VEHICULAR INTEGRATION FOR C4ISR/EW INTEROPERABILITY (VICTORY) STANDARD SPECIFICATIONS

Version V1.9
April 09, 2021

Released by the C5ISR Center MOSA Management Office
usarmy.apg.devcom-c5isr.mbx.mmo@army.mil

TABLE OF CONTENTS

1. Introduction 29

1.1 Background 29

1.2 Scope 29

1.3 Cybersecurity Policy 29

1.4 Maturity Levels 30

1.4.1 Preliminary 30

1.4.2 Informational 30

1.4.3 Experimental 30

1.4.4 Proposed Standard 31

1.4.5 Draft Standard 31

1.4.6 VICTORY Standard 31

2. Abbreviations and Acronyms 32

3. Document Conventions 45

3.1 Standards Key Words 45

3.2 Compliance 45

3.3 Maturity Level Markings 45

3.4 Applicability Markings 46

3.5 Usage of Specification Tags 46

3.6 Usage of VICTORY Type Tags 47

3.7 Usage of Notes 47

3.8 Usage of Mass Nouns 48

4. VICTORY System and Component Type Specifications 49

5. Common VICTORY Specifications 52

5.1 Common Data Interface Specifications <VT59901-V1.6> 52

5.2 Common Management Interface Specifications <VT59902-V1.9> 52

5.2.1 Data Interface Physical and Data Link Layer Specifications <VT59904-V1.9> 52

5.2.2 Copper Data Interface Physical and Data Link Layer Specifications <VT59905-V1.7> 53

5.2.3 Fiber Data Interface Physical and Data Link Layer Specifications <VT59906-V1.7> 53

5.2.4 Backplane Copper Data Interface Physical and Data Link Layer Specifications <VT59907-V1.9> 54

5.3 Authentication and Authorization Security Specifications <VT50400-V1.9> 54
5.4 Web Service Specifications <VT50420-V1.4> 54
5.5 Authorization Security Specifications <VT50430-V1.9> 55
5.6 Authentication Security Specifications <VT50440-V1.9> 55
6. Common VICTORY Interfaces 56
6.1 VDB Connector, Signaling, and Physical Media Interface <VT52303-V1.2> 56
6.2 Syslog-Based Health Publishing Interface <VT50103-V1.9> 56
6.3 SNMP-Based Health Publishing Interface <VT59903-V1.6> 56
6.4 Auto-Discovery Interface <VT59930-V1.6> 56
7. VICTORY Data Bus System Specifications 58
7.1 Shared Data Service Component Types 58
7.1.1 Time Synchronization Service <VT50000-V1.7> 58
7.1.2 Position Service <VT50100-V1.9> 59
7.1.3 Orientation Service <VT50200-V1.9> 60
7.1.4 Direction of Travel Service <VT50300-V1.9> 61
7.2 Access Control Component Types 61
7.2.1 Authentication Service <VT50401-V1.9> 61
7.2.2 Attribute Store Service <VT50800-V1.9> 62
7.2.3 Policy Decision Service <VT50500-V1.9> 63
7.2.4 Policy Enforcement Service <VT50600-V1.9> 63
7.2.5 Policy Store Service <VT50700-V1.9> 64
7.3 Infrastructure Component Types 64
7.3.1 Switch <VT50900-V1.9> 64
7.3.2 Router <VT51000-V1.9> 67
7.4 Information Assurance Component Types 70
7.4.1 Data Signing Service <VT52700-V1.9> 70
7.4.2 Signed Data Verification Service <VT52800-V1.9> 71
7.4.3 Data at Rest Encryptor <VT52400-V1.9> 71
7.4.4 Host-Based Antivirus <VT52500-V1.9> 72
7.4.5 Host-Based Firewall <VT51400-V1.9> 73
7.4.6 Inline Network Encryptor <VT51500-V1.6> 73
7.4.7 Intrusion Detection and Prevention System <VT51600-V1.9> 74

7.4.8 Network Firewall <VT51300-V1.9> 74
7.5 End Node Component Types 75
7.5.1 Shared Processing Unit <VT51900-V1.9> 75
7.5.2 Direct Attached Storage Encryption Management Interface <VT51908-V1.9> 77
7.5.3 Network Attached Storage <VT52000-V1.9> 77
7.5.4 Generic End Node <VT52100-V1.9> 77
7.6 System Management Component Types 80
7.6.1 VDB Management Service <VT52200-V1.9> 80
7.6.2 VICTORY Configuration Language Instance Document <VT55000-V1.9> 81
8. C4ISR/EW System and Component Type Specifications 84
8.1 Audio Communications Component Types 84
8.1.1 Streaming Audio Data Source <VT60001-V1.9> 84
8.1.2 Single-Instance Audio Data Source <VT60000-V1.9> 84
8.1.3 Intercom <VT60100-V1.9> 85
8.1.4 Legacy Radio <VT60200-V1.9> 86
8.2 Video and Imagery Situational Awareness Component Types 87
8.2.1 Video or Image Sensor <VT60300-V1.9> 87
8.2.2 Video or Image Encoder <VT60400-V1.9> 87
8.2.3 Camera Gimbal <VT60500-V1.9> 88
8.3 Recording and Logging Component Types 89
8.3.1 Mission Recorder <VT60700-V1.9> 89
8.4 Data Logger Component Types 90
8.4.1 Data Logger <VT62000-V1.9> 90
8.5 Extra-Vehicle Network Interface Component Types 91
8.5.1 Data Radio <VT60800-V1.9> 91
8.6 Electronic Warfare Component Types 93
8.6.1 Electronic Warfare Device <VT60900-V1.9> 93
8.7 C4ISR/EW Sensor Component Types 94
8.7.1 GPS Receiver <VT61000-V1.9> 94
8.8 C4ISR/EW System Component Types 96
8.8.1 Assured PNT <VT61600-V1.9> 96
9. Platform System Type Specifications 98

9.1 Automotive System Types 98
9.1.1 Automotive System <VT70000-V1.9> 98
9.1.2 Automotive System VICTORY Data Messages <VT70006-V1.9> 99
9.2 Power Distribution System Types 100
9.2.1 Power Distribution System <VT70100-V1.9> 100
9.3 Lethality System Types 101
9.3.1 Remote Weapon Station <VT70200-V1.9> 101
9.4 Logistics System Types 102
9.4.1 Vehicle Configuration Service <VT70800-V1.9> 102
9.5 Protection System Types 103
9.5.1 Threat Detection and Reporting Service <VT60600-V1.9> 103
9.5.2 Active Protection System <VT70900-V1.9> 103
9.6 Training System Types 105
9.6.1 Live Training Transformation Adapter <VT71000-V1.9> 105
9.7 MORA Component Type Specifications 106
9.7.1 Low Latency Switch <VT91010-V1.9> 106
9.7.2 MORA Device <VT92010-V1.9> 108
9.7.3 MORA Signal Resource Manager <VT92013-V1.9> 110
9.7.4 MORA Signal Port Manager <VT92016-V1.9> 112
9.8 MORA Data Messaging Specifications 115
9.8.1 MDM Protocol Specifications <VT96001-V1.9> 115
9.8.2 MDM Networking Specifications <VT96002-V1.9> 115
10. VDB Data Interface Specifications 116
10.1 VDB Physical Layer Protocols 116
10.1.1 10 Mbps Ethernet 116
10.1.2 100 Mbps Ethernet 116
10.1.3 Gigabit Ethernet 117
10.1.4 10 Gigabit Ethernet 117
10.1.5 Auto-Negotiation 117
10.2 Data Link Layer Protocols 118
10.2.1 Frame Structure <01013–20121113, Pro> 118
10.2.2 Media Access Control <01014–20121113, Pro> 118

10.2.3 Logical Link Control <01015–20120725, Pro> 118
10.2.4 Link Layer Switching <01016–20120725, Pro> 118
10.2.5 Link Layer Flow Control <01017–20121113, Pro> 118
10.3 Network Layer Protocols 118
10.3.1 Internet Protocol Version 4 Specification <01018–20120725, Pro> 118
10.3.2 Internet Protocol Version 6 Specification <01019–20110729, Exp> 119
10.3.3 IPv4 Datagram Transmission <01062–20120725, Pro> 119
10.3.4 IPv6 Datagram Transmission <01063–20120131, Exp> 119
10.3.5 Internet Control Message Protocol for IPv4 <01021–20120725, Pro> 119
10.3.6 Internet Control Message Protocol for IPv6 <01059–20120131, Exp> 119
10.3.7 Path MTU Discovery 120
10.3.8 IP Layer Security <01025–20180215, Pro> 120
10.4 Transport Layer Protocols 131
10.4.1 Transmission Control Protocol <01026–20120725, Pro> 131
10.4.2 User Datagram Protocol <01027–20120725, Pro> 131
10.4.3 Advanced Routing Concepts 131
10.4.4 VDB Subnetting Restrictions <01029–20120725, Pro> 132
10.4.5 Virtual Local Area Network (VLAN) <01077–20120131, Exp> 132
10.4.6 Multicast Routing Concept <Informational> 132
10.4.7 Multicast Group Subscription Management 132
10.4.8 Layer 2 Multicast Routing 133
10.4.9 Sparse Mode and Dense Mode 135
10.4.10 Multicast Data Addressing 136
10.5 Network Port Assignment <01035–20120725, Pro> 138
10.6 VDB Message Quality of Service 138
10.6.1 DiffServ <01037–20120725, Pro> 138
10.6.2 Quality of Service Levels <Informational> 139
10.6.3 DiffServ Code Point (DSCP) Assignments <01039–20120725, Pro> 140
10.6.4 QoS Mapping to Components <01036–20120725, Pro> 143
10.7 VDB Host Addressing 143
10.7.1 VDB Address Translation 143
10.7.2 VDB Host/Address Configuration 144

10.7.3 VDB Routing and Forwarding 145
10.7.4 VDB Node Identification 146
10.8 Service and Node Auto-Discovery 146
10.8.1 Background on Service and Node Auto-Discovery 147
10.8.2 Zero Configuration Networking <01044–20140630, Pro> 147
10.8.3 Auto-Discovery Security Concerns 151
10.9 VICTORY Connector Standards 152
10.9.1 MIL-DTL–38999 Series III 152
10.9.2 Other Ethernet Connectors <01081–20120131, Exp> 153
11. VDB Time Synchronization 154
11.1 VDB Time Synchronization Operational Concept <Informational> 154
11.2 VDB Time Synchronization Specifications 154
11.2.1 Network Time Protocol <02002–20121113, Pro> 155
11.2.2 Health Monitoring of Network Time Protocol <02004–20121113, Exp> 155
11.2.3 Network Time Protocol Management Specification <02005–20121113, Exp> 155
11.2.4 Precision Time Protocol <02003–20180215, Pro> 155
11.2.5 PTP End-to-End Transparent Clock Mode for Infrastructure Components <02008–20180215, Exp> 156
11.2.6 Health Monitoring of Precision Time Protocol <02006–20121113, Exp> 156
11.2.7 Precision Time Protocol Management Specification <02007–20121113, Exp> 156
12. Network Storage on the VDB 157
12.1 WebDAV <20028–20120731, Pro> 157
12.2 NFS <20029–20120731, Pro> 158
13. Access Control Framework 160
13.1 Authentication Methods <40001–20180215, Pro> 160
13.1.1 Password-Based Authentication for VICTORY Web Services 160
13.1.2 SAML-Based Authentication for VICTORY Web Services 160
13.1.3 PKI-Based Authentication for VICTORY Web Services 160
13.1.4 Biometrics-Based Authentication for VICTORY Web Services [Experimental] 161
13.2 Policy-Based Authorization <Informational> 161
13.3 Attribute-Based Access Control <Informational> 161
13.3.1 Attribute Store Service 164
13.3.2 Policy Decision Service 164

13.3.3 Policy Enforcement Service 164
13.3.4 Policy Store Service 164
13.4 Authorization Model <40003–20180215, Pro> 164
13.5 Authorization Architecture Entrypoints <40012–20210409, Pro> 164
13.6 Authentication Model <40013–20180215, Pro> 165
13.7 Authentication Entrypoints <40014–20210409, Pro> 165
13.8 Security Features <Informational> 166
13.9 Service Data Structures <Informational> 166
13.10 Information Flow <Informational> 167
14. Management Interfaces 169
14.1 Core Management Technologies 169
14.1.1 SOAP 169
14.1.2 Simple Network Management Protocol 171
14.1.3 Syslog Protocol <10005–20120725, Pro> 179
14.2 Common Management Parameters 179
14.2.1 Version Information Management Parameters <10007–20210409, Pro> 179
14.2.2 Version Information Complex Types <10031–20210409, Pro> 180
14.2.3 VICTORY Management Interface Types <10032–20210409, Pro> 181
14.2.4 VICTORY SOAP Component Types <10033–20210409, Pro> 184
14.2.5 VICTORY Data Management Parameters <10008–20210409, Pro> 196
14.2.6 Device Information Management Parameters <10009–20210409, Pro> 197
14.2.7 Health Publishing Management Parameters 198
14.2.8 Management Interface Error Types <10013–20210409, Pro> 202
14.3 Health Management 204
14.3.1 Health Management Overview <Informational> 204
14.3.2 Components of VICTORY Health Management <Informational> 204
14.3.3 Health Management Scenarios <Informational> 206
14.3.4 Health Level Management Parameters <10030–20210409, Pro> 207
14.3.5 Health Levels for End Users <10022–20131031, Pro> 207
14.3.6 Health Publishing Interfaces 208
14.3.7 Maintaining Fault and Status Uniqueness <Informational> 214
14.4 Notification Interfaces 214

14.4.1 Syslog-Based Asynchronous Notification Publishing Interface <10012–20120430, Exp> 214
14.4.2 Syslog-Based Asynchronous Notification Message Example <Informational> 215
14.5 Configuration Management Parameters for the VDB Management Service <10029–20210409, Pro> 215
14.6 Generic Component Status Interface 217
14.6.1 Generic Component Status Interface Information Content and API 217
14.7 Security for Management Interfaces 220
14.7.1 Web Services Security (WS-Security) 220
14.7.2 Data Confidentiality & Integrity for Management Interfaces <10018–20180215, Pro> 221
14.7.3 Data Authentication for Management Interfaces <10019–20150331, Pro> 221
15. Security Event Handling 222
15.1 Security Event Log Services 222
15.1.2 Security Event Log Data Specifications 222
15.1.3 Security Event Log Management Specification <14027–20131031, Exp> 225
15.1.4 Syslog Protocol Security Event Message Example <Informational> 228
15.1.5 Security Event Log Storage and Protection <10026–20130430, Exp> 229
15.1.6 Log Analysis Using PRI and SD-PARAM Values <10027–20130430, Exp> 229
15.1.7 Log Analysis Example <Informational> 229
15.2 Incident Response 232
15.2.1 Using VICTORY Management Interfaces for Incident Response <Informational> 232
16. VDB Component Types 236
16.1 Infrastructure Component Types 236
16.1.1 Generic Infrastructure Component Management Interface <12006–20180215, Pro> 236
16.1.2 Switch 237
16.1.3 Router 239
16.2 Information Assurance Component Types 239
16.2.1 Data Signing and Signed Data Verification Services 239
16.2.2 Data at Rest Encryptor 242
16.2.3 Host-Based Antivirus 244
16.2.4 Host-Based Firewall 245
16.2.5 Intrusion Detection and Prevention System 249
16.2.6 Network Firewall 250
16.3 Access Control Component Types 251

16.3.1 Attribute Store Service 251
16.3.2 Policy Decision Service 254
16.3.3 Policy Enforcement Service 256
16.3.4 Policy Store Service 259
16.3.5 Authentication Service 262
16.4 End Node Component Types 267
16.4.1 Generic End Node 267
16.4.2 Shared Processing Unit 269
16.4.3 Network Adapter 274
17. Application Layer Messaging 275
17.1 Application Layer Data Encoding <20001–20120725, Pro> 275
17.2 Application Layer Message Encapsulation <20002–20150331, Pro> 275
17.2.1 Field Byte Ordering 276
17.2.2 VDM Preamble (32 bits) 276
17.2.3 VDM Version Field (16 bits) 277
17.2.4 VDM Timestamp (80 bits) 277
17.2.5 VDM Sequence Number (32 bits) 277
17.2.6 VDM Interface ID (32 bits) 277
17.2.7 VDM Message Size (16 bits) 277
17.2.8 Number of Data Messages (16 bits) 277
17.2.9 VDM Data Index (VDM Encapsulation) 277
17.3 VICTORY Data Types <20003–20210409, Pro> 278
17.3.1 Heading 278
17.3.2 Speed 278
17.3.3 Rate of Climb 278
17.3.4 Roll 278
17.3.5 Pitch 278
17.3.6 Yaw 278
17.3.7 Latitude 279
17.3.8 Longitude 279
17.3.9 Altitude 279
17.3.10 MGRS 279

17.3.11 Azimuth 279
17.3.12 Elevation 279
17.3.13 Range 280
17.3.14 True Azimuth 280
17.3.15 Absolute Elevation 280
17.3.16 Relative Cartesian Coordinates 280
17.3.17 String Bounds 280
17.3.18 URI Bounds 280
17.4 Timestamp Format for Application Layer Data <20004–20140630, Pro> 281
17.5 Conversion from Textual to Binary Timestamps <20005–20120725, Pro> 282
17.6 Conversion from Binary to Textual Timestamps <20006–20140630, Pro> 282
18. C4ISR Systems Interfaces 284
18.1 VDB Shared Service Interfaces 284
18.1.1 Direction of Travel Interfaces 284
18.1.2 Orientation Interfaces 288
18.1.3 Position Interfaces 291
18.2 VICTORY Frame of Reference <20010–20110729, Exp> 295
18.2.1 Frame of Reference Tree Paradigm 295
18.2.2 Platform Frame of Reference 295
18.2.3 Orientation Coordinate System 295
18.2.4 Connector Frame of Reference 296
18.2.5 Sensor Frame of Reference 296
18.2.6 Configuration File 296
18.2.7 Position and Orientation Calculation 296
18.3 Situational Awareness Interfaces 297
18.4 Threat Detection and Reporting Interfaces 297
18.4.1 Threat Detection and Reporting Data Specification <20011–20210409, Pro> 297
18.4.2 Threat Detection and Reporting Management Parameters <15004–20210409, Pro> 300
18.4.3 Threat Detection and Reporting Complex Types <20015–20210409, Pro> 300
18.5 Mission Recorder Interfaces <Informational> 306
18.5.1 Mission Recorder Core Functionality <03001–20131031, Pro> 306
18.5.2 Mission Recorder Device Management Parameters <15015–20210409, Pro> 306

18.5.3 Mission Recorder Interface Complex Types <20027–20210409, Pro>..... 311
18.6 Data Logger 312
18.6.1 Data Logger Data Specification <20076–20210409, Pro>..... 312
18.6.2 Data Logger Management Parameters <15021–20210409, Pro>..... 314
18.6.3 Data Logger Complex Types <20077–20210409, Pro> 316
18.6.4 Data Logger FTP Specification <15022–20180215, Pro> 318
18.6.5 Data Logger Pcap Log File Format <20078–20150331, Pro> 319
18.7 Audio Communications Interfaces 320
18.7.1 Streaming Audio Data Specification <20016–20131031, Pro> 320
18.7.2 Streaming Audio Data Management Parameters <15007–20210409, Pro>..... 321
18.7.3 Single-Instance Audio Data Specification <20067–20131031, Pro>..... 321
18.7.4 Single-Instance Audio Data Management Parameters <15018–20210409, Pro> 321
18.7.5 Single-Instance Audio Sink Interface <Informational>..... 322
18.7.6 Legacy Radio Device Management Parameters <15025–20210409, Pro> 323
18.7.7 Legacy Radio Complex Types <15026–20210409, Pro>..... 326
18.7.8 Intercom Interfaces 327
18.8 Video and Imagery Situational Awareness Interfaces..... 330
18.8.1 VISA Data Interfaces..... 330
18.8.2 VISA Management Interfaces..... 334
18.8.3 VISA Complex Types <15011–20210409, Pro>..... 343
19. Platform Systems Interfaces 347
19.1 Power Distribution System Interfaces 347
19.1.1 Power Distribution System Data Specifications..... 347
19.1.2 Power Distribution System Data Management Parameters <16004–20210409, Pro>..... 358
19.1.3 Power Distribution System Device Management Parameters <16006–20210409, Pro> 365
19.1.4 Power Distribution System Interface Complex Types <20026–20210409, Pro>..... 366
19.2 Automotive System Interfaces..... 371
19.2.1 Automotive System Data Specifications 371
19.2.2 Automotive System Data Specification – V1.6 Extension 394
19.2.3 Automotive System Management Parameters <16001–20210409, Pro>..... 409
19.2.4 Automotive System Management Parameters – V1.6 <16005–20210409, Pro>..... 412
19.2.5 Automotive System Interface Complex Types <20022–20210409, Pro> 414

19.2.6 Automotive System Interface Simple Types <20066–20150331, Pro> 445

19.3 Lethality Systems Interfaces 453

19.3.1 Remote Weapon Station Interfaces 453

19.3.2 RWS Data Specification <20019–20210409, Pro> 454

19.3.3 RWS Device Management Parameters <16002–20210409, Pro> 455

19.3.4 RWS Interface Complex Types <20020–20210409, Pro> 455

19.4 Logistics Systems Interfaces 457

19.4.1 Vehicle Configuration Service 457

19.5 Training Systems Interfaces 466

19.5.1 Live Training Transformation (LT2) Adapter 466

20. VICTORY Configuration Language 472

20.1 VCL Schema <30044–20210409, Pro> 472

20.2 Configuration Type <30048–20131031, Pro> 473

20.3 VCL Instance Document Requirements <30045–20131031, Pro> 473

20.3.1 Stubbed Out VCL Instance Document Example 473

20.4 Tree View Requirements <30046–20131031, Pro> 474

20.5 Common Component Configuration Description Content <30047–20210409, Pro> 474

20.6 Position Service Configuration Description <30005–20210409, Pro> 479

20.6.1 Position Service Configuration Description Tree View 479

20.6.2 Position Service-Specific Configuration Description Content 481

20.6.3 Position Service Configuration Description Example 482

20.7 Orientation Service Configuration Description <30006–20210409, Pro> 485

20.7.1 Orientation Service Configuration Description Tree View 485

20.7.2 Orientation Service-Specific Configuration Description Content 486

20.7.3 Orientation Service Configuration Description Example 487

20.8 Direction of Travel Service Configuration Description <30007–20210409, Pro> 490

20.8.1 Direction of Travel Service Configuration Description Tree View 490

20.8.2 Direction of Travel Service-Specific Configuration Description Content 491

20.8.3 Direction of Travel Service Configuration Description Example 492

20.9 Streaming Audio Data Source Configuration Description <30008–20210409, Pro> 495

20.9.1 Streaming Audio Data Source Configuration Description Tree View 495

20.9.2 Streaming Audio Data Source-Specific Configuration Description Content 496

20.9.3 Streaming Audio Data Source Configuration Description Example 497

20.10 Single-Instance Audio Data Source Configuration Description <30009–20210409, Pro> 499

20.10.1 Single-Instance Audio Data Source Configuration Description Tree View 499

20.10.2 Single-Instance Audio Data Source-Specific Configuration Description Content 500

20.10.3 Single-Instance Audio Data Source Configuration Description Example 501

20.11 Intercom Configuration Description <30010–20210409, Pro> 503

20.11.1 Intercom Configuration Description Tree View 503

20.11.2 Intercom-Specific Configuration Description Content 504

20.11.3 Intercom Configuration Description Example 507

20.12 Legacy Radio Configuration Description <30053–20210409, Pro> 510

20.12.1 Legacy Radio Configuration Description Tree View 510

20.12.2 Legacy Radio-Specific Configuration Description Content 511

20.12.3 Legacy Radio Configuration Description Example 514

20.13 Video or Image Sensor Configuration Description <30012–20210409, Pro> 517

20.13.1 Video or Image Sensor Configuration Description Tree View 517

20.13.2 Video or Image Sensor-Specific Configuration Description Content 518

20.13.3 Video or Image Sensor Configuration Description Example 519

20.14 Video or Image Encoder Configuration Description <30013–20210409, Pro> 522

20.14.1 Video or Image Encoder Configuration Description Tree View 522

20.14.2 Video or Image Encoder-Specific Configuration Description Content 523

20.14.3 Video or Image Encoder Configuration Description Example 525

20.15 Camera Gimbal Configuration Description <30014–20210409, Pro> 528

20.15.1 Camera Gimbal Configuration Description Tree View 528

20.15.2 Camera Gimbal-Specific Configuration Description Content 529

20.15.3 Camera Gimbal Configuration Description Example 532

20.16 Threat Detection and Reporting Service Configuration Description <30015–20210409, Pro> 536

20.16.1 Threat Detection and Reporting Service Configuration Description Tree View 536

20.16.2 Threat Detection and Reporting Service-Specific Configuration Description Content 537

20.16.3 Threat Detection and Reporting Service Configuration Description Example 538

20.17 Mission Recorder Configuration Description <30016–20210409, Pro> 541

20.17.1 Mission Recorder Configuration Description Tree View 541

20.17.2 Mission Recorder-Specific Configuration Description Content 542

20.17.3 Mission Recorder Configuration Description Example 545

20.18 Data Logger Configuration Description <30049–20210409, Pro> 548

20.18.1 Data Logger Configuration Description Tree View 548

20.18.2 Data Logger-Specific Configuration Description Content 549

20.18.3 Data Logger Configuration Description Example 552

20.19 Automotive System Configuration Description <30019–20210409, Pro> 555

20.19.1 Automotive System Configuration Description Tree View 555

20.19.2 Automotive System-Specific Configuration Description Content 556

20.19.3 Automotive System Configuration Description Example 557

20.20 Power Distribution System Configuration Description <30020–20210409, Pro> 564

20.20.1 Power Distribution System Configuration Description Tree View 564

20.20.2 Power Distribution System-Specific Configuration Description Content 565

20.20.3 Power Distribution System Configuration Description Example 568

20.21 Remote Weapon Station Configuration Description <30021–20210409, Pro> 572

20.21.1 Remote Weapon Station Configuration Description Tree View 572

20.21.2 Remote Weapon Station-Specific Configuration Description Content 573

20.21.3 Remote Weapon Station Configuration Description Example 574

20.22 Vehicle Configuration Service Configuration Description <30050–20210409, Pro> 577

20.22.1 Vehicle Configuration Service Configuration Description Tree View 577

20.22.2 Vehicle Configuration Service-Specific Configuration Description Content 578

20.22.3 Vehicle Configuration Service Configuration Description Example 579

20.23 Live Training Transformation Adapter Configuration Description <30052–20210409, Pro> 581

20.23.1 Live Training Transformation Adapter Configuration Description Tree View 581

20.23.2 Live Training Transformation Adapter-Specific Configuration Description Content 582

20.23.3 Live Training Transformation Adapter Configuration Description Example 583

20.24 Attribute Store Service Configuration Description <30022–20210409, Pro> 586

20.24.1 Attribute Store Service Configuration Description Tree View 586

20.24.2 Attribute Store Service-Specific Configuration Description Content 586

20.24.3 Attribute Store Service Configuration Description Example 587

20.25 Policy Decision Service Configuration Description <30023–20210409, Pro> 589

20.25.1 Policy Decision Service Configuration Description Tree View 589

20.25.2 Policy Decision Service-Specific Configuration Description Content 589

20.25.3 Policy Decision Service Configuration Description Example 591
20.26 Policy Enforcement Service Configuration Description <30024–20210409, Pro> 593
20.26.1 Policy Enforcement Service Configuration Description Tree View 593
20.26.2 Policy Enforcement Service-Specific Configuration Description Content 593
20.26.3 Policy Enforcement Service Configuration Description Example 595
20.27 Policy Store Service Configuration Description <30025–20210409, Pro> 597
20.27.1 Policy Store Service Configuration Description Tree View 597
20.27.2 Policy Store Service-Specific Configuration Description Content 597
20.27.3 Policy Store Service Configuration Description Example 598
20.28 Authentication Service Configuration Description <30026–20210409, Pro> 600
20.28.1 Authentication Service Configuration Description Tree View 600
20.28.2 Authentication Service-Specific Configuration Description Content 600
20.28.3 Authentication Service Configuration Description Example 601
20.29 Host-Based Antivirus Configuration Description <30027–20210409, Pro> 603
20.29.1 Host-Based Antivirus Configuration Description Tree View 603
20.29.2 Host-Based Antivirus-Specific Configuration Description Content 604
20.29.3 Host-Based Antivirus Configuration Description Example 605
20.30 Host-Based Firewall Configuration Description <30028–20210409, Pro> 607
20.30.1 Host-Based Firewall Configuration Description Tree View 607
20.30.2 Host-Based Firewall-Specific Configuration Description Content 608
20.30.3 Host-Based Firewall Configuration Description Example 610
20.31 Network Firewall Configuration Description <30029–20210409, Pro> 613
20.31.1 Network Firewall Configuration Description Tree View 613
20.31.2 Network Firewall-Specific Configuration Description Content 614
20.31.3 Network Firewall Configuration Description Example 615
20.32 Intrusion Detection and Prevention System Configuration Description <30030–20210409, Pro> 618
20.32.1 Intrusion Detection and Prevention System Configuration Description Tree View 618
20.32.2 Intrusion Detection and Prevention System-Specific Configuration Description Content 619
20.32.3 Intrusion Detection and Prevention System Configuration Description Example 620
20.33 Data at Rest Encryptor Configuration Description <30031–20210409, Pro> 622
20.33.1 Data at Rest Encryptor Configuration Description Tree View 622
20.33.2 Data at Rest Encryptor-Specific Configuration Description Content 623

20.33.3 Data at Rest Encryptor Configuration Description Example 624
20.34 Data Signing Service Configuration Description <30032–20210409, Pro> 626
20.34.1 Data Signing Service Configuration Description Tree View 626
20.34.2 Data Signing Service-Specific Configuration Description Content 627
20.34.3 Data Signing Service Configuration Description Example 628
20.35 Signed Data Verification Service Configuration Description <30033–20210409, Pro> 630
20.35.1 Signed Data Verification Service Configuration Description Tree View 630
20.35.2 Signed Data Verification Service-Specific Configuration Description Content 631
20.35.3 Signed Data Verification Service Configuration Description Example 632
20.36 Generic End Node Configuration Description <30040–20210409, Pro> 634
20.36.1 Generic End Node Configuration Description Tree View 634
20.36.2 Generic End Node-Specific Configuration Description Content 635
20.36.3 Generic End Node Configuration Description Example 636
20.37 Network Attached Storage Configuration Description <30041–20210409, Pro> 639
20.37.1 Network Attached Storage Configuration Description Tree View 639
20.37.2 Network Attached Storage-Specific Configuration Description Content 639
20.37.3 Network Attached Storage Configuration Description Example 640
20.38 Shared Processing Unit Configuration Description <30042–20210409, Pro> 641
20.38.1 Shared Processing Unit Configuration Description Tree View 641
20.38.2 Shared Processing Unit-Specific Configuration Description Content 642
20.38.3 Shared Processing Unit Configuration Description Example 644
20.39 VDB Management Service Configuration Description <30043–20210409, Pro> 648
20.39.1 VDB Management Service Configuration Description Tree View 648
20.39.2 VDB Management Service-Specific Configuration Description Content 649
20.39.3 VDB Management Service Configuration Description Example 650
20.40 Configuration Management Parameters <10028–20210409, Pro> 652
APPENDIX A Zeroconf Announcement Strings Used By VICTORY 654
APPENDIX B VICTORY WSDLs 661
APPENDIX C VICTORY Schemas 663
APPENDIX D Leap Seconds Table From NIST 665
APPENDIX E VICTORY Specification Tag Cross-Reference 666
APPENDIX F VDM Data Type Values 683

APPENDIX G Video Encoder Pixel Formats 687

LIST OF TABLES

Table 1. Abbreviations and Acronyms 32

Table 2. Specification Groups 46

Table 3. Document Release Date To Document Version Mapping 47

Table 4. Common Data Interface Specifications <VT59901-V1.6> 52

Table 5. Common Management Interface Specifications <VT59902-V1.9> 52

Table 6. Data Interface Physical and Data Link Layer Specifications <VT59904-V1.9> 53

Table 7. Copper Data Interface Physical and Data Link Layer Specifications <VT59905-V1.7> 53

Table 8. Fiber Data Interface Physical and Data Link Layer Specifications <VT59906-V1.7> 53

Table 9. Backplane Copper Data Interface Physical and Data Link Layer Specifications <VT59907-V1.9> 54

Table 10. Authentication and Authorization Security Specifications <VT50400-V1.9> 54

Table 11. Web Service Specifications <VT50420-V1.4> 54

Table 12. Authorization Security Specifications <VT50430-V1.9> 55

Table 13. Authentication Security Specifications <VT50440-V1.9> 55

Table 14. VDB Connector, Signaling, and Physical Media Interface <VT52303-V1.2> 56

Table 15. Syslog-Based Health Publishing Interface <VT50103-V1.9> 56

Table 16. SNMP-Based Health Publishing Interface <VT59903-V1.6> 56

Table 17. Auto-Discovery Interface <VT59930-V1.6> 57

Table 18. Time Synchronization Service <VT50000-V1.7> 58

Table 19. Time Synchronization Interface <VT50001-V1.7> 58

Table 20. Network Time Protocol Specifications <VT50001.T1-V1.4> 58

Table 21. Precision Time Protocol Specifications <VT50001.T2-V1.7> 59

Table 22. Position Service <VT50100-V1.9> 59

Table 23. Position Data Interface <VT50101-V1.6> 59

Table 24. Position Management Interface <VT50102-V1.9> 59

Table 25. Orientation Service <VT50200-V1.9> 60

Table 26. Orientation Data Interface <VT50201-V1.6> 60

Table 27. Orientation Management Interface <VT50202-V1.9> 60

Table 28. Direction of Travel Service <VT50300-V1.9> 61

Table 29. Direction of Travel Data Interface <VT50301-V1.6> 61

Table 30. Direction of Travel Management Interface <VT50302-V1.9> 61

Table 31. Authentication Service <VT50401-V1.9> 62

Table 32. Authentication Service Management Interface <VT50402-V1.9> 62

Table 33. Attribute Store Service <VT50800-V1.9> 62

Table 34. Attribute Store Management Interface <VT50802-V1.9> 62

Table 35. Policy Decision Service <VT50500-V1.9> 63

Table 36. Policy Decision Management Interface <VT50502-V1.9> 63

Table 37. Policy Enforcement Service <VT50600-V1.9> 63

Table 38. Policy Enforcement Management Interface <VT50602-V1.9> 64

Table 39. Policy Store Service <VT50700-V1.9> 64

Table 40. Policy Store Management Interface <VT50702-V1.9> 64

Table 41. Switch <VT50900-V1.9> 65

Table 42. Switch Data Interface <VT50901-V1.9> 65

Table 43. Switch IPv6 Data Interface <VT50903-V1.9> 65

Table 44. Switch Data Interface Network Layer and Higher Specifications <VT50901.T4-V1.7> 65

Table 45. Switch IPv6 Data Interface Network Layer and Higher Specifications <VT50901.T7-V1.7> 66

Table 46. Switch Data Interface IPv4-Specific Specifications <VT50901.T5-V1.2> 66

Table 47. Switch Data Interface IPv6-Specific Specifications <VT50901.T6-V1.6> 67

Table 48. Switch Management Interface <VT50902-V1.7> 67

Table 49. Router <VT51000-V1.9> 67

Table 50. Router Data Interface <VT51001-V1.9> 67

Table 51. Router IPv6 Data Interface <VT51003-V1.9> 68

Table 52. Router Data Interface Network Layer and Higher Specifications <VT51001.T4-V1.7> 68

Table 53. Router IPv6 Data Interface Network Layer and Higher Specifications <VT51001.T8-V1.7> 68

Table 54. Router Data Interface IPv4-Specific Specifications <VT51001.T5-V1.7> 69

Table 55. Router Data Interface IPv6-Specific Specifications <VT51001.T6-V1.7> 70

Table 56. Router Management Interface <VT51002-V1.7> 70

Table 57. Data Signing Service <VT52700-V1.9> 70

Table 58. Data Signing Service Management Interface <VT52701-V1.9> 71

Table 59. Signed Data Verification Service <VT52800-V1.9> 71

Table 60. Signed Data Verification Service Management Interface <VT52801-V1.9> 71

Table 61. Data at Rest Encryptor <VT52400-V1.9> 72

Table 62. DAR Encryptor Management Interface <VT52402-V1.9> 72

Table 63. Host-Based Antivirus <VT52500-V1.9> 72

Table 64. Host-Based Antivirus Management Interface <VT52502-V1.9> 72

Table 65. Host-Based Firewall <VT51400-V1.9> 73

Table 66. Host-Based Firewall Management Interface <VT51402-V1.9> 73

Table 67. Inline Network Encryptor <VT51500-V1.6> 73

Table 68. INE Management Interface <VT51502-V1.6> 74

Table 69. Intrusion Detection and Prevention System <VT51600-V1.9> 74

Table 70. IDPS Management Interface <VT51602-V1.9> 74

Table 71. Network Firewall <VT51300-V1.9> 74

Table 72. Network Firewall Management Interface <VT51302-V1.9> 75

Table 73. Shared Processing Unit <VT51900-V1.9> 75

Table 74. Shared Processing Unit Management Interface <VT51902-V1.9> 75

Table 75. Shared Processing Unit Platform API Headless Profile <VT51912-V1.9> 76

Table 76. Shared Processing Unit Platform API Display Profile <VT51913-V1.9> 76

Table 77. Shared Processing Unit Low-Level Platform API <VT51909-V1.9> 76

Table 78. Shared Processing Unit Mid-Level Platform API <VT51910-V1.8> 77

Table 79. Direct Attached Storage Encryption Management Interface <VT51908-V1.9> 77

Table 80. Network Attached Storage <VT52000-V1.9> 77

Table 81. Network Attached Storage Interface <VT52001-V1.9> 77

Table 82. Generic End Node <VT52100-V1.9> 78

Table 83. Generic End Node Data Interface <VT52101-V1.9> 78

Table 84. Generic End Node IPv6 Data Interface <VT52103-V1.9> 78

Table 85. Generic End Node Data Interface IPv4-Specific Specifications <VT52101.T1-V1.7> 79

Table 86. Generic End Node Data Interface IPv6-Specific Specifications <VT52101.T2-V1.7> 79

Table 87. Generic End Node Management Interface <VT52102-V1.9> 80

Table 88. VDB Management Service <VT52200-V1.9> 80

Table 89. VDB System Management Interface <VT52201-V1.9> 81

Table 90. VICTORY Configuration Language Instance Document <VT55000-V1.9> 81

Table 91. VDB Configuration Description <VT55100-V1.9> 82

Table 92. Information Assurance Configuration Description <VT55200-V1.9> 82

Table 93. C4ISR/EW System Configuration Description <VT55300-V1.9> 82

Table 94. Platform System Configuration Description <VT55400-V1.9> 83

Table 95. Streaming Audio Data Source <VT60001-V1.9> 84

Table 96. Streaming Audio Data Interface <VT60004-V1.6> 84

Table 97. Streaming Audio Management Interface <VT60005-V1.9> 84

Table 98. Single-Instance Audio Data Source <VT60000-V1.9> 85

Table 99. Single-Instance Audio Data Interface <VT60002-V1.6> 85

Table 100. Single-Instance Audio Management Interface <VT60003-V1.9> 85

Table 101. Intercom <VT60100-V1.9> 85

Table 102. Intercom Management Interface <VT60101-V1.9> 86

Table 103. Legacy Radio <VT60200-V1.9> 86

Table 104. Legacy Radio Management Interface <VT60202-V1.9> 86

Table 105. Video or Image Sensor <VT60300-V1.9> 87

Table 106. Video or Image Sensor Management Interface <VT60301-V1.9> 87

Table 107. Video or Image Encoder <VT60400-V1.9> 88

Table 108. Video Streaming Data Interface <VT60401-V1.6> 88

Table 109. Video Encoder Management Interface <VT60403-V1.9> 88

Table 110. Camera Gimbal <VT60500-V1.9> 88

Table 111. Camera Gimbal Data Interface <VT60501-V1.7> 89

Table 112. Camera Gimbal Management Interface <VT60502-V1.9> 89

Table 113. Mission Recorder <VT60700-V1.9> 90

Table 114. Mission Recorder Device Management Interface <VT60702-V1.9> 90

Table 115. Data Logger <VT62000-V1.9> 90

Table 116. Data Logger Data Interface <VT62001-V1.9> 91

Table 117. Data Logger Management Interface <VT62002-V1.9> 91

Table 118. Data Radio <VT60800-V1.9> 91

Table 119. Data Radio Data Interface <VT60801-V1.6> 91

Table 120. Data Radio Data Interface Physical and Data Link Layer Specifications <VT60803-V1.6> 92

Table 121. Data Radio Data Interface Network Layer and Higher Specifications <VT60804-V1.6> 92

Table 122. Data Radio Management Interface <VT60802-V1.9> 93

Table 123. Electronic Warfare Device <VT60900-V1.9> 93

Table 124. Electronic Warfare Data Interface <VT60901-V1.9> 93

Table 125. Electronic Warfare Management Interface <VT60902-V1.9> 93

Table 126. Electronic Warfare VICTORY Data Messages <VT60904-V1.9> 94

Table 127. GPS Receiver <VT61000-V1.9> 94

Table 128. GPS Receiver Data Interface <VT61001-V1.9> 95

Table 129. GPS Receiver Management Interface <VT61002-V1.9> 95

Table 130. GPS Receiver VICTORY Data Messages <VT61003-V1.9> 95

Table 131. GPS Receiver Status VICTORY Data Messages <VT61004-V1.9> 96

Table 132. GPS Receiver Satellite Health Status VICTORY Data Messages <VT61005-V1.9> 96

Table 133. Assured PNT <VT61600-V1.9> 96

Table 134. Assured PNT Data Interface <VT61601-V1.9> 97

Table 135. Assured PNT Management Interface <VT61602-V1.9> 97

Table 136. Assured PNT VICTORY Data Messages <VT61603-V1.9> 97

Table 137. Automotive System <VT70000-V1.9> 98

Table 138. Automotive System Data Interface <VT70001-V1.9> 98

Table 139. Automotive System Management Interface <VT70002-V1.9> 98

Table 140. Automotive System VICTORY Data Messages <VT70006-V1.9> 99

Table 141. Power Distribution System <VT70100-V1.9> 100

Table 142. Power Distribution System Data Interface <VT70101-V1.9> 100

Table 143. Power Distribution System Management Interface <VT70102-V1.9> 100

Table 144. Power Distribution System VICTORY Data Messages <VT70103-V1.9> 101

Table 145. Remote Weapon Station <VT70200-V1.9> 101

Table 146. Remote Weapon Station Data Interface <VT70201-V1.9> 101

Table 147. Remote Weapon Station Management Interface <VT70202-V1.9> 102

Table 148. Vehicle Configuration Service <VT70800-V1.9> 102

Table 149. Vehicle Configuration Service Management Interface <VT70801-V1.9> 102

Table 150. Threat Detection and Reporting Service <VT60600-V1.9> 103

Table 151. Threat Detection and Reporting Data Interface <VT60601-V1.9> 103

Table 152. Threat Detection and Reporting Management Interface <VT60602-V1.9> 103

Table 153. Active Protection System <VT70900-V1.9> 104

Table 154. Active Protection System Data Interface <VT70901-V1.9> 104

Table 155. Active Protection System VICTORY Data Messages <VT70903-V1.9> 104

Table 156. Active Protection System Management Interface <VT70902-V1.9> 104

Table 157. Live Training Transformation Adapter <VT71000-V1.9> 105

Table 158. Live Training Transformation Adapter Data Interface <VT71001-V1.7> 105

Table 159. Live Training Transformation Adapter Management Interface <VT71002-V1.9> 105

Table 160. Low Latency Switch <VT91010-V1.9> 106

Table 161. Low Latency Switch Data Interface <VT91011-V1.9> 106

Table 162. Low Latency Switch Data Interface Physical and Data-Link Layer Specifications <VT91012-V1.9> 107

Table 163. Low Latency Copper Data Physical and Data-Link Layer Specifications <VT91013-V1.9> 107

Table 164. Low Latency Fiber Data Physical and Data-Link Layer Specifications <VT91014-V1.9> 108

Table 165. MORA Device <VT92010-V1.9> 109

Table 166. MORA Device Management Interface <VT92011-V1.9> 109

Table 167. MORA Device Operations Interface <VT90016-V1.9> 110

Table 168. MORA Signal Resource Manager <VT92013-V1.9> 110

Table 169. MORA Signal Resource Management Interface <VT92014-V1.9> 111

Table 170. MORA Signal Resource Manager Operations Interface <VT90017-V1.9> 111

Table 171. MORA Signal Port Manager <VT92016-V1.9> 112

Table 172. RF Layer Analog Data Interface <VT90011-V1.9> 112

Table 173. RF Layer Command Interface <VT90012-V1.9> 113

Table 174. RF Layer Context Interface <VT90013-V1.9> 113

Table 175. RF Layer Digital Data Interface <VT90014-V1.9> 114

Table 176. RF Layer Operations Interface <VT90015-V1.9> 114

Table 177. MDM Protocol Specifications <VT96001-V1.9> 115

Table 178. MDM Networking Specifications <VT96002-V1.9> 115

Table 179. Security Services 120

Table 180. Cryptographic Algorithms – Unicast Messaging 127

Table 181. Cryptographic Algorithms – Multicast Messaging 127

Table 182. Other IPsec Configuration Considerations 129

Table 183. IP Layer Security Protocols Specified by VICTORY 130

Table 184. IGMP (IPv4) Multicast Routing Standards 133

Table 185. MLD (IPv6) Multicast Routing Standards 133

Table 186. Multicast Snooping Standards 134

Table 187. Platform Independent Multicast - Sparse Mode 135

Table 188. Platform Independent Multicast - Dense Mode 136

Table 189. Classless Inter-Domain Routing and Subnetting Standards for IPv4 136

Table 190. Subnetting Standards for IPv6 137

Table 191. IPv4 Address Groups 137

Table 192. IPv6 Address Groups 138

Table 193. Mapping of DSCPs 140

Table 194. IPv4 Address Translation Standards 143

Table 195. IPv6 Address Translation Standards 144

Table 196. Dynamic Host Configuration Standards for IPv4 144

Table 197. Dynamic Host Configuration Standards for IPv6 145

Table 198. VDB Node Identification for IPv4 146

Table 199. VDB Node Identification for IPv6 146

Table 200. MIL-DTL–38999 Ethernet Connector Pinout 152

Table 201. MIL-DTL–38999 Dual Ethernet Connector Pinout 153

Table 202. Precision Time Synchronization Protocol 155

Table 203. IEEE 1588–2008: Sections Relevant to End-to-end Transparent Clock Implementations 156

Table 204. Network Storage Features and Corresponding WebDAV Specifications 157

Table 205. Network Storage Features and Corresponding NFS Specifications 158

Table 206. Core SNMP Specifications 171

Table 207. SNMP Variable Core Compliance Policy Table 174

Table 208. SNMP Variable Core Units of Compliance (Groups) 175

Table 209. VICTORY Management Interface Types 181

Table 210. VICTORY SOAP Component Types 184

Table 211. Health Parameter Mapping 201

Table 212. Transport Layer Security Specifications 221

Table 213. Event Priority Scores (EPS) 230

Table 214. Event Source Scores (ESS) 230

Table 215. Event ID Scores (EIS) 231

Table 216. Event Type Scores (ETS) 232

Table 217. VDM Header 276

Table 218. dateTime Data Type 281

Table 219. Timestamp Examples 281

Table 220. Leap Seconds Conversion – Unix Time 282

Table 221. Leap Seconds Conversion – NTP 283

Table 222. Ambient Conditions VDM Content 371

Table 223. Dash Display VDM Content 373

Table 224. Direct Lamp Control Data 1 VDM Content 374

Table 225. Direct Lamp Control Data 2 VDM Content 376

Table 226. Distance VDM Content 377

Table 227. Electrical Power 1 VDM Content 378

Table 228. Electrical Power 2 VDM Content 380

Table 229. Electronic Transmission Controller 2 VDM Content 381

Table 230. Engine Fluid Level and Pressure 1 VDM Content 382

Table 231. Engine Hours and Revolutions VDM Content 384

Table 232. Engine Speed and Load Factor VDM Content 385

Table 233. Engine Temperature 1 VDM Content 387

Table 234. Engine Temperature 2 VDM Content 388

Table 235. Engine Valve VDM Content 390

Table 236. Fuel Consumption VDM Content 392

Table 237. Fuel Economy VDM Content 393

Table 238. Active Diagnostic Trouble Codes VDM Content 395

Table 239. Alternator Temperature VDM Content 396

Table 240. Battery Temperature VDM Content 398

Table 241. Cab Message 1 VDM Content 399

Table 242. Combination Vehicle Weight VDM Content 401

Table 243. Electronic Engine Controller 1 VDM Content 402

Table 244. Electronic Engine Controller 2 VDM Content 403

Table 245. Fan Drive 1 VDM Content 405

Table 246. Tire Condition VDM Content 406

Table 247. Torque/Speed Control 1 VDM Content 408

Table 248. Active Diagnostic Trouble Codes 414

Table 249. Alternator Temperature 416

Table 250. Ambient Conditions 417

Table 251. Battery Temperature 418

Table 252. Cab Message 1 418

Table 253. Combination Vehicle Weight 420

Table 254. Dash Display 421

Table 255. Direct Lamp Control Data 1 422

Table 256. Direct Lamp Control Data 2 425

Table 257. Distance 425

Table 258. Electrical Power 1 426

Table 259. Electrical Power 2 427

Table 260. Electronic Engine Controller 1 428

Table 261. Electronic Engine Controller 2 429

Table 262. Electronic Transmission Controller 2 432

Table 263. Engine Fluid Level and Pressure 1 433

Table 264. Engine Hours and Revolutions 434

Table 265. Engine Speed and Load Factor 435

Table 266. Engine Temperature 1 436

Table 267. Engine Temperature 2 437

Table 268. Engine Valve 438

Table 269. Fan Drive 1 440

Table 270. Fuel Consumption 441

Table 271. Fuel Economy 441

Table 272. Tire Condition 443

Table 273. Torque/Speed Control 1 444

Table 274. Valid Automotive Data Types 445

Table 275. Valid Automotive Data Types With V1.6 Extensions 446

Table 276. Valid TSC1 Control Purposes 447

Table 277. Valid TSC1 Transmission Rates 447

Table 278. Valid Override Control Mode Priorities 447

Table 279. Valid Engine Requested Speed Control Conditions 448

Table 280. Valid Engine Override Control Modes 448

Table 281. Valid Tire Pressure Threshold Detections 448

Table 282. Valid Engine Starter Modes 449

Table 283. Valid Engine Torque Modes 449

Table 284. Valid Fan Drive States 450

Table 285. Valid Failure Mode Values 451

Table 286. Valid Range Severity Values 452

Table 287. Valid Valve Operation Status Values 452

Table 288. Valid Ranged Data Error Values 453

Table 289. Valid Transmission Range Values 453

Table 290. LT2 Data Type Values 469

Table 291. LT2 Adapter Data Interface Message 469

Table 292. Zeroconf Announcement Strings 654

Table 293. Announcement String TXT Field Descriptions 659

Table 294. VICTORY WSDLs 661

Table 295. VICTORY Schemas 663

Table 296. VICTORY Type Tags 666

Table 297. VICTORY Specification Tags 668

Table 298. VDM Data Type Values 683

Table 299. Video Encoder Pixel Formats 687

1. INTRODUCTION

1.1 Background

The Vehicular Integration for C4ISR/EW Interoperability (VICTORY) initiative is developing a framework for integration of Command, Control, Communications, Computers, Intelligence, Surveillance, Reconnaissance / Electronic Warfare (C4ISR/EW) and other electronics equipment on U.S. Army ground vehicles. The framework is made up of: 1) an architecture, which defines common terminology, systems, components, and interfaces; 2) a set of standard specifications, which are technical specifications of the items identified in the architecture; and 3) a set of reference designs, which provide guidance for how the architecture and standards can be instantiated to create designs against various types of requirements and environments.

The architecture, documented in "VICTORY Architecture – Version A," identifies the systems, components, and interfaces, but does not provide technical details. The technical details are specified in the "VICTORY Standard Specifications" (this document) and are intended to be used by the system acquisition and science and technology communities as a citable reference in new procurements, modernization activities, and engineering change proposals. The VICTORY Standard Specifications do not specify system designs; specific hardware and software components used to implement the VICTORY standard specification; hardware configurations; or mapping of software to hardware. Designs must be developed to meet the operational, functional, and performance requirements of the platform or product. VICTORY provides example designs to aid the community in understanding the options for deploying the specifications. These examples are documented in VICTORY reference design documents.

Information on the VICTORY initiative, including detailed technical and programmatic information, can be found on the public site at www.victory-standards.org.

1.2 Scope

This document defines the VICTORY Standard Specifications, and as such, provides the technical details of the systems, components, and interfaces identified in the architecture document. Each of these items is described by a specification. The VICTORY specifications will document the technical details of interfaces from the physical (PHY) layer to the application (APP) layer of the open systems interconnection (OSI) layered network model. The goal of the specifications is to provide enough technical detail that independent implementations of systems and components will result in a high level of interoperability at all levels.

A specification is the documentation of the technical details, and a standard is a specification that has been deemed a "standard." The VICTORY specifications are being defined, documented, and matured through a working group process. In this process, working groups made up of experts from government, industry, and academic organizations produce specification documents. A support organization called the VICTORY Standards Support Office (VSSO) provides oversight and support for these working groups and "validates" the specifications recommended by the working groups by analysis, prototyping, and/or integration of systems. Through the process, the working groups and the VSSO mature the specifications until they are deemed mature enough to become standards. The maturity levels that a specification goes through to become a standard are described in section 1.4.

1.3 Cybersecurity Policy

The process of integrating C4ISR on U.S. Army ground vehicles and promoting interoperability between components would not be complete if the aspect of security is not taken into account for national security. Information Systems (IS) security is of critical importance; this includes securing and protecting data, components, systems, and the vehicle as a whole. NIST Special Publication 800–37 articulates that an appropriate operational security posture must be applied to maintained IS, and to manage national security cybersecurity/information assurance. While it is beyond the scope of VICTORY to provide a complete basis for security certification, all military systems must ultimately adhere to Department of Defense and Intelligence Community policies. These policies include, but are not limited to:

- AR 25–2: Information Assurance
- CNSSI 1253: Security Categorization and Control Selection for National Security Systems
- DoDI 8500.01: Cybersecurity
- DoDI 8510.01: Risk Management Framework (RMF) for DoD Information Technology (IT)

- FIPS PUB 140–2: Security Requirements for Cryptographic Modules
- NIST SP 800–53: Security and Privacy Controls for Federal Information Systems and Organizations
- DCID 6/3: Protecting Sensitive Compartmented Information Within Information Systems
- ICD 503: Information Technology Systems Security Risk Management, Certification and Accreditation

Developers should apply cybersecurity/information security early-on and through all phases of the systems engineering process. Ultimately, it is their responsibility to determine the sufficiency of security controls, requirements, and policies to achieve security certification.

1.4 Maturity Levels

Specifications that are intended to be included in VICTORY Standard Specifications evolve through a set of maturity levels known as the "standards track." These maturity levels – "VICTORY Proposed Standard," "VICTORY Draft Standard," and "VICTORY Standard" – are defined and discussed in this section. Every specification will be labeled with a maturity level.

Even after a specification has been adopted, further evolution often occurs based on experience and the recognition of new requirements. The nomenclature and procedures of VICTORY standardization provide for the revision of VICTORY Standard Specifications with updated content. A set of maturity levels is defined to cover these and other specifications that are not considered to be on the standards track.

Not every specification will be on the standards track. A specification may be intended for eventual standardization but may not yet be ready to enter the standards track. A specification may have been superseded by a more recent VICTORY specification, or have otherwise fallen into disuse or disfavor. Specifications that are not on the standards track are labeled with one of three "off-track" maturity levels: "Preliminary," "Informational," or "Experimental." The specifications bearing these labels are included in a VICTORY Standard Specifications release primarily for informational purposes, as they provide insight into capabilities likely to become part of the standards track in the near future.

Source: IETF RFC 2026: "The Internet Standards Process – Revision 3"

1.4.1 Preliminary

The "Preliminary" designation typically denotes a specification that is part of some research or development effort. Such a specification is published for the general information of the VICTORY technical community and, as an archival record of the work, subject only to editorial considerations and to verification that there has been adequate coordination with the standards process. A Preliminary specification may be the output of an organized VICTORY research effort, a VICTORY Working Group, or it may be an individual contribution.

Source: IETF RFC 2026

1.4.2 Informational

An "Informational" specification is published for the general information of the VICTORY community and does not represent a VICTORY community consensus or recommendation. The Informational designation is intended to provide for the timely publication of a very broad range of responsible informational documents from many sources, subject only to editorial considerations and to verification that there has been adequate coordination with the VICTORY standards process.

Specifications that have been prepared outside the VICTORY community and are not incorporated into the VICTORY standards process may be published as an Informational publication, with the permission of the owner.

Source: IETF RFC 2026

1.4.3 Experimental

An "Experimental" specification is often published to allow testing by the VICTORY community. While an Experimental specification is not on the VICTORY standards track, it is usually elevated to the "Proposed" maturity level of the standards track once sufficient validation testing has been done.

Source: IETF RFC 2026

1.4.4 Proposed Standard

The entry-level maturity for the standards track is “Proposed Standard.” A Proposed Standard specification is generally stable, has resolved known design choices, is believed to be well understood, and appears to enjoy enough community interest to be considered valuable. However, further experience might result in a change or even retraction of the specification before it advances.

Granting Proposed Standard status to a specification that materially affects the core VICTORY protocols or that specifies behavior that may have significant operational impact on VICTORY configurations may first require implementation and/or operational experience.

A Proposed Standard should have no known technical omissions with respect to the requirements placed upon it. Limited technical omissions may be allowed if this specification is considered to be useful and necessary (and timely) even with known technical omissions.

Source: IETF RFC 2026

1.4.5 Draft Standard

A specification from which at least two independent and interoperable implementations have been developed, and for which sufficient successful operational experience has been obtained, may be elevated to the “Draft Standard” level. For the purpose of this section, “interoperable” means being functionally equivalent or containing interchangeable components of the system or process in which they are used. If patented or otherwise controlled technology is required for implementation, the separate implementations must also have resulted from separate exercise of the licensing process. Elevation to Draft Standard is a major advance in status, indicating a strong belief that the specification is very mature.

The requirement for at least two independent and interoperable implementations applies to all of the options and features of the specification. In cases in which one or more options or features have not been demonstrated in at least two interoperable implementations, the specification may advance to the Draft Standard level only if those options or features are removed.

A Draft Standard must be well understood and known to be quite stable, both in its semantics and as a basis for developing an implementation. A Draft Standard may still require additional or more widespread field experience since it is possible for implementations based on Draft Standard specifications to demonstrate unforeseen behavior when subjected to large-scale use in production environments. Contrary to the typical meaning of “draft,” in most cases specifications are not matured past this level, and are maintained through a configuration management process at this level for extended periods of time (years).

Source: IETF RFC 2026

1.4.6 VICTORY Standard

A specification for which significant implementation and successful operational experience has been obtained may be elevated to the “VICTORY Standard” level. A VICTORY Standard is characterized by a high degree of technical maturity and by a generally held belief that the specified protocol or service provides significant benefit to the VICTORY community. A VICTORY Standard is normally considered to be a final specification, and changes are likely to be made only to solve specific problems encountered. In most circumstances, it is reasonable for vendors to deploy implementations of Proposed, Draft, and VICTORY Standards into a disruption-sensitive environment.

Source: IETF RFC 2026

2. ABBREVIATIONS AND ACRONYMS

Table 1. Abbreviations and Acronyms

Acronym	Definition
2D	Two-Dimensional
3D	Three-Dimensional
AA	Authorization Architecture
AAR	After Action Review
ABAC	Attribute-Based Access Control
ABS	Anti-Lock Braking System
AC	Air Conditioning
ACL	Access Control List
ACM	Access Control Model
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
AN/ARC	Army Navy / Airborne Radio Communications
AN/PRC	Army Navy / Portable Radio Communications
AN/PSC	Army Navy / Portable Satellite Communications
AN/VRC	Army Navy / Vehicle Radio Communications
AoE	ATA over Ethernet
AOO	Action on Overflow
API	Application Programming Interface
APP Layer	Application Layer
APS	Active Protection System
APU	Auxiliary Power Unit
AR	Army Regulation
ARP	Address Resolution Protocol
AS	Attribute Store Service
ASCII	American Standard Code for Information Interchange
ASN	Abstract Syntax Notation
ASR	Anti-Slip Regulation
ATA-x	Advanced Technology Attachment

Acronym	Definition
B	Byte
BER	Basic Encoding Rules
BIAS	Biometric Identity Assurance Services
BSR	Bootstrap Router
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance, Reconnaissance
C4ISR/EW	Command, Control, Communications, Computers, Intelligence, Surveillance, Reconnaissance/Electronic Warfare
CA	Certificate Authority
CAC	Common Access Card
CAGE	Commercial And Government Entity
CAN	Controller Area Network
CBC	Cipher Block Chaining
CBM	Condition-Based Maintenance
CBRN	Chemical Biological Radiological Nuclear
CC/S/A/FA	Combatant Commands, Services, Defense Agencies, DoD/Army Field Activities
CD	Compact Disc
CIDR	Classless Inter-Domain Routing
CJCSM	Chairman of the Joint Chiefs of Staff Manual
CMRR	Center for Magnetic Recording Research
CMS	Cryptographic Message Syntax
CMSA/CD	Carrier Sense Multiple Access with Collision Detection
CNDSP	Computer Network Defense Service Provider
CNSS	Committee on National Security Systems
CNSSI	Committee on National Security Systems Instruction
COI	Community Of Interest
COMSEC	Communications Security
COTS	Commercial Off-The-Shelf
CP	Change Proposal
CPU	Central Processing Unit
CRIS	Common Relational Information Schema
CRL	Certificate Revocation List
CRUD	Create, Read, Update, Delete

Acronym	Definition
CSS	Central Security Service
CT	Ciphertext
CTI	Central Tire Inflation
DAC	Discretionary Access Control
DAR	Data at Rest
DARE	Data at Rest Encryptor
DCI	Data Communications Interface
DCID	Director of Central Intelligence Directive
DD	Detection Device
DER	Distinguished Encoding Rules
DF	Direction Finding
DH	Diffie-Hellman
DHCP	Dynamic Host Configuration Protocol
DiffServ	Differentiated Services
DISR	DoD Information Technology Standards Registry
DL	Data Link
DN	Distinguished Name
DNS	Domain Name System
DNS-SD	Domain Name System Service Discovery
DoD	Department of Defense
DoDI	Department of Defense Instruction
DPF	Diesel Particulate Filter
DS	Differentiated Services
DSCP	Differentiated Services Code Point
DSS	Digital Signature Standard
DVD	Digital Video Disc
EA	Electronic Attack
ECDSA	Elliptic Curve Digital Signature Algorithm
ECM	Engine Control Module
ECP Group	Elliptic Curve Group modulo a Prime
ECU	Electronic Control Unit

Acronym	Definition
EF	Expedited Forwarding
EGR	Exhaust Gas Recirculation
EIS	Event ID Score
EM	Electromagnetic
EP	Electronic Protection
EPS	Event Priority Score
ES	Event Score
ESP	Encapsulating Security Payload
ESS	Event Source Score
ETS	Event Type Score
EW	Electronic Warfare
EW	Event Window
EWID	Event Window ID
EWTS	Event Window Total Score
FASIT	Future Army System of Integrated Targets
FH	Frequency Hop
FIPS	Federal Information Processing Standards
FMI	Failure Mode Identifier
FoF	Force on Force
FOM	Figure of Merit
FoT	Force on Target
FOUO	For Official Use Only
FRAGO	Fragmentary Order
FSK	Frequency-Shift Keying
FTP	File Transfer Protocol
FTPS	FTP Secure
GB	Gigabyte
Gbps	Giga (1.0×10^9) bits per second
GCM	Galois/Counter Mode
GEO	Geolocation
GIG	Global Information Grid

Acronym	Definition
GMT	Greenwich Mean Time
GOTS	Government Off-The-Shelf
GPS	Global Positioning System
GTP	GIG Technical Profile
GUI	Graphical User Interface
GUID	Globally Unique Identifier
HMAC	Keyed-Hash Message Authentication Code
HMI	Human-Machine Interface
HMMWV	High Mobility Multipurpose Wheeled Vehicle
HTTP	Hypertext Transport Protocol
HTTPS	HTTP Secure
HVAC	Heating, Ventilation, and Air Conditioning
HW	Hardware
IA	Information Assurance
IANA	Internet Assigned Numbers Authority
IAWG	Information Assurance Working Group
IBAC	Identity-Based Access Control
ICAM	Identity, Credential, and Access Management
ICD	Intelligence Community Directive
ICD	Interface Control Document
ICMP	Internet Control Message Protocol
ICV	Integrity Check Value
ID	Identifier
ID	Instance Document
IDE	Integrated Drive Electronics
IDL	Interface Definition Language
IDPS	Intrusion Detection and Prevention System
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IED	Improvised Explosive Device
IEEE	Institute of Electrical and Electronic Engineers

Acronym	Definition
IETF	Internet Engineering Task Force
iFCP	Internet Fibre Channel Protocol
IGMP	Internet Group Management Protocol
IKE	Internet Key Exchange
IMAP	Internet Message Access Protocol
IMU	Inertial Measurement Unit
INCITS	InterNational Committee for Information Technology Standards
INE	Inline Network Encryptor
INS	Inertial Navigation System
IP	Internet Protocol
IPS	Intrusion Prevention System
IPsec	Internet Protocol Security
IPv4	Internet Protocol version 4
IPv4LL	IPv4 Link Local
IPv6	Internet Protocol version 6
IR	Infrared
IRC	Internet Relay Chat
IS	Information System
IS	Instrumentation Systems
IS	Interface Specification
iSCSI	Internet Small Computer System Interface
ISO	International Organization for Standardization
IT	Information Technology
ITAR	International Traffic in Arms Regulations
ITU	International Telecommunication Union
ITU-T	ITU's Telecommunication Standardization Sector
JMRC	Joint Multinational Readiness Center
KB	Kilobyte
LAN	Local Area Network
LCS	Log Collection Start/Stop
LD	Log File Deletion

Acronym	Definition
LE/CI	Law Enforcement and Counterintelligence
LLC	Logical Link Control
LOB	Line of Bearing
LPAN	Live Player Area Network
LPR	Line Printer
LR	Log File Retrieval
LT2	Live Training Transformation
LTS	Long Term Supported
MAC	Mandatory Access Control
MAC	Media Access Control
MAC	Message Authentication Code
MAC	Modular APS Controller
MAN	Manual
MB	Megabyte
Mbps	Mega (1.0×10^6) bits per second
mDNS	Multicast DNS
MGRS	Military Grid Reference System
MIB	Management Information Base
MILES	Multiple Integrated Laser Engagement System
MILS	Multiple Independent Levels of Security
MIME	Multipurpose Internet Mail Extensions
MISM	Motion Imagery System Matrix
MLD	Multicast Listener Discovery
MNAV	M-code Navigation
MORA	Modular Open RF Architecture
MPD	Message Processing and Dispatching
MT	MORA Type
MTU	Maximum Transmission Unit
MUOS	Mobile User Objective System
N/A	Not Applicable
NAS	Network Attached Storage

Acronym	Definition
NAT	Network Address Translation
NAT	Network Address Translator
NCES	Net-Centric Enterprise Services
NDP	Neighbor Discovery Protocol
NETCONF	Network Configuration Protocol
NFS	Network File System
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
NNTP	Network News Transfer Protocol
NSA	National Security Agency
NSN	National Stock Number
NTP	Network Time Protocol
NUC	Non-uniformity Compensation
OA&M	Operations, Administration, and Management
OASIS	Organization for the Advancement of Structured Information Standards
OBD	On-Board Diagnostic
OID	Object Identifier
OS	Operating System
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PA	Power Amplifier
PBAC	Policy-Based Access Control
PBAS	Precedence-Based Assured Services
PCAP	Packet Capture
PCM	Pulse Code Modulation
PDP	Policy Decision Point
PDS	Policy Decision Service
PDU	Protocol Data Unit
PEM	Privacy-Enhanced Electronic Mail
PEP	Policy Enforcement Point
PES	Policy Enforcement Service

Acronym	Definition
PFS	Perfect Forward Secrecy
PHB	Per Hop Behavior
PHY Layer	Physical Layer
PIM	Protocol Independent Multicast
PIM-DM	Protocol Independent Multicast-Dense Mode
PIM-SM	Protocol Independent Multicast-Sparse Mode
PIN	Personal Identification Number
PKCS	Public-Key Cryptography Standards
PKI	Public Key Infrastructure
PL	Precedence Level
PMTUD	Path Maximum Transmission Unit (MTU) Discovery
POP	Post Office Protocol
POP3	Post Office Protocol version 3
POSIX	Portable Operating System Interface
PRF	Pseudorandom Function
PRI	Priority
PRN	Pseudo-Random Noise
PS	Policy Store Service
PT	Plaintext
PTO	Power Take-Off
PTP	Precision Time Protocol
PTPv2	Precision Time Protocol version 2
PTR	Pointer Record
PTT	Push-to-Talk
PUB	Publication
PVT	Position/Velocity/Time
QoS	Quality of Service
RA	Reference Architecture
RAM	Random Access Memory
RBAC	Role-Based Access Control
RDMS	Range Data Measurement Subsystem

Acronym	Definition
RF	Radio Frequency
RFC	Request for Comment
RIP	Routing Information Protocol
RMF	Risk Management Framework
RP	Recommended Practice
RP	Rendezvous Point
RPC	Remote Procedure Call
RPG	Rocket-Propelled Grenade
RR	Resource Record
RSTP	Rapid Spanning Tree Protocol
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
RWS	Remote Weapon Station
SA	Security Association
SA	Situational Awareness
SA/A-S	Selective Availability/Anti-Spoofing
SAE	Society of Automotive Engineers
SAML	Security Assertion Markup Language
SAS	Serial Attached SCSI
SATA	Serial Advanced Technology Attachment
SC	Single Channel
SCR	Selective Catalytic Reduction
SCSI	Small Computer System Interface
SCTP	Stream Control Transmission Protocol
SDP	Session Description Protocol
SE	Standard Edition
SELS	Security Event Log Service
SHA	Secure Hash Algorithm
SIDC	Symbol Identification Code
SINCGARS	Single Channel Ground and Airborne Radio System

Acronym	Definition
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SLF	Size of the Log File
SMI	Structure of Management Information
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOA	Service-Oriented Architecture
SOI	Signal of Interest
SP	Special Publication
SPN	Suspect Parameter Number
SPU	Shared Processing Unit
SRV	Service Record
SRW	Soldier Radio Waveform
SSH	Secure Shell
SSL	Secure Socket Layer
SSM	Source-Specific Multicast
SSO	Single Sign-On
STIG	Security Technical Implementation Guide
STL	Standard Template Library
STP	Spanning Tree Protocol
SV	Space Vehicle
SW	Software
SWaP	Size, Weight, and Power
SWaP-C	Size, Weight, Power, and Cost
TAI	International Atomic Time
TASKORD	Tasking Order
TB	Terabyte
TBD	To Be Determined
TCP	Transmission Control Protocol
TESS	Tactical Engagement Simulation Systems
TFOM	Time Figure of Merit

Acronym	Definition
TLS	Transport Layer Security
TM	Transport Mappings
TPU	Terminal Player Unit
TR	Technical Report
TSC	Torque/Speed Control
TSPI	Time-Space-Position Information
TTP	Tactics, Techniques and Procedures
TX	Transmission
TXT	Text Record
UCS	Universal Character Set
UCSD	University of California, San Diego
UDP	User Datagram Protocol
UI	User Interface
UIC	Unit Identification Code
ULA	Unique Local Address
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Unique Reference Number
US	United States
USB	Universal Serial Bus
USM	User-Based Security Model
UTC	Coordinated Universal Time
UTF	UCS / Unicode Transformation Format
UUID	Universally Unique Identifier
VACM	View-Based Access Control Model
VCL	VICTORY Configuration Language
VCL ID	VCL Instance Document
VDB	VICTORY Data Bus
VDM	VICTORY Data Message
VIC	Vehicle Inter Communications
VICTORY	Vehicular Integration for C4ISR/EW Interoperability

Acronym	Definition
VISA	Video and Imagery Situational Awareness
VLAN	Virtual Local Area Network
VLSG	Vehicle Limiting Speed Governor
VM	Virtual Machine
VoIP	Voice Over Internet Protocol
VOX	Voice Operated Switch
VSSO	VICTORY Standards Support Office
VT	VICTORY Type
W3C	World Wide Web Consortium
WAN	Wide Area Network
WARNORD	Warning Order
WAVE	Waveform Audio File Format
WebDAV	Web Distributed Authoring and Versioning
WGS	World Geodetic System
WMI	Warfighter Machine Interface
WS	Web Services
WSDL	Web Services Description Language
WS-I	Web Services Interoperability
WSS	Web Services Security
XACML	eXtensible Access Control Markup Language
XML	Extensible Markup Language
XSD	XML Schema Definition Language
Z	Zulu
Zeroconf	Zero Configuration Networking

3. DOCUMENT CONVENTIONS

3.1 Standards Key Words

In this document, key words are used to signify the requirements in the standards. This section defines these words (derived from IETF RFC 2119) as they should be interpreted in VICTORY Specifications. Note that the force of these words is modified by the requirement level of the specification in which they are used.

- SHALL: This word means that the definition is an absolute requirement of the specification.
- SHALL NOT: This phrase means that the definition is an absolute prohibition of the specification.
- SHOULD: This word means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT: This phrase means that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- MAY: This word means that an item is truly optional. One implementation may choose to include the item because a particular marketplace requires it or because the implementation enhances the product while another implementation may omit the same item. An implementation that does not include a particular option SHALL be prepared to interoperate with another implementation that does include the option, though perhaps with reduced functionality. In the same vein, an implementation that does include a particular option SHALL be prepared to interoperate with another implementation that does not include the option (except, of course, for the feature the option provides).

3.2 Compliance

There is no concept of overall VICTORY compliance. VICTORY defines sets of component types, each of which defines requirements for interfaces. Compliance is designed, claimed, tested, and documented based on the component types relevant for the component or system in question. No component or system is likely to implement all of the component types, as not all of them are relevant for every system.

A system that implements a set of VICTORY component types, has been tested against the specifications and documented to be compliant with those component types can be said to be compliant with that list of VICTORY component types.

For the U.S. Army to benefit to the fullest extent, systems must be specified, designed, tested and documented to be compliant with as large a subset of the VICTORY component types as applies to each system.

3.3 Maturity Level Markings

Each interface will be described by one or more sections of the standard specifications document. Each section may have several levels of subsections. Each given section can be considered an individual specification (or standard), the composition of which represents the current state of the specification for each of the interfaces defined by the architecture.

As this set of specifications is being documented and matured over time, the individual specifications may be at different maturity levels. For that reason, each specification is marked with a maturity identifier: <Preliminary>, <Informational>, <Experimental>, <Proposed Standard>, <Draft Standard>, or <VICTORY Standard>. The meanings of these markings were described earlier in the section titled “Maturity Levels.” If a section is not marked, it inherits the maturity level of its parent section, if it has one. If a maturity level cannot be determined, it can be assumed to be a placeholder to be specified at a later time, and is thus <Preliminary>. The maturity identifier does not necessarily represent the maturity of a particular specification (or standard), but instead identifies the maturity of the acceptance (adoption) of the specification (or standard) into this standard specifications document.

3.4 Applicability Markings

Each specification inside a component or interface type table contains a field for marking the applicability of the specification. Specifications that have not yet been matured to proposed standard are optional, regardless of what is marked inside the applicability field. These experimental specifications are marked with "(+)" inside the applicability field of the type table. This indicates that compliance tests will not take this specification into account until it is matured to the "Proposed" standard level. At that point, the (+) will be removed and the compliance test suite artifacts will be updated to test the specification and take its applicability into account in determining compliance.

3.5 Usage of Specification Tags

This document incorporates the use of specification tags to track the individual specifications in a manner that is independent of the organization of the document. A specification tag is defined below:

<Spec ID - Date of Publication, Maturity Level>

The "Spec ID" is a 5-digit identification number with a range of 01001 - 99999. Spec ID shall be unique within the VICTORY specifications. This number is comprised of two fields: a 2-digit specification group, and a 3-digit identifier. Each specification group may have identifiers ranging from 001 to 999. Table 2 provides the list of possible specification groups. For example, the Spec ID "20002" denotes the specification group "Application Data Interfaces" with an identifier of "002."

Table 2. Specification Groups

Specification Group	Specification Group Description
01	Data Transport Interfaces
02	Time Synchronization Interfaces
03	Device Functionality
05	Non-Networked Nodes Interfaces
10	Common Management Parameters
11	VDB Management Interfaces
12	Infrastructure Node Management Interfaces
13	End Node Management Interfaces
14	Information Assurance Management Interfaces
15	C4ISR/EW Application Management Interfaces
16	Platform Systems Management Interfaces
20	Application Data Interfaces
30	VICTORY Configuration Language
40	Access Control Interfaces
47	MORA RF Interfaces
48	MORA Low Latency Bus
49	MORA Management Interfaces

The "Date of Publication" is the date of the last update of the specification.

The “Maturity Level” is the maturity level of the specification using one of the following abbreviations:

- **`Pre`** – Preliminary
- **`Inf`** – Informational
- **`Exp`** – Experimental
- **`Pro`** – Proposed Standard
- **`Dft`** – Draft Standard
- **`Std`** – VICTORY Standard

Each specification tag is contained within a header, and used to uniquely identify (mark) each specification contained in a version of the specifications document. A specification tag assigned to a heading indicates that the heading, and all subordinate headings, comprise a specification.

3.6 Usage of VICTORY Type Tags

The “VICTORY Type Tag” is an alphanumeric identification tag with a numeric range of 50000 – 99999 and a prefix of “VT.” These are used to identify the system, component, and interface type specification tables. The VICTORY Type Tag also contains a document version identifier with the format of ‘X.y’ where ‘X’ denotes a major release and ‘y’ denotes a minor release. Table 3 below provides a mapping between version identifiers and document release dates.

Table 3. Document Release Date To Document Version Mapping

Document Version	Document Release Date
1.2	July 30, 2012
1.3	November 9, 2012
1.4	June 30, 2014
1.5	May 2, 2013
1.6	March 31, 2015
1.7	February 15, 2018
1.8	April 18, 2019
1.9	April 09, 2021

The document version refers to the latest version of that document. For example, if there is a version 1.4.1, document version 1.4 refers to 1.4.1. These VICTORY component type tags will be referred to from outside the VICTORY Specifications document.

3.7 Usage of Notes

Throughout the VICTORY Specifications document, notes will be used to provide information relevant to the material being presented in the surrounding text. These notes will act as a supplement to guide the reader with rationale and advisories where they are deemed useful; however, the content of the notes is purely informational.

Either by their presence and/or removal, the notes shall not augment the rules and specifications presented in the VICTORY Standards document in any way.

NOTE: *This is an example of a VICTORY Specifications document note.*

3.8 Usage of Mass Nouns

There is controversy regarding whether the words *data*, *criteria*, and *media* should be treated as **plural nouns** or as **mass nouns**. Proper grammar is usually relatively straightforward: match singular subjects with singular verbs and match plural subjects with plural verbs. This becomes more complex with certain types of nouns that entered English from other languages.

Mass nouns are nouns that cannot be made singular, yet they take a singular verb. The old school method is to treat these nouns as normal plurals and use them with plural verbs. For grammar sticklers, this is the only correct way to use these nouns. For example:

- The singular form of data is the word datum. Traditionally, datum takes a singular verb, and data takes a plural verb.
  - These data are inconclusive.
  - This datum shows little without the rest of the statistics.
- The singular form of criteria is criterion. Similarly, criterion takes a singular verb, and criteria takes a plural verb.
  - As a whole, the criteria are fair.
  - This specific criterion is too harsh.
- The singular form of media is medium. This means medium uses a singular verb, and media uses a plural verb.
  - People say the media are biased.
  - The medium of the print newspaper is struggling.

In the real world, however, one is much more likely to hear *data*, *criteria*, and *media* used as mass nouns with singular verbs, like in these examples:

- The data matches the information entered.
- The selection criteria is optional and includes name and address.
- Copper media is limited by distance and signal interference.

Based on these real-world examples, the VICTORY Working Groups have decided that the VICTORY Specifications document will treat *data*, *criteria*, and *media* as **mass nouns** and pair them with singular verbs.

4. VICTORY SYSTEM AND COMPONENT TYPE SPECIFICATIONS

As described in "VICTORY Architecture – Version A," the VICTORY architecture is made up of a set of types that can be instantiated and integrated to create designs. The set includes system types, component types, and their interfaces. The architecture also identifies a structure called the VICTORY Data Bus (VDB), which is a conceptual entity that provides functions and interfaces as a composite, and which is made up of other component types.

Figure 1, Figure 2, and Figure 3 provide hierarchical breakdowns of the types for the VDB, C4ISR/EW Systems, and Platform Systems. Note that the system and component types are grouped hierarchically for convenience and accessibility to the reader. The groupings that have been chosen in this document should not affect strongly the content of the component type or interface specifications, nor should they be used to imply additional meaning to the functionality of the component types described herein.

The following sections walk through the component and system types that make up the VICTORY architecture. The top-level organization is as follows:

- VICTORY Data Bus: Component types that make up the VDB and the interfaces to be standardized
- C4ISR/EW Systems: Systems that the VDB will integrate, the component types that make them up, and the interfaces to be standardized
- Platform Systems: Systems that the VDB will interface and the interfaces to be standardized

Each section provides a set of tables: a system or component type table that lists the interfaces that are required, recommended, or optional for that system or component type, and interface specification tables that list the individual specifications that are required, recommended, or optional for that interface. These tables are intended to be the entry point for component implementers and system integrators for determining which specifications should be used in order to achieve compliance with VICTORY system types and component types.

Note that a hardware or software product will likely be designed to implement several system and component types. A product will implement the interfaces of each of the component types it represents. For instance, a vendor may design a device that implements the shared processing unit, position service, time synchronization service, and VDB management service component types. In this case, the developer must reference the component type tables and implement at least the union of the interfaces required by those component type specifications.

VDB Component Types	Shared Data Service Component Types	Time Synchronization Service
		Position Service
		Orientation Service
		Direction of Travel Service
	Access Control Component Types	Authentication Service
		Policy Decision Service
		Policy Enforcement Service
		Policy Store Service
		Attribute Store Service
	Infrastructure Component Types	Switch
		Router
	Information Assurance Component Types	Host-Based Antivirus
		Host-Based Firewall
		Data Signing Service
		Signed Data Verification Service
		Network Firewall
		Intrusion Detection and Prevention System
		Data at Rest Encryptor
		Inline Network Encryptor
	End Node Component Types	Shared Processing Unit
		Network Attached Storage
		Generic End Node
	System Management Component Types	VDB Management Service
		VICTORY Configuration Description

Figure 1. Hierarchical Breakdown of VICTORY Data Bus Types

C4ISR/EW Component Types	Audio Communications Component Types	Streaming Audio Data Source
		Single-Instance Audio Data Source
		Intercom
		Voice Radio
	Video and Imagery Situational Awareness Component Types	Video or Image Sensor
		Video or Image Encoder
		Camera Gimbal
	Recording and Logging Component Types	Mission Recorder
		Data Logger
	Extra-Vehicle Network Interface Component Types	Data Radio
	Electronic Warfare Component Types	Electronic Warfare Device
	C4ISR/EW Sensor Component Types	GPS Receiver

Figure 2. Hierarchical Breakdown of C4ISR/EW System Types

Platform System Types	Automotive System Types	Automotive Bus
	Power Distribution System Types	Power Distribution System
	Lethality System Types	Remote Weapon Station
		Weapon System
		Targeting System
	Logistics System Types	Vehicle Configuration Service
	Protection System Types	Threat Detection and Reporting Service
		Active Protection System
	Training System Types	Live Training Transformation Adapter

Figure 3. Hierarchical Breakdown of Platform System Types

5. COMMON VICTORY SPECIFICATIONS

To reduce redundancy in the system and component type specifications in the sections that follow, this section details sets of specifications that are common across many of the VICTORY component types.

5.1 Common Data Interface Specifications <VT59901-V1.6>

The table below identifies the specifications that are common for many component data interfaces.

Table 4. Common Data Interface Specifications <VT59901-V1.6>

Spec Title	Spec Tag	Applicability
Network Port Assignment	01035–20120725, Pro	Required
Application Layer Data Encoding	20001–20120725, Pro	Required
Application Layer Message Encapsulation	20002–20150331, Pro	Required
Timestamp Format for Application Layer Data	20004–20140630, Pro	Required

5.2 Common Management Interface Specifications <VT59902-V1.9>

The table below identifies the specifications that are common for many component management interfaces.

Table 5. Common Management Interface Specifications <VT59902-V1.9>

Spec Title	Spec Tag	Applicability
Transmission Control Protocol	01026–20120725, Pro	Required
Network Port Assignment	01035–20120725, Pro	Required
SOAP Compliance and Standards	10001–20120725, Pro	Required
Web Services Description Language	10002–20120725, Pro	Required
VICTORY Schemas and WSDLs	10003–20120725, Pro	Required
Version Information Management Parameters	10007–20210409, Pro	Required
Management Interface Error Types	10013–20210409, Pro	Required
Syslog-Based Health Publishing Management Specification	11005–20210409, Pro	Required
Web Service Specifications	VT50420-V1.4	Recommended

5.2.1 Data Interface Physical and Data Link Layer Specifications <VT59904-V1.9>

The table below identifies the specifications that are common for router, switch, and generic end node for the physical and data link layers.

Table 6. Data Interface Physical and Data Link Layer Specifications <VT59904-V1.9>

Spec Title	Spec Tag	Applicability
Copper Data Interface Physical and Data Link Layer Specifications	VT59905-V1.7	(1) Required
Fiber Data Interface Physical and Data Link Layer Specifications	VT59906-V1.7	
Backplane Copper Data Interface Physical and Data Link Layer Specifications	VT59907-V1.9	
Frame Structure	01013–20121113, Pro	Required
Media Access Control	01014–20121113, Pro	Required
Logical Link Control	01015–20120725, Pro	Required
Link Layer Switching	01016–20120725, Pro	Required
Link Layer Flow Control	01017–20121113, Pro	Required
Virtual Local Area Network (VLAN)	01077–20120131, Exp	(+) Required

(+) Experimental specifications are not included in compliance tests.

(1) One or more of VT59905, VT59906, VT59907 is Required

5.2.2 Copper Data Interface Physical and Data Link Layer Specifications <VT59905-V1.7>

The table below identifies the copper specifications that are common for router, switch, and generic end node for the physical and data link layers.

Table 7. Copper Data Interface Physical and Data Link Layer Specifications <VT59905-V1.7>

Spec Title	Spec Tag	Applicability
10BASE-T	01001–20121113, Pro	Required
100BASE-TX	01003–20121113, Pro	Required
Copper Auto-Negotiation	01011–20121113, Pro	Required
1000BASE-T	01006–20121113, Pro	Recommended
10GBASE-X/10GBASE-T	01009–20121113, Pro	Optional

5.2.3 Fiber Data Interface Physical and Data Link Layer Specifications <VT59906-V1.7>

The table below identifies the fiber specifications that are common for router, switch, and generic end node for the physical and data link layers.

Table 8. Fiber Data Interface Physical and Data Link Layer Specifications <VT59906-V1.7>

Spec Title	Spec Tag	Applicability
10BASE-FL	01002–20121113, Pro	(1) Required
100BASE-FX	01004–20121113, Pro	
100BASE-LX10	01005–20121113, Pro	

Spec Title	Spec Tag	Applicability
1000BASE-SX	01007–20150331, Pro	
1000BASE-LX	01008–20150331, Pro	
10GBASE-R	01010–20121113, Pro	

(1) One or more of 01002, 01004, 01005, 01007, 01008, 01010 is Required

5.2.4 Backplane Copper Data Interface Physical and Data Link Layer Specifications <VT59907-V1.9>

The table below identifies the specifications that are common for router, switch, and generic end node for the physical and data link layers.

Table 9. Backplane Copper Data Interface Physical and Data Link Layer Specifications <VT59907-V1.9>

Spec Title	Spec Tag	Applicability
1000BASE-KX	01095–20210409, Exp	(+) (1) Required
10GBASE-KR	01096–20210409, Exp	
Copper Auto-Negotiation	01011–20121113, Pro	Optional

(+) Experimental specifications are not included in compliance tests.

(1) One or more of 01095, 01096 is Required.

5.3 Authentication and Authorization Security Specifications <VT50400-V1.9>

The table below identifies the authentication and authorization specifications for securing components on the VDB.

Table 10. Authentication and Authorization Security Specifications <VT50400-V1.9>

Spec Title	Spec Tag	Applicability
Authentication Methods	40001–20180215, Pro	Required
Authentication Security Specifications	VT50440-V1.9	Required
Authorization Security Specifications	VT50430-V1.9	Recommended

5.4 Web Service Specifications <VT50420-V1.4>

The table below identifies the web service specifications for securing SOAP-based management interfaces.

Table 11. Web Service Specifications <VT50420-V1.4>

Spec Title	Spec Tag	Applicability
Web Services Framework	14018–20130315, Pro	Required
Web Services Security Specification	14019–20130315, Pro	Required

5.5 Authorization Security Specifications <VT50430-V1.9>

The table below identifies the authorization security specifications.

Table 12. Authorization Security Specifications <VT50430-V1.9>

Spec Title	Spec Tag	Applicability
Authorization Model	40003–20180215, Pro	Required
Authorization Architecture Entrypoints	40012–20210409, Pro	Required

5.6 Authentication Security Specifications <VT50440-V1.9>

The table below identifies the authentication security specifications.

Table 13. Authentication Security Specifications <VT50440-V1.9>

Spec Title	Spec Tag	Applicability
Authentication Model	40013–20180215, Pro	Required
Authentication Entrypoints	40014–20210409, Pro	Required

6. COMMON VICTORY INTERFACES

To reduce redundancy in the system and component type specifications in the sections that follow, this section details sets of interfaces that are common across many of the VICTORY component types.

6.1 VDB Connector, Signaling, and Physical Media Interface <VT52303-V1.2>

The table below identifies the specifications for the VDB connector, signaling, and physical media interface.

Table 14. VDB Connector, Signaling, and Physical Media Interface <VT52303-V1.2>

Spec Title	Spec Tag	Applicability
Ethernet	01079–20120131, Exp	(+) Recommended
Dual Ethernet	01080–20120131, Exp	(+) Recommended
Other Ethernet Connectors	01081–20120131, Exp	(+) Recommended

(+) Experimental specifications are not included in compliance tests.

6.2 Syslog-Based Health Publishing Interface <VT50103-V1.9>

The table below identifies the specifications for the Syslog-based health publishing interface.

Table 15. Syslog-Based Health Publishing Interface <VT50103-V1.9>

Spec Title	Spec Tag	Applicability
Syslog Protocol	10005–20120725, Pro	Required
Syslog Protocol Data Parameters	10011–20210409, Pro	Required
Syslog-Based Connectionless Health Publishing Specification	10023–20120725, Pro	Required

6.3 SNMP-Based Health Publishing Interface <VT59903-V1.6>

The table below identifies the specifications for the infrastructure component health publishing interface.

Table 16. SNMP-Based Health Publishing Interface <VT59903-V1.6>

Spec Title	Spec Tag	Applicability
Generic Infrastructure Component Health Status Specification	10016–20150331, Pro	Required

6.4 Auto-Discovery Interface <VT59930-V1.6>

The table below identifies the specifications for the auto-discovery interface.

Table 17. Auto-Discovery Interface <VT59930-V1.6>

Spec Title	Spec Tag	Applicability
Zero Configuration Networking	01044–20140630, Pro	Required

7. VICTORY DATA BUS SYSTEM SPECIFICATIONS

The VDB System Specifications group consists of the component types that are instantiated to create a vehicle network. An instance of a VDB is made up of hardware and software that implement several different VICTORY component types. Each of the VDB system-level interfaces is realized by a design component, which implements the standard interfaces of the component types with which it complies.

7.1 Shared Data Service Component Types

7.1.1 Time Synchronization Service <VT50000-V1.7>

The table below identifies the interfaces for the time synchronization service component type.

Table 18. Time Synchronization Service <VT50000-V1.7>

Spec Title	Spec Tag	Applicability
Time Synchronization Interface	VT50001-V1.7	Required

7.1.1.1 Time Synchronization Interface <VT50001-V1.7>

The tables below identify the specifications for the time synchronization interface.

NOTE: *A source for network time is required on a VICTORY-Compliant VDB. A software service or network component which implements the "time synchronization interface" must provide the capability for a source of time using both of these protocols (NTP and PTP). Network clients are free to use either or none of these provided network time sources.*

Table 19. Time Synchronization Interface <VT50001-V1.7>

Spec Title	Spec Tag	Applicability
Network Time Protocol Specifications	VT50001.T1-V1.4	Required
Precision Time Protocol Specifications	VT50001.T2-V1.7	Required

7.1.1.1.1 Network Time Protocol Specifications <VT50001.T1-V1.4>

The table below identifies the specifications for the network time protocol specifications.

Table 20. Network Time Protocol Specifications <VT50001.T1-V1.4>

Spec Title	Spec Tag	Applicability
Network Time Protocol	02002–20121113, Pro	Required
Health Monitoring of Network Time Protocol	02004–20121113, Exp	(+) Optional
Network Time Protocol Management Specification	02005–20121113, Exp	(+) Optional

(+) Experimental specifications are not included in compliance tests.

7.1.1.1.2 Precision Time Protocol Specifications <VT50001.T2-V1.7>

The table below identifies the specifications for the precision time protocol specifications.

Table 21. Precision Time Protocol Specifications <VT50001.T2-V1.7>

Spec Title	Spec Tag	Applicability
Precision Time Protocol	02003–20180215, Pro	Required
Health Monitoring of Precision Time Protocol	02006–20121113, Exp	(+) Optional
Precision Time Protocol Management Specification	02007–20121113, Exp	(+) Optional

(+) Experimental specifications are not included in compliance tests.

7.1.2 Position Service <VT50100-V1.9>

The table below identifies the interfaces for the position service component type.

Table 22. Position Service <VT50100-V1.9>

Spec Title	Spec Tag	Applicability
Position Data Interface	VT50101-V1.6	Required
Position Management Interface	VT50102-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.1.2.1 Position Data Interface <VT50101-V1.6>

The table below identifies the specifications for the position data interface.

Table 23. Position Data Interface <VT50101-V1.6>

Spec Title	Spec Tag	Applicability
Common Data Interface Specifications	VT59901-V1.6	Required
Position Data Specification	20009–20120725, Pro	Required

7.1.2.2 Position Management Interface <VT50102-V1.9>

The table below identifies the specifications for the position management interface.

Table 24. Position Management Interface <VT50102-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
VICTORY Data Management Parameters	10008–20210409, Pro	Required
Position Data Management Parameters	15003–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required

Spec Title	Spec Tag	Applicability
Position Service Configuration Description	30005–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

7.1.3 Orientation Service <VT50200-V1.9>

The table below identifies the interfaces for the orientation service component type.

Table 25. Orientation Service <VT50200-V1.9>

Spec Title	Spec Tag	Applicability
Orientation Data Interface	VT50201-V1.6	Required
Orientation Management Interface	VT50202-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.1.3.1 Orientation Data Interface <VT50201-V1.6>

The table below identifies the specifications for the orientation data interface.

Table 26. Orientation Data Interface <VT50201-V1.6>

Spec Title	Spec Tag	Applicability
Common Data Interface Specifications	VT59901-V1.6	Required
Orientation Data Specification	20008–20120725, Pro	Required

7.1.3.2 Orientation Management Interface <VT50202-V1.9>

The table below identifies the specifications for the orientation management interface.

Table 27. Orientation Management Interface <VT50202-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
VICTORY Data Management Parameters	10008–20210409, Pro	Required
Orientation Data Management Parameters	15002–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Orientation Service Configuration Description	30006–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

7.1.4 Direction of Travel Service <VT50300-V1.9>

The table below identifies the interfaces for the direction of travel service component type.

Table 28. Direction of Travel Service <VT50300-V1.9>

Spec Title	Spec Tag	Applicability
Direction of Travel Data Interface	VT50301-V1.6	Required
Direction of Travel Management Interface	VT50302-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.1.4.1 Direction of Travel Data Interface <VT50301-V1.6>

The table below identifies the specifications for the direction of travel data interface.

Table 29. Direction of Travel Data Interface <VT50301-V1.6>

Spec Title	Spec Tag	Applicability
Common Data Interface Specifications	VT59901-V1.6	Required
Direction of Travel Data Specification	20007–20120725, Pro	Required

7.1.4.2 Direction of Travel Management Interface <VT50302-V1.9>

The table below identifies the specifications for the direction of travel management interface.

Table 30. Direction of Travel Management Interface <VT50302-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
VICTORY Data Management Parameters	10008–20210409, Pro	Required
Direction of Travel Data Management Parameters	15001–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Direction of Travel Service Configuration Description	30007–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

7.2 Access Control Component Types

7.2.1 Authentication Service <VT50401-V1.9>

The table below identifies the interfaces for the authentication service component type.

Table 31. Authentication Service <VT50401-V1.9>

Spec Title	Spec Tag	Applicability
Authentication Service Management Interface	VT50402-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.2.1.1 Authentication Service Management Interface <VT50402-V1.9>

The table below identifies the specifications for the authentication service management interface.

Table 32. Authentication Service Management Interface <VT50402-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Authentication Service Management Specification	14028–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Required
Data Authentication for Management Interfaces	10019–20150331, Pro	Recommended
Authentication Service Configuration Description	30026–20210409, Pro	Recommended
Authentication Methods	40001–20180215, Pro	Recommended

7.2.2 Attribute Store Service <VT50800-V1.9>

The table below identifies the interfaces for the attribute store service component type.

Table 33. Attribute Store Service <VT50800-V1.9>

Spec Title	Spec Tag	Applicability
Attribute Store Management Interface	VT50802-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.2.2.1 Attribute Store Management Interface <VT50802-V1.9>

The table below identifies the specifications for the attribute store management interface.

Table 34. Attribute Store Management Interface <VT50802-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Attribute Store Management Specification	14015–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Required
Data Authentication for Management Interfaces	10019–20150331, Pro	Recommended

Spec Title	Spec Tag	Applicability
Attribute Store Service Configuration Description	30022–20210409, Pro	Recommended
Authentication Methods	40001–20180215, Pro	Recommended

7.2.3 Policy Decision Service <VT50500-V1.9>

The table below identifies the interfaces for the policy decision service component type.

Table 35. Policy Decision Service <VT50500-V1.9>

Spec Title	Spec Tag	Applicability
Policy Decision Management Interface	VT50502-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.2.3.1 Policy Decision Management Interface <VT50502-V1.9>

The table below identifies the specifications for the policy decision management interface.

Table 36. Policy Decision Management Interface <VT50502-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Policy Decision Management Specification	14014–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Required
Data Authentication for Management Interfaces	10019–20150331, Pro	Recommended
Policy Decision Service Configuration Description	30023–20210409, Pro	Recommended
Authentication Methods	40001–20180215, Pro	Recommended

7.2.4 Policy Enforcement Service <VT50600-V1.9>

The table below identifies the interfaces for the policy enforcement service component type.

Table 37. Policy Enforcement Service <VT50600-V1.9>

Spec Title	Spec Tag	Applicability
Policy Enforcement Management Interface	VT50602-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.2.4.1 Policy Enforcement Management Interface <VT50602-V1.9>

The table below identifies the specifications for the policy enforcement management interface.

Table 38. Policy Enforcement Management Interface <VT50602-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Policy Enforcement Management Specification	14013–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Required
Data Authentication for Management Interfaces	10019–20150331, Pro	Recommended
Policy Enforcement Service Configuration Description	30024–20210409, Pro	Recommended
Authentication Methods	40001–20180215, Pro	Recommended

7.2.5 Policy Store Service <VT50700-V1.9>

The table below identifies the interfaces for the policy store service component type.

Table 39. Policy Store Service <VT50700-V1.9>

Spec Title	Spec Tag	Applicability
Policy Store Management Interface	VT50702-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.2.5.1 Policy Store Management Interface <VT50702-V1.9>

The table below identifies the specifications for the policy store management interface.

Table 40. Policy Store Management Interface <VT50702-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Policy Store Management Specification	14022–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Required
Data Authentication for Management Interfaces	10019–20150331, Pro	Recommended
Policy Store Service Configuration Description	30025–20210409, Pro	Recommended
Authentication Methods	40001–20180215, Pro	Recommended

7.3 Infrastructure Component Types

7.3.1 Switch <VT50900-V1.9>

The table below identifies the interfaces for the switch component type.

Table 41. Switch <VT50900-V1.9>

Spec Title	Spec Tag	Applicability
Switch Data Interface	VT50901-V1.9	Required
Switch Management Interface	VT50902-V1.7	Required
SNMP-Based Health Publishing Interface	VT59903-V1.6	Required
VDB Connector, Signaling, and Physical Media Interface	VT52303-V1.2	Recommended
Switch IPv6 Data Interface	VT50903-V1.9	Recommended

7.3.1.1 Switch Data Interface <VT50901-V1.9>

The tables below identify the specifications for the switch data interface.

Table 42. Switch Data Interface <VT50901-V1.9>

Spec Title	Spec Tag	Applicability
Data Interface Physical and Data Link Layer Specifications	VT59904-V1.9	Required
Switch Data Interface Network Layer and Higher Specifications	VT50901.T4-V1.7	Required

7.3.1.2 Switch IPv6 Data Interface <VT50903-V1.9>

The table below identifies the specifications for the switch ipv6 data interface.

Table 43. Switch IPv6 Data Interface <VT50903-V1.9>

Spec Title	Spec Tag	Applicability
Data Interface Physical and Data Link Layer Specifications	VT59904-V1.9	Required
Switch IPv6 Data Interface Network Layer and Higher Specifications	VT50901.T7-V1.7	Required

7.3.1.3 Switch Data Interface Network Layer and Higher Specifications <VT50901.T4-V1.7>

The table below identifies the specifications for the switch data interface network layer and higher specifications.

Table 44. Switch Data Interface Network Layer and Higher Specifications <VT50901.T4-V1.7>

Spec Title	Spec Tag	Applicability
Switch Data Interface IPv4-Specific Specifications	VT50901.T5-V1.2	Required
Transmission Control Protocol	01026–20120725, Pro	Required
User Datagram Protocol	01027–20120725, Pro	Required
Static Host Addressing	01061–20120725, Pro	Required
QoS Mapping to Components	01036–20120725, Pro	Recommended
DiffServ	01037–20120725, Pro	Recommended
DiffServ Code Point (DSCP) Assignments	01039–20120725, Pro	Recommended

Spec Title	Spec Tag	Applicability
PTP End-to-End Transparent Clock Mode for Infrastructure Components	02008–20180215, Exp	(+) Optional

(+) Experimental specifications are not included in compliance tests.

7.3.1.4 Switch IPv6 Data Interface Network Layer and Higher Specifications <VT50901.T7-V1.7>

The table below identifies the specifications for the switch ipv6 data interface network layer and higher specifications.

Table 45. Switch IPv6 Data Interface Network Layer and Higher Specifications <VT50901.T7-V1.7>

Spec Title	Spec Tag	Applicability
Switch Data Interface IPv6-Specific Specifications	VT50901.T6-V1.6	Required
Transmission Control Protocol	01026–20120725, Pro	Required
User Datagram Protocol	01027–20120725, Pro	Required
Static Host Addressing	01061–20120725, Pro	Required
QoS Mapping to Components	01036–20120725, Pro	Recommended
DiffServ	01037–20120725, Pro	Recommended
DiffServ Code Point (DSCP) Assignments	01039–20120725, Pro	Recommended
PTP End-to-End Transparent Clock Mode for Infrastructure Components	02008–20180215, Exp	(+) Optional

(+) Experimental specifications are not included in compliance tests.

7.3.1.5 Switch Data Interface IPv4-Specific Specifications <VT50901.T5-V1.2>

The table below identifies the specifications for the switch data interface ipv4-specific specifications.

Table 46. Switch Data Interface IPv4-Specific Specifications <VT50901.T5-V1.2>

Spec Title	Spec Tag	Applicability
Internet Protocol Version 4 Specification	01018–20120725, Pro	Required
Internet Control Message Protocol for IPv4	01021–20120725, Pro	Required
IPv4 Datagram Transmission	01062–20120725, Pro	Required
IPv4 Private Address Space	01066–20120725, Pro	Required
VDB IPv4 Address Translation	01069–20120725, Pro	Required
IPv4 Dynamic Host Addressing	01071–20120725, Pro	Required
IGMP Snooping in IPv4	01024–20120725, Pro	Recommended

7.3.1.6 Switch Data Interface IPv6-Specific Specifications <VT50901.T6-V1.6>

The table below identifies the specifications for the switch data interface ipv6-specific specifications.

Table 47. Switch Data Interface IPv6-Specific Specifications <VT50901.T6-V1.6>

Spec Title	Spec Tag	Applicability
Internet Protocol Version 6 Specification	01019–20110729, Exp	(+) Required
Internet Control Message Protocol for IPv6	01059–20120131, Exp	(+) Required
IPv6 Datagram Transmission	01063–20120131, Exp	(+) Required
IPv6 Private Address Space	01067–20120725, Pro	Required
VDB IPv6 Address Translation	01070–20120725, Pro	Required
IPv6 Dynamic Host Addressing	01072–20120131, Exp	(+) Required
MLD Snooping in IPv6	01060–20120131, Exp	(+) Recommended

(+) Experimental specifications are not included in compliance tests.

7.3.1.7 Switch Management Interface <VT50902-V1.7>

The table below identifies the specifications for the switch management interface.

Table 48. Switch Management Interface <VT50902-V1.7>

Spec Title	Spec Tag	Applicability
Generic Infrastructure Component Management Interface	12006–20180215, Pro	Required

7.3.2 Router <VT51000-V1.9>

The table below identifies the interfaces for the router component type.

Table 49. Router <VT51000-V1.9>

Spec Title	Spec Tag	Applicability
Router Data Interface	VT51001-V1.9	Required
Router Management Interface	VT51002-V1.7	Required
SNMP-Based Health Publishing Interface	VT59903-V1.6	Required
VDB Connector, Signaling, and Physical Media Interface	VT52303-V1.2	Recommended
Router IPv6 Data Interface	VT51003-V1.9	Recommended

7.3.2.1 Router Data Interface <VT51001-V1.9>

The tables below identify the specifications for the router data interface.

Table 50. Router Data Interface <VT51001-V1.9>

Spec Title	Spec Tag	Applicability
Data Interface Physical and Data Link Layer Specifications	VT59904-V1.9	Required
Router Data Interface Network Layer and Higher Specifications	VT51001.T4-V1.7	Required

7.3.2.2 Router IPv6 Data Interface <VT51003-V1.9>

The table below identifies the specifications for the router ipv6 data interface.

Table 51. Router IPv6 Data Interface <VT51003-V1.9>

Spec Title	Spec Tag	Applicability
Data Interface Physical and Data Link Layer Specifications	VT59904-V1.9	Required
Router IPv6 Data Interface Network Layer and Higher Specifications	VT51001.T8-V1.7	Required

7.3.2.3 Router Data Interface Network Layer and Higher Specifications <VT51001.T4-V1.7>

The table below identifies the specifications for the router data interface network layer and higher specifications.

Table 52. Router Data Interface Network Layer and Higher Specifications <VT51001.T4-V1.7>

Spec Title	Spec Tag	Applicability
Router Data Interface IPv4-Specific Specifications	VT51001.T5-V1.7	Required
Transmission Control Protocol	01026–20120725, Pro	Required
User Datagram Protocol	01027–20120725, Pro	Required
VDB Subnetting Restrictions	01029–20120725, Pro	Required
Static Host Addressing	01061–20120725, Pro	Required
Protocol Independent Multicast - Sparse Mode (PIM-SM)	01073–20120725, Pro	Required
QoS Mapping to Components	01036–20120725, Pro	Required
DiffServ	01037–20120725, Pro	Recommended
DiffServ Code Point (DSCP) Assignments	01039–20120725, Pro	Recommended
Protocol Independent Multicast - Dense Mode (PIM-DM)	01074–20120725, Pro	Recommended
PTP End-to-End Transparent Clock Mode for Infrastructure Components	02008–20180215, Exp	(+) Optional

(+) Experimental specifications are not included in compliance tests.

7.3.2.4 Router IPv6 Data Interface Network Layer and Higher Specifications <VT51001.T8-V1.7>

The table below identifies the specifications for the router ipv6 data interface network layer and higher specifications.

Table 53. Router IPv6 Data Interface Network Layer and Higher Specifications <VT51001.T8-V1.7>

Spec Title	Spec Tag	Applicability
Router Data Interface IPv6-Specific Specifications	VT51001.T6-V1.7	Recommended
Transmission Control Protocol	01026–20120725, Pro	Required
User Datagram Protocol	01027–20120725, Pro	Required
VDB Subnetting Restrictions	01029–20120725, Pro	Required

Spec Title	Spec Tag	Applicability
Static Host Addressing	01061–20120725, Pro	Required
QoS Mapping to Components	01036–20120725, Pro	Required
Protocol Independent Multicast - Sparse Mode (PIM-SM)	01073–20120725, Pro	Required
DiffServ	01037–20120725, Pro	Recommended
DiffServ Code Point (DSCP) Assignments	01039–20120725, Pro	Recommended
Protocol Independent Multicast - Dense Mode (PIM-DM)	01074–20120725, Pro	Recommended
PTP End-to-End Transparent Clock Mode for Infrastructure Components	02008–20180215, Exp	(+) Optional

(+) Experimental specifications are not included in compliance tests.

7.3.2.5 Router Data Interface IPv4-Specific Specifications <VT51001.T5-V1.7>

The table below identifies the specifications for the router data interface ipv4-specific specifications.

Table 54. Router Data Interface IPv4-Specific Specifications <VT51001.T5-V1.7>

Spec Title	Spec Tag	Applicability
IPv4 Datagram Transmission	01062–20120725, Pro	Required
Internet Protocol Version 4 Specification	01018–20120725, Pro	Required
Internet Control Message Protocol for IPv4	01021–20120725, Pro	Required
Classless Inter-Domain Routing and Subnetting for IPv4	01054–20120725, Pro	Required
Path MTU Discovery for IPv4	01064–20180215, Pro	Required
IPv4 Private Address Space	01066–20120725, Pro	Required
IPv4 Network Address Translation	01068–20120725, Pro	Required
VDB IPv4 Address Translation	01069–20120725, Pro	Required
IPv4 Dynamic Host Addressing	01071–20120725, Pro	Required
Internet Group Management Protocol (IGMP)	01075–20120725, Pro	Required
IPv4 Static Routes	01086–20180215, Pro	Required
IGMP Forwarding Proxy	01087–20180215, Pro	Required
RIP	01088–20180215, Pro	Required
OSPF Version 2	01089–20180215, Pro	Required
BGP	01090–20180215, Pro	Recommended

7.3.2.6 Router Data Interface IPv6-Specific Specifications <VT51001.T6-V1.7>

The table below identifies the specifications for the router data interface ipv6-specific specifications.

Table 55. Router Data Interface IPv6-Specific Specifications <VT51001.T6-V1.7>

Spec Title	Spec Tag	Applicability
Internet Protocol Version 6 Specification	01019–20110729, Exp	(+) Required
Internet Control Message Protocol for IPv6	01059–20120131, Exp	(+) Required
IPv6 Datagram Transmission	01063–20120131, Exp	(+) Required
IPv6 Private Address Space	01067–20120725, Pro	Required
VDB IPv6 Address Translation	01070–20120725, Pro	Required
IPv6 Dynamic Host Addressing	01072–20120131, Exp	(+) Required
Multicast Listener Discovery (MLD)	01076–20140630, Pro	Required
Subnetting for IPv6	01082–20120131, Exp	(+) Required
BGP	01090–20180215, Pro	Recommended
IPv6 Static Routes	01091–20180215, Exp	(+) Required
MLD Forwarding Proxy	01092–20180215, Exp	(+) Required
RIPng	01093–20180215, Exp	(+) Required
OSPF for IPv6	01094–20180215, Exp	(+) Required

(+) Experimental specifications are not included in compliance tests.

7.3.2.7 Router Management Interface <VT51002-V1.7>

The table below identifies the specifications for the router management interface.

Table 56. Router Management Interface <VT51002-V1.7>

Spec Title	Spec Tag	Applicability
Generic Infrastructure Component Management Interface	12006–20180215, Pro	Required

7.4 Information Assurance Component Types

7.4.1 Data Signing Service <VT52700-V1.9>

The table below identifies the interfaces for the data signing service component type.

Table 57. Data Signing Service <VT52700-V1.9>

Spec Title	Spec Tag	Applicability
Data Signing Service Management Interface	VT52701-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.4.1.1 Data Signing Service Management Interface <VT52701-V1.9>

The table below identifies the specifications for the data signing service management interface.

Table 58. Data Signing Service Management Interface <VT52701-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Data Signing Service Management Specification	14025–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Required
Data Authentication for Management Interfaces	10019–20150331, Pro	Recommended
Data Signing Service Configuration Description	30032–20210409, Pro	Recommended
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

7.4.2 Signed Data Verification Service <VT52800-V1.9>

The table below identifies the interfaces for the signed data verification service component type.

Table 59. Signed Data Verification Service <VT52800-V1.9>

Spec Title	Spec Tag	Applicability
Signed Data Verification Service Management Interface	VT52801-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.4.2.1 Signed Data Verification Service Management Interface <VT52801-V1.9>

The table below identifies the specifications for the signed data verification service management interface.

Table 60. Signed Data Verification Service Management Interface <VT52801-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Signed Data Verification Service Management Specification	14026–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Recommended
Signed Data Verification Service Configuration Description	30033–20210409, Pro	Recommended
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

7.4.3 Data at Rest Encryptor <VT52400-V1.9>

The table below identifies the interfaces for the data at rest (DAR) encryptor component type.

Table 61. Data at Rest Encryptor <VT52400-V1.9>

Spec Title	Spec Tag	Applicability
DAR Encryptor Management Interface	VT52402-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.4.3.1 DAR Encryptor Management Interface <VT52402-V1.9>

The table below identifies the specifications for the DAR encryptor management interface.

Table 62. DAR Encryptor Management Interface <VT52402-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
DAR Encryptor Management Specification	14016–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Recommended
Data at Rest Encryptor Configuration Description	30031–20210409, Pro	Recommended
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

7.4.4 Host-Based Antivirus <VT52500-V1.9>

The table below identifies the interfaces for the host-based antivirus component type.

Table 63. Host-Based Antivirus <VT52500-V1.9>

Spec Title	Spec Tag	Applicability
Host-Based Antivirus Management Interface	VT52502-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.4.4.1 Host-Based Antivirus Management Interface <VT52502-V1.9>

The table below identifies the specifications for the host-based antivirus management interface.

Table 64. Host-Based Antivirus Management Interface <VT52502-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Host-Based Antivirus Management Specification	14023–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Recommended

Spec Title	Spec Tag	Applicability
Host-Based Antivirus Configuration Description	30027–20210409, Pro	Recommended
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

7.4.5 Host-Based Firewall <VT51400-V1.9>

The table below identifies the interfaces for the host-based firewall component type.

Table 65. Host-Based Firewall <VT51400-V1.9>

Spec Title	Spec Tag	Applicability
Host-Based Firewall Management Interface	VT51402-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.4.5.1 Host-Based Firewall Management Interface <VT51402-V1.9>

The table below identifies the specifications for the host-based firewall management interface.

Table 66. Host-Based Firewall Management Interface <VT51402-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Host-Based Firewall Management Specification	14020–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Recommended
Host-Based Firewall Configuration Description	30028–20210409, Pro	Recommended
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

7.4.6 Inline Network Encryptor <VT51500-V1.6>

The table below identifies the interfaces for the inline network encryptor (INE) component type.

Table 67. Inline Network Encryptor <VT51500-V1.6>

Spec Title	Spec Tag	Applicability
INE Management Interface	VT51502-V1.6	Required
VDB Connector, Signaling, and Physical Media Interface	VT52303-V1.2	Recommended

7.4.6.1 INE Management Interface <VT51502-V1.6>

The table below identifies the specifications for the INE management interface.

Table 68. INE Management Interface <VT51502-V1.6>

Spec Title	Spec Tag	Applicability
INE Management Specification	14002–20140630, Pro (1)	Required

(1) This VICTORY specification contains technical data marked as Distribution C as defined by DoDM 5200.01 and "For Official Use Only (FOUO)." As a result, this specification is provided in a separate document.

7.4.7 Intrusion Detection and Prevention System <VT51600-V1.9>

The table below identifies the interfaces for the intrusion detection and prevention system (IDPS) component type.

Table 69. Intrusion Detection and Prevention System <VT51600-V1.9>

Spec Title	Spec Tag	Applicability
IDPS Management Interface	VT51602-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.4.7.1 IDPS Management Interface <VT51602-V1.9>

The table below identifies the specifications for the IDPS management interface.

Table 70. IDPS Management Interface <VT51602-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
IDPS Management Specification	14006–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Recommended
Intrusion Detection and Prevention System Configuration Description	30030–20210409, Pro	Recommended
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

7.4.8 Network Firewall <VT51300-V1.9>

The table below identifies the interfaces for the network firewall component type.

Table 71. Network Firewall <VT51300-V1.9>

Spec Title	Spec Tag	Applicability
Network Firewall Management Interface	VT51302-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.4.8.1 Network Firewall Management Interface <VT51302-V1.9>

The table below identifies the specifications for the network firewall management interface.

Table 72. Network Firewall Management Interface <VT51302-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Device Information Management Parameters	10009–20210409, Pro	Optional
Network Firewall Management Specification	14003–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Recommended
Network Firewall Configuration Description	30029–20210409, Pro	Recommended
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

7.5 End Node Component Types

7.5.1 Shared Processing Unit <VT51900-V1.9>

The table below identifies the interfaces for the shared processing unit (SPU) component type.

Table 73. Shared Processing Unit <VT51900-V1.9>

Spec Title	Spec Tag	Applicability
Generic End Node Data Interface	VT52101-V1.9	Required
Shared Processing Unit Management Interface	VT51902-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
VDB Connector, Signaling, and Physical Media Interface	VT52303-V1.2	Recommended
Shared Processing Unit Platform API Headless Profile	VT51912-V1.9	Required
Shared Processing Unit Platform API Display Profile	VT51913-V1.9	Optional
Direct Attached Storage Encryption Management Interface	VT51908-V1.9	Optional
Auto-Discovery Interface	VT59930-V1.6	Optional
Generic End Node IPv6 Data Interface	VT52103-V1.9	Recommended

7.5.1.1 Shared Processing Unit Management Interface <VT51902-V1.9>

The table below identifies the specifications for the shared processing unit management interface.

Table 74. Shared Processing Unit Management Interface <VT51902-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required

Spec Title	Spec Tag	Applicability
Generic End Node Management Parameters	13003–20210409, Pro	Required
Shared Processing Unit Management Parameters	13006–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Recommended
Configuration Management Parameters	10028–20210409, Pro	Required
Shared Processing Unit Configuration Description	30042–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

7.5.1.2 Shared Processing Unit Platform API Headless Profile <VT51912-V1.9>

The table below identifies the specifications for the shared processing unit platform application programming interface headless profile.

Table 75. Shared Processing Unit Platform API Headless Profile <VT51912-V1.9>

Spec Title	Spec Tag	Applicability
Shared Processing Unit Low-Level Platform API	VT51909-V1.9	Required
Shared Processing Unit Mid-Level Platform API	VT51910-V1.8	Recommended

7.5.1.3 Shared Processing Unit Platform API Display Profile <VT51913-V1.9>

The table below identifies the specifications for the shared processing unit platform application programming interface display profile.

Table 76. Shared Processing Unit Platform API Display Profile <VT51913-V1.9>

Spec Title	Spec Tag	Applicability
Qt Requirements for the Display Unit SPU API Profile	13011–20210409, Pro	Required

7.5.1.4 Shared Processing Unit Low-Level Platform API <VT51909-V1.9>

The table below identifies the specifications for the shared processing unit low-level platform application programming interface.

Table 77. Shared Processing Unit Low-Level Platform API <VT51909-V1.9>

Spec Title	Spec Tag	Applicability
Qt Requirements for the Headless SPU API Profile	13010–20210409, Pro	Required
libBoost	13012–20210409, Pro	Required
POSIX	13009–20210409, Exp	(+) Required

(+) Experimental specifications are not included in compliance tests.

7.5.1.5 Shared Processing Unit Mid-Level Platform API <VT51910-V1.8>

The table below identifies the specifications for the shared processing unit mid-level platform application programming interface.

Table 78. Shared Processing Unit Mid-Level Platform API <VT51910-V1.8>

Spec Title	Spec Tag	Applicability
SPU Mid-Level API	13005–20190418, Pro	Recommended

7.5.2 Direct Attached Storage Encryption Management Interface <VT51908-V1.9>

The table below identifies the specifications for the direct attached storage encryption management interface.

Table 79. Direct Attached Storage Encryption Management Interface <VT51908-V1.9>

Spec Title	Spec Tag	Applicability
DAR Encryptor Management Specification	14016–20210409, Pro	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required

7.5.3 Network Attached Storage <VT52000-V1.9>

The table below identifies the interfaces for the network attached storage (NAS) component type.

Table 80. Network Attached Storage <VT52000-V1.9>

Spec Title	Spec Tag	Applicability
Network Attached Storage Interface	VT52001-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.5.3.1 Network Attached Storage Interface <VT52001-V1.9>

The table below identifies the specifications for the network attached storage interface (data and management are combined in these protocols).

Table 81. Network Attached Storage Interface <VT52001-V1.9>

Spec Title	Spec Tag	Applicability
WebDAV	20028–20120731, Pro	Required
NFS	20029–20120731, Pro	Recommended
Network Attached Storage Configuration Description	30041–20210409, Pro	Required

7.5.4 Generic End Node <VT52100-V1.9>

The table below identifies the interfaces for the generic end node component type.

Table 82. Generic End Node <VT52100-V1.9>

Spec Title	Spec Tag	Applicability
Generic End Node Data Interface	VT52101-V1.9	Required
Generic End Node Management Interface	VT52102-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Time Synchronization Interface	VT50001-V1.7	Required
VDB Connector, Signaling, and Physical Media Interface	VT52303-V1.2	Recommended
Auto-Discovery Interface	VT59930-V1.6	Optional
Generic End Node IPv6 Data Interface	VT52103-V1.9	Recommended

7.5.4.1 Generic End Node Data Interface <VT52101-V1.9>

The tables below identify the specifications for the generic end node data interface.

Table 83. Generic End Node Data Interface <VT52101-V1.9>

Spec Title	Spec Tag	Applicability
Data Interface Physical and Data Link Layer Specifications	VT59904-V1.9	Required
Generic End Node Data Interface IPv4-Specific Specifications	VT52101.T1-V1.7	Required
Transmission Control Protocol	01026–20120725, Pro	Required
User Datagram Protocol	01027–20120725, Pro	Required
VDB Subnetting Restrictions	01029–20120725, Pro	Required
Network Port Assignment	01035–20120725, Pro	Required
Static Host Addressing	01061–20120725, Pro	Required
DiffServ	01037–20120725, Pro	Recommended
DiffServ Code Point (DSCP) Assignments	01039–20120725, Pro	Recommended

7.5.4.2 Generic End Node IPv6 Data Interface <VT52103-V1.9>

The table below identifies the specifications for the generic end node ipv6 data interface.

Table 84. Generic End Node IPv6 Data Interface <VT52103-V1.9>

Spec Title	Spec Tag	Applicability
Data Interface Physical and Data Link Layer Specifications	VT59904-V1.9	Required
Generic End Node Data Interface IPv6-Specific Specifications	VT52101.T2-V1.7	Recommended
Transmission Control Protocol	01026–20120725, Pro	Required
User Datagram Protocol	01027–20120725, Pro	Required
VDB Subnetting Restrictions	01029–20120725, Pro	Required

Spec Title	Spec Tag	Applicability
Network Port Assignment	01035–20120725, Pro	Required
Static Host Addressing	01061–20120725, Pro	Required
DiffServ	01037–20120725, Pro	Recommended
DiffServ Code Point (DSCP) Assignments	01039–20120725, Pro	Recommended
Path MTU Discovery for IPv6	01065–20120725, Pro	Optional

7.5.4.2.1 Generic End Node Data Interface IPv4-Specific Specifications <VT52101.T1-V1.7>

The table below identifies the specifications for the generic end node data interface ipv4-specific specifications.

Table 85. Generic End Node Data Interface IPv4-Specific Specifications <VT52101.T1-V1.7>

Spec Title	Spec Tag	Applicability
Internet Protocol Version 4 Specification	01018–20120725, Pro	Required
Internet Control Message Protocol for IPv4	01021–20120725, Pro	Required
VDB Node Identification for IPv4	01042–20120725, Pro	Required
Classless Inter-Domain Routing and Subnetting for IPv4	01054–20120725, Pro	Required
IPv4 Datagram Transmission	01062–20120725, Pro	Required
IPv4 Private Address Space	01066–20120725, Pro	Required
VDB IPv4 Address Translation	01069–20120725, Pro	Required
IPv4 Dynamic Host Addressing	01071–20120725, Pro	Required
Internet Group Management Protocol (IGMP)	01075–20120725, Pro	Required
IP Layer Security	01025–20180215, Pro	Required

7.5.4.2.2 Generic End Node Data Interface IPv6-Specific Specifications <VT52101.T2-V1.7>

The table below identifies the specifications for the generic end node data interface ipv6-specific specifications.

Table 86. Generic End Node Data Interface IPv6-Specific Specifications <VT52101.T2-V1.7>

Spec Title	Spec Tag	Applicability
Internet Protocol Version 6 Specification	01019–20110729, Exp	(+) Required
Internet Control Message Protocol for IPv6	01059–20120131, Exp	(+) Required
IPv6 Datagram Transmission	01063–20120131, Exp	(+) Required
IPv6 Private Address Space	01067–20120725, Pro	Required
VDB IPv6 Address Translation	01070–20120725, Pro	Required

Spec Title	Spec Tag	Applicability
IPv6 Dynamic Host Addressing	01072–20120131, Exp	(+) Required
Multicast Listener Discovery (MLD)	01076–20140630, Pro	Required
Subnetting for IPv6	01082–20120131, Exp	(+) Required
VDB Node Identification for IPv6	01083–20120131, Exp	(+) Required
IP Layer Security	01025–20180215, Pro	Required

(+) Experimental specifications are not included in compliance tests.

7.5.4.3 Generic End Node Management Interface <VT52102-V1.9>

The table below identifies the specifications for the generic end node management interface.

Table 87. Generic End Node Management Interface <VT52102-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Generic End Node Management Parameters	13003–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Recommended
Configuration Management Parameters	10028–20210409, Pro	Required
Generic End Node Configuration Description	30040–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

7.6 System Management Component Types

7.6.1 VDB Management Service <VT52200-V1.9>

The table below identifies the interfaces for the VDB management service component type.

Table 88. VDB Management Service <VT52200-V1.9>

Spec Title	Spec Tag	Applicability
VDB System Management Interface	VT52201-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

7.6.1.1 VDB System Management Interface <VT52201-V1.9>

The table below identifies the specifications for the VDB system management interface.

Table 89. VDB System Management Interface <VT52201-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Health Level Management Parameters	10030–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Recommended
Configuration Management Parameters for the VDB Management Service	10029–20210409, Pro	Required
VDB Management Service Configuration Description	30043–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

7.6.2 VICTORY Configuration Language Instance Document <VT55000-V1.9>

A VICTORY Configuration Language Instance Document (VCL ID) is an Extensible Markup Language (XML) document that describes the configuration settings for a VICTORY implementation. It contains configuration descriptions for each of the component types that are implemented in a system. Some of these parameters are merely descriptive, while others are prescriptive. For prescriptive parameters, values of the parameters in the file can affect change in the configuration of the system when the configuration file is applied using the VDB System Management Interface of the VDB Management Service. Descriptive parameters provide the current parameter values, but the values in the VCL ID cannot affect change on the system configuration.

A VCL ID is made up of a set of component configuration descriptions. These configuration descriptions are grouped into four classes:

- VDB Configuration Description
- Information Assurance Configuration Description
- C4ISR/EW System Configuration Description
- Platform System Configuration Description

A compliant VCL ID *should* contain configuration descriptions for each of these classes, and *should* contain configuration descriptions for each VICTORY component type that is implemented.

Table 90. VICTORY Configuration Language Instance Document <VT55000-V1.9>

Spec Title	Spec Tag	Applicability
VCL Instance Document Requirements	30045–20131031, Pro	Recommended
Common Component Configuration Description Content	30047–20210409, Pro	Recommended
Configuration Type	30048–20131031, Pro	Recommended
VDB Configuration Description	VT55100-V1.9	Recommended
Information Assurance Configuration Description	VT55200-V1.9	Recommended
C4ISR/EW System Configuration Description	VT55300-V1.9	Recommended
Platform System Configuration Description	VT55400-V1.9	Recommended

7.6.2.1 VDB Configuration Description <VT55100-V1.9>

A compliant VDB Configuration Description *should* contain configuration descriptions for each of the VICTORY component types shown in Table 91 that is implemented in the system being described.

Table 91. VDB Configuration Description <VT55100-V1.9>

Spec Title	Spec Tag	Applicability
Generic End Node Configuration Description	30040–20210409, Pro	Recommended
Network Attached Storage Configuration Description	30041–20210409, Pro	Recommended
Shared Processing Unit Configuration Description	30042–20210409, Pro	Recommended
VDB Management Service Configuration Description	30043–20210409, Pro	Recommended

7.6.2.2 Information Assurance Configuration Description <VT55200-V1.9>

A compliant Information Assurance Configuration Description *should* contain configuration descriptions for each of the VICTORY component types shown in Table 92 that is implemented in the system being described.

Table 92. Information Assurance Configuration Description <VT55200-V1.9>

Spec Title	Spec Tag	Applicability
Attribute Store Service Configuration Description	30022–20210409, Pro	Recommended
Policy Decision Service Configuration Description	30023–20210409, Pro	Recommended
Policy Enforcement Service Configuration Description	30024–20210409, Pro	Recommended
Policy Store Service Configuration Description	30025–20210409, Pro	Recommended
Authentication Service Configuration Description	30026–20210409, Pro	Recommended
Host-Based Antivirus Configuration Description	30027–20210409, Pro	Recommended
Host-Based Firewall Configuration Description	30028–20210409, Pro	Recommended
Network Firewall Configuration Description	30029–20210409, Pro	Recommended
Intrusion Detection and Prevention System Configuration Description	30030–20210409, Pro	Recommended
Data at Rest Encryptor Configuration Description	30031–20210409, Pro	Recommended
Data Signing Service Configuration Description	30032–20210409, Pro	Recommended
Signed Data Verification Service Configuration Description	30033–20210409, Pro	Recommended

7.6.2.3 C4ISR/EW System Configuration Description <VT55300-V1.9>

A compliant C4ISR/EW System Configuration Description *should* contain configuration descriptions for each of the VICTORY component types shown in Table 93 that is implemented in the system being described.

Table 93. C4ISR/EW System Configuration Description <VT55300-V1.9>

Spec Title	Spec Tag	Applicability
Position Service Configuration Description	30005–20210409, Pro	Recommended
Orientation Service Configuration Description	30006–20210409, Pro	Recommended
Direction of Travel Service Configuration Description	30007–20210409, Pro	Recommended

Spec Title	Spec Tag	Applicability
Streaming Audio Data Source Configuration Description	30008–20210409, Pro	Recommended
Single-Instance Audio Data Source Configuration Description	30009–20210409, Pro	Recommended
Intercom Configuration Description	30010–20210409, Pro	Recommended
Video or Image Sensor Configuration Description	30012–20210409, Pro	Recommended
Video or Image Encoder Configuration Description	30013–20210409, Pro	Recommended
Camera Gimbal Configuration Description	30014–20210409, Pro	Recommended
Threat Detection and Reporting Service Configuration Description	30015–20210409, Pro	Recommended
Mission Recorder Configuration Description	30016–20210409, Pro	Recommended
Assured PNT Configuration Description	30036–20210409, Pro	Recommended
Data Logger Configuration Description	30049–20210409, Pro	Recommended
Legacy Radio Configuration Description	30053–20210409, Pro	Recommended

7.6.2.4 Platform System Configuration Description <VT55400-V1.9>

A compliant Platform System Configuration Description *should* contain configuration descriptions for each of the VICTORY component types shown in Table 94 that is implemented in the system being described.

Table 94. Platform System Configuration Description <VT55400-V1.9>

Spec Title	Spec Tag	Applicability
Electronic Warfare Device Configuration Description	30017–20210409, Pro (1)	Recommended
GPS Receiver Configuration Description	30018–20210409, Pro (2)	Recommended
Automotive System Configuration Description	30019–20210409, Pro	Recommended
Power Distribution System Configuration Description	30020–20210409, Pro	Recommended
Remote Weapon Station Configuration Description	30021–20210409, Pro	Recommended
Vehicle Configuration Service Configuration Description	30050–20210409, Pro	Recommended
Active Protection System Configuration Description	30051–20210409, Exp (2)	(+) Recommended
Live Training Transformation Adapter Configuration Description	30052–20210409, Pro	Recommended

(+) Experimental specifications are not included in compliance tests.

(1) This VICTORY specification contains technical data marked as Distribution C as defined by DoDM 5200.01. As a result, this specification is provided in a separate document.

(2) This VICTORY specification contains technical data whose export is restricted by the Arms Export Control Act (Title 22, U.S.C., Sec 2751 et seq.) or the Export Administration Act of 1979, as amended, Title 50, U.S.C., 2401, et seq. As a result, this specification is provided in a separate document.

8. C4ISR/EW SYSTEM AND COMPONENT TYPE SPECIFICATIONS

8.1 Audio Communications Component Types

8.1.1 Streaming Audio Data Source <VT60001-V1.9>

The table below identifies the interfaces for the streaming audio data source component type.

Table 95. Streaming Audio Data Source <VT60001-V1.9>

Spec Title	Spec Tag	Applicability
Streaming Audio Data Interface	VT60004-V1.6	Required
Streaming Audio Management Interface	VT60005-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

8.1.1.1 Streaming Audio Data Interface <VT60004-V1.6>

The table below identifies the specifications for the streaming audio data interface.

Table 96. Streaming Audio Data Interface <VT60004-V1.6>

Spec Title	Spec Tag	Applicability
Streaming Audio Data Specification	20016–20131031, Pro	Required

8.1.1.2 Streaming Audio Management Interface <VT60005-V1.9>

The table below identifies the specifications for the streaming audio management interface.

Table 97. Streaming Audio Management Interface <VT60005-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Streaming Audio Data Management Parameters	15007–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Streaming Audio Data Source Configuration Description	30008–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

8.1.2 Single-Instance Audio Data Source <VT60000-V1.9>

The table below identifies the interfaces for the single-instance audio data source component type.

Table 98. Single-Instance Audio Data Source <VT60000-V1.9>

Spec Title	Spec Tag	Applicability
Single-Instance Audio Data Interface	VT60002-V1.6	Required
Single-Instance Audio Management Interface	VT60003-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

8.1.2.1 Single-Instance Audio Data Interface <VT60002-V1.6>

The table below identifies the specifications for the single-instance audio data interface.

Table 99. Single-Instance Audio Data Interface <VT60002-V1.6>

Spec Title	Spec Tag	Applicability
Single-Instance Audio Data Specification	20067–20131031, Pro	Required

8.1.2.2 Single-Instance Audio Management Interface <VT60003-V1.9>

The table below identifies the specifications for the single-instance audio management interface.

Table 100. Single-Instance Audio Management Interface <VT60003-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Single-Instance Audio Data Management Parameters	15018–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Single-Instance Audio Data Source Configuration Description	30009–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

8.1.3 Intercom <VT60100-V1.9>

The table below identifies the interfaces for the intercom component type.

Table 101. Intercom <VT60100-V1.9>

Spec Title	Spec Tag	Applicability
Intercom Management Interface	VT60101-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

8.1.3.1 Intercom Management Interface <VT60101-V1.9>

The table below identifies the specifications for the intercom management interface.

Table 102. Intercom Management Interface <VT60101-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Device Information Management Parameters	10009–20210409, Pro	Required
Intercom Management Parameters	15019–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Intercom Configuration Description	30010–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

8.1.4 Legacy Radio <VT60200-V1.9>

The table below identifies the interfaces for the legacy radio component type.

NOTE: *The purpose of the Legacy Radio component type is not one of radio configuration, but rather its purpose is to provide simple information about the radio, e.g., transmit power, selected preset, waveform, etc. This information could be displayed to the occupants of the vehicle and would allow basic control of the radio, thereby providing a common and simple view of any radio regardless of the complexity of the radio. This component type is not specific to any single radio or even family of radios; it has been specified so that it can be used across a large number of radios.*

Table 103. Legacy Radio <VT60200-V1.9>

Spec Title	Spec Tag	Applicability
Legacy Radio Management Interface	VT60202-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

8.1.4.1 Legacy Radio Management Interface <VT60202-V1.9>

The table below identifies the specifications for the legacy radio management interface.

Table 104. Legacy Radio Management Interface <VT60202-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Device Information Management Parameters	10009–20210409, Pro	Required
Legacy Radio Device Management Parameters	15025–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended

Spec Title	Spec Tag	Applicability
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Legacy Radio Configuration Description	30053–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

8.2 Video and Imagery Situational Awareness Component Types

8.2.1 Video or Image Sensor <VT60300-V1.9>

The table below identifies the interfaces for the video or image sensor component type.

Table 105. Video or Image Sensor <VT60300-V1.9>

Spec Title	Spec Tag	Applicability
Video or Image Sensor Management Interface	VT60301-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

8.2.1.1 Video or Image Sensor Management Interface <VT60301-V1.9>

The table below identifies the specifications for the video or image sensor management interface.

Table 106. Video or Image Sensor Management Interface <VT60301-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Device Information Management Parameters	10009–20210409, Pro	Required
Video or Image Sensor Management Parameters	15008–20210409, Pro	Required
IR Video or Image Sensor Management Parameters	15027–20210409, Pro	Optional*
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Video or Image Sensor Configuration Description	30012–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

* Required if IR camera; optional otherwise

8.2.2 Video or Image Encoder <VT60400-V1.9>

The table below identifies the interfaces for the video or image encoder component type.

Table 107. Video or Image Encoder <VT60400-V1.9>

Spec Title	Spec Tag	Applicability
Video Streaming Data Interface	VT60401-V1.6	Required
Video Encoder Management Interface	VT60403-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

8.2.2.1 Video Streaming Data Interface <VT60401-V1.6>

The table below identifies the specifications for the video streaming data interface.

Table 108. Video Streaming Data Interface <VT60401-V1.6>

Spec Title	Spec Tag	Applicability
Network Port Assignment	01035–20120725, Pro	Required
Standard Definition Streaming Video on the VDB	20017–20150331, Pro	Required

8.2.2.2 Video Encoder Management Interface <VT60403-V1.9>

The table below identifies the specifications for the video encoder management interface.

Table 109. Video Encoder Management Interface <VT60403-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Device Information Management Parameters	10009–20210409, Pro	Optional
Video Encoder Management Parameters	15012–20210409, Pro	Required
Video Data Stream Management Parameters	15010–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Video or Image Encoder Configuration Description	30013–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

8.2.3 Camera Gimbal <VT60500-V1.9>

The table below identifies the interfaces for the camera gimbal component type.

Table 110. Camera Gimbal <VT60500-V1.9>

Spec Title	Spec Tag	Applicability
Camera Gimbal Data Interface	VT60501-V1.7	Required

Spec Title	Spec Tag	Applicability
Camera Gimbal Management Interface	VT60502-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

8.2.3.1 Camera Gimbal Data Interface <VT60501-V1.7>

The table below identifies the specifications for the camera gimbal data interface.

Table 111. Camera Gimbal Data Interface <VT60501-V1.7>

Spec Title	Spec Tag	Applicability
Common Data Interface Specifications	VT59901-V1.6	Required
Camera Orientation Data Specification	20018–20180215, Pro	Required
Home Position Data Specification	20082–20180215, Pro	Required

8.2.3.2 Camera Gimbal Management Interface <VT60502-V1.9>

The table below identifies the specifications for the camera gimbal management interface.

Table 112. Camera Gimbal Management Interface <VT60502-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Camera Gimbal Data Management Parameters	15028–20210409, Pro	Required
Device Information Management Parameters	10009–20210409, Pro	Required
Camera Gimbal Device Management Parameters	15020–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Camera Gimbal Configuration Description	30014–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

8.3 Recording and Logging Component Types

8.3.1 Mission Recorder <VT60700-V1.9>

The table below identifies the interfaces for the mission recorder component type.

Table 113. Mission Recorder <VT60700-V1.9>

Spec Title	Spec Tag	Applicability
Mission Recorder Core Functionality	03001–20131031, Pro	Required
Mission Recorder Device Management Interface	VT60702-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

8.3.1.1 Mission Recorder Device Management Interface <VT60702-V1.9>

The table below identifies the specifications for the mission recorder device management interface.

Table 114. Mission Recorder Device Management Interface <VT60702-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Device Information Management Parameters	10009–20210409, Pro	Required
Mission Recorder Device Management Parameters	15015–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Mission Recorder Configuration Description	30016–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

8.4 Data Logger Component Types

8.4.1 Data Logger <VT62000-V1.9>

The table below identifies the interfaces for the data logger component type.

Table 115. Data Logger <VT62000-V1.9>

Spec Title	Spec Tag	Applicability
Data Logger Data Interface	VT62001-V1.9	Required
Data Logger Management Interface	VT62002-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

8.4.1.1 Data Logger Data Interface <VT62001-V1.9>

The table below identifies the specifications for the data logger data interface.

Table 116. Data Logger Data Interface <VT62001-V1.9>

Spec Title	Spec Tag	Applicability
Common Data Interface Specifications	VT59901-V1.6	Required
Data Logger Data Specification	20076–20210409, Pro	Required

8.4.1.2 Data Logger Management Interface <VT62002-V1.9>

The table below identifies the specifications for the data logger management interface.

Table 117. Data Logger Management Interface <VT62002-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
VICTORY Data Management Parameters	10008–20210409, Pro	Required
Data Logger Management Parameters	15021–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Data Logger Configuration Description	30049–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

8.5 Extra-Vehicle Network Interface Component Types

8.5.1 Data Radio <VT60800-V1.9>

The table below identifies the interfaces for the data radio component type.

Table 118. Data Radio <VT60800-V1.9>

Spec Title	Spec Tag	Applicability
Data Radio Data Interface	VT60801-V1.6	Required
Data Radio Management Interface	VT60802-V1.9	Required
VDB Connector, Signaling, and Physical Media Interface	VT52303-V1.2	Required

8.5.1.1 Data Radio Data Interface <VT60801-V1.6>

The table below identifies the specifications for the data radio data interface.

Table 119. Data Radio Data Interface <VT60801-V1.6>

Spec Title	Spec Tag	Applicability
Data Radio Data Interface Physical and Data Link Layer Specifications	VT60803-V1.6	Required

Spec Title	Spec Tag	Applicability
Data Radio Data Interface Network Layer and Higher Specifications	VT60804-V1.6	Required

8.5.1.2 Data Radio Data Interface Physical and Data Link Layer Specifications <VT60803-V1.6>

The table below identifies the physical and data link layer specifications for the data radio data interface.

Table 120. Data Radio Data Interface Physical and Data Link Layer Specifications <VT60803-V1.6>

Spec Title	Spec Tag	Applicability
Frame Structure	01013–20121113, Pro	Required
Media Access Control	01014–20121113, Pro	Required
Logical Link Control	01015–20120725, Pro	Required
10BASE-T	01001–20121113, Pro	Required
100BASE-TX	01003–20121113, Pro	Required
Copper Auto-Negotiation	01011–20121113, Pro	Required

8.5.1.3 Data Radio Data Interface Network Layer and Higher Specifications <VT60804-V1.6>

The table below identifies the network layer and higher specifications for the data radio data interface.

Table 121. Data Radio Data Interface Network Layer and Higher Specifications <VT60804-V1.6>

Spec Title	Spec Tag	Applicability
Transmission Control Protocol	01026–20120725, Pro	Required
User Datagram Protocol	01027–20120725, Pro	Required
Static Host Addressing	01061–20120725, Pro	Required
Internet Protocol Version 4 Specification	01018–20120725, Pro	Required
Internet Control Message Protocol for IPv4	01021–20120725, Pro	Required
IPv4 Datagram Transmission	01062–20120725, Pro	Required
IPv4 Private Address Space	01066–20120725, Pro	Required
VDB IPv4 Address Translation	01069–20120725, Pro	Required
IPv4 Dynamic Host Addressing	01071–20120725, Pro	Recommended

8.5.1.4 Data Radio Management Interface <VT60802-V1.9>

The table below identifies the specifications for the data radio management interface.

Table 122. Data Radio Management Interface <VT60802-V1.9>

Spec Title	Spec Tag	Applicability
SNMP Core Specifications	10004–20150331, Pro	Required
Data Radio SNMP Specification	15024–20210409, Pro (1)	Required

(1) This VICTORY specification contains technical data whose export is restricted by the Arms Export Control Act (Title 22, U.S.C., Sec 2751 et seq.) or the Export Administration Act of 1979, as amended, Title 50, U.S.C., 2401, et seq. As a result, this specification is provided in a separate document.

8.6 Electronic Warfare Component Types

8.6.1 Electronic Warfare Device <VT60900-V1.9>

The table below identifies the interfaces for the electronic warfare device component type.

Table 123. Electronic Warfare Device <VT60900-V1.9>

Spec Title	Spec Tag	Applicability
Electronic Warfare Data Interface	VT60901-V1.9	Required
Electronic Warfare Management Interface	VT60902-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

8.6.1.1 Electronic Warfare Data Interface <VT60901-V1.9>

The table below identifies the specifications for the electronic warfare data interface.

Table 124. Electronic Warfare Data Interface <VT60901-V1.9>

Spec Title	Spec Tag	Applicability
Common Data Interface Specifications	VT59901-V1.6	Required
Electronic Warfare VICTORY Data Messages	VT60904-V1.9	Required

8.6.1.2 Electronic Warfare Management Interface <VT60902-V1.9>

The table below identifies the specifications for the electronic warfare management interface.

Table 125. Electronic Warfare Management Interface <VT60902-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Device Information Management Parameters	10009–20210409, Pro	Required
Electronic Warfare Device Management Parameters	15013–20210409, Pro (1)	Required
Electronic Warfare Data Management Parameters	15014–20210409, Pro (1)	Required

Spec Title	Spec Tag	Applicability
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Electronic Warfare Device Configuration Description	30017–20210409, Pro (1)	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

(1) This VICTORY specification contains technical data marked as Distribution C as defined by DoDM 5200.01. As a result, this specification is provided in a separate document.

8.6.1.3 Electronic Warfare VICTORY Data Messages <VT60904-V1.9>

The table below provides the list of electronic warfare VICTORY data messages.

Table 126. Electronic Warfare VICTORY Data Messages <VT60904-V1.9>

Spec Title	Spec Tag	Applicability
Electronic Warfare Line of Bearing VICTORY Data Message	20036–20210409, Pro (1)	(2) Required
Electronic Warfare Geolocation VICTORY Data Message	20037–20210409, Pro (1)	
Electronic Warfare Object Detection VICTORY Data Message	20038–20210409, Pro (1)	
Electronic Warfare Device Specific VICTORY Data Message	20039–20210409, Pro (1)	

(1) This VICTORY specification contains technical data marked as Distribution C as defined by DoDM 5200.01. As a result, this specification is provided in a separate document.

(2) One or more of 20036, 20037, 20038, 20039 is Required

8.7 C4ISR/EW Sensor Component Types

8.7.1 GPS Receiver <VT61000-V1.9>

The table below identifies the interfaces for the GPS receiver component type.

Table 127. GPS Receiver <VT61000-V1.9>

Spec Title	Spec Tag	Applicability
GPS Receiver Data Interface	VT61001-V1.9	Required
GPS Receiver Management Interface	VT61002-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

8.7.1.1 GPS Receiver Data Interface <VT61001-V1.9>

The table below identifies the specifications for the GPS receiver data interface.

Table 128. GPS Receiver Data Interface <VT61001-V1.9>

Spec Title	Spec Tag	Applicability
Common Data Interface Specifications	VT59901-V1.6	Required
GPS Receiver VICTORY Data Messages	VT61003-V1.9	Required

8.7.1.2 GPS Receiver Management Interface <VT61002-V1.9>

The table below identifies the specifications for the GPS receiver management interface.

Table 129. GPS Receiver Management Interface <VT61002-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Device Information Management Parameters	10009–20210409, Pro	Required
GPS Receiver Device Management Parameters	15016–20210409, Pro (1)	Required
GPS Receiver Data Management Parameters	15017–20210409, Pro (1)	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
GPS Receiver Configuration Description	30018–20210409, Pro (1)	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

(1) This VICTORY specification contains technical data whose export is restricted by the Arms Export Control Act (Title 22, U.S.C., Sec 2751 et seq.) or the Export Administration Act of 1979, as amended, Title 50, U.S.C., 2401, et seq. As a result, this specification is provided in a separate document.

8.7.1.3 GPS Receiver VICTORY Data Messages <VT61003-V1.9>

The table below provides the list of GPS receiver VICTORY data messages.

Table 130. GPS Receiver VICTORY Data Messages <VT61003-V1.9>

Spec Title	Spec Tag	Applicability
GPS Receiver Ionospheric Corrections VICTORY Data Message	20034–20131031, Pro (1)	(2) Required
GPS Receiver Status VICTORY Data Messages	VT61004-V1.9	
GPS Receiver Satellite Health Status VICTORY Data Messages	VT61005-V1.9	

(1) This VICTORY specification contains technical data whose export is restricted by the Arms Export Control Act (Title 22, U.S.C., Sec 2751 et seq.) or the Export Administration Act of 1979, as amended, Title 50, U.S.C., 2401, et seq. As a result, this specification is provided in a separate document.

(2) One or more of 20034, VT61004, VT61005 is Required

8.7.1.4 GPS Receiver Status VICTORY Data Messages <VT61004-V1.9>

The table below provides the list of GPS receiver status VICTORY data messages.

Table 131. GPS Receiver Status VICTORY Data Messages <VT61004-V1.9>

Spec Title	Spec Tag	Applicability
GPS Receiver Status VICTORY Data Message	20032–20210409, Pro (1)	Required
GPS Receiver Status 2 VICTORY Data Message	20085–20190418, Pro (1)	Required

(1) This VICTORY specification contains technical data whose export is restricted by the Arms Export Control Act (Title 22, U.S.C., Sec 2751 et seq.) or the Export Administration Act of 1979, as amended, Title 50, U.S.C., 2401, et seq. As a result, this specification is provided in a separate document.

8.7.1.5 GPS Receiver Satellite Health Status VICTORY Data Messages <VT61005-V1.9>

The table below provides the list of GPS receiver satellite health status VICTORY data messages.

Table 132. GPS Receiver Satellite Health Status VICTORY Data Messages <VT61005-V1.9>

Spec Title	Spec Tag	Applicability
GPS Receiver Satellite Health Status VICTORY Data Message	20035–20131031, Pro (1)	Required
GPS Receiver Satellite Health Status 2 VICTORY Data Message	20086–20210409, Pro (1)	Required

(1) This VICTORY specification contains technical data whose export is restricted by the Arms Export Control Act (Title 22, U.S.C., Sec 2751 et seq.) or the Export Administration Act of 1979, as amended, Title 50, U.S.C., 2401, et seq. As a result, this specification is provided in a separate document.

8.8 C4ISR/EW System Component Types

8.8.1 Assured PNT <VT61600-V1.9>

The table below identifies the interfaces for the assured PNT component type.

Table 133. Assured PNT <VT61600-V1.9>

Spec Title	Spec Tag	Applicability
Assured PNT Data Interface	VT61601-V1.9	Required
Assured PNT Management Interface	VT61602-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

8.8.1.1 Assured PNT Data Interface <VT61601-V1.9>

The table below identifies the specifications for the assured PNT data interface.

Table 134. Assured PNT Data Interface <VT61601-V1.9>

Spec Title	Spec Tag	Applicability
Common Data Interface Specifications	VT59901-V1.6	Required
Assured PNT VICTORY Data Messages	VT61603-V1.9	Required

8.8.1.2 Assured PNT Management Interface <VT61602-V1.9>

The table below identifies the specifications for the assured PNT management interface.

Table 135. Assured PNT Management Interface <VT61602-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Assured PNT System Management Parameters	15029–20210409, Pro (1)	Required
Assured PNT Data Management Parameters	15030–20210409, Pro (1)	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Assured PNT Configuration Description	30036–20210409, Pro (1)	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

(1) This VICTORY specification contains technical data marked as Distribution C as defined by DoDM 5200.01. As a result, this specification is provided in a separate document.

8.8.1.3 Assured PNT VICTORY Data Messages <VT61603-V1.9>

The table below provides the list of assured PNT VICTORY data messages.

Table 136. Assured PNT VICTORY Data Messages <VT61603-V1.9>

Spec Title	Spec Tag	Applicability
Assured PNT Blended Navigation VICTORY Data Message	20087–20210409, Pro (1)	(2) Required
Assured PNT Warnings VICTORY Data Message	20088–20210409, Pro (1)	
Assured PNT NAVWAR SA VICTORY Data Message	20090–20210409, Pro (1)	

(1) This VICTORY specification contains technical data marked as Distribution C as defined by DoDM 5200.01. As a result, this specification is provided in a separate document.

(2) One or more of 20087, 20088, 20090 is Required

9. PLATFORM SYSTEM TYPE SPECIFICATIONS

9.1 Automotive System Types

9.1.1 Automotive System <VT70000-V1.9>

The table below identifies the interfaces for the automotive system component type.

Table 137. Automotive System <VT70000-V1.9>

Spec Title	Spec Tag	Applicability
Automotive System Data Interface	VT70001-V1.9	Required
Automotive System Management Interface	VT70002-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

9.1.1.1 Automotive System Data Interface <VT70001-V1.9>

The table below identifies the specifications for the automotive system data interface.

Table 138. Automotive System Data Interface <VT70001-V1.9>

Spec Title	Spec Tag	Applicability
Common Data Interface Specifications	VT59901-V1.6	Required
Automotive System VICTORY Data Messages	VT70006-V1.9	Required

9.1.1.2 Automotive System Management Interface <VT70002-V1.9>

The table below identifies the specifications for the automotive system management interface.

Table 139. Automotive System Management Interface <VT70002-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Automotive System Management Parameters – V1.6	16005–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Automotive System Configuration Description	30019–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

9.1.2 Automotive System VICTORY Data Messages <VT70006-V1.9>

The table below provides the list of automotive system VICTORY data messages.

Table 140. Automotive System VICTORY Data Messages <VT70006-V1.9>

Spec Title	Spec Tag	Applicability
Automotive Ambient Conditions VICTORY Data Message	20040–20120725, Pro	(+) (1) Required
Automotive Dash Display VICTORY Data Message	20041–20120725, Pro	
Automotive Direct Lamp Control Data 1 VICTORY Data Message	20042–20120725, Pro	
Automotive Direct Lamp Control Data 2 VICTORY Data Message	20043–20120725, Pro	
Automotive Distance VICTORY Data Message	20044–20120725, Pro	
Automotive Electrical Power 1 VICTORY Data Message	20045–20120725, Pro	
Automotive Electrical Power 2 VICTORY Data Message	20046–20120725, Pro	
Automotive Electronic Transmission Controller 2 VICTORY Data Message	20047–20120725, Pro	
Automotive Engine Fluid Level and Pressure 1 VICTORY Data Message	20048–20120725, Pro	
Automotive Engine Hours and Revolutions VICTORY Data Message	20049–20120725, Pro	
Automotive Engine Speed and Load Factor VICTORY Data Message	20050–20120725, Pro	
Automotive Engine Temperature 1 VICTORY Data Message	20051–20120725, Pro	
Automotive Engine Temperature 2 VICTORY Data Message	20052–20120725, Pro	
Automotive Engine Valve VICTORY Data Message	20053–20120725, Pro	
Automotive Fuel Consumption VICTORY Data Message	20054–20120725, Pro	
Automotive Fuel Economy VICTORY Data Message	20055–20120725, Pro	
Automotive Active Diagnostic Trouble Codes VICTORY Data Message	20056–20210409, Exp	
Automotive Alternator Temperature VICTORY Data Message	20057–20140630, Pro	
Automotive Battery Temperature VICTORY Data Message	20058–20140630, Pro	
Automotive Cab Message 1 VICTORY Data Message	20059–20140630, Pro	
Automotive Combination Vehicle Weight VICTORY Data Message	20060–20140630, Pro	
Automotive Electronic Engine Controller 1 VICTORY Data Message	20061–20140630, Pro	
Automotive Electronic Engine Controller 2 VICTORY Data Message	20062–20140630, Pro	
Automotive Fan Drive 1 VICTORY Data Message	20063–20140630, Pro	
Automotive Tire Condition VICTORY Data Message	20064–20210409, Exp	
Automotive Torque/Speed Control 1 VICTORY Data Message	20065–20140630, Pro	

(+) Experimental specifications are not included in compliance tests.

(1) One or more of 20040, 20041, 20042, 20043, 20044, 20045, 20046, 20047, 20048, 20049, 20050, 20051, 20052, 20053, 20054, 20055, 20056, 20057, 20058, 20059, 20060, 20061, 20062, 20063, 20064, 20065 is Required

9.2 Power Distribution System Types

9.2.1 Power Distribution System <VT70100-V1.9>

The table below identifies the interfaces for the power distribution system component type.

Table 141. Power Distribution System <VT70100-V1.9>

Spec Title	Spec Tag	Applicability
Power Distribution System Data Interface	VT70101-V1.9	Required
Power Distribution System Management Interface	VT70102-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

9.2.1.1 Power Distribution System Data Interface <VT70101-V1.9>

The table below identifies the specifications for the power distribution system data interface.

Table 142. Power Distribution System Data Interface <VT70101-V1.9>

Spec Title	Spec Tag	Applicability
Common Data Interface Specifications	VT59901-V1.6	Required
Power Distribution System VICTORY Data Messages	VT70103-V1.9	Required

9.2.1.2 Power Distribution System Management Interface <VT70102-V1.9>

The table below identifies the specifications for the power distribution system management interface.

Table 143. Power Distribution System Management Interface <VT70102-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Power Distribution System Data Management Parameters	16004–20210409, Pro	Required
Power Distribution System Device Management Parameters	16006–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Power Distribution System Configuration Description	30020–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

9.2.1.3 Power Distribution System VICTORY Data Messages <VT70103-V1.9>

The table below provides the list of power distribution system VICTORY data messages.

Table 144. Power Distribution System VICTORY Data Messages <VT70103-V1.9>

Spec Title	Spec Tag	Applicability
Power System Device VICTORY Data Message	20069–20210409, Pro	(1) Required
Power System Device Threshold VICTORY Data Message	20070–20210409, Pro	
Platform Battery VICTORY Data Message	20071–20210409, Pro	
Platform Power Generation VICTORY Data Message	20072–20210409, Pro	
Platform Power VICTORY Data Message	20073–20131031, Pro	

(1) One or more of 20069, 20070, 20071, 20072, 20073 is Required

9.3 Lethality System Types

9.3.1 Remote Weapon Station <VT70200-V1.9>

The table below identifies the interfaces for the remote weapon station (RWS) component type.

Table 145. Remote Weapon Station <VT70200-V1.9>

Spec Title	Spec Tag	Applicability
Remote Weapon Station Data Interface	VT70201-V1.9	Required
Remote Weapon Station Management Interface	VT70202-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

9.3.1.1 Remote Weapon Station Data Interface <VT70201-V1.9>

The table below identifies the specifications for the remote weapon station data interface.

Table 146. Remote Weapon Station Data Interface <VT70201-V1.9>

Spec Title	Spec Tag	Applicability
Common Data Interface Specifications	VT59901-V1.6	Required
RWS Data Specification	20019–20210409, Pro	Required

9.3.1.2 Remote Weapon Station Management Interface <VT70202-V1.9>

The table below identifies the specifications for the remote weapon station management interface.

Table 147. Remote Weapon Station Management Interface <VT70202-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
VICTORY Data Management Parameters	10008–20210409, Pro	Required
Device Information Management Parameters	10009–20210409, Pro	Required
RWS Device Management Parameters	16002–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Remote Weapon Station Configuration Description	30021–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

9.4 Logistics System Types

9.4.1 Vehicle Configuration Service <VT70800-V1.9>

The table below identifies the interfaces for the vehicle configuration service component type.

Table 148. Vehicle Configuration Service <VT70800-V1.9>

Spec Title	Spec Tag	Applicability
Vehicle Configuration Service Management Interface	VT70801-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

9.4.1.1 Vehicle Configuration Service Management Interface <VT70801-V1.9>

The table below identifies the specifications for the vehicle configuration service management interface.

Table 149. Vehicle Configuration Service Management Interface <VT70801-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Vehicle Configuration Service Management Parameters	15023–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Vehicle Configuration Service Configuration Description	30050–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

9.5 Protection System Types

9.5.1 Threat Detection and Reporting Service <VT60600-V1.9>

The table below identifies the interfaces for the threat detection and reporting service component type.

Table 150. Threat Detection and Reporting Service <VT60600-V1.9>

Spec Title	Spec Tag	Applicability
Threat Detection and Reporting Data Interface	VT60601-V1.9	Required
Threat Detection and Reporting Management Interface	VT60602-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

9.5.1.1 Threat Detection and Reporting Data Interface <VT60601-V1.9>

The table below identifies the specifications for the threat detection and reporting data interface.

Table 151. Threat Detection and Reporting Data Interface <VT60601-V1.9>

Spec Title	Spec Tag	Applicability
Common Data Interface Specifications	VT59901-V1.6	Required
Threat Detection and Reporting Data Specification	20011–20210409, Pro	Required

9.5.1.2 Threat Detection and Reporting Management Interface <VT60602-V1.9>

The table below identifies the specifications for the threat detection and reporting management interface.

Table 152. Threat Detection and Reporting Management Interface <VT60602-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Threat Detection and Reporting Management Parameters	15004–20210409, Pro	Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Threat Detection and Reporting Service Configuration Description	30015–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

9.5.2 Active Protection System <VT70900-V1.9>

The table below identifies the interfaces for the active protection system (APS) component type.

Table 153. Active Protection System <VT70900-V1.9>

Spec Title	Spec Tag	Applicability
Active Protection System Data Interface	VT70901-V1.9	Required
Active Protection System Management Interface	VT70902-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

9.5.2.1 Active Protection System Data Interface <VT70901-V1.9>

The table below identifies the specifications for the active protection system data interface.

Table 154. Active Protection System Data Interface <VT70901-V1.9>

Spec Title	Spec Tag	Applicability
Common Data Interface Specifications	VT59901-V1.6	Required
Active Protection System VICTORY Data Messages	VT70903-V1.9	Required

9.5.2.2 Active Protection System VICTORY Data Messages <VT70903-V1.9>

The table below provides the list of active protection system VICTORY data messages.

Table 155. Active Protection System VICTORY Data Messages <VT70903-V1.9>

Spec Title	Spec Tag	Applicability
Active Protection System Engagement Report VICTORY Data Message	20080–20210409, Exp (1)	(+) Required
Active Protection System Status VICTORY Data Message	20083–20210409, Exp (1)	(+) Required

(+) Experimental specifications are not included in compliance tests.

(1) This VICTORY specification contains technical data whose export is restricted by the Arms Export Control Act (Title 22, U.S.C., Sec 2751 et seq.) or the Export Administration Act of 1979, as amended, Title 50, U.S.C., 2401, et seq. As a result, this specification is provided in a separate document.

9.5.2.3 Active Protection System Management Interface <VT70902-V1.9>

The table below identifies the specifications for the active protection system management interface.

Table 156. Active Protection System Management Interface <VT70902-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Active Protection System Data Management Parameters	16007–20210409, Exp (1)	(+) Required
Active Protection System Management Parameters	16008–20210409, Exp (1)	(+) Required
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended

Spec Title	Spec Tag	Applicability
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Active Protection System Configuration Description	30051–20210409, Exp (1)	(+) Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

(+) Experimental specifications are not included in compliance tests.

(1) This VICTORY specification contains technical data whose export is restricted by the Arms Export Control Act (Title 22, U.S.C., Sec 2751 et seq.) or the Export Administration Act of 1979, as amended, Title 50, U.S.C., 2401, et seq. As a result, this specification is provided in a separate document.

9.6 Training System Types

9.6.1 Live Training Transformation Adapter <VT71000-V1.9>

The table below identifies the interfaces for the live training transformation (LT2) adapter component type.

Table 157. Live Training Transformation Adapter <VT71000-V1.9>

Spec Title	Spec Tag	Applicability
Live Training Transformation Adapter Data Interface	VT71001-V1.7	Required
Live Training Transformation Adapter Management Interface	VT71002-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Auto-Discovery Interface	VT59930-V1.6	Optional

9.6.1.1 Live Training Transformation Adapter Data Interface <VT71001-V1.7>

The table below identifies the specifications for the live training transformation adapter data interface.

Table 158. Live Training Transformation Adapter Data Interface <VT71001-V1.7>

Spec Title	Spec Tag	Applicability
Common Data Interface Specifications	VT59901-V1.6	Required
Live Training Transformation Adapter Data Specification	20084–20180215, Pro	Required

9.6.1.2 Live Training Transformation Adapter Management Interface <VT71002-V1.9>

The table below identifies the specifications for the live training transformation adapter management interface.

Table 159. Live Training Transformation Adapter Management Interface <VT71002-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
Live Training Transformation Adapter Management Parameters	16009–20210409, Pro	Required

Spec Title	Spec Tag	Applicability
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Recommended
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Configuration Management Parameters	10028–20210409, Pro	Required
Live Training Transformation Adapter Configuration Description	30052–20210409, Pro	Required
Authentication and Authorization Security Specifications	VT50400-V1.9	Recommended

9.7 MORA Component Type Specifications

9.7.1 Low Latency Switch <VT91010-V1.9>

NOTE: *The Low Latency Switch <VT91010-V1.9> is equivalent to <MT91010-V2.4>, which is defined in the separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."*

The Low Latency Switch interfaces implementation shall be in accordance with Table 160 below.

Table 160. Low Latency Switch <VT91010-V1.9>

Spec Title	Spec Tag	Applicability
Low Latency Switch Data Interface	VT91011-V1.9	Required
Switch Management Interface	VT50902-V1.7	Required
SNMP-Based Health Publishing Interface	VT59903-V1.6	Required
MORA Low Latency Bus Performance Specification	48201–20160131, Exp (1)	(+) Required

(+) Experimental specifications are not included in compliance tests.

(1) This specification is located in a separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."

9.7.1.1 Low Latency Switch Data Interface <VT91011-V1.9>

NOTE: *The Low Latency Switch Data Interface <VT91011-V1.9> is equivalent to <MT91011-V2.4>, which is defined in the separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."*

The Low Latency Switch Data Interface implementation shall be in accordance with Table 161 below.

Table 161. Low Latency Switch Data Interface <VT91011-V1.9>

Spec Title	Spec Tag	Applicability
Low Latency Switch Data Interface Physical and Data-Link Layer Specifications	VT91012-V1.9	Required
Switch Data Interface Network Layer and Higher Specifications	VT50901.T4-V1.7	Required

9.7.1.2 Low Latency Switch Data Interface Physical and Data-Link Layer Specifications <VT91012-V1.9>

NOTE: *The Low Latency Switch Data Interface Physical and Data-Link Layer Specifications <VT91012-V1.9> is equivalent to <MT91012-V2.4>, which is defined in the separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."*

The Low Latency Switch Data Interface Physical and Data-Link Layer Specifications implementation shall be in accordance with Table 162 below.

Table 162. Low Latency Switch Data Interface Physical and Data-Link Layer Specifications <VT91012-V1.9>

Spec Title	Spec Tag	Applicability
Low Latency Copper Data Physical and Data-Link Layer Specifications	VT91013-V1.9	(1) Required
Low Latency Fiber Data Physical and Data-Link Layer Specifications	VT91014-V1.9	
Frame Structure	01013–20121113, Pro	Required
Media Access Control	01014–20121113, Pro	Required
Logical Link Control	01015–20120725, Pro	Required
Link Layer Switching	01016–20120725, Pro	Required
Link Layer Flow Control	01017–20121113, Pro	Required
Virtual Local Area Network (VLAN)	01077–20120131, Exp	(+) Required
Multicast Dissemination Over the Network Specification	48006–20160131, Exp (2)	(+) Required
Jumbo Frame Packet Support Specification	48007–20200707, Exp (2)	(+) Required
MDM Traffic Marking Specification	48008–20200707, Exp (2)	(+) Required

(+) Experimental specifications are not included in compliance tests.

(1) One or more of VT91013, VT91014 is Required

(2) This specification is located in a separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."

9.7.1.3 Low Latency Copper Data Physical and Data-Link Layer Specifications <VT91013-V1.9>

NOTE: *The Low Latency Copper Data Physical and Data-Link Layer Specifications <VT91013-V1.9> is equivalent to <MT91013-V2.4>, which is defined in the separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."*

The Low Latency Copper Data Physical and Data-Link Layer Specifications implementation shall be in accordance with Table 163 below.

Table 163. Low Latency Copper Data Physical and Data-Link Layer Specifications <VT91013-V1.9>

Spec Title	Spec Tag	Applicability
1000BASE-T	01006–20121113, Pro	(+) (1) Required
1000BASE-KX	01095–20210409, Exp	

Spec Title	Spec Tag	Applicability
10GBASE-X/10GBASE-T	01009–20121113, Pro	
10GBASE-KX4	48103–20140530, Exp (2)	
10GBASE-KR	01096–20210409, Exp	
40GBASE-KR4	48112–20170531, Exp (2)	
100GBASE-KR4	48114–20200707, Exp (2)	
Copper Auto-Negotiation	01011–20121113, Pro	Optional

(+) Experimental specifications are not included in compliance tests.

(1) One or more of 01006, 01095, 01009, 48103, 01096, 48112, 48114 is Required

(2) This specification is located in a separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”

9.7.1.4 Low Latency Fiber Data Physical and Data-Link Layer Specifications <VT91014-V1.9>

NOTE: *The Low Latency Fiber Data Physical and Data-Link Layer Specifications <VT91014-V1.9> is equivalent to <MT91014-V2.1>, which is defined in the separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”*

The Low Latency Fiber Data Physical and Data-Link Layer Specifications implementation shall be in accordance with Table 164 below.

Table 164. Low Latency Fiber Data Physical and Data-Link Layer Specifications <VT91014-V1.9>

Spec Title	Spec Tag	Applicability
10GBASE-SR	48102–20140530, Exp (2)	(+) (1) Required
40GBASE-SR4	48113–20140530, Exp (2)	

(+) Experimental specifications are not included in compliance tests.

(1) One or more of 48102, 48113 is Required

(2) This specification is located in a separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”

9.7.2 MORA Device <VT92010-V1.9>

NOTE: *The MORA Device <VT92010-V1.9> is equivalent to <MT92010-V2.4>, which is defined in the separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”*

The MORA Device component type provides device-level interfaces for a MORA Device. These include discovery and capability announcement, basic file input and output, configuration management, waveform management, status, and health. This component type also contains an operations interface for interacting with the MORA Signal Port Manager component type.

Any MORA Device shall implement the MORA Device component type in accordance with Table 165 below.

Table 165. MORA Device <VT92010-V1.9>

Spec Title	Spec Tag	Applicability
MORA Device Management Interface	VT92011-V1.9	Required
MORA Device Operations Interface	VT90016-V1.9	Required
Generic End Node Data Interface	VT52101-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Syslog Protocol Data Parameters for MORA	49101–20200707, Pro (1)	Required
Auto-Discovery Interface	VT59930-V1.6	Recommended

(1) This specification is located in a separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."

9.7.2.1 MORA Device Management Interface <VT92011-V1.9>

NOTE: *The MORA Device Management Interface <VT92011-V1.9> is equivalent to <MT92011-V2.4>, which is defined in the separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."*

The MORA Device Management Interface provides methods to manage the MORA Device. These methods include retrieving device description, device status, device configuration, built-in test results, and device files. The interface also provides the methods for setting device configuration and execution mode as well as sending device files.

The MORA Device Management Interface implementation shall be in accordance with Table 166 below.

Table 166. MORA Device Management Interface <VT92011-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
MORA Schemas and WSDLs	49103–20201002, Pro (1)	Required
Device Information Management Parameters	10009–20210409, Pro	Required
Generic End Node Management Parameters	13003–20210409, Pro	Required
MORA Device Management Parameters	49011–20210301, Pro (1)	Required
Waveform Management Parameters	49012–20201002, Pro (1)	Optional
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Optional
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Authentication and Authorization Security Specifications	VT50400-V1.9	Optional

(1) This specification is located in a separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."

9.7.2.2 MORA Device Operations Interface <VT90016-V1.9>

NOTE: *The MORA Device Operations Interface <VT90016-V1.9> is equivalent to <MT90016-V2.4>, which is defined in the separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."*

The MORA Device Operations Interface provides methods to interact with the MORA Signal Port Manager component type. These methods include signal resource execution mode, time of day announcement, and health updates. The interface also supports acknowledgement messaging.

The MORA Device Operations Interface implementation shall be in accordance with Table 167 below.

Table 167. MORA Device Operations Interface <VT90016-V1.9>

Spec Title	Spec Tag	Applicability
MDM Protocol Specifications	VT96001-V1.9	Required
MDM Command Message (Type 6) Specification	48615–20200707, Exp (1)	(+) Required
MDM Acknowledgement Message (Type 2) Specification	48611–20180928, Exp (1)	(+) Required
MDM Networking Specifications	VT96002-V1.9	Optional*
MDM Time of Day Message (Type 3) Specification	48612–20170531, Exp (1)	(+) Optional
MDM Health Message (Type 5) Specification	48614–20200707, Exp (1)	(+) Optional

(+) Experimental specifications are not included in compliance tests.

(1) This specification is located in a separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”

* Required for MDMs transiting a network

9.7.3 MORA Signal Resource Manager <VT92013-V1.9>

NOTE: *The MORA Signal Resource Manager <VT92013-V1.9> is equivalent to <MT92013-V2.4>, which is defined in the separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”*

The MORA Signal Resource Manager component type provides interfaces to retrieve and reserve RF capabilities. The component type provides interfaces for describing RF functions and capabilities of signal resources including antennas, RF conditioning, RF distribution, RF frequency translation, and signal domain conversion types. It also provides an interface to reserve signal ports and discover the endpoints for real-time control, context, and data provided over the ML2B. The MORA Signal Resource Manager component type also enables information exchange on antenna arrays, connections, manifolds, and RF performance. This component type also contains an operations interface for interacting with the MORA Signal Port Manager component type.

Any MORA Device that contains signal resources shall implement the MORA Signal Resource Manager component type in accordance with Table 168 below.

Table 168. MORA Signal Resource Manager <VT92013-V1.9>

Spec Title	Spec Tag	Applicability
MORA Signal Resource Management Interface	VT92014-V1.9	Required
MORA Signal Resource Manager Operations Interface	VT90017-V1.9	Required
Syslog-Based Health Publishing Interface	VT50103-V1.9	Required
Syslog Protocol Data Parameters for MORA	49101–20200707, Pro (1)	Required
Auto-Discovery Interface	VT59930-V1.6	Recommended

(1) This specification is located in a separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”

9.7.3.1 MORA Signal Resource Management Interface <VT92014-V1.9>

NOTE: *The MORA Signal Resource Management Interface <VT92014-V1.9> is equivalent to <MT92014-V2.4>, which is defined in the separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."*

The MORA Signal Resource Management Interface provides methods to understand and reserve partitioned signal resource capabilities formally defined as Signal Ports. This interface includes methods to discover signal port capabilities for signal resources including antennas, RF conditioning, RF frequency translation, and signal domain conversion types. The interface also provides for information exchange on antenna arrays, connections, manifolds, and RF performance.

The MORA Signal Resource Management Interface implementation shall be in accordance with Table 169 below.

Table 169. MORA Signal Resource Management Interface <VT92014-V1.9>

Spec Title	Spec Tag	Applicability
Common Management Interface Specifications	VT59902-V1.9	Required
MORA Schemas and WSDLs	49103–20201002, Pro (1)	Required
MORA Signal Resource Management Parameters	49014–20201002, Pro (1)	Required
Radio Frequency Distribution Management Parameters	49005–20201002, Pro (1)	Optional
Data Confidentiality & Integrity for Management Interfaces	10018–20180215, Pro	Optional
Data Authentication for Management Interfaces	10019–20150331, Pro	Optional
Authentication and Authorization Security Specifications	VT50400-V1.9	Optional

(1) This specification is located in a separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."

9.7.3.2 MORA Signal Resource Manager Operations Interface <VT90017-V1.9>

NOTE: *The MORA Signal Resource Manager Operations Interface <VT90017-V1.9> is equivalent to <MT90017-V2.4>, which is defined in the separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."*

The MORA Signal Resource Manager Operations Interface provides methods to interact with the MORA Signal Port Manager component type. These methods include the ability to exchange the identity of the user(s) who have reserved any signal port or switch group. These methods also include time of day announcement and health updates and supports acknowledgement messaging.

The MORA Signal Resource Manager Interface implementation shall be in accordance with Table 170 below.

Table 170. MORA Signal Resource Manager Operations Interface <VT90017-V1.9>

Spec Title	Spec Tag	Applicability
MDM Protocol Specifications	VT96001-V1.9	Required
MDM Acknowledgement Message (Type 2) Specification	48611–20180928, Exp (1)	(+) Required
MDM Time of Day Message (Type 3) Specification	48612–20170531, Exp (1)	(+) Required
MDM Signal Port User ID Message (Type 4) Specification	48613–20201002, Exp (1)	(+) Required
MDM Health Message (Type 5) Specification	48614–20200707, Exp (1)	(+) Required
MDM Networking Specifications	VT96002-V1.9	Optional*
MDM Switch Group User ID Message (Type 7) Specification	48617–20200707, Exp (1)	(+) Optional

(+) Experimental specifications are not included in compliance tests.

(1) This specification is located in a separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”

* Required for MDMs transiting a network

9.7.4 MORA Signal Port Manager <VT92016-V1.9>

NOTE: *The MORA Signal Port Manager <VT92016-V1.9> is equivalent to <MT92016-V2.4>, which is defined in the separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”*

The MORA Signal Port Manager is a component type that defines interfaces for high-speed and low latency control of signal ports.

Any MORA Device that contains signal resources shall implement the MORA Signal Port Manager component type in accordance with Table 171 below.

Table 171. MORA Signal Port Manager <VT92016-V1.9>

Spec Title	Spec Tag	Applicability
RF Layer Analog Data Interface	VT90011-V1.9	(1) Required
RF Layer Digital Data Interface	VT90014-V1.9	
RF Layer Command Interface	VT90012-V1.9	Required
RF Layer Context Interface	VT90013-V1.9	Required
RF Layer Operations Interface	VT90015-V1.9	Required

(1) One or more of VT90011, VT90014 is Required

9.7.4.1 RF Layer Analog Data Interface <VT90011-V1.9>

NOTE: *The RF Layer Analog Data Interface <VT90011-V1.9> is equivalent to <MT90011-V2.1>, which is defined in the separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”*

The RF Layer Analog Data Interface implementation shall be in accordance with Table 172 below.

Table 172. RF Layer Analog Data Interface <VT90011-V1.9>

Spec Title	Spec Tag	Applicability
Coaxial Cable Specification	47001–20150918, Pro (1)	Required
Optical Fiber Specification	47002–20140530, Exp (1)	(+) Optional

(+) Experimental specifications are not included in compliance tests.

(1) This specification is located in a separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”

9.7.4.2 RF Layer Command Interface <VT90012-V1.9>

NOTE: *The RF Layer Command Interface <VT90012-V1.9> is equivalent to <MT90012-V2.4>, which is defined in the separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”*

The RF Layer Command Interface for the transmission of packetized command data over the ML2B network shall be in accordance with Table 173 below.

Table 173. RF Layer Command Interface <VT90012-V1.9>

Spec Title	Spec Tag	Applicability
MDM Protocol Specifications	VT96001-V1.9	Required
MDM Networking Specifications	VT96002-V1.9	Required
MDM VRT Message Structure Specification	48607–20200707, Exp (1)	(+) Required
MDM VRT Context and Command Fields Specification	48616–20210301, Exp (1)	(+) Required
MDM VRT Command Packet Type Specification	48610–20200903, Exp (1)	(+) Required
MDM Acknowledgement Message (Type 2) Specification	48611–20180928, Exp (1)	(+) Required
MORA Low Latency Bus Performance Specification	48201–20160131, Exp (1)	(+) Recommended

(+) Experimental specifications are not included in compliance tests.

(1) This specification is located in a separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”

9.7.4.3 RF Layer Context Interface <VT90013-V1.9>

NOTE: *The RF Layer Context Interface <VT90013-V1.9> is equivalent to <MT90013-V2.4>, which is defined in the separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”*

The RF Layer Context Interface for the transmission of packetized RF context data over the ML2B network shall be in accordance with Table 174 below.

Table 174. RF Layer Context Interface <VT90013-V1.9>

Spec Title	Spec Tag	Applicability
MDM Protocol Specifications	VT96001-V1.9	Required
MDM Networking Specifications	VT96002-V1.9	Required
MDM VRT Message Structure Specification	48607–20200707, Exp (1)	(+) Required
MDM VRT Context and Command Fields Specification	48616–20210301, Exp (1)	(+) Required
MDM VRT Context Packet Type Specification	48609–20200903, Exp (1)	(+) Required
MDM Multicast Dissemination Specification	48801–20170531, Exp (1)	(+) Required
MORA Low Latency Bus Performance Specification	48201–20160131, Exp (1)	(+) Recommended

(+) Experimental specifications are not included in compliance tests.

(1) This specification is located in a separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”

9.7.4.4 RF Layer Digital Data Interface <VT90014-V1.9>

NOTE: *The RF Layer Digital Data Interface <VT90014-V1.9> is equivalent to <MT90014-V2.4>, which is defined in the separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”*

The RF Layer Digital Data Interface for the transmission of packetized RF data over the ML2B network shall be in accordance with Table 175 below.

Table 175. RF Layer Digital Data Interface <VT90014-V1.9>

Spec Title	Spec Tag	Applicability
MDM Protocol Specifications	VT96001-V1.9	Required
MDM Networking Specifications	VT96002-V1.9	Required
MDM VRT Message Structure Specification	48607–20200707, Exp (1)	(+) Required
MDM VRT Signal Data Packet Type Specification	48608–20200707, Exp (1)	(+) Required
MDM Multicast Dissemination Specification	48801–20170531, Exp (1)	(+) Required
MORA Low Latency Bus Performance Specification	48201–20160131, Exp (1)	(+) Recommended

(+) Experimental specifications are not included in compliance tests.

(1) This specification is located in a separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”

9.7.4.5 RF Layer Operations Interface <VT90015-V1.9>

NOTE: *The RF Layer Operations Interface <VT90015-V1.9> is equivalent to <MT90015-V2.4>, which is defined in the separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”*

The RF Layer Operations Interface for the transmission of packetized operations data over the ML2B network shall be in accordance with Table 176 below.

Table 176. RF Layer Operations Interface <VT90015-V1.9>

Spec Title	Spec Tag	Applicability
MDM Protocol Specifications	VT96001-V1.9	Required
MDM Acknowledgement Message (Type 2) Specification	48611–20180928, Exp (1)	(+) Required
MDM Time of Day Message (Type 3) Specification	48612–20170531, Exp (1)	(+) Required
MDM Signal Port User ID Message (Type 4) Specification	48613–20201002, Exp (1)	(+) Required
MDM Health Message (Type 5) Specification	48614–20200707, Exp (1)	(+) Required
MDM Command Message (Type 6) Specification	48615–20200707, Exp (1)	(+) Required
MDM Switch Group User ID Message (Type 7) Specification	48617–20200707, Exp (1)	(+) Required
MDM Networking Specifications	VT96002-V1.9	Optional*

(+) Experimental specifications are not included in compliance tests.

(1) This specification is located in a separate appendix titled “Modular Open RF Architecture (MORA) Specification, Version 2.4.”

* Required for MDMs transiting a network

9.8 MORA Data Messaging Specifications

9.8.1 MDM Protocol Specifications <VT96001-V1.9>

NOTE: *The MDM Protocol Specifications <VT96001-V1.9> is equivalent to <MT96001-V2.4>, which is defined in the separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."*

The MDM protocol specifications shall be in accordance with Table 177 below.

Table 177. MDM Protocol Specifications <VT96001-V1.9>

Spec Title	Spec Tag	Applicability
MDM Data Encoding Specification	48605–20160131, Exp (1)	(+) Required
MDM Encapsulation Specification	48606–20201002, Exp (1)	(+) Required

(+) Experimental specifications are not included in compliance tests.

(1) This specification is located in a separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."

9.8.2 MDM Networking Specifications <VT96002-V1.9>

NOTE: *The MDM Networking Specifications <VT96002-V1.9> is equivalent to <MT96002-V2.4>, which is defined in the separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."*

The MDM message structure is used in six distinct interfaces across three different MORA component types. MDM message interactions may occur within a single MORA Device (intra-device) or occur between multiple MORA Devices. MDM message interactions between devices require the message to transit a network and therefore shall be in accordance with the following networking specifications in Table 178 below.

Table 178. MDM Networking Specifications <VT96002-V1.9>

Spec Title	Spec Tag	Applicability
Low Latency Copper Data Physical and Data-Link Layer Specifications	VT91013-V1.9	(1) Required
Low Latency Fiber Data Physical and Data-Link Layer Specifications	VT91014-V1.9	
Standard for Ethernet Specification	48001–20160131, Exp (2)	(+) Required
Virtual LANs Specification	48002–20160131, Exp (2)	(+) Required
Standard for Internet Protocol and User Datagram Protocol Specification	48005–20160131, Exp (2)	(+) Required
Jumbo Frame Packet Support Specification	48007–20200707, Exp (2)	(+) Required
MDM Traffic Marking Specification	48008–20200707, Exp (2)	(+) Required

(+) Experimental specifications are not included in compliance tests.

(1) One or more of VT91013, VT91014 is Required

(2) This specification is located in a separate appendix titled "Modular Open RF Architecture (MORA) Specification, Version 2.4."

10. VDB DATA INTERFACE SPECIFICATIONS

The VDB supports the transport of data between many different systems and components, and, as such, must be prepared to implement a variety of physical to transport layer protocols that are utilized by these systems. For example, the VDB will be responsible for implementing unicast, multicast, and broadcast delivery mechanisms, as well as reliable and unreliable transports. The VDB Data Interface Specifications document the set of protocols that are supported by the VDB at each level up to the transport layer. They define the set of physical to transport layer protocols that must be supported by each of the component types that make up the VDB.

The specifications for the components and interfaces that support the system integrated by and interfaced by the VDB are called out in separate sections. Those specifications detail the application layer data formats and protocols, as well as the specific choices of transport layer protocols that have been selected for that application from the choices supported by the VDB data transport layer interfaces.

The following sections describe the set of physical to transport layer protocols that are to be supported by the VDB and its components. This will include the protocols that support message formation, addressing, delivery, routing, and QoS.

Later sections will provide the mapping of particular components that make up the VDB to these individual specifications.

10.1 VDB Physical Layer Protocols

The VDB data links shall leverage Ethernet for the core technology at the physical (PHY) layer. The specific variants and options of the Ethernet standard are called out in this section.

The VDB data interface shall support the bit rate and physical protocol standards as specified in the sections titled "10 Mbps Ethernet," "100 Mbps Ethernet," and "Gigabit Ethernet." Additionally, the VDB data interface may optionally support the bit rate and physical protocol standards as specified in the section titled "10 Gigabit Ethernet."

10.1.1 10 Mbps Ethernet

10.1.1.1 10BASE-T <01001–20121113, Pro>

Copper media connections using 10BASE-T Ethernet shall comply with IEEE 802.3–2008, Section 1, Clause 14.

10.1.1.2 10BASE-FL <01002–20121113, Pro>

Multi-mode or single-mode fiber media connections using 10BASE-FL Ethernet shall comply with IEEE 802.3–2008, Section 1, Clause 18.

10.1.2 100 Mbps Ethernet

10.1.2.1 100BASE-TX <01003–20121113, Pro>

Copper media connections using 100BASE-TX Ethernet shall comply with IEEE 802.3–2008, Section 2, Clause 25.

10.1.2.2 100BASE-FX <01004–20121113, Pro>

Multi-mode fiber media connections using 100BASE-FX Ethernet shall comply with IEEE 802.3–2008, Section 2, Clause 26.

10.1.2.3 100BASE-LX10 <01005–20121113, Pro>

Single-mode fiber media connections using 100BASE-LX10 Ethernet shall comply with IEEE 802.3–2008, Section 5, Clause 58.

10.1.3 Gigabit Ethernet

10.1.3.1 1000BASE-T <01006–20121113, Pro>

Copper media connections using 1000BASE-T Ethernet shall comply with IEEE 802.3–2008, Section 3, Clause 40.

10.1.3.2 1000BASE-SX <01007–20150331, Pro>

Multi-mode fiber media connections using 1000BASE-SX Ethernet shall comply with IEEE 802.3–2008, Section 3, Clause 38, and shall support auto-negotiation of speed, duplex, and flow control in the manner specified in IEEE 802.3–2008, Section 3, Clause 37.

10.1.3.3 1000BASE-LX <01008–20150331, Pro>

Multi-mode or single-mode fiber media connections using 1000BASE-LX Ethernet shall comply with IEEE 802.3–2008, Section 3, Clause 38, and shall support auto-negotiation of speed, duplex, and flow control in the manner specified in IEEE 802.3–2008, Section 3, Clause 37.

10.1.3.4 1000BASE-KX <01095–20210409, Exp>

Backplane connections using 1000BASE-KX shall comply with IEEE 802.3 clause 70.

10.1.4 10 Gigabit Ethernet

10.1.4.1 10GBASE-X/10GBASE-T <01009–20121113, Pro>

Copper media connections using 10 Gigabit Ethernet shall comply with one of the 10GBASE-X or 10GBASE-T physical standards as defined in IEEE 802.3–2008.

10.1.4.2 10GBASE-R <01010–20121113, Pro>

Fiber media connections using 10 Gigabit Ethernet shall comply with one of the 10GBASE-R physical standards as defined in IEEE 802.3–2008.

10.1.4.3 10GBASE-KR <01096–20210409, Exp>

Backplane connections using 10GBASE-KR shall comply with IEEE802.3 clause 72.

10.1.5 Auto-Negotiation

The VDB data interface shall support an auto-negotiation mechanism that is consistent with its particular physical channel medium (copper or fiber). These auto-negotiation mechanisms are described in the following sections.

10.1.5.1 Copper Auto-Negotiation <01011–20121113, Pro>

Copper media connections, as described in the preceding sections, shall support auto-negotiation of speed, duplex, and flow control in the manner specified in IEEE 802.3–2008, Section 2, Clause 28.

10.2 Data Link Layer Protocols

To support the communication of data across the network, the VDB shall implement each of the following Ethernet data link (DL) layer protocols.

10.2.1 Frame Structure <01013–20121113, Pro>

The VDB shall support the frame structure, field definitions, and Media Access Control (MAC) conventions as specified in IEEE 802.3–2008.

- Data link frames shall support 48-bit locally and universally administered addresses in a manner consistent with IEEE 802.3–2008.
- Data link frame structures shall support type-encapsulated and length-encapsulated frames as specified in IEEE 802.3–2008.

10.2.2 Media Access Control <01014–20121113, Pro>

The VDB shall support the Media Access Control (MAC) protocols as specified in IEEE 802.3–2008, Clauses 2, 3, and 4.

- The MAC protocols shall convey type- and length-encapsulated frames to support IP network layer protocols.

10.2.3 Logical Link Control <01015–20120725, Pro>

The VDB shall support the Logical Link Control (LLC) protocols as specified in IEEE 802.2–1998 to the extent necessary to support IP network layer protocols.

10.2.4 Link Layer Switching <01016–20120725, Pro>

The VDB shall conform to the requirements set forth in IEEE 802.1D–2004 for transparent bridging and Rapid Spanning Tree Protocol (RSTP) functionality.

10.2.5 Link Layer Flow Control <01017–20121113, Pro>

Those VDB implementations that support full-duplex Ethernet shall support flow control “PAUSE” frames as specified in IEEE 802.3–2008, Clause 31.

NOTE: *VDB Nodes with sufficiently large queuing buffers should be chosen to handle the “burstiness” of the traffic within the latency constraints of a particular implementation.*

10.3 Network Layer Protocols

To support the communication of data across the network, the VDB shall implement each of the following network layer protocols.

10.3.1 Internet Protocol Version 4 Specification <01018–20120725, Pro>

The VDB shall conform to the following Internet Protocol Version 4 (IPv4) core standards:

- IETF RFC 791: Internet Protocol

- IETF RFC 919: Broadcasting Internet Datagrams
- IETF RFC 922: Broadcasting Internet Datagrams in the Presence of Subnets

10.3.2 Internet Protocol Version 6 Specification <01019–20110729, Exp>

The VDB shall conform to the following Internet Protocol Version 6 (IPv6) core standards:

- IETF RFC 2460: Internet Protocol, Version 6 (IPv6) Specification
- IETF RFC 4193: Unique Local IPv6 Unicast Addresses
- IETF RFC 4291: IP Version 6 Addressing Architecture
- IETF RFC 4861: Neighbor Discovery for IP version 6 (IPv6)

In addition to the core RFCs above, and other IPv6 RFCs described in this document, the following RFCs are also considered necessary for comprehensive IPv6 support (note that some are only informational):

- IETF RFC 2375: IPv6 Multicast Address Assignments (Informational)
- IETF RFC 3587: IPv6 Global Unicast Address Format (Informational)
- IETF RFC 3849: IPv6 Address Prefix Reserved for Documentation
- IETF RFC 4302: IP Authentication Header
- IETF RFC 4620: IPv6 Node Information Queries (Experimental)
- IETF RFC 5095: Deprecation of Type 0 Routing Headers in IPv6
- IETF RFC 5722: Handling of Overlapping IPv6 Fragments
- IETF RFC 5871: IANA Allocation Guidelines for the IPv6 Routing Header
- IETF RFC 5952: A Recommendation for IPv6 Address Text Representation
- IETF RFC 6052: IPv6 Addressing of IPv4/IPv6 Translators
- IETF RFC 6085: Address Mapping of IPv6 Multicast Packets on Ethernet

10.3.3 IPv4 Datagram Transmission <01062–20120725, Pro>

The VDB shall conform to the following core standards for the transmission of IPv4 datagrams:

- IETF RFC 894: A Standard for the Transmission of IP Datagrams over Ethernet Networks
- IETF RFC 1042: A Standard for the Transmission of IP Datagrams over IEEE 802 Networks

10.3.4 IPv6 Datagram Transmission <01063–20120131, Exp>

The VDB shall conform to the following core standard for the transmission of IPv6 datagrams:

- IETF RFC 2464: Transmission of IPv6 Packets over Ethernet Networks

10.3.5 Internet Control Message Protocol for IPv4 <01021–20120725, Pro>

When implementing IPv4, the VDB shall conform to the following core Internet Control Message Protocol (ICMP) standards:

- IETF RFC 792: Internet Control Message Protocol
- IETF RFC 4884: Extended ICMP to Support Multi-Part Messages

10.3.6 Internet Control Message Protocol for IPv6 <01059–20120131, Exp>

When implementing IPv6, the VDB shall conform to the following core Internet Control Message Protocol (ICMP) standards, in addition to those for IPv4:

- IETF RFC 4443: Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification

10.3.7 Path MTU Discovery

Path Maximum Transmission Unit (MTU) Discovery (PMTUD) is used to avoid fragmentation of packets in IP-based networks. It provides a mechanism for the sender to discover the maximum size packet that can be sent along a path before it is fragmented.

10.3.7.1 Path MTU Discovery for IPv4 <01064–20180215, Pro>

For IPv4, PMTUD is provided by routers and is described by IETF RFC 1191: Path MTU Discovery. PMTUD support is required for VICTORY-compliant IPv4 routers.

10.3.7.2 Path MTU Discovery for IPv6 <01065–20120725, Pro>

For IPv6, PMTUD functionality is provided by end nodes, and is described by IETF RFC 1981: Path MTU Discovery for IP version 6. PMTUD may optionally be supported by VICTORY-compliant IPv6 end nodes.

10.3.8 IP Layer Security <01025–20180215, Pro>

A number of National Institute of Standards and Technology (NIST) Special Publications (SPs), Federal Information Processing Standards (FIPS) Publications, and Internet Engineering Task Force (IETF) Requests for Comments (RFCs) provided text for this specification. This text will be cited with a *Source:* line.

NOTE: *Some clarifications regarding the manner in which the Ethernet switch component is expected to support this specification can be found in the section titled "Applicability of Internet Protocol Specifications to the Switch Component."*

10.3.8.1 Cryptography Overview <Informational>

Cryptography is often used in an information technology security environment to protect data that is sensitive, has a high value, or is vulnerable to unauthorized disclosure or undetected modification during transmission or while in storage. Cryptography relies upon two basic components: an algorithm (or cryptographic methodology) and a cryptographic key. The algorithm is a mathematical function, and the key is a parameter used by that function.

Source: NIST SP 800–133: "Recommendation for Cryptographic Key Generation"

Cryptography is the science of mapping readable text, called plaintext, into an unreadable format, called ciphertext, and vice versa.

10.3.8.1.1 Security Services

Cryptography may be used to perform or support several basic security services: confidentiality, integrity authentication, source authentication, authorization, and non-repudiation. These services may also be required to protect cryptographic keying material. In addition, there are other cryptographic and non-cryptographic mechanisms that are used to support these security services. In general, a single cryptographic mechanism may provide more than one service (e.g., the use of digital signatures can provide integrity authentication and source authentication), but not all services.

In many applications, a combination of security services (confidentiality, integrity authentication, source authentication, and support for non-repudiation) is desired. Designers of secure systems often begin by considering which security services are needed to protect the information contained within and processed by the system. After these services have been determined, the designer then considers what mechanisms will best provide these services. Not all mechanisms are cryptographic in nature. However, cryptographic mechanisms consisting of algorithms, keys, and other keying material often provide the most cost-effective means of protecting the security of information. This is particularly true in applications where the information would otherwise be exposed to unauthorized entities. When properly implemented, some cryptographic algorithms provide multiple services. However, it is often the case that different algorithms need to be employed in order to provide all the desired services.

Source: NIST SP 800–57 Part 1: "Recommendation for Key Management, Part 1: General – Revision 4"

Ideally, security services would be provided using as few algorithms as possible. Security services are described in Table 179.

Table 179. Security Services

Service	Description
Confidentiality	Confidentiality is the property whereby information is not disclosed to unauthorized parties. Secrecy is a term that is often used synonymously with confidentiality. Confidentiality using cryptography is achieved using encryption to render the information unintelligible except by authorized entities. The information may become intelligible again by using decryption. In order for encryption to provide confidentiality, the cryptographic algorithm and mode of operation must be designed and implemented so that an unauthorized party cannot determine the secret or private keys associated with the encryption or be able to derive the plaintext directly without using the correct keys. *Source: NIST SP 800–57 Part 1 Revision 4*
Integrity / Data Integrity	Data integrity is a property whereby data has not been modified in an unauthorized manner since it was created, transmitted, or stored. Modification includes the insertion, deletion, and substitution of data. Cryptographic mechanisms, such as message authentication codes or digital signatures, can be used to detect (with a high probability) both accidental modifications (e.g., modifications that sometimes occur during noisy transmissions or by hardware memory failures) and deliberate modifications by an adversary. Non-cryptographic mechanisms are also often used to detect accidental modifications, but cannot be relied upon to detect deliberate modifications. *Source: NIST SP 800–57 Part 1 Revision 4*
Authentication	Two types of authentication services can be provided using cryptography: integrity authentication and source authentication. • An integrity authentication service is used to verify that data has not been modified. • A source authentication service is used to verify the identity of the user or system that created information. *Source: NIST SP 800–57 Part 1 Revision 4* ***NOTE:** Source Authentication is sometimes referred to as Peer Authentication.*
Authorization	Authorization is concerned with providing an official sanction or permission to perform a security function or activity (e.g., accessing a room). Authorization is considered a security service that is often supported by a cryptographic service. Normally, authorization is granted only after the execution of a successful source authentication service. *Source: NIST SP 800–57 Part 1 Revision 4*
Non-repudiation	Non-repudiation is a service that is used to provide assurance of the integrity and origin of data in such a way that the integrity and origin can be verified and validated by a third party as having originated from a specific entity in possession of the private key (i.e., the signatory). *Source: FIPS 186–4: "Digital Signature Standard (DSS)"*

10.3.8.2 Internet Protocol Security

Internet Protocol Security (IPsec) has emerged as the most commonly used network layer security protocol suite for protecting IP communications. IPsec is a framework of open standards for ensuring private communications over IP networks through the use of encryption and authentication. IPsec is not just a single protocol, but a combination of many protocols that work together to provide interoperable, high-quality, cryptographically-based security functions for both IPv4 and IPv6.

Two IPsec systems cannot interoperate unless they are using the same encryption and integrity authentication algorithms. Since insecure, incorrect, and/or misunderstood parameters can lead to a lack of interoperability or a vulnerable implementation, it was decided to specify algorithms and parameters that will help mitigate these problems.

IPsec has four basic components:

- Security Protocols
- Security Associations
- Key Management Methods
- Cryptographic Algorithms

And operates in two phases:

- Phase 1 negotiates the Security Associations (SAs) used to establish a single, reusable tunnel between two IPsec peers.
- Phase 2 uses the Phase 1 tunnel to negotiate Child SAs for transmitting user data.

To establish a secure tunnel, both peers must agree on the cryptographic algorithms and other security parameters that will be used to transmit data. Once negotiation is complete, each peer establishes a Security Association that defines the set of algorithms and security parameters. The IPsec "process" and the pieces that take part in it will be developed further in the sections that follow.

There have been three versions of IPsec, though just two of those are currently found in implementations: IPsec-v2 and IPsec-v3. Note that all IPsec versions are typically referred to simply as *IPsec*; distinguishing between them is done by determining the RFCs that specify them.

- IPsec-v2 is specified in IETF RFCs 2401, 2402, and 2406.
- IPsec-v3 is specified in IETF RFCs 4301, 4302, 4303, and 4835.

Though IPsec-v2 is still commonly used, it has been obsoleted by IPsec-v3. All new systems should implement IPsec-v3 since it contains many enhancements not found in either of the previous versions, including facilitating its use with other protocols. VICTORY implementations of IPsec shall use IPsec-v3.

10.3.8.2.1 Security Protocols

IPsec uses two protocols to provide traffic security services: Authentication Header (AH) and Encapsulating Security Payload (ESP). IPsec implementations must support ESP and may support AH. Both AH and ESP offer access control, enforced through the distribution of cryptographic keys and the management of traffic flows. These protocols may be applied individually or in combination with each other to provide IPv4 and IPv6 security services. However, most security requirements can be met through the use of ESP by itself. Each protocol supports two modes of use: transport mode and tunnel mode. In transport mode, AH and ESP provide protection primarily for next layer protocols; in tunnel mode, AH and ESP are applied to tunneled IP packets.

Source: IETF RFC 4301: "Security Architecture for the Internet Protocol"

VICTORY does not mandate that a certain mode be implemented.

10.3.8.2.1.1 Authentication Header

The IP Authentication Header (AH) is used to provide connectionless integrity and data origin authentication for IP datagrams and to provide protection against replays. AH provides authentication for as much of the IP header as possible, as well as for next level protocol data. However, some IP header fields may change in transit and the value of these fields, when the packet arrives at the receiver, may not be predictable by the sender. The values of such fields cannot be protected by AH. Thus, the protection provided to the IP header by AH is piecemeal.

Source: IETF RFC 4302: "IP Authentication Header"

According to NIST SP 800–57 Part 3 Revision 1, Authentication Header should not be used in IPsec-v3, therefore, it should not be used in VICTORY implementations.

10.3.8.2.1.2 Encapsulating Security Payload

The Encapsulating Security Payload (ESP) header is designed to provide a mix of security services in IPv4 and IPv6. ESP can be used to provide confidentiality, data origin authentication, connectionless integrity, an anti-replay service (a form of partial sequence integrity), and (limited) traffic flow confidentiality. Using encryption-only for confidentiality is allowed by ESP. However, it should be noted that, in general, this will provide defense only against passive attackers. Using encryption without a strong integrity mechanism on top of it (either in ESP or separately via AH) may render the confidentiality service insecure against some forms of active attacks.

Source: IETF RFC 4303: "IP Encapsulating Security Payload (ESP)"

Encapsulating Security Payload shall be used in a VICTORY implementation of IPsec-v3, and it shall provide both encryption/ confidentiality and integrity authentication.

10.3.8.2.2 Security Associations

Security Associations (SAs) are at the heart of IPsec. An SA is defined for a single unidirectional flow of data from a sending endpoint to a receiving endpoint; except in the case of multicast, where there could be multiple receiving endpoints. Each SA is uniquely identifiable. Both AH and ESP make use of SAs, and a major function of the Internet Key Exchange (IKE) protocol is the creation and maintenance of SAs. An SA is the establishment of a secure transmission session between two or more network entities that describes how each entity will use specific cryptographic algorithms and keys to communicate securely. SAs are valid for a specified period of time, a certain volume of traffic, or a combination of both.

Every Security Association contains information about the type of traffic it will protect, including:

- The IP addresses of the two endpoints.
- The type of encryption and/or authentication that is provided.
- The keys that are used to protect the data.
- How often the keys are refreshed.
- An identifier called a *Security Parameter Index (SPI)* that is used to uniquely identify the SA.

In IPsec-v3, a unicast SA is uniquely identified by the SPI and, optionally, by the ESP protocol; a multicast SA is identified by a combination of the SPI and the destination address and, optionally, the source address. Configuring SAs is implementation-specific and not described in this specification. Refer to the section titled "Protection for Multicast Traffic" and IETF RFC 4301 for more information.

10.3.8.2.2.1 Security Association-related Databases

The IPsec model presented in IETF RFC 4301 includes three databases that help with the processing of SA traffic. These include:

- **Security Policy Database (SPD)** – specifies the policies that determine how inbound and outbound traffic will be processed.
- **Security Association Database (SAD)** – contains the parameters that are associated with each established SA.
- **Peer Authorization Database (PAD)** – contains information necessary to conduct peer authentication, providing a link between the SPD and the SA management protocol (e.g., IKE).

Each peer maintains its own set of databases. Configuring these databases is implementation-specific and not defined in this specification. Refer to IETF RFC 4301 for more information.

10.3.8.2.3 Key Management Methods

Two classes of key management methods are specified for IPsec: manual keying and automated keying. Manual keying involves an agreement (in an unspecified manner) by the parties in a communication on the IPsec protections to be applied and the symmetric keys to be used. This has a major downside in that it severely limits the scalability of the security solution and requires rekeying to be done in an unspecified manner. An SA and its secret keys cannot be easily renewed in cases where the SA expires, has been used for the maximum allowable volume of traffic, or if its keys have been compromised.

To use automated keying, an automated negotiation between peers prior to exchanging IPsec-protected traffic determines the IPsec protections to be applied and the symmetric keys to be used. The same method can be used to maintain, delete, or renegotiate the SA (e.g., to rekey). This approach permits a decoupling of the key management mechanism from the other security mechanisms, thus facilitating the use of alternative key management methods without having to modify other security mechanisms.

Source: NIST SP 800–57 Part 3: "Recommendation for Key Management, Part 3: Application-Specific Key Management Guidance – Revision 1"

A manually generated key never expires. While manual keying can generate a false sense of security, it ultimately provides weaker security than automated keying and requires a large amount of coordination between peers. Manual keying also reduces the flexibility and options of IPsec: anti-replay, on-demand rekeying, and session-specific key management are not available. Automated keying, on the other hand, provides for widespread, automated distribution of keys and may generate several keys for a single SA. Since automatic key distribution is required for IPsec-v3, automated keying shall be used in a VICTORY implementation, **except** as specified, if needed, in the section titled "Protection for Multicast Traffic."

10.3.8.2.3.1 Internet Key Exchange

The Internet Key Exchange (IKE) protocol was designed specifically for use with IPsec. IKE is the default and preferred key management protocol for IPsec, and is used for source authentication, to negotiate, create, and manage SAs, and to negotiate authenticated keying material for use within those SAs. All IKE communications consist of pairs of messages – a request and a response. The message pair is commonly referred to as an "exchange." IKE utilizes a two-phase negotiation process and uses pseudo-random functions (PRFs) to generate the secret keys that are used in IKE SAs and Child (IPsec) SAs. IKE handles key management automatically and generates the necessary keying material for IPsec via an authenticated secure channel between the two IKE peers.

There are two versions of IKE in use: IKEv1 and IKEv2. Even though both versions perform mutual authentication and establish and maintain SAs, they are not interoperable. IKEv2 was designed to be more reliable and efficient than IKEv1, and addressed a number of IKEv1 weaknesses. IKEv2 also provides a built-in capability to establish and maintain connections across gateways that implement network address translation (NAT-Traversal). Since the cryptographic mechanisms used by IKEv2 are very similar to what ESP uses to protect IPsec packets, VICTORY specifies that IKEv2 shall be used, **except** as specified, if needed, in the section titled "Protection for Multicast Traffic."

10.3.8.2.4 Cryptographic Algorithms

A cryptographic algorithm is a mathematical function that can turn plaintext into ciphertext (encryption), and ciphertext into plaintext (decryption). Whenever security services are required, FIPS-approved or NIST-recommended cryptographic algorithms shall be used in VICTORY to support them. These algorithms have undergone an intensive security analysis and continue to undergo regular examination to determine whether they are still able to provide adequate security or not. Most cryptographic algorithms require cryptographic keys or other keying material. In some cases, increasing the size of the key can strengthen the algorithm.

Approved cryptographic algorithms fall into one of three classes:

- **Cryptographic hash functions** – Functions that map a bit string of arbitrary (although bounded) length to a fixed-length bit string.
- **Symmetric-key algorithms** – also known as secret-key algorithms – Cryptographic algorithms that transform data in a way that is fundamentally difficult to undo without knowledge of a *secret key*. The key is "symmetric" because the same key is used for a cryptographic operation and its inverse (e.g., encryption and decryption).
- **Asymmetric-key algorithms** – also known as public-key algorithms – Cryptographic algorithms that use two related keys, a *public key* and a *private key*. The keys work together such that a message encrypted by the public key can be decrypted by the corresponding private key. A property of the two keys is that determining the private key from the public key is not computationally feasible. Elliptic curve algorithms are a subset of this class.

The proper management of cryptographic keys is essential to the effective use of cryptography for security; both access to keys and the proliferation of keys need to be controlled. A key should only be accessible by an authorized entity, and only for the purpose for which it is authorized. Refer to NIST SP 800–133 for additional information about generating keys.

In a VICTORY implementation, the Advanced Encryption Standard (AES) shall be implemented as the encryption algorithm for ESP. While AES supports three key sizes, 256-bit keys shall be used. A number of confidentiality modes of operation can be used with AES; the modes suitable for VICTORY include Cipher Block Chaining (CBC) and Galois/Counter mode (GCM). Although ESP does not require that an *Integrity Check Value (ICV)* be present, AES-GCM does not allow a zero-length ICV.

> ***NOTE:*** *AES-GCM is a combined-mode algorithm, providing both encryption and integrity protection. Under many circumstances, combined-mode algorithms provide significant efficiency and throughput advantages over algorithms that provide only a single service. AES-GCM shall not be used with manually distributed keys.*

10.3.8.2.4.1 Cryptographic Suites

IPsec allows cryptographic algorithms to be selected individually. However, in an effort to promote interoperability in IPsec implementations, the National Security Agency (NSA) combined cryptographic algorithms that provide encryption, hashing, digital signature, and key exchange into a set called *Suite B Cryptography*. Suite B is a subset of the NIST-recommended algorithms presented in NIST SP 800–57 Part 3 Revision 1. While there are many factors that need to be considered in determining the best way to configure an IPsec implementation, Suite B only specifies cryptographic algorithms. Implementing the suites as a whole does not change the IPsec protocols in any way. The suites are specified in IETF RFC 6379, while the appropriate conventions for using the suites are specified in IETF RFC 6380. Suite B requires that key establishment and signature algorithms be based upon Elliptic Curve Cryptography and that the encryption algorithm be AES.

Perceiving a threat to cryptographic algorithms from quantum computers (see the section titled "The Quantum Computer Threat"), the NSA announced a second set of cryptographic algorithms that it believes will be better able to resist quantum computer attacks than those specified in Suite B. This set, known as the *Commercial National Security Algorithm (CNSA) Suite*, is approved to protect National Security Systems (NSS) up to (and including) the TOP SECRET level. More information on the NSA's cryptographic algorithm strategy and the CNSA Suite is available at: https://apps.nsa.gov/iaarchive/programs/iad-initiatives/cnsa-suite.cfm.

10.3.8.2.4.2 Protection for Multicast Traffic

For multicast traffic, there are multiple destination systems associated with a single SA. Some system or person will need to coordinate among all multicast groups to select an SPI or SPIs on behalf of each multicast group and then communicate the group's IPsec information to all of the legitimate members of that multicast group. Multiple senders to a multicast group should use a single SA (and hence SPI) for all traffic to that group when a symmetric key encryption or integrity algorithm is employed.

Source: IETF RFC 4301

As stated in NIST SP 800–119, "The current version of IPsec does not fully support protection for multicast traffic, because IPsec was designed specifically for protecting communications between two points, not among many points at once. Each multicast packet may have many recipients, which raises many IPsec-related issues." That fact notwithstanding, where there is an expressed requirement or desire to protect data transmissions within the platform that use multicast, the IPsec implementation must support multicast SAs and be used in accordance with IETF RFC 4301. Since IKE is inherently a two-party protocol, a key management/exchange algorithm/technique other than IKE may be utilized in an implementation that requires the use of multicast messaging.

While the IETF looks into how to modify IKE to better support multicast, one option that should work with multicast is to combine the protocols, techniques, and methods described in the following RFCs to build a *non-IKE* alternative. Manually configured SAs may be used in a non-IKE configuration.

- IETF RFC 2408: Internet Security Association and Key Management Protocol (ISAKMP)
- IETF RFC 2627: Key Management for Multicast: Issues and Architectures
- IETF RFC 3740: The Multicast Group Security Architecture
- IETF RFC 4301: Security Architecture for the IP Protocol
- IETF RFC 5374: Multicast Extensions to the Security Architecture for the Internet Protocol
- IETF RFC 6407: The Group Domain of Interpretation

Approved algorithms that should provide IKE-like cryptographic capabilities in a multicast messaging environment using ISAKMP (a subset of IKEv1) or another non-IKE protocol are listed in Table 181. **When a multicast-compatible version of IKE is released, it should be considered for incorporation into VICTORY multicast implementations.**

10.3.8.2.5 IPsec in IPv6 vs. IPsec in IPv4 <Informational>

All major aspects of IPsec under IPv6 are the same as they are under IPv4. IPsec is defined for both protocols in the same specifications. AH, ESP, tunnel mode, transport mode, SAs, and other aspects of the protocol are identical to those found within the IPv4 version. Coverage of the IP header in transport mode, ICMP selectors, and fragmentation handling are different.

Once IPv6 is widely deployed, it is more likely to be deployed with end-to-end addressing rather than with NAT. Thus, end-to-end IPsec is more feasible with IPv6. Work in the IETF on using IPsec to secure applications and on IPsec application program interfaces (APIs) may prove to be more useful in end-to-end IPv6 environments.

IPsec is a major component of IPv6 security and should be deployed, wherever possible, to secure IPv6 networks. Transition mechanisms allow existing IPv4 networks to coexist and interoperate with IPv6 networks, systems, and services.

Source: NIST SP 800–119: "Guidelines for the Secure Deployment of IPv6"

Since IPv4 and IPv6 have different datagram formats and addressing, there are some differences in the datagram formats used for AH and ESP in both tunnel and transport modes. For more information and illustrations on how those differences affect ESP, refer to IETF RFC 4303.

10.3.8.3 Implementing IPsec

The VICTORY Architecture calls for the use of IPsec protocols when a network layer protection mechanism is desired. Therefore, the parameters for the ESP and IKE protocols are specified. Since the ESP and IKE protocols are quite complicated, the interested reader is referred to NIST SP 800–77 and IETF RFC 7296, respectively, for additional information.

The IPsec "process" can be broken down into five main steps. In these steps, IKE refers to IKEv2; because of that, the *"IKE Phase X"* steps may differ if a multicast messaging environment using ISAKMP or a Non-IKE protocol has been implemented. The cryptographic algorithms listed are specified in Table 180 and Table 181. Every IPsec request message requires a response.

1. **Tunnel initiation:** An "IPsec tunnel" is initiated when a network layer protection mechanism is desired by two or more IPsec peers/endpoints. This starts the IKE process.
2. **IKE Phase 1:** Two request/response message pairs are exchanged between peers.
    a. **Message Pair 1:** Neither the request nor response is protected. The request contains the SPI, cryptographic algorithm proposals, a Diffie-Hellman shared secret, and a nonce (a randomly chosen number used as input to a cryptographic algorithm). The response confirms the cryptographic algorithms to be used and also includes a Diffie-Hellman shared secret and a nonce. This establishes an IKE SA.
        i. ***HMAC-SHA*** is used as the pseudo-random function to generate a hash.
        ii. A ***Diffie-Hellman ECP group*** uses the hash to generate a shared secret and to pass nonces. Encryption and integrity authentication keys are derived from the shared secret.
        iii. ***HMAC-SHA*** is used to provide integrity authentication.
        iv. ***ECDSA*** is used to sign a block of data and the other peer's nonce to authenticate the SA.
        v. ***AES-CBC*** is the encryption algorithm to be used in future messages.
    b. **Message Pair 2:** The second message pair exchanges the identities of the two peers and certificates. These messages authenticate the first message pair, validate the identities of the two peers, and establish the first Child SA (for ESP). Parts of these messages are encrypted using the algorithms agreed upon in the first message pair.
3. **IKE Phase 2:** A request/response message pair is exchanged to perform one of the following actions: create new Child SAs, rekey IKE SAs, rekey Child SAs, delete/close IKE SAs, delete/close Child SAs, or verify that a peer is still alive. Closing an IKE SA closes all associated Child SAs. Either peer may initiate this exchange. Data is transferred between peers using the Child SAs.

    NOTE: *A Diffie-Hellman group may be negotiated for a Child SA, which can provide perfect forward secrecy (PFS) for the Child SA keys.*

4. **Data transfer:** Inbound and outbound network traffic is encapsulated according to the set of algorithms and parameters provided by their respective negotiated Child SAs. These algorithms and parameters provide encryption and integrity authentication.
    a. ***AES-GCM*** is used to encrypt and provide integrity authentication to the data in a unicast messaging environment.
    b. ***AES-CBC*** and ***HMAC-SHA*** are used to encrypt and provide integrity authentication, respectively, to the data in a multicast messaging environment.
5. **Tunnel termination:** The IPsec tunnel is terminated when its Child SAs stop flowing, whether through deletion, rekeying, or timing out.

10.3.8.3.1 IPsec Configuration Parameters

The biggest security problem with IPsec results from selecting security services or parameters poorly. To avoid this, VICTORY specifies the use of a limited set of FIPS-approved or NIST-recommended cryptographic algorithms. The approved algorithms, along with their corresponding minimum key sizes, that shall be used to protect data in a VICTORY IPsec implementation are shown in Table 180 and Table 181. Note that Table 181 does not contain a specific protocol for cryptographic key exchange.

NOTE: *Not all implementations will require the use of all the algorithms listed in Table 180 and Table 181.*

The cryptographic algorithms shown are based upon a minimum "strength of security" indicated in the *Parameters* column. The strength of security determines the algorithm key length that must be used, which results in equivalent or greater key lengths for other cryptographic functions. This security model has been adopted by standards organizations including the IETF and NIST. The algorithms that are specified for ESP are used to provide IPsec protection to data traffic; the algorithms that are specified for IKE are used to protect IKE's own traffic.

Table 180. Cryptographic Algorithms – Unicast Messaging

IPsec Protocol	Cryptographic Service	Algorithm	Parameters	Suite B	CNSA Suite	Specifications
ESP	Encryption and Integrity Authentication	AES	256-bit keys, 16-octet ICV, GCM mode	Yes	Yes	IETF RFC 4106, NIST SP 800–38D, FIPS 197
IKEv2	Encryption	AES	256-bit keys, CBC mode	Yes	Yes	IETF RFC 3602, NIST SP 800–38A, FIPS 197
	Integrity Authentication	HMAC-SHA–384–192	384-bit keys	Yes	No	IETF RFC 4868, NIST SP 800–107, FIPS 198–1
	Pseudo-random Function	HMAC-SHA–384	384-bit keys (>= AES key size)	Yes	No	IETF RFC 4868, NIST SP 800–107, FIPS 198–1
	Source Authentication	ECDSA–384	Curve P–384	No	Yes	IETF RFC 4754, FIPS 186–4
	Key establishment and exchange	Elliptic Curve Diffie-Hellman Key Exchange	Curve P–384 (ECP group 20)	Yes	Yes	IETF RFC 5903, NIST SP 800–56A

Table 181. Cryptographic Algorithms – Multicast Messaging

IPsec Protocol	Cryptographic Service	Algorithm	Parameters	Suite B	CNSA Suite	Specifications
ESP	Encryption	AES	256-bit keys, CBC mode	No	Yes	IETF RFC 3602, NIST SP 800–38A, FIPS 197

IPsec Protocol	Cryptographic Service	Algorithm	Parameters	Suite B	CNSA Suite	Specifications
	Integrity Authentication	HMAC-SHA–384–192	384-bit keys	Yes	No	IETF RFC 4868, NIST SP 800–107, FIPS 198–1
ISAKMP OR Other Non-IKE Protocol OR Multicast-compatible IKE	Encryption	AES	256-bit keys, CBC mode	Yes	Yes	IETF RFC 3602, NIST SP 800–38A, FIPS 197
	Integrity Authentication	SHA–384	384-bit digest	No	Yes	FIPS 180–4
	Integrity Authentication	HMAC-SHA–384–192	384-bit keys	Yes	No	IETF RFC 4868, NIST SP 800–107, FIPS 198–1
	Pseudo-random Function	HMAC-SHA–384	384-bit keys (>= AES key size)	Yes	No	IETF RFC 4868, NIST SP 800–107, FIPS 198–1
	Source Authentication	ECDSA–384	Curve P–384	No	Yes	IETF RFC 4754, FIPS 186–4
	Source Authentication	RSA (with SHA–384)	3072-bit modulus	No	Yes	NIST SP 800–56B, FIPS 186–4
	Key establishment and exchange	Elliptic Curve Diffie-Hellman Key Exchange	Curve P–384 (ECP group 20)	Yes	Yes	IETF RFC 5903, NIST SP 800–56A

ESP Encryption and Integrity Authentication are achieved using either one or two algorithms, depending on the method of IP messaging.

1. With **unicast** messaging, AES-GCM is used to achieve both encryption and integrity authentication. Since AES-GCM provides integrity protection, a 16-octet ICV is stored in the ESP ICV field. Automated keying shall be used with this algorithm/mode. Refer to FIPS 197 for the set of technical security requirements to which the implementation and use of the AES algorithm must conform.
2. With **multicast** messaging, AES-CBC and HMAC-SHA are used together to achieve encryption and integrity authentication, respectively. AES-CBC can be used with either automated or manually distributed keys. Refer to FIPS 197 for the set of technical security requirements to which the implementation and use of the AES algorithm must conform. Refer to FIPS 198–1 for the set of technical security requirements to which the implementation of Keyed-Hash Message Authentication Code (HMAC) must conform.

IKEv2 Encryption is achieved using AES-CBC. Refer to FIPS 197 for the set of technical security requirements to which the implementation and use of the AES algorithm must conform.

IKEv2 Integrity Authentication is achieved using HMAC-SHA. Refer to FIPS 198–1 for the set of technical security requirements to which the implementation of HMAC must conform.

IKEv2 Pseudo-random Function (PRF) is a multi-purpose source of pseudo-randomness used for generating the keying material for all of the cryptographic algorithms used and for authentication of the IKE SA. All PRFs used with IKE must allow for variable-sized keys. IETF RFC 4868 describes the PRF that must be used. Refer to FIPS 198–1 for the set of technical security requirements to which the implementation of HMAC must conform.

IKEv2 Source Authentication is achieved using the elliptic curve digital signature algorithm (ECDSA). There are two methods for authenticating the two endpoints to each other: Pre-Shared Key and Digital Signature. Each endpoint has its own digital certificate that contains a public key. One endpoint uses the corresponding private key to digitally sign data before sending it to the other endpoint, which verifies the signature using the peer's public key. ECDSA cannot be specified independent of an associated hash function; the hash function needs to be specified as part of the signature algorithm. In addition to that, the choice of hash function depends on the elliptic curve group specified. As a result, specifying ECDSA–384, HMAC-SHA–384, and Diffie-Hellman 384-bit random ECP group 20 together represents an instantiation of ECDSA with 384-bit security strength. Refer to FIPS 186–4 for the set of technical security requirements to which the implementation and use of the ECDSA algorithm must conform.

NOTE: *On binary messages, a Base64 signature may not make sense. An implementation example of this scenario was to append the digital signature of the HMAC digest to the message after converting it to binary form. The length must correspond to the chosen signature strength: 96 octets for HMAC-SHA–384[–192].*

IKEv2 Elliptic Curve Diffie-Hellman (DH) group is used to generate a shared secret for the endpoints in a secure manner so that an observer of the IKE SA exchange cannot determine the shared secret. This shared secret is then used to generate a value that is used as input to the calculations for the Phase 1 and Phase 2 secret keys. Each DH group number corresponds to a key length and an encryption generator type. Groups using elliptic curves are more efficient than prime modulus (non-elliptic curve) groups; therefore, group 20 has been specified. Refer to IETF RFC 5903 for more information about this group.

10.3.8.3.2 Other IPsec Configuration Considerations

In addition to endpoint security, there are many other possible design aspects that need to be considered in an IPsec implementation. Table 182 describes several specific IPsec settings that were not addressed earlier.

Table 182. Other IPsec Configuration Considerations

Design Aspect	Description
Security Association (SA) Lifetimes	In IKEv2, each SA endpoint is responsible for enforcing its own lifetime policy and rekeying the SA when necessary. Generally, a shorter/smaller SA lifetime is synonymous with more secure communications, but with a higher cost. If the two lifetime policies are different, the endpoint with the shorter lifetime will always end up initiating rekeying. Conversely, if the two lifetime policies are the same, it is possible that both endpoints will initiate rekeying at the same time, resulting in redundant SAs. For VICTORY, the policies should include limits for both time interval (lifetime) and bytes of traffic (lifesize) before rekeying.
Perfect Forward Secrecy (PFS)	PFS causes a new shared secret to be created through a Diffie-Hellman exchange for each IPsec SA. Although PFS increases the overhead in establishing SAs, it provides greater security because each shared secret is unrelated to any others, both previous and subsequent. If a shared secret were to be compromised, the attacker would not be able to use it to gain access to data protected by any other SA. To this end, if PFS is configurable in a VICTORY implementation, it shall be enabled or set to 'on.'
Reuse of Diffie-Hellman Exponentials	Because computing Diffie-Hellman exponentials is computationally expensive, an endpoint may find it advantageous to reuse those exponentials for multiple connection setups. There are several reasonable strategies for doing this. An endpoint could choose a new exponential only periodically, though this could result in less-than-perfect forward secrecy if some connection lasts for less than the lifetime of the exponential. Or it could keep track of which exponential was used for each connection and delete the information associated with the exponential only when some corresponding connection was closed. This would allow the exponential to be reused without losing perfect forward secrecy at the cost of maintaining more state. Whether and when to reuse Diffie-Hellman exponentials are private decisions in the sense that they will not affect interoperability. An implementation that reuses exponentials may choose to remember the exponential used by the other endpoint on past exchanges and if one is reused to avoid the second half of the calculation. *Source: IETF RFC 7296: "Internet Key Exchange Protocol Version 2 (IKEv2)"*

10.3.8.4 The Quantum Computer Threat

As computers become faster and more powerful, cryptographic algorithms and their corresponding keys (key sizes) will need to be adjusted to stay ahead of adversaries trying to decrypt messages. This could mean several things:

1. Algorithms that are currently FIPS-approved or NIST-recommended get removed from those lists.
2. Algorithms that are currently FIPS-approved or NIST-recommended require a larger key size to maintain a satisfactory level of data security.
3. New algorithms get added to the FIPS-approved or NIST-recommended lists.

Quantum computers, should they become functional, could be capable of undermining all current cryptographic algorithms, rendering them unusable. In preparation, the effort to develop quantum resistant algorithms has begun. In the NSA's assessment, all asymmetric-key algorithms currently in use will need to be tested to determine whether or not they can withstand an attack from a quantum computer. Their current belief is that symmetric-key algorithms are not as susceptible to quantum computer attacks, especially with larger key sizes.

While quantum resistant algorithms are developed and tested, the algorithms listed in Table 180 and Table 181 are specified for VICTORY. These tables consist of algorithms from Suite B, the CNSA Suite, and NIST SP 800–57 Part 3 Revision 1 for ESP and IKE that can be used together to protect IP traffic. **The key and curve size parameters shown should be considered minimums; it is acceptable to use stronger/larger ones, if available and necessary.** The use of any combination of algorithms from Table 180 and Table 181 that provides both encryption and integrity authentication for ESP, and meets the requirements stipulated in Table 183, shall constitute VICTORY-compliance.

As quantum resistant algorithms become FIPS-approved or NIST-recommended, they should be incorporated into VICTORY IPsec implementations even if they are not listed in Table 180 or Table 181, as long as they have been approved for use by your local security authority. Refer to the NSA's Information Assurance website for the latest news and updates.

NIST SP 800–131A and an article published by Cisco titled *"Next Generation Encryption"* contain recommendations for selecting secure cryptographic algorithms, migrating from older/legacy algorithms to newer/more secure ones, and transitioning from smaller to larger key sizes. Also identified are cryptographic algorithms to avoid, algorithms being deprecated, and algorithms that are considered quantum computer resistant.

10.3.8.5 Summary

To summarize the recommendations made throughout this specification, Table 183 contains the minimal IPsec configuration that shall be implemented for VICTORY-compliance. Refer to Table 180 and Table 181 for the corresponding cryptographic algorithms.

A particular implementation may need to utilize features of IPsec (protocols, parameters, configurations, etc.) that are not listed in Table 183 (e.g., Transport vs. Tunnel Mode in ESP). Integrating these other features will not impact VICTORY-compliance as long as they are used ***in addition to*** the features listed below **and** interoperability is maintained.

Table 183. IP Layer Security Protocols Specified by VICTORY

Component	Protocol	Required
Secure IP Messaging	Internet Protocol Security (IPsec)	• *Version:* IPsec-v3
Security Protocol	Encapsulating Security Payload (ESP)	• *Version:* ESP-v3 (part of IPsec-v3) • *Service:* Encryption/Confidentiality • *Service:* Integrity Authentication
Key Management	Internet Key Exchange (IKE)	• *Version:* IKEv2 • *Traffic Type:* Unicast • *Method:* Automated Keying • *Service:* Encryption/Confidentiality • *Protection:* Cryptographic Algorithms / Key size combinations >= 256-bits • *Protection:* Perfect Forward Secrecy (PFS) Enabled / Set to 'on', if applicable

Component	Protocol	Required
	Multicast-compatible IKE OR Non-IKE Protocol	• *Version:* If IKE, latest IKE version that supports multicast • *Traffic Type:* Multicast • *Method:* Automated or Manual Keying • *Service:* Encryption/Confidentiality, if available • *Protection:* Cryptographic Algorithms / Key size combinations >= 256-bits, if applicable • *Protection:* PFS Enabled / Set to 'on', if applicable
Cryptographic Algorithms	FIPS-approved or NIST-recommended	• Algorithms listed in Table 180 or Table 181 • Quantum resistant algorithms, if available

10.4 Transport Layer Protocols

To support the communication of data across the network, the VDB shall implement each of the following transport layer protocols.

10.4.1 Transmission Control Protocol <01026–20120725, Pro>

The VDB shall support Transmission Control Protocol (TCP) by conforming to the following core TCP standard:

- IETF RFC 793: Transmission Control Protocol

The VDB shall conform to the following standard for TCP congestion control mechanisms:

- IETF RFC 2581: TCP Congestion Control

NOTE: *Some clarifications regarding the manner in which the Ethernet switch component is expected to support this specification can be found in the section titled "Applicability of Internet Protocol Specifications to the Switch Component."*

10.4.2 User Datagram Protocol <01027–20120725, Pro>

The VDB shall support User Datagram Protocol (UDP) by conforming to the following core UDP standard:

- IETF RFC 768: User Datagram Protocol

One of the core tenets of the VICTORY concept is that data will be shared between applications and components across a common bus. The data will be shared in a way that network traffic is kept to the minimum possible set.

NOTE: *Some clarifications regarding the manner in which the Ethernet switch component is expected to support this specification can be found in the section titled "Applicability of Internet Protocol Specifications to the Switch Component."*

10.4.3 Advanced Routing Concepts

The purpose of this subsection is to specify advanced routing topics such as network address translation (NAT) and the use of IPv4 private network address space.

10.4.3.1 IPv4 Private Address Space <01066–20120725, Pro>

The VICTORY Data Bus shall allow the use of IPv4 private address space as defined in IETF RFC 1918.

NOTE: *Some clarifications regarding the manner in which the Ethernet switch component is expected to support this specification can be found in the section titled "Applicability of Internet Protocol Specifications to the Switch Component."*

10.4.3.2 IPv6 Private Address Space <01067–20120725, Pro>

VICTORY Data Buses that support IPv6 shall allow the use of IPv6 Unique Local Address (ULA) space as defined in IETF RFC 4193.

NOTE: *Some clarifications regarding the manner in which the Ethernet switch component is expected to support this specification can be found in the section titled "Applicability of Internet Protocol Specifications to the Switch Component."*

10.4.3.3 IPv4 Network Address Translation <01068–20120725, Pro>

The VICTORY Data Bus shall allow for the use of Network Address Translation (NAT) as defined in IETF RFC 2663.

10.4.4 VDB Subnetting Restrictions <01029–20120725, Pro>

It is required that a VDB shall have at least one subnet. In addition, it is required that a single subnet shall not encompass more than one VDB.

10.4.5 Virtual Local Area Network (VLAN) <01077–20120131, Exp>

A Virtual LAN (VLAN) is a logical collection of network nodes that are grouped together into a single broadcast domain. VLANs can be used to separate connected nodes from the rest of the network, making communication between these nodes more secure. VLANs can also be used to restrict access to certain network resources. VLANs are mainly Layer 2 constructs which allow multiple Layer 3 networks to exist on the same switch.

In order to support VLANs on the VDB, a device shall implement the IEEE 802.1Q standard.

10.4.6 Multicast Routing Concept <Informational>

The method by which VICTORY will route data traffic in an optimal way is through the use of the multicast routing paradigm. From a broad perspective, multicast routing specifies that data only go where it is requested.

From the perspective of a data publisher, this means that data is only sent once to a special address called a multicast group. From the perspective of a data consumer, this means that to receive desired data, one must subscribe to the multicast group to which it is being published. For the routing nodes in the middle, this means keeping track of which consumers are interested in which groups, and only forwarding data along those links. In this way, only the minimum amount of traffic is sent and received by all publishers and consumers.

In practice, this is a bit more complicated. There are, for instance, at least two common ways that multicast routing paths, or trees, can be initialized. Each method has strengths and weaknesses. Furthermore, different subscription protocols have been specified for IPv4 and IPv6, and there is more than one version of each.

There are some caveats to consider with respect to multicast. For instance, multicast only works with stateless protocols. For this reason, the VDB will only provide multicast functionality for UDP. Additionally, multicast is prescribed at the Open Systems Interconnection (OSI) third layer, and so in the general case of a second layer (switch-only) network, it functions no differently than broadcast traffic. There are methods that may be employed to mitigate the inefficiency of broadcast traffic, and these methods are discussed in the sections below.

The following sections describe specific advice and requirements for the VDB, as well as the list of relevant RFCs that must be followed for each of the multicast-related technologies being prescribed.

10.4.7 Multicast Group Subscription Management

There are two main protocols used for subscribing multicast clients to a multicast group. One protocol is part of the IPv4 suite, and the other is only used with IPv6. The technologies are the IGMP and the MLD Protocols, respectively.

10.4.7.1 Internet Group Management Protocol (IGMP) <01075–20120725, Pro>

A VDB shall maintain the capability to forward dynamically configured IPv4 multicast datagrams. In order for the VDB as a whole to support this functionality, all VICTORY-compliant routers and devices with routing capability shall conform to the Internet Group Management Protocol (IGMP) version 3 standard as described in Table 184.

Table 184. IGMP (IPv4) Multicast Routing Standards

Applicability	Specification
Required	IETF RFC 3376 (Standards Track): Internet Group Management Protocol, Version 3
Informational	IETF RFC 3569 (Informational): An Overview of Source-Specific Multicast (SSM)
Required	IETF RFC 4604 (Standards Track): Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast

10.4.7.2 IGMP Forwarding Proxy <01087–20180215, Pro>

The VICTORY Data Bus shall support the use of IGMP forwarding proxy as defined in IETF RFC 4605: Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying").

10.4.7.3 Multicast Listener Discovery (MLD) <01076–20140630, Pro>

A VDB that supports IPv6 shall maintain the capability to forward dynamically configured IPv6 multicast datagrams. In order for the VDB as a whole to support this functionality, all VICTORY-compliant routers and devices with routing capability shall conform to the Multicast Listener Discovery (MLD) Protocol version 2 standard as described in Table 185.

Table 185. MLD (IPv6) Multicast Routing Standards

Applicability	Specification
Required	IETF RFC 2710 (Standards Track): Multicast Listener Discovery (MLD) for IPv6
Informational	IETF RFC 3569 (Informational): An Overview of Source-Specific Multicast (SSM)
Required	IETF RFC 3810 (Standards Track): Multicast Listener Discovery Version 2 (MLDv2) for IPv6
Required	IETF RFC 4604 (Standards Track): Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast
Recommended	IETF RFC 3590 (Standards Track): Source Address Selection for the Multicast Listener Discovery (MLD) Protocol

10.4.7.4 MLD Forwarding Proxy <01092–20180215, Exp>

The VICTORY Data Bus shall support the use of MLD forwarding proxy as defined in IETF RFC 4605: Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying").

10.4.8 Layer 2 Multicast Routing

As mentioned previously, IGMP and MLD are routing-related protocols, so they typically work only with layer 3 devices (e.g., routers). In order to support the benefits of multicast for layer 2 devices (e.g., switches), two additional technologies can be supported. The first is layer 2 static routing, and the second is snooping, which comes in two varieties, one for each version of the IP protocol.

10.4.8.1 IGMP Snooping in IPv4 <01024–20120725, Pro>

IGMP Snooping is a technique for extending the knowledge of end node multicast group subscriptions to layer 2 devices. A switch device 'snoops' by monitoring layer 3 IGMP group subscription request messages, and builds dynamic routing tables based upon them. It should be noted that not all devices that support snooping can do so at "wire speed." Any VICTORY device that supports IGMP snooping should do so at wire speed.

It is recommended that VICTORY-compliant layer 2 network switches that comprise the VDB support IGMP snooping for IPv4 traffic as presented in Table 186.

Table 186. Multicast Snooping Standards

Applicability	Comment	Specification
Informational	IPv4, Layer 2 Snooping recommended for switches at wire speed	IETF RFC 4541 (Informational): Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches

NOTE: *IGMP snooping is recommended for performance considerations in a dynamically configured IPv4 multicast environment, but there are some drawbacks to the technique. See the section titled "IGMP and MLD without a Router" for an elaboration of these drawbacks and their mitigations.*

10.4.8.2 MLD Snooping in IPv6 <01060–20120131, Exp>

MLD Snooping is a technique for extending the knowledge of end node multicast group subscriptions to layer 2 devices which support IPv6. A switch device 'snoops' by monitoring layer 3 MLD group subscription request messages, and builds dynamic routing tables based upon them. It should be noted that not all devices that support snooping can do so at "wire speed." Any VICTORY device that supports MLD snooping should do so at wire speed.

It is recommended that VICTORY-compliant layer 2 network switches that comprise the VDB support MLD snooping for IPv6 traffic as presented in Table 186.

NOTE: *MLD snooping is recommended for performance considerations in a dynamically configured IPv6 multicast environment, but there are some drawbacks to the technique. See the section titled "IGMP and MLD without a Router" for an elaboration of these drawbacks and their mitigations.*

10.4.8.3 IGMP and MLD without a Router <Informational>

The Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) protocols are used within a group of routed network subnets to automatically build up IPv4 and IPv6 multicast routing tables within IP routing devices. Within each of these routed subnets there may be one or more Ethernet switch devices. Generally, such devices work at the Ethernet layer, and do not understand that the IP traffic encapsulated in the Ethernet frames may be multicast. In such a case, the multicast traffic is broadcast to all clients within the switch device's broadcast domain. This can defeat much of the routing efficiency provided by the use of multicast on such networks, particularly networks that are extensively switched, and not routed.

There are a few common approaches to this problem. The first is to set up static multicast routes within the Ethernet switch devices. Although Ethernet switch devices are minimally layer 2 in their area of concern, many so-called "layer 3" Ethernet switch devices provide some capability to assign static routes to various multicast groups (addresses). On a network where all multicast services and clients are known *a priori*, this can be an effective mitigation. It is especially applicable to isolated networks that do not contain a router.

Static multicast routing essentially bypasses the use of IGMP and MLD, at least within a switched subdomain, but if the flexibility provided by those protocols is needed then a second approach is the use of IGMP and MLD snooping. Snooping allows the advantages of multicast routing to be extended to an Ethernet switch automatically, but it does have one major caveat. In order for IGMP and MLD snooping to function, there must be a router on the network. This can sometimes cause problems for isolated switch-only networks that want to take advantage of multicast routing.

The reason that IGMP and MLD snooping need a router is related to how the IGMP and MLD protocols operate. Both protocols require a dedicated network node, called an IGMP or MLD querier, that regularly polls a subnet for group interest. Without this regular poll, group memberships automatically time out, and multicast routes disappear. By protocol it is the router's job to serve this role as querier.

Interestingly enough, the necessity for a querier means that it is possible that enabling IGMP or MLD snooping on a network that lacks such a node will actually break a functioning multicast setup. This is because the default behavior of most network switches, where multicast is broadcast to all nodes, is typically disabled when snooping is enabled. A typical indication of this kind of malfunction is multicast that starts off working correctly then silently stops after a few minutes. This happens because each client's initial IGMP or MLD group request creates the correct entries in the switch's multicast routing table, but the entries expire when not regularly renewed by IGMP or MLD queries.

In order to mitigate this problem, many Ethernet switches that support IGMP and MLD snooping can also serve the role of querier, typically through an option that must be enabled (it is rarely enabled by default). Alternatively, a software-based IGMP and MLD querier can be set up to regularly poll the subnet.

10.4.9 Sparse Mode and Dense Mode

The core technologies typically used to build a multicast tree are Protocol Independent Multicast – Sparse Mode (PIM-SM) and Protocol Independent Multicast – Dense Mode (PIM-DM).

10.4.9.1 Protocol Independent Multicast - Sparse Mode (PIM-SM) <01073–20120725, Pro>

PIM-SM builds a multicast tree through the use of a rendezvous point (RP) for a group, to which all IGMP or MLD group subscription requests are sent. This RP serves as the logical center of the multicast tree, and all multicast data sent to the associated group address flows through it. Although it is not guaranteed to build a shortest-path tree, it is a very efficient method which ensures that a minimal amount of network traffic is generated for a multicast group.

Three additional things should be noted about PIM-SM. First, while the RP can be a single point of failure, it is possible to specify fallback options so that an additional RP can take over if the primary one goes down. Second, although the routing tree created through the use of an RP is not guaranteed to be the absolute shortest possible, there are methods by which such a tree can be transformed into a shortest path tree. Finally, PIM-SM is the primary method by which source-specific multicast is implemented.

PIM-SM is supported by both the IPv4 and IPv6 protocols, and so all VICTORY-compliant routers shall support PIM-SM as specified and described in Table 187.

Table 187. Platform Independent Multicast - Sparse Mode

Applicability	Specification
Required for IPv6	IETF RFC 3956 (Standards Track): Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address
Required	IETF RFC 4601 (Standards Track): Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)
Required	IETF RFC 5059 (Standards Track): Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)
Required	IETF RFC 5796 (Standards Track): Authentication and Confidentiality in Protocol Independent Multicast Sparse Mode (PIM-SM) Link-Local Messages

10.4.9.2 Protocol Independent Multicast - Dense Mode (PIM-DM) <01074–20120725, Pro>

PIM-DM follows an approach that is somewhat the opposite of that used by PIM-SM. Whereas in PIM-SM a client must specifically subscribe to a multicast group by sending a join message to the group's RP, in PIM-DM a client must specifically opt-out of receiving multicast data from a group. All multicast data is automatically flooded to all downstream nodes, and these nodes must send periodic prune requests upstream in order to stop the flow. This can be a beneficial configuration in the case where most clients are interested in all multicast data. It is also guaranteed to generate the shortest possible routing tree, does not depend on the existence of an RP, and is robust against a dynamically changing network. Despite these benefits, it should be noted that many IPv6 routers do not seem to implement PIM-DM.

Since PIM-DM is not supported in IPv6 and is typically useful only in niche cases, VICTORY-compliant routers are only recommended to support PIM-DM as specified and described in Table 188.

Table 188. Platform Independent Multicast - Dense Mode

Applicability	Specification
Required	IETF RFC 5059 (Standards Track): Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)
Recommended	IETF RFC 3973 (Experimental): Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)

10.4.9.3 Additional Considerations for Selecting PIM Mode

Selecting which of these multicast routing models is best to use for a specific VDB is a decision that will depend on a number of factors. As such, both are allowed as options on the VDB. It should be noted that the two technologies cannot be used at the same time on the same subnet, however, some network equipment allows for the specification of a Sparse-Dense mode. In Sparse-Dense mode, a failed RP will cause a subnet to fall back on dense mode. This feature is not called out in any specific RFC, but is allowed on the VDB.

10.4.10 Multicast Data Addressing

The following sections describe how multicast group addresses will be assigned to VICTORY-developed data streams.

10.4.10.1 Classless Inter-Domain Routing and Subnetting for IPv4 <01054–20120725, Pro>

Classless Inter-Domain Routing (CIDR) is specified by IETF RFC 4632. It describes how individual IP address subnets are assigned and routed on the Internet as a whole. It is closely related to Internet Standard Subnetting, which is defined by IETF RFC 950, and describes how to logically break up a range of IP addresses into easily routable and logically visible subnets.

Both CIDR and subnetting are well defined and actively used as components of IPv4 Ethernet networks. As such, all components of the VDB shall implement CIDR and subnetting as specified by IETF RFCs 4632 and 950.

Table 189 provides the list of specifications required to support CIDR and subnetting for IPv4, including DISR references.

Table 189. Classless Inter-Domain Routing and Subnetting Standards for IPv4

Type	Standard	Designator	DISR Reference
Host/Address Configuration	Classless Inter-Domain Routing	IETF RFC 4632	N/A
Host/Address Configuration	Internet Standard Subnetting Procedure	IETF RFC 950	IETF Standard 5

10.4.10.2 Subnetting for IPv6 <01082–20120131, Exp>

The subnetting model for IPv6 is very different from that of IPv4. It is described primarily in IETF RFC 4291, with some additional detail in IETF RFC 4861 (Neighbor Discovery), and with clarifications made in IETF RFC 5942. All components of the VDB that implement IPv6 shall implement subnetting as specified by IETF RFCs 4291, 4861, and 5942.

Table 190 provides the list of specifications required to support subnetting for IPv6, including DISR references.

Table 190. Subnetting Standards for IPv6

Type	Standard	Designator	DISR Reference
Host/Address Configuration	IP Version 6 Addressing Architecture	IETF RFC 4291	IETF RFC 4291
Address Translation	Neighbor Discovery for IP version 6 (IPv6)	IETF RFC 4861	IETF RFC 4861
Host/Address Configuration	IPv6 Subnet Model: The Relationship between Links and Subnet Prefixes	IETF RFC 5942	N/A

10.4.10.3 IPv4 Address Space <01033–20120725, Pro>

For IPv4, all multicast groups use addresses in the 224.0.0.0/4 address space. This space is further subdivided into sections that are assigned by the Internet Assigned Numbers Authority (IANA). The IANA prescribes the sections that are allowed for use in privately-scoped networks.

Two subnets, in particular, are called out by IETF RFC 2365 for use in certain scopes of a local network:

- Local Scope: 239.255.0.0/16
- Organization Local Scope: 239.192.0.0/14

For VICTORY-developed data streams, the Local Scope multicast group address range shall be used for data streams on a single platform. Such traffic is expected to not leave the platform. The Organization Local Scope multicast group subnet will be used for data that might be subscribed to by a client off of a platform.

Table 191 provides the required specifications describing how IPv4 address groups are subdivided.

Table 191. IPv4 Address Groups

Applicability	Comment	Specification
Required	Follow best practices	IETF RFC 2365 (Best Current Practice): Administratively Scoped IP Multicast
Required	Follow best practices	IETF RFC 5771 (Best Current Practice): IANA Guidelines for IPv4 Multicast Address Assignments

In all other cases, if a multicast group has already been assigned to a protocol or type of data, use that group as specified. An up-to-date database of such assignments can always be found at:
http://www.iana.org/assignments/multicast-addresses/multicast-addresses.xml.

10.4.10.4 IPv6 Address Space <01034–20110729, Exp>

For IPv6, all multicast groups use addresses in the FF00::/8 address space. Like IPv4, this space is further subdivided into sections that are assigned by the IANA. The IANA also prescribes the sections which are available for use in privately-scoped networks. The primary difference between IPv4 and IPv6 is the amount of address space available and the granularity built into the scoping.

IPv6 includes a four-bit portion of the address for use in scoping, and a four-bit portion for use as a flag field (these make up the second octet of an IPv6 multicast address).

Three scoped subnets, in particular, are recommended for use in the VDB for certain specified scopes:

- Link Local: FFx2::/16 (for data streams on a single VDB)
- Site Local: FFx5::/16 (for data streams intended to move between VDBs)
- Organization Local: FFx8::/16 (for data streams intended to move between platforms)

It should be noted that the 'x' in the above addresses represents a "don't care" value. In implementation, this 'x' will be replaced with a value derived from the flag field. The three flags currently defined for this four-bit field are the R, P, and T flags. On the VDB, these three flags are supported as specified in their defining RFCs.

Table 192 provides the required specifications describing how IPv6 address groups are subdivided and how the IPv6 scoping and flag fields function.

Table 192. IPv6 Address Groups

Applicability	Comment	Specification
Required	None	IETF RFC 3306 (Standards Track): Unicast-Prefix-based IPv6 Multicast Addresses
Required	None	IETF RFC 3956 (Standards Track): Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address
Required	None	IETF RFC 4291 (Standards Track): IP Version 6 Addressing Architecture
Required	None	IETF RFC 4489 (Standards Track): A Method for Generating Link-Scoped IPv6 Multicast Addresses

In all other cases, if a multicast group has already been assigned to a protocol or type of data, then use that group as specified. An up-to-date database of such assignments can always be found at: http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xhtml.

10.5 Network Port Assignment <01035–20120725, Pro>

Both the TCP and UDP transport layer protocols specified for use by VICTORY use a value called a port to differentiate one type of network traffic from another. A port value is used in conjunction with an IP address to create a unique address for a service or data stream. The Internet Assigned Numbers Authority (IANA) currently divides the range of ports into three categories:

- Well-known ports: 0 – 1023
- Registered ports: 1024 – 49151
- Ephemeral ports: 49152 – 65535

Most well-known ports require administrative access to an operating system. In addition, the IANA requests, but does not require, that most registered ports be known to them. Therefore, for VICTORY-developed data streams, it is recommended that ephemeral ports be used when possible, though the use of registered ports for VICTORY-developed data streams is allowed as long as there is no conflict. In all other cases, it is recommended that predefined and well-known port ranges be used as specified in the IANA port database available at: http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml. In all cases, care should be taken to avoid port resource conflicts.

10.6 VDB Message Quality of Service

10.6.1 DiffServ <01037–20120725, Pro>

Components in a VDB that support QoS shall support the Differentiated Services (DiffServ) standards as specified in:

- GTP–0009: GIG Technical Profile (GTP) – Quality of Service (QoS) (primary standard)
- IETF RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers
- IETF RFC 2597: Assured Forwarding of PHB (Per Hop Behavior) Group
- IETF RFC 3140: Per Hop Behavior Identification Codes
- IETF RFC 3246: An Expedited Forwarding PHB (Per Hop Behavior)

In addition, the following informational references may be of use for better understanding and implementing DiffServ, as specified by GTP–0009:

- IETF RFC 2475: An Architecture for Differentiated Services
- IETF RFC 2697: A Single Rate Three Color Marker
- IETF RFC 2698: A Two Rate Three Color Marker
- IETF RFC 3247: Supplemental Information for the New Definition of the EF PHB (Expedited Forwarding Per-Hop Behavior)
- IETF RFC 3260: New Terminology and Clarifications for Diffserv
- IETF RFC 4594: Configuration Guidelines for DiffServ Service Classes
- IETF RFC 5127: Aggregation of Diffserv Service Classes

The section titled "QoS Mapping to Components" describes which components of the VDB are required to support QoS and which components may optionally support QoS.

10.6.2 Quality of Service Levels <Informational>

GTP–0009 specifies the use of 42 different DiffServ Code Points (DSCPs). These are allocated based on a mix of three different service parameters: Precedence Level, Service Class, and Conformance Marking.

10.6.2.1 Precedence Level

GTP–0009 specifies the use of five Precedence Levels (PLs) in order to implement Precedence-Based Assured Services (PBAS). A PL is a user indication of the relative importance and urgency of a military message. The goal of PBAS is to ensure that higher precedence messages will, with high probability, receive the QoS associated with their service class at the expense of lower precedence messages if the network is congested.

The PLs specified are (from highest priority to lowest):

- Flash Override (FO)
- Flash (F)
- Immediate (I)
- Priority (P)
- Routine (R)

NOTE: *The use of Precedence Levels here are taken from GTP–0009. They do not correspond to those specified in the Type of Service (ToS) field of IETF RFC 791, which uses similar nomenclature.*

10.6.2.2 Service Class

As defined in IETF RFC 4594:

```
A "service class" represents a set of traffic that requires specific delay, loss, and
jitter characteristics from the network.  Conceptually, a service class pertains to
applications with similar characteristics and performance requirements, such as a
"High-Throughput Data" service class for applications like the web and electronic mail,
or a "Telephony" service class for real-time traffic such as voice and other telephony
services.  Such a service class may be defined locally in a Differentiated Services (DS)
domain, or across multiple DS domains, possibly extending end to end.
```

GTP–0009 specifies twelve Granular Service Classes contained in four Aggregate Service Classes. The Service Classes specified are (from highest priority to lowest):

- Network Control
- Inelastic/Real-Time
  - User Signaling
  - Short Messages
  - Voice
  - Multimedia Conferencing
  - Broadcast Video
- Preferred Elastic
  - Multimedia Streaming
  - Low-Latency Data
  - High-Throughput Data
  - Operations, Administration, and Management (OA&M)
- Elastic
  - Default/Best Effort
  - Low Priority Data

It should be noted that in Table 193, two of the aggregate service classes, Voice and Multimedia Conferencing, are further broken down into assured and non-assured traffic types, with the assured classes given a full range of priority levels.

10.6.2.3 Conformance Marking

GTP–0009 specifies a number of Conformance Markings. These markings indicate a degree of conformance to a governing Service Level Agreement (SLA). This allows for a form of dynamic, or feedback-based, traffic throttling. Examples cited are IETF RFCs 2697 and 2698. In Table 193, conformance markings do not get their own column, but are indicated by many of the N/As in the Precedence Level column.

10.6.3 DiffServ Code Point (DSCP) Assignments <01039–20120725, Pro>

Infrastructure Components of a VDB shall conform to the usage of particular DSCPs as specified in Table 193. This table is derived from GTP–0009.

DSCP values not specified herein that are covered by other RFC-specified standard uses are allowed to be used for platform-specific uses. The specifications in Table 193 and GTP–0009 are consistent with DiffServ IETF RFCs 2474, 2597, and 3246.

Table 193. Mapping of DSCPs

Aggregate Service Class	Granular Service Class	Precedence Level	DSCP	
			Binary	Decimal
Control	Network Control and Signaling	N/A	110 000	48

Aggregate Service Class	Granular Service Class	Precedence Level	DSCP	
			Binary	Decimal
Inelastic / Real-Time	User Signaling	N/A	101 000	40
	Short Message	FO	100 000	32
	Assured Voice	FO	101 001	41
		F	101 011	43
		I	101 101	45
		P	101 111	47
		R	110 001	49
	Non-Assured Voice	N/A	101 110	46
	Assured Multimedia Conferencing	FO	100 001	33
		F	100 011	35
		I	100 101	37
		P	100 111	39
		R	110 011	51
	Non-Assured Multimedia Conferencing	N/A	100 010	34
		N/A	100 100	36
		N/A	100 110	38
	Broadcast Video	N/A	011 000	24

Aggregate Service Class	Granular Service Class	Precedence Level	DSCP	
			Binary	Decimal
Preferred Elastic	Multimedia Streaming	FO	011 001	25
		F	011 011	27
		I	011 101	29
		P	011 111	31
		R	011 010	26
		N/A	011 100	28
		N/A	011 110	30
	Low Latency Data	FO	010 001	17
		F	010 011	19
		I	010 101	21
		P	010 111	23
		R	010 010	18
		N/A	010 100	20
		N/A	010 110	22
	High Throughput Data	FO	001 001	9
		F	001 011	11
		I	001 101	13
		P	001 111	15
		R	001 010	10
		N/A	001 100	12
		N/A	001 110	14
	OA&M	N/A	010 000	16
Elastic	Default (Best Effort)	N/A	000 000	0
	Low-Priority Data	N/A	001 000	8

NOTE: *As a clarification, GTP–0009 recognizes that few Ethernet switch devices are capable of using all 42 of the DSCP service classes described in Table 193. As such, only the four aggregate service classes are required for GTP–0009, and therefore VICTORY, compliance. Additional details for these various service classes can be found in the section titled "Quality of Service Levels."*

10.6.4 QoS Mapping to Components <01036–20120725, Pro>

VICTORY Services on End Nodes receiving data from the VDB are not required to differentiate the data based on the DiffServ markings. VICTORY Services on End Nodes sending data to the VDB shall utilize the DiffServ Code Points (DSCP) as described in GTP–0009, and shown in Table 193, to mark data that requires QoS differentiation. As specified in GTP–0009, general traffic sent on the VDB that does not require QoS differentiation shall use DSCP 0 (default).

Layer 3 Infrastructure components (routers, firewalls, INEs) shall support QoS as specified in GTP–0009.

Layer 2 Infrastructure components (switches) should support QoS as specified in GTP–0009.

NOTE: *Some clarifications regarding the manner in which the Ethernet switch component is expected to support this specification can be found in the section titled "Applicability of Internet Protocol Specifications to the Switch Component."*

10.7 VDB Host Addressing

The VDB is the core structure of the VICTORY architecture that integrates C4ISR/EW systems and implements interfaces to other systems. The VDB is nominally an IP network that facilitates interoperability between electronic systems that interface with it, as well as between the components of C4ISR/EW systems.

The VDB supports the capability for VDB hosts or end nodes to be addressed in a programmatic way. This is necessary so that individual end nodes can communicate with each other across the VDB. The host addressing scheme used by the VDB is defined by a selection of well-known and commonly used IP networking standards. These standards cover three general requirements of host addressing: address translation, host/address configuration, and node identification.

10.7.1 VDB Address Translation

Address translation on the VDB is the process of translating an end node address (nominally an IP address or OSI model network layer address) into a hardware address (OSI model link layer address). This is required so that data routing across the VDB can function properly.

10.7.1.1 VDB IPv4 Address Translation <01069–20120725, Pro>

The protocol selected for address resolution of IPv4 addresses on the VDB is the Address Resolution Protocol (ARP). This is the standard address resolution protocol used for IPv4 Ethernet networking. It is defined by IETF RFC 826.

Table 194 provides the list of specifications required for IPv4 address translation, including DISR references.

Table 194. IPv4 Address Translation Standards

Type	Standard	Designator	DISR Reference
Address Translation	Ethernet Address Resolution Protocol	IETF RFC 826	IETF Standard 37/IETF RFC 826

NOTE: *Some clarifications regarding the manner in which the Ethernet switch component is expected to support this specification can be found in the section titled "Applicability of Internet Protocol Specifications to the Switch Component."*

10.7.1.2 VDB IPv6 Address Translation <01070–20120725, Pro>

The protocol selected for address resolution of IPv6 addresses on the VDB is the Neighbor Discovery Protocol (NDP). This is the standard address resolution protocol used for IPv6 Ethernet networking. It is defined by IETF RFC 4861.

Table 195 provides the list of specifications required for IPv6 address translation, including DISR references.

Table 195. IPv6 Address Translation Standards

Type	Standard	Designator	DISR Reference
Address Translation	Neighbor Discovery for IP version 6 (IPv6)	IETF RFC 4861	IETF RFC 4861

NOTE: *Some clarifications regarding the manner in which the Ethernet switch component is expected to support this specification can be found in the section titled "Applicability of Internet Protocol Specifications to the Switch Component."*

10.7.2 VDB Host/Address Configuration

Host and address configuration protocols on the VDB determine how network layer addresses (IP addresses) are assigned and subnetted for routing.

10.7.2.1 Static Host Addressing <01061–20120725, Pro>

Static host addressing involves the assignment of IP addresses to hosts *a priori*. When static addressing is used, it is necessary for each individual network host to be manually configured with a unique IP address. The particular method required to configure a host's IP address statically is unique to the host. Ensuring that each IP address allocated statically is unique and non-conflicting is the responsibility of the network implementer.

It can be advantageous for a network to be designed with a static allocation of IP addresses. This can particularly be the case when boot-up time is at a premium.

NOTE: *Some clarifications regarding the manner in which the Ethernet switch component is expected to support this specification can be found in the section titled "Applicability of Internet Protocol Specifications to the Switch Component."*

10.7.2.2 IPv4 Dynamic Host Addressing <01071–20120725, Pro>

The Dynamic Host Configuration Protocol (DHCP) for IPv4, as defined in IETF RFC 2131, defines how individual end nodes acquire network configuration information from a centralized server. It is most commonly used to provide an end node with its IP address assignment but may also supply a number of other configuration parameters such as a hostname, a default routing gateway, or even a network location for an operating system kernel to boot. In addition, DHCP can be used to pass some parameters, such as hostname, from a client to a server.

All components of the VDB shall implement the DHCP protocol as specified by IETF RFC 2131.

This does not mean that all VDB implementations must use DHCP to assign addresses or other configuration information. It is entirely reasonable that a specific VDB implementation will use static IP address assignment for all end nodes. Even in such a case, each component of the VDB must support the DHCP protocol.

Table 196 provides the list of specifications for IPv4 dynamic host configuration, including DISR references.

Table 196. Dynamic Host Configuration Standards for IPv4

Type	Standard	Designator	DISR Reference
Host/Address Configuration	Dynamic Host Configuration Protocol (DHCP)	IETF RFC 2131	IETF RFC 2131

NOTE: *Some clarifications regarding the manner in which the Ethernet switch component is expected to support this specification can be found in the section titled "Applicability of Internet Protocol Specifications to the Switch Component."*

10.7.2.3 IPv6 Dynamic Host Addressing <01072–20120131, Exp>

The DHCP protocol has been updated for IPv6 and is described in IETF RFCs 3646 and 3736.

All components of the VDB that implement IPv6 shall implement the DHCP protocol as specified by IETF RFCs 3646 and 3736.

This does not mean that all VDB implementations must use DHCP to assign addresses or other configuration information. It is entirely reasonable that a specific VDB implementation will use static IP address assignment for all end nodes. Even in such a case, each component of the VDB must support the DHCP protocol.

Table 197 provides the list of specifications for IPv6 dynamic host configuration, including DISR references.

Table 197. Dynamic Host Configuration Standards for IPv6

Type	Standard	Designator	DISR Reference
Host/Address Configuration	DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)	IETF RFC 3646	N/A
Host/Address Configuration	Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6	IETF RFC 3736	N/A

NOTE: *Some clarifications regarding the manner in which the Ethernet switch component is expected to support this specification can be found in the section titled "Applicability of Internet Protocol Specifications to the Switch Component."*

10.7.3 VDB Routing and Forwarding

Routing and forwarding procedures and protocols on the VDB determine how network layer traffic (IP packets) are sent to the correct destination.

10.7.3.1 IPv4 Static Routes <01086–20180215, Pro>

The VICTORY Data Bus shall support the use of IPv4 static routing as defined in IETF RFC 1812, Section 7.4: Requirements for IP Version 4 Routers, Static Routing.

10.7.3.2 RIP <01088–20180215, Pro>

The VICTORY Data Bus shall support the use of the Routing Information Protocol (RIP) as defined in IETF RFC 2453: RIP Version 2.

10.7.3.3 OSPF Version 2 <01089–20180215, Pro>

The VICTORY Data Bus shall support the use of Open Shortest Path First (OSPF) as defined in IETF RFC 2328: OSPF Version 2.

10.7.3.4 BGP <01090–20180215, Pro>

The VICTORY Data Bus shall support the use of BGP as defined in IETF RFC 4271: A Border Gateway Protocol 4 (BGP–4).

10.7.3.5 IPv6 Static Routes <01091–20180215, Exp>

The VICTORY Data Bus shall support the use of IPv6 static routing as defined in IETF RFC 6434: IPv6 Node Requirements or IETF RFC 6204: Basic Requirements for IPv6 Customer Edge Routers.

10.7.3.6 RIPng <01093–20180215, Exp>

The VICTORY Data Bus shall support the use of the Routing Information Protocol (RIP) as defined in IETF RFC 2080: RIPng for IPv6.

10.7.3.7 OSPF for IPv6 <01094–20180215, Exp>

The VICTORY Data Bus shall support the use of Open Shortest Path First (OSPF) as defined in IETF RFC 5340: OSPF for IPv6.

10.7.4 VDB Node Identification

Node identification on the VDB is accomplished in a human-readable form primarily through the use of the Domain Name System (DNS). The DNS provides a methodology for requesting from a specific server the human-readable hostname and domain name mapping of a given IP address, and vice-versa.

10.7.4.1 VDB Node Identification for IPv4 <01042–20120725, Pro>

For IPv4, DNS is defined by IETF RFCs 1034 and 1035. All components of the VDB shall support the use of Domain Names and the DNS, as specified by IETF RFCs 1034 and 1035.

As with DHCP, although all VDB components must support DNS, it is not required that all VDB implementations use DNS for the assignment of hostnames. A specific VDB implementation may instead use other methods for the assignment of hostnames, such as shared static ‘hosts’ configuration files, or may forego the use of hostnames entirely.

Table 198 provides the list of specifications required for VDB node identification for IPv4, including DISR references.

Table 198. VDB Node Identification for IPv4

Type	Standard	Designator	DISR Reference
Node Identification	Domain Names - Concepts and Facilities	IETF RFC 1034	IETF Standard 13
Node Identification	Domain Names - Implementation	IETF RFC 1035	IETF Standard 13

10.7.4.2 VDB Node Identification for IPv6 <01083–20120131, Exp>

DNS has been extended by IETF RFC 3596 to support IPv6. Components of the VDB that support IPv6 shall support the DNS extensions for IPv6 described in IETF RFC 3596.

As with DHCP, although all VDB components must support DNS, it is not required that all VDB implementations use DNS for the assignment of hostnames. A specific VDB implementation may instead use other methods for the assignment of hostnames, such as shared static ‘hosts’ configuration files, or may forego the use of hostnames entirely.

Table 199 provides the list of specifications required for VDB node identification for IPv6, including DISR references.

Table 199. VDB Node Identification for IPv6

Type	Standard	Designator	DISR Reference
Node Identification	DNS Extensions to Support IP Version 6	IETF RFC 3596	IETF RFC 3596

10.8 Service and Node Auto-Discovery

The purpose of this section is to specify the technology to be used for identifying what nodes and services are running on a system integrated by a VDB.

10.8.1 Background on Service and Node Auto-Discovery

When configuring a network of computing nodes, it is often convenient to make various types of so-called “plug-and-play” functionality available to the network designer at various levels of the network stack. At the lowest layer, for example, a standard IP network will not work without each host having a unique IP address. This necessitates a certain amount of planning and configuration for each node, or the configuration of a DHCP server. Using network plug-and-play technologies at this level of the networking stack allows for local IP addresses to be assigned to nodes without the need for either static configurations or DHCP servers.

A similar concern may arise with respect to hostnames on a local network. Again, when a network designer wishes to make all nodes on a network accessible by hostname, it is necessary for each host to have either a static list of hostname-to-IP-address mappings (typically a “hosts” file), or for a DNS server to be configured for the network. An application of network “plug-and-play” technologies to this level of the networking stack allows for nodes to make local hostnames known without the necessity of static configurations, or DNS servers.

Finally, one of the primary purposes of a network is to allow for the sharing of services, such as storage and printing, from one host to another. Without any form of network plug-and-play, the sharing of services almost always requires *a priori* knowledge of service location, protocol, and arguments. (Although it is possible to use technologies such as DNS SRV [Service] records to share a pre-generated list of known services with nodes, this still requires *a priori* knowledge.) When plug-and-play technologies are applied to this level of the networking stack, it becomes possible for a node to automatically discover services of interest on a local network.

In general, when such plug-and-play technologies are applied to the discovery of network nodes through IP address and/or hostname, the term “node auto-discovery” is used. Similarly, when such technologies are applied to the discovery of network services, the term “service auto-discovery” is used.

Although most platform networks that are designed as instantiations of a VDB will likely not use node and service auto-discovery when deployed in the field, there are a number of situations in which such technologies may be very useful. In particular, the use of such technologies is often beneficial when a platform network is being designed, or when a platform is being serviced in a depot. In addition, there are circumstances where the use of such technologies can make it trivial to swap out equipment, such as VICTORY-compliant camera sensors, when they are damaged in the field.

When selecting a technology to fill this need, a number of considerations were evaluated. For example, it was understood that most platform networks would be designed with a requirement that all nodes have either a static IP address and hostname list, or an IP address and hostname list acquired from a DHCP and DNS server. Given this expectation, any technology selected for service auto-discovery would still need to function in such circumstances. In many circumstances, it is expected that the platform network designers will deploy their networks with node and service auto-discovery disabled. See the security concerns listed in the section titled “Auto-Discovery Security Concerns” for an elaboration. The technology selected cannot be too invasive into the normal operation of the VDB as to render it inoperable if disabled.

There are a number of competing technologies aiming to fill this void in the consumer electronics field. In addition, there are a few similar technologies aiming at the enterprise server market. Many of these technologies were evaluated against the considerations listed above. In the end, Zero Configuration Networking was selected as the means of providing Node and Service auto-discovery on the VDB.

10.8.2 Zero Configuration Networking <01044–20140630, Pro>

Zero Configuration Networking (Zeroconf) was originally designed as a replacement for the “Apple Talk” protocol for use on modern and future networks. It is primarily an extension to the existing DNS IETF standard, although it uses some lower-level technologies for the node IP address auto-assignment.

Zeroconf primarily consists of three technologies:

- Link-Local Addressing
- Multicast DNS (mDNS)
- DNS Service Discovery (DNS-SD)

10.8.2.1 Link-Local Addressing

Link-local addressing is the technology used by Zeroconf for the automatic assignment of IP addresses. For IPv4, this technology is often called IPv4LL (IPv4 Link-Local) addressing, or Auto-IP. For IPv6, this technology is called Stateless Address Autoconfiguration, and is essentially built into the protocol.

All end node components in a VDB shall support automatic assignment of IP addresses as specified in:

- IETF RFC 3927: Dynamic Configuration of IPv4 Link-Local Addresses
- IETF RFC 4862: IPv6 Stateless Address Autoconfiguration

It should be noted that infrastructure components do not necessarily need to support the automatic assignment of IP addresses, although they certainly are allowed to, and they should not hinder the implementation used by end node components in any way.

Finally, although it is required that all VICTORY-compliant end node components support IP address auto-assignment, it is not required that all VICTORY-compliant implementations must use this technology. As noted previously, it is expected that most VICTORY-compliant implementations will use either static IP address configurations or a DHCP server. It should also be noted that the use of IP address autoconfiguration does not actually preclude the use of static IP addresses, or a DHCP server, as local addressing can be used concurrently (in effect, giving a node more than one IP address). In actuality, this is almost always the case with IPv6 implementations, and is somewhat foundational to many of the features of IPv6.

10.8.2.2 Multicast DNS

Multicast DNS (mDNS) provides a means by which nodes can exchange DNS (hostname) records without the use of a DNS server. As the name implies, it is built upon multicast routing. It is the means used by Zeroconf to provide hostname and service details (through the use of DNS-SD) between nodes.

All end node components in a VDB shall support mDNS as specified in:

- IETF RFC 6762: Multicast DNS, February 2013

It should be noted that infrastructure components do not necessarily need to support mDNS, although they certainly are allowed to, and they should not hinder the implementation used by end node components in any way.

Finally, although it is required that all VICTORY-compliant end node components support mDNS, it is not required that all VICTORY-compliant implementations must use this technology. As noted previously, it is expected that most VICTORY-compliant implementations will use either static hostname configurations, or a DNS server. It should also be noted that the use of mDNS does not actually preclude the use of static hostnames, or a DNS server, as mDNS-provided local hostnames can be used concurrently (in effect, giving a node a hostname on more than one domain).

10.8.2.3 DNS Service Discovery

DNS Service Discovery (DNS-SD) is built primarily on top of mDNS and another technology called DNS SRV records. DNS SRV records are used in a DNS database to provide a generally static list of available network services to DNS clients. They provide the hostname and port of a given service type. For example, a DNS SRV record may specify that a web server is running on port 80 of the host example.com.

In addition to the use of DNS SRV records, another DNS record type, called a TXT record, is used by DNS-SD. TXT records are generally used in a DNS database to provide arbitrary human- or machine-readable text data to a client. DNS-SD specifies the use of this DNS field to provide additional well-formatted details of a service specified by an SRV record, usually as a key-value pair. For example, an LPR printer service may provide the name of the LPR queue as a TXT record (e.g., “rp=myqueue”).

All end node components in a VDB shall support DNS-SD, the DNS SRV record, and the DNS TXT record as specified in:

- IETF RFC 6763: DNS-Based Service Discovery, February 2013
- IETF RFC 2782: A DNS Resource Record (RR) for specifying the location of services (DNS SRV)
- IETF RFC 1035: Domain Names - Implementation and Specification

It should be noted that infrastructure components do not necessarily need to support DNS-SD, although they certainly are allowed to, and they should not hinder the implementation used by end node components in any way.

Finally, although all VICTORY-compliant end node components are required to support DNS-SD, it is not required that all VICTORY-compliant implementations must use this technology. VICTORY-compliant implementations can depend on *a priori* knowledge and a VCL file to specify the location of network services (and in fact, this is required). They can additionally use static DNS SRV records in a DNS database to provide service details. It should also be noted that the use of DNS-SD does not actually preclude the use of any of these additional methods concurrently.

A list of well-known DNS-SD SRV and TXT records is available at: http://www.dns-sd.org/ServiceTypes.html. It is possible to add additional services to this public list, if desired.

10.8.2.4 Zeroconf Announcement Strings

10.8.2.4.1 DNS Records used by Zeroconf

There are three different DNS records used by Zeroconf as part of DNS-SD. The combination of these three DNS records provides the following capabilities:

- Discovery of a service instance by type
- Discovery of a service instance by subtype
- Discovery of allowed arguments for a service instance

The three records used are the DNS SRV, Pointer Record (PTR), and TXT records. They are used as follows:

> ***NOTE:*** *The < and > symbols represent a "field" where a value should be substituted; they are not part of the format themselves. The provided examples should make this clear.*

- SRV: Provides the unique instance name of a component or service.
  - Record name uses the following format: <*instance-name*>.<*service-type*>.<*service-domain*>
    - <*instance-name*> is the unique name of a particular instance of a service.
      - <*instance-name*> has some limitation on length. For additional detail, please see IETF RFC 6763: DNS-Based Service Discovery.
    - <*service-type*> is the announcement string for the type of service being advertised.
      - The <*service-type*> shall contain only two underscore-initiated strings separated by a period, the first of which is the application protocol name, and the second of which must be either *_udp* or *_tcp*.
        - The selection of *_udp* or *_tcp* should be based on the transport protocol used by the service. Services that either use both or use neither, such as the workstation announcement value used only to provide network node topology, should use the *_tcp* value here.
    - <*service-domain*> is the network domain where the service is located.
      - This value may contain both a subdomain and domain, which shall be represented as: *subdomain.domain*
      - By convention, the value *local* is used when no network domain is set. Also by convention, *local* domains are typically only multicast within a single broadcast domain.
  - Record value points to the A or AAAA record (IPv4 or IPv6 Address) of the service, and includes the TCP or UDP port of the service.
  - Example SRV record names (note that the record values, the hostname/IP address and port, are not included):
    - A Workstation called "My_Computer"
      - *My_Computer._workstation._tcp.local*
    - The Zeroconf website at dns-sd.org:
      - *Zeroconf._http._tcp.dns-sd.org*
    - A VICTORY data interface called "VICTORY Data" on a local network (note the escaped space value)
      - *VICTORY\032Data._vic-vdm._udp.local*
- PTR: Serves two purposes.

- Purpose 1: Record name provides a generic name for all services of a specific type using the following format: *<service-type>.<service-domain>*
  - *<service-type>* and *<service-domain>* are defined as with the SRV record.
- Purpose 2: Record name provides a generic name for all services of a specific subtype of a type using the following format: *<service-subtype>._sub.<service-type>.<service-domain>*
  - *<service-type>* and *<service-domain>* are defined as with the SRV record.
  - *<service-subtype>* is an announcement string for a subset of services of type *<service-type>*
    - For example, if *<service-type>* is *_http._tcp* to indicate a web server, then a *<service-subtype>* value of *_router* could indicate a web server provided to manage a router device (**note:** this is a made up example).
  - *_sub* is a keyword that must come between the *<service-subtype>* and *<service-type>* fields to indicate that a subtype is being used.

    NOTE: *Implementation Note: The subtype name does not need to begin with an underscore (according to the specification, it can be any valid string of 8-bit characters, including escaped spaces and special characters), but some software will fail if it does not. Due to this, all VICTORY-specified service subtypes begin with an underscore.*
    The Apple Bonjour library represents subtypes as a comma-separated list after the service-type, and without the _sub keyword (e.g., ._vic-vdm._udp,_pos-dat). This does affect how subtypes are sometimes displayed and queried by tools built against this library, but does not affect the underlying PTR record (it still contains the _sub keyword).
- In both cases the Record value points to the SRV record that provides a particular instance of the service type and service subtype.
- Example PTR record names (note that the record values, the SRV records, are not included):
  - A computer workstation:
    - *_workstation._tcp.local*
    - Could point to the “My_Computer” SRV record previously described.
  - The website at dns-sd.org:
    - *_http._tcp.dns-sd.org*
    - Could point to the “Zeroconf” SRV record previously described.
  - A VICTORY data interface of a position service on a local network:
    - *_vic-vdm._udp.local*
    - *_pos-dat._sub._vic-vdm._udp.local*
    - Both of these could point to the VICTORY data interface called “VICTORY Data” previously described.

- TXT: Provides key-value arguments that can be used to convey additional information to a client. Typically used to provide meta-data that is necessary for communication with the announced service.
  - For additional detail, please see IETF RFC 6763: DNS-Based Service Discovery.

10.8.2.5 Announcement Strings for use by VICTORY

A list of announcement strings required by different components and services of VICTORY can be found in the appendix titled “Zeroconf Announcement Strings Used By VICTORY.”

10.8.2.6 Special Requirements for Services using Zeroconf

A Zeroconf announcement always points to a single network endpoint (address and port number). This limitation can cause problems for management interfaces and for services that can run at a URL which includes a full path beyond the hostname and port. These limitations are addressed as follows:

1. Services that run from a URL with a path:
   a. If a service runs from a full URL path, the TXT field ‘path’ is required to include the path portion of the URL. The path portion of the URL shall never include the hostname and port (this is a requirement of Zeroconf).
   b. In the following examples, the underlined text represents the portion of the path that must be included in the ‘path’ TXT field:

i. hostname.local:1234/some_path/service
ii. hostname.local:8080/some_other_path

2. VICTORY Management Interfaces: Management interfaces are usually comprised of multiple parameters (WSDL-defined SOAP components) running on multiple network endpoints, but a Zeroconf announcement, even with the 'path' variable, can only point to a single network endpoint. This can be overcome as follows:
    a. When a VICTORY Management service is announced, it is required that the network endpoint used in the announcement be that of the Version Information Management parameter group (SOAP endpoint).
    b. The Version Information Management parameter group includes a function call (getInterfaceSoapComponents) that can then be used to acquire the network endpoint specifications of the remaining management interface parameters.

10.8.2.7 Recommendations for Service Instance Names <Informational>

The following recommendations should be considered when implementing auto-discovery for VDB components and services.

1. A service instance name should be either based upon the unique name of the service or component (as is typically recommended in this document through IETF RFC 4122) or be user-friendly and meaningful (as is recommended by IETF RFC 6763). The service instance naming convention is ultimately determined by the implementor.
2. A service instance name should be suffixed by an integer in the case of a name conflict.
    a. The integer suffix should start at two and increase by one for each service name in conflict.
    b. The integer suffix should be formatted in one of two ways:
        i. The Bonjour library format: after a space, and contained within parentheses
            - E.g., Name_Already_Used is already used, Name_Already_Used (2) is used instead.
        ii. The Avahi library format: after a space and a pound (#) sign
            - E.g., Name_Already_Used is already used, so Name_Already_Used #2 is used instead.
    c. Refer to the note below for an explanation about service name conflicts.

NOTE: *Multiple Zeroconf announcement strings can point to the same endpoint (hostname/IP address and port combination), and multiple Zeroconf announcement strings can have the same instance name if they are of different types (e.g., _workstation._tcp and _vic-mgt._tcp could share the instance name "Position Node"),* ***BUT*** *no two announcement strings can have the same instance name if they are of the same type (i.e., two records cannot be identical, if they do not point to exactly the same place). This may be a concern in some cases.*

10.8.3 Auto-Discovery Security Concerns

It is important that one final note be made with respect to security concerns related to node and service auto-discovery. Any time that service and node auto-discovery is used in a network, any device connected to the network will know the addresses of all other network nodes, as well as all available network services. This is not limited to Zeroconf in particular, but to auto-discovery mechanisms in general. If such knowledge is considered privileged, then either service and node auto-discovery should be disabled, or a lower-level security mechanism (such as IPsec or IEEE 802.1X) should be used.

NOTE: *The knowledge of available services does not in any way weaken the services' own security mechanisms. For example, knowing that a Secure Shell (SSH) service is available on a network node does not imply that an unauthorized user will be able to circumvent the security built into SSH. Appropriate passwords and encryption keys would still be required to use the SSH service.*

NOTE: *The VICTORY IAWG described the security concerns as follows:*

Current and anticipated IAWG specifications do not preclude the use of auto-discovery mechanisms, such as Zeroconf, for initialization and discovery of devices and services on the VDB. Auto-discovery services can provide a useful functionality at both configuration-time and run-time. However, this functionality, as with any network service, can introduce certain vulnerabilities. These vulnerabilities and their associated risk levels must be understood, and mitigated as appropriate. Formal accreditation, acceptance of risk, and verification of appropriately chosen mitigation approaches are all beyond the scope and authority of the VICTORY initiative.

10.9 VICTORY Connector Standards

10.9.1 MIL-DTL–38999 Series III

10.9.1.1 Ethernet <01079–20120131, Exp>

A MIL-DTL–38999 type Ethernet connector shall utilize the pinout as described in the MoD Def Stan 23–09 specification. The pinout for this connector is shown in Table 200.

Table 200. MIL-DTL–38999 Ethernet Connector Pinout

Pair	Wire	Pin
4	2	1
	1	2
1	1	3
	2	4
2	2	5
	1	6
3	2	7
	1	8
Not Connected		9
		10
		11
		12
		13

10.9.1.2 Dual Ethernet <01080–20120131, Exp>

A MIL-DTL–38999 type Dual Ethernet connector shall utilize the pinout as described in the MoD Def Stan 23–09 specification. The pinout for this connector is shown in Table 201.

Table 201. MIL-DTL–38999 Dual Ethernet Connector Pinout

Ethernet Channel	Pair	Wire	Pin
A	4	2	1
		1	2
	1	1	3
		2	4
	2	2	5
		1	6
	3	2	7
		1	8
Not Connected			9
			10
			11
			12
			13
			14
B	4	2	15
		1	16
	1	1	17
		2	18
	2	2	19
		1	20
	3	2	21
		1	22

10.9.2 Other Ethernet Connectors <01081–20120131, Exp>

In some cases, size or other constraints necessitate the use of a connector other than a MIL-DTL–38999. In such circumstances, these other connectors shall use a near-identical pinout as described in MIL-DTL–38999 Series III with respect to the location of the pins, not the pin numbers.

NOTE: *In the future, as members of the VICTORY Working Groups verify specific connectors with the pinouts described above, the recommended mappings for those connectors will be added as subsections of this section.*

11. VDB TIME SYNCHRONIZATION

Time synchronization is an important aspect of any network-based system. To devices and applications requiring precise timing, it is imperative to operate correctly. Time synchronization is achieved by synchronizing device clocks with a common time source. In the case of a ground vehicle with a VDB, the method of synchronizing the clocks of devices attached to the VDB in the vehicles is by passing network packets that contain time information across the VDB network. This is a network-based time synchronization method.

11.1 VDB Time Synchronization Operational Concept <Informational>

Figure 4 depicts a typical time synchronization operation for two different platforms, one using Network Time Protocol (NTP) and the other using Precision Time Protocol (PTP).

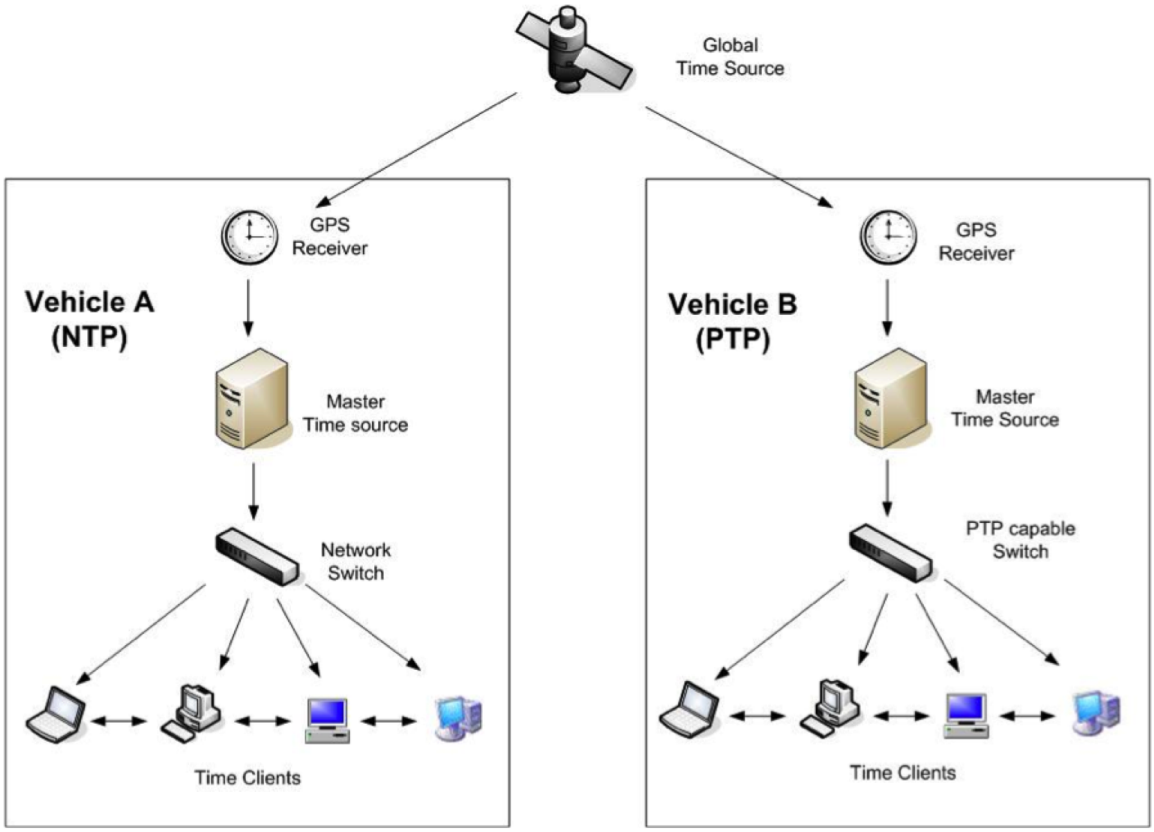

Figure 4. Network Time Synchronization Concept

11.2 VDB Time Synchronization Specifications

The VDB shall provide services for synchronizing the clocks of attached devices to a common time source. The service shall utilize the VDB network. The two most prominent technologies for network-based time synchronization are Network Time Protocol (NTP) and Precision Time Protocol (PTP). While PTP has the potential to provide more precise timing with hardware support, NTP is more ubiquitous and many devices support NTP without change. Many systems can utilize NTP for their time synchronization needs, while implementations with applications requiring high accuracy time synchronization between distributed nodes can utilize PTP. Even without specialized hardware support, PTP is generally at least as accurate as NTP.

The level of accuracy required for time synchronization in VDB systems is entirely dependent upon the application and mission needs of the vehicle platform. If a specific level of time keeping precision is required, it is insufficient to simply require PTP and expect for that level to be achieved. In such a case, the onus is on the vehicle implementer to architect a PTP solution that meets the requirements. This can be achieved with a combination of PTP-aware client and infrastructure hardware.

With respect to client (end node) hardware, greater accuracy can be achieved through the use of network interface cards (Ethernet interfaces) that provide hardware time-stamping of PTP packets upon ingress and egress. In addition, PTP-aware infrastructure components, such as Ethernet switches and IP routers, can greatly reduce jitter from the PTP traffic on the network through the use of end-to-end transparency features, as described in the section titled "PTP End-to-End Transparent Clock Mode for Infrastructure Components." Combining these methods with an accurate PTP grandmaster clock can provide a precision in the 100s, and possibly 10s, of nanoseconds.

11.2.1 Network Time Protocol <02002–20121113, Pro>

VDB implementations shall provide an NTPv4 service, which will provide time synchronization for systems for which the accuracy of NTP is sufficient. In order to synchronize with VICTORY-provided common time, devices attaching to the VDB shall have an NTP client. The following RFCs define the NTPv4 standard specification:

- IETF RFC 5905: Network Time Protocol Version 4: Protocol and Algorithms Specification
- IETF RFC 5906: Network Time Protocol Version 4: Autokey Specification
- IETF RFC 5907: Definitions of Managed Objects for Network Time Protocol Version 4 (NTPv4)
- IETF RFC 5908: Network Time Protocol (NTP) Server Option for DHCPv6

11.2.2 Health Monitoring of Network Time Protocol <02004–20121113, Exp>

IETF RFC 5907: Definitions of Managed Objects for Network Time Protocol Version 4 (NTPv4) provides a Management Information Base (MIB) for retrieving status and fault information on NTP. The name of this MIB is "ntpSnmpMIB." VICTORY implementations shall provide health monitoring of the required NTP service through the implementation of this MIB, as described by IETF RFC 5907.

11.2.3 Network Time Protocol Management Specification <02005–20121113, Exp>

The control and monitoring of NTP may be implemented using a network-management protocol such as SNMP and extensions to the MIB database. Alternatively, NTP control messages, as described in Sections 9.1 and 11.1 of IETF RFC 5905, may be used. For VICTORY implementations, at least one of these two methods shall be implemented for managing NTP services.

11.2.4 Precision Time Protocol <02003–20180215, Pro>

The specification for the use of precision time protocol version 2 (PTPv2) is described in Table 202. Table 202 calls out both the overall PTPv2 specification, and the specific subset of the PTPv2 specification that describes the on-the-wire transport that shall be used for VICTORY implementations of PTP. The transport called out for use on the VDB is UDP over multicast IPv4.

PTP can provide time synchronization accuracy on the order of 1 ms without special hardware support, which is similar to NTP. With hardware support, PTP can provide time synchronization accuracy on the order of tens of nanoseconds. Additional details about the kind of hardware support that can be used to increase PTP accuracy are described in the section titled "PTP End-to-End Transparent Clock Mode for Infrastructure Components."

In addition to NTP, VDB implementations shall provide a PTP v2 service as an additional method for time synchronization. In order to synchronize with VICTORY-provided common time, devices attaching to the VDB may have a PTP v2 client.

Table 202. Precision Time Synchronization Protocol

Standard	Title	Description
IEEE 1588–2008	IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems	Precision Time Protocol version 2 (PTPv2) overall specification

Standard	Title	Description
IEEE 1588–2008, Annex D	Transport of PTP over User Datagram Protocol over Internet Protocol Version 4	PTP Transport to be used on VICTORY

11.2.5 PTP End-to-End Transparent Clock Mode for Infrastructure Components <02008–20180215, Exp>

Many network infrastructure components, such as Ethernet switches and Internet Protocol Routers, provide a special capability to be "PTP aware." Such infrastructure components are able to monitor and modify PTP traffic as it moves through the network in order provide exceptional time precision. There are a few different methods outlined in the PTPv2 specification that can be utilized by infrastructure components in order to provide this additional precision.

One such method is called the End-to-End (E2E) Transparent Clock mode. When this mode is enabled on a PTP network, the infrastructure component continually measures the input-to-output queueing delay of PTP packets, modifying the packets on egress in order to indicate the measured delay. This, in essence, makes the infrastructure component "transparent" to the PTP time source client, removes a large source of PTP packet jitter, and allows for exceptional accuracy when combined with an accurate grandmaster source and client network interface hardware that prioritizes PTP packets and provides hardware-level timestamps at packet ingress and egress.

Table 203 lists the primary references from the IEEE 1588–2008 standards document (as listed in Table 202) for implementing PTP E2E transparent clock mode for infrastructure devices. Although these document sections are the ones most relevant to the implementation of E2E transparent mode, the whole of the IEEE 1588–2008 document is necessary for a complete implementation. As such, VICTORY compliant infrastructure components that implement PTP E2E transparent mode shall follow the guidelines listed in Table 203, as well as any other relevant and referenced sections of the PTP standard.

Table 203. IEEE 1588–2008: Sections Relevant to End-to-end Transparent Clock Implementations

Section Reference	Section Title
6.5.4	End-to-end transparent clocks
8.3	Data sets for transparent clocks
10	PTP for transparent clocks
11.5	Transparent clock residence time correction for PTP version 2 events

11.2.6 Health Monitoring of Precision Time Protocol <02006–20121113, Exp>

IEEE 1588–2008 provides a series of management messages. One of these messages, FAULT_LOG, shall be used for the health monitoring of PTP.

11.2.7 Precision Time Protocol Management Specification <02007–20121113, Exp>

PTP management messages, as described in IEEE 1588–2008, shall be implemented for the management and control of PTP.

NOTE: *Section 15.1.1 of IEEE 1588–2008 describes three PTP management mechanisms. VICTORY is requiring the first management mechanism, which is the mechanism specified in Section 15.2 of the document.*

12. NETWORK STORAGE ON THE VDB

Network storage on the VICTORY Data Bus is facilitated by two different technologies: Network File System (NFS) and Web Distributed Authoring and Versioning (WebDAV). A VICTORY-compliant system that is intended to provide storage for VICTORY services shall provide storage through WebDAV, and should provide storage through NFS. A VICTORY-compliant client system or service intended to access VICTORY storage may support either or both technologies.

All implementations of network storage shall provide the following features:

- Listing file objects
- Hierarchical representation of file objects
- Create, modify, read-only file operations
- Create, modify, read-only file meta-data operations
- Copy/move file operations
- File metadata has concept of ownership and permissions

The following sections describe the specific requirements for support of both WebDAV and NFS. The implementation of these protocols, as specified, will ensure that the features listed above are available to clients.

12.1 WebDAV <20028–20120731, Pro>

The Web Distributed Authoring and Versioning (WebDAV) protocol is an extension to the HTTP/1.1 protocol that allows clients to perform authoring operations on content remotely. WebDAV provides network-storage-related operations that fit into four main categories:

- Properties: create, remove, and query information about the content such as author and date
- Collections: create sets of documents and retrieve memberships (like a directory listing)
- Locking: the ability to prevent more than one person from editing the same file at a time
  - *This locking mechanism is limited to WebDAV shares only.*
- Namespace Operations: the ability to perform operations that change the mapping from URL to resources, such as copying and moving the file

WebDAV is particularly well suited to document access (including versioning). WebDAV shall be supported as specified by the RFCs listed below. Additional details can be found on the WebDAV Specifications page.

- IETF RFC 4918: HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV), June 2007
- IETF RFC 3253: Versioning Extensions to WebDAV (Web Distributed Authoring and Versioning), March 2002
- IETF RFC 5323: Web Distributed Authoring and Versioning (WebDAV) SEARCH, November 2008
- IETF RFC 4437: Web Distributed Authoring and Versioning (WebDAV) Redirect Reference Resources, March 2006
- IETF RFC 3648: Web Distributed Authoring and Versioning (WebDAV) Ordered Collections Protocol, December 2003
- IETF RFC 3744: Web Distributed Authoring and Versioning (WebDAV) Access Control Protocol, May 2004
- IETF RFC 4122: A Universally Unique IDentifier (UUID) URN Namespace, July 2005

Table 204 provides a mapping of each network storage feature to its corresponding RFC and VICTORY applicability.

Table 204. Network Storage Features and Corresponding WebDAV Specifications

Storage Feature	WebDAV Specification	Applicability
Listing file objects	IETF RFC 4918	Required
Hierarchical representation of file objects	IETF RFC 4918	Required
Create, modify, read-only file operations	IETF RFCs 4918, 3744	Required
Create, modify, read-only file meta-data operations	IETF RFCs 4918, 3744	Required

Storage Feature	WebDAV Specification	Applicability
Copy/move file operations	IETF RFC 4918	Required
File metadata has concept of ownership and permissions	IETF RFC 3744	Required
File locking	IETF RFC 4918	Required
Version Control	IETF RFC 3253	Optional
Searching	IETF RFC 5323	Optional
Redirection	IETF RFC 4437	Optional
Collection Ordering	IETF RFC 3648	Optional
Namespacing	IETF RFC 4122	Optional

NOTE: *When WebDAV is used in conjunction with NFS, special consideration should be given to the handling of file locks, since locks are usually enforceable only by the network file system that creates them.*

12.2 NFS <20029–20120731, Pro>

The Network File System (NFS) protocol was originally developed by Sun Microsystems in 1984, but it is now an open standard developed and maintained by the IETF. It provides the following features:

- Fine-grained access control
  - *Locking mechanisms are limited to NFS shares only.*
- Multiple authentication methods
- Strong security
- Transport can be UDP or TCP

NFS is particularly well suited for generic file access. NFS version 4 shall be supported as specified by the RFCs listed below:

- IETF RFC 3530: Network File System (NFS) version 4 Protocol, April 2003
- IETF RFC 5661: Network File System (NFS) Version 4 Minor Version 1 Protocol, January 2010

Table 205 provides a mapping of each network storage feature to its corresponding RFC.

Table 205. Network Storage Features and Corresponding NFS Specifications

Storage Feature	NFS Specification
Listing file objects	IETF RFCs 3530, 5661
Hierarchical representation of file objects	IETF RFCs 3530, 5661
Create, modify, read-only file operations	IETF RFCs 3530, 5661
Create, modify, read-only file meta-data operations	IETF RFCs 3530, 5661
Copy/move file operations	IETF RFCs 3530, 5661
File metadata has concept of ownership and permissions	IETF RFCs 3530, 5661
File locking	IETF RFCs 3530, 5661

NOTE: *When NFS is used in conjunction with WebDAV, special consideration should be given to the handling of file locks, since locks are usually enforceable only by the network file system that creates them.*

13. ACCESS CONTROL FRAMEWORK

The VDB access control framework provides a set of specifications for the communication of authentication and authorization information on the platform.

- ***Authentication*** proves the Principal is who the Principal claims to be.
- ***Authorization*** checks if the Principal can Act on the Resource requested.

Principals can be human (a user) or machine (a system, service, or software process). For Web Services, actions are operations on the service as defined in its WSDL, and are therefore service-specific.

Access control falls under the cybersecurity umbrella. Cybersecurity focuses on the protection of information and ensuring its confidentiality, integrity, and availability. Some of its key principles are listed below. Refer to the section titled "Cybersecurity Policy" for more information.

- Applies to all information technology that receives, processes, stores, displays, or transmits DoD information.
- Must be fully integrated into system life cycles.
- Information protection requirements are satisfied by the selection and implementation of appropriate security controls.

13.1 Authentication Methods <40001–20180215, Pro>

NOTE: *Management interfaces that provide their own security mechanisms (e.g., SNMPv3) shall be exempted from the requirements of this section.*

Principals shall be authenticated by one or more of the following methods:

- Password Method – Principal is authenticated through the use of a username and password.
- SAML Method – Principal is authenticated through the use of SAML assertions.
- PKI Method – Principal uses an X.509 certificate to authenticate. This certificate could be a software certificate or stored on a hardware token like a Smart Card.
- Biometrics Method – Principal is authenticated through the use of biometric measures.

NOTE: *For more information on implementing Security Tokens, refer to Section 6: Security Tokens in "Web Services Security: SOAP Message Security Version 1.1.1," 18 May 2012, OASIS Standard.*

13.1.1 Password-Based Authentication for VICTORY Web Services

Implementations that choose to utilize Password-Based Authentication shall support the following standard:

- Web Services Security Username Token Profile Version 1.1.1, 18 May 2012, OASIS Standard.

NOTE: *Transport Layer Security (TLS) is required if using UsernameToken. Not using TLS would allow passwords to be sent in plaintext across the network.*

13.1.2 SAML-Based Authentication for VICTORY Web Services

Implementations that choose to utilize SAML-Based Authentication shall support the following standard:

- Web Services Security SAML Token Profile Version 1.1.1, 18 May 2012, OASIS Standard.

13.1.3 PKI-Based Authentication for VICTORY Web Services

Implementations that choose to utilize PKI-Based Authentication shall support the following standard:

- Web Services Security X.509 Certificate Token Profile Version 1.1.1, 18 May 2012, OASIS Standard.

13.1.4 Biometrics-Based Authentication for VICTORY Web Services [Experimental]

Implementations that choose to utilize Biometrics-Based Authentication shall support the following standards:

- OASIS Biometric Identity Assurance Services (BIAS) SOAP Profile Version 1.0
- ANSI INCITS 442–2010 - Biometric Identity Assurance Services

Biometrics-Based Authentication is the use of measurable physical characteristics or personal behavioral traits (biometrics measures) to identify, or verify the claimed identity of, an individual. Examples of biometrics measures include facial images, fingerprints, and handwriting samples.

The OASIS Biometric Identity Assurance Services (BIAS) profile specifies how to use the XML definitions contained within the ANSI INCITS 442–2010 - Biometric Identity Assurance Services standard to invoke SOAP-based services that implement BIAS operations.

These SOAP-based services enable a software application to invoke biometric identity assurance operations remotely in a Service-Oriented Architecture (SOA) infrastructure. These services are technology agnostic and can support all methodologies (e.g., fingerprint, facial, iris, etc.).

The SOAP Profile implements the “abstract” services specified in ANSI InterNational Committee for Information Technology Standards (INCITS) 442–2010. The profile includes the WSDL / XML schema, data model / data dictionary, message structure / rules, error handling, and general use cases.

The OASIS BIAS profile complements the efforts of INCITS to provide the biometrics and security industries with a documented, open framework for deploying and invoking identity assurance capabilities that can be readily accessed as services. The OASIS and INCITS standards describe methods and bindings by which the INCITS BIAS framework can be used within XML-based Web services.

Biometrics-Based Authentication is currently at the experimental level. It has not gone through initial validation.

13.2 Policy-Based Authorization <Informational>

Policy-Based Authorization is the means for ensuring that only properly authorized Principals are able to access Resources based upon a well-defined security policy.

All authorization inquiries have the same general form:

```
Can {Principal X} perform {Action Y} on {Resource Z}?
```

Policies hold the rules that allow or block a Principal from accessing a Resource. Policy information is expressed using the XACML format. Refer to the Policy Store Service for more information.

13.3 Attribute-Based Access Control <Informational>

According to the Committee on National Security Systems (CNSS) Instruction No. 4009 National Information Assurance (IA) Glossary, Access Control is the process of granting or denying specific requests for obtaining and using information and related information processing services. Historically, Access Control has taken many forms:

- **Access Control List (ACL)**
  - A list of permissions associated with an object. The list specifies who or what is allowed to access the object and what operations are allowed to be performed on the object.
  - A mechanism that implements access control for a system resource by enumerating the system entities that are permitted to access the resource and stating, either implicitly or explicitly, the access modes granted to each entity.
- **Attribute-Based Access Control (ABAC)** – Access control based on attributes associated with and about subjects, objects, targets, initiators, resources, or the environment. An access control rule set defines the combination of attributes under which an access may take place.

- **Discretionary Access Control (DAC)** – A means of restricting access to objects based on the identity and need-to-know of subjects and/or groups to which the object belongs. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission on to any other subject (unless restrained by mandatory access control).
- **Identity-Based Access Control (IBAC)** – Access control based on the identity of the user (typically relayed as a characteristic of the process acting on behalf of that user) where access authorizations to specific objects are assigned based on user identity.
- **Mandatory Access Control (MAC)** – A means of restricting access to objects based on the sensitivity (as represented by a security label) of the information contained in the objects and the formal authorization (i.e., clearance, formal access approvals, and need-to-know) of subjects to access information of such sensitivity.
- **Policy-Based Access Control (PBAC)** – A form of access control that uses an authorization policy that is flexible in the types of evaluated parameters (e.g., identity, role, clearance, operational need, risk, heuristics).
- **Role-Based Access Control (RBAC)** – Access control based on user roles (i.e., a collection of access authorizations a user receives based on an explicit or implicit assumption of a given role). Role permissions may be inherited through a role hierarchy and typically reflect the permissions needed to perform defined functions within an organization. A given role may apply to a single individual or to several individuals.

Because of its flexible and extensible nature, the ABAC model has been chosen for VICTORY. It can easily implement the other types of access control. For example, RBAC could be considered a subset of ABAC where everyone has just one attribute, their role. Similarly, a simple way to implement ACLs in ABAC would be to give a particular attribute to each person requiring access.

Figure 5 below shows a diagram of the authorization architecture adopted by VICTORY. This VICTORY Authorization Architecture does not dictate or mandate a particular implementation of the architecture, but, rather, it requires the implementer to adhere to detailed interface specifications for the four major security component services.

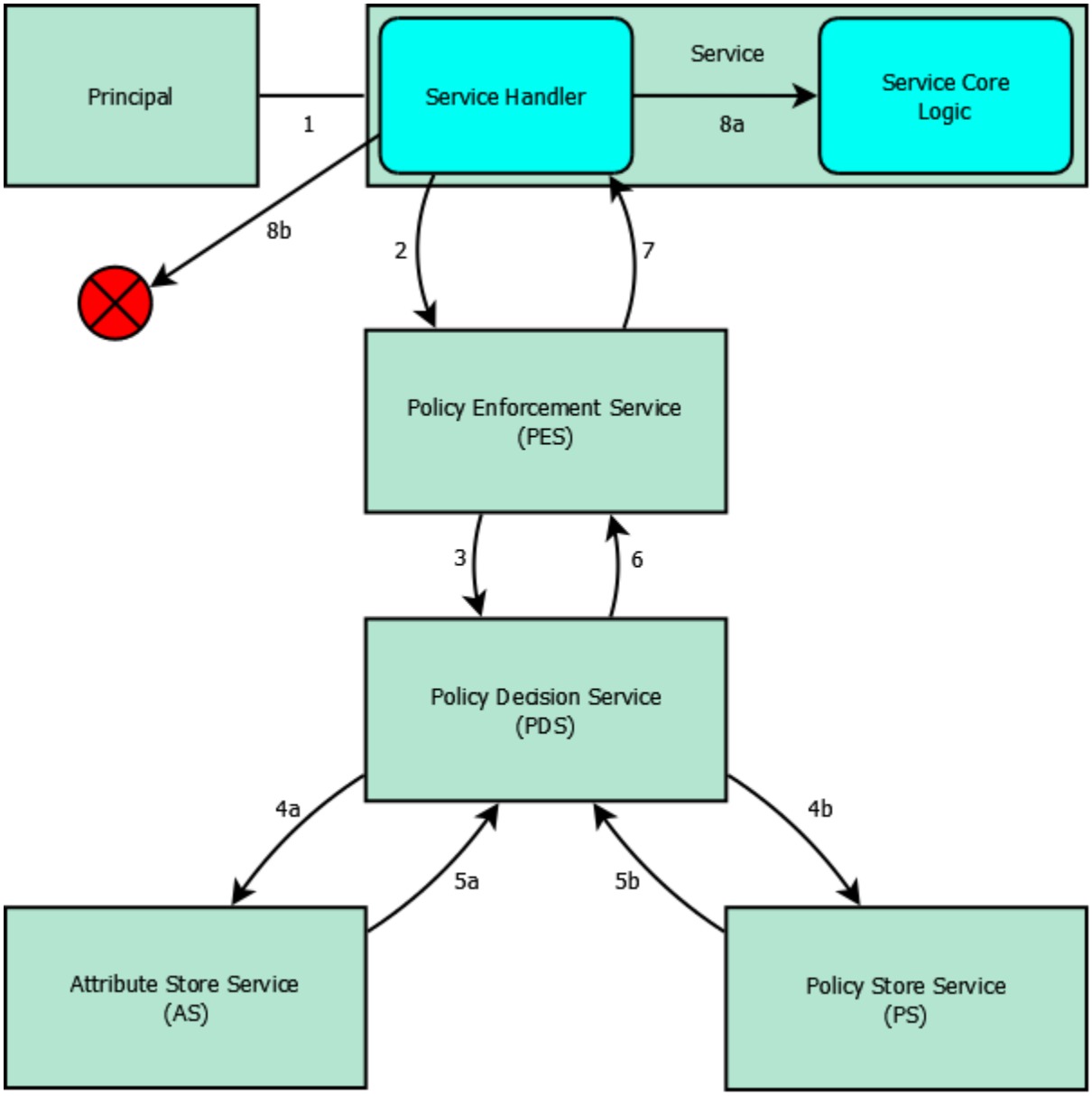

Figure 5. Authorization Architecture

1. The Principal sends a request along with the Authentication Information to the Service. The service must validate this Authentication information.
2. The Service Handler intercepts the message and requests Authorization information from the Policy Enforcement Service (PES).
3. The PES forwards this request on to the Policy Decision Service (PDS).
4. The PDS requests supporting information from the Attribute Store Service (AS) and Policy Store Service (PS).
5. The AS and PS return relevant attributes and policies to the PDS.
6. The PDS computes a decision and returns the result to the PES.
7. The PES forwards that information to the Service Handler.
8. The Principal's request is either:
    a. Allowed through to the Service, or
    b. Denied, with a "Not Authorized" SOAP fault generated.

13.3.1 Attribute Store Service

Refer to the Attribute Store Service component type.

13.3.2 Policy Decision Service

Refer to the Policy Decision Service component type.

13.3.3 Policy Enforcement Service

Refer to the Policy Enforcement Service component type.

13.3.4 Policy Store Service

Refer to the Policy Store Service component type.

13.4 Authorization Model <40003–20180215, Pro>

1. Attribute-Based Access Control (ABAC) shall be the fundamental model for authorization.
2. Each component utilizing the Authorization Model shall:
    a. Extract a Principal's credentials using Web Services Security Tokens as defined in the section titled "Authentication Methods."
    b. Construct a request and submit it to the Policy Enforcement Service located at the endpoint configured via Authorization Architecture Entrypoints.
    c. Evaluate the determination received by the Policy Enforcement Service.
        i. If permitted, continue execution.
        ii. If denied, a SoapFault shall be returned with a "Sender" Fault Code and a "Not Authorized" Fault Reason.

13.5 Authorization Architecture Entrypoints <40012–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the authorization architecture entrypoints. The authorization architecture entrypoints service shall implement functions that support the parameters as described in the following section.

The Authorization Architecture (AA) may have multiple points of entry. For instance, there could be multiple Policy Enforcement Services on the network.

A VICTORY Service that wishes to utilize the Authorization Architecture must provide a management interface for storing the location of each entry point into the Authorization Model that is used by the Service. This enables the dynamic setting of these values as well as allowing third party services to access these values.

Principal authentication via at least one of the Authentication Methods is required prior to being able to Set an architecture entry point. Web Services Security tokens will be used to extract the Principal's credentials as defined in the section titled "Authentication Methods."

Entry points that have been defined for this specification are listed below. The general nature of this interface will allow additional entry points to be added in future specifications should the need arise.

1. ***Policy Enforcement Service***
    a. Location of the Policy Enforcement Service to be used to authorize access requests.
    b. The PES location shall be in the format of a valid URI.
    c. The bounds of the PES location are described in the section titled "URI Bounds" under "VICTORY Data Types."

d. Operations include:
    i. GetPolicyEnforcementService()
        1. Shall return one of the following:
            a. The URI of the Policy Enforcement Service, if one has been set.
            b. The GENERIC_FAULT error, if the Policy Enforcement Service URI has not been set.
    ii. SetPolicyEnforcementService()
        1. The following errors are defined for SetPolicyEnforcementService:
            a. NO_ERROR, if the URI of the Policy Enforcement Service was successfully set.
            b. The OUT_OF_RANGE error may be returned if the Policy Enforcement Service URI was outside the range defined by this specification document.
            c. The GENERIC_FAULT error shall be returned if the operation was not successful for some other reason, such as a malformed URI.

13.6 Authentication Model <40013–20180215, Pro>

Each component utilizing the Authentication Model shall:

1. Extract a Principal's credentials using Web Services Security Tokens as defined in the section titled "Authentication Methods."
2. Construct a request and submit it to the Authentication Service located at the endpoint configured via Authentication Entrypoints.
3. Evaluate the determination received by the Authentication Service.
    a. If permitted, continue execution.
    b. If denied, a SoapFault shall be returned with a "Sender" Fault Code and a "Not Authorized" Fault Reason.

13.7 Authentication Entrypoints <40014–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the authentication entrypoints. The authentication entrypoints service shall implement functions that support the parameters as described in the following section.

The Authentication Service may have multiple points of entry. For instance, there could be multiple Authentication Services on the network.

A VICTORY Service that wishes to utilize the Authentication Service must provide a management interface for storing the location of each entry point into the Authentication Service that is used by the Service. This enables the dynamic setting of these values as well as allowing third party services to access these values.

Principal authentication via at least one of the Authentication Methods is required prior to being able to Set an Authentication entry point. Web Services Security tokens will be used to extract the Principal's credentials as defined in the section titled "Authentication Methods."

Entry points that have been defined for this specification are listed below. The general nature of this interface will allow additional entry points to be added in future specifications should the need arise.

1. ***Authentication Service***
    a. Location of the Authentication Service to be used to authenticate access requests.
    b. The AS location shall be in the format of a valid URI.
    c. The bounds of the AS location are described in the section titled "URI Bounds" under "VICTORY Data Types."
    d. Operations include:
        i. GetAuthenticationService()
            1. Shall return one of the following:
                a. The URI of the Authentication Service, if one has been set.

b. The GENERIC_FAULT error, if the Authentication Service URI has not been set.

ii. SetAuthenticationService()

1. The following errors are defined for SetAuthenticationService:
   a. NO_ERROR, if the URI of the Authentication Service was successfully set.
   b. The OUT_OF_RANGE error may be returned if the Authentication Service URI was outside the range defined by this specification document.
   c. The GENERIC_FAULT error shall be returned if the operation was not successful for some other reason, such as a malformed URI.

13.8 Security Features <Informational>

The security features met by this access control framework include the following:

Authentication: Most (if not all) service providers will require that consumers be authenticated before accepting a service request. Service consumers will also need to authenticate service providers when a response is received. Different authentication mechanisms should be supported, and these mechanisms should be configurable and interchangeable according to service-specific requirements. VICTORY currently identifies four Authentication Methods that can be used in support of this architecture. The one that is most recommended is the use of certificates. They can be either software or hardware certificates.

Authorization: In addition to authentication of a service consumer, access to a service will also require the consumer to possess certain privileges. These privileges feed an authorization check that is usually based on access control policies, for example: who can access a service and under what conditions. The authorization implementation should also be extensible to allow for domain- or community of interest (COI)-specific customizations.

Confidentiality: Protects the messages or documents that are carried over the underlying communication transport so that they cannot be made available to unauthorized parties. Sometimes only a fragment of the message or document (e.g., wrapped within a certain XML tag) may need to be kept confidential.

Message Integrity: Provides protection against unauthorized alteration of messages during transit.

Non-repudiation: Provides protection against false denial of involvement in a communication. Non-repudiation ensures that a sender cannot deny a message already sent, and a receiver cannot deny a message already received.

Authentication, authorization, confidentiality, and message integrity will be met with the various Web Services Security (WSS) specifications.

13.9 Service Data Structures <Informational>

The service data entities used within the security architecture are described below.

Secured Message: A SOAP message that is secured in accordance with WSS and utilizes XML Signature for message signatures. SOAP messages are used for all Web Service communications. Accordingly, a SOAP message is used whenever assertion requests, assertions, or authorization policies are transmitted.

Authentication Assertion: A Security Assertion Markup Language (SAML) assertion that contains security context information related to the act of authentication. The Authentication Assertion contains a SAML AuthnStatement that indicates things such as the issuer of the assertion (presumably the entity that performed the authentication), the date/time of authentication, the subject that was authenticated, and the method of authentication.

Attribute Request: A SAML Assertion Request seeking attributes for an entity from an attribute authority. The Attribute Request contains a SAML AttributeQuery that indicates things such as the object of the request. The Attribute Request is a request for an Attribute Assertion.

Attribute Assertion: A SAML assertion that contains security context information related to the attributes for an entity. The Attribute Assertion contains a SAML AttributeStatement that indicates things such as the issuer of the assertion (the attribute authority), the date/time of the assertion, the subject of the assertion (corresponding to the subject from the request), and the attributes associated with the subject. The Attribute Assertion is the response to an Attribute Assertion Request.

Authorization Decision Request: The Authorization Decision Request contains a SAML AuthzDecisionQuery that indicates things such as the action to be performed, the resource, and any evidence pertinent to the decision. The Authorization Decision Request is a request for an Authorization Decision Assertion.

Authorization Decision Assertion: A SAML assertion that contains security context information relating to the result of an Authorization Policy evaluation. The Authorization Decision Assertion contains a SAML AuthzDecisionStatement that indicates things such as the issuer of the assertion (presumably the entity that performed the Authorization Policy evaluation), the date/time of the decision, the action and resource (corresponding to the request), and the decision result (e.g., PERMIT, DENY). The Authorization Decision Assertion is the response to an Authorization Decision Request.

Authorization Policy: An eXtensible Access Control Markup Language (XACML) policy that contains the rules governing the actions that can be performed on one or more resources. An Authorization Policy is evaluated when an Authorization Decision Request is received. This evaluation provides the result for an Authorization Decision Assertion.

To reiterate, the service interfaces defined by this architecture are specifications, not implementations. The actual implementations may utilize best-of-breed commercial off-the-shelf (COTS) and government off-the-shelf (GOTS) technologies and may vary in different IT environments, but the specifications will remain stable and interoperable.

13.10 Information Flow <Informational>

Figure 6 below is a representation of the information flows that would occur during an authorization decision using the four services (AS, PS, PES, PDS) in the ABAC model.

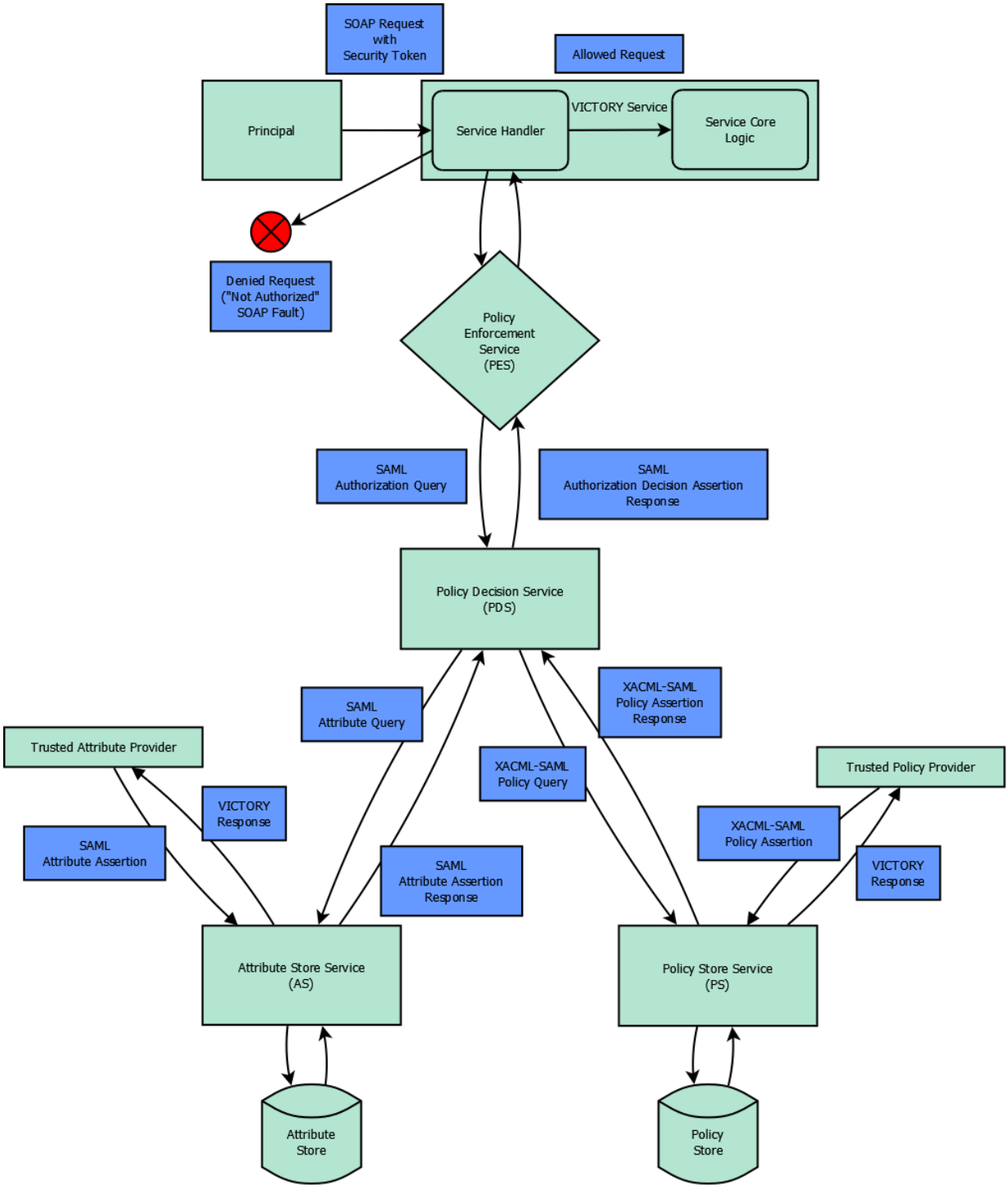

Figure 6. Message Flows

14. MANAGEMENT INTERFACES

Management interfaces are used to configure, control, report the status of, and manage faults on the VDB, its components, integrated systems and their components, and interfaced systems from the network. The concepts of configuration and control are sometimes coupled, referred to as configuration and control. Likewise, the concepts of status reporting and fault management are sometimes coupled, referred to as health reporting.

14.1 Core Management Technologies

14.1.1 SOAP

A core technology selected for the management of VICTORY-defined components and services is SOAP. SOAP can be used to implement the Get and Set operations that are the basis for management functionality. SOAP shall be used by all management interfaces on the VDB except where expressly noted in this document.

14.1.1.1 SOAP Compliance and Standards <10001–20120725, Pro>

All management interfaces based on SOAP shall conform to the following standards:

1. SOAP v1.2 parts 1 and 2.
2. SOAP Message Transmission Optimization Mechanism.

14.1.1.2 Web Services Description Language <10002–20120725, Pro>

SOAP interfaces are commonly defined by a Web Services Description Language (WSDL) file. A WSDL file is an XML-based Interface Definition Language (IDL) file that describes a SOAP interface in thorough enough detail that interfacing application code can often be auto-generated.

14.1.1.3 Background on WSDL Files

As defined by the WSDL 1.1 specification, a WSDL file contains the following components:

- Types: a container for data type definitions using some type system (such as the XML Schema Definition Language [XSD]).
- Message: an abstract, typed definition of the data being communicated.
- Operation: an abstract description of an action supported by the service.
- Port Type: an abstract set of operations supported by one or more endpoints.
- Binding: a concrete protocol and data format specification for a particular port type.
- Port: a single endpoint defined as a combination of a binding and a network address.
- Service: a collection of related endpoints.

The interfacing between these components is displayed in Figure 7 (image in the public domain).

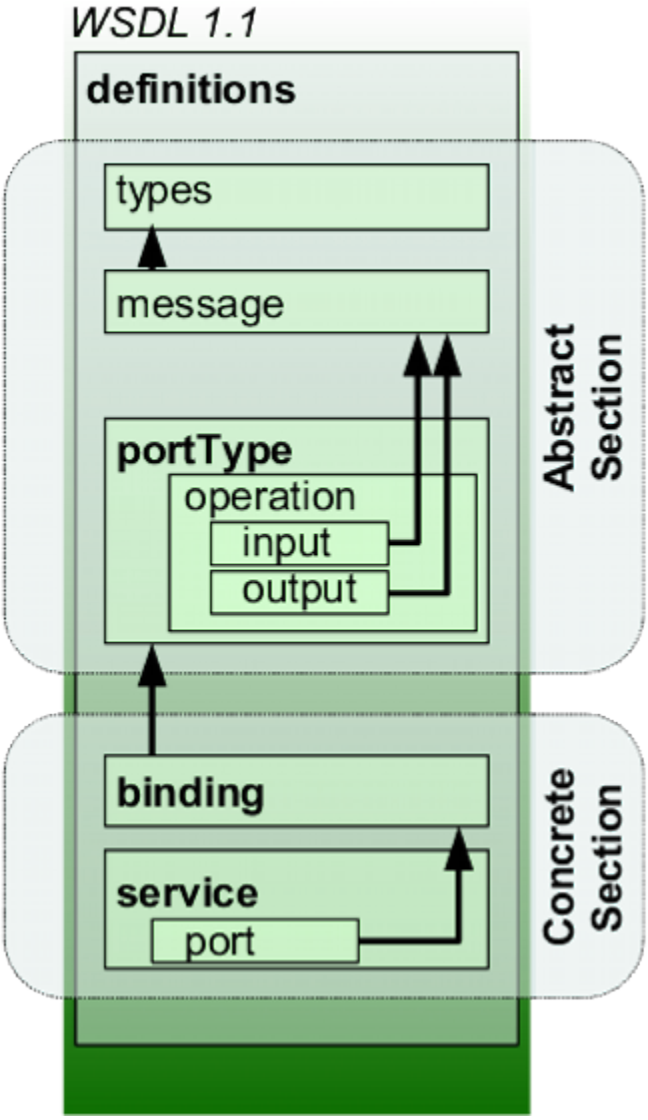

Figure 7. WSDL 1.1 Components

14.1.1.4 Background on Document Wrapped Style WSDL Files

There are three different styles of WSDL files commonly used that are recommended by the Web Services Interoperability (WS-I) Organization. These are Remote Procedure Call (RPC)/literal, document/literal (doc/lit), and a subset of the doc/lit style called doc/lit wrapped.

The WSDL file defined for the aggregate VDB management interface is designed to be a doc/lit wrapped WSDL file. This style was chosen primarily for the following three reasons:

1. Function calls defined by a doc/lit wrapped WSDL consist of sending XML documents that can be validated.
2. A doc/lit wrapped WSDL is WS-I Organization-compliant.
3. A doc/lit wrapped WSDL is compatible with all existing tools and libraries.

14.1.1.5 VICTORY Schemas and WSDLs <10003–20120725, Pro>

The appendix titled "VICTORY Schemas" provides the formal names of the schema files. These files contain additional details that describe the request and response message definitions used in management interface WSDLs, as well as providing data type definitions used to describe management interface parameters.

The section titled "Background on WSDL Files" provides information concerning the use and format of WSDL files. The WSDL files are listed in the appendix titled "VICTORY WSDLs."

14.1.2 Simple Network Management Protocol

Simple Network Management Protocol (SNMP) is a core technology that will be used for implementing the Get and Set functions for many devices with existing management interfaces, such as switches and routers. SNMP will be used to configure and control, as well as monitor the status and health of, these devices.

14.1.2.1 SNMP Core Specifications <10004–20150331, Pro>

SNMP versions 2c and 3 will be used as a core technology for managing some devices. The SNMP Core Specifications are defined by the RFCs described in Table 206. Support for these RFCs is required, as specified in the "VICTORY Compliance" column. It is important to note that full compliance with these RFCs, as they are stated, is not a simple yes or no operation. Most of the RFCs that define MIB modules also define optional and mandatory parts of those MIB modules. So, it is perfectly legitimate to be compliant with an RFC-defined MIB module, but not implement every MIB variable contained within it.

In most cases, the VICTORY specifications make no additional requirements than those already contained within the RFCs' conformance statements, but in some cases optional groups have been made mandatory, or mandatory groups optional. In order to clarify such cases, additional compliance information for the RFCs that define MIB modules is included in the associated compliance tables: Table 207 and Table 208. These associated compliance tables describe compliance requirements at a group, or "unit of conformance," level. These groups are taken directly from the RFCs. Each group has different conformance requirements, called Compliance Policies, which are taken from the RFCs. These Compliance Policies, and the requirements they assign to groups, are described briefly in Table 207. The MIB variable membership of each group is further described in Table 208. The final word on VICTORY compliance with respect to each of the RFC-defined compliance policies is contained in the column "VICTORY Compliance," which is part of Table 207. VICTORY-specified compliance statements apply at a Compliance Policy level, so groups that are optional within a Compliance Policy remain optional (unless specifically called out as otherwise) even though the Compliance Policy itself may be required.

For a full understanding of conformance with these RFCs, especially when it comes to MIB variable groups that are described as "conditionally mandatory" or "mandatory, read-only allowed," please refer to the actual RFCs. The conditions under which such a group is mandatory, for the former case, or read-only, in the latter, are described in depth in the RFCs.

Table 206. Core SNMP Specifications

Specification	Title	ASN.1 Definitions Name	MIB Modules Defined	Module Description	VICTORY Compliance
IETF RFC 1901	Introduction to Community-based SNMPv2	COMMUNITY-BASED-SNMPv2	N/A	N/A	Required for SNMPv2c
IETF RFC 2578	Structure of Management Information Version 2 (SMIv2)	SNMPv2-SMI	N/A	N/A	Required
IETF RFC 2579	Textual Conventions for SMIv2	SNMPv2-TC	N/A	N/A	Required

Specification	Title	ASN.1 Definitions Name	MIB Modules Defined	Module Description	VICTORY Compliance
IETF RFC 2580	Conformance Statements for SMIv2	SNMPv2-CONF	N/A	N/A	Required
IETF RFC 3411	An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks	SNMP-FRAMEWORK-MIB	snmpFrameworkMIB	The SNMP Management Architecture MIB	Required
IETF RFC 3412	Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)	SNMP-MPD-MIB	snmpMPDMIB	The MIB for Message Processing and Dispatching	Required
IETF RFC 3413	Simple Network Management Protocol (SNMP) Applications	SNMP-TARGET-MIB	snmpTargetMIB	This MIB module defines MIB objects which provide mechanisms to remotely configure the parameters used by an SNMP entity for the generation of SNMP messages.	Informational
		SNMP-NOTIFICATION-MIB	snmpNotificationMIB	This MIB module defines MIB objects which provide mechanisms to remotely configure the parameters used by an SNMP entity for the generation of notifications.	
		SNMP-PROXY-MIB	snmpProxyMIB	This MIB module defines MIB objects which provide mechanisms to remotely configure the parameters used by a proxy forwarding application.	
IETF RFC 3414	User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)	SNMP-USER-BASED-SM-MIB	snmpUsmMIB	The management information definitions for the SNMP User-based Security Model.	Required for SNMPv3

Specification	Title	ASN.1 Definitions Name	MIB Modules Defined	Module Description	VICTORY Compliance
IETF RFC 3415	View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)	SNMP-VIEW-BASED-ACM-MIB	snmpVacmMIB	The management information definitions for the View-based Access Control Model for SNMP.	Required for SNMPv3
IETF RFC 3416	Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)	SNMPv2-PDU	N/A	N/A	Required
IETF RFC 3417	Transport Mappings for the Simple Network Management Protocol (SNMP)	SNMPv2-TM	snmpv2tm	The MIB module for SNMP transport mappings.	Required
IETF RFC 3418	Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)	SNMPv2-MIB	snmpMIB	The MIB module for SNMP entities.	Required

Table 207. SNMP Variable Core Compliance Policy Table

Compliance Policy Name	Policy OID	Member Groups	Group Compliance	VICTORY Compliance	Specification	Description
snmpFrameworkMIBCompliance	1.3.6.1.6.3.10.3.1.1	snmpEngineGroup	mandatory	Required	IETF RFC 3411	Compliance statement for SNMP engines which implement the SNMP Management Framework MIB.
snmpMPDCompliance	1.3.6.1.6.3.11.3.1.1	snmpMPDGroup	mandatory	Required	IETF RFC 3412	Compliance statement for SNMP entities which implement the SNMP-MPD-MIB.
usmMIBCompliance	1.3.6.1.6.3.15.2.1.1	usmMIBBasicGroup	mandatory, partial read-only allowed	Required, SNMPv3	IETF RFC 3414	Compliance statement for SNMP engines which implement the SNMP-USER-BASED-SM-MIB.
vacmMIBCompliance	1.3.6.1.6.3.16.2.1.1	vacmBasicGroup	mandatory, partial read-only allowed	Required, SNMPv3	IETF RFC 3415	Compliance statement for SNMP engines which implement the SNMP View-based Access Control Model configuration MIB.
snmpBasicComplianceRev2	1.3.6.1.6.3.1.2.1.3	snmpGroup	mandatory	Required	IETF RFC 3418	Compliance statement for SNMP entities which implement this MIB module.
		snmpCommunityGroup	conditionally mandatory			
		snmpSetGroup	mandatory			
		systemGroup	mandatory			
		snmpBasicNotificationsGroup	mandatory			
		snmpWarmStartNotificationGroup	conditionally mandatory			
		snmpNotificationGroup	conditionally mandatory			

Table 208. SNMP Variable Core Units of Compliance (Groups)

Group Name	Group OID	Member Variables	Type	Specification	Description
snmpEngineGroup	1.3.6.1.6.3.10.3.2.1	snmpEngineID	Object	IETF RFC 3411	Objects for identifying and determining the configuration and current timeliness values of an SNMP engine.
		snmpEngineBoots			
		snmpEngineTime			
		snmpEngineMaxMessageSize			
snmpMPDGroup	1.3.6.1.6.3.11.3.2.1	snmpUnknownSecurityModels	Object	IETF RFC 3412	A collection of objects providing for remote monitoring of the SNMP Message Processing and Dispatching process.
		snmpInvalidMsgs			
		snmpUnknownPDUHandlers			

Group Name	Group OID	Member Variables	Type	Specification	Description
usmMIBBasicGroup	1.3.6.1.6.3.15.2.2.1	usmStatsUnsupportedSecLevels	Object	IETF RFC 3414	Objects providing for configuration of an SNMP engine which implements the SNMP User-based Security Model.
		usmStatsNotInTimeWindows			
		usmStatsUnknownUserNames			
		usmStatsUnknownEngineIDs			
		usmStatsWrongDigests			
		usmStatsDecryptionErrors			
		usmUserSpinLock			
		usmUserSecurityName			
		usmUserCloneFrom			
		usmUserAuthProtocol			
		usmUserAuthKeyChange			
		usmUserOwnAuthKeyChange			
		usmUserPrivProtocol			
		usmUserPrivKeyChange			
		usmUserOwnPrivKeyChange			
		usmUserPublic			
		usmUserStorageType			
		usmUserStatus			

Group Name	Group OID	Member Variables	Type	Specification	Description
vacmBasicGroup	1.3.6.1.6.3.16.2.2.1	vacmContextName	Object	IETF RFC 3415	A collection of objects providing for remote configuration of an SNMP engine which implements the SNMP view-based Access Control Model.
		vacmGroupName			
		vacmSecurityToGroupStorageType			
		vacmSecurityToGroupStatus			
		vacmAccessContextMatch			
		vacmAccessReadViewName			
		vacmAccessWriteViewName			
		vacmAccessNotifyViewName			
		vacmAccessStorageType			
		vacmAccessStatus			
		vacmViewSpinLock			
		vacmViewTreeFamilyMask			
		vacmViewTreeFamilyType			
		vacmViewTreeFamilyStorageType			
		vacmViewTreeFamilyStatus			
snmpGroup	1.3.6.1.6.3.1.2.2.8	snmpInPkts	Object	IETF RFC 3418	Objects providing basic instrumentation and control of an SNMP entity.
		snmpInBadVersions			
		snmpInASNParseErrs			
		snmpSilentDrops			
		snmpProxyDrops			
		snmpEnableAuthenTraps			

Group Name	Group OID	Member Variables	Type	Specification	Description
snmpCommunityGroup	1.3.6.1.6.3.1.2.2.9	snmpInBadCommunityNames	Object	IETF RFC 3418	Objects providing basic instrumentation of an SNMP entity which supports community-based authentication.
		snmpInBadCommunityUses			
snmpSetGroup	1.3.6.1.6.3.1.2.2.5	snmpSetSerialNo	Object	IETF RFC 3418	Objects which allow several cooperating command generator applications to coordinate their use of the set operation.
systemGroup	1.3.6.1.6.3.1.2.2.6	sysDescr	Object	IETF RFC 3418	Objects which are common to all managed systems.
		sysObjectID			
		sysUpTime			
		sysContact			
		sysName			
		sysLocation			
		sysServices			
		sysORLastChange			
		sysORID			
		sysORUpTime			
		sysORDescr			
snmpBasicNotificationsGroup	1.3.6.1.6.3.1.2.2.7	coldStart	Notification	IETF RFC 3418	Basic notifications implemented by an SNMP entity supporting command responder applications.
		authenticationFailure			
snmpWarmStartNotificationGroup	1.3.6.1.6.3.1.2.2.11	warmStart	Notification	IETF RFC 3418	Additional notification for an SNMP entity supporting command responder applications, if it is able to reinitialize itself such that its configuration is unaltered.
snmpNotificationGroup	1.3.6.1.6.3.1.2.2.12	snmpTrapOID	Object	IETF RFC 3418	Objects required for entities which support notification originator applications.
		snmpTrapEnterprise			

14.1.3 Syslog Protocol <10005–20120725, Pro>

Syslog has been selected as a core management technology of the VDB and its components. Specifically, Syslog will support notification of asynchronous events, such as faults, on the VDB.

The Syslog protocol shall be implemented as specified in the following RFCs:

- IETF RFC 5424: The Syslog Protocol
- IETF RFC 5425: Transport Layer Security (TLS) Transport Mapping for Syslog
- IETF RFC 5426: Transmission of Syslog Messages over UDP
- IETF RFC 5427: Textual Conventions for Syslog Management

Although the on-the-wire data formats of Syslog shall be implemented as specified in these RFCs, the transport layer requirements of the RFCs are not the same for VICTORY. Syslog is usually transported over a TCP socket, and support for this transport-layer protocol is a requirement of the Syslog RFCs, but is only a recommendation for VICTORY implementations. The UDP transport for Syslog, although only optional in the above RFCs (IETF RFC 5426, in particular), is the required transport for Syslog messages used by VICTORY.

Although the TCP transport for Syslog is only recommended for VICTORY, it should be noted that if it is available, then the TLS mechanisms described by IETF RFC 5425 are required to be available as well, although they do not all need to be used in every circumstance.

Currently, VICTORY does not make any use of Syslog over a TCP transport. This may change in the future as part of an optional capability (Connection-oriented Health Publishing).

14.2 Common Management Parameters

The following sections describe parameters that are shared across multiple types of management interfaces. The common management parameters are categorized into four groups:

- Version Information – Implemented by all management interfaces.
- VICTORY Data – Implemented by all interfaces that manage VICTORY data being sent to the VDB.
- Device Information – Implemented by all interfaces that manage a device.
- Health – Implemented by all VICTORY components.

14.2.1 Version Information Management Parameters <10007–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the version information management parameters. A textual description of the version information management parameters is shown below.

1. ***Interface Type***
    a. This parameter identifies the type of management interface that is implementing this instance of the Version Information SOAP component.
    b. If the interface is a VICTORY interface, the interface type shall be a string chosen from the "Management Interface Type" column in Table 209 contained in "VICTORY Management Interface Types."
        i. E.g., The Generic End Node Component Type shall have an interface type of GENERIC_END_NODE.
    c. If the interface is a MORA interface, the interface type shall be a string chosen from the "Management Interface Type" column of the "MORA Management Interface Types" table in the section titled "Version Information Management Parameters for MORA" of the appendix titled "Modular Open RF Architecture Specification."
        i. E.g., The MORA Device Component Type shall have an interface type of MORA_DEVICE.
    d. Operations include Get().
2. ***Interface Standard Version***
    a. This parameter identifies the version of the VICTORY standard implemented by the management interface.

b. The interface standard version shall be a string in the format of a major version, followed by a dot, followed by a minor version. The string may also follow the minor version with a dot, followed by a sub-minor version.

> ***NOTE:*** *Sub-minor releases are no longer applicable to the VICTORY standard. If a sub-minor version is provided in the Interface Standard Version, it can be ignored.*

   i. E.g., 1.4 or 1.4.1

c. Operations include Get().

3. ***Interface SOAP Components***
   a. This parameter identifies the network endpoints of the parameters (SOAP interfaces) associated with the management interface.
      i. A SOAP component is described in the section titled "SOAP Component" under "Version Information Complex Types."
      ii. The VERSION_INFORMATION SOAP component may be returned as part of this parameter.
   b. Operations include Get().

> ***NOTE:*** *The section titled "Management Interface Error Types" provides additional details on the errors defined for the version information management parameters along with a description of additional errors that were not specified but may be returned.*

14.2.2 Version Information Complex Types <10031–20210409, Pro>

> ***NOTE:*** *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

1. ***SOAP Component***
   a. **SOAP Component Type**
      i. The type of SOAP component.
      ii. If the interface is a VICTORY interface, the SOAP component type shall be a string chosen from the "SOAP Component Type" column of applicable rows in Table 210 contained in "VICTORY SOAP Component Types."
         1. A row is considered applicable when the value inside the "Management Interface Type" column in Table 210 contained in "VICTORY SOAP Component Types" matches the interface type that implements this instance of the Version Information SOAP Component.
      iii. If the interface is a MORA interface, the SOAP component type shall be a string chosen from the "SOAP Component Type" column of the "MORA SOAP Component Types" table in the section titled "Version Information Management Parameters for MORA" of the appendix titled "Modular Open RF Architecture Specification."
         1. A row is considered applicable when the value inside the "Management Interface Type" column in of the "MORA Management Interface Types" table in the section titled "Version Information Management Parameters for MORA" of the appendix titled "Modular Open RF Architecture Specification" matches the interface type that implements this instance of the Version Information SOAP Component.
      iv. The SOAP component type shall be provided as part of the SOAP component.
   b. **Ports**
      i. A list of WSDL ports associated with the SOAP component.
      ii. A SOAP component shall contain one or more ports.
         1. A port is described in the section titled "Port" under "Version Information Complex Types."
2. ***Port***
   a. **Standard Version**
      i. The version of the port.
      ii. The standard version shall be a string in the format of a major version, followed by a dot, followed by a minor version.
      iii. The standard version shall be a valid WSDL port version as defined in the "WSDL Port Version(s)" column of applicable rows in Table 210 contained in "VICTORY SOAP Component Types."
         1. A row is considered applicable when the value inside the "Management Interface Type" column in Table 210 contained in "VICTORY SOAP Component Types" matches the interface type that implements this instance of the Version Information SOAP Component

and the "SOAP Component Type" column in Table 210 contained in "VICTORY SOAP Component Types" matches the SOAP component type to which this port is associated.

iv. The standard version shall be provided as part of the port.

b. **URI**

i. The URI used in communicating with the WSDL port.

ii. The bounds of the URI are described in the section titled "URI Bounds" under "VICTORY Data Types."

iii. The URI shall be provided as part of the port.

14.2.3 VICTORY Management Interface Types <10032–20210409, Pro>

Table 209 lists the SOAP-based management interface types applicable to VICTORY component types.

NOTE: *This specification only includes language from VICTORY component types whose technical data has been approved for public release. VICTORY Management Interface Types for other VICTORY component types can be found in their respective controlled appendices.*

Table 209. VICTORY Management Interface Types

Component Type	Management Interface Type	Version Applicability					
		V1.4	V1.5	V1.6	V1.7	V1.8	V1.9
Position Service	POSITION	X	X	X	X	X	X
Orientation Service	ORIENTATION	X	X	X	X	X	X
Direction of Travel Service	DIRECTION_OF_TRAVEL	X	X	X	X	X	X
Attribute Store Service	ATTRIBUTE_STORE	X	X	X	X	X	X
Policy Decision Service	POLICY_DECISION	X	X	X	X	X	X
Policy Enforcement Service	POLICY_ENFORCEMENT	X	X	X	X	X	X
Policy Store Service	POLICY_STORE	X	X	X	X	X	X
Authentication Service	AUTHENTICATION			X	X	X	X
Data at Rest Encryptor	DATA_AT_REST	X	X	X	X	X	X
Data Signing Service	DATA_SIGNING		X	X	X	X	X
Signed Data Verification Service	SIGNED_DATA_VERIFICATION		X	X	X	X	X
Host-Based Antivirus	HOST_BASED_ANTIVIRUS	X	X	X	X	X	X

Component Type	Management Interface Type	Version Applicability					
		V1.4	V1.5	V1.6	V1.7	V1.8	V1.9
Host-Based Firewall	HOST_BASED_FIREWALL	X	X	X	X	X	X
Intrusion Detection and Prevention System	INTRUSION_DETECTION_AND_PREVENTION_SYSTEM	X	X	X	X	X	X
Network Firewall	FIREWALL	X	X	X	X	X	X
Shared Processing Unit	SHARED_PROCESSING_UNIT			X	X	X	X
Generic End Node	GENERIC_END_NODE			X	X	X	X
VDB Management Service	VDB_MANAGEMENT			X	X	X	X
Automotive System	AUTOMOTIVE_SYSTEM	X	X	X	X	X	X
Camera Gimbal	CAMERA_GIMBAL	X	X	X	X	X	X
Intercom	INTERCOM			X	X	X	X
Mission Recorder	MISSION_RECORDER			X	X	X	X
Power Distribution System	POWER_DISTRIBUTION_SYSTEM			X	X	X	X
Legacy Radio	LEGACY_RADIO				X	X	X
Remote Weapon Station	REMOTE_WEAPON_STATION	X	X	X	X	X	X
Single-Instance Audio Data Source	SINGLE_INSTANCE_AUDIO_SOURCE			X	X	X	X
Streaming Audio Data Source	STREAMING_AUDIO_SOURCE			X	X	X	X
Threat Detection and Reporting Service	THREAT_DETECTION_AND_REPORTING	X	X	X	X	X	X
Video Encoder	VIDEO_OR_IMAGE_ENCODER	X	X	X	X	X	X

Component Type	Management Interface Type	Version Applicability					
		V1.4	V1.5	V1.6	V1.7	V1.8	V1.9
Video Sensor	VIDEO_OR_IMAGE_SENSOR	X	X	X	X	X	X
Data Logger	DATA_LOGGER			X	X	X	X
Vehicle Configuration Service	VEHICLE_CONFIGURATION_SERVICE			X	X	X	X
LT2 Adapter	LT2_ADAPTER				X	X	X

14.2.4 VICTORY SOAP Component Types <10033–20210409, Pro>

Table 210 lists the SOAP component types that comprise each management interface type along with the WSDL ports defined for each SOAP component.

NOTE: *This specification only includes language from VICTORY component types whose technical data has been approved for public release. VICTORY SOAP Component Types for other VICTORY component types can be found in their respective controlled appendices.*

Table 210. VICTORY SOAP Component Types

Management Interface Type	SOAP Component Type	WSDL Port Version(s)					
		V1.4	V1.5	V1.6	V1.7	V1.8	V1.9
POSITION	POSITION_DATA						X
	VERSION_INFORMATION						X
	VICTORY_DATA						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
ORIENTATION	ORIENTATION_DATA						X
	VERSION_INFORMATION						X
	VICTORY_DATA						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X

Management Interface Type	SOAP Component Type	WSDL Port Version(s)					
		V1.4	V1.5	V1.6	V1.7	V1.8	V1.9
DIRECTION_OF_TRAVEL	DIRECTION_OF_TRAVEL_DATA						X
	VERSION_INFORMATION						X
	VICTORY_DATA						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
ATTRIBUTE_STORE	ATTRIBUTE_STORE						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
POLICY_DECISION	POLICY_DECISION						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
POLICY_ENFORCEMENT	POLICY_ENFORCEMENT						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X

Management Interface Type	SOAP Component Type	WSDL Port Version(s)					
		V1.4	V1.5	V1.6	V1.7	V1.8	V1.9
POLICY_STORE	POLICY_STORE						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
AUTHENTICATION	AUTHENTICATION						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
DATA_AT_REST	DARE						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
DATA_SIGNING	DATA_SIGNING						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X

Management Interface Type	SOAP Component Type	WSDL Port Version(s)					
		V1.4	V1.5	V1.6	V1.7	V1.8	V1.9
SIGNED_DATA_VERIFICATION	SIGNED_DATA_VERIFICATION						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
HOST_BASED_ANTIVIRUS	HOST_BASED_ANTIVIRUS						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
HOST_BASED_FIREWALL	HOST_BASED_FIREWALL						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X

Management Interface Type	SOAP Component Type	WSDL Port Version(s)					
		V1.4	V1.5	V1.6	V1.7	V1.8	V1.9
INTRUSION_DETECTION_AND_PREVENTION_SYSTEM	IDPS						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
FIREWALL	FIREWALL						X
	DEVICE_INFORMATION						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
SHARED_PROCESSING_UNIT	SHARED_PROCESSING_UNIT						X
	GENERIC_END_NODE						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X

Management Interface Type	SOAP Component Type	WSDL Port Version(s)					
		V1.4	V1.5	V1.6	V1.7	V1.8	V1.9
GENERIC_END_NODE	GENERIC_END_NODE						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
VDB_MANAGEMENT	HEALTH_LEVEL						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
AUTOMOTIVE_SYSTEM	AUTOMOTIVE_DATA						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X

Management Interface Type	SOAP Component Type	WSDL Port Version(s)					
		V1.4	V1.5	V1.6	V1.7	V1.8	V1.9
CAMERA_GIMBAL	CAMERA_GIMBAL_DATA						X
	CAMERA_GIMBAL_DEVICE						X
	DEVICE_INFORMATION						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
INTERCOM	INTERCOM						X
	DEVICE_INFORMATION						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X

Management Interface Type	SOAP Component Type	WSDL Port Version(s)					
		V1.4	V1.5	V1.6	V1.7	V1.8	V1.9
MISSION_RECORDER	MISSION_RECORDER						X
	DEVICE_INFORMATION						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
POWER_DISTRIBUTION_SYSTEM	POWER_DISTRIBUTION_SYSTEM_DATA						X
	POWER_DISTRIBUTION_SYSTEM_DEVICE						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
LEGACY_RADIO	LEGACY_RADIO_DEVICE						X
	DEVICE_INFORMATION						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X

Management Interface Type	SOAP Component Type	WSDL Port Version(s)					
		V1.4	V1.5	V1.6	V1.7	V1.8	V1.9
REMOTE_WEAPON_STATION	REMOTE_WEAPON_STATION_DEVICE						X
	VICTORY_DATA						X
	DEVICE_INFORMATION						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
SINGLE_INSTANCE_AUDIO_SOURCE	SINGLE_INSTANCE_AUDIO_DATA						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
STREAMING_AUDIO_SOURCE	STREAMING_AUDIO_DATA						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X

Management Interface Type	SOAP Component Type	WSDL Port Version(s)					
		V1.4	V1.5	V1.6	V1.7	V1.8	V1.9
THREAT_DETECTION_AND_REPORTING	THREAT_DATA						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
VIDEO_OR_IMAGE_ENCODER	VIDEO_DATA						X
	VIDEO_ENCODER						X
	DEVICE_INFORMATION						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X

Management Interface Type	SOAP Component Type	WSDL Port Version(s)					
		V1.4	V1.5	V1.6	V1.7	V1.8	V1.9
VIDEO_OR_IMAGE_SENSOR	VIDEO_SENSOR						X
	IR_VIDEO_SENSOR						X
	DEVICE_INFORMATION						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
DATA_LOGGER	DATA_LOGGER						X
	VICTORY_DATA						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X
VEHICLE_CONFIGURATION_SERVICE	VEHICLE_CONFIGURATION						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X

Management Interface Type	SOAP Component Type	WSDL Port Version(s)					
		V1.4	V1.5	V1.6	V1.7	V1.8	V1.9
LT2_ADAPTER	LT2_ADAPTER_DATA						X
	VERSION_INFORMATION						X
	HEALTH_PUBLISHING						X
	CONFIGURATION						X
	AA_ENTRYPOINTS						X
	AUTHN_ENTRYPOINTS						X

14.2.5 VICTORY Data Management Parameters <10008–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the VICTORY data management parameters. A textual description of the VICTORY data management parameters is shown below.

1. ***Data Enabled***
    a. This parameter identifies whether the VICTORY Data Interface managed by this interface is sending out VICTORY Data Messages (VDMs) on the VDB.
    b. A Boolean value of True means "enabled" (sending) and a value of False means "disabled" (not sending).
    c. Operations include Get() and Set().
2. ***Time Uncertainty***
    a. This parameter identifies the uncertainty of the timestamp applied to the VDMs being sent to the VDB by the VICTORY Data Interface managed by this interface.
    b. The measure of time uncertainty shall be in units of milliseconds.
    c. The time uncertainty shall be a real value at 1 σ.
    d. Operations include Get().
3. ***Data Transport Configuration***
    a. This parameter identifies the transport configurations used by the VICTORY Data Interface managed by this interface for sending out VDMs on the VDB.
    b. The contents of a data transport configuration are described in the section titled "Data Transport Configuration" under "VICTORY Data Interface Complex Types."
    c. The following errors are defined for the data transport configuration parameter:
        i. Setting multiple data transport configurations with the same unique name shall result in a GENERIC_FAULT error.
        ii. If network layer is included in the request, setting a data transport configuration with a network layer outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
        iii. Setting a data transport configuration with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
        iv. Setting a data transport configuration with a URI outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.
    d. Operations include Get() and Set().
4. ***Update Period***
    a. This parameter identifies the time between VDM updates sent by the VICTORY Data Interface managed by this interface.
    b. The update period shall be in units of milliseconds.
    c. The update period shall be a real value greater than zero.
    d. The following errors are defined for the update period parameter:
        i. Setting the update period to a value outside the bounds defined by this specification document may result in an OUT_OF_RANGE error.
        ii. Setting the update period to a value less than the minimum update period shall result in a NEAREST_VALUE error.
            1. The update period shall be changed to the minimum update period and shall be sent back in the response.
    e. Operations include Get() and Set().
5. ***Minimum Update Period***
    a. This parameter identifies the maximum frequency at which the VDM updates sent by the VICTORY Data Interface managed by this interface can occur.
    b. The minimum update period shall be in units of milliseconds.
    c. The minimum update period shall be a real value greater than zero.
    d. Operations include Get().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the VICTORY data management parameters along with a description of additional errors that were not specified but may be returned.*

14.2.5.1 VICTORY Data Interface Complex Types

NOTE: *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

1. ***Data Transport Configuration***
    a. **Unique Name**
        i. The unique name of a data transport configuration.
        ii. The bounds of the unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
        iii. The unique name shall be unique within the scope of all data transport configurations defined for the VICTORY Data Interface managed by this interface.
        iv. A data transport configuration shall provide a unique name.
    b. **Index**
        i. An index of a transport configuration to allow for ordinality of multiple transport configurations.
        ii. The index shall be an integer.
        iii. A data transport configuration may provide an index.
    c. **URI**
        i. Identifies the location from which VDMs are being sent to the VDB.
        ii. The bounds of the URI are described in the section titled "URI Bounds" under "VICTORY Data Types."
        iii. A data transport configuration shall provide a URI.
    d. **Network Layer**
        i. Describes the network protocol stack of a data transport configuration.
        ii. The network layer consists of hierarchical layer names, where each layer name shall be a string chosen from one of the following values:
            1. TCP
            2. UDP
            3. ICMP
            4. IGMP
            5. IPV4
            6. IPV6
            7. HTTP
            8. FTP
            9. SOAP
            10. SCTP
            11. VDM
        iii. A data transport configuration may provide a network layer.

14.2.6 Device Information Management Parameters <10009–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the device information management parameters. A textual description of the device information management parameters is shown below.

1. ***Manufacturer***
    a. This parameter identifies the manufacturer of the device being managed by this interface.
    b. The bounds of the manufacturer are described in the section titled "String Bounds" under "VICTORY Data Types."
    c. Operations include Get().
2. ***Model***
    a. This parameter identifies the model of the device being managed by this interface.
    b. The bounds of the model are described in the section titled "String Bounds" under "VICTORY Data Types."
    c. Operations include Get().
3. ***Equipment Identifier***
    a. This parameter identifies the unique equipment identifier of the device being managed by this interface.
    b. The bounds of the equipment identifier are described in the section titled "String Bounds" under "VICTORY Data Types."
    c. Operations include Get().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the device information management parameters along with a description of additional errors that were not specified but may be returned.*

14.2.7 Health Publishing Management Parameters

In order to control the Health Publishing Interfaces of VICTORY components, health publishing management parameters are provided. The management interface for the Syslog-Based Health Publishing Interface is a SOAP interface. The SNMP-Based Health Publishing Management Interface is an SNMP interface.

14.2.7.1 Syslog-Based Health Publishing Management Specification <11005–20210409, Pro>

The management of Syslog-based health publishing requires the implementation of a set of parameters that are a component of most VICTORY management interfaces. Descriptions of the parameters and types associated with syslog-based health publishing management are contained in the following sections.

14.2.7.1.1 Syslog-Based Health Publishing Management Parameters

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the syslog-based health publishing management parameters. A textual description of the syslog-based health publishing management parameters is shown below.

1. ***Data Transport Configuration***
    a. This parameter identifies the transport configurations used in sending asynchronous fault and status messages to the VDB.
    b. The contents of a data transport configuration are described in the section titled "Data Transport Configuration" under "VICTORY Data Interface Complex Types."
    c. Setting the data transport configurations shall result in the removal of all pre-existing transport configurations and the addition of all transport configurations provided in the request.
    d. The following errors are defined for the data transport configuration parameter:
        i. Setting multiple data transport configurations with the same unique name shall result in a GENERIC_FAULT error.
        ii. Adding a data transport configuration with a pre-existing unique name shall result in a GENERIC_FAULT error.
        iii. Removing a data transport configuration with a unique name that does not match the unique name of one of the data transport configurations of the health publishing interface shall result in a GENERIC_FAULT error.
        iv. Removing the only data transport configuration for the health publishing interface shall result in a GENERIC_FAULT error.
        v. If network layer is included in the request, setting or adding a data transport configuration with a network layer outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
        vi. Setting, adding, or removing a data transport configuration with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
        vii. Setting or adding a data transport configuration with a URI outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.
    e. Operations include Get(), Set(), Add(), and Remove().
2. ***Update Period***
    a. This parameter identifies the rate at which fault and status messages are sent to the VDB.
    b. Fault and status messages each have separate update rates.
        i. E.g., Fault messages could be sent out every 1000 milliseconds and status messages could be sent out every 5000 milliseconds.
    c. The update period shall be in units of milliseconds.
    d. The update period shall be a real value greater than zero.
    e. The following errors are defined for the update period parameter:
        i. Setting the update period to a value outside the bounds defined by this specification document may result in an OUT_OF_RANGE error.

f. Though a very small update period (e.g., an update period of five) is valid according to the specification document, the actual update period may be higher depending on the performance of the system. If an update period is set to a lower rate than what is achievable by the system, the system shall send out asynchronous fault and status messages as fast as possible.
g. Operations include Get() and Set().

3. ***Fault Severity Level***
    a. This parameter identifies which fault messages are being sent to the VDB with respect to the severity level of the fault.
    b. Enabling a fault severity level allows fault messages of the specified severity level to be sent to the VDB.
    c. Disabling a fault severity level prevents fault messages of the specified severity level from being sent to the VDB.
    d. Each fault severity level shall be a string chosen from one of the following values:
        i. EMERGENCY
        ii. ALERT
        iii. CRITICAL
        iv. ERROR
        v. WARNING
    e. The following errors are defined for the fault severity level parameter:
        i. Enabling or disabling a fault severity level with a fault severity level not inside the list of known fault severity levels may result in an OUT_OF_RANGE error.
    f. Operations include Get(), Enable(), and Disable().
4. ***Status Enabled***
    a. This parameter identifies whether status messages are being sent to the VDB.
    b. A Boolean value of True means “enabled” (sending) and a value of False means “disabled” (not sending).
    c. Operations include Get() and Set().
5. ***Acknowledge Fault***
    a. This parameter stops a fault from being periodically sent to the VDB.
    b. A fault is acknowledged based on its fault identifier and the unique name of the service/component where the fault originated.
    c. The following errors are defined for the acknowledge fault parameter:
        i. Acknowledging a fault with an identifier that does not match any of the faults of the specified fault origin shall result in a GENERIC_FAULT error.
        ii. Acknowledging a fault with an origin that does not match the unique name of an interface, service, or VICTORY component type on the VDB shall result in a GENERIC_FAULT error.
        iii. Only fault messages with the acknowledgeable flag set to True may be acknowledged. Acknowledging a fault that is not acknowledgeable shall result in a GENERIC_FAULT error.
        iv. Acknowledging a fault with an origin outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    d. A detailed description of the acknowledgeable flag is provided in the section titled “acknowledgeable” under “Syslog Protocol Data Parameters.”
    e. Operations include Get() and Set().
6. ***Resume Fault***
    a. This parameter cancels the previous acknowledgment of a fault, resulting in the fault periodically being sent to the VDB.
    b. A fault is resumed based on its fault identifier and the unique name of the service/component where the fault originated.
    c. The following errors are defined for the resume fault parameter:
        i. Resuming a fault with an identifier that does not match any of the faults of the specified fault origin shall result in a GENERIC_FAULT error.
        ii. Resuming a fault with an origin that does not match the unique name of an interface, service, or VICTORY component type on the VDB shall result in a GENERIC_FAULT error.
        iii. Resuming a fault with an origin outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    d. Operations include Set().
7. ***Active Fault***
    a. This parameter identifies a fault that has not yet been resolved.
    b. A list of faults that are currently active (OPERATIVE or ACKNOWLEDGED) can be retrieved.
    c. A detailed description of a fault is provided in the section titled “Fault” under “Syslog-Based Health Publishing Management Complex Types.”

d. Operations include Get().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the syslog-based health publishing management parameters along with a description of additional errors that were not specified but may be returned.*

14.2.7.1.2 Syslog-Based Health Publishing Management Complex Types

NOTE: *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

1. ***Fault***
    a. **Fault ID**
        i. This parameter identifies a fault.
        ii. This parameter shall be unique within an interface, but may be unique within a service or VICTORY component type.
        iii. This parameter shall be an integer greater than or equal to zero.
        iv. The fault identifier and fault origin shall be used for uniquely identifying a fault in the system. See the section titled "Maintaining Fault and Status Uniqueness" for more information on ensuring fault uniqueness.
        v. The fault ID shall be provided as part of the fault.
    b. **Origin**
        i. This parameter identifies the source of the fault.
        ii. The bounds of the origin are described in the section titled "String Bounds" under "VICTORY Data Types."
        iii. This parameter shall be unique within the scope of the VDB, representing the unique name of an interface, service, or VICTORY component type.
        iv. The fault identifier and fault origin shall be used for uniquely identifying a fault in the system. See the section titled "Maintaining Fault and Status Uniqueness" for more information on ensuring fault uniqueness.
        v. The origin shall be provided as part of the fault.
    c. **Fault Severity**
        i. This parameter identifies the severity of the fault.
        ii. The severity shall be a string chosen from one of the following values:
            1. EMERGENCY
            2. ALERT
            3. CRITICAL
            4. ERROR
            5. WARNING
        iii. The severity shall be provided as part of the fault.
    d. **Facility**
        i. This parameter identifies the facility of the fault.
        ii. The facility shall be a string chosen from one of the following values:
            1. DATA_INTERFACE
            2. MANAGEMENT_INTERFACE
            3. SERVICE
            4. HARDWARE_NODE
            5. UNCATEGORIZED
        iii. The facility shall be provided as part of the fault.
    e. **Category**
        i. This parameter identifies the category of the fault.
        ii. The category shall be a string chosen from one of the following values:
            1. CONFIGURATION_FAULT
            2. DATA_FORMAT_FAULT
            3. DATA_TRANSPORT_FAULT
            4. HARDWARE_FAULT
            5. POWER_FAULT
            6. SOFTWARE_FAULT

iii. The category may be provided as part of the fault.

f. **Suggested Action**
   i. This parameter identifies the suggested action as a result of the fault.
   ii. The suggested action shall be a string chosen from one of the following values:
      1. MONITOR
      2. RECONFIGURE
      3. RESTART_HARDWARE
      4. RESTART_SOFTWARE
      5. SHUTDOWN
   iii. The suggested action may be provided as part of the fault.
      1. Not including a value shall indicate no action is suggested for the fault.

g. **State**
   i. This parameter identifies the state of the fault.
   ii. The state shall be a string chosen from one of the following values:
      1. OPERATIVE
      2. ACKNOWLEDGED
      3. RESOLVED
   iii. The state shall be provided as part of the fault.

h. **Acknowledgeable**
   i. Indicates whether a fault can be acknowledged, and therefore, no longer periodically sent to the VDB via the Syslog protocol.
   ii. A message that will be periodically sent to the VDB shall be provided with a True value.
   iii. A Boolean value of True means the fault is acknowledgeable (will be periodically sent to the VDB until acknowledged) and a value of False means the fault is not acknowledgeable (will only be sent to the VDB once).
   iv. The acknowledgeable flag shall be provided as part of the fault.

i. **Message**
   i. This parameter identifies the actual fault message.
   ii. The bounds of the message are described in the section titled "String Bounds" under "VICTORY Data Types."
   iii. The message shall be provided as part of the fault.

j. **Timestamp**
   i. This parameter identifies the time the fault occurred in the system.
   ii. This parameter shall be in the format as described in the section titled "Timestamp Format for Application Layer Data."
   iii. The timestamp shall be provided as part of the fault.

14.2.7.1.3 Mapping Between Syslog and XML Health Parameters

Syslog provides built-in parameters for describing health and status messages. In addition to these parameters, VICTORY has defined three additional Syslog parameters that are required when sending Syslog messages asynchronously across the VDB. When querying for health data from the Syslog-Based Health Publishing Management Parameters (a SOAP interface, and part of a larger management interface), the health parameters defined in the section titled "Syslog-Based Health Publishing Management Complex Types" are returned. These returned values are XML-based, and not Syslog. The table below provides a mapping between the Syslog parameters and the parameters returned when querying for health data.

Table 211. Health Parameter Mapping

Syslog Parameter	VICTORY Fault Parameter	
PRI	Facility	Fault Severity
VERSION	N/A	
TIMESTAMP	Timestamp	
HOSTNAME	N/A	
APP-NAME	Origin	

Syslog Parameter	VICTORY Fault Parameter
PROCID	N/A
MSGID	Fault Category
SD-ID	N/A
id	Fault ID
state	State
acknowledgeable	Acknowledgeable
suggestedAction	Suggested Action
MSG	Message

14.2.7.2 SNMP-Based Health Publishing Management <Preliminary>

VICTORY-compliant infrastructure nodes shall provide health information to clients through a group of health publishing management parameters built upon SNMP. The specific details of the SNMP-Based Health Publishing Management Parameters for infrastructure nodes are described in the section titled "Generic Infrastructure Component Health Status Specification."

14.2.8 Management Interface Error Types <10013–20210409, Pro>

Error responses provide important feedback to a user when sending a SOAP request to a management interface. A response to a request for data shall return either the requested data or an error response indicating an error in processing the request. A response to a request for modifying existing data shall always return an error.

The following provides the possible errors returned from a management interface:

1. ***NO_ERROR***
    a. This error indicates that the management interface was successful in processing the SOAP request it received.
    b. A management interface shall not return an error response of NO_ERROR when data is being requested.
    c. For any request to modify existing data, a management interface shall return an error response of NO_ERROR when the request is successfully processed.
    d. When controlling a device, an error response of NO_ERROR does not indicate the device was successfully modified, only that the device has successfully received the data sent in the request. An additional request to retrieve the data may be necessary to verify that the device was successfully modified.
    e. When this error is returned, an additional error description may be provided.
        i. The bounds of the error description are described in the section titled "String Bounds" under "VICTORY Data Types."
2. ***NOT_SUPPORTED***
    a. A management interface shall be able to handle all SOAP requests defined in its WSDL file. If it does not actually implement the SOAP request, it shall return an error response of NOT_SUPPORTED.
    b. When this error is returned, an additional error description may be provided.
        i. The bounds of the error description are described in the section titled "String Bounds" under "VICTORY Data Types."
    c. Unless otherwise specified, NOT_SUPPORTED shall not be returned.
3. ***OUT_OF_RANGE***
    a. This error indicates that the data sent in the SOAP request was outside the range defined by this specification document.
    b. A management interface shall not return an error response of OUT_OF_RANGE if the data inside the request does not contain data restricted by this specification document.
    c. A management interface shall not return an error response of OUT_OF_RANGE if the bounded data provided in the request is not handled by the management interface.

d. A management interface shall return an error response of OUT_OF_RANGE if the data sent in the request is outside the bounds defined by this specification document and the management interface is handling the bounded data.
e. When this error is returned, the management interface shall not update the existing data with the data provided in the request.
f. When this error is returned, an additional error description may be provided.
    i. The bounds of the error description are described in the section titled "String Bounds" under "VICTORY Data Types."
g. Management parameters with upper and/or lower bounds are restricted inside both the specification document and the VICTORY Management Types schema. Some libraries used for the development of SOAP web services, such as gSOAP, handle bounded values inside the library itself. This prevents the management interface from sending a VICTORY-defined error when an out-of-bounds error occurs. If the management interface is not handling the out-of-bounds values, a library-specific error shall be provided. This error shall be an unrestricted string and the value that was being modified shall remain unchanged.
h. If a management request results in an error, and both OUT_OF_RANGE and GENERIC_FAULT are valid responses as per the specification, the OUT_OF_RANGE error will take higher precedence. This means the management interface shall not return a GENERIC_FAULT error and shall return an OUT_OF_RANGE error if the management interface is handling the bounded data. If the management interface is not handling the bounded data, a library-specific error shall be provided.
    i. E.g. Setting multiple data transport configurations with the same unique name typically results in a GENERIC_FAULT error. However, if the duplicated unique name is also outside the string bounds defined inside this specification document, an OUT_OF_RANGE error is also valid. In this case the management interface would return an OUT_OF_RANGE error if it is handling the bounded data.

4. ***NEAREST_VALUE***
    a. This error indicates that the data sent in the SOAP request was inside the bounds defined by this specification document but outside the scope of the device being controlled.
    b. A management interface shall return an error response of NEAREST_VALUE if the data sent in the request is within the bounds defined by this specification document but outside the scope of the device being controlled.
    c. When this error is returned, the management interface shall update the existing data to the nearest possible data understood by the device.
    d. When this error is returned, the management interface shall return the updated data in the response.
    e. When this error is returned, an additional error description may be provided.
        i. The bounds of the error description are described in the section titled "String Bounds" under "VICTORY Data Types."
5. ***NOT_AUTHORIZED***
    a. This error indicates that the source of the SOAP request has not yet been authenticated.
    b. When this error is returned, an additional error description may be provided.
        i. The bounds of the error description are described in the section titled "String Bounds" under "VICTORY Data Types."
6. ***AUTHORIZATION_EXPIRED***
    a. This error indicates that the source of the SOAP request was authenticated, but the security token used in the SOAP request has expired.
    b. When this error is returned, an additional error description may be provided.
        i. The bounds of the error description are described in the section titled "String Bounds" under "VICTORY Data Types."
7. ***GENERIC_FAULT***
    a. This error indicates that the management interface did not successfully process the SOAP request it received, or the error was not applicable to any of the other errors defined in this specification document.
    b. When a generic fault occurs, the management interface shall not execute the request.
    c. When this error is returned, an additional error description should be provided.
        i. The bounds of the error description are described in the section titled "String Bounds" under "VICTORY Data Types."
    d. If a management request results in an error, and both OUT_OF_RANGE and GENERIC_FAULT are valid responses as per the specification, the OUT_OF_RANGE error will take higher precedence. This means the management interface shall not return a GENERIC_FAULT error and shall return an OUT_OF_RANGE error if the management interface is handling the bounded data. If the management interface is not handling the bounded data, a library-specific error shall be provided.
        i. E.g. Setting multiple data transport configurations with the same unique name typically results in a GENERIC_FAULT error. However, if the duplicated unique name is also outside the string bounds

defined inside this specification document, an OUT_OF_RANGE error is also valid. In this case the management interface would return an OUT_OF_RANGE error if it is handling the bounded data.

14.3 Health Management

14.3.1 Health Management Overview <Informational>

From a VICTORY perspective, health management encompasses two related concepts: status and fault.

Status is related to the instantaneous state of a component (device or service). A status interface answers the question, "How well are you functioning?" A status interface is typically implemented as a synchronous and periodic report of the running state of a service or component, although, in some cases state changes may be conveyed as soon as they occur in an asynchronous manner.

Faults represent non-normative occurrences and are typically reported as soon as detected in an asynchronous manner. Faults often represent detrimental occurrences, but can usually be spaced on a spectrum of severity (e.g., from warning to critical). A fault interface typically makes the statement, "An event occurred of which you must be made aware."

For VICTORY-compliant services and nodes, both status and fault messages are managed under the concept of health management. The following sections describe the elements of health management. Many of these elements are only required by specific services and nodes in a VICTORY implementation. Although many of the requirements are called out in the following sections, the primary authority is the section titled "VICTORY System and Component Type Specifications." That section describes the requirement mappings in component specification tables.

14.3.2 Components of VICTORY Health Management <Informational>

VICTORY Health Management currently contains two high-level components. These components are:

1. Health Publishing Interface
    a. Provides the asynchronous health messages sent to the VDB.
    b. Implemented by every VICTORY service and device.
2. Health Publishing Management Interface
    a. Provides control over and the details of a Health Publishing Interface.
    b. Implemented by every VICTORY service and device.

There are two versions of each of these components, each of which is based on a different underlying publishing protocol (either Syslog or SNMP):

1. Syslog-Based Health Publishing:
    a. The Syslog-Based Health Publishing Interface for VICTORY services and generic end nodes is described in the section titled "Syslog-Based Data Transport for Health Publishing."
    b. The Syslog-Based Health Publishing Management Parameters for VICTORY services and generic end nodes is described in the section titled "Syslog-Based Health Publishing Management Parameters." These parameters are a component of the larger management interface of a service or generic end node.
2. SNMP-Based Health Publishing:
    a. The SNMP-Based Health Publishing Interface for infrastructure nodes is described in the section titled "SNMP-Based Data Transport for Health Publishing." Although this section exists, it is currently a stub, with additional details to be provided in a future revision.
    b. The SNMP-Based Health Publishing Management Interface for infrastructure nodes is described in the section titled "SNMP-Based Health Publishing Management." Although this section exists, it is currently a stub, with additional details to be provided in a future revision.

There are additional health components that are not yet defined. These additional components of VICTORY Health Management are primarily related to infrastructure nodes.

In order to better understand how the components of VICTORY Health Management relate to each other, refer to the section titled "Health Management Scenarios."

14.3.2.1 Syslog-Based Health Publishing <Informational>

Syslog-Based Health Publishing is primarily used for services and end nodes. It is built from two required components and one optional component. The combination of these components provides the Health Publishing Interface for a VICTORY service (such as a Position Service), or a generic end node (such as an SPU).

In order to provide a Health Publishing Interface, the following components are provided by a VICTORY service or generic end node:

- **Syslog-Based Health Publishing Management Parameters** – Contains parameters for controlling how fault and status messages are reported asynchronously over the VDB. This specification is described in the section titled "Syslog-Based Health Publishing Management Parameters."
- **Syslog-Based Connectionless Health Publishing Specification** – The asynchronous one-to-many data stream that relays fault and status messages from a service or generic end node to interested clients. Because this specification uses UDP multicast, it is not required that the service or generic end node keep track of interested clients. This specification is described in the section titled "Syslog-Based Connectionless Health Publishing Specification."

In addition to the components above, a Health Publishing Interface implemented by a service or generic end node may also include the following component:

- **Syslog-Based Connection-Oriented Health Publishing Specification** – A one-to-one (socket-based) version of the asynchronous data stream that relays fault and status messages from a service or generic end node to a single interested client. Implementing this specification implies that the service or generic end node will maintain a list of interested clients and will open a separate data stream (socket) for each. This specification is described in the section titled "Syslog-Based Connection-Oriented Health Publishing Specification."

Figure 8 describes how the Health Publishing Interface fits into the context of a VICTORY Service, while Figure 9 details how the Health Publishing Interface fits into the context of a Generic End Node.

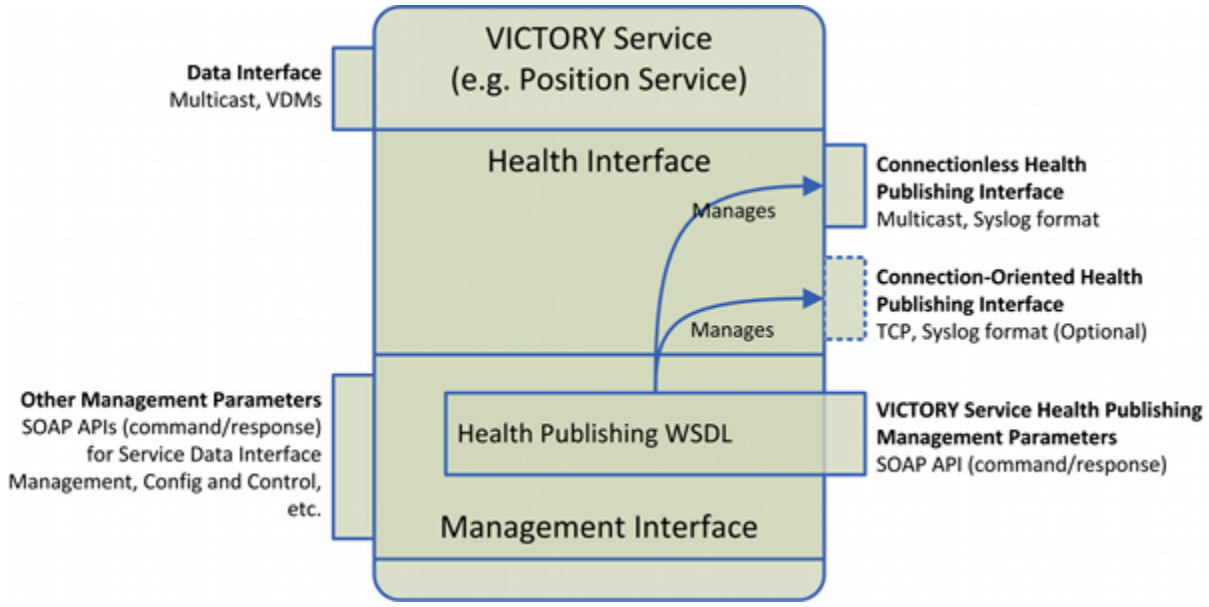

Figure 8. The Health Publishing Interface of a VICTORY Service

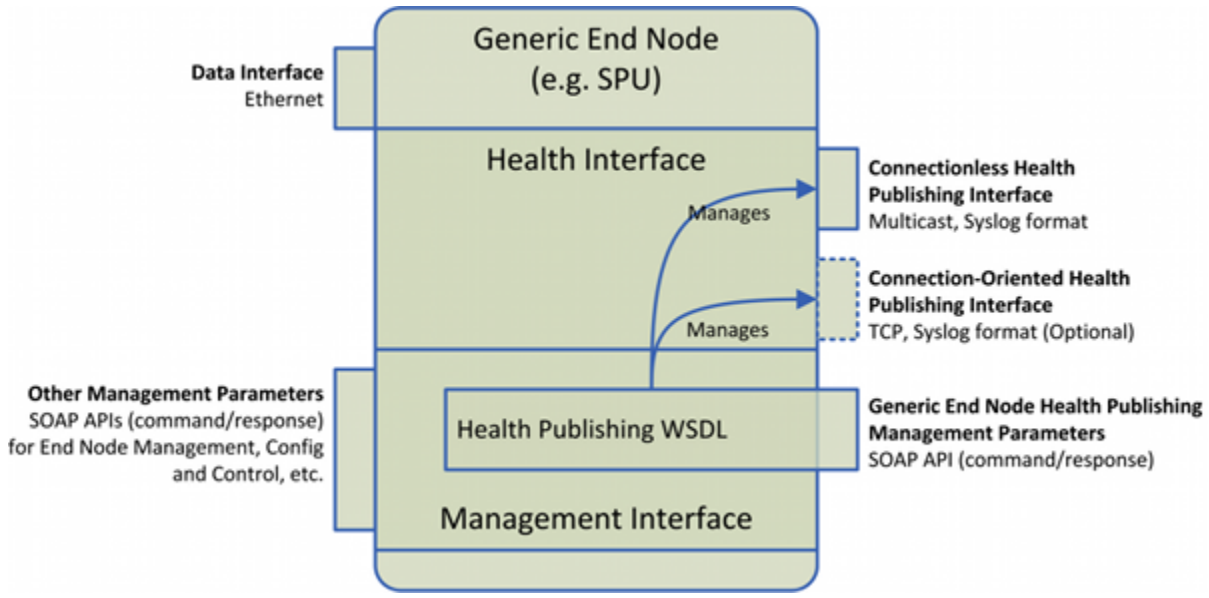

Figure 9. The Health Publishing Interface of a Generic End Node

14.3.3 Health Management Scenarios <Informational>

The following diagram is an attempt to answer the question, "How do the components of VICTORY Health Management interact?"

14.3.3.1 Scenario: Service and End Node Health Publishing

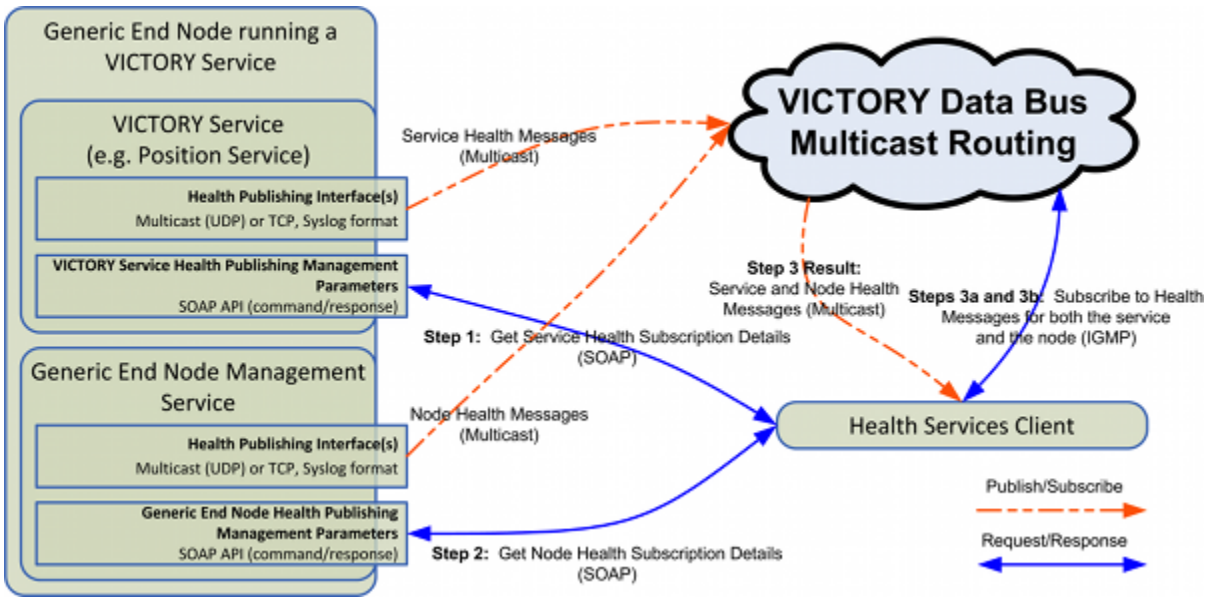

Figure 10. Service and End Node Health Publishing Example

Figure 10 describes Health Publishing, the most basic component of VICTORY Health Management. To clarify the diagram in Figure 10, the following simplifications have been made:

- The network infrastructure of the VDB has not been shown. There are infrastructure nodes (switches and routers) between the client and the end node.
- In order to show that the data is multicast, and so handled by the network infrastructure that is not shown, a "cloud" is used. The cloud represents miscellaneous switches and routers that keep track of multicast subscriptions and routing automatically.
- The end node and its services have been simplified. In reality, an end node and a service would contain many more management interfaces, as well as data interfaces.
- The Health Publishing Interfaces of the service and the end node have been shown at a high level. In actuality, the Health Publishing Interfaces of each can involve multiple Syslog interfaces. These are called out in a separate section.
- Only the one-to-many topology (the connectionless, multicast version) for health publishing is being displayed. There is also a one-to-one (connection-oriented) version of health publishing, but many of its details are still TBD.

In the illustration, a health client needs to monitor health messages being reported by a VICTORY Service (e.g., a Position Service). In addition, the client would like to monitor health messages being generated by the end node on which the service is running. In order to do so, it must know where the service and the end node are publishing their health messages (i.e., the multicast group). In order to achieve this, the client takes the following steps:

- **Step 1:** Get the health subscription details (multicast group) of the service from the service's Health Publishing Interface.
  - This involves making a "Get Data Transport Configuration" SOAP function call to the Health Publishing Management Parameters portion of the service's Management Interface.
  - The Health Publishing Management Parameters are described by the "Health Publishing WSDL." See the section titled "Syslog-Based Health Publishing Management Parameters" for additional details.
- **Step 2:** Get the health subscription details (multicast group) of the end node's Health Publishing Interface.
  - This is essentially identical to the function call made against the service in Step 1, as it is defined by the same WSDL. Consult the same section for additional details.
- **Step 3a:** Subscribe to the multicast group for the service's Health Publishing Interface.
  - Use a standard IGMP (or MLD for IPv6) multicast subscription request.
- **Step 3b:** Subscribe to the multicast group for the end node's Health Publishing Interface.
  - Again, use IGMP (or MLD).

After following these steps, the client should begin to receive health messages from both the service and the end node. These messages are formatted as a variant of the Syslog protocol. Additional details on this protocol, and its use for VICTORY health messages, can be found in the section titled "Syslog-Based Data Transport for Health Publishing."

14.3.4 Health Level Management Parameters <10030–20210409, Pro>

1. ***Health Level***
   a. This parameter indicates the current health level of the VICTORY component.
   b. The health level shall be a string chosen from one of the values defined in the section titled "Health Levels for End Users."
   c. Operations include Get().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the health level management parameters along with a description of additional errors that were not specified but may be returned.*

14.3.5 Health Levels for End Users <10022–20131031, Pro>

In order to simplify the understanding of health message severity for end users, health status levels shall be used whenever health information is conveyed to an end user of a VICTORY implementation (e.g., in a health reporting GUI). Health status levels describe the status, or state, of a component or service of the VICTORY Data Bus in terms of color. The primary intent is to provide an "indicator light" describing the overall health of the managed interface. The operating status is dependent on the state of a set of "key health components." These "key health components" are components that affect the primary functionality of a node or service (e.g., GPS lock for a position service). The selection of "key health components" for a node or service is left to the discretion of the service or node designer, although in the future this functionality may be added to the VCL.

14.3.5.1 Health Colors

Each “key health component” shall have a health status level with one of the following enumerated values:

1. ***RED***
    a. At least one “key health component” has an active fault condition that causes that component to be non-operational (a major feature is not available).
    b. Used whenever a node or service has one or more unresolved fault messages with an EMERGENCY or ALERT severity.
    c. For aggregate status, red overrides orange (see the section titled “VDB Aggregate Health”).
2. ***ORANGE***
    a. At least one “key health component” has an active fault condition that causes its performance to be degraded.
    b. Used whenever a node or service has one or more unresolved fault messages with a CRITICAL or ERROR severity.
    c. For aggregate status, orange overrides yellow (see the section titled “VDB Aggregate Health”).
3. ***YELLOW***
    a. At least one “key health component” has an active fault condition but its performance is not yet degraded.
    b. Used whenever a node or service has one or more unresolved fault messages with a WARNING severity.
    c. For aggregate status, yellow overrides green (see the section titled “VDB Aggregate Health”).
4. ***GREEN***
    a. No “key health component” has a fault condition.
    b. Used when no fault-status level (between WARNING and EMERGENCY) health messages are being generated by a node or service.

Additional details of the message severity levels associated with each of the health status colors can be found in the section titled “Syslog Protocol Data Parameters.”

14.3.5.2 VDB Aggregate Health

The overall health of the VDB can be defined as a “sum of its parts” and represented as a single health color (like a dashboard warning light). In other words, if every component and service comprising a VICTORY implementation has a green health color, then the VDB Aggregate Health is green as well. If even one component of the VDB is yellow, or orange, or red, then the VDB Aggregate Health is yellow or orange or red (respectively). The overall health color is always the same as the most severe health color of any one of all of the components of the VDB.

14.3.6 Health Publishing Interfaces

VICTORY components provide health information asynchronously to the VDB through a Health Publishing Interface. There are two protocols used for asynchronous health publishing for VICTORY components: Syslog and SNMP. Syslog is the primary method used by most VICTORY components, and it is used by all VICTORY services and most VICTORY hardware nodes. SNMP provides the same mechanism for infrastructure components, such as network switches and routers. The following sections describe the Health Publishing Interface based on each of these protocols.

14.3.6.1 Syslog-Based Data Transport for Health Publishing

Syslog-Based Health messages can be sent asynchronously to the VDB in one of two ways:

- Connectionless – Network efficient by using Syslog over UDP Multicast.
- Connection-Oriented – Highly reliable by using Syslog over a TCP Socket.

Refer to the section titled “Syslog Protocol” for additional details.

14.3.6.1.1 Syslog-Based Connectionless Health Publishing Specification <10023–20120725, Pro>

Connectionless Health Publishing requires that a VCL description of the health management interface for a service or node include the multicast group that must be subscribed to by a client wishing to sink the health data. It is not necessary that the service or node keep track of all clients that have subscribed since this will be handled by the multicast group subscriptions.

The behavior of faults and status messages inside the Connectionless Health Publishing Interface are very different from each other. Figure 11 shows the behavior of a fault inside the Syslog-Based Connectionless Health Publishing Specification.

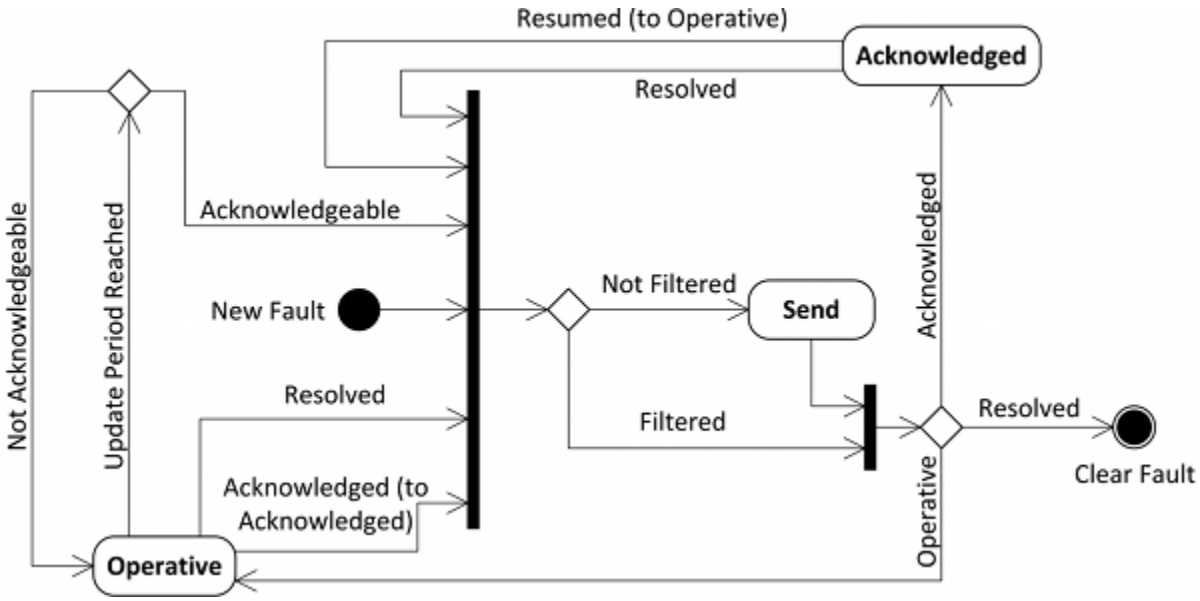

Figure 11. Fault Behavior for the Syslog-Based Connectionless Health Publishing Specification

A textual description of Figure 11 is provided below:

- When a new fault occurs, it has an initial state of OPERATIVE, is assigned a severity level, and can either be acknowledgeable or not. If the severity level of the fault is not being filtered (enabled), the fault will be encapsulated into a Syslog message and sent to the VDB one time to alert all subscribers that a new fault has occurred.
- While in an OPERATIVE state, a fault will proceed as follows:
  - If the severity level of the fault is not being filtered (enabled), acknowledgeable faults will be encapsulated into a Syslog message and sent to the VDB periodically (based on update period). This process will continue until one of the following occurs:
    - The fault state is set to ACKNOWLEDGED.
    - The fault state is set to RESOLVED.
    - The severity level of the fault is filtered out (disabled).
  - Faults that are not acknowledgeable do not send periodic Syslog messages to the VDB. Nothing happens until the state of the fault is set to RESOLVED.
- While in an ACKNOWLEDGED state, a fault will proceed as follows:
  - If the severity level of the fault is not being filtered (enabled), the fault will be encapsulated into a Syslog message and sent to the VDB one time to notify all subscribers that the state of the fault has been set to ACKNOWLEDGED.
  - If an acknowledged fault is resumed, its state will be set back to OPERATIVE.
    - If the severity level of the fault is not being filtered (enabled), the fault will be encapsulated into a Syslog message and sent to the VDB one time to notify all subscribers that the state of the fault has been set back to OPERATIVE.
- While in a RESOLVED state, a fault will proceed as follows:
  - If the severity level of the fault is not being filtered (enabled), the fault will be encapsulated into a Syslog message and sent to the VDB one time to notify all subscribers that the state of the fault has been set to RESOLVED.

- The fault will then be cleared from the Connectionless Health Publishing Interface.

Whereas fault publishing requires a complex series of steps, status publishing is much simpler. Figure 12 shows the behavior of status messages inside the Syslog-Based Connectionless Health Publishing Specification.

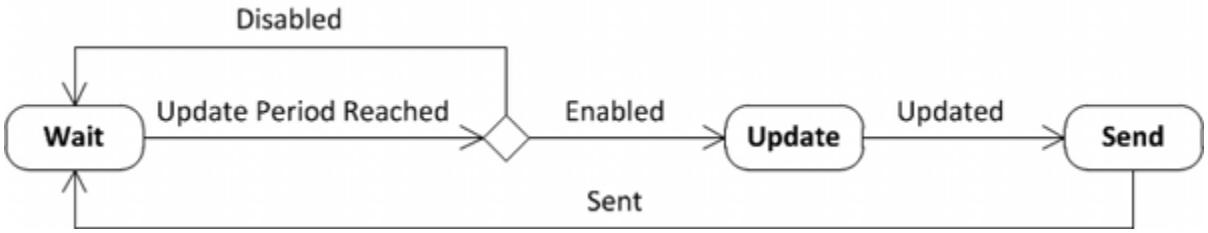

Figure 12. Status Behavior for the Syslog-Based Connectionless Health Publishing Specification

The status of the component from which a Connectionless Health Publishing Interface is sending status messages is determined by the most severe non-resolved fault that has been captured by the component. Unless disabled, the status of the component is periodically updated and sent to the VDB, similar to that of a heartbeat.

14.3.6.1.2 Syslog-Based Connection-Oriented Health Publishing Specification <Preliminary>

Connection-Oriented Health Publishing requires that a VCL description of the health management interface for a service or node include details of the service that can be notified of a client's desire to sink the health data. It is necessary for this second method that the service keeps track of all clients that have subscribed. The specific method that shall be used for subscription is TBD.

The behavior of faults and status messages inside a Connection-Oriented Health Publishing Interface is still TBD.

14.3.6.2 Syslog Protocol Data Parameters <10011–20210409, Pro>

The Syslog protocol defines a number of parameters that can be used to describe a fault or status message. A description of the Syslog parameters and how they relate to VICTORY are shown below.

1. ***PRI***
    a. This parameter indicates the priority of the message.
    b. Priority is derived from the Facility value and the Severity value (each term is defined by the Syslog protocol).
    c. For more information on the PRI parameter, refer to the Syslog protocol specification.
    d. The PRI parameter shall be included in a Syslog message for both faults and statuses.
    e. VICTORY uses Facility as follows:
        i. For sending health messages from an uncategorized VICTORY component, use facility numerical code 16 (defined as "local use 0" in the Syslog protocol).
        ii. For sending health messages from data interfaces, use facility numerical code 17 (defined as "local use 1" in the Syslog protocol).
        iii. For sending health messages from management interfaces, use facility numerical code 18 (defined as "local use 2" in the Syslog protocol).
        iv. For sending health messages from VICTORY services, use facility numerical code 19 (defined as "local use 3" in the Syslog protocol).
        v. For sending health messages from hardware nodes, use facility numerical code 20 (defined as "local use 4" in the Syslog protocol).
    f. VICTORY uses Severity as follows:
        i. For fault messages:
            1. Use severity numerical code 0 (Emergency) thru 4 (Warning).
            2. The severity numerical code represents the severity of the fault.
        ii. For status messages:
            1. Use severity numerical code 0 (Emergency) thru 4 (Warning) plus 6 (Informational).

2. The severity numerical code represents the most urgent (lowest) numerical code of all non-resolved faults being managed by the Health Publishing Interface.

2. ***VERSION***
    a. This parameter indicates the current version of the Syslog protocol specification.
    b. The VERSION parameter shall be included in a Syslog message for both faults and statuses.
    c. For more information on the VERSION parameter, refer to the Syslog protocol specification.
3. ***TIMESTAMP***
    a. This parameter indicates the date and time the fault occurred or the date and time the status of the interface, service, or component was monitored.
    b. If a fault is being periodically sent to the VDB, the timestamp of each periodic message shall be updated to the message send time in order to indicate that the fault is still occurring. The timestamp shall also reflect the message send time when a fault is acknowledged or resolved.
    c. The TIMESTAMP parameter shall be included in a Syslog message for both faults and statuses.
    d. For more information on the TIMESTAMP parameter, refer to the Syslog protocol specification.
4. ***HOSTNAME***
    a. This parameter indicates the machine that originated the Syslog message.
    b. The HOSTNAME parameter may be included in a Syslog message for both faults and statuses.
    c. For more information on the HOSTNAME parameter, refer to the Syslog protocol specification.
5. ***APP-NAME***
    a. This parameter indicates where the Syslog message originated.
    b. This parameter shall map to the unique name of the interface, service, or component.
    c. The APP-NAME shall not contain spaces. When sending a Syslog message, each space in a unique name shall be replaced with an underscore.
    d. The id and APP-NAME parameters shall be used for uniquely identifying a Syslog message. See the section titled "Maintaining Fault and Status Uniqueness" for more information on ensuring health message uniqueness.
    e. The APP-NAME parameter shall be included in a Syslog message for both faults and statuses even though it is declared as optional inside the Syslog protocol specification.
    f. For more information on the APP-NAME parameter, refer to the Syslog protocol specification.
6. ***PROCID***
    a. This parameter indicates there has been a discontinuity in Syslog reporting.
    b. The PROCID parameter may be included in a Syslog message for both faults and statuses.
    c. For more information on the PROCID parameter, refer to the Syslog protocol specification.
7. ***MSGID***
    a. This parameter indicates the category of the health message being sent to the VDB.
    b. For fault messages, the MSGID parameter shall be chosen from a list of known fault categories, including:
        i. CONFIGURATION_FAULT
        ii. DATA_FORMAT_FAULT
        iii. DATA_TRANSPORT_FAULT
        iv. HARDWARE_FAULT
        v. POWER_FAULT
        vi. SOFTWARE_FAULT
    c. For status messages, the MSGID parameter shall be chosen from a list of known status categories, including:
        i. STATUS
    d. The MSGID parameter shall be included in a Syslog message for both faults and statuses.
    e. For more information on the MSGID parameter, refer to the Syslog protocol specification.
8. ***SD-ID***
    a. This parameter indicates the unique structured-data identifier for VICTORY-specific Syslog parameters.
    b. The format of this parameter shall be healthParameters@<private enterprise number>
    c. Non-VICTORY IANA-registered SD-IDs are not required by VICTORY.
    d. The private enterprise number reserved for VICTORY is 41192.
        i. This private enterprise number shall be used for all VICTORY-related Syslog messages.
    e. The SD-ID parameter shall be included in a Syslog message for both faults and statuses.
    f. For more information on the SD-ID parameter, refer to the Syslog protocol specification.
9. ***SD-PARAM***
    a. This parameter indicates the name-value pairs used for the VICTORY-specific Syslog parameters.
    b. VICTORY is defining four additional parameters inside the Syslog messages.
        i. id
            1. A monotonically increasing number that indicates the identifier of the fault or status.

2. The id for faults shall increase by one (1) for every fault, and the id for statuses shall increase by one (1) for every status.
3. Example: id="100"
4. The id and APP-NAME parameters shall be used for uniquely identifying a Syslog message. See the section titled "Maintaining Fault and Status Uniqueness" for more information on ensuring health message uniqueness.
5. The id shall be an integer value greater than or equal to zero and less than or equal to 4,294,967,295.
6. When the id reaches its maximum value, it shall roll over to its minimum value.
7. Shall be included in a Syslog message for both faults and statuses.

ii. acknowledgeable
1. Indicates whether a fault can be acknowledged and therefore no longer periodically sent to the VDB. A message that will be periodically sent to the VDB must contain this parameter with a True value.
2. Value shall be a case-sensitive string representation of a Boolean where a value of True means the fault can be acknowledged and a value of False means the fault cannot be acknowledged.
3. Example: acknowledgeable="True"
4. Shall be included in a Syslog message for faults.
5. Shall not be included in a Syslog message for statuses.

iii. suggestedAction
1. Indicates the suggested action as a result of the fault.
2. Values include:
   a. SHUTDOWN
   b. MONITOR
   c. RECONFIGURE
   d. RESTART_HARDWARE
   e. RESTART_SOFTWARE
3. Example: suggestedAction="SHUTDOWN"
4. May be included in a Syslog message for faults. Not providing a suggested action for a Syslog fault message means no action is suggested as a result of the fault.
5. Shall not be included in a Syslog message for statuses.

iv. state
1. Indicates the current state of the fault.
2. The fault state parameter shall be chosen from a list of known fault states, including:
   a. OPERATIVE
   b. ACKNOWLEDGED
   c. RESOLVED
3. Example: state="RESOLVED"
4. Shall be included in a Syslog message for faults.
5. Shall not be included in a Syslog message for statuses.

c. For more information on the SD-PARAM parameter, refer to the Syslog protocol specification.

10. ***MSG***
a. This parameter indicates the actual message text of the Syslog message.
b. The bounds of the MSG are described in the section titled "String Bounds" under "VICTORY Data Types."
c. The MSG parameter shall be included in a Syslog message for both faults and statuses.
d. Refer to the Syslog protocol specification for any additional details on the MSG field.

14.3.6.3 Syslog Protocol Health Usage Examples <Informational>

14.3.6.3.1 Syslog Protocol Fault Message Example

The format of a typical Syslog message for a fault is as follows:

```
<PRI>VERSION TIMESTAMP HOSTNAME APP-NAME PROCID MSGID [SD-ID SD-PARAMS] MSG
```

Below is an example Syslog fault message for VICTORY.

```
<139>1 2012-02-18T22:14:15.000013Z - positionDataInterface - DATA_FORMAT_FAULT
[healthParameters@41192 id="100" acknowledgeable="True"
suggestedAction="RESTART_SOFTWARE" state="ACKNOWLEDGED"] Position Data Interface is
sending VDMs to the VDB in an incorrect format.
```

It can be broken down as follows:

- The value of the PRI parameter is 139, which indicates that the facility for this fault message is 17 (local use 1) and the severity for this fault message is 3 (Error). More specifically, this says a data interface has an error.
- The VERSION value is 1, which was retrieved from the most recent Syslog protocol specification document.
- The fault occurred on February 18, 2012 at 10:14:15.000013 PM Zulu time.
- The HOSTNAME was not provided, and thus was replaced by the dash character as required by the Syslog specification.
- The APP-NAME maps to the unique name of the interface, service, or component where the fault occurred.
  - In this case, a Position Data Interface sent out the error and the unique name of that Position Data Interface is positionDataInterface.
- The PROCID was not provided, and thus was replaced by the dash character as required by the Syslog specification.
- The MSGID maps to a fault category, which in this case was DATA_FORMAT_FAULT.
- The VICTORY-specific parameters are defined inside the [ ] brackets and provide the following information:
  - The IANA has reserved an SD-ID of 41192 for VICTORY.
  - The fault identifier is 100.
  - The fault is acknowledgeable.
  - The suggested action of this fault is to restart the software.
  - The fault has been acknowledged but not resolved.
- A free-form message is provided saying that the Position Data Interface is sending VDMs to the VDB in an incorrect format.

14.3.6.3.2 Syslog Protocol Status Message Example

The format of a typical Syslog message for a status is as follows:

```
<PRI>VERSION TIMESTAMP HOSTNAME APP-NAME PROCID MSGID [SD-ID SD-PARAMS] MSG
```

Below is an example Syslog status message for VICTORY.

```
<140>1 2012-02-18T22:14:15.000013Z - positionDataInterface - STATUS
[healthParameters@41192 id="100"] Position Data Interface has a health level of yellow.
```

It can be broken down as follows:

- The value of the PRI parameter is 140, which indicates that the facility for this status message is 17 (local use 1) and the most severe non-resolved fault captured by the Health Publishing Interface is 4 (Warning). See the details of the PRI parameter in the section titled “Syslog Protocol Data Parameters” for additional details.
- The VERSION value is 1, which was retrieved from the most recent Syslog protocol specification document.
- The status message was generated on February 18, 2012 at 10:14:15.000013 PM Zulu time.
- The HOSTNAME was not provided, and thus was replaced by the dash character as required by the Syslog specification.
- The APP-NAME maps to the unique name of the interface, service, or component to which the status pertains.
  - In this case, a Position Data Interface is sending out the status message, and the unique name of that Position Data Interface is positionDataInterface.
- The PROCID was not provided, and thus was replaced by the dash character as required by the Syslog specification.
- The MSGID maps to a status category, which is STATUS (the only acceptable value for status messages).
  - The MSGID parameter can be used for determining whether the message is a fault message or a status message.
- The VICTORY-specific parameters are defined inside the [ ] brackets and provide the following information:
  - The IANA has reserved an SD-ID of 41192 for VICTORY.
  - The status identifier is 100.
- A free-form message is provided saying that the Position Data Interface is sending out a status message that the component where the Health Publishing Interface is implemented has a current health level of yellow.

14.3.6.4 SNMP-Based Data Transport for Health Publishing

The SNMP-based data transport for Health Publishing is described in the section titled "Generic Infrastructure Component Health Status Specification."

14.3.7 Maintaining Fault and Status Uniqueness <Informational>

The fault identifier and origin shall be used for uniquely identifying a fault in the system. Similarly, the status identifier and origin shall be used for uniquely identifying a status in the system. Note that the fault/status identifier and origin are parameters defined by VICTORY. For information on how these parameters map to the Syslog parameters, see the section titled "Mapping Between Syslog and XML Health Parameters." Ensuring uniqueness is up to the implementer, but the following section provides a few examples as to how this could be accomplished.

NOTE: *In certain situations there may be a need to archive Syslog messages for long periods of time. In these situations, the identifier and origin may not be sufficient for maintaining message uniqueness as the identifier will eventually roll back to zero. To guarantee uniqueness, the message timestamp can be used as a supplement to the identifier and origin.*

14.3.7.1 Examples of Maintaining Fault and Status Uniqueness

The examples below refer to faults, but the same logic can be applied to status messages.

For the examples below, assume there is a service with a unique name of exampleService running a Position Data Interface and an Orientation Data Interface with unique names of examplePositionInterface and exampleOrientationInterface, respectively. Each data interface is providing health management. The faults being sent to the VDB could contain an origin of exampleService, examplePositionInterface, or exampleOrientationInterface, because the origin may be the unique name of an interface, service, or component.

In the case where the origin comes from the unique name of the service (exampleService), the fault identifier would need to be managed in a way that ensures fault uniqueness. One way to ensure fault uniqueness would be to have faults coming from the Position Data Interface (x) have identifiers in the range of $0 <= x <= y$ and faults coming from the Orientation Data Interface (z) have identifiers in the range of $z > y$. If both the Position Data Interface and the Orientation Data Interface started their fault identifier at zero, there would be no way of knowing whether a fault with an origin of exampleService and a fault identifier of zero came from the Position Data Interface or the Orientation Data Interface.

In the case where the origin comes from the unique name of the individual data interfaces (examplePositionInterface or exampleOrientationInterface), the fault identifier does not need to be unique within the two data interfaces. In this example, it is perfectly valid for both the Position Data Interface and the Orientation Data Interface to start their fault identifiers at zero, because a fault with an origin of examplePositionInterface and a fault identifier of zero is distinguishable from a fault with an origin of exampleOrientationInterface and a fault identifier of zero.

14.4 Notification Interfaces

One of the core management functions, Notify, is intended to provide a method of indicating that some sort of non-fault event has occurred on the VDB, such as a configuration change. The purpose of this section is to specify how asynchronous notification of events on the VDB will function.

14.4.1 Syslog-Based Asynchronous Notification Publishing Interface <10012–20120430, Exp>

The core technology that has been selected for asynchronous notification for VICTORY services and generic end nodes on the VDB is Syslog. This is the same protocol used for publishing fault and status health messages from VICTORY services and generic end nodes. The Syslog protocol was described previously in the section titled "Syslog Protocol." This section will focus primarily on how notification events are different from other health messages.

A Syslog message for notifications is essentially a Syslog message for statuses with a severity numerical code of 5: Notice. For more information on a Syslog message, see the section titled "Syslog-Based Connectionless Health Publishing Specification" and the example in the section titled "Syslog-Based Asynchronous Notification Message Example."

14.4.2 Syslog-Based Asynchronous Notification Message Example <Informational>

The format of a typical Syslog message for a notification is as follows:

```
<PRI>VERSION TIMESTAMP HOSTNAME APP-NAME PROCID - [SD-ID SD-PARAMS] MSG
```

Below is an example Syslog notification message for VICTORY.

```
<141>1 2012-02-18T22:14:15.000013Z - positionDataInterface - - [healthParameters@41192
id="100"] setDataEnabled=False
```

It can be broken down as follows:

- The value of the PRI parameter is 141, which indicates that the facility for this status message is 17 (local use 1) and the severity for this fault message is 5 (Notice). More specifically, this says a data interface is sending out a notification-specific status message.
- The VERSION value is 1, which was retrieved from the most recent Syslog protocol specification document.
- The notification message was generated on February 18, 2012 at 10:14:15.000013 PM Zulu time.
- The HOSTNAME was not provided, and thus was replaced by the dash character as required by the Syslog specification.
- The APP-NAME maps to the unique name of the interface, service, or component that the notification pertains to.
  - In this case, a Position Data Interface is sending out the status message, and the unique name of that Position Data Interface is positionDataInterface.
- The PROCID was not provided, and thus was replaced by the dash character as required by the Syslog specification.
- A notification message shall not provide a MSGID, so that was replaced by the dash character as required by the Syslog specification.
- The VICTORY-specific parameters are defined inside the [ ] brackets and provide the following information:
  - The IANA has reserved an SD-ID of 41192 for VICTORY.
  - The status identifier is 100.
- A free-form message is provided saying that the position data is no longer being sent to the VDB because it has been disabled.

14.5 Configuration Management Parameters for the VDB Management Service <10029–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the configuration management parameters for the VDM management service. A textual description of the configuration management parameters for the VDM management service is shown below.

1. ***Configuration***
    a. This parameter indicates the configuration of the VICTORY components on the VDB.
    b. The configuration shall be a valid VCL Instance Document (VCL ID) as defined in the section titled "VCL Schema."
        i. On a Set() operation:
            1. This parameter provides a VCL ID describing the desired configuration of the components on the VDB.
            2. The VDB Management Service shall configure itself by parsing the portion of the VCL ID as defined in the section titled "VDB Management Service Configuration Description."
            3. Optional Mutable, Configurable, or Volatile Configuration Item elements do not have to be included in a Set() Configuration request unless otherwise specified in the applicable Configuration Description specification. If an element is not included, it shall remain unchanged except in the event that not including an element represents the removal of its respective value, as specified in the applicable Configuration Description specification. If they are included, they shall not be ignored.
                a. Omitting an optional mutable, configurable, or volatile Configuration type from a Set() Configuration request shall not alter the values of the

Configuration type, i.e., the omitted Configuration type and all of its values are unchanged by the Set() Configuration request.

4. If a Configuration Item or Configuration Group element does not have a Configuration Type, i.e., neither it or any of its parent elements have a Configuration Type, and that element appears as a list or as optional inside the tree view of the applicable Configuration Description specification, it should not be modified from how it appears inside the Get() configuration response. This includes the existence of optional elements, the quantity of elements that are a list, or any attributes associated with the element. Any modifications made to this element inside a Set() Configuration request shall be ignored and the element shall not be changed inside the VCL ID.
5. A structural error in the config description for the VDB Management Service (e.g., duplicate config items where only one config item is allowed, missing required items, etc.) shall result in a GENERIC_FAULT error.
6. A string bounds error in the configuration description may result in a OUT_OF_RANGE error.
    a. A string bounds error occurs when the Configuration Item or Configuration Group string bounds requirements described in the section titled "String Bounds" under "VICTORY Data Types" are not satisfied.
7. If no structural or string bounds error occurs, the VDB Management Service shall parse the configuration descriptions for the remaining component types in the VCL ID that contain a Configuration SOAP Component and send a Set() Configuration request to the Configuration SOAP Component of the component type's Management Interface.
8. Any errors returned by the individual components and the errors encountered while configuring the VDB management service shall be aggregated and returned by the VDB Management Service.
    a. For each component defined in the VCL ID (including the VDB management service), if no errors are encountered, a single NO_ERROR error, along with the unique name of the component, shall be included in the aggregate error response.
    b. For each component defined in the VCL ID (including the VDB management service), all errors returned by the component shall be included in the aggregate error response along with the unique name of the component.

        NOTE: *Optional Mutable elements whose respective management parameter is returning a NOT_SUPPORTED error when being changed inside its applicable management interface should return a similar NOT_SUPPORTED error when modified inside a Set() Configuration request.*

        i. Each error shall contain one of the errors defined in the section titled "Management Interface Error Types."
    c. Regardless of whether or not errors are encountered, the VDB Management service shall continue processing the VCL ID.
    d. If the VDB Management Service encounters an error while performing a Set() Configuration request, the VDB Management Service shall include a GENERIC_FAULT error along with the unique name of the offensive component in the aggregate error response.
9. The VDB Management Service shall store the last known configuration for any component that does not implement the Configuration SOAP Component. A component type without a Configuration SOAP Component can be identified by checking the soapComponent configuration groups within the component type's configuration description.

ii. On a Get() operation:

1. This parameter generates a VCL ID describing the current configuration of the known components of the VDB.
2. The VDB Management Service shall send a Get() Configuration request to the Configuration SOAP Component for all components that implement the Configuration SOAP Component and aggregate the responses into one VCL ID.
3. Optional elements do not have to be returned inside the Get() Configuration response if they have no value. If a value exists (e.g., has been previously "set," or had an

implementation-dependent initial value), it shall be returned inside the Get() Configuration response.

> ***NOTE:*** *It is possible that parameters with Configuration Types of Configurable or Volatile may have been updated through a Set() Configuration request, but have not yet been applied since the component has not been restarted. In this case, the component should return the current values for these parameters, not the values that will take effect upon a restart.*

4. The VDB Management Service shall send the last known configuration for any component that does not implement the Configuration SOAP Component. A component type without a Configuration SOAP Component can be identified by checking the soapComponent configuration groups within the component type's configuration description.
5. If any component on the VDB returns an error in response to the Get() Configuration request from the VDB Management Service, the VDB Management Service shall aggregate all error responses from the component along with the unique name of the component.
6. If the VDB management service encounters an error while performing a Get() Configuration request on a component, it shall return the unique name of the component along with a GENERIC_FAULT error and shall be aggregated in the error response.

c. Operations include Get() and Set().

2. ***Configuration Version***
   a. This parameter identifies the version of the configuration used by the management interface.

   > ***NOTE:*** *The configuration version is an implementation-specific versioning of a VCL instance document.*

   b. The bounds of the configuration version are described in the section titled "String Bounds" under "VICTORY Data Types."
   c. Operations include Get().

> ***NOTE:*** *The section titled "Management Interface Error Types" provides additional details on the errors defined for the syslog-based health publishing management parameters along with a description of additional errors that were not specified but may be returned.*

14.6 Generic Component Status Interface

The purpose of this section is to specify the system and network status of the hardware and software components, including the health and performance of each such component.

14.6.1 Generic Component Status Interface Information Content and API

The purpose of this subsection is to specify the information content and API definition for the VICTORY component-level status interface. The VICTORY component-level status interface shall provide an overview of the hardware or software component status in the areas of health and performance (i.e., utilization).

14.6.1.1 Generic Component Health Levels for End Users <10014–20120430, Exp>

Health level "colors" describe how well a component is operating. As with the high-level VDB status interface, the primary intent is for this interface to serve not as a full diagnostic, but instead as an "indicator light."

Generic components of the VDB will use health level "colors" when presenting health information to an end user. Generic components are defined as:

- Generic End Nodes and Shared Processing Units
- Sensors and Network Adaptors
- Shared Data Services (such as position, orientation, and direction of travel)

The health level "colors" used by generic components are identical to those specified for the high-level VDB status interface, as defined in the section titled "Health Levels for End Users."

14.6.1.2 Generic End Node Component Performance Status Interface <10015–20110729, Exp>

The performance status describes the utilization of a component. As with the high-level VDB performance interface, the primary intent is for this interface to serve not as a full diagnostic, but instead as an "indicator light."

In order to supply the performance status, a GetUtilization SOAP API call will be added to each End-Node hardware component including:

- Shared Processing Units
- Sensors and Network Adaptors
- etc.

The GetUtilization API call will take the form:

- UtilizationCode GetUtilization()

The return value of this call (UtilizationCode) shall take on one of the following enumerated values:

- Red
  - The component is over utilized
  - Over a red utilization "break point"
- Orange
  - The component is nearly over utilized
  - Over an orange utilization "break point"
- Yellow
  - The component is highly utilized
  - Over a yellow utilization "break point"
- Green
  - The component has acceptable utilization
  - Under a yellow utilization "break point"

For each component, a set of "break points" will be configurable using the VICTORY Configuration Language. The method of encoding these "break points" is TBD.

14.6.1.3 Generic Infrastructure Component Health Status Specification <10016–20150331, Pro>

Health status describes how well an infrastructure component is operating. The Simple Network Management Protocol (SNMP) will be used to acquire the health and status of infrastructure components, such as:

- Network Switches
- Network Routers

In order to monitor health and status, clients and agents shall require support for the SNMP notification mechanism, often referred to as "traps" and "informs." This mechanism is typically implemented by adhering to conformance statements in the following MIBs defined in IETF RFC 3413:

1. SNMP-TARGET-MIB: contains objects for defining management targets. It consists of two tables and conformance/compliance statements.
2. SNMP-NOTIFICATION-MIB: contains objects for the remote configuration of the parameters used by an SNMP entity for the generation of notifications.

It should be noted that IETF RFC 3413 is one of the core requirements for support of the SNMP standard, as called out in the section titled "SNMP Core Specifications" and Table 207. Even so, some components, such as the switch component, are not required to implement all of SNMP, and do not have to necessarily follow the specification called out in "SNMP Core Specifications." The only requirement levied by this current specification is that all components that are required to implement traps provide a mechanism through which they can be configured either by following IETF RFC 3413, and the SNMP-NOTIFICATION-MIB in particular, or through another openly accessible and well documented interface (e.g., a web or command line interface).

In addition, the following notifications shall be supported:

1. Defined in IETF RFC 3418:
    a. authenticationFailure (OID: 1.3.6.1.6.3.1.1.5.5)
2. Defined in IETF RFC 2863:
    a. linkUp (OID: 1.3.6.1.6.3.1.1.5.4)
    b. linkDown (OID: 1.3.6.1.6.3.1.1.5.3)

For additional details on how to configure SNMP traps and informs through the use of the SNMP-defined mechanism described in IETF RFC 3413, see the section titled "Configuring SNMP Notifications Through SNMP."

14.6.1.4 Configuring SNMP Notifications Through SNMP <Informational>

The primary purpose of this section is not to add additional requirements, but rather to describe how to use one of the core SNMP capabilities.

Clients can configure the agents for trap and inform destinations using the following Object Identifiers (OIDs):

1. snmpTargetAddrTable: contains information about transport domains and addresses.
    a. snmpTargetAddrName
    b. snmpTargetAddrTDomain
    c. snmpTargetAddrTAddress
    d. snmpTargetAddrTagList
    e. snmpTargtAddrParams
    f. snmpTargetAddrStorageType
    g. snmpTargetAddrRowStatus
2. snmpTargetParamsTable: contains information about SNMP version and security information to be used when sending messages to particular transport domains and addresses.
    a. snmpTargetParamsName
    b. snmpTargetParamsMPModel
    c. snmpTargetParamsSecurityModel
    d. snmpTargetParamsSecurityName
    e. snmpTargetParamsSecurityLevel
    f. snmpTargetParamsStorageType
    g. snmpTargetParamsRowStatus
3. snmpNotifyTable: contains entries which select which entries in the snmpTargetAddrTable should be used for generating notifications, and the type of notifications to be generated.
    a. snmpNotifyName
    b. snmpNotifyTag
    c. snmpNotifyType
    d. snmpNotifyStorageType
    e. snmpNotifyRowStatus

The term *trap*, as it applies to SNMP, was first used to describe asynchronous notifications in the SNMPv1 standard. As of SNMPv2 the term has been superseded by the more general term *notification*. Even so, the term is still commonly used, and is even used in the SNMPv2 documentation to refer to a notification where no acknowledgment is sent from the client to the agent. The alternative to traps are *informs*, a type of SNMP notification in which a client is expected to send acknowledgment of the notification back to the agent. For the purposes of VICTORY, both SNMP traps and informs may be used.

For additional information, please refer to IETF RFC 3413, which provides an example for how to subscribe to and use SNMP traps and informs.

14.7 Security for Management Interfaces

The Transport Layer Security (TLS) protocol provides communications security over the VDB to components and end nodes. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery. TLS is a cryptographic protocol that provides security for communications over IP-based networks.

All VICTORY management interfaces that do not provide their own security protection may do so by using a combination of Hypertext Transfer Protocol (HTTP) with TLS. This level of security is typically indicated by the use of HTTPS in the protocol portion of the URI of the SOAP endpoint. A VICTORY Management Interface WSDL may be updated to indicate the use of this level of security protection by modifying the URI provided inside the "location" attribute of the **soap:address** element. When not included in the WSDL document, the HTTPS URI may be described in the Management Interface transport configurations portion of the VICTORY Configuration Language (VCL) file.

14.7.1 Web Services Security (WS-Security)

VICTORY adopts Web Services Security: SOAP Message Security as the base on which to build a security framework, but leaves when to encrypt, sign, etc., to the implementer and the platform designers. This WS-Security information can be described *a priori*, but doing so does not facilitate interoperability between various vendors. Therefore, VICTORY has adopted various additions to allow the WSDL document to hold interoperable WS-Security information.

14.7.1.1 Web Services Security: SOAP Message Security

This specification provides specific information on encryption, signing, and security tokens.

14.7.1.2 Web Services Policy 1.5 – Framework

This specification provides a framework to write general web service policies.

14.7.1.3 WS-SecurityPolicy 1.2

This specification provides the set of security policy assertions related to WS-Security to be used with WS-Policy.

14.7.1.4 Web Services Policy 1.5 – Attachment

This specification allows the policies created to secure parts of the WSDL. This allows the security information to be distributed to clients.

14.7.1.5 Web Services Framework <14018–20130315, Pro>

The following specifications shall be implemented to provide SOAP message security details in the WSDL.

1. Web Services Policy 1.5 - Framework, 04 September 2007, W3C Recommendation.
2. WS-SecurityPolicy 1.2, 25 April 2012, OASIS Standard incorporating Approved Errata 01.
3. Web Services Policy 1.5 - Attachment, 04 September 2007, W3C Recommendation.

14.7.1.6 Web Services Security Specification <14019–20130315, Pro>

The following specification shall be implemented to provide SOAP message security.

- Web Services Security: SOAP Message Security Version 1.1.1, 18 May 2012, OASIS Standard.

14.7.2 Data Confidentiality & Integrity for Management Interfaces <10018–20180215, Pro>

Transport Layer Security (TLS) provides mechanisms to protect sensitive data during electronic dissemination across the Internet and other insecure channels. TLS protects data by using a set of cryptographic algorithms (cipher suite) to ensure the confidentiality, integrity, and authenticity of exchanged data. TLS shall be used to secure the management interfaces to components and services accessible from the VDB. Details are described in Table 212 below.

While both clients and servers should be configured to support TLS 1.1 and TLS 1.2, VICTORY requires the use of TLS 1.1, at a minimum; the use of TLS 1.2 is optional. To determine the best way to implement TLS, refer to NIST SP 800–52 Revision 1.

14.7.2.1 Planning for the Future

According to NIST SP 800–52 Revision 1, plans to migrate to TLS 1.2 should have already been developed. As revisions to NIST SP 800–52 are released in the future, old versions of TLS may be obsoleted, new, more secure versions may need to be implemented, and future versions of VICTORY may update Table 212 accordingly.

Table 212. Transport Layer Security Specifications

Applicability	Protocol Version	RFC
Required	TLS 1.1	IETF RFC 4346
Optional	TLS 1.2	IETF RFC 5246

14.7.3 Data Authentication for Management Interfaces <10019–20150331, Pro>

Authentication at the data transport level ensures that the service and/or client are the intended parties in the information exchange. These mechanisms help secure underlying communication with services and are supplemental to the VICTORY Access Control Framework.

This specification complements "Data Confidentiality & Integrity for Management Interfaces" by identifying methods for authenticating one or both endpoints of the TLS connection.

1. Server authentication shall be certificate-based.
    - Clients must trust the service's TLS certificate to be sure it is communicating with a valid service.
2. Client authentication shall be **one or more** of the following:
    a. TLS Client Certificate-Based
        - During the TLS handshake, services can request a TLS Client Certificate from the client. This allows the service to accept only valid trusted clients.
    b. Username & Password-Based
        - Services can implement HTTP Basic Authentication as per IETF RFC 7235. This allows the service to accept only clients with pre-shared information.
        - TLS MUST be implemented to secure the transfer of plaintext credentials.

NOTE: *Management interfaces that provide their own security mechanisms (e.g., SNMPv3) shall be exempted from the requirements of this section.*

15. SECURITY EVENT HANDLING

15.1 Security Event Log Services

15.1.1 Background <Informational>

A log (or log file) is a record of the events that have occurred on the components and systems connected to the VICTORY Data Bus. Logs are composed of log entries, each of which contains information related to a specific event.

Many logs within vehicles contain records related to cybersecurity. These cybersecurity logs are generated by many sources, including:

- Host and network firewalls
- Antivirus software
- Intrusion detection and prevention systems
- Inline network encryptors
- Attribute based access control (ABAC) system services
- Infrastructure components such as switches and routers
- C4ISR/EW applications and operating systems hosted on a generic end node or a shared processing unit (SPU)
- Data at rest encryptor

Cybersecurity log management encompasses the process for generating, transmitting, storing, analyzing, and disposing of cybersecurity log data. Log management is essential to ensuring that cybersecurity records are stored in sufficient detail and for an appropriate period of time.

Log analysis is beneficial for identifying security incidents, policy violations, anomalous behavior, and operational problems. Logs are also useful when performing auditing and forensic analysis. They can help in supporting internal investigations, establishing baselines, and identifying operational trends and long-term problems.

Log generation and storage can be overly complicated if inconsistent log formats are used by the varied sources reporting events. To address the issues of inconsistent content and format, VICTORY has adapted the Syslog protocol, as it enables each log generator to use the same high-level format for its logs and the same basic mechanism for transferring its log entries to a Syslog server running internally or on another host.

Syslog provides a simple framework for log generation, storage, and transfer that any operating system (OS), security component, or application could use if designed to do so. Many log sources either use Syslog as their native logging format or offer features that allow their log formats to be converted to the Syslog format. An exact description of the use of Syslog, including the VICTORY information assurance (IA)-specific parameters, is found in the section below. Further information on Syslog can also be found in the section titled "Health Management."

Log management also involves protecting the confidentiality and integrity of logs. The most recent version of Syslog (IETF RFC 5424) states that all implementations of the Syslog specification must support Transport Layer Security (TLS) as described in IETF RFC 5425. However, it is only recommended that deployments of this VICTORY specification use the TLS-based transport. Implementing the VICTORY standards should assist in facilitating efficient and effective log management.

The Security Event Log Service (SELS) is a service which collects, processes, and stores Syslog messages that are formatted as specified in the sections titled "Security Event Log Format" and "Log Analysis Using PRI and SD-PARAM Values."

15.1.2 Security Event Log Data Specifications

To send event log data to the SELS or to retrieve event log data from the SELS, the client and the SELS, or the manager and the SELS, must use a common set of protocols and data formats as described in the sections below.

15.1.2.1 Security Event Log Format <10024–20210409, Exp>

It is recommended that all VICTORY component types and services adhere to this specification when generating and transferring security event logs.

The Syslog protocol defines a number of parameters that can be used to describe a security log message. A description of the Syslog parameters is described in IETF RFC 5424. The information provided below shows how Syslog is to be used in generating VICTORY security event logs.

1. ***PRI***
    a. This parameter indicates the priority of the message.
    b. Priority is derived from the Facility value and the Severity value (each term is defined by the Syslog protocol).
2. ***VERSION***
    a. This parameter indicates the current version of the Syslog protocol specification.
3. ***TIMESTAMP***
    a. This parameter indicates the date and time the event occurred.
    b. If an event is being periodically sent to the VDB, the timestamp of each periodic message shall be updated to indicate that the event is still occurring.
4. ***HOSTNAME***
    a. This parameter indicates the machine that originated the Syslog message.
5. ***APP-NAME***
    a. This parameter indicates the unique name of the interface, service, or component where the Syslog message originated.
    b. This parameter shall map to the unique name of the interface, service, or component found in the VCL file.
6. ***PROCID***
    a. This parameter indicates there has been a discontinuity in Syslog reporting.
7. ***MSGID***
    a. This parameter identifies the type of message being sent to the VDB.
8. ***SD-ID***
    a. This parameter indicates the unique structured-data identifier for the VICTORY IA-specific Syslog parameters.
    b. The format of this parameter shall be VICTORYIA@41192.
9. ***SD-PARAM***
    a. This parameter indicates the name-value pairs used for the VICTORY-specific Syslog parameters.
    b. VICTORY is defining four additional security log parameters inside the Syslog messages. The four additional parameters are: id, eventSource, eventID, and eventType.
        i. id
            1. A monotonically increasing number that indicates the identifier of the event.
            2. Example: id=“100”
            3. The id and APP-NAME parameters shall be used for uniquely identifying a Syslog message.
            4. The id shall be an integer value greater than or equal to zero and less than or equal to 4,294,967,295.
            5. When the id reaches its maximum value, it shall roll over to its minimum value.
        ii. eventSource
            1. A number which identifies the source of the event being captured in a Syslog message.
            2. Valid eventSource values are:
                a. 0 = network-based firewall
                b. 1 = host-based firewall
                c. 2 = host-based antivirus
                d. 3 = idps
                e. 4 = ine
                f. 5 = data at rest encryptor
                g. 6 = router
                h. 7 = switch
                i. 8 = operating system
                j. 9 = attribute store service
                k. 10 = policy store service
                l. 11 = policy enforcement service

m. 12 = policy decision service
n. 13 = c4isr/ew application
o. 14 = spu
p. 15 = other

3. Example: eventSource="0"

iii. eventID

1. A number indicating the type of event being logged.
2. Valid eventID values are:
   a. 0 = Unauthorized connection attempt
   b. 1 = Logon
   c. 2 = System, device, or application startup/shutdown
   d. 3 = Privileged use
   e. 4 = Policy change
   f. 5 = Configuration change
   g. 6 = Hardware failure
   h. 7 = Resource exhaustion
   i. 8 = Patch or change to system code, firmware, or application software (upgrade or downgrade)
   j. 9 = Failed login
   k. 10 = Failed login to privileged account
   l. 11 = Other
3. Example: eventID="0"

iv. eventType

1. A number used to further characterize an event. Ideally, this parameter is placed in a Syslog message by the message originator. In doing so, the Syslog client must keep track of logs it generates for a specific period of time so that the Syslog generator can determine whether or not the event has been seen before. It is possible that this field is populated by the Syslog server log if the originator is unable to populate the field.
2. When the Syslog client is unable to answer any of the three questions which make up the eventType, then the Syslog client must answer that question in the negative resulting in an eventType="0". The Syslog server may populate the eventType based upon additional information known to the server but not to the client.
3. Valid eventType values are:
   a. 0 = Event does not require immediate action and event is non-actionable and event has never been seen before.
   b. 1 = Event does not require immediate action and event is non-actionable and event has been seen before.
   c. 2 = Event does not require immediate action and event is actionable and event has never been seen before.
   d. 3 = Event does not require immediate action and event is actionable and event has been seen before.
   e. 4 = Event does require immediate action and event is actionable and event has never been seen before.
   f. 5 = Event does require immediate action and event is actionable and event has been seen before.
4. Example: eventType="3"

10. ***MSG***
    a. This parameter indicates the actual Syslog message.
    b. The bounds of the MSG are described in the section titled "String Bounds" under "VICTORY Data Types."

15.1.2.2 Security Event Log Transfer <10025–20180215, Exp>

IETF RFC 5424 is a fairly new (2009) standard. Being new, it requires the use of the latest version (1.2) of TLS (IETF RFC 5246). IETF RFC 5425 also requires the use of mutual authentication with both the client and the server using X.509 certificates. However, with VICTORY, TLS 1.1 is recommended and TLS 1.2 is optional. If using TLS, a server certificate is required, and either a client password or client certificate is required. A client certificate is optional if a client password is used. Refer to the section titled "Data Authentication for Management Interfaces" for more information. When there is a conflict between IETF RFCs 5424, 5425, 5426 and VICTORY, the VICTORY specifications shall override.

15.1.3 Security Event Log Management Specification <14027–20131031, Exp>

The security event log management specification configures and controls the Security Event Log Service (SELS). The SELS is a service which processes security event logs. This event log processing is then an enabler for security log analysis. It is the results of the analysis which may then trigger an incident response.

15.1.3.1 Core Technologies

The core technology selected for the management specification is SOAP 1.2, specified by the World Wide Web Consortium (W3C).

The core technology selected for the format of data being exchanged is XML 1.0, specified by the W3C.

15.1.3.2 Core Operations

These operations are used to implement the security event log management specification. The security event log management specification shall implement Get and Set functions that support the parameters as described in the following sections.

The following core operations are specified:

- Get – issued by a manager to the SELS to retrieve management data from the SELS.
- Set – issued by a manager to the SELS to set parameters of the SELS.

SELS management functions are based on a set of core operations:

- Get (parameter(s), value(s)): parameter(s) is an input; value(s) is an output.
- Set (parameter(s), value(s)): parameter(s) and value(s) are inputs.

15.1.3.3 Event Scoring Configuration and Control Parameters

Customizing the mapping of the Syslog priority <PRI>, eventID, eventSource, and eventType to their corresponding event priority score (EPS), event ID score (EIS), event source score (ESS), and event type score (ETS) provides very useful context to the log events. The mapping enables the scoring process to be tailored to the needs of a specific platform. See the section titled "Log Analysis Example" for additional information.

- ***Event Window ID (EWID)***

  EWID is the event window identification number of the window under consideration. Any additional configuration and control parameters fall under the current EWID until a new EWID is set.

  1. Set (EWID, n)
     a. n is an integer value greater than or equal to zero and less than or equal to 65535.
     b. EWID is the parameter being set to n.
  2. Get (EWID, n)
     a. EWID is the parameter being retrieved.
     b. n contains the value of the event window identification number (EWID).
     c. n is an integer value greater than or equal to zero and less than or equal to 65535.

- ***Event Window Total Score (EWTS)***
  1. Set (EWTS, m)
     a. m is an integer value greater than or equal to zero and less than or equal to 65535.
     b. EWTS is the parameter being set to m.

  Once the EWTS is set, the EWTS value becomes the upper limit for all of the other event scoring parameters.

  1. Get (EWTS, m)
     a. EWTS is the parameter being retrieved.

b. m contains the value of the event window total score (EWTS).
c. m is an integer value greater than or equal to zero and less than or equal to 65535.

- ***Event Priority Score (EPS)***
  1. Set (EPS, p, q)
     a. p represents the <PRI> value of the Syslog message.
     b. p is an integer value greater than or equal to zero and less than or equal to 191.
     c. q represents the event priority score (EPS) for a given value of p.
     d. q is an integer value greater than or equal to zero and less than or equal to EWTS.
     e. EPS is the parameter being set.
  2. Get (EPS, p, q)
     a. EPS is the parameter being retrieved.
     b. p represents the <PRI> value of the Syslog message.
     c. p is an integer value greater than or equal to zero and less than or equal to 191.
     d. q contains the event priority score (EPS) for the given value of p.
     e. q is an integer value greater than or equal to zero and less than or equal to EWTS.
- ***Event Source Score (ESS)***
  1. Set (ESS, r, s)
     a. r represents the eventSource which generated the Syslog message.
     b. r is an integer value greater than or equal to zero and less than or equal to 15.
     c. s represents the event source score (ESS) for a given value of r.
     d. s is an integer value greater than or equal to zero and less than or equal to EWTS.
     e. ESS is the parameter being set.
  2. Get (ESS, r, s)
     a. ESS is the parameter being retrieved.
     b. r represents the eventSource which generated the Syslog message.
     c. r is an integer value greater than or equal to zero and less than or equal to 15.
     d. s contains the event source score (ESS) for the given value of r.
     e. s is an integer value greater than or equal to zero and less than or equal to EWTS.
- ***Event ID Score (EIS)***
  1. Set (EIS, t, u)
     a. t represents the eventID which occurred.
     b. t is an integer value greater than or equal to zero and less than or equal to 11.
     c. u represents the event ID score (EIS) for a given value of t.
     d. u is an integer value greater than or equal to zero and less than or equal to EWTS.
     e. EIS is the parameter being set.
  2. Get (EIS, t, u)
     a. EIS is the parameter being retrieved.
     b. t represents the eventID which occurred.
     c. t is an integer value greater than or equal to zero and less than or equal to 11.
     d. u contains the event ID score (EIS) for the given value of t.
     e. u is an integer value greater than or equal to zero and less than or equal to EWTS.
- ***Event Type Score (ETS)***
  1. Set (ETS, v, w)
     a. v represents the eventType which occurred.
     b. v is an integer value greater than or equal to zero and less than or equal to 5.
     c. w represents the event type score (ETS) for a given value of v.
     d. w is an integer value greater than or equal to zero and less than or equal to EWTS.
     e. ETS is the parameter being set.
  2. Get (ETS, v, w)
     a. ETS is the parameter being retrieved.
     b. v represents the eventType which occurred.
     c. v is an integer value greater than or equal to zero and less than or equal to 5.
     d. w contains the event type score (ETS) for the given value of v.
     e. w is an integer value greater than or equal to zero and less than or equal to EWTS.

15.1.3.4 Event Log File Configuration and Control Parameters

A log file is defined as the collection of Syslog messages between the start of the collection process to when the collecting ceases. A "collector" gathers the collection of Syslog messages for further analysis. A log file is typically found at the log "collector." The start/stop of the collection is managed through the security event log management specification.

The security event log management specification shall have the following configurable parameters:

- ***Log Collection Start/Stop (LCS)***

  The management specification shall allow for Log Collection Start/Stop (LCS).

  1. Set (LCS, s)
     a. s is a string input whose value is either "start" or "stop."
     b. LCS is the parameter being set.

- ***Size of the Log File (SLF)***

  The management specification shall set limits on the Size of the Log File (SLF) and an optional warning message may be configured to inform a system manager that the file size is approaching its established limit.

  If utilized, the warning message shall be sent when 95% of the log file size has been used. The warning message shall be "95% of log file size has been used."

  1. Set (SLF, r, s)
     a. r represents the maximum size of the log file.
     b. r is an integer value greater than or equal to zero.
     c. The measure of r shall be in megabytes (MB).
     d. s is used to control the optional warning message.
     e. s is a string whose value is either "enable" or "disable."
     f. SLF is the parameter being set.
  2. Get (SLF, r, s)
     a. SLF is the parameter being retrieved.
     b. r represents the maximum size of the log file.
     c. r is an integer value greater than or equal to zero.
     d. The measure of r shall be in megabytes (MB).
     e. s is used to control the optional warning message.
     f. s is a string whose value is either "enable" or "disable."

- ***Action on Overflow (AOO)***

  The management specification shall allow configuration of a system response to a log file that has reached its maximum size. A choice from the following may be configured:

  1. Shutdown – turn off system and wait for administrator override.
  2. Ignore – continue without logging events.
  3. Overwrite – erase oldest log entries and continue logging.

  An optional warning message may also be configured to inform a system manager that a log file limit response has been received. The response message shall be "log file limit response received."

  1. Set (AOO, d, e)
     a. d represents the action to be performed on overflow.
     b. d is a string whose value is either "shutdown," "ignore," or "overwrite."
     c. e is used to control the optional limit response message.
     d. e is a string whose value is either "enable" or "disable."
     e. AOO is the parameter being set.

- ***Log File Size (LFS)***

  The management specification shall be able to request the following information from the event log system.

1. Get (LFS, f)
    a. LFS is the parameter being retrieved.
    b. f contains the value of the log file size.
    c. f is an integer value greater than or equal to zero.
    d. The measure of f shall be in megabytes (MB).

- ***Log File Retrieval (LR)***

  The following shall be available:

  1. Retrieve all Logs.
  2. Retrieve all Logs before a specified date.

  The management specification shall be able to request log files from the system.

  1. Get (LR, g, h)
      a. g contains the value of the type of log retrieval desired.
      b. g is a string whose value is either “all logs” or a valid date.
      c. If g is a date, it shall be in the format MMDDYYYY.
      d. h contains the output of the log retrieval request.

- ***Log File Deletion (LD)***

  The following options shall be available:

  1. Delete all Logs.
  2. Delete all Logs before a specified date.

  The management specification shall be able to delete log files.

  1. Set (LD, j)
      a. j represents the type of log deletion desired.
      b. j is a string whose value is either “all” or a valid date.
      c. If j is a date, all log files dated prior to, but not equal to, j will be deleted.

15.1.4 Syslog Protocol Security Event Message Example <Informational>

The format of a typical Syslog message for an event is as follows:

```
<PRI>VERSION TIMESTAMP HOSTNAME APP-NAME PROCID MSGID [SD-ID SD-PARAMS] MSG
```

Below is an example Syslog security event message for VICTORY.

```
<34>1 2012-02-18T22:14:15.000013Z - FIREWALL - - [VICTORYIA@41192 id="37"
eventSource="0" eventID="0" eventType="3"] Failed connection attempt from unrecognized
IP address to network-based firewall.
```

It can be broken down as follows:

- The value of the PRI parameter is 34, which indicates that the facility for this security event message is 4 (security/ authorization messages) and the severity for this event message is 2 (critical). It should be noted that not all security logs will necessarily be of type 4 (security/authorization messages), although some may be.
- The VERSION value is 1, which was retrieved from the most recent Syslog protocol specification document.
- The event occurred on February 18, 2012 at 10:14:15.000013 PM Zulu time.
- The HOSTNAME was not provided, and thus was replaced by the dash character as required by the Syslog specification.
- The APP-NAME maps to the unique name of the interface, service, or component where the event occurred.
    - In this case, the FIREWALL sent out the message.
- The PROCID and MSGID were not provided, and thus each was replaced by the dash character as required by the Syslog specification.
- The VICTORY-specific parameters are defined inside the [ ] brackets.

- The number 41192 has been reserved for VICTORY Syslog messages, and thus was used in this example.
- The VICTORY-specific parameters provide the following information:
  - The security event log message identifier is 37.
  - The security event came from the "network-based firewall" (eventSource="0").
  - The security event was an invalid connection attempt (eventID="0").
  - The security event does not require immediate action, the event is actionable, and the event has been seen before (eventType="3").

- A free-form message is provided saying that there was a failed connection attempt from an unrecognized IP address to the network-based firewall.

15.1.5 Security Event Log Storage and Protection <10026–20130430, Exp>

It is recommended that the security event logs be stored on the shared processing unit (SPU). The SPU also provides the capability to protect the log information by encryption.

15.1.6 Log Analysis Using PRI and SD-PARAM Values <10027–20130430, Exp>

Using the priority (PRI) and the SD-PARAM values, a very basic log analysis capability can be achieved. The goal of the analysis is to be able to respond to an event or series of events. The log analysis will produce an event window total score (EWTS) which is the sum of all the event scores (ES) given to each event within the event window.

The event score for any one particular event is the sum of the event priority score (EPS), event source score (ESS), event ID score (EIS), and event type score (ETS). The event score may be reset when a response is taken, when a set number of events has occurred, when the event window total score has reached a specified value, or when the event window time has expired. The EWTS is a running total calculated by the entity which collects the various log events. The log analysis parameters are defined as follows:

Event Window (ew) – An integer value representing the maximum amount of time of the window in seconds, the maximum number of events within the window, or the maximum event window total score (EWTS) before the event window score is set back to zero.

Event Priority Score (EPS) – An integer value assigned to each of the values of the Syslog priority field. Each Syslog priority value should have a corresponding priority score.

Event Source Score (ESS) – An integer value assigned to each of the 16 different eventSource values.

Event ID Score (EIS) – An integer assigned to each of the 12 different eventID values.

Event Type Score (ETS) – An integer value assigned to each of the six different eventType values.

Event Score (ES) – An integer value representing the sum of the event priority score, the event source score, the event ID score, and the event type score for any one event. $ES_{ew} = EPS_{ew} + ESS_{ew} + EIS_{ew} + ETS_{ew}$ for ew = 0, 1, 2 … ew–1.

Event Window Total Score (EWTS) – A cumulative integer value of all the event scores within an event window. It may include one or many events. $EWTS = ES_0 + ES_1 + ES_2 + \ldots + ES_{ew}$

The parameters which are used to calculate the event window total score (EWTS) will be configurable by the log management interface. It is expected that the configurable values will need to be tuned for a particular implementation of the specification.

15.1.7 Log Analysis Example <Informational>

The example that follows is for illustrative use only and is not part of the specification. The tables below were generated by a potential user of this specification based upon that person's knowledge of the VICTORY architecture and the various infrastructure components, C4ISR/EW applications, and information assurance components and services on the vehicle platform. This site-specific knowledge has led the implementer to specify the below listed values for log analysis.

In this example the implementer specified one hour (3600 seconds) as the event window. The implementer could have specified that a number of events determined the event window, but in this instance chose a time interval. After one hour the event window total score (EWTS) will be reset to zero.

The maximum value for the event window total score is 1,000. When a score of 1,000 is reached, the event window total score will be reset to zero and the event window will be set back to zero. This is true whether the event window was specified in time or in the number of events.

The event priority scores are found in Table 213. The implementer was only interested in events with severity levels of 0, 1, 2, and 3, which correspond to emergency, alert, critical, and error. The implementer was also only interested in facilities 0, 4, 10, 13, and 14, which correspond to kernel messages, security/authorization messages (4 and 10), log audit, and log alert. A priority score of 20 was given to each of those Syslog messages. Priority values not shown in Table 213 were disregarded and given an event priority score (EPS) of zero.

Table 213. Event Priority Scores (EPS)

Severity	Facility	Priority Value	Priority Score
0	0	0	20
1	0	1	20
2	0	2	20
3	0	3	20
0	4	32	20
1	4	33	20
2	4	34	20
3	4	35	20
0	10	80	20
1	10	81	20
2	10	82	20
3	10	83	20
0	13	104	20
1	13	105	20
2	13	106	20
3	13	107	20
0	14	112	20
1	14	113	20
2	14	114	20
3	14	115	20

Table 214 shows the scores the implementer gave each of the event sources. The INE was rated highest with a score of 10.

Table 214. Event Source Scores (ESS)

eventSource	Source	Score
0	network-based firewall	2
1	host-based firewall	4

eventSource	Source	Score
2	host-based antivirus	4
3	idps	6
4	ine	10
5	data at rest encryptor	2
6	router	2
7	switch	1
8	operating system	2
9	attribute store service	3
10	policy store service	3
11	policy enforcement service	3
12	policy decision service	3
13	c4isr/ew application	2
14	spu	2
15	other	0

Table 215 shows the scores given for each of the event IDs. A failed login to a privileged account received the highest score of five.

Table 215. Event ID Scores (EIS)

eventID	Event	Score
0	Unauthorized connection attempt	2
1	Logon	0
2	System, device, or application startup/shutdown	0
3	Privileged use	2
4	Policy change	2
5	Configuration change	0
6	Hardware failure	2
7	Resource exhaustion	2
8	Patch or change to system code, firmware, or application software (upgrade or downgrade)	0
9	Failed login	0
10	Failed login to privileged account	5
11	Other	0

Table 216 shows the scores given to each of the six event types, with event type 4 receiving the highest score of 10.

Table 216. Event Type Scores (ETS)

eventType	Type	Score
0	Event does not require immediate action and event is non-actionable and event has never been seen before	2
1	Event does not require immediate action and event is non-actionable and event has been seen before	0
2	Event does not require immediate action and event is actionable and event has never been seen before	5
3	Event does not require immediate action and event is actionable and event has been seen before	4
4	Event does require immediate action and event is actionable and event has never been seen before	10
5	Event does require immediate action and event is actionable and event has been seen before	7

After setting all of these initial values, each event is processed and given an event score by adding up the event priority score, event source score, event ID score, and event type score. If the value of the event window total score is less than the maximum event window total score, and the event has not exceeded the event window, then the next event is processed and its score is added to the score of the previous event. This process continues until the event window total score has reached its maximum value or the event window has expired.

15.2 Incident Response

15.2.1 Using VICTORY Management Interfaces for Incident Response <Informational>

The logging and log generation specifications (10024–10027) provide a basic mechanism to perform log analysis. The log analysis process can lead to the identification of critical cybersecurity events. Once those events are identified, an incident response process can be initiated.

Nothing in this informational specification should be construed as platform requirements or even specific recommendations for the vehicle platforms. This specification is an informational specification.

Cybersecurity events are unique and so, also, are the responses to those events. Because of their uniqueness, it would not be possible to develop a specification for the response to any particular cybersecurity event. The goal of this informational specification is to enable the use of the VICTORY management interfaces as part of their response to a cybersecurity event.

A cybersecurity incident refers to any event with the intent to destroy data, run destructive or intrusive programs, or otherwise compromise the confidentiality, integrity, or availability of the victim’s data, applications, or operating system. It can cause widespread damage and disruption, necessitating extensive recovery efforts.

15.2.1.1 Cyber Incident Handling Process

The primary objectives of the cyber incident handling process as described in the Chairman of the Joint Chiefs of Staff Manual (CJCSM) 6510.01B are to:

1. Maintain a robust detection capability to ensure all suspicious activity is detected and reported so that further analysis can take place to determine if it is a reportable cyber event or incident.
2. Ensure the timely reporting of cyber incidents through appropriate technical and operational channels in a way that promotes an accurate, meaningful, and comprehensive understanding of the cyber incident throughout its life cycle.
3. Effectively contain events and incidents and isolate information systems (ISs) to minimize any damage or impact to Army and DoD/Army information networks, ISs, data, and services.
4. Safely acquire and preserve the integrity of data required for cyber incident analysis to help determine the technical/operational impact, root cause(s), scope, and nature of the cyber event or incident.
5. Provide an effective and comprehensive response that includes the recovery of any affected ISs and the return to a fully functioning, secure, operational state for all services and ISs.

Incident response and recovery includes the response steps performed to prevent further damage, restore the integrity of affected ISs, and implement follow-up strategies to prevent the incident from happening again. Much of the information found in this informational specification is taken directly from CJCSM 6510.01B.

15.2.1.1.1 Containment

Containment consists of short-term, tactical actions to stop an intruder's access to a compromised IS, limit the extent of an intrusion, and prevent an intruder from causing further damage. The primary objectives for containment are to:

- Regain control of the ISs involved in order to further analyze the cyber incident and return the IS to normal operation.
- Deny an intruder access to prevent him or her from continuing the malicious activity and from affecting other ISs and data.

While an intruder has access to an IS, the IS cannot be properly analyzed or restored. Performing containment:

- Prevents an intruder from accessing or exfiltrating DoD/Army data or other information.
- Prevents an intruder from destroying valuable evidence and tampering with ISs while they are being analyzed.
- Prevents an intruder from using DoD/Army ISs to attack other ISs.

Containment provides a reasonable security solution until sufficient information has been collected to address the vulnerabilities exploited and the damage sustained.

- It should be noted that some containment actions can be taken during the preliminary response phase of the incident handling life cycle.
- More containment steps may be warranted following an in-depth analysis, which may identify more affected ISs or malicious activities.

Containment steps can be executed iteratively with the steps in the detection and analysis phase. Containment strategies are executed by the Combatant Commands, Services, Defense agencies, DoD/Army field activities, hereafter referred to as CC/S/A/FA, responsible for the maintenance and operation of the affected DoD/Army information networks or ISs. This could be a local system administrator or could be the component's Computer Network Defense Service Provider (CNDSP).

Who executes the strategies will depend on the incident type, affected component, and local policy and procedures. Containment strategies will vary based on the type of incident. Containment activities may include any of the following, or any other strategies determined by the affected CC/S/A/FAs.

- **Blocking:** Blocking is the use of network access controls at the perimeter or enclave boundary to prevent the attacker from connecting to other DoD/Army information networks, ISs, or DoD/Army data and services.

  Certain incidents require immediate blocks at the host and/or gateway level if the risk of further compromise, data exfiltration, and continued threat to the DoD/Army information networks outweighs the benefit of monitoring adversary Tactics, Techniques and Procedures (TTPs) for a more comprehensive countermeasure. Other incidents may elicit a block if the result reduces risk and exposure to potential threats. Block requests may include both inbound and outbound traffic. IP address and TCP port blocks are used to prevent the spread of compromise from an identified external IS.

  Blocking can be achieved through the VICTORY management interfaces for the network firewall, host-based firewall, router, and switch.

- **Network Isolation:** Network isolation involves the use of network access controls to logically segment the network and restrict access to the affected hosts. Isolation strategies can include the following:
  - Disconnect the IS from any local area network (LAN). This will help prevent further contamination of the affected information network and IS.
  - Disconnect IS from the Internet or any other public networks. This will help to prevent inbound access, outbound traffic, or data exfiltration.
  - Disconnect or Isolate the affected network host and/or segment from the rest of the network. This can help to prevent further contamination or containing malicious activity to an IS or logical network segment. This will allow attacked ISs to still function, but will not spread malicious activity to the rest of the infrastructure. In some cases, this may be relevant to monitor adversarial activity while limiting the adversary's ability to attack other ISs.

Network isolation can be achieved through the VICTORY management interfaces for the network firewall, host-based firewall, intrusion detection and prevention system, router, and switch.

- **IS, Server, or Service Shutdown:** Shutting down an IS, server, or service may help limit damage or prevent further access to the IS by the adversary. However, it will also affect the ability to acquire certain valuable data for the incident analysis. The decision to pursue this containment strategy should be weighed carefully.
  - IS Shutdown. If it is determined that allowing the IS to function will destroy data or applications on the IS, the IS should be shut down with the commander's approval as a containment measure.
  - Server Shutdown. If it is determined that a particular server, such as an e-mail or Web server, requires shutdown until problems can be eliminated or to contain the spread of malicious code, the specific server should be shut down. Be advised that in addition to destroying non-volatile data, shutting down a server may adversely affect multiple users and mission operations. This decision should be made in coordination with the relevant command authority.
  - Service Shutdown. If sufficient analysis has been performed to correctly limit the scope of an intrusion to specific services, these services can be disabled (especially if no patch is available). Be advised that in addition to potentially destroying non-volatile data, shutting down a service may adversely affect multiple users and mission operations. This decision should be made in coordination with the relevant command authority.

The VICTORY management interfaces for the data at rest encryptor, host-based firewall, network firewall, intrusion detection and prevention system, generic end node, shared processing unit, router, and switch can be used to turn an information system, server, service, or network device on or off.

15.2.1.1.2 Eradication

Eradication involves taking action that removes the cause of the incident from the IS/network. Eradication consists of the steps required to eliminate the root cause(s) of an intrusion. All threats and risks should be removed from DoD/Army information networks and ISs before returning them to service. If the threat is not removed, then an IS can be easily compromised or breached again.

The primary objectives for eradication are to:

- Ensure the removal of the cause(s) of the malicious activity and any associated files.
- Ensure the elimination of any access methods used by the intruder, including vulnerabilities, physical security problems, or human error.

One should execute eradication steps after the first round of containment actions occur and then interactively with any further analysis and containment activities. Sometimes, full eradication can only happen after long-term policy and configuration management changes are put into place. In that case, the threat should be mitigated to the extent possible before rebuilding and reconnecting any affected ISs. Some ISs, due to the nature of the incident, may not need any eradication steps, or the eradication may occur as part of the recovery activities when the infected IS is wiped or erased and rebuilt. Eradication strategies are executed by the CC/S/A/FA responsible for the maintenance and operation of the affected information networks or ISs; this could be a local system administrator or could be the component's CNDSP. Who executes the strategies will depend on the incident type, affected component, and local policy and procedures.

Specific eradication actions depend on the nature of the incident.

- Remove Malware – Quarantine, delete, replace, or restore the integrity of infected files. In most cases, this will require rebuilding the IS from trusted media. This may also involve updating antivirus signatures. *This can be achieved through the VICTORY management interface for the host-based antivirus.*
  - Under most conditions, once an IS is compromised, the integrity of that IS cannot be verified until it has been restored from trusted media. If an IS contains malware, keeping it in operation is not recommended unless the complete integrity of that IS can be once again verified, or the IS is left running and monitored closely as part of an ongoing law enforcement and counterintelligence (LE/CI) case. In the latter case, the decision to leave the IS running must be based on approvals by authorized DoD/Army personnel. *The VICTORY management interface for the signed data verification service can help to verify the integrity of application software binaries.*
- Remediate or Mitigate Vulnerability – Remove vulnerabilities by installing any new operating IS or application patches to vulnerable software to prevent exploitation. If the IS cannot be patched for technical or operational reasons, mitigate the vulnerability by updating IS configurations and defenses to protect or segment the affected host. If a patch is not available, apply workarounds or temporary mitigation strategies.

- Modify Access Controls – Update user and network access controls. For instance, remove compromised user or administrator accounts; modify network access controls (e.g., IDS/IPS, firewall, content filtering); update baseline configurations; and remove any other access mechanisms used by the adversary.

No system should be rebuilt until system data has been adequately preserved and the vulnerability has been mitigated. Systems suffering from severe cyber incidents may need to be rebuilt from trusted media and have up-to-date antivirus software loaded and configured in accordance with Security Technical Implementation Guides (STIGs) and warning and tactical directives/orders (e.g., WARNORDs, FRAGOs, TASKORDs, etc.) prior to connecting the IS to the VICTORY Data Bus.

15.2.1.1.3 Recovery

Recovery consists of the steps necessary to restore the integrity of affected ISs, return the affected data, ISs, and information networks to an operational state, and implement follow-up strategies to prevent the incident from happening again. The main objectives of recovery are to:

- Restore the integrity of the IS by rebuilding it from trusted media when necessary.
- Implement proactive and reactive defensive and protective measures to prevent similar incidents from occurring in the future.
- Ensure all data and ISs are operating in a normal fashion.
- Ensure the complete resolution and closure of the incident.

Data and ISs are fully restored when the necessary patches and fixes applicable to the incident have been installed. If an IS is compromised, the integrity of anything on that IS is suspect. The intruder could have changed the kernel, binaries, data files, running processes, or memory. The only way to be sure an IS is free of malicious code and back doors is to reinstall the trusted media and then install any security patches and upgrades. This includes both OS and application patches.

Also, as part of the recovery process, any containment activities completed may need to be removed. This can include removing any blocks that are no longer necessary, re-enabling services, and reconnecting ISs and information networks to the LAN or Internet.

Who executes the recovery strategies will depend on the incident type, affected component, and local policies and procedures. Depending on the nature of the incident, recovery actions can include, but are not limited to, the following:

- Rebuild from Trusted Media – Reinstall the OS and applications from a trusted backup, or from original distribution media.
- Verify System Data – Review IS data to ensure its integrity. Someone who knows what user or other data was on the IS should review the data to ensure it has not been changed. Alternatively, restore IS data from a trusted backup.
- Change System Passwords – Change all passwords on the IS and possibly on all ISs that have trust relationships with the victim IS.
- Improve Network and Host Security – This includes the following:
    - Increase host and network monitoring.
    - Enable maximum host logging, auditing, and accounting programs.
    - Disable unnecessary services.
    - Verify there are no weaknesses in configuration files for those services.
    - Install all the latest vendor security patches and upgrades if approved and tested.
    - Update firewall rule-sets.
    - Update boundary router ACLs.
    - Update IDS/IPS signatures.
    - Review any DoD/Army guidance, advisories, or bulletins to see if there are other recovery actions or security enhancements that can be enabled or installed.
    - Install new security mechanisms that may help protect ISs or detect malicious activity. This can include programs like file integrity checkers, URL checkers, etc.
    - Implement any needed mail filtering.

Once all recovery steps have been completed and ISs have been tested to ensure they are operating normally, reconnect any hosts or information networks that were disconnected.

16. VDB COMPONENT TYPES

The VDB implements the physical, data link, and network layers of the OSI network reference model, as well as application layer interfaces for management of the VDB and its components. Overall, the VDB data transport specification defines an IP- and Ethernet-based switched network. The VDB management interface specifications define network APIs for configuring, controlling, reporting status, and managing faults in the VDB.

Not every VDB component that makes up or attaches to the VDB is required to support every specification. This section calls out the set of specifications that are required for each VDB component type.

> ***NOTE:*** *These specifications describe the protocols and capabilities of "VDB Components." The terms VDB component, physical component, node, and device are used somewhat interchangeably in this section to indicate physical components such as switches, shared processing units, routers, etc. However, VICTORY specifications are not intended to define devices, but rather a set of specifications that can be attributed to "device types." When the terms "physical component," "node," and "device" are used, what is really meant is a "component type." An implementer may choose to implement a single physical device that supports the specifications of one or more VDB Components. For instance, a single box could implement the Switch, Network Time Reference, and Shared Processing Unit specifications, effectively acting as each of those VDB components in a single physical device.*

VDB Components can be logically classified into infrastructure nodes, information assurance nodes, end nodes, and non-networked nodes. The following sections are organized by these categories.

16.1 Infrastructure Component Types

16.1.1 Generic Infrastructure Component Management Interface <12006–20180215, Pro>

VICTORY-compliant generic infrastructure components, such as Ethernet switches and Internet Protocol (IP) routers, shall provide a configuration mechanism. The mechanism shall meet the following requirements:

1. Configuration mechanism shall be well documented:
    a. As a common standard, such as SNMP or NETCONF (with custom MIBs and schemas well-defined and readily available).
    b. Through a freely available manual.
2. Configuration mechanism shall provide a capability to configure the following infrastructure node features (with the caveat noted below):
    a. Infrastructure node management accounts (usernames and passwords).
    b. Infrastructure node management accessibility (enable/disable management methods, set component IP address, configure logging, etc.).
    c. Multicast features (IGMP/MLD, IGMP snooping/MLD snooping, static routes).
    d. QoS (DSCP-based queuing, queue weighting).
    e. SNMP traps/informs.
    f. Addressing and routing (enable/disable (R)STP, DNS/DHCP server).
3. Caveat: Some of the features listed here are not required for VICTORY compliance, although they may be recommended, optional, or required for a particular program. Even so, if these features are supported by an infrastructure component, then they shall be configurable through the provided management interface.

Allowed configuration mechanisms include, but are not limited to:

1. SNMP
2. NETCONF
3. Command Line Interface (accessible via serial port, telnet, or SSH)
4. Web Interface

VICTORY-compliant Ethernet switches and Internet Protocol routers shall provide a monitoring mechanism. The mechanism shall meet the following requirements:

1. Monitoring mechanism shall be well documented:
    a. As a common standard, such as SNMP or NETCONF (with custom MIBs and schemas well-defined and readily available).
    b. Through a freely available manual.
2. Monitoring mechanism shall provide access to network interface configuration and statistic information.
    a. Information provided shall include the following:
        i. A list of interfaces
        ii. Interface details, such as hardware address, configured speed, and MTU
        iii. Statistics for interfaces including packets/frames/bytes in, out, and dropped, with separate counts for unicast, multicast, and broadcast packets
    b. Note that if SNMP is used to implement the monitoring mechanism, then it is sufficient to support the interfaces MIB object tree (1.3.6.1.2.1.2) as described in IETF RFC 2863 and as specified by the RFC MIB Compliance Statement ***ifCompliance3***.
        i. This note is not intended to imply that either SNMP, or this particular MIB module, must be supported, but is simply an acknowledgement that analogous data is provided by the specified MIB module.
3. For IP routers only, the monitoring mechanism shall provide access to the IP implementation information.
    a. Information provided shall include the following:
        i. Interface address to IP address mappings
        ii. Per-interface status (interface *up* and *down*, at a minimum)
        iii. Per-interface and device-wide statistics, including packets in, out, and dropped for unicast, multicast, and broadcast traffic
        iv. The IP routing table
    b. This information shall be provided for IPv4 and, when supported by the device, IPv6.
    c. Note that if SNMP is used to implement the monitoring mechansim, then it is sufficient to support:
        i. The ip MIB object tree (1.3.6.1.2.1.4) as described in IETF RFC 4293 by the RFC MIB Compliance Statement ***ipMIBCompliance2***.
        ii. The ipForward MIB object tree (1.3.6.1.2.1.4.24) as described in IETF RFC 4292 by the RFC MIB Compliance Statement ***ipForwardReadOnlyCompliance***.

In all cases, configuration of a VICTORY-compliant Ethernet switch or IP router shall not require undocumented application programming interfaces (APIs) or specialized software, unless that software is provided freely to all U.S. Government Agencies and Supporting Contractors. Exclusion to non-U.S. entities may apply based on International Traffic in Arms Regulations (ITAR) controls.

NOTE: *VICTORY-Compliant Ethernet switch components are, by definition, managed switches. Non-managed switches are not prohibited from use in a VDB implementation, however, such switches will not be considered to be VICTORY-compliant.*

16.1.2 Switch

16.1.2.1 Switch Data Interface

The specifications for the data interface for this component are provided in the Switch Data Interface <VT50901-V1.9> component type table for IPv4, and in the Switch IPv6 Data Interface <VT50903-V1.9> component type table for IPv6.

16.1.2.2 Switch Management Interface

The specifications for the management interface for this component are provided in the Switch Management Interface <VT50902-V1.7> component type table.

16.1.2.3 Applicability of Internet Protocol Specifications to the Switch Component <Informational>

In the Open Systems Interconnection (OSI) model, layer 2 is described as the “data link layer.” By definition, it deals with the transmission of data between two nodes connected by a layer 1 (physical layer). Similarly layer 3, described as the “network” or “internetworking” layer, is defined as the layer that deals with multi-node networks, and includes features such as addressing and routing.

Although the layer separations described by the OSI model are not followed explicitly by contemporary networking stacks, it is not unfair to describe Ethernet as a layer 2 protocol and IP as a layer 3 protocol. Even so, it is not uncommon for a generally beneficial feature, such as traffic prioritization, to require exceptions to the clean separation described by the OSI model. Moreover, the modern managed Ethernet switch, which is minimally a layer 2 device working in the realm of Media Access Control (MAC) addresses, provides a number of features that the OSI model would relegate to layer 3. This has become so common that a new term has been coined for such a device: the layer 3 switch. In addition, even ostensibly layer 2 Ethernet switches which provide only bare Ethernet functionality often include management interfaces exposed through higher level protocols, such as HTTP over TCP/IP, or SNMP over UDP/IP, that require the component itself to appear on the network as an endpoint, and not just an invisible-to-layer–3 part of the infrastructure.

Such exceptions, when considered against the requirements of an Ethernet switch, can lead to a great deal of confusion. Therefore, the intention of this section is to provide some guidance and clarification with respect to the requirements levied by the VICTORY standards on Ethernet switch components of the VICTORY Data Bus. These clarifications are organized below, by applicable specification tag, technology, and support paradigm.

1. For the following specifications, and with the noted exceptions and caveats, Ethernet switch devices provide the necessary support by transporting Ethernet frames containing the specified content without necessarily needing to understand the data content of the frames.
    a. IP Layer Security (01025):
        i. Frame Content: Encrypted IP datagrams conforming to the Internet Protocol Security (IPsec) protocol.
        ii. Exception: Ethernet switch devices which provide a capability to secure management interface traffic using IPsec are expected to provide IP datagrams which conform to the relevant standard, and a mechanism by which IPsec settings can be configured.
        iii. Caveat: Some Ethernet switch devices have been known to exhibit poor performance and dropped frames when passing IPsec traffic. In particular, caution should be exercised when using IPsec on packets that are near the maximum transmission unit (MTU) of the switch device, as the addition of IPsec headers will likely cause fragmentation. Such fragmentation can put a higher reassembly burden on client systems. One solution to this problem is to use path MTU discovery to determine the MTU of the data path, and then ensure that no packets are large enough to be fragmented after the IPsec layer has been added.
    b. Transmission Control Protocol (01026):
        i. Frame Content: IP datagrams carrying the TCP stream packets.
        ii. Exception: Ethernet switch devices which provide a TCP/IP-based management interface through a protocol such as HTTP are expected to support sourcing and syncing of TCP streams as specified in the relevant standards.
    c. User Datagram Protocol (01027):
        i. Frame Content: IP datagrams carrying the UDP datagram packets.
        ii. Exception: Ethernet switch devices which provide a UDP/IP-based management interface through a protocol such as SNMP are expected to support sourcing and syncing of UDP datagrams as specified in the relevant standards.
        iii. Caveat: In some cases a managed Ethernet switch will not necessarily support certain types of UDP/IP routing, such as broadcast and multicast, without additional configuration.
2. For the following specification, Ethernet switch devices provide the necessary support by having some understanding of the content of the IP datagrams carried within the Ethernet frames, and responding accordingly.
    a. QoS Mapping to Components (01036):
        i. Frame Content: IP datagrams marked with differentiated services code point (DSCP) values, used to indicate predetermined content priorities and queuing models.
        ii. Expected functionality: A method exists for setting up traffic priority queues and models, where the type of traffic is indicated by DSCP values. The traffic is then sorted into these queues based on the DSCP values marked in the IP datagram headers.
3. For the following specifications, and with the noted exceptions, Ethernet switch devices provide the necessary support in a manner similar to an end node component, and to other IP network devices appear no different than such a component. These capabilities are related to the configuration endpoints that many Ethernet switch components provide on top of an IP-based stack (such as a Web interface or Simple Network Management Protocol [SNMP] interface).
    a. Static Host Addressing (01061):
        i. Expected functionality: A mechanism is provided through which the configuration endpoint within the Ethernet switch can be assigned a static IP address.
    b. IPv4 Private Address Space (01066):

i. Expected functionality: A mechanism is provided through which the configuration endpoint within the Ethernet switch can be assigned an IPv4 address and subnet within one of the private address space ranges defined by IETF RFC 1918.

c. IPv6 Private Address Space (01067):
   i. Expected functionality: A mechanism is provided through which the configuration endpoint within the Ethernet switch can be assigned an IPv6 address and subnet according to the unique local address (ULA) scheme defined by IETF RFC 4193.

d. VDB IPv4 Address Translation (01069):
   i. Expected functionality: A mechanism is provided for the sourcing and sinking of Address Resolution Protocol (ARP) messages through which the switch configuration endpoint can resolve IP-layer network endpoint addresses, and make its own IPv4 address available to other network nodes.

e. VDB IPv6 Address Translation (01070):
   i. Expected functionality: A mechanism is provided for the sourcing and sinking of Neighbor Discovery Protocol (NDP) messages through which the switch configuration endpoint can resolve IP-layer network endpoint addresses, and make its own IPv6 address available to other network nodes.

f. IPv4 Dynamic Host Addressing (01071):
   i. Expected functionality: A mechanism is provided through which the configuration endpoint within the Ethernet switch can be assigned a dynamic IPv4 address through the use of the Dynamic Host Configuration Protocol (DHCP) protocol (as a DHCP client).
   ii. Exception: There is no requirement that a VICTORY-compliant Ethernet switch function as a DHCP server, although many “layer 3” Ethernet switches can provide this capability. However, if such a capability is provided, it is expected to conform to the relevant specifications.

g. IPv6 Dynamic Host Addressing (01072):
   i. Expected functionality: A mechanism is provided through which the configuration endpoint within the Ethernet switch can be assigned a dynamic IPv6 address through the use of the DHCPv6 protocol (as a DHCP client).
   ii. Exception: There is no requirement that a VICTORY-compliant Ethernet switch function as a DHCPv6 server, although many “layer 3” Ethernet switches can provide this capability. However, if such a capability is provided, it is expected to conform to the relevant specifications.

16.1.3 Router

16.1.3.1 Router Data Interface

The specifications for the data interface for this component are provided in the Router Data Interface <VT51001-V1.9> component type table for IPv4, and in the Router IPv6 Data Interface <VT51003-V1.9> component type table for IPv6.

16.1.3.2 Router Management Interface

The specifications for the management interface for this component are provided in the Router Management Interface <VT51002-V1.7> component type table.

16.2 Information Assurance Component Types

16.2.1 Data Signing and Signed Data Verification Services

16.2.1.1 Background <Informational>

The VICTORY Data Signing and Signed Data Verification services are Web services that can be used to determine the origin and integrity of any type of data. For example, they can be used to verify software binaries, configuration files, databases, or any kind of digital data. These services are based on IETF Cryptographic Message Syntax (CMS), which in turn is based upon the Public-Key Cryptography Standards (PKCS) #7. They leverage the CMS signed-data content type.

This specification uses the CMS signed-data type and X.509 certificates to cryptographically bind signed data to the identity of the data publisher. The publisher may be the government, a vehicle integrator, and/or an application developer. The definition of a publisher is not constrained in any way by this specification.

This specification assumes knowledge of CMS, X.509 certificates, and Public Key Infrastructure (PKI). This specification assumes the use of a PKI, but does not discuss the issuing or processing of X.509 certificates, which are outside the scope of the data signing and signed data verification services. The interested reader is referred to IETF RFC 5652 for details regarding CMS.

The signature is in a CMS *SignedData* structure. The signature asserts that the file originates from a specific data publisher and the file has not been altered since it was signed. The signature itself does not convey any information about the intent or quality of the data.

A CMS *SignedData* structure contains the data file's hash value, a signature created with the data publisher's private key, and the X.509v3 certificate which bind the data file publisher's signing key to its identity.

16.2.1.2 VICTORY Uses of CMS <Informational>

CMS can be used to digitally sign, digest, authenticate, and/or encrypt arbitrary message content. This specification will only describe how to use CMS to digitally sign and verify data of any content type. Much of the syntax within CMS will not be used in this specification.

CMS can support a variety of architectures for certificate-based key management, such as the one defined by the Public Key Infrastructure using X.509 (PKIX) working group. CMS values are generated using ASN.1 (Abstract Syntax Notation) and BER-encoding (Basic Encoding Rules).

This specification does not require the implementation of any particular algorithms. Rather, the signature and message digest algorithms are found in the signer's public key certificate. Unlike CMS, these services allow only one signature per content type. Even though CMS is general enough to support many different content types, this specification defines one protection content, *ContentInfo*. *ContentInfo* encapsulates a single identified content type, and the identified type may provide further encapsulation. An implementation that conforms to this specification shall implement the protection content, *ContentInfo*, and shall implement the *signed-data* content type described in IETF RFC 5652.

16.2.1.2.1 Signed-data Content Type

The *signed-data* content type consists of content of any type and zero or one signature values. The VICTORY application of the *signed-data* content type represents one signer's digital signature on content of the *data* content type.

```
SignedData ::= SEQUENCE {
    version CMSVersion,
    digestAlgorithms DigestAlgorithmIdentifiers,
    encapContentInfo EncapsulatedContentInfo,
    certificates [0] IMPLICIT CertificateSet OPTIONAL,
    crls [1] IMPLICIT RevocationInfoChoices OPTIONAL,
    signerInfos SignerInfos }
```

16.2.1.3 Data Signing and Signed Data Verification Service Message Exchanges

16.2.1.3.1 Data Signing Service Management Specification <14025–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the data signing service management specification. Figure 13 describes the message exchange which would be carried out by an entity (Client) that wishes to sign data of any content type. The data signing service (Service) needs the data to sign, the signer's certificate, and the signer's private key.

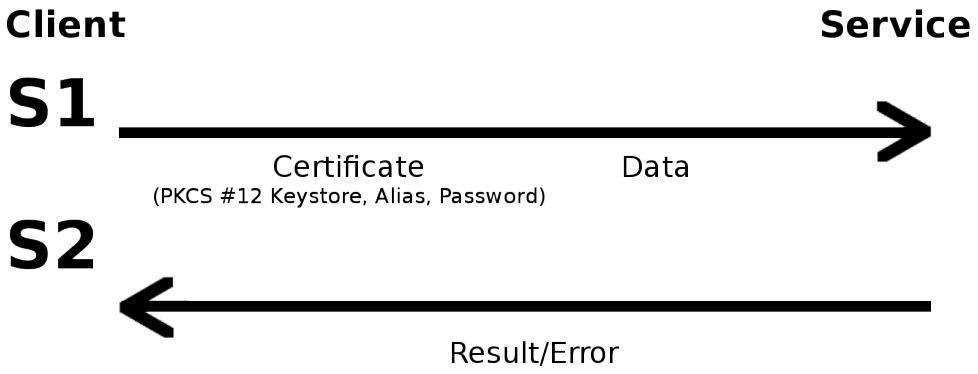

Figure 13. Data Signing Message Exchange

The **Certificate** contains the PKCS #12 archive (Base64 encoded), with the signer's X.509 certificate and private key stored inside, the alias for the key in the archive, and the password to access the archive. Refer to IETF RFC 4648: The Base16, Base32, and Base64 Data Encodings for more information on Base64 encoding.

- The bounds of the alias and password are described in the section titled "String Bounds" under "VICTORY Data Types."

NOTE: *Transport Layer Security (TLS) is required. Not using TLS would allow the password to be sent in plaintext across the network.*

The ***Data*** parameter is the actual data you wish to sign. The data is Base64 encoded.

The ***Result*** would be the actual ***signed-data*** content type as described in IETF RFC 5652. The signed data is Base64 encoded.

An ***Error*** is returned instead of the ***Result*** for the following issues:

- A GENERIC_FAULT error should be returned for the ERRNO in the error type.
- An OUT_OF_RANGE error may be returned if the alias and/or password is outside the range defined by this specification document.

The following cases define the ERRTXT in the error type:

1. Bad Keystore
    - This is returned when the PKCS #12 keystore is invalid or cannot be read.
2. Bad Password
    - This is returned when the password is incorrect for the keystore.
3. Bad Certificate
    - This is returned when the X.509 Certificate in the keystore is invalid or cannot be read.
4. Bad Alias
    - This is returned when the alias does not exist in the keystore.
5. Signing Failed
    - This is returned when the service could not sign the data.

16.2.1.3.2 Signed Data Verification Service Management Specification <14026–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the signed data verification service management specification. Figure 14 describes the messages that would be exchanged by an entity (Client) that wishes to verify previously signed data.

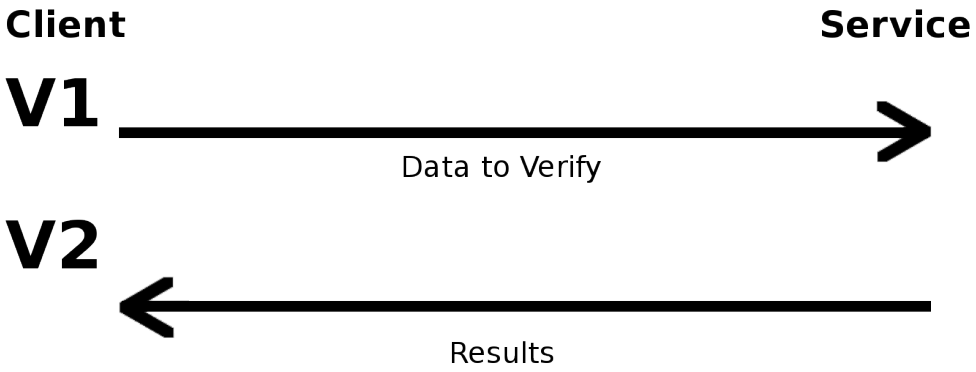

Figure 14. Signed Data Verification Message Exchange

The ***Data to Verify*** parameter is the actual data you wish to verify. The data conforms to the *signed-data* content type as described in IETF RFC 5652. It is Base64 encoded. Refer to IETF RFC 4648: The Base16, Base32, and Base64 Data Encodings for more information on Base64 encoding.

The ***Results*** parameter provides the specific results of the verification operation.

On a success, NO_ERROR should be returned for the ERRNO in the error type.

On a failure, a GENERIC_FAULT error should be returned for the ERRNO in the error type.

The following cases define the ERRTXT in the error type:

1. Bad Data
    - This is returned when the data is not CMS SignedData.
2. Verification Failed
    - This is returned when the verification did not succeed.

16.2.2 Data at Rest Encryptor

Encryption can secure data on storage devices when the devices leave the owners' control, such as when devices are repurposed, lost, or returned for repair. It can protect data from unauthorized discovery, lowering risks of data exposure while preserving retention requirements and assuring information security. Data at Rest (DAR) encryption also provides emergency data destruction without destroying equipment or losing data. The DAR Encryptor (DARE) may be either a hardware or software device.

16.2.2.1 DAR Encryption Storage Types <Informational>

If there is an expressed requirement or desire to use encryption to protect data at rest on the platform, it may occur within host-based software storage, direct attached storage, or network attached storage.

16.2.2.1.1 DAR Encryption Host-Based Software Storage

Host-based software storage is a program resident in memory on a particular device. It often stores its data in large container files on the host itself. These containers can be mounted on the system to act as a virtual drive that will encrypt everything sent to it.

16.2.2.1.2 DAR Encryption Direct Attached Storage

Direct Attached Storage refers to a hardware device that is directly attached to a data processor. A local hard drive is an example of direct attached storage.

16.2.2.1.3 DAR Encryption Network Attached Storage

Network Attached Storage refers to a broad spectrum of storage devices and storage networks. These are distinct from direct attached storage in that storage is performed remote to the data processor device. A network-enabled external hard drive is an example of network attached storage.

16.2.2.2 DAR Encryptor Management Specification <14016–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the DAR encryptor management specification. The DAR encryptor management specification shall implement functions that support the parameters as described in the following sections.

16.2.2.2.1 DAR Encryptor Configuration Parameters

1. ***Authenticator Type***
    a. The type of authentication used by the DARE instantiation.
    b. The authenticator type shall be chosen from a list of known authenticator types.
        - PASSWORD
        - CAC
        - PIN
        - STARTUP_TOKEN
    c. Operations include Get() and Set().
2. ***DAR Security Policy***
    a. The security policy is a human-readable file containing information about the policies in effect on the device.
    b. The security policy shall be binary encoded into Base64.
        i. Refer to IETF RFC 4648: The Base16, Base32, and Base64 Data Encodings for more information on Base64 encoding.
    c. Operations include Get() and Set().
3. ***Classification Level***
    a. The classification level for a particular DARE instantiation.
    b. The classification level shall be chosen from a list of known classification levels.
        - UNCLASSIFIED
        - CONFIDENTIAL
        - SECRET
        - TOPSECRET
    c. Operations include Get().
4. ***Crypto Type***
    a. The certification level for the DARE instantiation.
    b. The crypto type shall be chosen from a list of known crypto types.
        - NIST_CERTIFIED
        - NSA_APPROVED
        - UNEVALUATED
    c. Operations include Get().
5. ***Recovery Agent***
    a. Defines the entity that acts as the recovery agent.
    b. The recovery agent shall be in the format of a valid URI.
    c. The bounds of the recovery agent are described in the section titled "URI Bounds" under "VICTORY Data Types."
    d. The following error is defined for the recovery agent parameter:

i. Setting a recovery agent outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.

e. Operations include Get() and Set().

16.2.2.2.2 DAR Encryptor Control Parameters

1. ***DAREStatus***
    a. The current status of the DARE instantiation.
    b. The DARE status shall be chosen from a list of known statuses.
        - ON
        - OFF
    c. Operations include Get() and Set().
2. ***Zeroize***
    a. A message to zeroize the DARE instantiation.
    b. Operations include Set().

> ***NOTE:*** *Once the DARE instantiation has been zeroized, the behavior of the DAR Encryptor Management Interface is undefined.*

16.2.3 Host-Based Antivirus

A Host-Based Antivirus is a piece of software running on a single host that stops and removes malicious software on that host only. It is designed to protect the host, as well as the applications and data on it.

16.2.3.1 Host-Based Antivirus Features <Informational>

The following features have been identified as the minimal set for a host-based antivirus.

1. ***Defense against malware*** – Safeguards systems and files from viruses, spyware, worms, Trojans, and other security risks.
2. ***Detects and cleans malware***
3. ***Real-time scanning*** – An active 24/7 scan of the host system.
4. ***Guards against exploits that target vulnerabilities in applications***
5. ***Centralized management*** – A platform for deploying policies, managing security alerts, and viewing automated reports.

16.2.3.2 Host-Based Antivirus Management Specification <14023–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the host-based antivirus management specification. The host-based antivirus management specification shall implement functions that support the parameters as described in the following sections.

16.2.3.2.1 Host-Based Antivirus Configuration Parameters

The following configuration parameters have been identified as the minimal set for a host-based antivirus.

1. ***Virus Signatures***
    a. **setSigDBUpdate**
        i. Sends a binary file to update the virus signatures for the antivirus.
    b. **getSigDBVersion**
        i. Gets a string representing the current version of the virus signature database.
2. ***Program Updates***
    a. **setProgramUpdate**
        i. Sends a binary file to update the antivirus program.
    b. **getProgramVersion**
        i. Gets a string representing the current version of the antivirus program.

16.2.3.2.2 Host-Based Antivirus Control Parameters

The following control parameters have been identified as the minimal set for a host-based antivirus.

1. ***Virus Scanning***
    a. **setScan**
        i. Initiates a scan of the system.
    b. **getLastScanTime**
        i. Returns the last time a scan was run as a dateTime.
2. ***Logging***
    a. **getLog**
        i. Returns Logs as a Base64 binary file for the duration specified from the time given.
            1. Refer to IETF RFC 4648: The Base16, Base32, and Base64 Data Encodings for more information on Base64 encoding.
        ii. If time is not present, default to the current time.
        iii. If duration is not present, default to one (1) day.
        iv. If duration and time are not present, return all Logs.

16.2.4 Host-Based Firewall

A Host-Based Firewall performs functions similar to a Network Firewall, but has a smaller scope. A host-based firewall is a piece of software running on a single host that can restrict incoming and outgoing network activity for that host only.

16.2.4.1 Host-Based Firewall Features <Informational>

The following features have been identified for a host-based firewall.

1. ***Logging***

    All firewalls provide some capability for logging attacks for later review, allowing administrators to watch for anomalous activity. Logs would contain information about ports and affected packets, and are useful for forensics purposes.

2. ***Port Filtering***

    General port filtering allows for the enabling and/or disabling of TCP/UDP ports. This minimizes the attack surface of the host and provides generic protection.

3. ***Application Filtering***

    Application filtering allows specific applications and services to have targeted rules, which override the general port filtering settings.

16.2.4.2 Host-Based Firewall Management Specification <14020–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the host-based firewall management specification. The host-based firewall management specification shall implement functions that support the parameters as described in the following sections.

16.2.4.2.1 Host-Based Firewall Configuration Parameters

1. ***Port Logging***

    Logs based on the local/destination port(s) of the incoming packet or the remote/destination port(s) of the outgoing packet.

    The default logging status for ports without a rule configuration is OFF.

a. **addPortLoggingRule** – Adds a port logging rule.
    i. Request
        1. PortLoggingRule: Rules for setting log status for an inclusive range of ports.
            a. direction: {IN, OUT}.
            b. portRange: (0 – 65535 inclusive)
                i. portMin: Lowest port number.
                ii. portMax: Highest port number.
            c. status: {ON, OFF}.
        2. New rules overwrite existing values.
    ii. Response
        1. NO_ERROR
            a. Rule added successfully.
        2. GENERIC_FAULT
            a. portMin > portMax.
        3. NOT_SUPPORTED
            a. Logging by port is not supported.
b. **getPortLoggingRules** – Returns all the port ranges that have logging enabled (status = ON).
    i. portLoggingResponse:
        1. IN: 0 to *N* portRanges for enabled inbound logging.
        2. OUT: 0 to *N* portRanges for enabled outbound logging.
    ii. NOT_SUPPORTED
        1. Logging by port is not supported.
    iii. portRanges must be a minimal set. For example:
        1. portRanges must not overlap.
        2. portRanges must not have duplicates.
        3. portRanges must be a contiguous block with no adjacent ranges of the same status.

2. ***Port Filtering***

Allows/Blocks based on the local/destination port(s) of the incoming packet or the remote/destination port(s) of the outgoing packet.

a. **addPortFilteringRule** – Adds a port filtering rule.
    i. Request
        1. portFilteringRule: Rules for setting Block status for an inclusive range of ports.
            a. direction: {IN, OUT}.
            b. ipProtocol: {TCP, UDP, BOTH}.
            c. portRange: (0 – 65535 inclusive)
                i. portMin: Lowest port number.
                ii. portMax: Highest port number.
            d. action:
                i. ALLOW: Accepts the connection.
                ii. BLOCK: Blocks the connection.
                iii. MODE_DEFAULT: Action taken depends on "Port Filtering Mode Default."
        2. New rules overwrite existing values.
    ii. Response
        1. NO_ERROR
            a. Rule added successfully.
        2. GENERIC_FAULT
            a. portMin > portMax.
b. **getPortFilteringRules** – Returns all the ports that are being Allowed/Blocked explicitly.
    i. portFilteringResponse:
        1. IN
            a. TCP
                i. allowedPorts: 0 to *N* portRanges.
                ii. blockedPorts: 0 to *N* portRanges.
            b. UDP
                i. allowedPorts: 0 to *N* portRanges.

ii. blockedPorts: 0 to *N* portRanges.

2. OUT
   a. TCP
      i. allowedPorts: 0 to *N* portRanges.
      ii. blockedPorts: 0 to *N* portRanges.
   b. UDP
      i. allowedPorts: 0 to *N* portRanges.
      ii. blockedPorts: 0 to *N* portRanges.

ii. portRanges must be a minimal set. For example:
1. portRanges must not overlap.
2. portRanges must not have duplicates.
3. portRanges must be a contiguous block with no adjacent ranges of the same action.

3. ***Application Filtering***

Allows/Blocks based on the local/destination port(s) of the incoming packet or the remote/destination port(s) of the outgoing packet that is tied to a configured application.

a. **addAppFilteringRule** – Adds a rule to a new or existing application ruleset.
   i. Request:
      1. appID: Unique identifier for an application on the host.
         a. The path to the executable in Unix notation.
            i. Example: /c/program files/internet explorer/iexplore.exe or /usr/bin/firefox
         b. The bounds of the appID are described in the section titled "String Bounds" under "VICTORY Data Types."
      2. portFilteringRule: Rules for setting the filtering action for an inclusive range of ports.
         a. direction: {IN, OUT}.
         b. ipProtocol: {TCP, UDP, BOTH}.
         c. portRange: (0 – 65535 inclusive)
            i. portMin: Lowest port number.
            ii. portMax: Highest port number.
         d. action:
            i. ALLOW: Accepts the connection.
            ii. BLOCK: Blocks the connection.
            iii. MODE_DEFAULT: Action taken depends on "Application Filtering Mode Default."
      3. New rules overwrite existing values.
      4. Application Filtering Rules take precedence over Port Filtering Rules.
   ii. Response:
      1. NO_ERROR
         a. Rule added successfully.
      2. GENERIC_FAULT
         a. portMin > portMax.
      3. NOT_SUPPORTED
         a. Application Filtering is not supported.
      4. OUT_OF_RANGE
         a. May be returned if the appID is outside the range defined by this specification document.
b. **getAppIds** – Returns a collection of currently configured application IDs.
   i. Response:
      1. 0 to *N* appIDs.
         a. The bounds of each appID are described in the section titled "String Bounds" under "VICTORY Data Types."
      2. NOT_SUPPORTED
         a. Application Filtering is not supported.
c. **getAppFilteringRules** – Returns all the ports that are being Allowed/Blocked explicitly, by application ID.
   i. Request:
      1. 1 to *N* appIDs.

a. The bounds of each appID are described in the section titled "String Bounds" under "VICTORY Data Types."

ii. Response:

1. appFilteringRulesResponse: A collection of Application Filtering Rulesets.
    a. appFilteringResponse: (1 to *N*)
        i. appID
            1. The bounds of each appID are described in the section titled "String Bounds" under "VICTORY Data Types."
        ii. portFilteringResponse: (0 to 1)
            1. IN
                1. TCP
                    1. allowedPorts: 0 to *N* portRanges.
                    2. blockedPorts: 0 to *N* portRanges.
                2. UDP
                    1. allowedPorts: 0 to *N* portRanges.
                    2. blockedPorts: 0 to *N* portRanges.
            2. OUT
                1. TCP
                    1. allowedPorts: 0 to *N* portRanges.
                    2. blockedPorts: 0 to *N* portRanges.
                2. UDP
                    1. allowedPorts: 0 to *N* portRanges.
                    2. blockedPorts: 0 to *N* portRanges.
2. If an appID does not exist, the appFilteringResponse shall only contain the requested appID. portFilteringResponse must be null.
3. NOT_SUPPORTED
    a. Application Filtering is not supported.
4. portRanges must be a minimal set. For example:
    a. portRanges must not overlap.
    b. portRanges must not have duplicates.
    c. portRanges must be a contiguous block with no adjacent ranges of the same action.
5. OUT_OF_RANGE
    a. May be returned if one or more appIDs is outside the range defined by this specification document.

d. **removeAppFilteringRules** – Removes the application's appID and all associated rules.

i. Request:

1. 1 to *N* appIDs.
    a. The bounds of each appID are described in the section titled "String Bounds" under "VICTORY Data Types."

ii. Response:

1. NO_ERROR
    a. Rule(s) deleted successfully.
2. NOT_SUPPORTED
    a. Application Filtering is not supported.
3. OUT_OF_RANGE
    a. May be returned if one or more appIDs is outside the range defined by this specification document.

16.2.4.2.2 Host-Based Firewall Control Parameters

The Host-Based Firewall Management Interface shall accept and track rule manipulation and log collection even if the related functionality's status is OFF. For instance, if Port Filtering is disabled, configuration changes shall still be accepted and applied to the firewall when the Port Filtering functionality is re-enabled.

1. ***Logging Status***
    a. Sets the Logging capability within the Firewall as enabled or disabled.
    b. {ON, OFF}.

c. Set Response:
    i. NO_ERROR
        1. Logging capability was successfully toggled.
d. Operations include Get() and Set().

2. ***Port Filtering Status***
    a. Sets the Port Filtering capability within the Firewall as enabled or disabled.
    b. {ON, OFF}.
    c. Set Response:
        i. NO_ERROR
            1. Port filtering capability was successfully toggled.
    d. Operations include Get() and Set().
    e. Set Response:
        i. NO_ERROR
            1. Port filtering mode was set successfully.
3. ***Application Filtering Status***
    a. Sets the Application Filtering capability within the Firewall as enabled or disabled.
    b. {ON, OFF}.
    c. Set Response:
        i. NO_ERROR
            1. Application filtering capability was successfully toggled.
    d. Operations include Get() and Set().
    e. If application filtering is not supported, a NOT_SUPPORTED error shall be returned.
4. ***Application Filtering Mode Default***
    a. Sets the default behavior to use for Application Filtering.
    b. {ALLOW, BLOCK}.
    c. Set Response:
        i. NO_ERROR
            1. Application filtering mode was set successfully.
    d. Operations include Get() and Set().
    e. Affects both TCP and UDP configurations.
    f. If application filtering is not supported, a NOT_SUPPORTED error shall be returned.
5. ***Port Filtering Mode Default***
    a. Sets the default behavior to use for Port Filtering.
    b. {ALLOW, BLOCK}.
    c. Set Response:
        i. NO_ERROR
            1. Port filtering mode was successfully toggled.
    d. Operations include Get() and Set().
    e. Affects both TCP and UDP configurations.
6. ***Get Logs***

    Returns logged traffic related to the port logging rules (at the time they were logged) and may include other traffic log data, such as other protocol traffic with no notion of ports.

    a. Returns Logs as a Base64 binary file for the duration specified from the time given.
        i. Refer to IETF RFC 4648: The Base16, Base32, and Base64 Data Encodings for more information on Base64 encoding.
    b. If time is not present, default to the current time.
    c. If duration is not present, default to one (1) day.
    d. If duration and time are not present, return all Logs.

16.2.5 Intrusion Detection and Prevention System

The core functionality of an Intrusion Detection System (IDS) device is to detect unauthorized and/or malicious network traffic. This is accomplished through various methods including deep packet inspection, pattern matching, and attack signatures. An Intrusion Prevention System (IPS) goes further, taking action to protect a network once detection has occurred. As these systems have much in common, they are specified together as an Intrusion Detection and Prevention System (IDPS).

See also NIST Special Publication 800–94: Guide to Intrusion Detection and Prevention Systems (IDPS) for general information.

16.2.5.1 IDPS Management Specification <14006–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the IDPS management specification. The IDPS management specification shall implement functions that support the parameters as described in the following section.

16.2.5.1.1 IDPS Configuration Parameters

1. ***setRules***

   Send a ruleset to the IDPS.

   a. Request
      i. Base64 binary file that represents the ruleset.
         1. Refer to IETF RFC 4648: The Base16, Base32, and Base64 Data Encodings for more information on Base64 encoding.
         2. VICTORY does not define the ruleset format, so it may be vendor-specific.
         3. MIME Content type is optional, but may be used to identify the binary file.
   b. Response
      i. Management Interface Error Types
2. ***getRules***

   Get the current ruleset from the IDPS.

   a. Response
      i. Shall return one of the following:
         1. The GENERIC_FAULT error (for example, if the ruleset has not been set).
         2. Base64 binary file that represents the ruleset.
            a. VICTORY does not define the ruleset format, so it may be vendor-specific.
            b. MIME Content type is optional, but may be used to identify the binary file.

16.2.6 Network Firewall

The core functionality of the Network Firewall component is the control of IP traffic between different sub-networks within the larger IP-based network. Network firewalls allow access across boundaries to authorized traffic and block unauthorized traffic from entering or leaving certain areas of the network. As a general rule, firewalls are required to be present at the VDB boundaries with the larger extra-vehicular network, even though both may be within the same security domain.

A network firewall can be either software-based or hardware-based.

16.2.6.1 Network Firewall Management Specification <14003–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the network firewall management specification. The network firewall management specification shall implement functions that support the parameters as described in the following section.

16.2.6.1.1 Network Firewall Configuration Parameters

1. ***setRules***

   Send a ruleset to the Network Firewall.

   a. Request

i. Base64 binary file that represents the ruleset.
    1. Refer to IETF RFC 4648: The Base16, Base32, and Base64 Data Encodings for more information on Base64 encoding.
    2. VICTORY does not define the ruleset format, so it may be vendor-specific.
    3. MIME Content type is optional, but may be used to identify the binary file.

b. Response
    i. Management Interface Error Types

2. ***getRules***

Get the current ruleset from the Network Firewall.

a. Response
    i. Shall return one of the following:
        1. The GENERIC_FAULT error (for example, if the ruleset has not been set).
        2. Base64 binary file that represents the ruleset.
            a. VICTORY does not define the ruleset format, so it may be vendor-specific.
            b. MIME Content type is optional, but may be used to identify the binary file.

16.3 Access Control Component Types

16.3.1 Attribute Store Service

The Attribute Store Service is one of the four services that comprise the VICTORY Authorization Architecture (AA).

The Attribute Store Service is the repository for entity attribute information for the platform and stores attribute information, both static and dynamic, about entities. In order to support policy-based decisions, various attributes about the Principal need to be stored and retrieved. This attribute information is used to help the Policy Decision Service (PDS) determine whether access to a provider resource should be granted or not. The Attribute Store Service provides this standard access mechanism for accessing attributes. The request-response mechanism is based on the SAML Protocol.

Attribute Store Services serve as SAML Attribute Authorities and must be trusted by the services that use them. Attribute Store Services issue attributes as Attribute Statements contained in SAML Assertions. PDS implementations are required to trust only SAML Attribute Assertions issued by a well-known, trusted SAML Authority.

16.3.1.1 Attribute Store Management Specification <14015–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the attribute store management specification. Refer to the Core document for SAML 2.0 with errata for more information. The attribute store management specification shall implement functions that support the parameters as described in the following sections.

16.3.1.1.1 Attribute Store Configuration Parameters

1. ***addAttributes()***
    a. Uses a SAML Assertion with SAML AttributeStatements.
2. ***getAttributes()***
    a. Uses a SAML AttributeQuery element.
    b. Refer to Section 6: Assertion Query/Request Profile in "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0 – Errata Composite," Working Draft 06, 1 December 2009, for information on how to set up a SAML <AttributeQuery> and <Response> message.
3. ***getSubjects()***
    a. This parameter returns a list of Subjects for which the Attribute Store contains data.
    b. Returns a list of SAML Subject elements.
4. ***removeAttributes()***
    a. Uses a SAML AttributeQueryType to specify which attributes to delete.

16.3.1.1.2 Entity Attributes

The following minimal listing of entity attributes should be available upon request from the Attribute Store Service:

- Globally Unique Identifier (GUID)
- First Name
- Last Name
- Middle Name
- Role
- Clearance
- Rank

NOTE: *In order to lessen integration effort and maximize the portability of security policies, it is desired that entity attribute formats and values be standardized. The Department of the Army has developed a reference architecture and guidance for deploying ABAC systems. (See "U.S. Army – Identity, Credential, and Access Management (ICAM) - Reference Architecture (RA) v1.0," Version 1.0, 19 October 2012.) Implementers should reference this document to determine the applicability of the attribute values and formats contained in the Enterprise Identity Attribute Data Repository and Synchronization Service (EIADRSS).*

16.3.1.1.3 Additional Schema Restrictions

The SAML specification that these messages are based on includes a wide range of use-cases, many of which do not apply to VICTORY. The following are parts of the message that should not be used and additional information on various SAML parts.

1. ***addAttributes()***
    a. Ignore
        i. SAML Assertion
            1. Subject
                a. SubjectConfirmation
                    - This is mainly used for SAML Single Sign-On (SSO), which is not the case here.
            2. Advice
                - The SAML Spec states that, "this information MAY be ignored by applications without affecting either the semantics or the validity of the assertion." VICTORY will always ignore it.
            3. Condition
                a. OneTimeUse
                    - The SAML Spec states that, "relying parties may choose to retain assertions or the information they contain in some other form, for reuse." This does not fit the context of the Attribute Store for adding attributes.
                b. ProxyRestriction
                    - The SAML Spec states that ProxyRestriction, "specifies limitations that the asserting party imposes on relying parties that in turn wish to act as asserting parties and issue subsequent assertions of their own on the basis of the information contained in the original assertion." This does not fit the context of the Attribute Store for adding attributes.
                c. AudienceRestriction
                    - The SAML Spec states that, "the <AudienceRestriction> element specifies that the assertion is addressed to one or more specific audiences identified by <Audience> elements. Although a SAML relying party that is outside the audiences specified is capable of drawing conclusions from an assertion, the SAML asserting party explicitly makes no representation as to accuracy or trustworthiness to such a party." This would be used in cases where the assertions

could be broadcasted to different services. It does not fit with our connection-based Attribute Store.

4. Statement
   - This is an extension point for new Statements. Attribute Store does not use it.
5. AuthnStatement
   - This is for statements about SAML Authentication. Attribute Store does not use it.
6. AuthzDecisionStatement
   - This is for statements about SAML Authorization Decisions. Attribute Store does not use it.

b. Adapt

i. All Attribute Name Formats identifiers defined in Section 8.2 of [SAMLCore] must be implemented.
ii. All Name Identifier Formats identifiers defined in Section 8.3 of [SAMLCore] must be implemented.
iii. If using Encrypted Attribute or Encrypted ID, it must be encrypted with the public key of the Attribute Store.
iv. When the Issuer element is used in the Response, the value of this element must correspond to the unique name of the Attribute Store Service.

2. ***getAttributes()***

a. Ignore

i. SAML AttributeQuery
   1. Subject
      a. SubjectConfirmation
         - This is mainly used for SAML SSO, which is not the case here.

ii. SAML Response
   1. Assertion
      a. Subject
         i. SubjectConfirmation
            - This is mainly used for SAML SSO, which is not the case here.
      b. Advice
         - The SAML Spec states that, “this information MAY be ignored by applications without affecting either the semantics or the validity of the assertion.” VICTORY will always ignore it.
      c. Condition
         i. ProxyRestriction
            - The SAML Spec states that ProxyRestriction, “specifies limitations that the asserting party imposes on relying parties that in turn wish to act as asserting parties and issue subsequent assertions of their own on the basis of the information contained in the original assertion.” This does not fit the context of the Attribute Store for getting attributes.
         ii. AudienceRestriction
            - The SAML Spec states that, “the <AudienceRestriction> element specifies that the assertion is addressed to one or more specific audiences identified by <Audience> elements. Although a SAML relying party that is outside the audiences specified is capable of drawing conclusions from an assertion, the SAML asserting party explicitly makes no representation as to accuracy or trustworthiness to such a party.” This would be used in cases where the assertions could be broadcasted to different services. It does not fit with our connection-based Attribute Store.

b. Adapt

i. All Attribute Name Formats identifiers defined in Section 8.2 of [SAMLCore] must be implemented.
ii. All Name Identifier Formats identifiers defined in Section 8.3 of [SAMLCore] must be implemented.
iii. If using Encrypted IDs, they must be encrypted with the public key of the Attribute Store.
iv. Response should return encrypted Assertions if Encrypted IDs are provided.

v. When the Issuer element is used in the Response, the value of this element must correspond to the unique name of the Attribute Store Service.

3. ***getSubjects()***
    a. Ignore
        i. Subject
            1. SubjectConfirmation
                - This is mainly used for SAML SSO, which is not the case here.
    b. Adapt
        i. All Attribute Name Formats identifiers defined in Section 8.2 of [SAMLCore] must be implemented.
        ii. All Name Identifier Formats identifiers defined in Section 8.3 of [SAMLCore] must be implemented.
        iii. If using Encrypted IDs, they must be encrypted with the public key of the Attribute Store.
        iv. Response should return encrypted Assertions if Encrypted IDs are provided.
4. ***removeAttributes()***
    a. Ignore
        i. SAML AttributeQuery
            1. Subject
                a. SubjectConfirmation
                    - This is mainly used for SAML SSO, which is not the case here.
            2. Consent
                - Not useful for the remove function.
    b. Adapt
        i. All Attribute Name Formats identifiers defined in Section 8.2 of [SAMLCore] must be implemented.
        ii. All Name Identifier Formats identifiers defined in Section 8.3 of [SAMLCore] must be implemented.
        iii. If using Encrypted IDs, they must be encrypted with the public key of the Attribute Store.
        iv. When the Issuer element is used in the Response, the value of this element must correspond to the unique name of the Attribute Store Service.

16.3.2 Policy Decision Service

The Policy Decision Service (PDS) is one of the four services that comprise the VICTORY Authorization Architecture.

The Policy Decision Service makes authorization decisions based upon entity attributes and system policies and returns those decisions to the Policy Enforcement Service (PES). The PES provides the Principal's credentials, the Resource that the Principal is trying to access, and the action that is being taken. Entity attributes are received from the Attribute Store Service, while system policies are received from the Policy Store Service. Based on the information received, the PDS computes a decision and sends it back to the PES.

The PDS serves as a SAML Authorization authority and must be trusted by the services that use it. The PDS issues an Authorization Decision Statement contained within a SAML Assertion. The Assertion is returned to the PES via a SAML Response message.

16.3.2.1 Policy Decision Management Specification <14014–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the policy decision management specification. Refer to the Core document for SAML 2.0 with errata for more information. The policy decision management specification shall implement functions that support the parameters as described in the following sections.

16.3.2.1.1 Policy Decision Configuration Parameters

1. ***getAuthorization()***
    a. Uses a SAML AuthzDecisionQuery element.
    b. Refer to Section 6: Assertion Query/Request Profile in "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0 – Errata Composite," Working Draft 06, 1 December 2009, for information on how to set up a SAML <AuthzDecisionQuery> and <Response> message.
2. ***Attribute Store Service (AS) Location***
    a. Location of AS to contact.
    b. The AS location shall be in the format of a valid URI.
    c. The bounds of the AS location are described in the section titled "URI Bounds" under "VICTORY Data Types."

d. The following error is defined for the AS location parameter:
    i. Setting an AS location outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.
e. Operations include getAS() and setAS().

3. ***Policy Store Service (PS) Location***
    a. Location of PS to contact.
    b. The PS location shall be in the format of a valid URI.
    c. The bounds of the PS location are described in the section titled "URI Bounds" under "VICTORY Data Types."
    d. The following error is defined for the PS location parameter:
        i. Setting a PS location outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.
    e. Operations include getPS() and setPS().

16.3.2.1.2 Additional Schema Restrictions

The SAML specification that these messages are based on includes a wide range of use-cases, many of which do not apply to VICTORY. The following are parts of the message that should not be used and additional information on various SAML parts.

1. ***getAuthorization()***
    a. SAML AuthzDecisionQuery/Response
        i. Ignore
            1. SAML Evidence
                a. AssertionIDRef
                    - Not used.
                b. AssertionURIRef
                    - Not used.
            2. Subject
                a. SubjectConfirmation
                    - This is mainly used for SAML Single Sign-On (SSO), which is not the case here.
            3. Advice
                - The SAML Spec states that, "this information MAY be ignored by applications without affecting either the semantics or the validity of the assertion." VICTORY will always ignore it.
            4. Condition
                a. ProxyRestriction
                    - The SAML Spec states that ProxyRestriction, "specifies limitations that the asserting party imposes on relying parties that in turn wish to act as asserting parties and issue subsequent assertions of their own on the basis of the information contained in the original assertion." Since the assertions are not shared widely but queried, this was deemed unnecessary.
                b. AudienceRestriction
                    - The SAML Spec states that, "the <AudienceRestriction> element specifies that the assertion is addressed to one or more specific audiences identified by <Audience> elements. Although a SAML relying party that is outside the audiences specified is capable of drawing conclusions from an assertion, the SAML asserting party explicitly makes no representation as to accuracy or trustworthiness to such a party." This would be used in cases where the assertions could be broadcasted to different services. It does not fit with our connection-based Policy Decision Service.
            5. Statement
                - This is an extension point for new Statements. Policy Decision does not use it.
            6. AttributeStatement
                - This is for statements about SAML Attributes. Policy Decision does not use it.
            7. AuthnStatement

- This is for statements about SAML Authentication. Policy Decision does not use it.

ii. Adapt

1. There must be at least one AuthzDecisionStatement per different decision needed.
2. All Name Identifier Formats identifiers defined in Section 8.3 of [SAMLCore] must be implemented.
3. If using Encrypted IDs, they must be encrypted with the public key of the Policy Decision Service.
4. Response should return encrypted Assertions if Encrypted IDs are provided.
5. When the Issuer element is used in the Response, the value of this element must correspond to the unique name of the Policy Decision Service.

16.3.3 Policy Enforcement Service

The Policy Enforcement Service (PES) is one of the four services that comprise the VICTORY Authorization Architecture.

The PES is the entry-point into the VICTORY Authorization Architecture for a service on the platform and provides a common abstract access point to the authorization mechanisms. A protected resource's handler queries the PES with information about the requester, the action, and the resource itself. In a pure VICTORY implementation, the PES relays that information to the Policy Decision Service (PDS). In the event that a platform decides to utilize an alternate authorization framework, all VICTORY services requesting authorization would communicate with the PES as defined, and the PES would reroute and perform the necessary logic to use the alternate framework. In this way, non-VICTORY authorization methods can be used in an interoperable way. Refer to Figure 15 below.

The PES is not a Policy Enforcement Point that proxies all requests to a resource and performs the action of allowing or blocking the request to the resource; instead, the PES returns a decision back to the protected resource's handler. As the handler must allow or block the request, the response from the PES must be either a PERMIT or a DENY. The initial authorization decision may be modified with additional logic and considerations before it is returned to the handler. One scenario could include the use of supplemental authorization mechanisms to be considered after querying the VICTORY Authorization Architecture. The PES could utilize this additional information and logic to refine or override the PDS's decision.

If the final authorization decision, including any custom logic or information, is an INDETERMINATE decision, the PES must return a DENY decision back to the resource's handler.

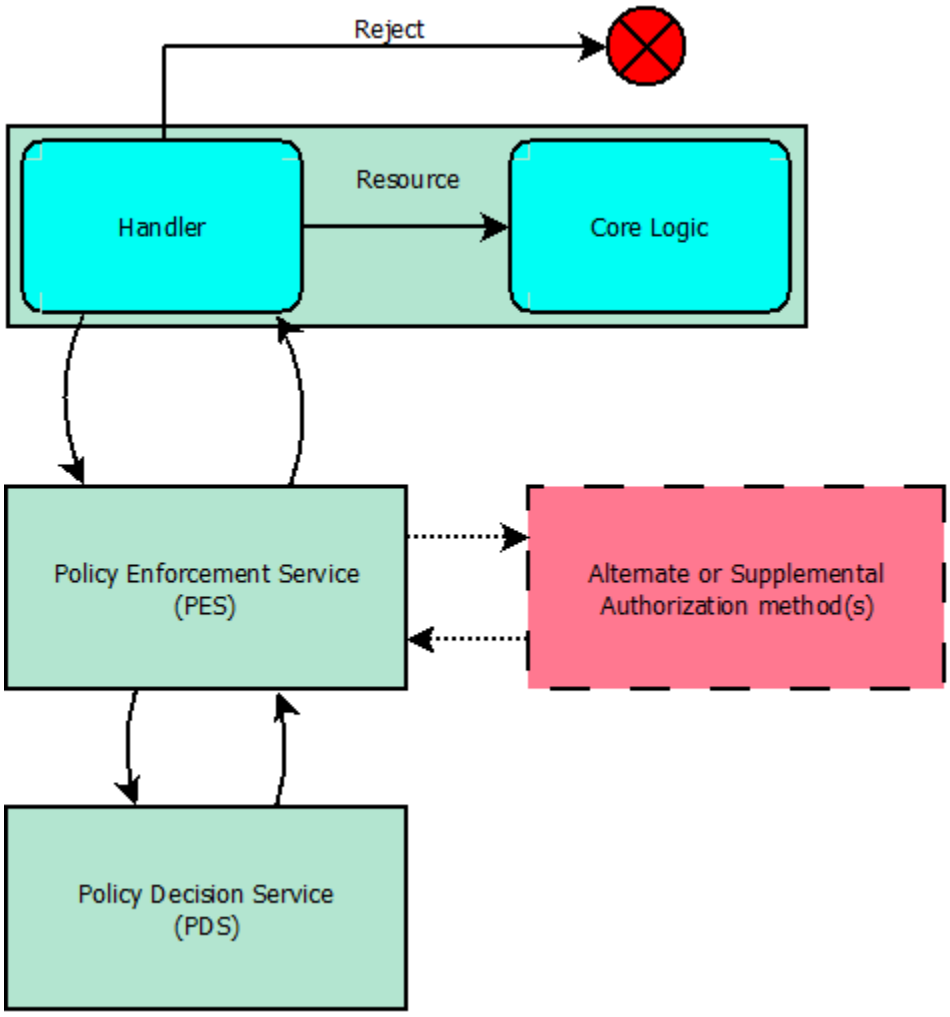

Figure 15. Policy Enforcement Information Flow

In summary, the PES relays information from a protected resource's handler to the authorization mechanism in use, abstracts that mechanism, provides an interoperable interface, supports flexibility with non-VICTORY frameworks, and provides a space at the entry-point to the authorization architecture for vendor value-add customization.

16.3.3.1 Policy Enforcement Management Specification <14013–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the policy enforcement management specification. Refer to the Core document for SAML 2.0 with errata for more information. The policy enforcement management specification shall implement functions that support the parameters as described in the following sections.

16.3.3.1.1 Policy Enforcement Configuration Parameters

1. ***getAuthorization()***
    a. Uses a SAML AuthzDecisionQuery element.
    b. Refer to Section 6: Assertion Query/Request Profile in "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0 – Errata Composite," Working Draft 06, 1 December 2009, for information on how to set up a SAML <AuthzDecisionQuery> and <Response> message.
    c. Request
        i. Subject

1. The Subject for getAuthorization() shall be the following for the indicated Authentication methods:
    a. Username Token
        i. The Username contained in the tag.
    b. X.509 Token
        i. The Subject of the Certificate.
    c. SAML Token
        i. The SAML Subject.

ii. Action
1. The Action for getAuthorization() shall be the following:
    a. Action shall be the string representation of the QName of the WSDL Operation.
        i. The bounds of the action are described in the section titled "String Bounds" under "VICTORY Data Types."
    b. This is defined as "{namespaceURI}localPart" as described on James Clark's XML Namespaces web page.
        i. Example for Orientation Service:
            1. Namespace: http://www.victory-standards.org/WSDLs/OrientationData.wsdl
            2. Operation: getOrientation
            3. Action: {http://www.victory-standards.org/WSDLs/OrientationData.wsdl}getOrientation

iii. Resource
1. The Resource for getAuthorization() shall be the full URI of the WSDL port being accessed.
2. The bounds of the resource are described in the section titled "URI Bounds" under "VICTORY Data Types."

d. Response
    i. PERMIT
    ii. DENY
    iii. Management Interface Error Types

2. ***PDS Location***
    a. Location of the PDS to contact.
    b. The PDS location shall be in the format of a valid URI.
    c. The bounds of the PDS location are described in the section titled "URI Bounds" under "VICTORY Data Types."
    d. The following error is defined for the PDS location parameter:
        i. Setting a PDS location outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.
    e. Operations include getPDS() and setPDS().

16.3.3.1.2 Additional Schema Restrictions

The SAML specification that these messages are based on includes a wide range of use-cases, many of which do not apply to VICTORY. The following are parts of the message that should not be used and additional information on various SAML parts.

1. ***getAuthorization()***
    a. SAML AuthzDecisionQuery/Response
        i. Ignore
            1. SAML Evidence
                a. AssertionIDRef
                    - Not used.
                b. AssertionURIRef
                    - Not used.
            2. Subject
                a. SubjectConfirmation
                    - This is mainly used for SAML Single Sign-On (SSO), which is not the case here.

3. Advice
    - The SAML Spec states that, “this information MAY be ignored by applications without affecting either the semantics or the validity of the assertion.” VICTORY will always ignore it.
4. Condition
    a. ProxyRestriction
        - The SAML Spec states that ProxyRestriction, “specifies limitations that the asserting party imposes on relying parties that in turn wish to act as asserting parties and issue subsequent assertions of their own on the basis of the information contained in the original assertion.” Since the assertions are not shared widely but queried, this was deemed unnecessary.
    b. AudienceRestriction
        - The SAML Spec states that, “the <AudienceRestriction> element specifies that the assertion is addressed to one or more specific audiences identified by <Audience> elements. Although a SAML relying party that is outside the audiences specified is capable of drawing conclusions from an assertion, the SAML asserting party explicitly makes no representation as to accuracy or trustworthiness to such a party.” This would be used in cases where the assertions could be broadcasted to different services. It does not fit with our connection-based Policy Enforcement Service.
5. Statement
    - This is an extension point for new Statements. Policy Enforcement does not use it.
6. AttributeStatement
    - This is for statements about SAML Attributes. Policy Enforcement does not use it.
7. AuthnStatement
    - This is for statements about SAML Authentication. Policy Enforcement does not use it.

ii. Adapt
1. The Decision must not be Indeterminate.
2. There must be at least one AuthzDecisionStatement per different decision needed.
3. All Name Identifier Formats identifiers defined in Section 8.3 of [SAMLCore] must be implemented.
4. If using Encrypted IDs, they must be encrypted with the public key of the Policy Enforcement Service.
5. Response should return encrypted Assertions if Encrypted IDs are provided.
6. When the Issuer element is used in the Response, the value of this element must correspond to the unique name of the Policy Enforcement Service.

16.3.4 Policy Store Service

The Policy Store Service (PS) is one of the four services that comprise the VICTORY Authorization Architecture.

The Policy Store Service is the repository for policy information for the platform and provides security policy information to the Policy Decision Service (PDS) to help the PDS determine whether access to a provider resource should be granted or not.

Policies must be stored in a location where they can be easily retrieved. The Policy Store Service provides the mechanism to transfer security policies. The PS uses XACML as the standard way to transfer policies.

16.3.4.1 Policy Store Management Specification <14022–20210409, Pro>

The appendix titled “VICTORY WSDLs” provides the name of the file containing the web service details for the policy store management specification. Refer to the Core document for SAML 2.0 with errata for more information.

All Policies sent to or returned by the Policy Store must be in the XACML Format and must be processed as such. Refer to the XACML 2.0 specification for additional information on XACML.

The policy store management specification shall implement functions that support the parameters as described in the following sections.

16.3.4.1.1 Policy Store Configuration Parameters

1. ***addPolicies()***
    a. Uses a SAML Assertion with an XACML-SAML XACMLPolicyStatement.
    b. Refer to the XACML Technical Committee page for additional information on XACML.
    c. Refer to the SAML 2.0 Profile of XACML for additional information on integrating SAML 2.0 with XACML.
2. ***getPolicies()***
    a. Uses an XACML-SAML XACMLPolicyQuery.
    b. Refer to Section 6: Assertion Query/Request Profile in "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0 – Errata Composite," Working Draft 06, 1 December 2009, for information on how to set up a SAML <XACMLPolicyQuery> and <Response> message.
    c. Refer to the SAML 2.0 Profile of XACML for additional information on integrating SAML 2.0 with XACML.
3. ***removePolicies()***
    a. Uses an XACML-SAML XACMLPolicyQueryType to specify what policies to delete.
    b. The syntax of the XACMLPolicyQueryType is the same as getPolicies().

16.3.4.1.2 PolicySetId and PolicyId Rules

PolicySetIds and PolicyIds are tracked separately. That means that they do not have to be unique from each other. Thus, the following rules must be followed.

1. If a Policy is added that shares the same ID as a Policy contained in a PolicySet, a GENERIC_FAULT error must be thrown.
2. When adding a PolicySet, if that PolicySet contains a Policy/PolicySet that shares an ID with an existing Policy/PolicySet respectively, a GENERIC_FAULT error must be thrown.
3. If a Policy is added that shares the same ID as an existing Policy, the existing Policy is overwritten.
4. If a PolicySet is added that shares the same ID as an existing PolicySet, the existing PolicySet is overwritten.
5. If removing by ID, if it is an ID of a Policy/PolicySet contained in a PolicySet, a GENERIC_FAULT error must be thrown.

16.3.4.1.3 Additional Schema Restrictions

The SAML specification that these messages are based on includes a wide range of use-cases, many of which do not apply to VICTORY. The following are parts of the message that should not be used and additional information on various SAML parts.

1. ***addPolicies()***
    a. Ignore
        i. SAML Assertion
            1. Subject
                a. SubjectConfirmation
                - This is mainly used for SAML Single Sign-On (SSO), which is not the case here.
            2. Advice
            - The SAML Spec states that, "this information MAY be ignored by applications without affecting either the semantics or the validity of the assertion." VICTORY will always ignore it.
            3. Condition
                a. OneTimeUse
                - The SAML Spec states that, "relying parties may choose to retain assertions or the information they contain in some other form, for reuse." This does not fit the context of the Policy Store for adding policies.
                b. ProxyRestriction

- The SAML Spec states that ProxyRestriction, “specifies limitations that the asserting party imposes on relying parties that in turn wish to act as asserting parties and issue subsequent assertions of their own on the basis of the information contained in the original assertion.” This does not fit the context of the Policy Store for adding policies.

c. AudienceRestriction

- The SAML Spec states that, “the <AudienceRestriction> element specifies that the assertion is addressed to one or more specific audiences identified by <Audience> elements. Although a SAML relying party that is outside the audiences specified is capable of drawing conclusions from an assertion, the SAML asserting party explicitly makes no representation as to accuracy or trustworthiness to such a party.” This would be used in cases where the assertions could be broadcasted to different services. It does not fit with our connection-based Policy Store.

4. AttributeStatement
   - This is for statements about SAML Attributes. Policy Store does not use it.
5. AuthnStatement
   - This is for statements about SAML Authentication. Policy Store does not use it.
6. AuthzDecisionStatement
   - This is for statements about SAML Authorization Decisions. Policy Store does not use it.

b. Adapt

i. All Name Identifier Formats identifiers defined in Section 8.3 of [SAMLCore] must be implemented.
ii. If using Encrypted IDs, they must be encrypted with the public key of the Policy Store Service.
iii. When the Issuer element is used in the Response, the value of this element must correspond to the unique name of the Policy Store Service.

2. ***getPolicies()***

a. Ignore

i. SAML Response
   1. Assertion
      a. Subject
         i. SubjectConfirmation
            - This is mainly used for SAML SSO, which is not the case here.
      b. Advice
         - The SAML Spec states that, “this information MAY be ignored by applications without affecting either the semantics or the validity of the assertion.” VICTORY will always ignore it.
      c. Condition
         i. ProxyRestriction
            - The SAML Spec states that ProxyRestriction, “specifies limitations that the asserting party imposes on relying parties that in turn wish to act as asserting parties and issue subsequent assertions of their own on the basis of the information contained in the original assertion.” This does not fit the context of the Policy Store for getting policies.
         ii. AudienceRestriction
            - The SAML Spec states that, “the <AudienceRestriction> element specifies that the assertion is addressed to one or more specific audiences identified by <Audience> elements. Although a SAML relying party that is outside the audiences specified is capable of drawing conclusions from an assertion, the SAML asserting party explicitly makes no representation as to accuracy or trustworthiness to such a party.” This would be used in

cases where the assertions could be broadcasted to different services. It does not fit with our connection-based Policy Store.

b. Adapt
   i. All Name Identifier Formats identifiers defined in Section 8.3 of [SAMLCore] must be implemented.
   ii. VersionMatch in IdReferenceType does NOT imply that older versions are stored and queried. Versioning of overwritten policies are outside the scope of this specification.
   iii. When the Issuer element is used in the Response, the value of this element must correspond to the unique name of the Policy Store Service.

3. ***removePolicies()***
   a. Ignore
      i. SAML XACMLPolicyQuery
         1. Consent
            - Not useful for the remove function.
   b. Adapt
      i. All Name Identifier Formats identifiers defined in Section 8.3 of [SAMLCore] must be implemented.
      ii. VersionMatch in IdReferenceType does NOT imply that older versions are stored and queried. Versioning of overwritten policies are outside this scope.
      iii. When the Issuer element is used in the Response, the value of this element must correspond to the unique name of the Policy Store Service.

16.3.5 Authentication Service

The Authentication Service provides platforms the option of enabling a central storage for authentication credentials. Instead of each service having to authenticate X.509 Certificates, for example, this function can be pushed to a trusted device on the network. This could enable smaller services and better (more correct) checking of authentication information. Note that this service is for checking to see if a Username/Password, X.509 Certificate, or SAML Token is from the person that they claim to be (authentication). For authorization, refer to the section that deals with the Authorization Architecture.

16.3.5.1 Authentication Service Management Specification <14028–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the authentication service management specification. The authentication service management specification shall implement functions that support the parameters as described in the following sections.

16.3.5.1.1 Overview

The Authentication Service stores trusted Authentication information and validates unknown Authentication information against it. Instead of requiring every VICTORY service to store usernames and passwords or PKI trust chains, these services can consult a centralized repository of Authentication information. The intent is to push some of the Authentication management burden away from the individual VICTORY services. To facilitate ease of use, the Authentication Service is designed to support the Authentication Methods of the VICTORY specification.

> ***NOTE:*** *Many of the profiles require that the message be signed by the requester. The Authentication Service will only tell you if the provided credentials can be trusted, it does not tell you if the message comes from that entity (the message is signed correctly). That is left to the particular service.*

16.3.5.1.2 X.509 Token Authentication

For X.509 Token details, refer to Web Services Security X.509 Certificate Token Profile Version 1.1.1, 18 May 2012, OASIS Standard.

According to the Web Services Security X.509 Certificate Token Profile, there are several ways that an X.509 certificate can be referenced in the Token.

- ***Binary Security Token***

The simplest way is to just include the actual X.509 certificate(s) in the Token. For this method, one must trust the provided certificate(s). The Authentication Service allows you to submit the certificates to verify that they can be trusted (certification paths to a root CA, not expired, etc.).

There are three different binary forms that are accepted: PKCS #7 file, PKIPath, and an X.509 certificate.

- ***Subject Key Identifier***

  This way is just a reference to a Certificate's X.509v3 SubjectKeyIdentifier extension. Since the actual certificate is not provided, a service can request the referenced certificate from the Authentication Service. The Authentication Service would only contain the certificate if it was trusted.

- ***Issuer and Serial Number***

  The last way is by referring to the tuple of the Issuer and the Serial Number of the X.509 certificate. Like previously, the service can request the referenced certificate from the Authentication Service. The Authentication Service would only contain the certificate if it was trusted.

16.3.5.1.3 SAML Token Authentication

For SAML Token details, refer to Web Services Security SAML Token Profile Version 1.1.1, 18 May 2012, OASIS Standard.

SAML Tokens can be sent to the Authentication Service for validation. These Tokens must be signed by a trusted source and the Authentication Service can only tell whether or not the particular Token is signed by a trusted authority. It cannot make any statements about the message that contains the Token. For example, it cannot determine if that message was sent by that SAML Subject or not.

16.3.5.1.4 Username Token Authentication

For Username Token details, refer to Web Services Security Username Token Profile Version 1.1.1, 18 May 2012, OASIS Standard.

> ***NOTE:*** *Transport Layer Security (TLS) is required. Not using TLS would allow passwords to be sent in plaintext across the network.*

There are two ways a Username Token can be used:

- ***Plaintext***

  The Username and PasswordText are sent without additional processing.

- ***Digest***

  The password is subjected to a Digest Algorithm as specified in the profile.

The Username Token profile allows the use of Nonce and a Created timestamp. These have various uniqueness requirements as spelled out in the profile. The Authentication Service does not track these. Therefore, it is the responsibility of the client.

16.3.5.1.5 Authentication Service Configuration Parameters

1. ***checkUser()***

   Checks if a Username/Password or a Username/Digest Password can be authenticated.

   a. Request
      i. Username
         - The bounds of the username are described in the section titled "String Bounds" under "VICTORY Data Types."
      ii. Either '1' or '2' below shall be provided.

1. Password
    - Plaintext password.
    - The bounds of the password are described in the section titled "String Bounds" under "VICTORY Data Types."
2. Digest Password
    a. Digest
        - A Password subjected to Digest Algorithm is defined in the Username Token profile.
        - The bounds of the digest are described in the section titled "String Bounds" under "VICTORY Data Types."
    b. Nonce
        - A Nonce used in Digest Algorithm is defined in the Username Token profile.
        - The bounds of the nonce are described in the section titled "String Bounds" under "VICTORY Data Types."
        - Nonce may be provided as part of the Digest Password.
    c. Created
        - A Timestamp used in Digest Algorithm is defined in the Username Token profile.
        - The bounds of the timestamp are described in the section titled "String Bounds" under "VICTORY Data Types."
        - Created may be provided as part of the Digest Password.

b. Response
    i. Management Interface Error Types
        1. The GENERIC_FAULT error shall be returned for the following:
            a. Authentication Failure
                i. ERROR_TEXT must not be included.
        2. The OUT_OF_RANGE error may be returned for the following:
            a. The string entered was outside the range defined by this specification document.

2. ***addUsername()***

Adds a Username and Password.

a. Request
    i. Username
        - The bounds of the username are described in the section titled "String Bounds" under "VICTORY Data Types."
    ii. Password
        - Plaintext password.
        - The bounds of the password are described in the section titled "String Bounds" under "VICTORY Data Types."

b. Response
    i. Management Interface Error Types
        1. The OUT_OF_RANGE error may be returned for the following:
            a. The string entered was outside the range defined by this specification document.

3. ***removeUsername()***

Removes a Username.

a. Request
    i. Username
    ii. The bounds of the username are described in the section titled "String Bounds" under "VICTORY Data Types."

b. Response
    i. Management Interface Error Types
        1. The OUT_OF_RANGE error may be returned for the following:

a. The string entered was outside the range defined by this specification document.

4. ***getUsernames()***

   Returns all Usernames.

   a. Response
      i. Zero-to-many Usernames
      ii. Management Interface Error Types

5. ***addTrustedCert()***

   Adds certificate(s) that can be trusted in the form of an X.509 certificate, an X509PKIPathv1, or a PKCS #7 file. For example, Root CAs.

   a. Request
      i. X.509 certificate
         - An X.509v3 certificate.
         - Distinguished Encoding Rules (DER) or Privacy-Enhanced Electronic Mail (PEM) format.
         - Base64 Encoded.
            - Refer to IETF RFC 4648: The Base16, Base32, and Base64 Data Encodings for more information on Base64 encoding.

      ii. X509PKIPathv1
         - An ordered list of X.509 certificates packaged in a PKIPath.
         - Base64 Encoded.

      iii. PKCS #7 file
         - A list of X.509 certificates and (optionally) certificate revocation lists (CRLs) packaged in a PKCS #7 wrapper.
         - Base64 Encoded.

   b. Response
      i. Management Interface Error Types

6. ***removeTrustedCert()***

   Removes a trusted certificate by Issuer and Serial Number.

   a. Request
      i. Issuer
         - The Issuer field in an X.509v3 certificate.
         - Refer to IETF RFC 5280.

      ii. Serial Number
         - The Serial Number field in an X.509v3 certificate.
         - Integer Format.

   b. Response
      i. Management Interface Error Types

7. ***getTrustedCert()***

   Returns the certificate associated with an Issuer and Serial number, or a SubjectKeyIdentifier.

   a. Request
      i. Issuer
         - The Issuer field in an X.509v3 certificate.
         - Refer to IETF RFC 5280.

      ii. Serial Number
         - The Serial Number field in an X.509v3 certificate.
         - Integer Format.

      iii. SubjectKeyIdentifier
         - The SubjectKeyIdentifier field in an X.509v3 certificate.
         - HEX Format.
         - Strip out OCTET STRING encoding prefixes.

- 04160414EDE200AA9F3CC7E27136C0A5AEA2FA5A99104283 -> EDE200AA9F3CC7E27136C0A5AEA2FA5A99104283

b. Response
   i. X.509 certificate
      - An X.509v3 certificate.
      - DER format.
      - Base64 Encoded.
   ii. Management Interface Error Types

8. ***getListOfTrustedCerts()***

Returns a list of the Issuer and Serial Numbers with optional Subject of all the current trusted certificates.

a. Response
   i. Issuer
      - The Issuer field in an X.509v3 certificate.
      - Refer to IETF RFC 5280.
   ii. Serial Number
      - The Serial Number field in an X.509v3 certificate.
      - Integer Format.
   iii. Friendly Subject
      - The Subject field in an X.509v3 certificate.
      - This is an optional field. The intent is for User Interface (UI) information.
   iv. Management Interface Error Types

9. ***checkX509Cert()***

Validates that the given certificate(s) can be trusted.

a. Request
   i. X.509 certificate
      - An X.509v3 certificate.
      - DER or PEM format.
      - Base64 Encoded.
   ii. X509PKIPathv1
      - An ordered list of X.509 certificates packaged in a PKIPath.
      - Certificates packaged must form a single chain.
      - Base64 Encoded.
   iii. PKCS #7 file
      - A list of X.509 certificates and (optionally) certificate revocation lists (CRLs) packaged in a PKCS #7 wrapper.
      - Certificates packaged must form a single chain.
      - Base64 Encoded.
b. Response
   i. Management Interface Error Types
      1. The GENERIC_FAULT error shall be returned for the following:
         a. The provided certificate(s) do not chain to a trusted authority.

10. ***checkSAMLToken()***

Validates that the given SAML Token has been signed by a trusted authority.

a. Request
   i. SAML Token
      - Base64 Encoded.
b. Response
   i. Management Interface Error Types
      1. The GENERIC_FAULT error shall be returned for the following:
         a. The Token was not signed by a trusted authority.

16.4 End Node Component Types

The components presented in the following sections are devices that attach to the network, and either provide services not related to transporting data, adapt non-networking systems to the VDB, or host software (shared processing units).

16.4.1 Generic End Node

16.4.1.1 Generic End Node Management Parameters <13003–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the generic end node management parameters. A textual description of the generic end node management parameters is shown below.

1. ***Ethernet Interfaces***
    a. This parameter indicates the list of ethernet interfaces on the generic end node.
    > ***NOTE:*** *Loopback should not be returned inside the list of ethernet interfaces.*
    b. The contents of an ethernet interface are described in the section titled "Ethernet Interface" under "Generic End Node Complex Types."
    c. Operations include Get().
2. ***DHCP Enabled***
    a. This parameter indicates whether DHCP is enabled on the specified ethernet interface.
    b. The ethernet interface shall be specified by its unique name.
    c. This parameter contains one flag indicating whether DHCPv4 is enabled or disabled and a second flag for indicating whether DHCPv6 is enabled or disabled.
        i. For each flag, a Boolean value of True means DHCP is enabled and a value of False means DHCP is disabled.
    d. The following errors are defined for the DHCP enabled parameter:
        i. Getting the DHCP enabled flags with a unique name that does not match the unique name of an ethernet interface on the generic end node shall result in a GENERIC_FAULT error.
        ii. Getting the DHCP enabled flags with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    e. Operations include Get().
3. ***Host Name***
    a. This parameter identifies the hostname of the generic end node.
    b. The bounds of the host name are described in the section titled "String Bounds" under "VICTORY Data Types."
    c. The following errors are defined for the host name parameter:
        i. Setting a host name to a value outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    d. Operations include Get() and Set().
4. ***Date Time***
    a. This parameter identifies the current system date/time of the generic end node.
    b. The date/time shall be in standard dateTime format.
    c. The following errors are defined for the date/time parameter:
        i. Setting the date/time of the generic end node outside the format as described in the section titled "Timestamp Format for Application Layer Data" may result in an OUT_OF_RANGE error.
    d. Operations include Get() and Set().
5. ***Auto-Discovery Enabled***
    a. This parameter identifies whether the generic end node has zero configuration enabled or disabled.
    b. Setting auto-discovery to False on the generic end node shall result in all services on the generic end node not being able to announce themselves via zero configuration.
    c. A Boolean value of True means "enabled" and a value of False means "disabled."
    d. The following errors are defined for the auto-discovery enabled parameter:
        i. Getting or setting the auto-discovery enabled flag on an generic end node that does not implement the optional auto-discovery interface shall result in a NOT_SUPPORTED error.
    e. Operations include Get() and Set().
6. ***Display Blackout Enabled***
    a. A Generic End Node with a display shall include the display blackout enabled parameter.

b. This parameter identifies whether the display of the generic end node is in blackout mode.
c. A Boolean value of True means the display and indicators are turned off and a value of False means the display and indicators are turned on.
d. The following errors are defined for the display blackout enabled parameter:
    i. Getting or setting the display blackout enabled flag on a generic end node that contains no display or indicators shall result in a NOT_SUPPORTED error.
e. Operations include Get() and Set().

7. ***Generic End Node***
    a. This parameter puts the generic end node is standby mode for conserving power, restarts the generic end node, and purges the generic end node.
    b. Purging shall render data on the generic end node unrecoverable by keyboard or laboratory attack.
    c. Formatting or repartitioning storage is not a purge.
    d. Purging shall follow guidelines established in NSA/CSS Manual 9–12 and AR 25–2, as appropriate.

> ***NOTE:*** *Purging products approved by Army Best Business Practices and the NSA National Information Assurance Partnership (NIAP) Common Criteria include, but are not limited to, White Canyon WipeDrive 6.1 and the University of California, San Diego (UCSD) Center for Magnetic Recording Research (CMRR) Secure Erase.*

    e. The following errors are defined for the generic end node parameter:
        i. Attempting to purge a generic end node that does not have the capability to purge shall result in a NOT_SUPPORTED error.
    f. Operations include Restart(), Standby(), and Purge().

> ***NOTE:*** *The section titled "Management Interface Error Types" provides additional details on the errors defined for the generic end node management parameters along with a description of additional errors that were not specified but may be returned.*

16.4.1.2 Generic End Node Complex Types <13004–20210409, Pro>

> ***NOTE:*** *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

1. ***Ethernet Interface***
    a. **Interface Name**
        i. The name of the ethernet interface (e.g., eth0, wlan0, etc.).
        ii. The bounds of the interface name are described in the section titled "String Bounds" under "VICTORY Data Types."
        iii. The interface name shall be provided as part of the ethernet interface.
    b. **Unique Name**
        i. A VICTORY handle for the ethernet interface.
        ii. The unique name shall be a string unique within the scope of the ethernet interfaces on the generic end node.
        iii. The bounds of the unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
        iv. The unique name shall be provided as part of the ethernet interface.
    c. **MAC Address**
        i. The MAC address of the ethernet interface.
        ii. The bounds of the MAC address are described in the section titled "String Bounds" under "VICTORY Data Types."
        iii. The MAC address shall be provided as part of the ethernet interface.
    d. **IP Configurations**
        i. The list of IP configurations associated with the ethernet interface.
        ii. The contents of an IP configuration are described in the section titled "IP Configuration" under "Generic End Node Complex Types."
        iii. IP configurations may be provided as part of the ethernet interface.
2. ***IP Configuration***
    a. **Type**
        i. Indicates if the IP configuration is static or automatic.

ii. The type shall be a string chosen from one of the following values:
    1. STATIC
    2. DHCP_V4
    3. DHCP_V6
    4. LINK_LOCAL
    5. STATELESS_AUTOCONFIGURATION
iii. The type shall be provided as part of the IP configuration.

b. **Address**
i. The address of the IP configuration.
ii. The address shall be a string in the format of a valid IPv4 or IPv6 address.
iii. The address shall be provided as part of the IP configuration.

c. **Subnet**
i. The subnet mask of the IP configuration.
ii. The subnet mask shall be a string in the format of a valid IPv4 address or a slash-prefix form (e.g., /8, /64, etc.).
iii. The subnet mask shall be a valid subnet mask.
iv. The subnet mask shall be provided as part of the IP configuration.

d. **Broadcast**
i. The broadcast address of the IP configuration.
ii. The broadcast shall be a string in the format of a valid IPv4 address.
iii. The broadcast may be provided as part of the IP configuration.

e. **Gateway**
i. The gateway address of the IP configuration.
ii. The gateway shall be a string in the format of a valid IPv4 or IPv6 address.
iii. The gateway may be provided as part of the IP configuration.

16.4.2 Shared Processing Unit

The Shared Processing Unit (SPU) is a hardware device that will typically host multiple software applications, each of which will share the device's hardware resources, such as processing resource and non-volatile storage. Optionally, an SPU can also provide warfighter machine interface (WMI) devices, such as displays, keyboards, and mice. The SPU will host at least one VDB interface for implementing communications with the VDB. If the device hosts a Separation Kernel, and therefore interfaces with multiple security domains, it will host multiple VDB interfaces for communicating with each network.

16.4.2.1 SPU Functions

Shared processing units host software that processes data on the VDB, implement the shared data services, or provide WMIs. Shared processing components can typically be shared by a number of different software applications and are utilized to implement more than one type of functionality, as opposed to legacy stovepipe systems.

16.4.2.2 SPU Data Interface

The SPU Data Interface shall adhere to the Generic End Node Data Interface as defined in the section titled "Generic End Node Data Interface."

16.4.2.3 SPU Health Publishing Interface

See the section titled "Syslog-Based Connectionless Health Publishing Specification" for a detailed description of the syslog-based connectionless health publishing interface required when managing an SPU.

16.4.2.4 Shared Processing Unit Management Parameters <13006–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the SPU management parameters. A textual description of the SPU management parameters is shown below.

1. ***Services***
    a. This parameter identifies the VICTORY services on the shared processing unit.
    b. A detailed description of a service is provided in the section titled "Service" under "Shared Processing Unit Complex Types."
    c. Operations include Get().
2. ***Service***
    a. This parameter controls the VICTORY services on the shared processing unit.
    b. The VICTORY service being controlled shall be specified by its unique name.
    c. The following errors are defined for the service parameter.
        i. Starting, stopping, or restarting a VICTORY service with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
        ii. Starting, stopping, or restarting a VICTORY service with a unique name that does not match the unique name of a VICTORY service on the shared processing unit shall result in a GENERIC_FAULT error.
    d. Operations include Start(), Stop(), and Restart().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the shared processing unit management parameters along with a description of additional errors that were not specified but may be returned.*

16.4.2.5 Shared Processing Unit Complex Types <13013–20210409, Pro>

NOTE: *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

1. ***Service***
    a. **Unique Name**
        i. The unique name of the VICTORY service on the SPU.
        ii. The bounds of the unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
        iii. The unique name shall be unique within the scope of all other services on the SPU.
        iv. The unique name shall be provided as part of the VICTORY service.
    b. **State**
        i. The current state of the VICTORY service.
        ii. The state of the service shall be a string chosen from one of the following values:
            1. STOPPED
            2. STARTING
            3. RUNNING
        iii. The state shall be provided as part of the VICTORY service.
    c. **Component Types**
        i. A list of VICTORY component types implemented by the VICTORY service.
        ii. A detailed description of a component type is provided in the section titled "Component Type" under "Shared Processing Unit Complex Types."
        iii. At least one component type shall be implemented by the VICTORY service.
2. ***Component Type***
    a. **Unique Name**
        i. The unique name of the VICTORY component type.
        ii. The bounds of the unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
        iii. The unique name shall be a unique name of one of the VICTORY component types on the VDB.
        iv. The unique name shall be provided as part of the VICTORY component type.
    b. **Type**
        i. The type of VICTORY component implemented on the VICTORY service.
        ii. The type shall be a string representing a VICTORY component type.
            1. Valid values can be found inside the sections of the individual component configuration descriptions.
                a. For example, the section titled "Position Service-Specific Configuration Description Content" under "Position Service Configuration Description"

contains a “Position Service Component.” The ***type*** attribute for the position service component is “POSITION_SERVICE”, so “POSITION_SERVICE” is a valid component type value.

iii. The type shall be provided as part of the VICTORY component type.

16.4.2.6 SPU Platform API Description <Informational>

All VICTORY-compliant shared processing units (SPUs) will provide a standard application programming interface (API) for software development. This API will provide services to VICTORY-compliant software at multiple levels of abstraction from the actual hardware of an SPU. At a minimum, the SPU API will provide access to storage, network, processing, and memory resources. In addition, the SPU API will provide mechanisms for multiprocessing, interprocess communication, synchronization, and process control. The SPU API may additionally offer access to graphic and display capabilities, peripherals, and virtualization capabilities for code isolation at both the bytecode and machine virtualization levels. Finally, the SPU API will offer access to the computing resources of an SPU through multiple computer programming language bindings.

Three levels of abstraction have been defined for the SPU API. The lowest level is called the SPU Low-Level API. This level defines the API for native software, which is compiled against a specific computer architecture to run on the SPU. The middle level is called the SPU Mid-Level API. This level defines the API for software running on a bytecode-interpreting virtual machine. Software implemented against the Mid-Level API is typically compiled to an architecture-agnostic bytecode. The highest level API, which has not yet been defined by VICTORY, is called the SPU High-Level API. This level defines the API for software that is interpreted, rather than compiled. Software defined at this level is often described as a “script,” and the programming languages that operate at this level are often called “scripting languages.” By defining three levels of abstraction, the SPU API allows for most of the possible tradeoffs between portability and performance.

16.4.2.6.1 SPU Mid-Level API Description

A Java virtual machine (VM) implementation is software that executes Java bytecode on standard operating systems. When combined with a standard set of APIs and libraries, the Java VM allows developers to write programs in high-level languages, compile them once, and then run the programs on a variety of different devices, regardless of platform or operating system configurations. By specifying a standard Java Runtime Environment, which consists of a Java VM and a standard set of Java APIs and libraries that will be available on a VICTORY-compliant shared processing unit, developers can be guaranteed that the high-level language software compiled to Java bytecode will run on any shared processing unit in a VICTORY implementation.

16.4.2.6.2 SPU Low-Level API Description

The Low-Level SPU API consists of a suite of runtime and build-time software libraries. The primary purpose of this suite of libraries is to provide a compile-time abstraction from the underlying operating system of a shared processing unit, which can serve as a practical design target for new software development. A secondary purpose is to “ease the pain” of porting software from one VICTORY-compliant SPU to another. In other words, the feature provided by the low-level SPU API is a standard VICTORY compile-time environment that can be used to generate native applications. This is in contrast to the mid-level SPU API, which provides a standard runtime for “build once, run anywhere” applications.

16.4.2.6.3 SPU Platform API Profiles

It is reasonable to assume that different shared processing units will provide different portions of the SPU API based on their capabilities. Therefore, in addition to being defined at three levels of abstraction, the SPU API is also scaled according to different profiles based primarily on the capabilities of the shared processing unit. For example, an SPU that is intended to primarily function as a source for VICTORY Data Messages (VDMs) will in many cases interface with a user only through the VDB (the network). Such a system will often lack a display, and so, likely has little use for many of the graphical user interface (GUI) libraries and capabilities provided by the SPU API. For such systems, a “headless” profile is defined.

16.4.2.7 Headless SPU API Profile

All VICTORY-compliant shared processing units that do not support a display shall, at a minimum, provide the Headless SPU API profile. The specific details of this profile are called out separately for each SPU API abstraction level, and in the VT table titled "Shared Processing Unit Platform API Headless Profile."

16.4.2.8 Display Unit SPU API Profile

All VICTORY-compliant shared processing units that support a display shall, at a minimum, provide the Display Unit SPU API profile. The specific details of this profile are called out separately for each SPU API abstraction level, and in the VT table titled "Shared Processing Unit Platform API Display Profile."

It should be noted that, in most cases, support of the Display Unit SPU API profile includes support of the Headless SPU API profile.

16.4.2.9 SPU Mid-Level API <13005–20190418, Pro>

All VICTORY-compliant shared processing units (SPUs) that are required to implement the SPU Mid-Level API shall provide a Java virtual machine implementation that is compatible with a long term supported (LTS) version of The Java™ Virtual Machine Specification: Java™ Platform Standard Edition (SE). The selected version of Java™ Platform SE shall be supported at the time of the SPU's certification of compliance. For an SPU component, this is at the time of its manufacture and delivery. For a system supplying an SPU as a member of its implemented VICTORY component types, this is at the time of the system's certification of compliance. The Java virtual machine implementation included on a VICTORY-compliant SPU is required to support, at a minimum, an equivalent implementation of all of the APIs and libraries included in the Java™ Platform SE.

A list of the APIs included in the Java™ Platform, Standard Edition API Specifications can be found at: https://docs.oracle.com/en/java/javase/.

16.4.2.10 SPU Low-Level API

All implementations of the VICTORY shared processing unit low-level API shall provide some portion of an implementation of POSIX, Qt, and libBoost.

16.4.2.10.1 POSIX <13009–20210409, Exp>

All VICTORY-compliant shared processing units (SPUs) that implement POSIX shall support the latest version of POSIX. The selected version of POSIX shall be supported at the time of the SPU's certification of compliance. For an SPU component, this is the time of its manufacture and delivery. For a system supplying an SPU as a member of its implemented VICTORY component types, this is at the time of the system's certification of compliance. The latest POSIX specification is freely available at: http://pubs.opengroup.org/onlinepubs/9699919799/.

SPUs that conform to either the Headless SPU API Profile or the Display Unit SPU API Profile have the same POSIX requirements.

SPUs shall support POSIX by adhering to the "Base Conformance" required by the POSIX specification, as defined by Section 2.1.3 "POSIX Conformance." In addition, the "Encryption" and "XSI STREAMS" Option Groups shall be supported.

> ***NOTE:*** *An SPU's operating system (OS) does not need to meet a standard of "strict conformance" to the POSIX standard (i.e., such an OS does not need to be certified by the Open Group), but rather, it must be capable of passing the Open Group's freely available conformance tests.*

16.4.2.10.2 Qt <Informational>

Unlike POSIX, Qt is not a standard, but rather a freely available abstraction library. The Qt library is composed of modules, and different subsets of these modules are required by the Headless SPU API Profile and the Display Unit SPU API Profile.

The core requirements for Qt (the minimum version required for VICTORY compliance) are described in the section titled Qt Requirements for the Headless SPU API Profile.

16.4.2.10.2.1 Qt Requirements for the Headless SPU API Profile <13010–20210409, Pro>

The versions of Qt that shall be supported are Qt version 5.6 or later. The different point-release versions of Qt 5.X are compile-time backwards compatible with each other.

SPUs that conform to the Headless SPU API Profile shall provide the following Qt modules:

- QtCore
- QtNetwork
- QtSQL
- QtTest
- QtXmlPatterns

16.4.2.10.2.2 Qt Requirements for the Display Unit SPU API Profile <13011–20210409, Pro>

SPUs that conform to the Display Unit SPU API Profile shall meet all requirements, and provide all modules specified by the Headless SPU API Profile, along with the following additional Qt modules:

- QtGui
- QtMultimedia
- QtQML
- QtQuick
- QtWebEngine
- QtWidgets
- QtSVG

16.4.2.10.3 libBoost <13012–20210409, Pro>

LibBoost is not a single library, but rather a collection of “best-practice” cross-platform C++ libraries. The libBoost libraries provide extensions to the C++ standard template library (STL), platform abstraction for standard functionality (such as threads and input/output), and efficient number processing structures. It is primarily this last capability of the libBoost libraries that is of interest to VICTORY.

The libBoost libraries are very modular, and many of the libraries are “header-only.” The advantage of “header-only” libraries is that they are only required at build time (i.e., they are only necessary on a build platform, and can be removed from an “in-the-field” SPU). As such, it might be somewhat confusing for an SPU implementer to know what is meant by the requirement that VICTORY-compliant SPUs support portions of the libBoost libraries. A simple litmus test is the following: If the SPU is a “build environment” where software will be compiled, then all portions of the following libBoost requirements must be supported; if the SPU is not a “build environment,” but will only run pre-compiled binaries, then only the libraries marked as having runtime requirements must be supported.

SPUs that conform to either the Headless SPU API Profile or the Display Unit SPU API Profile have the same libBoost requirements.

VICTORY-compliant SPUs shall support libBoost version 1.66 or later. All libraries in the category “Math and Numerics” shall be supported:

NOTE: *libBoost versions prior to 1.66 contain Regex library security issues and should not be used.*

- Accumulators
- Geometry (post 1.40)
- Integer
- Interval

- Math
- Math Common Factor
- Math Octonion
- Math Quaternion
- Math/Special Functions (has runtime requirements)
- Math/Statistical Distributions
- Multi-Array
- Numeric Conversion
- Operators
- Random
- Ratio (post 1.40)
- Rational
- uBLAS

In addition, all libraries that were included in the C++ Technical Report 1 (TR1), and that have not already been listed, shall also be supported:

- Array
- Bind
- Function
- Function/Hash
- Member Function
- Ref
- Regex (has runtime requirements)
- Smart Ptr
- Tuple
- Type Traits
- Unordered

16.4.3 Network Adapter

16.4.3.1 Network Adapter Functions

Network Adapter is a generic term for a device needed to help legacy (or non-standard) systems interface with the VDB. This device will provide some form of data conversion at the physical, transport, and application layers that the native device or system does not implement in the VICTORY standard way. Typically, a network adapter will be designed specifically for the native device that it is adapting to the VDB. The goal is that future components and systems will begin adopting the VICTORY Standard, and thus the need for network adapters will decrease over time.

16.4.3.2 Network Adapter Interfaces

A network adapter shall have at least one VDB physical interface, and shall utilize any other logical interfaces required to interact with the native system. The network adapter shall implement the interfaces of the type of device that it is adapting. As such, there are no network adapter-specific interfaces.

17. APPLICATION LAYER MESSAGING

The VDB data transport interface supports the delivery of messages of many types. The application layer messaging specifications define the format and protocols for the messaging interfaces between applications that utilize the VDB. In many cases, existing standards are ubiquitous and technically sufficient (e.g., streaming video and audio), so VICTORY specifications adopt those standards. In other cases, existing standards did not exist or were not directly applicable (e.g., position data), so VICTORY-specific message formats and delivery protocols were developed.

For application interfaces that adopted existing standards, the application layer messaging specifications are described in the application interface sections.

Other application interfaces shall utilize the VICTORY Data Message (VDM) format, a VICTORY-specific messaging specification. The VDM format is described in the following sections.

17.1 Application Layer Data Encoding <20001–20120725, Pro>

The VDB data messages shall be encoded using Extensible Markup Language (XML) except where performance or implementation constraints make the use of XML prohibitive, or when an accepted standard message format exists. Examples of this would be control loops with real-time constraints or high-rate messages (e.g., digitized video). VICTORY-specific messages shall be encoded in VDMs. The VDM data content and encoding formats are application dependent. The specific formats for the various applications are specified in later sections.

NOTE: *The specification for VDM XML formats includes syntactic and semantic components. The appendix titled "VICTORY Schemas" provides the name of the schema file which contains additional details for the Schema that describes the syntax of the XML language for VDMs. The information in the specific VDM data interface sections in this document provides context, meaning, and additional constraints to the elements and attributes contained in the XML schema. In order to create VDMs that are fully VICTORY-compliant, both the XML schema and the information in specific VDM data interface sections must be taken into account.*

17.2 Application Layer Message Encapsulation <20002–20150331, Pro>

All VDMs shall be encapsulated into the payload of User Datagram Protocol (UDP) datagrams and sent on the VDB using the UDP transport protocol, as defined in IETF RFC 768.

Applications that generate VDMs over IPv4 shall use an appropriate IP address as defined in the section titled "IPv4 Address Space." This includes the use of multicast addresses for multicast routing, which is expected to be the most common routing paradigm used by VDMs.

Datagram fragmentation at the data link level, due to a VDM exceeding the Ethernet maximum transmission unit (MTU) defined for a particular instance of a VDB, shall be handled by the Internet Protocol stack as defined in IETF RFC 791 and IETF RFC 815.

The VDM size shall be limited so as to not cause the total UDP datagram size to exceed the maximum IP packet size as defined in IETF RFC 791.

All VICTORY-specific data shall be encapsulated in the data octets field (commonly referred to as the datagram payload) of the UDP datagram. A VDM shall include a VDM-specific header, described below and represented in the table below. A VDM shall contain at least one VICTORY-specific data message. The maximum number of VICTORY-specific messages that may be contained in a VDM is limited by the maximum allowable size for the UDP datagram payload.

NOTE: *The decision to use UDP for VDMs imposes a maximum size for any VICTORY-specific data message. The maximum size of an IP message is 65,535 bytes. The maximum size of the UDP datagram, subtracting the 20-byte IP header, is 65,515 bytes. The maximum size of the UDP datagram payload, subtracting the 8-byte UDP header, is then 65,507 bytes. Thus, the maximum size for a complete VDM (the encoded data messages plus the VICTORY-specific payload header) is 65,507 bytes. Subtracting the minimum 36-byte VDM header results in a 65,471-byte maximum size for a single XML-encoded data message string.*

Table 217. VDM Header

32-bits			
16-bits		16-bits	
8-bits	8-bits	8-bits	8-bits
“0”	“V”	“D”	“M”
V_MAJ	V_MIN	TS_SB5	TS_SB4
TS_SB3	TS_SB2	TS_SB1	TS_SB0
TS_NSB3	TS_NSB2	TS_NSB1	TS_NSB0
VDM_SEQ_NUMBER			
INTERFACE_ID			
VDM_SIZE		NUM_DATA_MESSAGES	
TYPE		OFFSET	
LENGTH		RESERVED	
+			
+			
+			
TYPE		OFFSET	
LENGTH		RESERVED	
DATA_PAYLOAD • • •			

NOTE: *The size of the data message (DATA_PAYLOAD) is variable. The length of the data message shall not be such that it causes the VDM to exceed the maximum allowable size for the UDP datagram payload.*

17.2.1 Field Byte Ordering

All VDM fields shall be put onto the network in “network byte order” as defined in IETF RFC 2781.

17.2.2 VDM Preamble (32 bits)

The VDM shall be identified by a 32-bit field using the following sequence of four (4) UTF–8 (8-bit Unicode Transformation Format) characters: “0” (0x30), “V” (0x56), “D” (0x44), and “M” (0x4D).

17.2.3 VDM Version Field (16 bits)

The version of the VDM shall be identified via two fields: Major Version (V_MAJ) and Minor Version (V_MIN). The major version of the VDM shall be identified by an 8-bit, unsigned integer greater than or equal to zero and less than 256. The minor version of the VDM shall be identified by an 8-bit, unsigned integer greater than or equal to zero and less than 256. The VDM Version Field indicates the version of the VDM-specific (binary) header.

NOTE: *As of 15 FEB 2018, versions 1.2, 1.3, 1.4, 1.5, 1.6, and 1.7 of the VDM-specific header are identical.*

17.2.4 VDM Timestamp (80 bits)

The VDM shall contain a VICTORY binary timestamp as defined in the section titled "Timestamp Format for Application Layer Data." The Seconds field of the timestamp shall be defined using the fields TS_SB0, TS_SB1, TS_SB2, TS_SB3, TS_SB4, and TS_SB5 in the table above. Where TS_SB5 contains the most significant 8 bits of the timestamp Seconds field, and TS_SB0 contains the least significant 8 bits of the timestamp Seconds field. The Nanoseconds field of the timestamp shall be defined using the fields TS_NSB0, TS_NSB1, TS_NSB2, and TS_NSB3. Where TS_NSB3 contains the most significant 8 bits of the timestamp Nanoseconds field, and TS_NSB0 contains the least significant 8 bits of the timestamp Nanoseconds field.

17.2.5 VDM Sequence Number (32 bits)

The VDM shall contain a monotonically increasing number that indicates the sequence number of the VDM. This number shall be a 32-bit, unsigned integer contained in the VDM_SEQ_NUMBER field. The sequence number shall increase by one (1) for each VDM sent from the sending interface. When the sequence number reaches the maximum value for a 32-bit, unsigned integer, it shall "roll over" to a value of zero.

17.2.6 VDM Interface ID (32 bits)

The VDM shall contain a 32-bit, unsigned integer that identifies the source of the VDM. The interface ID shall be unique within the context of the VDB. The unique ID is contained in the INTERFACE_ID field.

17.2.7 VDM Message Size (16 bits)

The VDM shall contain a 16-bit, unsigned integer that provides the size of the VDM in bytes. The size of the VDM includes both the size of the VDM header and the size of the VDM data payload. The VDM_SIZE field contains this number.

17.2.8 Number of Data Messages (16 bits)

The VDM shall contain a 16-bit, unsigned integer that provides the number of VICTORY-specific data messages contained in the VDM. The NUM_DATA_MESSAGES field contains this number.

17.2.9 VDM Data Index (VDM Encapsulation)

The VDM shall provide an index for any VICTORY-specific data messages contained in the VDM. A data index shall provide the following information: ***data type***, ***offset***, and ***length***.

1. The ***data type*** shall be a 16-bit, unsigned integer contained in the TYPE field.
    a. The acceptable values for this field are described in the "Message Value" column for each row referenced by the "Data Type" column of Table 298 in the appendix titled "VDM Data Type Values."
2. The ***offset*** shall be a 16-bit, unsigned integer contained in the OFFSET field.
    a. The integer indicates an offset (in bytes) from the beginning of the VDM to the beginning of the data message (DATA_PAYLOAD).
3. The ***length*** shall be a 16-bit, unsigned integer contained in the LENGTH field.
    a. The integer provides the length (in bytes) of the data message (DATA_PAYLOAD).

The VDM Data Index shall have a 16-bit reserved field (RESERVED) following the LENGTH field. The VDM shall provide a data index for each VICTORY-specific data message (DATA_PAYLOAD) contained in the VDM.

NOTE: *New VDM data types may be added in future releases of this document, so it is necessary that interfaces designed to sink VDMs must be able to handle unknown data types. The VDM data type is provided in the binary header of the VDM to aid in top-level filtering of VDMs. If an interface implementation receives a VDM of an unknown type, the interface should discard the VDM and not try to retrieve data from that VDM.*

NOTE: *An 'Experimental' data type has been set aside. This data type allows for experimental data to be encapsulated in a VDM. However, this data does not necessarily conform to any VICTORY specified data format.*

17.3 VICTORY Data Types <20003–20210409, Pro>

This section describes data types that are shared between both data and management interfaces.

17.3.1 Heading

1. Heading of the platform, where heading is ordinarily calculated from True North and proceeds clockwise.

   NOTE: *The heading is not necessarily the same as the pointing direction of the platform. The platform could be moving backwards, or slipping.*
2. The heading shall be in units of arc degrees.
3. The heading shall be a real value greater than 0° and less than or equal to 360°.

17.3.2 Speed

1. Speed of the platform, where speed is defined as the instantaneous rate of travel along the heading direction.
2. The speed shall be in units of meters per second.
3. The speed shall be a real value with a range greater than or equal to 0 m/s.

17.3.3 Rate of Climb

1. Rate of climb of the platform, where rate of climb is defined as the rate at which the platform increases its altitude.
2. The rate of climb shall be in units of meters per second.
3. The rate of climb shall be a positive or negative real value.

17.3.4 Roll

1. The roll of the platform, where roll is defined as the rotation angle about the longitudinal (fore-and-aft) axis of the platform.
2. The roll shall be in units of arc degrees.
3. The roll shall be a real value greater than or equal to –180° and less than or equal to 180°, where positive roll is clockwise relative to the horizon.

17.3.5 Pitch

1. The pitch of the platform, where pitch is defined as the rotation angle about the transverse (side-to-side) axis of the platform.
2. The pitch shall be in units of arc degrees.
3. The pitch shall be a real value greater than or equal to –90° and less than or equal to 90°, where positive pitch is nose up relative to the horizon.

17.3.6 Yaw

1. The yaw of the platform, where yaw is defined as the rotation angle about the vertical (top-bottom) axis of the platform.
2. The yaw shall be in units of arc degrees.

3. The yaw shall be a real value greater than or equal to –180° and less than or equal to 180°, where positive yaw is nose right relative to True North.

17.3.7 Latitude

1. The latitude of the platform, where latitude is defined as the angle from the equatorial plane to the perpendicular of the ellipsoid through a given point with northwards treated as positive.
2. The latitude shall be in units of arc degrees.
3. The latitude shall be a real value greater than or equal to –90° and less than or equal to 90°, where north latitudes are positive and south latitudes are negative.

17.3.8 Longitude

1. The longitude of the platform, where longitude is defined as the angle from the prime meridian plane to the meridian plane of a given point with eastward treated as positive.
2. The longitude shall be in units of arc degrees.
3. The longitude shall be a real value greater than –180° and less than or equal to 180°, where east longitudes are positive and west longitudes are negative.

17.3.9 Altitude

1. The altitude of the platform, where altitude is defined as a height, and the chosen reference is mean sea level.

   NOTE: *VICTORY uses the Earth Gravitational Model from 1996 (EGM96) to determine how much the mean sea level varies from the geodetic datum used to calculate altitude.*
2. The altitude shall be in units of meters.
3. The altitude shall be a positive or negative real value.

17.3.10 MGRS

1. The position of the platform represented in the Military Grid Reference System (MGRS).
2. The MGRS shall be a string.

   NOTE: *Refer to Section 3–4 in DMA Technical Manual 8358.1 for more information on the MGRS string format.*

17.3.11 Azimuth

1. The direction of an object of interest from a fixed point (called the origin) on the sensor, expressed as the angle between the projection in the sensor's horizontal mounting plane from the vector designated front or forward by the sensor and the object of interest.
2. The azimuth shall be in units of arc degrees.
3. The azimuth shall be a real value greater than or equal to 0° and less than 360°.
    a. The angle is expressed as 0° when aligned with the forward vector within the mounting plane. It grows positively in a clockwise direction within the horizontal mounting plane when looking downward toward the origin from a vector pointing upward, as defined by the sensor, that is perpendicular to the mounting plane. From the sensor's perspective, the angle grows positively to the right and is 0° when centered.

17.3.12 Elevation

1. The angle between a vector from a fixed point on the sensor to the object of interest and the sensor mounting plane that contains the fixed point on the sensor.
2. The elevation shall be in units of arc degrees.
3. The elevation shall be a real value greater than or equal to –90° and less than or equal to 90°.
    a. The angle is expressed as 0° when the object of interest is on the sensor mounting plane that contains the fixed point on the sensor. Positive angles are above the sensor mounting plane that contains the fixed point on the sensor, and negative angles are below the sensor mounting plane that contains the fixed point on the sensor.

17.3.13 Range

1. The length of a vector from a fixed point on the sensor to the object of interest.
2. The range shall be in units of meters.
3. The range shall be a real value greater than or equal to zero.

17.3.14 True Azimuth

1. The direction of an object of interest from a fixed point on the vehicle (the vehicle origin), expressed as the angle between the projection in the horizontal plane (parallel to the horizon), between the vector pointing north (from True North) and the object of interest.
2. The true azimuth shall be in units of arc degrees.
3. The true azimuth shall be a real value greater than or equal to 0° and less than 360°.
    a. The angle is expressed as 0° when aligned with the vector pointing north from True North. It grows positively in a clockwise direction within the horizontal plane.

17.3.15 Absolute Elevation

1. Absolute elevation is defined as the angle between a vector from a fixed point on the vehicle (the vehicle origin) pointing in the direction of the object of interest and the horizontal plane.
2. The absolute elevation shall be in units of arc degrees.
3. The absolute elevation shall be a real value greater than or equal to –90° and less than or equal to 90°.
    a. The angle is expressed as 0° when the object of interest is on the horizontal plane that contains the fixed point on the vehicle (the vehicle origin). Positive angles are above the horizontal plane that contains the fixed point on the vehicle, and negative angles are below the horizontal plane that contains the fixed point on the vehicle.

17.3.16 Relative Cartesian Coordinates

Positions with respect to the platform can be defined uniquely by X, Y, and Z, where:

1. X, Y, and Z are measures along orthogonal axes oriented in a “right hand” configuration.
2. Positive X shall point forward down the centerline of the platform.
3. Positive Y shall point 90° left of forward (port).
4. Z shall be orthogonal to X and Y, pointing upward in the platform’s “at rest” stance.
5. X, Y, and Z shall be positive or negative real values.
6. X, Y, and Z shall be in units of meters.

17.3.17 String Bounds

1. The character restrictions of the string data type.
2. The string shall be comprised of only printable ASCII characters.
3. The length of the string shall be greater than or equal to one character and less than or equal to 256 characters.

17.3.18 URI Bounds

1. The character restrictions of the Uniform Resource Identifier (URI) data type.
2. The URI shall be a string in a format defined in IETF RFC 3986.
3. The URI shall be comprised of only printable ASCII characters.
4. The length of the URI shall be greater than or equal to one character and less than or equal to 256 characters.

17.4 Timestamp Format for Application Layer Data <20004–20140630, Pro>

VICTORY data messages are, in general, to be encoded as XML strings and encapsulated into UDP datagrams for transport on the VDB. The VDMs include timestamps in a textual format. Time sources will generate timestamps in a binary format, and computers require a binary format in order to interpret the timestamps correctly. The result is that there will be two timestamp formats: one XML encoded, and the other binary encoded.

It is desirable for these two timestamp formats to have the same precision. The VICTORY epoch shall be the same as the Unix epoch (IEEE P1003.1) and defined as 00:00:00.0 UTC on January 1, 1970. Coordinated Universal Time (UTC) shall be used, otherwise known as Zulu time (Z).

Timestamps represented as XML text shall be encoded using the **dateTime** XML schema data type as defined in Section 3.2.7 of "XML Schema Part 2: Datatypes Second Edition." The XML encoded timestamp shall be further restricted to the following format: 'YYYY'-'MM'-'DD'T'hh':'mm':'ss'.'sssssssss'Z. The restriction is that it shall include at most nine decimal digits in the Nanoseconds representation, and shall always be specified Zulu time with the "Z" suffix. The table below defines the fields for the **dateTime** data type.

Table 218. dateTime Data Type

Field	Definition	Minimum	Maximum
YYYY	Year	1970	–
MM	Month	1	12
DD	Day	1	31
T	Time of day follows	–	–
hh	Hour	0	24
mm	Minutes	0	59
ss	Seconds	0	60
sssssssss	Nanoseconds	0	999999999
Z	Zulu	–	–

Timestamps represented in binary shall be encoded as defined in the section titled "Platform Systems Interfaces Configuration & Control from Network" in Section 5.3.2 of IEEE 1588 version 2. The IEEE 1588 timestamp represents TAI (International Atomic Time) (not UTC) time with Seconds and Nanoseconds fields. The Seconds field shall represent the number of whole seconds since the VICTORY epoch, counting leap seconds, and shall be represented with a 48-bit unsigned integer. The Nanoseconds field shall represent the decimal fractional portion of the timestamp in units of nanoseconds, and shall be represented with a 32-bit unsigned integer. The Nanoseconds field shall always be less than 10^9.

The table below provides examples for the two timestamp formats.

Table 219. Timestamp Examples

Binary Format (in Hexadecimal)			
Date/Time	**dateTime**	**Seconds**	**Nanoseconds**
January 1, 1970 00:00:00 UTC	1970–01–01T00:00:00.000000000Z	0x00000000000A	0x00000000
June 21, 1974 22:30:00 UTC	1974–06–21T22:30:00.000000000Z	0x00000868CC75	0x00000000
June 14, 2002 19:00:12.001 UTC	2002–06–14T19:00:12.001000000Z	0x00003D0A3D5C	0x000F4240

Binary Format (in Hexadecimal)			
Date/Time	**dateTime**	**Seconds**	**Nanoseconds**
October 16, 1992 11:15:45 932ns	1992–10–16T11:15:45.000000932Z	0x00002ADEA47C	0x000003A4

The conversion from textual to binary timestamps and back again shall be non-ambiguous, and successive conversions back and forth in either direction for a particular timestamp shall not alter the time represented by the timestamp. The method of conversion between textual and binary timestamps shall adhere to steps described in the sections titled "Conversion from Textual to Binary Timestamps" and "Conversion from Binary to Textual Timestamps."

17.5 Conversion from Textual to Binary Timestamps <20005–20120725, Pro>

1. Convert the dateTime string, without the fractional seconds, into the appropriate number of whole seconds, not including leap seconds, since the VICTORY epoch. A value of 60 in the Seconds field, indicates that we are IN a leap second, and shall be treated as 59 for the purposes of this step of the conversion.
2. Convert whole seconds from UTC to TAI, according to the system leapfile in the appendix titled "Leap Seconds Table From NIST," by adding the indicated offset on the line starting with the largest number less than the UTC time + 2208988800. If the Seconds field above was initially 60, then add 1 to the resulting TAI time.
3. Store the number of seconds in the 48-bit unsigned integer Seconds field.
4. Extract the optional fractional Seconds part of the dateTime string, minus the "." prefix. This will be the fractional Seconds string.
5. Pad or truncate the fractional Seconds string to a total of nine characters, appending "0"s if the number of characters is less than nine, and ignoring any characters past the 9th. This will be the Nanoseconds string.
6. Convert the resulting Nanoseconds string into an unsigned integer, and store that in the 32-bit unsigned integer Nanoseconds field.

17.6 Conversion from Binary to Textual Timestamps <20006–20140630, Pro>

1. Convert the Nanoseconds field into a nine-character string version of the number of nanoseconds. This will be the fractional Seconds string, and will be nine characters long.
2. Convert seconds from TAI to UTC according to the system leapfile:
    a. If TAI + 2208988800 – the indicated offset on any line is equal to the first number on the following line, then subtract 1 and the offset on the first line from TAI before converting to the UTC string. Finally, change the Seconds string to 60 after Step 3 to indicate the leap second.
    b. Otherwise, find the line with the greatest offset where TAI time + 2208988800 – the indicated offset is greater than or equal to the first number on the line. Subtract this offset from TAI before converting to the UTC string.
3. Concatenate the string created in Step 2 with a "." followed by the Nanoseconds string created in Step 1.
4. Add a "Z" to the end of the string created in Step 3. This is the dateTime string.

The tables below provide the proper XML to binary timestamp conversion over a leap second for both Unix time and NTP. The "?"s indicate that the exact values are unknown since Unix Time and NTP may either repeat the previous second or halt time for a second, depending on implementation. PTP/TAI time is continuous. The "half seconds" are added for explanatory purposes, and would be accounted for in the Nanoseconds field.

Table 220. Leap Seconds Conversion – Unix Time

dateTime	Unix Time	Offset	TAI,PTP	TAI,PTP(hex)
2008–12–31T23:59.59.000000000Z	1230767999	33	1230768032	0x0000495C07A0
2008–12–31T23:59.59.500000000Z	1230767999.5	33	1230768033	0x0000495C07A0
2008–12–31T23:59.60.000000000Z	?	33	1230768033	0x0000495C07A1
2008–12–31T23:59.60.500000000Z	?	33	1230768034	0x0000495C07A1

dateTime	Unix Time	Offset	TAI,PTP	TAI,PTP(hex)
2009–01–01T00:00.00.000000000Z	1230768000	34	1230768034	0x0000495C07A2
2009–01–01T00:00.00.500000000Z	1230768000.5	34	1230768035	0x0000495C07A2

Table 221. Leap Seconds Conversion – NTP

dateTime	NTP	Offset	TAI,PTP	TAI,PTP(hex)
2008–12–31T23:59.59.000000000Z	3439756799	33	1230768032	0x0000495C07A0
2008–12–31T23:59.59.500000000Z	3439756799.5	33	1230768033	0x0000495C07A0
2008–12–31T23:59.60.000000000Z	?	33	1230768033	0x0000495C07A1
2008–12–31T23:59.60.500000000Z	?	33	1230768034	0x0000495C07A1
2009–01–01T00:00.00.000000000Z	3439756800	34	1230768034	0x0000495C07A2
2009–01–01T00:00.00.500000000Z	3439756800.5	34	1230768035	0x0000495C07A2

18. C4ISR SYSTEMS INTERFACES

The following sections describe the C4ISR systems that the VDB will integrate. The VDB will serve as the primary transport mechanism for data that is shared between components of these systems. The VDB, therefore, must serve as more than just an interface that these components can use. It is to be a critical part of each C4ISR system, and these systems will utilize shared service interfaces provided by the VDB, wherever possible, rather than implementing their own stovepipe solutions.

18.1 VDB Shared Service Interfaces

The VDB includes several shared service interfaces that are made available to the VDB nodes to reduce duplicated functionality. The standard will identify shared data services and will standardize their interfaces. The standard will not specify how the interfaces are to be implemented: how many and which software components will implement these interfaces, which processing units they will run on, or the programming language for implementation. The following will describe each of the shared service interfaces and the interfaces to be standardized.

18.1.1 Direction of Travel Interfaces

The Direction of Travel Interfaces provide information regarding the current direction of travel of the vehicle. The data will be available to all VDB network nodes that wish to subscribe to the Direction of Travel Data Interface. This logical interface is implemented by the VDB, most likely by a software component. Whether it is a stand-alone software component, or is implemented as part of an aggregate VDB, is up to the implementer.

18.1.1.1 Direction of Travel Data Specification <20007–20120725, Pro>

18.1.1.1.1 Direction of Travel Message Content

The Direction of Travel Data Interface provides information regarding the current direction of travel of the vehicle. This information will be available to all VDB network nodes that require direction of travel data.

1. ***Heading***
    a. A detailed description of the heading data is provided in the section titled "Heading" under "VICTORY Data Types."

    > ***NOTE:*** *The heading is not necessarily the same as the pointing direction of the platform. The platform could be moving backwards, or slipping.*

    b. The heading shall be provided as part of the direction of travel message.
2. ***Heading Valid***
    a. An indication that the heading data inside the Direction of Travel VDM is valid.
    b. A Boolean value of True indicates the data is valid, False otherwise.
    c. The valid flag shall be provided as part of the direction of travel message.
3. ***Heading Uncertainty***
    a. The uncertainty of the heading measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
    b. The measure of uncertainty shall be in arc degrees.
    c. The uncertainty shall be a real value at 1 σ.
    d. If the heading valid flag is True, the heading uncertainty shall be provided as part of the direction of travel message.
    e. If the heading valid flag is False, the heading uncertainty shall not be provided as part of the direction of travel message.
4. ***Heading Estimated***
    a. An indication that the heading data inside the Direction of Travel VDM is estimated.
    b. A Boolean value of True indicates the data is estimated, False otherwise.

   c. If the heading valid flag is True, the heading estimated flag shall be provided as part of the direction of travel message.
   d. If the heading valid flag is False, the heading estimated flag shall not be provided as part of the direction of travel message.
5. ***Speed***
   a. A detailed description of the speed data is provided in the section titled "Speed" under "VICTORY Data Types."
   b. The speed may be provided as part of the direction of travel message.
6. ***Speed Valid***
   a. An indication that the speed data inside the Direction of Travel VDM is valid.
   b. A Boolean value of True indicates the data is valid, False otherwise.
   c. If speed is provided, the speed valid flag shall be provided as part of the direction of travel message.
7. ***Speed Uncertainty***
   a. The uncertainty of the speed measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
   b. The measure of uncertainty shall be in meters per second.
   c. The uncertainty shall be a real value at 1 σ.
   d. If the speed valid flag is True, the speed uncertainty shall be provided as part of the direction of travel message.
   e. If the speed valid flag is False, the speed uncertainty shall not be provided as part of the direction of travel message.
8. ***Speed Estimated***
   a. An indication that the speed data inside the Direction of Travel VDM is estimated.
   b. A Boolean value of True indicates the data is estimated, False otherwise.
   c. If the speed valid flag is True, the speed estimated flag shall be provided as part of the direction of travel message.
   d. If the speed valid flag is False, the speed estimated flag shall not be provided as part of the direction of travel message.
9. ***Rate of Climb***
   a. A detailed description of the rate of climb data is provided in the section titled "Rate of Climb" under "VICTORY Data Types."
   b. The rate of climb may be provided as part of the direction of travel message.
10. ***Rate of Climb Valid***
   a. An indication that the rate of climb data inside the Direction of Travel VDM is valid.
   b. A Boolean value of True indicates the data is valid, False otherwise.
   c. If rate of climb is provided, the rate of climb valid flag shall be provided as part of the direction of travel message.
11. ***Rate of Climb Uncertainty***
   a. The uncertainty of the rate of climb measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
   b. The measure of uncertainty shall be in meters per second.
   c. The uncertainty shall be a real value at 1 σ.
   d. If the rate of climb valid flag is True, the rate of climb uncertainty shall be provided as part of the direction of travel message.
   e. If the rate of climb valid flag is False, the rate of climb uncertainty shall not be provided as part of the direction of travel message.
12. ***Rate of Climb Estimated***
   a. An indication that the rate of climb data inside the Direction of Travel VDM is estimated.
   b. A Boolean value of True indicates the data is estimated, False otherwise.
   c. If the rate of climb valid flag is True, the rate of climb estimated flag shall be provided as part of the direction of travel message.
   d. If the rate of climb valid flag is False, the rate of climb estimated flag shall not be provided as part of the direction of travel message.
13. ***Timestamp***
   a. A timestamp corresponding to the time at which the direction of travel data was sampled.
   b. The timestamp shall be in the format as described in the section titled "Timestamp Format for Application Layer Data."
   c. The Timestamp shall be provided as part of the direction of travel message.
14. ***Sequence Number***

NOTE: *The sequence number is intended to be used to detect messages that did not arrive or were dropped.*

a. A monotonically increasing number that indicates the sequence number of the Direction of Travel message sent from a particular Direction of Travel Data Interface.
b. The sequence number shall increase by one (1) for each Direction of Travel message sent from the interface.
c. When the sequence number reaches its maximum value, it shall roll over to its minimum value.
d. The sequence number shall be provided as part of the direction of travel message.

15. ***Interface ID***
    a. An identification number, unique within the scope of the VDB, used to identify a particular Direction of Travel Data Interface.
    b. The interface ID shall be an integer value greater than or equal to zero.
    c. The interface ID shall be provided as part of the direction of travel message.

18.1.1.1.2 Direction of Travel Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as a Direction of Travel VDM. The appendix titled "VICTORY Schemas" provides the name of the schema file. The file describes the Direction of Travel VDM, provides additional details for the VICTORY Data Types Schema, and describes the Direction of Travel type that generates the XML string used by the Direction of Travel VDM. An example Direction of Travel VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:vm="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <directionOfTravelMessage interfaceId="214748364"
        timestamp="2006-05-04T18:13:51.0Z"
        sequenceNumber="214748364">
        <vdt:heading valid="true" uncertainty="0.001"
            estimated="false">38.898646</vdt:heading>
        <vdt:speed valid="false">50</vdt:speed>
    </directionOfTravelMessage>
</vdm:messages>
```

Figure 16. Example Direction of Travel VDM

18.1.1.2 Direction of Travel Data Management Parameters <15001–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the direction of travel data management parameters. A textual description of the direction of travel data management parameters is shown below.

1. ***Direction of Travel***

NOTE: *Though this management interface provides operations by which to get and set data defined for this component, it is the intent that the data interface of this component be the primary mechanism for receiving data from this component. The Get operation from the management interface does not contain certain information available from the data interface used for determining the validity and accuracy of the data. Please refer to Section 17 for more information concerning the data interface and the methods by which data is made available on the network. The intent of the Get operation is for the retrieval of data in-between the defined update rate for the data interface of this component. The Set method has been defined as a way by which one could pre-populate the component's data when the component is not presently generating data. The Set method should be used sparingly and should not be part of the normal use of this component.*

a. This parameter indicates the direction of travel of the platform.

b. Retrieving the direction of travel allows VDB network nodes to obtain a direction of travel update without having to wait for the next update period.
c. The contents of the direction of travel are described in the section titled "Direction of Travel" under "Direction of Travel Complex Types."
d. The following errors are defined for the direction of travel parameter:
    i. Setting any of the heading, speed, or rate of climb values outside the range defined by this specification document may result in an OUT_OF_RANGE error.
    ii. Setting any of the heading, speed, or rate of climb values inside the range defined by this specification document but unrecognized by the Direction of Travel Data Interface shall result in a NEAREST_VALUE error.
        1. The value shall be changed to the nearest value known by the Direction of Travel Data Interface and shall be returned in the response.
e. Operations include Get() and Set().

2. ***Nominal Data Available***
    a. This parameter indicates which direction of travel measurements are being provided by the Direction of Travel Data Interface.
    b. The contents of this parameter are described in the section titled "Nominal Data Available" under "Direction of Travel Complex Types."
    c. Operations include Get().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the direction of travel data management parameters along with a description of additional errors that were not specified but may be returned.*

18.1.1.3 Direction of Travel Complex Types <20012–20210409, Pro>

NOTE: *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

1. ***Direction of Travel***
    a. **Heading**
        i. A detailed description of the heading data is provided in the section titled "Heading" under "VICTORY Data Types."

        NOTE: *The heading is not necessarily the same as the pointing direction of the platform. The platform could be moving backwards, or slipping.*

        ii. The heading may be provided as part of the direction of travel.
    b. **Speed**
        i. A detailed description of the speed data is provided in the section titled "Speed" under "VICTORY Data Types."
        ii. The speed may be provided as part of the direction of travel.
    c. **Rate of Climb**
        i. A detailed description of the rate of climb data is provided in the section titled "Rate of Climb" under "VICTORY Data Types."
        ii. The rate of climb may be provided as part of the direction of travel.
2. ***Nominal Data Available***
    a. **Heading Available**
        i. An indication that the heading data inside the direction of travel VDM is both available and valid.
        ii. A Boolean value of True indicates the data is available and valid, False otherwise.
        iii. The heading available flag shall be provided as part of the nominal data available.
    b. **Speed Available**
        i. An indication that the speed data inside the direction of travel VDM is both available and valid.
        ii. A Boolean value of True indicates the data is available and valid, False otherwise.
        iii. The speed available flag shall be provided as part of the nominal data available.
    c. **Rate of Climb Available**
        i. An indication that the rate of climb data inside the Direction of Travel VDM is both available and valid.
        ii. A Boolean value of True indicates the data is available and valid, False otherwise.

iii. The rate of climb available flag shall be provided as part of the nominal data available.

d. **Description**
   i. A detailed description of the direction of travel nominal capabilities.
   ii. The bounds of the description are described in the section titled "String Bounds" under "VICTORY Data Types."
   iii. The description may be provided as part of the nominal data available.

18.1.2 Orientation Interfaces

The Orientation Interfaces provide information regarding the current orientation of the vehicle with respect to a predetermined frame of reference. The data will be available to all VDB network nodes that wish to subscribe to the Orientation Data Interface. This logical interface is implemented by the VDB, most likely by a software component. Whether it is a stand-alone software component or is implemented as part of an aggregate VDB service is up to the implementer.

18.1.2.1 Orientation Data Specification <20008–20120725, Pro>

18.1.2.1.1 Orientation Message Content

The Orientation Data Interface provides information regarding the current orientation of the vehicle. This information will be available to all VDB network nodes that require orientation data.

1. ***Roll***
   a. A detailed description of the roll data is provided in the section titled "Roll" under "VICTORY Data Types."
   b. The roll shall be provided as part of the orientation message.
2. ***Roll Valid***
   a. An indication that the roll data inside the Orientation VDM is valid.
   b. A Boolean value of True indicates the data is valid, False otherwise.
   c. The roll valid flag shall be provided as part of the orientation message.
3. ***Roll Uncertainty***
   a. The uncertainty of the roll measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
   b. The measure of uncertainty shall be in arc degrees.
   c. The uncertainty shall be a real value at 1 σ.
   d. If the roll valid flag is True, the roll uncertainty shall be provided as part of the orientation message.
   e. If the roll valid flag is False, the roll uncertainty shall not be provided as part of the orientation message.
4. ***Roll Estimated***
   a. An indication that the roll data inside the Orientation VDM is estimated.
   b. A Boolean value of True indicates the data is estimated, False otherwise.
   c. If the roll valid flag is True, the roll estimated flag shall be provided as part of the orientation message.
   d. If the roll valid flag is False, the roll estimated flag shall not be provided as part of the orientation message.
5. ***Pitch***
   a. A detailed description of the pitch data is provided in the section titled "Pitch" under "VICTORY Data Types."
   b. The pitch shall be provided as part of the orientation message.
6. ***Pitch Valid***
   a. An indication that the pitch data inside the Orientation VDM is valid.
   b. A Boolean value of True indicates the data is valid, False otherwise.
   c. The pitch valid flag shall be provided as part of the orientation message.
7. ***Pitch Uncertainty***
   a. The uncertainty of the pitch measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
   b. The measure of uncertainty shall be in arc degrees.
   c. The uncertainty shall be a real value at 1 σ.
   d. If the pitch valid flag is True, the pitch uncertainty shall be provided as part of the orientation message.
   e. If the pitch valid flag is False, the pitch uncertainty shall not be provided as part of the orientation message.
8. ***Pitch Estimated***
   a. An indication that the pitch data inside the Orientation VDM is estimated.
   b. A Boolean value of True indicates the data is estimated, False otherwise.

c. If the pitch valid flag is True, the pitch estimated flag shall be provided as part of the orientation message.
d. If the pitch valid flag is False, the pitch estimated flag shall not be provided as part of the orientation message.

9. ***Yaw***
   a. A detailed description of the yaw data is provided in the section titled "Yaw" under "VICTORY Data Types."
   b. The yaw shall be provided as part of the orientation message.
10. ***Yaw Valid***
   a. An indication that the yaw data inside the Orientation VDM is valid.
   b. A Boolean value of True indicates the data is valid, False otherwise.
   c. The yaw valid flag shall be provided as part of the orientation message.
11. ***Yaw Uncertainty***
   a. The uncertainty of the yaw measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
   b. The measure of uncertainty shall be in arc degrees.
   c. The uncertainty shall be a real value at 1 σ.
   d. If the yaw valid flag is True, the yaw uncertainty shall be provided as part of the orientation message.
   e. If the yaw valid flag is False, the yaw uncertainty shall not be provided as part of the orientation message.
12. ***Yaw Estimated***
   a. An indication that the yaw data inside the Orientation VDM is estimated.
   b. A Boolean value of True indicates the data is estimated, False otherwise.
   c. If the yaw valid flag is True, the yaw estimated flag shall be provided as part of the orientation message.
   d. If the yaw valid flag is False, the yaw estimated flag shall not be provided as part of the orientation message.
13. ***Timestamp***
   a. A timestamp corresponding to the time at which the orientation data was sampled.
   b. The timestamp shall be in the format as described in the section titled "Timestamp Format for Application Layer Data."
   c. The timestamp shall be provided as part of the orientation message.
14. ***Sequence Number***
   a. A monotonically increasing number that indicates the sequence number of the Orientation message sent from a particular Orientation Data Interface.
   b. The sequence number shall increase by one (1) for each Orientation message sent from the interface.
   c. When the sequence number reaches its maximum value, it shall roll over to its minimum value.

      NOTE: *The sequence number is intended to be used to detect messages that did not arrive or were dropped.*

   d. The sequence number shall be provided as part of the orientation message.
15. ***Interface ID***
   a. An identification number, unique within the scope of the VDB, used to identify a particular Orientation Data Interface.
   b. The interface ID shall be an integer value greater than or equal to zero.
   c. The interface ID shall be provided as part of the orientation message.

18.1.2.1.2 Orientation Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Orientation VDM. The appendix titled "VICTORY Schemas" provides the name of the schema file. The file describes the Orientation VDM, provides additional details for the VICTORY Data Types Schema, and describes the Orientation type that generates the XML string used by the Orientation VDM. An example Orientation VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:vm="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <orientationMessage interfaceId="20"
        timestamp="2006-05-04T18:13:51.0Z"
```

```
        sequenceNumber="50">
        <vdt:roll valid="true" uncertainty="0" estimated="false">5.1332</vdt:roll>
        <vdt:pitch valid="true" uncertainty="0" estimated="false">10.5832</vdt:pitch>
        <vdt:yaw valid="true" uncertainty="0" estimated="false">29.1020</vdt:yaw>
    </orientationMessage>
</vdm:messages>
```

Figure 17. Example Orientation VDM

18.1.2.2 Orientation Data Management Parameters <15002–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the orientation data management parameters. A textual description of the orientation data management parameters is shown below.

1. ***Orientation***

   NOTE: *Though this management interface provides operations by which to get and set data defined for this component, it is the intent that the data interface of this component be the primary mechanism for receiving data from this component. The Get operation from the management interface does not contain certain information available from the data interface used for determining the validity and accuracy of the data. Please refer to Section 17 for more information concerning the data interface and the methods by which data is made available on the network. The intent of the Get operation is for the retrieval of data in-between the defined update rate for the data interface of this component. The Set method has been defined as a way by which one could pre-populate the component's data when the component is not presently generating data. The Set method should be used sparingly and should not be part of the normal use of this component.*

   a. This parameter indicates the orientation of the platform.
   b. Retrieving the orientation allows VDB network nodes to obtain an orientation update without having to wait for the next update period.
   c. The contents of this parameter are described in the section titled "Orientation" under "Orientation Complex Types."
   d. The following errors are defined for the orientation parameter:
      i. Setting any of the roll, pitch, or yaw values outside the range defined by this specification document may result in an OUT_OF_RANGE error.
      ii. Setting any of the roll, pitch, or yaw values inside the range defined by this specification document but unrecognized by the Orientation Data Interface shall result in a NEAREST_VALUE error.
          1. The value shall be changed to the nearest value known by the Orientation Data Interface and shall be returned in the response.
   e. Operations include Get() and Set().
2. ***Nominal Data Available***
   a. This parameter indicates which orientation measurements are being provided by the Orientation Data Interface.
   b. The contents of this parameter are described in the section titled "Nominal Data Available" under "Orientation Complex Types."
   c. Operations include Get().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the orientation data management parameters along with a description of additional errors that were not specified but may be returned.*

18.1.2.3 Orientation Complex Types <20013–20210409, Pro>

NOTE: *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

1. ***Orientation***
   a. **Roll**

i. A detailed description of the roll data is provided in the section titled "Roll" under "VICTORY Data Types."
ii. The roll may be provided as part of the orientation.

b. **Pitch**
i. A detailed description of the pitch data is provided in the section titled "Pitch" under "VICTORY Data Types."
ii. The pitch may be provided as part of the orientation.

c. **Yaw**
i. A detailed description of the yaw data is provided in the section titled "Yaw" under "VICTORY Data Types."
ii. The yaw may be provided as part of the orientation.

2. ***Nominal Data Available***
a. **Roll Available**
i. An indication that the roll data inside the orientation VDM is both available and valid.
ii. A Boolean value of True indicates the data is available and valid, False otherwise.
iii. The roll available flag shall be provided as part of the nominal data available.

b. **Pitch Available**
i. An indication that the pitch data inside the orientation VDM is both available and valid.
ii. A Boolean value of True indicates the data is available and valid, False otherwise.
iii. The pitch available flag shall be provided as part of the nominal data available.

c. **Yaw Available**
i. An indication that the yaw data inside the orientation VDM is both available and valid.
ii. A Boolean value of True indicates the data is available and valid, False otherwise.
iii. The yaw available flag shall be provided as part of the nominal data available.

d. **Description**
i. A detailed description of the orientation nominal capabilities.
ii. The bounds of the description are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The description may be provided as part of the nominal data available.

18.1.3 Position Interfaces

The Position Interfaces provide information regarding the current physical location of the vehicle with respect to a predetermined reference. This information is typically described in a geospatial coordinate system such as latitude, longitude, and altitude, information usually obtained from a device such as a GPS receiver or an inertial navigation system (INS) unit. The information will be available to all VDB network nodes that wish to subscribe to the Position Data Interface. This logical interface is implemented by the VDB, most likely by a software component. Whether it is a stand-alone software component, or is implemented as part of an aggregate VDB service, is up to the implementer.

18.1.3.1 Position Data Specification <20009–20120725, Pro>

18.1.3.1.1 Position Message Content

The Position Data Interface provides information regarding the current position of the vehicle. This information will be available to all VDB network nodes that require position data.

1. ***Latitude***
a. A detailed description of the latitude data is provided in the section titled "Latitude" under "VICTORY Data Types."
b. The latitude shall be provided as part of the position message.
2. ***Latitude Valid***
a. An indication that the latitude data inside the Position VDM is valid.
b. A Boolean value of True indicates the data is valid, False otherwise.
c. The latitude valid flag shall be provided as part of the position message.
3. ***Latitude Uncertainty***
a. The uncertainty of the latitude measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.

   b. The measure of uncertainty shall be in arc degrees.
   c. The uncertainty shall be a real value at 1 σ.
   d. If the latitude valid flag is True, the latitude uncertainty shall be provided as part of the position message.
   e. If the latitude valid flag is False, the latitude uncertainty shall not be provided as part of the position message.
4. ***Latitude Estimated***
   a. An indication that the latitude data inside the Position VDM is estimated.
   b. A Boolean value of True indicates the data is estimated, False otherwise.
   c. If the latitude valid flag is True, the latitude estimated flag shall be provided as part of the position message.
   d. If the latitude valid flag is False, the latitude estimated flag shall not be provided as part of the position message.
5. ***Longitude***
   a. A detailed description of the longitude data is provided in the section titled "Longitude" under "VICTORY Data Types."
   b. The longitude shall be provided as part of the position message.
6. ***Longitude Valid***
   a. An indication that the longitude data inside the Position VDM is valid.
   b. A Boolean value of True indicates the data is valid, False otherwise.
   c. The longitude valid flag shall be provided as part of the position message.
7. ***Longitude Uncertainty***
   a. The uncertainty of the longitude measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
   b. The measure of uncertainty shall be in arc degrees.
   c. The uncertainty shall be a real value at 1 σ.
   d. If the longitude valid flag is True, the longitude uncertainty shall be provided as part of the position message.
   e. If the longitude valid flag is False, the longitude uncertainty shall not be provided as part of the position message.
8. ***Longitude Estimated***
   a. An indication that the longitude data inside the Position VDM is estimated.
   b. A Boolean value of True indicates the data is estimated, False otherwise.
   c. If the longitude valid flag is True, the longitude estimated flag shall be provided as part of the position message.
   d. If the longitude valid flag is False, the longitude estimated flag shall not be provided as part of the position message.
9. ***Altitude***
   a. A detailed description of the altitude data is provided in the section titled "Altitude" under "VICTORY Data Types."
   b. The altitude shall be provided as part of the position message.
10. ***Altitude Valid***
   a. An indication that the altitude data inside the Position VDM is valid.
   b. A Boolean value of True indicates the data is valid, False otherwise.
   c. The altitude valid flag shall be provided as part of the position message.
11. ***Altitude Uncertainty***
   a. The uncertainty of the altitude measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
   b. The measure of uncertainty shall be in meters.
   c. The uncertainty shall be a real value at 1 σ.
   d. If the altitude valid flag is True, the altitude uncertainty shall be provided as part of the position message.
   e. If the altitude valid flag is False, the altitude uncertainty shall not be provided as part of the position message.
12. ***Altitude Estimated***
   a. An indication that the altitude data inside the Position VDM is estimated.
   b. A Boolean value of True indicates the data is estimated, False otherwise.
   c. If the altitude valid flag is True, the altitude estimated flag shall be provided as part of the position message.
   d. If the altitude valid flag is False, the altitude estimated flag shall not be provided as part of the position message.
13. ***Timestamp***
   a. A timestamp corresponding to the time at which the position data was sampled.
   b. The timestamp shall be in the format as described in the section titled "Timestamp Format for Application Layer Data."
   c. The timestamp shall be provided as part of the position message.
14. ***Sequence Number***

a. A monotonically increasing number that indicates the sequence number of the Position message sent from a particular Position Data Interface.
b. The sequence number shall increase by one (1) for each Position message sent from the interface.
c. When the sequence number reaches its maximum value, it shall roll over to its minimum value.

> ***NOTE:*** *The sequence number is intended to be used to detect messages that did not arrive or were dropped.*

d. The sequence number shall be provided as part of the position message.

15. ***Interface ID***
    a. An identification number, unique within the scope of the VDB, used to identify a particular Position Data Interface.
    b. The interface ID shall be an integer value greater than or equal to zero.
    c. The interface ID shall be provided as part of the position message.
16. ***MGRS Position***
    a. A detailed description of the MGRS data is provided in the section titled "MGRS" under "VICTORY Data Types."
    b. The MGRS position should be provided as part of the position message.

18.1.3.1.2 Position Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as a Position VDM. The appendix titled "VICTORY Schemas" provides the name of the schema file. The file describes the Position VDM, provides additional details for the VICTORY Data Types Schema, and describes the Position type that generates the XML string used by the Position VDM. An example Position VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:vm="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <positionMessage interfaceId="25"
        timestamp="2006-05-04T18:13:51.0Z"
        sequenceNumber="40">
        <vdt:latitude valid="true" uncertainty="0.001"
            estimated="false">38.898648</vdt:latitude>
        <vdt:longitude valid="true" uncertainty="0.001"
            estimated="false">-77.037692</vdt:longitude>
        <vdt:altitude valid="false">2000</vdt:altitude>
        <vdt:MGRS>4QFJ1234567890</vdt:MGRS>
    </positionMessage>
</vdm:messages>
```

Figure 18. Example Position VDM

18.1.3.2 Position Data Management Parameters <15003–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the position data management parameters. A textual description of the position data management parameters is shown below.

1. ***Geodetic Datum***
    a. This parameter identifies a list of references from which position measurements are made.
    b. Each geodetic datum shall be a string chosen from the following list of Datum Code Values:
        i. WGS_84

c. Operations include Get().

2. ***Position***

> ***NOTE:*** *Though this management interface provides operations by which to get and set data defined for this component, it is the intent that the data interface of this component be the primary mechanism for receiving data from this component. The Get operation from the management interface does not contain certain information available from the data interface used for determining the validity and accuracy of the data. Please refer to Section 17 for more information concerning the data interface and the methods by which data is made available on the network. The intent of the Get operation is for the retrieval of data in-between the defined update rate for the data interface of this component. The Set method has been defined as a way by which one could pre-populate the component's data when the component is not presently generating data. The Set method should be used sparingly and should not be part of the normal use of this component.*

   a. This parameter indicates the position of the platform.
   b. Retrieving the position allows VDB network nodes to obtain a position update without having to wait for the next update period.
   c. The contents of this parameter are described in the section titled "Position" under "Position Complex Types."
   d. The following errors are defined for the position parameter:
      i. Setting any of the latitude, longitude, altitude, or MGRS values outside the range defined by this specification document may result in an OUT_OF_RANGE error.
      ii. Setting any of the latitude, longitude, altitude, or MGRS values inside the range defined by this specification document but unrecognized by the Position Data Interface shall result in a NEAREST_VALUE error.
         1. The value shall be changed to the nearest value known by the Position Data Interface and shall be returned in the response.
   e. Operations include Get() and Set().

> ***NOTE:*** *Set Position, when originally written, was intended for initializing the Position Service. Because it is described as "volatile," any Set Position request would likely be overwritten and updated by the data provided by a connected device. It is recommended that Set Position only be used to initialize the Position Service.*

3. ***Nominal Data Available***
   a. This parameter indicates which position measurements are being provided by the Position Data Interface.
   b. The contents of this parameter are described in the section titled "Nominal Data Available" under "Position Complex Types."
   c. Operations include Get().

> ***NOTE:*** *The section titled "Management Interface Error Types" provides additional details on the errors defined for the position data management parameters along with a description of additional errors that were not specified but may be returned.*

18.1.3.3 Position Complex Types <20014–20210409, Pro>

> ***NOTE:*** *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

1. ***Position***
   a. **Latitude**
      i. A detailed description of the latitude data is provided in the section titled "Latitude" under "VICTORY Data Types."
      ii. The latitude may be provided as part of the position.
   b. **Longitude**
      i. A detailed description of the longitude data is provided in the section titled "Longitude" under "VICTORY Data Types."
      ii. The longitude may be provided as part of the position.
   c. **Altitude**
      i. A detailed description of the altitude data is provided in the section titled "Altitude" under "VICTORY Data Types."
      ii. The altitude may be provided as part of the position.
   d. **MGRS**

i. A detailed description of the MGRS data is provided in the section titled "MGRS" under "VICTORY Data Types."
ii. The MGRS may be provided as part of the position.

2. ***Nominal Data Available***
   a. **Latitude Available**
      i. An indication that the latitude data inside the position VDM is both available and valid.
      ii. A Boolean value of True indicates the data is available and valid, False otherwise.
      iii. The latitude available flag shall be provided as part of the nominal data available.
   b. **Longitude Available**
      i. An indication that the longitude data inside the position VDM is both available and valid.
      ii. A Boolean value of True indicates the data is available and valid, False otherwise.
      iii. The longitude available flag shall be provided as part of the nominal data available.
   c. **Altitude Available**
      i. An indication that the altitude data inside the position VDM is both available and valid.
      ii. A Boolean value of True indicates the data is available and valid, False otherwise.
      iii. The altitude available flag shall be provided as part of the nominal data available.
   d. **Description**
      i. A detailed description of the position nominal capabilities.
      ii. The bounds of the description are described in the section titled "String Bounds" under "VICTORY Data Types."
      iii. The description may be provided as part of the nominal data available.

18.2 VICTORY Frame of Reference <20010–20110729, Exp>

18.2.1 Frame of Reference Tree Paradigm

The existence of multiple sensors on a platform mounted to movable surfaces makes a tree view of the position layout a convenient representation. From the tree of translation/rotation information, it is possible to obtain absolute position and orientation of a sensor with respect to the Vehicle Frame of Reference. The elements of this tree are rooted on a vehicle frame of reference typically in the hull of the vehicle, and branch off based on the degrees of freedom between the hull and the sensing elements.

18.2.2 Platform Frame of Reference

The origin of the platform frame of reference may vary from implementation to implementation, depending upon locations of sensors, and the geometry of the platform. The origin consists of a position and an orientation.

The platform frame of reference for position shall consist of three orthogonal axes oriented in a "right hand" configuration. Positive x shall point forward down the centerline of the platform, positive y shall point 90 degrees left of forward (port), and z shall be orthogonal to x and y, pointing upward in the platform's "at rest" stance. X, y, and z shall be in meters.

The platform frame of reference for orientation shall be relative to earth centered, earth fixed coordinates, in which the plane formed by x and y is perpendicular to the earth's gravitational force, and where z points in the opposite direction of earth's gravitational force.

> ***NOTE:*** *The platform frame of reference for position should be the mounting point of a GPS antenna. If the platform has no GPS, then an accessible point close to the center of mass should be used. The platform frame of reference for orientation should be defined by the "at rest" position of the platform, and by magnetic north. Zero roll and pitch should be when the x and y axes of the vehicle are in the "at rest" position. Zero yaw should be when the platform is pointed at magnetic north (the x axis is pointed north). The relative angles of the Inertial Measurement Unit's (IMU's) orientation with respect to the platform frame of reference for orientation will be applied to orientation measurements to determine the "absolute" orientation with respect to the earth.*

18.2.3 Orientation Coordinate System

Relative orientation shall be defined by three angles: phi, theta, and psi, corresponding to the three Euler angles, representing roll, pitch, and yaw in arc degrees. The orientations of these angles shall be as described in the section titled "VICTORY Data Types."

For purposes of further calculations, roll, pitch, and yaw shall each be constrained to –180° to 180°, but can be normalized with pitch constrained from –90° to 90° and roll/yaw adjusted accordingly.

18.2.4 Connector Frame of Reference

In cases where a sensor's position may change in translation or rotation with respect to the platform's origin, such as when a sensor is mounted to a ball, turret, or mast, it is necessary to produce a set of transformational parameters that represent this transformation. Two sets of triplets shall be required for each connector, the first static pair to handle fixed translational offset and rotation in normal axes, the second dynamic pair to handle rotation and elongation. For connectors that change dimension, such as a telescoping mast, the static pair should rotate the Local Frame of Reference Axes such that one of the axes is parallel to the direction of elongation. Connectors such as turrets should have a dynamic pair to denote a position and orientation along the turret's center of rotation, so the static pair would be null except for a single angle which would change as the turret rotates.

18.2.5 Sensor Frame of Reference

A sensor shall be treated as a connector for purposes of calculations and the configuration file, with the difference being that sensors would likely never have further children referencing them. The orientation of the sensor shall use the same conventions as the other position-related systems, having x axis normal to the face of the sensor, z in the logical up direction with respect to the sensor, and y orthogonal and pointing left from the sensor's perspective (pointing right when looking at the input to the sensor). The "zero" position point in the sensor frame of reference shall be the point used as the translational offset for the sensor. The "zero" orientation offset for the sensor shall define the sensor's direction of gaze, if it has one.

18.2.6 Configuration File

The frame of reference parameters for the connectors and sensors on a platform shall be described in a configuration file following the VICTORY Configuration Language schema. The XML format of this description is TBD.

The description shall include the static translational (x, y, z) and the static rotational triplet (phi, theta, psi), as well as masks identifying which of the fields can be dynamic. The static transformational triplet pair shall be given with respect to the parent connector's dynamic reference frame depending upon the attachment point for each connector and sensor. The instantaneous position shall be defined by a set of zero to six dynamic parameters referenced to the connector's local static frame of reference. The translation and orientation dynamic masks specify which of the six parameters are dynamic for the connector.

The value of the static translational and rotational triplets, as well as the translational and rotational dynamic masks, shall be specified in the configuration file. The values of the dynamic translational and rotational parameters will not be given in the configuration file, but will be tracked as the system operates and applied against the static triplets to obtain the instantaneous translation and orientation of the connector.

Optionally, the configuration file may specify the default values and the valid ranges of the dynamic parameters. The configuration file shall also have a unique identifier, an optional user-readable name, and a reference to the parent's unique identifier for each connector and sensor. The vehicle Frame of Reference, the root of the coordinate system tree, shall have a unique identifier but no parent reference.

18.2.7 Position and Orientation Calculation

The position and orientation for a particular sensor with respect to the platform frame of reference is to be calculated using the coordinate transformation tree contained in the configuration file, along with the instantaneous values of any dynamic translational and rotational parameters. The position and orientation of a sensor will be determined by executing the transformations resulting from traversal of the coordinate transformation tree from the node for that sensor, up through all parent nodes to the platform frame of reference.

NOTE: *The mathematical transformations necessary to calculate the instantaneous frame of reference for a sensor are not described in this specification.*

NOTE: *It may be most efficient for a central service to keep track of vehicular and connector data and perform all transformational arithmetic necessary to track the instantaneous frame of reference for each sensor, providing the resulting instantaneous frame of reference data to applications requiring it. Alternatively, each node that requires the information could perform the computations itself based on the configuration file and the runtime values of the dynamic parameters. Either design is compatible with the specification.*

NOTE: *Each connector has two pairs of triplets, of which one pair is static and contained in the configuration file. The dynamic set of parameters must be obtained at runtime from interfaces to the sensor and platform actuators. The details of these interfaces are TBD.*

18.3 Situational Awareness Interfaces

Situational Awareness (SA) refers to the tracking of red (enemy) and blue (friendly) forces, the mapping of known threats, and the identification of boundaries and battle plans on maps. These interfaces are currently controlled by separate US Army programs. Whether or not VICTORY will define SA-specific interfaces is TBD.

18.4 Threat Detection and Reporting Interfaces

Threat detection components are responsible for detecting active threats to the vehicle platform and reporting them to the warfighter. Examples of threat detection systems that are currently being deployed on vehicles include acoustic sensing devices that detect and report on supersonic sniper shots, as well as radar systems that detect subsonic rocket-propelled grenade (RPG) attacks. While many of these systems are being fielded on current force platforms, they have either very limited or no integration with other C4ISR/EW systems. The warfighter could benefit greatly from integration of these systems, and the VICTORY interfaces that follow provide a framework for this integration.

A number of different components may be used to implement a platform's threat detection and reporting capabilities. Because the technologies that are utilized by these components are constantly changing, VICTORY will not attempt to standardize the components themselves. However, VICTORY dictates that these components must have a VDB interface, at some point on the system, to share information with other VICTORY nodes. The following sections describe the logical standards that are to be implemented for this VDB interface.

18.4.1 Threat Detection and Reporting Data Specification <20011–20210409, Pro>

A new threat being detected by a threat detection system or sensor shall result in the contents of that threat being published to the VDB in the format described below. The threat shall only be sent to the VDB once. A complete list of threats that have yet to be deleted from the Threat Detection and Reporting service can be retrieved using the Reported Threats parameter inside the section titled "Threat Detection and Reporting Management Parameters."

18.4.1.1 Threat Detection Message Content

The Threat Detection and Reporting Data Interface provides information regarding threats that have been detected by various threat detection systems and/or sensors. This information will be available to all VDB network nodes that require threat detection data.

1. ***Relative Position***
    a. A detailed description of the relative position of a threat is provided in the section titled "Relative Position" under "Threat Detection and Reporting Complex Types."
    b. The relative position of the threat shall be provided as part of the threat message.
2. ***Observer Position***
    a. A detailed description of the observer position of a threat is provided in the section titled "Observer Position" under "Threat Detection and Reporting Complex Types."
    b. The observer position of the threat should be provided as part of the threat message.
3. ***Absolute Position***
    a. A detailed description of the absolute position of a threat is provided in the section titled "Absolute Position" under "Threat Detection and Reporting Complex Types."

b. The absolute position of the threat may be provided as part of the threat message.

4. ***Unique Name***
   a. The unique name of the threat.
   b. The bounds of the unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
   c. The unique name shall be unique within the scope of all other threats reported by the Threat Detection and Reporting Data Interface.
   d. The unique name shall be provided as part of the threat message.
5. ***Threat Type***
   a. The type of threat that has been detected.
   b. The threat type shall be a string chosen from one of the following values:
      i. INVALID
      ii. UNKNOWN
      iii. SMALL_ARMS
      iv. SUBSONIC
      v. SUPERSONIC
      vi. CBRN
      vii. ELECTRONIC_WARFARE
      viii. UNEXPLODED_ORDNANCE
      ix. LASER
      x. NON_LETHAL
      xi. LAUNCH_POINT
      xii. EXPERIMENTAL
      xiii. ROCKET_PROPELLED_GRENADE
      xiv. ANTI_TANK_GUIDED_MISSILE
      xv. RECOILLESS_RIFLE
      xvi. TANK_FIRED_HEAT_CE
      xvii. TANK_FIRED_KINETIC_ENERGY
      xviii. LASER_BEAM_RIDER
      xix. LASER_DESIGNATOR
      xx. LASER_RANGEFINDER
      xxi. LASER_SEMI_ACTIVE_HOMING
      xxii. SEMI_AUTOMATIC_COMMAND_TO_LINE_OF_SIGHT
      xxiii. UNDERBODY_BLAST_LOW_YIELD
      xxiv. UNDERBODY_BLAST_HIGH_YIELD
      xxv. TOP_ATTACK_FALLER
      xxvi. TOP_ATTACK_FLYER
      xxvii. IMPROVISED_EXPLOSIVE_DEVICE
      xxviii. FIRE_AND_FORGET_MISSILE
      xxix. MAN_PORTABLE_AIR_DEFENSE_SYSTEM
      xxx. SURFACE_TO_AIR_MISSILE
      xxxi. SMALL_ARMS_FIRE
   c. The threat type may be provided as part of the threat message.

> ***NOTE:*** *Some threat types reported by the Threat Detection and Reporting Data Interface may also be reported by more specific VICTORY component types. The Threat Detection interface is intended to report threat types that are currently not separately reported and to provide a summary report for all threats. Components implementing a more specific interface are encouraged to also implement this general one to allow collection of summary data.*

> ***NOTE:*** *This list is not all-inclusive, but lists common threat categories. Threats that do not fit an existing category should be published as "EXPERIMENTAL" with additional data in the Threat Description.*

6. ***Threat Description***
   a. An additional description that can be used to detail the threat in a human readable format or using the Symbol Identification Code (SIDC) scheme as defined within MIL-STD–2525C: Common Warfighting Symbology.
   b. The bounds of the threat description are described in the section titled "String Bounds" under "VICTORY Data Types."
   c. The threat description may be provided as part of the threat message.
7. ***Timestamp***
   a. A timestamp corresponding to the time at which the threat was detected.

b. The timestamp shall be in the format as described in the section titled "Timestamp Format for Application Layer Data."
c. The timestamp shall be provided as part of the threat message.

8. ***Sequence Number***
   a. A monotonically increasing number that indicates the sequence number of the Threat Detection message sent from a particular Threat Detection and Reporting Data Interface.
   b. The sequence number shall increase by one (1) for each Threat Detection message sent from the interface.
   c. When the sequence number reaches its maximum value, it shall roll over to its minimum value.

      NOTE: *The sequence number is intended to be used to detect messages that did not arrive or were dropped.*

   d. The sequence number shall be provided as part of the threat message.
9. ***Interface ID***
   a. An identification number, unique within the scope of the VDB, used to identify a particular Threat Detection and Reporting Data Interface.
   b. The interface ID shall be an integer value greater than or equal to zero.
   c. The interface ID shall be provided as part of the threat message.

18.4.1.2 Threat Detection Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as a Threat Detection VDM. The appendix titled "VICTORY Schemas" provides the name of the schema file. The file describes the Threat Detection VDM, provides additional details for the VICTORY Data Types Schema, and describes the Threat Detection type that generates the XML string used by the Threat Detection VDM. An example Threat Detection VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:vm="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <threatDetectionMessage interfaceId="1"
        timestamp="2006-05-04T18:13:51.0Z"
        sequenceNumber="5">
        <vst:relativePosition>
            <vst:azimuth>21.1232</vst:azimuth>
            <vst:elevation>2.0239</vst:elevation>
            <vst:range>2100</vst:range>
        </vst:relativePosition>
        <vst:observerPosition>
            <vst:latitude>38.898648</vst:latitude>
            <vst:longitude>-77.037692</vst:longitude>
            <vst:altitude>542</vst:altitude>
        </vst:observerPosition>
        <vst:threatType>SUBSONIC</vst:threatType>
        <vst:threatDescription>SHGPEWR--------</vst:threatDescription>
        <vst:uniqueName>threat1</vst:uniqueName>
    </threatDetectionMessage>
</vdm:messages>
```

Figure 19. Example Threat Detection VDM

18.4.2 Threat Detection and Reporting Management Parameters <15004–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the threat detection and reporting management parameters. A textual description of the threat detection and reporting management parameters is shown below.

1. ***Reported Threats***
    a. This parameter indicates the threats reported by a threat detection system or sensor.
    b. The contents of a threat are described in the section titled "Threat" under "Threat Detection and Reporting Complex Types."
    c. Removing a threat using the unique name, as defined in the section titled "Threat Detection and Reporting Complex Types," shall result in the removal of that threat from the list of reported threats.
        i. If a unique name is not provided, all threats in the reported threats list shall be removed.
    d. The following errors are defined for the reported threats parameter:
        i. Removing a threat using a unique name that does not match the unique name of one of the reported threats shall result in a GENERIC_FAULT error.
        ii. Removing a threat with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    e. Operations include Get() and Remove().
2. ***Data Enabled***
    a. This parameter identifies whether the Threat Detection and Reporting Data Interface managed by this interface is sending out VICTORY Data Messages (VDMs) on the VDB.
    b. A Boolean value of True means "enabled" (sending) and a value of False means "disabled" (not sending).
    c. If enabled is True, a VDM shall be sent every time a new threat is identified.
    d. If enabled is True, a new VDM may be sent for an existing threat if the ***Relative Position***, ***Observer Position***, or ***Absolute Position*** has changed.
    e. Operations include Get() and Set().
3. ***Time Uncertainty***
    a. This parameter identifies the uncertainty of the timestamp applied to the VDMs being sent to the VDB by the Threat Detection and Reporting Data Interface managed by this interface.
    b. The measure of time uncertainty shall be in units of milliseconds.
    c. The time uncertainty shall be a real value at 1 σ.
    d. Operations include Get().
4. ***Data Transport Configuration***
    a. This parameter identifies the transport configuration used by the Threat Detection and Reporting Data Interface managed by this interface for sending out VDMs on the VDB.
    b. The contents of a data transport configuration are described in the section titled "Data Transport Configuration" under "VICTORY Data Interface Complex Types."
    c. The following errors are defined for the data transport configuration parameter:
        i. Setting multiple data transport configurations with the same unique name shall result in a GENERIC_FAULT error.
        ii. If network layer is included in the request, setting a data transport configuration with a network layer outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
        iii. Setting a data transport configuration with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
        iv. Setting a data transport configuration with a URI outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.
    d. Operations include Get() and Set().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the threat detection and reporting management parameters along with a description of additional errors that were not specified but may be returned.*

18.4.3 Threat Detection and Reporting Complex Types <20015–20210409, Pro>

NOTE: *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

1. ***Threat***
    a. **Unique Name**
        i. The unique name of the threat.
        ii. The bounds of the unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
        iii. The unique name shall be unique within the scope of all other threats reported by the Threat Detection and Reporting Data Interface.
        iv. The unique name shall be provided as part of the threat.
    b. **Relative Position**
        i. The position of the threat relative to the sensor that observed the threat..
        ii. A detailed description of the relative position is provided in the section titled "Relative Position" under "Threat Detection and Reporting Complex Types."
        iii. The relative position shall be provided as part of the threat.
    c. **Timestamp**
        i. A timestamp corresponding to the time at which the threat was detected.
        ii. The timestamp shall be in the format as described in the section titled "Timestamp Format for Application Layer Data."
        iii. The timestamp shall be provided as part of the threat.
    d. **Observer Position**
        i. The position of the threat observer.
        ii. A detailed description of the observer position is provided in the section titled "Observer Position" under "Threat Detection and Reporting Complex Types."
        iii. The observer position should be provided as part of the threat.
    e. **Absolute Position**
        i. The position of the threat.
        ii. A detailed description of the absolute position is provided in the section titled "Absolute Position" under "Threat Detection and Reporting Complex Types."
        iii. The absolute position may be provided as part of the threat.
    f. **Threat Type**
        i. The type of threat that was detected.
        ii. The threat type shall be a string chosen from the following values:
            1. INVALID
            2. UNKNOWN
            3. SMALL_ARMS
            4. SUBSONIC
            5. SUPERSONIC
            6. CBRN
            7. ELECTRONIC_WARFARE
            8. UNEXPLODED_ORDNANCE
            9. LASER
            10. NON_LETHAL
            11. LAUNCH_POINT
            12. EXPERIMENTAL
            13. ROCKET_PROPELLED_GRENADE
            14. ANTI_TANK_GUIDED_MISSILE
            15. RECOILLESS_RIFLE
            16. TANK_FIRED_HEAT_CE
            17. TANK_FIRED_KINETIC_ENERGY
            18. LASER_BEAM_RIDER
            19. LASER_DESIGNATOR
            20. LASER_RANGEFINDER
            21. LASER_SEMI_ACTIVE_HOMING
            22. SEMI_AUTOMATIC_COMMAND_TO_LINE_OF_SIGHT
            23. UNDERBODY_BLAST_LOW_YIELD
            24. UNDERBODY_BLAST_HIGH_YIELD
            25. TOP_ATTACK_FALLER
            26. TOP_ATTACK_FLYER
            27. IMPROVISED_EXPLOSIVE_DEVICE
            28. FIRE_AND_FORGET_MISSILE

29. MAN_PORTABLE_AIR_DEFENSE_SYSTEM
30. SURFACE_TO_AIR_MISSILE
31. SMALL_ARMS_FIRE

iii. The threat type may be provided as part of the threat.

g. **Threat Description**

i. An additional description that can be used to detail the threat in a human readable format or using the Symbol Identification Code (SIDC) scheme as defined within MIL-STD–2525C: Common Warfighting Symbology.
ii. The bounds of the threat description are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The threat description may be provided as part of the threat.

2. ***Relative Position***

a. **Azimuth**

i. The azimuth of the threat.
ii. A detailed description of the azimuth data is provided in the section titled "Azimuth" under "VICTORY Data Types."
iii. The azimuth shall be provided as part of the relative position.

b. **Elevation**

i. The elevation of the threat.
ii. A detailed description of the elevation data is provided in the section titled "Elevation" under "VICTORY Data Types."
iii. The elevation shall be provided as part of the relative position.

c. **Range**

i. The range of the threat.
ii. A detailed description of the range data is provided in the section titled "Range" under "VICTORY Data Types."
iii. The range shall be provided as part of the relative position.

d. **Azimuth Valid**

i. An indication that the azimuth data inside the relative position is valid.
ii. A Boolean value of True indicates the data is valid, False otherwise.
iii. The azimuth valid flag shall be provided as part of the relative position.

e. **Elevation Valid**

i. An indication that the elevation data inside the relative position is valid.
ii. A Boolean value of True indicates the data is valid, False otherwise.
iii. The elevation valid flag shall be provided as part of the relative position.

f. **Range Valid**

i. An indication that the range data inside the relative position is valid.
ii. A Boolean value of True indicates the data is valid, False otherwise.
iii. The range valid flag shall be provided as part of the relative position.

g. **Azimuth Uncertainty**

i. The uncertainty of the azimuth measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
ii. The measure of uncertainty shall be in units of arc degrees.
iii. The uncertainty shall be a real value at 1 σ.
iv. If the azimuth valid flag is True, the azimuth uncertainty shall be provided as part of the relative position.
v. If the azimuth valid flag is False, the azimuth uncertainty shall not be provided as part of the relative position.

h. **Elevation Uncertainty**

i. The uncertainty of the elevation measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
ii. The measure of uncertainty shall be in units of arc degrees.
iii. The uncertainty shall be a real value at 1 σ.
iv. If the elevation valid flag is True, the azimuth uncertainty shall be provided as part of the relative position.
v. If the elevation valid flag is False, the azimuth uncertainty shall not be provided as part of the relative position.

i. **Range Uncertainty**

i. The uncertainty of the range measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
ii. The measure of uncertainty shall be in units of meters.
iii. The uncertainty shall be a real value at 1 σ.
iv. If the range valid flag is True, the azimuth uncertainty shall be provided as part of the relative position.
v. If the range valid flag is False, the azimuth uncertainty shall not be provided as part of the relative position.

j. **Azimuth Estimated**
i. An indication that the azimuth data inside the relative position is estimated.
ii. A Boolean value of True indicates the data is estimated, False otherwise.
iii. If the azimuth valid flag is True, the azimuth estimated flag shall be provided as part of the relative position.
iv. If the azimuth valid flag is False, the azimuth estimated flag shall not be provided as part of the relative position.

k. **Elevation Estimated**
i. An indication that the elevation data inside the relative position is estimated.
ii. A Boolean value of True indicates the data is estimated, False otherwise.
iii. If the elevation valid flag is True, the elevation estimated flag shall be provided as part of the relative position.
iv. If the elevation valid flag is False, the elevation estimated flag shall not be provided as part of the relative position.

l. **Range Estimated**
i. An indication that the range data inside the relative position is estimated.
ii. A Boolean value of True indicates the data is estimated, False otherwise.
iii. If the range valid flag is True, the range estimated flag shall be provided as part of the relative position.
iv. If the range valid flag is False, the range estimated flag shall not be provided as part of the relative position.

3. ***Observer Position***

a. **Latitude**
i. The latitude of the threat observer.
ii. A detailed description of the latitude data is provided in the section titled "Latitude" under "VICTORY Data Types."
iii. The latitude shall be provided as part of the observer position.

b. **Longitude**
i. The longitude of the threat observer.
ii. A detailed description of the longitude data is provided in the section titled "Longitude" under "VICTORY Data Types."
iii. The longitude shall be provided as part of the observer position.

c. **Altitude**
i. The altitude of the threat observer.
ii. A detailed description of the altitude data is provided in the section titled "Altitude" under "VICTORY Data Types."
iii. The altitude shall be provided as part of the observer position.

d. **MGRS**
i. The MGRS position of the threat observer.
ii. A detailed description of the MGRS data is provided in the section titled "MGRS" under "VICTORY Data Types."
iii. The MGRS may be provided as part of the observer position.

e. **Latitude Valid**
i. An indication that the latitude data inside the observer position is valid.
ii. A Boolean value of True indicates the data is valid, False otherwise.
iii. The latitude valid flag shall be provided as part of the observer position.

f. **Longitude Valid**
i. An indication that the longitude data inside the observer position is valid.
ii. A Boolean value of True indicates the data is valid, False otherwise.
iii. The longitude valid flag shall be provided as part of the observer position.

g. **Altitude Valid**

i. An indication that the altitude data inside the observer position is valid.
ii. A Boolean value of True indicates the data is valid, False otherwise.
iii. The altitude valid flag shall be provided as part of the observer position.

h. **Latitude Uncertainty**
i. The uncertainty of the latitude measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
ii. The measure of uncertainty shall be in units of arc degrees.
iii. The uncertainty shall be a real value at 1 σ.
iv. If the latitude valid flag is True, the latitude uncertainty shall be provided as part of the observer position.
v. If the latitude valid flag is False, the latitude uncertainty shall not be provided as part of the observer position.

i. **Longitude Uncertainty**
i. The uncertainty of the longitude measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
ii. The measure of uncertainty shall be in units of arc degrees.
iii. The uncertainty shall be a real value at 1 σ.
iv. If the longitude valid flag is True, the longitude uncertainty shall be provided as part of the observer position.
v. If the longitude valid flag is False, the longitude uncertainty shall not be provided as part of the observer position.

j. **Altitude Uncertainty**
i. The uncertainty of the altitude measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
ii. The measure of uncertainty shall be in units of meters.
iii. The uncertainty shall be a real value at 1 σ.
iv. If the altitude valid flag is True, the altitude uncertainty shall be provided as part of the observer position.
v. If the altitude valid flag is False, the altitude uncertainty shall not be provided as part of the observer position.

k. **Latitude Estimated**
i. An indication that the latitude data inside the observer position is estimated.
ii. A Boolean value of True indicates the data is estimated, False otherwise.
iii. If the latitude valid flag is True, the latitude estimated flag shall be provided as part of the observer position.
iv. If the latitude valid flag is False, the latitude estimated flag shall not be provided as part of the observer position.

l. **Longitude Estimated**
i. An indication that the longitude data inside the observer position is estimated.
ii. A Boolean value of True indicates the data is estimated, False otherwise.
iii. If the longitude valid flag is True, the longitude estimated flag shall be provided as part of the observer position.
iv. If the longitude valid flag is False, the longitude estimated flag shall not be provided as part of the observer position.

m. **Altitude Estimated**
i. An indication that the altitude data inside the observer position is estimated.
ii. A Boolean value of True indicates the data is estimated, False otherwise.
iii. If the altitude valid flag is True, the altitude estimated flag shall be provided as part of the observer position.
iv. If the altitude valid flag is False, the altitude estimated flag shall not be provided as part of the observer position.

4. ***Absolute Position***

a. **Latitude**
i. The latitude of the threat.
ii. A detailed description of the latitude data is provided in the section titled "Latitude" under "VICTORY Data Types."
iii. The latitude shall be provided as part of the absolute position.

b. **Longitude**
i. The longitude of the threat.

ii. A detailed description of the longitude data is provided in the section titled "Longitude" under "VICTORY Data Types."
iii. The longitude shall be provided as part of the absolute position.

c. **Altitude**
i. The altitude of the threat.
ii. A detailed description of the altitude data is provided in the section titled "Altitude" under "VICTORY Data Types."
iii. The altitude shall be provided as part of the absolute position.

d. **MGRS**
i. The MGRS position of the threat.
ii. A detailed description of the MGRS data is provided in the section titled "MGRS" under "VICTORY Data Types."
iii. The MGRS may be provided as part of the absolute position.

e. **Latitude Valid**
i. An indication that the latitude data inside the absolute position is valid.
ii. A Boolean value of True indicates the data is valid, False otherwise.
iii. The latitude valid flag shall be provided as part of the absolute position.

f. **Longitude Valid**
i. An indication that the longitude data inside the absolute position is valid.
ii. A Boolean value of True indicates the data is valid, False otherwise.
iii. The longitude valid flag shall be provided as part of the absolute position.

g. **Altitude Valid**
i. An indication that the altitude data inside the absolute position is valid.
ii. A Boolean value of True indicates the data is valid, False otherwise.
iii. The altitude valid flag shall be provided as part of the abosolute position.

h. **Latitude Uncertainty**
i. The uncertainty of the latitude measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
ii. The measure of uncertainty shall be in units of arc degrees.
iii. The uncertainty shall be a real value at 1 σ.
iv. If the latitude valid flag is True, the latitude uncertainty shall be provided as part of the absolute position.
v. If the latitude valid flag is False, the latitude uncertainty shall not be provided as part of the absolute position.

i. **Longitude Uncertainty**
i. The uncertainty of the longitude measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
ii. The measure of uncertainty shall be in units of arc degrees.
iii. The uncertainty shall be a real value at 1 σ.
iv. If the longitude valid flag is True, the longitude uncertainty shall be provided as part of the absolute position.
v. If the longitude valid flag is False, the longitude uncertainty shall not be provided as part of the absolute position.

j. **Altitude Uncertainty**
i. The uncertainty of the altitude measurement, where uncertainty is defined as a measure of the inherent variability of repeated measurements of a quantity.
ii. The measure of uncertainty shall be in units of meters.
iii. The uncertainty shall be a real value at 1 σ.
iv. If the altitude valid flag is True, the altitude uncertainty shall be provided as part of the absolute position.
v. If the altitude valid flag is False, the altitude uncertainty shall not be provided as part of the absolute position.

k. **Latitude Estimated**
i. An indication that the latitude data inside the absolute position is estimated.
ii. A Boolean value of True indicates the data is estimated, False otherwise.
iii. If the latitude valid flag is True, the latitude estimated flag shall be provided as part of the absolute position.
iv. If the latitude valid flag is False, the latitude estimated flag shall not be provided as part of the absolute position.

l. **Longitude Estimated**
   i. An indication that the longitude data inside the absolute position is estimated.
   ii. A Boolean value of True indicates the data is estimated, False otherwise.
   iii. If the longitude valid flag is True, the longitude estimated flag shall be provided as part of the absolute position.
   iv. If the longitude valid flag is False, the longitude estimated flag shall not be provided as part of the absolute position.

m. **Altitude Estimated**
   i. An indication that the altitude data inside the absolute position is estimated.
   ii. A Boolean value of True indicates the data is estimated, False otherwise.
   iii. If the altitude valid flag is True, the altitude estimated flag shall be provided as part of the absolute position.
   iv. If the altitude valid flag is False, the altitude estimated flag shall not be provided as part of the absolute position.

18.5 Mission Recorder Interfaces <Informational>

Mission recording components are responsible for the storage of VICTORY-defined types of network data for retrieval at a later time. The hardware components that are used to support this capability can range widely, but will typically involve a network-ready processing device, as well as a mass (non-volatile) storage device to retain the network data. Because the mission recording hardware will vary depending upon mission objectives and platform limitations, the VICTORY architecture aims to standardize only the interfaces that the component hosts.

18.5.1 Mission Recorder Core Functionality <03001–20131031, Pro>

A mission recorder shall provide the following core capabilities:

- The mission recorder shall host at least one VDB interface, to sink data (record) from the network and provide data retrieval (playback) of that data.
- The mission recorder shall record data of different types, defined by VICTORY, according to mission objectives and component limitations.
- The mission recorder is only required to record data from VICTORY interfaces that are at or above the "proposed" maturity level.

  NOTE: *As new VICTORY interfaces are matured to the "proposed" level in this specifications document, the specification version of the mission recorder will be increased to match the latest document version.*
- The mission recorder shall provide the capability to record selected data occurring at a time prior to the command to record.
- The mission recorder shall provide the capability to playback recorded data at the rate at which it was recorded, and optionally, at a faster or slower rate.
- The mission recorder shall playback recorded data using the UDP transport protocol.

  NOTE: *Data recorded from a TCP source will need to be converted into the appropriate UDP format for playback.*

18.5.2 Mission Recorder Device Management Parameters <15015–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the mission recorder device management parameters. A textual description of the mission recorder device management parameters is shown below.

1. ***Data Stream***
   a. This parameter manages active mission recorder data streams.
   b. A data stream is described in the section titled "Mission Recorder Data Stream" under "Mission Recorder Interface Complex Types."
   c. Removing a data stream shall result in the removal of all recorded data for that data stream.
   d. The following errors are defined for the data stream parameter:
      i. When adding a data stream, if the unique name provided in the request already exists on the mission recorder, a GENERIC_FAULT error shall be returned.

ii. Adding a data stream of a type not supported by the mission recorder device may result in an OUT_OF_RANGE error.
iii. Removing a data stream with an unknown unique name shall result in a GENERIC_FAULT error.
iv. Removing a data stream with a state of PLAYBACK or RECORDING shall result in a GENERIC_FAULT error.
v. Adding or removing a data stream with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
vi. Adding a recording transport configuration or playback transport configuration with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
vii. Adding a recording transport configuration or playback transport configuration with a URI outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.
viii. Adding a comment to the data stream with a comment outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
ix. Adding a recorded file to the data stream with a file name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.

e. Operations include Add() and Remove().
i. The Data Stream State shall not be set through the Add() or Remove() operations.

> ***NOTE:*** *Information about the data streams must persist if the device is restarted.*

2. ***Available Data Streams***
a. This parameter identifies the data streams that have been configured on the mission recorder.
b. A data stream is described in the section titled "Mission Recorder Data Stream" under "Mission Recorder Interface Complex Types."
c. Operations include Get().
3. ***Data Stream Recording Transport Configuration***
a. This parameter identifies the transport configuration used by the mission recorder to record data from the VDB for the specified data stream.
b. The contents of a transport configuration are described in the section titled "Data Transport Configuration" under "VICTORY Data Interface Complex Types."
c. The data stream shall be specified by its unique name.
i. The bounds of the data stream unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
d. The following errors are defined for the data stream recording transport configuration parameter:
i. Setting the recording transport configuration while the stream state is RECORDING shall result in a GENERIC_FAULT error.
ii. Retrieving or setting the recording transport configuration with an unknown data stream unique name shall result in a GENERIC_FAULT error.
iii. Setting the recording transport configuration with a transport configuration unique name that already exists on the mission recorder shall result in a GENERIC_FAULT error.
iv. If network layer is included in the request, setting a data stream recording transport configuration with a network layer outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
v. Retrieving the recording transport configuration with a data stream unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
vi. Setting the recording transport configuration with a data stream unique name or transport configuration unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
vii. Setting the recording transport configuration with a URI outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.
e. Operations include Get() and Set().
4. ***Data Stream Playback Transport Configurations***
a. This parameter identifies the transport configurations used by the mission recorder device for playing back recorded data on the VDB for the specified data stream.
b. The contents of a transport configuration are described in the section titled "Data Transport Configuration" under "VICTORY Data Interface Complex Types."
c. The data stream shall be specified by its unique name.
i. The bounds of the data stream unique name are described in the section titled "String Bounds" under "VICTORY Data Types."

d. Setting playback transport configurations removes all existing playback transport configurations and adds the playback transport configurations provided in the request.
e. The following errors are defined for the data stream playback transport configurations parameter:
    i. Setting, adding, or removing playback transport configurations while the stream state is PLAYBACK shall result in a GENERIC_FAULT error.
    ii. Retrieving, setting, adding, or removing playback transport configurations with an unknown data stream unique name shall result in a GENERIC_FAULT error.
    iii. When adding or setting playback transport configurations, if any unique name provided in the request is duplicated or already exists on the mission recorder, a GENERIC_FAULT error shall be returned.
    iv. Removing the last playback transport configuration shall result in a GENERIC_FAULT error.
    v. If network layer is included in the request, setting or adding a data stream playback transport configuration with a network layer outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
    vi. Retrieving, setting, or adding a data stream playback transport configuration with a data stream unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    vii. Removing a data stream playback transport configuration with a data stream unique name or transport configuration unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    viii. Setting or adding a data playback transport configuration with a URI outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.
f. Operations include Get(), Set(), Add(), and Remove().

5. ***Data Stream Type***
    a. This parameter identifies the type of the specified data stream.
    b. The data stream shall be specified by its unique name.
        i. The bounds of the data stream unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
    c. The following errors are defined for the data stream type parameter:
        i. Retrieving the data stream type with an unknown data stream unique name shall result in a GENERIC_FAULT error.
        ii. Retrieving the data stream type with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    d. Operations include Get().
6. ***Data Stream Past Data Enabled***
    a. This parameter identifies whether data coming from the specified data stream will be buffered when recording is not active.

    > ***NOTE:*** *The size of the buffer is implementation-specific, but should be indicated in the configuration description for a mission recorder device.*

    b. The data stream shall be specified by its unique name.
        i. The bounds of the data stream unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
    c. A Boolean value of True means the data coming from the specified data stream will be buffered and a value of False means the data coming from the specified data stream will not be buffered.
    d. The following errors are defined for the data stream past data enabled parameter:
        i. Retrieving or setting the past data enabled flag with an unknown data stream unique name shall result in a GENERIC_FAULT error.
        ii. Retrieving or setting the past data enabled flag with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    e. Operations include Get() and Set().
7. ***Data Stream Comments***
    a. This parameter provides a mechanism for adding additional comments about the specified data stream.
    b. The data stream shall be specified by its unique name.
        i. The bounds of the data stream unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
    c. The bounds of each comment are described in the section titled "String Bounds" under "VICTORY Data Types."
    d. Each comment shall contain a unique name and an unrestricted string representing the comment.

e. Setting data stream comments replaces all existing comments with the comments provided in the request.
f. The following errors are defined for the data stream comments parameter:
    i. Retrieving, setting, adding, or removing data stream comments with an unknown data stream unique name shall result in a GENERIC_FAULT error.
    ii. When adding or setting data stream comments, if any unique name provided in the request is duplicated or already exists on the VDB, a GENERIC_FAULT error shall be returned.
    iii. Removing a data stream comment with an unknown unique name shall result in a GENERIC_FAULT error.
    iv. Adding or setting a data stream comment with a comment or comment unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    v. Retrieving, setting, adding, or removing data stream comments with a data stream unique name outside the string bounds defined by this specification may result in an OUT_OF_RANGE error.
g. Operations include Get(), Set(), Add(), and Remove().

8. ***Data Stream State***
    a. This parameter provides information about the state of a mission recorder data stream.
    b. The data stream shall be specified by its unique name.
    c. The bounds of the data stream unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
    d. The following errors are defined for the data stream state parameter:
        i. Retrieving the data stream state with an unknown data stream unique name shall result in a GENERIC_FAULT error.
        ii. Retrieving the data stream state with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    e. Operations include Get().
9. ***Available Permanent Storage***
    a. This parameter indicates the total size of available permanent storage on the mission recorder device.
    b. The available permanent storage shall be a real value greater than or equal to zero.
    c. The available permanent storage shall be given in units chosen from one of the following values:
        i. B
            1. Indicates a size in bytes, where 1 byte is equal to 8 bits.
        ii. KB
            1. Indicates a size in kilobytes, where 1 kilobyte is equal to 1024 bytes.
        iii. MB
            1. Indicates a size in megabytes, where 1 megabyte is equal to 1024^2 bytes.
        iv. GB
            1. Indicates a size in gigabytes, where 1 gigabyte is equal to 1024^3 bytes.
        v. TB
            1. Indicates a size in terabytes, where 1 terabyte is equal to 1024^4 bytes.
    d. Operations include Get().
10. ***Recording***
    a. This parameter starts or stops recording a specified data stream available to the mission recorder device.

> ***NOTE:*** *The term "record" means that data from the selected data stream is written to media so that it can be retrieved at a later time. The mission recorder device provides a mechanism by which certain data streams can be flagged for the purpose of recording data that occurred in the past (i.e., buffering). For example, a threat detection could trigger the recorder device to begin recording. For the purpose of post-mission analysis, it may be desirable to record data streams immediately prior to the threat detection event. The mission recorder device provides this mechanism via the "Past Data Enabled" flag in the "Mission Recorder Data Stream" as defined in the section titled "Mission Recorder Interface Complex Types."*

> ***NOTE:*** *Each recording must create a corresponding entry in the "Recorded Files" list of the "Mission Recorder Data Stream" as defined in the section titled "Mission Recorder Interface Complex Types."*

    b. The data stream recording that is being started or stopped shall be specified by its unique name.
        i. The bounds of the data stream unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
    c. If the start operation is successful, the stream state shall be changed to RECORDING.
    d. If the stop operation is successful, the stream state shall be changed to IDLE.
    e. The following errors are defined for the recording parameter:

i. Starting or stopping the recording of a data stream with an unknown data stream unique name shall result in a GENERIC_FAULT error.
ii. Attempting to start recording a stream when the stream state is RECORDING or PLAYBACK shall result in a GENERIC_FAULT error.
iii. Stopping recording on a stream when the stream state is IDLE or PLAYBACK shall result in a GENERIC_FAULT error.
iv. Starting or stopping the data stream recording with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.

f. Operations include Start() and Stop().

11. ***Playback***
    a. This parameter starts or stops playing back a specified recorded data stream.
    b. The data stream to playback shall be specified by its unique name.
        i. The bounds of the data stream unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
    c. When starting a playback, selection criteria may include a start time and/or an end time.
        i. Providing a start time but no end time shall result in the playback of all data recorded on or after the start time.
        ii. Providing an end time but no start time shall result in the playback of all data recorded on or before the end time.
        iii. Providing neither a start time nor an end time shall result in the playback of all recorded data.
    d. When starting a playback, a positive, real-valued scale factor may be provided for the speed of playback.

> ***NOTE:*** *The scale factor is a multiplier. For example, a factor of 2.0 results in playback at 2x real time.*

    e. If the start operation is successful, the stream state shall be changed to PLAYBACK.
    f. If the stop operation is successful, the stream state shall be changed to IDLE.
    g. The following errors are defined for the playback parameter:
        i. Starting or stopping playback on a data stream with an unknown data stream unique name shall result in a GENERIC_FAULT error.
        ii. Starting playback on a data stream when the stream state is RECORDING or PLAYBACK shall result in a GENERIC_FAULT error.
        iii. When starting playback on a data stream, if the start time provided in the request is earlier than the start of the recording or the end time provided in the request is later than the end of the recording, a NEAREST_VALUE error shall be returned.
            1. If the start time was earlier than the start of the recording, the start time shall be changed to the start time of the recording.
            2. If the end time was later than the end of the recording, the end time shall be changed to the end time of the recording.
        iv. When starting playback on a data stream, if the start time provided in the request is later than the current time, a GENERIC_FAULT error shall be returned.
        v. When starting playback on a data stream, if the device does not support the requested scale factor, a NEAREST_VALUE error shall be returned.
            1. The scale factor shall be changed to the nearest scale factor supported by the device and shall be sent back in the response.
        vi. Stopping playback on a data stream when the stream state is RECORDING or IDLE shall result in a GENERIC_FAULT error.
        vii. Starting or stopping the data stream playback with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    h. Operations include Start() and Stop().

12. ***Recorded Data***
    a. This parameter erases recorded data based on selection criteria.
    b. The selection criteria shall include the unique name of the data stream having data erased.
        i. The bounds of the data stream unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
    c. The selection criteria may include an "erase before" timestamp.
        i. Providing a time shall result in the erasure of all data recorded on or before the provided time.
        ii. Not providing a time shall result in the erasure of all recorded data for that data stream.
    d. The following errors are defined for the erase recorded data parameter:
        i. Attempting to erase data for a data stream with an unknown data stream unique name shall result in a GENERIC_FAULT error.

ii. If the selected stream is in the state of RECORDING or PLAYBACK, a GENERIC_FAULT error shall be returned.
iii. If the time provided in the request is earlier than the start of the recorded data for the selected stream, a NEAREST_VALUE error shall be returned.
1. The actual start time of the recording shall be sent back in the response.
iv. Attempting to erase recorded data with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.

e. Operations include Erase().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the mission recorder device management parameters along with a description of additional errors that were not specified but may be returned.*

18.5.3 Mission Recorder Interface Complex Types <20027–20210409, Pro>

NOTE: *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

1. ***Mission Recorder Data Stream***
   a. **Unique Name**
      i. The unique name of the data stream.
      ii. The unique name shall be unique within the scope of all data streams on the mission recorder.
      iii. The bounds of the unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
      iv. The unique name shall be provided as part of the mission recorder data stream.
   b. **Data Stream Type**
      i. The type(s) of data being provided by the data stream.
      ii. The data stream type shall be a list of strings chosen from any combination of the following:
         1. STD_DEFINITION_STREAMING_VIDEO
         2. STREAMING_AUDIO
         3. The "Data Type" column of Table 298 in the appendix titled "VDM Data Type Values."

            NOTE: *Mission recorder does not support experimental data streams. Thus, any VDM inside Table 298 with a "Message Value" column that begins with "0xF" cannot be selected as a mission recorder data stream.*

      iii. The data stream type shall be provided as part of the mission recorder data stream.
   c. **Past Data Enabled**
      i. A flag indicating whether or not the data coming from the data stream can be stored even when not being recorded by the mission recorder.
      ii. A Boolean value of True means the data can be stored and a value of False means the data cannot be stored.
      iii. The past data enabled flag shall be provided as part of the mission recorder data stream.
   d. **State**
      i. The state of the data stream.
      ii. The state shall be a string chosen from the following values:
         1. RECORDING
            a. Indicates that stream data is being recorded.
         2. PLAYBACK
            a. Indicates that stream data is being played back.
         3. IDLE
            a. Indicates that no action is occurring on stream data.
      iii. The state shall be provided as part of the mission recorder data stream.
   e. **Recording Transport Configuration**
      i. The transport configuration used by the mission recorder for recording data from the VDB.
      ii. The contents of a transport configuration are described in the section titled "Data Transport Configuration" under "VICTORY Data Interface Complex Types."
      iii. The recording transport configuration shall be provided as part of the mission recorder data stream.
   f. **Playback Transport Configurations**

i. The transport configurations used by the mission recorder for playing back recorded data on the VDB.
ii. The contents of a transport configuration are described in the section titled "Data Transport Configuration" under "VICTORY Data Interface Complex Types."
iii. The playback transport configurations shall be provided as part of the mission recorder data stream.

g. **Comments**
i. Provides a mechanism for adding additional comments about the data stream.
ii. The bounds of each comment are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The comments may be provided as part of the mission recorder data stream.

h. **Recorded Files**
i. Provides information about recordings created by the mission recorder.
ii. The contents of a recording file information type are described inside the section titled "Mission Recorder File Information" under "Mission Recorder Interface Complex Types."
iii. Recording file information may be provided for each recorded file as part of the mission recorder data stream.

2. ***Mission Recorder File Information***

a. **File Name**
i. The name of the recorded file.
ii. The bounds of the file name are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The file name shall be provided as part of the mission recorder file information.

b. **Start Time**
i. The start time of the file recording.
ii. The start time shall use the timestamp format given in the section titled "Timestamp Format for Application Layer Data."
iii. The start time shall be provided as part of the mission recorder file information.

c. **End Time**
i. The end time of the file recording.
ii. The end time shall use the timestamp format given in the section titled "Timestamp Format for Application Layer Data."
iii. The end time shall be provided as part of the mission recorder file information.

18.6 Data Logger

The data logger component type is similar to the mission recorder in that it receives and stores network traffic in a persistent medium. The difference is that instead of providing a "playback" function for recreating the message traffic as received, it instead stores the data into a "log," which can be retrieved in bulk.

18.6.1 Data Logger Data Specification <20076–20210409, Pro>

18.6.1.1 Data Logger Message Content

The Data Logger Data Interface provides information regarding the current storage information of the data logger. This information will be available to all VDB network nodes that require storage data from the data logger.

The Data Logger Data Interface shall provide the following:

1. ***Current Log Size***
a. This element describes the current log size.
b. The current log size is the size of the log file created when retrieving a log without a stream or time filter.
i. The contents of the current log size are described in the section titled "Size" under "Data Logger Complex Types."
2. ***Available Storage***
a. This element describes the remaining storage space available to the data logger.
b. The available storage space is the current amount of free disk space available to the data logger.

i. The contents of the size of available storage space are described in the section titled "Size" under "Data Logger Complex Types."

3. ***Maximum Log Size***
   a. This element indicates the maximum log size of the data logger.
   b. The maximum log size is the largest possible size for a log file created when retrieving a log without a stream or time filter.
      i. The contents of the maximum log size are described in the section titled "Size" under "Data Logger Complex Types."
   c. The maximum log size shall have a size greater than zero.
4. ***Timestamp***
   a. A timestamp corresponding to the time at which the storage information was sampled.
   b. The timestamp shall be in the format as described in the section titled "Timestamp Format for Application Layer Data."
5. ***Sequence Number***
   a. A monotonically increasing number that indicates the sequence number of the data logger message sent from a particular Data Logger Data Interface.
   b. The sequence number shall increase by one (1) for each data logger message sent from the interface.
   c. When the sequence number reaches its maximum value, it shall roll over to its minimum value.

   NOTE: *The sequence number is intended to be used to detect messages that did not arrive or were dropped.*
6. ***Interface ID***
   a. An identification number, unique within the scope of the VDB, used to identify a particular Data Logger Data Interface.
   b. The interface ID shall be an integer value greater than or equal to zero.

18.6.1.2 Data Logger Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as a Data Logger VDM. The appendix titled "VICTORY Schemas" provides the name of the schema file. The file describes the Data Logger VDM, provides additional details for the VICTORY Data Types Schema, and describes the Data Logger type that generates the XML string used by the Data Logger VDM. An example Data Logger VDM is provided below.

```
<vdm:messages xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd">
    <dataLoggerStorageMessage interfaceId="4"
        timestamp="2006-05-04T18:13:51.0Z"
        sequenceNumber="214748364">
        <vt:currentLogSize>
            <vst:size>29</vst:size>
            <vst:units>KB</vst:units>
        </vt:currentLogSize>
        <vt:maximumLogSize>
            <vst:size>1024</vst:size>
            <vst:units>KB</vst:units>
        </vt:maximumLogSize>
        <vt:availableStorage>
            <vst:size>1024</vst:size>
            <vst:units>KB</vst:units>
        </vt:availableStorage>
    </dataLoggerStorageMessage>
</vdm:messages>
```

Figure 20. Example Data Logger VDM

18.6.2 Data Logger Management Parameters <15021–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the data logger management parameters. A textual description of the data logger management parameters is shown below.

1. ***Streams***
    a. This parameter identifies a stream of network data that shall be captured and stored by the data logger.
        i. The contents of the stream are described in the section titled "Stream" under "Data Logger Complex Types."
        ii. Including an element in a stream shall restrict the messages captured by the data logger.
    b. Adding a stream shall consist of a single stream.
    c. Setting a list of streams shall overwrite the current stream list.
    d. Adding or setting a stream or list of streams and including an element in a stream filter shall limit the messages captured by the data logger.
    e. Removing a stream shall cause the data logger to stop capturing messages defined in that stream.
    f. When a stream is removed from the list of streams, the data logger logs shall not be affected.
    g. Retrieving the streams shall return the list of all active streams.
    h. The following errors are defined for the streams parameter:
        i. Removing a stream with a unique name not known to the data logger shall result in a GENERIC_FAULT error.
        ii. Adding a stream with the same unique name of a stream known to the data logger shall result in a GENERIC_FAULT error.
        iii. Setting a stream with multiple streams containing the same unique name shall result in a GENERIC_FAULT error.
        iv. Adding or Setting a stream with an unknown VDM type may result in an OUT_OF_RANGE error.
        v. Adding or Setting a stream with an invalid health type may result in an OUT_OF_RANGE error.
        vi. Adding or Setting a stream with an invalid timestamp may result in an OUT_OF_RANGE error.
        vii. Adding or Setting a stream with an end time before the request was received shall result in a GENERIC_FAULT error.
        viii. Adding or Setting a stream with a start time before the request was received shall result in a NEAREST_VALUE error.
            1. The start time shall be changed to the time of the request.
        ix. Adding or Setting a stream with a start time after the end time shall result in a GENERIC_FAULT error.
        x. Adding or Setting a stream with an end time or start time within the bounds defined by this specification document, but too precise for the data logger, shall result in a NEAREST_VALUE error.
            1. The time(s) shall be returned along with the NEAREST_VALUE error.
        xi. Adding or Setting a stream with an invalid URI may result in an OUT_OF_RANGE error.
        xii. Adding or Setting a stream with a health type inside the range defined by this specification document, but unrecognized by the data logger, shall result in a NOT_SUPPORTED error.
        xiii. Adding, Setting, or Removing a stream with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
        xiv. Adding or Setting a stream with a URI outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.
    i. Operations include Get(), Set(), Add(), and Remove().
2. ***Logging***
    a. This parameter starts or stops the logging of the streams.
    b. A Start Logging request shall cause the data logger to begin logging messages from the current list of streams known to the data logger.
    c. A Stop Logging request shall cause the data logger to stop logging messages on all streams.
    d. Start and Stop Logging shall not cause the data logger to lose any messages captured while logging.
    e. Operations include Start() and Stop().
3. ***Log***
    a. This parameter shall return a URI that can be used by a client to retrieve a log file.
        i. The URI shall be a valid Uniform Resource Identifier (URI) formatted as a string as defined in IETF RFC 3986.

1. The URI shall contain a scheme component of File Transfer Protocol (FTP) or File Transfer Protocol Secure (FTPS).
2. The bounds of the URI are described in the section titled "URI Bounds" under "VICTORY Data Types."
3. The URI shall contain an authority component.
    a. The authority shall contain a host and port.
    b. The authority may contain a userinfo subcomponent.
        i. The userinfo is defined as 'user:password@'.
4. The URI shall contain an absolute path to the log file.

ii. The method for a client to retrieve the log is described in the section titled "Data Logger FTP Specification."
iii. The format of the log file is described in the section titled "Data Logger Pcap Log File Format."

b. This parameter's request may include a list of stream filters, a time filter, or both.
    i. The contents of a stream filter are described in the section titled "Stream Filter" under "Data Logger Complex Types."
        1. A stream filter shall include any combination of a unique name, a stream type, and a URI.
            a. Including an element in a stream filter shall restrict the messages included in the log file to the messages that match the filter.
        2. Messages matching any of the stream filters shall be included in the log file.
    ii. The contents of the time filter are described in the section titled "Time Filter" under "Data Logger Complex Types."
        1. The time filter shall include a start time, a stop time, or both.
            a. Including a start time in the filter shall restrict the messages provided in the log file to messages captured after the start time.
            b. Including an end time in the filter shall restrict the messages provided in the log file to messages captured before the end time.
c. The following errors are defined for the log parameter:
    i. Retrieving a log with an unknown VDM Type may result in an OUT_OF_RANGE error.
    ii. Retrieving a log with an invalid health type may result in an OUT_OF_RANGE error.
    iii. Retrieving a log with an invalid timestamp may result in an OUT_OF_RANGE error.
    iv. Retrieving a log with a start time or end time after the request was made shall result in a GENERIC_FAULT error.
    v. Retrieving a log with a start time after the end time shall result in a GENERIC_FAULT error.
    vi. Retrieving a log with an end time or start time within the bounds defined by this specification document, but too precise for the data logger, shall result in a NEAREST_VALUE error.
        1. The time(s) shall be returned along with the NEAREST_VALUE error.
    vii. Retrieving a log with an invalid URI may result in an OUT_OF_RANGE error.
    viii. Retrieving a log with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    ix. Retrieving a log with a URI outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.
d. Operations include Get().

4. ***Data***
    a. This parameter erases data (messages) stored in the data logger.
    b. This parameter may include a time filter.
        i. Including a time filter in the request shall result in only the data captured between the start time and the end time.
        ii. Including only a start time shall result in the erasure of all data after that time.
        iii. Including only an end time shall result in the erasure of all data before that time.
        iv. Not including a time filter in the request shall erase all messages captured by the data logger.
        v. The contents of the time filter are described in the section titled "Time Filter" under "Data Logger Complex Types."
    c. The following errors are defined for the data parameter:
        i. Erasing data with a start time or end time after the request was made shall result in a GENERIC_FAULT error.
        ii. Erasing data with a start time after the end time shall result in a GENERIC_FAULT error.
        iii. Erasing data with an end time or start time within the bounds defined by this specification document, but too precise for the data logger, shall result in a NEAREST_VALUE error.
            1. The time(s) shall be returned along with the NEAREST_VALUE error.

d. Operations include Erase().

5. ***Current Log Size***
    a. This parameter describes the current log size.
    b. The current log size is the size of the log file created when retrieving a log without a stream or time filter.
        i. The contents of the current log size are described in the section titled "Size" under "Data Logger Complex Types."
    c. Operations include Get().
6. ***Available Storage***
    a. This parameter describes the storage space currently available to the data logger.
    b. The available storage space is the current amount of free disk space available to the data logger.
        i. The contents of available storage are described in the section titled "Size" under "Data Logger Complex Types."
    c. Operations include Get().
7. ***Maximum Log Size***
    a. This parameter indicates the maximum log size of the data logger.
    b. The maximum log size is the largest possible size for a log file created when retrieving a log without a stream or time filter.
        i. The contents of the maximum log size are described in the section titled "Size" under "Data Logger Complex Types."
    c. The maximum log size shall have a size greater than zero.
    d. Upon reaching the maximum log size, if the overwrite enabled parameter is True, the data logger shall overwrite the oldest captured message, and if the overwrite enabled parameter is False, the data logger shall no longer record any messages until the log has either been cleared or the maximum log size is increased.
    e. The following errors are defined for the maximum log size parameter:
        i. Setting the maximum log size to a unit's value outside the bounds defined by this specification document may result in an OUT_OF_RANGE error.
        ii. Setting the maximum log size to a size value outside the bounds defined by this specification document may result in an OUT_OF_RANGE error.
        iii. Setting the maximum log size to a size value larger than the combination of the available storage space and the current log size shall result in a NEAREST_VALUE error.
            1. The maximum log size shall be returned along with the NEAREST_VALUE error.
        iv. Setting the maximum log size to a size value inside the bounds defined by this specification document, but not allowed by the data logger, shall result in a NEAREST_VALUE error.
            1. The maximum log size shall be returned along with the NEAREST_VALUE error.
        v. Setting the maximum log size to a size value smaller than the current log size shall result in a NEAREST_VALUE error.
            1. The maximum log size shall be returned along with the NEAREST_VALUE error.
    f. Operations include Get() and Set().
8. ***Overwrite Enabled***
    a. This parameter indicates whether or not the data logger is overwriting old captured messages.
    b. A Boolean value of True means the data logger shall overwrite the oldest captured message when the maximum log size has been reached, and a value of False means the data logger shall no longer record any messages until the log has been cleared or the maximum log size is increased.
    c. The following errors are defined for the overwrite enabled parameter:
        i. Setting the overwrite enabled flag to True where the data logger does not support the ability to overwrite logs shall result in a NOT_SUPPORTED error.
    d. Operations include Get() and Set().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the data logger management parameters along with a description of additional errors that were not specified but may be returned.*

18.6.3 Data Logger Complex Types <20077–20210409, Pro>

NOTE: *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

1. ***Stream***
    a. **Unique Name**

i. The unique name identifies the unique identifier of the stream.
ii. The unique name shall be unique within the scope of all streams on the data logger.
iii. The bounds of the unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
iv. The unique name shall be provided as part of the stream.

b. **URI**
i. The URI of the stream identifies where the data logger will capture the messages.
ii. The URI shall be a valid Uniform Resource Identifier (URI) formatted as a string as defined in IETF RFC 3986.
iii. The bounds of the URI are described in the section titled "URI Bounds" under "VICTORY Data Types."
iv. The URI shall be provided as part of the stream.

c. **Stream Type**
i. The stream type identifies the type of data the data logger shall capture from the URI.
ii. The stream type shall be defined as a choice between a VDM stream and a health stream.
1. The contents of the stream type are described in the section titled "Stream Type" under "Data Logger Complex Types."
iii. The stream type shall be provided as part of the stream.

d. **Time Filter**
i. The time filter identifies a period of time the data logger will log messages from the URI.
ii. The time filter shall include either a start time or a stop time.
iii. The time filter may include both a start time and a stop time.
iv. The time filter may be provided as part of the stream.

2. ***Stream Filter***

a. **Unique Name**
i. The unique name identifies the unique name of a stream to include in the log file.
ii. The bounds of the unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The unique name may be provided as part of the stream filter.

b. **URI**
i. The URI of the stream identifies the URI of a captured message stream.
1. Providing a URI as part of the stream filter shall result in all messages captured via that URI.
ii. The URI shall be a valid Uniform Resource Identifier (URI) formatted as a string as defined in IETF RFC 3986.
iii. The bounds of the URI are described in the section titled "URI Bounds" under "VICTORY Data Types."
iv. The URI may be provided as part of the stream filter.

c. **Stream Type**
i. The type of stream to include in the log file.
1. Providing a stream type as part of the stream filter shall result in all messages that match the stream type.
ii. The stream type shall be defined as a choice between a VDM stream and a health stream.
1. The contents of the stream type are described in the section titled "Stream Type" under "Data Logger Complex Types."
iii. The stream type may be provided as part of the stream filter.

3. ***Stream Type***

a. **VDM Stream**
i. A VDM stream identifies the stream type as a stream containing VICTORY Data Messages (VDMs).
ii. A VDM stream shall be defined as a choice between a VDM type and an interface ID.
iii. The contents of the VDM stream are described in the section titled "VDM Stream" under "Data Logger Complex Types."
iv. The VDM stream may be provided as part of the stream type.

b. **Health Stream**
i. A health stream identifies the stream type as a stream containing health messages.
ii. A health stream shall be defined as a restricted string.
1. The string shall be a valid absolute object identifier (OID), a relative OID, or 'syslog'.
iii. A health stream of 'syslog' describes a stream type of only syslog messages.

iv. A health stream of an absolute OID or a relative OID describes the applicable Simple Network Management Protocol (SNMP) trap or notify messages.
v. The health stream may be provided as part of the stream type.

4. ***VDM Stream***
   a. **VDM Type**
      i. A VDM type identifies a message type within a VICTORY Data Message.
      ii. Each message in a VDM contains a type value in the header of the VDM as described in the section titled "Application Layer Message Encapsulation."
      iii. No matter the messages contained in the VDM, as long as one type header field matches the VDM type, that VDM shall be part of the VDM stream.
      iv. A VDM type shall be a valid VDM message type value.
         1. The acceptable values for the VDM type are described in the "Data Type" column of Table 298 in the appendix titled "VDM Data Type Values."
      v. The VDM type may be provided as part of the VDM stream.
   b. **Interface ID**
      i. The interface ID identifies a specific instance of a VICTORY Data Interface.
      ii. A VDM contains the interface ID as a field in the header as described in the section titled "Application Layer Message Encapsulation."
      iii. Any VDM that contains the provided interface ID shall be part of the VDM stream.
      iv. The interface ID shall be an integer value greater than zero.
      v. The interface ID may be provided as part of the VDM stream.
5. ***Time Filter***
   a. **Start Time**
      i. The start time of the time filter.
      ii. The start time shall use the string representation of the timestamp format as described in the section titled "Timestamp Format for Application Layer Data."
      iii. The start time may be provided as part of the time filter.
   b. **End Time**
      i. The end time of the time filter.
      ii. The end time shall use the string representation of the timestamp format as described in the section titled "Timestamp Format for Application Layer Data."
      iii. The end time may be provided as part of the time filter.
6. ***Size***
   a. **Size**
      i. The size identifies the value of the size.
      ii. The size shall be a real value greater than or equal to zero.
      iii. The size shall be provided as part of the size.
   b. **Units**
      i. The units identifies the data unit for the size.
      ii. The units shall be a string chosen from one of the following values:
         1. B
            a. Indicates a size in bytes, where 1 byte is equal to 8 bits.
         2. KB
            a. Indicates a size in kilobytes, where 1 kilobyte is equal to 1024 bytes.
         3. MB
            a. Indicates a size in megabytes, where 1 megabyte is equal to 1024^2 bytes.
         4. GB
            a. Indicates a size in gigabytes, where 1 gigabyte is equal to 1024^3 bytes.
         5. TB
            a. Indicates a size in terabytes, where 1 terabyte is equal to 1024^4 bytes.
      iii. The units shall be provided as part of the size.

18.6.4 Data Logger FTP Specification <15022–20180215, Pro>

When requesting a log from the data logger, the data logger shall provide a URI that a client can use to retrieve the actual log file.

1. The URI points to an endpoint on the data logger where a client can request the actual log file.

2. The URI provided shall allow an FTP client to retrieve the file using a standard FTP interface.
    a. An FTP connection is described in IETF RFC 959.
    b. The port contained in the URI shall be the URI to initiate the FTP command interface.
        i. The data logger FTP server shall only accept the following commands.
            1. USER NAME
            2. PASSWORD
            3. SYSTEM
            4. PASSIVE
            5. RETRIEVE
            6. QUIT
            7. TRANSFER MODE
                a. S (Stream) shall be the only accepted MODE command.
            8. REPRESENTATION TYPE
                a. I (Image Type) shall be the only accepted TYPE command.
                    i. Image Type transfers the file as pure binary data. This ensures there is no corruption when translating the data.
3. The URI of the log file shall either contain a scheme of FTP or FTPS.
    a. The scheme of FTPS implies the data logger implements Transport Layer Security (TLS) via "Data Confidentiality & Integrity for Management Interfaces."
    b. An FTPS connection is described in IETF RFC 4217.
        i. The data logger shall additionally accept the following commands when using FTPS.
            1. AUTHENTICATION/SECURITY MECHANISM
            2. DATA CHANNEL PROTECTION LEVEL
            3. PROTECTION BUFFER SIZE
4. The data logger shall keep the log file available for at least 60 seconds after requesting the log file URI.

18.6.5 Data Logger Pcap Log File Format <20078–20150331, Pro>

The log file returned via the FTP request shall be in the standard big-endian packet capture (pcap) file format.

The pcap file format is described below:

1. ***Global Header:*** 24 bytes
    a. The global header shall be at the beginning of the file and make up the first 24 bytes of the log.
    b. If the pcap file is formatted using little-endian, by convention, the bytes in the definitions below shall be swapped (e.g., 0xa1b2c3d4 is redefined as 0x4d3c2b1a).
    c. **Magic Number:** 4 bytes
        i. The magic number is used to identify the pcap file.
        ii. The magic number shall be defined as 0xa1b2c3d4.
    d. **Major Version:** 2 bytes
        i. The major version of the file format.
        ii. The major version shall be defined as 0x0002.
    e. **Minor Version:** 2 bytes
        i. The minor version of the file format.
        ii. The minor version shall be defined as 0x0004.
    f. **GMT to Local Correction:** 4 bytes
        i. The GMT to local correction of the timestamps of the packet headers.
        ii. Data logger log files shall use UTC as the GMT to local correction.
        iii. The GMT to local correction shall be defined as 0x00000000.
    g. **Accuracy:** 4 bytes
        i. The accuracy describes the number of significant figures on the timestamps.
        ii. The accuracy for the data logger is not defined, but shall be provided.
    h. **Snap Length:** 4 bytes
        i. The maximum length of the captured packets.
        ii. The data logger shall use the maximum UDP packet size + IP header.
        iii. The snap length shall be defined as 0x0000ffff.
    i. **Network:** 4 bytes
        i. The network describes the data link type.

ii. The data logger shall use a data link type of "LINKTYPE_ETHERNET".
iii. The network shall be defined as 0x00000001.

2. ***Packet Header:*** 16 bytes
    a. **Timestamp Seconds:** 4 bytes
        i. The timestamp the packet was received by the data logger in seconds.
    b. **Timestamp Microseconds:** 4 bytes
        i. The microsecond offset of the time the packet was received by the data logger.
    c. **Included Packet Length:** 4 bytes
        i. The size of the packet following this packet header.
    d. **Original Packet Length:** 4 bytes
        i. The original size of the packet when captured by the data logger.
        ii. This value should match the included packet length.
3. ***Packet Data***
    a. The packet data of the message received by the data logger.
    b. The packet data is the entire Ethernet packet.

A pcap file shall consist of the Global Header, followed by a Packet Header, the corresponding Packet Data, another Packet Header, the corresponding Packet Data, and so on.

The pcap file can be represented as follows:

```
Global Header|Packet Header1|Packet Data1|Packet Header2|Packet Data2| ... |Packet
HeaderN|Packet DataN
```

NOTE: *The vertical bar or pipe character ('|') in the above representation is used to delineate the fields in the pcap file. There is no delineating character between the fields in the actual pcap file.*

18.7 Audio Communications Interfaces

Audio interfaces may be implemented in VICTORY-compliant platforms in a number of different ways. Some deployments will feature full-blown digitized audio, using the VDB as the main transport mechanism for audio packets to appropriate digitized audio nodes. In other cases, however, the VDB will have a physical interface to only a single point in the voice communications component, and will be used primarily for providing physical connectivity between VDB nodes and the component. Regardless of how the specific platform implements voice communications, the standards identified in the following sections are to be implemented for integration with the VDB.

18.7.1 Streaming Audio Data Specification <20016–20131031, Pro>

The Streaming Audio Data Interface shall provide voice communication capabilities on the VDB via the Voice over Internet Protocol (VoIP). The Session Initiation Protocol (SIP) shall be used to control the VoIP session as defined in IETF RFC 3261. Port 5060 shall be used when UDP is used as the transport protocol and port 5061 shall be used when TLS over TCP is used as the transport protocol.

Streaming audio data shall be transported over the VDB via the Real-time Transport Protocol (RTP) as defined in IETF RFC 3551. UDP shall be used as the transport layer for RTP.

RTP ports shall be negotiated between the two SIP endpoints as defined in IETF RFC 4566: SDP: Session Description Protocol. The port numbers available for RTP/RTCP range from 1024 to 65535. If default values are indicated, IETF RFC 3551 recommends ports 5004 and 5005 for RTP and RTCP (RTP Control Protocol), respectively, but usable port numbers shall not be limited to those two.

The Streaming Audio Data Interface shall support, at a minimum, encoding streaming audio using the ITU-T uLaw G.711 codec. This encoding algorithm corresponds to RTP payload type '0'. Other codecs in addition to G.711 may also be supported, using appropriate payload types generally defined in IETF RFC 3551. SIP endpoints shall negotiate which codec is to be used as defined in IETF RFC 4566.

NOTE: *The specifications provided for streaming audio are only for "voice quality" audio (e.g., 8000 Hz, mono).*

18.7.2 Streaming Audio Data Management Parameters <15007–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the streaming audio data management parameters. A textual description of the streaming audio data management parameters is shown below.

1. ***Data Enabled***
    a. This parameter identifies whether the audio data stream managed by this interface is sending out audio data on the VDB.
    b. A Boolean value of True means "enabled" (sending) and a value of False means "disabled" (not sending).
    c. Operations include Get() and Set().
2. ***SIP Endpoint***
    a. This parameter identifies the URI of the SIP endpoint.
    b. The SIP endpoint shall be a valid URI as defined in IETF RFC 3986.
    c. The bounds of the SIP endpoint are described in the section titled "String Bounds" under "VICTORY Data Types."
    d. Operations include Get().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the streaming audio data management parameters along with a description of additional errors that were not specified but may be returned.*

18.7.3 Single-Instance Audio Data Specification <20067–20131031, Pro>

On a vehicle platform, applications may have a need to play audio alarms or notifications to alert users to various circumstances within or external to the system. One example of this behavior would be a threat detection sensor playing a "shot fired" message when it detects a gunshot. VICTORY allows for streaming such audio over the VDB to one or more audio sinks, which can then play the sound over a speaker, headset, etc. Streaming single-instance audio over the network eliminates the need for components to supply their own audio output devices or for the system to require extra hardware to handle audio switching among multiple components.

The Single-Instance Audio Data Interface shall provide streaming audio on the VDB via Real-time Transport Protocol (RTP) as defined in IETF RFC 3551. Additionally, to facilitate the setup of media sessions between audio sources and sinks, the data interface shall implement a Real Time Streaming Protocol (RTSP) server as defined in IETF RFC 2326. An audio sink can then set up a media session using RTSP, after which the data interface will stream its single-instance audio files as needed. The audio sink is not guaranteed to provide any RTSP control over the audio server beyond the minimum RTSP requests listed below.

The audio server shall support, at a minimum, the following RTSP requests:

- OPTIONS
- DESCRIBE (must support the Session Description Protocol [SDP] as defined in IETF RFC 4566)
- SETUP
- PLAY
- TEARDOWN

NOTE: *Guidelines for a minimal RTSP server can be found in Appendix D of IETF RFC 2326.*

For actual audio streaming, the audio server should prefer discontinuous transmission (silence suppression) over continuous transmission. That is, it should only send RTP packets when a sound is to be played. To ensure interoperability between any VICTORY audio source and sink, the Single-Instance Audio Data Interface shall support, at a minimum, encoding streaming audio using the ITU-T uLaw G.711 codec. Additional supported codecs can be included in the SDP description in order of priority.

18.7.4 Single-Instance Audio Data Management Parameters <15018–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the single-instance audio data management parameters. A textual description of the single-instance audio data management parameters is shown below.

1. ***RTSP Server Transport Configurations***
    a. This parameter identifies the transport configurations used for communicating with the RTSP server.

b. The contents of transport configurations are described in the section titled "Data Transport Configuration" under "VICTORY Data Interface Complex Types."
c. The following errors are defined for the RTSP server transport configurations parameter:
    i. Setting multiple RTSP server transport configurations with the same unique name shall result in a GENERIC_FAULT error.
    ii. Adding an RTSP server transport configuration with a pre-existing unique name shall result in a GENERIC_FAULT error.
    iii. Removing an RTSP server transport configuration with a unique name that does not match the unique name of one of the RTSP server transport configurations shall result in a GENERIC_FAULT error.
    iv. Removing the only RTSP server transport configuration shall result in a GENERIC_FAULT error.
    v. If network layer is included in the request, setting or adding an RTSP server transport configuration with a network layer outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
    vi. Setting, adding, or removing a data transport configuration with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    vii. Setting or adding a data transport configuration with a URI outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.
d. Operations include Get(), Set(), Add(), and Remove().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the single-instance audio data management parameters along with a description of additional errors that were not specified but may be returned.*

18.7.5 Single-Instance Audio Sink Interface <Informational>

The Single-Instance Audio Sink shall receive streaming audio on the VDB via Real-time Transport Protocol (RTP) as defined in IETF RFC 3551. Additionally, to facilitate the setup of the media session between audio source and sink, the sink shall implement a Real Time Streaming Protocol (RTSP) client as defined in IETF RFC 2326. The audio sink can then set up a media session with the audio server using RTSP, after which it will receive RTP packets as the server generates them. The audio sink is not guaranteed any RTSP control over the audio server beyond setting up/tearing down a session and opening the stream.

The audio sink shall implement, at a minimum, the following RTSP requests:

- DESCRIBE (must support SDP as defined in IETF RFC 4566)
- SETUP
- PLAY
- TEARDOWN

NOTE: *Guidelines for a minimal RTSP client can be found in Appendix D of IETF RFC 2326.*

To ensure interoperability between any VICTORY audio source and sink, the Single-Instance Audio Sink shall support, at a minimum, decoding streaming audio using the ITU-T uLaw G.711 codec. If both the audio server and sink mutually support other codecs, another desired codec can be chosen from the server's SDP description.

An example set of communications between a Single-Instance Audio service and sink is shown below:

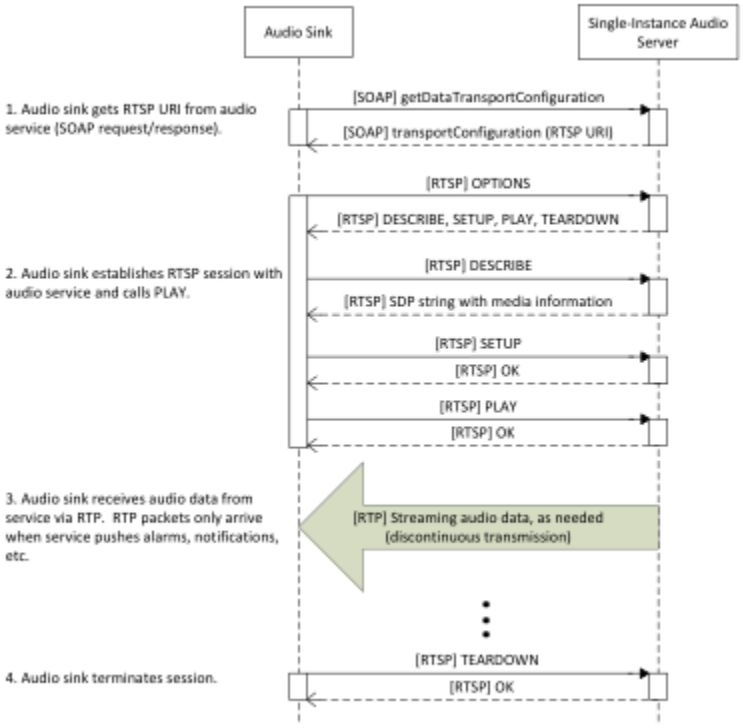

Figure 21. Audio Service-Sink Communication

18.7.6 Legacy Radio Device Management Parameters <15025–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the legacy radio device management parameters. A textual description of the legacy radio device management parameters is shown below.

NOTE: *The purpose of the Legacy Radio component type is not one of radio configuration, but rather its purpose is to provide simple information about the radio (this information could be displayed to the occupants of the vehicle). This specification defines parameters selected to address a number of different types of legacy radios, but they are not specific to any one radio, or even a family of radios. As these parameters are not specific to any particular radio, it is not possible to define radio-specific behaviors for the various get and set functions for the parameters defined below or how these parameters map to specific radios. It is encouraged that any adapter developed using the parameters defined below should be accompanied by documentation that defines the mapping between these parameters and the parameters specific to the radio. Furthermore, the behavior of the radio with respect to the get and set functions should be described in this accompanying documentation.*

1. ***Power***
    a. This parameter provides the legacy radio power level.
    b. The power shall be a string chosen from one of the following values:
        i. LOW
        ii. MEDIUM
        iii. HIGH
        iv. PA
    c. The following errors are defined for the power parameter:
        i. Setting the power level to a value not inside the list of valid legacy radio power levels may result in an OUT_OF_RANGE error.
    d. Operations include Get() and Set().
2. ***Selected Preset***
    a. This parameter provides the selected preset of the legacy radio.
    b. The contents of a preset are described in the section titled "Preset" under "Legacy Radio Complex Types."

c. The following errors are defined for the selected preset parameter:
   i. Setting the selected preset using a preset ID that does not match the preset ID of one of the presets available on the legacy radio shall result in a GENERIC_FAULT error.
   ii. Setting the selected preset with a preset ID outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
   iii. Getting the selected preset when the legacy radio does not have a preset selected shall result in a GENERIC_FAULT error.

> ***NOTE:*** *The error response should contain an additional error description stating that the radio does not currently have a preset selected.*

d. Operations include Get() and Set().

3. ***Selected Preset Status***
   a. This parameter provides the status of the preset that has been selected.
   b. The preset status shall be a string chosen from one of the following values:
      i. PENDING
         1. The preset has been selected but has not yet reached a condition that can be identified as READY or INOPERATIVE.
      ii. READY
         1. The selected preset is READY for use.
      iii. INOPERATIVE
         1. The selected preset has failed to reach a READY state. No further change to READY is expected.
      iv. NONE_SELECTED
         1. No preset has been selected.
   c. Operations include Get().
4. ***Available Presets***
   a. This parameter provides a list of the available radio presets and how they are configured.
   b. The contents of a preset are described in the section titled "Preset" under "Legacy Radio Complex Types."
   c. Operations include Get().
5. ***Radio Type***
   a. A parameter indicating the type of radio being managed by this interface.
   b. The radio type shall be a string chosen from one of the following values:
      i. AN_ARC_220
      ii. AN_ARC_231
      iii. AN_PRC_117F
      iv. AN_PRC_117G
      v. AN_PRC_148
      vi. AN_PRC_148_JEM
      vii. AN_PRC_150
      viii. AN_PRC_152
      ix. AN_PSC_5D
      x. RT_1523
      xi. OTHER
         1. If the value 'OTHER' is returned, it can be accompanied with additional text. The additional text will be added to the value 'OTHER' and a ';' will separate the additional information.

> ***NOTE:*** *'OTHER;AN_PRC_154' is an example of the use of 'OTHER' and additional information.*

   c. Operations include Get().
6. ***Available***
   a. This parameter indicates that the VICTORY interface is connected to the legacy radio and control and status information from the radio is accessible.
   b. A Boolean value of True means "available" and a value of False means "not available."
   c. Operations include Get().
7. ***Volume Step Maximum***
   a. This parameter identifies the maximum volume step level that can be used to control the volume of the legacy radio.
   b. The volume step maximum shall be an integer value greater than zero.
   c. Operations include Get().

8. ***Volume Step Setting***
    a. This parameter identifies the current volume step setting of the legacy radio.
    b. The volume step setting shall be an integer within the range of 0 to *n*, where *n* = ***Volume Step Maximum*** for the radio.
        i. A value of 0 shall indicate that the radio volume shall be rendered as quiet as possible.

> ***NOTE:*** *The management interface can use a legacy radio's separate mute control (if available) as an optimization, but it has to result in equivalent behavior.*

    c. The following errors are defined for the volume step setting parameter:
        i. Setting the volume step setting to a value less than 0 may result in an OUT_OF_RANGE error.
        ii. Setting the volume step setting to a value greater than *n* shall result in a NEAREST_VALUE error.
    d. Operations include Get() and Set().

> ***NOTE:*** *The Legacy Radio Management Interface, upon request to set the volume step setting, would translate the "step" number into the required discrete control value a radio requires. As examples, both the SINCGARS radio and the PRC–155 radio have exactly nine (9) discrete values (steps) at which volume can be set, however, in the SINCGARS radio the discrete values that control the volume range from 1–9, whereas in the PRC–155 the discrete values range from 0–8. Thus, a command to set volume to "step 5" in either the SINCGARS or PRC–155 would obtain the mid-point volume between steps 1–9, even though at the discrete output the translator for SINCGARS would command a setting that differs from the discrete output for the PRC–155.*

> ***NOTE:*** *It is not required that the Legacy Radio Management Interface use all of the potential range for the actual radio or that the mapping be linear. For example, if the lower half of the radio's range is not practically useful in the current installation, the interface can map the "1" volume level to the 50% volume setting for the radio.*

9. ***Zeroize Capability***
    a. This parameter provides the zeroize capability for the legacy radio.
    b. A Boolean value of True indicates the radio can be zeroized and a value of False indicates that it cannot be zeroized.
    c. Operations include Get().
10. ***Zeroize Responses***
    a. This parameter provides the zeroize responses available regarding the legacy radio. These responses are what can be returned regarding this legacy radio when the Get() function for the ***Zeroize Status*** parameter is used.
    b. The responses shall be a string comprised of one or more of the following values:
        i. ZEROIZED
        ii. NOT_ZEROIZED
        iii. IN_PROGRESS
        iv. FAILED
        v. NOT_REPORTABLE
    c. A response that contains more than one value from the above list shall use a ';' as the delimiter betweens values.
    d. Operations include Get().
11. ***Zeroize Status***
    a. This parameter provides the current status of the legacy radio with respect to its zeroize function.
    b. The zeroize status shall be a string chosen from one of the following values:
        i. ZEROIZED
        ii. NOT_ZEROIZED
        iii. IN_PROGRESS
        iv. FAILED
        v. NOT_REPORTABLE
    c. Operations include Get().
12. ***Zeroize***
    a. This parameter will zeroize the radio using the most complete and comprehensive form of zeroization available on the legacy radio.
    b. The following errors are defined for the zeroize parameter:
        i. Attempting to zeroize a legacy radio that does not support the zeroize function shall result in a NOT_SUPPORTED error.
    c. Operations include Set().

> ***NOTE:*** *Once the legacy radio has been zeroized, the behavior of the Legacy Radio Management Interface is undefined.*

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the legacy radio device management parameters along with a description of additional errors that were not specified but may be returned.*

18.7.7 Legacy Radio Complex Types <15026–20210409, Pro>

NOTE: *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

1. ***Preset***
    a. **Name**
        i. A human-readable name of the radio preset.
        ii. The bounds of the name are described in the section titled "String Bounds" under "VICTORY Data Types."
        iii. The name shall be provided as part of the preset.
    b. **Preset ID**
        i. The unique identifier of the preset.
        ii. The bounds of the preset ID are described in the section titled "String Bounds" under "VICTORY Data Types."
        iii. The preset ID shall be unique within the scope of the radio.

            NOTE: *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

        iv. The preset ID shall be provided as part of the preset.
    c. **Short Name**
        i. The shorthand version of the preset's name.
        ii. The bounds of the short name are described in the section titled "String Bounds" under "VICTORY Data Types."
        iii. The short name may be provided as part of the preset.
    d. **Status**
        i. The status of the preset.
        ii. The status of the preset shall be a string chosen from one of the following values:
            1. FUNCTIONAL
                a. The preset is defined and can be expected to reach a READY state upon selection for use.
            2. NOT_FUNCTIONAL
                a. The preset is defined, but cannot be expected to reach a READY state upon selection for use.
        iii. The status shall be provided as part of the preset.
    e. **Waveform**
        i. A human-readable identification of the waveform used by the preset.
        ii. The bounds of the waveform are described in the section titled "String Bounds" under "VICTORY Data Types."
        iii. The waveform may be provided as part of the preset.
    f. **Transmit Frequency**
        i. The transmit frequency of the preset.
        ii. The transmit frequency shall be an integer greater than zero.
        iii. The transmit frequency shall be in units of Hertz (Hz).
        iv. The transmit frequency may be provided as part of the preset.

            NOTE: *For presets without a fixed frequency (e.g., a frequency hopping preset), this parameter is not used.*

            1. If the transmit frequency is provided, the receive frequency shall also be provided.
    g. **Receive Frequency**
        i. The receive frequency of the preset.
        ii. The receive frequency shall be an integer greater than zero.
        iii. The receive frequency shall be in units of Hertz (Hz).
        iv. The receive frequency may be provided as part of the preset.

            NOTE: *For presets without a fixed frequency (e.g., a frequency hopping preset), this parameter is not used.*

1. If the receive frequency is provided, the transmit frequency shall also be provided.

h. **Mode**
   i. The mode of the radio preset.
   ii. The bounds of the mode are described in the section titled "String Bounds" under "VICTORY Data Types."
   iii. The mode may be provided as part of the preset.

i. **Security Mode**
   i. The security mode of the radio preset.
   ii. The security mode shall be a string chosen from one of the following values:
      1. PLAIN
      2. CIPHER
   iii. The security mode may be provided as part of the preset.

18.7.8 Intercom Interfaces

The vehicle intercom system is common to all ground combat vehicles and to many tactical vehicles. The vehicle intercom system enables voice communications among crew members and voice radios on the vehicle during noisy vehicle environments and provides audio alerts and warnings to the crew. The VICTORY architecture states that a vehicle intercom system provides a primary transport mechanism for audio data for communications system deployments that will not utilize the VDB. The intercom contains provisions for standard interfaces that the VDB nodes can utilize to communicate with, configure, and control the vehicle intercom system. For example, a WMI running on a VDB Common Display and Processing Unit may be used to send control commands, configuration commands, and status messages with the vehicle intercom system. These interfaces are identified in the following sections.

18.7.8.1 Intercom Management Parameters <15019–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the intercom management parameters. A textual description of the intercom management parameters is shown below.

1. ***Unit ID***
   a. This parameter provides the unique identifier of the intercom.
   b. The unit identifier shall be an integer greater than or equal to zero.
   c. Operations include Get().
2. ***Workgroup***
   a. This parameter provides the unique identifier of the intercom workgroup.
   b. The workgroup shall be an integer greater than or equal to zero.
   c. Operations include Get().
3. ***Type***
   a. This parameter provides the unique type of this intercom.
   b. The intercom shall be a string chosen from one of the following values:
      i. VIC_3
      ii. VIC_5
      iii. NETCOM_V
      iv. TACTICOM
      v. TOCNET
   c. Operations include Get().
4. ***Available***
   a. This parameter indicates whether or not the intercom is available.
   b. A Boolean value of True means "available" and a value of False means "not available."
   c. Operations include Get().
5. ***Audio Codecs***
   a. This parameter indicates the codecs used for audio compression on the intercom.
   b. The intercom may use more than one type of codec.
   c. The bounds of each audio codec are described in the section titled "String Bounds" under "VICTORY Data Types."
   d. At a minimum, the intercom shall support the G_711 codec.
   e. Operations include Get().
6. ***Loudspeaker Available***

   a. This parameter indicates whether or not there is a loudspeaker available on the intercom.
   b. A Boolean value of True means "available" and a value of False means "not available."
   c. Operations include Get().
7. ***Asset Groups***
   a. This parameter provides a list of asset groupings within the intercom.
   b. The contents of an asset group are described in the section titled "Intercom Asset Group" under "Intercom Complex Types."
   c. Setting asset groups results in the removal of all pre-existing asset groups and the addition of all asset groups provided in the request.
   d. The following errors are defined for the asset groups parameter:
      i. Setting multiple asset groups with the same intercom group ID shall result in a GENERIC_FAULT error.
      ii. Adding an asset group with a pre-existing intercom group ID shall result in a GENERIC_FAULT error.
      iii. Removing an asset group with an intercom group ID that does not match the intercom group ID of one of the asset groups shall result in a GENERIC_FAULT error.
      iv. Setting, adding, or removing an asset group with an intercom group ID outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
      v. Setting or adding an asset group with an asset outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
   e. Operations include Get(), Set(), Add(), and Remove().
8. ***Crew Stations***
   a. This parameter provides a list of crew stations on the intercom.
   b. The contents of a crew station are described in the section titled "Crew Station" under "Intercom Complex Types."
   c. Operations include Get().
9. ***Connected Radios***
   a. This parameter provides a list of radios connected to the intercom.
   b. The contents of a connected radio are described in the section titled "Connected Radio" under "Intercom Complex Types."
   c. Operations include Get().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the intercom management parameters along with a description of additional errors that were not specified but may be returned.*

18.7.8.2 Intercom Complex Types <20068–20210409, Pro>

NOTE: *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

1. ***Intercom Asset Group***
   a. **Intercom Group ID**
      i. The identifier of the intercom group.
      ii. The bounds of the intercom group identifier are described in the section titled "String Bounds" under "VICTORY Data Types."
      iii. The intercom group identifier shall be unique within all intercom group identifiers on the intercom.
      iv. The intercom group identifier shall be provided as part of the intercom asset group.
   b. **Assets**
      i. A list of assets within the intercom group.
      ii. The bounds of each asset are described in the section titled "String Bounds" under "VICTORY Data Types."
      iii. Assets may include identifiers of crew stations, radios, alarms, loudspeakers, external audio sources, etc.
      iv. The assets list shall be provided as part of the intercom asset group.
2. ***Crew Station***
   a. **Crew Station ID**
      i. The unique identifier of the crew station.

ii. The bounds of the crew station identifier are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The crew station identifier shall be unique within all crew station identifiers on the intercom.
iv. The crew station identifier shall be provided as part of the crew station.

b. **Volume**
i. The volume level for the crew station.
ii. This parameter shall be an integer greater than or equal to zero.
iii. The volume shall be provided as part of the crew station.

c. **PTT**
i. The push-to-talk (PTT) type.
ii. The PTT type shall be a string chosen from one of the following values:
1. PTT
a. Push-to-Talk
2. VOX
a. Voice Activated
3. LIVE
a. Live Microphone
iii. The PTT shall be provided as part of the crew station.

d. **Active Channels**
i. The list of working mode audio channels upon which the crew station will transmit audio.
ii. The bounds of each active channel are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The list of active channels may be provided as part of the crew station.
1. Not providing any active channels shall indicate the crew station is not transmitting on any audio channels.

e. **Receive Only Channels**
i. The list of monitor audio channels the crew station receives.
ii. The bounds of each receive only channel are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The list of receive only channels may be provided as part of the crew station.
1. Not providing any receive only channels shall indicate the crew station is not receiving any audio channels.

3. ***Connected Radio***

a. **Intercom Radio ID**
i. The identifier of the radio connected to the intercom.
ii. The bounds of the intercom radio identifier are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The radio identifier shall be unique within all intercom radio identifiers on the intercom.
iv. The intercom radio identifier shall be provided as part of the connected radio.

b. **Radio Type**
i. The type of radio connected to the intercom.
ii. The radio type shall be a string chosen from one of the following values:
1. AN_VRC_87
2. AN_VRC_88
3. AN_VRC_89
4. AN_VRC_90
5. AN_VRC_91
6. AN_VRC_92
7. AN_PRC_117
8. AN_PRC_119
9. SINCGARS
10. OTHER
iii. The radio type shall be provided as part of the connected radio.

18.8 Video and Imagery Situational Awareness Interfaces

Video and Imagery Situational Awareness (VISA) components are implemented on platforms to provide the warfighter with various views of the area surrounding the platform. VICTORY advocates that these video streams be placed on the VDB and be made available to any and all VDB network nodes. As such, the VISA components that are required to publish VISA data to the network shall have physical VDB interfaces.

Each design may include different physical components to implement VISA. However, most designs will incorporate physical components such as video sensors, pan and tilt modules, and video codecs (for digitization purposes). Furthermore, software components, like WMIs and configuration and control APIs, will be implemented for providing video streams to the user. The protocols identified in the following sections shall be used for providing streaming video on the VDB and management of VISA components.

18.8.1 VISA Data Interfaces

18.8.1.1 Standard Definition Streaming Video on the VDB <20017–20150331, Pro>

Streaming video on the VDB shall be encoded and formatted according to Recommended Practice (RP) 9720e - MISM, Low Bandwidth Motion Imagery, as described in the "Motion Imagery Standards Profile." Streaming video on the VDB shall be transported using Real-time Transport Protocol (RTP), IETF RFC 3550, over UDP. The video stream shall be controlled using the Real Time Streaming Protocol (RTSP), IETF RFC 2326.

> ***NOTE:*** *As stated in the "Motion Imagery Standards Profile," RP 9720e is suited for situational awareness applications where lower bandwidths and resolutions are tolerable. A device that provides video playback should support higher resolutions and data rates than source devices. With this in mind, the supported resolution at the streaming video sink device should be a minimum of 640x480 with a nominal data rate of 1.5 Mbps.*

A modified version of RP 9720e - MISM, Low Bandwidth Motion Imagery is provided in Figure 22 below for reference:

System Level	MISM							
	L2.2H	L2.1H	L2.1M	L2.0M	L1.3H	L1.2H	L1.1H	L1.0H
Common Description/ Intended Application	Medium / Distribution	Low-Medium / Distribution		Low / Distribution	Low / Distribution	Very Low / Distribution	Very Low / Distribution	Lowest / Distribution
System Attributes: Spatial Definition	Medium	Low - Medium		Low	Low	Low	Low	Very Low
System Attributes: Temporal Definition	Medium	Medium		Medium	Medium	Low	Very Low	Low
System Attributes: Generation Resiliency	Low	Low		Very Low	Very Low	Very Low	Very Low	Lowest
Applicable Standard (Note: Other Profiles /Practices may apply)	H.264 L2.2	H.264 L2.1	MPEG2 MP@ML	MPEG-1	H.264 L1.3	H.264 L1.2	H.264 L1.1	H.264 L1.0
Horizontal Resolution (Nominal)	640 - 720	320 - 352		320 - 352		320 - 352		160 - 176
Vertical Resolution (Nominal)	480 - 576	480 - 576		480 - 576	240 - 288p	240 - 288p		120 - 144p
Bit Depth (bits) (Nominal)	8	8		8		8		8
Frame Rate (FPS)	24 - 30	24 - 30		24 - 30		12 - 15	6 - 7	12 - 15
Compression Ratio (Nominal)	110:1	83:1	55:1	83:1	83:1	83:1	83:1	166:1
Data Rate (Nominal)	1.5 Mb/s	1.0 Mb/s	1.5 Mb/s	1.0 Mb/s	512 Kb/s	256 Kb/s	128 Kb/s	32 Kb/s
Data Rate Range (Kbits/s)	1,024 -1,500	768 - 1,024	1,024 - 1,500	768 - 1,024	384 - 768	192 - 384	56 - 192	< 56
Candidate Transport Channel (Nominal Rates)	T1/ E1	T1/ E1		T1/ E1	Partial T1/E1	Wireless	Wireless	Wireless
Allowed Transport Protocols	RTP/RTSP ~~Xon2~~	RTP/RTSP ~~Xon2~~		RTP/RTSP ~~Xon2~~	~~Xon2~~ RTP/RTSP	~~Xon2~~ RTP/RTSP	~~Xon2~~ RTP/RTSP	~~Xon2~~ RTP/RTSP
Recommended Transport Protocols	RTP/RTSP ~~Xon2~~	RTP/RTSP ~~Xon2~~		RTP/RTSP ~~Xon2~~	RTP/RTSP	RTP/RTSP	RTP/RTSP	RTP/RTSP

Figure 22. VICTORY Modified RP 9720e - MISM, Low Bandwidth Motion Imagery

18.8.1.2 Camera Orientation Data Specification <20018–20180215, Pro>

18.8.1.2.1 Camera Orientation Message Content

The Camera Gimbal Data Interface provides information regarding the current orientation of a camera gimbal. This information will be available to all VDB network nodes that require camera gimbal orientation data.

1. ***Timestamp***
    a. A timestamp corresponding to the time at which the orientation data was sampled.
    b. The timestamp shall be in the format as described in the section titled "Timestamp Format for Application Layer Data."
    c. The timestamp shall be provided as part of the camera orientation message.
2. ***Sequence Number***
    a. A monotonically increasing number that indicates the sequence number of the Camera Orientation message sent from a particular Camera Gimbal Data Interface.
    b. The sequence number shall increase by one (1) for each Camera Orientation message sent from the interface.
    c. When the sequence number reaches its maximum value, it shall roll over to its minimum value.

    NOTE: *The sequence number is intended to be used to detect messages that did not arrive or were dropped.*

    d. The sequence number shall be provided as part of the camera orientation message.
3. ***Interface ID***
    a. An identification number, unique within the scope of the VDB, used to identify a particular Camera Gimbal Data Interface.
    b. The interface ID shall be an integer value greater than or equal to zero.
    c. The interface ID shall be provided as part of the camera orientation message.
4. ***Camera Orientation***
    a. A detailed description of camera orientation data is provided in the section titled "Camera Orientation" under "VISA Complex Types."
    b. At least one of pan, tilt, or camera roll shall be provided as part of the camera orientation message.

18.8.1.2.2 Camera Orientation Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as a Camera Orientation VDM. The appendix titled "VICTORY Schemas" provides the name of the schema file. The file describes the Camera Orientation VDM, provides additional details for the VICTORY Data Types Schema, and describes the Camera Orientation type that generates the XML string used by the Camera Orientation VDM. An example Camera Orientation VDM is provided below.

```
<vdm:messages xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:vm="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <cameraOrientationMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z"
        sequenceNumber="22" >
        <vdt:cameraOrientation>
            <vdt:pan>5.1332</vdt:pan>
            <vdt:tilt>12.1353</vdt:tilt>
        </vdt:cameraOrientation>
    </cameraOrientationMessage>
</vdm:messages>
```

Figure 23. Example Camera Orientation VDM

18.8.1.3 Home Position Data Specification <20082–20180215, Pro>

18.8.1.3.1 Home Position Message Content

The Camera Gimbal Data Interface provides information regarding the current home position of a camera gimbal. This information will be available to all VDB network nodes that require camera gimbal orientation data.

Updating the home position of the camera gimbal being managed by this interface shall result in a home position VDM being sent to the VDB with the updated home position of the camera gimbal.

1. ***Timestamp***
    a. A timestamp corresponding to the time at which the home position data was updated.
    b. The timestamp shall be in the format as described in the section titled "Timestamp Format for Application Layer Data."
    c. The timestamp shall be provided as part of the home position message.
2. ***Sequence Number***
    a. A monotonically increasing number that indicates the sequence number of the Home Position message sent from a particular Camera Gimbal Data Interface.
    b. The sequence number shall increase by one (1) for each Home Position message sent from the interface.
    c. When the sequence number reaches its maximum value, it shall roll over to its minimum value.

    > ***NOTE:*** *The sequence number is intended to be used to detect messages that did not arrive or were dropped.*

    d. The sequence number shall be provided as part of the home position message.
3. ***Interface ID***
    a. An identification number, unique within the scope of the VDB, used to identify a particular Camera Gimbal Data Interface.
    b. The interface ID shall be an integer value greater than or equal to zero.
    c. The interface ID shall be provided as part of the home position message.
4. ***Home Position***
    a. A detailed description of home position data is provided in the section titled "Home Position" under "VISA Complex Types."
    b. At least one of absolute pan, absolute tilt, or absolute camera roll shall be provided as part of the home position message.

18.8.1.3.2 Home Position Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as a Home Position VDM. The appendix titled "VICTORY Schemas" provides the name of the schema file. The file describes the Home Position VDM, provides additional details for the VICTORY Data Types Schema, and describes the Home Position type that generates the XML string used by the Home Position VDM. An example Home Position VDM is provided below.

```
<vdm:messages xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:vm="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <homePositionMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z"
        sequenceNumber="22" >
        <vdt:homePosition>
            <vdt:absolutePan>5.1332</vdt:absolutePan>
            <vdt:absoluteTilt>12.1353</vdt:absoluteTilt>
        </vdt:homePosition>
```

```
    </homePositionMessage>
</vdm:messages>
```

Figure 24. Example Home Position VDM

18.8.2 VISA Management Interfaces

18.8.2.1 Camera Gimbal Management

18.8.2.1.1 Camera Gimbal Data Management Parameters <15028–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the camera gimbal data management parameters. A textual description of the camera gimbal data management parameters is shown below.

1. ***Data Enabled***
    a. Identifies whether the Camera Gimbal Data Interface is sending out data on the VDB for the specified type of camera gimbal data.
    b. The type of camera gimbal data shall be a string chosen from one of the following values:
        i. CAMERA_ORIENTATION
        ii. HOME_POSITION
    c. Each type of camera gimbal data shall contain a data enabled flag.
        i. A Boolean value of True means "enabled" (sending) and a value of False means "disabled" (not sending).
    d. The data enabled flag for each type of camera gimbal data shall be controlled independently.
    e. If enabled is True and the type of camera gimbal data is HOME_POSITION, a VDM shall be sent every time the home position has been updated.
    f. If enabled is True and the type of camera gimbal data is CAMERA_ORIENTATION, a VDM shall be sent at the rate specified by the Update Period parameter.
    g. The following errors are defined for the data enabled parameter:
        i. Retrieving or setting the data enabled flag with a type of camera gimbal data not inside the list of valid camera gimbal data types may result in an OUT_OF_RANGE error.
    h. Operations include Get() and Set().
2. ***Update Period***
    a. Identifies the frequency at which CAMERA_ORIENTATION VDM updates are sent to the VDB.
    b. The update period shall be in units of milliseconds.
    c. The update period shall be a real value greater than zero.
    d. The following errors are defined for the update period parameter:
        i. Setting the update period to a value outside the bounds defined by this specification document may result in an OUT_OF_RANGE error.
        ii. Setting the update period to a value less than the minimum update period shall result in a NEAREST_VALUE error.
            1. The update period shall be changed to the minimum update period and shall be sent back in the response.
    e. Operations include Get() and Set().
3. ***Minimum Update Period***
    a. Identifies the maximum frequency at which the CAMERA_ORIENTATION VDM updates can be sent to the VDB from the Camera Gimbal Data Interface.
    b. The minimum update period shall be in units of milliseconds.
    c. The minimum update period shall be a real value greater than zero.
    d. Operations include Get().
4. ***Data Transport Configuration***
    a. Identifies the transport configurations used by the Camera Gimbal Data Interface for sending out VDMs to the VDB for the specified type of camera gimbal data.
    b. The type of camera gimbal data shall be a string chosen from one of the following values:
        i. CAMERA_ORIENTATION

ii. HOME_POSITION

c. Each type of camera gimbal data shall contain a set of data transport configurations.
   i. The contents of data transport configurations are described in the section titled "Data Transport Configuration" under "VICTORY Data Interface Complex Types."
d. The data transport configurations for each type of camera gimbal data shall be controlled independently.
e. Setting the data transport configurations results in the removal of all pre-existing transport configurations and the addition of all transport configurations provided in the request.
f. The following errors are defined for the data transport configurations parameter:
   i. Setting multiple data transport configurations with the same unique name shall result in a GENERIC_FAULT error.
   ii. Adding a data transport configuration with a pre-existing unique name shall result in a GENERIC_FAULT error.
   iii. Removing a data transport configuration with a unique name that does not match the unique name of one of the camera gimbal data transport configurations shall result in a GENERIC_FAULT error.
   iv. Removing the only data transport configuration for the specified type of camera gimbal data shall result in a GENERIC_FAULT error.
   v. Retrieving, setting, adding, or removing data transport configurations with a type of camera gimbal data not inside the list of valid camera gimbal data types may result in an OUT_OF_RANGE error.
   vi. If network layer is included in the request, setting or adding a data transport configuration with a network layer outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
   vii. Setting, adding, or removing a data transport configuration with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
   viii. Setting or adding a data transport configuration with a URI outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.
g. Operations include Get(), Set(), Add(), and Remove().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the camera gimbal data management parameters along with a description of additional errors that were not specified but may be returned.*

18.8.2.1.2 Camera Gimbal Device Management Parameters <15020–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the camera gimbal device management parameters. A textual description of the camera gimbal device management parameters is shown below.

1. ***Relative Camera Orientation***
   a. This parameter identifies the current relative orientation of the camera gimbal.
   b. The contents of this parameter are described in the section titled "Camera Orientation" under "VISA Complex Types."
   c. At least one of pan, tilt, or camera roll shall be provided when retrieving or setting the relative camera orientation of the camera gimbal.
   d. The following errors are defined for the relative camera orientation parameter:
      i. Setting the camera orientation with a pan, tilt, or camera roll value outside the range defined by this specification document may result in an OUT_OF_RANGE error.
      ii. Setting the camera orientation with a pan, tilt, or camera roll value inside the range defined by this specification document but unrecognized by the camera gimbal shall result in a NEAREST_VALUE error.
         1. The values not recognized by the camera gimbal shall be changed to the nearest value recognized by the camera gimbal that is still within the range defined by this specification document and shall be sent back in the response.
   e. Operations include Get() and Set().
2. ***Relative Camera Position***
   a. This parameter identifies the current position of the camera gimbal, relative to the camera sensor's frame of reference.
   b. The contents of this parameter are described in the section titled "Relative Camera Position" under "VISA Complex Types."

c. At least one of x, y, or z shall be provided when retrieving or setting the relative camera position of the camera gimbal.
d. Operations include Get() and Set().

3. ***Slew***
   a. This parameter starts and stops the slewing of the camera gimbal.
   b. The slew shall use a camera-centric right-handed coordinate system, where the origin is at the center of the camera gimbal's axes of rotation.
   c. When the camera gimbal axes are in an orthogonal position, they shall be defined as follows:
      i. X-axis positive "forward," or "into" the image produced by the camera.
      ii. Y-axis positive "leftward."
      iii. Z-axis positive "upward."

      > ***NOTE:*** *Since axes are affected by rotation in a camera gimbal (not always orthogonal), and dependent on order of operations (which is not specified, because it is camera-specific), the camera gimbal axis definitions are only useful as a starting place to define general directionality.*

   d. When starting a slew, the camera gimbal shall rotate about one or more of the axes.
      i. The contents of a rotation are described in the section titled "Rotation" under "VISA Complex Types."
   e. Rotations about all axes shall be relative to current position.
   f. Starting a slew shall result in the stopping of all current motion, if in progress, and replace it with the newly specified motion.
   g. The following errors are defined for the slew parameter:
      i. Starting a slew with multiple rotations of the same type of rotation shall result in a GENERIC_FAULT error.
      ii. Starting or stopping a slew with a type of rotation outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
      iii. Starting a slew with a direction outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
      iv. Starting a slew with a type of rate outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
      v. Starting a slew with a rate, distance, or time value outside the range specified by this document may result an OUT_OF_RANGE error.
      vi. Starting a slew with a rate, distance, or time value inside the range defined by this specification document but unrecognized by the camera gimbal shall result in a NEAREST_VALUE error.
         1. The values not recognized by the camera gimbal shall be changed to the nearest value recognized by the camera gimbal that is still within the range defined by this specification document and shall be sent back in the response.
   h. Operations include Start() and Stop().

4. ***Home Position***
   a. This parameter manages the home position of the camera gimbal.
   b. The contents of this parameter are described in the section titled "Home Position" under "VISA Complex Types."

      > ***NOTE:*** *The home position values are camera gimbal-specific, so this document will only provide general requirements. How the camera gimbal interprets these values is completely up to the camera gimbal.*

   c. Setting a home position with no values shall result in the camera gimbal capturing its current location and setting those values to the home position.
      i. When setting the home position to any location other than the current location of the camera gimbal, at least one of absolute pan, absolute tilt, and absolute camera roll shall be provided.
   d. At least one of absolute pan, absolute tilt, or absolute camera roll shall be provided when retrieving the home position of the camera gimbal.
   e. The following errors are defined for the home position parameter:
      i. Setting a home position with an absolute pan, absolute tilt, or absolute camera roll value outside the range specified by this document may result an OUT_OF_RANGE error.
      ii. Setting a home position with an absolute pan, absolute tilt, or absolute camera roll value inside the range defined by this specification document but unrecognized by the camera gimbal shall result in a NEAREST_VALUE error.

1. The values not recognized by the camera gimbal shall be changed to the nearest value recognized by the camera gimbal that is still within the range defined by this specification document and shall be sent back in the response.

iii. Setting the home position of a camera gimbal that does not support manual control of the home position shall result in a NOT_SUPPORTED error.

f. Operations include Get(), Set(), and GoTo().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the camera gimbal device management parameters along with a description of additional errors that were not specified but may be returned.*

18.8.2.2 Video or Image Sensor Management

18.8.2.2.1 Video or Image Sensor Management Parameters <15008–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the video or image sensor management parameters. A textual description of the video or image sensor management parameters is shown below.

1. ***Focus***
    a. This parameter identifies the focus of the camera being managed by this interface, where focus is defined as an apparatus used to adjust the focal length of an optical system in order to make an image distinct or clear.
    b. The focus shall be a percentage.
    c. The focus shall be a real value greater than or equal to zero (minimum focal length) and less than or equal to 100 (maximum focal length).
    d. The following errors are defined for the focus parameter:
        i. Setting the focus to a value outside the range defined by this specification document may result in an OUT_OF_RANGE error.
        ii. Setting the focus to a value inside the range defined by this specification document but unrecognized by the camera shall result in a NEAREST_VALUE error.
            1. The focus shall be changed to the nearest value recognized by the camera that is still within the range defined by this specification document and shall be sent back in the response.
        iii. Setting the focus when auto-focus is set to True shall result in a GENERIC_FAULT error.
    e. Operations include Get() and Set().
2. ***Zoom***
    a. This parameter identifies the zoom of the camera being managed by this interface, where zoom is defined as a rapid change in the focal length, allowing rapid change in the size of an image.
    b. The zoom shall be a percentage.
    c. The zoom shall be a real value greater than or equal to zero (minimum zoom) and less than or equal to 100 (maximum zoom).
    d. The following errors are defined for the zoom parameter:
        i. Setting the zoom to a value outside the range defined by this specification document may result in an OUT_OF_RANGE error.
        ii. Setting the zoom to a value inside the range defined by this specification document but unrecognized by the camera shall result in a NEAREST_VALUE error.
            1. The zoom shall be changed to the nearest value recognized by the camera that is still within the range defined by this specification document and shall be sent back in the response.
    e. Operations include Get() and Set().
3. ***Field of View***
    a. This parameter identifies the current field of view of the camera being managed by this interface, where field of view is defined as the area of the inspection captured on the camera's imager.
    b. The field of view shall be in units of degrees.
    c. The field of view shall be a real value greater than or equal to zero (minimum field of view) and less than or equal to 180 (maximum field of view).
    d. The following errors are defined for the field of view parameter:

i. Setting the field of view to a value outside the range defined by this specification document may result in an OUT_OF_RANGE error.
ii. Setting the field of view to a value inside the range defined by this specification document but unrecognized by the camera shall result in a NEAREST_VALUE error.
1. The field of view shall be changed to the nearest value recognized by the camera that is still within the range defined by this specification document and shall be sent back in the response.

e. Operations include Get() and Set().

> ***NOTE:*** *The field of view can be calculated using the size of the sensor in the direction measured, d, and focal length, f.*
> *Field of View = 2*arctan(d/2f).*

4. ***Exposure***
   a. This parameter identifies how long to add up photons on the focal point.
   b. The exposure shall be in units of microseconds.
   c. The exposure shall be a real value greater than or equal to zero.
   d. The following errors are defined for the exposure parameter:
      i. Setting the exposure to a value outside the range defined by this specification document may result in an OUT_OF_RANGE error.
      ii. Setting the exposure to a value inside the range defined by the specification document but unrecognized by the camera shall result in a NEAREST_VALUE error.
         1. The exposure shall be changed to the nearest value recognized by the camera that is still within the range defined by this specification document and shall be sent back in the response.
      iii. Retrieving or setting the exposure of a camera that does not provide exposure shall result in a NOT_SUPPORTED error.
      iv. Setting the exposure of a camera that does not support manual control of exposure shall result in a NOT_SUPPORTED error.
      v. Setting the exposure when auto-exposure is set to True shall result in a GENERIC_FAULT error.
   e. Operations include Get() and Set().
5. ***Auto Exposure***
   a. This parameter identifies whether or not auto-exposure is turned on.
   b. A Boolean value of True means auto-exposure is on and a value of False means auto-exposure is off.
   c. The following errors are defined for the auto-exposure parameter:
      i. Retrieving or setting the auto-exposure of a camera that does not provide auto-exposure shall result in a NOT_SUPPORTED error.
      ii. Setting the auto-exposure of a camera that does not support manual control of auto-exposure shall result in a NOT_SUPPORTED error.
   d. Operations include Get() and Set().
6. ***White Balance***
   a. This parameter identifies the white balance of the camera.
   b. The white balance shall be represented by either a preset or a temperature.
      i. If preset, the white balance shall be a string chosen from one of the following values:
         1. AUTO
         2. TUNGSTEN
         3. FLUORESCENT
         4. DAYLIGHT_SUNNY
         5. CLOUDY
         6. FLASH
         7. SHADE
      ii. If temperature, the white balance shall be in units of Kelvin.
      iii. If temperature, the white balance shall be a real value greater than or equal to zero.
   c. The following errors are defined for the white balance parameter:
      i. Setting the white balance preset to a preset outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
      ii. Setting the white balance temperature to a temperature outside the range defined by this specification document may result in an OUT_OF_RANGE error.
      iii. Setting the white balance temperature to a value inside the range defined by the specification document but unrecognized by the camera shall result in a NEAREST_VALUE error.

1. The white balance temperature shall be changed to the nearest value recognized by the camera that is still within the range defined by this specification document and shall be sent back in the response.

iv. Retrieving, setting, or capturing the white balance of a camera that does not provide white balance shall result in a NOT_SUPPORTED error.

v. Setting or capturing the white balance of a camera that does not support manual control of white balance shall result in a NOT_SUPPORTED error.

d. Operations include Get(), Set(), and Capture().

7. ***Iris***
   a. This parameter identifies the openness of the camera iris.
   b. The iris shall be a percentage.
   c. The iris shall be a real value greater than or equal to zero (fully closed) and less than or equal to 100 (fully open).
   d. The following errors are defined for the iris parameter:
      i. Setting the iris to a value outside the range defined by this specification document may result in an OUT_OF_RANGE error.
      ii. Setting the iris to a value inside the range defined by the specification document but unrecognized by the camera shall result in a NEAREST_VALUE error.
         1. The iris shall be changed to the nearest value recognized by the camera that is still within the range defined by this specification document and shall be sent back in the response.
      iii. Retrieving or setting the iris of a camera that does not provide iris shall result in a NOT_SUPPORTED error.
      iv. Setting the iris of a camera that does not support manual control of iris shall result in a NOT_SUPPORTED error.
   e. Operations include Get() and Set().
8. ***Integration Time***
   a. This parameter identifies the integration time of the camera.
   b. The integration time shall be in units of microseconds.
   c. The integration time shall be a real value greater than or equal to zero.
   d. The following errors are defined for the integration time parameter:
      i. Setting the integration time to a value outside the range defined by this specification document may result in an OUT_OF_RANGE error.
      ii. Setting the integration time to a value inside the range defined by the specification document but unrecognized by the camera shall result in a NEAREST_VALUE error.
         1. The integration time shall be changed to the nearest value recognized by the camera that is still within the range defined by this specification document and shall be sent back in the response.
      iii. Retrieving or setting the integration time of a camera that does not provide integration time shall result in a NOT_SUPPORTED error.
      iv. Setting the integration time of a camera that has a factory set integration time shall result in a NOT_SUPPORTED error.
   e. Operations include Get() and Set().
9. ***Auto Focus***
   a. This parameter identifies whether or not auto-focus is turned on.
   b. A Boolean value of True means auto-focus is on and a value of False means auto-focus is off.
   c. The following errors are defined for the auto-focus parameter:
      i. Retrieving or setting the auto-focus of a camera that does not provide auto-focus shall result in a NOT_SUPPORTED error.
      ii. Setting the auto-focus of a camera that does not support manual control of auto-focus shall result in a NOT_SUPPORTED error.
   d. Operations include Get() and Set().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the video or image sensor management parameters along with a description of additional errors that were not specified but may be returned.*

18.8.2.2.2 IR Video or Image Sensor Management Parameters <15027–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the IR (infrared) video or image sensor management parameters. A textual description of the IR video or image sensor management parameters is shown below.

Even though each of the parameters inside this specification allows for a NOT_SUPPORTED error to be returned, NOT_SUPPORTED shall not be returned if the IR video or image sensor supports the parameter.

1. ***Gain***
    a. This parameter identifies the multiplicative offset of camera pixels.
    b. The gain shall be a percentage.
    c. The gain shall be a real value greater than or equal to zero (minimum gain) and less than or equal to 100 (maximum gain).
    d. The following errors are defined for the gain parameter:
        i. Setting the gain to a value outside the range defined by this specification document may result in an OUT_OF_RANGE error.
        ii. Setting the gain to a value inside the range defined by the specification document but unrecognized by the camera shall result in a NEAREST_VALUE error.
            1. The gain shall be changed to the nearest value recognized by the camera that is still within the range defined by this specification document and shall be sent back in the response.
        iii. Retrieving or setting the gain of a camera that does not provide gain shall result in a NOT_SUPPORTED error.
        iv. Setting the gain of a camera that does not support manual control of gain shall result in a NOT_SUPPORTED error.
    e. Operations include Get() and Set().
2. ***Level***
    a. This parameter identifies the additive offset of camera pixels.
    b. The level shall be a percentage.
    c. The level shall be a real value greater than or equal to zero (minimum level) and less than or equal to 100 (maximum level).
    d. The following errors are defined for the level parameter:
        i. Setting the level to a value outside the range defined by this specification document may result in an OUT_OF_RANGE error.
        ii. Setting the level to a value inside the range defined by the specification document but unrecognized by the camera shall result in a NEAREST_VALUE error.
            1. The level shall be changed to the nearest value recognized by the camera that is still within the range defined by this specification document and shall be sent back in the response.
        iii. Retrieving or setting the level of a camera that does not provide level shall result in a NOT_SUPPORTED error.
        iv. Setting the level of a camera that does not support manual control of level shall result in a NOT_SUPPORTED error.
    e. Operations include Get() and Set().
3. ***Polarity***
    a. This parameter identifies the polarity/palette of the camera.
    b. The polarity shall be a string chosen from one of the following values:
        i. WHITE_HOT
        ii. BLACK_HOT
    c. The following errors are defined for the polarity parameter:
        i. Setting the polarity to a polarity outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
    d. Retrieving or setting the polarity of a camera that does not provide polarity shall result in a NOT_SUPPORTED error.
    e. Setting the polarity of a camera that does not support manual control of polarity shall result in a NOT_SUPPORTED error.
    f. Operations include Get() and Set().
4. ***Supported Color Table Size***

a. This parameter identifies the number of color tuples that comprise a camera color table.
b. The size shall be an integer value greater than zero.
c. The following errors are defined for the supported color table size parameter:
    i. Retrieving the supported color table size of a camera that has its own internal set of color tables shall result in a NOT_SUPPORTED error.
    ii. Retrieving the supported color table size of a camera that does not provide color tables shall result in a NOT_SUPPORTED error.
d. Operations include Get().

5. ***Color Tables***
    a. This parameter identifies the color tables available for the camera.
    b. The contents of a color table are described in the section titled "Color Table" under "VISA Complex Types."
    c. Setting the color tables shall result in the removal of all pre-existing color tables and the addition of all color tables provided in the request.
    d. Removing the Current Color Table or setting new color tables that results in the removal of the Current Color Table shall result in there being no color table assigned as the Current Color Table.
        i. Retrieving the Current Color Table results in no color table being provided in the response until a new Current Color Table has been set.
    e. The following errors are defined for the color tables parameter:
        i. Setting multiple color tables with the same camera table unique identifier shall result in a GENERIC_FAULT error.
        ii. Adding a color table with a pre-existing color table unique identifier shall result in a GENERIC_FAULT error.
        iii. Removing a color table using a unique identifier that does not match the unique identifier of one of the camera color tables shall result in a GENERIC_FAULT error.
        iv. Adding or setting a color table with an RGB value outside the range defined by this specification document may result in an OUT_OF_RANGE error.
        v. Adding or setting a color table with an amount of color tuples not equal to the Supported Color Table Size shall result in a GENERIC_FAULT error.
        vi. Adding, setting, or removing a color table with a unique identifier outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
        vii. Setting, adding, or removing color tables on a camera that has its own internal set of color tables shall result in a NOT_SUPPORTED error.
        viii. Retrieving, setting, adding, or removing color tables of a camera that does not provide color tables shall result in a NOT_SUPPORTED error.
    f. Operations include Get(), Set(), Add(), and Remove().

6. ***Current Color Table***
    a. This parameter identifies the color table being used by the camera.
    b. The color table shall be represented by its unique identifier.
    c. Retrieving the current color table before one has been set or after the removal of the color table previously set as the current table shall result in no color table being provided in the response.
    d. If the color table set as the current color table exists inside the list of Color Tables, retrieving the current color table shall result in the current color table being provided in the response.
    e. The following errors are defined for the current color table parameter:
        i. Setting the current color table using a unique identifier that does not match any of the color table unique identifiers shall result in a GENERIC_FAULT error.
        ii. Retrieving or setting the current color table of a camera that does not provide color tables shall result in a NOT_SUPPORTED error.
        iii. Setting the current color table of a camera that does not support external control of the color table being used by the camera shall result in a NOT_SUPPORTED error.
        iv. Setting the current color table of a camera that has its own internal set of color tables and only contains one color table shall result in a NOT_SUPPORTED error.
        v. Setting the current color table with a unique identifier outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    f. Operations include Get() and Set().

7. ***NUC***
    a. This parameter executes the factory-calibrated non-uniformity compensation (NUC) coefficients to maintain a high image quality.
    b. The following errors are defined for the NUC parameter:

i. Executing a NUC command of a camera that does not support a manual NUC shall result in a NOT_SUPPORTED error.

c. Operations include Execute().

8. ***Shutter***
   a. This parameter identifies the amount of optical path blocking.
   b. The shutter shall be a percentage.
   c. The shutter shall be a real value greater than or equal to zero (fully blocked) and less than or equal to 100 (fully open).
   d. The following errors are defined for the shutter parameter:
      i. Setting the shutter to a value outside the range defined by this specification document may result in an OUT_OF_RANGE error.
      ii. Setting the shutter to a value inside the range defined by the specification document but unrecognized by the camera shall result in a NEAREST_VALUE error.
         1. The shutter shall be changed to the nearest value recognized by the camera that is still within the range defined by this specification document and shall be sent back in the response.
      iii. Retrieving or setting the shutter of a camera that does not provide shutter shall result in a NOT_SUPPORTED error.
      iv. Setting the shutter of a camera that does not support manual control of shutter shall result in a NOT_SUPPORTED error.
   e. Operations include Get() and Set().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the IR video or image sensor management parameters along with a description of additional errors that were not specified but may be returned.*

18.8.2.3 Video Data Stream Management

18.8.2.3.1 Video Data Stream Management Parameters <15010–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the video data stream management parameters. A textual description of the video data stream management parameters is shown below.

1. ***Data Enabled***
   a. This parameter identifies whether the video data stream managed by this interface is sending out video data on the Data Bus.
   b. A Boolean value of True means "enabled" (sending) and a value of False means "disabled" (not sending).
   c. Operations include Get() and Set().
2. ***Data Transport Configurations***
   a. This parameter identifies the transport configurations used by this interface for sending out video data on the VDB.
   b. The contents of data transport configurations are described in the section titled "Data Transport Configuration" under "VICTORY Data Interface Complex Types."
   c. The following errors are defined for the data transport configurations parameter:
      i. Setting multiple data transport configurations with the same unique name shall result in a GENERIC_FAULT error.
      ii. If network layer is included in the request, setting a data transport configuration with a network layer outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
      iii. Setting a data transport configuration with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
      iv. Setting a data transport configuration with a URI outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.
   d. Operations include Get() and Set().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the video data stream management parameters along with a description of additional errors that were not specified but may be returned.*

18.8.2.4 Video Encoder Management

18.8.2.4.1 Video Encoder Management Parameters <15012–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the video encoder management parameters. A textual description of the video encoder management parameters is shown below.

1. ***Video Resolution***
    a. The number of distinct pixels in each dimension that can be displayed on the video encoder.
    b. This parameter may provide vertical and horizontal resolution.
    c. The vertical and horizontal resolutions shall be an integer greater than zero.
    d. The vertical and horizontal resolutions shall be in units of pixels.
    e. The following errors are defined for the video resolution parameter:
        i. Setting the vertical or horizontal resolution to a value less than or equal to zero may result in an OUT_OF_RANGE error.
        ii. Setting the vertical or horizontal resolution to a value within the range defined by this specification document, but is not a resolution that can be handled by the video encoder, shall result in a NEAREST_VALUE error.
            1. The actual resolution values sent to the video encoder shall be provided in the response.
    f. Operations include Get() and Set().
2. ***Video Frame Rate***
    a. The frequency at which the video encoder produces unique consecutive images.
    b. The frame rate shall be a real value greater than zero.
    c. The frame rate shall be in units of frames per second.
    d. The following errors are defined for the video frame rate parameter:
        i. Setting the frame rate to a value less than or equal to zero may result in an OUT_OF_RANGE error.
        ii. Setting the frame rate to a value within the range defined by this specification document, but is not a frame rate that can be handled by the video encoder, shall result in a NEAREST_VALUE error.
        iii. The actual frame rate value sent to the video encoder shall be provided in the response.
    e. Operations include Get() and Set().
3. ***Video Pixel Format***
    a. The pixel format of the video encoder.
    b. When available, the pixel format should be chosen from one of the pixel formats defined in the appendix titled "Video Encoder Pixel Formats."
    c. The bounds of the pixel format are described in the section titled "String Bounds" under "VICTORY Data Types."
    d. The following errors are defined for the pixel format parameter:
        i. Retrieving the pixel format of an encoder that does not provide pixel format shall result in a NOT_SUPPORTED error.
    e. Operations include Get().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the video encoder management parameters along with a description of additional errors that were not specified but may be returned.*

18.8.3 VISA Complex Types <15011–20210409, Pro>

NOTE: *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

1. ***Relative Camera Position***
    a. **X**
        i. A detailed description of the X coordinate is provided in the section titled "Relative Cartesian Coordinates" under "VICTORY Data Types."
        ii. The X coordinate may be provided as part of the relative camera position.
    b. **Y**
        i. A detailed description of the Y coordinate is provided in the section titled "Relative Cartesian Coordinates" under "VICTORY Data Types."

ii. The Y coordinate may be provided as part of the relative camera position.

c. **Z**

i. A detailed description of the Z coordinate is provided in the section titled "Relative Cartesian Coordinates" under "VICTORY Data Types."

ii. The Z coordinate may be provided as part of the relative camera position.

2. ***Camera Orientation***

a. **Pan**

i. The rotation angle about the Z-axis, where the Z-axis is positive "upward."

ii. The pan shall be in units of arc degrees.

iii. The pan shall be a real value, where positive pan is left and negative pan is right.

iv. The pan shall be relative to the home position of the camera gimbal.

v. The pan may be provided as part of the camera orientation.

b. **Tilt**

i. The rotation angle about the Y-axis, where the Y-axis is positive "leftward."

ii. The tilt shall be in units of arc degrees.

iii. The tilt shall be a real value, where positive tilt is down and negative tilt is up.

iv. The tilt shall be relative to the home position of the camera gimbal.

v. The tilt may be provided as part of the camera orientation.

c. **Camera Roll**

i. The rotation angle about the X-axis, where the X-axis is positive "forward," or "into" the image produced by the camera

ii. The camera roll shall be in units of arc degrees.

iii. The camera roll shall be a real value, where positive camera roll is roll right and negative camera roll is roll left.

iv. The camera roll shall be relative to the home position of the camera gimbal.

v. The camera roll may be provided as part of the camera orientation.

3. ***Rotation***

a. **Rotation Type**

i. The type of camera gimbal rotation.

ii. The rotation type shall be a string chosen from one of the following values:

1. PAN
    a. Pan shall be rotation about the Z-axis, where left is positive and right is negative.
2. TILT
    a. Tilt shall be rotation about the Y-axis, where down is positive and up is negative.
3. CAMERA_ROLL
    a. Camera roll shall be rotation about the X-axis, where roll right is positive and roll left is negative.

iii. For all types of rotation, positive rotation shall be anti-clockwise looking toward the origin from the positive axis side.

iv. The rotation type shall be provided as part of the rotation.

b. **Direction**

i. The direction of the rotation.

ii. The direction shall be a string chosen from one of the following values:

1. POSITIVE
2. NEGATIVE

iii. The direction shall be provided as part of the rotation.

c. **Rate**

i. The rate of rotation.

ii. A detailed description of a "Rotation Rate" is provided in the "VISA Complex Types" section.

iii. The rate may be provided as part of the rotation.

1. Not providing a rate shall result in the camera gimbal slewing at a camera-specified "default" rate for the specified type of rotation.

d. **Distance**

i. The distance, relative to the current position, to slew the camera.

ii. The distance shall be in units of arc degrees.

iii. The distance shall be a real value greater than zero.

iv. If time is not provided, distance may be provided.

v. If time is provided, distance shall not be provided.

e. **Time**

i. The amount of time to slew the camera.
ii. The time shall be in units of microseconds.
iii. The time shall be an integer value greater than zero.
iv. If distance is not provided, time may be provided.
v. If distance is provided, time shall not be provided.

4. ***Rotation Rate***

a. **Rate Type**

i. The type of rate.
ii. The rate type shall be shall be a string chosen from one of the following values:
1. ARC_DEGREES_PER_SECOND
2. PERCENTAGE_OF_MAX
iii. The rate type shall be provided as part of the rotation rate.

b. **Rate**

i. The rate of rotation of the specified rotation type.
ii. If the rate type is ARC_DEGREES_PER_SECOND:
1. The rate shall be in units of arc degrees per second.
2. The rate shall be a real value greater than zero.
iii. If the rate type is PERCENTAGE_OF_MAX:
1. The rate shall be a percentage of max rotation of the camera gimbal.
2. The rate shall be a real value greater than zero and less than or equal to 100.
iv. The rate shall be provided as part of the rotation rate.

5. ***Home Position***

a. **Absolute Pan**

i. The absolute position of the camera gimbal's pan rotation.
ii. The absolute pan shall be in units of arc degrees.
iii. The absolute pan shall be a real value.
iv. The absolute pan may be provided as part of the home position.

b. **Absolute Tilt**

i. The absolute position of the camera gimbal's tilt rotation.
ii. The absolute tilt shall be in units of arc degrees.
iii. The absolute tilt shall be a real value.
iv. The absolute tilt may be provided as part of the home position.

c. **Absolute Camera Roll**

i. The absolute position of the camera gimbal's roll rotation.
ii. The absolute camera roll shall be in units of arc degrees.
iii. The absolute camera roll shall be a real value.
iv. The absolute camera roll may be provided as part of the home position.

6. ***Color Table***

a. **ID**

i. The unique identifier of the color table.
ii. The bounds of the ID are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The ID shall be unique within the scope of all other color tables available on the camera.

> ***NOTE:*** *VICTORY recommends the the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness.*

iv. The ID shall be provided as part of the color table.

b. **Color Tuples**

i. The list of color tuples that comprise the color table.
ii. A detailed description of a "Color Tuple" is provided in the "VISA Complex Types" section.
iii. At least one color tuple shall be provided as part of the color table.

7. ***Color Tuple***

a. **R**

i. The red color value of the color tuple.
ii. The red color value shall be an integer greater than or equal to zero and less than or equal to 255.
iii. The red color value shall be provided as part of the color tuple.

b. **G**

i. The green color value of the color tuple.
ii. The green color value shall be an integer greater than or equal to zero and less than or equal to 255.
iii. The green color value shall be provided as part of the color tuple.

c. **B**

i. The blue color value of the color tuple.
ii. The blue color value shall be an integer greater than or equal to zero and less than or equal to 255.
iii. The blue color value shall be provided as part of the color tuple.

19. PLATFORM SYSTEMS INTERFACES

Most vehicle platforms will host a variety of electronic systems that pertain to status, control, and/or diagnostics of the vehicle unit and the resources that it utilizes. While these systems are outside of the purview of the VICTORY architecture, VICTORY does provide standard interfaces for communicating between the C4ISR/EW components on the VDB and these systems. The following sections will identify some of these platform systems and identify the interfaces that they will share as part of the VICTORY architecture.

19.1 Power Distribution System Interfaces

One of the most common platform systems on current force platforms involves power distribution to the various systems the platform hosts. These Power Distribution Systems will enable the monitoring and control of power usage by any number of different electronic systems, including C4ISR/EW systems, protection systems, and wireless communications systems. The VICTORY architecture contains provisions for standard interfaces that the VDB nodes can utilize to communicate with the platform Power Management System. For example, a Warfighter Machine Interface (WMI) running on a VDB Common Display and Processing Unit may be used to communicate control commands and status messages with the Power Distribution System. These interfaces are identified in the sections that follow.

19.1.1 Power Distribution System Data Specifications

The Power Distribution System Data Interface provides information about the individual devices on the power distribution system, as well as power and battery information related to the overall platform. This information will be available to all VDB network nodes as required.

The Power Distribution System Data Interface may provide VDB-compliant messages of the following data types:

- Power System Device
- Power System Device Threshold
- Platform Battery
- Platform Power Generation
- Platform Power

19.1.1.1 Power System Device VICTORY Data Message <20069–20210409, Pro>

19.1.1.1.1 Power System Device Message Content

1. ***Timestamp***
    a. A timestamp corresponding to the time at which the power system device data was sampled from the power bus.
    b. The timestamp shall be in the format as described in the section titled "Timestamp Format for Application Layer Data."
    c. The timestamp shall be provided as part of the power system device data.
2. ***Sequence Number***
    a. A monotonically increasing number that indicates the sequence number of the power system device message sent from a particular Power Distribution System Data Interface.
    b. The sequence number shall increase by one (1) for each power system device message sent from the interface.
    c. When the sequence number reaches its maximum value, it shall roll over to its minimum value.
    d. The sequence number shall be provided as part of the power system device data.
3. ***Interface ID***
    a. An identification number, unique within the scope of the VDB, used to identify a particular Power Distribution System Data Interface.
    b. The interface ID shall be an integer value greater than or equal to zero.
    c. The interface ID shall be provided as part of the power system device data.
4. ***Device***
    a. Identifies a top-level power system device.

b. A top-level power system device shall fall into one of three categories:
    i. Power controller (e.g., power distribution unit).
    ii. Virtual device, used only for the categorization of load devices (e.g., communications, displays, etc.).
    iii. Device powered by system power.
c. If the top-level power system device is a power controller or a device powered by system power, the power system device message shall include all available data for the top-level power system device.
d. If the top-level power system device is a virtual device, the power system device message shall only include the unique identifier and the name of the top-level power system device.
    i. A virtual device is not an actual physical device in the system, so it does not contain power system device data.
e. A detailed description of the power system device data is provided in the section titled "Power System Device" under "Power Distribution System Interface Complex Types."
f. Device data shall be provided as part of the power system device data.

5. ***Load Device***
    a. If the top-level power system device is a power controller, the load device is the device being controlled by the top-level power system device.
        i. When the top-level power system device is a power controller, the power system device message shall contain one or more load devices.
    b. If the top-level power system device is a virtual device, the load device is the device that falls under the category of the top-level power system device.
        i. When the top-level power system device is a virtual device, the power system device message shall contain one or more load devices.
    c. If the top-level power system device is a device being powered by system power, the power system device message shall not contain any load devices.
        i. The top-level power system device is not associated with any other device on the power system, which is why it does not contain any load devices.
    d. The power system device message shall include all available data for each load device associated with the top-level power system device.
    e. A detailed description of the power system device data is provided in the section titled "Power System Device" under "Power Distribution System Interface Complex Types."
    f. Load device data may be provided as part of the power system device data.

19.1.1.1.2 Power System Device Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as a Power System Device VDM. Several Power System Device VDMs are provided below along with a brief description of each.

```
<vdm:messages xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <powerSystemDeviceMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:device>
            <vst:deviceId>pdu1</vst:deviceId>
            <vst:type>POWER_CONTROLLER</vst:type>
            <vst:name>SPDU20-0U</vst:name>
            <vst:current>75.0</vst:current>
            <vst:lowVoltage>52.8</vst:lowVoltage>
            <vst:temperature>42.6</vst:temperature>
            <vst:combatOverride>false</vst:combatOverride>
            <vst:loadDevices>
                <vst:loadDevice>
                    <vst:deviceId>sincgars1</vst:deviceId>
                    <vst:type>SYSTEM_POWERED_DEVICE</vst:type>
                    <vst:name>VRC_92F</vst:name>
```

```
                    <vst:current>38.5</vst:current>
                    <vst:lowVoltage>27.4</vst:lowVoltage>
                    <vst:temperature>35.8</vst:temperature>
                    <vst:tripStatus>NOT_TRIPPED</vst:tripStatus>
                    <vst:powerModeStatus>NORM</vst:powerModeStatus>
                    <vst:powerConnectorDisconnectInterlockStatus>
                        GOOD
                    </vst:powerConnectorDisconnectInterlockStatus>
                </vst:loadDevice>
                <vst:loadDevice>
                    <vst:deviceId>sincgars2</vst:deviceId>
                    <vst:type>SYSTEM_POWERED_DEVICE</vst:type>
                    <vst:name>VRC_92F</vst:name>
                    <vst:current>36.5</vst:current>
                    <vst:lowVoltage>25.4</vst:lowVoltage>
                    <vst:temperature>39.8</vst:temperature>
                    <vst:tripStatus>TRIPPED</vst:tripStatus>
                    <vst:powerModeStatus>NORM</vst:powerModeStatus>
                    <vst:powerConnectorDisconnectInterlockStatus>
                        GOOD
                    </vst:powerConnectorDisconnectInterlockStatus>
                </vst:loadDevice>
            </vst:loadDevices>
        </vst:device>
    </powerSystemDeviceMessage>
</vdm:messages>
```

Figure 25. Example Power System Device VDM with Two SINCGARS Radios

The example above shows the top-level power system device as being a power distribution unit that controls and sends power to two SINCGARS radios. The power distribution unit includes its unique identifier and its name, along with all other power system device data that is available from the power distribution unit. Each SINCGARS radio also includes its unique identifier and its name, along with all other power system device data that is available from the SINCGARS radio.

```
<vdm:messages xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <powerSystemDeviceMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:device>
            <vst:deviceId>comms</vst:deviceId>
                <vst:name>Communications Devices</vst:name>
                <vst:type>VIRTUAL_DEVICE</vst:type>
                <vst:loadDevices>
                    <vst:loadDevice>
                        <vst:deviceId>sincgars1</vst:deviceId>
                        <vst:type>SYSTEM_POWERED_DEVICE</vst:type>
                        <vst:name>VRC_92F</vst:name>
                        <vst:current>38.5</vst:current>
                        <vst:lowVoltage>27.4</vst:lowVoltage>
                        <vst:temperature>35.8</vst:temperature>
                        <vst:tripStatus>NOT_TRIPPED</vst:tripStatus>
                        <vst:combatOverride>false</vst:combatOverride>
                        <vst:powerModeStatus>NORM</vst:powerModeStatus>
                        <vst:powerConnectorDisconnectInterlockStatus>
```

```
                            GOOD
                        </vst:powerConnectorDisconnectInterlockStatus>
                    </vst:loadDevice>
                    <vst:loadDevice>
                        <vst:deviceId>sincgars2</vst:deviceId>
                        <vst:type>SYSTEM_POWERED_DEVICE</vst:type>
                        <vst:name>VRC_92F</vst:name>
                        <vst:current>36.5</vst:current>
                        <vst:lowVoltage>25.4</vst:lowVoltage>
                        <vst:temperature>39.8</vst:temperature>
                        <vst:tripStatus>TRIPPED</vst:tripStatus>
                        <vst:combatOverride>false</vst:combatOverride>
                        <vst:powerModeStatus>NORM</vst:powerModeStatus>
                        <vst:powerConnectorDisconnectInterlockStatus>
                            GOOD
                        </vst:powerConnectorDisconnectInterlockStatus>
                    </vst:loadDevice>
                </vst:loadDevices>
        </vst:device>
    </powerSystemDeviceMessage>
</vdm:messages>
```

Figure 26. Example Power System Device VDM

The example above shows the top-level power system device as being a virtual device that acts as a container for all communications devices on the power system. The top-level power system device includes its unique identifier and its name only. There is no other power system device data associated with a virtual device. Each SINCGARS radio also includes its unique identifier and its name, along with all other power system device data that is available from the SINCGARS radio. The location from which the SINCGARS radios are actually being controlled cannot be determined from this message. This message simply allows for the retrieval of power system device data based on a category of devices.

```
<vdm:messages xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <powerSystemDeviceMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:device>
            <vst:deviceId>sincgars1</vst:deviceId>
            <vst:type>SYSTEM_POWERED_DEVICE</vst:type>
            <vst:name>VRC_92F</vst:name>
            <vst:current>38.5</vst:current>
            <vst:lowVoltage>27.4</vst:lowVoltage>
            <vst:temperature>35.8</vst:temperature>
            <vst:tripStatus>NOT_TRIPPED</vst:tripStatus>
            <vst:combatOverride>false</vst:combatOverride>
            <vst:powerModeStatus>NORM</vst:powerModeStatus>
            <vst:powerConnectorDisconnectInterlockStatus>
                GOOD
            </vst:powerConnectorDisconnectInterlockStatus>
        </vst:device>
    </powerSystemDeviceMessage>
</vdm:messages>
```

Figure 27. Additional Example Power System Device VDM

The example above shows a case where the top-level power system device is connected directly to system power. These devices do not contain any load devices because they are not providing power to any devices and they are not a virtual device. The SINCGARS radio includes its unique identifier and its name, along with all other power system device data that is available from the SINCGARS radio.

19.1.1.2 Power System Device Threshold VICTORY Data Message <20070–20210409, Pro>

19.1.1.2.1 Power System Device Threshold Message Content

1. ***Timestamp***
    a. A timestamp corresponding to the time at which the power system device threshold data was sampled from the power bus.
    b. The timestamp shall be in the format as described in the section titled "Timestamp Format for Application Layer Data."
    c. The timestamp shall be provided as part of the power system device threshold data.
2. ***Sequence Number***
    a. A monotonically increasing number that indicates the sequence number of the power system device threshold message sent from a particular Power Distribution System Data Interface.
    b. The sequence number shall increase by one (1) for each power system device threshold message sent from the interface.
    c. When the sequence number reaches its maximum value, it shall roll over to its minimum value.
    d. The sequence number shall be provided as part of the power system device threshold data.
3. ***Interface ID***
    a. An identification number, unique within the scope of the VDB, used to identify a particular Power Distribution System Data Interface.
    b. The interface ID shall be an integer value greater than or equal to zero.
    c. The interface ID shall be provided as part of the power system device threshold data.

    > ***NOTE:*** *The devices and load devices provided in the power system device threshold message refer to the same devices and load devices inside the power system device message. The threshold data has been broken out into a separate message because threshold data is unlikely to change, so threshold data may not need to be continuously sent to the VDB.*

4. ***Device***
    a. Identifies a top-level power system device in the system.
    b. A top-level power system device shall fall under one of three categories:
        i. Power controller (e.g., power distribution unit).
        ii. Virtual device, used only for the categorization of load devices (e.g., communications, displays, etc.).
        iii. Device powered by system power.
            1. These devices are not associated with any other devices on the power system.
    c. If the top-level power system device is a power controller or a device powered by system power, the power system device threshold message shall include all available threshold data for the top-level power system device.
    d. If the top-level power system device is a virtual device, the power system device threshold message shall only include the unique identifier and the name of the top-level power system device.
        i. A virtual device is not an actual physical device in the system so it does not contain power system device threshold data.
    e. A detailed description of the power system device threshold data is provided in the section titled "Power System Device Threshold" under "Power Distribution System Interface Complex Types."
    f. Device data shall be provided as part of the power system device threshold data.
5. ***Load Device***
    a. If the top-level power system device is a power controller, the load device is the device being controlled by the top-level device.
        i. When the top-level power system device is a power controller, the power system device threshold message shall contain one or more load devices.
    b. If the top-level power system device is a virtual device, the load device is the device that falls under the category of the top-level power system device.
        i. When the top-level power system device is a virtual device, the power system device threshold message shall contain one or more load devices.

c. If the top-level power system device is a device being powered by system power, the power system device threshold message shall not contain any load devices.
    i. The top-level power system device is not associated with any other device on the power system, which is why it does not contain any load devices.
d. The power system device threshold message shall include all available threshold data for each load device associated with the top-level power system device.
e. A detailed description of the power system device threshold data is provided in the section titled "Power System Device" under "Power Distribution System Interface Complex Types."
f. Load device data may be provided as part of the power system device threshold data.

19.1.1.2.2 Power System Device Threshold Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as a Power System Device Threshold VDM. Several example Power System Device Threshold VDMs are provided below along with a brief description of each.

```
<vdm:messages xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <powerSystemDeviceThresholdMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:device>
            <vst:deviceId>pdu1</vst:deviceId>
            <vst:type>POWER_CONTROLLER</vst:type>
            <vst:name>SPDU20-0U</vst:name>
            <vst:lowVoltageThresholds>
                <vst:currentTripThreshold>50</vst:currentTripThreshold>
                <vst:currentOnlineThreshold>30</vst:currentOnlineThreshold>
                <vst:upperVoltageThreshold>32</vst:upperVoltageThreshold>
                <vst:lowerVoltageThreshold>12</vst:lowerVoltageThreshold>
            </vst:lowVoltageThresholds>
            <vst:temperatureThresholds>
                <vst:upperTemperatureThreshold>50</vst:upperTemperatureThreshold>
                <vst:lowerTemperatureThreshold>-40</vst:lowerTemperatureThreshold>
            </vst:temperatureThresholds>
            <vst:loadDevices>
                <vst:loadDevice>
                    <vst:deviceId>sincgars1</vst:deviceId>
                    <vst:type>SYSTEM_POWERED_DEVICE</vst:type>
                    <vst:name>VRC_92F</vst:name>
                    <vst:lowVoltageThresholds>
                        <vst:currentTripThreshold>50</vst:currentTripThreshold>
                        <vst:currentOnlineThreshold>30</vst:currentOnlineThreshold>
                        <vst:upperVoltageThreshold>32</vst:upperVoltageThreshold>
                        <vst:lowerVoltageThreshold>12</vst:lowerVoltageThreshold>
                    </vst:lowVoltageThresholds>
                    <vst:temperatureThresholds>
                        <vst:upperTemperatureThreshold>
                            50
                        </vst:upperTemperatureThreshold>
                        <vst:lowerTemperatureThreshold>
                            -40
                        </vst:lowerTemperatureThreshold>
                    </vst:temperatureThresholds>
                </vst:loadDevice>
                <vst:loadDevice>
                    <vst:deviceId>sincgars2</vst:deviceId>
```

```
                    <vst:type>SYSTEM_POWERED_DEVICE</vst:type>
                    <vst:name>VRC_92F</vst:name>
                    <vst:lowVoltageThresholds>
                        <vst:currentTripThreshold>50</vst:currentTripThreshold>
                        <vst:currentOnlineThreshold>30</vst:currentOnlineThreshold>
                        <vst:upperVoltageThreshold>32</vst:upperVoltageThreshold>
                        <vst:lowerVoltageThreshold>12</vst:lowerVoltageThreshold>
                    </vst:lowVoltageThresholds>
                    <vst:temperatureThresholds>
                        <vst:upperTemperatureThreshold>
                            50
                        </vst:upperTemperatureThreshold>
                        <vst:lowerTemperatureThreshold>
                            -40
                        </vst:lowerTemperatureThreshold>
                    </vst:temperatureThresholds>
                </vst:loadDevice>
            </vst:loadDevices>
        </vst:device>
    </powerSystemDeviceThresholdMessage>
</vdm:messages>
```

Figure 28. Example Power System Device Threshold VDM with Two SINCGARS Radios

The example above shows the top-level device as being a power distribution unit that controls, and sends power to, two SINCGARS radios. The power distribution unit includes its unique identifier and its name, along with all other power system device threshold data that is available from the power distribution unit. Each SINCGARS radio also includes its unique identifier and its name, along with all other power system device threshold data that is available from the SINCGARS radio.

```
<vdm:messages xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <powerSystemDeviceThresholdMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:device>
            <vst:deviceId>comms</vst:deviceId>
            <vst:type>VIRTUAL_DEVICE</vst:type>
            <vst:name>Communications Devices</vst:name>
            <vst:loadDevices>
                <vst:loadDevice>
                    <vst:deviceId>sincgars1</vst:deviceId>
                    <vst:type>SYSTEM_POWERED_DEVICE</vst:type>
                    <vst:name>VRC_92F</vst:name>
                    <vst:lowVoltageThresholds>
                        <vst:currentTripThreshold>
                            50
                        </vst:currentTripThreshold>
                        <vst:currentOnlineThreshold>
                            30
                        </vst:currentOnlineThreshold>
                        <vst:upperVoltageThreshold>
                            32
                        </vst:upperVoltageThreshold>
                        <vst:lowerVoltageThreshold>
                            12
```

```
                        </vst:lowerVoltageThreshold>
                    </vst:lowVoltageThresholds>
                    <vst:temperatureThresholds>
                        <vst:upperTemperatureThreshold>
                            50
                        </vst:upperTemperatureThreshold>
                        <vst:lowerTemperatureThreshold>
                            -40
                        </vst:lowerTemperatureThreshold>
                    </vst:temperatureThresholds>
                </vst:loadDevice>
                <vst:loadDevice>
                    <vst:deviceId>sincgars2</vst:deviceId>
                    <vst:type>SYSTEM_POWERED_DEVICE</vst:type>
                    <vst:name>VRC_92F</vst:name>
                    <vst:lowVoltageThresholds>
                        <vst:currentTripThreshold>50</vst:currentTripThreshold>
                        <vst:currentOnlineThreshold>30</vst:currentOnlineThreshold>
                        <vst:upperVoltageThreshold>32</vst:upperVoltageThreshold>
                        <vst:lowerVoltageThreshold>12</vst:lowerVoltageThreshold>
                    </vst:lowVoltageThresholds>
                    <vst:temperatureThresholds>
                        <vst:upperTemperatureThreshold>
                            50
                        </vst:upperTemperatureThreshold>
                        <vst:lowerTemperatureThreshold>
                            -40
                        </vst:lowerTemperatureThreshold>
                        </vst:temperatureThresholds>
                </vst:loadDevice>
            </vst:loadDevices>
        </vst:device>
    </powerSystemDeviceThresholdMessage>
</vdm:messages>
```

Figure 29. Example Power System Device Threshold VDM

The example above shows the top-level device as being a virtual device that acts as a container for all communications devices on the system. The top-level device includes its unique identifier and its name only. There is no other power system device threshold data associated with a virtual device. Each SINCGARS radio also includes its unique identifier and its name, along with all other power system device threshold data that is available from the SINCGARS radio. The location from which the SINCGARS radios are actually being controlled cannot be determined from this message. This message simply allows for the retrieval of power system device threshold data based on a category of devices.

```
<vdm:messages xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <powerSystemDeviceThresholdMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:device>
            <vst:deviceId>sincgars1</vst:deviceId>
            <vst:type>SYSTEM_POWERED_DEVICE</vst:type>
            <vst:name>VRC_92F</vst:name>
            <vst:lowVoltageThresholds>
                <vst:currentTripThreshold>50</vst:currentTripThreshold>
```

```
                <vst:currentOnlineThreshold>30</vst:currentOnlineThreshold>
                <vst:upperVoltageThreshold>32</vst:upperVoltageThreshold>
                <vst:lowerVoltageThreshold>12</vst:lowerVoltageThreshold>
            </vst:lowVoltageThresholds>
            <vst:temperatureThresholds>
                <vst:upperTemperatureThreshold>50</vst:upperTemperatureThreshold>
                <vst:lowerTemperatureThreshold>-40</vst:lowerTemperatureThreshold>
            </vst:temperatureThresholds>
        </vst:device>
    </powerSystemDeviceThresholdMessage>
</vdm:messages>
```

Figure 30. Additional Example Power System Device Threshold VDM

The example above shows a case where the top-level device is connected directly to system power. These devices do not contain any load devices because they are not providing power to any devices. The SINCGARS radio provides its unique identifier and its name, along with all other power system device threshold data that is available from the SINCGARS radio.

19.1.1.3 Platform Battery VICTORY Data Message <20071–20210409, Pro>

19.1.1.3.1 Platform Battery Message Content

1. ***Timestamp***
    a. A timestamp corresponding to the time at which the platform battery data was sampled from the power bus.
    b. The timestamp shall be in the format as described in the section titled "Timestamp Format for Application Layer Data."
    c. The timestamp shall be provided as part of the platform battery message.
2. ***Sequence Number***
    a. A monotonically increasing number that indicates the sequence number of the platform battery message sent from a particular Power Distribution System Data Interface.
    b. The sequence number shall increase by one (1) for each platform battery message sent from the interface.
    c. When the sequence number reaches its maximum value, it shall roll over to its minimum value.
    d. The sequence number shall be provided as part of the platform battery message.
3. ***Interface ID***
    a. An identification number, unique within the scope of the VDB, used to identify a particular Power Distribution System Data Interface.
    b. The interface ID shall be an integer value greater than or equal to zero.
    c. The interface ID shall be provided as part of the platform battery message.
4. ***Battery***
    a. Identifies a battery on the platform.
    b. A detailed description of the platform battery data is provided in the section titled "Platform Battery" under "Power Distribution System Interface Complex Types."
    c. Battery data shall be provided as part of the platform battery message.

19.1.1.3.2 Platform Battery Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as a Platform Battery VDM. An example Platform Battery VDM is provided below.

```
<vdm:messages xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <platformBatteryMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <battery>
            <vst:batteryId>battery1</vst:batteryId>
            <vst:name>Armasafe Plus 12 FV 120</vst:name>
            <vst:stateOfCharge>92</vst:stateOfCharge>
            <vst:stateOfHealth>98</vst:stateOfHealth>
            <vst:current>-47</vst:current>
            <vst:lowVoltage>24.75</vst:lowVoltage>
            <vst:temperature>45</vst:temperature>
            <vst:chargeEnabled>true</vst:chargeEnabled>
            <vst:interconnected>true</vst:interconnected>
            <vst:chargeCurrent>12.2</vst:chargeCurrent>
            <vst:chargeVoltage>27.9</vst:chargeVoltage>
        </battery>
    </platformBatteryMessage>
</vdm:messages>
```

Figure 31. Example Platform Battery VDM

19.1.1.4 Platform Power Generation VICTORY Data Message <20072–20210409, Pro>

Platform power generation refers to the devices that provide power to the other devices and batteries on the power system.

19.1.1.4.1 Platform Power Generation Message Content

1. ***Timestamp***
    a. A timestamp corresponding to the time at which the platform power generation data was sampled from the power bus.
    b. The timestamp shall be in the format as described in the section titled "Timestamp Format for Application Layer Data."
    c. The timestamp shall be provided as part of the platform power generation message.
2. ***Sequence Number***
    a. A monotonically increasing number that indicates the sequence number of the platform power generation message sent from a particular Power Distribution System Data Interface.
    b. The sequence number shall increase by one (1) for each platform power generation message sent from the interface.
    c. When the sequence number reaches its maximum value, it shall roll over to its minimum value.
    d. The sequence number shall be provided as part of the platform power generation message.
3. ***Interface ID***
    a. An identification number, unique within the scope of the VDB, used to identify a particular Power Distribution System Data Interface.
    b. The interface ID shall be an integer value greater than or equal to zero.
    c. The interface ID shall be provided as part of the platform power generation message.
4. ***Generator***
    a. Identifies the generator on the platform.
    b. A platform power generation message shall contain a single generator.
    c. A detailed description of the generator data is provided in the section titled "Generator" under "Power Distribution System Interface Complex Types."
    d. Generator data may be provided as part of the platform power generation message.
5. ***Auxiliary Power Unit***
    a. Identifies the auxiliary power unit (APU) on the platform.
    b. A detailed description of the auxiliary power unit data is provided in the section titled "Auxiliary Power Unit" under "Power Distribution System Interface Complex Types."
    c. Auxiliary power unit data may be provided as part of the platform power generation message.

19.1.1.4.2 Platform Power Generation Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as a Platform Power Generation VDM. An example Platform Power Generation VDM is provided below.

```
<vdm:messages xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <platformPowerGenerationMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:generator>
            <vst:generatorId>generator1</vst:generatorId>
            <vst:name>HMMWV Generator</vst:name>
            <vst:voltage>24.7</vst:voltage>
            <vst:temperature>32</vst:temperature>
            <vst:shaftSpeed>2000</vst:shaftSpeed>
            <vst:utilization>43</vst:utilization>
            <vst:faults>
                <vst:fault>
                    <vst:faultCode>A1</vst:faultCode>
                    <vst:faultStatus>GOOD</vst:faultStatus>
                </vst:fault>
                <vst:fault>
                    <vst:faultCode>B1</vst:faultCode>
                    <vst:faultStatus>FAULTED</vst:faultStatus>
                </vst:fault>
            </vst:faults>
            <vst:outputEnabled>false</vst:outputEnabled>
        </vst:generator>
        <vst:auxiliaryPowerUnit>
            <vst:auxiliaryPowerUnitId>apu1</vst:auxiliaryPowerUnitId>
            <vst:name>HMMWV APU</vst:name>
            <vst:power>18.4</vst:power>
            <vst:voltage>24.7</vst:voltage>
            <vst:temperature>32</vst:temperature>
            <vst:faults>
                <vst:fault>
                <vst:faultCode>C1</vst:faultCode>
                <vst:faultStatus>GOOD</vst:faultStatus>
                </vst:fault>
            </vst:faults>
            <vst:outputEnabled>false</vst:outputEnabled>
        </vst:auxiliaryPowerUnit>
    </platformPowerGenerationMessage>
</vdm:messages>
```

Figure 32. Example Platform Power Generation VDM

19.1.1.5 Platform Power VICTORY Data Message <20073–20131031, Pro>

Platform Power refers to the overall power of the vehicle (platform). The overall power is comprised of all the devices on the platform power system.

19.1.1.5.1 Platform Power Message Content

1. ***Timestamp***
    a. A timestamp corresponding to the time at which the platform power data was sampled from the power management system bus.
    b. The timestamp shall be in the format as described in the section titled "Timestamp Format for Application Layer Data."
    c. The timestamp shall be provided as part of the platform power message.
2. ***Sequence Number***
    a. A monotonically increasing number that indicates the sequence number of the platform power message sent from a particular Power Distribution System Data Interface.
    b. The sequence number shall increase by one (1) for each platform power message sent from the interface.
    c. When the sequence number reaches its maximum value, it shall roll over to its minimum value.
    d. The sequence number shall be provided as part of the platform power message.
3. ***Interface ID***
    a. An identification number, unique within the scope of the VDB, used to identify a particular Power Distribution System Data Interface.
    b. The interface ID shall be an integer value greater than or equal to zero.
    c. The interface ID shall be provided as part of the platform power message.
4. ***Platform Power***
    a. Identifies the power of the platform.
    b. A detailed description of the platform power data is provided in the section titled "Platform Power" under "Power Distribution System Interface Complex Types."
    c. Platform power data shall be provided as part of the platform power message.

19.1.1.5.2 Platform Power Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as a Platform Power VDM. An example Platform Power VDM is provided below.

```
<vdm:messages xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <platformPowerMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:instantaneousPowerConsumed>1350</vst:instantaneousPowerConsumed>
        <vst:instantaneousPowerGenerated>5400</vst:instantaneousPowerGenerated>
        <vst:systemVoltage>27.4</vst:systemVoltage>
        <vst:totalInstantaneousEnginePower>240</vst:totalInstantaneousEnginePower>
        <vst:availableEnginePower>50</vst:availableEnginePower>
    </platformPowerMessage>
</vdm:messages>
```

Figure 33. Example Platform Power VDM

19.1.2 Power Distribution System Data Management Parameters <16004–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the power distribution system data management parameters. A textual description of the power distribution system data management parameters is shown below.

NOTE: *The data of interest and data enabled flags control the flow of data on the VICTORY platform. When a type of power data has a data of interest flag set to False, the data is not monitored from the power bus. As a result, the data cannot be sent to the VDB from the Power Distribution System Data Interface (regardless of the data enabled flag), nor can the data be retrieved from the Power Distribution System Management Interface. For all intents and purposes, data with a data of interest flag set to False is considered by VICTORY to not be available on the power bus. When a type of power data has a data of interest flag set to True, the data is monitored from the power bus. The data enabled flag can then be used for controlling whether the data is encapsulated into a VDM and sent to the VDB from the Power Distribution System Data Interface. Regardless of the data enabled flag, the most recent data received from the power bus may be retrieved from the Power Distribution System Management Interface if the data of interest flag is set to True.*

1. ***Power System Device IDs***
    a. Identifies the top-level power system devices managed by this interface.
    b. Top-level power system devices include virtual devices, devices that provide power to load devices, or devices that receive their power from system power and do not supply power to any other devices.
    c. The power system device ID for each device managed by the interface shall be returned. A detailed description of the power system device ID is provided in the section titled "Power System Device" under "Power Distribution System Interface Complex Types."
    d. Operations include Get().
2. ***Platform Battery IDs***
    a. Identifies the batteries on the platform that are managed by this interface.
    b. The platform battery ID for each battery managed by the interface shall be returned. A detailed description of the platform battery ID is provided in the section titled "Platform Battery" under "Power Distribution System Interface Complex Types."
    c. Operations include Get().
3. ***Power System Device Data***
    a. Identifies the most recent power system device data received from the power bus for both the top-level power system device and the load devices associated with the top-level power system device.
    b. The unique name of a top-level power system device managed by this interface shall be provided in the request.
    c. The following errors are defined for the power system device data parameter:
        i. Retrieving power system device data with a device identifier not associated with a power system device shall result in a GENERIC_FAULT error.
        ii. Retrieving power system device data for a device that is not available from the power bus or is not being monitored (data of interest flag set to False) shall result in a NOT_SUPPORTED error.
        iii. Retrieving power system device data with a device identifier outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    d. A detailed description of the power system device data is provided in the section titled "Power System Device" under "Power Distribution System Interface Complex Types."
    e. Operations include Get().
4. ***Power System Device Threshold Data***
    a. Identifies the most recent power system device threshold data received from the power bus for both the top-level power system device and the load devices associated with the top-level power system device.
    b. The unique name of a top-level power system device managed by this interface shall be provided in the request.
    c. The following errors are defined for the power system device threshold data parameter:
        i. Retrieving power system device threshold data with a device identifier not associated with a power system device shall result in a GENERIC_FAULT error.
        ii. Retrieving power system device threshold data for a device that is not available from the power bus or is not being monitored (data of interest flag set to False) shall result in a NOT_SUPPORTED error.
        iii. Retrieving power system device threshold data with a device identifier outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    d. A detailed description of the power system device threshold data is provided in the section titled "Power System Device Threshold" under "Power Distribution System Interface Complex Types."
    e. Operations include Get().
5. ***Platform Battery Data***
    a. Identifies the most recent battery data received from the power bus for a battery on the platform.
    b. The unique name of a battery on the platform shall be provided in the request.
    c. The following errors are defined for the platform battery data parameter:
        i. Retrieving platform battery data with a device identifier not associated with a battery on the platform shall result in a GENERIC_FAULT error.

ii. Retrieving platform battery data for a battery that is not available from the power bus or is not being monitored (data of interest flag set to False) shall result in a NOT_SUPPORTED error.
iii. Retrieving platform battery data with a battery identifier outside the range defined by this specification document may result in an OUT_OF_RANGE error.

d. A detailed description of the platform battery data is provided in the section titled "Platform Battery" under "Power Distribution System Interface Complex Types."
e. Operations include Get().

6. ***Platform Power Generation Data***
   a. Identifies the most recent platform power generation data received from the power bus.
   b. The following errors are defined for the platform power generation data parameter:
      i. Retrieving platform power generation data that is not available from the power bus or is not being monitored (data of interest flag set to False) shall result in a NOT_SUPPORTED error.
   c. A detailed description of the platform power generation data is provided in the sections titled "Generator" and "Auxiliary Power Unit" under "Power Distribution System Interface Complex Types."
   d. Operations include Get().
7. ***Platform Power Data***
   a. Identifies the most recent platform power data retrieved from the power bus.
   b. The following errors are defined for the platform power data parameter:
      i. Retrieving platform power data that is not available from the power bus or is not being monitored (data of interest flag set to False) shall result in a NOT_SUPPORTED error.
   c. A detailed description of the platform power data is provided in the section titled "Platform Power" under "Power Distribution System Interface Complex Types."
   d. Operations include Get().
8. ***Data of Interest***
   a. Identifies the power distribution system data being monitored from the power bus.
   b. The following types of power distribution system data contain a single data of interest flag:
      i. PLATFORM_POWER_GENERATION
      ii. PLATFORM_POWER
   c. The following types of power distribution system data contain a data of interest flag for each top-level power system device being managed by this interface:
      i. POWER_SYSTEM_DEVICE
      ii. POWER_SYSTEM_DEVICE_THRESHOLD
   d. The following types of power distribution system data contain a data of interest flag for each battery being managed by this interface:
      i. PLATFORM_BATTERY
   e. The type of power distribution system data shall be a string chosen from one of the following values:
      i. POWER_SYSTEM_DEVICE
      ii. POWER_SYSTEM_DEVICE_THRESHOLD
      iii. PLATFORM_BATTERY
      iv. PLATFORM_POWER_GENERATION
      v. PLATFORM_POWER
   f. The unique name of a top-level power system device or a battery shall be specified for the following types of power distribution system data:
      i. POWER_SYSTEM_DEVICE
      ii. POWER_SYSTEM_DEVICE_THRESHOLD
      iii. PLATFORM_BATTERY
   g. The bounds of the unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
   h. Each type of power distribution system data shall contain a data of interest flag.
      i. A Boolean value of True means the data is monitored and a value of False means it is not monitored.
      ii. The data of interest flag for each type of power distribution system data shall be controlled independently.
   i. The following errors are defined for the data of interest parameter:
      i. Retrieving or setting the data of interest flag for power distribution system data that is not available on the power bus shall result in a NOT_SUPPORTED error.
      ii. Retrieving or setting the data of interest flag with a type of power distribution system data not inside the list of known types of power distribution system data may result in an OUT_OF_RANGE error.
      iii. Providing a unique name when retrieving or setting the data of interest flag shall result in a GENERIC_FAULT error for the following types of power distribution system data:

1. PLATFORM_POWER_GENERATION
2. PLATFORM_POWER

iv. Not providing a unique name when retrieving or setting the data of interest flag shall result in a GENERIC_FAULT error for the following types of power distribution system data:
1. POWER_SYSTEM_DEVICE
2. POWER_SYSTEM_DEVICE_THRESHOLD
3. PLATFORM_BATTERY

v. Retrieving or setting the data of interest flag with an incorrect unique name shall result in a GENERIC_FAULT error for the following types of power distribution system data:
1. POWER_SYSTEM_DEVICE
2. POWER_SYSTEM_DEVICE_THRESHOLD
3. PLATFORM_BATTERY

vi. Retrieving or setting the data of interest flag with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.

j. Operations include Get() and Set().

9. ***Data Enabled***

a. Identifies the power distribution system data sending VDMs on the VDB.

b. The following types of power distribution system data contain a single data enabled flag:
i. PLATFORM_POWER_GENERATION
ii. PLATFORM_POWER

c. The following types of power distribution system data contain a data enabled flag for each top-level power system device being managed by this interface:
i. POWER_SYSTEM_DEVICE
ii. POWER_SYSTEM_DEVICE_THRESHOLD

d. The following types of power distribution system data contain a data enabled flag for each battery being managed by this interface:
i. PLATFORM_BATTERY

e. The type of power distribution system data shall be a string chosen from one of the following values:
i. POWER_SYSTEM_DEVICE
ii. POWER_SYSTEM_DEVICE_THRESHOLD
iii. PLATFORM_BATTERY
iv. PLATFORM_POWER_GENERATION
v. PLATFORM_POWER

f. The unique name of a top-level power system device or a battery shall be specified for the following types of power distribution system data:
i. POWER_SYSTEM_DEVICE
ii. POWER_SYSTEM_DEVICE_THRESHOLD
iii. PLATFORM_BATTERY

g. The bounds of the unique name are described in the section titled "String Bounds" under "VICTORY Data Types."

h. Each type of power distribution system data shall contain a data enabled flag.
i. A Boolean value of True means "enabled" (sending) and a value of False means "disabled" (not sending).
ii. The data enabled flag for each type of power distribution system data shall be controlled independently.

i. The following errors are defined for the data enabled parameter:
i. Retrieving the data enabled flag for power distribution system data that is not available on the power bus or is not being monitored from the power bus (data of interest flag set to False) shall result in a NOT_SUPPORTED error.
ii. Retrieving or setting the data enabled flag with a type of power distribution system data not inside the list of known types of power distribution system data may result in an OUT_OF_RANGE error.
iii. Providing a unique name when retrieving or setting the data enabled flag shall result in a GENERIC_FAULT error for the following types of power distribution system data:
1. PLATFORM_POWER_GENERATION
2. PLATFORM_POWER

iv. Not providing a unique name when retrieving or setting the data enabled flag shall result in a GENERIC_FAULT error for the following types of power distribution system data:
1. POWER_SYSTEM_DEVICE
2. POWER_SYSTEM_DEVICE_THRESHOLD

3. PLATFORM_BATTERY

v. Retrieving or setting the data enabled flag with an incorrect unique name shall result in a GENERIC_FAULT error for the following types of power distribution system data:
   1. POWER_SYSTEM_DEVICE
   2. POWER_SYSTEM_DEVICE_THRESHOLD
   3. PLATFORM_BATTERY

vi. Retrieving or setting the data enabled flag with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.

j. Operations include Get() and Set().

10. ***Update Period***

a. Identifies the time between VDM updates for the specified type of power distribution system data.

b. The following types of power distribution system data contain a single update period:
   i. PLATFORM_POWER_GENERATION
   ii. PLATFORM_POWER

c. The following types of power distribution system data contain an update period for each top-level power system device being managed by this interface:
   i. POWER_SYSTEM_DEVICE
   ii. POWER_SYSTEM_DEVICE_THRESHOLD

d. The following types of power distribution system data contain an update period for each battery being managed by this interface:
   i. PLATFORM_BATTERY

e. The type of power distribution system data shall be a string chosen from one of the following values:
   i. POWER_SYSTEM_DEVICE
   ii. POWER_SYSTEM_DEVICE_THRESHOLD
   iii. PLATFORM_BATTERY
   iv. PLATFORM_POWER_GENERATION
   v. PLATFORM_POWER

f. The unique name of a top-level power system device or a battery shall be specified for the following types of power distribution system data:
   i. POWER_SYSTEM_DEVICE
   ii. POWER_SYSTEM_DEVICE_THRESHOLD
   iii. PLATFORM_BATTERY

g. The bounds of the unique name are described in the section titled "String Bounds" under "VICTORY Data Types."

h. Each type of power distribution system data shall contain an update period.
   i. The update period shall be in units of milliseconds.
   ii. The update period shall be a real value greater than zero.
   iii. The update period for each type of power distribution system data shall be controlled independently.

i. The following errors are defined for the update period parameter:
   i. Retrieving or setting the update period for power distribution system data that is not available on the power bus shall result in a NOT_SUPPORTED error.
   ii. Retrieving or setting the update period with a type of power distribution system data not inside the list of known types of power distribution system data may result in an OUT_OF_RANGE error.
   iii. Providing a unique name when retrieving or setting the update period shall result in a GENERIC_FAULT error for the following types of power distribution system data:
      1. PLATFORM_POWER_GENERATION
      2. PLATFORM_POWER
   iv. Not providing a unique name when retrieving or setting the update period shall result in a GENERIC_FAULT error for the following types of power distribution system data:
      1. POWER_SYSTEM_DEVICE
      2. POWER_SYSTEM_DEVICE_THRESHOLD
      3. PLATFORM_BATTERY
   v. Retrieving or setting the update period with an incorrect unique name shall result in a GENERIC_FAULT error for the following types of power distribution system data:
      1. POWER_SYSTEM_DEVICE
      2. POWER_SYSTEM_DEVICE_THRESHOLD
      3. PLATFORM_BATTERY
   vi. Setting the update period to a value outside the bounds defined by this specification document may result in an OUT_OF_RANGE error.

vii. Setting the update period to a value less than the minimum update period shall result in a NEAREST_VALUE error.
1. The update period shall be changed to the minimum update period and shall be sent back in the response.
viii. Retrieving or setting the update period with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.

j. Though a very small update period (e.g., an update period of five) is within the range defined by this specification document (greater than zero), the actual update period may be higher depending on the performance of the system. If an update period is set to a lower rate than what is achievable by the system, the system shall send out VDMs as fast as possible.
k. Operations include Get() and Set().

11. ***Minimum Update Period***
a. The maximum frequency at which updates can be sent from the power bus.
b. The following types of power distribution system data contain a single minimum update period:
i. PLATFORM_POWER_GENERATION
ii. PLATFORM_POWER
c. The following types of power distribution system data contain a minimum update period for each top-level power system device being managed by this interface:
i. POWER_SYSTEM_DEVICE
ii. POWER_SYSTEM_DEVICE_THRESHOLD
d. The following types of power distribution system data contain a minimum update period for each battery being managed by this interface:
i. PLATFORM_BATTERY
e. The type of power distribution system data shall be a string chosen from one of the following values:
i. POWER_SYSTEM_DEVICE
ii. POWER_SYSTEM_DEVICE_THRESHOLD
iii. PLATFORM_BATTERY
iv. PLATFORM_POWER_GENERATION
v. PLATFORM_POWER
f. The unique name of a top-level power system device or a battery shall be specified for the following types of power distribution system data:
i. POWER_SYSTEM_DEVICE
ii. POWER_SYSTEM_DEVICE_THRESHOLD
iii. PLATFORM_BATTERY
g. The bounds of the unique name are described in the section titled "String Bounds" under "VICTORY Data Types."
h. Each type of power distribution system data shall contain a minimum update period.
i. The minimum update period shall be in units of milliseconds.
ii. The minimum update period shall be a real value greater than zero.
i. The following errors are defined for the minimum update period parameter:
i. Retrieving the minimum update period for power distribution system data that is not available on the power bus shall result in a NOT_SUPPORTED error.
ii. Retrieving the minimum update period with a type of power distribution system data not inside the list of known types of power distribution system data may result in an OUT_OF_RANGE error.
iii. Providing a unique name when retrieving the minimum update period shall result in a GENERIC_FAULT error for the following types of power distribution system data:
1. PLATFORM_POWER_GENERATION
2. PLATFORM_POWER
iv. Not providing a unique name when retrieving the minimum update period shall result in a GENERIC_FAULT error for the following types of power distribution system data:
1. POWER_SYSTEM_DEVICE
2. POWER_SYSTEM_DEVICE_THRESHOLD
3. PLATFORM_BATTERY
v. Retrieving the minimum update period with an incorrect unique name shall result in a GENERIC_FAULT error for the following types of power distribution system data:
1. POWER_SYSTEM_DEVICE
2. POWER_SYSTEM_DEVICE_THRESHOLD
3. PLATFORM_BATTERY

vi. Retrieving the minimum update period with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.

j. Operations include Get().

12. ***Data Transport Configuration***

a. The transport configurations used for sending out power distribution system VDMs on the VDB.

b. The following types of power distribution system data contain a single set of data transport configurations:
   i. PLATFORM_POWER_GENERATION
   ii. PLATFORM_POWER

c. The following types of power distribution system data contain a set of data transport configurations for each top-level power system device being managed by this interface:
   i. POWER_SYSTEM_DEVICE
   ii. POWER_SYSTEM_DEVICE_THRESHOLD

d. The following types of power distribution system data contain a set of data transport configurations for each battery being managed by this interface:
   i. PLATFORM_BATTERY

e. The type of power distribution system data shall be a string chosen from one of the following values:
   i. POWER_SYSTEM_DEVICE
   ii. POWER_SYSTEM_DEVICE_THRESHOLD
   iii. PLATFORM_BATTERY
   iv. PLATFORM_POWER_GENERATION
   v. PLATFORM_POWER

f. The unique name of a top-level power system device or a battery shall be specified for the following types of power distribution system data:
   i. POWER_SYSTEM_DEVICE
   ii. POWER_SYSTEM_DEVICE_THRESHOLD
   iii. PLATFORM_BATTERY

g. The bounds of the unique name are described in the section titled "String Bounds" under "VICTORY Data Types."

h. Each type of power distribution system data shall contain data transport configurations.
   i. The contents of data transport configurations are described in the section titled "Data Transport Configuration" under "VICTORY Data Interface Complex Types."
   ii. The transport configurations for each type of power distribution system data shall be controlled independently.

i. Setting the data transport configurations results in the removal of all pre-existing transport configurations and the addition of all transport configurations provided in the request.

j. The following errors are defined for the data transport configuration parameter:
   i. Retrieving, setting, adding, or removing transport configurations for power distribution system data that is not available on the power bus shall result in a NOT_SUPPORTED error.
   ii. Retrieving, setting, adding, or removing transport configurations with a type of power distribution system data not inside the list of known types of power distribution system data may result in an OUT_OF_RANGE error.
   iii. Providing a unique name when retrieving, setting, adding, or removing transport configurations shall result in a GENERIC_FAULT error for the following types of power distribution system data:
      1. PLATFORM_POWER_GENERATION
      2. PLATFORM_POWER
   iv. Not providing a unique name when retrieving, setting, adding, or removing transport configurations shall result in a GENERIC_FAULT error for the following types of power distribution system data:
      1. POWER_SYSTEM_DEVICE
      2. POWER_SYSTEM_DEVICE_THRESHOLD
      3. PLATFORM_BATTERY
   v. Retrieving, setting, adding, or removing transport configurations with an incorrect unique name shall result in a GENERIC_FAULT error for the following types of power distribution system data:
      1. POWER_SYSTEM_DEVICE
      2. POWER_SYSTEM_DEVICE_THRESHOLD
      3. PLATFORM_BATTERY
   vi. Setting multiple data transport configurations with the same unique name shall result in a GENERIC_FAULT error.
   vii. Adding a data transport configuration with a unique name that matches the unique name of one of the pre-existing power data transport configurations shall result in a GENERIC_FAULT error.

viii. Removing a data transport configuration with a unique name that does not match the unique name of one of the power data transport configurations shall result in a GENERIC_FAULT error.
ix. Removing the only data transport configuration for the specified type of power distribution system data shall result in a GENERIC_FAULT error.
x. If network layer is included in the request, setting or adding a data transport configuration with a network layer outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
xi. Setting, adding, or removing a data transport configuration with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
xii. Setting or adding a data transport configuration with a URI outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.

k. Operations include Get(), Set(), Add(), and Remove().

13. ***Nominal Data Available***
    a. Identifies which data parameters from the selected power data of interest are available from the power bus.
    b. Nominal data available shall not be provided for each type of power distribution system data where the data of interest flag is set to False.
        i. Conversely, nominal data available shall be provided for each type of power distribution system data where the data of interest flag is set to True.
    c. For each piece of data provided, a Boolean value of False means the data is not being sent from the power bus, True otherwise.
    d. Operations include Get().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the power distribution system data management parameters along with a description of additional errors that were not specified but may be returned.*

19.1.3 Power Distribution System Device Management Parameters <16006–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the power distribution system device management parameters. A textual description of the power distribution system device management parameters is shown below.

1. ***Power Controller Device Enabled***
    a. This parameter is used to enable or disable the output of a power controller device.
    b. The device shall be specified by its unique identifier.
    c. A Boolean value of False means all power outputs for the device must be dropped to zero and the power mode status value must be set to OFF.
        i. For more details on power mode status, see the section titled "Power System Device" under "Power Distribution System Interface Complex Types."
    d. The following errors are defined for the power controller device enabled parameter:
        i. Retrieving or setting the power controller device enabled flag with an incorrect device ID shall result in a GENERIC_FAULT error.
        ii. Retrieving or setting the power controller device enabled flag with a device ID outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
        iii. Retrieving or setting the power controller device enabled flag for a device that is not a power controller device (i.e., the type of device is not POWER_CONTROLLER) shall result in a NOT_SUPPORTED error.
        iv. Retrieving or setting the power controller device enabled flag for a device where POWER_SYSTEM_DEVICE data is not available or is not being monitored (data of interest flag set to False) shall result in a NOT_SUPPORTED error.
    e. Operations include Get() and Set().
2. ***Device Trip Status***
    a. This parameter provides the "tripped" state of a power system device. This parameter can also be used to reset a power system device that is in the TRIPPED state.
    b. The device shall be specified by its unique identifier.
    c. The contents of the trip status are described in the section titled "Power System Device" under "Power Distribution System Interface Complex Types."
    d. The following errors are defined for the device trip status parameter:

i. Getting or resetting the device trip status with an incorrect device ID shall result in a GENERIC_FAULT error.
ii. Resetting a device whose trip status is LOAD_SHED shall result in a NOT_SUPPORTED error.
iii. Retrieving or resetting the device trip status with a device ID outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
iv. Retrieving or resetting the device trip status for a device that is a virtual device (i.e., the type of device is VIRTUAL_DEVICE) shall result in a NOT_SUPPORTED error.
v. Retrieving or resetting the device trip status for a device where POWER_SYSTEM_DEVICE data is not available or is not being monitored (data of interest flag set to False) shall result in a NOT_SUPPORTED error.

e. Operations include Get() and Reset().
i. The Reset() operation will cause the device to reset the current trip state to NOT_TRIPPED.

3. ***APU Enabled***
a. This parameter is used to control the operation of an APU on the platform.
b. A Boolean value of False means all power outputs for the device must be dropped to zero and the output enabled flag must be set to False.
i. For more details on the output enabled flag, see the section titled "Auxiliary Power Unit" under "Power Distribution System Interface Complex Types."
c. The following errors are defined for the APU enabled parameter:
i. Retrieving or setting the APU enabled flag when PLATFORM_POWER_GENERATION data is not available from the power bus or is not being monitored (data of interest flag set to False) shall result in a NOT_SUPPORTED error.
ii. Retrieving or setting the APU enabled flag when there is no auxiliary power unit on the platform shall result in a NOT_SUPPORTED error.
d. Operations include Get() and Set().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the power distribution system device management parameters along with a description of additional errors that were not specified but may be returned.*

19.1.4 Power Distribution System Interface Complex Types <20026–20210409, Pro>

1. ***Power System Device***
a. **Device ID**
i. The identifier of the power system device.
ii. The bounds of the device identifier are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The device identifier shall be unique within the scope of the VDB.
iv. The device identifier shall be provided as part of the power system device data.
b. **Type**
i. The type of device.
ii. The device type shall be a string chosen from one of the following values:
1. POWER_CONTROLLER
2. VIRTUAL_DEVICE
3. SYSTEM_POWERED_DEVICE
iii. The type shall be provided as part of the power system device data.
c. **Name**
i. The human-readable name of the power system device.
ii. The bounds of the device name are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The device name shall be provided as part of the power system device data.
d. **Current**
i. The current draw of the power system device.
ii. The current shall be a real value.
iii. The current shall be in units of Amps.
iv. The current may be provided as part of the power system device data.
e. **Low Voltage**
i. The voltage of the power system device with a low voltage supply system.

ii. The voltage shall be a real value.
iii. The voltage shall be in units of Volts.
iv. The voltage may be provided as part of the power system device data.

f. **Temperature**
i. The temperature of the power system device.
ii. The temperature shall be a real value.
iii. The temperature shall be greater than or equal to –273.15.
iv. The temperature shall be in units of degrees Celsius.
v. The temperature may be provided as part of the power system device data.

g. **Trip Status**
i. The trip status of circuit protection for the power system device.
ii. The trip status shall be a string chosen from one of the following values:
1. TRIPPED
2. NOT_TRIPPED
3. LOAD_SHED
iii. The trip status may be provided as part of the power system device data.

h. **Combat Override**
i. The status of the power system device combat override.
ii. A Boolean value of True indicates that the device is in a combat override state, False otherwise.
iii. The combat override status may be provided as part of the power system device data.

i. **Power Mode Status**
i. The power mode status of the power system device.
ii. The power mode status shall be a string chosen from one of the following values:
1. OFF
2. STANDBY
3. LOW
4. NORM
iii. The power mode status may be provided as part of the power system device data.

j. **Power Connector Disconnect Interlock Status**
i. The connector disconnect interlock status for each power connector of the power system device.
ii. The connector disconnect interlock status shall be a string chosen from one of the following values:
1. GOOD
2. FAULT
iii. The connector disconnect interlock status may be provided as part of the power system device data.

2. ***Power System Device Threshold***

a. **Device ID**
i. The identifier of the power system device.
ii. The bounds of the device identifier are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The device identifier shall be unique within the scope of the VDB.
iv. The device identifier shall be provided as part of the power system device threshold data.

b. **Type**
i. The type of device.
ii. The device type shall be a string chosen from one of the following values:
1. POWER_CONTROLLER
2. VIRTUAL_DEVICE
3. SYSTEM_POWERED_DEVICE
iii. The type shall be provided as part of the power system device threshold data.

c. **Name**
i. The human-readable name of the power system device.
ii. The bounds of the device name are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The device name shall be provided as part of power system device threshold data.

d. **Low Voltage Thresholds**
i. Low-voltage thresholds may include an upper and lower voltage threshold.
ii. Upper and lower voltage thresholds shall be real values.
iii. Upper and lower voltage thresholds shall be in units of Volts.
iv. Low-voltage thresholds may include a current trip threshold and a current on-line threshold.
v. Current trip and current on-line thresholds shall be a real value.

vi. Current trip and current on-line thresholds shall be in units of Amps.
vii. Low-voltage thresholds may be provided as part of the power system device threshold data.

e. **Temperature Thresholds**
i. Temperature thresholds may include an upper and lower threshold.
ii. Each temperature threshold shall be a real value.
iii. Each temperature threshold shall be greater than or equal to –273.15.
iv. Each temperature threshold shall be in units of degrees Celsius.
v. Temperature thresholds may be provided as part of the power system device threshold data.

3. ***Platform Battery***

a. **Battery ID**
i. The identifier of the battery on the platform.
ii. The bounds of the battery identifier are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The battery identifier shall be unique within the scope of the VDB.
iv. The battery identifier shall be provided as part of the platform battery data.

b. **Name**
i. The human-readable name of the battery on the platform.
ii. The bounds of the battery name are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The battery name shall be provided as part of the platform battery data.

c. **State of Charge**
i. The state of charge of the battery on the platform.
ii. The state of charge shall be a real value.
iii. The state of charge shall be greater than or equal to zero and less than or equal to 100.
iv. The state of charge value shall be in units of percent.
v. The state of charge may be provided as part of the platform battery data.

d. **State of Health**
i. The state of health of the battery on the platform.
ii. The state of health shall be a real value.
iii. The state of health shall be greater than or equal to zero and less than or equal to 100.
iv. The state of health shall be in units of percent.
v. The state of health may be provided as part of the platform battery data.

e. **Current**
i. The current of the battery on the platform.
ii. The current shall be a real value.
iii. The current shall be in units of Amps.
iv. The current may be provided as part of the platform battery data.

f. **Low Voltage**
i. The voltage of the battery with a low voltage supply system.
ii. The voltage shall be a real value.
iii. The voltage shall be in units of Volts.
iv. The voltage may be provided as part of the platform battery data.

g. **Temperature**
i. The temperature of the battery on the platform.
ii. The temperature shall be a real value.
iii. The temperature shall be greater than or equal to –273.15.
iv. The temperature shall be in units of degrees Celsius.
v. The temperature may be provided as part of the platform battery data.

h. **Charge Enabled**
i. Used to indicate if the battery is being charged.
ii. A Boolean value of True indicates the battery is charging, False otherwise.
iii. The state of the charge enabled request shall be provided as part of the platform battery data.

i. **Interconnected**
i. The status of the battery interconnect.
ii. A Boolean value of True indicates the battery is connected, False otherwise.
iii. The status of the battery interconnect may be provided as part of the platform battery data.

j. **Charge Current**
i. The charge current of the battery on the platform.
ii. The charge current shall be a real value.

iii. The charge current shall be in units of Amps.
iv. The charge current may be provided as part of the platform battery data.

k. **Charge Voltage**
i. The charge voltage of the battery on the platform.
ii. The charge voltage shall be a real value.
iii. The charge voltage shall be in units of Volts.
iv. The charge voltage may be provided as part of the platform battery data.

4. ***Generator***
a. **Name**
i. The human-readable name of the generator on the platform.
ii. The bounds of the generator name are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The generator name shall be provided as part of the generator data.

b. **Voltage**
i. The voltage of the generator on the platform.
ii. The voltage shall be a real value.
iii. The voltage shall be in units of Volts.
iv. The voltage may be provided as part of the generator data.

c. **Temperature**
i. The temperature of the generator on the platform.
ii. The temperature shall be a real value.
iii. The temperature shall be greater than or equal to –273.15.
iv. The temperature shall be in units of degrees Celsius.
v. The temperature may be provided as part of the generator data.

d. **Shaft Speed**
i. The shaft speed of the generator on the platform.
ii. The shaft speed shall be a real value.
iii. The shaft speed shall be greater than or equal to zero.
iv. The shaft speed shall be in units of revolutions per minute.
v. The shaft speed may be provided as part of the generator data.

e. **Utilization**
i. The percent utilization of the generator on the platform.
ii. The percent utilization shall be a real value.
iii. The percent utilization shall be greater than or equal to zero and less than or equal to 100.
iv. The percent utilization shall be in units of percent.
v. The percent utilization may be provided as part of the generator data.

f. **Faults**
i. The status of all possible generator faults.
ii. A generator fault shall contain a fault code and a fault status.
iii. The bounds of the fault code are described in the section titled "String Bounds" under "VICTORY Data Types."
iv. The fault status shall be a string chosen from one of the following values:
1. GOOD
2. FAULTED
v. The generator may report on any number of faults.
vi. The faults may be provided as part of the generator data.

g. **Output Enabled**
i. The output status of the generator.
ii. A Boolean value of True indicates the generator power output is enabled, False otherwise.
iii. The output status shall be provided as part of the generator data.

5. ***Auxiliary Power Unit***
a. **Name**
i. The human-readable name of the auxiliary power unit on the platform.
ii. The bounds of the auxiliary power unit name are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The auxiliary power unit name shall be provided as part of the auxiliary power unit data.

b. **Power**
i. The power of the auxiliary power unit on the platform.
ii. The power shall be a real value.

iii. The power shall be in units of watts.
iv. The power may be provided as part of the auxiliary power unit data.

c. **Voltage**
i. The voltage of the auxiliary power unit on the platform.
ii. The voltage shall be a real value.
iii. The voltage shall be in units of Volts.
iv. The voltage may be provided as part of the auxiliary power unit data.

d. **Temperature**
i. The temperature of the auxiliary power unit on the platform.
ii. The temperature shall be a real value.
iii. The temperature shall be greater than or equal to –273.15.
iv. The temperature shall be in units of degrees Celsius.
v. The temperature may be provided as part of the auxiliary power unit data.

e. **Faults**
i. The status of all possible auxiliary power unit faults.
ii. An auxiliary power unit fault shall contain a fault code and a fault status.
iii. The bounds of the fault code are described in the section titled "String Bounds" under "VICTORY Data Types."
iv. The fault status shall be a string chosen from one of the following values:
1. GOOD
2. FAULTED
v. The auxiliary power unit may report on any number of faults.
vi. The faults may be provided as part of the auxiliary power unit data.

f. **Output Enabled**
i. The output status of the auxiliary power unit.
ii. A Boolean value of True indicates the power output of the auxiliary power unit is enabled, False otherwise.
iii. The output status shall be provided as part of the auxiliary power unit data.

6. ***Platform Power***

> ***NOTE:*** *Though all platform power data is optional, at least one of the following pieces of platform power data shall be provided in order to be VICTORY-compliant.*

a. **Instantaneous Power Consumed**
i. The instantaneous power being consumed by the entire platform.
ii. The instantaneous power consumed shall be a real value.
iii. The instantaneous power consumed shall be in units of watts.
iv. The instantaneous power consumed may be provided as part of the platform power data.

b. **Instantaneous Power Generated**
i. The instantaneous power being generated by the platform.
ii. The instantaneous power generated value shall be a real value.
iii. The instantaneous power generated value shall be in units of watts.
iv. The instantaneous power generated may be provided as part of the platform power data.

c. **System Voltage**
i. The voltage of the power system on the platform.
ii. The voltage value shall be a real value.
iii. The voltage value shall be in units of Volts.
iv. The system voltage may be provided as part of the platform power data.

d. **Total Instantaneous Engine Power**
i. The amount of the engine power produced on the platform.
ii. The power value shall be a real value.
iii. The power value shall be in units of kilowatts (kW).
iv. The power may be provided as part of the platform power data.

e. **Available Engine Power**
i. The amount of engine power available on the platform.
ii. The available power value shall be a real value.
iii. The available power value shall be in units of kilowatts (kW).
iv. The available power may be provided as part of the platform power data.

19.2 Automotive System Interfaces

The platform Automotive System (vetronics) is implemented to control and provide status of vehicular entities, such as the engine, transmission, and vehicle fuel levels. Typically, these systems require a non-Ethernet-based data bus to connect the electronics associated with the various vehicular mechanical systems. Although this system is outside the purview of the VICTORY architecture, VICTORY provides a standard interface for receiving status and diagnostics information from the platform Automotive System.

19.2.1 Automotive System Data Specifications

The Automotive System Data Interface provides information concerning the status of vehicular entities. This information will be available to all VDB network nodes that require automotive data.

The Automotive System Data Interface may provide VICTORY Data Messages for the following automotive data types:

- Ambient Conditions
- Dash Display
- Direct Lamp Control Data 1
- Direct Lamp Control Data 2
- Distance
- Electrical Power 1
- Electrical Power 2
- Electronic Transmission Controller 2
- Engine Fluid Level and Pressure 1
- Engine Hours and Revolutions
- Engine Speed and Load Factor
- Engine Temperature 1
- Engine Temperature 2
- Engine Valve
- Fuel Consumption
- Fuel Economy

19.2.1.1 Automotive Ambient Conditions VICTORY Data Message <20040–20120725, Pro>

19.2.1.1.1 Automotive Ambient Conditions Message Content

Table 222. Ambient Conditions VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the ambient conditions data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the ambient conditions VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each ambient conditions VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Ambient Conditions	The ambient conditions data received from the automotive bus.	Ambient Conditions	N/A	N/A	Required	N/A

19.2.1.1.2 Automotive Ambient Conditions Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Ambient Conditions VDM. An example Automotive Ambient Conditions VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveAmbientConditionsMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:barometricPressure>
            <vst:data>12.5</vst:data>
        </vst:barometricPressure>
        <vst:cabInteriorTemperature>
            <vst:data>-272.96875</vst:data>
        </vst:cabInteriorTemperature>
        <vst:ambientAirTemperature>
            <vst:error>ERROR</vst:error>
        </vst:ambientAirTemperature>
        <vst:engineAirIntakeTemperature>
            <vst:data>10</vst:data>
        </vst:engineAirIntakeTemperature>
    </automotiveAmbientConditionsMessage>
</vdm:messages>
```

Figure 34. Example Automotive Ambient Conditions VDM

19.2.1.2 Automotive Dash Display VICTORY Data Message <20041–20120725, Pro>

19.2.1.2.1 Automotive Dash Display Message Content

Table 223. Dash Display VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the dash display data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the dash display VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each dash display VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Dash Display	The dash display data received from the automotive bus.	Dash Display	N/A	N/A	Required	N/A

19.2.1.2.2 Automotive Dash Display Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Dash Display VDM. An example Automotive Dash Display VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveDashDisplayMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:washerFluidLevel>
            <vst:data>50</vst:data>
        </vst:washerFluidLevel>
        <vst:engineFuelFilterDifferentialPressure>
            <vst:data>10</vst:data>
```

```
        </vst:engineFuelFilterDifferentialPressure>
        <vst:engineOilFilterDifferentialPressure>
            <vst:error>ERROR</vst:error>
        </vst:engineOilFilterDifferentialPressure>
        <vst:cargoAmbientTemperature>
            <vst:data>-86.575</vst:data>
        </vst:cargoAmbientTemperature>
        <vst:fuelLevel1>
            <vst:error>ERROR</vst:error>
        </vst:fuelLevel1>
        <vst:fuelLevel2>
            <vst:error>ERROR</vst:error>
        </vst:fuelLevel2>
    </automotiveDashDisplayMessage>
</vdm:messages>
```

Figure 35. Example Automotive Dash Display VDM

19.2.1.3 Automotive Direct Lamp Control Data 1 VICTORY Data Message <20042–20120725, Pro>

19.2.1.3.1 Automotive Direct Lamp Control Data 1 Message Content

Table 224. Direct Lamp Control Data 1 VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the direct lamp control 1 data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the direct lamp control 1 VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each direct lamp control 1 VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Direct Lamp Control Data 1	The direct lamp control 1 data received from the automotive bus.	Direct Lamp Control Data 1	N/A	N/A	Required	N/A

19.2.1.3.2 Automotive Direct Lamp Control Data 1 Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Direct Lamp Control Data 1 VDM. An example Automotive Direct Lamp Control Data 1 VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveDirectLampControlData1Message interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:engineProtectLamp>
            <vst:data>true</vst:data>
        </vst:engineProtectLamp>
        <vst:engineAmberWarningLamp>
            <vst:data>false</vst:data>
        </vst:engineAmberWarningLamp>
        <vst:engineRedStopLamp>
            <vst:error>ERROR</vst:error>
        </vst:engineRedStopLamp>
        <vst:obdMalfunctionIndicatorLamp>
            <vst:error>ERROR</vst:error>
        </vst:obdMalfunctionIndicatorLamp>
        <vst:engineBrakeActiveLamp>
            <vst:data>true</vst:data>
        </vst:engineBrakeActiveLamp>
        <vst:compressionBrakeEnableSwitchIndicatorLamp>
            <vst:error>NOT_AVAILABLE</vst:error>
        </vst:compressionBrakeEnableSwitchIndicatorLamp>
        <vst:engineOilPressureLowLamp>
            <vst:data>true</vst:data>
        </vst:engineOilPressureLowLamp>
        <vst:engineCoolantTemperatureHighLamp>
            <vst:data>true</vst:data>
        </vst:engineCoolantTemperatureHighLamp>
        <vst:engineCoolantLevelLowLamp>
            <vst:data>true</vst:data>
        </vst:engineCoolantLevelLowLamp>
        <vst:engineIdleManagementActiveLamp>
            <vst:data>false</vst:data>
        </vst:engineIdleManagementActiveLamp>
        <vst:engineAirFilterRestrictionLamp>
            <vst:data>false</vst:data>
        </vst:engineAirFilterRestrictionLamp>
        <vst:engineFuelFilterRestrictionLamp>
            <vst:error>ERROR</vst:error>
        </vst:engineFuelFilterRestrictionLamp>
        <vst:engineWaitToStartLamp>
            <vst:error>NOT_AVAILABLE</vst:error>
        </vst:engineWaitToStartLamp>
    </automotiveDirectLampControlData1Message>
</vdm:messages>
```

Figure 36. Example Automotive Direct Lamp Control Data 1 VDM

19.2.1.4 Automotive Direct Lamp Control Data 2 VICTORY Data Message <20043–20120725, Pro>

19.2.1.4.1 Automotive Direct Lamp Control Data 2 Message Content

Table 225. Direct Lamp Control Data 2 VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the direct lamp control 2 data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the direct lamp control 2 VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each direct lamp control 2 VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Direct Lamp Control Data 2	The direct lamp control 2 data received from the automotive bus.	Direct Lamp Control Data 2	N/A	N/A	Required	N/A

19.2.1.4.2 Automotive Direct Lamp Control Data 2 Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Direct Lamp Control Data 2 VDM. An example Automotive Direct Lamp Control Data 2 VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveDirectLampControlData2Message interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:vehicleBatteryVoltageLowLamp>
            <vst:data>true</vst:data>
        </vst:vehicleBatteryVoltageLowLamp>
        <vst:vehicleFuelLevelLowLamp>
```

```
            <vst:data>true</vst:data>
        </vst:vehicleFuelLevelLowLamp>
        <vst:vehicleAirPressureLowLamp>
            <vst:data>false</vst:data>
        </vst:vehicleAirPressureLowLamp>
        <vst:vehicleHvacRecirculationLamp>
            <vst:data>false</vst:data>
        </vst:vehicleHvacRecirculationLamp>
        <vst:vehicleBatteryChargingLamp>
            <vst:error>NOT_AVAILABLE</vst:error>
        </vst:vehicleBatteryChargingLamp>
    </automotiveDirectLampControlData2Message>
</vdm:messages>
```

Figure 37. Example Automotive Direct Lamp Control Data 2 VDM

19.2.1.5 Automotive Distance VICTORY Data Message <20044–20120725, Pro>

19.2.1.5.1 Automotive Distance Message Content

Table 226. Distance VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the distance data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the distance VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each distance VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Distance	The distance data received from the automotive bus.	Distance	N/A	N/A	Required	N/A

19.2.1.5.2 Automotive Distance Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Distance VDM. An example Automotive Distance VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveDistanceMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:tripDistance>
            <vst:data>50</vst:data>
        </vst:tripDistance>
        <vst:totalVehicleDistance>
            <vst:data>500</vst:data>
        </vst:totalVehicleDistance>
    </automotiveDistanceMessage>
</vdm:messages>
```

Figure 38. Example Automotive Distance VDM

19.2.1.6 Automotive Electrical Power 1 VICTORY Data Message <20045–20120725, Pro>

19.2.1.6.1 Automotive Electrical Power 1 Message Content

Table 227. Electrical Power 1 VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the electrical power 1 data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the electrical power 1 VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each electrical power 1 VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Electrical Power 1	The electrical power 1 data received from the automotive bus.	Electrical Power 1	N/A	N/A	Required	N/A

19.2.1.6.2 Automotive Electrical Power 1 Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Electrical Power 1 VDM. An example Automotive Electrical Power 1 VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveElectricalPower1Message interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:netBatteryCurrent>
            <vst:error>ERROR</vst:error>
        </vst:netBatteryCurrent>
        <vst:alternatorCurrent>
            <vst:data>10</vst:data>
        </vst:alternatorCurrent>
        <vst:chargingSystemPotential>
            <vst:data>50</vst:data>
        </vst:chargingSystemPotential>
        <vst:powerInput1>
            <vst:error>ERROR</vst:error>
        </vst:powerInput1>
        <vst:keyswitchBatteryPotential>
            <vst:data>25</vst:data>
        </vst:keyswitchBatteryPotential>
    </automotiveElectricalPower1Message>
</vdm:messages>
```

Figure 39. Example Automotive Electrical Power 1 VDM

19.2.1.7 Automotive Electrical Power 2 VICTORY Data Message <20046–20120725, Pro>

19.2.1.7.1 Automotive Electrical Power 2 Message Content

Table 228. Electrical Power 2 VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the electrical power 2 data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the electrical power 2 VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each electrical power 2 VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Electrical Power 2	The electrical power 2 data received from the automotive bus.	Electrical Power 2	N/A	N/A	Required	N/A

19.2.1.7.2 Automotive Electrical Power 2 Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Electrical Power 2 VDM. An example Automotive Electrical Power 2 VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveElectricalPower2Message interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:powerInput2>
            <vst:data>50</vst:data>
        </vst:powerInput2>
        <vst:ecuPowerOutputSupplyVoltage1>
            <vst:data>50</vst:data>
```

```
        </vst:ecuPowerOutputSupplyVoltage1>
        <vst:ecuPowerOutputSupplyVoltage2>
            <vst:data>50</vst:data>
        </vst:ecuPowerOutputSupplyVoltage2>
        <vst:ecuPowerOutputSupplyVoltage3>
            <vst:data>50</vst:data>
        </vst:ecuPowerOutputSupplyVoltage3>
    </automotiveElectricalPower2Message>
</vdm:messages>
```

Figure 40. Example Automotive Electrical Power 2 VDM

19.2.1.8 Automotive Electronic Transmission Controller 2 VICTORY Data Message <20047–20120725, Pro>

19.2.1.8.1 Automotive Electronic Transmission Controller 2 Message Content

Table 229. Electronic Transmission Controller 2 VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the electronic transmission controller 2 data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the electronic transmission controller 2 VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each electronic transmission controller 2 VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Electronic Transmission Controller 2	The electronic transmission controller 2 data received from the automotive bus.	Electronic Transmission Controller 2	N/A	N/A	Required	N/A

19.2.1.8.2 Automotive Electronic Transmission Controller 2 Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Electronic Transmission Controller 2 VDM. An example Automotive Electronic Transmission Controller 2 VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveElectronicTransmissionController2Message interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:transmissionCurrentGear>
            <vst:data>80</vst:data>
        </vst:transmissionCurrentGear>
        <vst:transmissionActualGearRatio>
            <vst:data>40</vst:data>
        </vst:transmissionActualGearRatio>
        <vst:transmissionSelectedGear>
            <vst:error>ERROR</vst:error>
        </vst:transmissionSelectedGear>
        <vst:transmissionRequestedRange>
            <vst:data>DRIVE</vst:data>
        </vst:transmissionRequestedRange>
        <vst:transmissionCurrentRange>
            <vst:data>DRIVE</vst:data>
        </vst:transmissionCurrentRange>
    </automotiveElectronicTransmissionController2Message>
</vdm:messages>
```

Figure 41. Example Automotive Electronic Transmission Controller 2 VDM

19.2.1.9 Automotive Engine Fluid Level and Pressure 1 VICTORY Data Message <20048–20120725, Pro>

19.2.1.9.1 Automotive Engine Fluid Level and Pressure 1 Message Content

Table 230. Engine Fluid Level and Pressure 1 VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the engine fluid level and pressure 1 data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Sequence Number	A monotonically increasing number that indicates the sequence number of the engine fluid level and pressure 1 VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each engine fluid level and pressure 1 VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Engine Fluid Level and Pressure 1	The engine fluid level and pressure 1 data received from the automotive bus.	Engine Fluid Level and Pressure 1	N/A	N/A	Required	N/A

19.2.1.9.2 Automotive Engine Fluid Level and Pressure 1 Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Engine Fluid Level and Pressure 1 VDM. An example Automotive Engine Fluid Level and Pressure 1 VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveEngineFluidLevelAndPressure1Message interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:engineFuelDeliveryPressure>
            <vst:error>ERROR</vst:error>
        </vst:engineFuelDeliveryPressure>
        <vst:engineExtendedCrankcaseBlowbyPressure>
            <vst:data>30</vst:data>
        </vst:engineExtendedCrankcaseBlowbyPressure>
        <vst:engineOilLevel>
            <vst:data>50</vst:data>
        </vst:engineOilLevel>
        <vst:engineOilPressure>
            <vst:error>ERROR</vst:error>
        </vst:engineOilPressure>
        <vst:engineCrankcasePressure>
            <vst:error>ERROR</vst:error>
        </vst:engineCrankcasePressure>
        <vst:engineCoolantPressure>
            <vst:error>ERROR</vst:error>
```

```
        </vst:engineCoolantPressure>
        <vst:engineCoolantLevel>
            <vst:data>25</vst:data>
        </vst:engineCoolantLevel>
    </automotiveEngineFluidLevelAndPressure1Message>
</vdm:messages>
```

Figure 42. Example Automotive Engine Fluid Level and Pressure 1 VDM

19.2.1.10 Automotive Engine Hours and Revolutions VICTORY Data Message <20049–20120725, Pro>

19.2.1.10.1 Automotive Engine Hours and Revolutions Message Content

Table 231. Engine Hours and Revolutions VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the engine hours and revolutions data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the engine hours and revolutions VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each engine hours and revolutions VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Engine Hours and Revolutions	The engine hours and revolutions data received from the automotive bus.	Engine Hours and Revolutions	N/A	N/A	Required	N/A

19.2.1.10.2 Automotive Engine Hours and Revolutions Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Engine Hours and Revolutions VDM. An example Automotive Engine Hours and Revolutions VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
```

```
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveEngineHoursAndRevolutionsMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:engineHoursOfOperation>
            <vst:data>245</vst:data>
        </vst:engineHoursOfOperation>
        <vst:engineRevolutions>
            <vst:data>2456</vst:data>
        </vst:engineRevolutions>
    </automotiveEngineHoursAndRevolutionsMessage>
</vdm:messages>
```

Figure 43. Example Automotive Engine Hours and Revolutions VDM

19.2.1.11 Automotive Engine Speed and Load Factor VICTORY Data Message <20050–20120725, Pro>

19.2.1.11.1 Automotive Engine Speed and Load Factor Message Content

Table 232. Engine Speed and Load Factor VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the engine speed and load factor data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the engine speed and load factor VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each engine speed and load factor VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Engine Speed and Load Factor	The engine speed and load factor data received from the automotive bus.	Engine Speed and Load Factor	N/A	N/A	Required	N/A

19.2.1.11.2 Automotive Engine Speed and Load Factor Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Engine Speed and Load Factor VDM. An example Automotive Engine Speed and Load Factor VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveEngineSpeedAndLoadFactorMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:tripMaximumEngineSpeed>
            <vst:error>ERROR</vst:error>
        </vst:tripMaximumEngineSpeed>
        <vst:tripAverageEngineSpeed>
            <vst:data>40</vst:data>
        </vst:tripAverageEngineSpeed>
        <vst:tripDriveAverageLoadFactor>
            <vst:data>50</vst:data>
        </vst:tripDriveAverageLoadFactor>
        <vst:totalDriveAverageLoadFactor>
            <vst:error>ERROR</vst:error>
        </vst:totalDriveAverageLoadFactor>
        <vst:totalEngineCruiseTime>
            <vst:data>1000</vst:data>
        </vst:totalEngineCruiseTime>
    </automotiveEngineSpeedAndLoadFactorMessage>
</vdm:messages>
```

Figure 44. Example Automotive Engine Speed and Load Factor VDM

19.2.1.12 Automotive Engine Temperature 1 VICTORY Data Message <20051–20120725, Pro>

19.2.1.12.1 Automotive Engine Temperature 1 Message Content

Table 233. Engine Temperature 1 VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the engine temperature 1 data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the engine temperature 1 VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each engine temperature 1 VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Engine Temperature 1	The engine temperature 1 data received from the automotive bus.	Engine Temperature 1	N/A	N/A	Required	N/A

19.2.1.12.2 Automotive Engine Temperature 1 Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Engine Temperature 1 VDM. An example Automotive Engine Temperature 1 VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveEngineTemperature1Message interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:engineCoolantTemperature>
            <vst:error>ERROR</vst:error>
        </vst:engineCoolantTemperature>
```

```
        <vst:engineFuelTemperature1>
            <vst:data>-86.575</vst:data>
        </vst:engineFuelTemperature1>
        <vst:engineOilTemperature1>
            <vst:error>ERROR</vst:error>
        </vst:engineOilTemperature1>
        <vst:engineTurbochargerOilTemperature>
            <vst:error>ERROR</vst:error>
        </vst:engineTurbochargerOilTemperature>
        <vst:engineIntercoolerTemperature>
            <vst:data>-26.575</vst:data>
        </vst:engineIntercoolerTemperature>
        <vst:engineIntercoolerThermostatOpening>
            <vst:error>ERROR</vst:error>
        </vst:engineIntercoolerThermostatOpening>
    </automotiveEngineTemperature1Message>
</vdm:messages>
```

Figure 45. Example Automotive Engine Temperature 1 VDM

19.2.1.13 Automotive Engine Temperature 2 VICTORY Data Message <20052–20120725, Pro>

19.2.1.13.1 Automotive Engine Temperature 2 Message Content

Table 234. Engine Temperature 2 VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the engine temperature 2 data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the engine temperature 2 VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each engine temperature 2 VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Engine Temperature 2	The engine temperature 2 data received from the automotive bus.	Engine Temperature 2	N/A	N/A	Required	N/A

19.2.1.13.2 Automotive Engine Temperature 2 Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Engine Temperature 2 VDM. An example Automotive Engine Temperature 2 VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveEngineTemperature2Message interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:engineOilTemperature2>
            <vst:error>ERROR</vst:error>
        </vst:engineOilTemperature2>
        <vst:engineEcuTemperature>
            <vst:data>-46.575</vst:data>
        </vst:engineEcuTemperature>
        <vst:engineExhaustGasRecirculation1DifferentialPressure>
            <vst:data>0</vst:data>
        </vst:engineExhaustGasRecirculation1DifferentialPressure>
        <vst:engineExhaustGasRecirculation1Temperature>
            <vst:data>-56.235</vst:data>
        </vst:engineExhaustGasRecirculation1Temperature>
    </automotiveEngineTemperature2Message>
</vdm:messages>
```

Figure 46. Example Automotive Engine Temperature 2 VDM

19.2.1.14 Automotive Engine Valve VICTORY Data Message <20053–20120725, Pro>

19.2.1.14.1 Automotive Engine Valve Message Content

Table 235. Engine Valve VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the engine valve data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the engine valve VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each engine valve VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Engine Valve	The engine valve data received from the automotive bus.	Engine Valve	N/A	N/A	Required	N/A

19.2.1.14.2 Automotive Engine Valve Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Engine Valve VDM. An example Automotive Engine Valve VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveEngineValveMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:engineDesiredThrottleValve1Position>
            <vst:error>ERROR</vst:error>
        </vst:engineDesiredThrottleValve1Position>
        <vst:engineThrottleValve1PreliminaryFmi>
            <vst:data>ABNORMAL_UPDATE_RATE</vst:data>
```

```
        </vst:engineThrottleValve1PreliminaryFmi>
        <vst:engineThrottleValve1TemperatureStatus>
            <vst:data>IN_RANGE</vst:data>
        </vst:engineThrottleValve1TemperatureStatus>
        <vst:engineThrottleValve1OperationStatus>
            <vst:data>NORMAL</vst:data>
        </vst:engineThrottleValve1OperationStatus>
        <vst:engineDesiredThrottleValve2Position>
            <vst:data>25</vst:data>
        </vst:engineDesiredThrottleValve2Position>
        <vst:engineThrottleValve2PreliminaryFmi>
            <vst:data>ROOT_CAUSE_NOT_KNOWN</vst:data>
        </vst:engineThrottleValve2PreliminaryFmi>
        <vst:engineThrottleValve2TemperatureStatus>
            <vst:data>IN_RANGE</vst:data>
        </vst:engineThrottleValve2TemperatureStatus>
        <vst:engineThrottleValve2OperationStatus>
            <vst:data>NORMAL</vst:data>
        </vst:engineThrottleValve2OperationStatus>
        <vst:engineFuelValve1PreliminaryFmi>
            <vst:data>SPECIAL_INSTRUCTIONS</vst:data>
        </vst:engineFuelValve1PreliminaryFmi>
        <vst:engineFuelValve1TemperatureStatus>
            <vst:data>IN_RANGE</vst:data>
        </vst:engineFuelValve1TemperatureStatus>
        <vst:engineFuelValve1OperationStatus>
            <vst:data>NORMAL</vst:data>
        </vst:engineFuelValve1OperationStatus>
        <vst:engineFuelValve2PreliminaryFmi>
            <vst:data>BAD_INTELLIGENT_DEVICE_OR_COMPONENT</vst:data>
        </vst:engineFuelValve2PreliminaryFmi>
        <vst:engineFuelValve2TemperatureStatus>
            <vst:data>IN_RANGE</vst:data>
        </vst:engineFuelValve2TemperatureStatus>
        <vst:engineFuelValve2OperationStatus>
            <vst:data>NORMAL</vst:data>
        </vst:engineFuelValve2OperationStatus>
    </automotiveEngineValveMessage>
</vdm:messages>
```

Figure 47. Example Automotive Engine Valve VDM

19.2.1.15 Automotive Fuel Consumption VICTORY Data Message <20054–20120725, Pro>

19.2.1.15.1 Automotive Fuel Consumption Message Content

Table 236. Fuel Consumption VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the fuel consumption data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the fuel consumption VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each fuel consumption VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Fuel Consumption	The fuel consumption data received from the automotive bus.	Fuel Consumption	N/A	N/A	Required	N/A

19.2.1.15.2 Automotive Fuel Consumption Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Fuel Consumption VDM. An example Automotive Fuel Consumption VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveFuelConsumptionMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:engineTripFuel>
```

```
            <vst:data>50</vst:data>
        </vst:engineTripFuel>
        <vst:engineTotalFuelUsed>
            <vst:data>500</vst:data>
        </vst:engineTotalFuelUsed>
    </automotiveFuelConsumptionMessage>
</vdm:messages>
```

Figure 48. Example Automotive Fuel Consumption VDM

19.2.1.16 Automotive Fuel Economy VICTORY Data Message <20055–20120725, Pro>

19.2.1.16.1 Automotive Fuel Economy Message Content

Table 237. Fuel Economy VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the fuel economy data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the fuel economy VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each fuel economy VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Fuel Economy	The fuel economy data received from the automotive bus.	Fuel Economy	N/A	N/A	Required	N/A

19.2.1.16.2 Automotive Fuel Economy Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Fuel Economy VDM. An example Automotive Fuel Economy VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
```

```
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveFuelEconomyMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:engineFuelRate>
            <vst:error>ERROR</vst:error>
        </vst:engineFuelRate>
        <vst:engineInstantaneousFuelEconomy>
            <vst:data>50</vst:data>
        </vst:engineInstantaneousFuelEconomy>
        <vst:engineAverageFuelEconomy>
            <vst:error>ERROR</vst:error>
        </vst:engineAverageFuelEconomy>
        <vst:engineThrottleValve1Position>
            <vst:error>ERROR</vst:error>
        </vst:engineThrottleValve1Position>
        <vst:engineThrottleValve2Position>
            <vst:data>60</vst:data>
        </vst:engineThrottleValve2Position>
    </automotiveFuelEconomyMessage>
</vdm:messages>
```

Figure 49. Example Automotive Fuel Economy VDM

19.2.2 Automotive System Data Specification – V1.6 Extension

The Automotive System Data Interface has been extended to include the status of additional vehicular entities. This information will be available to all VDB network nodes that require these extensions.

The Automotive System Data Interface may provide VICTORY Data Messages for the following automotive data types:

- Active Diagnostic Trouble Codes
- Alternator Temperature
- Battery Temperature
- Cab Message 1
- Combination Vehicle Weight
- Electronic Engine Controller 1
- Electronic Engine Controller 2
- Fan Drive 1
- Tire Condition
- Torque/Speed Control 1

19.2.2.1 Automotive Active Diagnostic Trouble Codes VICTORY Data Message <20056–20210409, Exp>

19.2.2.1.1 Automotive Active Diagnostic Trouble Codes Message Content

Table 238. Active Diagnostic Trouble Codes VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the active diagnostic trouble codes data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the active diagnostic trouble codes VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each active diagnostic trouble codes VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Active Diagnostic Trouble Codes	The active diagnostic trouble codes data received from the automotive bus.	Active Diagnostic Trouble Codes	N/A	N/A	Required	N/A

19.2.2.1.2 Automotive Active Diagnostic Trouble Codes Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Active Diagnostic Trouble Codes VDM. An example Automotive Active Diagnostic Trouble Codes VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveActiveDiagnosticTroubleCodesMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:engineProtectLamp>
            <vst:data>ON</vst:data>
        </vst:engineProtectLamp>
```

```
        <vst:engineAmberWarningLamp>
            <vst:data>ON</vst:data>
        </vst:engineAmberWarningLamp>
        <vst:engineRedStopLamp>
            <vst:data>OFF</vst:data>
        </vst:engineRedStopLamp>
        <vst:obdMalfunctionIndicatorLamp>
            <vst:data>OFF</vst:data>
        </vst:obdMalfunctionIndicatorLamp>
        <vst:flashProtectLamp>
            <vst:data>OFF</vst:data>
        </vst:flashProtectLamp>
        <vst:flashAmberWarningLamp>
            <vst:data>ON</vst:data>
        </vst:flashAmberWarningLamp>
        <vst:flashRedStopLamp>
            <vst:data>ON</vst:data>
        </vst:flashRedStopLamp>
        <vst:flashMalfunctionIndicatorLamp>
            <vst:data>OFF</vst:data>
        </vst:flashMalfunctionIndicatorLamp>
        <vst:suspectParameterNumber>
            <vst:data>SPN111</vst:data>
        </vst:suspectParameterNumber>
        <vst:failureModeIdentifier>
            <vst:error>FMI Error</vst:error>
        </vst:failureModeIdentifier>
        <vst:occuranceCount>
            <vst:data>24</vst:data>
        </vst:occuranceCount>
        <vst:spnConversionMethod>
            <vst:data>1</vst:data>
        </vst:spnConversionMethod>
    </automotiveActiveDiagnosticTroubleCodesMessage>
</vdm:messages>
```

Figure 50. Example Automotive Active Diagnostic Trouble Codes VDM

19.2.2.2 Automotive Alternator Temperature VICTORY Data Message <20057–20140630, Pro>

19.2.2.2.1 Automotive Alternator Temperature Message Content

Table 239. Alternator Temperature VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the alternator temperature data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Sequence Number	A monotonically increasing number that indicates the sequence number of the alternator temperature VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each alternator temperature VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Alternator Temperature	The alternator temperature data received from the automotive bus.	Alternator Temperature	N/A	N/A	Required	N/A

19.2.2.2.2 Automotive Alternator Temperature Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Alternator Temperature VDM. An example Automotive Alternator Temperature VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveAlternatorTemperatureMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:engineAlternatorBearing1Temperature>
            <vst:data>98.6</vst:data>
        </vst:engineAlternatorBearing1Temperature>
        <vst:engineAlternatorBearing2Temperature>
            <vst:data>30</vst:data>
        </vst:engineAlternatorBearing2Temperature>
        <vst:engineAlternatorWinding1Temperature>
            <vst:error>ERROR</vst:error>
        </vst:engineAlternatorWinding1Temperature>
        <vst:engineAlternatorWinding2Temperature>
            <vst:data>10</vst:data>
        </vst:engineAlternatorWinding2Temperature>
        <vst:engineAlternatorWinding3Temperature>
            <vst:data>-10</vst:data>
        </vst:engineAlternatorWinding3Temperature>
    </automotiveAlternatorTemperatureMessage>
</vdm:messages>
```

Figure 51. Example Automotive Alternator Temperature VDM

19.2.2.3 Automotive Battery Temperature VICTORY Data Message <20058–20140630, Pro>

19.2.2.3.1 Automotive Battery Temperature Message Content

Table 240. Battery Temperature VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the battery temperature data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the battery temperature VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each battery temperature VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Battery Temperature	The battery temperature data received from the automotive bus.	Battery Temperature	N/A	N/A	Required	N/A

19.2.2.3.2 Automotive Battery Temperature Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Battery Temperature VDM. An example Automotive Battery Temperature VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveBatteryTemperatureMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
```

```
        <vst:battery1Temperature>
            <vst:error>ERROR</vst:error>
        </vst:battery1Temperature>
        <vst:battery2Temperature>
            <vst:data>20.1</vst:data>
        </vst:battery2Temperature>
    </automotiveBatteryTemperatureMessage>
</vdm:messages>
```

Figure 52. Example Automotive Battery Temperature VDM

19.2.2.4 Automotive Cab Message 1 VICTORY Data Message <20059–20140630, Pro>

19.2.2.4.1 Automotive Cab Message 1 Message Content

Table 241. Cab Message 1 VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the cab message 1 data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the cab message 1 VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each cab message 1 VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Cab Message 1	The cab message 1 data received from the automotive bus.	Cab Message 1	N/A	N/A	Required	N/A

19.2.2.4.2 Automotive Cab Message 1 Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Cab Message 1 VDM. An example Automotive Cab Message 1 VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
```

```
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveCabMessage1Message interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:requestedPercentFanSpeed>
            <vst:error>ERROR</vst:error>
        </vst:requestedPercentFanSpeed>
        <vst:operatorSeatDirectionSwitch>
            <vst:data>false</vst:data>
        </vst:operatorSeatDirectionSwitch>
        <vst:seatBeltSwitch>
            <vst:error>ERROR</vst:error>
        </vst:seatBeltSwitch>
        <vst:vlsgDecrementSwitch>
            <vst:data>true</vst:data>
        </vst:vlsgDecrementSwitch>
        <vst:vlsgIncrementSwitch>
            <vst:error>ERROR</vst:error>
        </vst:vlsgIncrementSwitch>
        <vst:vlsgEnableSwitch>
            <vst:error>ERROR</vst:error>
        </vst:vlsgEnableSwitch>
        <vst:dieselParticulateFilterRegenerationInhibitSwitch>
            <vst:data>true</vst:data>
        </vst:dieselParticulateFilterRegenerationInhibitSwitch>
        <vst:dieselParticulateFilterRegenerationForceSwitch>
            <vst:error>ERROR</vst:error>
        </vst:dieselParticulateFilterRegenerationForceSwitch>
        <vst:automaticGearShiftingEnableSwitch>
            <vst:data>false</vst:data>
        </vst:automaticGearShiftingEnableSwitch>
        <vst:engineAutomaticStartEnableSwitch>
            <vst:data>true</vst:data>
        </vst:engineAutomaticStartEnableSwitch>
        <vst:selectedMaximumVehicleSpeedLimit>
            <vst:data>6.2</vst:data>
        </vst:selectedMaximumVehicleSpeedLimit>
    </automotiveCabMessage1Message>
</vdm:messages>
```

Figure 53. Example Automotive Cab Message 1 VDM

19.2.2.5 Automotive Combination Vehicle Weight VICTORY Data Message <20060–20140630, Pro>

19.2.2.5.1 Automotive Combination Vehicle Weight Message Content

Table 242. Combination Vehicle Weight VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the combination vehicle weight data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the combination vehicle weight VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each combination vehicle weight VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Combination Vehicle Weight	The combination vehicle weight data received from the automotive bus.	Combination Vehicle Weight	N/A	N/A	Required	N/A

19.2.2.5.2 Automotive Combination Vehicle Weight Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Combination Vehicle Weight VDM. An example Automotive Combination Vehicle Weight VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveCombinationVehicleWeightMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:poweredVehicleWeight>
            <vst:data>2300</vst:data>
        </vst:poweredVehicleWeight>
```

```
        <vst:grossCombinationVehicleWeight>
            <vst:data>4330</vst:data>
        </vst:grossCombinationVehicleWeight>
    </automotiveCombinationVehicleWeightMessage>
</vdm:messages>
```

Figure 54. Example Automotive Combination Vehicle Weight VDM

19.2.2.6 Automotive Electronic Engine Controller 1 VICTORY Data Message <20061–20140630, Pro>

19.2.2.6.1 Automotive Electronic Engine Controller 1 Message Content

Table 243. Electronic Engine Controller 1 VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the electronic engine controller 1 data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the electronic engine controller 1 VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each electronic engine controller 1 VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Electronic Engine Controller 1	The electronic engine controller 1 data received from the automotive bus.	Electronic Engine Controller 1	N/A	N/A	Required	N/A

19.2.2.6.2 Automotive Electronic Engine Controller 1 Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Electronic Engine Controller 1 VDM. An example Automotive Electronic Engine Controller 1 VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
```

```
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveElectronicEngineController1Message interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:engineTorqueMode>
            <vst:data>CRUISE_CONTROL</vst:data>
        </vst:engineTorqueMode>
        <vst:driversDemandEnginePercentTorque>
            <vst:data>10</vst:data>
        </vst:driversDemandEnginePercentTorque>
        <vst:actualEnginePercentTorque>
            <vst:data>30</vst:data>
        </vst:actualEnginePercentTorque>
        <vst:engineSpeed>
            <vst:data>50</vst:data>
        </vst:engineSpeed>
        <vst:engineStarterMode>
            <vst:data>START_FINISHED</vst:data>
        </vst:engineStarterMode>
        <vst:engineDemandPercentTorque>
            <vst:error>ERROR</vst:error>
        </vst:engineDemandPercentTorque>
    </automotiveElectronicEngineController1Message>
</vdm:messages>
```

Figure 55. Example Automotive Electronic Engine Controller 1 VDM

19.2.2.7 Automotive Electronic Engine Controller 2 VICTORY Data Message <20062–20140630, Pro>

19.2.2.7.1 Automotive Electronic Engine Controller 2 Message Content

Table 244. Electronic Engine Controller 2 VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the electronic engine controller 2 data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the electronic engine controller 2 VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each electronic engine controller 2 VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Electronic Engine Controller 2	The electronic engine controller 2 data received from the automotive bus.	Electronic Engine Controller 2	N/A	N/A	Required	N/A

19.2.2.7.2 Automotive Electronic Engine Controller 2 Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Electronic Engine Controller 2 VDM. An example Automotive Electronic Engine Controller 2 VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveElectronicEngineController2Message interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:acceleratorPedal1LowIdleSwitch>
            <vst:data>false</vst:data>
        </vst:acceleratorPedal1LowIdleSwitch>
        <vst:acceleratorPedalKickdownSwitch>
            <vst:data>true</vst:data>
        </vst:acceleratorPedalKickdownSwitch>
        <vst:roadSpeedLimitStatus>
            <vst:error>ERROR</vst:error>
        </vst:roadSpeedLimitStatus>
        <vst:acceleratorPedal2LowIdleSwitch>
            <vst:data>false</vst:data>
        </vst:acceleratorPedal2LowIdleSwitch>
        <vst:acceleratorPedalPosition1>
            <vst:data>50</vst:data>
        </vst:acceleratorPedalPosition1>
        <vst:enginePercentLoadAtCurrentSpeed>
            <vst:data>62.5</vst:data>
        </vst:enginePercentLoadAtCurrentSpeed>
        <vst:remoteAcceleratorPedalPosition>
            <vst:data>50</vst:data>
        </vst:remoteAcceleratorPedalPosition>
        <vst:acceleratorPedalPosition2>
            <vst:error>ERROR</vst:error>
        </vst:acceleratorPedalPosition2>
        <vst:vehicleAccelerationRateLimitStatus>
            <vst:data>false</vst:data>
        </vst:vehicleAccelerationRateLimitStatus>
        <vst:momentaryEngineMaximumPowerEnableFeedback>
            <vst:error>ERROR</vst:error>
```

```
        </vst:momentaryEngineMaximumPowerEnableFeedback>
        <vst:dpfThermalManagementActive>
            <vst:data>false</vst:data>
        </vst:dpfThermalManagementActive>
        <vst:scrThermalManagementActive>
            <vst:data>true</vst:data>
        </vst:scrThermalManagementActive>
        <vst:actualMaximumAvailableEnginePercentTorque>
            <vst:error>ERROR</vst:error>
        </vst:actualMaximumAvailableEnginePercentTorque>
        <vst:estimatedPumpingPercentTorque>
            <vst:data>0</vst:data>
        </vst:estimatedPumpingPercentTorque>
    </automotiveElectronicEngineController2Message>
</vdm:messages>
```

Figure 56. Example Automotive Electronic Engine Controller 2 VDM

19.2.2.8 Automotive Fan Drive 1 VICTORY Data Message <20063–20140630, Pro>

19.2.2.8.1 Automotive Fan Drive 1 Message Content

Table 245. Fan Drive 1 VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the fan drive 1 data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the fan drive 1 VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each fan drive 1 VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Fan Drive 1	The fan drive 1 data received from the automotive bus.	Fan Drive 1	N/A	N/A	Required	N/A

19.2.2.8.2 Automotive Fan Drive 1 Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Fan Drive 1 VDM. An example Automotive Fan Drive 1 VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveFanDrive1Message interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:estimatedPercentFanSpeed>
            <vst:error>ERROR</vst:error>
        </vst:estimatedPercentFanSpeed>
        <vst:fanDriveState>
            <vst:data>FAN_OFF</vst:data>
        </vst:fanDriveState>
        <vst:fanSpeed>
            <vst:data>50</vst:data>
        </vst:fanSpeed>
        <vst:hydraulicFanMotorPressure>
            <vst:error>ERROR</vst:error>
        </vst:hydraulicFanMotorPressure>
        <vst:fanDriveBypassCommandStatus>
            <vst:data>50</vst:data>
        </vst:fanDriveBypassCommandStatus>
    </automotiveFanDrive1Message>
</vdm:messages>
```

Figure 57. Example Automotive Fan Drive 1 VDM

19.2.2.9 Automotive Tire Condition VICTORY Data Message <20064–20210409, Exp>

19.2.2.9.1 Automotive Tire Condition Message Content

Table 246. Tire Condition VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the tire condition data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Sequence Number	A monotonically increasing number that indicates the sequence number of the tire condition VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each tire condition VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Tire Condition	The tire condition data received from the automotive bus.	Tire Condition	N/A	N/A	Required	N/A

19.2.2.9.2 Automotive Tire Condition Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Tire Condition VDM. An example Automotive Tire Condition VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages
    xmlns:vmt="http://www.victory-standards.org/Schemas/VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveTireConditionMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:tireLocation>
            <vst:data>0</vst:data>
        </vst:tireLocation>
        <vst:tirePressure>
            <vst:data>50</vst:data>
        </vst:tirePressure>
        <vst:tireTemperature>
            <vst:data>-86.575</vst:data>
        </vst:tireTemperature>
        <vst:ctiWheelSensorStatus>
            <vst:data>false</vst:data>
        </vst:ctiWheelSensorStatus>
        <vst:ctiTireStatus>
            <vst:data>false</vst:data>
        </vst:ctiTireStatus>
        <vst:ctiWheelEndElectricalFault>
            <vst:error>ERROR</vst:error>
        </vst:ctiWheelEndElectricalFault>
        <vst:tireAirLeakageRate>
            <vst:error>ERROR</vst:error>
```

```
        </vst:tireAirLeakageRate>
        <vst:tirePressureThresholdDetection>
            <vst:data>NO_WARNING_PRESSURE</vst:data>
        </vst:tirePressureThresholdDetection>
    </automotiveTireConditionMessage>
</vdm:messages>
```

Figure 58. Example Automotive Tire Condition VDM

19.2.2.10 Automotive Torque/Speed Control 1 VICTORY Data Message <20065–20140630, Pro>

19.2.2.10.1 Automotive Torque/Speed Control 1 Message Content

Table 247. Torque/Speed Control 1 VDM Content

Name	Description	Data Type	Range	Units	Applicability	Additional Requirements
Timestamp	The time the torque/speed control 1 data was sampled from the automotive bus.	dateTime	Timestamp Format for Application Layer Data	N/A	Required	The timestamp shall only be updated when new data is received from the automotive bus. This is especially important when the update period of the VDM is at a faster rate than what is available from the automotive bus.
Sequence Number	A monotonically increasing number that indicates the sequence number of the torque/speed control 1 VDM sent from the Automotive System Data Interface.	Integer	x>=0	N/A	Required	The sequence number shall increase by one for each torque/speed control 1 VDM sent from the Automotive System Data Interface and shall roll over to its minimum value once the maximum value has been reached.
Interface ID	An identification number, unique within the scope of the VDB, used to identify the Automotive System Data Interface.	Integer	x>=0	N/A	Required	N/A
Torque/ Speed Control 1	The torque/speed control 1 data received from the automotive bus.	Torque/ Speed Control 1	N/A	N/A	Required	N/A

19.2.2.10.2 Automotive Torque/Speed Control 1 Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as an Automotive Torque/Speed Control 1 VDM. An example Automotive Torque/Speed Control 1 VDM is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vdm:messages xmlns:vmt="http://www.victory-standards.org/Schemas/
VICTORYManagementTypes.xsd"
    xmlns:vst="http://www.victory-standards.org/Schemas/VICTORYSharedTypes.xsd"
```

```
    xmlns:tns="http://www.victory-standards.org/Schemas/VICTORYMessages.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd"
    xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VDM.xsd VDM.xsd">
    <automotiveTorqueSpeedControl1Message interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z" sequenceNumber="10">
        <vst:engineOverrideControlMode>
            <vst:data>SPEED_CONTROL</vst:data>
        </vst:engineOverrideControlMode>
        <vst:engineRequestedSpeedControlConditions>
            <vst:data>
                TRANSIENT_OPTIMIZED_DRIVELINE_DISENGAGED_NON_LOCKUP_CONDITIONS
            </vst:data>
        </vst:engineRequestedSpeedControlConditions>
        <vst:overrideControlModePriority>
            <vst:data>HIGH</vst:data>
        </vst:overrideControlModePriority>
        <vst:engineRequestedSpeedAndSpeedLimit>
            <vst:error>ERROR</vst:error>
        </vst:engineRequestedSpeedAndSpeedLimit>
        <vst:engineRequestedTorqueAndTorqueLimit>
            <vst:data>100</vst:data>
        </vst:engineRequestedTorqueAndTorqueLimit>
        <vst:tsc1TransmissionRate>
            <vst:error>ERROR</vst:error>
        </vst:tsc1TransmissionRate>
        <vst:tsc1ControlPurpose>
            <vst:error>ERROR</vst:error>
        </vst:tsc1ControlPurpose>
    </automotiveTorqueSpeedControl1Message>
</vdm:messages>
```

Figure 59. Example Automotive Torque/Speed Control 1 VDM

19.2.3 Automotive System Management Parameters <16001–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the automotive system management parameters. The management parameters below refer to the V1.4 port defined inside the automotive system WSDL. A textual description of the automotive system management parameters is shown below.

NOTE: *The data of interest and data enabled flags control the flow of data on the VICTORY platform. When a type of automotive data has a data of interest flag set to False, the data is not monitored from the automotive bus. As a result, the data cannot be sent to the VDB from the Automotive System Data Interface (regardless of the data enabled flag), nor can the data be retrieved from the Automotive System Management Interface. For all intents and purposes, data with a data of interest flag set to False is considered by VICTORY to not be available on the automotive bus. When a type of automotive data has a data of interest flag set to True, the data is monitored from the automotive bus. The data enabled flag can then be used for controlling whether the data is encapsulated into a VDM and sent to the VDB from the Automotive System Data Interface. Regardless of the data enabled flag, the most recent data received from the automotive bus may be retrieved from the Automotive System Management Interface if the data of interest flag is set to True.*

1. ***Automotive Data***
    a. Identifies the most recent data received from the automotive bus for the specified type of automotive data.
    b. The type of automotive data shall be a string chosen from one of the values inside Table 274.
    c. The following errors are defined for the automotive data parameter:
        i. Retrieving automotive data that is not available from the automotive bus or is not being monitored (data of interest flag set to False) shall result in a NOT_SUPPORTED error.

ii. Retrieving automotive data with a type of automotive data not inside the list of valid types of automotive data may result in an OUT_OF_RANGE error.

d. The contents of the automotive data are described in the section titled "Automotive System Interface Complex Types."

e. Operations include Get().

2. ***Data of Interest***

a. Identifies the automotive data being monitored from the automotive bus.

b. The type of automotive data shall be a string chosen from one of the values inside Table 274.

c. Each type of automotive data shall contain a data of interest flag.

i. A Boolean value of True means the data is monitored and a value of False means it is not monitored.

ii. The data of interest flag for each type of automotive data shall be controlled independently.

d. The following errors are defined for the data of interest parameter:

i. Retrieving or setting the data of interest flag for automotive data that is not available on the automotive bus shall result in a NOT_SUPPORTED error.

ii. Retrieving or setting the data of interest flag with a type of automotive data not inside the list of known types of automotive data may result in an OUT_OF_RANGE error.

e. Operations include Get() and Set().

3. ***Data Enabled***

a. Identifies the automotive data sending VDMs on the VDB.

b. The type of automotive data shall be a string chosen from one of the values inside Table 274.

c. Each type of automotive data shall contain a data enabled flag.

i. A Boolean value of True means "enabled" (sending) and a value of False means "disabled" (not sending).

ii. The data enabled flag for each type of automotive data shall be controlled independently.

d. The following errors are defined for the data enabled parameter:

i. Retrieving the data enabled flag for automotive data that is not available on the automotive bus or is not being monitored from the automotive bus (data of interest flag set to False) shall result in a NOT_SUPPORTED error.

ii. Retrieving or setting the data enabled flag with a type of automotive data not inside the list of known types of automotive data may result in an OUT_OF_RANGE error.

e. Operations include Get() and Set().

4. ***Update Period***

a. Identifies the time between VDM updates for the specified type of automotive data.

b. The type of automotive data shall be a string chosen from one of the values inside Table 274.

c. Each type of automotive data shall contain an update period.

i. The update period shall be in units of milliseconds.

ii. The update period shall be a real value greater than zero.

iii. The update period for each type of automotive data shall be controlled independently.

d. The following errors are defined for the update period parameter:

i. Retrieving or setting the update period for automotive data that is not available on the automotive bus or is not being monitored from the automotive bus (data of interest flag set to False) shall result in a NOT_SUPPORTED error.

ii. Retrieving or setting the update period with a type of automotive data not inside the list of known types of automotive data may result in an OUT_OF_RANGE error.

iii. Setting the update period to a value outside the bounds defined by this specification document may result in an OUT_OF_RANGE error.

iv. Setting the update period to a value less than the minimum update period shall result in a NEAREST_VALUE error.

1. The update period shall be changed to the minimum update period and shall be sent back in the response.

e. Though a very small update period (e.g., an update period of five) is within the range defined by this specification document (greater than zero), the actual update period may be higher depending on the performance of the system. If an update period is set to a lower rate than what is achievable by the system, the system shall send out VDMs as fast as possible.

f. Operations include Get() and Set().

5. ***Minimum Update Period***

a. The maximum frequency at which updates can be sent from the automotive bus.

b. The type of automotive data shall be a string chosen from one of the values inside Table 274.

c. Each type of automotive data shall contain a minimum update period.

i. The minimum update period shall be in units of milliseconds.
ii. The minimum update period shall be a real value greater than zero.

d. The following errors are defined for the minimum update period parameter:
i. Retrieving the minimum update period for automotive data that is not available on the automotive bus or is not being monitored from the automotive bus (data of interest flag set to False) shall result in a NOT_SUPPORTED error.
ii. Retrieving the minimum update period with a type of automotive data not inside the list of known types of automotive data may result in an OUT_OF_RANGE error.

e. Operations include Get().

6. ***Data Transport Configuration***
a. The transport configurations used for sending out automotive VDMs on the VDB.
b. The type of automotive data shall be a string chosen from one of the values inside Table 274.
c. Each type of automotive data shall contain data transport configurations.
i. The contents of data transport configurations are described in the section titled "Data Transport Configuration" under "VICTORY Data Interface Complex Types."
ii. The transport configurations for each type of automotive data shall be controlled independently.
d. Setting the data transport configurations results in the removal of all pre-existing transport configurations and the addition of all transport configurations provided in the request.
e. The following errors are defined for the data transport configuration parameter:
i. Retrieving or setting the data transport configurations for automotive data that is not available on the automotive bus or is not being monitored from the automotive bus (data of interest flag set to False) shall result in a NOT_SUPPORTED error.
ii. Retrieving or setting the data transport configurations with a type of automotive data not inside the list of known types of automotive data may result in an OUT_OF_RANGE error.
iii. Setting multiple data transport configurations with the same unique name shall result in a GENERIC_FAULT error.
iv. Adding or setting a data transport configuration with a URI outside the format defined by this specification document may result in an OUT_OF_RANGE error.
v. Adding a data transport configuration with a pre-existing unique name shall result in a GENERIC_FAULT error.
vi. Removing a data transport configuration with a unique name that does not match the unique name of one of the data transport configurations of the automotive system shall result in a GENERIC_FAULT error.
vii. Removing the only data transport configuration for the automotive system shall result in a GENERIC_FAULT error.
viii. If network layer is included in the request, setting or adding a data transport configuration with a network layer outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
ix. Setting, adding, or removing a data transport configuration with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
x. Setting or adding a data transport configuration with a URI outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.
f. Operations include Get(), Set(), Add(), and Remove().

7. ***Nominal Data Available***
a. Identifies which data parameters from the selected automotive data of interest are available from the automotive bus.
b. Nominal data available shall not be provided for each type of automotive data where the data of interest flag is set to False.
i. Conversely, nominal data available shall be provided for each type of automotive data where the data of interest flag is set to True.
c. For each piece of data provided:
i. A Boolean value of True means the data is available and is being sent from the automotive bus.
ii. A Boolean value of False means the data is not being sent from the automotive bus or that the data is not available from the automotive bus.
d. Operations include Get().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the automotive system management parameters along with a description of additional errors that were not specified but may be returned.*

19.2.4 Automotive System Management Parameters – V1.6 <16005–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the automotive system management parameters. The management parameters below refer to the V1.6 port defined inside the automotive system WSDL. A textual description of the automotive system management parameters is shown below.

NOTE: *The data of interest and data enabled flags control the flow of data on the VICTORY platform. When a type of automotive data has a data of interest flag set to False, the data is not monitored from the automotive bus. As a result, the data cannot be sent to the VDB from the Automotive System Data Interface (regardless of the data enabled flag), nor can the data be retrieved from the Automotive System Management Interface. For all intents and purposes, data with a data of interest flag set to False is considered by VICTORY to not be available on the automotive bus. When a type of automotive data has a data of interest flag set to True, the data is monitored from the automotive bus. The data enabled flag can then be used for controlling whether the data is encapsulated into a VDM and sent to the VDB from the Automotive System Data Interface. Regardless of the data enabled flag, the most recent data received from the automotive bus may be retrieved from the Automotive System Management Interface if the data of interest flag is set to True.*

1. ***Automotive Data 2***
    a. Identifies the most recent data received from the automotive bus for the specified type of automotive data.
    b. The type of automotive data shall be a string chosen from one of the values inside Table 275.
    c. The following errors are defined for the automotive data parameter:
        i. Retrieving automotive data that is not available from the automotive bus or is not being monitored (data of interest flag set to False) shall result in a NOT_SUPPORTED error.
        ii. Retrieving automotive data with a type of automotive data not inside the list of valid types of automotive data may result in an OUT_OF_RANGE error.
    d. The contents of the automotive data are described in the section titled "Automotive System Interface Complex Types."
    e. Operations include Get().
2. ***Data of Interest***
    a. Identifies the automotive data being monitored from the automotive bus.
    b. The type of automotive data shall be a string chosen from one of the values inside Table 275.
    c. Each type of automotive data shall contain a data of interest flag.
        i. A Boolean value of True means the data is monitored and a value of False means it is not monitored.
        ii. The data of interest flag for each type of automotive data shall be controlled independently.
    d. The following errors are defined for the data of interest parameter:
        i. Retrieving or setting the data of interest flag for automotive data that is not available on the automotive bus shall result in a NOT_SUPPORTED error.
        ii. Retrieving or setting the data of interest flag with a type of automotive data not inside the list of known types of automotive data may result in an OUT_OF_RANGE error.
    e. Operations include Get() and Set().
3. ***Data Enabled***
    a. Identifies the automotive data sending VDMs on the VDB.
    b. The type of automotive data shall be a string chosen from one of the values inside Table 275.
    c. Each type of automotive data shall contain a data enabled flag.
        i. A Boolean value of True means "enabled" (sending) and a value of False means "disabled" (not sending).
        ii. The data enabled flag for each type of automotive data shall be controlled independently.
    d. The following errors are defined for the data enabled parameter:
        i. Retrieving the data enabled flag for automotive data that is not available on the automotive bus or is not being monitored from the automotive bus (data of interest flag set to False) shall result in a NOT_SUPPORTED error.
        ii. Retrieving or setting the data enabled flag with a type of automotive data not inside the list of known types of automotive data may result in an OUT_OF_RANGE error.
    e. Operations include Get() and Set().
4. ***Update Period***
    a. Identifies the time between VDM updates for the specified type of automotive data.
    b. The type of automotive data shall be a string chosen from one of the values inside Table 275.
    c. Each type of automotive data shall contain an update period.
        i. The update period shall be in units of milliseconds.

ii. The update period shall be a real value greater than zero.
iii. The update period for each type of automotive data shall be controlled independently.

d. The following errors are defined for the update period parameter:
i. Retrieving or setting the update period for automotive data that is not available on the automotive bus or is not being monitored from the automotive bus (data of interest flag set to False) shall result in a NOT_SUPPORTED error.
ii. Retrieving or setting the update period with a type of automotive data not inside the list of known types of automotive data may result in an OUT_OF_RANGE error.
iii. Setting the update period to a value outside the bounds defined by this specification document may result in an OUT_OF_RANGE error.
iv. Setting the update period to a value less than the minimum update period shall result in a NEAREST_VALUE error.
1. The update period shall be changed to the minimum update period and shall be sent back in the response.

e. Though a very small update period (e.g., an update period of five) is within the range defined by this specification document (greater than zero), the actual update period may be higher depending on the performance of the system. If an update period is set to a lower rate than what is achievable by the system, the system shall send out VDMs as fast as possible.
f. Operations include Get() and Set().

5. ***Minimum Update Period***
a. The maximum frequency at which updates can be sent from the automotive bus.
b. The type of automotive data shall be a string chosen from one of the values inside Table 275.
c. Each type of automotive data shall contain a minimum update period.
i. The minimum update period shall be in units of milliseconds.
ii. The minimum update period shall be a real value greater than zero.
d. The following errors are defined for the minimum update period parameter:
i. Retrieving the minimum update period for automotive data that is not available on the automotive bus or is not being monitored from the automotive bus (data of interest flag set to False) shall result in a NOT_SUPPORTED error.
ii. Retrieving the minimum update period with a type of automotive data not inside the list of known types of automotive data may result in an OUT_OF_RANGE error.
e. Operations include Get().

6. ***Data Transport Configuration***
a. The transport configurations used for sending out automotive VDMs on the VDB.
b. The type of automotive data shall be a string chosen from one of the values inside Table 275.
c. Each type of automotive data shall contain data transport configurations.
i. The contents of data transport configurations are described in the section titled "Data Transport Configuration" under "VICTORY Data Interface Complex Types."
ii. The transport configurations for each type of automotive data shall be controlled independently.
d. Setting the data transport configurations results in the removal of all pre-existing transport configurations and the addition of all transport configurations provided in the request.
e. The following errors are defined for the data transport configuration parameter:
i. Retrieving or setting the data transport configurations for automotive data that is not available on the automotive bus or is not being monitored from the automotive bus (data of interest flag set to False) shall result in a NOT_SUPPORTED error.
ii. Retrieving or setting the data transport configurations with a type of automotive data not inside the list of known types of automotive data may result in an OUT_OF_RANGE error.
iii. Setting multiple data transport configurations with the same unique name shall result in a GENERIC_FAULT error.
iv. Adding or setting a data transport configuration with a URI outside the format defined by this specification document may result in an OUT_OF_RANGE error.
v. Adding a data transport configuration with a pre-existing unique name shall result in a GENERIC_FAULT error.
vi. Removing a data transport configuration with a unique name that does not match the unique name of one of the data transport configurations of the automotive system shall result in a GENERIC_FAULT error.
vii. Removing the only data transport configuration for the automotive system shall result in a GENERIC_FAULT error.

viii. If network layer is included in the request, setting or adding a data transport configuration with a network layer outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
ix. Setting, adding, or removing a data transport configuration with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
x. Setting or adding a data transport configuration with a URI outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.

f. Operations include Get(), Set(), Add(), and Remove().

7. ***Nominal Data Available 2***
   a. Identifies which data parameters from the selected automotive data of interest are available from the automotive bus.
   b. Nominal data available shall not be provided for each type of automotive data where the data of interest flag is set to False.
      i. Conversely, nominal data available shall be provided for each type of automotive data where the data of interest flag is set to True.
   c. For each piece of data provided:
      i. A Boolean value of True means the data is available and is being sent from the automotive bus.
      ii. A Boolean value of False means the data is not being sent from the automotive bus or that the data is not available from the automotive bus.
   d. Operations include Get().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the automotive system management parameters along with a description of additional errors that were not specified but may be returned.*

19.2.5 Automotive System Interface Complex Types <20022–20210409, Pro>

NOTE: *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

NOTE: *The nominal data available management parameter is used for indicating what data is available on the automotive bus.*

19.2.5.1 Active Diagnostic Trouble Codes Complex Type

Table 248. Active Diagnostic Trouble Codes

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Protect Lamp	Status of the engine protect lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	"ERROR" or "NOT_AVAILABLE"	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Amber Warning Lamp	Status of the engine amber warning lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	"ERROR" or "NOT_AVAILABLE"	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Red Stop Lamp	Status of the engine red stop lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
OBD Malfunction Indicator Lamp	Status of the on-board diagnostic (OBD) malfunction indicator lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Flash Protect Lamp	Status of the flash protect lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Flash Amber Warning Lamp	Status of the flash amber warning lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Flash Red Stop Lamp	Status of the flash red stop lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Flash Malfunction Indicator Lamp	Status of the flash malfunction indicator lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Suspect Parameter Number	Identifies the suspect parameter number.	Data	String	Printable ASCII; 1<=length<=256	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Failure Mode Identifier	Identifies the failure mode identifier.	Data	String	Printable ASCII; 1<=length<=256	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	"ERROR" or "NOT_AVAILABLE"	N/A	
Occurrence Count	Identifies the occurrence count.	Data	Integer	0<=x<=126	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	"ERROR" or "NOT_AVAILABLE"	N/A	
SPN Conversion Method	Identifies the method used for converting the suspect parameter number (SPN).	Data	String	Printable ASCII; 1<=length<=256	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	"ERROR" or "NOT_AVAILABLE"	N/A	

19.2.5.2 Alternator Temperature Complex Type

Table 249. Alternator Temperature

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Alternator Bearing 1 Temperature	Temperature of the bearing inside the alternator. Bearing 1 is the left or rear bearing.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Alternator Bearing 2 Temperature	Temperature of the bearing inside the alternator. Bearing 2 is the right or front bearing.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Alternator Winding 1 Temperature	Temperature of the windings inside the alternator.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Alternator Winding 2 Temperature	Temperature of the windings inside the alternator.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Alternator Winding 3 Temperature	Temperature of the windings inside the alternator.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.3 Ambient Conditions Complex Type

Table 250. Ambient Conditions

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Barometric Pressure	Absolute air pressure of the atmosphere.	Data	Real	x>=0	Kilopascals	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Cab Interior Temperature	Temperature of air inside the part of the vehicle that encloses the driver and vehicle operating system.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Ambient Air Temperature	Temperature of air surrounding the vehicle.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Air Intake Temperature	Temperature of air entering vehicle air induction system.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Road Surface Temperature	Temperature of road surface over which vehicle is operating.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.4 Battery Temperature Complex Type

Table 251. Battery Temperature

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Battery 1 Temperature	Temperature of battery 1. The relation to physical location is determined by the equipment manufacturer.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Battery 2 Temperature	Temperature of battery 2. The relation to physical location is determined by the equipment manufacturer.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.5 Cab Message 1 Complex Type

Table 252. Cab Message 1

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Requested Percent Fan Speed	Fan speed as a ratio of the actual fan drive (current speed) to the fully engaged fan drive (maximum fan speed).	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Operator Seat Direction Switch	Indicates whether the operator seat is in the forward driving position.	Data	Boolean	true (facing forward); false (not facing forward)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	"ERROR" or "NOT_AVAILABLE"	N/A	
Seat Belt Switch	Indicates whether the seat belt is buckled.	Data	Boolean	true (buckled); false (not buckled)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	"ERROR" or "NOT_AVAILABLE"	N/A	
Vehicle Limiting Speed Governor Decrement Switch	Indicates whether the speed governor is decreasing.	Data	Boolean	true (decreasing); false (not decreasing)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	"ERROR" or "NOT_AVAILABLE"	N/A	
Vehicle Limiting Speed Governor Increment Switch	Indicates whether the speed governor is increasing.	Data	Boolean	true (increasing); false (not increasing)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	"ERROR" or "NOT_AVAILABLE"	N/A	
Vehicle Limiting Speed Governor Enable Switch	Indicates whether the speed governor is enabled.	Data	Boolean	true (enabled); false (disabled)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	"ERROR" or "NOT_AVAILABLE"	N/A	
Diesel Particulate Filter Regeneration Inhibit Switch	Indicates whether the switch that inhibits diesel particulate filter regeneration is active.	Data	Boolean	true (active); false (not active)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	"ERROR" or "NOT_AVAILABLE"	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Diesel Particulate Filter Regeneration Force Switch	Indicates whether the switch that forces diesel particulate filter regeneration is active.	Data	Boolean	true (active); false (not active)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	
Automatic Gear Shifting Enable Switch	Indicates whether automated gear shifting is enabled.	Data	Boolean	true (enabled); false (disabled)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	
Engine Automatic Start Enable Switch	Indicates whether the idle management system is enabled.	Data	Boolean	true (enabled); false (disabled)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	
Selected Maximum Vehicle Speed Limit	The user-selected maximum speed of the vehicle.	Data	Real	x>=0	Kilometers per hour	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.6 Combination Vehicle Weight Complex Type

Table 253. Combination Vehicle Weight

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Powered Vehicle Weight	Total mass imposed by the tires of the powered vehicle on the road surface. Does not include the trailer.	Data	Integer	x>=0	Kilograms	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Gross Combination Vehicle Weight	The total weight of the truck and all attached trailers.	Data	Integer	x>=0	Kilograms	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.7 Dash Display Complex Type

Table 254. Dash Display

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Washer Fluid Level	Ratio of volume of liquid to total container volume of fluid reservoir in windshield wash system.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Fuel Filter Differential Pressure	Change in fuel delivery pressure, measured across the filter, due to accumulation of solid or semisolid matter on the filter element.	Data	Real	x>=0	Kilopascals	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Oil Filter Differential Pressure	Change in engine oil pressure, measured across the filter, due to the filter and any accumulation of solid or semisolid material on or in the filter.	Data	Real	x>=0	Kilopascals	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Cargo Ambient Temperature	Temperature of air inside vehicle container used to accommodate cargo.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Fuel Level 1	Ratio of volume of fuel to the total volume of the fuel storage container. When Fuel Level 2 is not used, Fuel Level 1 represents the total fuel in all fuel storage containers. When Fuel Level 2 is used, Fuel Level 1 represents the fuel level in the primary or left-side fuel storage container.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Fuel Level 2	Ratio of volume of fuel to the total volume of fuel in the second or right-side storage container. When Fuel Level 2 is not used, Fuel Level 1 represents the total fuel in all fuel storage containers.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.8 Direct Lamp Control Data 1 Complex Type

Table 255. Direct Lamp Control Data 1

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Protect Lamp	Status of the engine protect lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	
Engine Amber Warning Lamp	Status of the engine amber warning lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	
Engine Red Stop Lamp	Status of the engine red stop lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
OBD Malfunction Indicator Lamp	Status of the on-board diagnostic (OBD) malfunction indicator lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Brake Active Lamp	Status of the engine brake active lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Compression Brake Enable Switch Indicator Lamp	Status of the compression brake enable switch indicator lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Oil Pressure Low Lamp	Status of the engine oil pressure low lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Coolant Temperature High Lamp	Status of the engine coolant temperature high lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Coolant Level Low Lamp	Status of the engine coolant level low lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Idle Management Active Lamp	Status of the engine idle management active lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Air Filter Restriction Lamp	Status of the engine air filter restriction lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Fuel Filter Restricted Lamp	Status of the engine fuel filter restricted lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Wait to Start Lamp	Status of the engine wait to start lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Control Module 1 Ready for Use Lamp	Status of the engine control module 1 ready for use lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Control Module 2 Ready for Use Lamp	Status of the engine control module 2 ready for use lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Control Module 3 Ready for Use Lamp	Status of the engine control module 3 ready for use lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.

19.2.5.9 Direct Lamp Control Data 2 Complex Type

Table 256. Direct Lamp Control Data 2

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Vehicle Battery Voltage Low Lamp	Status of the vehicle battery voltage low lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	
Vehicle Fuel Level Low Lamp	Status of the vehicle fuel level low lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	
Vehicle Air Pressure Low Lamp	Status of the vehicle air pressure low lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	
Vehicle HVAC Recirculation Lamp	Status of the vehicle heating, ventilation, and air conditioning (HVAC) recirculation lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	
Vehicle Battery Charging Lamp	Status of the vehicle battery charging lamp.	Data	Boolean	true (On); false (Off)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	

19.2.5.10 Distance Complex Type

Table 257. Distance

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Trip Distance	Distance traveled during all or part of a journey.	Data	Real	x>=0	Kilometers	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Total Vehicle Distance	Accumulated distance traveled by vehicle during its operation.	Data	Real	x>=0	Kilometers	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.11 Electrical Power 1 Complex Type

Table 258. Electrical Power 1

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Net Battery Current	Net flow of electrical current into/out of the battery or batteries.	Data	Real	N/A	Amps	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Alternator Current	Measure of electrical current flow from the alternator.	Data	Real	N/A	Amps	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Charging System Potential	Electrical potential measured at the charging system output. The charging system may be any device charging the batteries. This includes alternators, generators, solid state charger, and other charging devices.	Data	Real	x>=0	Volts	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Power Input 1	Measures the first source of battery potential as measured at the input of the ECM/actuator coming from one or more batteries, irrespective of the distance between the component and the battery. This is also used when ECMs are interconnected in a series configuration, where the source of power is coming directly or indirectly from the same battery/batteries.	Data	Real	x>=0	Volts	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Keyswitch Battery Potential	Battery potential measured at the input of the electronic control unit supplied through a keyswitch or similar switching device.	Data	Real	x>=0	Volts	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.12 Electrical Power 2 Complex Type

Table 259. Electrical Power 2

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Power Input 2	Measures the second source of battery potential as measured at the input of the ECM/actuator coming from one or more batteries, irrespective of the distance between the component and the battery. This is also used when ECMs are interconnected in a series configuration, where the source of power is coming directly or indirectly from the same battery/ batteries.	Data	Real	x>=0	Volts	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
ECU Power Output Supply Voltage 1	The first power output from an ECU.	Data	Real	x>=0	Volts	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
ECU Power Output Supply Voltage 2	The second power output from an ECU.	Data	Real	x>=0	Volts	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
ECU Power Output Supply Voltage 3	The third power output from an ECU.	Data	Real	x>=0	Volts	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.13 Electronic Engine Controller 1 Complex Type

Table 260. Electronic Engine Controller 1

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Torque Mode	Indicates which engine torque mode is currently generating, limiting, or controlling the torque.	Data	String	Valid Engine Torque Modes	N/A	Required
Driver's Demand Engine - Percent Torque	The requested torque output of the engine by the driver.	Data	Real	x	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Actual Engine - Percent Torque	The calculated output torque of the engine.	Data	Real	x	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Speed	Actual engine speed, which is calculated over a minimum crankshaft angle of 720 degrees divided by the number of cylinders.	Data	Real	x>=0	Revolutions per minute	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Starter Mode	There are several phases in a starting action and different reasons why a start cannot take place.	Data	String	Valid Engine Starter Modes	N/A	Required

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Demand - Percent Torque	The requested torque output of the engine by all dynamic internal inputs, including smoke control, noise control, and low and high speed governing.	Data	Real	x	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.14 Electronic Engine Controller 2 Complex Type

Table 261. Electronic Engine Controller 2

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Accelerator Pedal 1 Low Idle Switch	Indicates whether the accelerator pedal 1 is low.	Data	Boolean	true (low); false (not low)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	
Accelerator Pedal Kickdown Switch	Indicates whether the accelerator pedal kickdown is opened or closed.	Data	Boolean	true (open/active); false (closed/passive)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	
Road Speed Limit Status	Indicates whether the system used to limit maximum vehicle velocity is active.	Data	Boolean	true (active); false (not active)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Accelerator Pedal 2 Low Idle Switch	Indicates whether the accelerator pedal 2 is low.	Data	Boolean	true (low); false (not low)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	
Accelerator Pedal Position 1	The ratio of actual position of the analog engine speed/torque request input device (such as an accelerator pedal or throttle lever) to the maximum position of the input device.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Percent Load at Current Speed	The ratio of actual engine percent torque (indicated) to maximum indicated torque available at the current engine speed, clipped to zero torque during engine braking.	Data	Real	x>=0	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Remote Accelerator Pedal Position	The ratio of actual position of the remote analog engine speed/ torque request input device (such as an accelerator pedal or throttle lever) to the maximum position of the input device.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Accelerator Pedal Position 2	The ratio of actual position of the second analog engine speed/ torque request input device (such as an accelerator pedal or throttle lever) to the maximum position of the input device.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Vehicle Acceleration Rate Limit Status	Indicates whether the system used to limit maximum forward vehicle acceleration is active.	Data	Boolean	true (active); false (not active)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_AVAILABLE”	N/A	
Momentary Engine Maximum Power Enable Feedback	Indicates whether maximum engine power available from the vehicle is being momentarily requested.	Data	Boolean	true (requesting maximum power); false (not requesting maxiumum power)	N/A	If available, data or error (fault) is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“UNDEFINED_PARAMETER” or “NOT_SPECIFIED”	N/A	
DPF Thermal Management Active	Indicates whether the exhaust temperatures have been elevated for regeneration of the diesel particulate filter (DPF) aftertreatment system or in preparation of regeneration of the diesel particulate filter aftertreatment system.	Data	Boolean	true (active); false (not active)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“UNDEFINED_PARAMETER” or “NOT_SPECIFIED”	N/A	
SCR Thermal Management Active	Indicates whether the exhaust temperatures have been elevated for regeneration of the selective catalytic reduction (SCR) aftertreatment system or in preparation of regeneration of the SCR aftertreatment system.	Data	Boolean	true (active); false (not active)	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“UNDEFINED_PARAMETER” or “NOT_SPECIFIED”	N/A	
Actual Maximum Available Engine - Percent Torque	The maximum amount of torque that the engine can immediately deliver as a percentage of the reference engine torque.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Estimated Pumping - Percent Torque	The calculated torque that indicates the estimated amount of torque loss due to the engine air handling system.	Data	Real	x	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.15 Electronic Transmission Controller 2 Complex Type

Table 262. Electronic Transmission Controller 2

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Transmission Current Gear	The gear currently engaged in the transmission or the last gear engaged while the transmission is in the process of shifting to the new or selected gear. Transitions toward a destination gear will not be indicated.	Data	Integer	x	Gears	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Transmission Selected Gear	The gear that the transmission will attempt to achieve during the current shift, if a shift is in progress, or the next shift, if one is pending.	Data	Integer	x	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Transmission Actual Gear Ratio	Actual ratio of input shaft speed to output shaft speed.	Data	Real	x	Gears	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Transmission Requested Range	Range selected by the operator.	Data	String	Valid Transmission Range Values	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	"ERROR" or "NOT_AVAILABLE"	N/A	
Transmission Current Range	Range currently commanded by the transmission control system.	Data	String	Valid Transmission Range Values	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	"ERROR" or "NOT_AVAILABLE"	N/A	

19.2.5.16 Engine Fluid Level and Pressure 1 Complex Type

Table 263. Engine Fluid Level and Pressure 1

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Fuel Delivery Pressure	Gauge pressure of fuel in system as delivered from the supply pump to the injection pump.	Data	Real	x>=0	Kilopascals	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Extended Crankcase Blowby Pressure	Differential crankcase blow-by pressure as measured through a tube with a venturi.	Data	Real	x>=0	Kilopascals	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Oil Level	Ratio of current volume of engine sump oil to maximum required volume.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Oil Pressure	Gauge pressure of oil in the engine lubrication system as provided by the oil pump.	Data	Real	x>=0	Kilopascals	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Crankcase Pressure	Gauge pressure inside the engine crankcase.	Data	Real	x	Kilopascals	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Coolant Pressure	Gauge pressure of liquid found in the engine cooling system.	Data	Real	x>=0	Kilopascals	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Coolant Level	Ratio of liquid found in the engine cooling system to total cooling system volume.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.17 Engine Hours and Revolutions Complex Type

Table 264. Engine Hours and Revolutions

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Hours of Operation	Accumulated time of operation of engine.	Data	Real	x>=0	Hours	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Revolutions	Accumulated number of revolutions of engine crankshaft during its operation.	Data	Integer	x>=0	Revolutions	If available, data or error is required.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.18 Engine Speed and Load Factor Complex Type

Table 265. Engine Speed and Load Factor

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Trip Maximum Engine Speed	Maximum engine speed achieved since the last trip reset.	Data	Real	x>=0	Revolutions per minute	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Trip Average Engine Speed	Average speed of the engine since the last trip reset.	Data	Real	x>=0	Revolutions per minute	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Trip Drive Average Load Factor	Average engine load factor while engine speed is greater than zero, vehicle speed is greater than or equal to 2 km/h, and both the power take-off (PTO) and remote PTO governors are not in the hold state, since the last trip reset. Engine operation during cruise control operation is included.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Total Drive Average Load Factor	Average engine load factor while engine speed is greater than zero, vehicle speed is greater than or equal to 2 km/h, and both the PTO and remote PTO governors are not in the hold state, over the life of the engine. Engine operation during cruise control operation is included.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Total Engine Cruise Time	Total time that the engine has operated in the cruise hold state, excluding time in accelerator override, over the life of the engine.	Data	Real	x>=0	Hours	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.19 Engine Temperature 1 Complex Type

Table 266. Engine Temperature 1

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Coolant Temperature	Temperature of liquid found in the engine cooling system.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Fuel Temperature 1	Temperature of fuel (or gas) passing through the first fuel control system.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Oil Temperature 1	Temperature of the engine lubricant.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Turbocharger Oil Temperature 1	Temperature of the turbocharger lubricant.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Intercooler Temperature 1	Temperature of the liquid found in the intercooler located after the turbocharger.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Intercooler Thermostat Opening	The current position of the thermostat used to regulate the temperature of the engine intercooler. A value of 0% represents the thermostat being completely closed and 100% represents the thermostat being completely open.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.20 Engine Temperature 2 Complex Type

Table 267. Engine Temperature 2

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Oil Temperature 2	Temperature of the engine lubricant.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine ECU Temperature	Temperature of the engine electronic control unit (ECU).	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Exhaust Gas Recirculation 1 Differential Pressure	Differential pressure across the exhaust gas recirculation (EGR) system.	Data	Real	x	Kilopascals	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Exhaust Gas Recirculation 1 Temperature	Temperature of the recirculated exhaust gas.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.21 Engine Valve Complex Type

Table 268. Engine Valve

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Fuel Valve 1 Failure Mode	Identifies the failure mode, if applicable, detected in engine fuel valve 1.	Data	String	Valid Failure Mode Values	N/A	If available, data is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Fuel Valve 1 Temperature Status	Identifies the status of engine fuel valve 1 drive circuitry temperature.	Data	String	Valid Range Severity Values	N/A	If available, data is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Fuel Valve 1 Operation Status	Identifies the status of engine fuel valve 1.	Data	String	Valid Valve Operation Status Values	N/A	If available, data is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Desired Throttle Valve 1 Position	The desired position of throttle valve 1 that is regulating the fluid, usual air/fuel mixture to the engine, as commanded by the Engine Control Unit. 0% represents no supply and 100% represents full supply.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Throttle Valve 1 Failure Mode	Identifies the failure mode, if applicable, detected in engine throttle valve 1.	Data	String	Valid Failure Mode Values	N/A	If available, data is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Throttle Valve 1 Temperature Status	Identifies the status of engine throttle valve 1 drive circuitry temperature.	Data	String	Valid Range Severity Values	N/A	If available, data is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Throttle Valve 1 Operation Status	Identifies the status of engine throttle valve 1.	Data	String	Valid Valve Operation Status Values	N/A	If available, data is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Desired Throttle Valve 2 Position	The desired position of throttle valve 2 that is regulating the fluid, usual air/fuel mixture to the engine, as commanded by the Engine Control Unit. 0% represents no supply and 100% represents full supply.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Throttle Valve 2 Failure Mode	Identifies the failure mode, if applicable, detected in engine throttle valve 2.	Data	String	Valid Failure Mode Values	N/A	If available, data is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Throttle Valve 2 Temperature Status	Identifies the status of engine throttle valve 2 drive circuitry temperature.	Data	String	Valid Range Severity Values	N/A	If available, data is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Throttle Valve 2 Operation Status	Identifies the status of engine throttle valve 2.	Data	String	Valid Valve Operation Status Values	N/A	If available, data is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Fuel Valve 2 Failure Mode	Identifies the failure mode, if applicable, detected in engine fuel valve 2.	Data	String	Valid Failure Mode Values	N/A	If available, data is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Fuel Valve 2 Temperature Status	Identifies the status of engine fuel valve 2 drive circuitry temperature.	Data	String	Valid Range Severity Values	N/A	If available, data is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
Engine Fuel Valve 2 Operation Status	Identifies the status of engine fuel valve 2.	Data	String	Valid Valve Operation Status Values	N/A	If available, data is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.

19.2.5.22 Fan Drive 1 Complex Type

Table 269. Fan Drive 1

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Estimated Percent Fan Speed	Estimated fan speed as a ratio of the fan drive (current speed) to the fully engaged fan drive (maximum fan speed).	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Fan Drive State	State, or mode of operation, of the fan drive.	N/A	String	Valid Fan Drive States	N/A	Required
Fan Speed	The speed of the fan associated with the engine coolant system.	Data	Real	x>=0	Revolutions per minute	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Hydraulic Fan Motor Pressure	The hydraulic pressure used to drive the fan system, sensed before the hydraulic fan motor.	Data	Real	x>=0	Kilopascals	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Fan Drive Bypass Command Status	Status of the Fan Drive Bypass Command as being commanded by the Engine Control Unit. 0% indicates no bypass, whereas 100% is fully bypassed.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.23 Fuel Consumption Complex Type

Table 270. Fuel Consumption

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Trip Fuel	Fuel consumed during all or part of a journey.	Data	Real	x>=0	Liters	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Total Fuel Used	Accumulated amount of fuel used during vehicle operation.	Data	Real	x>=0	Liters	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.24 Fuel Economy Complex Type

Table 271. Fuel Economy

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Fuel Rate	Amount of fuel consumed by the engine per unit of time.	Data	Real	x>=0	Liters per hour	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Instantaneous Fuel Economy	Current fuel economy at current vehicle velocity.	Data	Real	x>=0	Kilometers per liter	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“INFINITE”, “UNDEFINED_PARAMETER”, “ERROR”, or “NOT_AVAILABLE”	N/A	
Engine Average Fuel Economy	Average of instantaneous fuel economy for that segment of vehicle operation of interest.	Data	Real	x>=0	Kilometers per liter	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Throttle Valve 1 Position	The position of the valve used to regulate the supply of a fluid, usually air or fuel/air mixture, to an engine. 0% represents no supply and 100% is full supply.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Engine Throttle Valve 2 Position	The sensed position feedback of the valve, coming from a second electrical actuator for a second throttle plate, used to regulate the supply of a fluid, usually air or fuel/air mixture. 0% represents no supply and 100% is full supply.	Data	Real	0<=x<=100	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

19.2.5.25 Tire Condition Complex Type

Table 272. Tire Condition

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Tire Location	Identifies which tire is associated with the parametric data in this message.	Data	String	Printable ASCII; 1<=length<=256	N/A	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Printable ASCII; 1<=length<=256	N/A	
Tire Pressure	Pressure at which air is contained in the cavity formed by the tire and rim.	Data	Real	x>=0	Kilopascals	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Tire Temperature	Temperature at the surface of the tire sidewall.	Data	Real	x>= –273.15	Degrees Celsius	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
CTI Wheel Sensor Status	Indicates whether the wheel is being monitored by the central tire inflation (CTI) controller.	Data	Boolean	true (monitored); false (not monitored)	N/A	If supported, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“NOT_DEFINED” or “NOT_SUPPORTED”	N/A	
CTI Tire Status	Indicates whether there is a leak detected in the tire.	Data	Boolean	true (leak detected); false (leak not detected)	N/A	If supported, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	“ERROR” or “NOT_SUPPORTED”	N/A	
CTI Wheel End Electrical Fault	Indicates whether there is an electrical fault on the CTI wheel interface.	Data	Boolean	true (fault); false (no fault)	N/A	If supported, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Printable ASCII; 1<=length<=256	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Tire Air Leakage Rate	Pressure loss rate of a tire.	Data	Real	x>=0	Pascals per second	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
Tire Pressure Threshold Detection	The pressure level of the tire.	N/A	String	Valid Tire Pressure Threshold Detections	N/A	Required

19.2.5.26 Torque/Speed Control 1 Complex Type

Table 273. Torque/Speed Control 1

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Override Control Mode	The override control mode.	Data	String	Valid Engine Override Control Modes	N/A	Required
Engine Requested Speed Control Conditions	The governor characteristics that are desired during speed control.	Data	String	Valid Engine Requested Speed Control Conditions	N/A	Required
Override Control Mode Priority	The priority of the override control mode.	Data	String	Valid Override Control Mode Priorities	N/A	Required
Engine Requested Speed/ Speed Limit	The engine speed at which the engine is expected to operate if the speed control mode is active or the engine speed which the engine is not expected to exceed if the speed limit mode is active.	Data	Real	x>=0	Revolutions per minute	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	

Name	Description	Data/ Error	Data Type	Range	Units	Applicability
Engine Requested Torque/ Torque Limit	The engine torque at which the engine is expected to operate if the torque control mode is active or the engine torque which the engine is not expected to exceed if the torque limit mode is active.	Data	Real	x	Percent	If available, data or error is required. If not available, the parameter is not present in the complex type or contains the error NOT_AVAILABLE.
		Error	String	Valid Ranged Data Error Values	N/A	
TSC1 Transmission Rate	The transmission rate at which the sending device will transmit the torque/speed control 1 (TSC1) message.	Data	String	Valid TSC1 Transmission Rates	N/A	Required
TSC1 Control Purpose	Indicates which control mode the sending device is using to generate the TSC1 command.	Data	String	Valid TSC1 Control Purposes	N/A	Required

19.2.6 Automotive System Interface Simple Types <20066–20150331, Pro>

19.2.6.1 Automotive Data Types Simple Type

Table 274. Valid Automotive Data Types

Value
AMBIENT_CONDITIONS
BATTERY_TEMPERATURE
DIRECT_LAMP_CONTROL_DATA_1
DIRECT_LAMP_CONTROL_DATA_2
DISTANCE
ELECTRICAL_POWER_1
ELECTRICAL_POWER_2
ELECTRONIC_TRANSMISSION_CONTROLLER_2
ENGINE_FLUID_LEVEL_AND_PRESSURE_1
ENGINE_HOURS_AND_REVOLUTIONS
ENGINE_SPEED_AND_LOAD_FACTOR
ENGINE_TEMPERATURE_1
ENGINE_TEMPERATURE_2
ENGINE_VALVE
FUEL_CONSUMPTION

Value
FUEL_ECONOMY

19.2.6.2 Automotive Data Types With V1.6 Extension Simple Type

Table 275. Valid Automotive Data Types With V1.6 Extensions

Value
ACTIVE_DIAGNOSTIC_TROUBLE_CODES
ALTERNATOR_TEMPERATURE
AMBIENT_CONDITIONS
BATTERY_TEMPERATURE
CAB_MESSAGE_1
COMBINATION_VEHICLE_WEIGHT
DASH_DISPLAY
DIRECT_LAMP_CONTROL_DATA_1
DIRECT_LAMP_CONTROL_DATA_2
DISTANCE
ELECTRICAL_POWER_1
ELECTRICAL_POWER_2
ELECTRONIC_ENGINE_CONTROLLER_1
ELECTRONIC_ENGINE_CONTROLLER_2
ELECTRONIC_TRANSMISSION_CONTROLLER_2
ENGINE_FLUID_LEVEL_AND_PRESSURE_1
ENGINE_HOURS_AND_REVOLUTIONS
ENGINE_SPEED_AND_LOAD_FACTOR
ENGINE_TEMPERATURE_1
ENGINE_TEMPERATURE_2
ENGINE_VALVE
FAN_DRIVE_1
FUEL_CONSUMPTION
FUEL_ECONOMY
TIRE_CONDITION
TORQUE_SPEED_CONTROL_1

19.2.6.3 TSC1 Control Purposes Simple Type

Table 276. Valid TSC1 Control Purposes

Value	Additional Requirements
ACCELERATOR_PEDAL	
CRUISE_CONTROL	
PTO_GOVERNOR	
ROAD_SPEED_GOVERNOR	
ENGINE_PROTECTION	
TEMPORARY_POWER_TRAIN_CONTROL	
UNDEFINED_PARAMETER	
NOT_AVAILABLE	Valid when the automotive bus is not sending any data for the parameter, or if the automotive bus is sending a value that means ‘not available’.

19.2.6.4 TSC1 Transmission Rates Simple Type

Table 277. Valid TSC1 Transmission Rates

Value	Additional Requirements
1000_MS	
750_MS	
500_MS	
250_MS	
100_MS	
50_MS	
20_MS	
STANDARD_10_MS	
NOT_AVAILABLE	Valid when the automotive bus is not sending any data for the parameter, or if the automotive bus is sending a value that means ‘not available’.

19.2.6.5 Override Control Mode Priorities Simple Type

Table 278. Valid Override Control Mode Priorities

Value	Additional Requirements
HIGHEST	
HIGH	
MEDIUM	

Value	Additional Requirements
LOW	
NOT_AVAILABLE	Valid when the automotive bus is not sending any data for the parameter, or if the automotive bus is sending a value that means ‘not available’.

19.2.6.6 Engine Requested Speed Control Conditions Simple Type

Table 279. Valid Engine Requested Speed Control Conditions

Value	Additional Requirements
TRANSIENT_OPTIMIZED_DRIVELINE_DISENGAGED_NON_LOCKUP_CONDITIONS	
STABILITY_OPTIMIZED_DRIVELINE_DISENGAGED_NON_LOCKUP_CONDITIONS	
STABILITY_OPTIMIZED_DRIVELINE_ENGAGED_LOCKUP_CONDITION_1	
STABILITY_OPTIMIZED_DRIVELINE_ENGAGED_LOCKUP_CONDITION_2	
NOT_AVAILABLE	Valid when the automotive bus is not sending any data for the parameter, or if the automotive bus is sending a value that means ‘not available’.

19.2.6.7 Engine Override Control Modes Simple Type

Table 280. Valid Engine Override Control Modes

Value	Additional Requirements
OVERRIDE_DISABLED	
SPEED_CONTROL	
TORQUE_CONTROL	
SPEED_TORQUE_LIMIT_CONTROL	
NOT_AVAILABLE	Valid when the automotive bus is not sending any data for the parameter, or if the automotive bus is sending a value that means ‘not available’.

19.2.6.8 Tire Pressure Threshold Detections Simple Type

Table 281. Valid Tire Pressure Threshold Detections

Value	Additional Requirements
EXTREME_OVER_PRESSURE	Valid only when tire pressure threshold detection data is available from the automotive bus.
OVER_PRESSURE	
NO_WARNING_PRESSURE	
UNDER_PRESSURE	
EXTREME_UNDER_PRESSURE	

Value	Additional Requirements
NOT_DEFINED	
ERROR	
NOT_AVAILABLE	Valid when the automotive bus is not sending any data for the parameter, or if the automotive bus is sending a value that means ‘not available’.

19.2.6.9 Engine Starter Modes Simple Type

Table 282. Valid Engine Starter Modes

Value	Additional Requirements
START_NOT_REQUESTED	Valid only when engine starter mode data is available from the automotive bus.
GEAR_NOT_ENGAGED	
GEAR_ENGAGED	
START_FINISHED	
STARTER_INHIBITED_ENGINE_ALREADY_RUNNING	
STARTER_INHIBITED_ENGINE_NOT_READY_FOR_START	
STARTER_INHIBITED_DRIVELINE_ENGAGED	
STARTER_INHIBITED_ACTIVE_IMMOBILIZER	
STARTER_INHIBITED_DUE_TO_STARTER_OVER_TEMP	
STARTER_INHIBITED_DUE_TO_INTAKE_AIR_SHUTOFF_VALVE_BEING_ACTIVE	
STARTER_INHIBITED_UNKNOWN_REASON	
ERROR	
UNDEFINED_PARAMETER	
NOT_AVAILABLE	Valid when the automotive bus is not sending any data for the parameter, or if the automotive bus is sending a value that means ‘not available’.

19.2.6.10 Engine Torque Modes Simple Type

Table 283. Valid Engine Torque Modes

Value	Additional Requirements
LOW_IDLE_GOVERNOR/ NO_REQUEST	Valid only when engine torque mode data is available from the automotive bus.
ACCELERATOR_PEDAL/ OPERATOR_SELECTION	

Value	Additional Requirements
CRUISE_CONTROL	
PTO_GOVERNOR	
ROAD_SPEED_GOVERNOR	
ASR_CONTROL	
TRANSMISSION_CONTROL	
ABS_CONTROL	
TORQUE_LIMITING	
HIGH_SPEED_GOVERNOR	
BRAKING_SYSTEM	
REMOTE_ACCELERATOR	
SERVICE_PROCEDURE	
NOT_DEFINED	
OTHER	
NOT_AVAILABLE	Valid when the automotive bus is not sending any data for the parameter, or if the automotive bus is sending a value that means ‘not available’.

19.2.6.11 Fan Drive States Simple Type

Table 284. Valid Fan Drive States

Value	Additional Requirements
FAN_OFF	Valid only when fan drive state data is available from the automotive bus.
ENGINE_SYSTEM	
EXCESSIVE_ENGINE_AIR_TEMP	
EXCESSIVE_ENGINE_OIL_TEMP	
EXCESSIVE_ENGINE_COOLANT_TEMP	
EXCESSIVE_TRANSMISSION_OIL_TEMP	
EXCESSIVE_HYDRAULIC_OIL_TEMP	
DEFAULT_OPERATION	
REVERSE_OPERATION	
MANUAL_CONTROL	
TRANSMISSION_RETARDER	
AC_SYSTEM	
TIMER	

Value	Additional Requirements
ENGINE_BRAKE	
OTHER	
NOT_AVAILABLE	Valid when the automotive bus is not sending any data for the parameter, or if the automotive bus is sending a value that means 'not available'.

19.2.6.12 Failure Mode Values Simple Type

Table 285. Valid Failure Mode Values

Value	Note
DATA_VALID_BUT_ABOVE_NORMAL_OPERATIONAL_RANGE_MOST_SEVERE_LEVEL	
DATA_VALID_BUT_BELOW_NORMAL_OPERATIONAL_RANGE_MOST_SEVERE_LEVEL	
DATA_ERRATIC_INTERMITTENT_OR_INCORRECT	
VOLTAGE_ABOVE_NORMAL_OR_SHORTED_TO_HIGH_SOURCE	
VOLTAGE_BELOW_NORMAL_OR_SHORTED_TO_LOW_SOURCE	
CURRENT_BELOW_NORMAL_OR_OPEN_CIRCUIT	
CURRENT_ABOVE_NORMAL_OR_GROUNDED_CIRCUIT	
MECHANICAL_SYSTEM_NOT_RESPONDING_OR_OUT_OF_ADJUSTMENT	
ABNORMAL_FREQUENCY_OR_PULSE_WIDTH_OR_PERIOD	
ABNORMAL_UPDATE_RATE	
ABNORMAL_RATE_OF_CHANGE	
ROOT_CAUSE_NOT_KNOWN	
BAD_INTELLIGENT_DEVICE_OR_COMPONENT	
OUT_OF_CALIBRATION	
SPECIAL_INSTRUCTIONS	
DATA_VALID_BUT_ABOVE_NORMAL_OPERATING_RANGE_LEAST_SEVERE_LEVEL	
DATA_VALID_BUT_ABOVE_NORMAL_OPERATING_RANGE_MODERATELY_SEVERE_LEVEL	
DATA_VALID_BUT_BELOW_NORMAL_OPERATING_RANGE_LEAST_SEVERE_LEVEL	
DATA_VALID_BUT_BELOW_NORMAL_OPERATING_RANGE_MODERATELY_SEVERE_LEVEL	
RECEIVED_NETWORK_DATA_IN_ERROR	
DATA_DRIFTED_HIGH	
DATA_DRIFTED_LOW	
CONDITION_EXISTS	May represent no failure.
UNDEFINED_PARAMETER	

Value	Note
RESERVED	

19.2.6.13 Range Severity Values Simple Type

Table 286. Valid Range Severity Values

Value	Additional Requirements
HIGH_MOST_SEVERE	
HIGH_LEAST_SEVERE	
IN_RANGE	
LOW_LEAST_SEVERE	
LOW_MOST_SEVERE	
NOT_DEFINED	
ERROR	
NOT_AVAILABLE	Valid when the automotive bus is not sending any data for the parameter, or if the automotive bus is sending a value that means ‘not available’.

19.2.6.14 Valve Operation Status Simple Type

Table 287. Valid Valve Operation Status Values

Value	Note
NORMAL	
ALARM	Fully operational, but needs service
ALARM_HIGH_SEVERITY	Functional, but transient performance may be reduced
DERATE_ACTIVE	Torque output reduced due to environmental conditions
CONTROLLED_SHUTDOWN_ACTIVE	Driving to the default position - usually this means closed
UNCONTROLLED_SHUTDOWN_ACTIVE	Actuator current is off so it is limp
UNDEFINED_PARAMETER	
NOT_AVAILABLE	Valid when the automotive bus is not sending any data for the parameter, or if the automotive bus is sending a value that means ‘not available’.
ERROR	

19.2.6.15 Ranged Data Error Values Simple Type

Table 288. Valid Ranged Data Error Values

Value	Additional Requirements
PARAMETER_SPECIFIC_INDICATOR	
UNDEFINED_PARAMETER	
ERROR	
NOT_AVAILABLE	Valid when the automotive bus is not sending any data for the parameter, or if the automotive bus is sending a value that means 'not available'.

19.2.6.16 Transmission Range Values Simple Type

Table 289. Valid Transmission Range Values

Value	Note
PARK	
REVERSE	"REVERSE" may contain a numerical modifier. The numerical modifier shall be preceded by an '_' e.g., REVERSE_1.
DRIVE	"DRIVE" may contain a numerical modifier. The numerical modifier shall be preceded by an '_' e.g., DRIVE_1.
NEUTRAL	"NEUTRAL" may contain a numerical modifier. The numerical modifier shall be preceded by an '_' e.g., NEUTRAL_1.
LOW	"LOW" may contain a numerical modifier. The numerical modifier shall be preceded by an '_' e.g., LOW_1.
<string representation of an integer value>	Examples include: 1, 2, and 3.

19.3 Lethality Systems Interfaces

Lethality Systems (weapons) are often implemented on vehicle platforms for the purpose of engaging the enemy. While these systems are clearly outside the purview of the VICTORY architecture, it is nonetheless beneficial to have a VDB interface for communicating management information to the weapons systems. This interface will necessarily be outside of the fire loop of the system. These weapons systems will benefit greatly from having an interface with C4ISR/EW systems and other VDB components, as they will be able to share data regarding the location and nature of threats.

19.3.1 Remote Weapon Station Interfaces

A Remote Weapon Station (RWS) is a weapons system that can be controlled remotely by the warfighter. Current RWS systems include computing systems, display technologies, and weapons components as part of the full system. VICTORY provides a standard interface between the RWS system and the VDB, and as such, the RWS system requires a physical VDB interface.

NOTE: *The RWS interface is only specified for the interface between the RWS and the VDB. The VICTORY specification does not refer to the interfaces between components that comprise an RWS system.*

VICTORY may also be used to facilitate the transport of data from other C4ISR/EW systems (or VICTORY components) to the RWS system. For example, Threat Detection and Reporting data messages could be used to create targets based upon recently detected shots. These targets could then be made available to the RWS for the purpose of automatically slewing the weapons system to the threat.

19.3.2 RWS Data Specification <20019–20210409, Pro>

19.3.2.1 Remote Weapon Station Message Content

The Remote Weapon Station Data Interface provides information concerning the pointing direction of the RWS. This information will be available to all VDB network nodes that require RWS pointing direction data.

The Remote Weapon Station Data Interface shall provide the following:

1. ***RWS Pointing Direction***
    a. **Azimuth**
        i. A detailed description of the azimuth data is provided in the section titled "Azimuth" under "VICTORY Data Types."
        ii. Azimuth shall be provided inside the pointing direction portion of an RWS VDM.
    b. **Elevation**
        i. A detailed description of the elevation data is provided in the section titled "Elevation" under "VICTORY Data Types."
        ii. Elevation shall be provided inside the pointing direction portion of an RWS VDM.
2. ***Timestamp***
    a. A timestamp corresponding to the time at which the RWS data was sampled.
    b. The timestamp shall be in the format as described in the section titled "Timestamp Format for Application Layer Data."
3. ***Sequence Number***
    a. A monotonically increasing number that indicates the sequence number of the Remote Weapon Station message sent from a particular Remote Weapon Station Data Interface.
    b. The sequence number shall increase by one (1) for each Remote Weapon Station message sent from the interface.
    c. When the sequence number reaches its maximum value, it shall roll over to its minimum value.

    NOTE: *The sequence number is intended to be used to detect messages that did not arrive or were dropped.*
4. ***Interface ID***
    a. An identification number, unique within the scope of the VDB, used to identify a particular Remote Weapon Station Data Interface.
    b. The interface ID shall be an integer value greater than or equal to zero.

19.3.2.2 Remote Weapon Station Message Encoding

The information described in the previous section shall be encoded into an XML string formatted as a Remote Weapon Station VDM. The appendix titled "VICTORY Schemas" provides the name of the schema file. The file describes the Remote Weapon Station VDM, provides additional details for the VICTORY Data Types Schema, and describes the RWS type that generates the XML string used by the Remote Weapon Station VDM. An example Remote Weapon Station VDM is provided below.

```
<vdm:messages xmlns:vdm="http://www.victory-standards.org/Schemas/VDM.xsd"
    xmlns:vdt="http://www.victory-standards.org/Schemas/VICTORYDataTypes.xsd">
    <remoteWeaponsStationMessage interfaceId="1"
        timestamp="2011-01-01T12:00:00.000000000Z"
        sequenceNumber="14">
        <vdt:pointingDirection>
            <vdt:azimuth>21.25</vdt:azimuth>
            <vdt:elevation>2.37</vdt:elevation>
        </vdt:pointingDirection>
    </remoteWeaponsStationMessage>
</vdm:messages>
```

Figure 60. Example Remote Weapon Station VDM

19.3.3 RWS Device Management Parameters <16002–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the RWS device management parameters. A textual description of the RWS device management parameters is shown below.

1. ***Target List***
    a. This parameter identifies a list of targets known by the RWS interface.
    b. The contents of a target are described in the section titled "Target" under "RWS Interface Complex Types."
    c. Operations include Get().
2. ***Target***
    a. This parameter handles targets known by the RWS interface.
    b. A target is selected, updated, or removed by using the unique identifier of the target.
    c. The contents of a target are described in the section titled "Target" under "RWS Interface Complex Types."
    d. The following errors are defined for the target parameter:
        i. Adding or updating a target with a relative position, absolute position, priority, or confidence outside the range defined by this specification document may result in an OUT_OF_RANGE error.
        ii. Adding a target with a pre-existing target identifier shall result in a GENERIC_FAULT error.
        iii. Removing, updating, or selecting a target with a target identifier that does not exist shall result in a GENERIC_FAULT error.
        iv. Removing, updating, or selecting a target with a target identifier outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.
    e. Operations include Add(), Update(), Remove(), and Select().

    NOTE: *Select() target would nominate a target to be the primary focus, thus a function similar to a Set() operation. This is a requested action and should return a NO_ERROR.*
3. ***Actively Engaged***
    a. This parameter indicates whether the RWS is currently engaged.
    b. A Boolean value of True means "engaged" and a value of False means "not engaged."
    c. Operations include Get().
4. ***Frame of Reference [Experimental]***
    a. This parameter identifies the physical location of the turret with reference to the vehicle reference point.
    b. The frame of reference shall be presented in a format as defined in the section titled "VICTORY Frame of Reference."
    c. Operations include Get().
    d. This parameter is experimental. It is not required and is not supported in the Remote Weapon Station Device WSDL.
5. ***Ammunition Count***
    a. This parameter identifies the current ammunition count of the gun attached to the turret.
    b. The ammunition count shall be an integer greater than or equal to zero.
    c. Operations include Get().
6. ***Ammunition Type***
    a. This parameter identifies the type of ammunition used by the RWS.
    b. The bounds of the ammunition type are described in the section titled "String Bounds" under "VICTORY Data Types."
    c. Operations include Get().
7. ***Pointing Direction***
    a. This parameter identifies the current pointing direction of the RWS turret.
    b. The format of a pointing direction is described in the section titled "RWS Pointing Direction" under "Remote Weapon Station Message Content."
    c. Operations include Get().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the RWS device management parameters along with a description of additional errors that were not specified but may be returned.*

19.3.4 RWS Interface Complex Types <20020–20210409, Pro>

NOTE: *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

1. ***Target***
    a. **Target ID**
        i. The unique identifier of the target.
        ii. The bounds of the target ID are described in the section titled "String Bounds" under "VICTORY Data Types."
        iii. The target ID shall be provided as part of the target.
    b. **Relative Position**
        i. The position of the target relative to the remote weapon system.
        ii. A detailed description of the relative position data is provided in the section titled "Relative Position" under "RWS Interface Complex Types."
        iii. The relative position shall be provided as part of the target.
    c. **Absolute Position**
        i. The absolute position of the target.
        ii. A detailed description of the relative position data is provided in the section titled "Absolute Position" under "RWS Interface Complex Types."
        iii. The absolute position may be provided as part of the target.
    d. **Priority**
        i. The priority number assigned to the target.
        ii. The lower the priority number, the higher the priority of the target.
        iii. The priority shall be an integer greater than or equal to 1 (highest priority).
        iv. The same priority value may be used by any number of targets.
        v. The priority shall be provided as part of the target.
    e. **Confidence**
        i. The confidence level assigned to the target, indicating the confidence that the target exists.
        ii. The unit of measure shall be a percentage.
        iii. The confidence shall be a real value greater than or equal to zero (lowest confidence level) and less than or equal to 100 (highest confidence level).
        iv. The confidence shall be provided as part of the target.
    f. **Timestamp**
        i. The date and time the target was detected.
        ii. The timestamp shall be in standard dateTime format as specified in the section titled "Timestamp Format for Application Layer Data."
        iii. The timestamp shall be provided as part of the target.
    g. **Selected**
        i. An indication that the target is currently selected.
        ii. A Boolean value of True indicates that the target is selected, False otherwise.
        iii. The selected flag shall be provided as part of the target.
2. ***Relative Position***
    a. **Azimuth**
        i. The azimuth of the target.
        ii. A detailed description of the azimuth data is provided in the section titled "Azimuth" under "VICTORY Data Types."
        iii. The azimuth shall be provided as part of the relative position.
    b. **Elevation**
        i. The elevation of the target.
        ii. A detailed description of the elevation data is provided in the section titled "Elevation" under "VICTORY Data Types."
        iii. The elevation shall be provided as part of the relative position.
    c. **Range**
        i. The range of the target.
        ii. A detailed description of the range data is provided in the section titled "Range" under "VICTORY Data Types."
        iii. The range shall be provided as part of the relative position.
3. ***Absolute Position***
    a. **Latitude**
        i. The latitude of the target.
        ii. A detailed description of the latitude data is provided in the section titled "Latitude" under "VICTORY Data Types."
        iii. The latitude shall be provided as part of the absolute position.

b. **Longitude**
   i. The longitude of the target.
   ii. A detailed description of the longitude data is provided in the section titled "Longitude" under "VICTORY Data Types."
   iii. The longitude shall be provided as part of the absolute position.
c. **Altitude**
   i. The altitude of the target.
   ii. A detailed description of the altitude data is provided in the section titled "Altitude" under "VICTORY Data Types."
   iii. The altitude shall be provided as part of the absolute position.
d. **MGRS**
   i. The MGRS position of the target.
   ii. A detailed description of the MGRS data is provided in the section titled "MGRS" under "VICTORY Data Types."
   iii. The MGRS may be provided as part of the absolute position.

19.4 Logistics Systems Interfaces

Various Logistics Systems are currently being fielded to support the tracking and movement of soldiers and vehicles in the field. These systems are not considered to be within the purview of the VICTORY architecture, but VICTORY nonetheless provides a standard interface for interfacing with the VDB. This interface with the VDB will allow for the sharing of information between the Logistics Systems and VICTORY components, as well as other systems that have interfaces with the VDB.

19.4.1 Vehicle Configuration Service

The function of the vehicle configuration service component type is to store and manage the "as designed" and "as maintained" configurations of the vehicle and its integrated systems.

19.4.1.1 Vehicle Configuration Service Management Parameters <15023–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the vehicle configuration service management parameters. A textual description of the vehicle configuration service management parameters is shown below.

1. ***Vehicle Configuration***
   a. Identifies the current configuration of the vehicle.
   b. The contents of the vehicle configuration are described in the section titled "Vehicle Configuration Service Complex Types."
   c. Operations include Get().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the vehicle configuration service management parameters along with a description of additional errors that were not specified but may be returned.*

19.4.1.2 Vehicle Configuration Service Complex Types <20079–20210409, Pro>

NOTE: *The term "complex types" is used when multiple data types are used to describe an aggregate data type. For example, a complex type to describe a position could be comprised of a latitude, longitude, and altitude.*

1. ***Vehicle Configuration***
   a. **Platform**
      i. Provides information on the vehicle platform.
      ii. A detailed description of the platform data is provided in the section titled "Platform" under "Vehicle Configuration Service Complex Types."
      iii. The platform data shall be provided as part of the vehicle configuration.
2. ***Platform***

a. **GUID**
    i. The globally unique identifier (GUID) of the platform.
    ii. The platform GUID shall be a string representing a valid GUID.
    iii. The bounds of the platform GUID are described in the section titled "String Bounds" under "VICTORY Data Types."
    iv. The platform GUID shall be provided as part of the platform.
b. **Name**
    i. The human-readable name of the platform.
    ii. The bounds of the platform name are described in the section titled "String Bounds" under "VICTORY Data Types."
    iii. The platform name shall be provided as part of the platform.
c. **Tag**
    i. The tag of the platform.
    ii. The bounds of the tag are described in the section titled "String Bounds" under "VICTORY Data Types."
    iii. The tag may be provided as part of the platform.
d. **Active Status**
    i. Identifies whether or not the platform is active.
    ii. The active status shall be a string chosen from one of the following values:
        1. UNDETERMINED
        2. ACTIVE
        3. INACTIVE
        4. INTERMITTENT
    iii. The active status shall be provided as part of the platform.
e. **Serial Number**
    i. The serial number of the platform.
    ii. The serial number shall be a string representing a valid serial number.
    iii. The bounds of the serial number are described in the section titled "String Bounds" under "VICTORY Data Types."
    iv. The serial number shall be provided as part of the platform.
f. **Hull Asset ID**
    i. The identifier of the platform hull asset.
    ii. The bounds of the hull asset ID are described in the section titled "String Bounds" under "VICTORY Data Types."
    iii. The hull asset ID may be provided as part of the platform.
g. **Platform Site ID**
    i. The identifier of the platform site.
    ii. The platform site ID shall be a 64-bit integer.
    iii. The platform site ID shall be provided as part of the platform.
h. **Source CRIS Site ID**
    i. The value to map to MIMOSA Common Relational Information Schema (CRIS) database, in particular the site ID.
    ii. The bounds of the source CRIS site ID are described in the section titled "String Bounds" under "VICTORY Data Types."
    iii. The source CRIS site ID shall be provided as part of the platform.
i. **Platform Model**
    i. Provides platform model information about the vehicle.
    ii. A detailed description of the platform model data is provided in the section titled "Platform Model" under "Vehicle Configuration Service Complex Types."
    iii. The platform model data shall be provided as part of the platform.
j. **Assets**
    i. The functional components of a system. Identifies the model for a component included in a platform model. The components are structured into hierarchical form (subcomponents). Assets are installed to perform lower level component functionality.
    ii. A detailed description of an asset is provided in the section titled "Asset" under "Vehicle Configuration Service Complex Types."
    iii. At least one asset shall be provided as part of the platform.
k. **Unit**
    i. The unit associated with the platform.

ii. A detailed description of a unit is provided in the section titled "Unit" under "Vehicle Configuration Service Complex Types."
iii. The unit may be provided as part of the platform.

3. ***Platform Model***
   a. **ID**
      i. The unique identifier of the platform model.
      ii. The platform ID shall be a 64-bit integer.
      iii. The platform ID shall be provided as part of the platform model.
   b. **Name**
      i. The human-readable name of the platform model.
      ii. The bounds of the platform name are described in the section titled "String Bounds" under "VICTORY Data Types."
      iii. The platform name shall be provided as part of the platform model.
   c. **Type**
      i. The type of platform model.
      ii. The bounds of the platform type are described in the section titled "String Bounds" under "VICTORY Data Types."
      iii. The platform type shall be provided as part of the platform model.
   d. **NSN**
      i. The national stock number (NSN) of the platform model.
      ii. The national stock number shall be a string representing a valid NSN.
      iii. The bounds of the NSN are described in the section titled "String Bounds" under "VICTORY Data Types."
      iv. The NSN shall be provided as part of the platform model.
   e. **Description**
      i. A description of the platform model.
      ii. The bounds of the description are described in the section titled "String Bounds" under "VICTORY Data Types."
      iii. The description shall be provided as part of the platform model.
   f. **Manufacturer**
      i. Provides manufacturer information about the platform model.
      ii. A detailed description of the manufacturer data is provided in the section titled "Manufacturer" under "Vehicle Configuration Service Complex Types."
      iii. The manufacturer data shall be provided as part of the platform model.
   g. **Platform Family**
      i. Provides family information about the platform model.
      ii. A detailed description of the platform family data is provided in the section titled "Platform Family" under "Vehicle Configuration Service Complex Types."
      iii. The platform family data may be provided as part of the platform model.
   h. **Components**
      i. The functional components of a system. Identifies the model for a component included in a platform model. The components are structured into hierarchical form (subcomponents).
      ii. A detailed description of a component is provided in the section titled "Component" under "Vehicle Configuration Service Complex Types."
      iii. At least one component shall be provided as part of the platform model.
   i. **Accessory Components**
      i. The alternative components of the platform model.
      ii. A detailed description of a component is provided in the section titled "Component" under "Vehicle Configuration Service Complex Types."
      iii. Accessory components may be provided as part of the platform model.
   j. **Capabilities Model**
      i. The capabilities of the platform model.
      ii. A detailed description of a capability model is provided in the section titled "Capabilities Model" under "Vehicle Configuration Service Complex Types."
      iii. Capabilities models may be provided as part of the platform model.
   k. **Measurement Location Model**
      i. Measurement locations included in the platform model.
      ii. A detailed description of a measurement location model is provided in the section titled "Measurement Location Model" under "Vehicle Configuration Service Complex Types."

iii. At least one measurement location model shall be provided as part of the platform model.

l. **Fault Model**

i. The fault models used by the platform model.

ii. A detailed description of a fault model is provided in the section titled "Fault Model" under "Vehicle Configuration Service Complex Types."

iii. At least one fault model shall be provided as part of the platform model.

4. ***Manufacturer***

a. **GUID**

i. The globally unique identifier of the manufacturer.

ii. The manufacturer GUID shall be a string representing a valid GUID.

iii. The bounds of the manufacturer GUID are described in the section titled "String Bounds" under "VICTORY Data Types."

iv. The manufacturer GUID shall be provided as part of the manufacturer.

b. **Name**

i. The human-readable name of the manufacturer.

ii. The bounds of the manufacturer name are described in the section titled "String Bounds" under "VICTORY Data Types."

iii. The manufacturer name shall be provided as part of the manufacturer.

c. **Division**

i. The division within the manufacturer responsible for building this model of platform.

ii. The bounds of the division are described in the section titled "String Bounds" under "VICTORY Data Types."

iii. The division may be provided as part of the manufacturer.

d. **CAGE Code**

i. The unique Commercial and Government Entity (CAGE) identifier assigned to suppliers to various government or defense agencies, as well as to government agencies themselves, and also various organizations.

ii. The bounds of the CAGE code are described in the section titled "String Bounds" under "VICTORY Data Types."

iii. The CAGE code shall be provided as part of the manufacturer.

e. **Phone**

i. The phone number of the manufacturer.

ii. The phone number shall be a string representing a valid phone number.

iii. The bounds of the phone number are described in the section titled "String Bounds" under "VICTORY Data Types."

iv. The phone number may be provided as part of the manufacturer.

5. ***Platform Family***

a. **ID**

i. The unique identifier of the platform family.

ii. The platform family ID shall be a 64-bit integer.

iii. The platform family ID shall be provided as part of the platform family.

b. **Name**

i. The human-readable name of the platform family.

ii. The bounds of the platform family name are described in the section titled "String Bounds" under "VICTORY Data Types."

iii. The platform family name shall be provided as part of the platform family.

c. **Birth Site ID**

i. The identifier of the platform family birth site.

ii. The bounds of the birth site ID are described in the section titled "String Bounds" under "VICTORY Data Types."

iii. The birth site ID may be provided as part of the platform family.

6. ***Component***

a. **ID**

i. The unique identifier of the component.

ii. The component ID shall be a 64-bit integer.

iii. The component ID shall be provided as part of the component.

b. **Name**

i. The human-readable name of the component.

ii. The bounds of the component name are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The component name shall be provided as part of the component.

c. **Short Name**
i. The short name of the component.
ii. The bounds of the short name are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The component short name may be provided as part of the component.

d. **Model**
i. The model of the component.
ii. The bounds of the component model are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The component model may be provided as part of the component.

e. **Description**
i. A description of the component.
ii. The bounds of the description are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The description shall be provided as part of the component.

f. **Category**
i. The category of the component.
ii. The bounds of the component category are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The component category may be provided as part of the component.

g. **Type**
i. The type of component.
ii. The bounds of the component type are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The component type may be provided as part of the component.

h. **Source Component ID**
i. The identifier for where the component originated.
ii. The source component ID shall be a 64-bit integer.
iii. The source component ID may be provided as part of the component.

i. **Registry Code GUID**
i. The globally unique identifier for this component from the authoritative source.
ii. The registry code GUID shall be a string representing a valid GUID.
iii. The bounds of the registry code GUID are described in the section titled "String Bounds" under "VICTORY Data Types."
iv. The registry code GUID may be provided as part of the component.

j. **Manufacturer Part**
i. Provides manufacturer part information about the component.
ii. A detailed description of the manufacturer part data is provided in the section titled "Manufacturer Part" under "Vehicle Configuration Service Complex Types."
iii. The manufacturer part data may be provided as part of the component.

k. **Alternative Component**
i. An alternative to this component.
ii. A detailed description of a component is provided in the section titled "Component" under "Vehicle Configuration Service Complex Types."
iii. The alternative component may be provided as part of the component.

l. **Components**
i. The sub-components of the component.
ii. A detailed description of a component is provided in the section titled "Component" under "Vehicle Configuration Service Complex Types."
iii. The components may be provided as part of the component.

7. ***Manufacturer Part***

a. **GUID**
i. The globally unique identifier of the manufacturer part.
ii. The manufacturer part GUID shall be a string representing a valid GUID.
iii. The bounds of the manufacturer part GUID are described in the section titled "String Bounds" under "VICTORY Data Types."

iv. The manufacturer part GUID shall be provided as part of the manufacturer part.

b. **Name**
   i. The human-readable name of the manufacturer part.
   ii. The bounds of the manufacturer part name are described in the section titled "String Bounds" under "VICTORY Data Types."
   iii. The manufacturer part name shall be provided as part of the manufacturer part.

c. **Number**
   i. The manufacturer part number.
   ii. The bounds of the manufacturer part number are described in the section titled "String Bounds" under "VICTORY Data Types."
   iii. The manufacturer part number shall be provided as part of the manufacturer part.

d. **NSN**
   i. The national stock number of the manufacturer part.
   ii. The national stock number shall be a string representing a valid NSN.
   iii. The bounds of the NSN are described in the section titled "String Bounds" under "VICTORY Data Types."
   iv. The NSN shall be provided as part of the manufacturer part.

e. **Manufacturer**
   i. Provides manufacturer information about the manufacturer part.
   ii. A detailed description of the manufacturer data is provided in the section titled "Manufacturer" under "Vehicle Configuration Service Complex Types."
   iii. The manufacturer data may be provided as part of the manufacturer part.

8. ***Capabilities Model***

a. **ID**
   i. The unique identifier of the capability.
   ii. The capability ID shall be a 64-bit integer.
   iii. The capability ID shall be provided as part of the capabilities model.

b. **Name**
   i. The human-readable name of the capability.
   ii. The bounds of the capability name are described in the section titled "String Bounds" under "VICTORY Data Types."
   iii. The capability name shall be provided as part of the capabilities model.

c. **Type ID**
   i. The type of the capability.
   ii. The type ID shall be a 64-bit integer.
   iii. The type ID may be provided as part of the capabilities model.

d. **Description**
   i. The description of the capability.
   ii. The bounds of the description are described in the section titled "String Bounds" under "VICTORY Data Types."
   iii. The description shall be provided as part of the capabilities model.

e. **Top Level Capability**
   i. Indicates whether the capability is a top-level capability or a sub-capability.
   ii. A Boolean value of True means the capability is a top-level capability and False means the capability is a sub-capability.
   iii. The top level capability flag shall be provided as part of the capabilities model.

f. **Source Capability ID**
   i. Identifies the parent capability of a sub-capability.
   ii. The source capability ID shall be a 64-bit integer.
   iii. The source capability ID shall be provided as part of the capabilities model.

g. **Sub Capability ID**
   i. Identifies the child capability of the capability.
   ii. The sub capability ID shall be a 64-bit integer.
   iii. The sub capability ID may be provided as part of the capabilities model.

9. ***Measurement Location Model***

a. **ID**
   i. The unique identifier of the measurement location.
   ii. The component ID shall be a 64-bit integer.
   iii. The component ID shall be provided as part of the measurement location model.

b. **Name**
   i. The human-readable name of the measurement location.
   ii. The bounds of the component name are described in the section titled "String Bounds" under "VICTORY Data Types."
   iii. The component name shall be provided as part of the measurement location model.
c. **Type ID**
   i. The type of the measurement location.
   ii. The type ID shall be a 64-bit integer.
   iii. The type ID shall be provided as part of the measurement location model.
d. **Unit ID**
   i. The unit identifier for the measurement location.
   ii. The unit ID shall be a 64-bit integer.
   iii. The unit ID shall be provided as part of the measurement location model.
e. **Min Range**
   i. The minimum allowable range for the measurement.
   ii. The min range shall be a 32-bit integer.
   iii. The min range may be provided as part of the measurement location model.
f. **Max Range**
   i. The maximum allowable range for the measurement.
   ii. The max range shall be a 32-bit integer.
   iii. The max range may be provided as part of the measurement location model.
g. **Data Source Type ID**
   i. The source of the measurement data.
   ii. The data source type ID shall be a 64-bit integer.
   iii. The data source type ID may be provided as part of the measurement location model.
h. **Collection Duration**
   i. Limits the duration of the measurement.
   ii. The collection duration shall be a 64-bit integer.
   iii. The collection duration may be provided as part of the measurement location model.
i. **Duration Measurement Unit ID**
   i. The unit identifier associated with the collection duration.
   ii. The duration measurement unit ID shall be a 64-bit integer.
   iii. The duration measurement unit ID may be provided as part of the measurement location model.
j. **Measurement Interval**
   i. The interval between measurements.
   ii. The measurement interval shall be a 32-bit integer.
   iii. The measurement interval may be provided as part of the measurement location model.
k. **Interval Measurement Unit ID**
   i. The unit identifier associated with the measurement interval.
   ii. The interval measurement unit ID shall be a 64-bit integer.
   iii. The interval measurement unit ID may be provided as part of the measurement location model.
l. **Registry Code GUID**
   i. The globally unique identifier of the registry code of the measurement location.
   ii. The registry code GUID shall be a string representing a valid GUID.
   iii. The bounds of the registry code GUID are described in the section titled "String Bounds" under "VICTORY Data Types."
   iv. The registry code GUID shall be provided as part of the measurement location model.

10. ***Fault Model***
a. **Platform Model ID**
   i. The platform model using the fault model.
   ii. The platform model ID shall be a 64-bit integer.
   iii. The platform model ID shall be provided as part of the fault model.
b. **ID**
   i. The unique identifier of the fault model.
   ii. The fault ID shall be a 64-bit integer.
   iii. The fault ID shall be provided as part of the fault model.
c. **Name**
   i. The human-readable name of the fault model.

ii. The bounds of the fault name are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The fault name shall be provided as part of the fault model.

d. **System Code**
i. The platform-unique identifier for this fault.
ii. The bounds of the system code are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The system code shall be provided as part of the fault model.

e. **Type ID**
i. The manufacturer's identifier for this fault.
ii. The type ID shall be a 64-bit integer.
iii. The type ID shall be provided as part of the fault model.

f. **Description**
i. The description of the fault.
ii. The description shall be an unsigned integer.
iii. The description shall be provided as part of the fault model.

g. **Label**
i. The label applied to the fault.
ii. The bounds of the label are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The label may be provided as part of the fault model.

h. **Severity ID**
i. The severity level associated with the fault.
ii. The severity ID shall be a 64-bit integer.
iii. The severity ID shall be provided as part of the fault model.

i. **Effect ID**
i. The identifier of the effect of the fault.
ii. The effect ID shall be a 64-bit integer.
iii. The effect ID may be provided as part of the fault model.

11. ***Asset***

a. **Component ID**
i. The unique identifier for the component of this asset.
ii. The component ID shall be a 64-bit integer.
iii. The component ID shall be provided as part of the asset.

b. **GUID**
i. The globally unique identifier of the asset.
ii. The asset GUID shall be a string representing a valid GUID.
iii. The bounds of the asset GUID are described in the section titled "String Bounds" under "VICTORY Data Types."
iv. The asset GUID shall be provided as part of the asset.

c. **Name**
i. The human-readable name of the asset.
ii. The bounds of the asset name are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The asset name shall be provided as part of the asset.

d. **Active Status**
i. Identifies whether or not the asset is active.
ii. The active status shall be a string chosen from one of the following values:
1. UNDETERMINED
2. ACTIVE
3. INACTIVE
4. INTERMITTENT
iii. The active status shall be provided as part of the asset.

e. **Serial Number**
i. The serial number of the asset.
ii. The serial number shall be a string representing a valid serial number.
iii. The bounds of the serial number are described in the section titled "String Bounds" under "VICTORY Data Types."
iv. The serial number shall be provided as part of the asset.

f. **Manufacturer Part GUID**
   i. The globally unique identifier of the manufacturer for the asset.
   ii. The manufacturer part GUID shall be a string representing a valid GUID.
   iii. The bounds of the manufacturer part GUID are described in the section titled "String Bounds" under "VICTORY Data Types."
   iv. The manufacturer part GUID shall be provided as part of the asset.

g. **Asset Type ID**
   i. The type of asset.
   ii. The asset type ID shall be a 64-bit integer.
   iii. The asset type ID shall be provided as part of the asset.

h. **Source Asset ID**
   i. The authoritative source identifier for the asset.
   ii. The bounds of the source asset ID are described in the section titled "String Bounds" under "VICTORY Data Types."
   iii. The source asset ID shall be provided as part of the asset.

i. **Registry Code GUID**
   i. The globally unique identifier of the registry code of the asset.
   ii. The registry code GUID shall be a string representing a valid GUID.
   iii. The bounds of the registry code GUID are described in the section titled "String Bounds" under "VICTORY Data Types."
   iv. The registry code GUID shall be provided as part of the asset.

j. **Is Alternative**
   i. Indicates whether or not the asset is an alternative.
   ii. A Boolean value of True means the asset is an alternative and False means the asset is not an alternative.
   iii. The is alternative flag shall be provided as part of the asset.

k. **Install By**
   i. Identifies the installer of the asset.
   ii. The bounds of the install by are described in the section titled "String Bounds" under "VICTORY Data Types."
   iii. Install by may be provided as part of the asset.

l. **Install Date**
   i. The date when the asset was installed.
   ii. The install date shall be in standard dateTime format as specified in the section titled "Timestamp Format for Application Layer Data."
   iii. The install date may be provided as part of the asset.

m. **Remove By**
   i. Identifies the remover of the asset.
   ii. The bounds of the remove by are described in the section titled "String Bounds" under "VICTORY Data Types."
   iii. Remove by may be provided as part of the asset.

n. **Remove Date**
   i. The date when the asset was removed.
   ii. The remove date shall be in standard dateTime format as specified in the section titled "Timestamp Format for Application Layer Data."
   iii. The remove date may be provided as part of the asset.

12. ***Unit***

a. **GUID**
   i. The globally unique identifier of the unit.
   ii. The unit GUID shall be a string representing a valid GUID.
   iii. The bounds of the unit GUID are described in the section titled "String Bounds" under "VICTORY Data Types."
   iv. The unit GUID shall be provided as part of the unit.

b. **Name**
   i. The human-readable name of the unit.
   ii. The bounds of the unit name are described in the section titled "String Bounds" under "VICTORY Data Types."
   iii. The unit name shall be provided as part of the unit.

c. **Unit Identification Code**

i. The Unit Identification Code (UIC) associated with the unit.
ii. The bounds of the UIC are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The UIC shall be provided as part of the unit.

d. **Description**
i. The description of the unit.
ii. The bounds of the description are described in the section titled "String Bounds" under "VICTORY Data Types."
iii. The description may be provided as part of the unit.

e. **US Agency**
i. The United States agency associated with the unit.
ii. The US agency shall be a 64-bit integer.
iii. The US agency shall be provided as part of the unit.

f. **Major Command ID**
i. The major command associated with the unit.
ii. The major command ID shall be a 64-bit integer.
iii. The major command ID may be provided as part of the unit.

g. **Active Status**
i. Identifies whether or not the unit is active.
ii. The active status shall be a string chosen from one of the following values:
1. UNDETERMINED
2. ACTIVE
3. INACTIVE
4. INTERMITTENT
iii. The active status may be provided as part of the unit.

h. **Type**
i. The type of unit.
ii. The unit type shall be a 64-bit integer.
iii. The unit type may be provided as part of the unit.

19.5 Training Systems Interfaces

Various training systems are currently being fielded to support the training of soldiers in the field. The aim of the interfaces defined in this section are to allow the training community to leverage existing sensors and systems on the vehicles, and to use the VDB for integrating their test systems, thus making their operations more efficient.

19.5.1 Live Training Transformation (LT2) Adapter

The Live Training Transformation (LT2) initiative within the US Army training community is continuing to evolve an architecture for integrating live training systems for Force on Force (FoF) and Force on Target (FoT) training. Live FoF training involves training in a realistic, live environment using tactical engagement simulation systems (TESS) to simulate both the firing capabilities and vulnerability of the vehicle. Vehicle TESS systems utilize Multiple Integrated Laser Engagement System (MILES) transmitters to simulate weapon firing, and MILES detectors to determine casualty effects on targets. Examples of other components that are part of live FoF vehicle training systems include simulators for battle field and weapons effects, a vehicle kill indicator, an audio/visual display, a master controller (processor that hosts control software), an instrumentation system for transmitting data for after action review (AAR), and a vehicle interface module to interface with vehicle power and data systems. Figure 61 shows an example of the LT2 adapter and its use for the live FoF use case.

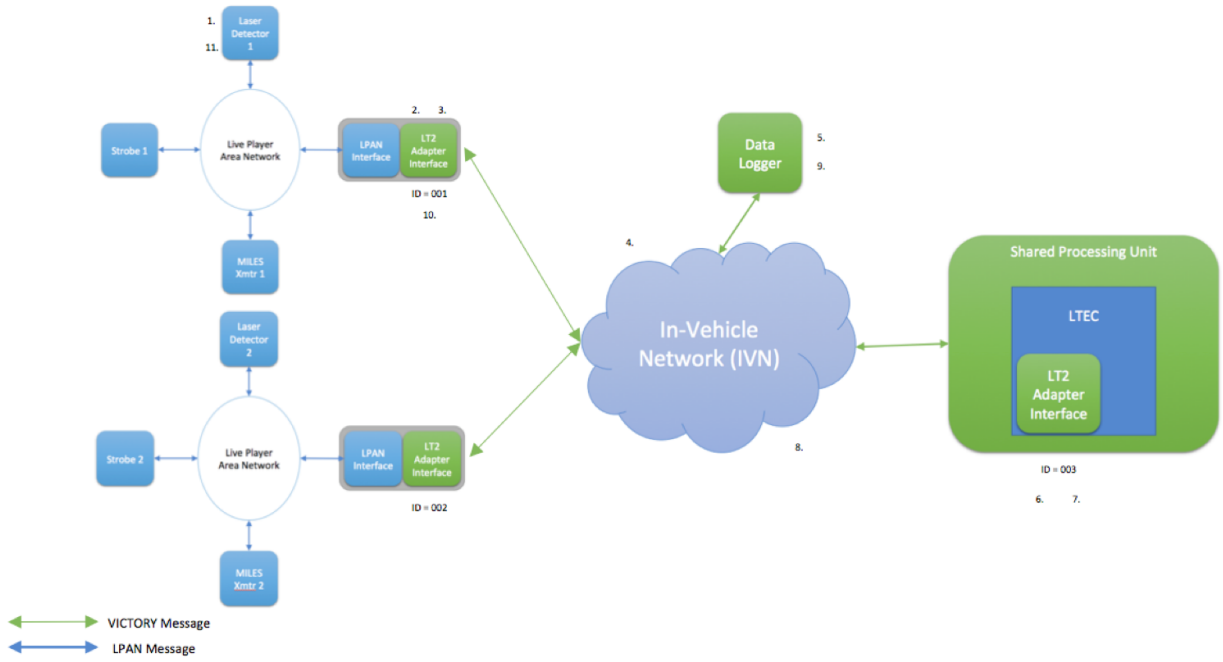

Figure 61. LT2 Interface Concept

The LT2 adapter process begins with an LT2 message generated by an LT2 device connected to the Live Player Area Network (LPAN). In this case, the Laser Energy Detected LPAN message has been generated by "Laser Detector 1":

NOTE: *The messages used in this example are binary messages with formats specified by the appropriate live training standard. For example, for live Force on Force training applications, the LT2 Application message would be defined by the* ***Live PAN Standard PRF-PT–00549*** *and* ***LT2 Message Set Description for Live PAN Components, Document PRF-PT–00635****.*

1. Association Request Emitted
2. Association Request Received by LT2 Adapter App
3. Association Request → LT2 VICTORY Message
    a. Source ID = 001, Destination ID = 003
4. LT2 VICTORY Message sent to IVN
    a. Source ID = 001, Destination ID = 003
5. LT2 VICTORY Message Logged
    a. Source ID = 001, Destination ID = 003
6. LT2 VICTORY Message Received by LTEC and processed
7. LTEC responds to association request. LTEC Association Response → LT2 VICTORY Message
    a. Source ID = 003, Destination ID = 001
8. LT2 VICTORY Message sent to IVN
    a. Source ID = 003, Destination ID = 001
9. LT2 VICTORY Message Logged
    a. Source ID = 003, Destination ID = 001
10. LT2 VICTORY Message received by LT2 Adapter App
11. LT2 VICTORY Message → Association Response
12. Association Response sent to requesting device

Once the device has successfully associated, it is enabled to send an LPAN Message. The LPAN Message follows the same path the association messages followed through the LT2 adapter, over the VDB, and to the receiving application. Any responses to be sent follow the same path as the association response.

The message source and destination within the LT2 VICTORY Message can be used to dictate which LT2 adapter on the network should receive the message if multiple adapters are present.

19.5.1.1 Live Training Transformation Adapter Data Specification <20084–20180215, Pro>

19.5.1.1.1 Live Training Transformation Adapter Message Content

The Live Training Transformation Adapter Data Interface provides a mechanism by which to transport LT2 Application Messages across the VDB. This information will be available to all VDB network nodes that require LT2 Application Message data.

> ***NOTE:*** *The LT2 Application Message is a binary message with its format specified by the appropriate live training standard. For example, for live Force on Force training applications, the LT2 Application message would be defined by the* ***Live PAN Standard PRF-PT–00549*** *and* ***LT2 Message Set Description for Live PAN Components, Document PRF-PT–00635****. The LT2 Adapter data message does not use XML to encapsulate any data.*

1. ***LT2 Application Message Source***
    a. An indicator, unique within the scope of the VDB, used to identify the source of the LT2 Application Message.
        > ***NOTE:*** *The LT2 Application Message Source is not tied to the source ID of the LT2 Adapter component type. It is specific to the LT2 system.*
    b. The LT2 Application Message Source shall be a 64-bit integer value greater than or equal to zero.
    c. The LT2 Application Message Source shall be provided as part of the LT2 adapter message.
2. ***LT2 Application Message Destination***
    a. An indicator, unique within the scope of the VDB, used to identify the destination of the LT2 Application Message.
        > ***NOTE:*** *The LT2 Application Message Destination is specific to the LT2 system.*
    b. The LT2 Application Message Destination shall be a 64-bit integer value greater than or equal to zero.
    c. The LT2 Application Message Destination shall be provided as part of the LT2 adapter message.
3. ***LT2 Data Type***
    a. An indicator as to the type of LT2 data encapsulated in the message.
    b. LT2 Data Type shall be a 16-bit, unsigned integer chosen from Table 290.
        > ***NOTE:*** *A type of "Experimental" has been provided for the purpose of allowing LT2 data types that have not yet been defined, but may later be considered for inclusion in this specification.*
    c. The LT2 Data Type shall be provided as part of the LT2 adapter message.
    d. The LT2 Data Type shall be followed by a 16-bit reserved field.
4. ***LT2 Application Message***
    a. A binary encoded LT2 application message in accordance with one or more of the following live training standards:
        i. Live Training Transformation (LT2) Message Set Description for Live PAN Components (**PRF-PT–0063**)
        ii. Live Training Transformation (LT2) Live Player Area Network Standard (**PRF-PT–00549**)
        iii. Instrumentation System (IS) / Tactical Engagement Simulation Systems (TESS) Interface Standard (**PRF-PT–00552**)
        iv. Performance Specification for the Future Army System of Integrated Targets (FASIT) (**PRF-PT–00468**)
        v. Interface Control Document (ICD) for the Data Communications Interface (DCI) and the Terminal Player Unit (TPU) to Detection Device (DD) Interface Design for the Joint Multinational Readiness Center (JMRC) Instrumentation Systems (IS) Range Data Measurement Subsystem (RDMS) (**706014G**)
        vi. Play Unit Interface Design Document for the Digital Multipurpose Range Complex (**111610–001–84-R13**)
    b. The LT2 Application Message shall be provided as part of the LT2 adapter message.
    c. The size of the LT2 Application Message shall not exceed 65,455 bytes.
        > ***NOTE:*** *The choice to use UDP for VDMs imposes a maximum size for any VICTORY-specific data message. The maximum size of an IP message is 65,535 bytes. The maximum size of the UDP datagram, subtracting the 20-byte IP header, is 65,515 bytes. The maximum size of the UDP datagram payload, subtracting the 8-byte UDP header, is then 65,507 bytes. Thus, the maximum size for a complete VDM (the encoded data messages plus the VICTORY-specific payload header) is 65,507 bytes. Subtracting the minimum 36-byte VDM header, and the 20-bytes for the LT2 message source and LT2 message destination, results in a 65,451-byte maximum size for LT2 application message.*

Table 290. LT2 Data Type Values

Message Type Name	Message Type Value
Live PAN	0x0001
IS TESS	0x0002
FASIT	0x0003
JMRC	0x0004
DMPRC	0x0005
Experimental	0xFFFF

19.5.1.1.2 Live Training Transformation Adapter Data Interface Message Encoding

The information described in the previous section shall be encoded as binary data as prescribed in Table 291. This binary data shall be sent in a VICTORY Data Message (VDM) as the **DATA_PAYLOAD** defined in the section titled "Application Layer Message Encapsulation."

Table 291. LT2 Adapter Data Interface Message

32-bits	
16-bits	**16-bits**
MESSAGE_SOURCE1	
MESSAGE_SOURCE0	
MESSAGE_DESTINATION1	
MESSAGE_DESTINATION0	
LT2_DATA_TYPE	RESERVED
LT2_APPLICATION_MESSAGE • • •	

NOTE: *The size of the LT2 application message (LT2_APPLICATION_MESSAGE) is variable. The length of the message shall not be such that it causes the VDM to exceed the maximum allowable size for the UDP datagram payload.*

19.5.1.1.2.1 LT2 Application Message Source (64-bits)

The LT2 Adapter Data Interface Message shall contain a 64-bit, unsigned integer to identify the source of the LT2 Application message. The source shall be defined using the fields MESSAGE_SOURCE1 and MESSAGE_SOURCE0 in the table above. Where MESSAGE_SOURCE1 contains the most significant 32 bits of the message source, and MESSAGE_SOURCE0 contains the least significant 32 bits.

19.5.1.1.2.2 LT2 Application Destination Source (64-bits)

The LT2 Adapter Data Interface Message shall contain a 64-bit, unsigned integer to identify the destination of the LT2 Application message. The destination shall be defined using the fields MESSAGE_DESTINATION1 and MESSAGE_DESTINATION0 in the table above. Where MESSAGE_DESTINATION1 contains the most significant 32 bits of the message destination, and MESSAGE_DESTINATION0 contains the least significant 32 bits.

19.5.1.1.2.3 LT2 Data Type (16-bits)

The LT2 Adapter Data Interface Message shall contain a 16-bit, unsigned integer to identify the type of the LT2 Application message. The data type shall be defined using the field LT2_DATA_TYPE in the table above.

19.5.1.1.2.4 LT2 Application Message (variable size)

The LT2 Application Message is a binary message with its format specified by the appropriate live training standard. For example, for live Force on Force (FoF) training applications, the LT2 Application message would be defined by the Live PAN Standard, PRF-PT–00549 and the LT2 Cluster Message Set Description (CMSD), PRF-PT–00634. The LT2 Adapter data message does not use XML to encapsulate any data. While the LT2 Adapter component type is being defined to initially support live FoF training applications, it may also be used for live Force on Target, virtual, and/or constructive training applications in the future.

19.5.1.2 Live Training Transformation Adapter Management Parameters <16009–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the LT2 adapter management parameters. A textual description of the LT2 adapter management parameters is shown below.

1. ***Data Enabled***
    a. This parameter identifies whether the Live Training Transformation Adapter Data Interface is sending out VICTORY Data Messages (VDMs) on the VDB.
    b. A Boolean value of True means "enabled" (sending) and a value of False means "disabled" (not sending).
    c. If enabled, a VDM shall be sent every time an LT2 application message is received from an LT2 component/ device (e.g., from the Live Player Area Network [LPAN]).
    d. Operations include Get() and Set().
2. ***Time Uncertainty***
    a. This parameter identifies the uncertainty of the timestamp applied to the VDMs being sent to the VDB by the Live Training Transformation Adapter Data Interface.
    b. The measure of time uncertainty shall be in units of milliseconds.
    c. The time uncertainty shall be a real value at 1 σ.
    d. Operations include Get().
3. ***Data Transport Configuration***
    a. This parameter identifies the transport configuration used by the Live Training Transformation Adapter Data Interface for sending out VDMs on the VDB.
    b. The contents of a data transport configuration are described in the section titled "Data Transport Configuration" under "VICTORY Data Interface Complex Types."
    c. The following errors are defined for the data transport configuration parameter:
        i. Setting multiple data transport configurations with the same unique name shall result in a GENERIC_FAULT error.
        ii. Setting a data transport configuration with a URI outside the format defined by this specification document may result in an OUT_OF_RANGE error.
        iii. If network layer is included in the request, setting a data transport configuration with a network layer outside the list of valid values defined by this specification document may result in an OUT_OF_RANGE error.
        iv. Setting a data transport configuration with a unique name outside the string bounds defined by this specification document may result in an OUT_OF_RANGE error.

v. Setting a data transport configuration with a URI outside the URI bounds defined by this specification document may result in an OUT_OF_RANGE error.

d. Operations include Get() and Set().

NOTE: *The section titled "Management Interface Error Types" provides additional details on the errors defined for the LT2 adapter management parameters along with a description of additional errors that were not specified but may be returned.*

20. VICTORY CONFIGURATION LANGUAGE

20.1 VCL Schema <30044–20210409, Pro>

The VICTORY Configuration Language (VCL) is an XML file used for the configuration of VICTORY components. A graphical representation of the VCL is shown below:

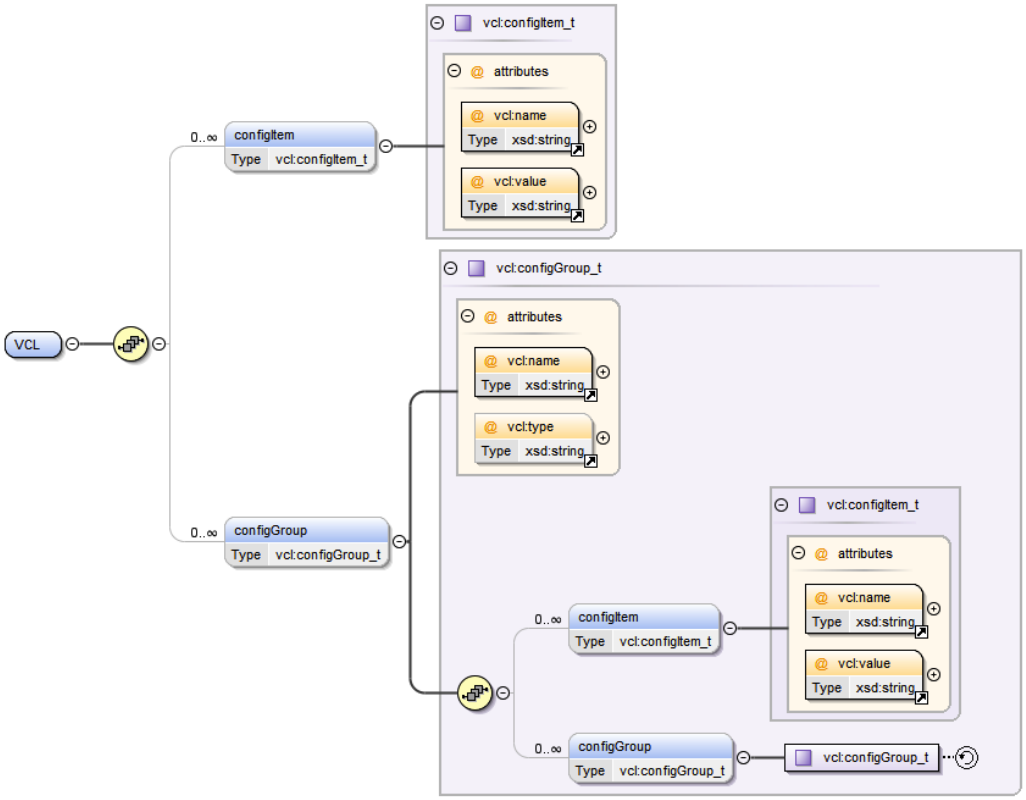

Figure 62. VCL

The VCL provides two concepts for describing VICTORY components:

1. ***Configuration Item***
    a. An element used to encapsulate configuration values.
    b. This element shall contain a ***name*** attribute.
    c. This element shall contain a ***value*** attribute.
    d. The ***name*** and ***value*** attributes shall be strings.
    e. The bounds of the ***name*** and ***value*** attributes are described in the section titled "String Bounds" under "VICTORY Data Types."
2. ***Configuration Group***
    a. An element used to encapsulate both Configuration Item and Configuration Group elements.
    b. This element shall contain a ***name*** attribute.
    c. This element may contain a ***type*** element.
    d. The ***name*** and ***type*** attributes shall be strings.
    e. The bounds of the ***name*** and ***type*** attributes are described in the section titled "String Bounds" under "VICTORY Data Types."

The configuration of VICTORY components is specified inside a VCL Instance Document (VCL ID) which may be stored in a file, or otherwise in a text string in the memory of a computer. The VCL ID shall be formatted in accordance with XML 1.0, as specified in "W3C Recommendation: Extensible Markup Language (XML) 1.0 (Fifth Edition)."

20.2 Configuration Type <30048–20131031, Pro>

1. ***Reportable***
    a. Not configured through the VCL or the management interface.
    b. If a reportable value is included during a set request and is different than its current value, the value in the request shall be ignored.
2. ***Configurable***
    a. Configured only through the VCL.
        i. Changes made to these values through the VCL shall not take effect until the next reboot of the component.
    b. Configurable values are not considered volatile because these values are not constantly changing.
3. ***Mutable***
    a. Configured through the VCL and the management interface.
        i. Changes made to these values through either the VCL or the management interface shall take effect immediately.
    b. Not considered volatile because they are not constantly changing.
4. ***Volatile***
    a. Configured through the VCL and the management interface.
        i. Changes made to these values through the VCL shall not take effect until the next reboot of the component.
            1. Used as initial configuration.
        ii. Changes made to these values through the management interface shall take effect immediately but are expected to be quickly overwritten.
    b. Updated by other sources, such as an external sensor.

20.3 VCL Instance Document Requirements <30045–20131031, Pro>

To be valid, a VICTORY Configuration Language Instance Document (VCL ID) shall meet the following requirements:

- The configuration description of all VICTORY components on the VDB shall be contained inside a single VCL ID.
- The root element of the VCL ID shall be a "VCL" element.
- The "VCL" element shall contain 1 to *N* child Configuration Group elements, where *N* is equal to the number of VICTORY components on the VDB.

20.3.1 Stubbed Out VCL Instance Document Example

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/
VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/
VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup name="de11788a-6983-4d64-986c-ff7eeea47bb0" type="POSITION_SERVICE">
        <!--Position Service-Specific Configuration Description defined inside the
VICTORY specification document would be encapsulated here.-->
    </vcl:configGroup>
    <vcl:configGroup name="deeff43a-6983-4d64-986c-ff7eeea47bb0" type="GENERIC_END_NODE">
        <!--Generic End Node-Specific Configuration Description defined inside the
VICTORY specification document would be encapsulated here.-->
    </vcl:configGroup>
    <vcl:configGroup name="de11788a-2345-4d64-986c-ff7eeea47bb0"
```

```
type="ATTRIBUTE_STORE_SERVICE">
        <!--Attribute Store Service-Specific Configuration Description defined inside
the VICTORY specification document would be encapsulated here.-->
    </vcl:configGroup>
</vcl:VCL>
```

20.4 Tree View Requirements <30046–20131031, Pro>

- Configurations specified using the [0,1] notation may contain either zero or one entry. These values are optional.
- Configurations specified using the [1,∞) notation shall be required and may contain more than one entry.
- Configurations specified using the [0,∞) notation may contain any number of entries, including zero.
- Configurations with no special notation shall be required and shall only contain a single entry.
- The configuration type of each tree element is specified inside the parentheses.

  NOTE: *Not all tree elements contain a configuration type.*

  - All descendants to an element with a configuration type contain the same configuration type as the ancestor element.
- The hierarchy of the tree view shall be followed when building up a VCL ID.
- The ordering of the elements at each level of the hierarchy does not matter when building up a VCL ID, except for the VCL schema validation restrictions.

20.5 Common Component Configuration Description Content <30047–20210409, Pro>

1. ***Unique Name***
   a. A Configuration Item element used to describe the value of a unique name on the VDB.
   b. The ***name*** attribute shall be “uniqueName”.
   c. The ***value*** attribute shall be a string.
2. ***Interface ID***
   a. A Configuration Item element used to describe the interface ID of the component.
   b. The ***name*** attribute shall be “interfaceId”.
   c. The ***value*** attribute shall be a string representation of an integer greater than or equal to zero.
   d. The ***value*** attribute shall be unique within the scope of the VDB.
   e. The ***value*** attribute shall match the interface ID inside all VDMs sent by the component.
3. ***Transport Configuration***
   a. A Configuration Group element used to encapsulate a single transport configuration.
   b. The ***name*** attribute shall be “transportConfiguration”.
   c. This element shall not contain a ***type*** attribute.
4. ***URI***
   a. A Configuration Item element used to describe the value of a URI.
   b. The ***name*** attribute shall be “uri”.
   c. The ***value*** attribute shall be a string.
   d. The ***value*** attribute shall be a valid URI as defined in IETF RFC 3986.
5. ***Version Information SOAP Component***
   a. A Configuration Group element used to encapsulate the version information SOAP component.
   b. The ***name*** attribute shall be “soapComponent”.
   c. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be “VERSION_INFORMATION”.
6. ***Interface Standard Version***
   a. A Configuration Item element used to describe the version of the VICTORY Standard Specifications the component implements.
   b. The ***name*** attribute shall be “interfaceStandardVersion”.
   c. The ***value*** attribute shall be a string.
   d. The ***value*** attribute shall contain a major version followed by a “.” followed by a minor version.
      i. E.g., 1.4
   e. The ***value*** attribute may contain an additional “.” followed by a sub minor version.
      i. E.g., 1.4.2
7. ***VICTORY Data SOAP Component***

   a. A Configuration Group element used to encapsulate the VICTORY data SOAP component.
   b. The ***name*** attribute shall be “soapComponent”.
   c. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be “VICTORY_DATA”.
8. ***Data Enabled***
   a. A Configuration Item element used to describe whether or not data (e.g., VDMs) is being sent to the VDB by the component.
   b. The ***name*** attribute shall be “dataEnabled”.
   c. The ***value*** attribute shall be a case-sensitive string representation of a Boolean value where “true” or “1” means the component is sending data to the VDB and “false” or “0” means the component is not sending data to the VDB.
9. ***Data Transport Configurations***
   a. A Configuration Group element used to encapsulate transport configurations for the data interface portion of a component.
   b. These transport configurations are used to retrieve data being sent by the component.
   c. The ***name*** attribute shall be “dataTransportConfigurations”.
   d. This element shall not contain a ***type*** attribute.
10. ***Update Period***
   a. A Configuration Item element used to describe the time between data updates sent to the VDB by the component.
   b. The ***name*** attribute shall be “updatePeriod”.
   c. The ***value*** attribute shall be a string representation of a real value greater than zero and greater than or equal to the minimum update period.
   d. The ***value*** attribute shall be in units of milliseconds.
11. ***Minimum Update Period***
   a. A Configuration Item element used to describe the maximum frequency at which data updates may be sent to the VDB by the component.
   b. The ***name*** attribute shall be “minimumUpdatePeriod”.
   c. The ***value*** attribute shall be a string representation of a real value greater than zero.
   d. The ***value*** attribute shall be in units of milliseconds.
12. ***Device Information SOAP Component***
   a. A Configuration Group element used to encapsulate the device information SOAP component.
   b. The ***name*** attribute shall be “soapComponent”.
   c. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be “DEVICE_INFORMATION”.
13. ***Manufacturer***
   a. A Configuration Item element used to describe the manufacturer of the device.
   b. The ***name*** attribute shall be “manufacturer”.
   c. The ***value*** attribute shall be a string.
14. ***Model***
   a. A Configuration Item element used to describe the model of the device.
   b. The ***name*** attribute shall be “model”.
   c. The ***value*** attribute shall be a string.
15. ***Equipment ID***
   a. A Configuration Item element used to describe the equipment identifier of the device.
   b. The ***name*** attribute shall be “equipmentId”.
   c. The ***value*** attribute shall be a string.
16. ***Health Publishing SOAP Component***
   a. A Configuration Group element used to encapsulate the health publishing SOAP component.
   b. The ***name*** attribute shall be “soapComponent”.
   c. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be “HEALTH_PUBLISHING”.
17. ***Fault Update Period***
   a. A Configuration Item element used to describe the time between Syslog fault messages sent to the VDB by the component.
   b. The ***name*** attribute shall be “faultUpdatePeriod”.
   c. The ***value*** attribute shall be a string representation of a real value greater than zero.
   d. The ***value*** attribute shall be in units of milliseconds.
18. ***Status Update Period***

   a. A Configuration Item element used to describe the time between Syslog status messages sent to the VDB by the component.
   b. The ***name*** attribute shall be “statusUpdatePeriod”.
   c. The ***value*** attribute shall be a string representation of a real value greater than zero.
   d. The ***value*** attribute shall be in units of milliseconds.

19. ***Enabled Fault Severity Level***
   a. A Configuration Item element used to describe which fault messages are being sent to the VDB with respect to the fault severity.
   b. The ***name*** attribute shall be “enabledFaultSeverityLevel”.
   c. The ***value*** attribute shall be a string chosen from one of the following values:
      i. EMERGENCY
      ii. ALERT
      iii. CRITICAL
      iv. ERROR
      v. WARNING

20. ***Status Enabled***
   a. A Configuration Item element used to describe whether or not the component is sending Syslog status messages to the VDB.
   b. The ***name*** attribute shall be “statusEnabled”.
   c. The ***value*** attribute shall be a case-sensitive string representation of a Boolean value where “true” or “1” means the component is sending status messages and “false” or “0” means the component is not sending status messages.

21. ***Configuration SOAP Component***
   a. A Configuration Group element used to encapsulate the configuration SOAP component.
   b. The ***name*** attribute shall be “soapComponent”.
   c. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be “CONFIGURATION”.

22. ***Port Standard Version***
   a. A Configuration Item element used to describe the version of the SOAP component specification the WSDL port implements.
   b. The ***name*** attribute shall be “portStandardVersion”.
   c. The ***value*** attribute shall be a string.
   d. The ***value*** attribute shall contain a major version followed by a “.” followed by a minor version.
      i. E.g., 1.6

23. ***Ports***
   a. A Configuration Group element used to encapsulate the WSDL ports associated with the SOAP component.
   b. The ***name*** attribute shall be “ports”.
   c. This element shall not contain a ***type*** attribute.

24. ***Port***
   a. A Configuration Group element used to encapsulate a single WSDL port associated with the SOAP component.
   b. The ***name*** attribute shall be “port”.
   c. This element shall not contain a ***type*** attribute.

25. ***Announcement Enabled***
   a. A Configuration Item element used to describe whether or not the component can be discovered with zero configuration.
   b. The ***name*** attribute shall be “announcementEnabled”.
   c. The ***value*** attribute shall be a case-sensitive string representation of a Boolean value where “true” or “1” means the component can be discovered with zero configuration and “false” or “0” means the component cannot be discovered with zero configuration.

26. ***Enabled Fault Severity Levels***
   a. A Configuration Group element used to encapsulate the enabled fault severity levels of the health publishing SOAP component.
   b. The ***name*** attribute shall be “enabledFaultSeverityLevels”.
   c. This element shall not contain a ***type*** attribute.

27. ***Data of Interest***
   a. A Configuration Item element used to describe data to be monitored from an external source (e.g., automotive bus).
   b. The ***name*** attribute shall be “dataOfInterest”.

c. The ***value*** attribute shall be a case-sensitive string representation of a Boolean value where "true" or "1" means the data is to be monitored and "false" or "0" means it is not to be monitored.

> ***NOTE:*** *If the "Data of Interest" is set to "false" or "0," all other configuration items associated with data would not be instantiated by the Set() request and a "NOT_SUPPORTED" error would be returned.*

28. ***Unit ID***
    a. A Configuration Item element used to describe the unique identifier of a unit.
    b. The ***name*** attribute shall be "unitId".
    c. The ***value*** attribute shall be a string representation of an integer greater than or equal to zero.
29. ***Workgroup***
    a. A Configuration Item element used to describe the unique identifier of the workgroup.
    b. The ***name*** attribute shall be "workgroup".
    c. The ***value*** attribute shall be a string representation of an integer greater than or equal to zero.
30. ***Health Level SOAP Component***
    a. A Configuration Group element used to encapsulate the health level SOAP component.
    b. The ***name*** attribute shall be "soapComponent".
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "HEALTH_LEVEL".
31. ***Software Name***
    a. A Configuration Item element used to describe the name of the implementation.
    b. The ***name*** attribute shall be "softwareName".
    c. The ***value*** attribute shall be a string.
32. ***Software Version***
    a. A Configuration Item element used to describe the version of the implementation.
    b. The ***name*** attribute shall be "softwareVersion".
    c. The ***value*** attribute shall be a string.
33. ***Generic End Node SOAP Component***
    a. A Configuration Group element used to encapsulate the generic end node SOAP component.
    b. The ***name*** attribute shall be "soapComponent".
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "GENERIC_END_NODE".
34. ***Ethernet Interfaces***
    a. A Configuration Group element used to encapsulate the ethernet interfaces on the generic end node.
    b. The ***name*** attribute shall be "ethernetInterfaces".
    c. This element shall not contain a ***type*** attribute.
35. ***Ethernet Interface***
    a. A Configuration Group element used to encapsulate an ethernet interface on the generic end node.
    b. The ***name*** attribute shall be "ethernetInterface".
    c. This element shall not contain a ***type*** attribute.
36. ***Interface Name***
    a. A Configuration Item element used to describe the name of the operating system interface (e.g., eth0, wlan0, etc.).
    b. The ***name*** attribute shall be "interfaceName".
    c. The ***value*** attribute shall be a string.
37. ***AA Entrypoints SOAP Component***
    a. A Configuration Group element used to encapsulate the authorization architecture (AA) entrypoints SOAP component.

    > ***NOTE:*** *While the AA Entrypoints and Authentication Entrypoints SOAP components are both Optional, AA Entrypoints cannot be present in a configuration without Authentication Entrypoints.*

    b. The ***name*** attribute shall be "soapComponent".
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "AA_ENTRYPOINTS".
38. ***PES Location***
    a. A Configuration Item element used to describe the location of the policy enforcement service (PES).
    b. The ***name*** attribute shall be "PESLocation".
    c. The ***value*** attribute shall be a valid URI as defined in IETF RFC 3986.
39. ***Authentication Entrypoints SOAP Component***
    a. A Configuration Group element used to encapsulate the authentication entrypoints SOAP component.
    b. The ***name*** attribute shall be "soapComponent".

c. This element shall contain a ***type*** attribute.
    i. The ***type*** attribute shall be "AUTHN_ENTRYPOINTS".

40. ***Authentication Service Location***
    a. A Configuration Item element used to describe the location of the authentication service.
    b. The ***name*** attribute shall be "authNLocation".
    c. The ***value*** attribute shall be a valid URI as defined in IETF RFC 3986.
41. ***MAC Address***
    a. A Configuration Item element used to describe the media access control (MAC) address of the ethernet interface.
    b. The ***name*** attribute shall be "macAddress".
    c. The ***value*** attribute shall be a string.
42. ***Static IP Configurations***
    a. A Configuration Group element used to encapsulate the static IP configurations of the ethernet interface.
    b. The ***name*** attribute shall be "staticIpConfigurations".
    c. This element shall not contain a ***type*** attribute.
43. ***Static IP Configuration***
    a. A Configuration Group element used to encapsulate a static IP configuration of the ethernet interface.
    b. The ***name*** attribute shall be "staticIpConfiguration".
    c. This element shall not contain a ***type*** attribute.
44. ***Address***
    a. A Configuration Item element used to describe the address of the IP configuration.
    b. The ***name*** attribute shall be "address".
    c. The ***value*** attribute shall be a string in the format of a valid IPv4 or IPv6 address.
45. ***Subnet***
    a. A Configuration Item element used to describe the subnet mask of the IP configuration.
    b. The ***name*** attribute shall be "subnet".
    c. The ***value*** attribute shall be a string in the format of a valid IPv4 address or a slash-prefix form (e.g., /8, /64, etc.).
46. ***Broadcast***
    a. A Configuration Item element used to describe the broadcast of the IP configuration.
    b. The ***name*** attribute shall be "broadcast".
    c. The ***value*** attribute shall be a string in the format of a valid IPv4 or IPv6 address.
47. ***Gateway***
    a. A Configuration Item element used to describe the gateway of the IP configuration.
    b. The ***name*** attribute shall be "gateway".
    c. The ***value*** attribute shall be a string in the format of a valid IPv4 or IPv6 address.
48. ***Auto IP Configurations***
    a. A Configuration Group element used to encapsulate the auto IP configurations of the ethernet interface.
    b. The ***name*** attribute shall be "autoIpConfigurations".
    c. This element shall not contain a ***type*** attribute.
49. ***Auto IP Configuration***
    a. A Configuration Group element used to encapsulate an auto IP configuration of the ethernet interface.
    b. The ***name*** attribute shall be "autoIpConfiguration".
    c. This element shall not contain a ***type*** attribute.
50. ***Type***
    a. A Configuration Item element used to describe the type of auto IP configuration.
    b. The ***name*** attribute shall be "type".
    c. The ***value*** attribute shall be a string chosen from one of the following values:
        i. DHCP_V4
        ii. DHCP_V6
        iii. LINK_LOCAL
        iv. STATELESS_AUTOCONFIGURATION
51. ***DHCPv4 Enabled***
    a. A Configuration Item element used to describe whether DHCPv4 is enabled on the ethernet interface.
    b. The ***name*** attribute shall be "dhcpv4Enabled".
    c. The ***value*** attribute shall be a case-sensitive string representation of a Boolean value where "true" or "1" means DHCPv4 is enabled on the ethernet interface and "false" or "0" means DHCPv4 is not enabled on the ethernet interface.
52. ***DHCPv6 Enabled***

a. A Configuration Item element used to describe whether DHCPv6 is enabled on the ethernet interface.
b. The ***name*** attribute shall be "dhcpv6Enabled".
c. The ***value*** attribute shall be a case-sensitive string representation of a Boolean value where "true" or "1" means DHCPv6 is enabled on the ethernet interface and "false" or "0" means DHCPv6 is not enabled on the ethernet interface.

53. ***Hostname***
    a. A Configuration Item element used to describe the hostname of the generic end node.
    b. The ***name*** attribute shall be "hostname".
    c. The ***value*** attribute shall be a string.
54. ***Date/Time***
    a. A Configuration Item element used to describe the date/time of the generic end node.
    b. The ***name*** attribute shall be "dateTime".
    c. The ***value*** attribute shall be a string representation of standard date/time format.
55. ***Auto-Discovery Enabled***
    a. A Configuration Item element used to describe whether auto-discovery is enabled on the generic end node.
    b. The ***name*** attribute shall be "autodiscoveryEnabled".
    c. The ***value*** attribute shall be a case-sensitive string representation of a Boolean value where "true" or "1" means auto-discovery is enabled on the generic end node and "false" or "0" means auto-discovery is not enabled on the generic end node.
56. ***Display Blackout Enabled***
    a. A Configuration Item element used to describe whether the display on the generic end node is blacked out.
    b. The ***name*** attribute shall be "displayBlackoutEnabled".
    c. The ***value*** attribute shall be a case-sensitive string representation of a Boolean value where "true" or "1" means the display and indicator lights on the generic end node are turned off and "false" or "0" means the display and indicator lights on the generic end node are turned on.
    d. A unit with a display shall include the display blackout enabled configuration item.
57. ***Related Components***
    a. A Configuration Group element used to encapsulate related components on the VDB.
    b. The ***name*** attribute shall be "relatedComponents".
    c. This element shall not contain a ***type*** attribute.
58. ***Related Video or Image Situational Awareness Component***
    a. A Configuration Group element used to encapsulate a video or imagery situational awareness component related to the specified video or imagery situational awareness component.
    b. The ***name*** attribute shall be a string representing the unique name of a video or imagery situational awareness component on the VDB.
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be a string representing one of the following types of video or imagery situational awareness components:
            1. VIDEO_SENSOR
            2. VIDEO_ENCODER
            3. CAMERA_GIMBAL
59. ***Version Information URI***
    a. A Configuration Item element used to describe the version information URI of a component on the VDB.
    b. The ***name*** attribute shall be "versionInformationUri".
    c. The ***value*** attribute shall be a string representing the version information URI of a component on the VDB that is related to the specified component.

20.6 Position Service Configuration Description <30005–20210409, Pro>

20.6.1 Position Service Configuration Description Tree View

The tree view below shows the hierarchy of the Position Service configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Position Service Component
    - Interface ID (Configurable)
    - Announcement Enabled (Configurable)

- Position Data SOAP Component
  - Geodetic Datum (Reportable)
  - Position [0,1] (Volatile)
    - Latitude [0,1]
    - Longitude [0,1]
    - Altitude [0,1]
    - MGRS [0,1]
  - Ports (Reportable)
    - Port [1,∞)
      - Port Standard Version
      - URI
- Version Information SOAP Component
  - Interface Standard Version (Reportable)
  - Ports (Reportable)
    - Port [1,∞)
      - Port Standard Version
      - URI
- VICTORY Data SOAP Component
  - Data Enabled (Mutable)
  - Update Period (Mutable)
  - Minimum Update Period (Reportable)
  - Data Transport Configurations (Mutable)
    - Transport Configuration [1,∞)
      - Unique Name
      - URI
  - Ports (Reportable)
    - Port [1,∞)
      - Port Standard Version
      - URI
- Health Publishing SOAP Component
  - Fault Update Period (Mutable)
  - Status Update Period (Mutable)
  - Status Enabled (Mutable)
  - Enabled Fault Severity Levels (Mutable)
    - Enabled Fault Severity Level [0,∞)
  - Data Transport Configurations (Mutable)
    - Transport Configuration [1,∞)
      - Unique Name
      - URI
  - Ports (Reportable)
    - Port [1,∞)
      - Port Standard Version
      - URI
- Configuration SOAP Component
  - Ports (Reportable)
    - Port [1,∞)
      - Port Standard Version
      - URI
- AA Entrypoints SOAP Component [0,1]
  - PES Location (Mutable)
  - Ports (Reportable)
    - Port [1,∞)
      - Port Standard Version
      - URI
- Authentication Entrypoints SOAP Component [0,1]

- Authentication Service Location (Mutable)
- Ports (Reportable)
  - Port [1,∞)
    - Port Standard Version
    - URI

20.6.2 Position Service-Specific Configuration Description Content

1. ***Position Service Component***
   a. A Configuration Group element used to encapsulate the position service configuration description.
   b. The ***name*** attribute shall be a string representing the unique name of the position service.
   c. The ***name*** attribute shall be unique within the scope of the VDB.

   > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

   d. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "POSITION_SERVICE".
2. ***Position Data SOAP Component***
   a. A Configuration Group element used to encapsulate the position data SOAP component.
   b. The ***name*** attribute shall be "soapComponent".
   c. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "POSITION_DATA".
3. ***Geodetic Datum***
   a. A Configuration Item element used to describe the geodetic datum.
   b. The ***name*** attribute shall be "geodeticDatum".
   c. The ***value*** attribute shall be a string chosen from the following values:
      i. WGS_84
4. ***Position***
   a. A Configuration Group element used to encapsulate the initial position of the platform.
   b. The ***name*** attribute shall be "position".
   c. This element shall not contain a ***type*** attribute.
5. ***Latitude***
   a. A Configuration Item element used to describe the initial latitude of the platform.
   b. The ***name*** attribute shall be "latitude".
   c. The ***value*** attribute shall be a string representation of a real value greater than or equal to –90 and less than or equal to 90.
   d. The ***value*** attribute shall be in arc degrees.
6. ***Longitude***
   a. A Configuration Item element used to describe the initial longitude of the platform.
   b. The ***name*** attribute shall be "longitude".
   c. The ***value*** attribute shall be a string representation of a real value greater than –180 and less than or equal to 180.
   d. The ***value*** attribute shall be in arc degrees.
7. ***Altitude***
   a. A Configuration Item element used to describe the initial altitude of the platform.
   b. The ***name*** attribute shall be "altitude".
   c. The ***value*** attribute shall be a string representation of a real value.
   d. The ***value*** attribute shall be in meters.
8. ***MGRS***
   a. A Configuration Item element used to describe the initial MGRS of the platform.
   b. The ***name*** attribute shall be "mgrs".
   c. The ***value*** attribute shall be a string.

   > ***NOTE:*** *Refer to Army Field Manual 3–25.26, 20 July 2001, for more information on the MGRS string format.*

20.6.3 Position Service Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Position Service is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="POSITION_SERVICE" name="de11788a-6983-4d64-986c-ff7eeea47bb0">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configItem value="8" name="interfaceId"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/PositionService/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7dec-11d0-a765-00a0c91e6bf2" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10100" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/PositionService/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
```

```
        </vcl:configGroup>
        <vcl:configGroup type="POSITION_DATA" name="soapComponent">
            <vcl:configItem value="WGS_84" name="geodeticDatum"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/PositionService/positionData" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="position">
                <vcl:configItem value="0" name="altitude"/>
                <vcl:configItem value="0" name="latitude"/>
                <vcl:configItem value="0" name="longitude"/>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/PositionService/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VICTORY_DATA" name="soapComponent">
            <vcl:configItem value="true" name="dataEnabled"/>
            <vcl:configItem value="100" name="minimumUpdatePeriod"/>
            <vcl:configItem value="1000" name="updatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-11ec-11d0-a765-00a0c93e6bf4" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10000" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/PositionService/victoryData" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
```

```
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/PositionService/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/PositionService/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.7 Orientation Service Configuration Description <30006–20210409, Pro>

20.7.1 Orientation Service Configuration Description Tree View

The tree view below shows the hierarchy of the Orientation Service configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Orientation Service Component
  - Interface ID (Configurable)
  - Announcement Enabled (Configurable)
  - Orientation Data SOAP Component
    - Orientation [0,1] (Volatile)
      - Roll [0,1]
      - Pitch [0,1]
      - Yaw [0,1]
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - VICTORY Data SOAP Component
    - Data Enabled (Mutable)
    - Update Period (Mutable)
    - Minimum Update Period (Reportable)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)

        - Port [1,∞)
            - Port Standard Version
            - URI
    - AA Entrypoints SOAP Component [0,1]
        - PES Location (Mutable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI
    - Authentication Entrypoints SOAP Component [0,1]
        - Authentication Service Location (Mutable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI

20.7.2 Orientation Service-Specific Configuration Description Content

1. ***Orientation Service Component***
    a. A Configuration Group element used to encapsulate the orientation service configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the orientation service.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "ORIENTATION_SERVICE".
2. ***Orientation Data SOAP Component***
    a. A Configuration Group element used to encapsulate the orientation data SOAP component.
    b. The ***name*** attribute shall be "soapComponent".
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "ORIENTATION_DATA".
3. ***Orientation***
    a. A Configuration Group element used to encapsulate the initial orientation of the platform.
    b. The ***name*** attribute shall be "orientation".
    c. This element shall not contain a ***type*** attribute.
4. ***Roll***
    a. A Configuration Item element used to describe the initial roll of the platform.
    b. The ***name*** attribute shall be "roll".
    c. The ***value*** attribute shall be a string representation of a real value greater than or equal to –180 and less than or equal to 180.
    d. The ***value*** attribute shall be in arc degrees.
5. ***Pitch***
    a. A Configuration Item element used to describe the initial pitch of the platform.
    b. The ***name*** attribute shall be "pitch".
    c. The ***value*** attribute shall be a string representation of a real value greater than or equal to –90 and less than or equal to 90.
    d. The ***value*** attribute shall be in arc degrees.
6. ***Yaw***
    a. A Configuration Item element used to describe the initial yaw of the platform.
    b. The ***name*** attribute shall be "yaw".
    c. The ***value*** attribute shall be a string representation of a real value greater than or equal to –180 and less than or equal to 180.
    d. The ***value*** attribute shall be in arc degrees.

20.7.3 Orientation Service Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing an Orientation Service is provided below.

NOTE: *The VCL ID below is for a get configuration operation. Reportable values shall not be present in a VCL ID during a set configuration operation.*

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="ORIENTATION_SERVICE" name="6865a590-245c-11e3-8224-0800200c9a66">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configItem value="9" name="interfaceId"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/OrientationService/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7dec-11d0-a765-00a0c91e6bf1" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10124" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/OrientationService/healthPublishing" name="uri"/>
```

```
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
    <vcl:configGroup type="ORIENTATION_DATA" name="soapComponent">
        <vcl:configGroup name="orientation">
            <vcl:configItem value="0" name="pitch"/>
            <vcl:configItem value="180" name="roll"/>
            <vcl:configItem value="0" name="yaw"/>
        </vcl:configGroup>
        <vcl:configGroup name="ports">
            <vcl:configGroup name="port">
                <vcl:configItem value="1.9" name="portStandardVersion"/>
                <vcl:configItem value="http://192.168.1.10:53714/OrientationService/orientationData" name="uri"/>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
    <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
        <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
        <vcl:configGroup name="ports">
            <vcl:configGroup name="port">
                <vcl:configItem value="1.9" name="portStandardVersion"/>
                <vcl:configItem value="http://192.168.1.10:53714/OrientationService/versionInformation" name="uri"/>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
    <vcl:configGroup type="VICTORY_DATA" name="soapComponent">
        <vcl:configItem value="true" name="dataEnabled"/>
        <vcl:configItem value="100" name="minimumUpdatePeriod"/>
        <vcl:configItem value="1000" name="updatePeriod"/>
        <vcl:configGroup name="dataTransportConfigurations">
            <vcl:configGroup name="transportConfiguration">
                <vcl:configItem value="a81d4fae-11ec-11d1-a765-00a0c93e6bf4" name="uniqueName"/>
                <vcl:configItem value="multicast://239.192.1.1:10002" name="uri"/>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup name="ports">
            <vcl:configGroup name="port">
                <vcl:configItem value="1.9" name="portStandardVersion"/>
                <vcl:configItem value="http://192.168.1.10:53714/OrientationService/victoryData" name="uri"/>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
    <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
        <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
```

```
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/OrientationService/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/OrientationService/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.8 Direction of Travel Service Configuration Description <30007–20210409, Pro>

20.8.1 Direction of Travel Service Configuration Description Tree View

The tree view below shows the hierarchy of the Direction of Travel Service configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Direction of Travel Service Component
  - Interface ID (Configurable)
  - Announcement Enabled (Configurable)
  - Direction of Travel Data SOAP Component
    - Direction of Travel [0,1] (Volatile)
      - Heading [0,1]
      - Speed [0,1]
      - Rate of Climb [0,1]
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - VICTORY Data SOAP Component
    - Data Enabled (Mutable)
    - Update Period (Mutable)
    - Minimum Update Period (Reportable)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)

        - Port [1,∞)
            - Port Standard Version
            - URI
    - AA Entrypoints SOAP Component [0,1]
        - PES Location (Mutable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI
    - Authentication Entrypoints SOAP Component [0,1]
        - Authentication Service Location (Mutable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI

20.8.2 Direction of Travel Service-Specific Configuration Description Content

1. ***Direction of Travel Service Component***
    a. A Configuration Group element used to encapsulate the direction of travel service configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the direction of travel service.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "DIRECTION_OF_TRAVEL_SERVICE".
2. ***Direction of Travel Data SOAP Component***
    a. A Configuration Group element used to encapsulate the direction of travel data SOAP component.
    b. The ***name*** attribute shall be "soapComponent".
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "DIRECTION_OF_TRAVEL_DATA".
3. ***Direction of Travel***
    a. A Configuration Group element used to encapsulate the initial direction of travel of the platform.
    b. The ***name*** attribute shall be "directionOfTravel".
    c. This element shall not contain a ***type*** attribute.
4. ***Heading***
    a. A Configuration Item element used to describe the initial heading of the platform.
    b. The ***name*** attribute shall be "heading".
    c. The ***value*** attribute shall be a string representation of a real value greater than 0 and less than or equal to 360.
    d. The ***value*** attribute shall be in arc degrees.
5. ***Speed***
    a. A Configuration Item element used to describe the initial speed of the platform.
    b. The ***name*** attribute shall be "speed".
    c. The ***value*** attribute shall be a string representation of a real value greater than or equal to zero.
    d. The ***value*** attribute shall be in meters per second.
6. ***Rate of Climb***
    a. A Configuration Item element used to describe the initial rate of climb of the platform.
    b. The ***name*** attribute shall be "rateOfClimb".
    c. The ***value*** attribute shall be a string representation of a positive or negative real value.
    d. The ***value*** attribute shall be in meters per second.

20.8.3 Direction of Travel Service Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Direction of Travel Service is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="DIRECTION_OF_TRAVEL_SERVICE" name="e6fa45b0-2226-11e3-8224-0800200c9a66">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configItem value="12" name="interfaceId"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DirectionOfTravelService/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="DIRECTION_OF_TRAVEL_DATA" name="soapComponent">
            <vcl:configGroup name="directionOfTravel">
                <vcl:configItem value="360" name="heading"/>
                <vcl:configItem value="0" name="rateOfClimb"/>
                <vcl:configItem value="0" name="speed"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DirectionOfTravelService/directionOfTravel"
                        name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7dec-11d0-a765-00a0c91e6bf6" name="uniqueName"/>
```

```
                    <vcl:configItem value="multicast://239.192.1.1:10111" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DirectionOfTravelService/healthPublishing"
name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DirectionOfTravelService/versionInformation"
                        name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VICTORY_DATA" name="soapComponent">
            <vcl:configItem value="true" name="dataEnabled"/>
            <vcl:configItem value="100" name="minimumUpdatePeriod"/>
            <vcl:configItem value="1000" name="updatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-11ec-11d2-a765-00a0c93e6bf4" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10004" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DirectionOfTravelService/victoryData" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
```

```
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DirectionOfTravelService/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DirectionOfTravelService/authnEntrypoints"
name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.9 Streaming Audio Data Source Configuration Description <30008–20210409, Pro>

20.9.1 Streaming Audio Data Source Configuration Description Tree View

The tree view below shows the hierarchy of the Streaming Audio Data Source configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Streaming Audio Data Source
  - Announcement Enabled (Configurable)
  - Streaming Audio Data SOAP Component
    - Data Enabled (Mutable)
    - SIP Endpoint URI (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - AA Entrypoints SOAP Component [0,1]
    - PES Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Authentication Entrypoints SOAP Component [0,1]
    - Authentication Service Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI

20.9.2 Streaming Audio Data Source-Specific Configuration Description Content

1. ***Streaming Audio Data Source Component***
    a. A Configuration Group element used to encapsulate the streaming audio data source configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the streaming audio data source.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "STREAMING_AUDIO_SOURCE".
2. ***Streaming Audio Data SOAP Component***
    a. A Configuration Group element used to encapsulate the streaming audio data SOAP component.
    b. The ***name*** attribute shall be "soapComponent".
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "STREAMING_AUDIO_DATA".
3. ***SIP Endpoint URI***
    a. A Configuration Item element used to identify the URI of the Session Initiation Protocol (SIP) endpoint.
    b. The ***name*** attribute shall be "sipEndPointUri".
    c. The ***value*** attribute shall be a string.
    d. The ***value*** attribute shall be a valid URI as defined in IETF RFC 3986.

20.9.3 Streaming Audio Data Source Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Streaming Audio Data Source component is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="STREAMING_AUDIO_SOURCE" name="7d9fadbd-8f86-4208-8fd6-254ce7e33d33">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/StreamingAudio/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7dec-11d0-a765-00a0c91e6bfa" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10101" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/StreamingAudio/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
```

```
        <vcl:configGroup type="STREAMING_AUDIO_DATA" name="soapComponent">
            <vcl:configItem value="true" name="dataEnabled"/>
            <vcl:configItem value="sip:10.50.50.80:5060" name="sipEndPointUri"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/StreamingAudio/streamingAudioData" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/StreamingAudio/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/StreamingAudio/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/StreamingAudio/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.10 Single-Instance Audio Data Source Configuration Description <30009–20210409, Pro>

20.10.1 Single-Instance Audio Data Source Configuration Description Tree View

The tree view below shows the hierarchy of the Single-Instance Audio Data Source configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Single-Instance Audio Data Source
  - Announcement Enabled (Configurable)
  - Single-Instance Audio Data SOAP Component
    - RTSP Server Transport Configurations (Reportable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - AA Entrypoints SOAP Component [0,1]
    - PES Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Authentication Entrypoints SOAP Component [0,1]
    - Authentication Service Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)

- Port Standard Version
- URI

20.10.2 Single-Instance Audio Data Source-Specific Configuration Description Content

1. ***Single-Instance Audio Data Source Component***
    a. A Configuration Group element used to encapsulate the single-instance audio data source configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the single-instance audio data source.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "SINGLE_INSTANCE_AUDIO_SOURCE".
2. ***Single-Instance Audio Data SOAP Component***
    a. A Configuration Group element used to encapsulate the single-instance audio data SOAP component.
    b. The ***name*** attribute shall be "soapComponent".
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "SINGLE_INSTANCE_AUDIO_DATA".
3. ***RTSP Server Transport Configurations***
    a. A Configuration Group element used to encapsulate transport configurations for the Real Time Streaming Protocol (RTSP) server.
    b. These transport configurations are used to retrieve the RTSP server(s) URI(s).
    c. The ***name*** attribute shall be "rtspServerTransportConfigurations".
    d. This element shall not contain a ***type*** attribute.

20.10.3 Single-Instance Audio Data Source Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Single-Instance Audio Data Source is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="SINGLE_INSTANCE_AUDIO_SOURCE" name="d01c3f5f-b75d-42c2-891b-331015fd3c95">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SingleInstanceAudio/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="581d4fae-7dec-11d0-a765-555555555555" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10102" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SingleInstanceAudio/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
```

```
        <vcl:configGroup type="SINGLE_INSTANCE_AUDIO_DATA" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SingleInstanceAudio/singleInstanceData" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="rtspServerTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81aaaae-1aac-1aa0-aaa5-00a0c93e6bf4" name="uniqueName"/>
                    <vcl:configItem value="sip:10.50.50.80:5555" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SingleInstanceAudio/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SingleInstanceAudio/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SingleInstanceAudio/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.11 Intercom Configuration Description <30010–20210409, Pro>

20.11.1 Intercom Configuration Description Tree View

The tree view below shows the hierarchy of the Intercom configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Intercom Component
  - Announcement Enabled (Configurable)
  - Intercom SOAP Component
    - Unit ID (Configurable)
    - Workgroup (Configurable)
    - Intercom Type (Reportable)
    - Intercom Asset Groups (Mutable)
      - Intercom Asset Group [0,∞)
        - Intercom Group ID
        - Assets
          - Asset [1,∞)
    - Crew Stations (Reportable)
      - Crew Station [0,∞)
        - Crew Station ID
        - Active Channels
          - Active Channel [0,∞)
        - Receive Channels
          - Receive Channel [0,∞)
    - Connected Radios (Configurable)
      - Connected Radio [0,∞)
        - Intercom Radio ID
        - Radio Type
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Device Information SOAP Component
    - Equipment ID (Reportable)
    - Manufacturer (Reportable)
    - Model (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)

- Data Transport Configurations (Mutable)
  - Transport Configuration [1,∞)
    - Unique Name
    - URI
- Ports (Reportable)
  - Port [1,∞)
    - Port Standard Version
    - URI

- Configuration SOAP Component
  - Ports (Reportable)
    - Port [1,∞)
      - Port Standard Version
      - URI

- AA Entrypoints SOAP Component [0,1]
  - PES Location (Mutable)
  - Ports (Reportable)
    - Port [1,∞)
      - Port Standard Version
      - URI

- Authentication Entrypoints SOAP Component [0,1]
  - Authentication Service Location (Mutable)
  - Ports (Reportable)
    - Port [1,∞)
      - Port Standard Version
      - URI

20.11.2 Intercom-Specific Configuration Description Content

1. ***Intercom Component***
   a. A Configuration Group element used to encapsulate the intercom configuration description.
   b. The ***name*** attribute shall be a string representing the unique name of the intercom.
   c. The ***name*** attribute shall be unique within the scope of the VDB.

   > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

   d. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "INTERCOM".
2. ***Intercom SOAP Component***
   a. A Configuration Group element used to encapsulate the intercom SOAP component.
   b. The ***name*** attribute shall be "soapComponent".
   c. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "INTERCOM".
3. ***Intercom Type***
   a. A Configuration Item element used to describe the intercom type.
   b. The ***name*** attribute shall be "intercomType".
   c. The ***value*** attribute shall be a string chosen from one of the following values:
      i. VIC_3
      ii. VIC_5
      iii. NETCOM_V
      iv. TACTICOM
      v. TOCNET
4. ***Intercom Asset Groups***
   a. A Configuration Group element used to encapsulate the initial configuration of the intercom asset groups.
   b. The ***name*** attribute shall be "intercomAssetGroups".
   c. This element shall not contain a ***type*** attribute.
5. ***Intercom Asset Group***
   a. A Configuration Group element used to describe an intercom asset group.

   b. The ***name*** attribute shall be “intercomAssetGroup”.
   c. This element shall not contain a ***type*** attribute.
6. ***Intercom Group ID***
   a. A Configuration Item element used to describe the unique identifier of the intercom group.
   b. The ***name*** attribute shall be “intercomGroupId”.
   c. The ***value*** attribute shall be a string.
7. ***Assets***
   a. A Configuration Group element used to describe the list of assets within the intercom group.
   b. The ***name*** attribute shall be “assets”.
   c. This element shall not contain a ***type*** attribute.
8. ***Asset***
   a. A Configuration Item element used to describe an asset.
   b. The ***name*** attribute shall be “asset”.
   c. The ***value*** attribute shall be a string.
9. ***Crew Stations***
   a. A Configuration Group element used to encapsulate the configuration of intercom crew stations.
   b. The ***name*** attribute shall be “crewStations”.
   c. This element shall not contain a ***type*** attribute.
10. ***Crew Station***
   a. A Configuration Group element used to encapsulate the initial configuration of the intercom crew station.
   b. The ***name*** attribute shall be “crewStation”.
   c. This element shall not contain a ***type*** attribute.
11. ***Crew Station ID***
   a. A Configuration Item element used to describe the unique identifier of the crew station.
   b. The ***name*** attribute shall be “crewStationId”.
   c. The ***value*** attribute shall be a string.
12. ***Active Channels***
   a. A Configuration Group element used to encapsulate the configuration of the intercom crew station active channels.
   b. The ***name*** attribute shall be “activeChannels”.
   c. This element shall not contain a ***type*** attribute.
13. ***Active Channel***
   a. A Configuration Item element used to describe the configuration of the intercom crew station active channel.
   b. The ***name*** attribute shall be “activeChannel”.
   c. The ***value*** attribute shall be a string.
14. ***Receive Channels***
   a. A Configuration Group element used to encapsulate the configuration of the intercom crew station receive channels.
   b. The ***name*** attribute shall be “receiveChannels”.
   c. This element shall not contain a ***type*** attribute.
15. ***Receive Channel***
   a. A Configuration Item element used to describe the configuration of the intercom crew station receive channel.
   b. The ***name*** attribute shall be “receiveChannel”.
   c. The ***value*** attribute shall be a string.
16. ***Connected Radios***
   a. A Configuration Group element used to encapsulate the configuration of the intercom radios.
   b. The ***name*** attribute shall be “connectedRadios”.
   c. This element shall not contain a ***type*** attribute.
17. ***Connected Radio***
   a. A Configuration Group element used to describe the radio connected to the intercom.
   b. The ***name*** attribute shall be “connectedRadio”.
   c. This element shall not contain a ***type*** attribute.
18. ***Intercom Radio ID***
   a. A Configuration Item element used to describe the unique identifier for the radio.
   b. The ***name*** attribute shall be “intercomRadioId”.
   c. The ***value*** attribute shall be a string.
19. ***Radio Type***
   a. A Configuration Item element used to describe the type of the radio.
   b. The ***name*** attribute shall be “radioType”.

c. The ***value*** attribute shall be a string chosen from one of the following values:
   i. AN_VRC_87
   ii. AN_VRC_88
   iii. AN_VRC_89
   iv. AN_VRC_90
   v. AN_VRC_91
   vi. AN_VRC_92
   vii. AN_PRC_117
   viii. AN_PRC_119
   ix. SINCGARS
   x. OTHER

20.11.3 Intercom Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing an Intercom is provided below.

NOTE: *The VCL ID below is for a get configuration operation. Reportable values shall not be present in a VCL ID during a set configuration operation.*

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="INTERCOM" name="d5b32e15-8503-485d-a4d7-f1ecb3d15abd">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Intercom/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="DEVICE_INFORMATION" name="soapComponent">
            <vcl:configItem value="1" name="equipmentId"/>
            <vcl:configItem value="VIS" name="manufacturer"/>
            <vcl:configItem value="VIS-1" name="model"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Intercom/deviceInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7dec-11d0-a765-00a0c91e6bf0" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10105" name="uri"/>
                </vcl:configGroup>
```

```
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Intercom/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="INTERCOM" name="soapComponent">
            <vcl:configItem value="VIC_5" name="intercomType"/>
            <vcl:configItem value="1" name="unitId"/>
            <vcl:configItem value="5" name="workgroup"/>
            <vcl:configGroup name="connectedRadios">
                <vcl:configGroup name="connectedRadio">
                    <vcl:configItem value="023e95e3-287b-41e4-ac02-3d9c69e049e1" name="intercomRadioId"/>
                    <vcl:configItem value="AN_PRC_117" name="radioType"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="intercomAssetGroups">
                <vcl:configGroup name="intercomAssetGroup">
                    <vcl:configItem value="group1" name="intercomGroupId"/>
                    <vcl:configGroup name="assets">
                        <vcl:configItem value="loudspeaker" name="asset"/>
                        <vcl:configItem value="MP3 player" name="asset"/>
                    </vcl:configGroup>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="crewStations">
                <vcl:configGroup name="crewStation">
                    <vcl:configItem value="CMDR" name="crewStationId"/>
                    <vcl:configGroup name="activeChannels">
                        <vcl:configItem value="NET1" name="activeChannel"/>
                    </vcl:configGroup>
                    <vcl:configGroup name="receiveChannels">
                        <vcl:configItem value="NET1" name="receiveChannel"/>
                    </vcl:configGroup>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
```

```
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Intercom/intercom" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Intercom/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
            <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
                <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
                <vcl:configGroup name="ports">
                    <vcl:configGroup name="port">
                        <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Intercom/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Intercom/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.12 Legacy Radio Configuration Description <30053–20210409, Pro>

20.12.1 Legacy Radio Configuration Description Tree View

The tree view below shows the hierarchy of the Legacy Radio configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Legacy Radio Component
  - Announcement Enabled (Configurable)
  - Legacy Radio Device SOAP Component
    - Power (Mutable)
    - Radio Type (Reportable)
    - Volume Step Maximum (Reportable)
    - Zeroize Capability (Reportable)
    - Zeroize Responses (Reportable)
    - Available Presets (Reportable)
      - Preset [1,∞)
        - Name
        - Id
        - Short Name [0,1]
        - Waveform [0,1]
        - Transmit Frequency [0,1]
        - Receive Frequency [0,1]
        - Mode [0,1]
        - Security Mode [0,1]
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Device Information SOAP Component
    - Equipment ID (Reportable)
    - Manufacturer (Reportable)
    - Model (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)

        - Port [1,∞)
            - Port Standard Version
            - URI
    - Configuration SOAP Component
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI
    - AA Entrypoints SOAP Component [0,1]
        - PES Location (Mutable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI
    - Authentication Entrypoints SOAP Component [0,1]
        - Authentication Service Location (Mutable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI

20.12.2 Legacy Radio-Specific Configuration Description Content

1. ***Legacy Radio Component***
    a. A Configuration Group element used to encapsulate the legacy radio configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the legacy radio.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "LEGACY_RADIO".
2. ***Legacy Radio Device SOAP Component***
    a. A Configuration Group element used to encapsulate the legacy radio device SOAP component.
    b. The ***name*** attribute shall be "soapComponent".
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "LEGACY_RADIO_DEVICE".
3. ***Power***
    a. A Configuration Item element used to describe the legacy radio power level.
    b. The ***name*** attribute shall be "power".
    c. The ***value*** attribute shall be a string chosen from one of the following values:
        i. LOW
        ii. MEDIUM
        iii. HIGH
        iv. PA
4. ***Available Presets***
    a. A Configuration Group element used to provide the list of presets available for the legacy radio.
    b. The ***name*** value shall be "availablePresets".
    c. This element shall not contain a ***type*** attribute.
5. ***Preset***
    a. A Configuration Group element used to encapsulate a legacy radio preset.
    b. The ***name*** value shall be "preset".
    c. This element shall not contain a ***type*** attribute.
6. ***Name***
    a. A Configuration Item element used to describe the name of a legacy radio preset.
    b. The ***name*** attribute shall be "name".
    c. The ***value*** attribute shall be a string.
7. ***ID***

a. A Configuration Item element used to describe the unique identifier of the legacy radio preset.
b. The ***name*** attribute shall be "id".
c. The ***value*** attribute shall be a string.
d. The ***value*** attribute shall be unique within the scope of the legacy radio.

> ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

8. ***Short Name***
   a. A Configuration Item element used to describe the shorthand name of the legacy radio preset.
   b. The ***name*** attribute shall be "shortName".
   c. The ***value*** attribute shall be a string.
9. ***Waveform***
   a. A Configuration Item element used to describe the waveform of a legacy radio preset.
   b. The ***name*** attribute shall be "waveform".
   c. The ***value*** attribute shall be a string.
10. ***Transmit Frequency***
    a. A Configuration Item element used to describe the transmit frequency of a legacy radio preset.

    > ***NOTE:*** *For a preset without a fixed frequency (e.g., a frequency hopping preset), this element is not used.*

    b. The ***name*** attribute shall be "txFrequency".
    c. The ***value*** attribute shall be a string representation of an integer greater than zero.
    d. The ***value*** attribute shall be in units of Hertz (Hz).
    e. If the Receive Frequency configuration item is provided, this configuration item shall also be provided.
11. ***Receive Frequency***
    a. A Configuration Item element used to describe the receive frequency of a legacy radio preset.

    > ***NOTE:*** *For a preset without a fixed frequency (e.g., a frequency hopping preset), this element is not used.*

    b. The ***name*** attribute shall be "rxFrequency".
    c. The ***value*** attribute shall be a string representation of an integer greater than zero.
    d. The ***value*** attribute shall be in units of Hertz (Hz).
    e. If the Transmit Frequency configuration item is provided, this configuration item shall also be provided.
12. ***Mode***
    a. A Configuration Item element used to describe the mode of a legacy radio preset.
    b. The ***name*** attribute shall be "mode".
    c. The ***value*** attribute shall be a string.

    > ***NOTE:*** *Single Channel, Frequency Hopping, Cue, Upper Side Band, Lower Side Band, etc., are examples of modes that could be returned as part of the preset.*

13. ***Security Mode***
    a. A Configuration Item element used to describe the security mode of a legacy radio preset.
    b. The ***name*** attribute shall be "securityMode".
    c. The ***value*** attribute shall be a string chosen from one of the following values:
       i. PLAIN
       ii. CIPHER
14. ***Radio Type***
    a. A Configuration Item element used to describe the radio type of a legacy radio.
    b. The ***name*** attribute shall be "radioType".
    c. The ***value*** attribute shall be a string chosen from one of the following values:
       i. AN_ARC_220
       ii. AN_ARC_231
       iii. AN_PRC_117F
       iv. AN_PRC_117G
       v. AN_PRC_148
       vi. AN_PRC_148_JEM
       vii. AN_PRC_150
       viii. AN_PRC_152
       ix. AN_PSC_5D
       x. RT_1523
       xi. OTHER

1. If the value ‘OTHER’ is returned, it can be accompanied with additional text. The additional text will be added to the value ‘OTHER’ and a ‘;’ will separate the additional information.

> ***NOTE:*** *‘OTHER;AN_PRC_154’ is an example of the use of ‘OTHER’ and additional information.*

15. ***Volume Step Maximum***
    a. A Configuration Item element used to describe the maximum volume step of a legacy radio.
    b. The ***name*** attribute shall be “volumeStepMaximum”.
    c. The ***value*** attribute shall be a string representation of an integer greater than zero.
16. ***Zeroize Capability***
    a. A Configuration Item element used to describe the capability to zeroize the legacy radio.
    b. The ***name*** attribute shall be “zeroizeCapability”.
    c. The ***value*** attribute shall be a case-sensitive string representation of a Boolean value where “true” or “1” means the radio can be zeroized and “false” or “0” means the radio cannot be zeroized.
17. ***Zeroize Responses***
    a. A Configuration Item element used to describe the responses available from the legacy radio should there be a request to zeroize it.
    b. The ***name*** attribute shall be “zeroizeResponses”.
    c. The ***value*** attribute shall be a string chosen from one or more of the following values. Multiple values shall be delimited using a ‘;’.
        i. ZEROIZED
        ii. NOT_ZEROIZED
        iii. IN_PROGRESS
        iv. FAILED
        v. NOT_REPORTABLE

20.12.3 Legacy Radio Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Legacy Radio is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="LEGACY_RADIO" name="c8e2ebc0-2bb0-11e3-8224-0800200c9a66">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/LegacyRadio/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="LEGACY_RADIO_DEVICE" name="soapComponent">
            <vcl:configItem value="MEDIUM" name="power"/>
            <vcl:configItem value="RT-1523" name="radioType"/>
            <vcl:configItem value="9" name="volumeStepMaximum"/>
            <vcl:configItem value="true" name="zeroizeCapability"/>
            <vcl:configItem value="ZEROIZED;NOT_ZEROIZED;IN_PROGRESS;FAILED" name="zeroizeResponses"/>
            <vcl:configGroup name="availablePresets">
                <vcl:configGroup name="preset">
                    <vcl:configItem value="NET1" name="name"/>
                    <vcl:configItem value="017a204d-d226-4f65-9c43-087e82b75940" name="id"/>
                    <vcl:configItem value="N1" name="shortName"/>
                    <vcl:configItem value="SINCGARS" name="waveform"/>
                    <vcl:configItem value="470000000" name="txFrequency"/>
                    <vcl:configItem value="470000000" name="rxFrequency"/>
                    <vcl:configItem value="Single Channel" name="mode"/>
                    <vcl:configItem value="PLAIN" name="securityMode"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/LegacyRadio/legacyRadio" name="uri"/>
```

```
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
    <vcl:configGroup type="DEVICE_INFORMATION" name="soapComponent">
        <vcl:configItem value="1" name="equipmentId"/>
        <vcl:configItem value="Harris" name="manufacturer"/>
        <vcl:configGroup name="ports">
            <vcl:configGroup name="port">
                <vcl:configItem value="1.9" name="portStandardVersion"/>
                <vcl:configItem value="http://192.168.1.10:53714/LegacyRadio/deviceInformation" name="uri"/>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
    <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
        <vcl:configItem value="1000" name="faultUpdatePeriod"/>
        <vcl:configItem value="true" name="statusEnabled"/>
        <vcl:configItem value="5000" name="statusUpdatePeriod"/>
        <vcl:configGroup name="dataTransportConfigurations">
            <vcl:configGroup name="transportConfiguration">
                <vcl:configItem value="b91d4fae-7dec-11d0-a765-00a0c91e6bfa" name="uniqueName"/>
                <vcl:configItem value="multicast://239.192.1.1:10107" name="uri"/>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup name="enabledFaultSeverityLevels">
            <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
            <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
            <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
        </vcl:configGroup>
        <vcl:configGroup name="ports">
            <vcl:configGroup name="port">
                <vcl:configItem value="1.9" name="portStandardVersion"/>
                <vcl:configItem value="http://192.168.1.10:53714/LegacyRadio/healthPublishing" name="uri"/>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
    <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
        <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
        <vcl:configGroup name="ports">
            <vcl:configGroup name="port">
                <vcl:configItem value="1.9" name="portStandardVersion"/>
                <vcl:configItem value="http://192.168.1.10:53714/LegacyRadio/versionInformation" name="uri"/>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
```

```
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/LegacyRadio/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/LegacyRadio/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.13 Video or Image Sensor Configuration Description <30012–20210409, Pro>

20.13.1 Video or Image Sensor Configuration Description Tree View

The tree view below shows the hierarchy of the Video or Image Sensor configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Video or Image Sensor Component
  - Announcement Enabled (Configurable)
  - Related Components (Configurable)
    - Related Video or Image Situational Awareness Component [0,∞)
      - Version Information URI
  - Video or Image Sensor SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - IR Video or Image Sensor SOAP Component [0,1]
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Device Information SOAP Component
    - Equipment ID (Reportable)
    - Manufacturer (Reportable)
    - Model (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)

            - Port Standard Version
            - URI
    - AA Entrypoints SOAP Component [0,1]
        - PES Location (Mutable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI
    - Authentication Entrypoints SOAP Component [0,1]
        - Authentication Service Location (Mutable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI

20.13.2 Video or Image Sensor-Specific Configuration Description Content

1. ***Video or Image Sensor Component***
    a. A Configuration Group element used to encapsulate the video or image sensor configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the video or image sensor.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be “VIDEO_SENSOR”.
2. ***Video or Image Sensor SOAP Component***
    a. A Configuration Group element used to encapsulate the video or image sensor SOAP component.
    b. The ***name*** attribute shall be “soapComponent”.
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be “VIDEO_SENSOR”.
3. ***IR Video or Image Sensor SOAP Component***
    a. A Configuration Group element used to encapsulate the IR video or image sensor SOAP component.
    b. The ***name*** attribute shall be “soapComponent”.
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be “IR_VIDEO_SENSOR”.

20.13.3 Video or Image Sensor Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Video or Image Sensor is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="VIDEO_SENSOR" name="abe587a1-e4b0-4251-b43e-06f7339b8fe5">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup name="relatedComponents">
            <vcl:configGroup type="CAMERA_GIMBAL" name="camera-gimbal-uuid-string">
                <vcl:configItem name="versionInformationUri"
                    value="http://192.168.1.10:53714/CameraGimbal/versionInformation"/>
            </vcl:configGroup>
            <vcl:configGroup type="VIDEO_ENCODER" name="video-encoder-uuid-string">
                <vcl:configItem name="versionInformationUri"
                    value="http://192.168.1.10:53714/VideoEncoder/versionInformation"/>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VideoSensor/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="DEVICE_INFORMATION" name="soapComponent">
            <vcl:configItem value="1" name="equipmentId"/>
            <vcl:configItem value="axis" name="manufacturer"/>
            <vcl:configItem value="ptz" name="model"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VideoSensor/deviceInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
```

```
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7dec-11d0-a765-00a0c91e6bfe" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10106" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VideoSensor/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VideoSensor/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VIDEO_SENSOR" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VideoSensor/videoSensor" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="IR_VIDEO_SENSOR" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VideoSensor/irVideoSensor" name="uri"/>
                </vcl:configGroup>
```

```
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VideoSensor/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VideoSensor/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.14 Video or Image Encoder Configuration Description <30013–20210409, Pro>

20.14.1 Video or Image Encoder Configuration Description Tree View

The tree view below shows the hierarchy of the Video or Image Encoder configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Video or Image Encoder Component
  - Announcement Enabled (Configurable)
  - Related Components (Configurable)
    - Related Video or Image Situational Awareness Component [0,∞)
      - Version Information URI
  - Video or Image Encoder SOAP Component
    - Video Frame Rate (Mutable)
    - Video Resolution (Mutable)
      - Horizontal Resolution
      - Vertical Resolution
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Device Information SOAP Component
    - Equipment ID (Reportable)
    - Manufacturer (Reportable)
    - Model (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Video Data SOAP Component
    - Data Enabled (Mutable)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)

            - Unique Name
            - URI
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI
    - Configuration SOAP Component
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI
    - AA Entrypoints SOAP Component [0,1]
        - PES Location (Mutable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI
    - Authentication Entrypoints SOAP Component [0,1]
        - Authentication Service Location (Mutable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI

20.14.2 Video or Image Encoder-Specific Configuration Description Content

1. ***Video or Image Encoder Component***
    a. A Configuration Group element used to encapsulate the video or image encoder configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the video or image encoder.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be “VIDEO_ENCODER”.
2. ***Video or Image Encoder SOAP Component***
    a. A Configuration Group element used to encapsulate the video or image encoder SOAP component.
    b. The ***name*** attribute shall be “soapComponent”.
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be “VIDEO_ENCODER”.
3. ***Video Data SOAP Component***
    a. A Configuration Group element used to encapsulate the video data SOAP component.
    b. The ***name*** attribute shall be “soapComponent”.
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be “VIDEO_DATA”.
4. ***Video Resolution***
    a. A Configuration Group element used to encapsulate the number of distinct pixels in each dimension that can be displayed by the video encoder.
    b. The ***name*** attribute shall be “videoResolution”.
    c. This element shall not contain a ***type*** attribute.
5. ***Horizontal Resolution***
    a. A Configuration Item element used to describe the horizontal resolution.
    b. The ***name*** attribute shall be “horizontalResolution”.
    c. The ***value*** attribute shall be a string representation of an integer greater than or equal to zero.
    d. The ***value*** attribute shall be in units of pixels.
6. ***Vertical Resolution***
    a. A Configuration Item element used to describe the vertical resolution.

b. The ***name*** attribute shall be “verticalResolution”.
c. The ***value*** attribute shall be a string representation of an integer greater than or equal to zero.
d. The ***value*** attribute shall be in units of pixels.

7. ***Video Frame Rate***
   a. A Configuration Item element used to describe the frequency at which the video encoder produces unique consecutive images.
   b. The ***name*** attribute shall be “videoFrameRate”.
   c. The ***value*** attribute shall be a string representation of a real value greater than zero.
   d. The ***value*** attribute shall be in units of frames per second.

20.14.3 Video or Image Encoder Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Video or Image Encoder is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="VIDEO_ENCODER" name="f11f13f5-09da-42a7-ba36-6bcc1452dd81">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup name="relatedComponents">
            <vcl:configGroup type="CAMERA_GIMBAL" name="camera-gimbal-uuid-string">
                <vcl:configItem name="versionInformationUri"
                    value="http://192.168.1.10:53714/CameraGimbal/versionInformation"/>
            </vcl:configGroup>
            <vcl:configGroup type="VIDEO_SENSOR" name="video-sensor-uuid-string">
                <vcl:configItem name="versionInformationUri"
                    value="http://192.168.1.10:53714/VideoSensor/versionInformation"/>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VideoEncoder/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="DEVICE_INFORMATION" name="soapComponent">
            <vcl:configItem value="1" name="equipmentId"/>
            <vcl:configItem value="axis" name="manufacturer"/>
            <vcl:configItem value="ptz" name="model"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VideoEncoder/deviceInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
```

```
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7dec-11d0-a766-00a0c91e6bf6" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10119" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VideoEncoder/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VideoEncoder/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VIDEO_DATA" name="soapComponent">
            <vcl:configItem value="true" name="dataEnabled"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-11e7-11d0-aaaf-00a0c93e6bf4" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10006" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VideoEncoder/videoData" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
```

```
        </vcl:configGroup>
        <vcl:configGroup type="VIDEO_ENCODER" name="soapComponent">
            <vcl:configItem value="30" name="videoFrameRate"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VideoEncoder/videoEncoder" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="videoResolution">
                <vcl:configItem value="480" name="horizontalResolution"/>
                <vcl:configItem value="360" name="verticalResolution"/>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VideoEncoder/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VideoEncoder/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.15 Camera Gimbal Configuration Description <30014–20210409, Pro>

20.15.1 Camera Gimbal Configuration Description Tree View

The tree view below shows the hierarchy of the Camera Gimbal configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Camera Gimbal Component
  - Interface ID (Configurable)
  - Announcement Enabled (Configurable)
  - Related Components (Configurable)
    - Related Video or Image Situational Awareness Component [0,∞)
      - Version Information URI
  - Camera Gimbal Data SOAP Component
    - Update Period (Mutable)
    - Minimum Update Period (Reportable)
    - Camera Gimbal Data Types
      - Camera Gimbal Data Type [1,∞)
        - Data Enabled (Mutable)
        - Data Transport Configurations (Mutable)
          - Transport Configuration [1,∞)
            - Unique Name
            - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Camera Gimbal Device SOAP Component
    - Relative Camera Orientation [0,1] (Volatile)
      - Pan [0,1]
      - Tilt [0,1]
      - Camera Roll [0,1]
    - Relative Camera Position [0,1] (Volatile)
      - X [0,1]
      - Y [0,1]
      - Z [0,1]
    - Home Position (Mutable)
      - Absolute Pan [0,1]
      - Absolute Tilt [0,1]
      - Absolute Camera Roll [0,1]
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Device Information SOAP Component
    - Equipment ID (Reportable)
    - Manufacturer (Reportable)

        - Model (Reportable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI
    - Health Publishing SOAP Component
        - Fault Update Period (Mutable)
        - Status Update Period (Mutable)
        - Status Enabled (Mutable)
        - Enabled Fault Severity Levels (Mutable)
            - Enabled Fault Severity Level [0,∞)
        - Data Transport Configurations (Mutable)
            - Transport Configuration [1,∞)
                - Unique Name
                - URI
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI
    - Configuration SOAP Component
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI
    - AA Entrypoints SOAP Component [0,1]
        - PES Location (Mutable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI
    - Authentication Entrypoints SOAP Component [0,1]
        - Authentication Service Location (Mutable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI

20.15.2 Camera Gimbal-Specific Configuration Description Content

1. ***Camera Gimbal Component***
    a. A Configuration Group element used to encapsulate the camera gimbal configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the camera gimbal.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be “CAMERA_GIMBAL”.
2. ***Camera Gimbal Device SOAP Component***
    a. A Configuration Group element used to encapsulate the camera gimbal device SOAP component.
    b. The ***name*** attribute shall be “soapComponent”.
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be “CAMERA_GIMBAL_DEVICE”.
3. ***Camera Gimbal Data SOAP Component***
    a. A Configuration Group element used to encapsulate the camera gimbal data SOAP component.
    b. The ***name*** attribute shall be “soapComponent”.
    c. This element shall contain a ***type*** attribute.

i. The ***type*** attribute shall be “CAMERA_GIMBAL_DATA”.

4. ***Relative Camera Orientation***
    a. A Configuration Group element used to encapsulate the relative orientation of the camera gimbal.
    b. The ***name*** attribute shall be “relativeCameraOrientation”.
    c. This element shall not contain a ***type*** attribute.
5. ***Pan***
    a. A Configuration Item element used to describe the pan of the camera gimbal.
    b. The ***name*** attribute shall be “pan”.
    c. The ***value*** attribute shall be a string representation of a real value.
    d. The ***value*** attribute shall be in units of arc degrees.
6. ***Tilt***
    a. A Configuration Item element used to describe the tilt of the camera gimbal.
    b. The ***name*** attribute shall be “tilt”.
    c. The ***value*** attribute shall be a string representation of a real value.
    d. The ***value*** attribute shall be in units of arc degrees.
7. ***Camera Roll***
    a. A Configuration Item element used to describe the roll of the camera gimbal.
    b. The ***name*** attribute shall be “cameraRoll”.
    c. The ***value*** attribute shall be a string representation of a real value.
    d. The ***value*** attribute shall be in units of arc degrees.
8. ***Relative Camera Position***
    a. A Configuration Group element used to encapsulate the relative position of the camera gimbal.
    b. The ***name*** attribute shall be “relativeCameraPosition”.
    c. This element shall not contain a ***type*** attribute.
9. ***X***
    a. A Configuration Item element used to describe the camera gimbal position X.
    b. The ***name*** attribute shall be “x”.
    c. The ***value*** attribute shall be a string representation of a real value.
    d. The ***value*** attribute shall be in units of meters.
10. ***Y***
    a. A Configuration Item element used to describe the camera gimbal position Y.
    b. The ***name*** attribute shall be “y”.
    c. The ***value*** attribute shall be a string representation of a real value.
    d. The ***value*** attribute shall be in units of meters.
11. ***Z***
    a. A Configuration Item element used to describe the camera gimbal position Z.
    b. The ***name*** attribute shall be “z”.
    c. The ***value*** attribute shall be a string representation of a real value.
    d. The ***value*** attribute shall be in units of meters.
12. ***Home Position***
    a. A Configuration Group element used to encapsulate the home position of the camera gimbal.
    b. The ***name*** attribute shall be “homePosition”.
    c. This element shall not contain a ***type*** attribute.
13. ***Absolute Pan***
    a. A Configuration Item element used to describe the absolute pan of the camera gimbal home position.
    b. The ***name*** attribute shall be “absolutePan”.
    c. The ***value*** attribute shall be a string representation of a real value.
    d. The ***value*** attribute shall be in units of arc degrees.
14. ***Absolute Tilt***
    a. A Configuration Item element used to describe the absolute tilt of the camera gimbal home position.
    b. The ***name*** attribute shall be “absoluteTilt”.
    c. The ***value*** attribute shall be a string representation of a real value.
    d. The ***value*** attribute shall be in units of arc degrees.
15. ***Absolute Camera Roll***
    a. A Configuration Item element used to describe the absolute camera roll of the camera gimbal home position.
    b. The ***name*** attribute shall be “absoluteCameraRoll”.
    c. The ***value*** attribute shall be a string representation of a real value.
    d. The ***value*** attribute shall be in units of arc degrees.
16. ***Camera Gimbal Data Types***

a. A Configuration Group element used to encapsulate a list of camera gimbal data types.
b. The ***name*** attribute shall be “cameraGimbalDataTypes”.
c. This element shall not contain a ***type*** attribute.

17. ***Camera Gimbal Data Type***
    a. A Configuration Group element used to encapsulate a camera gimbal data type.
    b. The ***name*** attribute shall be “cameraGimbalDataType”.
    c. The ***type*** attribute shall be a string defined in the “Data Type” column of a “Camera Gimbal” row in Table 298 in the appendix titled “VDM Data Type Values.”

20.15.3 Camera Gimbal Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Camera Gimbal is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="CAMERA_GIMBAL" name="9fc63b42-c9d3-4128-842e-1ba4be247abc">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup name="relatedComponents">
            <vcl:configGroup type="VIDEO_SENSOR" name="video-sensor-uuid-string">
                <vcl:configItem name="versionInformationUri"
                    value="http://192.168.1.10:53714/VideoSensor/versionInformation"/>
            </vcl:configGroup>
            <vcl:configGroup type="VIDEO_ENCODER" name="video-encoder-uuid-string">
                <vcl:configItem name="versionInformationUri"
                    value="http://192.168.1.10:53714/VideoEncoder/versionInformation"/>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configItem value="14" name="interfaceId"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/CameraGimbal/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="CAMERA_GIMBAL_DATA" name="soapComponent">
            <vcl:configItem value="1000" name="updatePeriod"/>
            <vcl:configItem value="100" name="minimumUpdatePeriod"/>
            <vcl:configGroup name="cameraGimbalDataTypes">
                <vcl:configGroup type="CAMERA_ORIENTATION" name="cameraGimbalDataType">
                    <vcl:configItem value="true" name="dataEnabled"/>
                    <vcl:configGroup name="dataTransportConfigurations">
                        <vcl:configGroup name="transportConfiguration">
                            <vcl:configItem value="c8794fae-7dec-11d0-a765-00a0c91e6b01" name="uniqueName"/>
                            <vcl:configItem value="multicast://239.192.1.1:10040" name="uri"/>
                        </vcl:configGroup>
```

```
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup type="HOME_POSITION" name="cameraGimbalDataType">
                <vcl:configItem value="true" name="dataEnabled"/>
                <vcl:configGroup name="dataTransportConfigurations">
                    <vcl:configGroup name="transportConfiguration">
                        <vcl:configItem value="ae194fae-7dec-11d0-a765-00a0c91e6b01" name="uniqueName"/>
                        <vcl:configItem value="multicast://239.192.1.1:10050" name="uri"/>
                    </vcl:configGroup>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup name="ports">
            <vcl:configGroup name="port">
                <vcl:configItem value="1.9" name="portStandardVersion"/>
                <vcl:configItem value="http://192.168.1.10:53714/CameraGimbal/cameraGimbalData" name="uri"/>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
    <vcl:configGroup type="CAMERA_GIMBAL_DEVICE" name="soapComponent">
        <vcl:configGroup name="relativeCameraOrientation">
            <vcl:configItem value="0" name="pan"/>
            <vcl:configItem value="0" name="cameraRoll"/>
            <vcl:configItem value="0" name="tilt"/>
        </vcl:configGroup>
        <vcl:configGroup name="relativeCameraPosition">
            <vcl:configItem value="0" name="x"/>
            <vcl:configItem value="0" name="y"/>
            <vcl:configItem value="0" name="z"/>
        </vcl:configGroup>
        <vcl:configGroup name="homePosition">
            <vcl:configItem value="0" name="absolutePan"/>
            <vcl:configItem value="0" name="absoluteTilt"/>
            <vcl:configItem value="0" name="absoluteCameraRoll"/>
        </vcl:configGroup>
        <vcl:configGroup name="ports">
            <vcl:configGroup name="port">
                <vcl:configItem value="1.9" name="portStandardVersion"/>
                <vcl:configItem value="http://192.168.1.10:53714/CameraGimbal/cameraGimbalDevice" name="uri"/>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
    <vcl:configGroup type="DEVICE_INFORMATION" name="soapComponent">
        <vcl:configItem value="1" name="equipmentId"/>
```

```
            <vcl:configItem value="axis" name="manufacturer"/>
            <vcl:configItem value="ptz" name="model"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/CameraGimbal/deviceInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="24748585-3277-11e3-aa6e-0800200c9a66" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10103" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/CameraGimbal/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/CameraGimbal/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
```

```
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/CameraGimbal/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/CameraGimbal/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.16 Threat Detection and Reporting Service Configuration Description <30015–20210409, Pro>

20.16.1 Threat Detection and Reporting Service Configuration Description Tree View

The tree view below shows the hierarchy of the Threat Detection and Reporting Service configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Threat Detection and Reporting Service Component
  - Interface ID (Configurable)
  - Announcement Enabled (Configurable)
  - Threat Detection and Reporting Data SOAP Component
    - Data Enabled (Mutable)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - AA Entrypoints SOAP Component [0,1]
    - PES Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Authentication Entrypoints SOAP Component [0,1]
    - Authentication Service Location (Mutable)

- Ports (Reportable)
  - Port [1,∞)
    - Port Standard Version
    - URI

20.16.2 Threat Detection and Reporting Service-Specific Configuration Description Content

1. ***Threat Detection and Reporting Service Component***
   a. A Configuration Group element used to encapsulate the threat detection and reporting service configuration description.
   b. The ***name*** attribute shall be a string representing the unique name of the threat detection and reporting service.
   c. The ***name*** attribute shall be unique within the scope of the VDB.

   > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

   d. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "THREAT_SERVICE".
2. ***Threat Detection and Reporting Data SOAP Component***
   a. A Configuration Group element used to encapsulate the threat detection and reporting data SOAP component.
   b. The ***name*** attribute shall be "soapComponent".
   c. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "THREAT_DATA".

20.16.3 Threat Detection and Reporting Service Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Threat Detection and Reporting Service is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="THREAT_SERVICE" name="fb748639-377b-4d76-9bd4-ff4869661f00">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configItem value="15" name="interfaceId"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/ThreatService/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7dec-11d0-a765-00a0c91e6bf9" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10122" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/ThreatService/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
```

```
        </vcl:configGroup>
        <vcl:configGroup type="THREAT_DATA" name="soapComponent">
            <vcl:configItem value="true" name="dataEnabled"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-11e7-11d0-a765-00a0c93e6bf4" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10007" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/ThreatService/threatData" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/ThreatService/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/ThreatService/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/ThreatService/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
```

```
    </vcl:configGroup>
</vcl:VCL>
```

20.17 Mission Recorder Configuration Description <30016–20210409, Pro>

20.17.1 Mission Recorder Configuration Description Tree View

The tree view below shows the hierarchy of the Mission Recorder configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Mission Recorder Component
  - Announcement Enabled (Configurable)
  - Mission Recorder SOAP Component
    - Data Streams
      - Data Stream [0,∞) (Mutable)
        - Unique Name
        - Past Data Enabled
        - Data Stream Types
          - Data Stream Type [1,∞)
        - Recording Transport Configuration
          - Transport Configuration
            - Unique Name
            - URI
        - Playback Transport Configuration
          - Transport Configuration [1,∞)
            - Unique Name
            - URI
        - Comments
          - Comment [0,∞)
            - Comment Unique Name
            - Comment Value
        - Recorded Files
          - Recorded File [0,∞)
            - File Name
            - Start Time
            - End Time
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Device Information SOAP Component
    - Equipment ID (Reportable)
    - Manufacturer (Reportable)
    - Model (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)

- Status Update Period (Mutable)
- Status Enabled (Mutable)
- Enabled Fault Severity Levels (Mutable)
  - Enabled Fault Severity Level [0,∞)
- Data Transport Configurations (Mutable)
  - Transport Configuration [1,∞)
    - Unique Name
    - URI
- Ports (Reportable)
  - Port [1,∞)
    - Port Standard Version
    - URI

- Configuration SOAP Component
  - Ports (Reportable)
    - Port [1,∞)
      - Port Standard Version
      - URI

- AA Entrypoints SOAP Component [0,1]
  - PES Location (Mutable)
  - Ports (Reportable)
    - Port [1,∞)
      - Port Standard Version
      - URI

- Authentication Entrypoints SOAP Component [0,1]
  - Authentication Service Location (Mutable)
  - Ports (Reportable)
    - Port [1,∞)
      - Port Standard Version
      - URI

20.17.2 Mission Recorder-Specific Configuration Description Content

1. ***Mission Recorder Component***
   a. A Configuration Group element used to encapsulate the mission recorder configuration description.
   b. The ***name*** attribute shall be a string representing the unique name of the mission recorder.
   c. The ***name*** attribute shall be unique within the scope of the VDB.

   > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

   d. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "MISSION_RECORDER".
2. ***Mission Recorder SOAP Component***
   a. A Configuration Group element used to encapsulate the mission recorder SOAP component.
   b. The ***name*** attribute shall be "soapComponent".
   c. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "MISSION_RECORDER".
3. ***Data Streams***
   a. A Configuration Group element used to encapsulate the data streams available to the mission recorder.
   b. The ***name*** attribute shall be "dataStreams".
   c. This element shall not contain a ***type*** attribute.
4. ***Data Stream***
   a. A Configuration Group element used to encapsulate a data stream available to the mission recorder.
   b. The ***name*** attribute shall be "dataStream".
   c. This element shall not contain a ***type*** attribute.
5. ***Data Stream Types***
   a. A Configuration Group element used to encapsulate the data types of the data stream.
   b. The ***name*** attribute shall be "dataStreamTypes".

   c. This element shall not contain a ***type*** attribute.
6. ***Data Stream Type***
   a. A Configuration Item element used to identify a single data stream type.
   b. The ***name*** attribute shall be "dataStreamType".
   c. The ***value*** attribute shall be a string chosen from one of the following:
      i. STD_DEFINITION_STREAMING_VIDEO
      ii. STREAMING_AUDIO
      iii. The "Data Type" column of Table 298 in the appendix titled "VDM Data Type Values."

      > ***NOTE:*** *Mission recorder does not support experimental data streams. Thus, any VDM inside Table 298 with a "Message Value" column that begins with "0xF" cannot be selected as a mission recorder data stream.*
7. ***Recording Transport Configuration***
   a. A Configuration Group element used to describe the data stream transport configuration used by the mission recorder to record data from the VDB.
   b. The ***name*** attribute shall be "recordingTransportConfiguration".
   c. This element shall not contain a ***type*** attribute.
8. ***Playback Transport Configuration***
   a. A Configuration Group element used to describe the data stream transport configuration used by the mission recorder for playing back recorded data on the VDB.
   b. The ***name*** attribute shall be "playbackTransportConfiguration".
   c. This element shall not contain a ***type*** attribute.
9. ***Past Data Enabled***
   a. A Configuration Item element used to describe whether or not the data coming from the data stream can be stored even when not being recorded by the mission recorder.
   b. The ***name*** attribute shall be "pastDataEnabled".
   c. The ***value*** attribute shall be a case-sensitive string representation of a Boolean value where "true" or "1" means the data can be stored and "false" or "0" means the data cannot be stored.
10. ***Comments***
    a. A Configuration Group element used to encapsulate comments about the data stream.
    b. The ***name*** attribute shall be "comments".
    c. This element shall not contain a ***type*** attribute.
11. ***Comment***
    a. A Configuration Group element used to encapsulate a comment and the comment's unique name.
    b. The ***name*** attribute shall be "comment".
    c. This element shall not contain a ***type*** attribute.
12. ***Comment Value***
    a. A Configuration Item element used to encapsulate a text of the comment.
    b. The ***name*** attribute shall be "commentValue".
    c. The ***value*** attribute shall be a string.
13. ***Comment Unique Name***
    a. A Configuration Item element used to encapsulate the unique name of the comment.
    b. The ***name*** attribute shall be "commentUniqueName".
    c. The ***value*** attribute shall be a string.
    d. The ***value*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*
14. ***Recorded Files***
    a. A Configuration Group element used to encapsulate the files of the data stream.
    b. The ***name*** attribute shall be "recordedFiles".
    c. This element shall not contain a ***type*** attribute.
15. ***Recorded File***
    a. A Configuration Group element used to encapsulate a file of the data stream.
    b. The ***name*** attribute shall be "recordedFile".
    c. This element shall not contain a ***type*** attribute.
16. ***File Name***
    a. A Configuration Item element used to encapsulate the name of the file.
    b. The ***name*** attribute shall be "fileName".
    c. The ***value*** attribute shall be a string.

17. ***Start Time***
    a. A Configuration Item element used to encapsulate the start time of the file recording.
    b. The ***name*** attribute shall be “startTime”.
    c. The ***value*** attribute shall be a string representation of a timestamp.
    d. The timestamp shall be in the format as described in the section titled “Timestamp Format for Application Layer Data.”
18. ***End Time***
    a. A Configuration Item element used to encapsulate the end time of the file recording.
    b. The ***name*** attribute shall be “endTime”.
    c. The ***value*** attribute shall be a string representation of a timestamp.
    d. The timestamp shall be in the format as described in the section titled “Timestamp Format for Application Layer Data.”

20.17.3 Mission Recorder Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Mission Recorder is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="MISSION_RECORDER" name="e710829d-4409-4e87-88d0-6ce819b80821">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/MissionRecoder/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="DEVICE_INFORMATION" name="soapComponent">
            <vcl:configItem value="1" name="equipmentId"/>
            <vcl:configItem value="Manufacturer1" name="manufacturer"/>
            <vcl:configItem value="Model1" name="model"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/MissionRecorder/deviceInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="9c09c047-e4b1-4276-8ff5-7c8e090141b3" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10117" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
```

```
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/MissionRecorder/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="MISSION_RECORDER" name="soapComponent">
            <vcl:configGroup name="dataStreams">
                <vcl:configGroup name="dataStream">
                    <vcl:configItem value="stream1" name="dataStreamUniqueName"/>
                    <vcl:configItem value="true" name="pastDataEnabled"/>
                    <vcl:configGroup name="dataStreamTypes">
                        <vcl:configItem value="POSITION" name="dataStreamType"/>
                    </vcl:configGroup>
                    <vcl:configGroup name="recordingTransportConfiguration">
                        <vcl:configGroup name="transportConfiguration">
                            <vcl:configItem value="fc09c047-e4b1-4276-8ff5-7c8e09022222" name="uniqueName"/>
                            <vcl:configItem value="multicast://239.192.1.1:10202" name="uri"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
                    <vcl:configGroup name="playbackTransportConfiguration">
                        <vcl:configGroup name="transportConfiguration">
                            <vcl:configItem value="aaaaa047-e4b1-4276-8ff5-7c8e09011111" name="uniqueName"/>
                            <vcl:configItem value="multicast://239.192.1.1:10201" name="uri"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
                    <vcl:configGroup name="comments">
                        <vcl:configGroup name="comment">
                            <vcl:configItem value="comment1" name="commentUniqueName"/>
                            <vcl:configItem value="I hope there are VDMs on this address" name="commentValue"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
                    <vcl:configGroup name="recordedFiles">
                        <vcl:configGroup name="recordedFile">
                            <vcl:configItem value="file1.txt" name="fileName"/>
                            <vcl:configItem value="" name="startTime"/>
                            <vcl:configItem value="" name="endTime"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
```

```
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/MissionRecorder/missionRecorderDevice" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/MissionRecorder/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/MissionRecorder/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/MissionRecorder/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.18 Data Logger Configuration Description <30049–20210409, Pro>

20.18.1 Data Logger Configuration Description Tree View

The tree view below shows the hierarchy of the Data Logger configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Data Logger Component
  - Interface ID (Configurable)
  - Announcement Enabled (Configurable)
  - Data Logger SOAP Component
    - Overwrite Enabled (Mutable)
    - Maximum Log Size (Mutable)
      - Size
      - Units
    - Streams (Mutable)
      - Stream [0,∞)
        - Unique Name
        - URI
        - Stream Type
          - VDM Stream [0,1]
            - VDM Type [0,1]
            - Interface ID [0,1]
          - Health Stream [0,1]
        - Time Filter [0,1]
          - Start Time [0,1]
          - End Time [0,1]
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - VICTORY Data SOAP Component
    - Data Enabled (Mutable)
    - Update Period (Mutable)
    - Minimum Update Period (Reportable)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)

- Enabled Fault Severity Levels (Mutable)
  - Enabled Fault Severity Level [0,∞)
- Data Transport Configurations (Mutable)
  - Transport Configuration [1,∞)
    - Unique Name
    - URI
- Ports (Reportable)
  - Port [1,∞)
    - Port Standard Version
    - URI

◦ Configuration SOAP Component
- Ports (Reportable)
  - Port [1,∞)
    - Port Standard Version
    - URI

◦ AA Entrypoints SOAP Component [0,1]
- PES Location (Mutable)
- Ports (Reportable)
  - Port [1,∞)
    - Port Standard Version
    - URI

◦ Authentication Entrypoints SOAP Component [0,1]
- Authentication Service Location (Mutable)
- Ports (Reportable)
  - Port [1,∞)
    - Port Standard Version
    - URI

20.18.2 Data Logger-Specific Configuration Description Content

1. ***Data Logger Component***
   a. A Configuration Group element used to encapsulate the data logger configuration description.
   b. The ***name*** attribute shall be a string representing the unique name of the data logger.
   c. The ***name*** attribute shall be unique within the scope of the VDB.

   > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

   d. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "DATA_LOGGER".
2. ***Data Logger SOAP Component***
   a. A Configuration Group element used to encapsulate the data logger SOAP component.
   b. The ***name*** attribute shall be "soapComponent".
   c. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "DATA_LOGGER".
3. ***Overwrite Enabled***
   a. A Configuration Item element used to describe whether the data logger will overwrite when the maximum log size is reached.
   b. The ***name*** attribute shall be "overwriteEnabled".
   c. The ***value*** attribute shall be a case-sensitive string representation of a Boolean value where "true" or "1" means the component is sending status messages and "false" or "0" means the component is not sending status messages.
4. ***Maximum Log Size***
   a. A Configuration Group element used to describe the maximum log size of the data logger.
      i. The maximum log size is the maximum amount of disk space the data logger will use to store log messages.
   b. The ***name*** attribute shall be "maximumLogSize".
   c. This element shall not contain a ***type*** attribute.

5. ***Streams***
    a. A Configuration Group element used to encapsulate the streams being logged by the data logger.
    b. The ***name*** attribute shall be “streams”.
    c. This element shall not contain a ***type*** attribute.
    d. Not including the Streams Configuration Group would remove all streams from the data logger.
6. ***Stream***
    a. A Configuration Group element used to encapsulate a stream being logged by the data logger.
    b. The ***name*** attribute shall be “stream”.
    c. This element shall not contain a ***type*** attribute.
7. ***Stream Type***
    a. A Configuration Group element used to encapsulate the type of data being logged from the stream.
    b. The ***name*** attribute shall be “streamType”.
    c. This element shall not contain a ***type*** attribute.
    d. The stream type shall contain either a VDM Stream Configuration Group or a Health Stream Configuration Item.
8. ***VDM Stream***
    a. A Configuration Group element used to encapsulate the types of VDM data being logged from the stream.
    b. The ***name*** attribute shall be “vdmStream”.
    c. This element shall not contain a ***type*** attribute.
    d. The VDM stream shall contain either a VDM Type Configuration Item or an Interface ID Configuration Item.
9. ***VDM Type***
    a. A Configuration Item element used to identify the type of VDM being logged from the stream.
    b. The ***name*** attribute shall be “vdmType”.
    c. The ***value*** attribute shall be a string representation of the “Data Type” column of Table 298 in the appendix titled “VDM Data Type Values.”
10. ***Interface ID***
    a. A Configuration Item element used to describe the interface ID of the messages being logged from the stream.
    b. The ***name*** attribute shall be “interfaceId”.
    c. The ***value*** attribute shall be a string representation of an integer greater than or equal to zero.
11. ***Health Stream***
    a. A Configuration Item element used to identify the type of health messages to capture from the stream.
    b. The ***name*** attribute shall be “healthStream”.
    c. The ***value*** attribute shall be a string defined as “syslog,” a relative object identifier (OID), or an absolute OID.
12. ***Time Filter***
    a. A Configuration Group element used to encapsulate the start time and/or end time of the logged stream.
    b. The ***name*** attribute shall be “timeFilter”.
    c. This element shall not contain a ***type*** attribute.
13. ***Start Time***
    a. A Configuration Item element used to specify the start time the data logger began logging messages on the stream.
    b. The ***name*** attribute shall be “startTime”.
    c. The ***value*** attribute shall be a string representation of a timestamp.
    d. The timestamp shall be in the format as described in the section titled “Timestamp Format for Application Layer Data.”
14. ***End Time***
    a. A Configuration Item element used to specify the end time the data logger stopped logging messages on the stream.
    b. The ***name*** attribute shall be “endTime”.
    c. The ***value*** attribute shall be a string representation of a timestamp.
    d. The timestamp shall be in the format as described in the section titled “Timestamp Format for Application Layer Data.”
15. ***Size***
    a. A Configuration Item element used to specify the quantity of the storage size.
    b. The ***name*** attribute shall be “size”.
    c. The ***value*** attribute shall be a real value greater than zero.
16. ***Units***
    a. A Configuration Item element used to specify the units of the storage size.
    b. The ***name*** attribute shall be “units”.
    c. The ***value*** attribute shall be a string chosen from one of the following values:

i. B
ii. KB
iii. MB
iv. GB
v. TB

20.18.3 Data Logger Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Data Logger is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup name="datalogger1" type="DATA_LOGGER">
        <vcl:configItem name="announcementEnabled" value="false"/>
        <vcl:configItem name="interfaceId" value="99"/>
        <vcl:configGroup name="soapComponent" type="CONFIGURATION">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem name="portStandardVersion" value="1.9"/>
                    <vcl:configItem name="uri" value="http://192.168.1.10:53714/DataLogger/configuration"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup name="soapComponent" type="DATA_LOGGER">
            <vcl:configItem name="overwriteEnabled" value="false"/>
            <vcl:configGroup name="maximumLogSize">
                <vcl:configItem name="size" value="30"/>
                <vcl:configItem name="units" value="MB"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem name="portStandardVersion" value="1.9"/>
                    <vcl:configItem name="uri" value="http://192.168.1.10:53714/DataLogger/dataLogger"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="streams">
                <vcl:configGroup name="stream">
                    <vcl:configItem name="uniqueName" value="PositionVDM"/>
                    <vcl:configItem name="uri" value="multicast://239.192.1.1:10001"/>
                    <vcl:configGroup name="streamType">
                        <vcl:configGroup name="vdmStream">
                            <vcl:configItem name="vdmType" value="POSITION"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
```

```
            </vcl:configGroup>
            <vcl:configGroup name="stream">
                <vcl:configItem name="uniqueName" value="PositionHealth"/>
                <vcl:configItem name="uri" value="multicast://239.192.1.1:10003"/>
                <vcl:configGroup name="streamType">
                    <vcl:configItem name="healthStream" value="syslog"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
    <vcl:configGroup name="soapComponent" type="HEALTH_PUBLISHING">
        <vcl:configItem name="faultUpdatePeriod" value="1000"/>
        <vcl:configItem name="statusEnabled" value="true"/>
        <vcl:configItem name="statusUpdatePeriod" value="1000"/>
        <vcl:configGroup name="dataTransportConfigurations">
            <vcl:configGroup name="transportConfiguration">
                <vcl:configItem name="uniqueName" value="24748585-3299-11e3-aa6e-0800200c9a66"/>
                <vcl:configItem name="uri" value="multicast://239.192.1.1:10150"/>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup name="enabledFaultSeverityLevels">
            <vcl:configItem name="enabledFaultSeverityLevel" value="EMERGENCY"/>
            <vcl:configItem name="enabledFaultSeverityLevel" value="ALERT"/>
            <vcl:configItem name="enabledFaultSeverityLevel" value="CRITICAL"/>
            <vcl:configItem name="enabledFaultSeverityLevel" value="ERROR"/>
            <vcl:configItem name="enabledFaultSeverityLevel" value="WARNING"/>
        </vcl:configGroup>
        <vcl:configGroup name="ports">
            <vcl:configGroup name="port">
                <vcl:configItem name="portStandardVersion" value="1.9"/>
                <vcl:configItem name="uri" value="http://192.168.1.10:53714/DataLogger/healthPublishing"/>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
    <vcl:configGroup name="soapComponent" type="VERSION_INFORMATION">
        <vcl:configItem name="interfaceStandardVersion" value="1.9"/>
        <vcl:configGroup name="ports">
            <vcl:configGroup name="port">
                <vcl:configItem name="portStandardVersion" value="1.9"/>
                <vcl:configItem name="uri" value="http://192.168.1.10:53714/DataLogger/versionInformation"/>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
    <vcl:configGroup name="soapComponent" type="VICTORY_DATA">
```

```
            <vcl:configItem name="dataEnabled" value="true"/>
            <vcl:configItem name="minimumUpdatePeriod" value="100"/>
            <vcl:configItem name="updatePeriod" value="5000"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem name="uniqueName" value="a81d4fae-11ec-11d5-a765-00a0c93e6bf4"/>
                    <vcl:configItem name="uri" value="multicast://239.192.1.1:10050"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem name="portStandardVersion" value="1.9"/>
                    <vcl:configItem name="uri" value="http://192.168.1.10:53714/DataLogger/victoryData"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DataLogger/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DataLogger/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.19 Automotive System Configuration Description <30019–20210409, Pro>

20.19.1 Automotive System Configuration Description Tree View

The tree view below shows the hierarchy of the Automotive System configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Automotive System Component
  - Interface ID (Configurable)
  - Announcement Enabled (Configurable)
  - Automotive Data SOAP Component
    - Automotive Data Types
      - Automotive Data Type [1,∞)
        - Data of Interest (Mutable)
        - Data Enabled (Mutable)
        - Update Period (Mutable)
        - Minimum Update Period (Reportable)
        - Data Transport Configurations (Mutable)
          - Transport Configuration [1,∞)
            - Unique Name
            - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - AA Entrypoints SOAP Component [0,1]
    - PES Location (Mutable)
    - Ports (Reportable)

        - Port [1,∞)
            - Port Standard Version
            - URI
    - Authentication Entrypoints SOAP Component [0,1]
        - Authentication Service Location (Mutable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI

20.19.2 Automotive System-Specific Configuration Description Content

1. ***Automotive System Component***
    a. A Configuration Group element used to encapsulate the automotive system configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the automotive system.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "AUTOMOTIVE_SYSTEM".
2. ***Automotive Data SOAP Component***
    a. A Configuration Group element used to encapsulate the automotive data SOAP component.
    b. The ***name*** attribute shall be "soapComponent".
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "AUTOMOTIVE_DATA".
3. ***Automotive Data Types***
    a. A Configuration Group element used to encapsulate a list of automotive data types.

    > ***NOTE:*** *All automotive data types not encapsulated inside this element are not supported by the automotive system.*

    b. The ***name*** attribute shall be "automotiveDataTypes".
    c. This element shall not contain a ***type*** attribute.
4. ***Automotive Data Type***
    a. A Configuration Group element used to encapsulate a particular automotive data type.
    b. The ***name*** attribute shall be "automotiveDataType".
    c. The ***type*** attribute shall be a string defined in the "Data Type" column of an "Automotive System" row in Table 298 in the appendix titled "VDM Data Type Values."

20.19.3 Automotive System Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing an Automotive System is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="AUTOMOTIVE_SYSTEM" name="be5a5172-6833-457d-95ff-199030492514">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configItem value="22" name="interfaceId"/>
        <vcl:configGroup type="AUTOMOTIVE_DATA" name="soapComponent">
            <vcl:configGroup name="automotiveDataTypes">
                <vcl:configGroup type="AMBIENT_CONDITIONS" name="automotiveDataType">
                    <vcl:configItem value="true" name="dataOfInterest"/>
                    <vcl:configItem value="true" name="dataEnabled"/>
                    <vcl:configItem value="1000" name="updatePeriod"/>
                    <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                    <vcl:configGroup name="dataTransportConfigurations">
                        <vcl:configGroup name="transportConfiguration">
                            <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efeaaa" name="uniqueName"/>
                            <vcl:configItem value="multicast://239.192.1.1:20051" name="uri"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
                </vcl:configGroup>
                <vcl:configGroup type="DASH_DISPLAY" name="automotiveDataType">
                    <vcl:configItem value="true" name="dataOfInterest"/>
                    <vcl:configItem value="true" name="dataEnabled"/>
                    <vcl:configItem value="1000" name="updatePeriod"/>
                    <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                    <vcl:configGroup name="dataTransportConfigurations">
                        <vcl:configGroup name="transportConfiguration">
                            <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efeaab" name="uniqueName"/>
                            <vcl:configItem value="multicast://239.192.1.1:20052" name="uri"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
                </vcl:configGroup>
                <vcl:configGroup type="DISTANCE" name="automotiveDataType">
                    <vcl:configItem value="true" name="dataOfInterest"/>
                    <vcl:configItem value="true" name="dataEnabled"/>
```

```
                    <vcl:configItem value="1000" name="updatePeriod"/>
                    <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                    <vcl:configGroup name="dataTransportConfigurations">
                        <vcl:configGroup name="transportConfiguration">
                            <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efeaac" name="uniqueName"/>
                            <vcl:configItem value="multicast://239.192.1.1:20053" name="uri"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
                </vcl:configGroup>
                <vcl:configGroup type="ELECTRICAL_POWER_1" name="automotiveDataType">
                    <vcl:configItem value="true" name="dataOfInterest"/>
                    <vcl:configItem value="true" name="dataEnabled"/>
                    <vcl:configItem value="1000" name="updatePeriod"/>
                    <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                    <vcl:configGroup name="dataTransportConfigurations">
                        <vcl:configGroup name="transportConfiguration">
                            <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efeaad" name="uniqueName"/>
                            <vcl:configItem value="multicast://239.192.1.1:20054" name="uri"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
                </vcl:configGroup>
                <vcl:configGroup type="ELECTRICAL_POWER_2" name="automotiveDataType">
                    <vcl:configItem value="true" name="dataOfInterest"/>
                    <vcl:configItem value="true" name="dataEnabled"/>
                    <vcl:configItem value="1000" name="updatePeriod"/>
                    <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                    <vcl:configGroup name="dataTransportConfigurations">
                        <vcl:configGroup name="transportConfiguration">
                            <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efeaae" name="uniqueName"/>
                            <vcl:configItem value="multicast://239.192.1.1:20055" name="uri"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
                </vcl:configGroup>
                <vcl:configGroup type="ENGINE_FLUID_LEVEL_AND_PRESSURE_1" name="automotiveDataType">
                    <vcl:configItem value="true" name="dataOfInterest"/>
                    <vcl:configItem value="true" name="dataEnabled"/>
                    <vcl:configItem value="1000" name="updatePeriod"/>
                    <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                    <vcl:configGroup name="dataTransportConfigurations">
                        <vcl:configGroup name="transportConfiguration">
                            <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efeaaf" name="uniqueName"/>
                            <vcl:configItem value="multicast://239.192.1.1:20056" name="uri"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
```

```
            </vcl:configGroup>
            <vcl:configGroup type="ENGINE_HOURS_AND_REVOLUTIONS" name="automotiveDataType">
                <vcl:configItem value="true" name="dataOfInterest"/>
                <vcl:configItem value="true" name="dataEnabled"/>
                <vcl:configItem value="1000" name="updatePeriod"/>
                <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                <vcl:configGroup name="dataTransportConfigurations">
                    <vcl:configGroup name="transportConfiguration">
                        <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efeaa0" name="uniqueName"/>
                        <vcl:configItem value="multicast://239.192.1.1:20057" name="uri"/>
                    </vcl:configGroup>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup type="ENGINE_SPEED_AND_LOAD_FACTOR" name="automotiveDataType">
                <vcl:configItem value="true" name="dataOfInterest"/>
                <vcl:configItem value="true" name="dataEnabled"/>
                <vcl:configItem value="1000" name="updatePeriod"/>
                <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                <vcl:configGroup name="dataTransportConfigurations">
                    <vcl:configGroup name="transportConfiguration">
                        <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efeaa1" name="uniqueName"/>
                        <vcl:configItem value="multicast://239.192.1.1:20058" name="uri"/>
                    </vcl:configGroup>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup type="ENGINE_TEMPERATURE_1" name="automotiveDataType">
                <vcl:configItem value="true" name="dataOfInterest"/>
                <vcl:configItem value="true" name="dataEnabled"/>
                <vcl:configItem value="1000" name="updatePeriod"/>
                <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                <vcl:configGroup name="dataTransportConfigurations">
                    <vcl:configGroup name="transportConfiguration">
                        <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efeaa2" name="uniqueName"/>
                        <vcl:configItem value="multicast://239.192.1.1:20059" name="uri"/>
                    </vcl:configGroup>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup type="ENGINE_TEMPERATURE_2" name="automotiveDataType">
                <vcl:configItem value="true" name="dataOfInterest"/>
                <vcl:configItem value="true" name="dataEnabled"/>
                <vcl:configItem value="1000" name="updatePeriod"/>
                <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                <vcl:configGroup name="dataTransportConfigurations">
                    <vcl:configGroup name="transportConfiguration">
```

```
                    <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efeaa4" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:20061" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="FUEL_CONSUMPTION" name="automotiveDataType">
            <vcl:configItem value="true" name="dataOfInterest"/>
            <vcl:configItem value="true" name="dataEnabled"/>
            <vcl:configItem value="1000" name="updatePeriod"/>
            <vcl:configItem value="100" name="minimumUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efeaa5" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:20062" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="FUEL_ECONOMY" name="automotiveDataType">
            <vcl:configItem value="true" name="dataOfInterest"/>
            <vcl:configItem value="true" name="dataEnabled"/>
            <vcl:configItem value="1000" name="updatePeriod"/>
            <vcl:configItem value="100" name="minimumUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efeaa6" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:20063" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="ENGINE_VALVE" name="automotiveDataType">
            <vcl:configItem value="true" name="dataOfInterest"/>
            <vcl:configItem value="true" name="dataEnabled"/>
            <vcl:configItem value="1000" name="updatePeriod"/>
            <vcl:configItem value="100" name="minimumUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efeaa7" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:20064" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="DIRECT_LAMP_CONTROL_DATA_1" name="automotiveDataType">
            <vcl:configItem value="true" name="dataOfInterest"/>
            <vcl:configItem value="true" name="dataEnabled"/>
```

```
                <vcl:configItem value="1000" name="updatePeriod"/>
                <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                <vcl:configGroup name="dataTransportConfigurations">
                    <vcl:configGroup name="transportConfiguration">
                        <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efeaa8" name="uniqueName"/>
                        <vcl:configItem value="multicast://239.192.1.1:20065" name="uri"/>
                    </vcl:configGroup>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup type="DIRECT_LAMP_CONTROL_DATA_2" name="automotiveDataType">
                <vcl:configItem value="true" name="dataOfInterest"/>
                <vcl:configItem value="true" name="dataEnabled"/>
                <vcl:configItem value="1000" name="updatePeriod"/>
                <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                <vcl:configGroup name="dataTransportConfigurations">
                    <vcl:configGroup name="transportConfiguration">
                        <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efeaa9" name="uniqueName"/>
                        <vcl:configItem value="multicast://239.192.1.1:20066" name="uri"/>
                    </vcl:configGroup>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup type="ELECTRONIC_TRANSMISSION_CONTROLLER_2" name="automotiveDataType">
                <vcl:configItem value="true" name="dataOfInterest"/>
                <vcl:configItem value="true" name="dataEnabled"/>
                <vcl:configItem value="1000" name="updatePeriod"/>
                <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                <vcl:configGroup name="dataTransportConfigurations">
                    <vcl:configGroup name="transportConfiguration">
                        <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efea1a" name="uniqueName"/>
                        <vcl:configItem value="multicast://239.192.1.1:20067" name="uri"/>
                    </vcl:configGroup>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup name="ports">
            <vcl:configGroup name="port">
                <vcl:configItem value="1.9" name="portStandardVersion"/>
                <vcl:configItem value="http://192.168.1.10:53714/AutomotiveSystem/automotiveData" name="uri"/>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
    <vcl:configGroup type="CONFIGURATION" name="soapComponent">
        <vcl:configGroup name="ports">
            <vcl:configGroup name="port">
```

```
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/AutomotiveSystem/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="ebd24b09-90b6-4dc0-8b30-524a69efeb71" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10139" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/AutomotiveSystem/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/AutomotiveSystem/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/AutomotiveSystem/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
```

```
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/AutomotiveSystem/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.20 Power Distribution System Configuration Description <30020–20210409, Pro>

20.20.1 Power Distribution System Configuration Description Tree View

The tree view below shows the hierarchy of the Power Distribution System configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Power Distribution System Component
  - Interface ID (Configurable)
  - Announcement Enabled (Configurable)
  - Power Distribution System Data SOAP Component
    - Power Distribution System Data Types
      - Power Distribution System Data Type [1,∞)
        - Device ID [0,1] (Reportable)
        - Device Type [0,1] (Reportable)
        - Battery ID [0,1] (Reportable)
        - Name [0,1] (Reportable)
        - APU Name [0,1] (Reportable)
        - Data of Interest (Mutable)
        - Data Enabled (Mutable)
        - Update Period (Mutable)
        - Minimum Update Period (Reportable)
        - Data Transport Configurations (Mutable)
          - Transport Configuration [1,∞)
            - Unique Name
            - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Power Distribution System Device SOAP Component
    - Power Controller Devices
      - Power Controller Device [0,∞)
        - Power Controller Device ID (Reportable)
        - Enabled (Mutable)
    - Auxiliary Power Unit [0,1]
      - Enabled (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)

- Data Transport Configurations (Mutable)
  - Transport Configuration [1,∞)
    - Unique Name
    - URI
- Ports (Reportable)
  - Port [1,∞)
    - Port Standard Version
    - URI

- Configuration SOAP Component
  - Ports (Reportable)
    - Port [1,∞)
      - Port Standard Version
      - URI

- AA Entrypoints SOAP Component [0,1]
  - PES Location (Mutable)
  - Ports (Reportable)
    - Port [1,∞)
      - Port Standard Version
      - URI

- Authentication Entrypoints SOAP Component [0,1]
  - Authentication Service Location (Mutable)
  - Ports (Reportable)
    - Port [1,∞)
      - Port Standard Version
      - URI

20.20.2 Power Distribution System-Specific Configuration Description Content

1. ***Power Distribution System Component***
   a. A Configuration Group element used to encapsulate the power distribution system configuration description.
   b. The ***name*** attribute shall be a string representing the unique name of the power distribution system.
   c. The ***name*** attribute shall be unique within the scope of the VDB.

   > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

   d. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "POWER_DISTRIBUTION_SYSTEM".
2. ***Power Distribution System Data SOAP Component***
   a. A Configuration Group element used to encapsulate the power distribution system data SOAP component.
   b. The ***name*** attribute shall be "soapComponent".
   c. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "POWER_DISTRIBUTION_SYSTEM_DATA".
3. ***Power Distribution System Device SOAP Component***
   a. A Configuration Group element used to encapsulate the power distribution system device SOAP component.
   b. The ***name*** attribute shall be "soapComponent".
   c. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "POWER_DISTRIBUTION_SYSTEM_DEVICE".
4. ***Power Distribution System Data Types***
   a. A Configuration Group element used to encapsulate a list of power distribution system data types.

   > ***NOTE:*** *All power distribution system data types not encapsulated inside this element are not supported by the power distribution system.*

   b. The ***name*** attribute shall be "powerDataTypes".
   c. This element shall not contain a ***type*** attribute.
5. ***Power Distribution System Data Type***
   a. A Configuration Group element used to encapsulate a particular power distribution system data type.
   b. The ***name*** attribute shall be "powerDataType".

c. The ***type*** attribute shall be a string defined in the "Data Type" column of a "Power Distribution System" row in Table 298 in the appendix titled "VDM Data Type Values."

6. ***Device ID***
   a. A Configuration Item element used to describe the unique identifier of a device on the power distribution system.
   b. The Device ID Configuration Item element shall be provided for the following power data types:
      i. POWER_SYSTEM_DEVICE
      ii. POWER_SYSTEM_DEVICE_THRESHOLD
   c. The Device ID Configuration Item element shall not be provided for all other power data types.
   d. The ***name*** attribute shall be "deviceId".
   e. The ***value*** attribute shall be a string representing the unique identifier of the device.
   f. The ***value*** attribute shall be unique within the scope of the VDB.

   NOTE: *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

7. ***Device Type***
   a. A Configuration Item element used to describe the type of device on the power distribution system.
   b. The Device Type Configuration Item element shall be provided for the following power data types:
      i. POWER_SYSTEM_DEVICE
      ii. POWER_SYSTEM_DEVICE_THRESHOLD
   c. The Device Type Configuration Item element shall not be provided for all other power data types.
   d. The ***name*** attribute shall be "deviceType".
   e. The ***value*** attribute shall be a string chosen from one of the following values:
      i. POWER_CONTROLLER
      ii. VIRTUAL_DEVICE
      iii. DEVICE_POWERED_BY_SYSTEM_DEVICE

8. ***Battery ID***
   a. A Configuration Item element used to describe the unique identifier of a battery on the power distribution system.
   b. The Battery ID Configuration Item element shall be provided for the following power data types:
      i. PLATFORM_BATTERY
   c. The Battery ID Configuration Item element shall not be provided for all other power data types.
   d. The ***name*** attribute shall be "batteryId".
   e. The ***value*** attribute shall be a string representing the unique identifier of the battery.
   f. The ***value*** attribute shall be unique within the scope of the VDB.

   NOTE: *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

9. ***Power Controller Device ID***
   a. A Configuration Item element used to describe the unique identifier of a power controller device on the power distribution system.
   b. The ***name*** attribute shall be "powerControllerDeviceId".
   c. The ***value*** attribute shall be a string representing the unique identifier of the power controller device.
   d. The ***value*** attribute shall be unique within the scope of the VDB.

   NOTE: *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

10. ***Power Controller Device Enabled***
    a. A Configuration Item element used to describe whether or not the power controller device is enabled.
    b. The ***name*** attribute shall be "enabled".
    c. The ***value*** attribute shall be a case-sensitive string representation of a Boolean value where:
       i. "False" or "0" means:
          1. Power mode status is set to OFF for the power controller device.
             a. Power mode status is defined in the section titled "Power System Device" under "Power Distribution System Interface Complex Types."
          2. All power outputs drop to zero.
       ii. "True" or "1" means:
          1. Power mode status is set to ON for the power controller device.
             a. Power mode status is defined in the section titled "Power System Device" under "Power Distribution System Interface Complex Types."

2. All power outputs are enabled.

11. ***Power Controller Devices***
    a. A Configuration Group element used to encapsulate a list of power controller devices.
    b. The ***name*** attribute shall be “powerControllerDevices”.
    c. This element shall not contain a ***type*** attribute.
12. ***Power Controller Device***
    a. A Configuration Group element used to encapsulate a single power controller device.
    b. The ***name*** attribute shall be “powerContollerDevice”.
    c. This element shall not contain a ***type*** attribute.
13. ***Auxiliary Power Unit***
    a. A Configuration Group element used to encapsulate an auxiliary power unit (APU).
    b. The ***name*** attribute shall be “apu”.
    c. This element shall not contain a ***type*** attribute.
14. ***APU Enabled***
    a. A Configuration Item element used to describe whether or not the APU is enabled.
    b. The ***name*** attribute shall be “enabled”.
    c. The ***value*** attribute shall be a case-sensitive string representation of a Boolean value where:
        i. “False” or “0” means:
            1. Output enabled is set to “false” for the APU.
                a. Output enabled is defined in the section titled “Auxiliary Power Unit” under “Power Distribution System Interface Complex Types.”
            2. All power outputs drop to zero.
        ii. “True” or “1” means:
            1. Output enabled is set to “true” for the APU.
                a. Output enabled is defined in the section titled “Auxiliary Power Unit” under “Power Distribution System Interface Complex Types.”
            2. All power outputs are enabled.
15. ***Name***
    a. A Configuration Item element used to describe the human-readable name of a device, battery, or generator on the power distribution system.
    b. The Name Configuration Item element shall be provided for the following power data types:
        i. POWER_SYSTEM_DEVICE
        ii. POWER_SYSTEM_DEVICE_THRESHOLD
        iii. PLATFORM_BATTERY
        iv. PLATFORM_POWER_GENERATION
    c. The Name Configuration Item element shall not be provided for the PLATFORM_POWER power data type.
    d. For PLATFORM_POWER_GENERATION, the Name Configuration Item element shall represent the human-readable name of the generator.
    e. The ***name*** attribute shall be “name”.
    f. The ***value*** attribute shall be a string.
16. ***APU Name***
    a. A Configuration Item element used to describe the human-readable name of the APU on the power distribution system.
    b. If the power distribution system contains an APU, the APU Name Configuration Item element shall be provided for the PLATFORM_POWER_GENERATION power data type.
    c. If the power distribution system does not contain an APU, the APU Name Configuration Item element shall not be provided for the PLATFORM_POWER_GENERATION power data type.
    d. The APU Name Configuration Item element shall not be provided for all other power data types.
    e. The ***name*** attribute shall be “apuName”.
    f. The ***value*** attribute shall be a string.

20.20.3 Power Distribution System Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Power Distribution System is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="POWER_DISTRIBUTION_SYSTEM" name="ab8855e0-35ad-11e3-aa6e-0800200c9a00">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configItem value="16" name="interfaceId"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/PowerDistributionSystem/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="ab8855e0-35ad-11e3-aa6e-0800200c9a01" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10104" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/PowerDistributionSystem/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
```

```
        </vcl:configGroup>
        <vcl:configGroup type="POWER_DISTRIBUTION_SYSTEM_DATA" name="soapComponent">
            <vcl:configGroup name="powerDataTypes">
                <vcl:configGroup type="PLATFORM_BATTERY" name="powerDataType">
                    <vcl:configItem value="ab8855e0-35ad-11e3-aa6e-0800200c9a01" name="batteryId"/>
                    <vcl:configItem value="Battery1" name="name"/>
                    <vcl:configItem value="true" name="dataOfInterest"/>
                    <vcl:configItem value="true" name="dataEnabled"/>
                    <vcl:configItem value="1000" name="updatePeriod"/>
                    <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                    <vcl:configGroup name="dataTransportConfigurations">
                        <vcl:configGroup name="transportConfiguration">
                            <vcl:configItem value="ab8855e0-35ad-11e3-aa6e-0800200c9a02" name="uniqueName"/>
                            <vcl:configItem value="multicast://239.192.1.1:10010" name="uri"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
                </vcl:configGroup>
                <vcl:configGroup type="PLATFORM_POWER" name="powerDataType">
                    <vcl:configItem value="true" name="dataOfInterest"/>
                    <vcl:configItem value="true" name="dataEnabled"/>
                    <vcl:configItem value="1000" name="updatePeriod"/>
                    <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                    <vcl:configGroup name="dataTransportConfigurations">
                        <vcl:configGroup name="transportConfiguration">
                            <vcl:configItem value="ab8855e0-35ad-11e3-aa6e-0800200c9a03" name="uniqueName"/>
                            <vcl:configItem value="multicast://239.192.1.1:10011" name="uri"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
                </vcl:configGroup>
                <vcl:configGroup type="PLATFORM_POWER_GENERATION" name="powerDataType">
                    <vcl:configItem value="Generator1" name="name"/>
                    <vcl:configItem value="APU1" name="apuName"/>
                    <vcl:configItem value="true" name="dataOfInterest"/>
                    <vcl:configItem value="true" name="dataEnabled"/>
                    <vcl:configItem value="1000" name="updatePeriod"/>
                    <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                    <vcl:configGroup name="dataTransportConfigurations">
                        <vcl:configGroup name="transportConfiguration">
                            <vcl:configItem value="ab8855e0-35ad-11e3-aa6e-0800200c9a04" name="uniqueName"/>
                            <vcl:configItem value="multicast://239.192.1.1:10012" name="uri"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
                </vcl:configGroup>
                <vcl:configGroup type="POWER_SYSTEM_DEVICE" name="powerDataType">
```

```
                <vcl:configItem value="ab8855e0-35ad-11e3-aa6e-0800200c9a05" name="deviceId"/>
                <vcl:configItem value="POWER_CONTROLLER" name="deviceType"/>
                <vcl:configItem value="Device1" name="name"/>
                <vcl:configItem value="true" name="dataOfInterest"/>
                <vcl:configItem value="true" name="dataEnabled"/>
                <vcl:configItem value="1000" name="updatePeriod"/>
                <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                <vcl:configGroup name="dataTransportConfigurations">
                    <vcl:configGroup name="transportConfiguration">
                        <vcl:configItem value="ab8855e0-35ad-11e3-aa6e-0800200c9a06" name="uniqueName"/>
                        <vcl:configItem value="multicast://239.192.1.1:10013" name="uri"/>
                    </vcl:configGroup>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup type="POWER_SYSTEM_DEVICE_THRESHOLD" name="powerDataType">
                <vcl:configItem value="ab8855e0-35ad-11e4-aa6e-0800200c9a05" name="deviceId"/>
                <vcl:configItem value="POWER_CONTROLLER" name="deviceType"/>
                <vcl:configItem value="Device1" name="name"/>
                <vcl:configItem value="true" name="dataOfInterest"/>
                <vcl:configItem value="true" name="dataEnabled"/>
                <vcl:configItem value="1000" name="updatePeriod"/>
                <vcl:configItem value="100" name="minimumUpdatePeriod"/>
                <vcl:configGroup name="dataTransportConfigurations">
                    <vcl:configGroup name="transportConfiguration">
                        <vcl:configItem value="ab8855e0-35ad-11e3-aa6e-0800200c9a07" name="uniqueName"/>
                        <vcl:configItem value="multicast://239.192.1.1:10014" name="uri"/>
                    </vcl:configGroup>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup name="ports">
            <vcl:configGroup name="port">
                <vcl:configItem value="1.9" name="portStandardVersion"/>
                <vcl:configItem value="http://192.168.1.10:53714/PowerDistributionSystem/powerDistributionSystemData"
                    name="uri"/>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
    <vcl:configGroup type="POWER_DISTRIBUTION_SYSTEM_DEVICE" name="soapComponent">
        <vcl:configGroup name="apu">
            <vcl:configItem value="ab8855e0-35ad-11e3-aa6e-0800200c9a09" name="apuId"/>
            <vcl:configItem value="true" name="enabled"/>
        </vcl:configGroup>
        <vcl:configGroup name="powerControllerDevices">
```

```
                <vcl:configGroup name="powerControllerDevice">
                    <vcl:configItem value="true" name="enabled"/>
                    <vcl:configItem value="ab8855e0-35ad-11e3-aa6e-0800200c9a08" name="powerControllerDeviceId"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/PowerDistributionSystem/powerDistributionSystemDevice"
                        name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/PowerDistributionSystem/versionInformation"
                        name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/PowerDistributionSystem/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/PowerDistributionSystem/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.21 Remote Weapon Station Configuration Description <30021–20210409, Pro>

20.21.1 Remote Weapon Station Configuration Description Tree View

The tree view below shows the hierarchy of the Remote Weapon Station (RWS) configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Remote Weapon Station Component
  - Interface ID (Configurable)
  - Announcement Enabled (Configurable)
  - Remote Weapon Station Device SOAP Component
    - Ammunition Type (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - VICTORY Data SOAP Component
    - Data Enabled (Mutable)
    - Update Period (Mutable)
    - Minimum Update Period (Reportable)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Device Information SOAP Component
    - Equipment ID (Reportable)
    - Manufacturer (Reportable)
    - Model (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)

        - Port [1,∞)
            - Port Standard Version
            - URI
    - Configuration SOAP Component
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI
    - AA Entrypoints SOAP Component [0,1]
        - PES Location (Mutable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI
    - Authentication Entrypoints SOAP Component [0,1]
        - Authentication Service Location (Mutable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI

20.21.2 Remote Weapon Station-Specific Configuration Description Content

1. ***Remote Weapon Station Component***
    a. A Configuration Group element used to encapsulate the remote weapon station configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the remote weapon station.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be “REMOTE_WEAPON_STATION”.
2. ***Remote Weapon Station Device SOAP Component***
    a. A Configuration Group element used to encapsulate the remote weapon station device SOAP component.
    b. The ***name*** attribute shall be “soapComponent”.
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be “REMOTE_WEAPON_STATION_DEVICE”.
3. ***Ammunition Type***
    a. A Configuration Item element used to describe the type of ammunition being used by the RWS.
    b. The ***name*** attribute shall be “ammunitionType”.
    c. The ***value*** attribute shall be a string.

20.21.3 Remote Weapon Station Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Remote Weapon Station is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="REMOTE_WEAPON_STATION" name="517075c0-2521-11e3-8224-0800200c9a66">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configItem value="7" name="interfaceId"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/RwsService/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="DEVICE_INFORMATION" name="soapComponent">
            <vcl:configItem value="1" name="equipmentId"/>
            <vcl:configItem value="swri" name="manufacturer"/>
            <vcl:configItem value="superRWS" name="model"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/RwsService/deviceInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7dec-11d0-a765-00a0c91e6bf3" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10151" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
```

```
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/RwsService/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="REMOTE_WEAPON_STATION_DEVICE" name="soapComponent">
            <vcl:configItem value="someKindOfProjectile" name="ammunitionType"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/RwsService/remoteWeaponStationDevice" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/RwsService/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VICTORY_DATA" name="soapComponent">
            <vcl:configItem value="true" name="dataEnabled"/>
            <vcl:configItem value="100" name="minimumUpdatePeriod"/>
            <vcl:configItem value="1000" name="updatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-11ec-11d3-a765-00a0c93e6bf4" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10008" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/RwsService/victoryData" name="uri"/>
```

```
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/RwsService/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/RwsService/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.22 Vehicle Configuration Service Configuration Description <30050–20210409, Pro>

20.22.1 Vehicle Configuration Service Configuration Description Tree View

The tree view below shows the hierarchy of the Vehicle Configuration Service configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Vehicle Configuration Service Component
  - Announcement Enabled (Configurable)
  - Vehicle Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - AA Entrypoints SOAP Component [0,1]
    - PES Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Authentication Entrypoints SOAP Component [0,1]
    - Authentication Service Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI

20.22.2 Vehicle Configuration Service-Specific Configuration Description Content

1. ***Vehicle Configuration Service Component***
    a. A Configuration Group element used to encapsulate the vehicle configuration service configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the vehicle configuration service.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "VEHICLE_CONFIGURATION_SERVICE".
2. ***Vehicle Configuration SOAP Component***
    a. A Configuration Group element used to encapsulate the vehicle configuration SOAP component.
    b. The ***name*** attribute shall be "soapComponent".
    c. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "VEHICLE_CONFIGURATION".

20.22.3 Vehicle Configuration Service Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Vehicle Configuration Service is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="VEHICLE_CONFIGURATION_SERVICE" name="de11788a-6983-4d64-986c-ff7eeea47bb1">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VehicleConfigurationService/configuration"
name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7dec-11d0-a765-00a0c91e6bf7" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10127" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VehicleConfigurationService/healthPublishing"
                        name="uri"/>
                </vcl:configGroup>
```

```
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VEHICLE_CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VehicleConfigurationService/vehicleConfiguration"
                        name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VehicleConfigurationService/versionInformation"
                        name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VehicleConfigurationService/aaEntrypoints"
name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VehicleConfigurationService/authnEntrypoints"
                        name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.23 Live Training Transformation Adapter Configuration Description <30052–20210409, Pro>

20.23.1 Live Training Transformation Adapter Configuration Description Tree View

The tree view below shows the hierarchy of the Live Training Transformation (LT2) Adapter configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- LT2 Adapter Component
  - Interface ID (Configurable)
  - Announcement Enabled (Configurable)
  - LT2 Adapter Data SOAP Component
    - Data Enabled (Mutable)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - AA Entrypoints SOAP Component [0,1]
    - PES Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Authentication Entrypoints SOAP Component [0,1]
    - Authentication Service Location (Mutable)

- Ports (Reportable)
  - Port [1,∞)
    - Port Standard Version
    - URI

20.23.2 Live Training Transformation Adapter-Specific Configuration Description Content

1. ***LT2 Adapter Component***
   a. A Configuration Group element used to encapsulate the LT2 adapter configuration description.
   b. The ***name*** attribute shall be a string representing the unique name of the LT2 adapter.
   c. The ***name*** attribute shall be unique within the scope of the VDB.

   > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

   d. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "LT2_ADAPTER".
2. ***LT2 Adapter Data SOAP Component***
   a. A Configuration Group element used to encapsulate the LT2 adapter data SOAP component.
   b. The ***name*** attribute shall be "soapComponent".
   c. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "LT2_ADAPTER_DATA".

20.23.3 Live Training Transformation Adapter Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing an LT2 Adapter is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="LT2_ADAPTER" name="fb748639-377b-4d76-9bd4-ff4869661ff0">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configItem value="74" name="interfaceId"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Lt2Adapter/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7dec-11d0-a765-00a0c91e6fbf" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10128" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Lt2Adapter/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
```

```
        </vcl:configGroup>
        <vcl:configGroup type="LT2_ADAPTER_DATA" name="soapComponent">
            <vcl:configItem value="true" name="dataEnabled"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-11ec-11d2-a765-00a0c93e6fb4" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10000" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Lt2Adapter/lt2AdapterData" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Lt2Adapter/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Lt2Adapter/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Lt2Adapter/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
```

```
    </vcl:configGroup>
</vcl:VCL>
```

20.24 Attribute Store Service Configuration Description <30022–20210409, Pro>

20.24.1 Attribute Store Service Configuration Description Tree View

The tree view below shows the hierarchy of the Attribute Store Service configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Attribute Store Service Component
  - Announcement Enabled (Configurable)
  - Attribute Store SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI

20.24.2 Attribute Store Service-Specific Configuration Description Content

1. ***Attribute Store Service Component***
   a. A Configuration Group element used to encapsulate the attribute store service configuration description.
   b. The ***name*** attribute shall be a string representing the unique name of the attribute store service.
   c. The ***name*** attribute shall be unique within the scope of the VDB.

      NOTE: *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

   d. The ***type*** attribute shall be "ATTRIBUTE_STORE_SERVICE".
2. ***Attribute Store SOAP Component***
   a. A Configuration Group element used to encapsulate the attribute store SOAP component.
   b. The ***name*** attribute shall be "soapComponent".
   c. The ***type*** attribute shall be "ATTRIBUTE_STORE".

20.24.3 Attribute Store Service Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing an Attribute Store Service is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="ATTRIBUTE_STORE_SERVICE" name="de11788a-6983-4d22-986c-ff7eeea47bb0">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/AttributeStoreService/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="ATTRIBUTE_STORE" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/AttributeStoreService/attributeStore" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7d22-11d0-a765-00a0c91e6bf6" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10110" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
```

```
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/AttributeStoreService/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/AttributeStoreService/versionInformation"
name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.25 Policy Decision Service Configuration Description <30023–20210409, Pro>

20.25.1 Policy Decision Service Configuration Description Tree View

The tree view below shows the hierarchy of the Policy Decision Service configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Policy Decision Service Component
  - Announcement Enabled (Configurable)
  - Policy Decision SOAP Component
    - Attribute Store Service (AS) Location (Mutable)
    - Policy Store Service (PS) Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI

20.25.2 Policy Decision Service-Specific Configuration Description Content

1. ***Policy Decision Service Component***
   a. A Configuration Group element used to encapsulate the policy decision service configuration description.
   b. The ***name*** attribute shall be a string representing the unique name of the policy decision service.
   c. The ***name*** attribute shall be unique within the scope of the VDB.

      NOTE: *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

   d. The ***type*** attribute shall be "POLICY_DECISION_SERVICE".
2. ***Policy Decision SOAP Component***
   a. A Configuration Group element used to encapsulate the policy decision SOAP component.

   b. The ***name*** attribute shall be “soapComponent”.
   c. The ***type*** attribute shall be “POLICY_DECISION”.
3. ***Attribute Store Service (AS) Location***
   a. A Configuration Item element used to identify the location of the attribute store service.
   b. The ***name*** attribute shall be “ASLocation”.
   c. The ***value*** attribute shall be a string in the format of a valid URI.
4. ***Policy Store Service (PS) Location***
   a. A Configuration Item element used to identify the location of the policy store service.
   b. The ***name*** attribute shall be “PSLocation”.
   c. The ***value*** attribute shall be a string in the format of a valid URI.

20.25.3 Policy Decision Service Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Policy Decision Service is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="POLICY_DECISION_SERVICE" name="de11788a-6983-4d23-986c-ff7eeea47bb0">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/PolicyDecisionService/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="POLICY_DECISION" name="soapComponent">
            <vcl:configItem name="ASLocation" value="https://192.168.1.10:53714/AttributeStoreService/attributeStore"/>
            <vcl:configItem name="PSLocation" value="https://192.168.1.10:53714/PolicyStoreService/policyStore"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/PolicyDecisionService/policyDecision" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7d23-11d0-a765-00a0c91e6bf6" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10112" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
```

```
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/PolicyDecisionService/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/PolicyDecisionService/versionInformation"
name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.26 Policy Enforcement Service Configuration Description <30024–20210409, Pro>

20.26.1 Policy Enforcement Service Configuration Description Tree View

The tree view below shows the hierarchy of the Policy Enforcement Service configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Policy Enforcement Service Component
  - Announcement Enabled (Configurable)
  - Policy Enforcement SOAP Component
    - PDS Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI

20.26.2 Policy Enforcement Service-Specific Configuration Description Content

1. ***Policy Enforcement Service Component***
   a. A Configuration Group element used to encapsulate the policy enforcement service configuration description.
   b. The ***name*** attribute shall be a string representing the unique name of the policy enforcement service.
   c. The ***name*** attribute shall be unique within the scope of the VDB.

      NOTE: *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

   d. The ***type*** attribute shall be "POLICY_ENFORCEMENT_SERVICE".
2. ***Policy Enforcement SOAP Component***
   a. A Configuration Group element used to encapsulate the policy enforcement SOAP component.
   b. The ***name*** attribute shall be "soapComponent".

c. The ***type*** attribute shall be “POLICY_ENFORCEMENT”.

3. ***PDS Location***
   a. A Configuration Item element used to identify the location of the policy decision service (PDS).
   b. The ***name*** attribute shall be “PDSLocation”.
   c. The ***value*** attribute shall be a string in the format of a valid URI.

20.26.3 Policy Enforcement Service Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Policy Enforcement Service is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="POLICY_ENFORCEMENT_SERVICE" name="de11788a-6983-4d24-986c-ff7eeea47bb0">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/PolicyEnforcementService/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="POLICY_ENFORCEMENT" name="soapComponent">
            <vcl:configItem name="PDSLocation" value="https://192.168.1.10:53714/PolicyDecisionService/policyDecision"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/PolicyEnforcementService/policyEnforcement"
name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7d24-11d0-a765-00a0c91e6bf6" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10113" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
```

```
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/PolicyEnforcementService/healthPublishing"
name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/PolicyEnforcementService/versionInformation"
                        name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.27 Policy Store Service Configuration Description <30025–20210409, Pro>

20.27.1 Policy Store Service Configuration Description Tree View

The tree view below shows the hierarchy of the Policy Store Service configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Policy Store Service Component
  - Announcement Enabled (Configurable)
  - Policy Store SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI

20.27.2 Policy Store Service-Specific Configuration Description Content

1. ***Policy Store Service Component***
   a. A Configuration Group element used to encapsulate the policy store service configuration description.
   b. The ***name*** attribute shall be a string representing the unique name of the policy store service.
   c. The ***name*** attribute shall be unique within the scope of the VDB.

      NOTE: *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

   d. The ***type*** attribute shall be "POLICY_STORE_SERVICE".
2. ***Policy Store SOAP Component***
   a. A Configuration Group element used to encapsulate the policy store SOAP component.
   b. The ***name*** attribute shall be "soapComponent".
   c. The ***type*** attribute shall be "POLICY_STORE".

20.27.3 Policy Store Service Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Policy Store Service is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="POLICY_STORE_SERVICE" name="de11788a-6983-4d25-986c-ff7eeea47bb0">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/PolicyStoreService/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="POLICY_STORE" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/PolicyStoreService/policyStore" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7d25-11d0-a765-00a0c91e6bf6" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10114" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
```

```
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/PolicyStoreService/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/PolicyStoreService/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.28 Authentication Service Configuration Description <30026–20210409, Pro>

20.28.1 Authentication Service Configuration Description Tree View

The tree view below shows the hierarchy of the Authentication Service configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Authentication Service Component
  - Announcement Enabled (Configurable)
  - Authentication Service SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI

20.28.2 Authentication Service-Specific Configuration Description Content

1. ***Authentication Service Component***
   a. A Configuration Group element used to encapsulate the authentication service configuration description.
   b. The ***name*** attribute shall be a string representing the unique name of the authentication service.
   c. The ***name*** attribute shall be unique within the scope of the VDB.

      NOTE: *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

   d. The ***type*** attribute shall be "AUTHENTICATION_SERVICE".
2. ***Authentication Service SOAP Component***
   a. A Configuration Group element used to encapsulate the authentication service SOAP component.
   b. The ***name*** attribute shall be "soapComponent".
   c. The ***type*** attribute shall be "AUTHENTICATION".

20.28.3 Authentication Service Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing an Authentication Service is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="AUTHENTICATION_SERVICE" name="de11788a-6983-4d26-986c-ff7eeea47bb0">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/AuthenticationService/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHENTICATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/AuthenticationService/authentication" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7d26-11d0-a765-00a0c91e6bf6" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10115" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
```

```
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/AuthenticationService/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="https://192.168.1.10:53714/AuthenticationService/versionInformation"
name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.29 Host-Based Antivirus Configuration Description <30027–20210409, Pro>

20.29.1 Host-Based Antivirus Configuration Description Tree View

The tree view below shows the hierarchy of the Host-Based Antivirus configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Host-Based Antivirus Component
  - Announcement Enabled (Configurable)
  - Host-Based Antivirus SOAP Component
    - Last Scan Time (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - AA Entrypoints SOAP Component [0,1]
    - PES Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Authentication Entrypoints SOAP Component [0,1]
    - Authentication Service Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI

20.29.2 Host-Based Antivirus-Specific Configuration Description Content

1. ***Host-Based Antivirus Component***
    a. A Configuration Group element used to encapsulate the host-based antivirus configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the host-based antivirus.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. The ***type*** attribute shall be "HOST_BASED_ANTIVIRUS_SERVICE".
2. ***Host-Based Antivirus SOAP Component***
    a. A Configuration Group element used to encapsulate the host-based antivirus SOAP component.
    b. The ***name*** attribute shall be "soapComponent".
    c. The ***type*** attribute shall be "HOST_BASED_ANTIVIRUS".
3. ***Last Scan Time***
    a. A Configuration Item element used to identify the last time an antivirus scan was performed.
    b. The ***name*** attribute shall be "lastScanTime".
    c. The ***value*** attribute shall be a string that conforms to XML dateTime format.

20.29.3 Host-Based Antivirus Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Host-Based Antivirus is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="HOST_BASED_ANTIVIRUS_SERVICE" name="de11788a-6983-4d27-986c-ff7eeea47bb0">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/HostBasedAntivirus/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HOST_BASED_ANTIVIRUS" name="soapComponent">
            <vcl:configItem name="lastScanTime" value="0001-01-01T00:00:00"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/HostBasedAntivirus/hostBasedAntivirus" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7d27-11d0-a765-00a0c91e6bf6" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10116" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
```

```
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/HostBasedAntivirus/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/HostBasedAntivirus/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/HostBasedAntivirus/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/HostBasedAntivirus/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.30 Host-Based Firewall Configuration Description <30028–20210409, Pro>

20.30.1 Host-Based Firewall Configuration Description Tree View

The tree view below shows the hierarchy of the Host-Based Firewall configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Host-Based Firewall Component
  - Announcement Enabled (Configurable)
  - Host-Based Firewall SOAP Component
    - Logging (Mutable)
    - Port Filtering (Mutable)
    - Application Filtering [0,1] (Mutable)
    - Application Filtering Mode Default [0,1] (Mutable)
      - Mode Default Inbound
      - Mode Default Outbound
    - Port Filtering Mode Default (Mutable)
      - Mode Default Inbound
      - Mode Default Outbound
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - AA Entrypoints SOAP Component [0,1]
    - PES Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version

        - URI
    - Authentication Entrypoints SOAP Component [0,1]
        - Authentication Service Location (Mutable)
        - Ports (Reportable)
            - Port [1,∞)
                - Port Standard Version
                - URI

20.30.2 Host-Based Firewall-Specific Configuration Description Content

1. ***Host-Based Firewall Component***
    a. A Configuration Group element used to encapsulate the host-based firewall configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the host-based firewall.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. The ***type*** attribute shall be "HOST_BASED_FIREWALL_SERVICE".
2. ***Host-Based Firewall SOAP Component***
    a. A Configuration Group element used to encapsulate the host-based firewall SOAP component.
    b. The ***name*** attribute shall be "soapComponent".
    c. The ***type*** attribute shall be "HOST_BASED_FIREWALL".
3. ***Logging***
    a. A Configuration Item element used to set the Logging capability within the Firewall.
    b. The ***name*** attribute shall be "logging".
    c. The ***value*** attribute shall be a string chosen from one of the following values:
        i. ON
        ii. OFF
4. ***Port Filtering***
    a. A Configuration Item element used to set the Port Filtering capability within the Firewall.
    b. The ***name*** attribute shall be "portFiltering".
    c. The ***value*** attribute shall be a string chosen from one of the following values:
        i. ON
        ii. OFF
5. ***Application Filtering***
    a. A Configuration Item element used to set the Application Filtering capability within the Firewall.
    b. The ***name*** attribute shall be "appFiltering".
    c. The ***value*** attribute shall be a string chosen from one of the following values:
        i. ON
        ii. OFF
6. ***Application Filtering Mode Default***
    a. A Configuration Group element used to encapsulate inbound and outbound default actions for Application Filtering.
    b. The ***name*** attribute shall be "appFilteringModeDefault".
    c. This element shall not contain a ***type*** attribute.
7. ***Mode Default Inbound***
    a. A Configuration Item element used to set the inbound Filtering Mode Default capability within the Firewall.
    b. The ***name*** attribute shall be "modeDefaultIn".
    c. The ***value*** attribute shall be a string chosen from one of the following values:
        i. ALLOW
        ii. BLOCK
8. ***Mode Default Outbound***
    a. A Configuration Item element used to set the outbound Filtering Mode Default capability within the Firewall.
    b. The ***name*** attribute shall be "modeDefaultOut".
    c. The ***value*** attribute shall be a string chosen from one of the following values:
        i. ALLOW
        ii. BLOCK
9. ***Port Filtering Mode Default***

a. A Configuration Group element used to encapsulate inbound and outbound default actions for Port Filtering.
b. The ***name*** attribute shall be “portFilteringModeDefault”.
c. This element shall not contain a ***type*** attribute.

20.30.3 Host-Based Firewall Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Host-Based Firewall is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="HOST_BASED_FIREWALL_SERVICE" name="de11788a-6983-4d28-986c-ff7eeea47bb0">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/HostBasedFirewall/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HOST_BASED_FIREWALL" name="soapComponent">
            <vcl:configItem name="logging" value="ON"/>
            <vcl:configItem name="portFiltering" value="ON"/>
            <vcl:configItem name="appFiltering" value="OFF"/>
            <vcl:configGroup name="appFilteringModeDefault">
                <vcl:configItem name="modeDefaultIn" value="BLOCK"/>
                <vcl:configItem name="modeDefaultOut" value="ALLOW"/>
            </vcl:configGroup>
            <vcl:configGroup name="portFilteringModeDefault">
                <vcl:configItem name="modeDefaultIn" value="BLOCK"/>
                <vcl:configItem name="modeDefaultOut" value="ALLOW"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/HostBasedFirewall/hostBasedFirewall" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
```

```
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7d28-11d0-a765-00a0c91e6bf6" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10118" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/HostBasedFirewall/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/HostBasedFirewall/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/HostBasedFirewall/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/HostBasedFirewall/authnEntrypoints" name="uri"/>
```

```
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.31 Network Firewall Configuration Description <30029–20210409, Pro>

20.31.1 Network Firewall Configuration Description Tree View

The tree view below shows the hierarchy of the Network Firewall configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Network Firewall Component
  - Announcement Enabled (Configurable)
  - Network Firewall SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Device Information SOAP Component
    - Equipment ID (Reportable)
    - Manufacturer (Reportable)
    - Model (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - AA Entrypoints SOAP Component [0,1]
    - PES Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI

- Authentication Entrypoints SOAP Component [0,1]
  - Authentication Service Location (Mutable)
  - Ports (Reportable)
    - Port [1,∞)
      - Port Standard Version
      - URI

20.31.2 Network Firewall-Specific Configuration Description Content

1. ***Network Firewall Component***
   a. A Configuration Group element used to encapsulate the network firewall configuration description.
   b. The ***name*** attribute shall be a string representing the unique name of the network firewall.
   c. The ***name*** attribute shall be unique within the scope of the VDB.

   > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

   d. The ***type*** attribute shall be "FIREWALL_SERVICE".
2. ***Network Firewall SOAP Component***
   a. A Configuration Group element used to encapsulate the network firewall SOAP component.
   b. The ***name*** attribute shall be "soapComponent".
   c. The ***type*** attribute shall be "FIREWALL".

20.31.3 Network Firewall Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Network Firewall is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="FIREWALL_SERVICE" name="de11788a-6983-4d29-986c-ff7eeea47bb0">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Firewall/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="FIREWALL" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Firewall/firewall" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="DEVICE_INFORMATION" name="soapComponent">
            <vcl:configItem value="1" name="equipmentId"/>
            <vcl:configItem value="axis" name="manufacturer"/>
            <vcl:configItem value="ptz" name="model"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Firewall/deviceInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
```

```
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7d29-11d0-a765-00a0c91e6bf6" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10120" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Firewall/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Firewall/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Firewall/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/Firewall/authnEntrypoints" name="uri"/>
```

```
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.32 Intrusion Detection and Prevention System Configuration Description <30030–20210409, Pro>

20.32.1 Intrusion Detection and Prevention System Configuration Description Tree View

The tree view below shows the hierarchy of the Intrusion Detection and Prevention System (IDPS) configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- IDPS Component
  - Announcement Enabled (Configurable)
  - IDPS SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - AA Entrypoints SOAP Component [0,1]
    - PES Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Authentication Entrypoints SOAP Component [0,1]
    - Authentication Service Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI

20.32.2 Intrusion Detection and Prevention System-Specific Configuration Description Content

1. ***IDPS Component***
    a. A Configuration Group element used to encapsulate the IDPS configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the IDPS.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. The ***type*** attribute shall be "IDPS_SERVICE".
2. ***IDPS SOAP Component***
    a. A Configuration Group element used to encapsulate the IDPS SOAP component.
    b. The ***name*** attribute shall be "soapComponent".
    c. The ***type*** attribute shall be "IDPS".

20.32.3 Intrusion Detection and Prevention System Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing an Intrusion Detection and Prevention System is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="IDPS_SERVICE" name="de11788a-6983-4d30-986c-ff7eeea47bb0">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/IDPS/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="IDPS" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/IDPS/idps" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7d30-11d0-a765-00a0c91e6bf6" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10121" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
```

```
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/IDPS/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/IDPS/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/IDPS/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/IDPS/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.33 Data at Rest Encryptor Configuration Description <30031–20210409, Pro>

20.33.1 Data at Rest Encryptor Configuration Description Tree View

The tree view below shows the hierarchy of the Data at Rest (DAR) Encryptor (DARE) configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- DAR Encryptor Component
  - Announcement Enabled (Configurable)
  - DAR Encryptor SOAP Component
    - Authenticator Type (Mutable)
    - DAR Security Policy [0,1] (Mutable)
    - Classification Level (Reportable)
    - Crypto Type (Reportable)
    - Recovery Agent (Mutable)
    - DARE Status (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - AA Entrypoints SOAP Component [0,1]
    - PES Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Authentication Entrypoints SOAP Component [0,1]
    - Authentication Service Location (Mutable)

- Ports (Reportable)
  - Port [1,∞)
    - Port Standard Version
    - URI

20.33.2 Data at Rest Encryptor-Specific Configuration Description Content

1. ***DAR Encryptor Component***
   a. A Configuration Group element used to encapsulate the DAR encryptor configuration description.
   b. The ***name*** attribute shall be a string representing the unique name of the DAR encryptor.
   c. The ***name*** attribute shall be unique within the scope of the VDB.

   > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

   d. The ***type*** attribute shall be "DARE_SERVICE".
2. ***DAR Encryptor SOAP Component***
   a. A Configuration Group element used to encapsulate the DAR encryptor SOAP component.
   b. The ***name*** attribute shall be "soapComponent".
   c. The ***type*** attribute shall be "DARE".
3. ***Authenticator Type***
   a. A Configuration Item element used to identify the type of authentication used by the DARE instantiation.
   b. The ***name*** attribute shall be "authenticatorType".
   c. The ***value*** attribute shall be a string chosen from one of the following values:
      i. PASSWORD
      ii. CAC
      iii. PIN
      iv. STARTUP_TOKEN
4. ***DAR Security Policy***
   a. A Configuration Item element used to list the DAR policies.
   b. The ***name*** attribute shall be "darPolicy".
   c. The ***value*** attribute shall be a human readable string.
5. ***Classification Level***
   a. A Configuration Item element used to identify the classification level of the DARE instantiation.
   b. The ***name*** attribute shall be "classificationLevel".
   c. The ***value*** attribute shall be a string chosen from one of the following values:
      i. UNCLASSIFIED
      ii. CONFIDENTIAL
      iii. SECRET
      iv. TOPSECRET
6. ***Crypto Type***
   a. A Configuration Item element used to identify the certification level of the DARE instantiation.
   b. The ***name*** attribute shall be "cryptoType".
   c. The ***value*** attribute shall be a string chosen from one of the following values:
      i. NIST_CERTIFIED
      ii. NSA_APPROVED
      iii. UNEVALUATED
7. ***Recovery Agent***
   a. A Configuration Item element used to identify the entity that acts as the recovery agent.
   b. The ***name*** attribute shall be "recoveryAgent".
   c. The ***value*** attribute shall be a string in the format of a valid URI.
8. ***DARE Status***
   a. A Configuration Item element used to describe the status of the DARE instantiation.
   b. The ***name*** attribute shall be "dareStatus".
   c. The ***value*** attribute shall be a string chosen from one of the following values:
      i. ON
      ii. OFF

20.33.3 Data at Rest Encryptor Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Data at Rest Encryptor is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="DARE_SERVICE" name="de11788a-6983-4d31-986c-ff7eeea47bb0">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DarEncryptor/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="DARE" name="soapComponent">
            <vcl:configItem name="authenticatorType" value="PASSWORD"/>
            <vcl:configItem name="darPolicy" value="Everything is write-append only"/>
            <vcl:configItem name="classificationLevel" value="UNCLASSIFIED"/>
            <vcl:configItem name="cryptoType" value="UNEVALUATED"/>
            <vcl:configItem name="recoveryAgent" value="urn:APG:bob"/>
            <vcl:configItem name="dareStatus" value="ON"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DarEncryptor/dare" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7d31-11d0-a765-00a0c91e6bf6" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10123" name="uri"/>
```

```
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DarEncryptor/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DarEncryptor/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DarEncryptor/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DarEncryptor/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.34 Data Signing Service Configuration Description <30032–20210409, Pro>

20.34.1 Data Signing Service Configuration Description Tree View

The tree view below shows the hierarchy of the Data Signing Service configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Data Signing Service Component
  - Announcement Enabled (Configurable)
  - Data Signing SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - AA Entrypoints SOAP Component [0,1]
    - PES Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Authentication Entrypoints SOAP Component [0,1]
    - Authentication Service Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI

20.34.2 Data Signing Service-Specific Configuration Description Content

1. ***Data Signing Service Component***
    a. A Configuration Group element used to encapsulate the data signing service configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the data signing service.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. The ***type*** attribute shall be "DATA_SIGNING_SERVICE".
2. ***Data Signing SOAP Component***
    a. A Configuration Group element used to encapsulate the data signing SOAP component.
    b. The ***name*** attribute shall be "soapComponent".
    c. The ***type*** attribute shall be "DATA_SIGNING".

20.34.3 Data Signing Service Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Data Signing Service is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="DATA_SIGNING_SERVICE" name="de11788a-6983-4d32-986c-ff7eeea47bb0">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DataSigningService/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="DATA_SIGNING" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DataSigningService/dataSigning" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7d32-11d0-a765-00a0c91e6bf6" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10125" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
```

```
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DataSigningService/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DataSigningService/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DataSigningService/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/DataSigningService/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.35 Signed Data Verification Service Configuration Description <30033–20210409, Pro>

20.35.1 Signed Data Verification Service Configuration Description Tree View

The tree view below shows the hierarchy of the Signed Data Verification Service configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Signed Data Verification Service Component
  - Announcement Enabled (Configurable)
  - Signed Data Verification SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels (Mutable)
      - Enabled Fault Severity Level [0,∞)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - AA Entrypoints SOAP Component [0,1]
    - PES Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Authentication Entrypoints SOAP Component [0,1]
    - Authentication Service Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI

20.35.2 Signed Data Verification Service-Specific Configuration Description Content

1. ***Signed Data Verification Service Component***
    a. A Configuration Group element used to encapsulate the signed data verification service configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the signed data verification service.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. The ***type*** attribute shall be "SIGNED_DATA_VERIFICATION_SERVICE".
2. ***Signed Data Verification SOAP Component***
    a. A Configuration Group element used to encapsulate the signed data verification SOAP component.
    b. The ***name*** attribute shall be "soapComponent".
    c. The ***type*** attribute shall be "SIGNED_DATA_VERIFICATION".

20.35.3 Signed Data Verification Service Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Signed Data Verification Service is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="SIGNED_DATA_VERIFICATION_SERVICE" name="de11788a-6983-4d33-986c-ff7eeea47bb0">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SignedDataVerificationService/configuration"
                        name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="SIGNED_DATA_VERIFICATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SignedDataVerificationService/signedDataVerification"
                        name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7d33-11d0-a765-00a0c91e6bf6" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10126" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
```

```
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SignedDataVerificationService/healthPublishing"
                        name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SignedDataVerificationService/versionInformation"
                        name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SignedDataVerificationService/aaEntrypoints"
                        name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SignedDataVerificationService/authnEntrypoints"
                        name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.36 Generic End Node Configuration Description <30040–20210409, Pro>

20.36.1 Generic End Node Configuration Description Tree View

The tree view below shows the hierarchy of the Generic End Node configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Generic End Node Component
  - Announcement Enabled (Configurable)
  - Generic End Node SOAP Component
    - Hostname [0,1] (Mutable)
    - Date/Time [0,1] (Volatile)
    - Auto-Discovery Enabled [0,1] (Mutable)
    - Display Blackout Enabled [0,1] (Mutable)
    - Ethernet Interfaces (Reportable)
      - Ethernet Interface [1,∞)
        - Interface Name
        - Unique Name
        - MAC Address
        - DHCPv4 Enabled
        - DHCPv6 Enabled
        - Static IP Configurations
          - Static IP Configuration [0,∞)
            - Unique Name
            - Address
            - Subnet
            - Broadcast [0,1]
            - Gateway [0,1]
        - Auto IP Configurations
          - Auto IP Configuration [0,∞)
            - Unique Name
            - Type
            - Address
            - Subnet
            - Broadcast [0,1]
            - Gateway [0,1]
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels
      - Enabled Fault Severity Level [0,∞) (Mutable)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)

        - Unique Name
        - URI
    - Ports (Reportable)
        - Port [1,∞)
            - Port Standard Version
            - URI
- Configuration SOAP Component
    - Ports (Reportable)
        - Port [1,∞)
            - Port Standard Version
            - URI
- AA Entrypoints SOAP Component [0,1]
    - PES Location (Mutable)
    - Ports (Reportable)
        - Port [1,∞)
            - Port Standard Version
            - URI
- Authentication Entrypoints SOAP Component [0,1]
    - Authentication Service Location (Mutable)
    - Ports (Reportable)
        - Port [1,∞)
            - Port Standard Version
            - URI

20.36.2 Generic End Node-Specific Configuration Description Content

1. ***Generic End Node Component***
    a. A Configuration Group element used to encapsulate the generic end node configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the generic end node.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be “GENERIC_END_NODE”.

20.36.3 Generic End Node Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Generic End Node is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="GENERIC_END_NODE" name="5b5debbc-1955-11e3-9a3b-f23c91aec05e">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/GenericEndNode/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="GENERIC_END_NODE" name="soapComponent">
            <vcl:configItem value="true" name="autodiscoveryEnabled"/>
            <vcl:configItem value="2013-10-31T09:06:12z" name="dateTime"/>
            <vcl:configItem value="false" name="displayBlackoutEnabled"/>
            <vcl:configItem value="myHostname" name="hostname"/>
            <vcl:configGroup name="ethernetInterfaces">
                <vcl:configGroup name="ethernetInterface">
                    <vcl:configItem value="true" name="dhcpv4Enabled"/>
                    <vcl:configItem value="false" name="dhcpv6Enabled"/>
                    <vcl:configItem value="eth0" name="interfaceName"/>
                    <vcl:configItem value="63-98-09-32-00-DE" name="macAddress"/>
                    <vcl:configItem value="5b5dfeea-1955-11e3-9a3b-f23c91aec05e" name="uniqueName"/>
                    <vcl:configGroup name="autoIpConfigurations">
                        <vcl:configGroup name="autoIpConfiguration">
                            <vcl:configItem value="10.30.10.10" name="address"/>
                            <vcl:configItem value="10.30.10.255" name="broadcast"/>
                            <vcl:configItem value="10.30.10.1" name="gateway"/>
                            <vcl:configItem value="10.1.1.0" name="subnet"/>
                            <vcl:configItem value="DHCP_V4" name="type"/>
                            <vcl:configItem value="5b5e035e-1955-11e3-9a3b-f23c91aec05e" name="uniqueName"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
```

```
                <vcl:configGroup name="staticIpConfigurations">
                    <vcl:configGroup name="staticIpConfiguration">
                        <vcl:configItem value="http://192.168.1.10" name="address"/>
                        <vcl:configItem value="http://192.168.1.255" name="broadcast"/>
                        <vcl:configItem value="http://192.168.1.1" name="gateway"/>
                        <vcl:configItem value="255.255.255.0" name="subnet"/>
                        <vcl:configItem value="5b5e0160-1955-11e3-9a3b-f23c91aec05e" name="uniqueName"/>
                    </vcl:configGroup>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/GenericEndNode/genericEndNode" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="5b5df77e-1955-11e3-9a3b-f23c91aec05e" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10148" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/GenericEndNode/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
```

```
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/GenericEndNode/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/GenericEndNode/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/GenericEndNode/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.37 Network Attached Storage Configuration Description <30041–20210409, Pro>

20.37.1 Network Attached Storage Configuration Description Tree View

The tree view below shows the hierarchy of the Network Attached Storage configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Network Attached Storage Component
  - WebDAV
    - Software Name (Reportable)
    - Software Version (Reportable)
    - Share Addresses (Reportable)
      - URI [1,∞)
  - NFS [0,1]
    - Software Name (Reportable)
    - Software Version (Reportable)
    - Share Addresses (Reportable)
      - URI [1,∞)

20.37.2 Network Attached Storage-Specific Configuration Description Content

1. ***Network Attached Storage Component***
   a. A Configuration Group element used to encapsulate the network attached storage configuration description.
   b. The ***name*** attribute shall be a string representing the unique name of the network attached storage.
   c. The ***name*** attribute shall be unique within the scope of the VDB.

      NOTE: *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

   d. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "NETWORK_ATTACHED_STORAGE".
2. ***WebDAV***
   a. A Configuration Group element used to encapsulate the Web Distributed Authoring and Versioning (WebDAV) configuration information.
   b. The ***name*** attribute shall be "webDav".
   c. This element shall not contain a ***type*** attribute.
3. ***NFS***
   a. A Configuration Group element used to encapsulate the Network File System (NFS) configuration information.
   b. The ***name*** attribute shall be "nfs".
   c. This element shall not contain a ***type*** attribute.
4. ***Share Addresses***
   a. A Configuration Group element used to encapsulate the share addresses for a WebDAV or NFS server.
   b. The ***name*** attribute shall be "shareAddresses".
   c. This element shall not contain a ***type*** attribute.
   d. The URI(s) contained within this Configuration Group shall be formatted as specified in IETF RFC 2224: NFS URL Scheme.

20.37.3 Network Attached Storage Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing Network Attached Storage is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="NETWORK_ATTACHED_STORAGE" name="081d40ae-7d0c-11d0-0065-00a0c00e6cf0"/>
        <vcl:configGroup name="webDav">
            <vcl:configItem name="softwareName" value="WebDAV Server"/>
            <vcl:configItem name="softwareVersion" value="v1.5"/>
            <vcl:configGroup name="shareAddresses">
                <vcl:configItem name="uri" value="http://192.168.1.155:53714"/>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup name="nfs">
            <vcl:configItem name="softwareName" value="NFS Server"/>
            <vcl:configItem name="softwareVersion" value="v4.01"/>
            <vcl:configGroup name="shareAddresses">
                <vcl:configItem name="uri" value="http://192.168.1.140:53714"/>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.38 Shared Processing Unit Configuration Description <30042–20210409, Pro>

20.38.1 Shared Processing Unit Configuration Description Tree View

The tree view below shows the hierarchy of the Shared Processing Unit (SPU) configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- Shared Processing Unit Component
  - Announcement Enabled (Configurable)
  - Shared Processing Unit SOAP Component
    - Services
      - Service [0,∞)
        - Unique Name (Reportable)
        - Start Service (Configurable)
        - VICTORY Component Types (Reportable)
          - VICTORY Component Type [1,∞)
            - Unique Name
            - Type
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Generic End Node SOAP Component
    - Hostname [0,1] (Mutable)
    - Date/Time [0,1] (Volatile)
    - Auto-Discovery Enabled [0,1] (Mutable)
    - Display Blackout Enabled [0,1] (Mutable)
    - Ethernet Interfaces (Reportable)
      - Ethernet Interface [1,∞)
        - Interface Name
        - Unique Name
        - MAC Address
        - DHCPv4 Enabled
        - DHCPv6 Enabled
        - Static IP Configurations
          - Static IP Configuration [0,∞)
            - Unique Name
            - Address
            - Subnet
            - Broadcast [0,1]
            - Gateway [0,1]
        - Auto IP Configurations
          - Auto IP Configuration [0,∞)
            - Unique Name
            - Type
            - Address
            - Subnet
            - Broadcast [0,1]
            - Gateway [0,1]
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Version Information SOAP Component

- Interface Standard Version (Reportable)
- Ports (Reportable)
  - Port [1,∞)
    - Port Standard Version
    - URI

◦ Health Publishing SOAP Component
- Fault Update Period (Mutable)
- Status Update Period (Mutable)
- Status Enabled (Mutable)
- Enabled Fault Severity Levels
  - Enabled Fault Severity Level [0,∞) (Mutable)
- Data Transport Configurations (Mutable)
  - Transport Configuration [1,∞)
    - Unique Name
    - URI
- Ports (Reportable)
  - Port [1,∞)
    - Port Standard Version
    - URI

◦ Configuration SOAP Component
- Ports (Reportable)
  - Port [1,∞)
    - Port Standard Version
    - URI

◦ AA Entrypoints SOAP Component [0,1]
- PES Location (Mutable)
- Ports (Reportable)
  - Port [1,∞)
    - Port Standard Version
    - URI

◦ Authentication Entrypoints SOAP Component [0,1]
- Authentication Service Location (Mutable)
- Ports (Reportable)
  - Port [1,∞)
    - Port Standard Version
    - URI

20.38.2 Shared Processing Unit-Specific Configuration Description Content

1. ***Shared Processing Unit Component***
   a. A Configuration Group element used to encapsulate the shared processing unit configuration description.
   b. The ***name*** attribute shall be a string representing the unique name of the shared processing unit.
   c. The ***name*** attribute shall be unique within the scope of the VDB.

   > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

   d. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "SHARED_PROCESSING_UNIT".
2. ***Shared Processing Unit SOAP Component***
   a. A Configuration Group element used to encapsulate the shared processing unit SOAP component.
   b. The ***name*** attribute shall be "soapComponent".
   c. This element shall contain a ***type*** attribute.
      i. The ***type*** attribute shall be "SHARED_PROCESSING_UNIT".
3. ***Services***
   a. A Configuration Group element used to encapsulate the services on the shared processing unit.
   b. The ***name*** attribute shall be "services".
   c. This element shall not contain a ***type*** attribute.

4. ***Service***
    a. A Configuration Group element used to encapsulate a service on the shared processing unit.
    b. The ***name*** attribute shall be “service”.
    c. This element shall not contain a ***type*** attribute.
5. ***Start Service***
    a. A Configuration Item element used to describe whether the service should be started or stopped when the system is booting.
    b. The ***name*** attribute shall be “startService”.
    c. The ***value*** attribute shall be a case-sensitive string representation of a Boolean value where “true” or “1” means the service should be started and “false” or “0” means the service should remain stopped.
6. ***VICTORY Component Types***
    a. A Configuration Group element used to encapsulate the VICTORY component types implemented by a service on the shared processing unit.
    b. The ***name*** attribute shall be “victoryComponentTypes”.
    c. This element shall not contain a ***type*** attribute.
7. ***VICTORY Component Type***
    a. A Configuration Group element used to encapsulate a VICTORY component type implemented by a service on the shared processing unit.
    b. The ***name*** attribute shall be “victoryComponentType”.
    c. This element shall not contain a ***type*** attribute.
8. ***Type***
    a. A Configuration Item element used to describe the type of VICTORY component implemented by a service on the shared processing unit.
    b. The ***name*** attribute shall be “type”.
    c. The ***value*** attribute shall be a string representation of a valid VICTORY component type value as described inside the component Configuration Group element of each component configuration description.

20.38.3 Shared Processing Unit Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a Shared Processing Unit is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="SHARED_PROCESSING_UNIT" name="5b5debbc-1955-11e3-9a3b-f23c91aec05e">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SharedProcessingUnit/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="GENERIC_END_NODE" name="soapComponent">
            <vcl:configItem value="true" name="autodiscoveryEnabled"/>
            <vcl:configItem value="2013-10-31T09:06:12z" name="dateTime"/>
            <vcl:configItem value="false" name="displayBlackoutEnabled"/>
            <vcl:configItem value="myHostname" name="hostname"/>
            <vcl:configGroup name="ethernetInterfaces">
                <vcl:configGroup name="ethernetInterface">
                    <vcl:configItem value="true" name="dhcpv4Enabled"/>
                    <vcl:configItem value="false" name="dhcpv6Enabled"/>
                    <vcl:configItem value="eth0" name="interfaceName"/>
                    <vcl:configItem value="63-98-09-32-00-DE" name="macAddress"/>
                    <vcl:configItem value="5b5dfeea-1955-11e3-9a3b-f23c91aec05f" name="uniqueName"/>
                    <vcl:configGroup name="autoIpConfigurations">
                        <vcl:configGroup name="autoIpConfiguration">
                            <vcl:configItem value="10.30.10.10" name="address"/>
                            <vcl:configItem value="10.30.10.255" name="broadcast"/>
                            <vcl:configItem value="10.30.10.1" name="gateway"/>
                            <vcl:configItem value="10.1.1.0" name="subnet"/>
                            <vcl:configItem value="DHCP_V4" name="type"/>
                            <vcl:configItem value="5b5e035e-1955-11e3-9a3b-f23c91aec05f" name="uniqueName"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
```

```
                <vcl:configGroup name="staticIpConfigurations">
                    <vcl:configGroup name="staticIpConfiguration">
                        <vcl:configItem value="http://192.168.1.10" name="address"/>
                        <vcl:configItem value="http://192.168.1.255" name="broadcast"/>
                        <vcl:configItem value="http://192.168.1.1" name="gateway"/>
                        <vcl:configItem value="255.255.255.0" name="subnet"/>
                        <vcl:configItem value="5b5e0160-1955-11e3-9a3b-f23c91aec05f" name="uniqueName"/>
                    </vcl:configGroup>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SharedProcessingUnit/genericEndNode" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="5b5df77e-1955-11e3-9a3b-f23c91aec05f" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:10149" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SharedProcessingUnit/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="SHARED_PROCESSING_UNIT" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
```

```
                    <vcl:configItem value="http://192.168.1.10:53714/SharedProcessingUnit/sharedProcessingUnit"
name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="services">
                <vcl:configGroup name="service">
                    <vcl:configItem value="6865a590-245c-11e3-8224-0800200c9a66" name="uniqueName"/>
                    <vcl:configItem value="false" name="startService"/>
                    <vcl:configGroup name="victoryComponentTypes">
                        <vcl:configGroup name="victoryComponentType">
                            <vcl:configItem value="6865a590-245c-11e3-8224-0800200c9a67" name="uniqueName"/>
                            <vcl:configItem value="ORIENTATION_SERVICE" name="type"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
                </vcl:configGroup>
                <vcl:configGroup name="service">
                    <vcl:configItem value="de11788a-6983-4d64-986c-ff7eeea47bb0" name="uniqueName"/>
                    <vcl:configItem value="false" name="startService"/>
                    <vcl:configGroup name="victoryComponentTypes">
                        <vcl:configGroup name="victoryComponentType">
                            <vcl:configItem value="de11788a-6983-4d64-986c-ff7eeea47bb1" name="uniqueName"/>
                            <vcl:configItem value="POSITION_SERVICE" name="type"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
                </vcl:configGroup>
                <vcl:configGroup name="service">
                    <vcl:configItem value="e6fa45b0-2226-11e3-8224-0800200c9a66" name="uniqueName"/>
                    <vcl:configItem value="false" name="startService"/>
                    <vcl:configGroup name="victoryComponentTypes">
                        <vcl:configGroup name="victoryComponentType">
                            <vcl:configItem value="e6fa45b0-2226-11e3-8224-0800200c9a67" name="uniqueName"/>
                            <vcl:configItem value="DIRECTION_OF_TRAVEL_SERVICE" name="type"/>
                        </vcl:configGroup>
                    </vcl:configGroup>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SharedProcessingUnit/versionInformation" name="uri"/>
                </vcl:configGroup>
```

```
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SharedProcessingUnit/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/SharedProcessingUnit/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.39 VDB Management Service Configuration Description <30043–20210409, Pro>

20.39.1 VDB Management Service Configuration Description Tree View

The tree view below shows the hierarchy of the VDB Management Service configuration description. For a list of rules associated with the tree view, see the section titled "Tree View Requirements."

- VDB Management Service Component
  - Announcement Enabled (Configurable)
  - Version Information SOAP Component
    - Interface Standard Version (Reportable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Publishing SOAP Component
    - Fault Update Period (Mutable)
    - Status Update Period (Mutable)
    - Status Enabled (Mutable)
    - Enabled Fault Severity Levels
      - Enabled Fault Severity Level [0,∞) (Mutable)
    - Data Transport Configurations (Mutable)
      - Transport Configuration [1,∞)
        - Unique Name
        - URI
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Configuration SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Health Level SOAP Component
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - AA Entrypoints SOAP Component [0,1]
    - PES Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI
  - Authentication Entrypoints SOAP Component [0,1]
    - Authentication Service Location (Mutable)
    - Ports (Reportable)
      - Port [1,∞)
        - Port Standard Version
        - URI

20.39.2 VDB Management Service-Specific Configuration Description Content

1. ***VDB Management Service Component***
    a. A Configuration Group element used to encapsulate the VDB management service configuration description.
    b. The ***name*** attribute shall be a string representing the unique name of the VDB management service.
    c. The ***name*** attribute shall be unique within the scope of the VDB.

    > ***NOTE:*** *VICTORY recommends the use of Universally Unique Identifiers (UUIDs) for maintaining uniqueness. Refer to IETF RFC 4122 for more information about UUIDs.*

    d. This element shall contain a ***type*** attribute.
        i. The ***type*** attribute shall be "VDB_MANAGEMENT_SERVICE".

20.39.3 VDB Management Service Configuration Description Example

The information described in the previous sections shall be encoded into an XML string and inserted inside the VCL Instance Document (VCL ID). An example VCL ID describing a VDB Management Service is provided below.

```
<?xml version="1.0" encoding="UTF-8"?>
<vcl:VCL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:vcl="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd"
    xsi:schemaLocation="http://www.victory-standards.org/Schemas/VICTORYConfigurationLanguage.xsd
        VICTORYConfigurationLanguage.xsd">
    <vcl:configGroup type="VDB_MANAGEMENT_SERVICE" name="a81d4fae-7dec-11d0-a765-00a0c91e6bf0">
        <vcl:configItem value="true" name="announcementEnabled"/>
        <vcl:configItem value="11" name="interfaceId"/>
        <vcl:configGroup type="CONFIGURATION" name="soapComponent">
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VDBManagementService/configuration" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_LEVEL" name="soapComponent">
            <vcl:configItem value="GREEN" name="healthLevel"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VDBManagementService/healthLevel" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="HEALTH_PUBLISHING" name="soapComponent">
            <vcl:configItem value="1000" name="faultUpdatePeriod"/>
            <vcl:configItem value="true" name="statusEnabled"/>
            <vcl:configItem value="5000" name="statusUpdatePeriod"/>
            <vcl:configGroup name="dataTransportConfigurations">
                <vcl:configGroup name="transportConfiguration">
                    <vcl:configItem value="a81d4fae-7dec-11d0-a765-00a0c91e6bf5" name="uniqueName"/>
                    <vcl:configItem value="multicast://239.192.1.1:13100" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
            <vcl:configGroup name="enabledFaultSeverityLevels">
                <vcl:configItem value="EMERGENCY" name="enabledFaultSeverityLevel"/>
```

```
                <vcl:configItem value="ALERT" name="enabledFaultSeverityLevel"/>
                <vcl:configItem value="CRITICAL" name="enabledFaultSeverityLevel"/>
            </vcl:configGroup>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VDBManagementService/healthPublishing" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="VERSION_INFORMATION" name="soapComponent">
            <vcl:configItem value="1.9" name="interfaceStandardVersion"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VDBManagementService/versionInformation" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AA_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="PESLocation" value="https://192.168.1.10:53714/PolicyEnforcementService/
policyEnforcement"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VDBManagementService/aaEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
        <vcl:configGroup type="AUTHN_ENTRYPOINTS" name="soapComponent">
            <vcl:configItem name="authNLocation" value="https://192.168.1.10:53714/AuthenticationService/authentication"/>
            <vcl:configGroup name="ports">
                <vcl:configGroup name="port">
                    <vcl:configItem value="1.9" name="portStandardVersion"/>
                    <vcl:configItem value="http://192.168.1.10:53714/VDBManagementService/authnEntrypoints" name="uri"/>
                </vcl:configGroup>
            </vcl:configGroup>
        </vcl:configGroup>
    </vcl:configGroup>
</vcl:VCL>
```

20.40 Configuration Management Parameters <10028–20210409, Pro>

The appendix titled "VICTORY WSDLs" provides the name of the file containing the web service details for the configuration management parameters. A textual description of the configuration management parameters is shown below.

Any security settings applied to an individual operation MUST also be present for the Get/Set Configuration. Get/Set Configuration MUST NOT allow a user to bypass ANY security settings.

1. ***Configuration***
    a. This parameter indicates the configuration of a VICTORY component on the VDB.
    b. The configuration shall be a valid VCL ID as defined in the section titled "VCL Schema."
        i. On a Set() operation:
            1. The parameter shall be a VCL ID describing, at a minimum, the configuration required by the component receiving the request.
            2. Optional Mutable, Configurable, or Volatile Configuration Item elements do not have to be included in a Set() Configuration request unless otherwise specified in the applicable Configuration Description specification. If an element is not included, it shall remain unchanged except in the event that not including an element represents the removal of its respective value, as specified in the applicable Configuration Description specification. If they are included, they shall not be ignored.
            3. If a Configuration Item or Configuration Group element does not have a Configuration Type, i.e., neither it or any of its parent elements have a Configuration Type, and that element appears as a list or as optional inside the tree view of the applicable Configuration Description specification, it should not be modified from how it appears inside the Get() configuration response. This includes the existence of optional elements, the quantity of elements that are a list, or any attributes associated with the element. Any modifications made to this element inside a Set() Configuration request shall be ignored and the element shall not be changed inside the VCL ID.
            4. A structural error in the configuration description (e.g., duplicate configuration items where only one configuration item is allowed, missing required items, etc.) shall result in a GENERIC_FAULT error.
            5. A string bounds error in the configuration description may result in a OUT_OF_RANGE error.
                a. A string bounds error occurs when the Configuration Item or Configuration Group string bounds requirements described in the section titled "String Bounds" under "VICTORY Data Types" are not satisfied.
            6. If no structural or string bounds error occurs, the component shall configure itself by parsing its portion of the VCL ID as defined in the applicable component configuration description section.
                a. If no errors are encountered, a single NO_ERROR error, along with the unique name of the component, shall be included in the error response.
                b. All errors encountered while attempting to set a Mutable, Configurable, or Volatile Configuration Item value shall be included in the error response along with the unique name of the component.

                > ***NOTE:*** *Optional Mutable elements whose respective management parameter is returning a NOT_SUPPORTED error when being changed inside its applicable management interface should return a similar NOT_SUPPORTED error when provided inside a Set() Configuration request.*

                    i. Setting the value of a Configurable parameter the component does not have the ability to change shall result in a NOT_SUPPORTED error.
                    ii. Setting a parameter that is restricted due to security settings shall result in a NOT_AUTHORIZED error.
                    iii. Each error shall contain one of the errors defined in the section titled "Management Interface Error Types."

c. Regardless of whether or not errors are encountered, the component shall continue processing its portion of the VCL ID.

ii. On a Get() operation:

1. This parameter generates a VCL ID describing the current configuration of the component receiving the request.
2. Optional elements do not have to be returned inside the Get() Configuration response if they have no value. If a value exists (e.g., has been previously "set," or had an implementation-dependent initial value), it shall be returned inside the Get() Configuration response.
3. All errors encountered while attempting to get a Configuration Item value shall be included in the error response along with the unique name of the component.
    a. Getting a parameter that is restricted due to security settings shall result in a NOT_AUTHORIZED error.
    b. Each error shall contain one of the errors defined in the section titled "Management Interface Error Types."

> ***NOTE:*** *It is possible that parameters with Configuration Types of Configurable or Volatile may have been updated through a Set() Configuration request, but have not yet been applied since the component has not been restarted. In this case, the component should return the current values for these parameters, not the values that will take effect upon a restart.*

c. Operations include Get() and Set().

2. ***Configuration Version***
    a. This parameter identifies the version of the configuration used by the management interface.

> ***NOTE:*** *The configuration version is an implementation-specific versioning of a VCL instance document.*

    b. The bounds of the configuration version are described in the section titled "String Bounds" under "VICTORY Data Types."
    c. Operations include Get().

> ***NOTE:*** *The section titled "Management Interface Error Types" provides additional details on the errors defined for the configuration management parameters along with a description of additional errors that were not specified but may be returned.*

APPENDIX A: ZEROCONF ANNOUNCEMENT STRINGS USED BY VICTORY

Table 292. Zeroconf Announcement Strings

VICTORY Service / Component Name	Announcement String Type	Announcement String Subtype	DNS-TXT[1]	Existing Standard
End Node Component	_workstation._tcp	N/A	N/A	Yes
Network Time Protocol (NTP) Service	_ntp._udp	N/A	N/A	Yes
WebDAV Shared Storage	_webdav._tcp _webdavs._tcp (for TLS)	N/A	u p path	Yes
Network File System (NFS) Shared Storage	_nfs._tcp	N/A	path	Yes
VICTORY-Authored Services	_vic-mgt._tcp	N/A	txtvers(3) wsdls path tls	No
Generic End Node	_vic-mgt._tcp	_gen-end-node	txtvers(3) wsdls path tls	No
Shared Processing Unit	_vic-mgt._tcp	_spu	txtvers(3) wsdls path tls	No
VDB Management Service	_vic-mgt._tcp	_vdb	txtvers(3) wsdls path tls	No
Attribute Store Service	_vic-mgt._tcp	_att-store	txtvers(3) wsdls path tls	No

VICTORY Service / Component Name	Announcement String Type	Announcement String Subtype	DNS-TXT[1]	Existing Standard
Authentication Service	_vic-mgt._tcp	_auth	txtvers(3) wsdls path tls	No
Data at Rest Encryptor (DARE)	_vic-mgt._tcp	_dare	txtvers(3) wsdls path tls	No
Data Signing Service	_vic-mgt._tcp	_data-sign	txtvers(3) wsdls path tls	No
Host-Based Antivirus	_vic-mgt._tcp	_host-anti	txtvers(3) wsdls path tls	No
Host-Based Firewall	_vic-mgt._tcp	_host-fwl	txtvers(3) wsdls path tls	No
Intrusion Detection and Prevention System (IDPS)	_vic-mgt._tcp	_idps	txtvers(3) wsdls path tls	No
Network Firewall	_vic-mgt._tcp	_net-fwl	txtvers(3) wsdls path tls	No

VICTORY Service / Component Name	Announcement String Type	Announcement String Subtype	DNS-TXT[1]	Existing Standard
Policy Decision Service	_vic-mgt._tcp	_pol-dec	txtvers(3) wsdls path tls	No
Policy Enforcement Service	_vic-mgt._tcp	_pol-enf	txtvers(3) wsdls path tls	No
Policy Store Service	_vic-mgt._tcp	_pol-store	txtvers(3) wsdls path tls	No
Signed Data Verification Service	_vic-mgt._tcp	_data-verif	txtvers(3) wsdls path tls	No
Audio Service	_vic-mgt._tcp	_audio	txtvers(3) wsdls path tls	No
Automotive System	_vic-mgt._tcp	_auto-system	txtvers(3) wsdls path tls	No
Camera Gimbal	_vic-mgt._tcp	_gimbal	txtvers(3) wsdls path tls	No

VICTORY Service / Component Name	Announcement String Type	Announcement String Subtype	DNS-TXT[1]	Existing Standard
Direction of Travel Service	_vic-mgt._tcp	_dot	txtvers(3) wsdls path tls	No
Orientation Service	_vic-mgt._tcp	_orientation	txtvers(3) wsdls path tls	No
Position Service	_vic-mgt._tcp	_position	txtvers(3) wsdls path tls	No
Power Distribution System	_vic-mgt._tcp	_power-dist	txtvers(3) wsdls path tls	No
Legacy Radio	_vic-mgt._tcp	_legacy-radio	txtvers(3) wsdls path tls	No
Remote Weapon Station	_vic-mgt._tcp	_rws	txtvers(3) wsdls path tls	No
Threat Detection and Reporting Service	_vic-mgt._tcp	_threat	txtvers(3) wsdls path tls	No

VICTORY Service / Component Name	Announcement String Type	Announcement String Subtype	DNS-TXT[1]	Existing Standard
Video or Image Sensor	_vic-mgt._tcp	_video	txtvers(3) wsdls path tls	No
Video Encoder	_vic-mgt._tcp	_video-enc	txtvers(3) wsdls path tls	No
Mission Recorder	_vic-mgt._tcp	_recorder	txtvers(3) wsdls path tls	No
Streaming Audio Data Source	_vic-mgt._tcp	_audio-str	txtvers(3) wsdls path tls	No
Single-Instance Audio Data Source	_vic-mgt._tcp	_audio-si	txtvers(3) wsdls path tls	No
Intercom	_vic-mgt._tcp	_intercom	txtvers(3) wsdls path tls	No
Data Logger	_vic-mgt._tcp	_data-logger	txtvers(3) wsdls path tls	No

VICTORY Service / Component Name	Announcement String Type	Announcement String Subtype	DNS-TXT[1]	Existing Standard
Vehicle Configuration Service	_vic-mgt._tcp	_veh-cfg	txtvers(3) wsdls path tls	No
LT2 Adapter	_vic-mgt._tcp	_lt2-adptr	txtvers(3) wsdls path tls	No
Audio Streaming Data Interface (SIP, for VoIP)	_sip._udp	N/A	[2]	Yes
Video Streaming Data Interface	_rtsp._tcp	N/A	path [3]	Yes

Table 293. Announcement String TXT Field Descriptions

TXT Field	Usage/Meaning	Required	txtvers Version Added
txtvers	Version number of the DNS-TXT fields used by the VICTORY specification. Currently, the only values accepted by VICTORY for the “txtvers” field are 1, 2, and 3. When a new TXT field is added, or the meaning of an existing TXT field is changed, the “txtvers” field for that announcement will be incremented for backwards compatibility. In Table 292, the value that is expected is reported in parentheses. For example, if the entry is txtvers(3) for an announcement string, then it is required that an implementation of that announcement for this version of VICTORY will include txtvers=3 in the TXT field. For reference, the “txtvers Version Added” field records when a TXT field value was first added to VICTORY. Knowing when a field was added allows a client to understand which fields it is expected to parse when it receives an announcement string with an equivalent txtvers value.	Yes	1
wsdls	A comma-separated list of URLs (minus the hostname[4] and port, in most cases). Each entry shall refer to the location of one of the parameter-describing WSDLs that comprise the VICTORY Management Interface that is being announced. NOTE: The hostname[4] and port shall be excluded in all cases where they are the same as the hostname[4] and port that is already being announced by the zeroconf DNS-SD entry. This is required by the DNS-SD and mDNS specifications.	No	1
path	The “filesystem path” to the resource being announced. In all cases, this must be a relative path, with the assumption that the full URI of the resource would include the announced hostname[4] and port in addition to the value of the “path” field. This field is common in both VICTORY-defined service announcements and existing standard announcements. For all VICTORY Management Interface announcements (_vic-mgt._tcp), this field shall point to the location of the “Version Information Management Parameters” network endpoint, presuming that the endpoint is running at a full URL, and not at the base URL (the hostname[4] and port only).	No	2
tls	Indicates whether Transport Layer Security (TLS) is enabled for the service or component being announced. If present, it must be either *True* or *False*. If absent, it is presumed to be *True*.	No	3

TXT Field	Usage/Meaning	Required	txtvers Version Added
u	Username (used by WebDAV)	No	N/A
p	Password (used by WebDAV)	No	N/A

APPENDIX B: VICTORY WSDLS

Table 294. VICTORY WSDLs

Specification Title	WSDL File Name
Attribute Store Management Specification	AttributeStore.wsdl
Authentication Entrypoints	AuthenticationEntrypoints.wsdl
Authentication Service Management Specification	Authentication.wsdl
Authorization Architecture Entrypoints	AuthorizationArchitectureEntrypoints.wsdl
Automotive System Management Parameters	AutomotiveData.wsdl
Camera Gimbal Data Management Parameters	CameraGimbalData.wsdl
Camera Gimbal Device Management Parameters	CameraGimbal.wsdl
Configuration Management Parameters	Configuration.wsdl
Configuration Management Parameters for the VDB Management Service	Configuration.wsdl
DAR Encryptor Management Specification	DARE.wsdl
Data Logger Management Parameters	DataLogger.wsdl
Data Signing Service Management Specification	DataSigning.wsdl
Device Information Management Parameters	DeviceInformation.wsdl
Direction of Travel Data Management Parameters	DirectionOfTravelData.wsdl
Generic End Node Management Parameters	GenericEndNode.wsdl
Health Level Management Parameters	HealthLevel.wsdl
Host-Based Antivirus Management Specification	HostBasedAntivirus.wsdl
Host-Based Firewall Management Specification	HostBasedFirewall.wsdl
IDPS Management Specification	IDPS.wsdl
Intercom Management Parameters	Intercom.wsdl
Legacy Radio Device Management Parameters	LegacyRadioDevice.wsdl
Live Training Transformation Adapter Management Parameters	Lt2Adapter.wsdl
Mission Recorder Device Management Parameters	MissionRecorder.wsdl
Network Firewall Management Specification	Firewall.wsdl
Orientation Data Management Parameters	OrientationData.wsdl
Policy Decision Management Specification	PolicyDecision.wsdl
Policy Enforcement Management Specification	PolicyEnforcement.wsdl
Policy Store Management Specification	PolicyStore.wsdl

Specification Title	WSDL File Name
Position Data Management Parameters	PositionData.wsdl
Power Distribution System Data Management Parameters	PowerDistributionSystemData.wsdl
Power Distribution System Device Management Parameters	PowerDistributionSystemDevice.wsdl
RWS Device Management Parameters	RemoteWeaponStationDevice.wsdl
Shared Processing Unit Management Parameters	SharedProcessingUnit.wsdl
Signed Data Verification Service Management Specification	SignedDataVerification.wsdl
SINCGARS Device Management Parameters	SincgarsDevice.wsdl
Single-Instance Audio Data Management Parameters	SingleInstanceAudioData.wsdl
Streaming Audio Data Management Parameters	StreamingAudioData.wsdl
Syslog-Based Health Publishing Management Specification	HealthPublishing.wsdl
Threat Detection and Reporting Management Parameters	ThreatData.wsdl
Vehicle Configuration Service Management Parameters	VehicleConfiguration.wsdl
Version Information Management Parameters	VersionInformation.wsdl
VICTORY Data Management Parameters	VICTORYData.wsdl
Video Data Stream Management Parameters	VideoData.wsdl
Video Encoder Management Parameters	VideoEncoder.wsdl
Video or Image Sensor Management Parameters	VideoSensor.wsdl

APPENDIX C: VICTORY SCHEMAS

Table 295. VICTORY Schemas

Schema Name	Schema File Name
Attribute Store Management Types	AttributeStoreManagementTypes.xsd
Authentication Entrypoints Types	AuthenticationEntrypointsTypes.xsd
Authentication Management Types	AuthenticationManagementTypes.xsd
Authorization Architecture Entrypoints Types	AuthorizationArchitectureEntrypointsTypes.xsd
Data at Rest Encryptor Management Types	DAREManagementTypes.xsd
Data Signing Management Types	DataSigningManagementTypes.xsd
Firewall Management Types	FirewallManagementTypes.xsd
Host-Based Antivirus Types	HostBasedAntivirusTypes.xsd
Host-Based Firewall Management Types	HostBasedFirewallManagementTypes.xsd
Information Assurance Types	InformationAssuranceTypes.xsd
Intrusion Detection and Prevention System Management Types	IDPSManagementTypes.xsd
OASIS Access Control XACML 2.0 Context	access_control-xacml–2.0-context-schema-os.xsd
OASIS Access Control XACML 2.0 Policy	access_control-xacml–2.0-policy-schema-os.xsd
OASIS SAML Schema Assertion 2.0	saml-schema-assertion–2.0.xsd
OASIS SAML Schema Protocol 2.0	saml-schema-protocol–2.0.xsd
OASIS XACML 2.0 Profile SAML 2.0 v2 Assertion	xacml–2.0-profile-saml2.0-v2-schema-assertion-wd–13.xsd
OASIS XACML 2.0 Profile SAML 2.0 v2 Protocol	xacml–2.0-profile-saml2.0-v2-schema-protocol-wd–13.xsd
Policy Decision Management Types	PolicyDecisionManagementTypes.xsd
Policy Enforcement Management Types	PolicyEnforcementManagementTypes.xsd
Policy Store Management Types	PolicyStoreManagementTypes.xsd
Signed Data Verification Management Types	SignedDataVerificationManagementTypes.xsd
VICTORY Configuration Language	VICTORYConfigurationLanguage.xsd
VICTORY Data Messages	VDM.xsd
VICTORY Data Types	VICTORYDataTypes.xsd
VICTORY Management Types	VICTORYManagementTypes.xsd
VICTORY Messages	VICTORYMessages.xsd
VICTORY Shared Types	VICTORYSharedTypes.xsd
XML Encryption	xenc-schema.xsd

Schema Name	Schema File Name
XML Multimedia Internet Message Extensions	xmlmime.xsd
XML Signature	xmldsig-core-schema.xsd

APPENDIX D: LEAP SECONDS TABLE FROM NIST

The National Institute of Standards and Technology (NIST) leap seconds table is available for download at: https://www.ietf.org/timezones/data/leap-seconds.list.

APPENDIX E: VICTORY SPECIFICATION TAG CROSS-REFERENCE

The VICTORY specification document uses VICTORY Type Tags (VT Tags) to identify the system, component, and interface type specification tables. The VT Tag is a 5-digit alphanumeric identification tag with a numeric range of 50000 - 99999 with a prefix of "VT." The specification document includes Specification Tags (Spec IDs) to track the individual specifications. The Spec ID is a 5-digit number with a range of 01001 - 99999.

This appendix provides a detailed list of the VT Tags and Spec IDs that have been assigned, and includes a note if a particular Tag/ID has been retired.

Table 296. VICTORY Type Tags

VT Tag	VT Name
VT50000	Time Synchronization Service
VT50100	Position Service
VT50200	Orientation Service
VT50300	Direction of Travel Service
VT50400	Authentication and Authorization Security Specifications
VT50401	Authentication Service
VT50420	Web Service Specifications
VT50430	Authorization Security Specifications
VT50500	Policy Decision Service
VT50600	Policy Enforcement Service
VT50700	Policy Store Service
VT50800	Attribute Store Service
VT50900	Switch
VT51000	Router
VT51100	[Retired]
VT51300	Network Firewall
VT51400	Host-Based Firewall
VT51500	Inline Network Encryptor
VT51600	Intrusion Detection and Prevention System
VT51700	[Retired]
VT51800	[Retired]
VT51900	Shared Processing Unit
VT52000	Network Attached Storage
VT52100	Generic End Node

VT Tag	VT Name
VT52200	VDB Management Service
VT52300	[Retired]
VT52303	VDB Connector, Signaling, and Physical Media Interface
VT52400	Data at Rest Encryptor
VT52500	Host-Based Antivirus
VT52600	[Retired]
VT52700	Data Signing Service
VT52800	Signed Data Verification Service
VT55000	VICTORY Configuration Language Instance Document
VT55100	VDB Configuration Description
VT55200	Information Assurance Configuration Description
VT55300	C4ISR/EW System Configuration Description
VT55400	Platform System Configuration Description
VT60000	Single-Instance Audio Data Source
VT60001	Streaming Audio Data Source
VT60100	Intercom
VT60200	Legacy Radio
VT60300	Video or Image Sensor
VT60400	Video or Image Encoder
VT60500	Camera Gimbal
VT60600	Threat Detection and Reporting Service
VT60700	Mission Recorder
VT60800	Data Radio
VT60900	Electronic Warfare Device
VT61000	GPS Receiver
VT61100	[Retired]
VT61200	[Retired]
VT61300	[Retired]
VT61500	[Retired]
VT61600	Assured PNT
VT62000	Data Logger

VT Tag	VT Name
VT70000	Automotive System
VT70100	Power Distribution System
VT70200	Remote Weapon Station
VT70800	Vehicle Configuration Service
VT70900	Active Protection System
VT71000	Live Training Transformation Adapter
VT71100	[Retired]
VT71200	[Retired]
VT71300	[Retired]
VT71400	[Retired]
VT91010	Low Latency Switch
VT92010	MORA Device
VT92013	MORA Signal Resource Manager
VT92016	MORA Signal Port Manager

Table 297. VICTORY Specification Tags

Spec ID	Specification Title
01001	10BASE-T
01002	10BASE-FL
01003	100BASE-TX
01004	100BASE-FX
01005	100BASE-LX10
01006	1000BASE-T
01007	1000BASE-SX
01008	1000BASE-LX
01009	10GBASE-X/10GBASE-T
01010	10GBASE-R
01011	Copper Auto-Negotiation
01012	[Retired]
01013	Frame Structure
01014	Media Access Control
01015	Logical Link Control

Spec ID	Specification Title
01016	Link Layer Switching
01017	Link Layer Flow Control
01018	Internet Protocol Version 4 Specification
01019	Internet Protocol Version 6 Specification
01020	[Retired]
01021	Internet Control Message Protocol for IPv4
01022	[Retired]
01023	[Retired]
01024	IGMP Snooping in IPv4
01025	IP Layer Security
01026	Transmission Control Protocol
01027	User Datagram Protocol
01028	[Retired]
01029	VDB Subnetting Restrictions
01030	[Retired]
01031	[Retired]
01032	[Retired]
01033	IPv4 Address Space
01034	IPv6 Address Space
01035	Network Port Assignment
01036	QoS Mapping to Components
01037	DiffServ
01038	[Retired]
01039	DiffServ Code Point (DSCP) Assignments
01040	[Retired]
01041	[Retired]
01042	VDB Node Identification for IPv4
01043	[Retired]
01044	Zero Configuration Networking
01045	[Retired]
01046	[Retired]

Spec ID	Specification Title
01047	[Retired]
01048	[Retired]
01049	[Retired]
01050	[Retired]
01051	[Retired]
01052	[Retired]
01053	[Retired]
01054	Classless Inter-Domain Routing and Subnetting for IPv4
01055	[Retired]
01056	[Retired]
01057	[Retired]
01058	[Retired]
01059	Internet Control Message Protocol for IPv6
01060	MLD Snooping in IPv6
01061	Static Host Addressing
01062	IPv4 Datagram Transmission
01063	IPv6 Datagram Transmission
01064	Path MTU Discovery for IPv4
01065	Path MTU Discovery for IPv6
01066	IPv4 Private Address Space
01067	IPv6 Private Address Space
01068	IPv4 Network Address Translation
01069	VDB IPv4 Address Translation
01070	VDB IPv6 Address Translation
01071	IPv4 Dynamic Host Addressing
01072	IPv6 Dynamic Host Addressing
01073	Protocol Independent Multicast - Sparse Mode (PIM-SM)
01074	Protocol Independent Multicast - Dense Mode (PIM-DM)
01075	Internet Group Management Protocol (IGMP)
01076	Multicast Listener Discovery (MLD)
01077	Virtual Local Area Network (VLAN)

Spec ID	Specification Title
01078	[Retired]
01079	Ethernet
01080	Dual Ethernet
01081	Other Ethernet Connectors
01082	Subnetting for IPv6
01083	VDB Node Identification for IPv6
01084	[Retired]
01085	[Retired]
01086	IPv4 Static Routes
01087	IGMP Forwarding Proxy
01088	RIP
01089	OSPF Version 2
01090	BGP
01091	IPv6 Static Routes
01092	MLD Forwarding Proxy
01093	RIPng
01094	OSPF for IPv6
01095	1000BASE-KX
01096	10GBASE-KR
02001	[Retired]
02002	Network Time Protocol
02003	Precision Time Protocol
02004	Health Monitoring of Network Time Protocol
02005	Network Time Protocol Management Specification
02006	Health Monitoring of Precision Time Protocol
02007	Precision Time Protocol Management Specification
02008	PTP End-to-End Transparent Clock Mode for Infrastructure Components
03001	Mission Recorder Core Functionality
10001	SOAP Compliance and Standards
10002	Web Services Description Language
10003	VICTORY Schemas and WSDLs

Spec ID	Specification Title
10004	SNMP Core Specifications
10005	Syslog Protocol
10006	[Retired]
10007	Version Information Management Parameters
10008	VICTORY Data Management Parameters
10009	Device Information Management Parameters
10010	[Retired]
10011	Syslog Protocol Data Parameters
10012	Syslog-Based Asynchronous Notification Publishing Interface
10013	Management Interface Error Types
10014	Generic Component Health Levels for End Users
10015	Generic End Node Component Performance Status Interface
10016	Generic Infrastructure Component Health Status Specification
10017	[Retired]
10018	Data Confidentiality & Integrity for Management Interfaces
10019	Data Authentication for Management Interfaces
10020	[Retired]
10021	[Retired]
10022	Health Levels for End Users
10023	Syslog-Based Connectionless Health Publishing Specification
10024	Security Event Log Format
10025	Security Event Log Transfer
10026	Security Event Log Storage and Protection
10027	Log Analysis Using PRI and SD-PARAM Values
10028	Configuration Management Parameters
10029	Configuration Management Parameters for the VDB Management Service
10030	Health Level Management Parameters
10031	Version Information Complex Types
10032	VICTORY Management Interface Types
10033	VICTORY SOAP Component Types
11001	[Retired]

Spec ID	Specification Title
11002	[Retired]
11003	[Retired]
11004	[Retired]
11005	Syslog-Based Health Publishing Management Specification
11006	[Retired]
11007	[Retired]
11008	[Retired]
12001	[Retired]
12002	[Retired]
12003	[Retired]
12004	[Retired]
12005	[Retired]
12006	Generic Infrastructure Component Management Interface
13001	[Retired]
13002	[Retired]
13003	Generic End Node Management Parameters
13004	Generic End Node Complex Types
13005	SPU Mid-Level API
13006	Shared Processing Unit Management Parameters
13007	[Retired]
13008	[Retired]
13009	POSIX
13010	Qt Requirements for the Headless SPU API Profile
13011	Qt Requirements for the Display Unit SPU API Profile
13012	libBoost
13013	Shared Processing Unit Complex Types
14001	[Retired]
14002	INE Management Specification
14003	Network Firewall Management Specification
14004	[Retired]
14005	[Retired]

Spec ID	Specification Title
14006	IDPS Management Specification
14007	[Retired]
14008	[Retired]
14009	[Intentionally left blank]
14010	[Retired]
14011	[Retired]
14012	[Retired]
14013	Policy Enforcement Management Specification
14014	Policy Decision Management Specification
14015	Attribute Store Management Specification
14016	DAR Encryptor Management Specification
14017	[Retired]
14018	Web Services Framework
14019	Web Services Security Specification
14020	Host-Based Firewall Management Specification
14021	[Retired]
14022	Policy Store Management Specification
14023	Host-Based Antivirus Management Specification
14024	[Retired]
14025	Data Signing Service Management Specification
14026	Signed Data Verification Service Management Specification
14027	Security Event Log Management Specification
14028	Authentication Service Management Specification
15001	Direction of Travel Data Management Parameters
15002	Orientation Data Management Parameters
15003	Position Data Management Parameters
15004	Threat Detection and Reporting Management Parameters
15005	[Retired]
15006	[Retired]
15007	Streaming Audio Data Management Parameters
15008	Video or Image Sensor Management Parameters

Spec ID	Specification Title
15009	[Retired]
15010	Video Data Stream Management Parameters
15011	VISA Complex Types
15012	Video Encoder Management Parameters
15013	Electronic Warfare Device Management Parameters
15014	Electronic Warfare Data Management Parameters
15015	Mission Recorder Device Management Parameters
15016	GPS Receiver Device Management Parameters
15017	GPS Receiver Data Management Parameters
15018	Single-Instance Audio Data Management Parameters
15019	Intercom Management Parameters
15020	Camera Gimbal Device Management Parameters
15021	Data Logger Management Parameters
15022	Data Logger FTP Specification
15023	Vehicle Configuration Service Management Parameters
15024	Data Radio SNMP Specification
15025	Legacy Radio Device Management Parameters
15026	Legacy Radio Complex Types
15027	IR Video or Image Sensor Management Parameters
15028	Camera Gimbal Data Management Parameters
15029	Assured PNT System Management Parameters
15030	Assured PNT Data Management Parameters
16001	Automotive System Management Parameters
16002	RWS Device Management Parameters
16003	[Retired]
16004	Power Distribution System Data Management Parameters
16005	Automotive System Management Parameters – V1.6
16006	Power Distribution System Device Management Parameters
16007	Active Protection System Data Management Parameters
16008	Active Protection System Management Parameters
16009	Live Training Transformation Adapter Management Parameters

Spec ID	Specification Title
20001	Application Layer Data Encoding
20002	Application Layer Message Encapsulation
20003	VICTORY Data Types
20004	Timestamp Format for Application Layer Data
20005	Conversion from Textual to Binary Timestamps
20006	Conversion from Binary to Textual Timestamps
20007	Direction of Travel Data Specification
20008	Orientation Data Specification
20009	Position Data Specification
20010	VICTORY Frame of Reference
20011	Threat Detection and Reporting Data Specification
20012	Direction of Travel Complex Types
20013	Orientation Complex Types
20014	Position Complex Types
20015	Threat Detection and Reporting Complex Types
20016	Streaming Audio Data Specification
20017	Standard Definition Streaming Video on the VDB
20018	Camera Orientation Data Specification
20019	RWS Data Specification
20020	RWS Interface Complex Types
20021	[Retired]
20022	Automotive System Interface Complex Types
20023	[Retired]
20024	[Retired]
20025	Electronic Warfare Interface Complex Types
20026	Power Distribution System Interface Complex Types
20027	Mission Recorder Interface Complex Types
20028	WebDAV
20029	NFS
20030	[Retired]
20031	GPS Receiver Complex Types

Spec ID	Specification Title
20032	GPS Receiver Status VICTORY Data Message
20033	[Retired]
20034	GPS Receiver Ionospheric Corrections VICTORY Data Message
20035	GPS Receiver Satellite Health Status VICTORY Data Message
20036	Electronic Warfare Line of Bearing VICTORY Data Message
20037	Electronic Warfare Geolocation VICTORY Data Message
20038	Electronic Warfare Object Detection VICTORY Data Message
20039	Electronic Warfare Device Specific VICTORY Data Message
20040	Automotive Ambient Conditions VICTORY Data Message
20041	Automotive Dash Display VICTORY Data Message
20042	Automotive Direct Lamp Control Data 1 VICTORY Data Message
20043	Automotive Direct Lamp Control Data 2 VICTORY Data Message
20044	Automotive Distance VICTORY Data Message
20045	Automotive Electrical Power 1 VICTORY Data Message
20046	Automotive Electrical Power 2 VICTORY Data Message
20047	Automotive Electronic Transmission Controller 2 VICTORY Data Message
20048	Automotive Engine Fluid Level and Pressure 1 VICTORY Data Message
20049	Automotive Engine Hours and Revolutions VICTORY Data Message
20050	Automotive Engine Speed and Load Factor VICTORY Data Message
20051	Automotive Engine Temperature 1 VICTORY Data Message
20052	Automotive Engine Temperature 2 VICTORY Data Message
20053	Automotive Engine Valve VICTORY Data Message
20054	Automotive Fuel Consumption VICTORY Data Message
20055	Automotive Fuel Economy VICTORY Data Message
20056	Automotive Active Diagnostic Trouble Codes VICTORY Data Message
20057	Automotive Alternator Temperature VICTORY Data Message
20058	Automotive Battery Temperature VICTORY Data Message
20059	Automotive Cab Message 1 VICTORY Data Message
20060	Automotive Combination Vehicle Weight VICTORY Data Message
20061	Automotive Electronic Engine Controller 1 VICTORY Data Message
20062	Automotive Electronic Engine Controller 2 VICTORY Data Message

Spec ID	Specification Title
20063	Automotive Fan Drive 1 VICTORY Data Message
20064	Automotive Tire Condition VICTORY Data Message
20065	Automotive Torque/Speed Control 1 VICTORY Data Message
20066	Automotive System Interface Simple Types
20067	Single-Instance Audio Data Specification
20068	Intercom Complex Types
20069	Power System Device VICTORY Data Message
20070	Power System Device Threshold VICTORY Data Message
20071	Platform Battery VICTORY Data Message
20072	Platform Power Generation VICTORY Data Message
20073	Platform Power VICTORY Data Message
20074	[Retired]
20075	GPS Datum Values
20076	Data Logger Data Specification
20077	Data Logger Complex Types
20078	Data Logger Pcap Log File Format
20079	Vehicle Configuration Service Complex Types
20080	Active Protection System Engagement Report VICTORY Data Message
20081	Active Protection System Complex Types
20082	Home Position Data Specification
20083	Active Protection System Status VICTORY Data Message
20084	Live Training Transformation Adapter Data Specification
20085	GPS Receiver Status 2 VICTORY Data Message
20086	GPS Receiver Satellite Health Status 2 VICTORY Data Message
20087	Assured PNT Blended Navigation VICTORY Data Message
20088	Assured PNT Warnings VICTORY Data Message
20089	Assured PNT Complex Types
20090	Assured PNT NAVWAR SA VICTORY Data Message
30001	[Retired]
30002	[Retired]
30003	[Retired]

Spec ID	Specification Title
30004	[Retired]
30005	Position Service Configuration Description
30006	Orientation Service Configuration Description
30007	Direction of Travel Service Configuration Description
30008	Streaming Audio Data Source Configuration Description
30009	Single-Instance Audio Data Source Configuration Description
30010	Intercom Configuration Description
30011	[Retired]
30012	Video or Image Sensor Configuration Description
30013	Video or Image Encoder Configuration Description
30014	Camera Gimbal Configuration Description
30015	Threat Detection and Reporting Service Configuration Description
30016	Mission Recorder Configuration Description
30017	Electronic Warfare Device Configuration Description
30018	GPS Receiver Configuration Description
30019	Automotive System Configuration Description
30020	Power Distribution System Configuration Description
30021	Remote Weapon Station Configuration Description
30022	Attribute Store Service Configuration Description
30023	Policy Decision Service Configuration Description
30024	Policy Enforcement Service Configuration Description
30025	Policy Store Service Configuration Description
30026	Authentication Service Configuration Description
30027	Host-Based Antivirus Configuration Description
30028	Host-Based Firewall Configuration Description
30029	Network Firewall Configuration Description
30030	Intrusion Detection and Prevention System Configuration Description
30031	Data at Rest Encryptor Configuration Description
30032	Data Signing Service Configuration Description
30033	Signed Data Verification Service Configuration Description
30034	[Retired]

Spec ID	Specification Title
30035	[Retired]
30036	Assured PNT Configuration Description
30037	[Retired]
30038	[Retired]
30039	[Retired]
30040	Generic End Node Configuration Description
30041	Network Attached Storage Configuration Description
30042	Shared Processing Unit Configuration Description
30043	VDB Management Service Configuration Description
30044	VCL Schema
30045	VCL Instance Document Requirements
30046	Tree View Requirements
30047	Common Component Configuration Description Content
30048	Configuration Type
30049	Data Logger Configuration Description
30050	Vehicle Configuration Service Configuration Description
30051	Active Protection System Configuration Description
30052	Live Training Transformation Adapter Configuration Description
30053	Legacy Radio Configuration Description
40001	Authentication Methods
40002	[Retired]
40003	Authorization Model
40004	[Retired]
40005	[Retired]
40006	[Retired]
40007	[Retired]
40008	[Retired]
40009	[Retired]
40010	[Retired]
40011	[Retired]
40012	Authorization Architecture Entrypoints

Spec ID	Specification Title
40013	Authentication Model
40014	Authentication Entrypoints
43001	[Retired]
44001	[Retired]
47001	Coaxial Cable Specification
47002	Optical Fiber Specification
48001	Standard for Ethernet Specification
48002	Virtual LANs Specification
48005	Standard for Internet Protocol and User Datagram Protocol Specification
48006	Multicast Dissemination Over the Network Specification
48007	Jumbo Frame Packet Support Specification
48008	MDM Traffic Marking Specification
48102	10GBASE-SR
48103	10GBASE-KX4
48112	40GBASE-KR4
48113	40GBASE-SR4
48114	100GBASE-KR4
48201	MORA Low Latency Bus Performance Specification
48605	MDM Data Encoding Specification
48606	MDM Encapsulation Specification
48607	MDM VRT Message Structure Specification
48608	MDM VRT Signal Data Packet Type Specification
48609	MDM VRT Context Packet Type Specification
48610	MDM VRT Command Packet Type Specification
48611	MDM Acknowledgement Message (Type 2) Specification
48612	MDM Time of Day Message (Type 3) Specification
48613	MDM Signal Port User ID Message (Type 4) Specification
48614	MDM Health Message (Type 5) Specification
48615	MDM Command Message (Type 6) Specification
48616	MDM VRT Context and Command Fields Specification
48617	MDM Switch Group User ID Message (Type 7) Specification

Spec ID	Specification Title
48801	MDM Multicast Dissemination Specification
49005	Radio Frequency Distribution Management Parameters
49011	MORA Device Management Parameters
49012	Waveform Management Parameters
49014	MORA Signal Resource Management Parameters
49016	MORA Common Complex Types
49017	Waveform Complex Types
49018	Radio Frequency Distribution Complex Types
49101	Syslog Protocol Data Parameters for MORA
49102	Version Information Management Parameters for MORA
49103	MORA Schemas and WSDLs

APPENDIX F: VDM DATA TYPE VALUES

Table 298. VDM Data Type Values

Component Type	Message Name	Data Type	Message Value	Version Applicability				
				V1.4	V1.6	V1.7	V1.8	V1.9
Invalid	N/A	N/A	0x0000, 0xFFFF	N/A	N/A	N/A	N/A	N/A
Position Service	Position	POSITION	0x0001	X	X	X	X	X
Orientation Service	Orientation	ORIENTATION	0x0002	X	X	X	X	X
Direction of Travel Service	Direction of Travel	DIRECTION_OF_TRAVEL	0x0003	X	X	X	X	X
Threat Detection and Reporting Service	Threat	THREAT	0x0004	X	X	X	X	X
Remote Weapon Station	Remote Weapon Station	REMOTE_WEAPON_STATION	0x0005	X	X	X	X	X

Component Type	Message Name	Data Type	Message Value	Version Applicability				
				V1.4	V1.6	V1.7	V1.8	V1.9
Automotive System	Ambient Conditions	AMBIENT_CONDITIONS	0x0006	X	X	X	X	X
	Dash Display	DASH_DISPLAY	0x0007	X	X	X	X	X
	Distance	DISTANCE	0x0008	X	X	X	X	X
	Electrical Power 1	ELECTRICAL_POWER_1	0x0009	X	X	X	X	X
	Electrical Power 2	ELECTRICAL_POWER_2	0x000A	X	X	X	X	X
	Engine Fluid Level and Pressure 1	ENGINE_FLUID_LEVEL_AND_PRESSURE_1	0x000B	X	X	X	X	X
	Engine Hours And Revolutions	ENGINE_HOURS_AND_REVOLUTIONS	0x000C	X	X	X	X	X
	Engine Speed And Load Factor	ENGINE_SPEED_AND_LOAD_FACTOR	0x000D	X	X	X	X	X
	Engine Temperature 1	ENGINE_TEMPERATURE_1	0x000E	X	X	X	X	X
	Engine Temperature 2	ENGINE_TEMPERATURE_2	0x000F	X	X	X	X	X
	Fuel Consumption	FUEL_CONSUMPTION	0x0010	X	X	X	X	X
	Fuel Economy	FUEL_ECONOMY	0x0011	X	X	X	X	X
	Engine Valve	ENGINE_VALVE	0x0012	X	X	X	X	X
	Direct Lamp Control Data 1	DIRECT_LAMP_CONTROL_DATA_1	0x0013	X	X	X	X	X
	Direct Lamp Control Data 2	DIRECT_LAMP_CONTROL_DATA_2	0x0014	X	X	X	X	X
	Electronic Transmission Controller 2	ELECTRONIC_TRANSMISSION_CONTROLLER_2	0x0015	X	X	X	X	X
	Active Diagnostic Trouble Codes	ACTIVE_DIAGNOSTIC_TROUBLE_CODES	0xF024		X	X	X	X

Component Type	Message Name	Data Type	Message Value	Version Applicability				
				V1.4	V1.6	V1.7	V1.8	V1.9
	Alternator Temperature	ALTERNATOR_TEMPERATURE	0x0025		X	X	X	X
	Battery Temperature	BATTERY_TEMPERATURE	0x0026		X	X	X	X
	Cab Message 1	CAB_MESSAGE_1	0x0027		X	X	X	X
	Combination Vehicle Weight	COMBINATION_VEHICLE_WEIGHT	0x0028		X	X	X	X
	Electronic Engine Controller 1	ELECTRONIC_ENGINE_CONTROLLER_1	0x0029		X	X	X	X
	Electronic Engine Controller 2	ELECTRONIC_ENGINE_CONTROLLER_2	0x002A		X	X	X	X
	Fan Drive 1	FAN_DRIVE_1	0x002B		X	X	X	X
	Tire Condition	TIRE_CONDITION	0xF02C		X	X	X	X
	Torque/ Speed Control 1	TORQUE_SPEED_CONTROL_1	0x002D		X	X	X	X
Power Distribution System	Power System Device	POWER_SYSTEM_DEVICE	0x001A	X	X	X	X	X
	Power System Device Threshold	POWER_SYSTEM_DEVICE_THRESHOLD	0x001B	X	X	X	X	X
	Platform Battery	PLATFORM_BATTERY	0x001C	X	X	X	X	X
	Platform Power Generation	PLATFORM_POWER_GENERATION	0x001D	X	X	X	X	X
	Platform Power	PLATFORM_POWER	0x001E	X	X	X	X	X
Camera Gimbal	Camera Orientation	CAMERA_ORIENTATION	0x002E	X	X	X	X	X
	Home Position	HOME_POSITION	0x0031			X	X	X

Component Type	Message Name	Data Type	Message Value	Version Applicability				
				V1.4	V1.6	V1.7	V1.8	V1.9
Data Logger	Data Logger Storage	DATA_LOGGER_STORAGE	0x002F		X	X	X	X
Live Training Transformation Adapter	LT2 Adapter	LT2_ADAPTER	0x0030			X	X	X
Experimental	N/A	N/A	0xF***	N/A	N/A	N/A	N/A	N/A

APPENDIX G: VIDEO ENCODER PIXEL FORMATS

This appendix provides a comprehensive list of common pixel formats. While not required, implementors of a video encoder should use one of the pixel formats from this list, when available.

Table 299. Video Encoder Pixel Formats

Pixel Format	Description
AV_PIX_FMT_YUV420P	planar YUV 4:2:0, 12bpp, (1 Cr & Cb sample per 2x2 Y samples)
AV_PIX_FMT_YUYV422	packed YUV 4:2:2, 16bpp, Y0 Cb Y1 Cr
AV_PIX_FMT_RGB24	packed RGB 8:8:8, 24bpp, RGBRGB…
AV_PIX_FMT_BGR24	packed RGB 8:8:8, 24bpp, BGRBGR…
AV_PIX_FMT_YUV422P	planar YUV 4:2:2, 16bpp, (1 Cr & Cb sample per 2x1 Y samples)
AV_PIX_FMT_YUV444P	planar YUV 4:4:4, 24bpp, (1 Cr & Cb sample per 1x1 Y samples)
AV_PIX_FMT_YUV410P	planar YUV 4:1:0, 9bpp, (1 Cr & Cb sample per 4x4 Y samples)
AV_PIX_FMT_YUV411P	planar YUV 4:1:1, 12bpp, (1 Cr & Cb sample per 4x1 Y samples)
AV_PIX_FMT_GRAY8	Y , 8bpp.
AV_PIX_FMT_MONOWHITE	Y , 1bpp, 0 is white, 1 is black, in each byte pixels are ordered from the msb to the lsb.
AV_PIX_FMT_MONOBLACK	Y , 1bpp, 0 is black, 1 is white, in each byte pixels are ordered from the msb to the lsb.
AV_PIX_FMT_PAL8	8 bit with AV_PIX_FMT_RGB32 palette
AV_PIX_FMT_YUVJ420P	planar YUV 4:2:0, 12bpp, full scale (JPEG)
AV_PIX_FMT_YUVJ422P	planar YUV 4:2:2, 16bpp, full scale (JPEG)
AV_PIX_FMT_YUVJ444P	planar YUV 4:4:4, 24bpp, full scale (JPEG)
AV_PIX_FMT_XVMC_MPEG2_MC	XVideo Motion Acceleration via common packet passing.
AV_PIX_FMT_XVMC_MPEG2_IDCT	
AV_PIX_FMT_UYVY422	packed YUV 4:2:2, 16bpp, Cb Y0 Cr Y1
AV_PIX_FMT_UYYVYY411	packed YUV 4:1:1, 12bpp, Cb Y0 Y1 Cr Y2 Y3
AV_PIX_FMT_BGR8	packed RGB 3:3:2, 8bpp, (msb)2B 3G 3R(lsb)
AV_PIX_FMT_BGR4	packed RGB 1:2:1 bitstream, 4bpp, (msb)1B 2G 1R(lsb), a byte contains two pixels, the first pixel in the byte is the one composed by the 4 msb bits
AV_PIX_FMT_BGR4_BYTE	packed RGB 1:2:1, 8bpp, (msb)1B 2G 1R(lsb)
AV_PIX_FMT_RGB8	packed RGB 3:3:2, 8bpp, (msb)2R 3G 3B(lsb)
AV_PIX_FMT_RGB4	packed RGB 1:2:1 bitstream, 4bpp, (msb)1R 2G 1B(lsb), a byte contains two pixels, the first pixel in the byte is the one composed by the 4 msb bits
AV_PIX_FMT_RGB4_BYTE	packed RGB 1:2:1, 8bpp, (msb)1R 2G 1B(lsb)

Pixel Format	Description
AV_PIX_FMT_NV12	planar YUV 4:2:0, 12bpp, 1 plane for Y and 1 plane for the UV components, which are interleaved (first byte U and the following byte V)
AV_PIX_FMT_NV21	planar YUV 4:2:0, 12bpp, 1 plane for Y and 1 plane for the UV components, which are interleaved (first byte Y and the following byte U)
AV_PIX_FMT_ARGB	packed ARGB 8:8:8:8, 32bpp, ARGBARGB…
AV_PIX_FMT_RGBA	packed RGBA 8:8:8:8, 32bpp, RGBARGBA…
AV_PIX_FMT_ABGR	packed ABGR 8:8:8:8, 32bpp, ABGRABGR…
AV_PIX_FMT_BGRA	packed BGRA 8:8:8:8, 32bpp, BGRABGRA…
AV_PIX_FMT_GRAY16BE	Y , 16bpp, big-endian.
AV_PIX_FMT_GRAY16LE	Y , 16bpp, little-endian.
AV_PIX_FMT_YUV440P	planar YUV 4:4:0 (1 Cr & Cb sample per 1x2 Y samples)
AV_PIX_FMT_YUVJ440P	planar YUV 4:4:0 full scale (JPEG)
AV_PIX_FMT_YUVA420P	planar YUV 4:2:0, 20bpp, (1 Cr & Cb sample per 2x2 Y & A samples)
AV_PIX_FMT_VDPAU_H264	H.264 HW decoding with VDPAU
AV_PIX_FMT_VDPAU_MPEG1	MPEG–1 HW decoding with VDPAU
AV_PIX_FMT_VDPAU_MPEG2	MPEG–2 HW decoding with VDPAU
AV_PIX_FMT_VDPAU_WMV3	WMV3 HW decoding with VDPAU
AV_PIX_FMT_VDPAU_VC1	VC–1 HW decoding with VDPAU
AV_PIX_FMT_RGB48BE	packed RGB 16:16:16, 48bpp, 16R, 16G, 16B, the 2-byte value for each R/G/B component is stored as big-endian
AV_PIX_FMT_RGB48LE	packed RGB 16:16:16, 48bpp, 16R, 16G, 16B, the 2-byte value for each R/G/B component is stored as little-endian
AV_PIX_FMT_RGB565BE	packed RGB 5:6:5, 16bpp, (msb) 5R 6G 5B(lsb), big-endian
AV_PIX_FMT_RGB565LE	packed RGB 5:6:5, 16bpp, (msb) 5R 6G 5B(lsb), little-endian
AV_PIX_FMT_RGB555BE	packed RGB 5:5:5, 16bpp, (msb)1X 5R 5G 5B(lsb), big-endian , X=unused/undefined
AV_PIX_FMT_RGB555LE	packed RGB 5:5:5, 16bpp, (msb)1X 5R 5G 5B(lsb), little-endian, X=unused/undefined
AV_PIX_FMT_BGR565BE	packed BGR 5:6:5, 16bpp, (msb) 5B 6G 5R(lsb), big-endian
AV_PIX_FMT_BGR565LE	packed BGR 5:6:5, 16bpp, (msb) 5B 6G 5R(lsb), little-endian
AV_PIX_FMT_BGR555BE	packed BGR 5:5:5, 16bpp, (msb)1X 5B 5G 5R(lsb), big-endian , X=unused/undefined
AV_PIX_FMT_BGR555LE	packed BGR 5:5:5, 16bpp, (msb)1X 5B 5G 5R(lsb), little-endian, X=unused/undefined
AV_PIX_FMT_VAAPI_MOCO	HW acceleration through VA API at motion compensation entry-point
AV_PIX_FMT_VAAPI_IDCT	HW acceleration through VA API at IDCT entry-point
AV_PIX_FMT_VAAPI_VLD	HW decoding through VA API

Pixel Format	Description
AV_PIX_FMT_YUV420P16LE	planar YUV 4:2:0, 24bpp, (1 Cr & Cb sample per 2x2 Y samples), little-endian
AV_PIX_FMT_YUV420P16BE	planar YUV 4:2:0, 24bpp, (1 Cr & Cb sample per 2x2 Y samples), big-endian
AV_PIX_FMT_YUV422P16LE	planar YUV 4:2:2, 32bpp, (1 Cr & Cb sample per 2x1 Y samples), little-endian
AV_PIX_FMT_YUV422P16BE	planar YUV 4:2:2, 32bpp, (1 Cr & Cb sample per 2x1 Y samples), big-endian
AV_PIX_FMT_YUV444P16LE	planar YUV 4:4:4, 48bpp, (1 Cr & Cb sample per 1x1 Y samples), little-endian
AV_PIX_FMT_YUV444P16BE	planar YUV 4:4:4, 48bpp, (1 Cr & Cb sample per 1x1 Y samples), big-endian
AV_PIX_FMT_VDPAU_MPEG4	MPEG4 HW decoding with VDPAU
AV_PIX_FMT_DXVA2_VLD	HW decoding through DXVA2
AV_PIX_FMT_RGB444LE	packed RGB 4:4:4, 16bpp, (msb)4X 4R 4G 4B(lsb), little-endian, X=unused/undefined
AV_PIX_FMT_RGB444BE	packed RGB 4:4:4, 16bpp, (msb)4X 4R 4G 4B(lsb), big-endian, X=unused/undefined
AV_PIX_FMT_BGR444LE	packed BGR 4:4:4, 16bpp, (msb)4X 4B 4G 4R(lsb), little-endian, X=unused/undefined
AV_PIX_FMT_BGR444BE	packed BGR 4:4:4, 16bpp, (msb)4X 4B 4G 4R(lsb), big-endian, X=unused/undefined
AV_PIX_FMT_YA8	8bit gray, 8bit alpha
AV_PIX_FMT_Y400A	alias for AV_PIX_FMT_YA8
AV_PIX_FMT_GRAY8A	alias for AV_PIX_FMT_YA8
AV_PIX_FMT_BGR48BE	packed RGB 16:16:16, 48bpp, 16B, 16G, 16R, the 2-byte value for each R/G/B component is stored as big-endian
AV_PIX_FMT_BGR48LE	packed RGB 16:16:16, 48bpp, 16B, 16G, 16R, the 2-byte value for each R/G/B component is stored as little-endian
AV_PIX_FMT_YUV420P9BE	planar YUV 4:2:0, 13.5bpp, (1 Cr & Cb sample per 2x2 Y samples), big-endian
AV_PIX_FMT_YUV420P9LE	planar YUV 4:2:0, 13.5bpp, (1 Cr & Cb sample per 2x2 Y samples), little-endian
AV_PIX_FMT_YUV420P10BE	planar YUV 4:2:0, 15bpp, (1 Cr & Cb sample per 2x2 Y samples), big-endian
AV_PIX_FMT_YUV420P10LE	planar YUV 4:2:0, 15bpp, (1 Cr & Cb sample per 2x2 Y samples), little-endian
AV_PIX_FMT_YUV422P10BE	planar YUV 4:2:2, 20bpp, (1 Cr & Cb sample per 2x1 Y samples), big-endian
AV_PIX_FMT_YUV422P10LE	planar YUV 4:2:2, 20bpp, (1 Cr & Cb sample per 2x1 Y samples), little-endian
AV_PIX_FMT_YUV444P9BE	planar YUV 4:4:4, 27bpp, (1 Cr & Cb sample per 1x1 Y samples), big-endian
AV_PIX_FMT_YUV444P9LE	planar YUV 4:4:4, 27bpp, (1 Cr & Cb sample per 1x1 Y samples), little-endian
AV_PIX_FMT_YUV444P10BE	planar YUV 4:4:4, 30bpp, (1 Cr & Cb sample per 1x1 Y samples), big-endian
AV_PIX_FMT_YUV444P10LE	planar YUV 4:4:4, 30bpp, (1 Cr & Cb sample per 1x1 Y samples), little-endian
AV_PIX_FMT_YUV422P9BE	planar YUV 4:2:2, 18bpp, (1 Cr & Cb sample per 2x1 Y samples), big-endian
AV_PIX_FMT_YUV422P9LE	planar YUV 4:2:2, 18bpp, (1 Cr & Cb sample per 2x1 Y samples), little-endian
AV_PIX_FMT_VDA_VLD	hardware decoding through VDA

Pixel Format	Description
AV_PIX_FMT_GBRP	planar GBR 4:4:4 24bpp
AV_PIX_FMT_GBRP9BE	planar GBR 4:4:4 27bpp, big-endian
AV_PIX_FMT_GBRP9LE	planar GBR 4:4:4 27bpp, little-endian
AV_PIX_FMT_GBRP10BE	planar GBR 4:4:4 30bpp, big-endian
AV_PIX_FMT_GBRP10LE	planar GBR 4:4:4 30bpp, little-endian
AV_PIX_FMT_GBRP16BE	planar GBR 4:4:4 48bpp, big-endian
AV_PIX_FMT_GBRP16LE	planar GBR 4:4:4 48bpp, little-endian
AV_PIX_FMT_YUVA420P9BE	planar YUV 4:2:0 22.5bpp, (1 Cr & Cb sample per 2x2 Y & A samples), big-endian
AV_PIX_FMT_YUVA420P9LE	planar YUV 4:2:0 22.5bpp, (1 Cr & Cb sample per 2x2 Y & A samples), little-endian
AV_PIX_FMT_YUVA422P9BE	planar YUV 4:2:2 27bpp, (1 Cr & Cb sample per 2x1 Y & A samples), big-endian
AV_PIX_FMT_YUVA422P9LE	planar YUV 4:2:2 27bpp, (1 Cr & Cb sample per 2x1 Y & A samples), little-endian
AV_PIX_FMT_YUVA444P9BE	planar YUV 4:4:4 36bpp, (1 Cr & Cb sample per 1x1 Y & A samples), big-endian
AV_PIX_FMT_YUVA444P9LE	planar YUV 4:4:4 36bpp, (1 Cr & Cb sample per 1x1 Y & A samples), little-endian
AV_PIX_FMT_YUVA420P10BE	planar YUV 4:2:0 25bpp, (1 Cr & Cb sample per 2x2 Y & A samples, big-endian)
AV_PIX_FMT_YUVA420P10LE	planar YUV 4:2:0 25bpp, (1 Cr & Cb sample per 2x2 Y & A samples, little-endian)
AV_PIX_FMT_YUVA422P10BE	planar YUV 4:2:2 30bpp, (1 Cr & Cb sample per 2x1 Y & A samples, big-endian)
AV_PIX_FMT_YUVA422P10LE	planar YUV 4:2:2 30bpp, (1 Cr & Cb sample per 2x1 Y & A samples, little-endian)
AV_PIX_FMT_YUVA444P10BE	planar YUV 4:4:4 40bpp, (1 Cr & Cb sample per 1x1 Y & A samples, big-endian)
AV_PIX_FMT_YUVA444P10LE	planar YUV 4:4:4 40bpp, (1 Cr & Cb sample per 1x1 Y & A samples, little-endian)
AV_PIX_FMT_YUVA420P16BE	planar YUV 4:2:0 40bpp, (1 Cr & Cb sample per 2x2 Y & A samples, big-endian)
AV_PIX_FMT_YUVA420P16LE	planar YUV 4:2:0 40bpp, (1 Cr & Cb sample per 2x2 Y & A samples, little-endian)
AV_PIX_FMT_YUVA422P16BE	planar YUV 4:2:2 48bpp, (1 Cr & Cb sample per 2x1 Y & A samples, big-endian)
AV_PIX_FMT_YUVA422P16LE	planar YUV 4:2:2 48bpp, (1 Cr & Cb sample per 2x1 Y & A samples, little-endian)
AV_PIX_FMT_YUVA444P16BE	planar YUV 4:4:4 64bpp, (1 Cr & Cb sample per 1x1 Y & A samples, big-endian)
AV_PIX_FMT_YUVA444P16LE	planar YUV 4:4:4 64bpp, (1 Cr & Cb sample per 1x1 Y & A samples, little-endian)
AV_PIX_FMT_VDPAU	HW acceleration through VDPAU
AV_PIX_FMT_XYZ12LE	packed XYZ 4:4:4, 36 bpp, (msb) 12X, 12Y, 12Z (lsb), the 2-byte value for each X/Y/Z is stored as little-endian, the 4 lower bits are set to 0
AV_PIX_FMT_XYZ12BE	packed XYZ 4:4:4, 36 bpp, (msb) 12X, 12Y, 12Z (lsb), the 2-byte value for each X/Y/Z is stored as big-endian, the 4 lower bits are set to 0
AV_PIX_FMT_NV16	interleaved chroma YUV 4:2:2, 16bpp, (1 Cr & Cb sample per 2x1 Y samples)

Pixel Format	Description
AV_PIX_FMT_NV20LE	interleaved chroma YUV 4:2:2, 20bpp, (1 Cr & Cb sample per 2x1 Y samples), little-endian
AV_PIX_FMT_NV20BE	interleaved chroma YUV 4:2:2, 20bpp, (1 Cr & Cb sample per 2x1 Y samples), big-endian
AV_PIX_FMT_YVYU422	packed YUV 4:2:2, 16bpp, Y0 Cr Y1 Cb
AV_PIX_FMT_VDA	HW acceleration through VDA
AV_PIX_FMT_YA16BE	16bit gray, 16bit alpha (big-endian)
AV_PIX_FMT_YA16LE	16bit gray, 16bit alpha (little-endian)
AV_PIX_FMT_QSV	HW acceleration through QSV
AV_PIX_FMT_MMAL	HW acceleration though MMAL
AV_PIX_FMT_D3D11VA_VLD	HW decoding through Direct3D11
AV_PIX_FMT_RGBA64BE	packed RGBA 16:16:16:16, 64bpp, 16R, 16G, 16B, 16A, the 2-byte value for each R/G/B/A component is stored as big-endian
AV_PIX_FMT_RGBA64LE	packed RGBA 16:16:16:16, 64bpp, 16R, 16G, 16B, 16A, the 2-byte value for each R/G/B/A component is stored as little-endian
AV_PIX_FMT_BGRA64BE	packed RGBA 16:16:16:16, 64bpp, 16B, 16G, 16R, 16A, the 2-byte value for each R/G/B/A component is stored as big-endian
AV_PIX_FMT_BGRA64LE	packed RGBA 16:16:16:16, 64bpp, 16B, 16G, 16R, 16A, the 2-byte value for each R/G/B/A component is stored as little-endian
AV_PIX_FMT_0RGB	packed RGB 8:8:8, 32bpp, XRGBXRGB… X=unused/undefined
AV_PIX_FMT_RGB0	packed RGB 8:8:8, 32bpp, RGBXRGBX… X=unused/undefined
AV_PIX_FMT_0BGR	packed BGR 8:8:8, 32bpp, XBGRXBGR… X=unused/undefined
AV_PIX_FMT_BGR0	packed BGR 8:8:8, 32bpp, BGRXBGRX… X=unused/undefined
AV_PIX_FMT_YUVA444P	planar YUV 4:4:4 32bpp, (1 Cr & Cb sample per 1x1 Y & A samples)
AV_PIX_FMT_YUVA422P	planar YUV 4:2:2 24bpp, (1 Cr & Cb sample per 2x1 Y & A samples)
AV_PIX_FMT_YUV420P12BE	planar YUV 4:2:0,18bpp, (1 Cr & Cb sample per 2x2 Y samples), big-endian
AV_PIX_FMT_YUV420P12LE	planar YUV 4:2:0,18bpp, (1 Cr & Cb sample per 2x2 Y samples), little-endian
AV_PIX_FMT_YUV420P14BE	planar YUV 4:2:0,21bpp, (1 Cr & Cb sample per 2x2 Y samples), big-endian
AV_PIX_FMT_YUV420P14LE	planar YUV 4:2:0,21bpp, (1 Cr & Cb sample per 2x2 Y samples), little-endian
AV_PIX_FMT_YUV422P12BE	planar YUV 4:2:2,24bpp, (1 Cr & Cb sample per 2x1 Y samples), big-endian
AV_PIX_FMT_YUV422P12LE	planar YUV 4:2:2,24bpp, (1 Cr & Cb sample per 2x1 Y samples), little-endian
AV_PIX_FMT_YUV422P14BE	planar YUV 4:2:2,28bpp, (1 Cr & Cb sample per 2x1 Y samples), big-endian
AV_PIX_FMT_YUV422P14LE	planar YUV 4:2:2,28bpp, (1 Cr & Cb sample per 2x1 Y samples), little-endian
AV_PIX_FMT_YUV444P12BE	planar YUV 4:4:4,36bpp, (1 Cr & Cb sample per 1x1 Y samples), big-endian

Pixel Format	Description
AV_PIX_FMT_YUV444P12LE	planar YUV 4:4:4,36bpp, (1 Cr & Cb sample per 1x1 Y samples), little-endian
AV_PIX_FMT_YUV444P14BE	planar YUV 4:4:4,42bpp, (1 Cr & Cb sample per 1x1 Y samples), big-endian
AV_PIX_FMT_YUV444P14LE	planar YUV 4:4:4,42bpp, (1 Cr & Cb sample per 1x1 Y samples), little-endian
AV_PIX_FMT_GBRP12BE	planar GBR 4:4:4 36bpp, big-endian
AV_PIX_FMT_GBRP12LE	planar GBR 4:4:4 36bpp, little-endian
AV_PIX_FMT_GBRP14BE	planar GBR 4:4:4 42bpp, big-endian
AV_PIX_FMT_GBRP14LE	planar GBR 4:4:4 42bpp, little-endian
AV_PIX_FMT_GBRAP	planar GBRA 4:4:4:4 32bpp
AV_PIX_FMT_GBRAP16BE	planar GBRA 4:4:4:4 64bpp, big-endian
AV_PIX_FMT_GBRAP16LE	planar GBRA 4:4:4:4 64bpp, little-endian
AV_PIX_FMT_YUVJ411P	planar YUV 4:1:1, 12bpp, (1 Cr & Cb sample per 4x1 Y samples) full scale (JPEG)
AV_PIX_FMT_BAYER_BGGR8	bayer, BGBG..(odd line), GRGR..(even line), 8-bit samples
AV_PIX_FMT_BAYER_RGGB8	bayer, RGRG..(odd line), GBGB..(even line), 8-bit samples
AV_PIX_FMT_BAYER_GBRG8	bayer, GBGB..(odd line), RGRG..(even line), 8-bit samples
AV_PIX_FMT_BAYER_GRBG8	bayer, GRGR..(odd line), BGBG..(even line), 8-bit samples
AV_PIX_FMT_BAYER_BGGR16LE	bayer, BGBG..(odd line), GRGR..(even line), 16-bit samples, little-endian
AV_PIX_FMT_BAYER_BGGR16BE	bayer, BGBG..(odd line), GRGR..(even line), 16-bit samples, big-endian
AV_PIX_FMT_BAYER_RGGB16LE	bayer, RGRG..(odd line), GBGB..(even line), 16-bit samples, little-endian
AV_PIX_FMT_BAYER_RGGB16BE	bayer, RGRG..(odd line), GBGB..(even line), 16-bit samples, big-endian
AV_PIX_FMT_BAYER_GBRG16LE	bayer, GBGB..(odd line), RGRG..(even line), 16-bit samples, little-endian
AV_PIX_FMT_BAYER_GBRG16BE	bayer, GBGB..(odd line), RGRG..(even line), 16-bit samples, big-endian
AV_PIX_FMT_BAYER_GRBG16LE	bayer, GRGR..(odd line), BGBG..(even line), 16-bit samples, little-endian
AV_PIX_FMT_BAYER_GRBG16BE	bayer, GRGR..(odd line), BGBG..(even line), 16-bit samples, big-endian
AV_PIX_FMT_YUV440P10LE	planar YUV 4:4:0,20bpp, (1 Cr & Cb sample per 1x2 Y samples), little-endian
AV_PIX_FMT_YUV440P10BE	planar YUV 4:4:0,20bpp, (1 Cr & Cb sample per 1x2 Y samples), big-endian
AV_PIX_FMT_YUV440P12LE	planar YUV 4:4:0,24bpp, (1 Cr & Cb sample per 1x2 Y samples), little-endian
AV_PIX_FMT_YUV440P12BE	planar YUV 4:4:0,24bpp, (1 Cr & Cb sample per 1x2 Y samples), big-endian
PIX_FMT_YUV420P	planar YUV 4:2:0, 12bpp, (1 Cr & Cb sample per 2x2 Y samples)
PIX_FMT_YUYV422	packed YUV 4:2:2, 16bpp, Y0 Cb Y1 Cr
PIX_FMT_RGB24	packed RGB 8:8:8, 24bpp, RGBRGB…
PIX_FMT_BGR24	packed RGB 8:8:8, 24bpp, BGRBGR…

Pixel Format	Description
PIX_FMT_YUV422P	planar YUV 4:2:2, 16bpp, (1 Cr & Cb sample per 2x1 Y samples)
PIX_FMT_YUV444P	planar YUV 4:4:4, 24bpp, (1 Cr & Cb sample per 1x1 Y samples)
PIX_FMT_YUV410P	planar YUV 4:1:0, 9bpp, (1 Cr & Cb sample per 4x4 Y samples)
PIX_FMT_YUV411P	planar YUV 4:1:1, 12bpp, (1 Cr & Cb sample per 4x1 Y samples)
PIX_FMT_GRAY8	Y , 8bpp.
PIX_FMT_MONOWHITE	Y , 1bpp, 0 is white, 1 is black, in each byte pixels are ordered from the msb to the lsb.
PIX_FMT_MONOBLACK	Y , 1bpp, 0 is black, 1 is white, in each byte pixels are ordered from the msb to the lsb.
PIX_FMT_PAL8	8 bit with PIX_FMT_RGB32 palette
PIX_FMT_YUVJ420P	planar YUV 4:2:0, 12bpp, full scale (JPEG)
PIX_FMT_YUVJ422P	planar YUV 4:2:2, 16bpp, full scale (JPEG)
PIX_FMT_YUVJ444P	planar YUV 4:4:4, 24bpp, full scale (JPEG)
PIX_FMT_XVMC_MPEG2_MC	XVideo Motion Acceleration via common packet passing.
PIX_FMT_XVMC_MPEG2_IDCT	
PIX_FMT_UYVY422	packed YUV 4:2:2, 16bpp, Cb Y0 Cr Y1
PIX_FMT_UYYVYY411	packed YUV 4:1:1, 12bpp, Cb Y0 Y1 Cr Y2 Y3
PIX_FMT_BGR8	packed RGB 3:3:2, 8bpp, (msb)2B 3G 3R(lsb)
PIX_FMT_BGR4	packed RGB 1:2:1 bitstream, 4bpp, (msb)1B 2G 1R(lsb), a byte contains two pixels, the first pixel in the byte is the one composed by the 4 msb bits
PIX_FMT_BGR4_BYTE	packed RGB 1:2:1, 8bpp, (msb)1B 2G 1R(lsb)
PIX_FMT_RGB8	packed RGB 3:3:2, 8bpp, (msb)2R 3G 3B(lsb)
PIX_FMT_RGB4	packed RGB 1:2:1 bitstream, 4bpp, (msb)1R 2G 1B(lsb), a byte contains two pixels, the first pixel in the byte is the one composed by the 4 msb bits
PIX_FMT_RGB4_BYTE	packed RGB 1:2:1, 8bpp, (msb)1R 2G 1B(lsb)
PIX_FMT_NV12	planar YUV 4:2:0, 12bpp, 1 plane for Y and 1 plane for the UV components, which are interleaved (first byte U and the following byte V)
PIX_FMT_NV21	planar YUV 4:2:0, 12bpp, 1 plane for Y and 1 plane for the UV components, which are interleaved (first byte V and the following byte U)
PIX_FMT_ARGB	packed ARGB 8:8:8:8, 32bpp, ARGBARGB…
PIX_FMT_RGBA	packed RGBA 8:8:8:8, 32bpp, RGBARGBA…
PIX_FMT_ABGR	packed ABGR 8:8:8:8, 32bpp, ABGRABGR…
PIX_FMT_BGRA	packed BGRA 8:8:8:8, 32bpp, BGRABGRA…
PIX_FMT_GRAY16BE	Y , 16bpp, big-endian.
PIX_FMT_GRAY16LE	Y , 16bpp, little-endian.

Pixel Format	Description
PIX_FMT_YUV440P	planar YUV 4:4:0 (1 Cr & Cb sample per 1x2 Y samples)
PIX_FMT_YUVJ440P	planar YUV 4:4:0 full scale (JPEG)
PIX_FMT_YUVA420P	planar YUV 4:2:0, 20bpp, (1 Cr & Cb sample per 2x2 Y & A samples)
PIX_FMT_VDPAU_H264	H.264 HW decoding with VDPAU
PIX_FMT_VDPAU_MPEG1	MPEG–1 HW decoding with VDPAU
PIX_FMT_VDPAU_MPEG2	MPEG–2 HW decoding with VDPAU
PIX_FMT_VDPAU_WMV3	WMV3 HW decoding with VDPAU
PIX_FMT_VDPAU_VC1	VC–1 HW decoding with VDPAU
PIX_FMT_RGB48BE	packed RGB 16:16:16, 48bpp, 16R, 16G, 16B, the 2-byte value for each R/G/B component is stored as big-endian
PIX_FMT_RGB48LE	packed RGB 16:16:16, 48bpp, 16R, 16G, 16B, the 2-byte value for each R/G/B component is stored as little-endian
PIX_FMT_RGB565BE	packed RGB 5:6:5, 16bpp, (msb) 5R 6G 5B(lsb), big-endian
PIX_FMT_RGB565LE	packed RGB 5:6:5, 16bpp, (msb) 5R 6G 5B(lsb), little-endian
PIX_FMT_RGB555BE	packed RGB 5:5:5, 16bpp, (msb)1A 5R 5G 5B(lsb), big-endian, most significant bit to 0
PIX_FMT_RGB555LE	packed RGB 5:5:5, 16bpp, (msb)1A 5R 5G 5B(lsb), little-endian, most significant bit to 0
PIX_FMT_BGR565BE	packed BGR 5:6:5, 16bpp, (msb) 5B 6G 5R(lsb), big-endian
PIX_FMT_BGR565LE	packed BGR 5:6:5, 16bpp, (msb) 5B 6G 5R(lsb), little-endian
PIX_FMT_BGR555BE	packed BGR 5:5:5, 16bpp, (msb)1A 5B 5G 5R(lsb), big-endian, most significant bit to 1
PIX_FMT_BGR555LE	packed BGR 5:5:5, 16bpp, (msb)1A 5B 5G 5R(lsb), little-endian, most significant bit to 1
PIX_FMT_VAAPI_MOCO	HW acceleration through VA API at motion compensation entry-point
PIX_FMT_VAAPI_IDCT	HW acceleration through VA API at IDCT entry-point
PIX_FMT_VAAPI_VLD	HW decoding through VA API
PIX_FMT_YUV420P16LE	planar YUV 4:2:0, 24bpp, (1 Cr & Cb sample per 2x2 Y samples), little-endian
PIX_FMT_YUV420P16BE	planar YUV 4:2:0, 24bpp, (1 Cr & Cb sample per 2x2 Y samples), big-endian
PIX_FMT_YUV422P16LE	planar YUV 4:2:2, 32bpp, (1 Cr & Cb sample per 2x1 Y samples), little-endian
PIX_FMT_YUV422P16BE	planar YUV 4:2:2, 32bpp, (1 Cr & Cb sample per 2x1 Y samples), big-endian
PIX_FMT_YUV444P16LE	planar YUV 4:4:4, 48bpp, (1 Cr & Cb sample per 1x1 Y samples), little-endian
PIX_FMT_YUV444P16BE	planar YUV 4:4:4, 48bpp, (1 Cr & Cb sample per 1x1 Y samples), big-endian
PIX_FMT_VDPAU_MPEG4	MPEG4 HW decoding with VDPAU
PIX_FMT_DXVA2_VLD	HW decoding through DXVA2

Pixel Format	Description
PIX_FMT_RGB444LE	packed RGB 4:4:4, 16bpp, (msb)4A 4R 4G 4B(lsb), little-endian, most significant bits to 0
PIX_FMT_RGB444BE	packed RGB 4:4:4, 16bpp, (msb)4A 4R 4G 4B(lsb), big-endian, most significant bits to 0
PIX_FMT_BGR444LE	packed BGR 4:4:4, 16bpp, (msb)4A 4B 4G 4R(lsb), little-endian, most significant bits to 1
PIX_FMT_BGR444BE	packed BGR 4:4:4, 16bpp, (msb)4A 4B 4G 4R(lsb), big-endian, most significant bits to 1
PIX_FMT_GRAY8A	8bit gray, 8bit alpha
PIX_FMT_BGR48BE	packed RGB 16:16:16, 48bpp, 16B, 16G, 16R, the 2-byte value for each R/G/B component is stored as big-endian
PIX_FMT_BGR48LE	packed RGB 16:16:16, 48bpp, 16B, 16G, 16R, the 2-byte value for each R/G/B component is stored as little-endian
PIX_FMT_YUV420P9BE	planar YUV 4:2:0, 13.5bpp, (1 Cr & Cb sample per 2x2 Y samples), big-endian
PIX_FMT_YUV420P9LE	planar YUV 4:2:0, 13.5bpp, (1 Cr & Cb sample per 2x2 Y samples), little-endian
PIX_FMT_YUV420P10BE	planar YUV 4:2:0, 15bpp, (1 Cr & Cb sample per 2x2 Y samples), big-endian
PIX_FMT_YUV420P10LE	planar YUV 4:2:0, 15bpp, (1 Cr & Cb sample per 2x2 Y samples), little-endian
PIX_FMT_YUV422P10BE	planar YUV 4:2:2, 20bpp, (1 Cr & Cb sample per 2x1 Y samples), big-endian
PIX_FMT_YUV422P10LE	planar YUV 4:2:2, 20bpp, (1 Cr & Cb sample per 2x1 Y samples), little-endian
PIX_FMT_YUV444P9BE	planar YUV 4:4:4, 27bpp, (1 Cr & Cb sample per 1x1 Y samples), big-endian
PIX_FMT_YUV444P9LE	planar YUV 4:4:4, 27bpp, (1 Cr & Cb sample per 1x1 Y samples), little-endian
PIX_FMT_YUV444P10BE	planar YUV 4:4:4, 30bpp, (1 Cr & Cb sample per 1x1 Y samples), big-endian
PIX_FMT_YUV444P10LE	planar YUV 4:4:4, 30bpp, (1 Cr & Cb sample per 1x1 Y samples), little-endian
PIX_FMT_YUV422P9BE	planar YUV 4:2:2, 18bpp, (1 Cr & Cb sample per 2x1 Y samples), big-endian
PIX_FMT_YUV422P9LE	planar YUV 4:2:2, 18bpp, (1 Cr & Cb sample per 2x1 Y samples), little-endian
PIX_FMT_VDA_VLD	hardware decoding through VDA
PIX_FMT_GBRP	planar GBR 4:4:4 24bpp
PIX_FMT_GBRP9BE	planar GBR 4:4:4 27bpp, big endian
PIX_FMT_GBRP9LE	planar GBR 4:4:4 27bpp, little endian
PIX_FMT_GBRP10BE	planar GBR 4:4:4 30bpp, big endian
PIX_FMT_GBRP10LE	planar GBR 4:4:4 30bpp, little endian
PIX_FMT_GBRP16BE	planar GBR 4:4:4 48bpp, big endian
PIX_FMT_GBRP16LE	planar GBR 4:4:4 48bpp, little endian

Pixel Format	Description
PIX_FMT_RGBA64BE	packed RGBA 16:16:16:16, 64bpp, 16R, 16G, 16B, 16A, the 2-byte value for each R/G/B/A component is stored as big-endian
PIX_FMT_RGBA64LE	packed RGBA 16:16:16:16, 64bpp, 16R, 16G, 16B, 16A, the 2-byte value for each R/G/B/A component is stored as little-endian
PIX_FMT_BGRA64BE	packed RGBA 16:16:16:16, 64bpp, 16B, 16G, 16R, 16A, the 2-byte value for each R/G/B/A component is stored as big-endian
PIX_FMT_BGRA64LE	packed RGBA 16:16:16:16, 64bpp, 16B, 16G, 16R, 16A, the 2-byte value for each R/G/B/A component is stored as little-endian
PIX_FMT_0RGB	packed RGB 8:8:8, 32bpp, 0RGB0RGB…
PIX_FMT_RGB0	packed RGB 8:8:8, 32bpp, RGB0RGB0…
PIX_FMT_0BGR	packed BGR 8:8:8, 32bpp, 0BGR0BGR…
PIX_FMT_BGR0	packed BGR 8:8:8, 32bpp, BGR0BGR0…
PIX_FMT_YUVA444P	planar YUV 4:4:4 32bpp, (1 Cr & Cb sample per 1x1 Y & A samples)
PIX_FMT_YUVA422P	planar YUV 4:2:2 24bpp, (1 Cr & Cb sample per 2x1 Y & A samples)
PIX_FMT_YUV420P12BE	planar YUV 4:2:0,18bpp, (1 Cr & Cb sample per 2x2 Y samples), big-endian
PIX_FMT_YUV420P12LE	planar YUV 4:2:0,18bpp, (1 Cr & Cb sample per 2x2 Y samples), little-endian
PIX_FMT_YUV420P14BE	planar YUV 4:2:0,21bpp, (1 Cr & Cb sample per 2x2 Y samples), big-endian
PIX_FMT_YUV420P14LE	planar YUV 4:2:0,21bpp, (1 Cr & Cb sample per 2x2 Y samples), little-endian
PIX_FMT_YUV422P12BE	planar YUV 4:2:2,24bpp, (1 Cr & Cb sample per 2x1 Y samples), big-endian
PIX_FMT_YUV422P12LE	planar YUV 4:2:2,24bpp, (1 Cr & Cb sample per 2x1 Y samples), little-endian
PIX_FMT_YUV422P14BE	planar YUV 4:2:2,28bpp, (1 Cr & Cb sample per 2x1 Y samples), big-endian
PIX_FMT_YUV422P14LE	planar YUV 4:2:2,28bpp, (1 Cr & Cb sample per 2x1 Y samples), little-endian
PIX_FMT_YUV444P12BE	planar YUV 4:4:4,36bpp, (1 Cr & Cb sample per 1x1 Y samples), big-endian
PIX_FMT_YUV444P12LE	planar YUV 4:4:4,36bpp, (1 Cr & Cb sample per 1x1 Y samples), little-endian
PIX_FMT_YUV444P14BE	planar YUV 4:4:4,42bpp, (1 Cr & Cb sample per 1x1 Y samples), big-endian
PIX_FMT_YUV444P14LE	planar YUV 4:4:4,42bpp, (1 Cr & Cb sample per 1x1 Y samples), little-endian
PIX_FMT_GBRP12BE	planar GBR 4:4:4 36bpp, big endian
PIX_FMT_GBRP12LE	planar GBR 4:4:4 36bpp, little endian
PIX_FMT_GBRP14BE	planar GBR 4:4:4 42bpp, big endian
PIX_FMT_GBRP14LE	planar GBR 4:4:4 42bpp, little endian

1. The meaning of the TXT Field Values are described in Table 293. ←
2. See the website "DNS SRV (RFC 2782) Service Types" for a complete list of DNS-TXT keys for SIP. ←
3. Although no DNS-TXT keys are defined for RTSP at "DNS SRV (RFC 2782) Service Types", support for the path variable is common where it is implemented. ←
4. Hostname is interchangeable with IP address for the situations in which hostnames are not being used. ←