REST API
SEL-5056 v2.2.0
DRAFT ONLY

Schweitzer Engineering Laboratories, Inc. [1]

Last Updated October 15, 2020

[1] For assistance or feedback contact robert_meine@selinc.com

Contents

1 Introduction **1**

2 Conventions **3**

2.1 URIs . . . 3

2.2 Displaying HTTP Messages . . . 3

3 Getting Started **5**

I Overview **7**

4 API Onboarding and REST Interface Access **9**

4.1 API Onboarding . . . 9

4.1.1 API Discovery . . . 9

4.1.2 API Access Prerequisites . . . 9

4.1.3 API Applications . . . 9

4.2 Accessing the REST Interface . . . 10

5 Tutorial **11**

5.1 Commissioning the SEL-5056 . . . 12

5.2 Using the OData Schema, Part 1 . . . 13

5.3 Logging in with a username and password . . . 18

5.4 Refreshing the token . . . 19

5.5 Validating a token . . . 20

5.6 Logging Out . . . 21

5.7 Creating a Permission Level 3 user . . . 22

5.8 Finding the Permission Level 3 user . . . 24

5.9 Creating Configuration Nodes . . . 26

5.10 Adopting Switches . . . 32

5.11 Adopting Links . . . 35

5.12 Adopting the Hosts, including failover mode . . . 37

5.13 Creating Logical Conenctions . . . 42

5.14 Creating CSTs . . . 46

5.15 Adding an offline host . . . 48

5.16 Managing Flow Entries . . . 50

5.17 Managing Group Entries . . . 54

5.18 Managing Meter Entries . . . 58

6 Data Modal **61**

6.1 OpenFlow Data Model 61
6.1.1 Simple View of an OpenFlow Switch 62
6.2 Logical Programming Data Model 62
6.3 SEL-2740S Management Data Model 62
6.4 SEL-5056 Data Model 62
6.5 Topology Model 63
6.6 Defined versus variable settings 63
6.6.1 Attributes 64
6.6.2 Tags 64
6.6.3 Configurations 64

II REST Overview 65

7 HTTP 67
7.1 HTTPS 67
7.1.1 Headers 67
7.1.2 Response Codes 67
7.1.3 Common Format of HTTP Methods 69

8 OData 71
8.1 OData Version 71
8.2 Service Roots 71
8.2.1 Common Properties 71
8.2.2 Property Attributes 72
8.2.3 Filtering 72

9 OAuth Authentication 73
9.1 Including tokens in a request 73
9.2 OAuth and Connect ID Parameters 74
9.2.1 Grant Types 74
9.2.2 Client ID and Client Secret 74
9.2.3 Scope 75
9.2.4 Roles 75
9.3 Endpoints 75
9.3.1 Obtaining a token 75
9.3.2 Resource Owner Password Credentials 76
9.3.3 Client Credentials 77
9.3.4 Inspecting a token 78
9.3.5 Extending the expiration time of a Token 79
9.4 Refreshing Token 80
9.5 Managing Permissions 80
9.6 Troubleshooting 80
9.6.1 Registered Applications 80

III OData Endpoints 83

10 User Management 87
10.1 User Accounts 87
10.1.1 Read users 87
10.1.2 Create users 89
10.1.3 Update users (PUT) 91
10.1.4 Update users (PATCH) 91
10.1.5 Change the current user's password 93
10.1.6 Delete users 94
10.2 User Roles 94
10.2.1 Read user roles 94
10.3 Authentication Services 95
10.3.1 Read created authentication services 95
10.3.2 Create authentication services 96
10.3.3 Update created authentication services (PUT) 98
10.3.4 Update created authentication services (PATCH) 98
10.3.5 Delete created authentication services 99
10.3.6 Test an authentication service 100
10.4 Authentication Service Groups 101
10.4.1 Read authentication service groups 101
10.4.2 Create authentication service groups 102
10.4.3 Update authentication service groups (PUT) 102
10.4.4 Update authentication service groups (PATCH) 103
10.4.5 Delete authentication service groups 104
10.5 Preferences 104
10.5.1 Get the value of a user preference 105
10.5.2 Read user preferences 105
10.5.3 Add or modify a user preference 106

11 Topology Management 109
11.1 Nodes 110
11.1.1 Read configured nodes 112
11.1.2 Read operational nodes 113
11.1.3 Create configured nodes 114
11.1.4 Update configured nodes (PUT) 116
11.1.5 Update configured nodes (PATCH) 117
11.1.6 Merge two nodes by combining the ports of one into the other 118
11.1.7 Replace the configuration object for an adopted node 119
11.1.8 Delete configured nodes 119
11.1.9 Remove a node entry 120
11.1.10 Adopt a node with default settings 121
11.1.11 Adopt a node with a preconfigured node object 121
11.1.12 Unadopt a node 122
11.1.13 Get nodes by IP address 122
11.1.14 Get nodes by an attribute 123
11.1.15 Run host discovery for a particular IP address 123

11.1.16 Prevent the controller from sending arps to the node . . . 124
11.1.17 Allow the controller to send arps to the node . . . 125
11.2 OpenFlow Switches . . . 125
11.2.1 Added an offline host to a port . . . 125
11.3 SEL-2740S Switches . . . 126
11.3.1 Get nodes that are unsynchronized . . . 127
11.3.2 Read switches to synchronize . . . 127
11.3.3 Factory default reset a SEL-2740S . . . 134
11.3.4 Firmware upgrade a SEL-2740S . . . 134
11.3.5 Reboot a SEL-2740S . . . 135
11.3.6 Replan the IB management logical connection to a switch . . . 136
11.3.7 Read flow table diagnostics . . . 137
11.4 Traditional Switches . . . 138
11.4.1 Add a traditional switch . . . 138
11.5 Ports . . . 138
11.5.1 Read configured ports . . . 139
11.5.2 Read detected ports . . . 140
11.5.3 Create configured ports . . . 141
11.5.4 Update configured ports (PUT) . . . 142
11.5.5 Update configured ports (PATCH) . . . 143
11.5.6 Replace the configuration object for an adopted port . . . 144
11.5.7 Delete configured ports . . . 144
11.5.8 Get list of nodes that are unsynchronized . . . 145
11.5.9 Adopt a port with a preconfigured port object . . . 145
11.5.10 Unadopt a port . . . 146
11.5.11 Detect the other port of an SEL relay device . . . 146
11.5.12 Add SEL relay failover mode to a port . . . 147
11.5.13 Same as MarkPortAsSelRelayFailover and AddOperationalLinkFromHostToSwitch together . . . 148
11.5.14 Same as MarkPortAndAddRelayFailoverLink except blocks until the link is discovered or timed out . . . 148
11.5.15 Remove SEL relay failover mode from a port . . . 149
11.5.16 Filter ports by a port attribute . . . 149
11.6 OpenFlow Ports . . . 150
11.6.1 Read OpenFlow port descriptions . . . 150
11.6.2 Modify the OpenFlow settings of a port . . . 151
11.7 SEL-2740S Ports . . . 152
11.7.1 Read OpenFlow port diagnostics . . . 152
11.8 Links . . . 153
11.8.1 Read configured links . . . 153
11.8.2 Read detected links . . . 154
11.8.3 Create configured links . . . 155
11.8.4 Add an operational link from host to switch . . . 156
11.8.5 Update configured links (PUT) . . . 156
11.8.6 Update configured links (PATCH) . . . 157
11.8.7 Replace the configuration object for a link . . . 158
11.8.8 Delete configured links . . . 159

11.8.9 Adopt a link with default settings . . . 159
11.8.10 Adopt a link with a predefined link object . . . 160
11.8.11 Unadopt a link . . . 160
11.8.12 Filter links by a port attribute . . . 161
11.9 Default Adoption Settings . . . 162
11.9.1 Read default adoption settings and default ARP SPA IP address . . . 162
11.9.2 Update default adoption settings and default ARP SPA IP address (PUT) . . . 163
11.9.3 Update default adoption settings and default ARP SPA IP address (PATCH) 163

12 Logical Programming 165
12.1 Logical Connections . . . 165
12.1.1 Read created logical connections . . . 166
12.1.2 Create logical connections . . . 166
12.1.3 Replan all logical connections . . . 169
12.1.4 Delete created logical connections . . . 170
12.1.5 Delete all logical connections . . . 170
12.1.6 Resubmit a logical connection . . . 171
12.1.7 Replan the IB management logical connection to a switch . . . 172
12.1.8 Get the Openflow counters for a logical connection . . . 173
12.1.9 Get the OpenFlow entries for a logical connection . . . 174
12.1.10 Get path plans for a logical connection . . . 174
12.2 Communication Service Types . . . 175
12.2.1 Read created communication service types (CST) . . . 175
12.2.2 Create communication service types (CST) . . . 176
12.2.3 Update created communication service types (CST) (PUT) . . . 179
12.2.4 Update created communication service types (CST) (PATCH) . . . 179
12.2.5 Delete created communication service types (CST) . . . 180
12.3 VID Reservation . . . 181
12.3.1 Read reserved VIDs . . . 181
12.3.2 Reserve a VID or range of VIDs . . . 182
12.3.3 Delete reserved VIDs . . . 184

13 OpenFlow Programming 185
13.1 Flow Entries . . . 185
13.1.1 Read flow entries . . . 187
13.1.2 Create flow entries . . . 188
13.1.3 Update flow entries (PUT) . . . 190
13.1.4 Reset the counters for a flow entry . . . 192
13.1.5 Delete flow entries . . . 193
13.1.6 Read flow entry diagnostics . . . 194
13.1.7 Resolving Aliased field values . . . 194
13.1.8 Write-actions . . . 196
13.1.9 Meter . . . 196
13.2 Group Entries . . . 196
13.2.1 Read Group entries . . . 197
13.2.2 Create Group entries . . . 198
13.2.3 Update Group entries (PUT) . . . 202

13.2.4 Update Group entries (PATCH) . . . 203
13.2.5 Delete Group entries . . . 206
13.2.6 Read Group entry descriptions . . . 206
13.2.7 Read group entry diagnostics . . . 207
13.3 Meter Entries . . . 207
13.3.1 Read Meter entries . . . 208
13.3.2 Create Meter entries . . . 209
13.3.3 Update Meter entries (PUT) . . . 210
13.3.4 Update Meter entries (PATCH) . . . 211
13.3.5 Delete Meter entries . . . 212
13.3.6 Read meter descriptions . . . 212
13.3.7 Read meter features . . . 213
13.3.8 Read meter diagnostics . . . 214

14 Controller Management 215
14.1 Commissioning . . . 215
14.1.1 Commissioned status of SEL-5056 . . . 215
14.2 Database Management . . . 215
14.2.1 Backup the SEL-5056 database . . . 216
14.2.2 Restore a SEL-5056 database . . . 216
14.3 Event Categories . . . 217
14.3.1 Read event categories . . . 217
14.3.2 Update event categories (PATCH) . . . 218
14.3.3 Update event categories (PUT) . . . 219
14.4 Event Types . . . 220
14.4.1 Read event categories . . . 220
14.5 Session Management . . . 221
14.5.1 Read maximum login attempts, lockout seconds, and login attempt windows 221
14.5.2 Read usage policy, maximum number of concurrent sessions, and session timeout 222
14.5.3 Update maximum login attempts, lockout seconds, and login attempt windows (PATCH) . . . 223
14.5.4 Update maximum login attempts, lockout seconds, and login attempt windows (PUT) . . . 224
14.5.5 Update usage policy, maximum number of concurrent sessions, and session timeout (PATCH) . . . 225
14.5.6 Update usage policy, maximum number of concurrent sessions, and session timeout (PUT) . . . 226
14.6 Strings . . . 226
14.6.1 Read localization strings . . . 226
14.7 System Messages . . . 227
14.7.1 Read system messages . . . 227

15 Packet In/Out 229
15.1 Packet In/Out Management . . . 229
15.1.1 Read Capture packets sent to the controller . . . 229
15.1.2 Inject a packet into the network . . . 230

16 Application Management 233

16.1 Application Management . . . 233
16.1.1 Request application registration information from an external URL . . . 234
16.1.2 Create registered applications . . . 235
16.1.3 Delete registered applications . . . 236
16.1.4 Read registered applications . . . 236
16.1.5 Enable an application . . . 237
16.1.6 Disable an application . . . 237

17 Certificate Management 239
17.1 Certificate Management . . . 239
17.1.1 Read X.509 Certificates . . . 239
17.1.2 Get a certificate in PEM format . . . 240
17.1.3 Upload an X.509 certificate . . . 241
17.1.4 Revoke an X.509 certificate . . . 242
17.1.5 Revoke a X.509 certificate by its purpose . . . 242
17.1.6 Revoke an X.509 certificate by its thumprint . . . 243

IV Additional Topics 245

18 Situational Awareness, Diagnostics, and Counters 249
18.1 Counters . . . 249
18.1.1 Direct OpenFlow Counters . . . 249
18.1.2 Logical Connections . . . 250
18.2 Diagnostics . . . 250
18.2.1 OpenFlow . . . 250
18.3 Situational Awareness . . . 250
18.3.1 Logical Connections . . . 250

19 Asynchronous Data Exchange with SignalR 251
19.1 Hubs . . . 251
19.2 Authorization . . . 251
19.3 Permissions . . . 252
19.4 WebSocket . . . 252

20 Permissions and Roles 257

V ODата Definitions 259

21 Native OData Types 263
21.1 Edm.DateTimeOffset . . . 263
21.2 Edm.Duration . . . 263

22 Primary OData Types 265
22.1 Sel.BlueFrame.ApplicationRegistration.ApplicationLink . . . 265
22.2 Sel.BlueFrame.Core.EventBus.EventCategory . . . 266
22.3 Sel.BlueFrame.Core.EventBus.EventType . . . 266
22.3.1 Sel.BlueFrame.Core.EventBus.ClearedEventType . . . 267

22.4 Sel.BlueFrame.Core.EventBus.LocaleString 267
22.5 Sel.BlueFrame.Core.SecurityManager.AuthService 267
22.5.1 Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService 267
22.6 Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup 268
22.7 Sel.BlueFrame.Core.SecurityManager.Role 269
22.8 Sel.BlueFrame.Core.SecurityManager.User 269
22.8.1 Sel.BlueFrame.Core.SecurityManager.LocalUser 270
22.8.2 Sel.BlueFrame.Core.SecurityManager.AuthServiceUser 270
22.9 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction 271
22.10 Sel.BlueFrame.RestBroker.Plumbing.RestSettings 271
22.11 Sel.BlueFrame.SecurityManager.SecurityManagerSettings 271
22.12 Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation 272
22.13 Sel.BlueFrame.UserPreferences.Preference 273
22.14 Sel.Sel5056.LearnAndLock.LearnAndLockSettings 273
22.15 Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection . 273
22.16 Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession 274
22.17 Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType . 276
22.17.1 Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceTypePreDefined 277
22.18 Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection 277
22.19 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow 278
22.20 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats 279
22.21 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group 280
22.22 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc 280
22.23 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats 281
22.24 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter 281
22.25 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions 282
22.26 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures 282
22.27 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats 283
22.28 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc 283
22.29 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats 283
22.30 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ReceivedPacket 284
22.31 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats 284
22.32 Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation 284
22.32.1 Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidDestinationReservation 285
22.33 Sel.Sel5056.SystemMessaging.SystemMessage 285
22.34 Sel.Sel5056.TopologyManager.Links.ConfigLink 286
22.35 Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink 286
22.36 Sel.Sel5056.TopologyManager.NetworkSettings 287
22.37 Sel.Sel5056.TopologyManager.Nodes.ConfigNode 287
22.37.1 Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode 288
22.38 Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode 289
22.39 Sel.Sel5056.TopologyManager.Ports.ConfigPort 290
22.40 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort 290
22.41 Sel.Sel5056.TopologyManager.Synchronization.SynchronizationRequest 291

22.41.1 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowAddSynchronizationRequest . . . 292
22.41.2 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowDeleteSynchronizationRequest . . . 292
22.41.3 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModifySynchronizationRequest . . . 293
22.41.4 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupAddSynchronizationRequest . . . 293
22.41.5 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupDeleteSynchronizationRequest . . . 294
22.41.6 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupModifySynchronizationRequest . . . 294
22.41.7 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterAddSynchronizationRequest . . . 294
22.41.8 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterDeleteSynchronizationRequest . . . 295
22.41.9 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterModifySynchronizationRequest . . . 295
22.41.10Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization 296
22.41.11Sel.Sel5056.SapphirePlugin.Synchronization.FirmwareUpgradeRequiredSynchronization . . . 296

23 Secondary OData Types 297
23.1 Sel.BlueFrame.ApplicationRegistration.AuthenticationType . . . 297
23.2 Sel.BlueFrame.CommissioningManager.CommissioningConcern . . . 297
23.2.1 Sel.BlueFrame.CommissioningManager.UserCommissioner.UserCommissioningConcern . . . 298
23.3 Sel.BlueFrame.Core.EventBus.Behaviors.Behavior . . . 298
23.3.1 Sel.BlueFrame.EventBus.Syslog.SyslogBehavior . . . 298
23.3.2 Sel.BlueFrame.EventBus.WindowsEventLog.WindowsEventLogBehavior . . 299
23.3.3 Sel.BlueFrame.Particle.AlarmContact.AlarmContactBehavior . . . 299
23.3.4 Sel.BlueFrame.Particle.LocalEventStore.LocalEventStoreBehavior . . . 299
23.4 Sel.BlueFrame.Core.EventBus.LocaleRecord . . . 299
23.5 Sel.BlueFrame.Core.FileArgument . . . 300
23.6 Sel.BlueFrame.Core.FileReturnValue . . . 300
23.7 Sel.BlueFrame.Core.Networking.NetworkInterfaceAddress . . . 300
23.8 Sel.BlueFrame.Core.SecurityManager.Permission . . . 301
23.8.1 Sel.BlueFrame.Core.DataBrokerPermission . . . 301
23.8.2 Sel.BlueFrame.Core.SecurityAttributePermission . . . 301
23.8.3 Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission . . . 301
23.8.4 Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission . 302
23.9 Sel.BlueFrame.Core.SecurityManager.UserFailedLogin . . . 302
23.10 Sel.BlueFrame.Core.TrustAuthority.CertificatePurpose . . . 302
23.10.1 Sel.BlueFrame.TrustAuthority.TrustedCertificates.TrustedCertificatePurpose 303
23.10.2 Sel.Sel5056.Drivers.OpenFlow.Trust.OpenFlowDeviceCertificate . . . 303
23.10.3 Sel.Sel5056.Drivers.OpenFlow.Trust.OpenFlowServerCertificate . . . 303

23.10.4 Sel.BlueFrame.TrustAuthority.WebCertificateManager.WebCertificatePurpose . . . 303
23.10.5 Sel.BlueFrame.TrustAuthority.InternalCertificateAuthority.InternalCaPurpose . . . 303
23.10.6 Sel.Sel5056.SapphirePlugin.Trust.SapphireDeviceCertificate . . . 303
23.10.7 Sel.Sel5056.SapphirePlugin.Trust.SapphireServerCertificate . . . 303
23.11 Sel.BlueFrame.SecurityManager.AuthServiceGroupDescription . . . 304
23.12 Sel.BlueFrame.SecurityManager.AuthServiceResult . . . 304
23.13 Sel.BlueFrame.SecurityManager.AuthServiceResultCollection . . . 304
23.14 Sel.BlueFrame.SecurityManager.GroupCollection . . . 304
23.15 Sel.BlueFrame.UserPreferences.Setting . . . 305
23.16 Sel.FileArgument . . . 305
23.17 Sel.PathPlanHop . . . 305
23.18 Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions 305
23.19 Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.MeterMetadata . . . 306
23.20 Sel.Sel5056.LearnAndLock.CSV.AdoptedHostInfo . . . 306
23.21 Sel.Sel5056.LearnAndLock.CSV.AdoptedLinkInfo . . . 307
23.22 Sel.Sel5056.LearnAndLock.CSV.AdoptedSwitchInfo . . . 307
23.23 Sel.Sel5056.LearnAndLock.Configuration.AutoAdoptConfig . . . 308
23.24 Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.Configuration.LogicalConnectionLearningConfig . . . 308
23.25 Sel.Sel5056.LearnAndLock.ProposedCommunicationServiceType . . . 309
23.26 Sel.Sel5056.LearnAndLock.Sessions.AutoAdoptSession . . . 310
23.27 Sel.Sel5056.LearnAndLock.Sessions.LogicalConnectionLearningSession . . . 310
23.28 Sel.Sel5056.LearnAndLock.Status.LearnAndLockCounters . . . 310
23.29 Sel.Sel5056.LearnAndLock.Status.LearnAndLockStateBeginEndTimestamps . . . 311
23.30 Sel.Sel5056.LogicalConnectionValidation.DataObject.AutoFillSettings . . . 311
23.31 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Action . . . 312
23.31.1 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetQueueAction . . . 313
23.31.2 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupAction . . . 313
23.31.3 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputAction . . . 313
23.31.4 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PopVlanAction . . . 314
23.31.5 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PushVlanAction . . . 314
23.31.6 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetFieldAction . . . 314
23.32 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BandStat . . . 314
23.33 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket . . . 314
23.33.1 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupBucketByAlias . . . 315
23.33.2 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortBucketByAlias . . . 315
23.34 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BucketStats . . . 316
23.35 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowData . . . 316
23.36 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat . . . 316
23.37 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupData . . . 317
23.38 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat . . . 317
23.39 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction . . . 318
23.39.1 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ApplyActions . . . 319
23.39.2 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ClearActions . . . 319
23.39.3 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GoToTable . . . 319

23.39.4 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterInstruction 319
23.39.5 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions 319
23.40 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match 319
23.41 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.MatchField 320
23.41.1 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpOp 323
23.41.2 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpSpa 323
23.41.3 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpTpa 323
23.41.4 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthDst 323
23.41.5 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthSrc 323
23.41.6 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthType 324
23.41.7 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.InPort 324
23.41.8 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.IpProto 324
23.41.9 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Dst 324
23.41.10 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Src 324
23.41.11 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.TcpDst 325
23.41.12 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.TcpSrc 326
23.41.13 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.UdpDst 326
23.41.14 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.UdpSrc 326
23.41.15 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.VlanPcp 326
23.41.16 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.VlanVid 326
23.42 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand 326
23.42.1 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandDrop 327
23.43 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandHeader 327
23.44 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDesc 327
23.45 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStat 328
23.46 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port 328
23.47 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStat 329
23.48 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStat 329
23.49 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency . 330
23.49.1 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.FlowDeleteDependency 331
23.49.2 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupDeleteDependency 331
23.49.3 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupDependency 331
23.49.4 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.MeterDependency 331
23.49.5 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.FlowModifyDependency 332
23.49.6 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupModifyDependency 332
23.50 Sel.Sel5056.PathPlanner.DataObject.PathPlanHopCollection 332
23.51 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.-AmDestination 332
23.51.1 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.-Types.RelayFailoverDestination 333
23.51.2 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.-Types.SimpleDestination 333

23.51.3 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.-Types.TraditionalTiePointDestination . . . 333
23.51.4 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.-Types.ProgrammableDestination . . . 334
23.51.5 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.-Types.InbandDestination . . . 334
23.52 Sel.Sel5056.PathPlanner.RestController.DirectionalLink . . . 334
23.53 Sel.Sel5056.PathPlanner.RestController.FlowReferences . . . 335
23.54 Sel.Sel5056.PathPlanner.RestController.FlowWithStats . . . 335
23.55 Sel.Sel5056.PathPlanner.RestController.GroupReferences . . . 335
23.56 Sel.Sel5056.PathPlanner.RestController.GroupWithStats . . . 336
23.57 Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats . . . 336
23.58 Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo . . . 337
23.59 Sel.Sel5056.PathPlanner.RestController.PathPlanInfo . . . 337
23.60 Sel.Sel5056.Tags.AmTag . . . 338
23.60.1 Sel.Sel5056.Tags.LogicalConnectionTag . . . 339
23.60.2 Sel.Sel5056.Tags.InBandPortIdTag . . . 339
23.60.3 Sel.Sel5056.Tags.GenericArpFlowTag . . . 339
23.60.4 Sel.Sel5056.Tags.SystemTag . . . 339
23.60.5 Sel.Sel5056.Tags.OpenFlowPortIdTag . . . 339
23.60.6 Sel.Sel5056.Tags.KeyValueTag . . . 339
23.60.7 Sel.Sel5056.Tags.InBandTag . . . 339
23.60.8 Sel.Sel5056.Tags.ControllerTag . . . 340
23.60.9 Sel.Sel5056.Tags.IpAddressTag . . . 340
23.60.10 Sel.Sel5056.Tags.MacAddressTag . . . 340
23.60.11 Sel.Sel5056.Tags.ReadoptTag . . . 340
23.60.12 Sel.Sel5056.Tags.ControllerRouteTag . . . 340
23.60.13 Sel.Sel5056.Tags.NoArpTag . . . 340
23.60.14 Sel.Sel5056.LogicalConnectionValidation.Tags.MatchableCst . . . 341
23.60.15 Sel.Sel5056.LearnAndLock.AutoAdoption.Tags.AutoAdoptionTag . . . 341
23.61 Sel.Sel5056.TopologyManager.Attributes.AmAttribute . . . 341
23.61.1 Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr 342
23.61.2 Sel.Sel5056.TopologyManager.Attributes.Operational.Port.Sel2740SPortAttr 343
23.61.3 Sel.Sel5056.TopologyManager.Attributes.Operational.Port.Ethernet-ApplicationAttr . . . 343
23.61.4 Sel.Sel5056.TopologyManager.Attributes.Operational.Port.SelRelay-FailoverModeAttr . . . 343
23.61.5 Sel.Sel5056.TopologyManager.Attributes.Operational.Port.Ethernet-ApplicationNodeAttr . . . 343
23.61.6 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.OpenFlowAttr . 344
23.61.7 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.IpAttr . . . 344
23.61.8 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr 344
23.61.9 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.Firmware-UpdateAttr . . . 345
23.61.10 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.UserHintedIp-AddressAttr . . . 345
23.61.11 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.NoArpAttr . . . 346

23.61.12 Sel.Sel5056.TopologyManager.Attributes.Operational.Link.OfdpDiscovered . 346
23.61.13 Sel.Sel5056.TopologyManager.Attributes.Operational.Link.OpenFlowVerified 346
23.61.14 Sel.Sel5056.TopologyManager.Attributes.Operational.Link.ArpVerified . . . 346
23.61.15 Sel.Sel5056.TopologyManager.Attributes.Operational.Link.UserHintedLinkAttr 346
23.61.16 Sel.Sel5056.TopologyManager.Attributes.Operational.ControllerAttr 346
23.61.17 Sel.Sel5056.TopologyManager.Attributes.Operational.EthernetAttr 346
23.61.18 Sel.Sel5056.TopologyManager.Attributes.Operational.DoNotExpireAttr . . 347
23.61.19 Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedMappedSwitchPortAttr 347
23.61.20 Sel.Sel5056.TopologyManager.Attributes.Operational.NetworkingAbstractionAttr 347
23.61.21 Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedNodeAttr . 347
23.61.22 Sel.Sel5056.Southbound.DeviceManagement.CommissioningStateAttr 348
23.62 Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationProviderInformation 348
23.63 Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractSettingBase . . . 349
23.63.1 Sel.Sel5056.TopologyManager.Configurations.Abstraction.TraditionalSwitch 349
23.64 Sel.Sel5056.TopologyManager.Configurations.AmConfig 349
23.64.1 Sel.Sel5056.TopologyManager.Configurations.ControllerConfig 350
23.64.2 Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriority . . 350
23.64.3 Sel.Sel5056.TopologyManager.Configurations.OpenFlow.OpenFlowConfig . 350
23.64.4 Sel.Sel5056.TopologyManager.Configurations.FailOver.SelRelayFailoverModeConfig 350
23.64.5 Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractionConfig 350
23.64.6 Sel.Sel5056.TopologyManager.Configurations.Sapphire.AlternateIpConfig . . 350
23.65 Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.AmSystemPriority 351
23.65.1 Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.-SystemCritical 352
23.65.2 Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.-SystemRequired 352
23.66 Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.RateTypeBase 352
23.66.1 Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.Burst 352
23.66.2 Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.-KilobitesPerSecond 352
23.66.3 Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.-PacketsPerSecond 353
23.66.4 Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.Stats 353
23.67 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.ConfigBase . 353
23.67.1 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.NoForward 353
23.67.2 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.NoPacketIn 354
23.67.3 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.NoReceive 354

23.67.4 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.PortDown . . . 354
23.68 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.FeatureBase 354
23.68.1 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.AutoNegotiation . . . 355
23.68.2 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.Copper . . . 355
23.68.3 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.Fiber 355
23.68.4 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.FortyGigaBitFullDuplex . . . 355
23.68.5 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneGigaBitFullDuplex . . . 355
23.68.6 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneGigaBitHalfDuplex . . . 356
23.68.7 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneHundredGigaBitFullDuplex . . . 356
23.68.8 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneHundredMegaBitFullDuplex . . . 356
23.68.9 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneHundredMegaBitHalfDuplex . . . 356
23.68.10 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneTeraBitFullDuplex . . . 356
23.68.11 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OtherBitRate . . . 356
23.68.12 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.Pause 356
23.68.13 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.PauseAsymmetric . . . 356
23.68.14 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.TenGigaBitFullDuplex . . . 356
23.68.15 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.TenMegaBitFullDuplex . . . 357
23.68.16 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.TenMegaBitHalfDuplex . . . 357

24 OData Actions 359
24.1 Sel.AcceptAllProposedLearnedLogicalConnections . . . 359
24.2 Sel.AcceptLearnedLogicalConnections . . . 359
24.3 Sel.AcknowledgeLearnAndLockSession . . . 359
24.4 Sel.AddAbstractNode . . . 359
24.5 Sel.AddOperationalLinkFromHostToSwitch . . . 360
24.6 Sel.AddSilentHost . . . 360
24.7 Sel.Adopt . . . 360
24.7.1 Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode . . . 360
24.7.2 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort . . . 360
24.7.3 Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink . . . 360
24.8 Sel.AdoptWithConfig . . . 361
24.8.1 Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode . . . 361

24.8.2 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort 361
24.8.3 Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink 361
24.9 Sel.AsPem 361
24.10 Sel.Backup 362
24.11 Sel.BeginLearnAndLock 362
24.12 Sel.ClearLearnedLogicalConnections 362
24.13 Sel.ClearTransaction 362
24.13.1 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction 363
24.13.2 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction 363
24.13.3 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction 363
24.13.4 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction 363
24.13.5 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction 363
24.14 Sel.Commit 363
24.14.1 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction 363
24.14.2 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction 364
24.14.3 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction 364
24.14.4 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction 364
24.14.5 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction 364
24.15 Sel.DeclineAllProposedLearnedLogicalConnections 364
24.16 Sel.DeclineLearnedLogicalConnections 364
24.17 Sel.DeleteAll 364
24.18 Sel.DeleteLearnAndLockSession 365
24.19 Sel.DetectSelRelayFailoverPort 365
24.20 Sel.DisableApplicationLink 365
24.21 Sel.DisableControllerArpToNode 365
24.22 Sel.Disconnect 365
24.23 Sel.EnableApplicationLink 365
24.24 Sel.EnableControllerArpToNode 366
24.25 Sel.FactoryDefaultReset 366
24.26 Sel.FirmwareUpgrade 366
24.27 Sel.GetAvailableGroups 366
24.28 Sel.GetHardTimeoutTimeRemaining 366
24.29 Sel.GetLearnAndLockCounters 366
24.30 Sel.GetLinkByPortAttribute 367
24.31 Sel.GetLogicalConnectionStats 367
24.32 Sel.GetNodeByAttribute 367
24.33 Sel.GetNodesByIp 367
24.34 Sel.GetNodesThatRequireSynchronization 367
24.35 Sel.GetOpenFlowInformation 368
24.36 Sel.GetPathPlanInformation 368
24.37 Sel.GetPortByAttribute 368
24.38 Sel.GetStateBeginEndTimestamps 368
24.39 Sel.GetStatusMessages 368
24.40 Sel.GetUserPreference 369
24.41 Sel.Heartbeat 369
24.42 Sel.InitiateApplicationRegistration 369
24.43 Sel.IsCommissioned 369

24.44 Sel.LearnAndLockIndividualSessionExportCSV 369
24.45 Sel.MarkPortAndAddRelayFailoverLink 369
24.46 Sel.MarkPortAndPerformBlockingDetectSelRelayFailover 370
24.47 Sel.MarkPortAsSelRelayFailover 370
24.48 Sel.MergeNodes 370
24.49 Sel.PacketOut 370
24.50 Sel.ProposeLearnedLogicalConnections 370
24.51 Sel.Reboot 371
24.52 Sel.RemoveSelRelayFailoverFromPort 371
24.53 Sel.ReplaceConfig 371
24.53.1 Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode 371
24.53.2 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort 371
24.53.3 Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink 372
24.54 Sel.ReplanAll 372
24.55 Sel.ReplanInbandPath 372
24.56 Sel.ReserveVlanVidRange 372
24.57 Sel.ResetCounters 372
24.58 Sel.Restore 372
24.59 Sel.ResubmitLogicalConnection 373
24.60 Sel.RevokeCertificate 373
24.61 Sel.RevokeCertificateByPurpose 373
24.62 Sel.RevokeCertificateByThumbprint 373
24.63 Sel.SendPortMod 373
24.63.1 Sel.Sel5056.TopologyManager.Ports.ConfigPort 374
24.63.2 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort 374
24.64 Sel.SetDone 374
24.65 Sel.SetUserPreference 374
24.66 Sel.StopLearnAndLock 375
24.67 Sel.SynchronizeConfiguration 375
24.68 Sel.SynchronizeOpenFlowConfiguration 375
24.69 Sel.TestAuthenticateUser 375
24.70 Sel.Unadopt 375
24.70.1 Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode 376
24.70.2 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort 376
24.70.3 Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink 376
24.71 Sel.UpdatePassword 376
24.72 Sel.UploadCertificate 376
24.73 Sel.UserInitiatedHostDiscovery 376

25 OData Enums 379
25.1 Sel.BlueFrame.Core.ErrorState 379
25.2 Sel.BlueFrame.Core.EventBus.Enums.DurationType 380
25.3 Sel.BlueFrame.Core.EventBus.Enums.FacilityCode 380
25.4 Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel 381
25.5 Sel.BlueFrame.Core.TrustAuthority.Enums.CertificateStatus 382
25.6 Sel.BlueFrame.Core.TrustAuthority.Enums.UserInteraction 382
25.7 Sel.BlueFrame.EventBus.Syslog.RFCType 382

25.8 Sel.BlueFrame.EventBus.Syslog.TransportType 383
25.9 Sel.BlueFrame.SecurityManager.AuthServiceResultState 383
25.10 Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.NodesConfiguration . . . 383
25.11 Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.OutputConfiguration . . 384
25.12 Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.TableConfiguration . . . 384
25.13 Sel.Sel5056.LearnAndLock.Enums.EndReason 384
25.14 Sel.Sel5056.LearnAndLock.LearnAndLockStateStatus 385
25.15 Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.AutoAcceptStatus 386
25.16 Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnectionState 386
25.17 Sel.Sel5056.LogicalConnectionValidation.Enums.CstCommunicationType 387
25.18 Sel.Sel5056.LogicalConnectionValidation.Enums.LogicalConnectionOrigin 387
25.19 Sel.Sel5056.OpenFlowPlugin.Enums.OF13MeterBandType 387
25.20 Sel.Sel5056.OpenFlowPlugin.Enums.OfpGroupType 388
25.21 Sel.Sel5056.SapphirePlugin.Synchronization.SapphireEndpoint 388
25.22 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.DeviceFramework . . . 389
25.23 Sel.Sel5056.TopologyManager.Enums.CommunicationBand 389
25.24 Sel.Sel5056.TopologyManager.Enums.DiscoveryTrustState 390
25.25 Sel.Sel5056.TopologyManager.Enums.OfpPortStatus 390
25.26 Sel.Sel5056.TopologyManager.Enums.PacketSource 391
25.27 Sel.Sel5056.TopologyManager.Enums.State 392
25.28 System.Diagnostics.EventLogEntryType 392
25.29 System.DirectoryServices.Protocols.AuthType 393

VI Appendix 395

A Summary of Changes to this document 397

B Summary of Changes to REST Interface 399

B.1 v2.2.0.0 399
B.1.1 Added Endpoints 399
B.1.2 Removed Endpoints 400
B.1.3 Added Types 401
B.1.4 Removed Types 403
B.1.5 Added Properties 405
B.1.6 Removed Properties 406
B.1.7 Added Property Attributes 406
B.1.8 Removed Property Attributes 407
B.1.9 Removed Property Attributes 407

List of Figures

5.1 Tutorial Network Topology 11

List of Tables

5.1 Header values for /api/default/security/Commission 12
5.2 Body values for Commissioning the SEL-5056 12
5.3 Header values for /configuration/settings.json 13
5.4 Header values for /identity/connect/token 18
5.5 Body values for Logging in with a username and password 18
5.6 Header values for /api/default/settings/Heartbeat 19
5.7 Header values for /identity/connect/accesstokenvalidation 20
5.8 Parameter values for /identity/connect/accesstokenvalidation 20
5.9 Header values for /identity/connect/revocation 21
5.10 Body values for Logging Out 22
5.11 Header values for /api/default/security/users 23
5.12 Body values for Creating a Permission Level 3 user 23
5.13 Header values for /api/default/security/users 25
5.14 Parameter values for /api/default/security/users 25
5.15 Header values for /identity/connect/token 26
5.16 Body values for Creating Configuration Nodes 27
5.17 Header values for /api/default/config/nodes 28
5.18 Body values for Creating Configuration Nodes 28
5.19 Header values for /api/default/config/nodes 31
5.20 Body values for Creating Configuration Nodes 31
5.21 Header values for /api/default/operational/nodes 33
5.22 Header values for /api/default/operational/ports 35
5.23 Parameter values for /api/default/operational/ports 35
5.24 Header values for /api/default/operational/links 37
5.25 Header values for /api/default/config/nodes 37
5.26 Parameter values for /api/default/config/nodes 37
5.27 Header values for /api/default/operational/nodes 38
5.28 Body values for Adopting the Hosts, including failover mode 39
5.29 Header values for /api/default/operational/ports 39

5.30 Header values for /api/default/operational/links . . . 41
5.31 Header values for /api/default/operational/ports . . . 41
5.32 Header values for /api/default/operational/ports . . . 42
5.33 Header values for /api/default/config/logicalConnections . . . 42
5.34 Body values for Creating Logical Conenctions . . . 43
5.35 Header values for /api/default/config/logicalConnections . . . 44
5.36 Body values for Creating Logical Conenctions . . . 44
5.37 Header values for /api/default/config/communicationServiceTypes . . . 46
5.38 Body values for Creating CSTs . . . 46
5.39 Header values for /api/default/config/nodes . . . 49
5.40 Body values for Adding an offline host . . . 49
5.41 Header values for /api/default/config/flows . . . 50
5.42 Body values for Managing Flow Entries . . . 51
5.43 Header values for /api/default/config/groups . . . 54
5.44 Body values for Managing Group Entries . . . 55
5.45 Header values for /api/default/config/meters . . . 58
5.46 Body values for Managing Meter Entries . . . 58

8.2 OData Service Roots . . . 71

9.1 Supported OAuth Grant Types . . . 74
9.2 OAuth Grant Types Requirements . . . 74
9.3 OAuth Grant Types Client ID and Secret . . . 75
9.5 OAuth/Token Endpoints . . . 75
9.9 Resource Owner Password Credentials Token Request Parameters . . . 76
9.11 Client Credentials Token Request Parameters . . . 77

10.3 Default Values for Sel.BlueFrame.Core.SecurityManager.LocalUser . . . 89
10.5 Default Values for Sel.BlueFrame.Core.SecurityManager.AuthServiceUser . . . 90
10.9 Default Values for Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService . 97
10.12 Default Values for Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup . . . 102

11.5 Default Values for Sel.Sel5056.TopologyManager.Nodes.ConfigNode . . . 114
11.7 Default Values for Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode . . . 115
11.13 Default Values for Sel.Sel5056.TopologyManager.Ports.ConfigPort . . . 141
11.18 Default Values for Sel.Sel5056.TopologyManager.Links.ConfigLink . . . 155

12.3 Default Values for Sel.Sel5056.LogicalConnectionValidation.DataObject.-CommunicationServiceType . . . 177

13.3 Default Values for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow . . . 188
13.7 Default Values for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group . . . 198
13.10 Default Values for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter . . . 209

16.3 Default Values for Sel.BlueFrame.ApplicationRegistration.ApplicationLink . . . 235

21.2 Edm.* Types . . . 263

22.2 Properties for Sel.BlueFrame.ApplicationRegistration.ApplicationLink . . . 265
22.4 Properties for Sel.BlueFrame.Core.EventBus.EventCategory . . . 266

22.6 Properties for Sel.BlueFrame.Core.EventBus.EventType 266
22.8 Sel.BlueFrame.Core.EventBus.ClearedEventType 267
22.10 Properties for Sel.BlueFrame.Core.EventBus.LocaleString 267
22.12 Properties for Sel.BlueFrame.Core.SecurityManager.AuthService 267
22.14 Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService 268
22.16 Properties for Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup 269
22.18 Properties for Sel.BlueFrame.Core.SecurityManager.Role 269
22.20 Properties for Sel.BlueFrame.Core.SecurityManager.User 270
22.22 Sel.BlueFrame.Core.SecurityManager.LocalUser 270
22.24 Sel.BlueFrame.Core.SecurityManager.AuthServiceUser 270
22.26 Properties for Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction . . . 271
22.28 Properties for Sel.BlueFrame.RestBroker.Plumbing.RestSettings 271
22.30 Properties for Sel.BlueFrame.SecurityManager.SecurityManagerSettings 272
22.32 Properties for Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation 272
22.34 Properties for Sel.BlueFrame.UserPreferences.Preference 273
22.36 Properties for Sel.Sel5056.LearnAndLock.LearnAndLockSettings 273
22.38 Properties for Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection 274
22.40 Properties for Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession 276
22.42 Properties for Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType 277
22.44 Properties for Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection 278
22.46 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow 279
22.48 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats 280
22.50 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group 280
22.52 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc 281
22.54 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats 281
22.56 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter 282
22.58 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions . . 282
22.60 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures 282
22.62 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats 283
22.64 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc 283
22.66 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats 283
22.68 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ReceivedPacket 284
22.70 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats 284
22.72 Properties for Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation . 285
22.74 Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidDestinationReservation . . . 285
22.76 Properties for Sel.Sel5056.SystemMessaging.SystemMessage 285
22.78 Properties for Sel.Sel5056.TopologyManager.Links.ConfigLink 286
22.80 Properties for Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink 287
22.82 Properties for Sel.Sel5056.TopologyManager.NetworkSettings 287
22.84 Properties for Sel.Sel5056.TopologyManager.Nodes.ConfigNode 288
22.86 Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode 289
22.88 Properties for Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode . . . 289
22.90 Properties for Sel.Sel5056.TopologyManager.Ports.ConfigPort 290
22.92 Properties for Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort 291

22.94 Properties for Sel.Sel5056.TopologyManager.Synchronization.SynchronizationRequest 291
22.96 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowAdd-SynchronizationRequest 292
22.98 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowDelete-SynchronizationRequest 293
22.100 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModify-SynchronizationRequest 293
22.102 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupAdd-SynchronizationRequest 294
22.104 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupDelete-SynchronizationRequest 294
22.106 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupModify-SynchronizationRequest 294
22.108 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterAdd-SynchronizationRequest 295
22.110 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterDelete-SynchronizationRequest 295
22.112 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterModify-SynchronizationRequest 296
22.114 Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization 296

23.1 Properties for Sel.BlueFrame.ApplicationRegistration.AuthenticationType 297
23.2 Properties for type Sel.BlueFrame.CommissioningManager.UserCommissioner.-UserCommissioningConcern 298
23.3 Properties for Sel.BlueFrame.Core.EventBus.Behaviors.Behavior 298
23.4 Properties for type Sel.BlueFrame.EventBus.Syslog.SyslogBehavior 299
23.5 Properties for type Sel.BlueFrame.EventBus.WindowsEventLog.WindowsEventLogBehavior 299
23.6 Properties for Sel.BlueFrame.Core.EventBus.LocaleRecord 300
23.7 Properties for Sel.BlueFrame.Core.FileArgument 300
23.8 Properties for Sel.BlueFrame.Core.FileReturnValue 300
23.9 Properties for Sel.BlueFrame.Core.Networking.NetworkInterfaceAddress 300
23.10 Properties for Sel.BlueFrame.Core.SecurityManager.Permission 301
23.11 Properties for type Sel.BlueFrame.Core.DataBrokerPermission 301
23.12 Properties for type Sel.BlueFrame.Core.SecurityAttributePermission 301
23.13 Properties for type Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission . 302
23.14 Properties for type Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission 302
23.15 Properties for Sel.BlueFrame.Core.SecurityManager.UserFailedLogin 302
23.16 Properties for type Sel.Sel5056.Drivers.OpenFlow.Trust.OpenFlowDeviceCertificate 303
23.17 Properties for type Sel.Sel5056.SapphirePlugin.Trust.SapphireDeviceCertificate . . 303
23.18 Properties for Sel.BlueFrame.SecurityManager.AuthServiceGroupDescription . . . 304
23.19 Properties for Sel.BlueFrame.SecurityManager.AuthServiceResult 304
23.20 Properties for Sel.BlueFrame.SecurityManager.AuthServiceResultCollection 304
23.21 Properties for Sel.BlueFrame.SecurityManager.GroupCollection 304
23.22 Properties for Sel.BlueFrame.UserPreferences.Setting 305

23.23 Properties for Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions . . . 306
23.24 Properties for Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.MeterMetadata . . . 306
23.25 Properties for Sel.Sel5056.LearnAndLock.CSV.AdoptedHostInfo . . . 307
23.26 Properties for Sel.Sel5056.LearnAndLock.CSV.AdoptedLinkInfo . . . 307
23.27 Properties for Sel.Sel5056.LearnAndLock.CSV.AdoptedSwitchInfo . . . 307
23.28 Properties for Sel.Sel5056.LearnAndLock.Configuration.AutoAdoptConfig . . . 308
23.29 Properties for Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.Configuration.LogicalConnectionLearningConfig . . . 309
23.30 Properties for Sel.Sel5056.LearnAndLock.ProposedCommunicationServiceType . . . 309
23.31 Properties for Sel.Sel5056.LearnAndLock.Sessions.AutoAdoptSession . . . 310
23.32 Properties for Sel.Sel5056.LearnAndLock.Sessions.LogicalConnectionLearningSession 310
23.33 Properties for Sel.Sel5056.LearnAndLock.Status.LearnAndLockCounters . . . 310
23.34 Properties for Sel.Sel5056.LearnAndLock.Status.LearnAndLockStateBeginEndTimestamps . . . 311
23.35 Properties for Sel.Sel5056.LogicalConnectionValidation.DataObject.AutoFillSettings 311
23.36 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetQueueAction 313
23.37 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupAction . . 313
23.38 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputAction . 314
23.39 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PushVlanAction 314
23.40 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetFieldAction . 314
23.41 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BandStat . . . 314
23.42 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket . . . 315
23.43 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupBucketByAlias . . . 315
23.44 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortBucketByAlias315
23.45 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BucketStats . . . 316
23.46 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowData . . . 316
23.47 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat . . . 317
23.48 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupData . . . 317
23.49 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat . . . 318
23.50 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction . . . 318
23.51 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GoToTable . . . 319
23.52 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterInstruction 319
23.53 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match . . . 319
23.54 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.MatchField320
23.55 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpSpa . . . 323
23.56 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpTpa . . . 323
23.57 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthDst . . . 323
23.58 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthSrc . . . 324
23.59 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.InPort . . . 324

23.60 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv-4Dst . . . 324
23.61 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv-4Src . . . 324
23.62 Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.-VlanVid . . . 326
23.63 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand . . . 326
23.64 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandHeader . . . 327
23.65 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDesc . . . 327
23.66 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStat . . . 328
23.67 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port . . . 328
23.68 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStat . . . 329
23.69 Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStat . . . 329
23.70 Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency . . . 330
23.71 Properties for type Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.-FlowDeleteDependency . . . 331
23.72 Properties for type Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.-GroupDeleteDependency . . . 331
23.73 Properties for type Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.-GroupDependency . . . 331
23.74 Properties for type Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.-MeterDependency . . . 332
23.75 Properties for type Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.-FlowModifyDependency . . . 332
23.76 Properties for type Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.-GroupModifyDependency . . . 332
23.77 Properties for Sel.Sel5056.PathPlanner.DataObject.PathPlanHopCollection . . . 332
23.78 Properties for Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.-DestinationDescriptions.Types.AmDestination . . . 333
23.79 Properties for type Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.-DestinationDescriptions.Types.RelayFailoverDestination . . . 333
23.80 Properties for type Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.-DestinationDescriptions.Types.SimpleDestination . . . 333
23.81 Properties for type Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.-DestinationDescriptions.Types.TraditionalTiePointDestination . . . 334
23.82 Properties for type Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.-DestinationDescriptions.Types.ProgrammableDestination . . . 334
23.83 Properties for type Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.-DestinationDescriptions.Types.InbandDestination . . . 334
23.84 Properties for Sel.Sel5056.PathPlanner.RestController.DirectionalLink . . . 334
23.85 Properties for Sel.Sel5056.PathPlanner.RestController.FlowReferences . . . 335
23.86 Properties for Sel.Sel5056.PathPlanner.RestController.FlowWithStats . . . 335
23.87 Properties for Sel.Sel5056.PathPlanner.RestController.GroupReferences . . . 336
23.88 Properties for Sel.Sel5056.PathPlanner.RestController.GroupWithStats . . . 336
23.89 Properties for Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats . . . 337
23.90 Properties for Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo . . . 337

23.91 Properties for Sel.Sel5056.PathPlanner.RestController.PathPlanInfo 337
23.92 Properties for type Sel.Sel5056.Tags.LogicalConnectionTag 339
23.93 Properties for type Sel.Sel5056.Tags.InBandPortIdTag 339
23.94 Properties for type Sel.Sel5056.Tags.GenericArpFlowTag 339
23.95 Properties for type Sel.Sel5056.Tags.OpenFlowPortIdTag 339
23.96 Properties for type Sel.Sel5056.Tags.KeyValueTag 339
23.97 Properties for type Sel.Sel5056.Tags.InBandTag 340
23.98 Properties for type Sel.Sel5056.Tags.ControllerTag 340
23.99 Properties for type Sel.Sel5056.Tags.IpAddressTag 340
23.100Properties for type Sel.Sel5056.Tags.MacAddressTag 340
23.101Properties for type Sel.Sel5056.Tags.ControllerRouteTag 340
23.102Properties for type Sel.Sel5056.LearnAndLock.AutoAdoption.Tags.AutoAdoptionTag341
23.103Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr 343
23.104Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationAttr 343
23.105Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationNodeAttr 344
23.106Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.Node.OpenFlowAttr 344
23.107Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.Node.IpAttr344
23.108Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr 345
23.109Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.Node.FirmwareUpdateAttr 345
23.110Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.Node.UserHintedIpAddressAttr 346
23.111Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.EthernetAttr 347
23.112Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedMappedSwitchPortAttr 347
23.113Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.NetworkingAbstractionAttr 347
23.114Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedNodeAttr 348
23.115Properties for type Sel.Sel5056.Southbound.DeviceManagement.CommissioningStateAttr 348
23.116Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationProviderInformation 348
23.117Properties for type Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriority 350
23.118Properties for type Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractionConfig 350
23.119Properties for type Sel.Sel5056.TopologyManager.Configurations.Sapphire.AlternateIpConfig 350

24.1 Properties for Sel.AcceptLearnedLogicalConnections 359

24.2 Properties for Sel.AcknowledgeLearnAndLockSession 359
24.3 Properties for Sel.AddAbstractNode 360
24.4 Properties for Sel.AddOperationalLinkFromHostToSwitch 360
24.5 Properties for Sel.AddSilentHost 360
24.6 Properties for Sel.AdoptWithConfig 361
24.7 Properties for Sel.AdoptWithConfig 361
24.8 Properties for Sel.AdoptWithConfig 361
24.9 Properties for Sel.Backup 362
24.10 Properties for Sel.BeginLearnAndLock 362
24.11 Properties for Sel.ClearLearnedLogicalConnections 362
24.12 Properties for Sel.DeclineLearnedLogicalConnections 364
24.13 Properties for Sel.DeleteLearnAndLockSession 365
24.14 Properties for Sel.FirmwareUpgrade 366
24.15 Properties for Sel.GetHardTimeoutTimeRemaining 366
24.16 Properties for Sel.GetLearnAndLockCounters 367
24.17 Properties for Sel.GetLinkByPortAttribute 367
24.18 Properties for Sel.GetNodeByAttribute 367
24.19 Properties for Sel.GetNodesByIp 367
24.20 Properties for Sel.GetPortByAttribute 368
24.21 Properties for Sel.GetStateBeginEndTimestamps 368
24.22 Properties for Sel.GetStatusMessages 368
24.23 Properties for Sel.GetUserPreference 369
24.24 Properties for Sel.InitiateApplicationRegistration 369
24.25 Properties for Sel.LearnAndLockIndividualSessionExportCSV 369
24.26 Properties for Sel.MarkPortAndAddRelayFailoverLink 370
24.27 Properties for Sel.MergeNodes 370
24.28 Properties for Sel.PacketOut 370
24.29 Properties for Sel.ProposeLearnedLogicalConnections 371
24.30 Properties for Sel.ReplaceConfig 371
24.31 Properties for Sel.ReplaceConfig 371
24.32 Properties for Sel.ReplaceConfig 372
24.33 Properties for Sel.ReserveVlanVidRange 372
24.34 Properties for Sel.Restore 373
24.35 Properties for Sel.RevokeCertificateByPurpose 373
24.36 Properties for Sel.RevokeCertificateByThumbprint 373
24.37 Properties for Sel.SendPortMod 374
24.38 Properties for Sel.SendPortMod 374
24.39 Properties for Sel.SetUserPreference 374
24.40 Properties for Sel.SynchronizeConfiguration 375
24.41 Properties for Sel.SynchronizeOpenFlowConfiguration 375
24.42 Properties for Sel.TestAuthenticateUser 375
24.43 Properties for Sel.UpdatePassword 376
24.44 Properties for Sel.UploadCertificate 376
24.45 Properties for Sel.UserInitiatedHostDiscovery 377

25.1 Sel.BlueFrame.Core.ErrorState 379
25.2 Sel.BlueFrame.Core.EventBus.Enums.DurationType 380

25.3 Sel.BlueFrame.Core.EventBus.Enums.FacilityCode 381
25.4 Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel 381
25.5 Sel.BlueFrame.Core.TrustAuthority.Enums.CertificateStatus 382
25.6 Sel.BlueFrame.Core.TrustAuthority.Enums.UserInteraction 382
25.7 Sel.BlueFrame.EventBus.Syslog.RFCType 382
25.8 Sel.BlueFrame.EventBus.Syslog.TransportType 383
25.9 Sel.BlueFrame.SecurityManager.AuthServiceResultState 383
25.10 Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.NodesConfiguration . . . 384
25.11 Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.OutputConfiguration . . 384
25.12 Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.TableConfiguration . . . 384
25.13 Sel.Sel5056.LearnAndLock.Enums.EndReason 385
25.14 Sel.Sel5056.LearnAndLock.LearnAndLockStateStatus 385
25.15 Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.AutoAcceptStatus 386
25.16 Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnectionState386
25.17 Sel.Sel5056.LogicalConnectionValidation.Enums.CstCommunicationType 387
25.18 Sel.Sel5056.LogicalConnectionValidation.Enums.LogicalConnectionOrigin 387
25.19 Sel.Sel5056.OpenFlowPlugin.Enums.OF13MeterBandType 388
25.20 Sel.Sel5056.OpenFlowPlugin.Enums.OfpGroupType 388
25.21 Sel.Sel5056.SapphirePlugin.Synchronization.SapphireEndpoint 389
25.22 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.DeviceFramework . . . 389
25.23 Sel.Sel5056.TopologyManager.Enums.CommunicationBand 390
25.24 Sel.Sel5056.TopologyManager.Enums.DiscoveryTrustState 390
25.25 Sel.Sel5056.TopologyManager.Enums.OfpPortStatus 391
25.26 Sel.Sel5056.TopologyManager.Enums.PacketSource 391
25.27 Sel.Sel5056.TopologyManager.Enums.State 392
25.28 System.Diagnostics.EventLogEntryType 392
25.29 System.DirectoryServices.Protocols.AuthType 393

B.2 Added Properties 406
B.4 Removed Properties 406
B.6 Added Property Attributes 406
B.8 Removed Property Attributes 407
B.10 Modified Property Attributes 407

List of Examples

9.1 Resource Owner Password Credentials - Obtaining a Token 76
9.2 Client Credentials - Obtaining a Token 77
9.3 Inspecting a token 78
9.4 Resource Owner Password Credentials - Revoking a Token 78
9.5 Client Credentials - Revoking a Token 79
9.6 Extending the expiration time of a Token 79
10.7 Add a permission level 3 user 90

10.8 Disable a user . . . 93
11.9 Upgrade a SEL-274XS . . . 135
12.10 POST a unicast LC . . . 168
12.11 POST a multicast LC . . . 168
12.12 POST a unicast CST . . . 177
12.13 POST a multicast CST . . . 178
12.14 Reserve a range of values . . . 183
12.15 Reserve a single VID . . . 183
13.16 Create a simple flow entry . . . 189
13.17 Update a simple flow entry by changing Outport . . . 191
13.18 Add a simple All group . . . 199
13.19 Add a simple FF group . . . 200
13.20 Modify (PUT) the action buckets of an All group . . . 202
13.21 Modify (PATCH) the action buckets of an All group . . . 205
14.22 Restore the SEL-5056 database . . . 217
19.23Subscribing to all Flow entry changes using Websockets . . . 252

Visit `https://selinc.com/products/5056` for further information on the SEL-5056.

Chapter 1

Introduction

The purpose of this documentation is to describe the SEL-5056 REST interface API by providing list of endpoints, tutorials, SignalR, and other material. Documentation for the SEL-274XS REST interface API including accessing information found on the device view will be in another document.

This document is focused on using direct HTTP calls for the REST interface and not does cover any language specific bindings or utilities.

This document assumes that you understand how to make HTTP calls, are familiar with using JSON, OpenFlow, and operating the SEL-5056. Familiarity with OData is not assumed but will be helpful.

Chapter 2

Conventions

This section covers extra formatting of this document.

2.1 URIs

To save space, the first part of URIs are removed (schema and authority). Assuming the FQDN (Fully qualified domain name) of the SEL-5056 is localhost, then the URI, as used in the HTTP request itself, of: /api/default/config/nodes/ corresponds to the URL address of: https://localhost/api/default/config/nodes/.

All OData REST calls (i.e. those that are not OAuth or SingalR) are prefixed by /api/default/ in this version of the SEL-5056. Occasionally when context is clear outside of HTTP messages, the /api/default prefix may be replaced with the character ~, so the URI ~/config/nodes/ corresponds to https://localhost/api/default/config/nodes/.

2.2 Displaying HTTP Messages

To fit the width of this document and to reduce the size the HTTP messages included in this document, HTTP messages may be formatted in the following manner:

- Most responses from the SEL-5056 are chunked so have a transfer encoding of chucked. In these examples, the body is shown unchunked, but the Transfer-Encoding header is still indicates the original chunked Transfer-Encoding.
- Due to the width of this document some of the code has been modified to fit the line length limits such as the insertion of new lines.
- If the Transfer-Encoding is JSON, the body is displayed in a modified "pretty print" style to be more readable.
- Only the necessary minimal HTTP set of headers are included in requests for examples or explanations.

- Removal of the following headers from the SEL-5056 HTTP responses:
  - Cache-Control
  - Date
  - OData-Version
  - Pragma
  - Location
  - Server
- Change the OAuth token to 000...000.

Chapter 3

Getting Started

There are several ways to get started.

- Start with the tutorial section and follow along to familiarize yourself with a selected list of endpoints.
- Start at the OAuth section to become familiar with authentication and then look through the Endpoint definitions sections for specific endpoints.
- Start at Data model section to get a better sense of the relationship between objects. This is implicit in the tutorials, but made explicit there. The endpoint sections covers only a limited part of the data model to that specific in each section.

Part I

Overview

Chapter 4

API Onboarding and REST Interface Access

The API is embedded in the SEL-5056 service and is accessible while the SEL-5056 service is running at the address defined in the Fully Qualified Domain Name setting. See the SEL-5056 instruction manual for further information on configuring this setting.

4.1 API Onboarding

The API is accessible at the URI defined in the settings of the SEL-5056. By default this value is localhost with a TCP port of 443 so the base URI is https://localhost/. You may change both values using the SEL-5056 system settings.

To access the API to make OData or SignalR calls, only a token is required. This in turn requires credentials covered here.

4.1.1 API Discovery

OData exposes the metadata of the REST API on the REST interface that provides a list of endpoints and their properties. See Service Roots for more information.

4.1.2 API Access Prerequisites

There is no key, license, or account required to access the REST interface.

4.1.3 API Applications

If you wish to create an external application using the lient credentials or authorization code grants], you first need to understand OAuth token management and then register an application Application Management.

4.2 Accessing the REST Interface

The SEL-5056 only supports HTTPS. Before a user, application, or script can make REST calls (to create a logical connection for example), they must first obtain an OAuth access token.

The overall process is:

- Obtain credentials with the appropriate permissions or use the Implicit grant type with the appropriate role
- Obtain an OAuth token using a supported grant type
- Make REST calls
- (Optional) Revoke the token (i.e. log out)

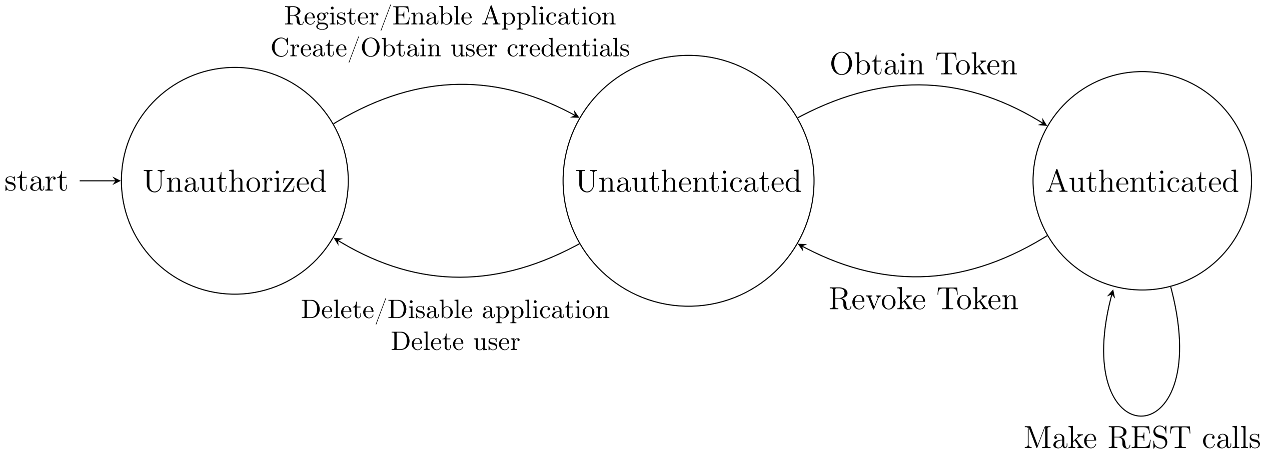
There are two paths to obtaining credentials:

- Register an application
- Add a user or obtain the credentials of a current user

The implicit grant type does not require user credentials.

After credentials are obtained, a token can be requested from the SEL-5056 using a supported OAuth type. Token management is handled by OAuth/ID Connect. After you have a token, you may now make REST calls using the token assuming the token has the appropriate permission(s) for the endpoint(s).

Following is a summary of the process:



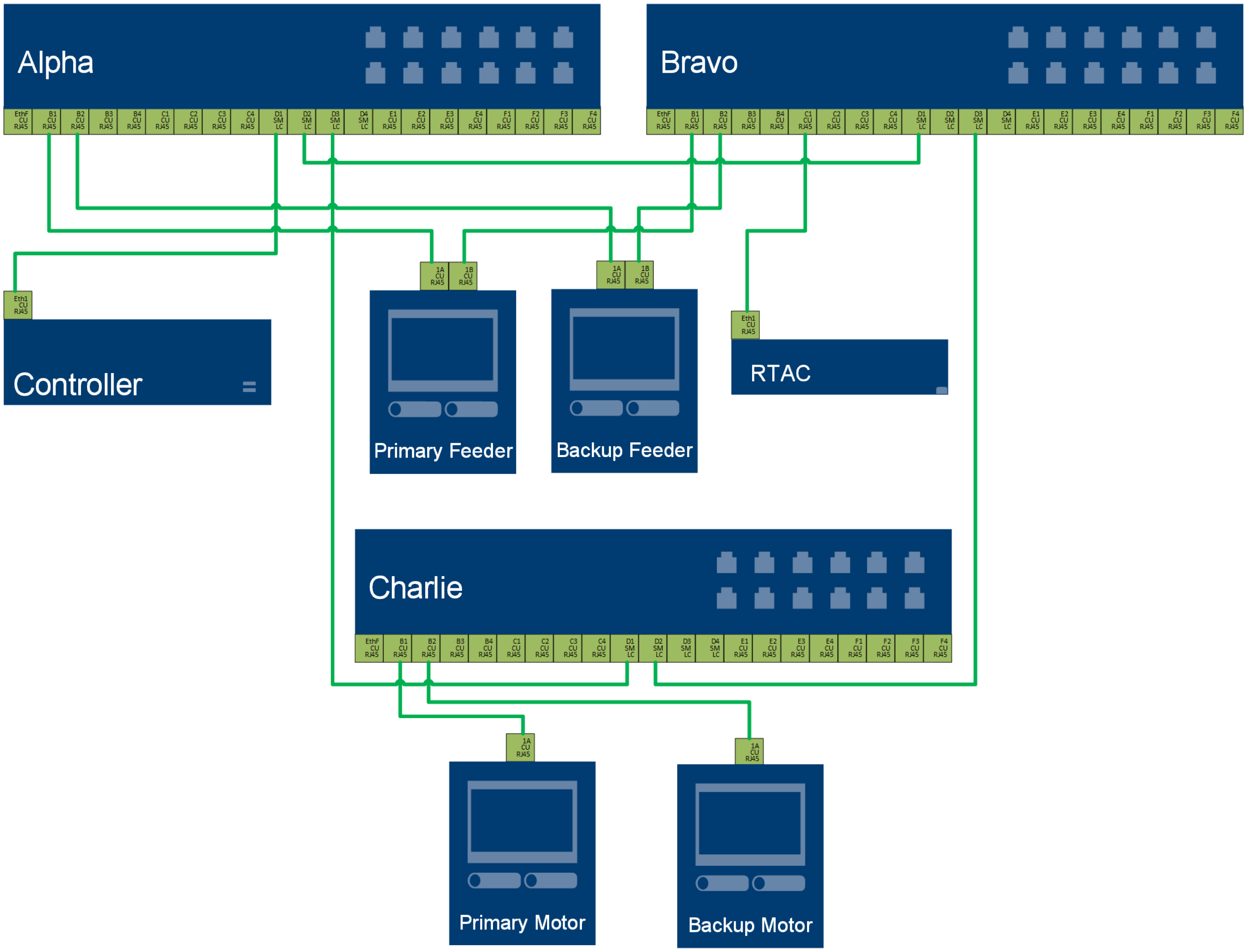
Chapter 5

Tutorial

See the conventions section for further information about the format of HTTP messages in this section.

This section covers making several common SEL-5056 REST call such as those to setup the network in figure 5.1.

Figure 5.1: Tutorial Network Topology



5.1 Commissioning the SEL-5056

We are going to start by commissioning the database with an initial user with the Security Administrator role.

Request

Key	Value
Endpoint	/api/default/security/Commission
Method	POST
Content-Type	application/json

Table 5.1: Header values for /api/default/security/Commission

We need 1 properties for the json body.

Property	Value

commissioningList	`[`
	`  {`
	`    "@odata.type": "#Sel.BlueFrame.CommissioningManager.UserCommissioner.UserCommissioningConcern",`
	`    "username": "admin",`
	`    "password": "Example1!",`
	`  }`
	`]`

Table 5.2: Body values for Commissioning the SEL-5056

```
POST /api/default/security/Commission HTTP/1.1
Content-Type: application/json
HOST: localhost
Content-Length: 167

{
  "commissioningList":   [
    {
      "@odata.type": "#Sel.BlueFrame.CommissioningManager.UserCommissioner.UserCommissioningConcern",
      "username": "admin",
      "password": "Example1!",
    }
  ]
}
```

Reply

```
HTTP/1.1 200 OK
Content-Length:  0
```

Successful response code is 200.

5.2 Using the OData Schema, Part 1

The SEL-5056 contains endpoint metadata information so information about endpoints can be obtained directly from the REST interface. In this section, we cover looking at the REST interface to find out what is available. Information on OData service roots and SEL-5056 version number can be obtained from the config.json endpoint.

Request

Key	Value
Endpoint	/configuration/settings.json
Method	GET
Content-Type	None

Table 5.3: Header values for /configuration/settings.json

```
GET /configuration/settings.json HTTP/1.1
HOST: localhost
```

Reply

```
HTTP/1.1 200 OK
Content-Type:  application/json
Content-Length:  326

{
  "webUri":  "https://localhost",
  "controllerVersion":  "2.2.0",
  "buildDate":  "2020-07-29T16:09:05.000-06:00",
  "trees":    [
    "api/default/settings",
    "api/default/security",
    "api/default/config",
    "api/default/operational",
    "api/default/certificate",
  ]
  "systemCommissioned":  "True",
  "remoteIp":  "127.0.0.1",
  "currentTime":  "2020-09-18T14:49:31.418-06:00",
}
```

We can confirm both the SEL-5056 version and build dates. The current system time is also included. The second part are the trees, in the trees properties.

This allows you to find all the roots that contain all of the endpoints outside of SignalR and OAuth.

Tree	Description
api/default/settings	Other settings
api/default/security	User management, database backup/restore
api/default/config	Objects representing settings not bellowing to specific physical devices
api/default/operational	Objects representing physical and virtual devices
api/default/certificate	X.509 certificate management

Going to any of these endpoints will show the futher endpoints that are available. For example, if we to go /api/default/config, then we will get the following JSON.

```
{
  "@odata.context": "https://localhost/api/default/config/$metadata",
  "value": [
    {
      "name": "nodes",
      "kind": "EntitySet",
      "url": "nodes"
    },
    {
      "name": "ports",
      "kind": "EntitySet",
      "url": "ports"
    },
    {
      "name": "links",
      "kind": "EntitySet",
      "url": "links"
    },
    {
      "name": "flows",
      "kind": "EntitySet",
      "url": "flows"
    },
    {
      "name": "groups",
      "kind": "EntitySet",
      "url": "groups"
    },
    {
      "name": "meters",
      "kind": "EntitySet",
      "url": "meters"
    },
    {
      "name": "logicalConnections",
      "kind": "EntitySet",
      "url": "logicalConnections"
    },
    {
      "name": "communicationServiceTypes",
      "kind": "EntitySet",
      "url": "communicationServiceTypes"
    },
    {
      "name": "vlanVidReservation",
      "kind": "EntitySet",
      "url": "vlanVidReservation"
    },
    {
      "name": "transactions",
      "kind": "EntitySet",
      "url": "transactions"
    }
  ]
}
```

We can see that for logical connections, the endpoint is listed as logicalConnections, the full url being /api/default/config/logicalConnections. This is the endpoint that we use to GET or POST logical connections.

To commission the SEL-5056 based the OData information obtained from the REST API, let's look at the /api/default/security endpoint. How each function is mapped to an endpoint is organized, but not expressed within the REST interface. The /api/default/security endpoint returns:

```
{
  "@odata.context": "https://localhost/api/default/security/$metadata",
  "value": [
    {
      "name": "roles",
      "kind": "EntitySet",
      "url": "roles"
    },
    {
      "name": "users",
      "kind": "EntitySet",
      "url": "users"
    },
    {
      "name": "moduleRoles",
      "kind": "EntitySet",
      "url": "moduleRoles"
    },
    {
      "name": "authServices",
      "kind": "EntitySet",
      "url": "authServices"
    },
    {
      "name": "authServiceGroups",
      "kind": "EntitySet",
      "url": "authServiceGroups"
    },
    {
      "name": "applicationLinks",
      "kind": "EntitySet",
      "url": "applicationLinks"
    },
    {
      "name": "transactions",
      "kind": "EntitySet",
      "url": "transactions"
    }
  ]
}
```

The XML document containing the metadata that contains a description of the endpoints and objects available on this branch can be obtained by going to the @odata.context link, here: https://localhost/api/default/security/$metadata

At the bottom of the document is the element <Schema Namespace="Sel">. This defines the endpoints (Entity sets) and actions available on that tree (in this case /api/default/security).

One of the actions has the Name="Commission", which includes the endpoint we used above. The endpoints has one paramenter, commissionsingList of Type Collection(Sel.BlueFrame.CommissioningManager.CommissioningConcern). This tells us that this paramenter is a list (from Collection) and the type of the elements in the list are of type Sel.BlueFrame.CommissioningManager.CommissioningConcern.

If we search for Sel.BlueFrame.CommissioningManager.CommissioningConcern in the XML document, we see the following XML:

```
<Schema Namespace="Sel.BlueFrame.CommissioningManager.UserCommissioner">
   <ComplexType Name="UserCommissioningConcern"
      BaseType="Sel.BlueFrame.CommissioningManager.CommissioningConcern">
      <Property Name="username" Type="Edm.String"
         CharacterSet="Printable7BitAscii" MaxLength="128"
         MinLength="1" Nullable="false"/>
      <Property Name="password" Type="Edm.String" CharacterSet="Printable7BitAscii"
         MaxLength="128" MinLength="8"
         RegularExpression="^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[^A-Z|^a-z|^0-9])"/>
    </ComplexType>
</Schema>
```

From this we can obtained the odata.type and the parameters used in the body of the request. The odata.type is Sel.BlueFrame.CommissioningManager.UserCommissioner.UserCommissioningConcern because the containing namespace is Sel.BlueFrame.CommissioningManager.UserCommissioner and the Name of the ComplexType we use if UserCommissionsingConcern. A # character is placed before the odata.type value and a @ before odata.type. The two properties are username and password. From this we can predict that the HTTP body would be:

```
{"commissioningList": [{"@odata.type":
 "#Sel.BlueFrame.CommissioningManager.UserCommissioner.UserCommissioningConcern",
"username": X, "password": Y}]}
```

, where X and Y are the values for username and password, respectively.

Now that the database is commissioned, we can add in a PL3 user. But first we need to log in using the commissioned user. We have the username and password from above, now we need the user role because permissions are required. We can see from the Add users endpoint that the required role is "SecurityAdministrator".

5.3 Logging in with a username and password

Logging in means getting an IDserver token. This token gives you permissions to do a set of actions. Permissions are covered later in this tutorial. You obtain a token using credentials and a grant type. For now, we are only concerned about the password grant type. This is where you send a username and password and obtain a token. Other OAuth grant types are covered later. Note that OAuth endpoints are not a part of the OData endpoints.

Request

Key	Value
Endpoint	/identity/connect/token
Method	POST
Content-Type	application/x-www-form-urlencoded

Table 5.4: Header values for /identity/connect/token

We need 7 properties for the url encoded body.

Property	Value
grant_type	password
username	admin
password	Example1!
acr_values	role:SecurityAdministrator
client_secret	Rest Interface
scope	Rest
client_id	password-client
state	0

Table 5.5: Body values for Logging in with a username and password

```
POST /identity/connect/token HTTP/1.1
HOST: localhost
Content-Type:  application/x-www-form-urlencoded
Content-Length:  171

grant_type=password&
username=admin&
password=Example1%21&
acr_values=role%3ASecurityAdministrator&
client_secret=Rest%20Interface&
scope=Rest&
client_id=password-client&
state=0
```

Reply

```
HTTP/1.1 200 OK
Content-Type:  application/json
Content-Length:  137

{
  "access_token":  "000...000",
  "expires_in":  "900",
  "token_type":  "Bearer",
  "scope":  "Rest",
}
```

In the response is the property access_token. For any request that requires the role Security Administrator, this token must be passed. We will see this in the next call. The token will expire in 900 seconds. To extend the duration of the token, we can user the Heartbeat endpoint.

5.4 Refreshing the token

Tokens are refreshed (i.e. the expiration date is set 900 seconds) each time you use it. You may use the Heartbeat endpoint to keep the token valid without making another REST call.

Request

Key	Value
Endpoint	/api/default/settings/Heartbeat
Method	POST
Content-Type	None

Table 5.6: Header values for /api/default/settings/Heartbeat

```
POST /api/default/settings/Heartbeat HTTP/1.1
HOST: localhost
Authorization:  Bearer 000...000
Content-Length:  0
```

Reply

```
HTTP/1.1 200 OK
Content-Length:  0
```

If successful, the response code is 200.

5.5 Validating a token

How do we know that a token is valid? We go to the token validation endpoint. Here the token must be passed as a URL parameter not in a body.

Request

Key	Value
Endpoint	/identity/connect/accesstokenvalidation
Method	POST
Content-Type	None

Table 5.7: Header values for /identity/connect/accesstokenvalidation

Key	Value
token	000...000

Table 5.8: Parameter values for /identity/connect/accesstokenvalidation

```
POST /identity/connect/accesstokenvalidation?token=000...000 HTTP/1.1
HOST: localhost
Content-Length:  0
```

Reply

```
HTTP/1.1 200 OK
Content-Type:  application/json
Content-Length:  308

{
  "aud":    [
    "https://localhost/identity/resources",
    "Rest",
  ]
  "iss":  "https://localhost/identity",
  "nbf":  "1600563952",
  "exp":  "1600564852",
  "client_id":  "password-client",
  "scope":    [
    "Rest",
  ]
  "sub":  "admin",
  "name":  "admin",
  "amr":  "local",
  "idp":  "local",
  "auth_time":  "1600563952",
  "role":  "SecurityAdministrator",
  "ipAddress":  "127.0.0.1",
}
```

What does all this mean? For us, we only care about the exp date. This is in number of seconds since epoch, which is January 1, 1970 at 00:00:00 (UTC). As mentioned above, the current system time is available the endpoint /configuration/settings.json.
If the token is not valid, the response is code 400.

5.6 Logging Out

The final token related action is revocation, i.e. logging out.

Request

Key	Value
Endpoint	/identity/connect/revocation
Method	POST
Content-Type	application/x-www-form-urlencoded

Table 5.9: Header values for /identity/connect/revocation

We need 3 properties for the url encoded body.

Property	Value
token	9fa7823d61ca099abf7f0e9ea53ae6bfeae6dafa31bff6de0a9983c8816e6adf
token_type_hint	access_token
client_id	password-client
client_secret	Rest Interface

Table 5.10: Body values for Logging Out

```
POST /identity/connect/revocation HTTP/1.1
HOST: localhost
Content-Type:  application/x-www-form-urlencoded
Authorization:  Bearer 000...000
Content-Length:  156

token=9fa7823d61ca099abf7f0e9ea53ae6bfeae6dafa31bff6de0a9983c8816e6adf&
token_type_hint=access_token&
client_id=password-client&
client_secret=Rest%20Interface
```

Reply

```
HTTP/1.1 200 OK
Content-Length:  0
```

If we check the status of the token, we see that it is expired because the response is HTTP code 400.

Request

Reply

5.7 Creating a Permission Level 3 user

First we log back in with our SecurityAdministrator user

Request

Reply

Now we can create a user with permission level 3 (PermissionLevel3). We will also give this user the monitor (Monitor) role.

Request

Key	Value
Endpoint	/api/default/security/users

Method	POST
Content-Type	application/json

Table 5.11: Header values for /api/default/security/users

We need 4 properties for the json body.

Property	Value
@odata.type	`"#Sel.BlueFrame.Core.SecurityManager.LocalUser",`
roles	`[ "PermissionLevel3", "Monitor", ]`
username	`"pl3",`
clearTextPass	`"Example2!",`

Table 5.12: Body values for Creating a Permission Level 3 user

```
POST /api/default/security/users HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 155

{
  "@odata.type": "#Sel.BlueFrame.Core.SecurityManager.LocalUser",
  "roles":   [
    "PermissionLevel3",
    "Monitor",
  ]
  "username": "pl3",
  "clearTextPass": "Example2!",
}
```

Reply

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 543

{
  "@odata.context": "https://localhost/api/default/security/$metadata#users-
/Sel.BlueFrame.Core.SecurityManager.LocalUser/$entity",
  "@odata.type": "#Sel.BlueFrame.Core.SecurityManager.LocalUser",
  "id": "a6fdcb7300a224a9b814cb43190ed581",
  "displayName": "pl3",
  "enabled": "True",
  "isLockedOut": "False",
  "lastSuccessfulLogin": "0001-01-01T00:00:00-07:00",
  "lastSuccessLoginIp": "None",
  "lockedOutUntil": "0001-01-01T00:00:00-07:00",
  "lockoutTime": "0001-01-01T00:00:00-07:00",
  "roles":   [
    "PermissionLevel3",
    "Monitor",
  ]
  "username": "pl3",
  "clearTextPass": "None",
  "failedLogins":   [
  ]
}
```

For this POST call, we receive the new user definition back.

5.8 Finding the Permission Level 3 user

We have received the created user back with the POST, but what if we would like to retrieve the user at a later time, for example, to see if the user is locked out so that is why he is unable to log back in.

As we posted to the /api/default/security/users endpoint, we can GET the /api/default/security/users endpoint. There are two types of GET, individual GET (/api/default/security/users(id')) or a GET all (/api/default/security/users/). To use the individual GET, we need to know the id (the value of the id property). We don't have the id, so the other option is to get the complete list back and then iterate to find the user with the username pl3.

Another method is to use OData filter to retrieve the user using the value of an attribute, in this case the displayName (or username).

Request

Key	Value

Endpoint	/api/default/security/users
Method	GET
Content-Type	None

Table 5.13: Header values for /api/default/security/users

Key	Value
$filter	displayName eq 'pl3'

Table 5.14: Parameter values for /api/default/security/users

```
GET /api/default/security/users?$filter=displayName%20eq%20'pl3' HTTP/1.1
HOST: localhost
Authorization:  Bearer 000...000
```

Reply

```
HTTP/1.1 200 OK
Content-Type:  application/json
Content-Length:  502

{
  "@odata.context":  "https://localhost/api/default/security/$metadata-
#users",
  "value":    [
    {
      "@odata.type":  "#Sel.BlueFrame.Core.SecurityManager.LocalUser",
      "id":  "a23cee85b1ebb4e328b0a1af7997e47c",
      "displayName":  "pl3",
      "enabled":  "True",
      "isLockedOut":  "False",
      "lastSuccessfulLogin":  "0001-01-01T00:00:00-07:00",
      "lastSuccessLoginIp":  "None",
      "lockedOutUntil":  "0001-01-01T00:00:00-07:00",
      "lockoutTime":  "0001-01-01T00:00:00-07:00",
      "roles":        [
        "PermissionLevel3",
        "Monitor",
      ]
      "username":  "pl3",
      "clearTextPass":  "None",
      "failedLogins":        [
      ]
    }
  ]
}
```

5.9 Creating Configuration Nodes

With the permission level 3 user created, we can now log in, using the role PermissionLevel3

Request

Key	Value
Endpoint	/identity/connect/token
Method	POST
Content-Type	application/x-www-form-urlencoded

Table 5.15: Header values for /identity/connect/token

We need 7 properties for the url encoded body.

Property	Value
grant_type	password
username	pl3
password	Example2!
acr_values	role:PermissionLevel3
client_secret	Rest Interface
scope	Rest
client_id	password-client
state	0

Table 5.16: Body values for Creating Configuration Nodes

```
POST /identity/connect/token HTTP/1.1
HOST: localhost
Content-Type:  application/x-www-form-urlencoded
Content-Length:  164

grant_type=password&
username=pl3&
password=Example2%21&
acr_values=role%3APermissionLevel3&
client_secret=Rest%20Interface&
scope=Rest&
client_id=password-client&
state=0
```

Reply

```
HTTP/1.1 200 OK
Content-Type:  application/json
Content-Length:  137

{
  "access_token":  "000...000",
  "expires_in":  "900",
  "token_type":  "Bearer",
  "scope":  "Rest",
}
```

First, we create a configuration nodes for the SEL-2740Ss. The minimum settings we need to provide are the IPv4 address of the switch, the subnet of the switch, the default gateway of the switch, and the IPv4 address of the Controller. The other settings will be covered later.

Request

Key	Value
Endpoint	/api/default/config/nodes
Method	POST
Content-Type	application/json

Table 5.17: Header values for /api/default/config/nodes

We need 10 properties for the json body.

Property	Value
@odata.type	`"#Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode",`
displayName	`"Alpha",`
controllerIp	`"192.168.1.1",`
defaultGateway	`"192.168.1.1",`
ipAddress	`"192.168.1.100",`
subnetMask	`"255.255.0.0",`
alarmMinimumDuration	`"1",`
enablePtp	`"False",`
enableSnmp	`"False",`
ntpServers	`[                   ]`

Table 5.18: Body values for Creating Configuration Nodes

```
POST /api/default/config/nodes HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 305

{
  "@odata.type": "#Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode",
  "displayName": "Alpha",
  "controllerIp": "192.168.1.1",
  "defaultGateway": "192.168.1.1",
  "ipAddress": "192.168.1.100",
  "subnetMask": "255.255.0.0",
  "alarmMinimumDuration": "1",
  "enablePtp": "False",
  "enableSnmp": "False",
  "ntpServers":   [
  ]
}
```

Reply

```
HTTP/1.1 201 Created
Content-Type:  application/json
Content-Length:  1440

{
  "@odata.context":  "https://localhost/api/default/config/$metadata#nodes/Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode/$entity",
  "@odata.type":  "#Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode",
  "id":  "aa3ebe3914748486d9b924311c5d05b3",
  "state":  "Configured",
  "displayName":  "Alpha",
  "linkedKey":  "None",
  "ports":    [
  ]
  "additionalTrustedCertificates":    [
  ]
  "alarmMinimumDuration":  "1",
  "controllerIp":  "192.168.1.1",
  "defaultGateway":  "192.168.1.1",
  "enablePtp":  "False",
  "enableSnmp":  "False",
  "ipAddress":  "192.168.1.100",
  "ntpServers":    [
  ]
  "subnetMask":  "255.255.0.0",
  "configurations":    [
  ]
  "tags":    [
  ]
  "eventCategories":    [
    {
      "key":  "default",
      "id":  "default",
      "behaviors":        [
        {
          "@odata.type":  "#Sel.BlueFrame.Particle.AlarmContact.AlarmContactBehavior",
          "name":  "default",
          "severity":  "Error",
        }
        {
          "@odata.type":  "#Sel.BlueFrame.Particle.LocalEventStore.LocalEventStoreBehavior",
          "name":  "default",
          "severity":  "Informational",
        }
      ]
    }
    {
      "key":  "Openflow",
      "id":  "Openflow",
      "behaviors":        [
        {
          "@odata.type":  "#Sel.BlueFrame.Particle.AlarmContact.AlarmContact-
Behavior",
          "name":  "default",
          "severity":  "Notice",
        }
      ]
    }
    {
      "key":  "Link",
      "id":  "Link",
      "behaviors":        [
        {
          "@odata.type":  "#Sel.BlueFrame.Particle.AlarmContact.AlarmContact-
Behavior",
          "name":  "default",
          "severity":  "Notice",
        }
      ]
    }
    {
      "key":  "Security",
      "id":  "Security",
      "behaviors":        [
      ]
    }
    {
      "key":  "Configuration",
      "id":  "Configuration",
      "behaviors":        [
      ]
    }
    {
      "key":  "System Integrity",
      "id":  "System Integrity",
      "behaviors":        [
      ]
    }
    {
      "key":  "Chassis and Module",
      "id":  "Chassis and Module",
      "behaviors":        [
      ]
    }
  ]
}
```

As you can see in the response, the created object is also returned in the response along with additional parameters, both read-only and optional.

The 5 hosts are much simpler as only the displayName is needed.

1. Primary Feeder
2. Backup Feeder
3. Primary Motor
4. Backup Motor
5. RTAC

Here is the information for the endpoint for creating the first host.

Request

Key	Value
Endpoint	/api/default/config/nodes
Method	POST
Content-Type	application/json

Table 5.19: Header values for /api/default/config/nodes

We need 2 properties for the json body.

Property	Value
displayName	`"Primary Feeder",`
tags	`[     ]`

Table 5.20: Body values for Creating Configuration Nodes

```
POST /api/default/config/nodes HTTP/1.1
Content-Type:  application/json
HOST: localhost
Authorization:  Bearer 000...000
Content-Length:  45

{
  "displayName":  "Primary Feeder",
  "tags":    [
  ]
}
```

Reply

```
HTTP/1.1 201 Created
Content-Type:  application/json
Content-Length:  231

{
  "@odata.context":  "https://localhost/api/default/config/$metadata#nodes/-
$entity",
  "id":  "a061a1101d68b4d8bb4ac328c20e6e2f",
  "state":  "Configured",
  "displayName":  "Primary Feeder",
  "linkedKey":  "None",
  "ports":    [
  ]
  "configurations":    [
  ]
  "tags":    [
  ]
}
```

The remaining switches and hosts have the same form, but with different displayName values and ipAddress values.

5.10 Adopting Switches

Now that the configuration objects are setup, the next step is to adopt the switches and hosts with those configuration objects. There are two adopt functions: an adopt with an previously created configuration node and an adopt with an auto-generated configuration node.

We cannot adopt a switch using a default profile so we will be using the an adopt with an previously created configuration node endpoint. Therefore we need two parameters, the binding parameter, which is the operational id, and the configuration id.

When a SEL-2740S node appears in the topology, we need to match it against the configuration node. There are several ways to do this:

1. Match the datapath ID against the predicted one
2. Match the placement in the topology against the predicted one
3. Match based on logic such as there is only one switch so the one that shows up must be that one

Unlike hosts, SEL-2740S switches do not have the desired IP address at start up. For the first switch, we will assume that there is only one unadopted switch and that switch is Alpha.

First we look for unadopted switches. SEL2740S operational nodes have a SelSapphireAttr (Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr) attribute.

We can do this by GETing all of the other operational node endpoints and looking for the one with the SelSapphireAttr.

Request

Key	Value
Endpoint	/api/default/operational/nodes
Method	GET
Content-Type	None

Table 5.21: Header values for /api/default/operational/nodes

```
GET /api/default/operational/nodes HTTP/1.1
HOST: localhost
Authorization:  Bearer 000...000
```

Reply

```
HTTP/1.1 200 OK
Content-Type:  application/json
Content-Length:  3769

{
  "@odata.context":  "https://localhost/api/default/operational/$metadata-
#nodes",
  "value":    [
    {
      "id":  "a45a10ea7bdb3423eab55b254e969f3b",
      "displayName":  "Controller",
      "state":  "Adopted",
      "trustState":  "Internal",
      "linkedKey":  "a92d1ccca0339497097b883ca263659a",
      "ports":       [
        "a1fd97928e0aa4cc9b3dd90a69f6b753",
        "aa25d2dcfb3ea4affa06ce149bc7d02d",
        "ab33d6dc702794cdaadcd23fa57da6e3",
        "a629e33b59de94f6c9fd40a34479f509",
        "a8ba4dcd1070e4f2daea8aeaba403a41",
        "ac7dce79daf6e4beaad894e60ac13677",
        "ae0f59eceac5346e7ad25ad8312e780e",
      ]
      "attributes":       [
        {
          "@odata.type":  "#Sel.Sel5056.TopologyManager.Attributes.-
Operational.Node.IpAttr",
          "ipAddress":  "169.254.168.147",
        }
        {
          "@odata.type":  "#Sel.Sel5056.TopologyManager.Attributes.-
Operational.Node.IpAttr",
          "ipAddress":  "169.254.124.180",
        }
        {
          "@odata.type":  "#Sel.Sel5056.TopologyManager.Attributes.-
Operational.EthernetAttr",
          "macAddress":  "B831B5967D39",
          "id":  "B831B5967D39",
        }
        {
          "@odata.type":  "#Sel.Sel5056.TopologyManager.Attributes.-
Operational.EthernetAttr",
          "macAddress":  "0A0027000008",
          "id":  "0A0027000008",
        }
        {
          "@odata.type":  "#Sel.Sel5056.TopologyManager.Attributes.-
Operational.Node.IpAttr",
          "ipAddress":  "192.168.1.1",
        }
```

5.11 Adopting Links

Adopting switches automatically adopted some of the links. There is still the link between Bravo and Charlie that is unadopted. Because we can adopt a link with a default profile and we don't have any need to create a profile before hand, we will use the an adopt with an auto-generated configuration node endpoint. However, first we need to get the id of the operational link to be adopt to adopt it. There are several methods for obtaining this, here are two possibilities.

1. Start at Bravo or Charlie and then find port D2 and then get the object id of the attached link
2. Retrieve all of the links and then find if the attached ports are D2 on Bravo or Charlie

We will do the first. We will use OData filtering to obtain the port in one query and then use the first member of the attachedLinked array to obtain the link id that we need.

Request

Key	Value
Endpoint	/api/default/operational/ports
Method	GET
Content-Type	None

Table 5.22: Header values for /api/default/operational/ports

Key	Value
$filter	attributes/any(t: isof(t, 'Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr') and t/Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr/prettyDataPathId eq '00000030A71412F4' and t/Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr/portId eq 10)

Table 5.23: Parameter values for /api/default/operational/ports

```
GET /api/default/operational/ports?$filter=attributes/any(t%3A-
%20isof(t%2C%20'Sel.Sel5056.TopologyManager.Attributes.Operational-
.Port.OpenFlowPortAttr')%20and%20t/Sel.Sel5056.TopologyManager-
.Attributes.Operational.Port.OpenFlowPortAttr/prettyDataPathId%20eq%20-
'00000030A71412F4'%20and%20t/Sel.Sel5056.TopologyManager.Attributes-
.Operational.Port.OpenFlowPortAttr/portId%20eq%2010) HTTP/1.1
HOST: localhost
Authorization: Bearer 000...000
```

Reply

```
HTTP/1.1 200 OK
Content-Type:  application/json
Content-Length:  2317

{
  "@odata.context":  "https://localhost/api/default/operational/$metadata-
#ports",
  "value":    [
    {
      "id":  "a1b5e44b827704978802a4f8548f986e",
      "displayName":  "OpenFlow:00000030A71412F4:D2(10)",
      "isConnected":  "True",
      "state":  "Adopted",
      "trustState":  "Internal",
      "linkedKey":  "ab668532f729748dab0a7071062fa82b",
      "parentNode":  "a3a6598a296e24ce5978714d934b717e",
      "attachedLinks":        [
        "abf3efa8ff3a54a0c8343c20d88712ed",
      ]
      "attributes":        [
        {
          "@odata.type":  "#Sel.Sel5056.TopologyManager.Attributes.-
Operational.Port.Sel2740SPortAttr"
        }
        {
          "@odata.type":  "#Sel.Sel5056.TopologyManager.Attributes.-
Operational.Port.OpenFlowPortAttr",
          "portId":  "10",
          "dataPathId":  "208961540852",
          "prettyDataPathId":  "00000030A71412F4",
          "currentFeatures":  "8224",
          "ofState":  "Live",
          "currentSpeed":  "1000000",
          "maxSpeed":  "1000000",
          "name":  "ethd-2",
          "hardwareAddress":  "0030A71412F4",
          "peer":  "0",
          "id":  "00000030A71412F4:10",
          "advertised":            [
            {
              "@odata.type":  "#Sel.Sel5056.TopologyManager.OpenFlow.-
PortMod.FeatureFlagObjects.AutoNegotiation"
            }
            {
              "@odata.type":  "#Sel.Sel5056.TopologyManager.OpenFlow.-
PortMod.FeatureFlagObjects.Copper"
            }
            {
              "@odata.type":  "#Sel.Sel5056.TopologyManager.OpenFlow.-
PortMod.FeatureFlagObjects.OneGigaBitFullDuplex"
            }
```

Now we can adopt the link

Request

Key	Value
Endpoint	/api/default/operational/links
Object id	abf3efa8ff3a54a0c8343c20d88712ed
Action	Adopt
Method	POST
Content-Type	None

Table 5.24: Header values for /api/default/operational/links

```
POST /api/default/operational/links('abf3efa8ff3a54a0c8343c20d88712ed')-
/Adopt HTTP/1.1
HOST: localhost
Authorization:  Bearer 000...000
Content-Length:  0
```

Reply

```
HTTP/1.1 200 OK
Content-Length:  0
```

Now the switches and all of the links between switches are adopted. The next step is to adopt the hosts

5.12 Adopting the Hosts, including failover mode

Request

Reply

Request

Key	Value
Endpoint	/api/default/config/nodes
Method	GET
Content-Type	None

Table 5.25: Header values for /api/default/config/nodes

Key	Value
$filter	displayName eq 'Primary Feeder'

Table 5.26: Parameter values for /api/default/config/nodes

```
GET /api/default/config/nodes?$filter=displayName%20eq%20'Primary%20Feeder'
HTTP/1.1
HOST: localhost
Authorization:  Bearer 000...000
```

Reply

```
HTTP/1.1 200 OK
Content-Type:  application/json
Content-Length:  269

{
  "@odata.context":  "https://localhost/api/default/config/$metadata#nodes",
  "value":    [
    {
      "id":  "a061a1101d68b4d8bb4ac328c20e6e2f",
      "state":  "Configured",
      "displayName":  "Primary Feeder",
      "linkedKey":  "None",
      "ports":        [
        "ac596af02e66d4ce69aa8699b80c5457",
      ]
      "configurations":        [
      ]
      "tags":        [
      ]
    }
  ]
}
```

Request

Key	Value
Endpoint	/api/default/operational/nodes
Object id	a15d4fe3d77e64b61afdf87d43fc500c
Action	AdoptWithConfig
Method	POST
Content-Type	application/json

Table 5.27: Header values for /api/default/operational/nodes

We need 1 properties for the json body.

Property	Value
configKey	`"a061a1101d68b4d8bb4ac328c20e6e2f",`

Table 5.28: Body values for Adopting the Hosts, including failover mode

```
POST /api/default/operational/nodes(’a15d4fe3d77e64b61afdf87d43fc500c’)-
/AdoptWithConfig HTTP/1.1
Content-Type:  application/json
HOST: localhost
Authorization:  Bearer 000...000
Content-Length:  49

{
  "configKey":  "a061a1101d68b4d8bb4ac328c20e6e2f",
}
```

Reply

```
HTTP/1.1 200 OK
Content-Length:  0
```

Request

Key	Value
Endpoint	/api/default/operational/ports
Object id	a2d01b7ddceaf4be38efc9c39fc31e60
Method	GET
Content-Type	None

Table 5.29: Header values for /api/default/operational/ports

```
GET /api/default/operational/ports(’a2d01b7ddceaf4be38efc9c39fc31e60’) HTTP-
/1.1
HOST: localhost
Authorization:  Bearer 000...000
```

Reply

```
HTTP/1.1 200 OK
Content-Type:  application/json
Content-Length:  835

{
  "@odata.context":  "https://localhost/api/default/operational/$metadata-
#ports/$entity",
  "id":  "a2d01b7ddceaf4be38efc9c39fc31e60",
  "displayName":  "IP:192.168.1.150",
  "isConnected":  "True",
  "state":  "Unadopted",
  "trustState":  "Untrusted",
  "linkedKey":  "None",
  "parentNode":  "a15d4fe3d77e64b61afdf87d43fc500c",
  "attachedLinks":    [
    "a72c90da635c94a7a86699cc51168ab9",
  ]
  "attributes":    [
    {
      "@odata.type":  "#Sel.Sel5056.TopologyManager.Attributes.Operational.-
Node.IpAttr",
      "ipAddress":  "192.168.1.150",
    }
    {
      "@odata.type":  "#Sel.Sel5056.TopologyManager.Attributes.Operational.-
Port.EthernetApplicationAttr",
      "ethType":  "35000",
      "macAddress":  "0030A703D8AB",
      "id":  "35000:0030A703D8AB",
      "providerInformation":        [
        {
          "vlan":  "901",
          "destinationMac":  "010CCD010001",
        }
      ]
    }
    {
      "@odata.type":  "#Sel.Sel5056.TopologyManager.Attributes.Operational.-
EthernetAttr",
      "macAddress":  "0030A703D8AA",
      "id":  "0030A703D8AA",
    }
  ]
}
```

Request

Key	Value
Endpoint	/api/default/operational/links
Object id	a72c90da635c94a7a86699cc51168ab9
Action	Adopt
Method	POST
Content-Type	None

Table 5.30: Header values for /api/default/operational/links

```
POST /api/default/operational/links('a72c90da635c94a7a86699cc51168ab9')-
/Adopt HTTP/1.1
HOST: localhost
Authorization:  Bearer 000...000
Content-Length:  0
```

Reply

```
HTTP/1.1 200 OK
Content-Length:  0
```

Request

Key	Value
Endpoint	/api/default/operational/ports
Object id	a2d01b7ddceaf4be38efc9c39fc31e60
Action	MarkPortAsSelRelayFailover
Method	POST
Content-Type	None

Table 5.31: Header values for /api/default/operational/ports

```
POST /api/default/operational/ports('a2d01b7ddceaf4be38efc9c39fc31e60')-
/MarkPortAsSelRelayFailover HTTP/1.1
HOST: localhost
Authorization:  Bearer 000...000
Content-Length:  0
```

Reply

```
HTTP/1.1 200 OK
Content-Length:  0
```

Request

Key	Value
Endpoint	/api/default/operational/ports
Object id	a2d01b7ddceaf4be38efc9c39fc31e60
Action	DetectSelRelayFailoverPort
Method	POST
Content-Type	None

Table 5.32: Header values for /api/default/operational/ports

```
POST /api/default/operational/ports('a2d01b7ddceaf4be38efc9c39fc31e60')-
/DetectSelRelayFailoverPort HTTP/1.1
HOST: localhost
Authorization:  Bearer 000...000
Content-Length:  0
```

Reply

```
HTTP/1.1 200 OK
Content-Length:  0
```

The Backup Feeder is the same. The other hosts can skip the failover steps. Now the network is completely adopted.

5.13 Creating Logical Conenctions

Now that the network is completely adopted, we can create the logical connections (LC). We first start with the NTP LC between the primary feeder and RTAC. Because this is a unicast IP LC, we also need to program the ARP. Like the other calls, the ids of the nodes are required. Finding those ids are not shown here.

Request

Key	Value
Endpoint	/api/default/config/logicalConnections
Method	POST
Content-Type	application/json

Table 5.33: Header values for /api/default/config/logicalConnections

We need 3 properties for the json body.

Property	Value

sourceEndPoints	`[  "a061a1101d68b4d8bb4ac328c20e6e2f",]`
destinationEndPoints	`[  "a04fcab0939ea41b4af211261bb9283f",]`
communicationServiceTypeId	`"NTP Client",`

Table 5.34: Body values for Creating Logical Conenctions

```
POST /api/default/config/logicalConnections HTTP/1.1
Content-Type:  application/json
HOST: localhost
Authorization:  Bearer 000...000
Content-Length:  163

{
  "sourceEndPoints":    [
    "a061a1101d68b4d8bb4ac328c20e6e2f",
  ]
  "destinationEndPoints":    [
    "a04fcab0939ea41b4af211261bb9283f",
  ]
  "communicationServiceTypeId":  "NTP Client",
}
```

Reply

```
HTTP/1.1 201 Created
Content-Type:  application/json
Content-Length:  370

{
  "@odata.context":  "https://localhost/api/default/config/$metadata-
#logicalConnections/$entity",
  "id":  "ab1a21e1490d641cd9c6fde3689e305d",
  "sourceEndPoints":    [
    "a061a1101d68b4d8bb4ac328c20e6e2f",
  ]
  "destinationEndPoints":    [
    "a04fcab0939ea41b4af211261bb9283f",
  ]
  "communicationServiceTypeId":  "NTP Client",
  "connectionOrigin":  "EndUser",
  "errorState":  "InProgress",
  "errors":    [
  ]
  "version":  "1",
}
```

Request

Key	Value
Endpoint	/api/default/config/logicalConnections
Method	POST
Content-Type	application/json

Table 5.35: Header values for /api/default/config/logicalConnections

We need 3 properties for the json body.

Property	Value
sourceEndPoints	`[ "a061a1101d68b4d8bb4ac328c20e6e2f",]`
destinationEndPoints	`[ "a04fcab0939ea41b4af211261bb9283f",]`
communicationServiceTypeId	`"Bidirectional ARP",`

Table 5.36: Body values for Creating Logical Conenctions

```
POST /api/default/config/logicalConnections HTTP/1.1
Content-Type:  application/json
HOST: localhost
Authorization:  Bearer 000...000
Content-Length:  170

{
  "sourceEndPoints":    [
    "a061a1101d68b4d8bb4ac328c20e6e2f",
  ]
  "destinationEndPoints":    [
    "a04fcab0939ea41b4af211261bb9283f",
  ]
  "communicationServiceTypeId":  "Bidirectional ARP",
}
```

Reply

```
HTTP/1.1 201 Created
Content-Type:  application/json
Content-Length:  377

{
  "@odata.context":  "https://localhost/api/default/config/$metadata-
#logicalConnections/$entity",
  "id":  "a6765f2f96305414db922ffdf490e5ad",
  "sourceEndPoints":    [
    "a061a1101d68b4d8bb4ac328c20e6e2f",
  ]
  "destinationEndPoints":    [
    "a04fcab0939ea41b4af211261bb9283f",
  ]
  "communicationServiceTypeId":  "Bidirectional ARP",
  "connectionOrigin":  "EndUser",
  "errorState":  "InProgress",
  "errors":    [
  ]
  "version":  "1",
}
```

Other IP LCs are handled the same way

5.14 Creating CSTs

For the GOOSE LCs, we need to create our own CSTs.

Request

Key	Value
Endpoint	/api/default/config/communicationServiceTypes
Method	POST
Content-Type	application/json

Table 5.37: Header values for /api/default/config/communicationServiceTypes

We need 3 properties for the json body.

Property	Value
cstCommunicationType	`"Multicast",`
displayName	`"GOOSE 901",`

trafficMatch	`{`
	`  "matchFields":    [`
	`    {`
	`      "@odata.type":  "Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.VlanVid",`
	`      "value":  "901",`
	`    }`
	`    {`
	`      "@odata.type":  "Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthType",`
	`      "value":  "GOOSE",`
	`    }`
	`  ]`
	`}`

Table 5.38: Body values for Creating CSTs

```
POST /api/default/config/communicationServiceTypes HTTP/1.1
Content-Type:  application/json
HOST: localhost
Authorization:  Bearer 000...000
Content-Length:  300

{
  "cstCommunicationType":  "Multicast",
  "displayName":  "GOOSE 901",
  "trafficMatch":    {
    "matchFields":      [
      {
        "@odata.type":  "Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
MatchFields.VlanVid",
        "value":  "901",
      }
      {
        "@odata.type":  "Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
MatchFields.EthType",
        "value":  "GOOSE",
      }
    ]
  }
}
```

Reply

```
HTTP/1.1 201 Created
Content-Type:  application/json
Content-Length:  596

{
  "@odata.context":  "https://localhost/api/default/config/$metadata-
#communicationServiceTypes/$entity",
  "id":  "a688d8322fed545738c235ad6167ec26",
  "priority":  "2000",
  "numFailableNodes":  "0",
  "numFailableLinks":  "0",
  "cstCommunicationType":  "Multicast",
  "errorState":  "Success",
  "errors":    [
  ]
  "displayName":  "GOOSE 901",
  "version":  "1",
  "trafficMatch":    {
    "matchFields":      [
      {
        "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
MatchFields.VlanVid",
        "value":  "901",
        "mask":  "None",
      }
      {
        "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
MatchFields.EthType",
        "value":  "GOOSE",
      }
    ]
  }
  "setQueue":  "None",
  "autoFillSettings":  "None",
  "tags":    [
  ]
}
```

5.15 Adding an offline host

Offline hosts are created in three steps. First the configuration node object is created with the appropriate IP and MAC addresses as needed. Second, obtain the operational id of the switch port the offline host will be attached to. Third, the AddSilentHost function is used to bind the configuration node to the switch port. We have already covered obtaining switch operational port ids, so this step will not be shown below, but assume that the operational id presented is for port C1(5).

Request

Key	Value
Endpoint	/api/default/config/nodes
Method	POST
Content-Type	application/json

Table 5.39: Header values for /api/default/config/nodes

We need 2 properties for the json body.

Property	Value
displayName	"Tap",

tags	[
	{
	"@odata.type": "#Sel.Sel5056.Tags.IpAddressTag",
	"ipAddress": "192.168.1.2",
	}
	]

Table 5.40: Body values for Adding an offline host

```
POST /api/default/config/nodes HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 111

{
  "displayName": "Tap",
  "tags":   [
    {
      "@odata.type": "#Sel.Sel5056.Tags.IpAddressTag",
      "ipAddress": "192.168.1.2",
    }
  ]
}
```

Reply

```
HTTP/1.1 201 Created
Content-Type:  application/json
Content-Length:  313

{
  "@odata.context":  "https://localhost/api/default/config/$metadata#nodes/-
$entity",
  "id":  "a3896e6b2849a4e748a3a6838adf982e",
  "state":  "Configured",
  "displayName":  "Tap",
  "linkedKey":  "None",
  "ports":    [
  ]
  "configurations":    [
  ]
  "tags":    [
    {
      "@odata.type":  "#Sel.Sel5056.Tags.IpAddressTag",
      "ipAddress":  "192.168.1.2",
      "id":  "192.168.1.2",
    }
  ]
}
```

5.16 Managing Flow Entries

Now that we have explored using the automation provided by LCs and CSTs, we will proceed with managing OpenFlow entries directly. First we add a simple entry to send table miss traffic from Alpha to the Tap device we added.

Request

Key	Value
Endpoint	/api/default/config/flows
Method	POST
Content-Type	application/json

Table 5.41: Header values for /api/default/config/flows

We need 5 properties for the json body.

Property	Value
displayName	`"Alpha Tap",`

node	`"aa3ebe3914748486d9b924311c5d05b3",`
priority	`"100",`
tableId	`"3",`

instructions	`[`
	`  {`
	`    "actions":      [`
	`      {`
	`        "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputAction",`
	`        "outPort":  "5",`
	`      }`
	`    ]`
	`    "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions"`
	`  }`
	`]`

Table 5.42: Body values for Managing Flow Entries

```
POST /api/default/config/flows HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 304

{
  "displayName": "Alpha Tap",
  "node": "aa3ebe3914748486d9b924311c5d05b3",
  "priority": "100",
  "tableId": "3",
  "instructions":    [
    {
      "actions":        [
        {
          "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputAction",
          "outPort": "5",
        }
      ]
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions"
    }
  ]
}
```

Reply

```
HTTP/1.1 201 Created
Content-Type:  application/json
Content-Length:  762

{
  "@odata.context":  "https://localhost/api/default/config/$metadata#flows/$entity",
  "id":  "a9e908a5ac88b45078f39d0088d723c3",
  "outPort":  "5",
  "outGroup":  "None",
  "inPort":  "None",
  "tableId":  "3",
  "bufferId":  "0",
  "hardTimeout":  "0",
  "idleTimeout":  "0",
  "noPacketCounts":  "False",
  "noByteCounts":  "False",
  "resetCounts":  "False",
  "checkOverlap":  "False",
  "priority":  "100",
  "lastEdit":  "2020-09-30T16:42:13.8464684-06:00",
  "errorState":  "InProgress",
  "errors":    [
  ]
  "enabled":  "True",
  "node":  "aa3ebe3914748486d9b924311c5d05b3",
  "cookie":  "1882",
  "displayName":  "Alpha Tap",
  "version":  "1",
  "match":    {
    "matchFields":      [
    ]
  }
  "instructions":    [
    {
      "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions",
      "actions":        [
        {
          "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputAction",
          "maxLength":  "65535",
          "outPort":  "5",
        }
      ]
    }
  ]
  "tags":    [
  ]
}
```

5.17 Managing Group Entries

Request

Key	Value
Endpoint	/api/default/config/groups
Method	POST
Content-Type	application/json

Table 5.43: Header values for /api/default/config/groups

We need 5 properties for the json body.

Property	Value
displayName	`"D2 then D3",`
node	`"aa3ebe3914748486d9b924311c5d05b3",`
groupId	`"0",`
groupType	`"FastFailover",`

buckets	`[`
	`  {`
	`    "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket",`
	`    "watchPort":  "9",`
	`    "actions":     [`
	`      {`
	`        "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputAction",`
	`        "outPort":  "10",`
	`      }`
	`    ]`
	`  }`
	`  {`
	`    "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket",`
	`    "watchPort":  "10",`
	`    "actions":     [`
	`      {`
	`        "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputAction",`
	`        "outPort":  "11",`
	`      }`
	`    ]`
	`  }`
	`]`

Table 5.44: Body values for Managing Group Entries

```
POST /api/default/config/groups HTTP/1.1
Content-Type:  application/json
HOST: localhost
Authorization:  Bearer 000...000
Content-Length:  522

{
  "displayName":  "D2 then D3",
  "node":  "aa3ebe3914748486d9b924311c5d05b3",
  "groupId":  "0",
  "groupType":  "FastFailover",
  "buckets":    [
    {
      "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket",
      "watchPort":  "9",
      "actions":        [
        {
          "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
OutputAction",
          "outPort":  "10",
        }
      ]
    }
    {
      "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket",
      "watchPort":  "10",
      "actions":        [
        {
          "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
OutputAction",
          "outPort":  "11",
        }
      ]
    }
  ]
}
```

Reply

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 770

{
  "@odata.context": "https://localhost/api/default/config/$metadata#groups/-
$entity",
  "id": "ab850efa03969493691238036696c5d2",
  "groupType": "FastFailover",
  "groupId": "0",
  "errorState": "InProgress",
  "errors":   [
  ]
  "node": "aa3ebe3914748486d9b924311c5d05b3",
  "enabled": "True",
  "displayName": "D2 then D3",
  "version": "1",
  "buckets":   [
    {
      "id": "a3d0e229448ac48d9ae9dd9f9c084884",
      "outGroup": "None",
      "outPort": "10",
      "watchGroup": "Any",
      "watchPort": "9",
      "actions":       [
        {
          "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
OutputAction",
          "maxLength": "65535",
          "outPort": "10",
        }
      ]
    }
    {
      "id": "a9d11e8d07f1e4e1c8db6060616ce83c",
      "outGroup": "None",
      "outPort": "11",
      "watchGroup": "Any",
      "watchPort": "10",
      "actions":       [
        {
          "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
OutputAction",
          "maxLength": "65535",
          "outPort": "11",
        }
      ]
    }
  ]
  "tags":   [
  ]
```

5.18 Managing Meter Entries

Request

Key	Value
Endpoint	/api/default/config/meters
Method	POST
Content-Type	application/json

Table 5.45: Header values for /api/default/config/meters

We need 5 properties for the json body.

Property	Value
displayName	`"SV Meter",`
node	`"aa3ebe3914748486d9b924311c5d05b3",`
meterId	`"1",`

flags	`[`
	`{`
	`"@odata.type": "#Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.PacketsPerSecond"`
	`}`
	`]`

meterBands	`[`
	`{`
	`"@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandDrop",`
	`"rate": "100",`
	`"burstSize": "1000",`
	`}`
	`]`

Table 5.46: Body values for Managing Meter Entries

```
POST /api/default/config/meters HTTP/1.1
Content-Type:  application/json
HOST: localhost
Authorization:  Bearer 000...000
Content-Length:  323

{
  "displayName":  "SV Meter",
  "node":  "aa3ebe3914748486d9b924311c5d05b3",
  "meterId":  "1",
  "flags":    [
    {
      "@odata.type":  "#Sel.Sel5056.TopologyManager.OpenFlow.Meter.-
RateTypeFlagObjects.PacketsPerSecond"
    }
  ]
  "meterBands":    [
    {
      "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand-
Drop",
      "rate":  "100",
      "burstSize":  "1000",
    }
  ]
}
```

Reply

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 504

{
  "@odata.context": "https://localhost/api/default/config/$metadata#meters/-
$entity",
  "id": "a68162ea210a6463abfd364b7ee5ad02",
  "meterId": "1",
  "errorState": "InProgress",
  "errors":   [
  ]
  "enabled": "True",
  "node": "aa3ebe3914748486d9b924311c5d05b3",
  "displayName": "SV Meter",
  "version": "1",
  "flags":   [
    {
      "@odata.type": "#Sel.Sel5056.TopologyManager.OpenFlow.Meter.-
RateTypeFlagObjects.PacketsPerSecond"
    }
  ]
  "meterBands":   [
    {
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand-
Drop",
      "burstSize": "1000",
      "rate": "100",
    }
  ]
  "tags":   [
  ]
}
```

Chapter 6

Data Modal

This section is an overview of the structure and relationships of data in the SEL-5056. Specific section later in this document cover specific points of the data modal.

The SEL-5056 communicates with several external parties. The most important are the user (or processes operating on behalf of the user) and the switches. However, there are others, such as end devices and LDAP and syslog servers.

6.1 OpenFlow Data Model

The switching behavior of OpenFlow switches operates partly or completely according to the OpenFlow configuration depending on the switch.

The SEL-5056 operates as the OpenFlow server to all OpenFlow devices it communicates over OpenFlow with and assumes that it is the only server managing the OpenFlow settings of a device. Although other servers may be present, a change of configuration that is actively managed by the SEL-5056 will result in a sychronization request. If a change should occur from another server, that change would also need to be pushed to the SEL-5056 as the SEL-5056 considers any change that does not match it's own configuration as a deviation.

An OpenFlow switch has several objects under management, the high levels being the OpenFlow tables and ports. The OpenFlow entry objects are flow entries (contained in flow tables), group entries (contained in the group table), and meter entries (contained in the meter table).

An OpenFlow switch therefore contains one or more flow tables, a group table, a meter table, and a set of ports and their associated constituents. The SEL-5056 itself contains one flow table, one group table, one meter table, and one table for port configuration for all switches together.

The SEL-5056 supports all of the OpenFlow capabilities of the switches but does not support all OpenFlow properties or operations.

Further details on the SEL-5056 data modal can be found: here, here, and here.

The OpenFlow specification 1.3.4 contain additional details about the OpenFlow data model. The OpenFlow protocol also defines how the information is encoded between the Controller and the SEL-5056. This is hidden from the user using TLS. Some of the OpenFlow information, specifically the diagnostics and features, are stored in memory and are available on the REST interface.

6.1.1 Simple View of an OpenFlow Switch

At a high level view, an OpenFlow only switch (as defined by the OpenFlow specification) forwards packets based on the behavior defined in the flow, group, and meter entries programmed to the switch.

6.2 Logical Programming Data Model

Logical programming is an automated method for creating OpenFlow programming. Further details are covered here and here. This method does not replace OpenFlow entries, but only how they are generated. By defining LC (logical connections), OpenFlow entries are generated.

6.3 SEL-2740S Management Data Model

Although the SEL-5056 can manage devices through the OpenFlow interface, the OpenFlow protocol only manages a subset of all necessary or desired functionality. Firmware management and initial configuration are outside of the OpenFlow protocol and so are part of the management protocol.

For the SEL-2740S, the SEL-5056 manages not only the OpenFlow settings, it also manages the other switches settings through a separate protocol (referred to as the Management protocol in this document). This management protocol manages all of the non-OpenFlow settings of the switch. The SEL-5056 can also be managed by something else, but the REST calls are not covered in this document. This document assumes that all management of the SEL-2740S occurs from the SEL-5056.

The management functionality for the SEL-2740S is contained within the same space as other functionality. This document identifies the scope of any device call.

6.4 SEL-5056 Data Model

The core of the SEL-5056 is exposed through a JSON-encoded OData schemas. Through OData, the SEL-5056 API is divided into 5 service roots (or trees). Each tree contains a set of objects and actions that can be performed on those objects.

A list can be found here 8.2.

The most important trees for network operations is the configuration and operational trees. The configuration tree holds the settings and actions for objects that can exist independent of physical objects, referred to as configuration objects. The operational tree holds the settings and actions for objects that are dependent on physical objects. Here the term physical can cover both real objects and also virtual objects. These are referred to as operational objects.

For an example, consider OpenFlow flow entries. On the configuration tree are the flow entries with their associated settings, e.g. table id, match fields. One of the settings is a node, which is the id of a configuration node. On the operational tree are the statistics/diagnostics for those flow entries. The flow entries are tied to an operational node through the configuration node setting.

Because the flow entries are tied to configuration objects and not operational objects, if the operational object is replaced, only the relationship between the configuration node and the operational node needs to change, which the SEL-5056 automatically handles during the adoption and unadopt processes. Also, because the flow entries are tied to configuration objects, no operational objects are necessary to pre-configure the network.

6.5 Topology Model

Physical devices and the topology must be represented by objects. In the SEL-5056, those topology objects are: nodes, ports, and links.

Nodes are the physical devices, ports are the interfaces of a physical device, and links are connections between interfaces.

Each of these three types are further divided into two types of objects: configuration and operational. The configuration objects hold the settings of a device independent of whether or not a physical object exists and the operational object represents some understanding of the physical object. There are 4 types of nodes, discussed further here.

6.6 Defined versus variable settings

Each object has one or more properties, some of which are defined settings, such as the node setting for a flow entry. Another type of setting are variable settings contained within the attributes, tag, or configuration settings. This is where settings that may or may not apply to an object or settings with a variable count are found, such as the IP address for operational objects.

6.6.1 Attributes

Some properties of objects are stored in attributes instead of in specific properties. For example, the MAC address of an operational node is stored as an attribute not as a property named something like macAddress.

Attributes are stored as unsorted lists of dictionaries. Each dictionary contains an @odata.type properties along with zero or more properties dependent on the @odata.type. Not all attributes can be used with all objects.

TODO List of objects to attributes

An IP address attribute looks like:

```
{
    "@odata.type": "#Sel.Sel5056.TopologyManager.Attributes.Operational.Node.IpAttr",
    "ipAddress": "192.168.1.157"
}
```

An object may have 0 or more attributes.

6.6.2 Tags

Tags are another method that additional, non specific property data can be stored in an object. Like attributes, these tags are stored as unsorted list of dictionaries, where each dictionary contains one or more properties. All tags, like attributes contain the @odata.type property.

6.6.3 Configurations

Stuff about configurations

Part II

REST Overview

Chapter 7

HTTP

7.1 HTTPS

The SEL-5056 requires HTTP over TLS (HTTPS) connections to the rest interface. HTTP nor HTTP redirect to HTTPS are supported.

7.1.1 Headers

The Authorization header is used to send the token along with the REST request.

Other headers are required per the HTTP standard, with may include

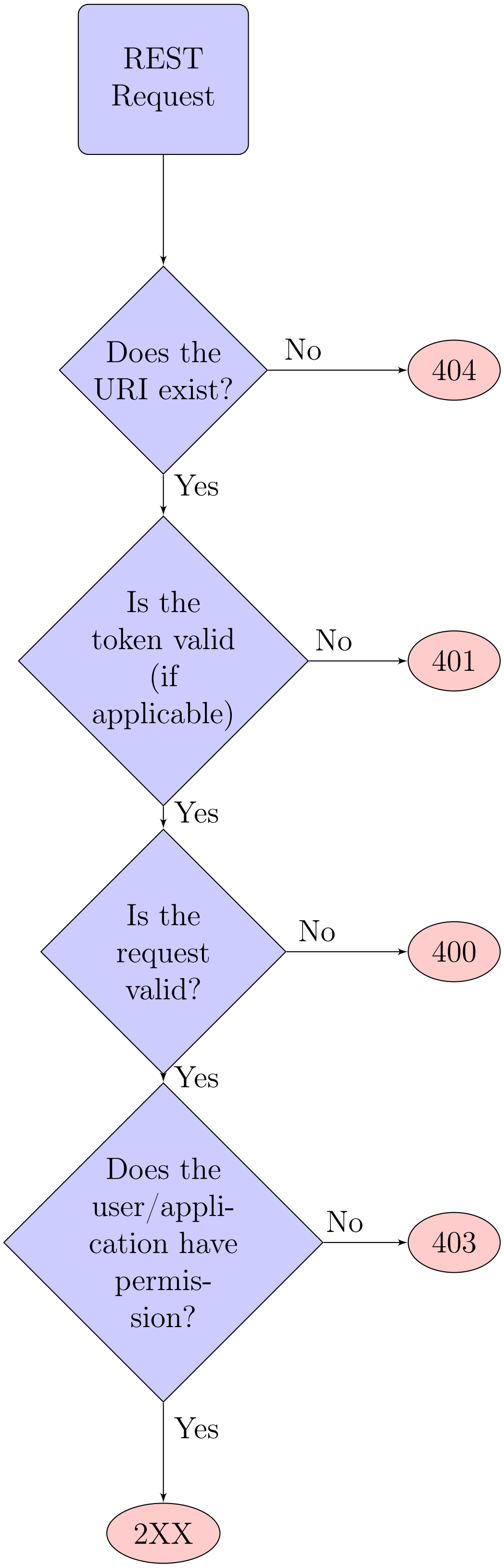
- Host

- Content-Length

- Content-Type

7.1.2 Response Codes

There is a precedence in which the request is handled (relevant ones shown below)]



Deleting an object that does not exist returns 200.

7.1.3 Common Format of HTTP Methods

GET

GET has 2 types and 3 forms:

Number of Objects Targeted	Form 1	Form 2
All	~/root/entity	
One	~/root/entity('id')	~/root/entity/id

Example

Number of Objects Targeted	Form 1	Form 2
All	~/config/nodes/	
One	~/config/nodes('abc...def')	~/config/nodes/abc...def

POST

POSTing is used both for adding objects to an entity set and also calling an action. POST has 2 types and 3 forms:

Function	Form 1	Form 2
Adding multiple objects	Not supported	
Adding an object	~/root/entity	
Bound action	~/root/entity('id')/function	~/root/entity/id/function
Unbound action	~/root/entity	

Where function is the function name, such as Adopt. This document uses the ('id) form of providing the id.

Functions have two forms:

- (function) such as Adopt
- Sel.(function) such as Sel.Adopt

The first form is used in this document.

Example

Function	Form 1	Form 2
Adding an object	~/config/nodes	
Bound action	~/operational/nodes('abc...def')/Adopt	~/operational/nodes/abc...def/Adopt
Unbound action	~/operational/GetPortByAttribute	

PATCH or PUT

There are several methods to modify an object, depending on the object.

- PATCH or PUT the object
- PUT only the object
- An action to perform the action

For example, change the name of a CST, you may PUT or PATCH. However, add a linkedKey to an operational object, you must use the Adopt function.

DELETE

DELETE has 2 forms:

Number of Objects Targeted	Form 1	Form 2
Multiple	Not supported	
One	˜/root/entity('id')	˜/root/entity/id

The first format is used in this document.

Chapter 8

OData

8.1 OData Version

The SEL-5056 only supports version 4.0. If the OData-Version header or OData-MaxVersion is used, it must be set to 4.0

```
OData-Version: 4.0
```

Other values return a 404.

8.2 Service Roots

The SEL-5056 has 5 service roots:

root	description
certificate	X.509 certification endpoints
config	Configuration endpoints
operational	Operational endpoints
security	User and application endpoints
settings	Other endpoints

Table 8.2: OData Service Roots

The lists of available root and the API version can be found on the REST interface at: ~/configuration/settings.json

The entities at each root can be found by accessing: ~/root/, such as ~/api/default/certificate/

The OData schema metadata that describes the endpoints are accessible at roots by appending /$metadata to the service root, such as /api/default/config$metadata.

8.2.1 Common Properties

Many of the endpoints share common properties.

property	description
displayName	Friendly name
id	Unique key for the item, usually a UUID
errorState	State of the item
errors	Errors of the item, if in the error state
version	Version of the item, increments for each change

8.2.2 Filtering

The SEL-5056 through OData supports returning a subtest of results when using the GET method with some endpoints. There are several web resources available. This section covers filtering specific to the objects of the SEL-5056.

Many endpoints contain primitive type properties (strings, numbers, boolean, or null) in the outermost level. These are filterable by $filter=(property) (operator) (value).

For example, to find a (because cookies are unique) flow entry with a cookie of 1000 (an integer), you can use:
/api/default/config/flows?$filter=cookie eq 1000

To find all flows that are enabled (boolean):
/api/default/config/flows?$filter=enabled eq true

To find all flows that have an errorState of Success (string):
/api/default/config/flows?$filter=errorState eq 'Success'

Within a flow entry can be multiple complex types, such as the match object, the instruction array, and the tag array.

To filter on these advanced types, you can use the form:
$filter=<complex type name>/any(t: isof(t, '<odata type>') and t/<odata type>/<complex type property> <operator> <value>)

For example, to find all of the flows that belong to a logical connection with id a51a0437c95c3424da8f60568011e84a, you can use:
/api/default/config/flows?$filter=tags/any(t: isof(t, 'Sel.Sel5056.Tags.LogicalConnectionTag') and t/Sel.Sel5056.Tags.LogicalConnectionTag/logicalConnectionId eq 'a51a0437c95c3424da8f60568011e84a')

The tag type is Sel.Sel5056.Tags.LogicalConnectionTag and the property with the id is the logicalConnectionId property.

Chapter 9

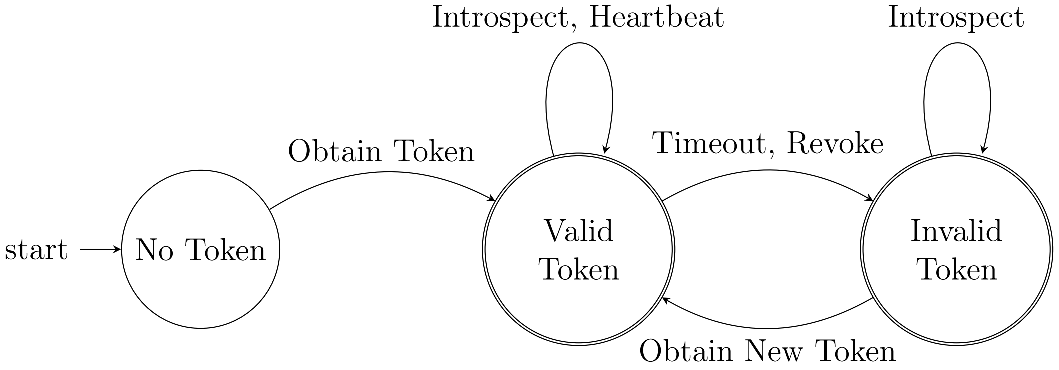
OAuth Authentication

This section explains the process of obtain, managing, and revoking authorization to the REST interface of the SEL-5056. OAuth handles the authorization and Connect ID the identity for interaction with the SEL-5056. OAuth is not explained in detail in this document, but as a popular protocol, resources are available on the web.

This section assumes that you have credentials or a registered application as necessary for obtaining a token.

The SEL-5056 is both the authorization server and the resource server for OAuth.

Overview of the process:



The simplest grant type is the Resource Owner Password Credentials grant, where the external client uses a known username and password to obtain a token (example here). The other three grant types rely on other methods to obtain a token.

The details on obtaining a token are presented below.

9.1 Including tokens in a request

Tokens are of type Bearer (RFC 6750)

Token must be appended to the Authorization header to OData endpoints (i.e. those that are not for token management or SignalR) if the endpoint requires permissions to access. For example, if the token received by the SEL-5056 is c9e4fc629059422f25a63864003e185b, then to make a call to get the PermissionLevel3 role, you would make the following HTTP GET request (modify HOST to match your setup):

```
GET /api/default/security/roles('PermissionLevel3') HTTP/1.1
```

```
HOST: localhost
Authorization: Bearer c9e4fc629059422f25a63864003e185b
```

Listing 9.1: Using an OAuth token

This is independent of the grant type used to obtain the token.

SignalR requests use a different header for including the token in the HTTP header.

9.2 OAuth and Connect ID Parameters

9.2.1 Grant Types

The SEL-5056 supports the following grant types and supported clients:

Grant Type	Typical Clients	Registered Application Required
Authorization Code	Applications	Yes
Implicit	UI	No
Resource Owner Password Credentials	Any	No
Client Credentials	Applications	Yes

Table 9.1: Supported OAuth Grant Types

Some grant types are only available to registered applications. In the following table, columns 2-5 refer to application specific requirements for registered applications.

Grant Type	Application Registration	Client Certificate Upload	IP Address Binding	Permission List	User login required for token
Authorization Code	Required	Yes	Yes	Yes	Yes
Implicit	Not Possible	No	No	No	Yes
Resource Owner Password Credentials	Not Possible	No	No	No	Yes
Client Credentials	Required	Yes	No	Yes	No

Table 9.2: OAuth Grant Types Requirements

Each grant type has a string representation that is used when obtaining a token.

Grant Type	Grant type string
Authorization Code	authorization_code
Implicit	(None)
Resource Owner Password Credentials	password
Client Credentials	client_credentials

9.2.2 Client ID and Client Secret

Each grant type has a corresponding client id and one also has a client secret.

Grant Type	Client ID	Client Secret
Authorization Code	(application name)-AuthorizationCode	(None)
Implicit	web-client	(None)
Resource Owner Password Credentials	password-client	Rest Interface
Client Credentials	(application name)-ClientCredentials	(None)

Table 9.3: OAuth Grant Types Client ID and Secret

9.2.3 Scope

The scope must be "Rest". The permissions assigned to the role or the application control what actions a token allows the bearer to do. See 9.5 for further information.

9.2.4 Roles

Users must have a role to obtain a token. The role determines what actions the user can perform. The following roles are available:

Role	Role string	Description
Security Administrator	SecurityAdministrator	Manage users, certificates and other non-network settings
Permissions Level 3	PermissionLevel3	Manage network settings
Monitor	Monitor	Read only version of Permission Level 3

Roles are encoded into the token request using the acr_values parameter, such as acr_values=role:PermissionLevel3.
Applications do not have roles, instead, they have permissions.

9.3 Endpoints

Endpoint	Purpose
/identity/connect/token	Obtaining a token
/identity/connect/authorize	Obtaining an authorization code
/identity/connect/revocation	Revoking a token ("logging off")
/identity/connect/accesstokenvalidation	Confirm whether or not a token is valid
/identity/connect/introspect	Token Introspection
/api/default/settings/Heartbeat	Extend expiration of token

Table 9.5: OAuth/Token Endpoints

The following OAuth functionality are not supported:

- Refresh tokens

9.3.1 Obtaining a token

URL: /identity/connect/token
Method: POST

Request Structure

The request structure depends on the grant type. The general parameters common to all grant types are:

name	values	description
grant_type	(One of the grant types defined here)	OAuth Grant Type
client_id	password-client	OAuth Client ID
scope	Rest	OAuth Scope
state	(Any opaque string)	OAuth State

See Resource Owner Password Credentials.

Response Structure

All response structures are the same, regardless of grant type.

name	values	description
access_token	Token	Bearer token used in HTTP requests that require a token
expires_in	900	Number of seconds that token is valid from point of issue
token_type	Bearer	OAuth token type

9.3.2 Resource Owner Password Credentials

Request Structure

Including the general parameters, the following are required for this grant type:

grant_type	(One of the grant types defined here)	OAuth Grant Type
client_id	password-client	OAuth Client ID
scope	Rest	OAuth Scope
state	(Any opaque string)	OAuth State
username	(Username of user logging in)	-
password	(Password of user logging in)	-
acr_values	role:(Role Name)	user's role
client_secret	Rest Interface	OAuth Client Secret

Table 9.9: Resource Owner Password Credentials Token Request Parameters

Resource Owner Password Credentials Example

The user is "example" with password "!qazxsw@" with role Permission Level 3. The token is c9e4fc629059422f25a63864003e185b in this example.

Example 1.

```
POST /identity/connect/token HTTP/1.1
HOST: 192.168.255.1
Content-Type: application/x-www-form-urlencoded
Content-Length: 167
```

```
grant_type=password&
username=example&
password=1qazXSW%40&
acr_values=role%3APermissionLevel3&
client_secret=Rest%20Interface&
client_id=password-client&
scope=Rest&
state=0


HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, max-age=0, private
Pragma: no-cache
Content-Length: 90
Content-Type: application/json; charset=utf-8
Server: Microsoft-HTTPAPI/2.0
Date: Fri, 31 May 2019 18:44:21 GMT

{"access_token":"c9e4fc629059422f25a63864003e185b","expires_in
   ":900,"token_type":"Bearer"}
```

Listing 9.2: "Password Credentials Token (Normalized with line wrapping)"

9.3.3 Client Credentials

Request Structure

Including the general parameters, the following are required for this grant type:

grant_type	(One of the grant types defined here)	OAuth Grant Type
client_id	password-client	OAuth Client ID
scope	Rest	OAuth Scope
state	(Any opaque string)	OAuth State

Table 9.11: Client Credentials Token Request Parameters

Client Credentials Example

The application name is "example".

Example 2.

```
POST /identity/connect/token HTTP/1.1
HOST: 192.168.255.1
Content-Type: application/x-www-form-urlencoded
Content-Length: 98

grant_type=client_credentials&scope=Rest&client_id=example-
   ClientCredentials&state=0
```

```
HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, max-age=0, private
Pragma: no-cache
Content-Length: 91
Content-Type: application/json; charset=utf-8
Server: Microsoft-HTTPAPI/2.0
Date: Fri, 31 May 2019 19:37:43 GMT

{"access_token":"b0ec811b4e6a025659bf08dd48756289","expires_in
   ":5000,"token_type":"Bearer"}
```

Listing 9.3: "Client Credentials Token Request (Normalized with line wrapping)"

9.3.4 Inspecting a token

Example 3.

```
POST /identity/connect/accesstokenvalidation HTTP/1.1
HOST: localhost
Content-Length: 38

token=8014fe15fbf92f531d3cd40d2e424e80


HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, max-age=0, private
Pragma: no-cache
Content-Length: 293
Content-Type: application/json; charset=utf-8
Server: Microsoft-HTTPAPI/2.0
Date: Tue, 02 Jul 2019 20:39:51 GMT

{"aud":"https://localhost/identity/resources","iss":"https://
   localhost/identity","nbf":1562099842,"exp":1562100742,"client_id
   ":"password-client","scope":"Rest","sub":"admin","name":"admin","
   amr":"password","idp":"idsrv","auth_time":1562099842,"role":"
   PermissionLevel3","IpAddress":"127.0.0.1"}
```

Listing 9.4: "Inspecting a token"

Revoking a Token

Example 4.
To revoke your token (i.e. "logout"):

```
POST /identity/connect/revocation HTTP/1.1
HOST: 192.168.255.1
```

```
Content-Type: application/x-www-form-urlencoded
Authorization: Bearer faf5e69b8bb3865ce2284289f7d5fc19
Content-Length: 124

token=c9e4fc629059422f25a63864003e185b&token_type_hint=access_token&
   client_id=password-client&client_secret=Rest%20Interface


HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, max-age=0, private
Pragma: no-cache
Content-Length: 0
Server: Microsoft-HTTPAPI/2.0
Date: Fri, 31 May 2019 19:32:49 GMT
```

Listing 9.5: "Password Credentials Token Revocation (Normalized with line wrapping)"

Example 5.

```
POST /identity/connect/revocation HTTP/1.1
HOST: 192.168.255.1
Content-Type: application/x-www-form-urlencoded
Authorization: Bearer b0ec811b4e6a025659bf08dd48756289
Content-Length: 117

token=b0ec811b4e6a025659bf08dd48756289&token_type_hint=access_token&
   client_id=example-ClientCredentials


HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, max-age=0, private
Pragma: no-cache
Content-Length: 0
Server: Microsoft-HTTPAPI/2.0
Date: Fri, 31 May 2019 19:37:43 GMT
```

Listing 9.6: "Client Credentials Token Revocation (Normalized with line wrapping)"

Client credential sessions do not count against session limit.

9.3.5 Extending the expiration time of a Token

Applies to all grant types and requires no permissions.
Example 6.

```
POST /api/default/settings/Heartbeat HTTP/1.1
HOST: localhost
Authorization: Bearer c9e4fc629059422f25a63864003e185b
Content-Length: 0
```

```
HTTP/1.1 200 OK
Content-Length: 0
Server: Microsoft-HTTPAPI/2.0
Date: Tue, 02 Jul 2019 20:32:50 GMT
```

Listing 9.7: "Client Credentials Token Revocation (Normalized with line wrapping)"

9.4 Refreshing Token

The SEL-5056 does not support refresh tokens. The Heartbeat end point can be used to extend the expiration time of the token. The expiration time of the token is also reset back to the original value whenever the token is used.

9.5 Managing Permissions

The permissions of a client using the authorization grant is the intersection of the requested permissions at registration and the permissions of the role of the user authorizing the client. The permissions of a client using the client credentials grant is all of the permissions requested at registration.

User permissions are covered here.

9.6 Troubleshooting

If you receive a 401 Unauthorized when making any REST call, check that the token exists and is valid. You can do this using the /identity/connect/introspect endpoint. Also confirm that you have added the Authorization header to the HTTP request of the form:

```
Authorization: Bearer X
```

where X is the token received.
For example

```
GET /api/default/config/nodes/ HTTP/1.1
HOST: 192.168.1.1
Authorization: Bearer d61f3cc466a02535c2a93cecb08ff07ed532ce2b3129e816e1c667539906bbae
```

9.6.1 Registered Applications

This part applies only if you registered an application using one of methods outlined in Application Management.

If you receive 400 Bad Request with error invalid_client when obtaining a token, check that the application is registered AND enabled.

If you receive 401 Unauthorized when making any REST call after obtaining, check that:

1. The certificate that has the same thumbprint as certificateThumbprint if you used the /security/applicationLinks endpoint to register is uploaded and not revoked

2. The certificate that you used during the TLS exchanged when using the /security/applicationLinks/InitiateApplicationRegistration/ endpoint to register is uploaded and not revoked

3. You did not register the application with ipAddress of 127.0.0.1.

4. The IP address that the SEL-5056 see for the application is the same as the applicationIp passed during registration

Part III

OData Endpoints

The following chapters cover the OData REST endpoints divided by their general function. The OAuth endpoints are covered here and the SingalR endpoints are covered here.

Each chapter contains an introduction followed by sections for each group of endpoints. For example, chapter 9 covers user management. The first subsection covers the /security/users endpoints.

The endpoints for each group are divided into different purposes, the most basic being CRUD (create, read, update, and delete). Other purposes could include diagnostics or other actions. The HTTP method is also included.

The section is then divided into each endpoint-method combination. For example, Read Users, which covers the endpoint /api/default/security/users/ and the method GET.

For the GET method, the return types and the JSON body of each return type is displayed. Further details on each property of the JSON can be found in part 3 by following the link.

Roles and permissions cover the user role that can use the endpoint and also the SEL-5056 permission for use with applications.

Finally, examples or further discussion can follow.

Commands are carried by HTTP. There are 5 methods covered in this document.

- GET
- POST
- PATCH
- PUT
- DELETE

See HTTP for further information on formatting URIs.

Chapter 10

User Management

There are three types of users:

- Local
- External (created before the user logs in and associated to an authentication service)
- Remote (authenticated through LDAP with no requirement to be precreated)

The first two users are managed through User Accounts. The later is managed through authentication services. LDAP is the only supported authentication service.

10.1 User Accounts

purpose	endpoints	methods
read	/security/users/ /security/users('id')/	GET GET
create	/security/users/	POST
update	/security/users('id')/ /security/users('id')/ /security/users('id')/UpdatePassword	PUT PATCH POST
delete	/security/users('id')/	DELETE

10.1.1 Read users

URL: /api/default/security/users/
Method: GET
OData Type: Sel.BlueFrame.Core.SecurityManager.User
There are 2 type(s) that can be returned: Sel.BlueFrame.Core.SecurityManager.LocalUser, Sel.BlueFrame.Core.SecurityManager.AuthServiceUser The return will be a list of one or more of these types show below

Response Structure for Sel.BlueFrame.Core.SecurityManager.LocalUser

```
{
  "@odata.type":  "#Sel.BlueFrame.Core.SecurityManager.LocalUser",
  "clearTextPass":  Edm.String,
  "displayName":  Edm.String,
  "enabled":  Edm.Boolean,
  "failedLogins":  [ Sel.BlueFrame.Core.SecurityManager.UserFailedLogin, ],
  "id":  Edm.String (UUID),
  "isLockedOut":  Edm.Boolean,
  "lastSuccessLoginIp":  Edm.String,
  "lastSuccessfulLogin":  Edm.DateTimeOffset,
  "lockedOutUntil":  Edm.DateTimeOffset,
  "lockoutTime":  Edm.DateTimeOffset,
  "roles":  [ Edm.String, ] (Sel.BlueFrame.Core.SecurityManager.Role.id),
  "username":  Edm.String,
}
```

Response Structure for Sel.BlueFrame.Core.SecurityManager.AuthServiceUser

```
{
  "@odata.type":  "#Sel.BlueFrame.Core.SecurityManager.AuthServiceUser",
  "authService":  Edm.String (Sel.BlueFrame.Core.SecurityManager.AuthService.id),
  "displayName":  Edm.String,
  "enabled":  Edm.Boolean,
  "failedLogins":  [ Sel.BlueFrame.Core.SecurityManager.UserFailedLogin, ],
  "id":  Edm.String (UUID),
  "isLockedOut":  Edm.Boolean,
  "lastSuccessLoginIp":  Edm.String,
  "lastSuccessfulLogin":  Edm.DateTimeOffset,
  "lockedOutUntil":  Edm.DateTimeOffset,
  "lockoutTime":  Edm.DateTimeOffset,
  "roles":  [ Edm.String, ] (Sel.BlueFrame.Core.SecurityManager.Role.id),
  "username":  Edm.String,
}
```

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "User",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SecurityTree"
}
```

10.1.2 Create users

URL: /api/default/security/users/
Method: POST
OData Type: Sel.BlueFrame.Core.SecurityManager.User
There are 2 type(s) that can be POSTed: Sel.BlueFrame.Core.SecurityManager.LocalUser, Sel.BlueFrame.Core.SecurityManager.AuthServiceUser

Request Structure for Sel.BlueFrame.Core.SecurityManager.LocalUser

Property	Default Value
enabled	true
failedLogins	[]
isLockedOut	false
lastSuccessLoginIp	null
lastSuccessfulLogin	0001-01-01T00:00:00-07:00
lockedOutUntil	0001-01-01T00:00:00-07:00
lockoutTime	0001-01-01T00:00:00-07:00
roles	[]

Table 10.3: Default Values for Sel.BlueFrame.Core.SecurityManager.LocalUser

```
{
  "@odata.type":  "#Sel.BlueFrame.Core.SecurityManager.LocalUser",
  "clearTextPass":  Edm.String,
  "enabled":  Edm.Boolean,
  "failedLogins":  [ Sel.BlueFrame.Core.SecurityManager.UserFailedLogin, ],
  "isLockedOut":  Edm.Boolean,
  "lastSuccessLoginIp":  Edm.String,
  "lastSuccessfulLogin":  Edm.DateTimeOffset,
  "lockedOutUntil":  Edm.DateTimeOffset,
  "lockoutTime":  Edm.DateTimeOffset,
  "roles":  [ Edm.String, ] (Sel.BlueFrame.Core.SecurityManager.Role.id),
  "username":  Edm.String,
}
```

Request Structure for Sel.BlueFrame.Core.SecurityManager.AuthServiceUser

Property	Default Value
displayName	Engineer
enabled	true
failedLogins	[]
isLockedOut	false
lastSuccessLoginIp	null
lastSuccessfulLogin	0001-01-01T00:00:00-07:00
lockedOutUntil	0001-01-01T00:00:00-07:00

lockoutTime	0001-01-01T00:00:00-07:00
roles	[]

Table 10.5: Default Values for Sel.BlueFrame.Core.SecurityManager.AuthServiceUser

```
{
  "@odata.type":  "#Sel.BlueFrame.Core.SecurityManager.AuthServiceUser",
  "authService":  Edm.String (Sel.BlueFrame.Core.SecurityManager.AuthService.id),
  "displayName":  Edm.String,
  "enabled":  Edm.Boolean,
  "failedLogins":  [ Sel.BlueFrame.Core.SecurityManager.UserFailedLogin, ],
  "id":  Edm.String (UUID),
  "isLockedOut":  Edm.Boolean,
  "lastSuccessLoginIp":  Edm.String,
  "lastSuccessfulLogin":  Edm.DateTimeOffset,
  "lockedOutUntil":  Edm.DateTimeOffset,
  "lockoutTime":  Edm.DateTimeOffset,
  "roles":  [ Edm.String, ] (Sel.BlueFrame.Core.SecurityManager.Role.id),
  "username":  Edm.String,
}
```

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "User",
   "module": "DataBroker",
   "permissionTag": "Create",
   "tree": "SecurityTree"
}
```

Examples

Add a permission level 3 user Example 7.

```
POST /api/default/security/users/ HTTP/1.1
Content-Type: application/json
HOST: 192.168.56.1
Authorization: Bearer 92
   a721ebbc2df860e876f55a2e6a3eaf8faccd379c43ae20ca9682d7433007cf
Content-Length: 144

{
   "@odata.type": "#Sel.BlueFrame.Core.SecurityManager.LocalUser",
   "clearTextPass": "1Example$",
   "roles": [
      "PermissionLevel3"
   ],
   "username": "pl3"
}
```

Listing 10.1: Add a permission level 3 user

10.1.3 Update users (PUT)

URL: /api/default/security/users('id')/
Method: PUT
OData Type: Sel.BlueFrame.Core.SecurityManager.User
Binding parameter (id): Sel.BlueFrame.Core.SecurityManager.User.id
There are 2 type(s) that can be modified: Sel.BlueFrame.Core.SecurityManager.LocalUser, Sel.BlueFrame.Core.SecurityManager.AuthServiceUser

Request Structure for Sel.BlueFrame.Core.SecurityManager.LocalUser

The request structure is the same as the GET response above. Some fields may be omitted if not POSTable. Any POSTable fields not included will be reset to the default state.

Request Structure for Sel.BlueFrame.Core.SecurityManager.AuthServiceUser

The request structure is the same as the GET response above. Some fields may be omitted if not POSTable. Any POSTable fields not included will be reset to the default state.

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "User",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "SecurityTree"
}
```

10.1.4 Update users (PATCH)

URL: /api/default/security/users('id')/
Method: PATCH
OData Type: Sel.BlueFrame.Core.SecurityManager.User
Binding parameter (id): Sel.BlueFrame.Core.SecurityManager.User.id

There are 2 type(s) that can be modified: Sel.BlueFrame.Core.SecurityManager.LocalUser, Sel.BlueFrame.Core.SecurityManager.AuthServiceUser

The following properties are patchable for type Sel.BlueFrame.Core.SecurityManager.LocalUser

- enabled
- isLockedOut
- roles
- username

Request Structure for Sel.BlueFrame.Core.SecurityManager.LocalUser

The request structure includes one or more properties that are patchable. Any property not included in the request structure will retain its current value (subject to any property relationships).
The following properties are patchable for type Sel.BlueFrame.Core.SecurityManager.AuthServiceUser

- displayName
- enabled
- isLockedOut
- roles
- username

Request Structure for Sel.BlueFrame.Core.SecurityManager.AuthServiceUser

The request structure includes one or more properties that are patchable. Any property not included in the request structure will retain its current value (subject to any property relationships).

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "User",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SecurityTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "User",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "SecurityTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "User",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "SecurityTree"
}
```

Examples

Disable a user **Example 8.**

```
PATCH /api/default/security/users('a13f7c00f65eb42449f15f43f1b47176
   ') HTTP/1.1
Content-Type: application/json
HOST: 192.168.56.1
Authorization: Bearer
   e4c75cae10c586347710727fc0dce8baad2f6d2f813bcbbb6849e89bc9a982e4
Content-Length: 20

{
   "enabled": "False"
}
```

Listing 10.2: Disable a user

10.1.5 Change the current user's password

URL: /api/default/security/users('id')/UpdatePassword
Method: POST
OData Type: Sel.UpdatePassword
Binding parameter (id): Sel.BlueFrame.Core.SecurityManager.User.id
There are 1 type(s) that can be POSTed: Sel.UpdatePassword

Request Structure

```
{
  "newPassword":  Edm.String,
  "oldPassword":  Edm.String,
}
```

User Roles with Required Permissions

- Monitor
- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

```
{
   "module": "Security Manager",
   "permissionTag": "ChangeOwnPassword"
}
```

10.1.6 Delete users

URL: /api/default/security/users('id')/
Method: DELETE
OData Type: Sel.BlueFrame.Core.SecurityManager.User
Binding parameter (id): Sel.BlueFrame.Core.SecurityManager.User.id

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "User",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "SecurityTree"
}
```

10.2 User Roles

purpose	endpoints	methods
read	/security/roles/ /security/roles('id')/	GET GET
create	Not supported	
update	Not supported	
delete	Not supported	

10.2.1 Read user roles

URL: /api/default/security/roles/

Method: GET
OData Type: Sel.BlueFrame.Core.SecurityManager.Role
There are 1 type(s) that can be returned: Sel.BlueFrame.Core.SecurityManager.Role The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "displayName":  Edm.String,
    "id":  Edm.String,
    "permissions":  [ Sel.BlueFrame.Core.SecurityManager.Permission, ],
    "roleName":  Edm.String,
  , ]
}
```

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Role",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SecurityTree"
}
```

10.3 Authentication Services

purpose	endpoints	methods
read	/security/authServices/ /security/authServices('id')/ /security/authServices('id')/GetAvailableGroups	GET GET GET
create	/security/authServices/	POST
update	/security/authServices('id')/ /security/authServices('id')/	PUT PATCH
delete	/security/authServices('id')/	DELETE
test	/security/authServices('id')/TestAuthenticateUser	POST

10.3.1 Read created authentication services

URL: /api/default/security/authServices/
Method: GET

OData Type: Sel.BlueFrame.Core.SecurityManager.AuthService
There are 1 type(s) that can be returned: Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService The return will be a list of one or more of these types show below

Response Structure for Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService

```
{
  "@odata.type":  "#Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService",
  "authType":  System.DirectoryServices.Protocols.AuthType,
  "bindDN":  Edm.String,
  "bindPassword":  Edm.String,
  "email":  Edm.String,
  "firstName":  Edm.String,
  "groupMemberAttribute":  Edm.String,
  "hostname":  Edm.String,
  "lastName":  Edm.String,
  "name":  Edm.String,
  "portNumber":  Edm.Int32,
  "protocolVersion":  Edm.Int32,
  "searchBase":  Edm.String,
  "userIdFilter":  Edm.String,
  "workPhone":  Edm.String,
  "errorState":  Sel.BlueFrame.Core.ErrorState,
  "errors":  [ Edm.String, ],
  "id":  Edm.String,
}
```

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "AuthService",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SecurityTree"
}
```

10.3.2 Create authentication services

URL: /api/default/security/authServices/
Method: POST
OData Type: Sel.BlueFrame.Core.SecurityManager.AuthService

There are 1 type(s) that can be POSTed: Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService

Request Structure for Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService

Property	Default Value
bindDN	null
bindPassword	null
email	null
firstName	null
groupMemberAttribute	null
lastName	null
searchBase	null
userIdFilter	null
workPhone	null

Table 10.9: Default Values for Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService

```
{
  "@odata.type":  "#Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService",
  "bindDN":  Edm.String,
  "bindPassword":  Edm.String,
  "email":  Edm.String,
  "firstName":  Edm.String,
  "groupMemberAttribute":  Edm.String,
  "hostname":  Edm.String,
  "lastName":  Edm.String,
  "name":  Edm.String,
  "portNumber":  Edm.Int32,
  "searchBase":  Edm.String,
  "userIdFilter":  Edm.String,
  "workPhone":  Edm.String,
}
```

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "AuthService",
   "module": "DataBroker",
   "permissionTag": "Create",
   "tree": "SecurityTree"
}
```

10.3.3 Update created authentication services (PUT)

URL: /api/default/security/authServices('id')/
Method: PUT
OData Type: Sel.BlueFrame.Core.SecurityManager.AuthService
Binding parameter (id): Sel.BlueFrame.Core.SecurityManager.AuthService.id
There are 1 type(s) that can be modified: Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService

Request Structure for Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService

The request structure is the same as the GET response above. Some fields may be omitted if not POSTable. Any POSTable fields not included will be reset to the default state.

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "AuthService",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "SecurityTree"
}
```

10.3.4 Update created authentication services (PATCH)

URL: /api/default/security/authServices('id')/
Method: PATCH
OData Type: Sel.BlueFrame.Core.SecurityManager.AuthService
Binding parameter (id): Sel.BlueFrame.Core.SecurityManager.AuthService.id
There are 1 type(s) that can be modified: Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService
The following properties are patchable for type Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService

- bindDN
- bindPassword
- email
- firstName
- groupMemberAttribute
- hostname

- lastName
- name
- portNumber
- searchBase
- userIdFilter
- workPhone

Request Structure for Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService

The request structure includes one or more properties that are patchable. Any property not included in the request structure will retain its current value (subject to any property relationships).

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "AuthService",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SecurityTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "AuthService",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "SecurityTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "AuthService",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "SecurityTree"
}
```

10.3.5 Delete created authentication services

URL: /api/default/security/authServices('id')/
Method: DELETE
OData Type: Sel.BlueFrame.Core.SecurityManager.AuthService
Binding parameter (id): Sel.BlueFrame.Core.SecurityManager.AuthService.id

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "AuthService",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "SecurityTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "AuthServiceGroup",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SecurityTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "AuthServiceGroup",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "SecurityTree"
}
```

10.3.6 Test an authentication service

URL: /api/default/security/authServices('id')/TestAuthenticateUser
Method: POST
OData Type: Sel.TestAuthenticateUser
Binding parameter (id): Sel.BlueFrame.Core.SecurityManager.AuthService.id
There are 1 type(s) that can be POSTed: Sel.TestAuthenticateUser

Request Structure

```
{
  "password":  Edm.String,
  "username":  Edm.String,
}
```

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "module": "Security Manager",
   "permissionTag": "AuthService.TestUserAuthenticate"
}
```

10.4 Authentication Service Groups

purpose	endpoints	methods
read	/security/authServiceGroups/ /security/authServiceGroups('id')/	GET GET
create	/security/authServiceGroups/	POST
update	/security/authServiceGroups('id')/ /security/authServiceGroups('id')/	PUT PATCH
delete	/security/authServiceGroups('id')/ /security/authServiceGroups('id')/	DELETE DELETE

10.4.1 Read authentication service groups

URL: /api/default/security/authServiceGroups/
Method: GET
OData Type: Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup
There are 1 type(s) that can be returned: Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup
The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "authService":  Edm.String (Sel.BlueFrame.Core.SecurityManager.AuthService.id),
    "displayName":  Edm.String,
    "dn":  Edm.String,
    "groupRoles":  [ Edm.String, ] (Sel.BlueFrame.Core.SecurityManager.Role.id),
    "id":  Edm.String,
  , ]
}
```

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "AuthServiceGroup",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SecurityTree"
}
```

10.4.2 Create authentication service groups

URL: /api/default/security/authServiceGroups/
Method: POST
OData Type: Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup
There are 1 type(s) that can be POSTed: Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup

Request Structure

Property	Default Value
groupRoles	[]

Table 10.12: Default Values for Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup

```
{
  "authService": Edm.String (Sel.BlueFrame.Core.SecurityManager.AuthService.id),
  "displayName": Edm.String,
  "dn": Edm.String,
  "groupRoles": [ Edm.String, ] (Sel.BlueFrame.Core.SecurityManager.Role.id),
}
```

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "AuthServiceGroup",
   "module": "DataBroker",
   "permissionTag": "Create",
   "tree": "SecurityTree"
}
```

10.4.3 Update authentication service groups (PUT)

URL: /api/default/security/authServiceGroups('id')/
Method: PUT
OData Type: Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup

Binding parameter (id): Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup.id
There are 1 type(s) that can be modified: Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup

Request Structure

The request structure is the same as the GET response above. Some fields may be omitted if not POSTable. Any POSTable fields not included will be reset to the default state.

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "AuthServiceGroup",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "SecurityTree"
}
```

10.4.4 Update authentication service groups (PATCH)

URL: /api/default/security/authServiceGroups('id')/
Method: PATCH
OData Type: Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup
Binding parameter (id): Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup.id
There are 1 type(s) that can be modified: Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup
The following fields are patchable:

- displayName
- groupRoles

Request Structure

The request structure includes one or more properties that are patchable. Any property not included in the request structure will retain its current value (subject to any property relationships).

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "AuthServiceGroup",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SecurityTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "AuthServiceGroup",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "SecurityTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "AuthServiceGroup",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "SecurityTree"
}
```

10.4.5 Delete authentication service groups

URL: /api/default/security/authServiceGroups('id')/
Method: DELETE
OData Type: Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup
Binding parameter (id): Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup.id

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "AuthServiceGroup",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "SecurityTree"
}
```

10.5 Preferences

purpose	endpoints	methods
read	/settings/GetUserPreference /settings/preferences/	POST GET
create	/settings/SetUserPreference	POST
delete	Not supported	
update	/settings/SetUserPreference	POST

10.5.1 Get the value of a user preference

URL: /api/default/settings/GetUserPreference
Method: POST
OData Type: Sel.GetUserPreference
There are 1 type(s) that can be POSTed: Sel.GetUserPreference

Request Structure

```
{
  "defaultValue":  Edm.String,
  "key":  Edm.String,
}
```

User Roles with Required Permissions

- Monitor
- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Preference",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SettingsTree"
}
```

10.5.2 Read user preferences

URL: /api/default/settings/preferences/
Method: GET
OData Type: Sel.BlueFrame.UserPreferences.Preference
There are 1 type(s) that can be returned: Sel.BlueFrame.UserPreferences.Preference The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "id":  Edm.String,
    "registry":  [ Sel.BlueFrame.UserPreferences.Setting, ],
    "username":  Edm.String,
  , ]
}
```

User Roles with Required Permissions

- Monitor
- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Preference",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SettingsTree"
}
```

10.5.3 Add or modify a user preference

URL: /api/default/settings/SetUserPreference
Method: POST
OData Type: Sel.SetUserPreference
There are 1 type(s) that can be POSTed: Sel.SetUserPreference

Request Structure

```
{
  "key":  Edm.String,
  "value":  Edm.String,
}
```

User Roles with Required Permissions

- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

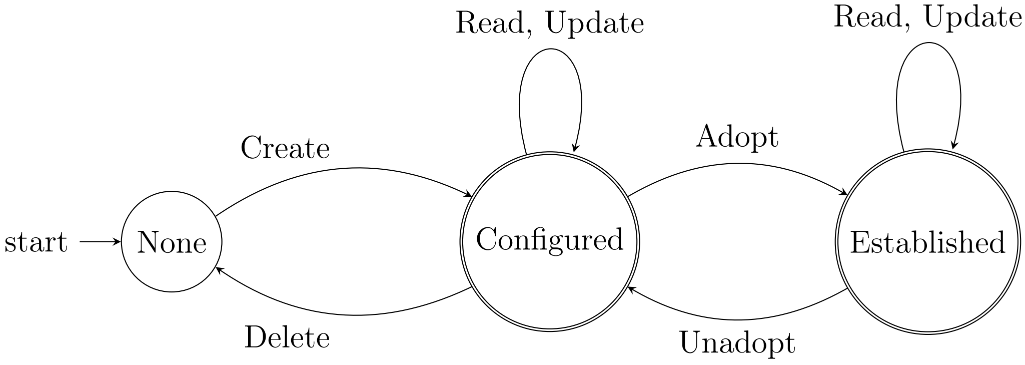
```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Preference",
   "module": "DataBroker",
   "permissionTag": "Create",
   "tree": "SettingsTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Preference",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SettingsTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Preference",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "SettingsTree"
}
```

Chapter 11

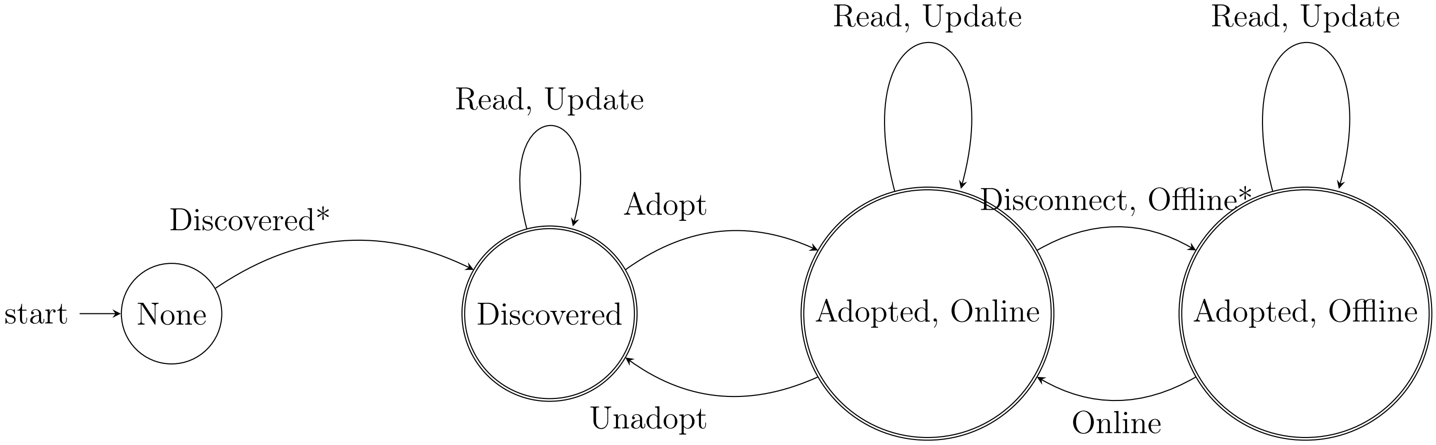
Topology Management

There are two types of objects, operational and configuration. Operational objects are those detected by the SEL-5056 through one of the discovery processes. Configuration are the abstract settings that are independent of the operational object. Configuration objects are applied to the operational object during adoption.

Configuration objects operate in one of two states: Configured and Established. Those in the configured state are created but not yet tied to an operational object.



Operational objects can be 3 states:



There are three types of objects:

- Nodes
- Ports
- Links

A node may have one or more ports, which may be connected to one or more links. These links are then connected to the port on the other side, which belongs to a node.

Starting Object	Property	Referenced Object Type
Node	ports	List of Port ids
Port	parentNode	Node
Port	attachedLinks	List of links ids
Link	endPoints	List of port ids

Node to Port to Link to Port to Node

11.1 Nodes

Nodes may be one of four types:

- Controller, representing the SEL-5056
- NetworkAbstraction, representing switches not controlled by SEL-5056
- SEL-2740S, representing the SEL-2740S
- Generic, all other nodes: including end devices and other types of OpenFlow switches.

The Controller and NetworkAbstraction are both subtypes of Generic determined by a tag within the data structure. The SEL-2740S is a derived type of Generic node.

IDing the node type

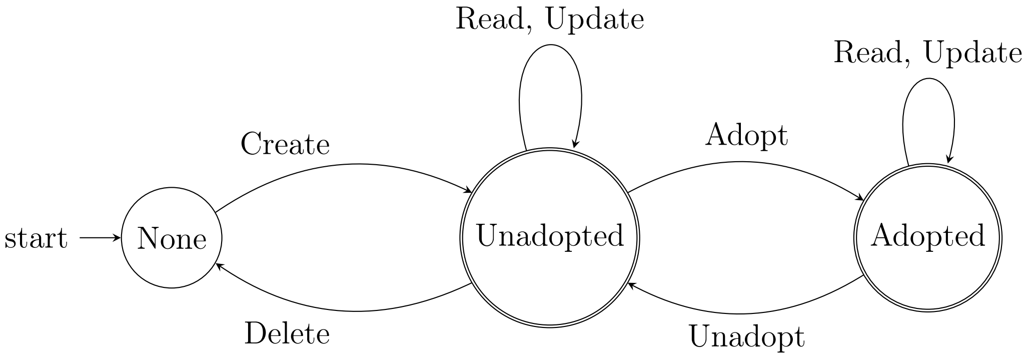
Type	Referenced Object Type
SEL2740S	Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode
NetworkAbstraction	configurations property contains { "@odata.type": "#Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractionConfig", "abstractionSetting": { "@odata.type": "#Sel.Sel5056.TopologyManager.Configurations.Abstraction.TraditionalSwitch" } }
Controller	configuration property contains { "@odata.type": "#Sel.Sel5056.TopologyManager.Configurations.ControllerConfig" }
Generic	All other nodes

SEL-2740S

flow tables : OpenFlow
group table : OpenFlow
meter table : OpenFlow
ports : OpenFlow
ptp : Management
ntp : Management
alarm contact: Management
ip interface: Management
syslog : Management

upgrade firmware()
reboot()
factory default reset()

purpose	endpoints	methods
read	/config/nodes/ /config/nodes('id')/ /operational/nodes/ /operational/nodes('id')/	GET GET GET GET
create	/config/nodes/	POST
update	/config/nodes('id')/ /config/nodes('id')/ /operational/MergeNodes /operational/nodes('id')/ReplaceConfig	PUT PATCH POST POST
delete	/config/nodes('id')/	DELETE
management	/operational/nodes('id')/Disconnect	POST
adopt	/operational/nodes('id')/Adopt /operational/nodes('id')/AdoptWithConfig	POST POST
unadopt	/operational/nodes('id')/Unadopt	POST
filtering	/operational/GetNodesByIp /operational/GetNodeByAttribute	POST POST
searching	/operational/UserInitiatedHostDiscovery	POST
ARP Management	/operational/nodes('id')/EnableControllerArpToNode /operational/nodes('id')/DisableControllerArpToNode	POST POST



11.1.1 Read configured nodes

URL: /api/default/config/nodes/
Method: GET
OData Type: Sel.Sel5056.TopologyManager.Nodes.ConfigNode
There are 2 type(s) that can be returned:

- Sel.Sel5056.TopologyManager.Nodes.ConfigNode
- Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

The return will be a list of one or more of these types show below

Response Structure for Sel.Sel5056.TopologyManager.Nodes.ConfigNode

```
{
  [
    "configurations":  [ Sel.Sel5056.TopologyManager.Configurations.AmConfig, ],
    "displayName":  Edm.String,
    "id":  Edm.String,
    "linkedKey":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.Operational-
NetworkNode.id),
    "ports":  [ Edm.String, ] (Sel.Sel5056.TopologyManager.Ports.ConfigPort.id),
    "state":  Sel.Sel5056.TopologyManager.Enums.State,
    "tags":  [ Sel.Sel5056.Tags.AmTag, ],
  , ]
}
```

Response Structure for Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

```
{
  "@odata.type":  "#Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode",
  "additionalTrustedCertificates":  [ Edm.String, ] (X.509 Thumprint),
  "alarmMinimumDuration":  Edm.Int64,
  "controllerIp":  Edm.String (IpAddress),
  "defaultGateway":  Edm.String (IpAddress),
  "enablePtp":  Edm.Boolean,
  "enableSnmp":  Edm.Boolean,
  "eventCategories":  [ Sel.BlueFrame.Core.EventBus.EventCategory, ],
  "ipAddress":  Edm.String,
  "ntpServers":  [ Edm.String, ] (ipv4address),
  "subnetMask":  Edm.String (IpAddress),
  "configurations":  [ Sel.Sel5056.TopologyManager.Configurations.AmConfig, ],
  "displayName":  Edm.String,
  "id":  Edm.String,
  "linkedKey":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.OperationalNetwork-
Node.id),
  "ports":  [ Edm.String, ] (Sel.Sel5056.TopologyManager.Ports.ConfigPort.id),
  "state":  Sel.Sel5056.TopologyManager.Enums.State,
  "tags":  [ Sel.Sel5056.Tags.AmTag, ],
}
```

Available User Roles with Required Permissions

- Monitor

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigNode",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
}
```

11.1.2 Read operational nodes

URL: /api/default/operational/nodes/
Method: GET
OData Type: Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
There are 1 type(s) that can be returned:

- Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "attributes":  [ Sel.Sel5056.TopologyManager.Attributes.AmAttribute, ],
    "displayName":  Edm.String,
    "id":  Edm.String,
    "linkedKey":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode.id),
    "ports":  [ Edm.String, ] (Sel.Sel5056.TopologyManager.Ports.-
OperationalNetworkPort.id),
    "state":  Sel.Sel5056.TopologyManager.Enums.State,
    "trustState":  Sel.Sel5056.TopologyManager.Enums.DiscoveryTrustState,
  , ]
}
```

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkNode",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

11.1.3 Create configured nodes

URL: /api/default/config/nodes/
Method: POST
OData Type: Sel.Sel5056.TopologyManager.Nodes.ConfigNode
There are 2 type(s) that can be POSTed:

- Sel.Sel5056.TopologyManager.Nodes.ConfigNode
- Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

Request Structure for Sel.Sel5056.TopologyManager.Nodes.ConfigNode

Property	Default Value
@odata.type	Not applicable
configurations	[]
displayName	null
ports	[]
tags	[]

Table 11.5: Default Values for Sel.Sel5056.TopologyManager.Nodes.ConfigNode

```
{
  "configurations":  [ Sel.Sel5056.TopologyManager.Configurations.AmConfig, ],
  "displayName":  Edm.String,
  "ports":  [ Edm.String, ] (Sel.Sel5056.TopologyManager.Ports.ConfigPort.id),
  "tags":  [ Sel.Sel5056.Tags.AmTag, ],
}
```

Request Structure for Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

Property	Default Value
additionalTrustedCertificates	[]
alarmMinimumDuration	1
configurations	[]
displayName	null
enablePtp	false
enableSnmp	false
eventCategories	[{ "behaviors": [{ "@odata.type": "#Sel.BlueFrame.Particle.AlarmContact.AlarmContactBehavior", "name": "default", "severity": "Error" }, { "@odata.type": "#Sel.BlueFrame.Particle.LocalEventStore.LocalEventStoreBehavior", "name": "default", "severity": "Informational" }], "id": "default", "key": "default" }, { "behaviors": [{ "@odata.type": "#Sel.BlueFrame.Particle.AlarmContact.AlarmContactBehavior", "name": "default", "severity": "Notice" }], "id": "Openflow", "key": "Openflow" }, { "behaviors": [{ "@odata.type": "#Sel.BlueFrame.Particle.AlarmContact.AlarmContactBehavior", "name": "default", "severity": "Notice" }], "id": "Link", "key": "Link" }, { "behaviors": [], "id": "Security", "key": "Security" }, { "behaviors": [], "id": "Configuration", "key": "Configuration" }, { "behaviors": [], "id": "System Integrity", "key": "System Integrity" }, { "behaviors": [], "id": "Chassis and Module", "key": "Chassis and Module" }]
ntpServers	[]
ports	[]
tags	[]

Table 11.7: Default Values for Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

```
{
  "@odata.type": "#Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode",
  "additionalTrustedCertificates": [ Edm.String, ] (X.509 Thumprint),
  "alarmMinimumDuration": Edm.Int64,
  "controllerIp": Edm.String (IpAddress),
  "defaultGateway": Edm.String (IpAddress),
  "enablePtp": Edm.Boolean,
  "enableSnmp": Edm.Boolean,
  "eventCategories": [ Sel.BlueFrame.Core.EventBus.EventCategory, ],
  "ipAddress": Edm.String,
  "ntpServers": [ Edm.String, ] (ipv4address),
  "subnetMask": Edm.String (IpAddress),
  "configurations": [ Sel.Sel5056.TopologyManager.Configurations.AmConfig, ],
  "displayName": Edm.String,
  "ports": [ Edm.String, ] (Sel.Sel5056.TopologyManager.Ports.ConfigPort.id),
  "tags": [ Sel.Sel5056.Tags.AmTag, ],
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigNode",
   "module": "DataBroker",
   "permissionTag": "Create",
   "tree": "ConfigTree"
}
```

11.1.4 Update configured nodes (PUT)

URL: /api/default/config/nodes('id')/
Method: PUT
OData Type: Sel.Sel5056.TopologyManager.Nodes.ConfigNode
Binding parameter (id): Sel.Sel5056.TopologyManager.Nodes.ConfigNode.id
There are 2 type(s) that can be modified:

- Sel.Sel5056.TopologyManager.Nodes.ConfigNode
- Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

Request Structure for Sel.Sel5056.TopologyManager.Nodes.ConfigNode

The request structure is the same as the GET response above. Some fields may be omitted if not POSTable. Any POSTable fields not included will be reset to the default state.

Request Structure for Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

The request structure is the same as the GET response above. Some fields may be omitted if not POSTable. Any POSTable fields not included will be reset to the default state.

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigNode",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "ConfigTree"
}
```

11.1.5 Update configured nodes (PATCH)

URL: /api/default/config/nodes('id')/
Method: PATCH
OData Type: Sel.Sel5056.TopologyManager.Nodes.ConfigNode
Binding parameter (id): Sel.Sel5056.TopologyManager.Nodes.ConfigNode.id
There are 2 type(s) that can be modified:

- Sel.Sel5056.TopologyManager.Nodes.ConfigNode
- Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

The following properties are patchable for type Sel.Sel5056.TopologyManager.Nodes.ConfigNode

- configurations
- displayName
- ports
- tags

The following properties are patchable for type Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

- additionalTrustedCertificates
- alarmMinimumDuration
- configurations
- controllerIp
- defaultGateway
- displayName
- enablePtp
- enableSnmp
- eventCategories
- ipAddress
- ntpServers
- ports
- subnetMask
- tags

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigNode",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigNode",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigNode",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "ConfigTree"
}
```

11.1.6 Merge two nodes by combining the ports of one into the other

URL: /api/default/operational/MergeNodes
Method: POST
OData Type: Sel.MergeNodes
There are 1 type(s) that can be POSTed:

- Sel.MergeNodes

Request Structure

```
{
  "firstOperationalNodeId":  Edm.String,
  "secondOperationalNodeId":  Edm.String,
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkNode",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
},
{
   "module": "TopologyManager",
   "permissionTag": "TopologyDiscoveryHinting"
},
{
   "module": "Synchronization",
   "permissionTag": "PerformSynchronization"
}
```

11.1.7 Replace the configuration object for an adopted node

URL: /api/default/operational/nodes('id')/ReplaceConfig
Method: POST
OData Type: Sel.ReplaceConfig
Binding parameter (id): Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id
There are 1 type(s) that can be POSTed:

- Sel.ReplaceConfig

Request Structure

```
{
  "configKey":  Edm.String,
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "TopologyManager",
   "permissionTag": "ReplaceObject"
}
```

11.1.8 Delete configured nodes

URL: /api/default/config/nodes('id')/
Method: DELETE
OData Type: Sel.Sel5056.TopologyManager.Nodes.ConfigNode
Binding parameter (id): Sel.Sel5056.TopologyManager.Nodes.ConfigNode.id

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigNode",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "ConfigTree"
}
```

11.1.9 Remove a node entry

URL: /api/default/operational/nodes('id')/Disconnect
Method: POST
OData Type: Sel.Disconnect
Binding parameter (id): Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id
There are 1 type(s) that can be POSTed:

- Sel.Disconnect

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkNode",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
},
{
   "module": "TopologyManager",
   "permissionTag": "DisconnectObject"
},
{
   "module": "TopologyManager",
   "permissionTag": "ReplaceObject"
}
```

11.1.10 Adopt a node with default settings

URL: /api/default/operational/nodes('id')/Adopt
Method: POST
OData Type: Sel.Adopt
Binding parameter (id): Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id
There are 1 type(s) that can be POSTed:

- Sel.Adopt

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "TopologyManager",
   "permissionTag": "AdoptObject"
}
```

11.1.11 Adopt a node with a preconfigured node object

URL: /api/default/operational/nodes('id')/AdoptWithConfig
Method: POST
OData Type: Sel.AdoptWithConfig
Binding parameter (id): Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id
There are 1 type(s) that can be POSTed:

- Sel.AdoptWithConfig

Request Structure

```
{
  "configKey":  Edm.String,
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "TopologyManager",
   "permissionTag": "AdoptObject"
}
```

11.1.12 Unadopt a node

URL: /api/default/operational/nodes('id')/Unadopt
Method: POST
OData Type: Sel.Unadopt
Binding parameter (id): Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id
There are 1 type(s) that can be POSTed:

- Sel.Unadopt

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "TopologyManager",
   "permissionTag": "UnadoptObject"
}
```

11.1.13 Get nodes by IP address

URL: /api/default/operational/GetNodesByIp
Method: POST
OData Type: Sel.GetNodesByIp
There are 1 type(s) that can be POSTed:

- Sel.GetNodesByIp

Request Structure

```
{
  "ip":  Edm.String,
}
```

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

None

11.1.14 Get nodes by an attribute

URL: /api/default/operational/GetNodeByAttribute
Method: POST
OData Type: Sel.GetNodeByAttribute
There are 1 type(s) that can be POSTed:

- Sel.GetNodeByAttribute

Request Structure

```
{
  "attribute": Sel.Sel5056.TopologyManager.Attributes.AmAttribute,
}
```

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkNode",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

11.1.15 Run host discovery for a particular IP address

URL: /api/default/operational/UserInitiatedHostDiscovery
Method: POST
OData Type: Sel.UserInitiatedHostDiscovery
There are 1 type(s) that can be POSTed:

- Sel.UserInitiatedHostDiscovery

Request Structure

```
{
  "ip": Edm.String,
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "TopologyManager",
   "permissionTag": "TopologyDiscoveryHinting"
}
```

11.1.16 Prevent the controller from sending arps to the node

URL: /api/default/operational/nodes('id')/EnableControllerArpToNode
Method: POST
OData Type: Sel.EnableControllerArpToNode
Binding parameter (id): Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id
There are 1 type(s) that can be POSTed:

- Sel.EnableControllerArpToNode

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkNode",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
},
{
   "module": "TopologyManager",
   "permissionTag": "TopologyDiscoveryHinting"
},
{
   "module": "Synchronization",
   "permissionTag": "PerformSynchronization"
}
```

11.1.17 Allow the controller to send arps to the node

URL: /api/default/operational/nodes('id')/DisableControllerArpToNode
Method: POST
OData Type: Sel.DisableControllerArpToNode
Binding parameter (id): Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id
There are 1 type(s) that can be POSTed:

- Sel.DisableControllerArpToNode

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkNode",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
},
{
   "module": "TopologyManager",
   "permissionTag": "TopologyDiscoveryHinting"
},
{
   "module": "Synchronization",
   "permissionTag": "PerformSynchronization"
}
```

11.2 OpenFlow Switches

purpose	endpoints	methods
Offline/Virtual Hosts	/operational/ports('id')/AddSilentHost	POST

11.2.1 Added an offline host to a port

URL: /api/default/operational/ports('id')/AddSilentHost
Method: POST
OData Type: Sel.AddSilentHost
Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id
There are 1 type(s) that can be POSTed:

- Sel.AddSilentHost

Request Structure

```
{
  "configNodeId":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode),
}
```

Available User Roles with Required Permissions

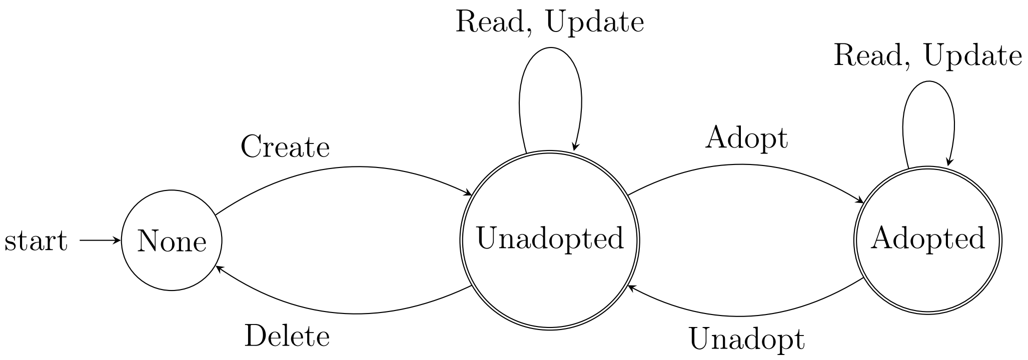
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkPort",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
},
{
   "module": "TopologyManager",
   "permissionTag": "TopologyDiscoveryHinting"
},
{
   "module": "Synchronization",
   "permissionTag": "PerformSynchronization"
}
```

11.3 SEL-2740S Switches

purpose	endpoints	methods
synchronization	/operational/GetNodesThatRequireSynchronization /operational/synchronization/	POST GET
device management	/operational/nodes('id')/FactoryDefaultReset /operational/nodes('id')/FirmwareUpgrade /operational/nodes('id')/Reboot /operational/nodes('id')/ReplanInbandPath	POST POST POST POST
diagnostics	/operational/tableStats/ /operational/tableStats('id')/	GET GET



11.3.1 Get nodes that are unsynchronized

URL: /api/default/operational/GetNodesThatRequireSynchronization
Method: POST
OData Type: Sel.GetNodesThatRequireSynchronization
There are 1 type(s) that can be POSTed:

- Sel.GetNodesThatRequireSynchronization

Request Structure

No request body

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "SynchronizationRequest",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

11.3.2 Read switches to synchronize

URL: /api/default/operational/synchronization/
Method: GET
OData Type: Sel.Sel5056.TopologyManager.Synchronization.SynchronizationRequest
There are 12 type(s) that can be returned:

- Sel.Sel5056.TopologyManager.Synchronization.SynchronizationRequest
- Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowAddSynchronizationRequest

- Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowDeleteSynchronizationRequest
- Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModifySynchronizationRequest
- Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupAddSynchronizationRequest
- Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupDeleteSynchronizationRequest
- Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupModifySynchronizationRequest
- Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterAddSynchronizationRequest
- Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterDeleteSynchronizationRequest
- Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterModifySynchronizationRequest
- Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization
- Sel.Sel5056.SapphirePlugin.Synchronization.FirmwareUpgradeRequiredSynchronization

The return will be a list of one or more of these types show below

Response Structure for Sel.Sel5056.TopologyManager.Synchronization.SynchronizationRequest

```
{
  [
    "configNodeId":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode),
    "description":  Edm.String,
    "errorState":  Sel.BlueFrame.Core.ErrorState,
    "errors":  [ Edm.String, ],
    "id":  Edm.String,
    "version":  Edm.Int32,
  , ]
}
```

Response Structure for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowAddSynchronizationRequest

```
{
  "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests-
.FlowAddSynchronizationRequest",   "alias":  Edm.String,
  "configFlowId":  Edm.String (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow),
  "cookie":  Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flowcookie),
  "dependencies":  [ Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.Open-
FlowDependency, ],
  "priorityConfiguration":  Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-
SystemPriorities.AmSystemPriority,
  "configNodeId":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode),
  "description":  Edm.String,
  "errorState":  Sel.BlueFrame.Core.ErrorState,
  "errors":  [ Edm.String, ],
  "id":  Edm.String,
  "version":  Edm.Int32,
}
```

Response Structure for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowDeleteSynchronizationRequest

```
{
  "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests-
.FlowDeleteSynchronizationRequest",   "alias":  Edm.String,
  "cookie":  Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flowcookie),
  "dependencies":  [ Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.Open-
FlowDependency, ],
  "priorityConfiguration":  Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-
SystemPriorities.AmSystemPriority,
  "referencedGroupId":  Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
GroupgroupId),
  "configNodeId":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode),
  "description":  Edm.String,
  "errorState":  Sel.BlueFrame.Core.ErrorState,
  "errors":  [ Edm.String, ],
  "id":  Edm.String,
  "version":  Edm.Int32,
}
```

Response Structure for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModifySynchronizationRequest

```
{
  "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests-
.FlowModifySynchronizationRequest",   "alias":  Edm.String,
  "configFlowId":  Edm.String (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow),
  "cookie":  Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flowcookie),
  "currentGroupId":  Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
GroupgroupId),
  "dependencies":  [ Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.Open-
FlowDependency, ],
  "newGroupId":  Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
GroupgroupId),
  "priorityConfiguration":  Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-
SystemPriorities.AmSystemPriority,
  "configNodeId":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode),
  "description":  Edm.String,
  "errorState":  Sel.BlueFrame.Core.ErrorState,
  "errors":  [ Edm.String, ],
  "id":  Edm.String,
  "version":  Edm.Int32,
}
```

Response Structure for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.-Requests.GroupAddSynchronizationRequest

```
{
  "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests-
.GroupAddSynchronizationRequest",  "alias":  Edm.String,
  "configGroupId":  Edm.String (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group),
  "dependencies":  [ Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.Open-
FlowDependency, ],
  "groupId":  Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupgroupId),
  "priorityConfiguration":  Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-
SystemPriorities.AmSystemPriority,
  "configNodeId":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode),
  "description":  Edm.String,
  "errorState":  Sel.BlueFrame.Core.ErrorState,
  "errors":  [ Edm.String, ],
  "id":  Edm.String,
  "version":  Edm.Int32,
}
```

Response Structure for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.-Requests.GroupDeleteSynchronizationRequest

```
{
  "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests-
.GroupDeleteSynchronizationRequest",  "alias":  Edm.String,
  "dependencies":  [ Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.Open-
FlowDependency, ],
  "groupId":  Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupgroupId),
  "priorityConfiguration":  Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-
SystemPriorities.AmSystemPriority,
  "configNodeId":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode),
  "description":  Edm.String,
  "errorState":  Sel.BlueFrame.Core.ErrorState,
  "errors":  [ Edm.String, ],
  "id":  Edm.String,
  "version":  Edm.Int32,
}
```

Response Structure for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupModifySynchronizationRequest

```
{
  "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests-
.GroupModifySynchronizationRequest",   "alias":  Edm.String,
  "configGroupId":  Edm.String (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group),
  "currentGroupIds":  [ Edm.Int64, ] (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
GroupgroupId),
  "dependencies":  [ Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.Open-
FlowDependency, ],
  "groupId":  Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupgroupId),
  "newGroupIds":  [ Edm.Int64, ] (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
GroupgroupId),
  "priorityConfiguration":  Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-
SystemPriorities.AmSystemPriority,
  "configNodeId":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode),
  "description":  Edm.String,
  "errorState":  Sel.BlueFrame.Core.ErrorState,
  "errors":  [ Edm.String, ],
  "id":  Edm.String,
  "version":  Edm.Int32,
}
```

Response Structure for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterAddSynchronizationRequest

```
{
  "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests-
.MeterAddSynchronizationRequest",   "alias":  Edm.String,
  "configMeterId":  Edm.String (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter),
  "dependencies":  [ Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.Open-
FlowDependency, ],
  "meterId":  Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MetermeterId),
  "priorityConfiguration":  Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-
SystemPriorities.AmSystemPriority,
  "configNodeId":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode),
  "description":  Edm.String,
  "errorState":  Sel.BlueFrame.Core.ErrorState,
  "errors":  [ Edm.String, ],
  "id":  Edm.String,
  "version":  Edm.Int32,
}
```

Response Structure for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterDeleteSynchronizationRequest

```
{
  "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests-
.MeterDeleteSynchronizationRequest",   "alias":  Edm.String,
  "dependencies":  [ Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.Open-
FlowDependency, ],
  "meterId":  Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MetermeterId),
  "priorityConfiguration":  Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-
SystemPriorities.AmSystemPriority,
  "configNodeId":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode),
  "description":  Edm.String,
  "errorState":  Sel.BlueFrame.Core.ErrorState,
  "errors":  [ Edm.String, ],
  "id":  Edm.String,
  "version":  Edm.Int32,
}
```

Response Structure for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterModifySynchronizationRequest

```
{
  "@odata.type":  "#Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests-
.MeterModifySynchronizationRequest",   "alias":  Edm.String,
  "configMeterId":  Edm.String (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter),
  "dependencies":  [ Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.Open-
FlowDependency, ],
  "meterId":  Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MetermeterId),
  "priorityConfiguration":  Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-
SystemPriorities.AmSystemPriority,
  "configNodeId":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode),
  "description":  Edm.String,
  "errorState":  Sel.BlueFrame.Core.ErrorState,
  "errors":  [ Edm.String, ],
  "id":  Edm.String,
  "version":  Edm.Int32,
}
```

Response Structure for Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization

```
{
  "@odata.type":  "#Sel.Sel5056.SapphirePlugin.Synchronization-
.SapphireConfigSynchronization",   "endpoint":  Sel.Sel5056.SapphirePlugin.-
Synchronization.SapphireEndpoint,
  "key":  Edm.String,
  "configNodeId":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode),
  "description":  Edm.String,
  "errorState":  Sel.BlueFrame.Core.ErrorState,
  "errors":  [ Edm.String, ],
  "id":  Edm.String,
  "version":  Edm.Int32,
}
```

Response Structure for Sel.Sel5056.SapphirePlugin.Synchronization.FirmwareUpgradeRequiredSynchronization

```
{
  "@odata.type":  "#Sel.Sel5056.SapphirePlugin.Synchronization-
.FirmwareUpgradeRequiredSynchronization",   "configNodeId":  Edm.String (Sel.-
Sel5056.TopologyManager.Nodes.ConfigNode),
  "description":  Edm.String,
  "errorState":  Sel.BlueFrame.Core.ErrorState,
  "errors":  [ Edm.String, ],
  "id":  Edm.String,
  "version":  Edm.Int32,
}
```

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "SynchronizationRequest",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

11.3.3 Factory default reset a SEL-2740S

URL: /api/default/operational/nodes('id')/FactoryDefaultReset
Method: POST
OData Type: Sel.FactoryDefaultReset
Binding parameter (id): Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id
There are 1 type(s) that can be POSTed:

- Sel.FactoryDefaultReset

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission",
   "deviceType": "SapphireDevice",
   "module": "DeviceManagement",
   "permissionTag": "FactoryDefaultReset"
},
{
   "@odata.type": "#Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission",
   "deviceType": "SapphireDevice",
   "module": "DeviceManagement",
   "permissionTag": "Administrator"
},
{
   "@odata.type": "#Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission",
   "deviceType": "SapphireDevice",
   "module": "DeviceManagement",
   "permissionTag": "Engineer"
}
```

11.3.4 Firmware upgrade a SEL-2740S

URL: /api/default/operational/nodes('id')/FirmwareUpgrade
Method: POST
OData Type: Sel.FirmwareUpgrade
Binding parameter (id): Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id
There are 1 type(s) that can be POSTed:

- Sel.FirmwareUpgrade

Request Structure

Firmware file upload uses the multipart request message format (RFC 7578).

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission",
   "deviceType": "SapphireDevice",
   "module": "DeviceManagement",
   "permissionTag": "FirmwareUpgrade"
},
{
   "@odata.type": "#Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission",
   "deviceType": "SapphireDevice",
   "module": "DeviceManagement",
   "permissionTag": "Engineer"
}
```

Examples

Upgrade a SEL-274XS Example 9.

```
POST /api/default/operational/nodes('
   aa4974f85c0424fe98c879164d871082')/Sel.FirmwareUpgrade HTTP/1.1
Content-Type: multipart/form-data; boundary
   =---------------------------194619196719097754702258645198
HOST: 192.168.56.1
Authorization: Bearer
   a915eb55bd8d304db3762d4f0b440b7d5b796c22869c2d9cd7c343ad51bd9583
Content-Length: 270

-----------------------------194619196719097754702258645198
Content-Disposition: form-data; name="fileName"; filename="firmware.
   img"
Content-Type: application/octet-stream

<--Firmware file contents-->
-----------------------------194619196719097754702258645198--
```

Listing 11.1: Upgrade a SEL-274XS

11.3.5 Reboot a SEL-2740S

URL: /api/default/operational/nodes('id')/Reboot

Method: POST
OData Type: Sel.Reboot
Binding parameter (id): Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id
There are 1 type(s) that can be POSTed:

- Sel.Reboot

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission",
   "deviceType": "SapphireDevice",
   "module": "DeviceManagement",
   "permissionTag": "Reboot"
},
{
   "@odata.type": "#Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission",
   "deviceType": "SapphireDevice",
   "module": "DeviceManagement",
   "permissionTag": "Administrator"
},
{
   "@odata.type": "#Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission",
   "deviceType": "SapphireDevice",
   "module": "DeviceManagement",
   "permissionTag": "Engineer"
}
```

11.3.6 Replan the IB management logical connection to a switch

URL: /api/default/operational/nodes('id')/ReplanInbandPath
Method: POST
OData Type: Sel.ReplanInbandPath
Binding parameter (id): Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id
There are 1 type(s) that can be POSTed:

- Sel.ReplanInbandPath

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission",
   "deviceType": "SapphireDevice",
   "module": "DeviceManagement",
   "permissionTag": "ReplanControllerRoute"
}
```

11.3.7 Read flow table diagnostics

URL: /api/default/operational/tableStats/
Method: GET
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats
There are 1 type(s) that can be returned:

- Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats

The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "dataPathId":  Edm.Int64,
    "id":  Edm.String,
    "tableStatList":  [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStat, ],
  , ]
}
```

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "TableStats",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

11.4 Traditional Switches

purpose	endpoints	methods
create	/operational/ports('id')/AddAbstractNode	POST

11.4.1 Add a traditional switch

URL: /api/default/operational/ports('id')/AddAbstractNode
Method: POST
OData Type: Sel.AddAbstractNode
Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id
There are 1 type(s) that can be POSTed:

- Sel.AddAbstractNode

Request Structure

```
{
  "configSetting":  Sel.Sel5056.TopologyManager.Configurations.Abstraction.-
AbstractSettingBase,
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkPort",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
},
{
   "module": "TopologyManager",
   "permissionTag": "TopologyDiscoveryHinting"
},
{
   "module": "Synchronization",
   "permissionTag": "PerformSynchronization"
}
```

11.5 Ports

purpose	endpoints	methods
read	/config/ports/ /config/ports('id')/ /operational/ports/ /operational/ports('id')/	GET GET GET GET
create	/config/ports/	POST
update	/config/ports('id')/ /config/ports('id')/ /operational/ports('id')/ReplaceConfig	PUT PATCH POST
delete	/config/ports('id')/	DELETE
adopt	/operational/ports('id')/Adopt /operational/ports('id')/AdoptWithConfig	POST POST
unadopt	/operational/ports('id')/Unadopt	POST
failover mode	/operational/ports('id')/DetectSelRelayFailoverPort /operational/ports('id')/MarkPortAsSelRelayFailover /operational/ports('id')/MarkPortAndAddRelayFailoverLink /operational/ports('id')/MarkPortAndPerformBlockingDetectSelRelayFailover /operational/ports('id')/RemoveSelRelayFailoverFromPort	POST POST POST POST POST
search	/operational/GetPortByAttribute	POST

11.5.1 Read configured ports

URL: /api/default/config/ports/
Method: GET
OData Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort
There are 1 type(s) that can be returned:

- Sel.Sel5056.TopologyManager.Ports.ConfigPort

The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "attachedLinks":  [ Edm.String, ] (Sel.Sel5056.TopologyManager.Links.Config-
Link),
    "configurations":  [ Sel.Sel5056.TopologyManager.Configurations.AmConfig, ],
    "displayName":  Edm.String,
    "displayNameSetByUser":  Edm.Boolean,
    "id":  Edm.String,
    "linkedKey":  Edm.String (Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id),
    "parentNode":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode.id),
    "state":  Sel.Sel5056.TopologyManager.Enums.State,
    "tags":  [ Sel.Sel5056.Tags.AmTag, ],
  , ]
}
```

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigPort",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
}
```

11.5.2 Read detected ports

URL: /api/default/operational/ports/
Method: GET
OData Type: Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
There are 1 type(s) that can be returned:

- Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "attachedLinks":  [ Edm.String, ] (Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink),
    "attributes":  [ Sel.Sel5056.TopologyManager.Attributes.AmAttribute, ],
    "displayName":  Edm.String,
    "id":  Edm.String,
    "isConnected":  Edm.Boolean,
    "linkedKey":  Edm.String (Sel.Sel5056.TopologyManager.Ports.ConfigPort.id),
    "parentNode":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.Operational-
NetworkNode),
    "state":  Sel.Sel5056.TopologyManager.Enums.State,
    "trustState":  Sel.Sel5056.TopologyManager.Enums.DiscoveryTrustState,
  , ]
}
```

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkPort",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

11.5.3 Create configured ports

URL: /api/default/config/ports/
Method: POST
OData Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort
There are 1 type(s) that can be POSTed:

- Sel.Sel5056.TopologyManager.Ports.ConfigPort

Request Structure

Property	Default Value
attachedLinks	[]
configurations	[]
displayName	null
displayNameSetByUser	true
tags	[]

Table 11.13: Default Values for Sel.Sel5056.TopologyManager.Ports.ConfigPort

```
{
  "attachedLinks":  [ Edm.String, ] (Sel.Sel5056.TopologyManager.Links.Config-
Link),
  "configurations": [ Sel.Sel5056.TopologyManager.Configurations.AmConfig, ],
  "displayName": Edm.String,
  "displayNameSetByUser": Edm.Boolean,
  "parentNode": Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode.id),
  "tags": [ Sel.Sel5056.Tags.AmTag, ],
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigPort",
   "module": "DataBroker",
   "permissionTag": "Create",
   "tree": "ConfigTree"
}
```

11.5.4 Update configured ports (PUT)

URL: /api/default/config/ports('id')/
Method: PUT
OData Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort
Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.ConfigPort.id
There are 1 type(s) that can be modified:

- Sel.Sel5056.TopologyManager.Ports.ConfigPort

Request Structure

The request structure is the same as the GET response above. Some fields may be omitted if not POSTable. Any POSTable fields not included will be reset to the default state.

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigPort",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "ConfigTree"
}
```

11.5.5 Update configured ports (PATCH)

URL: /api/default/config/ports('id')/
Method: PATCH
OData Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort
Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.ConfigPort.id
There are 1 type(s) that can be modified:

- Sel.Sel5056.TopologyManager.Ports.ConfigPort

The following fields are patchable:

- configurations
- displayName
- displayNameSetByUser
- parentNode
- tags

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigPort",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigPort",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigPort",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "ConfigTree"
}
```

11.5.6 Replace the configuration object for an adopted port

URL: /api/default/operational/ports('id')/ReplaceConfig
Method: POST
OData Type: Sel.ReplaceConfig
Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id
There are 1 type(s) that can be POSTed:

- Sel.ReplaceConfig

Request Structure

```
{
  "configKey":  Edm.String,
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "TopologyManager",
   "permissionTag": "ReplaceObject"
}
```

11.5.7 Delete configured ports

URL: /api/default/config/ports('id')/
Method: DELETE
OData Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort
Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.ConfigPort.id

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigPort",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "ConfigTree"
}
```

11.5.8 Get list of nodes that are unsynchronized

URL: /api/default/operational/ports('id')/Adopt
Method: POST
OData Type: Sel.Adopt
Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id
There are 1 type(s) that can be POSTed:

- Sel.Adopt

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "TopologyManager",
   "permissionTag": "AdoptObject"
}
```

11.5.9 Adopt a port with a preconfigured port object

URL: /api/default/operational/ports('id')/AdoptWithConfig
Method: POST
OData Type: Sel.AdoptWithConfig
Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id
There are 1 type(s) that can be POSTed:

- Sel.AdoptWithConfig

Request Structure

```
{
  "configKey":  Edm.String,
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "TopologyManager",
   "permissionTag": "AdoptObject"
}
```

11.5.10 Unadopt a port

URL: /api/default/operational/ports('id')/Unadopt
Method: POST
OData Type: Sel.Unadopt
Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id
There are 1 type(s) that can be POSTed:

- Sel.Unadopt

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "TopologyManager",
   "permissionTag": "UnadoptObject"
}
```

11.5.11 Detect the other port of an SEL relay device

URL: /api/default/operational/ports('id')/DetectSelRelayFailoverPort
Method: POST
OData Type: Sel.DetectSelRelayFailoverPort
Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id
There are 1 type(s) that can be POSTed:

- Sel.DetectSelRelayFailoverPort

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkPort",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
},
{
   "module": "TopologyManager",
   "permissionTag": "TopologyDiscoveryHinting"
},
{
   "module": "OpenFlowPlugin",
   "permissionTag": "SetFlowMod"
},
{
   "module": "OpenFlowPlugin",
   "permissionTag": "PerformSynchronization"
}
```

11.5.12 Add SEL relay failover mode to a port

URL: /api/default/operational/ports('id')/MarkPortAsSelRelayFailover
Method: POST
OData Type: Sel.MarkPortAsSelRelayFailover
Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id
There are 1 type(s) that can be POSTed:

- Sel.MarkPortAsSelRelayFailover

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "TopologyManager",
   "permissionTag": "TopologyDiscoveryHinting"
}
```

11.5.13 Same as MarkPortAsSelRelayFailover and AddOperationalLinkFromHostToSwitch together

URL: /api/default/operational/ports('id')/MarkPortAndAddRelayFailoverLink
Method: POST
OData Type: Sel.MarkPortAndAddRelayFailoverLink
Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id
There are 1 type(s) that can be POSTed:

- Sel.MarkPortAndAddRelayFailoverLink

Request Structure

```
{
  "switchPortId":  Edm.String,
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "TopologyManager",
   "permissionTag": "TopologyDiscoveryHinting"
}
```

11.5.14 Same as MarkPortAndAddRelayFailoverLink except blocks until the link is discovered or timed out

URL: /api/default/operational/ports('id')/MarkPortAndPerformBlockingDetectSelRelayFailover
Method: POST
OData Type: Sel.MarkPortAndPerformBlockingDetectSelRelayFailover
Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id
There are 1 type(s) that can be POSTed:

- Sel.MarkPortAndPerformBlockingDetectSelRelayFailover

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkPort",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
},
{
   "module": "TopologyManager",
   "permissionTag": "TopologyDiscoveryHinting"
},
{
   "module": "OpenFlowPlugin",
   "permissionTag": "SetFlowMod"
},
{
   "module": "OpenFlowPlugin",
   "permissionTag": "PerformSynchronization"
}
```

11.5.15 Remove SEL relay failover mode from a port

URL: /api/default/operational/ports('id')/RemoveSelRelayFailoverFromPort
Method: POST
OData Type: Sel.RemoveSelRelayFailoverFromPort
Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id
There are 1 type(s) that can be POSTed:

- Sel.RemoveSelRelayFailoverFromPort

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "TopologyManager",
   "permissionTag": "TopologyDiscoveryHinting"
}
```

11.5.16 Filter ports by a port attribute

URL: /api/default/operational/GetPortByAttribute

Method: POST
OData Type: Sel.GetPortByAttribute
There are 1 type(s) that can be POSTed:

- Sel.GetPortByAttribute

Request Structure

```
{
  "attribute":  Sel.Sel5056.TopologyManager.Attributes.AmAttribute,
}
```

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkPort",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

11.6 OpenFlow Ports

purpose	endpoints	methods
descriptions	/operational/portDesc/ /operational/portDesc('id')/	GET GET
update	/config/ports('id')/SendPortMod	POST

11.6.1 Read OpenFlow port descriptions

URL: /api/default/operational/portDesc/
Method: GET
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc
There are 1 type(s) that can be returned:

- Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc

The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "dataPathId":  Edm.Int64,
    "id":  Edm.String,
    "ports":  [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port, ],
  , ]
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "PortDesc",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

11.6.2 Modify the OpenFlow settings of a port

URL: /api/default/config/ports('id')/SendPortMod
Method: POST
OData Type: Sel.SendPortMod
Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.ConfigPort.id
There are 1 type(s) that can be POSTed:

- Sel.SendPortMod

Request Structure

```
{
  "configFlags":  [ Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
ConfigFlagObjects.ConfigBase, ],
  "featuresFlags":  [ Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.FeatureBase, ],
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "OpenFlowPlugin",
   "permissionTag": "SetFlowMod"
}
```

11.7 SEL-2740S Ports

purpose	endpoints	methods
diagnostics	/operational/portStats/ /operational/portStats('id')/	GET GET

11.7.1 Read OpenFlow port diagnostics

URL: /api/default/operational/portStats/
Method: GET
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats
There are 1 type(s) that can be returned:

- Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats

The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "dataPathId":  Edm.Int64,
    "id":  Edm.String,
    "portStatList":  [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStat, ],
  , ]
}
```

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "PortStats",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

11.8 Links

purpose	endpoints	methods
read	/config/links/ /config/links('id')/ /operational/links/ /operational/links('id')/	GET GET GET GET
create	/config/links/ /operational/ports('id')/AddOperationalLinkFromHostToSwitch	POST POST
update	/config/links('id')/ /config/links('id')/ /operational/links('id')/ReplaceConfig	PUT PATCH POST
delete	/config/links('id')/	DELETE
adopt	/operational/links('id')/Adopt /operational/links('id')/AdoptWithConfig	POST POST
unadopt	/operational/links('id')/Unadopt	POST
search	/operational/GetLinkByPortAttribute	POST

11.8.1 Read configured links

URL: /api/default/config/links/
Method: GET
OData Type: Sel.Sel5056.TopologyManager.Links.ConfigLink
There are 1 type(s) that can be returned:

- Sel.Sel5056.TopologyManager.Links.ConfigLink

The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "configurations":  [ Sel.Sel5056.TopologyManager.Configurations.AmConfig, ],
    "displayName":  Edm.String,
    "endPoints":  [ Edm.String, ] (Sel.Sel5056.TopologyManager.Ports.ConfigPort.id),
    "first":  Edm.String (Sel.Sel5056.TopologyManager.Ports.ConfigPort),
    "id":  Edm.String,
    "last":  Edm.String (Sel.Sel5056.TopologyManager.Ports.ConfigPort),
    "linkedKey":  Edm.String (Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.id),
    "state":  Sel.Sel5056.TopologyManager.Enums.State,
    "tags":  [ Sel.Sel5056.Tags.AmTag, ],
    "weight":  Edm.Int64,
  , ]
}
```

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigLink",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
}
```

11.8.2 Read detected links

URL: /api/default/operational/links/
Method: GET
OData Type: Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink
There are 1 type(s) that can be returned:

- Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "attributes":  [ Sel.Sel5056.TopologyManager.Attributes.AmAttribute, ],
    "displayName":  Edm.String,
    "endPoints":  [ Edm.String, ] (Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort),
    "first":  Edm.String (Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort),
    "id":  Edm.String,
    "last":  Edm.String (Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort),
    "linkedKey":  Edm.String (Sel.Sel5056.TopologyManager.Links.ConfigLink.id),
    "state":  Sel.Sel5056.TopologyManager.Enums.State,
    "trustState":  Sel.Sel5056.TopologyManager.Enums.DiscoveryTrustState,
  , ]
}
```

Available User Roles with Required Permissions

- Monitor

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkLink",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

11.8.3 Create configured links

URL: /api/default/config/links/
Method: POST
OData Type: Sel.Sel5056.TopologyManager.Links.ConfigLink
There are 1 type(s) that can be POSTed:

- Sel.Sel5056.TopologyManager.Links.ConfigLink

Request Structure

Property	Default Value
configurations	[]
displayName	null
first	a
last	z
tags	[]
weight	1

Table 11.18: Default Values for Sel.Sel5056.TopologyManager.Links.ConfigLink

```
{
  "configurations": [ Sel.Sel5056.TopologyManager.Configurations.AmConfig, ],
  "displayName": Edm.String,
  "endPoints": [ Edm.String, ] (Sel.Sel5056.TopologyManager.Ports.ConfigPort.id),
  "first": Edm.String (Sel.Sel5056.TopologyManager.Ports.ConfigPort),
  "last": Edm.String (Sel.Sel5056.TopologyManager.Ports.ConfigPort),
  "tags": [ Sel.Sel5056.Tags.AmTag, ],
  "weight": Edm.Int64,
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigLink",
   "module": "DataBroker",
   "permissionTag": "Create",
   "tree": "ConfigTree"
}
```

11.8.4 Add an operational link from host to switch

URL: /api/default/operational/ports('id')/AddOperationalLinkFromHostToSwitch
Method: POST
OData Type: Sel.AddOperationalLinkFromHostToSwitch
Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id
There are 1 type(s) that can be POSTed:

- Sel.AddOperationalLinkFromHostToSwitch

Request Structure

```
{
  "switchPortId":  Edm.String,
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "TopologyManager",
   "permissionTag": "TopologyDiscoveryHinting"
}
```

11.8.5 Update configured links (PUT)

URL: /api/default/config/links('id')/
Method: PUT
OData Type: Sel.Sel5056.TopologyManager.Links.ConfigLink
Binding parameter (id): Sel.Sel5056.TopologyManager.Links.ConfigLink.id
There are 1 type(s) that can be modified:

- Sel.Sel5056.TopologyManager.Links.ConfigLink

Request Structure

The request structure is the same as the GET response above. Some fields may be omitted if not POSTable. Any POSTable fields not included will be reset to the default state.

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigLink",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "ConfigTree"
}
```

11.8.6 Update configured links (PATCH)

URL: /api/default/config/links('id')/
Method: PATCH
OData Type: Sel.Sel5056.TopologyManager.Links.ConfigLink
Binding parameter (id): Sel.Sel5056.TopologyManager.Links.ConfigLink.id
There are 1 type(s) that can be modified:

- Sel.Sel5056.TopologyManager.Links.ConfigLink

The following fields are patchable:

- displayName
- endPoints
- first
- last
- tags
- weight

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigLink",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigLink",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigLink",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "ConfigTree"
}
```

11.8.7 Replace the configuration object for a link

URL: /api/default/operational/links('id')/ReplaceConfig
Method: POST
OData Type: Sel.ReplaceConfig
Binding parameter (id): Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.id
There are 1 type(s) that can be POSTed:

- Sel.ReplaceConfig

Request Structure

```
{
  "configKey":  Edm.String,
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "TopologyManager",
  "permissionTag": "ReplaceObject"
}
```

11.8.8 Delete configured links

URL: /api/default/config/links('id')/
Method: DELETE
OData Type: Sel.Sel5056.TopologyManager.Links.ConfigLink
Binding parameter (id): Sel.Sel5056.TopologyManager.Links.ConfigLink.id

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
  "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
  "dataType": "ConfigLink",
  "module": "DataBroker",
  "permissionTag": "Delete",
  "tree": "ConfigTree"
}
```

11.8.9 Adopt a link with default settings

URL: /api/default/operational/links('id')/Adopt
Method: POST
OData Type: Sel.Adopt
Binding parameter (id): Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.id
There are 1 type(s) that can be POSTed:

- Sel.Adopt

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
  "module": "TopologyManager",
  "permissionTag": "AdoptObject"
}
```

11.8.10 Adopt a link with a predefined link object

URL: /api/default/operational/links('id')/AdoptWithConfig
Method: POST
OData Type: Sel.AdoptWithConfig
Binding parameter (id): Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.id
There are 1 type(s) that can be POSTed:

- Sel.AdoptWithConfig

Request Structure

```
{
  "configKey":  Edm.String,
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "TopologyManager",
   "permissionTag": "AdoptObject"
}
```

11.8.11 Unadopt a link

URL: /api/default/operational/links('id')/Unadopt
Method: POST
OData Type: Sel.Unadopt
Binding parameter (id): Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.id
There are 1 type(s) that can be POSTed:

- Sel.Unadopt

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "TopologyManager",
   "permissionTag": "UnadoptObject"
}
```

11.8.12 Filter links by a port attribute

URL: /api/default/operational/GetLinkByPortAttribute
Method: POST
OData Type: Sel.GetLinkByPortAttribute
There are 1 type(s) that can be POSTed:

- Sel.GetLinkByPortAttribute

Request Structure

```
{
  "attribute":  Sel.Sel5056.TopologyManager.Attributes.AmAttribute,
}
```

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkPort",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkLink",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkLink",
   "module": "DataBroker",
   "permissionTag": "Subscribe",
   "tree": "OperationalTree"
}
```

11.9 Default Adoption Settings

purpose	endpoints	methods
read	/settings/networkSettings/ /settings/networkSettings('id')/	GET GET
update	/settings/networkSettings('id')/ /settings/networkSettings('id')/	PUT PATCH

11.9.1 Read default adoption settings and default ARP SPA IP address

URL: /api/default/settings/networkSettings/
Method: GET
OData Type: Sel.Sel5056.TopologyManager.NetworkSettings
There are 1 type(s) that can be returned:

- Sel.Sel5056.TopologyManager.NetworkSettings

The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "controllerArpSrcIpAddress":  Edm.String (ipv4address),
    "controllerArpSrcNetworkAddresses":  [ Sel.BlueFrame.Core.Networking.Network-
InterfaceAddress, ],
    "controllerIp":  Edm.String (ipv4address),
    "defaultGateway":  Edm.String (ipv4address),
    "id":  Edm.String,
    "ntpServers":  [ Edm.String, ] (ipv4address),
  , ]
}
```

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "NetworkSettings",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SettingsTree"
}
```

11.9.2 Update default adoption settings and default ARP SPA IP address (PUT)

URL: /api/default/settings/networkSettings('id')/
Method: PUT
OData Type: Sel.Sel5056.TopologyManager.NetworkSettings
Binding parameter (id): Sel.Sel5056.TopologyManager.NetworkSettings.id
There are 1 type(s) that can be modified:

- Sel.Sel5056.TopologyManager.NetworkSettings

Request Structure

The request structure is the same as the GET response above. Some fields may be omitted if not POSTable. Any POSTable fields not included will be reset to the default state.

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "NetworkSettings",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "SettingsTree"
}
```

11.9.3 Update default adoption settings and default ARP SPA IP address (PATCH)

URL: /api/default/settings/networkSettings('id')/
Method: PATCH
OData Type: Sel.Sel5056.TopologyManager.NetworkSettings
Binding parameter (id): Sel.Sel5056.TopologyManager.NetworkSettings.id
There are 1 type(s) that can be modified:

- Sel.Sel5056.TopologyManager.NetworkSettings

The following fields are patchable:

- controllerArpSrcIpAddress
- controllerArpSrcNetworkAddresses
- controllerIp
- defaultGateway
- ntpServers

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "NetworkSettings",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SettingsTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "NetworkSettings",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "SettingsTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "NetworkSettings",
   "module": "DataBroker",
   "permissionTag": "Subscribe",
   "tree": "SettingsTree"
}
```

Chapter 12

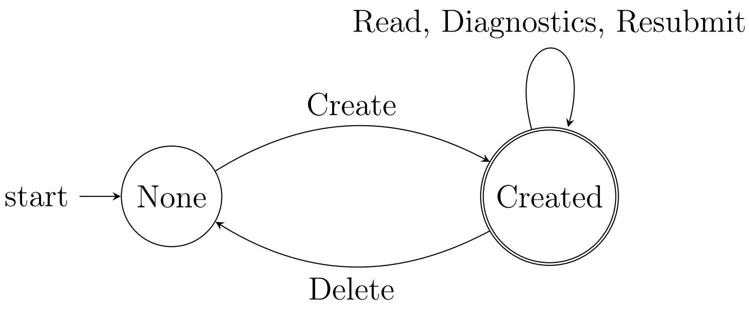
Logical Programming

The logical programming component provides an automated method to generating the OpenFlow programming necessary for an OpenFlow to forward traffic. Logical programming is comprised of two parts: Communication Service Types (CSTs) and Logical Connections (LCs). CSTs are the blueprint for a flow and the LCs is an instance of a flow.

12.1 Logical Connections

LCs required a CST, an adopted source and destination(s), and at least one adopted path. LCs are created between nodes (hence the endpoint endpoints are node ids).

purpose	endpoints	methods
read	/config/logicalConnections/ /config/logicalConnections('id')/	GET GET
create	/config/logicalConnections/ /config/logicalConnections/ReplanAll	POST POST
update	Not supported	
delete	/config/logicalConnections('id')/ /config/logicalConnections/DeleteAll	DELETE POST
resubmit	/config/logicalConnections('id')/ResubmitLogicalConnection /operational/nodes('id')/ReplanInbandPath	POST POST
diagnostics	/config/logicalConnections('id')/GetLogicalConnectionStats /config/logicalConnections('id')/GetOpenFlowInformation /config/logicalConnections('id')/GetPathPlanInformation	POST POST POST



12.1.1 Read created logical connections

URL: /api/default/config/logicalConnections/
Method: GET
OData Type: Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection
There are 1 type(s) that can be returned:

- Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection

The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "communicationServiceTypeId":  Edm.String (Sel.Sel5056.-
LogicalConnectionValidation.DataObject.CommunicationServiceType.id),
    "connectionOrigin":  Sel.Sel5056.LogicalConnectionValidation.Enums.Logical-
ConnectionOrigin,
    "destinationEndPoints":  [ Edm.String, ],
    "errorState":  Sel.BlueFrame.Core.ErrorState,
    "errors":  [ Edm.String, ],
    "id":  Edm.String,
    "sourceEndPoints":  [ Edm.String, ],
    "version":  Edm.Int32,
  , ]
}
```

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "LogicalConnection",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
}
```

12.1.2 Create logical connections

URL: /api/default/config/logicalConnections/
Method: POST
OData Type: Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection

There are 1 type(s) that can be POSTed:

- Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection

Request Structure

```
{
  "communicationServiceTypeId":  Edm.String (Sel.Sel5056.-
LogicalConnectionValidation.DataObject.CommunicationServiceType.id),
  "destinationEndPoints":  [ Edm.String, ],
  "sourceEndPoints":  [ Edm.String, ],
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigNode",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "LogicalConnection",
   "module": "DataBroker",
   "permissionTag": "Create",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "CommunicationServiceType",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
},
{
   "module": "LogicalConnectionValidation",
   "permissionTag": "ActivateLogicalConnection"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "VlanVidReservation",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
}
```

Examples

POST a unicast LC Example 10.

```
POST /api/default/config/logicalConnections HTTP/1.1
Content-Type: application/json
HOST: 192.168.56.1
Authorization: Bearer
   fadebc87db216dc80bfcfd1e70afede3c00844e7b0563c0b09d3c8d67d8a75bf
Content-Length: 163

{
   "communicationServiceTypeId": "NTP Client",
   "destinationEndPoints": [
      "aba2458f938f84886b017d24928dc15c"
   ],
   "sourceEndPoints": [
      "af4b603cf41724642b96c55e53817efb"
   ]
}
```

Listing 12.1: POST a unicast LC

POST a multicast LC Example 11.

```
POST /api/default/config/logicalConnections HTTP/1.1
Content-Type: application/json
HOST: 192.168.56.1
Authorization: Bearer 8
   bbc674349d7b1158373af896b5cdf5fa7874bd947eb24980409ecd89847c9fb
Content-Length: 206

{
   "communicationServiceTypeId": "PTP Power Profile",
   "destinationEndPoints": [
      "a31065d35724a4cf998f0b21b052ad4a",
      "a45d86de1fa7841d8ac33db1ca18efd7"
   ],
   "sourceEndPoints": [
      "aba2458f938f84886b017d24928dc15c"
   ]
}
```

Listing 12.2: POST a multicast LC

12.1.3 Replan all logical connections

URL: /api/default/config/logicalConnections/ReplanAll
Method: POST
OData Type: Sel.ReplanAll
There are 1 type(s) that can be POSTed:

- Sel.ReplanAll

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigNode",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "LogicalConnection",
   "module": "DataBroker",
   "permissionTag": "Create",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "LogicalConnection",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "CommunicationServiceType",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
},
{
   "module": "LogicalConnectionValidation",
   "permissionTag": "ActivateLogicalConnection"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "VlanVidReservation",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
}
```

12.1.4 Delete created logical connections

URL: /api/default/config/logicalConnections('id')/
Method: DELETE
OData Type: Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection
Binding parameter (id): Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection.id

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "LogicalConnection",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "ConfigTree"
},
{
   "module": "LogicalConnectionValidation",
   "permissionTag": "ActivateLogicalConnection"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "VlanVidReservation",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
}
```

12.1.5 Delete all logical connections

URL: /api/default/config/logicalConnections/DeleteAll
Method: POST
OData Type: Sel.DeleteAll
There are 1 type(s) that can be POSTed:

- Sel.DeleteAll

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "LogicalConnection",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "LogicalConnection",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "ConfigTree"
},
{
   "module": "LogicalConnectionValidation",
   "permissionTag": "ActivateLogicalConnection"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "VlanVidReservation",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
}
```

12.1.6 Resubmit a logical connection

URL: /api/default/config/logicalConnections('id')/ResubmitLogicalConnection
Method: POST
OData Type: Sel.ResubmitLogicalConnection
Binding parameter (id): Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection.id
There are 1 type(s) that can be POSTed:

- Sel.ResubmitLogicalConnection

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigNode",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkNode",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "LogicalConnection",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "LogicalConnection",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "ConfigTree"
}
```

12.1.7 Replan the IB management logical connection to a switch

URL: /api/default/operational/nodes('id')/ReplanInbandPath
Method: POST
OData Type: Sel.ReplanInbandPath
Binding parameter (id): Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id
There are 1 type(s) that can be POSTed:

- Sel.ReplanInbandPath

Request Structure

No request body

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission",
   "deviceType": "SapphireDevice",
   "module": "DeviceManagement",
   "permissionTag": "ReplanControllerRoute"
}
```

12.1.8 Get the Openflow counters for a logical connection

URL: /api/default/config/logicalConnections('id')/GetLogicalConnectionStats
Method: POST
OData Type: Sel.GetLogicalConnectionStats
Binding parameter (id): Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection.id
There are 1 type(s) that can be POSTed:

- Sel.GetLogicalConnectionStats

Request Structure

No request body

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "LogicalConnection",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
}
```

12.1.9 Get the OpenFlow entries for a logical connection

URL: /api/default/config/logicalConnections('id')/GetOpenFlowInformation
Method: POST
OData Type: Sel.GetOpenFlowInformation
Binding parameter (id): Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection.id
There are 1 type(s) that can be POSTed:

- Sel.GetOpenFlowInformation

Request Structure

No request body

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "LogicalConnection",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
}
```

12.1.10 Get path plans for a logical connection

URL: /api/default/config/logicalConnections('id')/GetPathPlanInformation
Method: POST
OData Type: Sel.GetPathPlanInformation
Binding parameter (id): Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection.id
There are 1 type(s) that can be POSTed:

- Sel.GetPathPlanInformation

Request Structure

No request body

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

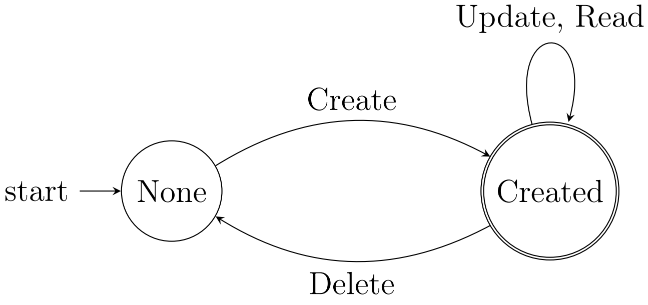
Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "LogicalConnection",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
}
```

12.2 Communication Service Types

CSTs are a blueprint for the OpenFlow entries created for an LCs. All of the settings for LCs are contained in the CST. CSTs may be used be 0, 1, or multiple LCs.

purpose	endpoints	methods
read	/config/communicationServiceTypes/ /config/communicationServiceTypes('id')/	GET GET
create	/config/communicationServiceTypes/	POST
update	/config/communicationServiceTypes('id')/ /config/communicationServiceTypes('id')/	PUT PATCH
delete	/config/communicationServiceTypes('id')/	DELETE



12.2.1 Read created communication service types (CST)

URL: /api/default/config/communicationServiceTypes/
Method: GET
OData Type: Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType
There are 1 type(s) that can be returned:

- Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType

The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "autoFillSettings":  Sel.Sel5056.LogicalConnectionValidation.DataObject.AutoFillSettings,
    "cstCommunicationType":  Sel.Sel5056.LogicalConnectionValidation.Enums.CstCommunicationType,
    "displayName":  Edm.String,
    "errorState":  Sel.BlueFrame.Core.ErrorState,
    "errors":  [ Edm.String, ],
    "id":  Edm.String,
    "numFailableLinks":  Edm.Int64,
    "numFailableNodes":  Edm.Int64,
    "priority":  Edm.Int32,
    "setQueue":  Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetQueueAction,
    "tags":  [ Sel.Sel5056.Tags.AmTag, ],
    "trafficMatch":  Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match,
    "version":  Edm.Int32,
  , ]
}
```

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "CommunicationServiceType",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
}
```

12.2.2 Create communication service types (CST)

URL: /api/default/config/communicationServiceTypes/
Method: POST
OData Type: Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType
There are 1 type(s) that can be POSTed:

- Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType

Request Structure

Property	Default Value

autoFillSettings	null
cstCommunicationType	UnicastBidirectional
numFailableLinks	0
numFailableNodes	0
priority	2000
setQueue	null
tags	[]
trafficMatch	{ "matchFields": [] }

Table 12.3: Default Values for Sel.Sel5056.LogicalConnectionValidation.DataObject.-CommunicationServiceType

```
{
  "autoFillSettings": Sel.Sel5056.LogicalConnectionValidation.DataObject.Auto-
FillSettings,
  "cstCommunicationType": Sel.Sel5056.LogicalConnectionValidation.Enums.Cst-
CommunicationType,
  "displayName": Edm.String,
  "numFailableLinks": Edm.Int64,
  "numFailableNodes": Edm.Int64,
  "priority": Edm.Int32,
  "setQueue": Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetQueueAction,
  "tags": [ Sel.Sel5056.Tags.AmTag, ],
  "trafficMatch": Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match,
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "CommunicationServiceType",
   "module": "DataBroker",
   "permissionTag": "Create",
   "tree": "ConfigTree"
}
```

Examples

POST a unicast CST Example 12.

```
POST /api/default/config/communicationServiceTypes HTTP/1.1
Content-Type: application/json
```

```
HOST: 192.168.56.1
Authorization: Bearer 658
   b07512eaa648040243cb33ef3c75561b4f1e8803256be16cf7f201c07753b
Content-Length: 234

{
   "cstCommunicationType": "UnicastBidirectional",
   "displayName": "DNP",
   "setQueue": {
      "queueId": 3
   },
   "trafficMatch": {
      "matchFields": [
         {
            "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
               DataTreeObjects.MatchFields.UdpDst",
            "value": "20001"
         }
      ]
   }
}
```

Listing 12.3: POST a unicast CST

POST a multicast CST Example 13.

```
POST /api/default/config/communicationServiceTypes HTTP/1.1
Content-Type: application/json
HOST: 192.168.56.1
Authorization: Bearer 4
   c742c932070cdfb4ac0d3a2a01df1c0ebb95f26b804806fb7cb422dac0660fb
Content-Length: 250

{
   "cstCommunicationType": "Multicast",
   "displayName": "ProfiNET Discovery",
   "setQueue": {
      "queueId": 1
   },
   "trafficMatch": {
      "matchFields": [
         {
            "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
               DataTreeObjects.MatchFields.EthDst",
            "value": "01:0E:CF:00:00:00"
         }
      ]
```

```
    }
}
```

Listing 12.4: POST a multicast CST

12.2.3 Update created communication service types (CST) (PUT)

URL: /api/default/config/communicationServiceTypes('id')/
Method: PUT
OData Type: Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType
Binding parameter (id): Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType.id
There are 1 type(s) that can be modified:

- Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType

Request Structure

The request structure is the same as the GET response above. Some fields may be omitted if not POSTable. Any POSTable fields not included will be reset to the default state.

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "CommunicationServiceType",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "ConfigTree"
}
```

12.2.4 Update created communication service types (CST) (PATCH)

URL: /api/default/config/communicationServiceTypes('id')/
Method: PATCH
OData Type: Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType
Binding parameter (id): Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType.id
There are 1 type(s) that can be modified:

- Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType

The following fields are patchable:

- autoFillSettings

- cstCommunicationType
- displayName
- numFailableLinks
- numFailableNodes
- priority
- setQueue
- tags
- trafficMatch

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "CommunicationServiceType",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "CommunicationServiceType",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "CommunicationServiceType",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "ConfigTree"
}
```

12.2.5 Delete created communication service types (CST)

URL: /api/default/config/communicationServiceTypes('id')/
Method: DELETE
OData Type: Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType
Binding parameter (id): Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType.id

Available User Roles with Required Permissions

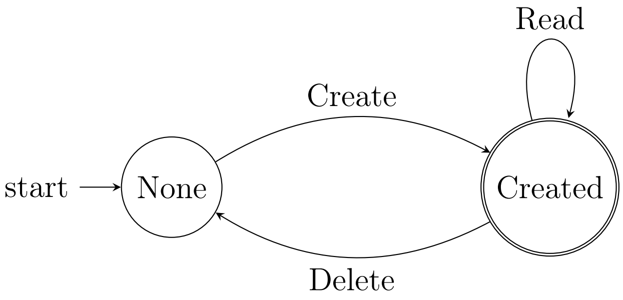
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "CommunicationServiceType",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "ConfigTree"
}
```

12.3 VID Reservation

purpose	endpoints	methods
read	/config/vlanVidReservation/ /config/vlanVidReservation('id')/	GET GET
create	/config/vlanVidReservation/ReserveVlanVidRange	POST
update	Not supported	
delete	/config/vlanVidReservation('id')/	DELETE



12.3.1 Read reserved VIDs

URL: /api/default/config/vlanVidReservation/
Method: GET
OData Type: Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation
There are 2 type(s) that can be returned:

- Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation

- Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidDestinationReservation

The return will be a list of one or more of these types show below

Response Structure for Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation

```
{
  [
    "creator":  Edm.String,
    "id":  Edm.String,
    "uniqueId":  Edm.String,
    "vlanVid":  Edm.Int32,
  , ]
}
```

Response Structure for Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidDestinationReservation

```
{
  "@odata.type":  "#Sel.Sel5056.PathProgrammer.VlanVidManager-
.VlanVidDestinationReservation",   "amDestination":  Sel.Sel5056.PathPlanner.-
PathFinding.PrefixTree.DestinationDescriptions.Types.AmDestination,
  "creator":  Edm.String,
  "id":  Edm.String,
  "uniqueId":  Edm.String,
  "vlanVid":  Edm.Int32,
}
```

Available User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "VlanVidReservation",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
}
```

12.3.2 Reserve a VID or range of VIDs

URL: /api/default/config/vlanVidReservation/ReserveVlanVidRange
Method: POST
OData Type: Sel.ReserveVlanVidRange
There are 1 type(s) that can be POSTed:

- Sel.ReserveVlanVidRange

Request Structure

```
{
  "end":  Edm.Int32,
  "start":  Edm.Int32,
}
```

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "module": "PathProgrammer",
   "permissionTag": "ReserveVlanVidPermission"
}
```

Examples

Reserve a range of values Example 14.

```
POST /api/default/config/vlanVidReservation/ReserveVlanVidRange HTTP
   /1.1
Content-Type: application/json
HOST: 192.168.56.1
Authorization: Bearer
   a915eb55bd8d304db3762d4f0b440b7d5b796c22869c2d9cd7c343ad51bd9583
Content-Length: 23

{
   "end": 10,
   "start": 5
}
```

Listing 12.5: Reserve a range of values

Reserve a single VID Example 15.

```
POST /api/default/config/vlanVidReservation/ReserveVlanVidRange HTTP
   /1.1
Content-Type: application/json
HOST: 192.168.56.1
Authorization: Bearer 0
   a82203669dea2702302cf016a01d46901ce408ab4fb1e1e14526651efbcb5db
```

```
Content-Length: 22

{
   "end": 5,
   "start": 5
}
```

Listing 12.6: Reserve a single VID

12.3.3 Delete reserved VIDs

URL: /api/default/config/vlanVidReservation('id')/
Method: DELETE
OData Type: Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation
Binding parameter (id): Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation.id

Available User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "VlanVidReservation",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "ConfigTree"
}
```

Chapter 13

OpenFlow Programming

There are three types of OpenFlow entries:

- Flow
- Group
- Meter

These entries are divided into two types of objects: configuration, which are managed by the user and the Controller, and operational, which are the statistics returned by the switches. The SEL-5056 stores the OpenFlow entries of each type for all switches in the same table. The node (and for flow entries, also table id) is a property in the table. Retrieving the all of the flow entries in the SEL-5056 returns all of the flow entries on all switches from all tables as programmed on the SEL-5056. Sorting and filtering is available through OData.

Flow entries contain group actions that reference group entries and also a meter instructions that references meter entries. Group entries may also contain group actions to reference other groups through chaining.

Diagnostics are contained in the flowStats, groupStats, and meterStats. These contain the OpenFlow statistics (or diagnostics) of the OpenFlow entries. Descriptions are contained in the flowStats, groupDescr, meterDescr endpoints. This is the data as retrieved from the switches rather than from the SEL-5056 tables. Requesting the flowStats returns all flowStats from all switches and all tables. The SEL-5056 compares the descriptions of the entries as reported by the switches against its own tables of what the switches should have to calculate differences. If there is a difference, a synchronization request may occur. Submitting the synchronization request resolves the difference by modifying the switch configuration.

13.1 Flow Entries

This section describes the flow entry data structure and its related endpoints and methods.

Flows entries may differ from the OpenFlow data model by two parts:

- Use of friendly names for values
- Use of aliased match fields, actions, and instructions.

Friendly names are strings that stand for integers. These can be used for InPort and others as described in the SEL-5056 instruction manual. Fields that can have friendly names can have multiple values that are equivalent. The translated value is normalized.
Aliased values are used to provide dynamic updates to flow entries by changes to other components, such as IP addresses. For example, if Ipv4DstByAlias is used instead of Ipv4Dst, then if the IP addresses of the underlying object referenced in the match field is changed, then the SEL-5056 will automatically update the flow entry so that during the next switch to SEL-5056 flow entry compare, the switch will be unsynchronized. Non-alised match fields would need to updating using the instructions below.
Flow entries are made of three parts:

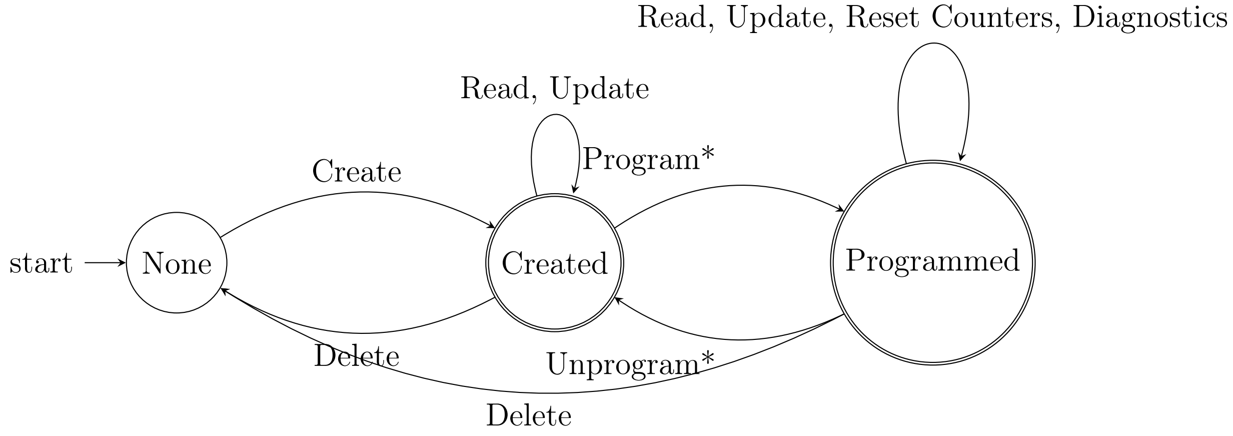
- General settings
- Match Fields
- Instructions (including Actions)

General settings can be further divided into two sets: those that are not defined by OpenFlow and those that are.
Those not defined by OpenFlow are:

- displayName
- enabled
- errorState
- errors
- id
- lastEdit
- node
- tags
- version

These are describe below. The rest are defined in the OpenFlow 1.3.4 specification.

purpose	endpoints	methods
read	/config/flows/ /config/flows('id')/	GET GET
create	/config/flows/	POST
update	/config/flows('id')/	PUT
reset counters	/config/flows('id')/ResetCounters	POST
delete	/config/flows('id')/	DELETE
diagnostics	/operational/flowStats/ /operational/flowStats('id')/	GET GET



13.1.1 Read flow entries

URL: /api/default/config/flows/
Method: GET
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow
There are 1 type(s) that can be returned: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "bufferId": Edm.Int64,
    "checkOverlap": Edm.Boolean,
    "cookie": Edm.Int64,
    "displayName": Edm.String,
    "enabled": Edm.Boolean,
    "errorState": Sel.BlueFrame.Core.ErrorState,
    "errors": [ Edm.String, ],
    "hardTimeout": Edm.Int32,
    "id": Edm.String,
    "idleTimeout": Edm.Int32,
    "inPort": Edm.Int64,
    "instructions": [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction, ],
    "lastEdit": Edm.DateTimeOffset,
    "match": Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match,
    "noByteCounts": Edm.Boolean,
    "noPacketCounts": Edm.Boolean,
    "node": Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode.id),
    "outGroup": Edm.Int64,
    "outPort": Edm.Int64,
    "priority": Edm.Int32,
    "resetCounts": Edm.Boolean,
    "tableId": Edm.Byte,
    "tags": [ Sel.Sel5056.Tags.AmTag, ],
    "version": Edm.Int32,
  , ]
}
```

User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Flow",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
}
```

13.1.2 Create flow entries

URL: /api/default/config/flows/
Method: POST
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow
There are 1 type(s) that can be POSTed: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow

Request Structure

Property	Default Value
bufferId	0
checkOverlap	false
displayName	null
enabled	true
hardTimeout	0
idleTimeout	0
instructions	[]
match	'matchFields': []
noByteCounts	false
noPacketCounts	false
node	null
priority	1000
resetCounts	false
tableId	0
tags	[]

Table 13.3: Default Values for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow

```
{
  "bufferId":  Edm.Int64,
  "checkOverlap":  Edm.Boolean,
  "displayName":  Edm.String,
  "enabled":  Edm.Boolean,
  "hardTimeout":  Edm.Int32,
  "idleTimeout":  Edm.Int32,
  "instructions":  [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction, ],
  "match":  Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match,
  "noByteCounts":  Edm.Boolean,
  "noPacketCounts":  Edm.Boolean,
  "node":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode.id),
  "priority":  Edm.Int32,
  "resetCounts":  Edm.Boolean,
  "tableId":  Edm.Byte,
  "tags":  [ Sel.Sel5056.Tags.AmTag, ],
}
```

User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Flow",
   "module": "DataBroker",
   "permissionTag": "Create",
   "tree": "ConfigTree"
}
```

Examples

Create a simple flow entry Example 16.

```
POST /api/default/config/flows HTTP/1.1
Content-Type: application/json
HOST: 192.168.1.1
Authorization: Bearer
   d61f3cc466a02535c2a93cecb08ff07ed532ce2b3129e816e1c667539906bbae
Content-Length: 411

{
   "displayName": "Simple Flow",
   "instructions": [
      {
         "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects
            .WriteActions",
         "actions": [
            {
               "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                  DataTreeObjects.OutputAction",
               "outPort": "9"
            }
         ]
      }
   ],
   "match": {
      "matchFields": [
         {
            "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
               DataTreeObjects.MatchFields.EthDst",
            "value": "012300000001"
         }
      ]
   },
   "node": "af4b603cf41724642b96c55e53817efb"
}
```

Listing 13.1: Create a simple flow entry

13.1.3 Update flow entries (PUT)

URL: /api/default/config/flows('id')/
Method: PUT
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow
Binding parameter (id): Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow.id
There are 1 type(s) that can be modified: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow

Request Structure

The request structure is the same as the GET response above. Some fields may be omitted if not POSTable. Any POSTable fields not included will be reset to the default state.

User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
  "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
  "dataType": "Flow",
  "module": "DataBroker",
  "permissionTag": "Update",
  "tree": "ConfigTree"
}
```

Examples

Update a simple flow entry by changing Outport Example 17.

```
PUT /api/default/config/flows('a895e430a0e2e4458af6d04207c7b15d')
   HTTP/1.1
Content-Type: application/json
HOST: 192.168.56.1
Authorization: Bearer
   b71f5f3da46d1202e81ac1bd1ebd9ea4e0aa26002170e16f28d72785e724e25e
Content-Length: 986

{
   "@odata.context": "https://192.168.56.1/api/default/config/
      $metadata#flows/$entity",
   "bufferId": 0,
   "checkOverlap": false,
   "cookie": 1015,
   "displayName": "Simple Flow",
   "enabled": true,
   "errorState": "Failure",
   "errors": [
      "Node does not support OpenFlow"
   ],
   "hardTimeout": 0,
   "id": "a895e430a0e2e4458af6d04207c7b15d",
   "idleTimeout": 0,
   "inPort": null,
   "instructions": [
      {
         "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects
            .WriteActions",
         "actions": [
            {
               "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                  DataTreeObjects.OutputAction",
               "maxLength": 65535,
               "outPort": "10"
            }
         ]
      }
   ],
    "lastEdit": "2020-06-11T13:44:15.5737887-06:00",
    "match": {
        "matchFields": [
            {
                "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                   DataTreeObjects.MatchFields.EthDst",
                "mask": null,
                "translatedValue": "01:23:00:00:00:01",
                "value": "012300000001"
            }
        ]
    },
    "noByteCounts": false,
    "noPacketCounts": false,
    "node": null,
    "outGroup": null,
    "outPort": 9,
    "priority": 1000,
    "resetCounts": false,
    "tableId": 0,
    "tags": [],
    "version": 1
}
```

Listing 13.2: Update a simple flow entry by changing Outport

13.1.4 Reset the counters for a flow entry

URL: /api/default/config/flows('id')/ResetCounters
Method: POST
OData Type: Sel.ResetCounters
Binding parameter (id): Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow.id
There are 1 type(s) that can be POSTed: Sel.ResetCounters

Request Structure

No request body

User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ConfigNode",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkNode",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Flow",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
},
{
   "module": "OpenFlowPlugin",
   "permissionTag": "ResetFlowCounters"
},
{
   "module": "OpenFlowPlugin",
   "permissionTag": "SetFlowMod"
}
```

13.1.5 Delete flow entries

URL: /api/default/config/flows('id')/
Method: DELETE
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow
Binding parameter (id): Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow.id

User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
    "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
    "dataType": "Flow",
    "module": "DataBroker",
    "permissionTag": "Delete",
    "tree": "ConfigTree"
}
```

13.1.6 Read flow entry diagnostics

URL: /api/default/operational/flowStats/
Method: GET
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats
There are 1 type(s) that can be returned: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats
The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "dataPathId":  Edm.Int64,
    "id":  Edm.String,
    "stats":  [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat, ],
    "transactionId":  Edm.Int64,
  , ]
}
```

User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "FlowStats",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

13.1.7 Resolving Aliased field values

There are two types of match fields, those that contain a static value and those that contain aliases to static values that can be updated automatically.

> In v2.2.0.0 the translated value will appear as another property for match fields so you don't need to resolve them manually.

The alised match fields are:

1. InPortByAlias
2. EthDstByAlias

3. EthSrcByAlias
4. Ipv4DstByAlias
5. Ipv4SrcByAlias
6. ArpTpaByAlias
7. ArpSpaByAlias

The value property of aliased match fields contain an id property that can be used to find the static value. Aliased fields are different than friendly names as friendly names are fixed.
If the id cannot be resolve as shown below to a specified object, then it is possible that the object has been deleted.

InPortByAlias

The value is the id of a configuration port.

1. Use the value to get the configuration port object
2. Use the linkedKey property value to get the operational port object
3. The OpenFlowPortAttr attribute value is the static value used

EthSrcByAlias and EthDstByAlias

The value is either the id of a SEL-2740S configuration node or the id of a configuration port.
If the value is the id of a SEL-2740S configuration node:

1. Use the value to get the configuration node (if not found, could be the id of a configuration port)
2. Use the linkedKey value to get the operational node
3. Use the last macAddress attribute value

TODO IB versus OOB
If the value is the id of a configuration port

1. Use the value to get the configuration port
2. Use the linkedKey value to get the operational port
3. Return the last EthernetAttr attribute value

EthernetApplicationByAlias

This resolves to the EthSrc address. Currently this is used to select the correct GOOSE source address as some devices publish GOOSE with a different MAC address than IP.

1. Use the value to get the configuration port
2. Use the linkedKey value to get the operational port
3. Return the last EthernetAttr attribute value

Ipv4SrcByAlias, Ipv4DstByAlias, ArpTpaByAlias, ArpSpaByAlias

This is the same as EthSrcByAlias and EthDstByAlias except the attributes are IpAttr and ipAddress.
The value is either the id of a SEL-2740S configuration node or the id of a configuration port.
If the value is the id of a SEL-2740S configuration node:

1. Use the value to get the configuration node (if not found, could be the id of a configuration port)

2. Use the linkedKey value to get the operational node

3. Use the last ipAddress attribute value

If the value is the id of a configuration port

1. Use the value to get the configuration port

2. Use the linkedKey value to get the operational port

3. Return the last IpAttr attribute value

Aliased Field	Instruction
InPortByAlias	value -> Configuration Ports -> Operational Ports -> OpenFlowPortAttr
EthSrcByAlias/EthDstByAlias	There is a difference if the value references a SEL-2740S or another device. First casevalue ->

TODO The following instructions can be alised

1. Write-actions

2. Meter

13.1.8 Write-actions

The alised actions are:

1. Output

2. Group

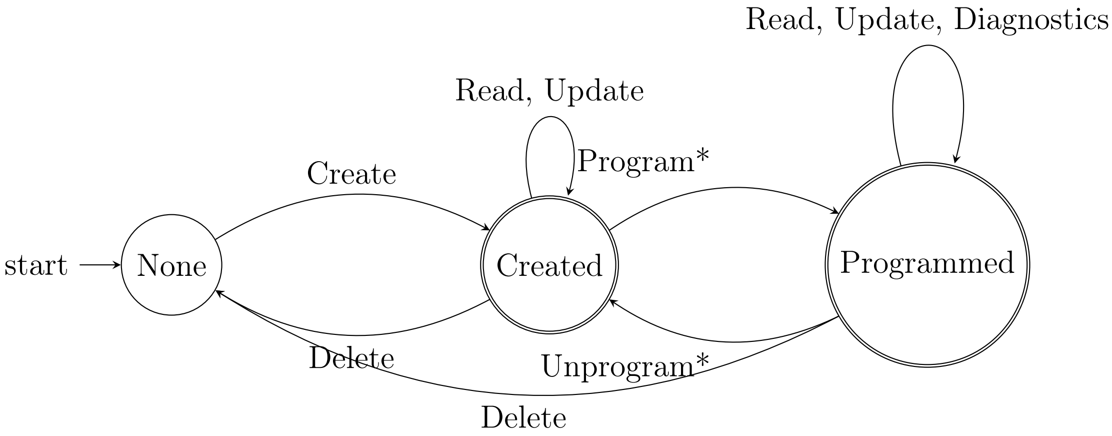
13.1.9 Meter

TODO

13.2 Group Entries

purpose	endpoints	methods

read	/config/groups/ /config/groups('id')/	GET GET
create	/config/groups/	POST
update	/config/groups('id')/ /config/groups('id')/	PUT PATCH
delete	/config/groups('id')/	DELETE
descriptions	/operational/groupDesc/ /operational/groupDesc('id')/	GET GET
diagnostics	/operational/groupStats/ /operational/groupStats('id')/	GET GET



13.2.1 Read Group entries

URL: /api/default/config/groups/
Method: GET
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group
There are 1 type(s) that can be returned: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group
The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "buckets":  [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket, ],
    "displayName":  Edm.String,
    "enabled":  Edm.Boolean,
    "errorState":  Sel.BlueFrame.Core.ErrorState,
    "errors":  [ Edm.String, ],
    "groupId":  Edm.Int64,
    "groupType":  Sel.Sel5056.OpenFlowPlugin.Enums.OfpGroupType,
    "id":  Edm.String,
    "node":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode.id),
    "tags":  [ Sel.Sel5056.Tags.AmTag, ],
    "version":  Edm.Int32,
  , ]
}
```

User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Group",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
}
```

13.2.2 Create Group entries

URL: /api/default/config/groups/
Method: POST
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group
There are 1 type(s) that can be POSTed: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group

Request Structure

Property	Default Value
buckets	[]
displayName	null
enabled	true
groupId	0
groupType	FastFailover
node	null
tags	[]

Table 13.7: Default Values for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group

```
{
  "buckets":  [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket, ],
  "displayName":  Edm.String,
  "enabled":  Edm.Boolean,
  "groupId":  Edm.Int64,
  "groupType":  Sel.Sel5056.OpenFlowPlugin.Enums.OfpGroupType,
  "node":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode.id),
  "tags":  [ Sel.Sel5056.Tags.AmTag, ],
}
```

User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Group",
   "module": "DataBroker",
   "permissionTag": "Create",
   "tree": "ConfigTree"
}
```

Examples

Add a simple All group Example 18.

```
POST /api/default/config/groups HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer
   e6f50626767cfe00421b1a28c13d9abe3edf8b1e5707c27ec8edc81cd096eb0a
Content-Length: 564

{
   "buckets": [
      {
         "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects
            .Bucket",
         "actions": [
            {
               "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                  DataTreeObjects.OutputAction",
               "outPort": "1"
            }
         ],
         "watchGroup": "Any",
         "watchPort": "Any"
      },
      {
         "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects
            .Bucket",
         "actions": [
            {
               "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                  DataTreeObjects.OutputAction",
               "outPort": "2"
            }
         ],
         "watchGroup": "Any",
         "watchPort": "Any"
      }
   ],
   "displayName": "Simple All Group",
   "groupId": 100,
   "groupType": "All",
   "node": "a5e75678a7a27453296fb0eecfb557cf"
}
```

Listing 13.3: Add a simple All group

Add a simple FF group Example 19.

```
POST /api/default/config/groups HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 56
   e83b720cd6eb871578a10a4ff29fec8f2206c08d90da3342b72a8d55e85277
Content-Length: 1217

{
   "buckets": [
      {
         "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects
            .Bucket",
         "actions": [
            {
               "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                  DataTreeObjects.GroupAction",
               "groupId": 100
            },
            {
               "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                  DataTreeObjects.SetQueueAction",
               "queueId": 4
            },
            {
               "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                  DataTreeObjects.SetFieldAction",
               "field": {
                  "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                     DataTreeObjects.MatchFields.VlanVid",
                  "value": "4001"
               }
            },
            {
               "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                  DataTreeObjects.SetFieldAction",
               "field": {
                  "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                     DataTreeObjects.MatchFields.VlanPcp",
                  "value": "7"
               }
            },
            {
               "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                  DataTreeObjects.PushVlanAction",
               "etherType": "33024"
            }
         ],
         "watchGroup": "ANY",
         "watchPort": "B1"
      },
      {
         "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects
            .Bucket",
         "actions": [
            {
               "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                  DataTreeObjects.OutputAction",
               "outPort": "B2"
            },
            {
               "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                  DataTreeObjects.PopVlanAction"
            }
         ],
         "watchGroup": "ANY",
         "watchPort": "B2"
      }
   ],
   "displayName": "Simple FF Group",
   "groupId": 101,
   "groupType": "FastFailover",
   "node": "a5e75678a7a27453296fb0eecfb557cf"
}
```

Listing 13.4: Add a simple FF group

13.2.3 Update Group entries (PUT)

URL: /api/default/config/groups('id')/
Method: PUT
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group
Binding parameter (id): Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.id
There are 1 type(s) that can be modified: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group

Request Structure

The request structure is the same as the GET response above. Some fields may be omitted if not POSTable. Any POSTable fields not included will be reset to the default state.

User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Group",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "ConfigTree"
}
```

Examples

Modify (PUT) the action buckets of an All group Example 20.

```
PUT /api/default/config/groups('a3477a965317443a6802f94ee4b11bfc')
   HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 4
   cca2a113660d9657e660fb48f3ea48850d30eec1ce383c75658fe321e08158c
Content-Length: 782

{
   "buckets": [
      {
         "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects
            .Bucket",
         "actions": [
            {
               "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                  DataTreeObjects.OutputAction",
               "outPort": "1"
            }
         ],
         "watchGroup": "Any",
         "watchPort": "Any"
      },
      {
         "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects
            .Bucket",
         "actions": [
            {
               "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                  DataTreeObjects.OutputAction",
               "outPort": "2"
            }
         ],
         "watchGroup": "Any",
         "watchPort": "Any"
      },
      {
         "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects
            .Bucket",
         "actions": [
            {
               "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                  DataTreeObjects.OutputAction",
               "outPort": "3"
            }
         ],
         "watchGroup": "Any",
         "watchPort": "Any"
      }
   ],
   "displayName": "Simple All Group",
   "groupId": 100,
   "groupType": "All",
   "node": "a5e75678a7a27453296fb0eecfb557cf"
}
```

Listing 13.5: Modify (PUT) the action buckets of an All group

13.2.4 Update Group entries (PATCH)

URL: /api/default/config/groups('id')/
Method: PATCH
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group
Binding parameter (id): Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.id

Data structures must be patched as complete units. So to update the action buckets (buckets), you include the entire set of action buckets after making all modifications, such as reordering, adding, deleting, or modifying. You may PATCH more than one field at a time. There are 1 type(s) that can be modified: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group
The following fields are patchable:

- buckets
- displayName
- enabled
- groupId
- groupType
- node
- tags

Request Structure

The request structure includes one or more properties that are patchable. Any property not included in the request structure will retain its current value (subject to any property relationships).

User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Group",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Group",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Group",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "ConfigTree"
}
```

Examples

Modify (PATCH) the action buckets of an All group Example 21.
This example shows how to modify the action buckets of a group entry. Modification could mean changing the order, having a different number of action buckets, and/or modify the buckets themselves, such as chaning an action. The desired bucket definition is then placed in the body of the message. Therefore, from this example, one cannot see what the original definition would have been.

```
PATCH /api/default/config/groups('a3477a965317443a6802f94ee4b11bfc')
    HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer
   a32ddcd7c27f70c88da9ffd7585e3c9459770eebdd347f879dbdefbd11da35f3
Content-Length: 449

{
   "buckets": [
      {
         "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects
            .Bucket",
         "actions": [
            {
               "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                  DataTreeObjects.OutputAction",
               "outPort": "3"
            }
         ],
         "watchGroup": "Any",
         "watchPort": "Any"
      },
      {
         "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects
            .Bucket",
         "actions": [
            {
               "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.
                  DataTreeObjects.OutputAction",
               "outPort": "4"
            }
         ],
         "watchGroup": "Any",
         "watchPort": "Any"
      }
   ]
}
```

Listing 13.6: Modify (PATCH) the action buckets of an All group

13.2.5 Delete Group entries

URL: /api/default/config/groups('id')/
Method: DELETE
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group
Binding parameter (id): Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.id

User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Group",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "ConfigTree"
}
```

13.2.6 Read Group entry descriptions

URL: /api/default/operational/groupDesc/
Method: GET
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc
There are 1 type(s) that can be returned: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "dataPathId":  Edm.Int64,
    "groups":  [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group, ],
    "id":  Edm.String,
    "transactionId":  Edm.Int64,
  , ]
}
```

User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "GroupDesc",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

13.2.7 Read group entry diagnostics

URL: /api/default/operational/groupStats/
Method: GET
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats
There are 1 type(s) that can be returned: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "dataPathId":  Edm.Int64,
    "id":  Edm.String,
    "stats":  [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat, ],
    "transactionId":  Edm.Int64,
  , ]
}
```

User Roles with Required Permissions

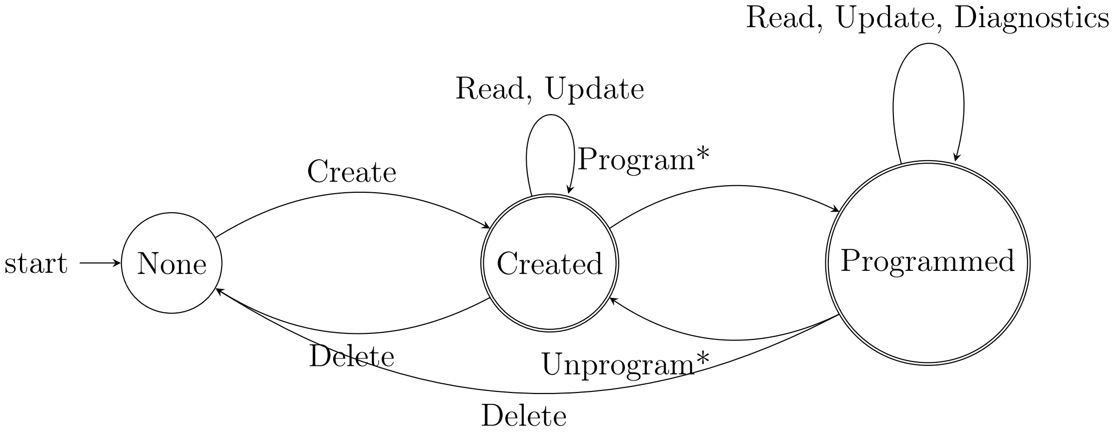
- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "GroupStats",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

13.3 Meter Entries

purpose	endpoints	methods
read	/config/meters/ /config/meters(’id’)/	GET GET
create	/config/meters/	POST
update	/config/meters(’id’)/ /config/meters(’id’)/	PUT PATCH
delete	/config/meters(’id’)/	DELETE
descriptions	/operational/meterDescriptions/ /operational/meterDescriptions(’id’)/	GET GET
features	/operational/meterFeatures/ /operational/meterFeatures(’id’)/	GET GET
diagnostics	/operational/meterstats/ /operational/meterstats(’id’)/	GET GET



13.3.1 Read Meter entries

URL: /api/default/config/meters/
Method: GET
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter
There are 1 type(s) that can be returned: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "displayName":  Edm.String,
    "enabled":  Edm.Boolean,
    "errorState":  Sel.BlueFrame.Core.ErrorState,
    "errors":  [ Edm.String, ],
    "flags":  [ Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.-
RateTypeBase, ],
    "id":  Edm.String,
    "meterBands":  [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand, ],
    "meterId":  Edm.Int64,
    "node":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode.id),
    "tags":  [ Sel.Sel5056.Tags.AmTag, ],
    "version":  Edm.Int32,
  , ]
}
```

User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Meter",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
}
```

13.3.2 Create Meter entries

URL: /api/default/config/meters/
Method: POST
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter
There are 1 type(s) that can be POSTed: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter

Request Structure

Property	Default Value
displayName	null
enabled	true
flags	[]
meterBands	[]
meterId	0
node	null
tags	[]

Table 13.10: Default Values for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter

```
{
  "displayName":  Edm.String,
  "enabled":  Edm.Boolean,
  "flags":  [ Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.RateTypeBase, ],
  "meterBands":  [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand, ],
  "meterId":  Edm.Int64,
  "node":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode.id),
  "tags":  [ Sel.Sel5056.Tags.AmTag, ],
}
```

User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Meter",
   "module": "DataBroker",
   "permissionTag": "Create",
   "tree": "ConfigTree"
}
```

13.3.3 Update Meter entries (PUT)

URL: /api/default/config/meters('id')/
Method: PUT
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter
Binding parameter (id): Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.id
There are 1 type(s) that can be modified: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter

Request Structure

The request structure is the same as the GET response above. Some fields may be omitted if not POSTable. Any POSTable fields not included will be reset to the default state.

User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Meter",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "ConfigTree"
}
```

13.3.4 Update Meter entries (PATCH)

URL: /api/default/config/meters('id')/
Method: PATCH
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter
Binding parameter (id): Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.id
There are 1 type(s) that can be modified: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter
The following fields are patchable:

- displayName
- enabled
- flags
- meterBands
- meterId
- node
- tags

Request Structure

The request structure includes one or more properties that are patchable. Any property not included in the request structure will retain its current value (subject to any property relationships).

User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Meter",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Meter",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "ConfigTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Meter",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "ConfigTree"
}
```

13.3.5 Delete Meter entries

URL: /api/default/config/meters('id')/
Method: DELETE
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter
Binding parameter (id): Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.id

User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "Meter",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "ConfigTree"
}
```

13.3.6 Read meter descriptions

URL: /api/default/operational/meterDescriptions/
Method: GET
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions
There are 1 type(s) that can be returned: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "dataPathId":  Edm.Int64,
    "id":  Edm.String,
    "meterDescriptionsList":  [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter-
Desc, ],
    "transactionId":  Edm.Int64,
  , ]
}
```

User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "MeterDescriptions",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

13.3.7 Read meter features

URL: /api/default/operational/meterFeatures/
Method: GET
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures
There are 1 type(s) that can be returned: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "bandTypes":  Sel.Sel5056.OpenFlowPlugin.Enums.OF13MeterBandType,
    "capabilities":  [ Sel.Sel5056.TopologyManager.OpenFlow.Meter.-
RateTypeFlagObjects.RateTypeBase, ],
    "dataPathId":  Edm.Int64,
    "id":  Edm.String,
    "maxBands":  Edm.Byte,
    "maxColor":  Edm.Byte,
    "maxMeters":  Edm.Int64,
  , ]
}
```

User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "MeterFeatures",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

13.3.8 Read meter diagnostics

URL: /api/default/operational/meterstats/
Method: GET
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats
There are 1 type(s) that can be returned: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats
The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "dataPathId":  Edm.Int64,
    "id":  Edm.String,
    "meterStatList":  [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStat, ],
    "transactionId":  Edm.Int64,
  , ]
}
```

User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "MeterStats",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

Chapter 14

Controller Management

Licensing has been removed in v2.2.0.

14.1 Commissioning

purpose	endpoints	methods
read	/security/IsCommissioned	POST

14.1.1 Commissioned status of SEL-5056

URL: /api/default/security/IsCommissioned
Method: POST
OData Type: Sel.IsCommissioned
There are 1 type(s) that can be POSTed: Sel.IsCommissioned

Request Structure

No request body

User Roles with Required Permissions

- Monitor
- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

None

14.2 Database Management

purpose	endpoints	methods
backup	/settings/Backup	POST
restore	/settings/Restore	POST

14.2.1 Backup the SEL-5056 database

URL: /api/default/settings/Backup
Method: POST
OData Type: Sel.Backup
There are 1 type(s) that can be POSTed: Sel.Backup

Request Structure

```
{
  "password": Edm.String,
}
```

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "module": "TrustAuthority",
   "permissionTag": "Backup"
}
```

14.2.2 Restore a SEL-5056 database

URL: /api/default/settings/Restore
Method: POST
OData Type: Sel.Restore
There are 1 type(s) that can be POSTed: Sel.Restore

Request Structure

Database restore file upload uses the multipart request message format (RFC 7578).

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "module": "TrustAuthority",
   "permissionTag": "Restore"
}
```

Examples

Restore the SEL-5056 database Example 22.

```
POST /api/default/settings/Restore HTTP/1.1
Content-Type: multipart/form-data; boundary
   =---------------------------226827793616512233304021929274
HOST: 192.168.56.1
Authorization: Bearer
   a915eb55bd8d304db3762d4f0b440b7d5b796c22869c2d9cd7c343ad51bd9583
Content-Length: 388

-----------------------------226827793616512233304021929274
Content-Disposition: form-data; name="password"


-----------------------------226827793616512233304021929274
Content-Disposition: form-data; name="fileName"; filename="database.
   zip"
Content-Type: application/x-zip-compressed

<--Database file contents-->
-----------------------------226827793616512233304021929274--
```

Listing 14.1: Restore the SEL-5056 database

14.3 Event Categories

purpose	endpoints	methods
read	/settings/eventCategories/ /settings/eventCategories('id')/	GET GET
update	/settings/eventCategories('id')/ /settings/eventCategories('id')/	PATCH PUT
delete	Not supported	
create	Not supported	

14.3.1 Read event categories

URL: /api/default/settings/eventCategories/
Method: GET
OData Type: Sel.BlueFrame.Core.EventBus.EventCategory
There are 1 type(s) that can be returned: Sel.BlueFrame.Core.EventBus.EventCategory The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "behaviors":  [ Sel.BlueFrame.Core.EventBus.Behaviors.Behavior, ],
    "id":  Edm.String,
    "key":  Edm.String,
  , ]
}
```

User Roles with Required Permissions

- Monitor
- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "EventCategory",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SettingsTree"
}
```

14.3.2 Update event categories (PATCH)

URL: /api/default/settings/eventCategories('id')/
Method: PATCH
OData Type: Sel.BlueFrame.Core.EventBus.EventCategory
Binding parameter (id): Sel.BlueFrame.Core.EventBus.EventCategory.id
There are 1 type(s) that can be modified: Sel.BlueFrame.Core.EventBus.EventCategory
The following fields are patchable:

- behaviors

Request Structure

The request structure includes one or more properties that are patchable. Any property not included in the request structure will retain its current value (subject to any property relationships).

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "EventCategory",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SettingsTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "EventCategory",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "SettingsTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "EventCategory",
   "module": "DataBroker",
   "permissionTag": "Subscribe",
   "tree": "SettingsTree"
}
```

14.3.3 Update event categories (PUT)

URL: /api/default/settings/eventCategories('id')/
Method: PUT
OData Type: Sel.BlueFrame.Core.EventBus.EventCategory
Binding parameter (id): Sel.BlueFrame.Core.EventBus.EventCategory.id
There are 1 type(s) that can be modified: Sel.BlueFrame.Core.EventBus.EventCategory

Request Structure

The request structure is the same as the GET response above. Some fields may be omitted if not POSTable. Any POSTable fields not included will be reset to the default state.

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "EventCategory",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "SettingsTree"
}
```

14.4 Event Types

purpose	endpoints	methods
read	/settings/eventTypes/ /settings/eventTypes('id')/	GET GET
update	Not supported	
delete	Not supported	
create	Not supported	

14.4.1 Read event categories

URL: /api/default/settings/eventTypes/
Method: GET
OData Type: Sel.BlueFrame.Core.EventBus.EventType
There are 2 type(s) that can be returned: Sel.BlueFrame.Core.EventBus.EventType, Sel.BlueFrame.Core.EventBus.ClearedEventType The return will be a list of one or more of these types show below

Response Structure for Sel.BlueFrame.Core.EventBus.EventType

```
{
  [
    "categoryKey":  Edm.String,
    "durationSetting":  Sel.BlueFrame.Core.EventBus.Enums.DurationType,
    "eventKey":  Edm.String,
    "facility":  Sel.BlueFrame.Core.EventBus.Enums.FacilityCode,
    "id":  Edm.String,
    "messageId":  Edm.String,
    "module":  Edm.String,
    "severity":  Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel,
  , ]
}
```

Response Structure for Sel.BlueFrame.Core.EventBus.ClearedEventType

```
{
  "@odata.type":  "#Sel.BlueFrame.Core.EventBus.ClearedEventType",
  "linkedEventKey":  Edm.String,
  "categoryKey":  Edm.String,
  "durationSetting":  Sel.BlueFrame.Core.EventBus.Enums.DurationType,
  "eventKey":  Edm.String,
  "facility":  Sel.BlueFrame.Core.EventBus.Enums.FacilityCode,
  "id":  Edm.String,
  "messageId":  Edm.String,
  "module":  Edm.String,
  "severity":  Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel,
}
```

User Roles with Required Permissions

- Monitor
- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "EventType",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SettingsTree"
}
```

14.5 Session Management

purpose	endpoints	methods
read	/settings/securityManager/ /settings/securityManager('id')/ /settings/webSettings/ /settings/webSettings('id')/	GET GET GET GET
update	/settings/securityManager('id')/ /settings/securityManager('id')/ /settings/webSettings('id')/ /settings/webSettings('id')/	PATCH PUT PATCH PUT
delete	Not supported	
create	Not supported	

14.5.1 Read maximum login attempts, lockout seconds, and login attempt windows

URL: /api/default/settings/securityManager/
Method: GET
OData Type: Sel.BlueFrame.SecurityManager.SecurityManagerSettings
There are 1 type(s) that can be returned: Sel.BlueFrame.SecurityManager.SecurityManagerSettings
The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "id":  Edm.String,
    "lockoutSeconds":  Edm.Int32,
    "loginAttemptWindow":  Edm.Int32,
    "maximumLoginAttempts":  Edm.Int32,
  , ]
}
```

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "SecurityManagerSettings",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SettingsTree"
}
```

14.5.2 Read usage policy, maximum number of concurrent sessions, and session timeout

URL: /api/default/settings/webSettings/
Method: GET
OData Type: Sel.BlueFrame.RestBroker.Plumbing.RestSettings
There are 1 type(s) that can be returned: Sel.BlueFrame.RestBroker.Plumbing.RestSettings The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "id":  Edm.String,
    "loginBanner":  Edm.String,
    "maxSessions":  Edm.Int64,
    "sessionTimeoutMinutes":  Edm.Int64,
  , ]
}
```

User Roles with Required Permissions

- Monitor
- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "RestSettings",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SettingsTree"
}
```

14.5.3 Update maximum login attempts, lockout seconds, and login attempt windows (PATCH)

URL: /api/default/settings/securityManager('id')/
Method: PATCH
OData Type: Sel.BlueFrame.SecurityManager.SecurityManagerSettings
Binding parameter (id): Sel.BlueFrame.SecurityManager.SecurityManagerSettings.id
There are 1 type(s) that can be modified: Sel.BlueFrame.SecurityManager.SecurityManagerSettings
The following fields are patchable:

- lockoutSeconds
- loginAttemptWindow
- maximumLoginAttempts

Request Structure

The request structure includes one or more properties that are patchable. Any property not included in the request structure will retain its current value (subject to any property relationships).

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "SecurityManagerSettings",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SettingsTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "SecurityManagerSettings",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "SettingsTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "SecurityManagerSettings",
   "module": "DataBroker",
   "permissionTag": "Subscribe",
   "tree": "SettingsTree"
}
```

14.5.4 Update maximum login attempts, lockout seconds, and login attempt windows (PUT)

URL: /api/default/settings/securityManager('id')/
Method: PUT
OData Type: Sel.BlueFrame.SecurityManager.SecurityManagerSettings
Binding parameter (id): Sel.BlueFrame.SecurityManager.SecurityManagerSettings.id
There are 1 type(s) that can be modified: Sel.BlueFrame.SecurityManager.SecurityManagerSettings

Request Structure

The request structure is the same as the GET response above. Some fields may be omitted if not POSTable. Any POSTable fields not included will be reset to the default state.

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "SecurityManagerSettings",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "SettingsTree"
}
```

14.5.5 Update usage policy, maximum number of concurrent sessions, and session timeout (PATCH)

URL: /api/default/settings/webSettings('id')/
Method: PATCH
OData Type: Sel.BlueFrame.RestBroker.Plumbing.RestSettings
Binding parameter (id): Sel.BlueFrame.RestBroker.Plumbing.RestSettings.id
There are 1 type(s) that can be modified: Sel.BlueFrame.RestBroker.Plumbing.RestSettings
The following fields are patchable:

- loginBanner
- maxSessions
- sessionTimeoutMinutes

Request Structure

The request structure includes one or more properties that are patchable. Any property not included in the request structure will retain its current value (subject to any property relationships).

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "RestSettings",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SettingsTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "RestSettings",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "SettingsTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "RestSettings",
   "module": "DataBroker",
   "permissionTag": "Subscribe",
   "tree": "SettingsTree"
}
```

14.5.6 Update usage policy, maximum number of concurrent sessions, and session timeout (PUT)

URL: /api/default/settings/webSettings('id')/
Method: PUT
OData Type: Sel.BlueFrame.RestBroker.Plumbing.RestSettings
Binding parameter (id): Sel.BlueFrame.RestBroker.Plumbing.RestSettings.id
There are 1 type(s) that can be modified: Sel.BlueFrame.RestBroker.Plumbing.RestSettings

Request Structure

The request structure is the same as the GET response above. Some fields may be omitted if not POSTable. Any POSTable fields not included will be reset to the default state.

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "RestSettings",
   "module": "DataBroker",
   "permissionTag": "Update",
   "tree": "SettingsTree"
}
```

14.6 Strings

purpose	endpoints	methods
read	/settings/strings/ /settings/strings('id')/	GET GET
update	Not supported	
delete	Not supported	
create	Not supported	

14.6.1 Read localization strings

URL: /api/default/settings/strings/
Method: GET
OData Type: Sel.BlueFrame.Core.EventBus.LocaleString
There are 1 type(s) that can be returned: Sel.BlueFrame.Core.EventBus.LocaleString The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "id":  Edm.String,
    "key":  Edm.String,
    "records":  [ Sel.BlueFrame.Core.EventBus.LocaleRecord, ],
    "variables":  [ Edm.String, ],
  , ]
}
```

User Roles with Required Permissions

- Monitor
- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "LocaleString",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SettingsTree"
}
```

14.7 System Messages

purpose	endpoints	methods
read	/settings/systemMessages/	GET
update	Not supported	
delete	Not supported	
create	Not supported	

14.7.1 Read system messages

URL: /api/default/settings/systemMessages/
Method: GET
OData Type: Sel.Sel5056.SystemMessaging.SystemMessage
There are 1 type(s) that can be returned: Sel.Sel5056.SystemMessaging.SystemMessage The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "id":  Edm.String,
    "message":  Edm.String,
    "module":  Edm.String,
    "severity":  Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel,
    "type":  Edm.String,
  , ]
}
```

User Roles with Required Permissions

- Monitor
- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "SystemMessage",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SettingsTree"
}
```

Chapter 15

Packet In/Out

15.1 Packet In/Out Management

purpose	endpoints	methods
read	/operational/packets/	GET
create	/operational/nodes('id')/PacketOut	POST
update	Not supported	
delete	Not supported	

15.1.1 Read Capture packets sent to the controller

URL: /api/default/operational/packets/
Method: GET
OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ReceivedPacket
There are 1 type(s) that can be returned: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ReceivedPacket The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "contents":  Edm.String,
    "cookie":  Edm.Int64,
    "dataPathId":  Edm.Int64,
    "id":  Edm.String,
    "monotonicId":  Edm.Int64,
    "port":  Edm.Int64,
    "prettyDataPathId":  Edm.String,
    "receivedAt":  Edm.Int64,
  , ]
}
```

User Roles with Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ReceivedPacket",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
}
```

15.1.2 Inject a packet into the network

URL: /api/default/operational/nodes('id')/PacketOut
Method: POST
OData Type: Sel.PacketOut
Binding parameter (id): Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id
There are 1 type(s) that can be POSTed: Sel.PacketOut

Request Structure

```
{
  "packetData":  Edm.String,
  "portId":  Edm.Int64,
}
```

User Roles with Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "OperationalNetworkNode",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "OperationalTree"
},
{
   "module": "OpenFlowPlugin",
   "permissionTag": "SendPacketOut"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "LogicalConnection",
   "module": "DataBroker",
   "permissionTag": "Create",
   "tree": "ConfigTree"
}
```

Chapter 16

Application Management

An application allows another method to get a token. Once an application or user obtains a token, the only difference is only what permissions as allowed for the token.
In general applications provide four benefits:

- Use of the Client Credentials and Authorization Code grant types
- Control over the permissions that the application has, versus the three fixed roles that user have access through the Resource Owner Password Credentials grant type
- Profile listed on the Application Management page.
- Client side X.509 Certificates

There are two methods for registering an application. You may have the Controller GET the information from the application or you may POST the information to the Controller.

16.1 Application Management

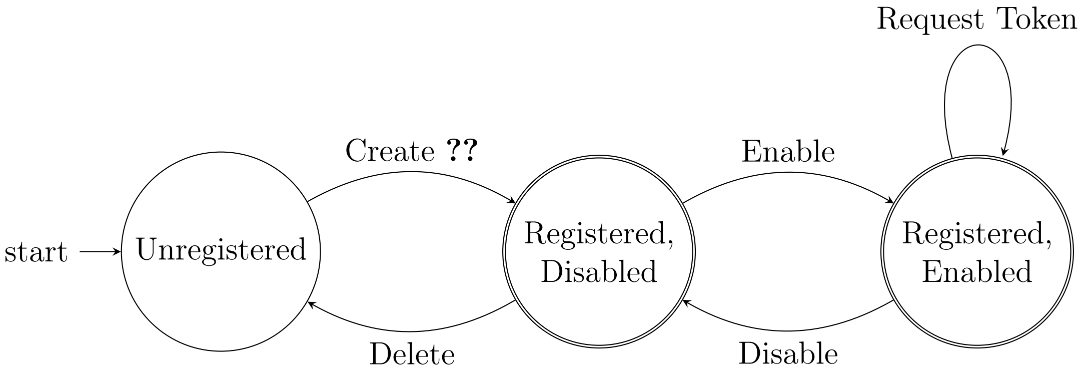
An application allows another method to get a token. Once an application or user obtains a token, the only difference is only what permissions as allowed for the token.
In general applications provide four benefits:

- Use of the Client Credentials and Authorization Code grant types
- Control over the permissions that the application has, versus the three fixed roles that user have access through the Resource Owner Password Credentials grant type
- Profile listed on the Application Management page.
- Client side X.509 Certificates

There are two methods for registering an application. You may have the Controller GET the information from the application or you may POST the information to the Controller.

purpose	endpoints	methods

create	/security/applicationLinks/InitiateApplicationRegistration /security/applicationLinks/	POST POST
delete	/security/applicationLinks('id')/	DELETE
read	/security/applicationLinks/ /security/applicationLinks('id')/	GET GET
update	Not supported	
enable	/security/applicationLinks('id')/EnableApplicationLink	POST
disable	/security/applicationLinks('id')/DisableApplicationLink	POST



16.1.1 Request application registration information from an external URL

URL: /api/default/security/applicationLinks/InitiateApplicationRegistration
Method: POST
OData Type: Sel.InitiateApplicationRegistration
There are 1 type(s) that can be POSTed: Sel.InitiateApplicationRegistration

Request Structure

```
{
  "url":  Edm.String,
}
```

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "module": "ApplicationRegistration",
   "permissionTag": "InitiateApplicationRegistration"
}
```

Troubleshooting

BadRequestError: Unable to validate certificate for remote application
The client certificate used by the web site serving the registration information is not present on the SEL-5056 or is revoked.

If the certification has been previously revoked and reuploaded, you may need to manually remove it from the database before reuploaded.

16.1.2 Create registered applications

URL: /api/default/security/applicationLinks/
Method: POST
OData Type: Sel.BlueFrame.ApplicationRegistration.ApplicationLink
There are 1 type(s) that can be POSTed: Sel.BlueFrame.ApplicationRegistration.ApplicationLink

Request Structure

Property	Default Value
description	null

Table 16.3: Default Values for Sel.BlueFrame.ApplicationRegistration.ApplicationLink

```
{
  "applicationIp":  Edm.String (IPv4 Address),
  "applicationName":  Edm.String,
  "authenticationTypes":  [ Sel.BlueFrame.ApplicationRegistration.AuthenticationType, ],
  "certificateThumbprint":  Edm.String (X.509 Thumbprint),
  "description":  Edm.String,
  "versions":  [ Edm.String, ],
}
```

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ApplicationLink",
   "module": "DataBroker",
   "permissionTag": "Create",
   "tree": "SecurityTree"
}
```

16.1.3 Delete registered applications

URL: /api/default/security/applicationLinks('id')/
Method: DELETE
OData Type: Sel.BlueFrame.ApplicationRegistration.ApplicationLink
Binding parameter (id): Sel.BlueFrame.ApplicationRegistration.ApplicationLink.id

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ApplicationLink",
   "module": "DataBroker",
   "permissionTag": "Delete",
   "tree": "SecurityTree"
}
```

16.1.4 Read registered applications

URL: /api/default/security/applicationLinks/
Method: GET
OData Type: Sel.BlueFrame.ApplicationRegistration.ApplicationLink
There are 1 type(s) that can be returned: Sel.BlueFrame.ApplicationRegistration.ApplicationLink
The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "applicationIcon":  Edm.String (Base64 encoded),
    "applicationIp":  Edm.String (IPv4 Address),
    "applicationName":  Edm.String,
    "authenticationTypes":  [ Sel.BlueFrame.ApplicationRegistration.-
AuthenticationType, ],
    "certificateThumbprint":  Edm.String (X.509 Thumbprint),
    "description":  Edm.String,
    "enabled":  Edm.Boolean,
    "id":  Edm.String,
    "versions":  [ Edm.String, ],
  , ]
}
```

User Roles with Required Permissions

- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "ApplicationLink",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SecurityTree"
}
```

16.1.5 Enable an application

URL: /api/default/security/applicationLinks('id')/EnableApplicationLink
Method: POST
OData Type: Sel.EnableApplicationLink
Binding parameter (id): Sel.BlueFrame.ApplicationRegistration.ApplicationLink.id
There are 1 type(s) that can be POSTed: Sel.EnableApplicationLink

Request Structure

No request body

User Roles with Required Permissions

- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

```
{
   "module": "ApplicationRegistration",
   "permissionTag": "EnableApplicationLink"
}
```

16.1.6 Disable an application

URL: /api/default/security/applicationLinks('id')/DisableApplicationLink
Method: POST
OData Type: Sel.DisableApplicationLink
Binding parameter (id): Sel.BlueFrame.ApplicationRegistration.ApplicationLink.id
There are 1 type(s) that can be POSTed: Sel.DisableApplicationLink

Request Structure

No request body

User Roles with Required Permissions

- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

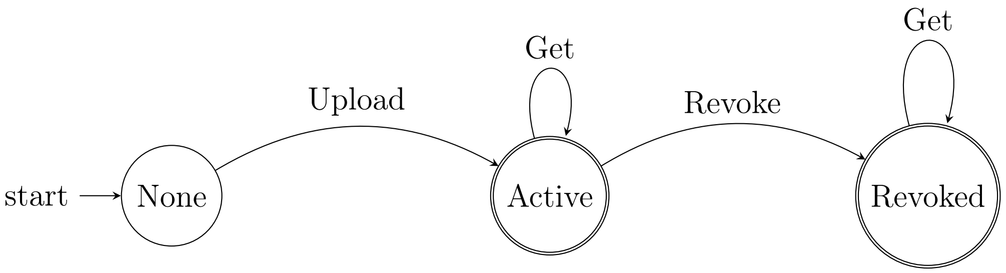
```
{
   "module": "ApplicationRegistration",
   "permissionTag": "DisableApplicationLink"
}
```

Chapter 17

Certificate Management

17.1 Certificate Management

purpose	endpoints	methods
get	/certificate/certificateInfo/ /certificate/certificateInfo('id')/AsPem /certificate/certificateInfo('id')/	GET POST GET
create	Not supported	
upload	/certificate/UploadCertificate	POST
delete	Not supported	
revoke	/certificate/certificateInfo('id')/RevokeCertificate /certificate/RevokeCertificateByPurpose /certificate/RevokeCertificateByThumbprint	POST POST POST
update	Not supported	



17.1.1 Read X.509 Certificates

URL: /api/default/certificate/certificateInfo/
Method: GET
OData Type: Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation
There are 1 type(s) that can be returned: Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation The return will be a list of one or more of these types show below

Response Structure

```
{
  [
    "certificateVersion":  Edm.Int32,
    "commonName":  Edm.String,
    "displayName":  Edm.String,
    "hasPrivateKey":  Edm.Boolean,
    "id":  Edm.String,
    "isSigningCertificate":  Edm.Boolean,
    "issuer":  Edm.String,
    "purpose":  Sel.BlueFrame.Core.TrustAuthority.CertificatePurpose,
    "purposeId":  Edm.String,
    "serialNumber":  Edm.String,
    "signatureAlgorithm":  Edm.String,
    "status":  Sel.BlueFrame.Core.TrustAuthority.Enums.CertificateStatus,
    "subject":  Edm.String,
    "thumbprint":  Edm.String,
    "userInteraction":  Sel.BlueFrame.Core.TrustAuthority.Enums.UserInteraction,
    "validEndDate":  Edm.DateTimeOffset,
    "validStartDate":  Edm.DateTimeOffset,
  , ]
}
```

User Roles with Required Permissions

- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "CertificateInformation",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "CertificateTree"
}
```

17.1.2 Get a certificate in PEM format

URL: /api/default/certificate/certificateInfo('id')/AsPem
Method: POST
OData Type: Sel.AsPem
Binding parameter (id): Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation.id
There are 1 type(s) that can be POSTed: Sel.AsPem

Request Structure

No request body

User Roles with Required Permissions

- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "CertificateInformation",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "CertificateTree"
}
```

17.1.3 Upload an X.509 certificate

URL: /api/default/certificate/UploadCertificate
Method: POST
OData Type: Sel.UploadCertificate
There are 1 type(s) that can be POSTed: Sel.UploadCertificate

Request Structure

```
{
  "base64Certificate":  Edm.String,
  "certificatePassword":  Edm.String,
  "purpose":  Sel.BlueFrame.Core.TrustAuthority.CertificatePurpose,
}
```

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
   "module": "TrustAuthority",
   "permissionTag": "Replace",
   "purpose": "TrustedCertificatePurpose"
}
```

17.1.4 Revoke an X.509 certificate

URL: /api/default/certificate/certificateInfo('id')/RevokeCertificate
Method: POST
OData Type: Sel.RevokeCertificate
Binding parameter (id): Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation.id
There are 1 type(s) that can be POSTed: Sel.RevokeCertificate

Request Structure

No request body

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "CertificateInformation",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "CertificateTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
   "module": "TrustAuthority",
   "permissionTag": "Revoke",
   "purpose": "TrustedCertificatePurpose"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
   "module": "TrustAuthority",
   "permissionTag": "Replace",
   "purpose": "TrustedCertificatePurpose"
}
```

17.1.5 Revoke a X.509 certificate by its purpose

URL: /api/default/certificate/RevokeCertificateByPurpose
Method: POST
OData Type: Sel.RevokeCertificateByPurpose
There are 1 type(s) that can be POSTed: Sel.RevokeCertificateByPurpose

Request Structure

```
{
  "purpose":  Sel.BlueFrame.Core.TrustAuthority.CertificatePurpose,
}
```

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "CertificateInformation",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "CertificateTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
   "module": "TrustAuthority",
   "permissionTag": "Revoke",
   "purpose": "WebCertificatePurpose"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
   "module": "TrustAuthority",
   "permissionTag": "Replace",
   "purpose": "WebCertificatePurpose"
}
```

17.1.6 Revoke an X.509 certificate by its thumprint

URL: /api/default/certificate/RevokeCertificateByThumbprint
Method: POST
OData Type: Sel.RevokeCertificateByThumbprint
There are 1 type(s) that can be POSTed: Sel.RevokeCertificateByThumbprint

Request Structure

```
{
  "thumbprint":  Edm.String,
}
```

User Roles with Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "CertificateInformation",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "CertificateTree"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
   "module": "TrustAuthority",
   "permissionTag": "Revoke",
   "purpose": "TrustedCertificatePurpose"
},
{
   "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
   "module": "TrustAuthority",
   "permissionTag": "Replace",
   "purpose": "TrustedCertificatePurpose"
}
```

Part IV

Additional Topics

This part covers additional functionality.

Chapter 18

Situational Awareness, Diagnostics, and Counters

This section summarizes the situational function of the SEL-5056. Situational awareness may be divided into several parts, which include diagnostics and counters.

18.1 Counters

Counters represent measurements of some network function. Counters can be divided into two types:

1. OpenFlow
2. Non-OpenFlow

Non-OpenFlow counters will be covered in a separate document.

OpenFlow counters can be divided into two types:

1. Direct, those that are defined in the OpenFlow specification
2. Indirect, organized Openflow

18.1.1 Direct OpenFlow Counters

The OpenFlow counters include the following:

Counter	Description	URL
Flows	Byte and Packet counts for those that are executed against the flow entry	/operational/flowStats/
Groups	Byte and Packet counts for those sent to the group entry	/operational/groupStats/
Action Bucket	Byte and Packet counts for those sent to the action bucket*	/operational/groupStats/
Meter	Byte and Packet counts for those sent to the meter (but not necessarily dropped)	/operational/meterstats/

Meter Band	Byte and Packet counts for those applied to the meter band (such as Drop)	/operational/meterstats/
Flow Table	Number of flows in a table	/operational/tableStats/
Port	Egress and Ingress byte and packet count per OpenFlow port	/operational/portStats/

* Not applicable for SEL-274XS switches

18.1.2 Logical Connections

The flow and group diagnostics for LCs can be obtained through the GetLogicalConnectionStats endpoint. This is the same information that can be obtained using the direct OpenFlow endpoints, but they have been filtered to only those used by the LC.

18.2 Diagnostics

Objects can be divided into two types. Those that have:

1. Separate operational information for associated configuration object, such as Flow Stats (operational) for flow entries (configuration), and

2. Operational information included in the object itself (other trees),

18.2.1 OpenFlow

In OpenFlow, group and meter descriptions as stored separately from the group and meter entry statistics. They are combined in flow entries. The SEL-5056 exposes this information in the same manner. The descriptions have been converted to the data structure of the SEl-5056. Differences between what is in the database versus the descriptions cause synchronize requests.

Type	Statistics (Counters)	Descriptions
Flows	/operational/flowStats/	/operational/flowStats/
Groups (and their action buckets)	/operational/groupStats/	/operational/groupDesc/
Meter (and their bands)	/operational/meterstats/	/operational/meterDescriptions/

18.3 Situational Awareness

18.3.1 Logical Connections

The path plans used to the program the flow and group entries for an LC are available from the GetPathPlanInformation endpoint.

Chapter 19

Asynchronous Data Exchange with SignalR

SignalR is an API abstraction for support for asynchronous data exchange. SignalR is used to receive notification that an object has been created, modified, or deleted. SignalR does not replace the need to get the object(s), but rather replaces the need to poll the object(s) for changes. The notification only contains the id of the object, not the change. Subscriptions are through hubs.

> SignalR operates differently than v2.1 and is not backwards compatible. This section has been updated to address those changes.

The actually protocols implemented are one of the following:

1. Websocket (RFC 6455)
2. SSE (Server side events)
3. Long polling
4. Forever Frame

Which is chosen depends on a number of factors, such as the host OS for the SEL-5056 and the client web browser.
This section covers a basic example of setting up a WebSocket connection with the SEL-5056.

19.1 Hubs

You can subscribe to some of object endpoints defined on the hub.
Hubs located at: /signalr/hubs.
TODO LIST OF HUBS and what they mean

19.2 Authorization

For authorization, a valid token obtained with the required permissions must be passed into the connect HTTP call of the form: Cookie: BearerToken=X, where X is the OAuth token.
For example, if the OAuth token obtained was 21bac929e567f84c459c8be5c67af2a0, then the header would be Cookie: BearerToken=21bac929e567f84c459c8be5c67af2a0

19.3 Permissions

There are a set of permissions specific to SignalR. If the user role has access to an endpoint, then the user also has permission to subscribe to the endpoint over SignalR, if a hub exists for it.
TODO LIST OF PERMISSIONS

19.4 WebSocket

Example 23.
The following will turn the HTTP socket into a websocket socket for subscribing to all flow entry changes.
First negotiate signalR

```
GET /signalr/negotiate?clientProtocol=1.5&connectionData=%5B%7B%22
   name%22%3A%22default-config-flows%22%7D%5D HTTP/1.1
HOST: 192.168.255.2



HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
Server: Microsoft-HTTPAPI/2.0
X-Content-Type-Options: nosniff
Date: Mon, 28 Oct 2019 22:53:35 GMT



267
{"Url":"/signalr","ConnectionToken":"AQAAANCMnd8BFdERjHoAwE/Cl+
   sBAAAAjJw60GhBKUG3qNIuuhcccAAAAAACAAAAAAAQZgAAAAEAACAAAAC6z5eMEeylGX9onLs
   +35AAAAAAOgAAAAAIAACAAAABZm2W8oa9vQi1VYFE4YbMOXFT3w+
   RDld8MDFF1xpfDazAAAADKzgQbXQivdReQAiUQvpjod0NSp1xKftb/
   OplXHLZIPiYJP3JUQ1Sm31zxwYn97QNAAAAAX4bSAogCl9BXGpSoX7U+L9s+
   eu1MeZOg2HdobUa/1DrrIIIXdnGNlJWFi95/I+96J5X0/QCkOoKrijY/KFl9ng
   ==","ConnectionId":"2e0b3834-54ae-4215-b11e-641bce56ddf1","
   KeepAliveTimeout":20.0,"DisconnectTimeout":30.0,"
   ConnectionTimeout":110.0,"TryWebSockets":true,"ProtocolVersion
   ":"1.5","TransportConnectTimeout":5.0,"LongPollDelay":0.0}
0
```

Listing 19.1: "Negotiating SignalR"

Then choose web sockets, passing in the OAuth token in a cookie header. The response body itself is formatted according to websocket standard.

```
GET /signalr/connect?transport=webSockets&clientProtocol=1.5&
   connectionToken=AQAAANCMnd8BFdERjHoAwE/Cl%2
   BsBAAAAjJw60GhBKUG3qNIuuhcccAAAAAACAAAAAAAQZgAAAAEAACAAAAC6z5eMEeylGX9onL
   %2B35AAAAAAOgAAAAAIAACAAAABZm2W8oa9vQi1VYFE4YbMOXFT3w%2
   BRDld8MDFF1xpfDazAAAADKzgQbXQivdReQAiUQvpjod0NSp1xKftb/
   OplXHLZIPiYJP3JUQ1Sm31zxwYn97QNAAAAAX4bSAogCl9BXGpSoX7U%2BL9s%2
   Beu1MeZOg2HdobUa/1DrrIIIXdnGNlJWFi95/I%2B96J5X0/QCkOoKrijY/KFl9ng
   %3D%3D&connectionData=%5B%7B%22name%22%3A%22default-config-flows
   %22%7D%5D&tid=3 HTTP/1.1
Origin: https://192.168.255.2
Sec-WebSocket-Key: oDwoWeKbFWhQ8jjeRC/VQQ==
Sec-WebSocket-Version: 13
Upgrade: websocket
Connection: Upgrade
Cookie: BearerToken=21bac929e567f84c459c8be5c67af2a0
HOST: 192.168.255.2


HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Server: Microsoft-HTTPAPI/2.0
Access-Control-Allow-Origin: https://192.168.255.2
Access-Control-Allow-Credentials: true
X-Content-Type-Options: nosniff
Connection: Upgrade
Sec-WebSocket-Accept: 21zNErwhWrRp/nOsgheD5yoUuEU=
Date: Mon, 28 Oct 2019 22:53:35 GMT


.{"C":"d-F0C04A7E-EY,0|Fm,0|Fn,1","S":1,"M":[]}
```

Listing 19.2: "Selecting websockets"

At this point the socket is now operating under the Websocket protocol and no longer HTTP. The next step is to send a subscribe message to the SEL-5056.
For flow entries, the hub at /signalr/hubs has the following:

```
{
      "Name": "default-config-flows",
      "ClientMethods": null,
      "ServerMethods": [
        {
          "Name": "SubscribeAll",
          "Parameters": []
        },
        {
          "Name": "UnsubscribeAll",
          "Parameters": []
        },
        {
          "Name": "SubscribeAllCreates",
          "Parameters": []
        },
        {
          "Name": "UnsubscribeAllCreates",
          "Parameters": []
        },
        {
          "Name": "SubscribeTo",
          "Parameters": [
            {
              "Name": "ids",
              "Type": "Array"
            }
          ]
        },
        {
          "Name": "UnsubscribeTo",
          "Parameters": [
            {
              "Name": "ids",
              "Type": "Array"
            }
          ]
        }
      ]
    }
```

Listing 19.3: "default-config-flows hub"

We want to subscribe to all. SubscribeAll does not have any arguments (parameters), so the message we need to send to the SEL-5056 through the Websocket (and no longer HTTP) is:

```
{"H":"default-config-flows","M":"SubscribeAll","A":[],"I":0}
```

Listing 19.4: "Subscribe to /config/flows"

Websockets requires a header and masking. Assuming that the masking value is 0xe40e9be2, then the request is a string of the following octets:

```
0x81 0xbc 0xe4 0xe 0x9b
0xe2 0x9f 0x2c 0xd3 0xc0
0xde 0x2c 0xff 0x87 0x82
0x6f 0xee 0x8e 0x90 0x23
0xf8 0x8d 0x8a 0x68 0xf2
0x85 0xc9 0x68 0xf7 0x8d
0x93 0x7d 0xb9 0xce 0xc6
0x43 0xb9 0xd8 0xc6 0x5d
0xee 0x80 0x97 0x6d 0xe9
0x8b 0x86 0x6b 0xda 0x8e
0x88 0x2c 0xb7 0xc0 0xa5
0x2c 0xa1 0xb9 0xb9 0x22
0xb9 0xab 0xc6 0x34 0xab
0x9f
```

Listing 19.5: "Masked Subscribe to /config/flows"

Ignoring the header, when a flow entry with the id of a8f40cc11155640c193bf7a65350aecd has been updated, the following message from the SEL-5056 is received:

```
{"C":"d-F0C04A7E-EY,0|Fm,0|Fn,2|FO,C","M":[{"H":"default-config-
   flows","M":"OnUpdate","A":["default","config","flows","
   a8f40cc11155640c193bf7a65350aecd"]}]}
```

Listing 19.6: "Notification of Flow Update"

Chapter 20

Permissions and Roles

Permissions are objects that represent the ability to read, modify, delete, or create objects. The SEL-5056 permission specifically is based on ability to use a method for a URI, which a few exceptions based on method, URI and object type.

Each token is associated with a list of permissions that are inherited from the permissions assigned to the role that a user used to log in or the list of permissions requested when registering an application.

For example, the permission

```
{
   "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
   "dataType": "User",
   "module": "DataBroker",
   "permissionTag": "Read",
   "tree": "SecurityTree"
}
```

allows the bearer to read users from the REST interface at the endpoint /api/default/security/users and method GET, as shown in the read users section.

Each of the roles are assigned a fix set of permissions. This list is available on the REST interface using the read user roles endpoint. The permissions available to a registered application can be found using the read registered applications endpoint.

Some endpoints do not require permissions to use and some require multiple permissions to use. Each endpoint described in endpoints section under Permissions for Registered Applications contain the permissions required to use the endpoints and method combination for registered applications that may have a variable set of permissions. For users with roles, the User Roles with Required Permissions contain the roles that can use the endpoint and method combination.

Part V

OData Definitions

This part covers the OData objects.
And is divided into 4 parts:

Chapter 21

Native OData Types

This short chapter covers the types defined by the OData standard that are also used by the SEL-5056. All of these types begin with Edm. More information can be found in the OData standard.

These types define ranges or sets of valid values. However, additional constraints may be present, some, but not all of which are listed in the following chapters of this part of the manual.

Name	Form	description
Edm.Boolean	true|false	Boolean
Edm.Byte	[0-9]+	Unsigned 8-bit integer
Edm.DateTimeOffset	See below	Date/time value with a time-zone offset
Edm.Duration	See below	Time duration
Edm.Int32	[0-9]+	Signed 32-bit integer
Edm.Int64	[0-9]+	Signed 64-bit integer
Edm.String	list of UTF-8 characters	String

Table 21.2: Edm.* Types

21.1 Edm.DateTimeOffset

TODO

21.2 Edm.Duration

TODO
P11DT23H59M59.999999999999S

Chapter 22

Primary OData Types

This section describes the OData types that are base types, except for those that describe the endpoints.

22.1 Sel.BlueFrame.ApplicationRegistration.ApplicationLink

name	attributes	description
applicationIcon	Type: Edm.String	Icon
applicationIp	DataClass: IpAddress Type: Edm.String	IPv4 address that the SEL-5056 sees when the application connects to the SEL-5056
applicationName	CharacterSet: Printable7BitAscii MaxLength: 48 Type: Edm.String	Name of application
authenticationTypes	MinLength: 1 Type: [Sel.BlueFrame.ApplicationRegistration.AuthenticationType]	TODO
certificateThumbprint	Type: Edm.String	Thumbprint of the X.509 certificate used to register
description	MaxLength: 4096 Type: Edm.String	String shown in the Application Management page
enabled	Type: Edm.Boolean	Shos if the application is enabled
id	Type: Edm.String	Common Property
versions	MinLength: 1 Type: [Edm.String]	Currently not used

Table 22.2: Properties for Sel.BlueFrame.ApplicationRegistration.ApplicationLink

Connections:
Sel.BlueFrame.ApplicationRegistration.ApplicationLink
—Sel.DisableApplicationLink.bindingParameter
—Sel.EnableApplicationLink.bindingParameter
—Sel.InitiateApplicationRegistration.bindingParameter

22.2 Sel.BlueFrame.Core.EventBus.EventCategory

name	attributes	description
behaviors	Type: [Sel.BlueFrame.Core.EventBus.Behaviors.Behavior]	TODO
id	Type: Edm.String	Common Property
key	Type: Edm.String	TODO

Table 22.4: Properties for Sel.BlueFrame.Core.EventBus.EventCategory

Connections:
Sel.BlueFrame.Core.EventBus.EventCategory
—Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode.eventCategories
——Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

22.3 Sel.BlueFrame.Core.EventBus.EventType

name	attributes	description
categoryKey	Indexed: True Type: Edm.String	TODO
durationSetting	Type: Sel.BlueFrame.Core.EventBus.Enums.DurationType	TODO
eventKey	Type: Edm.String	TODO
facility	Indexed: True Type: Sel.BlueFrame.Core.EventBus.Enums.FacilityCode	TODO
id	Type: Edm.String	TODO
messageId	Type: Edm.String	TODO
module	Indexed: True Type: Edm.String	TODO
severity	Indexed: True Type: Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel	TODO

Table 22.6: Properties for Sel.BlueFrame.Core.EventBus.EventType

The following types inherit from this type:
-Sel.BlueFrame.Core.EventBus.ClearedEventType
Connections:
Sel.BlueFrame.Core.EventBus.EventType

22.3.1 Sel.BlueFrame.Core.EventBus.ClearedEventType

name	attributes	description
linkedEventKey	Nullable: FalseReadOnly: TrueType: Edm.String	TODO

Table 22.8: Sel.BlueFrame.Core.EventBus.ClearedEventType

22.4 Sel.BlueFrame.Core.EventBus.LocaleString

name	attributes	description
id	Type: Edm.String	TODO
key	Type: Edm.String WriteOnce: true	TODO
records	Type: [Sel.BlueFrame.Core.EventBus.LocaleRecord]	TODO
variables	Type: [Edm.String]	TODO

Table 22.10: Properties for Sel.BlueFrame.Core.EventBus.LocaleString

Connections:
Sel.BlueFrame.Core.EventBus.LocaleString

22.5 Sel.BlueFrame.Core.SecurityManager.AuthService

name	attributes	description
errorState	DefaultValue: InProgress Type: Sel.BlueFrame.Core.ErrorState	Common Property
errors	Type: [Edm.String]	Common Property
id	Type: Edm.String	Common Property

Table 22.12: Properties for Sel.BlueFrame.Core.SecurityManager.AuthService

The following types inherit from this type:
-Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService
Connections:
Sel.BlueFrame.Core.SecurityManager.AuthService
—Sel.GetAvailableGroups.bindingParameter
—Sel.TestAuthenticateUser.bindingParameter

22.5.1 Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService

name	attributes	description
authType	Nullable: False ReadOnly: True Type: System.DirectoryServices.Protocols.AuthType	TODO

bindDN	MaxLength: 256 Type: Edm.String	TODO
bindPassword	Type: Edm.String	TODO
email	DefaultValue: mail MaxLength: 256 Type: Edm.String	TODO
firstName	DefaultValue: givenName MaxLength: 256 Type: Edm.String	TODO
groupMemberAttribute	DefaultValue: memberOf MaxLength: 256 Type: Edm.String	TODO
hostname	Nullable: False Type: Edm.String	TODO
lastName	DefaultValue: sn MaxLength: 256 Type: Edm.String	TODO
name	MaxLength: 32 MinLength: 1 Nullable: False Type: Edm.String Unique: True	TODO
portNumber	DefaultValue: 389 Max: 65535 Min: 1 Nullable: False Step: 1 Type: Edm.Int32	TODO
protocolVersion	Nullable: False ReadOnly: True Type: Edm.Int32	TODO
searchBase	MaxLength: 256 Type: Edm.String	TODO
userIdFilter	DefaultValue: (sAMAccountName=USERNAME) MaxLength: 256 Type: Edm.String	TODO
workPhone	DefaultValue: telephoneNumber MaxLength: 256 Type: Edm.String	TODO

Table 22.14: Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService

22.6 Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup

name	attributes	description
authService	MaxLength: 256 Type: Sel.BlueFrame.Core.SecurityManager.AuthService WriteOnce: true	ID of associated authentication service
displayName	MaxLength: 32 Type: Edm.String Unique: True	Common Property
dn	MaxLength: 256 Type: Edm.String WriteOnce: true	Distinguished name
groupRoles	Type: Sel.BlueFrame.Core.SecurityManager.Role	User roles of the group
id	Type: Edm.String	Common Property

Table 22.16: Properties for Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup

Connections:
Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup

22.7 Sel.BlueFrame.Core.SecurityManager.Role

name	attributes	description
displayName	Type: Edm.String	Common Property
id	Type: Edm.String	Common Property
permissions	Type: [Sel.BlueFrame.Core.SecurityManager.Permission]	List of permissions
roleName	Type: Edm.String	Name of role

Table 22.18: Properties for Sel.BlueFrame.Core.SecurityManager.Role

Connections:
Sel.BlueFrame.Core.SecurityManager.Role

22.8 Sel.BlueFrame.Core.SecurityManager.User

name	attributes	description
displayName	Type: Edm.String	Common Property
enabled	DefaultValue: True Type: Edm.Boolean	User can log in
failedLogins	Type: [Sel.BlueFrame.Core.SecurityManager.UserFailedLogin]	Number of failed logins for user since creation
id	Type: Edm.String	Common Property
isLockedOut	Type: Edm.Boolean	Is the account temporarly disabled
lastSuccessLoginIp	Type: Edm.String	IP address of the last successful login attempt

lastSuccessfulLogin	Type: Edm.DateTimeOffset	Local time of the last successful login attempt
lockedOutUntil	Type: Edm.DateTimeOffset	If the account is locked, the local time when the account is unlocked
lockoutTime	Type: Edm.DateTimeOffset	If the account is locked, when the account was locked
roles	Type: Sel.BlueFrame.Core.SecurityManager.Role	Roles belonging to the user
username	CaseInsensitive: true CharacterSet: Printable7BitAscii Indexed: True MaxLength: 128 MinLength: 1 Type: Edm.String Unique: True	User name of the user

Table 22.20: Properties for Sel.BlueFrame.Core.SecurityManager.User

The following types inherit from this type:
-Sel.BlueFrame.Core.SecurityManager.LocalUser
-Sel.BlueFrame.Core.SecurityManager.AuthServiceUser
Connections:
Sel.BlueFrame.Core.SecurityManager.User
—Sel.UpdatePassword.bindingParameter

22.8.1 Sel.BlueFrame.Core.SecurityManager.LocalUser

name	attributes	description
clearTextPass	CharacterSet: Printable7BitAscii MaxLength: 128 MinLength: 8 Type: Edm.String	TODO

Table 22.22: Sel.BlueFrame.Core.SecurityManager.LocalUser

22.8.2 Sel.BlueFrame.Core.SecurityManager.AuthServiceUser

name	attributes	description
authService	Type: Sel.BlueFrame.Core.SecurityManager.AuthService	TODO

Table 22.24: Sel.BlueFrame.Core.SecurityManager.AuthServiceUser

22.9 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction

name	attributes	description
id	Type: Edm.String	TODO

Table 22.26: Properties for Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction

Connections:
Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
—Sel.Commit.bindingParameter
—Sel.ClearTransaction.bindingParameter
—Sel.Commit.bindingParameter
—Sel.ClearTransaction.bindingParameter
—Sel.Commit.bindingParameter
—Sel.ClearTransaction.bindingParameter
—Sel.Commit.bindingParameter
—Sel.ClearTransaction.bindingParameter
—Sel.Commit.bindingParameter
—Sel.ClearTransaction.bindingParameter

22.10 Sel.BlueFrame.RestBroker.Plumbing.RestSettings

name	attributes	description
id	Type: Edm.String	Common Property
loginBanner	MaxLength: 4095 MinLength: 0 Type: Edm.String	TODO
maxSessions	Type: Edm.Int64	TODO
sessionTimeoutMinutes	Type: Edm.Int64	TODO

Table 22.28: Properties for Sel.BlueFrame.RestBroker.Plumbing.RestSettings

Connections:
Sel.BlueFrame.RestBroker.Plumbing.RestSettings

22.11 Sel.BlueFrame.SecurityManager.SecurityManagerSettings

name	attributes	description
id	Type: Edm.String	Common Property

lockoutSeconds	Type: Edm.Int32	Common Property
loginAttemptWindow	Type: Edm.Int32	Common Property
maximumLoginAttempts	Type: Edm.Int32	Common Property

Table 22.30: Properties for Sel.BlueFrame.SecurityManager.SecurityManagerSettings
Connections:
Sel.BlueFrame.SecurityManager.SecurityManagerSettings

22.12 Sel.BlueFrame.TrustAuthority.DataTreeObjects.-CertificateInformation

name	attributes	description
certificateVersion	Type: Edm.Int32	TODO
commonName	Indexed: True Type: Edm.String	TODO
displayName	Type: Edm.String	Common Property
hasPrivateKey	Type: Edm.Boolean	TODO
id	Type: Edm.String	Common Property
isSigningCertificate	Type: Edm.Boolean	TODO
issuer	Type: Edm.String	TODO
purpose	Type: Sel.BlueFrame.Core.TrustAuthority.-CertificatePurpose	TODO
purposeId	Indexed: True Type: Edm.String	TODO
serialNumber	Type: Edm.String	TODO
signatureAlgorithm	Type: Edm.String	TODO
status	Type: Sel.BlueFrame.Core.TrustAuthority.-Enums.CertificateStatus	Common Property
subject	Type: Edm.String	TODO
thumbprint	Indexed: True Type: Edm.String	TODO
userInteraction	Type: Sel.BlueFrame.Core.TrustAuthority.-Enums.UserInteraction	TODO
validEndDate	Type: Edm.DateTimeOffset	TODO
validStartDate	Type: Edm.DateTimeOffset	TODO

Table 22.32: Properties for Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation
Connections:
Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation
—Sel.AsPem.bindingParameter

—Sel.RevokeCertificate.bindingParameter

22.13 Sel.BlueFrame.UserPreferences.Preference

name	attributes	description
id	Type: Edm.String	TODO
registry	Type: [Sel.BlueFrame.UserPreferences.Setting]	TODO
username	Type: Edm.String WriteOnce: true	TODO

Table 22.34: Properties for Sel.BlueFrame.UserPreferences.Preference

Connections:
Sel.BlueFrame.UserPreferences.Preference

22.14 Sel.Sel5056.LearnAndLock.LearnAndLockSettings

name	attributes	description
id	Type: Edm.String	TODO
includeControllerInLearningRegion	DefaultValue: TrueType: Edm.Boolean	TODO
performAutoAdoption	DefaultValue: TrueType: Edm.Boolean	TODO
performLogicalConnectionLearning	DefaultValue: TrueType: Edm.Boolean	TODO
performNetworkReset	DefaultValue: FalseType: Edm.Boolean	TODO

Table 22.36: Properties for Sel.Sel5056.LearnAndLock.LearnAndLockSettings

Connections:
Sel.Sel5056.LearnAndLock.LearnAndLockSettings

22.15 Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.-LearnedLogicalConnection

name	attributes	description
destinationConfigNodeKeys	Type: [Edm.String]	TODO
destinationIps	Type: [Edm.String]	TODO
dontProposeReverse	Type: Edm.Boolean	TODO
id	Type: Edm.String	TODO
lastModifiedBy	Type: Edm.String	TODO
linkedCSTKey	Type: Edm.String	TODO
linkedLCKey	Type: Edm.String	TODO

linkedLLSessionKey	Type: Edm.String	TODO
proposedCommunicationServiceType	Type: Sel.Sel5056.LearnAndLock.ProposedCommunicationServiceType	TODO
sourceConfigNodeKeys	Type: [Edm.String]	TODO
sourceIps	Type: [Edm.String]	TODO
state	Type: Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnectionState	TODO

Table 22.38: Properties for Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection
Connections:
Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection

22.16 Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

name	attributes	description
adoptedHostsInfo	Type: [Sel.Sel5056.LearnAndLock.CSV.AdoptedHostInfo]	TODO
adoptedLinksInfo	Type: [Sel.Sel5056.LearnAndLock.CSV.AdoptedLinkInfo]	TODO
adoptedSwitchesInfo	Type: [Sel.Sel5056.LearnAndLock.CSV.AdoptedSwitchInfo]	TODO
autoAdoptConfig	Type: Sel.Sel5056.LearnAndLock.Configuration.AutoAdoptConfig	TODO
autoAdoptSession	Type: Sel.Sel5056.LearnAndLock.Sessions.AutoAdoptSession	TODO
autoAdoptionEndReason	Type: Sel.Sel5056.LearnAndLock.Enums.EndReason	TODO
autoAdoptionEndedByUserOrForReason	Type: Edm.String	TODO
autoAdoptionWasRun	Type: Edm.Boolean	TODO
beginSessionTime	Type: Edm.DateTimeOffset	TODO
endSessionTime	Type: Edm.DateTimeOffset	TODO
hardTimeoutRemaining	Type: Edm.Int32	TODO

id	Type: Edm.String	TODO
isActive	Type: Edm.Boolean	TODO
learnAndLockSessionEndReason	Type: Edm.String	TODO
logicalConnectionLearningConfig	Type: Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.Configuration.LogicalConnectionLearningConfig	TODO
logicalConnectionLearningEndReason	Type: Sel.Sel5056.LearnAndLock.Enums.EndReason	TODO
logicalConnectionLearningEndedByUser	Type: Edm.String	TODO
logicalConnectionLearningSession	Type: Sel.Sel5056.LearnAndLock.Sessions.LogicalConnectionLearningSession	TODO
logicalConnectionLearningWasRun	Type: Edm.Boolean	TODO
networkResetWasRun	Type: Edm.Boolean	TODO
sessionAcknowledged	Type: Edm.Boolean	TODO
sessionAcknowledgedTime	Type: Edm.DateTimeOffset	TODO
stateBeginEndTimestamps	Type: Sel.Sel5056.LearnAndLock.Status.LearnAndLockStateBeginEndTimestamps	TODO
stateStatus	Type: Sel.Sel5056.LearnAndLock.LearnAndLockStateStatus	TODO
statusCounters	Type: Sel.Sel5056.LearnAndLock.Status.LearnAndLockCounters	TODO
statusMessages	Type: [Edm.String]	TODO
usernameWhoAcknowledged	Type: Edm.String	TODO
usernameWhoStarted	Type: Edm.String	TODO

Table 22.40: Properties for Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

Connections:
Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession
—Sel.BeginLearnAndLock.bindingParameter
—Sel.SetDone.bindingParameter
—Sel.StopLearnAndLock.bindingParameter
—Sel.AcknowledgeLearnAndLockSession.bindingParameter
—Sel.AcceptLearnedLogicalConnections.bindingParameter
—Sel.ClearLearnedLogicalConnections.bindingParameter
—Sel.ProposeLearnedLogicalConnections.bindingParameter
—Sel.DeclineLearnedLogicalConnections.bindingParameter
—Sel.AcceptAllProposedLearnedLogicalConnections.bindingParameter
—Sel.DeclineAllProposedLearnedLogicalConnections.bindingParameter
—Sel.DeleteLearnAndLockSession.bindingParameter
—Sel.LearnAndLockIndividualSessionExportCSV.bindingParameter
—Sel.GetLearnAndLockCounters.bindingParameter
—Sel.GetHardTimeoutTimeRemaining.bindingParameter
—Sel.GetStatusMessages.bindingParameter
—Sel.GetStateBeginEndTimestamps.bindingParameter

22.17 Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType

name	attributes	description
autoFillSettings	Type: Sel.Sel5056.LogicalConnectionValidation.DataObject.AutoFillSettings	TODO
cstCommunicationType	DefaultValue: UnicastBidirectional Type: Sel.Sel5056.LogicalConnectionValidation.Enums.CstCommunicationType	Cast type and directionality of the LC. One of the following: 'UnicastBidirectional', 'Unicast', 'Multicast'
displayName	Indexed: True MaxLength: 512 Type: Edm.String Unique: True	Common Property
errorState	Deprecated: This is not implemented and is only set to success Type: Sel.BlueFrame.Core.ErrorState	Common Property
errors	Deprecated: Errors are never set Type: [Edm.String]	Common Property
id	Type: Edm.String	Common Property

numFailableLinks	DefaultValue: 1 Type: Edm.Int64	Link redundancy: must be 0 (no redundancy) or 1
numFailableNodes	Type: Edm.Int64	Node redundancy: must be 0
priority	DefaultValue: 2000 Type: Edm.Int32	Flow entry priority for the flow entries created for LC using this CST
setQueue	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetQueueAction	Priority Queue of the flow entries created for the LC. Must be 1 (lowest)-4 (highest)
tags	Type: [Sel.Sel5056.Tags.AmTag]	TODO
trafficMatch	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match	TODO
version	Type: Edm.Int32	Common Property

Table 22.42: Properties for Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType

The following types inherit from this type:
-Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceTypePreDefined
Connections:
Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType

22.17.1 Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceTypePreDefined

This type has no properties

22.18 Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection

name	attributes	description
communicationServiceTypeId	Indexed: True Type: Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType	id property of the CST
connectionOrigin	DefaultValue: EndUser Type: Sel.Sel5056.LogicalConnectionValidation.Enums.LogicalConnectionOrigin	Who created the LC
destinationEndPoints	Type: [Edm.String]	id property of the configuration node object(s) of the destination(s)
errorState	Type: Sel.BlueFrame.Core.ErrorState	Common Property

errors	Type: [Edm.String]	Common Property
id	Type: Edm.String	Common Property
sourceEndPoints	Type: [Edm.String]	id property of the configuration node object of the source
version	Type: Edm.Int32	Common Property

Table 22.44: Properties for Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection

Connections:
Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection
—Sel.GetLogicalConnectionStats.bindingParameter
—Sel.GetOpenFlowInformation.bindingParameter
—Sel.GetPathPlanInformation.bindingParameter
—Sel.ResubmitLogicalConnection.bindingParameter
—Sel.DeleteAll.bindingParameter
—Sel.ReplanAll.bindingParameter

22.19 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow

name	attributes	description
bufferId	Type: Edm.Int64	TODO
checkOverlap	DefaultValue: True Type: Edm.Boolean	Whether switch will reject the flow is there is an overlap with another flow entry
cookie	Type: Edm.Int64	OpenFlow cookie
displayName	Indexed: True MaxLength: 512 Type: Edm.String	Common Property
enabled	DefaultValue: True Type: Edm.Boolean	Whether the Flow is programmed to the switch
errorState	Type: Sel.BlueFrame.Core.ErrorState	Common Property
errors	Type: [Edm.String]	Common Property
hardTimeout	Type: Edm.Int32	Seconds before flow is deleted (o is disabled)
id	Type: Edm.String	Common Property
idleTimeout	Type: Edm.Int32	Seconds before unmatched flow is deleted (0 is disabled)
inPort	Type: Edm.Int64	Translated value of the InPort match field if present

instructions	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction]	Instruction set
lastEdit	Type: Edm.DateTimeOffset	Time flow entry was last modified
match	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match	Match criteria
noByteCounts	Type: Edm.Boolean	Whether to gather byte count diagnostics
noPacketCounts	Type: Edm.Boolean	Whether to gather packet count diagnostics
node	Indexed: True Type: Sel.Sel5056.TopologyManager.Nodes.ConfigNode	OpenFlow switch configuration node id
outGroup	Type: Edm.Int64	Translated value of the Group write-action if present
outPort	Type: Edm.Int64	Translated value of the Output write-action if present
priority	DefaultValue: 1000 Max: 65535 Min: 0 Step: 1 Type: Edm.Int32	Flow Priority
resetCounts	Type: Edm.Boolean	TODO
tableId	Max: 255 Min: 0 Step: 1 Type: Edm.Byte	Flow Table Number
tags	Type: [Sel.Sel5056.Tags.AmTag]	Common Property
version	Type: Edm.Int32	Common Property

Table 22.46: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow
—Sel.ResetCounters.bindingParameter

22.20 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats

name	attributes	description
dataPathId	Type: Edm.Int64	Datapath ID of the switch the flow belongs to

id	Type: Edm.String	Common Property
stats	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat]	Flow diagnostics object
transactionId	Type: Edm.Int64	TODO

Table 22.48: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats

22.21 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group

name	attributes	description
buckets	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket]	TODO
displayName	Indexed: True MaxLength: 512 Type: Edm.String	TODO
enabled	DefaultValue: True Type: Edm.Boolean	TODO
errorState	Type: Sel.BlueFrame.Core.ErrorState	TODO
errors	Type: [Edm.String]	TODO
groupId	Type: Edm.Int64	TODO
groupType	DefaultValue: FastFailover Type: Sel.Sel5056.OpenFlowPlugin.Enums.OfpGroupType	TODO
id	Type: Edm.String	TODO
node	Indexed: True Type: Sel.Sel5056.TopologyManager.Nodes.ConfigNode	TODO
tags	Type: [Sel.Sel5056.Tags.AmTag]	TODO
version	Type: Edm.Int32	TODO

Table 22.50: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc.groups
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc

22.22 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc

name	attributes	description
dataPathId	Type: Edm.Int64	TODO
groups	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group]	TODO
id	Type: Edm.String	Common Property
transactionId	Type: Edm.Int64	TODO

Table 22.52: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc

22.23 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats

name	attributes	description
dataPathId	Type: Edm.Int64	TODO
id	Type: Edm.String	Common Property
stats	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat]	TODO
transactionId	Type: Edm.Int64	TODO

Table 22.54: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats

22.24 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter

name	attributes	description
displayName	Indexed: True MaxLength: 512 Type: Edm.String	Common Property
enabled	DefaultValue: True Type: Edm.Boolean	TODO
errorState	Type: Sel.BlueFrame.Core.ErrorState	Common Property
errors	Type: [Edm.String]	Common Property
flags	Type: [Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.RateTypeBase]	TODO
id	Type: Edm.String	Common Property
meterBands	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand]	TODO
meterId	Type: Edm.Int64	OpenFlow entry ID

node	Indexed: True Type: Sel.Sel5056.TopologyManager.Nodes.ConfigNode	OpenFlow switch configuration node id
tags	Type: [Sel.Sel5056.Tags.AmTag]	Common Property
version	Type: Edm.Int32	Common Property

Table 22.56: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter

22.25 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions

name	attributes	description
dataPathId	Type: Edm.Int64	TODO
id	Type: Edm.String	Common Property
meterDescriptionsList	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDesc]	TODO
transactionId	Type: Edm.Int64	TODO

Table 22.58: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions

22.26 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures

name	attributes	description
bandTypes	Type: Sel.Sel5056.OpenFlowPlugin.Enums.OF13MeterBandType	TODO
capabilities	Type: [Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.RateTypeBase]	TODO
dataPathId	Type: Edm.Int64	TODO
id	Type: Edm.String	TODO
maxBands	Type: Edm.Byte	TODO
maxColor	Type: Edm.Byte	TODO
maxMeters	Type: Edm.Int64	TODO

Table 22.60: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures

22.27 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats

name	attributes	description
dataPathId	Type: Edm.Int64	TODO
id	Type: Edm.String	Common Property
meterStatList	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStat]	TODO
transactionId	Type: Edm.Int64	TODO

Table 22.62: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats

22.28 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc

name	attributes	description
dataPathId	Type: Edm.Int64	TODO
id	Type: Edm.String	TODO
ports	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port]	TODO

Table 22.64: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc

22.29 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats

name	attributes	description
dataPathId	Type: Edm.Int64	Datapath ID of the switch the flow belongs to
id	Type: Edm.String	Common Property
portStatList	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStat]	TODO

Table 22.66: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats

22.30 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ReceivedPacket

name	attributes	description
contents	Type: Edm.String	Base64 encoded contents of received packet
cookie	Type: Edm.Int64	TODO
dataPathId	Type: Edm.Int64	TODO
id	Type: Edm.String	Common Property
monotonicId	Type: Edm.Int64	Common Property
port	Type: Edm.Int64	Ingress port number of the packet
prettyDataPathId	Type: Edm.String	Ingress port number of the packet
receivedAt	Type: Edm.Int64	TODO

Table 22.68: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ReceivedPacket

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ReceivedPacket

22.31 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats

name	attributes	description
dataPathId	Type: Edm.Int64	TODO
id	Type: Edm.String	Common Property
tableStatList	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStat]	Common Property

Table 22.70: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats

22.32 Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation

name	attributes	description

creator	DefaultValue: System Type: Edm.String	Username who reserved the value, else 'Controller' if reserved by the SEL-5056
id	Type: Edm.String	Common Property
uniqueId	Indexed: True Type: Edm.String	Same as id property
vlanVid	Max: 4094 Min: 1 Step: 1 Type: Edm.Int32	Reserved VID

Table 22.72: Properties for Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation

The following types inherit from this type:
-Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidDestinationReservation
Connections:
Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation
—Sel.ReserveVlanVidRange.bindingParameter

22.32.1 Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidDestinationReservation

name	attributes	description
amDestination	Type: Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.AmDestination	TODO

Table 22.74: Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidDestinationReservation

22.33 Sel.Sel5056.SystemMessaging.SystemMessage

name	attributes	description
id	Type: Edm.String	Common Property
message	Type: Edm.String	Common Property
module	Type: Edm.String	Common Property
severity	Type: Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel	Common Property
type	Type: Edm.String	Common Property

Table 22.76: Properties for Sel.Sel5056.SystemMessaging.SystemMessage

Connections:
Sel.Sel5056.SystemMessaging.SystemMessage

22.34 Sel.Sel5056.TopologyManager.Links.ConfigLink

name	attributes	description
configurations	Type: [Sel.Sel5056.TopologyManager.Configurations.AmConfig]	TODO
displayName	AcceptServerChanges: true Indexed: True MaxLength: 512 MinLength: 0 Type: Edm.String	Common Property
endPoints	Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort	Configuration Port ID for each end of the link
first	Indexed: True Type: Edm.String	One of the two end points
id	Type: Edm.String	Common Property
last	Indexed: True Type: Edm.String	One of the two end points
linkedKey	Indexed: True Type: Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink	Associated Operational ID
state	DefaultValue: Configured Type: Sel.Sel5056.TopologyManager.Enums.State	Common Property
tags	Type: [Sel.Sel5056.Tags.AmTag]	Common Property
weight	DefaultValue: 1 Max: 2147483647 Min: 1 Step: 1 Type: Edm.Int64	TODO

Table 22.78: Properties for Sel.Sel5056.TopologyManager.Links.ConfigLink

Connections:
Sel.Sel5056.TopologyManager.Links.ConfigLink

22.35 Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

name	attributes	description
attributes	Type: [Sel.Sel5056.TopologyManager.Attributes.AmAttribute]	TODO
displayName	Type: Edm.String	Common Property
endPoints	Type: [Edm.String]	Operational Port ID for each end of the link

first	Indexed: True Type: Edm.String	One of the two end points
id	Type: Edm.String	Common Property
last	Indexed: True Type: Edm.String	One of the two end points
linkedKey	Indexed: True Type: Sel.Sel5056.TopologyManager.Links.-ConfigLink	Associated Configuration ID
state	Type: Sel.Sel5056.TopologyManager.Enums.-State	Common Property
trustState	Type: Sel.Sel5056.TopologyManager.Enums.-DiscoveryTrustState	TODO

Table 22.80: Properties for Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

Connections:
Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink
—Sel.Adopt.bindingParameter
—Sel.AdoptWithConfig.bindingParameter
—Sel.Unadopt.bindingParameter
—Sel.ReplaceConfig.bindingParameter

22.36 Sel.Sel5056.TopologyManager.NetworkSettings

name	attributes	description
controllerArpSrcIpAddress	DataClass: IpAddress DefaultValue: 0.0.0.0 Type: Edm.String	TODO
controllerArpSrcNetworkAddresses	Type: [Sel.BlueFrame.Core.Networking.-NetworkInterfaceAddress]	TODO
controllerIp	DataClass: IpAddress Type: Edm.String	TODO
defaultGateway	DataClass: IpAddress Type: Edm.String	TODO
id	Type: Edm.String	Common Property
ntpServers	Type: [Edm.String]	TODO

Table 22.82: Properties for Sel.Sel5056.TopologyManager.NetworkSettings

Connections:
Sel.Sel5056.TopologyManager.NetworkSettings

22.37 Sel.Sel5056.TopologyManager.Nodes.ConfigNode

name	attributes	description
configurations	Type: [Sel.Sel5056.TopologyManager.Configurations.AmConfig]	TODO
displayName	AcceptServerChanges: true Indexed: True MaxLength: 512 MinLength: 0 Type: Edm.String	Common Property
id	Type: Edm.String	Common Property
linkedKey	Indexed: True Type: Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode	Associated Operational ID
ports	Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort	Configuration Port ID of attached ports
state	DefaultValue: Configured Type: Sel.Sel5056.TopologyManager.Enums.State	Common Property
tags	Type: [Sel.Sel5056.Tags.AmTag]	Common Property

Table 22.84: Properties for Sel.Sel5056.TopologyManager.Nodes.ConfigNode

The following types inherit from this type:
-Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode
Connections:
Sel.Sel5056.TopologyManager.Nodes.ConfigNode

22.37.1 Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

name	attributes	description
additionalTrustedCertificates	Type: [Edm.String]	TODO
alarmMinimumDuration	DefaultValue: 1 Max: 30 Min: 1 Nullable: False Step: 1 Type: Edm.Int64	TODO
controllerIp	DataClass: IpAddress Nullable: False Type: Edm.String	TODO
defaultGateway	DataClass: IpAddress Nullable: False Type: Edm.String	TODO

enablePtp	DefaultValue: False Nullable: False Type: Edm.Boolean	TODO
enableSnmp	DefaultValue: False Nullable: False Type: Edm.Boolean	TODO
eventCategories	Type: [Sel.BlueFrame.Core.EventBus.EventCategory]	TODO
ipAddress	Nullable: False Type: Edm.String Unique: True	TODO
ntpServers	Type: [Edm.String]	TODO
subnetMask	DataClass: IpAddress Nullable: False Type: Edm.String	TODO

Table 22.86: Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

22.38 Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

name	attributes	description
attributes	Type: [Sel.Sel5056.TopologyManager.Attributes.AmAttribute]	TODO
displayName	Type: Edm.String	Common Property
id	Type: Edm.String	Common Property
linkedKey	Indexed: True Type: Sel.Sel5056.TopologyManager.Nodes.ConfigNode	Associated Configuration ID
ports	Type: [Edm.String]	Operational Port ID of attached ports
state	Type: Sel.Sel5056.TopologyManager.Enums.State	Common Property
trustState	Type: Sel.Sel5056.TopologyManager.Enums.DiscoveryTrustState	TODO

Table 22.88: Properties for Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

Connections:

Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
—Sel.Disconnect.bindingParameter
—Sel.DisableControllerArpToNode.bindingParameter
—Sel.EnableControllerArpToNode.bindingParameter
—Sel.PacketOut.bindingParameter
—Sel.FactoryDefaultReset.bindingParameter
—Sel.FirmwareUpgrade.bindingParameter

—Sel.Reboot.bindingParameter
—Sel.ReplanInbandPath.bindingParameter

22.39 Sel.Sel5056.TopologyManager.Ports.ConfigPort

name	attributes	description
attachedLinks	Type: [Edm.String]	Link IDs of attached links
configurations	Type: [Sel.Sel5056.TopologyManager.Configurations.AmConfig]	TODO
displayName	AcceptServerChanges: true Indexed: True MaxLength: 512 MinLength: 0 Type: Edm.String	Common Property
displayNameSetByUser	Type: Edm.Boolean	TODO
id	Type: Edm.String	Common Property
linkedKey	Indexed: True Type: Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort	Associated Operational ID
parentNode	Indexed: True Type: Sel.Sel5056.TopologyManager.Nodes.ConfigNode	Configuration Node ID to which the port belongs
state	DefaultValue: Configured Type: Sel.Sel5056.TopologyManager.Enums.State	Common Property
tags	Type: [Sel.Sel5056.Tags.AmTag]	Common Property

Table 22.90: Properties for Sel.Sel5056.TopologyManager.Ports.ConfigPort

Connections:
Sel.Sel5056.TopologyManager.Ports.ConfigPort
—Sel.SendPortMod.bindingParameter

22.40 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

name	attributes	description
attachedLinks	Type: [Edm.String]	Link IDs of attached links
attributes	Type: [Sel.Sel5056.TopologyManager.Attributes.AmAttribute]	TODO
displayName	Type: Edm.String	Common Property

id	Type: Edm.String	Common Property
isConnected	Type: Edm.Boolean	Common Property
linkedKey	Indexed: True Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort	Associated Configuration ID
parentNode	Indexed: True Type: Edm.String	Operational Node ID to which the port belongs
state	Type: Sel.Sel5056.TopologyManager.Enums.State	Common Property
trustState	Type: Sel.Sel5056.TopologyManager.Enums.DiscoveryTrustState	TODO

Table 22.92: Properties for Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

Connections:
Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
—Sel.MarkPortAsSelRelayFailover.bindingParameter
—Sel.RemoveSelRelayFailoverFromPort.bindingParameter
—Sel.DetectSelRelayFailoverPort.bindingParameter
—Sel.MarkPortAndPerformBlockingDetectSelRelayFailover.bindingParameter
—Sel.AddSilentHost.bindingParameter
—Sel.AddOperationalLinkFromHostToSwitch.bindingParameter
—Sel.MarkPortAndAddRelayFailoverLink.bindingParameter
—Sel.SendPortMod.bindingParameter
—Sel.AddAbstractNode.bindingParameter

22.41 Sel.Sel5056.TopologyManager.Synchronization.SynchronizationRequest

name	attributes	description
configNodeId	Indexed: True Type: Edm.String	Configuration Node ID
description	Type: Edm.String	TODO
errorState	Type: Sel.BlueFrame.Core.ErrorState	TODO
errors	Type: [Edm.String]	TODO
id	Type: Edm.String	Common Property
version	Type: Edm.Int32	Common Property

Table 22.94: Properties for Sel.Sel5056.TopologyManager.Synchronization.SynchronizationRequest

The following types inherit from this type:
-Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowAddSynchronizationRequest
-Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowDeleteSynchronizationRequest
-Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModifySynchronizationRequest

-Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupAddSynchronizationRequest
-Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupDeleteSynchronizationRequest
-Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupModifySynchronizationRequest
-Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterAddSynchronizationRequest
-Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterDeleteSynchronizationRequest
-Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterModifySynchronizationRequest
-Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization
-Sel.Sel5056.SapphirePlugin.Synchronization.FirmwareUpgradeRequiredSynchronization
Connections:
Sel.Sel5056.TopologyManager.Synchronization.SynchronizationRequest

22.41.1 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowAddSynchronizationRequest

name	attributes	description
alias	Type: Edm.String	TODO
configFlowId	Type: Edm.String	TODO
cookie	Nullable: False Type: Edm.Int64	Flow Entry ID
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]	TODO
priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.AmSystemPriority	TODO

Table 22.96: Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowAddSynchronizationRequest

22.41.2 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowDeleteSynchronizationRequest

name	attributes	description
alias	Type: Edm.String	TODO
cookie	Nullable: False Type: Edm.Int64	Flow Entry ID
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]	TODO

priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.AmSystemPriority	TODO
referencedGroupId	Nullable: False Type: Edm.Int64	TODO

Table 22.98: Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowDeleteSynchronizationRequest

22.41.3 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModifySynchronizationRequest

name	attributes	description
alias	Type: Edm.String	TODO
configFlowId	Type: Edm.String	TODO
cookie	Nullable: False Type: Edm.Int64	Flow Entry ID
currentGroupId	Nullable: False Type: Edm.Int64	TODO
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]	TODO
newGroupId	Nullable: False Type: Edm.Int64	TODO
priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.AmSystemPriority	TODO

Table 22.100: Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModifySynchronizationRequest

22.41.4 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupAddSynchronizationRequest

name	attributes	description
alias	Type: Edm.String	TODO
configGroupId	Type: Edm.String	TODO
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]	TODO
groupId	Nullable: False Type: Edm.Int64	TODO
priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.AmSystemPriority	TODO

Table 22.102: Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupAddSynchronizationRequest

22.41.5 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupDeleteSynchronizationRequest

name	attributes	description
alias	Type: Edm.String	TODO
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]	TODO
groupId	Nullable: False Type: Edm.Int64	TODO
priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.AmSystemPriority	TODO

Table 22.104: Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupDeleteSynchronizationRequest

22.41.6 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupModifySynchronizationRequest

name	attributes	description
alias	Type: Edm.String	TODO
configGroupId	Type: Edm.String	TODO
currentGroupIds	Nullable: False Type: [Edm.Int64]	TODO
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]	TODO
groupId	Nullable: False Type: Edm.Int64	TODO
newGroupIds	Nullable: False Type: [Edm.Int64]	TODO
priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.AmSystemPriority	TODO

Table 22.106: Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupModifySynchronizationRequest

22.41.7 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterAddSynchronizationRequest

name	attributes	description
alias	Type: Edm.String	TODO
configMeterId	Type: Edm.String	TODO
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]	TODO
meterId	Nullable: False Type: Edm.Int64	TODO
priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.AmSystemPriority	TODO

Table 22.108: Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterAddSynchronizationRequest

22.41.8 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterDeleteSynchronizationRequest

name	attributes	description
alias	Type: Edm.String	TODO
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]	TODO
meterId	Nullable: False Type: Edm.Int64	TODO
priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.AmSystemPriority	TODO

Table 22.110: Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterDeleteSynchronizationRequest

22.41.9 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterModifySynchronizationRequest

name	attributes	description
alias	Type: Edm.String	TODO
configMeterId	Type: Edm.String	TODO
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]	TODO
meterId	Nullable: False Type: Edm.Int64	TODO
priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.AmSystemPriority	TODO

Table 22.112: Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterModifySynchronizationRequest

22.41.10 Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization

name	attributes	description
endpoint	Nullable: False Type: Sel.Sel5056.SapphirePlugin.Synchronization.SapphireEndpoint	TODO
key	Type: Edm.String	TODO

Table 22.114: Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization

22.41.11 Sel.Sel5056.SapphirePlugin.Synchronization.FirmwareUpgradeRequiredSynchronization

This type has no properties

Chapter 23

Secondary OData Types

This section describes the OData types that are derived from base types described in TODO.

23.1 Sel.BlueFrame.ApplicationRegistration.AuthenticationType

name	attributes	description
clientId	Type: Edm.String	OAuth Client ID
grantType	Type: Edm.String	OAuth Grant Type
permissions	MinLength: 1 Type: [Sel.BlueFrame.Core.SecurityManager.Permission]	SEL-5056 Permissions
redirectUris	Type: [Edm.String]	OAuth Redirect URIs

Table 23.1: Properties for Sel.BlueFrame.ApplicationRegistration.AuthenticationType

Connections:
Sel.BlueFrame.ApplicationRegistration.AuthenticationType
—Sel.BlueFrame.ApplicationRegistration.ApplicationLink.authenticationTypes
——Sel.BlueFrame.ApplicationRegistration.ApplicationLink

23.2 Sel.BlueFrame.CommissioningManager.CommissioningConcern

This type has no properties
The following types inherit from this type:
-Sel.BlueFrame.CommissioningManager.UserCommissioner.UserCommissioningConcern
Connections:
Sel.BlueFrame.CommissioningManager.CommissioningConcern

23.2.1 Sel.BlueFrame.CommissioningManager.UserCommissioner.UserCommissioningConcern

name	attributes	description
password	CharacterSet: Printable7BitAscii MaxLength: 128 MinLength: 8 Type: Edm.String	Password of the added SecurityAdministrator user
username	CharacterSet: Printable7BitAscii MaxLength: 128 MinLength: 1 Nullable: False Type: Edm.String	Username of the added SecurityAdministrator user

Table 23.2: Properties for type Sel.BlueFrame.CommissioningManager.UserCommissioner.UserCommissioningConcern

23.3 Sel.BlueFrame.Core.EventBus.Behaviors.Behavior

name	attributes	description
name	Type: Edm.String	Name for the logging setting
severity	DefaultValue: Warning Type: Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel	Logging level

Table 23.3: Properties for Sel.BlueFrame.Core.EventBus.Behaviors.Behavior

The following types inherit from this type:
-Sel.BlueFrame.EventBus.Syslog.SyslogBehavior
-Sel.BlueFrame.EventBus.WindowsEventLog.WindowsEventLogBehavior
-Sel.BlueFrame.Particle.AlarmContact.AlarmContactBehavior
-Sel.BlueFrame.Particle.LocalEventStore.LocalEventStoreBehavior
Connections:
Sel.BlueFrame.Core.EventBus.Behaviors.Behavior
—Sel.BlueFrame.Core.EventBus.EventCategory.behaviors
——Sel.BlueFrame.Core.EventBus.EventCategory
—Sel.BlueFrame.Core.EventBus.EventCategory.behaviors
——Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode.eventCategories
———Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

23.3.1 Sel.BlueFrame.EventBus.Syslog.SyslogBehavior

name	attributes	description

ipAddress	DataClass: IpAddress Nullable: False Type: Edm.String	IP Address of the the Syslog server
port	DefaultValue: 514 Max: 65535 Min: 1 Nullable: False Step: 1 Type: Edm.Int32	TCP or UDP port of the service on the Syslog server
rfcType	DefaultValue: Rfc3164 Nullable: False Type: Sel.BlueFrame.EventBus.Syslog.RFCType	Format of the message
transportType	DefaultValue: Udp Nullable: False Type: Sel.BlueFrame.EventBus.Syslog.TransportType	UDP or TLS

Table 23.4: Properties for type Sel.BlueFrame.EventBus.Syslog.SyslogBehavior

23.3.2 Sel.BlueFrame.EventBus.WindowsEventLog.WindowsEventLogBehavior

name	attributes	description
alertMap	Nullable: FalseType: System.Diagnostics.EventLogEntryType	TODO
criticalMap	Nullable: FalseType: System.Diagnostics.EventLogEntryType	TODO
emergencyMap	Nullable: FalseType: System.Diagnostics.EventLogEntryType	TODO
errorMap	Nullable: FalseType: System.Diagnostics.EventLogEntryType	TODO
informationalMap	Nullable: FalseType: System.Diagnostics.EventLogEntryType	TODO
noticeMap	Nullable: FalseType: System.Diagnostics.EventLogEntryType	TODO
warningMap	Nullable: FalseType: System.Diagnostics.EventLogEntryType	TODO

Table 23.5: Properties for type Sel.BlueFrame.EventBus.WindowsEventLog.WindowsEventLogBehavior

23.3.3 Sel.BlueFrame.Particle.AlarmContact.AlarmContactBehavior

This type has no properties

23.3.4 Sel.BlueFrame.Particle.LocalEventStore.LocalEventStoreBehavior

This type has no properties

23.4 Sel.BlueFrame.Core.EventBus.LocaleRecord

name	attributes	description
locale	Type: Edm.String	TODO
value	Type: Edm.String	TODO

Table 23.6: Properties for Sel.BlueFrame.Core.EventBus.LocaleRecord

Connections:
Sel.BlueFrame.Core.EventBus.LocaleRecord
—Sel.BlueFrame.Core.EventBus.LocaleString.records
——Sel.BlueFrame.Core.EventBus.LocaleString

23.5 Sel.BlueFrame.Core.FileArgument

name	attributes	description
fileName	Type: Edm.String	TODO

Table 23.7: Properties for Sel.BlueFrame.Core.FileArgument

Connections:
Sel.BlueFrame.Core.FileArgument
—Sel.Restore.fileName

23.6 Sel.BlueFrame.Core.FileReturnValue

name	attributes	description
contentType	Type: Edm.String	TODO
fileName	Type: Edm.String	TODO

Table 23.8: Properties for Sel.BlueFrame.Core.FileReturnValue

Connections:
Sel.BlueFrame.Core.FileReturnValue

23.7 Sel.BlueFrame.Core.Networking.NetworkInterfaceAddress

name	attributes	description
fullAddress	Type: Edm.String	TODO

Table 23.9: Properties for Sel.BlueFrame.Core.Networking.NetworkInterfaceAddress

Connections:

Sel.BlueFrame.Core.Networking.NetworkInterfaceAddress
—Sel.Sel5056.TopologyManager.NetworkSettings.controllerArpSrcNetworkAddresses
——Sel.Sel5056.TopologyManager.NetworkSettings

23.8 Sel.BlueFrame.Core.SecurityManager.Permission

name	attributes	description
module	Type: Edm.String	TODO
permissionTag	Type: Edm.String	TODO

Table 23.10: Properties for Sel.BlueFrame.Core.SecurityManager.Permission

The following types inherit from this type:
-Sel.BlueFrame.Core.DataBrokerPermission
-Sel.BlueFrame.Core.SecurityAttributePermission
-Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission
-Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission
Connections:
Sel.BlueFrame.Core.SecurityManager.Permission
—Sel.BlueFrame.Core.SecurityManager.Role.permissions
——Sel.BlueFrame.Core.SecurityManager.Role
—Sel.BlueFrame.Core.SecurityManager.ModuleRole.permissions
——Sel.BlueFrame.Core.SecurityManager.ModuleRole
—Sel.BlueFrame.ApplicationRegistration.AuthenticationType.permissions
——Sel.BlueFrame.ApplicationRegistration.ApplicationLink.authenticationTypes
———Sel.BlueFrame.ApplicationRegistration.ApplicationLink

23.8.1 Sel.BlueFrame.Core.DataBrokerPermission

name	attributes	description
dataType	Type: Edm.String	TODO
tree	Type: Edm.String	TODO

Table 23.11: Properties for type Sel.BlueFrame.Core.DataBrokerPermission

23.8.2 Sel.BlueFrame.Core.SecurityAttributePermission

name	attributes	description
securityAttribute	Type: Edm.String	TODO

Table 23.12: Properties for type Sel.BlueFrame.Core.SecurityAttributePermission

23.8.3 Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission

name	attributes	description
purpose	Type: Edm.String	TODO

Table 23.13: Properties for type Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission

23.8.4 Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission

name	attributes	description
deviceType	Type: Edm.String	TODO

Table 23.14: Properties for type Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission

23.9 Sel.BlueFrame.Core.SecurityManager.UserFailedLogin

name	attributes	description
ipAddress	Type: Edm.String	TODO
loginTime	Type: Edm.DateTimeOffset	TODO

Table 23.15: Properties for Sel.BlueFrame.Core.SecurityManager.UserFailedLogin

Connections:
Sel.BlueFrame.Core.SecurityManager.UserFailedLogin
—Sel.BlueFrame.Core.SecurityManager.User.failedLogins
——Sel.BlueFrame.Core.SecurityManager.User

23.10 Sel.BlueFrame.Core.TrustAuthority.CertificatePurpose

This type has no properties
The following types inherit from this type:
-Sel.BlueFrame.TrustAuthority.TrustedCertificates.TrustedCertificatePurpose
-Sel.Sel5056.Drivers.OpenFlow.Trust.OpenFlowDeviceCertificate
-Sel.Sel5056.Drivers.OpenFlow.Trust.OpenFlowServerCertificate
-Sel.BlueFrame.TrustAuthority.WebCertificateManager.WebCertificatePurpose
-Sel.BlueFrame.TrustAuthority.InternalCertificateAuthority.InternalCaPurpose
-Sel.Sel5056.SapphirePlugin.Trust.SapphireDeviceCertificate
-Sel.Sel5056.SapphirePlugin.Trust.SapphireServerCertificate
Connections:
Sel.BlueFrame.Core.TrustAuthority.CertificatePurpose
—Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation.purpose
——Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation
—Sel.RevokeCertificateByPurpose.purpose

—Sel.UploadCertificate.purpose

23.10.1 Sel.BlueFrame.TrustAuthority.TrustedCertificates.TrustedCertificatePurpose

This type has no properties

23.10.2 Sel.Sel5056.Drivers.OpenFlow.Trust.OpenFlowDeviceCertificate

name	attributes	description
dataPathId	Nullable: FalseType: Edm.Int64	TODO

Table 23.16: Properties for type Sel.Sel5056.Drivers.OpenFlow.Trust.OpenFlowDeviceCertificate

23.10.3 Sel.Sel5056.Drivers.OpenFlow.Trust.OpenFlowServerCertificate

This type has no properties

23.10.4 Sel.BlueFrame.TrustAuthority.WebCertificateManager.WebCertificatePurpose

This type has no properties

23.10.5 Sel.BlueFrame.TrustAuthority.InternalCertificateAuthority.-InternalCaPurpose

This type has no properties

23.10.6 Sel.Sel5056.SapphirePlugin.Trust.SapphireDeviceCertificate

name	attributes	description
ipAddress	DataClass: IpAddressType: Edm.String	TODO
serialNumber	Type: Edm.String	TODO

Table 23.17: Properties for type Sel.Sel5056.SapphirePlugin.Trust.SapphireDeviceCertificate

23.10.7 Sel.Sel5056.SapphirePlugin.Trust.SapphireServerCertificate

This type has no properties

23.11 Sel.BlueFrame.SecurityManager.AuthServiceGroupDescription

name	attributes	description
displayName	Type: Edm.String	TODO
name	Type: Edm.String	TODO

Table 23.18: Properties for Sel.BlueFrame.SecurityManager.AuthServiceGroupDescription

Connections:
Sel.BlueFrame.SecurityManager.AuthServiceGroupDescription
—Sel.BlueFrame.SecurityManager.GroupCollection.groups

23.12 Sel.BlueFrame.SecurityManager.AuthServiceResult

name	attributes	description
attemptedTest	Type: Edm.String	TODO
message	Type: Edm.String	TODO
state	Type: Sel.BlueFrame.SecurityManager.AuthServiceResultState	TODO

Table 23.19: Properties for Sel.BlueFrame.SecurityManager.AuthServiceResult

Connections:
Sel.BlueFrame.SecurityManager.AuthServiceResult
—Sel.BlueFrame.SecurityManager.AuthServiceResultCollection.results

23.13 Sel.BlueFrame.SecurityManager.AuthServiceResultCollection

name	attributes	description
results	Type: [Sel.BlueFrame.SecurityManager.AuthServiceResult]	TODO

Table 23.20: Properties for Sel.BlueFrame.SecurityManager.AuthServiceResultCollection

Connections:
Sel.BlueFrame.SecurityManager.AuthServiceResultCollection

23.14 Sel.BlueFrame.SecurityManager.GroupCollection

name	attributes	description
groups	Type: [Sel.BlueFrame.SecurityManager.AuthServiceGroupDescription]	TODO

Table 23.21: Properties for Sel.BlueFrame.SecurityManager.GroupCollection

Connections:
Sel.BlueFrame.SecurityManager.GroupCollection

23.15 Sel.BlueFrame.UserPreferences.Setting

name	attributes	description
key	Type: Edm.String	TODO
value	Type: Edm.String	TODO

Table 23.22: Properties for Sel.BlueFrame.UserPreferences.Setting

Connections:
Sel.BlueFrame.UserPreferences.Setting
—Sel.BlueFrame.UserPreferences.Preference.registry
——Sel.BlueFrame.UserPreferences.Preference

23.16 Sel.FileArgument

This type has no properties
Connections:
Sel.FileArgument
—Sel.FirmwareUpgrade.fileName

23.17 Sel.PathPlanHop

This type has no properties
Connections:
Sel.PathPlanHop
—Sel.Sel5056.PathPlanner.DataObject.PathPlanHopCollection.pathPlanHops

23.18 Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions

name	attributes	description
displayName	MaxLength: 128 Type: Edm.String	TODO
match	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match	TODO
meterMetadata	Type: Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.MeterMetadata	TODO

nodesSetting	Deprecated: This property has no behavior associated with it, do not use it Type: Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.NodesConfiguration	TODO
outputSetting	Type: Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.OutputConfiguration	TODO
priority	Type: Edm.Int32	TODO
tableId	Type: Edm.Byte	TODO
tableSetting	Type: Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.TableConfiguration	TODO

Table 23.23: Properties for Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions

Connections:
Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions

23.19 Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.MeterMetadata

name	attributes	description
flags	Type: [Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.RateTypeBase]	TODO
meterBands	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand]	TODO
meterId	Type: Edm.Int64	TODO

Table 23.24: Properties for Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.MeterMetadata

Connections:
Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.MeterMetadata
—Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions.meterMetadata

23.20 Sel.Sel5056.LearnAndLock.CSV.AdoptedHostInfo

name	attributes	description
connectedSwitchPort1	Type: Edm.String	TODO

connectedSwitchPort2	Type: Edm.String	TODO
displayName	Type: Edm.String	TODO
id	Type: Edm.String	TODO
ipAddress	Type: Edm.String	TODO
redundancyDetected	Type: Edm.Boolean	TODO

Table 23.25: Properties for Sel.Sel5056.LearnAndLock.CSV.AdoptedHostInfo

Connections:
Sel.Sel5056.LearnAndLock.CSV.AdoptedHostInfo
—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.adoptedHostsInfo
——Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

23.21 Sel.Sel5056.LearnAndLock.CSV.AdoptedLinkInfo

name	attributes	description
endpoint_1_Node	Type: Edm.String	TODO
endpoint_1_Port	Type: Edm.String	TODO
endpoint_2_Node	Type: Edm.String	TODO
endpoint_2_Port	Type: Edm.String	TODO

Table 23.26: Properties for Sel.Sel5056.LearnAndLock.CSV.AdoptedLinkInfo

Connections:
Sel.Sel5056.LearnAndLock.CSV.AdoptedLinkInfo
—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.adoptedLinksInfo
——Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

23.22 Sel.Sel5056.LearnAndLock.CSV.AdoptedSwitchInfo

name	attributes	description
dataPathId	Type: Edm.Int64	TODO
displayName	Type: Edm.String	TODO
ipAddress	Type: Edm.String	TODO
type	Type: Edm.String	TODO

Table 23.27: Properties for Sel.Sel5056.LearnAndLock.CSV.AdoptedSwitchInfo

Connections:
Sel.Sel5056.LearnAndLock.CSV.AdoptedSwitchInfo
—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.adoptedSwitchesInfo
——Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

23.23 Sel.Sel5056.LearnAndLock.Configuration.AutoAdoptConfig

name	attributes	description
configNodes	Type: [Edm.String]	TODO
endingIpAddress	DefaultValue: 192.168.1.254 Type: Edm.String	TODO
maxNumberOfHostsToAdopt	DefaultValue: 0 Type: Edm.Int64	TODO
maxNumberOfOpenFlowSwitchesToAdopt	DefaultValue: 0 Type: Edm.Int64	TODO
startingIpAddress	DefaultValue: 192.168.1.2 Type: Edm.String	TODO
unlimitedNumberOfHostsToAdopt	DefaultValue: True Type: Edm.Boolean	TODO
unlimitedNumberOfOpenFlowSwitchesToAdopt	DefaultValue: True Type: Edm.Boolean	TODO

Table 23.28: Properties for Sel.Sel5056.LearnAndLock.Configuration.AutoAdoptConfig

Connections:
Sel.Sel5056.LearnAndLock.Configuration.AutoAdoptConfig
—Sel.BeginLearnAndLock.autoAdoptConfig
—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.autoAdoptConfig
——Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

23.24 Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.Configuration.LogicalConnectionLearningConfig

name	attributes	description
additionalLearningRegionIps	Type: [Edm.String]	TODO
autoAcceptStatus	DefaultValue: All Type: Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.AutoAcceptStatus	TODO

enable-Multicast-Learning	DefaultValue: False Type: Edm.Boolean	TODO
hardTimeout-Seconds	DefaultValue: 0 Max: 604800 Min: 0 Step: 1 Type: Edm.Int32	TODO

Table 23.29: Properties for Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.Configuration.-LogicalConnectionLearningConfig

Connections:
Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.Configuration.LogicalConnectionLearningConfig
—Sel.BeginLearnAndLock.logicalConnectionLearningConfig
—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.logicalConnectionLearningConfig
——Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

23.25 Sel.Sel5056.LearnAndLock.ProposedCommunicationServiceType

name	attributes	description
cst-Communication-Type	DefaultValue: UnicastBidirectional Type: Sel.Sel5056.LogicalConnectionValidation.-Enums.CstCommunicationType	TODO
displayName	MaxLength: 128 Type: Edm.String Unique: True	TODO
matchableCst	Type: Edm.Boolean	TODO
numFailable-Links	DefaultValue: 1 Type: Edm.Int64	TODO
numFailable-Nodes	Type: Edm.Int64	TODO
priority	DefaultValue: 2000 Type: Edm.Int32	TODO
setQueue	Type: Sel.Sel5056.OpenFlowPlugin.-DataTreeObjects.SetQueueAction	TODO
trafficMatch	Type: Sel.Sel5056.OpenFlowPlugin.-DataTreeObjects.Match	TODO

Table 23.30: Properties for Sel.Sel5056.LearnAndLock.ProposedCommunicationServiceType

Connections:
Sel.Sel5056.LearnAndLock.ProposedCommunicationServiceType
—Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection.proposed-

CommunicationServiceType
——Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection

23.26 Sel.Sel5056.LearnAndLock.Sessions.AutoAdoptSession

name	attributes	description
status	Type: Sel.Sel5056.LearnAndLock.LearnAndLockStateStatus	TODO

Table 23.31: Properties for Sel.Sel5056.LearnAndLock.Sessions.AutoAdoptSession

Connections:
Sel.Sel5056.LearnAndLock.Sessions.AutoAdoptSession
—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.autoAdoptSession
——Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

23.27 Sel.Sel5056.LearnAndLock.Sessions.LogicalConnectionLearningSession

name	attributes	description
status	Type: Sel.Sel5056.LearnAndLock.LearnAndLockStateStatus	TODO

Table 23.32: Properties for Sel.Sel5056.LearnAndLock.Sessions.LogicalConnectionLearningSession

Connections:
Sel.Sel5056.LearnAndLock.Sessions.LogicalConnectionLearningSession
—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.logicalConnectionLearningSession
——Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

23.28 Sel.Sel5056.LearnAndLock.Status.LearnAndLockCounters

name	attributes	description
hostsAdopted	DefaultValue: 0Type: Edm.Int64	TODO
logicalConnectionsAccepted	DefaultValue: 0Type: Edm.Int64	TODO
logicalConnectionsDeclined	DefaultValue: 0Type: Edm.Int64	TODO
logicalConnectionsProgrammed	DefaultValue: 0Type: Edm.Int64	TODO
logicalConnectionsProposed	DefaultValue: 0Type: Edm.Int64	TODO
switchesAdopted	DefaultValue: 0Type: Edm.Int64	TODO

Table 23.33: Properties for Sel.Sel5056.LearnAndLock.Status.LearnAndLockCounters

Connections:
Sel.Sel5056.LearnAndLock.Status.LearnAndLockCounters
—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.statusCounters
——Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

23.29 Sel.Sel5056.LearnAndLock.Status.LearnAndLockStateBeginEndTimestamps

name	attributes	description
beginAutoAdoption	Type: Edm.DateTimeOffset	TODO
beginInbandManagementReplanning	Type: Edm.DateTimeOffset	TODO
beginLogicalConnectionLearning	Type: Edm.DateTimeOffset	TODO
beginNetworkReset	Type: Edm.DateTimeOffset	TODO
beginRelayFailoverDetection	Type: Edm.DateTimeOffset	TODO
endAutoAdoption	Type: Edm.DateTimeOffset	TODO
endInbandManagementReplanning	Type: Edm.DateTimeOffset	TODO
endLogicalConnectionLearning	Type: Edm.DateTimeOffset	TODO
endNetworkReset	Type: Edm.DateTimeOffset	TODO
endRelayFailoverDetection	Type: Edm.DateTimeOffset	TODO

Table 23.34: Properties for Sel.Sel5056.LearnAndLock.Status.LearnAndLockStateBeginEndTimestamps

Connections:
Sel.Sel5056.LearnAndLock.Status.LearnAndLockStateBeginEndTimestamps
—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.stateBeginEndTimestamps
——Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

23.30 Sel.Sel5056.LogicalConnectionValidation.DataObject.AutoFillSettings

name	attributes	description
autoFillMacWhenAutoFillingIp	DefaultValue: FalseType: Edm.Boolean	TODO
disableSourceAutoFill	DefaultValue: FalseType: Edm.Boolean	TODO

Table 23.35: Properties for Sel.Sel5056.LogicalConnectionValidation.DataObject.AutoFillSettings

Connections:
Sel.Sel5056.LogicalConnectionValidation.DataObject.AutoFillSettings
—Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType.autoFillSettings
——Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType

23.31 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Action

This type has no properties
The following types inherit from this type:
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetQueueAction
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupAction
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputAction
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PopVlanAction
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PushVlanAction
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetFieldAction
Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Action
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction.actions
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat.instructions
———Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow.instructions
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow
——Sel.Sel5056.PathPlanner.RestController.FlowReferences.instructions
———Sel.Sel5056.PathPlanner.RestController.FlowWithStats.flow
————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
———Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.flowReferences
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat.instructions
———Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket.actions
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.buckets
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group
——Sel.Sel5056.PathPlanner.RestController.GroupReferences.buckets
———Sel.Sel5056.PathPlanner.RestController.GroupWithStats.group
————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.groupsReferences
———Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.groupsReferences
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.buckets
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc.groups
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction.actions
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat.instructions
———Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow.instructions
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow

——Sel.Sel5056.PathPlanner.RestController.FlowReferences.instructions

———Sel.Sel5056.PathPlanner.RestController.FlowWithStats.flow

————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences

———Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.flowReferences

——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat.instructions

———Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics

————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences

———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats

————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats

—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket.actions

——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.buckets

———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group

——Sel.Sel5056.PathPlanner.RestController.GroupReferences.buckets

———Sel.Sel5056.PathPlanner.RestController.GroupWithStats.group

————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.groupsReferences

———Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.groupsReferences

——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.buckets

———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc.groups

————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc

23.31.1 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetQueueAction

name	attributes	description
queueId	DefaultValue: 2Nullable: FalseType: Edm.Int64	TODO

Table 23.36: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetQueueAction

23.31.2 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupAction

name	attributes	description
groupId	Nullable: FalseType: Edm.Int64	TODO

Table 23.37: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupAction

23.31.3 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputAction

name	attributes	description
maxLength	DefaultValue: 65535 Max: 65535 Min: 0 Nullable: False Step: 1 Type: Edm.Int32	TODO
outPort	Type: Edm.String	TODO

Table 23.38: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputAction

23.31.4 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PopVlanAction

This type has no properties

23.31.5 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PushVlanAction

name	attributes	description
etherType	DefaultValue: 0x8100Type: Edm.String	TODO

Table 23.39: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PushVlanAction

23.31.6 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetFieldAction

name	attributes	description
field	Nullable: False Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.MatchField	TODO

Table 23.40: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetFieldAction

23.32 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BandStat

name	attributes	description
byteBandCount	Type: Edm.Int64	TODO
packetBandCount	Type: Edm.Int64	TODO

Table 23.41: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BandStat

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BandStat
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStat.bandStats
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats.meterStatList
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats

23.33 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket

name	attributes		description
actions	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Action]		TODO
id	Type: Edm.String		TODO
outGroup	Type: Edm.Int64		TODO
outPort	Type: Edm.Int64		TODO
watchGroup	DefaultValue: Any	Type: Edm.String	TODO
watchPort	DefaultValue: Any	Type: Edm.String	TODO

Table 23.42: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket

The following types inherit from this type:
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupBucketByAlias
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortBucketByAlias
Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.buckets
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group
—Sel.Sel5056.PathPlanner.RestController.GroupReferences.buckets
——Sel.Sel5056.PathPlanner.RestController.GroupWithStats.group
———Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.groupsReferences
——Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.groupsReferences
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.buckets
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc.groups
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc

23.33.1 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupBucketByAlias

name	attributes	description
watchGroupConfigId	Type: Edm.String	TODO

Table 23.43: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupBucketByAlias

23.33.2 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortBucketByAlias

name	attributes	description
watchPortConfigId	Type: Edm.String	TODO

Table 23.44: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortBucketByAlias

23.34 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BucketStats

name	attributes	description
byteCount	Type: Edm.Int64	TODO
packetCount	Type: Edm.Int64	TODO

Table 23.45: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BucketStats

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BucketStats
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat.bucketStatsList
——Sel.Sel5056.PathPlanner.RestController.GroupWithStats.statistics
———Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.groupsReferences
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats.stats
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat.bucketStatsList
——Sel.Sel5056.PathPlanner.RestController.GroupWithStats.statistics
———Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.groupsReferences
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats.stats
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats

23.35 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowData

name	attributes	description
cookie	Type: Edm.Int64	TODO
nodeId	Type: Edm.String	TODO
priority	Type: Edm.Int64	TODO
tableId	Type: Edm.Byte	TODO

Table 23.46: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowData

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowData

23.36 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat

name	attributes	description
byteCount	Type: Edm.Int64	TODO
checkOverlap	Type: Edm.Boolean	TODO

cookie	Type: Edm.Int64	TODO
duration	Type: Edm.Duration	TODO
flags	Type: Edm.Int32	TODO
hardTimeout	Type: Edm.Int32	TODO
idleTimeout	Type: Edm.Int32	TODO
instructions	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction]	TODO
match	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match	TODO
noByteCounts	Type: Edm.Boolean	TODO
noPacketCounts	Type: Edm.Boolean	TODO
packetCount	Type: Edm.Int64	TODO
priority	Type: Edm.Int32	TODO
resetCounts	Type: Edm.Boolean	TODO
tableId	Type: Edm.Byte	TODO

Table 23.47: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat
—Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
——Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats

23.37 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupData

name	attributes	description
groupId	Type: Edm.Int64	TODO
nodeId	Type: Edm.String	TODO

Table 23.48: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupData

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupData

23.38 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat

name	attributes	description

bucketStatsList	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BucketStats]	TODO
byteCount	Type: Edm.Int64	TODO
duration	Type: Edm.Duration	TODO
groupId	Type: Edm.Int64	TODO
packetCount	Type: Edm.Int64	TODO
refCount	Type: Edm.Int64	TODO

Table 23.49: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat
—Sel.Sel5056.PathPlanner.RestController.GroupWithStats.statistics
——Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.groupsReferences
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats.stats
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats

23.39 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction

name	attributes	description
actions	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Action]	TODO

Table 23.50: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction

The following types inherit from this type:
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ApplyActions
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ClearActions
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GoToTable
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterInstruction
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions
Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat.instructions
——Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
———Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow.instructions
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow
—Sel.Sel5056.PathPlanner.RestController.FlowReferences.instructions
——Sel.Sel5056.PathPlanner.RestController.FlowWithStats.flow
———Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
——Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.flowReferences
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat.instructions

——Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
———Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats

23.39.1 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ApplyActions

This type has no properties

23.39.2 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ClearActions

This type has no properties

23.39.3 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GoToTable

name	attributes	description
tableId	Max: 255Min: 1Nullable: FalseStep: 1Type: Edm.Byte	TODO

Table 23.51: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GoToTable

23.39.4 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterInstruction

name	attributes	description
meterId	Nullable: FalseType: Edm.Int64	TODO

Table 23.52: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterInstruction

23.39.5 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions

This type has no properties

23.40 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match

name	attributes	description
matchFields	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.MatchField]	TODO

Table 23.53: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat.match
——Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
———Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences

——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow.match
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow
—Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions.match
—Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType.trafficMatch
——Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType
—Sel.Sel5056.LearnAndLock.ProposedCommunicationServiceType.trafficMatch
——Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection.proposedCommunicationServiceType
———Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat.match
——Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
———Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats

23.41 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.MatchField

name	attributes	description
value	Type: Edm.String	TODO

Table 23.54: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.MatchField

The following types inherit from this type:
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpOp
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpSpa
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpTpa
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthDst
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthSrc
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthType
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.InPort
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.IpProto
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Dst
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Src
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.TcpDst
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.TcpSrc
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.UdpDst
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.UdpSrc
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.VlanPcp
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.VlanVid
Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.MatchField
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match.matchFields

——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat.match
———Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow.match
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow
——Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions.match
——Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType.trafficMatch
———Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType
——Sel.Sel5056.LearnAndLock.ProposedCommunicationServiceType.trafficMatch
———Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection.proposedCommunicationServiceType
————Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat.match
———Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetFieldAction.field
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction.actions
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat.instructions
————Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
—————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
—————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow.instructions
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow
———Sel.Sel5056.PathPlanner.RestController.FlowReferences.instructions
————Sel.Sel5056.PathPlanner.RestController.FlowWithStats.flow
—————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
————Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.flowReferences
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat.instructions
————Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
—————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
—————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket.actions
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.buckets
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group
———Sel.Sel5056.PathPlanner.RestController.GroupReferences.buckets
————Sel.Sel5056.PathPlanner.RestController.GroupWithStats.group
—————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.groupsReferences
————Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.groupsReferences
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.buckets
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc.groups

—————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction.actions
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat.instructions
————Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
—————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
—————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow.instructions
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow
———Sel.Sel5056.PathPlanner.RestController.FlowReferences.instructions
————Sel.Sel5056.PathPlanner.RestController.FlowWithStats.flow
—————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
————Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.flowReferences
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat.instructions
————Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
—————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
—————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket.actions
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.buckets
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group
———Sel.Sel5056.PathPlanner.RestController.GroupReferences.buckets
————Sel.Sel5056.PathPlanner.RestController.GroupWithStats.group
—————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.groupsReferences
————Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.groupsReferences
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.buckets
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc.groups
—————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match.matchFields
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat.match
———Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow.match
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow
——Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions.match
——Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType.trafficMatch
———Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType
——Sel.Sel5056.LearnAndLock.ProposedCommunicationServiceType.trafficMatch
———Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection.proposedCommunicationServiceType
————Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat.match
———Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
————Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences

———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetFieldAction.field

23.41.1 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpOp

This type has no properties

23.41.2 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpSpa

name	attributes	description
mask	DataClass: IpAddressType: Edm.String	TODO
translatedValue	ReadOnly: True Type: Edm.String	TODO

Table 23.55: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpSpa

23.41.3 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpTpa

name	attributes	description
mask	DataClass: IpAddressType: Edm.String	TODO
translatedValue	ReadOnly: True Type: Edm.String	TODO

Table 23.56: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpTpa

23.41.4 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthDst

name	attributes	description
mask	DataClass: MacAddressType: Edm.String	TODO
translatedValue	ReadOnly: True Type: Edm.String	TODO

Table 23.57: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthDst

23.41.5 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthSrc

name	attributes	description
mask	DataClass: MacAddressType: Edm.String	TODO

translatedValue	ReadOnly: True Type: Edm.String	TODO

Table 23.58: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthSrc

23.41.6 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthType

This type has no properties

23.41.7 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.InPort

name	attributes	description
translatedValue	ReadOnly: TrueType: Edm.Int64	TODO

Table 23.59: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.InPort

23.41.8 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.IpProto

This type has no properties

23.41.9 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Dst

name	attributes	description
mask	DataClass: IpAddressType: Edm.String	TODO
translatedValue	ReadOnly: True Type: Edm.String	TODO

Table 23.60: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Dst

23.41.10 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Src

name	attributes	description
mask	DataClass: IpAddressType: Edm.String	TODO
translatedValue	ReadOnly: True Type: Edm.String	TODO

Table 23.61: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Src

23.41.11 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.-TcpDst

This type has no properties

23.41.12 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.TcpSrc

This type has no properties

23.41.13 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.UdpDst

This type has no properties

23.41.14 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.UdpSrc

This type has no properties

23.41.15 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.VlanPcp

This type has no properties

23.41.16 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.VlanVid

name	attributes	description
mask	Type: Edm.String	TODO

Table 23.62: Properties for type Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.VlanVid

23.42 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand

name	attributes	description
burstSize	Type: Edm.Int64	TODO
rate	Type: Edm.Int64	TODO

Table 23.63: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand

The following types inherit from this type:
-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandDrop
Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.meterBands
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter
—Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.MeterMetadata.meterBands

——Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions.meterMetadata
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.meterBands

23.42.1 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandDrop

This type has no properties

23.43 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandHeader

name	attributes	description
burstSize	Type: Edm.Int64	TODO
length	Type: Edm.Int32	TODO
rate	Type: Edm.Int64	TODO
type	Type: Sel.Sel5056.OpenFlowPlugin.Enums.OF13MeterBandType	TODO

Table 23.64: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandHeader

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandHeader
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDesc.bands
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions.meterDescriptionsList
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions

23.44 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDesc

name	attributes	description
bands	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandHeader]	TODO
flags	Type: Edm.Int32	TODO
length	Type: Edm.Int32	TODO
meterId	Type: Edm.Int64	TODO

Table 23.65: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDesc

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDesc
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions.meterDescriptionsList
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions

23.45 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStat

name	attributes	description
bandStats	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BandStat]	TODO
byteInCount	Type: Edm.Int64	TODO
duration	Type: Edm.Duration	TODO
flowCount	Type: Edm.Int64	TODO
meterId	Type: Edm.Int64	TODO
packetInCount	Type: Edm.Int64	TODO

Table 23.66: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStat

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStat
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats.meterStatList
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats

23.46 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port

name	attributes	description
advertised	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.FeatureBase]	TODO
config	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.ConfigBase]	TODO
curr	Type: Edm.Int64	TODO
currentSpeed	Type: Edm.Int64	TODO
hardwareAddress	Type: Edm.String	TODO
maxSpeed	Type: Edm.Int64	TODO
name	Type: Edm.String	TODO
peer	Type: Edm.Int64	TODO
portId	Type: Edm.Int64	TODO
state	Type: Sel.Sel5056.TopologyManager.Enums.OfpPortStatus	TODO
supported	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.FeatureBase]	TODO

Table 23.67: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port

Connections:

Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc.ports
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc

23.47 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStat

name	attributes	description
collisions	Type: Edm.Int64	TODO
duration	Type: Edm.Duration	TODO
portNo	Type: Edm.Int64	TODO
rxBytes	Type: Edm.Int64	TODO
rxCrcErr	Type: Edm.Int64	TODO
rxDropped	Type: Edm.Int64	TODO
rxErrors	Type: Edm.Int64	TODO
rxFrameErr	Type: Edm.Int64	TODO
rxOverError	Type: Edm.Int64	TODO
rxPackets	Type: Edm.Int64	TODO
txBytes	Type: Edm.Int64	TODO
txDropped	Type: Edm.Int64	TODO
txErrors	Type: Edm.Int64	TODO
txPackets	Type: Edm.Int64	TODO

Table 23.68: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStat

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStat
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats.portStatList
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats

23.48 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStat

name	attributes	description
activeCount	Type: Edm.Int64	TODO
lookupCount	Type: Edm.Int64	TODO
matchCount	Type: Edm.Int64	TODO
tableId	Type: Edm.Byte	TODO

Table 23.69: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStat

Connections:
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStat

—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats.tableStatList
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats

23.49 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency

name	attributes	description
dependencyResolved	Type: Edm.Boolean	TODO

Table 23.70: Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency

The following types inherit from this type:
-Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.FlowDeleteDependency
-Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupDeleteDependency
-Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupDependency
-Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.MeterDependency
-Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.FlowModifyDependency
-Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupModifyDependency
Connections:
Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowAddSynchronizationRequest.dependencies
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowAddSynchronizationRequest
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowDeleteSynchronizationRequest.dependencies
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowDeleteSynchronizationRequest
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModifySynchronizationRequest.dependencies
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModifySynchronizationRequest
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupAddSynchronizationRequest.dependencies
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupAddSynchronizationRequest
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupDeleteSynchronizationRequest.dependencies
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupDeleteSynchronizationRequest
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupModifySynchronizationRequest.dependencies
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupModifySynchronizationRequest
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterAddSynchronizationRequest.-

dependencies
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterAddSynchronizationRequest
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterDeleteSynchronizationRequest.dependencies
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterDeleteSynchronizationRequest
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterModifySynchronizationRequest.dependencies
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterModifySynchronizationRequest

23.49.1 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.FlowDeleteDependency

name	attributes	description
cookie	Nullable: FalseType: Edm.Int64	TODO

Table 23.71: Properties for type Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.FlowDeleteDependency

23.49.2 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupDeleteDependency

name	attributes	description
groupId	Nullable: FalseType: Edm.Int64	TODO

Table 23.72: Properties for type Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupDeleteDependency

23.49.3 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupDependency

name	attributes	description
groupId	Nullable: FalseType: Edm.Int64	TODO

Table 23.73: Properties for type Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupDependency

23.49.4 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.MeterDependency

name	attributes	description
meterId	Nullable: FalseType: Edm.Int64	TODO

Table 23.74: Properties for type Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.MeterDependency

23.49.5 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.-FlowModifyDependency

name	attributes	description
cookie	Nullable: FalseType: Edm.Int64	TODO

Table 23.75: Properties for type Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.FlowModifyDependency

23.49.6 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.-GroupModifyDependency

name	attributes	description
groupId	Nullable: FalseType: Edm.Int64	TODO

Table 23.76: Properties for type Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupModifyDependency

23.50 Sel.Sel5056.PathPlanner.DataObject.PathPlanHopCollection

name	attributes	description
destinationNodeId	Type: Edm.String	TODO
pathPlanHops	Type: [Sel.PathPlanHop]	TODO
sourceNodeId	Type: Edm.String	TODO

Table 23.77: Properties for Sel.Sel5056.PathPlanner.DataObject.PathPlanHopCollection

Connections:
Sel.Sel5056.PathPlanner.DataObject.PathPlanHopCollection

23.51 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.-DestinationDescriptions.Types.AmDestination

name	attributes	description
getFirstNodeId	Type: Edm.String	TODO
uniqueId	Type: Edm.String	TODO

Table 23.78: Properties for Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.AmDestination

The following types inherit from this type:
-Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.RelayFailoverDestination
-Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.SimpleDestination
-Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.TraditionalTiePointDestination
-Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.ProgrammableDestination
-Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.InbandDestination
Connections:
Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.AmDestination
—Sel.Sel5056.PathPlanner.RestController.PathPlanInfo.pathDestination
—Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidDestinationReservation.amDestination
——Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidDestinationReservation
——Sel.ReserveVlanVidRange.bindingParameter

23.51.1 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.RelayFailoverDestination

name	attributes	description
egressPortId	Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort	TODO
nodeId	Type: Edm.String	TODO

Table 23.79: Properties for type Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.RelayFailoverDestination

23.51.2 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.SimpleDestination

name	attributes	description
nodeId	Type: Edm.String	TODO

Table 23.80: Properties for type Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.SimpleDestination

23.51.3 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.TraditionalTiePointDestination

name	attributes	description
nodeIds	Type: [Edm.String]	TODO
reachableOpenFlowPortId	Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort	TODO
sourcePortId	Type: Edm.String	TODO

Table 23.81: Properties for type Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.-DestinationDescriptions.Types.TraditionalTiePointDestination

23.51.4 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.-DestinationDescriptions.Types.ProgrammableDestination

name	attributes	description
nodeId	Type: Edm.String	TODO

Table 23.82: Properties for type Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.-DestinationDescriptions.Types.ProgrammableDestination

23.51.5 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.-DestinationDescriptions.Types.InbandDestination

name	attributes	description
inBandPort	Type: Edm.String	TODO
nodeId	Type: Edm.String	TODO

Table 23.83: Properties for type Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.-DestinationDescriptions.Types.InbandDestination

23.52 Sel.Sel5056.PathPlanner.RestController.DirectionalLink

name	attributes	description
egressPortId	Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort	TODO
ingressPortId	Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort	TODO
linkId	Type: Sel.Sel5056.TopologyManager.Links.ConfigLink	TODO

Table 23.84: Properties for Sel.Sel5056.PathPlanner.RestController.DirectionalLink

Connections:
Sel.Sel5056.PathPlanner.RestController.DirectionalLink
—Sel.Sel5056.PathPlanner.RestController.PathPlanInfo.primaryLinks
—Sel.Sel5056.PathPlanner.RestController.PathPlanInfo.failOverLinks
—Sel.Sel5056.PathPlanner.RestController.PathPlanInfo.notFailedAroundLinks

23.53 Sel.Sel5056.PathPlanner.RestController.FlowReferences

name	attributes	description
cookie	Type: Edm.Int64	TODO
displayName	Type: Edm.String	TODO
id	Type: Edm.String	TODO
instructions	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction]	TODO
node	Type: Edm.String	TODO
tags	Type: [Sel.Sel5056.Tags.AmTag]	TODO

Table 23.85: Properties for Sel.Sel5056.PathPlanner.RestController.FlowReferences

Connections:
Sel.Sel5056.PathPlanner.RestController.FlowReferences
—Sel.Sel5056.PathPlanner.RestController.FlowWithStats.flow
——Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
—Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.flowReferences

23.54 Sel.Sel5056.PathPlanner.RestController.FlowWithStats

name	attributes	description
dataPathId	Type: Edm.Int64	TODO
flow	Type: Sel.Sel5056.PathPlanner.RestController.FlowReferences	TODO
outputs	Type: Edm.String	TODO
statistics	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat	TODO
switchDisplayName	Type: Edm.String	TODO

Table 23.86: Properties for Sel.Sel5056.PathPlanner.RestController.FlowWithStats

Connections:
Sel.Sel5056.PathPlanner.RestController.FlowWithStats
—Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences

23.55 Sel.Sel5056.PathPlanner.RestController.GroupReferences

name	attributes	description

buckets	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket]	TODO
displayName	Type: Edm.String	TODO
groupId	Type: Edm.Int64	TODO
groupType	Type: Sel.Sel5056.OpenFlowPlugin.Enums.OfpGroupType	TODO
id	Type: Edm.String	TODO
node	Type: Edm.String	TODO
tags	Type: [Sel.Sel5056.Tags.AmTag]	TODO

Table 23.87: Properties for Sel.Sel5056.PathPlanner.RestController.GroupReferences

Connections:
Sel.Sel5056.PathPlanner.RestController.GroupReferences
—Sel.Sel5056.PathPlanner.RestController.GroupWithStats.group
——Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.groupsReferences
—Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.groupsReferences

23.56 Sel.Sel5056.PathPlanner.RestController.GroupWithStats

name	attributes	description
dataPathId	Type: Edm.Int64	TODO
group	Type: Sel.Sel5056.PathPlanner.RestController.GroupReferences	TODO
outputs	Type: Edm.String	TODO
statistics	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat	TODO
switchDisplayName	Type: Edm.String	TODO

Table 23.88: Properties for Sel.Sel5056.PathPlanner.RestController.GroupWithStats

Connections:
Sel.Sel5056.PathPlanner.RestController.GroupWithStats
—Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.groupsReferences

23.57 Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats

name	attributes	description
flowReferences	Type: [Sel.Sel5056.PathPlanner.RestController.FlowWithStats]	TODO
groupsReferences	Type: [Sel.Sel5056.PathPlanner.RestController.GroupWithStats]	TODO

Table 23.89: Properties for Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats

Connections:
Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats

23.58 Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo

name	attributes	description
flowReferences	Type: [Sel.Sel5056.PathPlanner.RestController.FlowReferences]	TODO
groupsReferences	Type: [Sel.Sel5056.PathPlanner.RestController.GroupReferences]	TODO

Table 23.90: Properties for Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo

Connections:
Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo

23.59 Sel.Sel5056.PathPlanner.RestController.PathPlanInfo

name	attributes	description
failOverLinks	Type: [Sel.Sel5056.PathPlanner.RestController.DirectionalLink]	TODO
failOverNodes	Type: [Edm.String]	TODO
notFailedAroundLinks	Type: [Sel.Sel5056.PathPlanner.RestController.DirectionalLink]	TODO
pathDestination	Type: Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.AmDestination	TODO
primaryLinks	Type: [Sel.Sel5056.PathPlanner.RestController.DirectionalLink]	TODO
primaryNodes	Type: [Edm.String]	TODO

Table 23.91: Properties for Sel.Sel5056.PathPlanner.RestController.PathPlanInfo

Connections:
Sel.Sel5056.PathPlanner.RestController.PathPlanInfo

23.60 Sel.Sel5056.Tags.AmTag

This type has no properties
The following types inherit from this type:
-Sel.Sel5056.Tags.LogicalConnectionTag
-Sel.Sel5056.Tags.InBandPortIdTag
-Sel.Sel5056.Tags.GenericArpFlowTag
-Sel.Sel5056.Tags.SystemTag
-Sel.Sel5056.Tags.OpenFlowPortIdTag
-Sel.Sel5056.Tags.KeyValueTag
-Sel.Sel5056.Tags.InBandTag
-Sel.Sel5056.Tags.ControllerTag
-Sel.Sel5056.Tags.IpAddressTag
-Sel.Sel5056.Tags.MacAddressTag
-Sel.Sel5056.Tags.ReadoptTag
-Sel.Sel5056.Tags.ControllerRouteTag
-Sel.Sel5056.Tags.NoArpTag
-Sel.Sel5056.LogicalConnectionValidation.Tags.MatchableCst
-Sel.Sel5056.LearnAndLock.AutoAdoption.Tags.AutoAdoptionTag
Connections:
Sel.Sel5056.Tags.AmTag
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow.tags
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.tags
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.tags
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter
—Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType.tags
——Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType
—Sel.Sel5056.PathPlanner.RestController.FlowReferences.tags
——Sel.Sel5056.PathPlanner.RestController.FlowWithStats.flow
———Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
——Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.flowReferences
—Sel.Sel5056.PathPlanner.RestController.GroupReferences.tags
——Sel.Sel5056.PathPlanner.RestController.GroupWithStats.group
———Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.groupsReferences
——Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.groupsReferences
—Sel.Sel5056.TopologyManager.Nodes.ConfigNode.tags
——Sel.Sel5056.TopologyManager.Nodes.ConfigNode
—Sel.Sel5056.TopologyManager.Ports.ConfigPort.tags
——Sel.Sel5056.TopologyManager.Ports.ConfigPort
—Sel.Sel5056.TopologyManager.Links.ConfigLink.tags
——Sel.Sel5056.TopologyManager.Links.ConfigLink
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.tags
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc.groups
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.tags

23.60.1 Sel.Sel5056.Tags.LogicalConnectionTag

name	attributes	description
id	Type: Edm.String	TODO
logicalConnectionId	Type: Edm.String	TODO

Table 23.92: Properties for type Sel.Sel5056.Tags.LogicalConnectionTag

23.60.2 Sel.Sel5056.Tags.InBandPortIdTag

name	attributes	description
inbandPortId	Nullable: FalseType: Edm.Int64	TODO

Table 23.93: Properties for type Sel.Sel5056.Tags.InBandPortIdTag

23.60.3 Sel.Sel5056.Tags.GenericArpFlowTag

name	attributes	description
configNodeId	Type: Edm.String	TODO
id	Type: Edm.String	TODO

Table 23.94: Properties for type Sel.Sel5056.Tags.GenericArpFlowTag

23.60.4 Sel.Sel5056.Tags.SystemTag

This type has no properties

23.60.5 Sel.Sel5056.Tags.OpenFlowPortIdTag

name	attributes	description
portId	Nullable: FalseType: Edm.Int64	TODO

Table 23.95: Properties for type Sel.Sel5056.Tags.OpenFlowPortIdTag

23.60.6 Sel.Sel5056.Tags.KeyValueTag

name	attributes	description
id	Type: Edm.String	TODO
key	Type: Edm.String	TODO
value	Type: Edm.String	TODO

Table 23.96: Properties for type Sel.Sel5056.Tags.KeyValueTag

23.60.7 Sel.Sel5056.Tags.InBandTag

name	attributes	description
configNodeId	Type: Edm.String	TODO
id	Type: Edm.String	TODO

Table 23.97: Properties for type Sel.Sel5056.Tags.InBandTag

23.60.8 Sel.Sel5056.Tags.ControllerTag

name	attributes	description
id	Type: Edm.String	TODO

Table 23.98: Properties for type Sel.Sel5056.Tags.ControllerTag

23.60.9 Sel.Sel5056.Tags.IpAddressTag

name	attributes	description
id	Type: Edm.String	TODO
ipAddress	DataClass: IpAddressType: Edm.String	TODO

Table 23.99: Properties for type Sel.Sel5056.Tags.IpAddressTag

23.60.10 Sel.Sel5056.Tags.MacAddressTag

name	attributes	description
id	Type: Edm.String	TODO
macAddress	DataClass: MacAddressType: Edm.String	TODO

Table 23.100: Properties for type Sel.Sel5056.Tags.MacAddressTag

23.60.11 Sel.Sel5056.Tags.ReadoptTag

This type has no properties

23.60.12 Sel.Sel5056.Tags.ControllerRouteTag

name	attributes	description
id	Type: Edm.String	TODO

Table 23.101: Properties for type Sel.Sel5056.Tags.ControllerRouteTag

23.60.13 Sel.Sel5056.Tags.NoArpTag

This type has no properties

23.60.14 Sel.Sel5056.LogicalConnectionValidation.Tags.MatchableCst

This type has no properties

23.60.15 Sel.Sel5056.LearnAndLock.AutoAdoption.Tags.AutoAdoptionTag

name	attributes	description
learnAndLockSessionId	Type: Edm.String	TODO

Table 23.102: Properties for type Sel.Sel5056.LearnAndLock.AutoAdoption.Tags.AutoAdoptionTag

23.61 Sel.Sel5056.TopologyManager.Attributes.AmAttribute

This type has no properties
The following types inherit from this type:
-Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.Port.Sel2740SPortAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.Port.SelRelayFailoverModeAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationNodeAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.Node.OpenFlowAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.Node.IpAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.Node.FirmwareUpdateAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.Node.UserHintedIpAddressAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.Node.NoArpAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.Link.OfdpDiscovered
-Sel.Sel5056.TopologyManager.Attributes.Operational.Link.OpenFlowVerified
-Sel.Sel5056.TopologyManager.Attributes.Operational.Link.ArpVerified
-Sel.Sel5056.TopologyManager.Attributes.Operational.Link.UserHintedLinkAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.ControllerAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.EthernetAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.DoNotExpireAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedMappedSwitchPortAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.NetworkingAbstractionAttr
-Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedNodeAttr
-Sel.Sel5056.Southbound.DeviceManagement.CommissioningStateAttr
Connections:
Sel.Sel5056.TopologyManager.Attributes.AmAttribute
—Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr.discoveryPortAttribute
——Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr.discoveryPortAttribute

——Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.attributes

———Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

——Sel.GetLinkByPortAttribute.attribute

——Sel.GetNodeByAttribute.attribute

——Sel.GetPortByAttribute.attribute

——Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.attributes

———Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

——Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.attributes

———Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

—Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.attributes

——Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

—Sel.GetLinkByPortAttribute.attribute

—Sel.GetNodeByAttribute.attribute

—Sel.GetPortByAttribute.attribute

—Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.attributes

——Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

—Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.attributes

——Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

23.61.1 Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr

name	attributes	description
advertised	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.FeatureBase]	TODO
config	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.ConfigBase]	TODO
currentFeatures	Nullable: False Type: Edm.Int64	TODO
currentSpeed	Nullable: False Type: Edm.Int64	TODO
dataPathId	Nullable: False Type: Edm.Int64	TODO
hardwareAddress	DataClass: MacAddress Type: Edm.String	TODO
id	ReadOnly: True Type: Edm.String	TODO
maxSpeed	Nullable: False Type: Edm.Int64	TODO
name	Type: Edm.String	TODO
ofState	Nullable: False Type: Sel.Sel5056.TopologyManager.Enums.OfpPortStatus	TODO

peer	Nullable: False Type: Edm.Int64	TODO
portId	Nullable: False Type: Edm.Int64	TODO
prettyDataPathId	ReadOnly: True Type: Edm.String	TODO
supported	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.FeatureBase]	TODO

Table 23.103: Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr

23.61.2 Sel.Sel5056.TopologyManager.Attributes.Operational.Port.Sel2740SPortAttr

This type has no properties

23.61.3 Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationAttr

name	attributes	description
ethType	Nullable: False Type: Edm.Int32	TODO
id	ReadOnly: True Type: Edm.String	TODO
macAddress	Type: Edm.String	TODO
providerInformation	Type: [Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationProviderInformation]	TODO

Table 23.104: Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationAttr

23.61.4 Sel.Sel5056.TopologyManager.Attributes.Operational.Port.SelRelayFailoverModeAttr

This type has no properties

23.61.5 Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationNodeAttr

name	attributes	description
id	ReadOnly: TrueType: Edm.String	TODO
macAddress	Type: Edm.String	TODO

Table 23.105: Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.Port.-EthernetApplicationNodeAttr

23.61.6 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.-OpenFlowAttr

name	attributes	description
dataPathId	Nullable: False Type: Edm.Int64	TODO
id	ReadOnly: TrueType: Edm.String	TODO
prettyDataPathId	ReadOnly: TrueType: Edm.String	TODO

Table 23.106: Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.Node.-OpenFlowAttr

23.61.7 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.-IpAttr

name	attributes	description
ipAddress	DataClass: IpAddressType: Edm.String	TODO

Table 23.107: Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.Node.Ip-Attr

23.61.8 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.-SelSapphireAttr

name	attributes	description
autoDiscoveryVersion	Type: Edm.String	TODO
communicationBand	Nullable: False Type: Sel.Sel5056.TopologyManager.Enums.-CommunicationBand	TODO
dataPathId	Nullable: False Type: Edm.Int64	TODO
deviceFramework	Nullable: False Type: Sel.Sel5056.TopologyManager.Attributes.-Operational.Node.DeviceFramework	TODO
discoveryMac	Type: Edm.String	TODO
discoveryPort	Nullable: False Type: Edm.Int64	TODO
discoveryPortAttribute	Type: Sel.Sel5056.TopologyManager.Attributes.-AmAttribute	TODO

discoverySource	Nullable: False Type: Sel.Sel5056.TopologyManager.Enums.PacketSource	TODO
ethFrontIp	Type: Edm.String	TODO
ethFrontMac	Type: Edm.String	TODO
ethRearIp	Type: Edm.String	TODO
ethRearMac	Type: Edm.String	TODO
failedEventQueryAttempts	Nullable: False Type: Edm.Int32	TODO
id	ReadOnly: True Type: Edm.String	TODO
isCommissioned	Nullable: False Type: Edm.Boolean	TODO
lastAutoDiscoveryReceived	Nullable: False Type: Edm.DateTimeOffset	TODO
lastQueriedEventId	Nullable: False Type: Edm.Int64	TODO
modelNumber	Type: Edm.String	TODO
name	Type: Edm.String	TODO
omniPort	Nullable: False Type: Edm.Int32	TODO
prettyDataPathId	ReadOnly: True Type: Edm.String	TODO
serialNumber	Type: Edm.String	TODO
x509Fingerprint	Type: Edm.String	TODO

Table 23.108: Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr

23.61.9 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.FirmwareUpdateAttr

name	attributes	description
firmwareFileUploadPercentComplete	Nullable: False Type: Edm.Int32	TODO
firmwareUploadDeviceStatus	Type: Edm.String	TODO
lastUpdate	Nullable: FalseType: Edm.DateTimeOffset	TODO

Table 23.109: Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.Node.FirmwareUpdateAttr

23.61.10 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.UserHintedIpAddressAttr

name	attributes	description
id	Type: Edm.String	TODO
ipAddress	DataClass: IpAddressType: Edm.String	TODO
openFlowConfigPortId	Type: Edm.String	TODO

Table 23.110: Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.Node.-UserHintedIpAddressAttr

23.61.11 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.-NoArpAttr

This type has no properties

23.61.12 Sel.Sel5056.TopologyManager.Attributes.Operational.Link.-OfdpDiscovered

This type has no properties

23.61.13 Sel.Sel5056.TopologyManager.Attributes.Operational.Link.-OpenFlowVerified

This type has no properties

23.61.14 Sel.Sel5056.TopologyManager.Attributes.Operational.Link.-ArpVerified

This type has no properties

23.61.15 Sel.Sel5056.TopologyManager.Attributes.Operational.Link.-UserHintedLinkAttr

This type has no properties

23.61.16 Sel.Sel5056.TopologyManager.Attributes.Operational.-ControllerAttr

This type has no properties

23.61.17 Sel.Sel5056.TopologyManager.Attributes.Operational.-EthernetAttr

name	attributes	description

id	ReadOnly: TrueType: Edm.String	TODO
macAddress	Type: Edm.String	TODO

Table 23.111: Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.EthernetAttr

23.61.18 Sel.Sel5056.TopologyManager.Attributes.Operational.DoNotExpireAttr

This type has no properties

23.61.19 Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedMappedSwitchPortAttr

name	attributes	description
configPortId	Type: Edm.String	TODO
id	Type: Edm.String	TODO

Table 23.112: Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedMappedSwitchPortAttr

23.61.20 Sel.Sel5056.TopologyManager.Attributes.Operational.NetworkingAbstractionAttr

name	attributes	description
id	Type: Edm.String	TODO
mappedEthernetApplicationPortAttr	Type: Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationAttr	TODO
mappedEthernetPortAttr	Type: Sel.Sel5056.TopologyManager.Attributes.Operational.EthernetAttr	TODO
mappedOpenFlowPortAttr	Type: Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr	TODO
mappedSilentHostAttr	Type: Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedMappedSwitchPortAttr	TODO

Table 23.113: Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.NetworkingAbstractionAttr

23.61.21 Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedNodeAttr

name	attributes	description
configNodeId	Type: Edm.String	TODO
id	Type: Edm.String	TODO

Table 23.114: Properties for type Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedNodeAttr

23.61.22 Sel.Sel5056.Southbound.DeviceManagement.CommissioningStateAttr

name	attributes	description
state	Nullable: FalseType: Sel.BlueFrame.Core.ErrorState	TODO
status	Type: Edm.String	TODO

Table 23.115: Properties for type Sel.Sel5056.Southbound.DeviceManagement.CommissioningStateAttr

23.62 Sel.Sel5056.TopologyManager.Attributes.-Operational.Port.EthernetApplicationProviderInformation

name	attributes	description
destinationMac	Type: Edm.String	TODO
vlan	Type: Edm.Int32	TODO

Table 23.116: Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationProviderInformation

Connections:
Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationProviderInformation
—Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationAttr.providerInformation
——Sel.Sel5056.TopologyManager.Attributes.Operational.NetworkingAbstractionAttr.mappedEthernetApplicationPortAttr
———Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr.discoveryPortAttribute
———Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.attributes
————Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
———Sel.GetLinkByPortAttribute.attribute
———Sel.GetNodeByAttribute.attribute
———Sel.GetPortByAttribute.attribute
———Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.attributes
————Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
———Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.attributes

————Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

23.63 Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractSettingBase

This type has no properties
The following types inherit from this type:
-Sel.Sel5056.TopologyManager.Configurations.Abstraction.TraditionalSwitch
Connections:
Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractSettingBase
—Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractionConfig.abstractionSetting
——Sel.Sel5056.TopologyManager.Nodes.ConfigNode.configurations
———Sel.Sel5056.TopologyManager.Nodes.ConfigNode
——Sel.Sel5056.TopologyManager.Ports.ConfigPort.configurations
———Sel.Sel5056.TopologyManager.Ports.ConfigPort
——Sel.Sel5056.TopologyManager.Links.ConfigLink.configurations
———Sel.Sel5056.TopologyManager.Links.ConfigLink
—Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractionConfig.abstractionSetting
—Sel.AddAbstractNode.configSetting

23.63.1 Sel.Sel5056.TopologyManager.Configurations.Abstraction.TraditionalSwitch

This type has no properties

23.64 Sel.Sel5056.TopologyManager.Configurations.AmConfig

This type has no properties
The following types inherit from this type:
-Sel.Sel5056.TopologyManager.Configurations.ControllerConfig
-Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriority
-Sel.Sel5056.TopologyManager.Configurations.OpenFlow.OpenFlowConfig
-Sel.Sel5056.TopologyManager.Configurations.FailOver.SelRelayFailoverModeConfig
-Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractionConfig
-Sel.Sel5056.TopologyManager.Configurations.Sapphire.AlternateIpConfig
Connections:
Sel.Sel5056.TopologyManager.Configurations.AmConfig
—Sel.Sel5056.TopologyManager.Nodes.ConfigNode.configurations
——Sel.Sel5056.TopologyManager.Nodes.ConfigNode
—Sel.Sel5056.TopologyManager.Ports.ConfigPort.configurations
——Sel.Sel5056.TopologyManager.Ports.ConfigPort
—Sel.Sel5056.TopologyManager.Links.ConfigLink.configurations
——Sel.Sel5056.TopologyManager.Links.ConfigLink

23.64.1 Sel.Sel5056.TopologyManager.Configurations.ControllerConfig

This type has no properties

23.64.2 Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriority

name	attributes	description
priority	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.AmSystemPriority	TODO

Table 23.117: Properties for type Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriority

23.64.3 Sel.Sel5056.TopologyManager.Configurations.OpenFlow.OpenFlowConfig

This type has no properties

23.64.4 Sel.Sel5056.TopologyManager.Configurations.FailOver.SelRelayFailoverModeConfig

This type has no properties

23.64.5 Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractionConfig

name	attributes	description
abstractionSetting	Nullable: False Type: Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractSettingBase	The type of the abstract node.

Table 23.118: Properties for type Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractionConfig

23.64.6 Sel.Sel5056.TopologyManager.Configurations.Sapphire.AlternateIpConfig

name	attributes	description
alternateIp	Nullable: False Type: Edm.String	TODO
alternateSubnetMask	DataClass: IpAddressNullable: FalseType: Edm.String	TODO

Table 23.119: Properties for type Sel.Sel5056.TopologyManager.Configurations.Sapphire.AlternateIpConfig

23.65 Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.AmSystemPriority

This type has no properties
The following types inherit from this type:
-Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.SystemCritical
-Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.SystemRequired
Connections:
Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.AmSystemPriority
—Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriority.priority
——Sel.Sel5056.TopologyManager.Nodes.ConfigNode.configurations
———Sel.Sel5056.TopologyManager.Nodes.ConfigNode
——Sel.Sel5056.TopologyManager.Ports.ConfigPort.configurations
———Sel.Sel5056.TopologyManager.Ports.ConfigPort
——Sel.Sel5056.TopologyManager.Links.ConfigLink.configurations
———Sel.Sel5056.TopologyManager.Links.ConfigLink
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowAddSynchronizationRequest.priorityConfiguration
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowAddSynchronizationRequest
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowDeleteSynchronizationRequest.priorityConfiguration
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowDeleteSynchronizationRequest
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModifySynchronizationRequest.priorityConfiguration
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModifySynchronizationRequest
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupAddSynchronizationRequest.priorityConfiguration
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupAddSynchronizationRequest
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupDeleteSynchronizationRequest.priorityConfiguration
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupDeleteSynchronizationRequest
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupModifySynchronizationRequest.priorityConfiguration
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupModifySynchronizationRequest
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterAddSynchronizationRequest.priorityConfiguration
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterAddSynchronizationRequest
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterDeleteSynchronizationRequest.priorityConfiguration
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterDeleteSynchronization-

Request
—Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterModifySynchronizationRequest.-priorityConfiguration
——Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterModifySynchronization-Request

23.65.1 Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-SystemPriorities.SystemCritical

This type has no properties

23.65.2 Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-SystemPriorities.SystemRequired

This type has no properties

23.66 Sel.Sel5056.TopologyManager.OpenFlow.Meter.-RateTypeFlagObjects.RateTypeBase

This type has no properties
The following types inherit from this type:
-Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.Burst
-Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.KilobitesPerSecond
-Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.PacketsPerSecond
-Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.Stats
Connections:
Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.RateTypeBase
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.flags
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter
—Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.MeterMetadata.flags
——Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions.meter-Metadata
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.flags
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures.capabilities
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures

23.66.1 Sel.Sel5056.TopologyManager.OpenFlow.Meter.-RateTypeFlagObjects.Burst

This type has no properties

23.66.2 Sel.Sel5056.TopologyManager.OpenFlow.Meter.-RateTypeFlagObjects.KilobitesPerSecond

This type has no properties

23.66.3 Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.PacketsPerSecond

This type has no properties

23.66.4 Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.Stats

This type has no properties

23.67 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.ConfigBase

This type has no properties
The following types inherit from this type:
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.NoForward
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.NoPacketIn
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.NoReceive
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.PortDown
Connections:
Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.ConfigBase
—Sel.SendPortMod.configFlags
—Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr.config
——Sel.Sel5056.TopologyManager.Attributes.Operational.NetworkingAbstractionAttr.mappedOpenFlowPortAttr
———Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr.discoveryPortAttribute
———Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.attributes
————Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
———Sel.GetLinkByPortAttribute.attribute
———Sel.GetNodeByAttribute.attribute
———Sel.GetPortByAttribute.attribute
———Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.attributes
————Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
———Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.attributes
————Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port.config
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc.ports
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc
—Sel.SendPortMod.configFlags

23.67.1 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.NoForward

This type has no properties

23.67.2 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-ConfigFlagObjects.NoPacketIn

This type has no properties

23.67.3 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-ConfigFlagObjects.NoReceive

This type has no properties

23.67.4 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-ConfigFlagObjects.PortDown

This type has no properties

23.68 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.FeatureBase

This type has no properties
The following types inherit from this type:
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.AutoNegotiation
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.Copper
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.Fiber
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.FortyGigaBitFullDuplex
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneGigaBitFullDuplex
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneGigaBitHalfDuplex
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneHundredGigaBitFullDuplex
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneHundredMegaBitFullDuplex
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneHundredMegaBitHalfDuplex
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneTeraBitFullDuplex
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OtherBitRate
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.Pause
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.PauseAsymmetric
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.TenGigaBitFullDuplex
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.TenMegaBitFullDuplex
-Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.TenMegaBitHalfDuplex
Connections:
Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.FeatureBase
—Sel.SendPortMod.featuresFlags
—Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr.advertised
——Sel.Sel5056.TopologyManager.Attributes.Operational.NetworkingAbstractionAttr.mappedOpenFlowPortAttr

———Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr.discoveryPortAttribute

———Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.attributes

————Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

———Sel.GetLinkByPortAttribute.attribute

———Sel.GetNodeByAttribute.attribute

———Sel.GetPortByAttribute.attribute

———Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.attributes

————Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

———Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.attributes

————Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

—Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr.supported

——Sel.Sel5056.TopologyManager.Attributes.Operational.NetworkingAbstractionAttr.mappedOpenFlowPortAttr

—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port.advertised

——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc.ports

———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc

—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port.supported

——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc.ports

———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc

—Sel.SendPortMod.featuresFlags

23.68.1 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.AutoNegotiation

This type has no properties

23.68.2 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.Copper

This type has no properties

23.68.3 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.Fiber

This type has no properties

23.68.4 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.FortyGigaBitFullDuplex

This type has no properties

23.68.5 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneGigaBitFullDuplex

This type has no properties

23.68.6 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.OneGigaBitHalfDuplex

This type has no properties

23.68.7 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.OneHundredGigaBitFullDuplex

This type has no properties

23.68.8 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.OneHundredMegaBitFullDuplex

This type has no properties

23.68.9 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.OneHundredMegaBitHalfDuplex

This type has no properties

23.68.10 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.OneTeraBitFullDuplex

This type has no properties

23.68.11 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.OtherBitRate

This type has no properties

23.68.12 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.Pause

This type has no properties

23.68.13 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.PauseAsymmetric

This type has no properties

23.68.14 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.TenGigaBitFullDuplex

This type has no properties

23.68.15 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.TenMegaBitFullDuplex

This type has no properties

23.68.16 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.TenMegaBitHalfDuplex

This type has no properties

Chapter 24

OData Actions

24.1 Sel.AcceptAllProposedLearnedLogicalConnections

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession
This type has no properties

24.2 Sel.AcceptLearnedLogicalConnections

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

name	attributes	description
learnedLogicalConnectionIds	Type: [Edm.String]	TODO

Table 24.1: Properties for Sel.AcceptLearnedLogicalConnections

24.3 Sel.AcknowledgeLearnAndLockSession

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

name	attributes	description
sessionId	Type: Edm.String	TODO

Table 24.2: Properties for Sel.AcknowledgeLearnAndLockSession

24.4 Sel.AddAbstractNode

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

name	attributes	description

configSetting	Type: Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractSettingBase	The type of abstract node

Table 24.3: Properties for Sel.AddAbstractNode

24.5 Sel.AddOperationalLinkFromHostToSwitch

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

name	attributes	description
switchPortId	Type: Edm.String	TODO

Table 24.4: Properties for Sel.AddOperationalLinkFromHostToSwitch

24.6 Sel.AddSilentHost

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

name	attributes	description
configNodeId	Type: Edm.String	TODO

Table 24.5: Properties for Sel.AddSilentHost

24.7 Sel.Adopt

This action is bound so requires an object id.
There are 3 instances of this action
-Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
-Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
-Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

24.7.1 Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

The binding parameter type is: Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
This type has no properties

24.7.2 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

The binding parameter type is: Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
This type has no properties

24.7.3 Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

The binding parameter type is: Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

This type has no properties

24.8 Sel.AdoptWithConfig

This action is bound so requires an object id.
There are 3 instances of this action
-Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
-Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
-Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

24.8.1 Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

The binding parameter type is: Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

name	attributes	description
configKey	Type: Edm.String	Sel.Sel5056.TopologyManager.Nodes.ConfigNode

Table 24.6: Properties for Sel.AdoptWithConfig

24.8.2 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

The binding parameter type is: Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

name	attributes	description
configKey	Type: Edm.String	Sel.Sel5056.TopologyManager.Nodes.ConfigNode

Table 24.7: Properties for Sel.AdoptWithConfig

24.8.3 Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

The binding parameter type is: Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

name	attributes	description
configKey	Type: Edm.String	Sel.Sel5056.TopologyManager.Nodes.ConfigNode

Table 24.8: Properties for Sel.AdoptWithConfig

24.9 Sel.AsPem

This action is bound so requires an object id.
The binding parameter type is: Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation
This type has no properties

24.10 Sel.Backup

This action is not bound.

name	attributes	description
password	Type: Edm.String	TODO

Table 24.9: Properties for Sel.Backup

24.11 Sel.BeginLearnAndLock

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

name	attributes	description
autoAdoptConfig	Type: Sel.Sel5056.LearnAndLock.Configuration.AutoAdoptConfig	TODO
logicalConnectionLearningConfig	Type: Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.Configuration.LogicalConnectionLearningConfig	TODO

Table 24.10: Properties for Sel.BeginLearnAndLock

24.12 Sel.ClearLearnedLogicalConnections

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

name	attributes	description
learnedLogicalConnectionIds	Type: [Edm.String]	TODO

Table 24.11: Properties for Sel.ClearLearnedLogicalConnections

24.13 Sel.ClearTransaction

This action is bound so requires an object id.
There are 5 instances of this action
-Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
-Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
-Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
-Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
-Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction

24.13.1 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction

The binding parameter type is: Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
This type has no properties

24.13.2 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction

The binding parameter type is: Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
This type has no properties

24.13.3 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction

The binding parameter type is: Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
This type has no properties

24.13.4 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction

The binding parameter type is: Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
This type has no properties

24.13.5 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction

The binding parameter type is: Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
This type has no properties

24.14 Sel.Commit

This action is bound so requires an object id.
There are 5 instances of this action
-Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
-Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
-Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
-Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
-Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction

24.14.1 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction

The binding parameter type is: Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
This type has no properties

24.14.2 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction

The binding parameter type is: Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
This type has no properties

24.14.3 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction

The binding parameter type is: Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
This type has no properties

24.14.4 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction

The binding parameter type is: Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
This type has no properties

24.14.5 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction

The binding parameter type is: Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
This type has no properties

24.15 Sel.DeclineAllProposedLearnedLogicalConnections

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession
This type has no properties

24.16 Sel.DeclineLearnedLogicalConnections

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

name	attributes	description
learnedLogicalConnectionIds	Type: [Edm.String]	TODO

Table 24.12: Properties for Sel.DeclineLearnedLogicalConnections

24.17 Sel.DeleteAll

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection

This type has no properties

24.18 Sel.DeleteLearnAndLockSession

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

name	attributes	description
sessionId	Type: Edm.String	TODO

Table 24.13: Properties for Sel.DeleteLearnAndLockSession

24.19 Sel.DetectSelRelayFailoverPort

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
This type has no properties

24.20 Sel.DisableApplicationLink

This action is bound so requires an object id.
The binding parameter type is: Sel.BlueFrame.ApplicationRegistration.ApplicationLink
This type has no properties

24.21 Sel.DisableControllerArpToNode

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
This type has no properties

24.22 Sel.Disconnect

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
This type has no properties

24.23 Sel.EnableApplicationLink

This action is bound so requires an object id.
The binding parameter type is: Sel.BlueFrame.ApplicationRegistration.ApplicationLink
This type has no properties

24.24 Sel.EnableControllerArpToNode

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
This type has no properties

24.25 Sel.FactoryDefaultReset

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
This type has no properties

24.26 Sel.FirmwareUpgrade

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

name	attributes	description
fileName	Type: Sel.FileArgument	TODO

Table 24.14: Properties for Sel.FirmwareUpgrade

24.27 Sel.GetAvailableGroups

This action is bound so requires an object id.
The binding parameter type is: Sel.BlueFrame.Core.SecurityManager.AuthService
This type has no properties

24.28 Sel.GetHardTimeoutTimeRemaining

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

name	attributes	description
sessionId	Type: Edm.String	TODO

Table 24.15: Properties for Sel.GetHardTimeoutTimeRemaining

24.29 Sel.GetLearnAndLockCounters

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

name	attributes	description
sessionId	Type: Edm.String	TODO

Table 24.16: Properties for Sel.GetLearnAndLockCounters

24.30 Sel.GetLinkByPortAttribute

This action is not bound.

name	attributes	description
attribute	Type: Sel.Sel5056.TopologyManager.Attributes.AmAttribute	TODO

Table 24.17: Properties for Sel.GetLinkByPortAttribute

24.31 Sel.GetLogicalConnectionStats

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection
This type has no properties

24.32 Sel.GetNodeByAttribute

This action is not bound.

name	attributes	description
attribute	Type: Sel.Sel5056.TopologyManager.Attributes.AmAttribute	TODO

Table 24.18: Properties for Sel.GetNodeByAttribute

24.33 Sel.GetNodesByIp

This action is not bound.

name	attributes	description
ip	Type: Edm.String	TODO

Table 24.19: Properties for Sel.GetNodesByIp

24.34 Sel.GetNodesThatRequireSynchronization

This action is not bound.
This type has no properties

24.35 Sel.GetOpenFlowInformation

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection
This type has no properties

24.36 Sel.GetPathPlanInformation

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection
This type has no properties

24.37 Sel.GetPortByAttribute

This action is not bound.

name	attributes	description
attribute	Type: Sel.Sel5056.TopologyManager.Attributes.AmAttribute	TODO

Table 24.20: Properties for Sel.GetPortByAttribute

24.38 Sel.GetStateBeginEndTimestamps

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

name	attributes	description
sessionId	Type: Edm.String	TODO

Table 24.21: Properties for Sel.GetStateBeginEndTimestamps

24.39 Sel.GetStatusMessages

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

name	attributes	description
sessionId	Type: Edm.String	TODO

Table 24.22: Properties for Sel.GetStatusMessages

24.40 Sel.GetUserPreference

This action is not bound.

name	attributes	description
defaultValue	Type: Edm.String	TODO
key	Type: Edm.String	TODO

Table 24.23: Properties for Sel.GetUserPreference

24.41 Sel.Heartbeat

This action is not bound.
This type has no properties

24.42 Sel.InitiateApplicationRegistration

This action is bound so requires an object id.
The binding parameter type is: Sel.BlueFrame.ApplicationRegistration.ApplicationLink

name	attributes	description
url	Type: Edm.String	TODO

Table 24.24: Properties for Sel.InitiateApplicationRegistration

24.43 Sel.IsCommissioned

This action is not bound.
This type has no properties

24.44 Sel.LearnAndLockIndividualSessionExportCSV

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

name	attributes	description
sessionId	Type: Edm.String	TODO

Table 24.25: Properties for Sel.LearnAndLockIndividualSessionExportCSV

24.45 Sel.MarkPortAndAddRelayFailoverLink

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

name	attributes	description
switchPortId	Type: Edm.String	TODO

Table 24.26: Properties for Sel.MarkPortAndAddRelayFailoverLink

24.46 Sel.MarkPortAndPerformBlockingDetectSelRelayFailover

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
This type has no properties

24.47 Sel.MarkPortAsSelRelayFailover

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
This type has no properties

24.48 Sel.MergeNodes

This action is not bound.

name	attributes	description
firstOperationalNodeId	Type: Edm.String	TODO
secondOperationalNodeId	Type: Edm.String	TODO

Table 24.27: Properties for Sel.MergeNodes

24.49 Sel.PacketOut

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

name	attributes	description
packetData	Type: Edm.String	TODO
portId	Type: Edm.Int64	TODO

Table 24.28: Properties for Sel.PacketOut

24.50 Sel.ProposeLearnedLogicalConnections

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

name	attributes	description
learnedLogicalConnectionIds	Type: [Edm.String]	TODO

Table 24.29: Properties for Sel.ProposeLearnedLogicalConnections

24.51 Sel.Reboot

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
This type has no properties

24.52 Sel.RemoveSelRelayFailoverFromPort

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
This type has no properties

24.53 Sel.ReplaceConfig

This action is bound so requires an object id.
There are 3 instances of this action
-Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
-Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
-Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

24.53.1 Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

The binding parameter type is: Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

name	attributes	description
configKey	Type: Edm.String	TODO

Table 24.30: Properties for Sel.ReplaceConfig

24.53.2 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

The binding parameter type is: Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

name	attributes	description
configKey	Type: Edm.String	TODO

Table 24.31: Properties for Sel.ReplaceConfig

24.53.3 Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

The binding parameter type is: Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

name	attributes	description
configKey	Type: Edm.String	TODO

Table 24.32: Properties for Sel.ReplaceConfig

24.54 Sel.ReplanAll

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection
This type has no properties

24.55 Sel.ReplanInbandPath

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
This type has no properties

24.56 Sel.ReserveVlanVidRange

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation

name	attributes	description
end	Type: Edm.Int32	TODO
start	Type: Edm.Int32	TODO

Table 24.33: Properties for Sel.ReserveVlanVidRange

24.57 Sel.ResetCounters

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow
This type has no properties

24.58 Sel.Restore

This action is not bound.

name	attributes	description
fileName	Type: Sel.BlueFrame.Core.FileArgument	TODO

password	Type: Edm.String	TODO

Table 24.34: Properties for Sel.Restore

24.59 Sel.ResubmitLogicalConnection

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection
This type has no properties

24.60 Sel.RevokeCertificate

This action is bound so requires an object id.
The binding parameter type is: Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation
This type has no properties

24.61 Sel.RevokeCertificateByPurpose

This action is not bound.

name	attributes	description
purpose	Type: Sel.BlueFrame.Core.TrustAuthority.CertificatePurpose	TODO

Table 24.35: Properties for Sel.RevokeCertificateByPurpose

24.62 Sel.RevokeCertificateByThumbprint

This action is not bound.

name	attributes	description
thumbprint	Type: Edm.String	TODO

Table 24.36: Properties for Sel.RevokeCertificateByThumbprint

24.63 Sel.SendPortMod

This action is bound so requires an object id.
There are 2 instances of this action
-Sel.Sel5056.TopologyManager.Ports.ConfigPort
-Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

24.63.1 Sel.Sel5056.TopologyManager.Ports.ConfigPort

The binding parameter type is: Sel.Sel5056.TopologyManager.Ports.ConfigPort

name	attributes	description
configFlags	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.ConfigBase]	TODO
featuresFlags	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.FeatureBase]	TODO

Table 24.37: Properties for Sel.SendPortMod

24.63.2 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

The binding parameter type is: Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

name	attributes	description
configFlags	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.ConfigBase]	TODO
featuresFlags	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.FeatureBase]	TODO

Table 24.38: Properties for Sel.SendPortMod

24.64 Sel.SetDone

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession
This type has no properties

24.65 Sel.SetUserPreference

This action is not bound.

name	attributes	description
key	Type: Edm.String	TODO
value	Type: Edm.String	TODO

Table 24.39: Properties for Sel.SetUserPreference

24.66 Sel.StopLearnAndLock

This action is bound so requires an object id.
The binding parameter type is: Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession
This type has no properties

24.67 Sel.SynchronizeConfiguration

This action is not bound.

name	attributes	description
configNodeId	Type: Edm.String	TODO

Table 24.40: Properties for Sel.SynchronizeConfiguration

24.68 Sel.SynchronizeOpenFlowConfiguration

This action is not bound.

name	attributes	description
configNodeId	Type: Edm.String	TODO

Table 24.41: Properties for Sel.SynchronizeOpenFlowConfiguration

24.69 Sel.TestAuthenticateUser

This action is bound so requires an object id.
The binding parameter type is: Sel.BlueFrame.Core.SecurityManager.AuthService

name	attributes	description
password	Type: Edm.String	TODO
username	Type: Edm.String	TODO

Table 24.42: Properties for Sel.TestAuthenticateUser

24.70 Sel.Unadopt

This action is bound so requires an object id.
There are 3 instances of this action
-Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
-Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
-Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

24.70.1 Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

The binding parameter type is: Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
This type has no properties

24.70.2 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

The binding parameter type is: Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
This type has no properties

24.70.3 Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

The binding parameter type is: Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink
This type has no properties

24.71 Sel.UpdatePassword

This action is bound so requires an object id.
The binding parameter type is: Sel.BlueFrame.Core.SecurityManager.User

name	attributes	description
newPassword	Type: Edm.String	TODO
oldPassword	Type: Edm.String	TODO

Table 24.43: Properties for Sel.UpdatePassword

24.72 Sel.UploadCertificate

This action is not bound.

name	attributes	description
base64Certificate	Type: Edm.String	TODO
certificatePassword	Type: Edm.String	TODO
purpose	Type: Sel.BlueFrame.Core.TrustAuthority.CertificatePurpose	TODO

Table 24.44: Properties for Sel.UploadCertificate

24.73 Sel.UserInitiatedHostDiscovery

This action is not bound.

name	attributes	description

ip	Type: Edm.String	TODO

Table 24.45: Properties for Sel.UserInitiatedHostDiscovery

Chapter 25

OData Enums

25.1 Sel.BlueFrame.Core.ErrorState

key	value	description
Failure	2	TODO
FlowStatTransactionFailure	4	Not used
InProgress	0	TODO
Success	1	TODO
ValidationError	3	TODO

Table 25.1: Sel.BlueFrame.Core.ErrorState

Relationship between this type and other types:
Sel.BlueFrame.Core.ErrorState
—Sel.BlueFrame.Core.SecurityManager.AuthService.errorState
——Sel.BlueFrame.Core.SecurityManager.AuthService
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow.errorState
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.errorState
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.errorState
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter
—Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection.errorState
——Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection
—Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType.errorState
——Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType
—Sel.Sel5056.Southbound.DeviceManagement.CommissioningStateAttr.state
——Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr.discoveryPortAttribute
——Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.attributes
———Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
——Sel.GetLinkByPortAttribute.attribute
——Sel.GetNodeByAttribute.attribute
——Sel.GetPortByAttribute.attribute
——Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.attributes

———Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
——Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.attributes
———Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.errorState
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc.groups
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.errorState
—Sel.Sel5056.TopologyManager.Synchronization.SynchronizationRequest.errorState
——Sel.Sel5056.TopologyManager.Synchronization.SynchronizationRequest

25.2 Sel.BlueFrame.Core.EventBus.Enums.DurationType

key	value	description
Momentary	0	TODO
Persistent	1	TODO

Table 25.2: Sel.BlueFrame.Core.EventBus.Enums.DurationType

Relationship between this type and other types:
Sel.BlueFrame.Core.EventBus.Enums.DurationType
—Sel.BlueFrame.Core.EventBus.EventType.durationSetting
——Sel.BlueFrame.Core.EventBus.EventType

25.3 Sel.BlueFrame.Core.EventBus.Enums.FacilityCode

key	value	description
ClockDaemon1	9	TODO
ClockDaemon2	15	TODO
FtpDaemon	11	TODO
InternalMessages	5	TODO
KernelMessages	0	TODO
LinePrinterSubsystem	6	TODO
LocalUse0	16	TODO
LocalUse1	17	TODO
LocalUse2	18	TODO
LocalUse3	19	TODO
LocalUse4	20	TODO
LocalUse5	21	TODO
LocalUse6	22	TODO
LocalUse7	23	TODO
LogAlert	14	TODO
LogAudit	13	TODO
MailSystem	2	TODO
NetworkNewsSubsystem	7	TODO

NtpSubsystem	12	TODO
SecurityOrAuthorizationMessages1	4	TODO
SecurityOrAuthorizationMessages2	10	TODO
SystemDaemons	3	TODO
UserLevelMessages	1	TODO
UucpSubsystem	8	TODO

Table 25.3: Sel.BlueFrame.Core.EventBus.Enums.FacilityCode

Relationship between this type and other types:
Sel.BlueFrame.Core.EventBus.Enums.FacilityCode
—Sel.BlueFrame.Core.EventBus.EventType.facility
——Sel.BlueFrame.Core.EventBus.EventType

25.4 Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel

key	value	description
Alert	1	TODO
Critical	2	TODO
Debug	7	TODO
Emergency	0	TODO
Error	3	TODO
Informational	6	TODO
Notice	5	TODO
Warning	4	TODO

Table 25.4: Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel

Relationship between this type and other types:
Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel
—Sel.BlueFrame.Core.EventBus.Behaviors.Behavior.severity
——Sel.BlueFrame.Core.EventBus.EventCategory.behaviors
———Sel.BlueFrame.Core.EventBus.EventCategory
——Sel.BlueFrame.Core.EventBus.EventCategory.behaviors
———Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode.eventCategories
————Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode
—Sel.BlueFrame.Core.EventBus.EventType.severity
——Sel.BlueFrame.Core.EventBus.EventType
—Sel.Sel5056.SystemMessaging.SystemMessage.severity
——Sel.Sel5056.SystemMessaging.SystemMessage
—Sel.BlueFrame.Core.EventBus.Behaviors.Behavior.severity
——Sel.BlueFrame.Core.EventBus.EventCategory.behaviors
———Sel.BlueFrame.Core.EventBus.EventCategory
——Sel.BlueFrame.Core.EventBus.EventCategory.behaviors
———Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode.eventCategories

————Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

25.5 Sel.BlueFrame.Core.TrustAuthority.Enums.CertificateStatus

key	value	description
Expired	2	TODO
Revoked	1	TODO
Valid	0	TODO

Table 25.5: Sel.BlueFrame.Core.TrustAuthority.Enums.CertificateStatus

Relationship between this type and other types:
Sel.BlueFrame.Core.TrustAuthority.Enums.CertificateStatus
—Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation.status
——Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation

25.6 Sel.BlueFrame.Core.TrustAuthority.Enums.UserInteraction

key	value	description
NotUserUploaded	0	TODO
UserUploaded	1	TODO

Table 25.6: Sel.BlueFrame.Core.TrustAuthority.Enums.UserInteraction

Relationship between this type and other types:
Sel.BlueFrame.Core.TrustAuthority.Enums.UserInteraction
—Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation.userInteraction
——Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation

25.7 Sel.BlueFrame.EventBus.Syslog.RFCType

key	value	description
Rfc3164	0	TODO
Rfc5424	1	TODO

Table 25.7: Sel.BlueFrame.EventBus.Syslog.RFCType

Relationship between this type and other types:
Sel.BlueFrame.EventBus.Syslog.RFCType
—Sel.BlueFrame.EventBus.Syslog.SyslogBehavior.rfcType
——Sel.BlueFrame.Core.EventBus.EventCategory.behaviors

———Sel.BlueFrame.Core.EventBus.EventCategory
——Sel.BlueFrame.Core.EventBus.EventCategory.behaviors
———Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode.eventCategories
————Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode
—Sel.BlueFrame.EventBus.Syslog.SyslogBehavior.rfcType

25.8 Sel.BlueFrame.EventBus.Syslog.TransportType

key	value	description
SecureTcp	1	TODO
Udp	0	TODO

Table 25.8: Sel.BlueFrame.EventBus.Syslog.TransportType

Relationship between this type and other types:
Sel.BlueFrame.EventBus.Syslog.TransportType
—Sel.BlueFrame.EventBus.Syslog.SyslogBehavior.transportType
——Sel.BlueFrame.Core.EventBus.EventCategory.behaviors
———Sel.BlueFrame.Core.EventBus.EventCategory
——Sel.BlueFrame.Core.EventBus.EventCategory.behaviors
———Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode.eventCategories
————Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode
—Sel.BlueFrame.EventBus.Syslog.SyslogBehavior.transportType

25.9 Sel.BlueFrame.SecurityManager.AuthServiceResultState

key	value	description
Error	2	TODO
Info	0	TODO
Success	3	TODO
Warning	1	TODO

Table 25.9: Sel.BlueFrame.SecurityManager.AuthServiceResultState

Relationship between this type and other types:
Sel.BlueFrame.SecurityManager.AuthServiceResultState
—Sel.BlueFrame.SecurityManager.AuthServiceResult.state
——Sel.BlueFrame.SecurityManager.AuthServiceResultCollection.results

25.10 Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.NodesConfiguration

key	value	description
All	0	TODO
Only2740S	1	TODO

Table 25.10: Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.NodesConfiguration

Relationship between this type and other types:
Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.NodesConfiguration
—Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions.nodesSetting

25.11 Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.OutputConfiguration

key	value	description
Controller	0	TODO
Local	1	TODO
NextTable	2	TODO

Table 25.11: Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.OutputConfiguration

Relationship between this type and other types:
Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.OutputConfiguration
—Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions.outputSetting

25.12 Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.TableConfiguration

key	value	description
AllTables	1	TODO
SpecificTable	0	TODO

Table 25.12: Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.TableConfiguration

Relationship between this type and other types:
Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.TableConfiguration
—Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions.tableSetting

25.13 Sel.Sel5056.LearnAndLock.Enums.EndReason

key	value	description
Automatic	1	TODO
ByUser	2	TODO
ForReason	3	TODO
NA	0	TODO
Timeout	4	TODO

Table 25.13: Sel.Sel5056.LearnAndLock.Enums.EndReason

Relationship between this type and other types:
Sel.Sel5056.LearnAndLock.Enums.EndReason
—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.autoAdoptionEndReason
——Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession
—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.logicalConnectionLearningEndReason
——Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

25.14 Sel.Sel5056.LearnAndLock.LearnAndLockStateStatus

key	value	description
AutoAdoptDone	3	TODO
AutoAdoptNotStarted	1	TODO
AutoAdoptRunning	2	TODO
Failure	11	TODO
LearnAndLockDone	9	TODO
LearnAndLockNotStarted	7	TODO
LearnAndLockRunning	8	TODO
LogicalConnectionLearningDone	6	TODO
LogicalConnectionLearningNotStarted	4	TODO
LogicalConnectionLearningRunning	5	TODO
None	0	TODO
Success	10	TODO

Table 25.14: Sel.Sel5056.LearnAndLock.LearnAndLockStateStatus

Relationship between this type and other types:
Sel.Sel5056.LearnAndLock.LearnAndLockStateStatus
—Sel.Sel5056.LearnAndLock.Sessions.AutoAdoptSession.status
——Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.autoAdoptSession
———Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession
—Sel.Sel5056.LearnAndLock.Sessions.LogicalConnectionLearningSession.status
——Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.logicalConnectionLearningSession
———Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession
—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.stateStatus

——Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

25.15 Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.-AutoAcceptStatus

key	value	description
All	0	TODO
ArpOnly	1	TODO
None	2	TODO

Table 25.15: Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.AutoAcceptStatus

Relationship between this type and other types:
Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.AutoAcceptStatus
—Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.Configuration.-LogicalConnectionLearningConfig.autoAcceptStatus
——Sel.BeginLearnAndLock.logicalConnectionLearningConfig
——Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.logicalConnectionLearningConfig
———Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

25.16 Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.-LearnedLogicalConnectionState

key	value	description
Accepted	2	TODO
Cleared	6	TODO
Declined	5	TODO
Disregarded	4	TODO
Failed	7	TODO
Filtered	3	TODO
None	0	TODO
Programmed	1	TODO
Proposed	8	TODO

Table 25.16: Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnectionState

Relationship between this type and other types:
Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnectionState
—Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection.state
——Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection

25.17 Sel.Sel5056.LogicalConnectionValidation.Enums.CstCommunicationType

key	value	description
ControllerRoute	4	IB Management LC
Multicast	2	TODO
Undefined	0	Not used
Unicast	1	TODO
UnicastBidirectional	3	TODO

Table 25.17: Sel.Sel5056.LogicalConnectionValidation.Enums.CstCommunicationType

Relationship between this type and other types:
Sel.Sel5056.LogicalConnectionValidation.Enums.CstCommunicationType
—Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType.cstCommunicationType
——Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType
—Sel.Sel5056.LearnAndLock.ProposedCommunicationServiceType.cstCommunicationType
——Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection.proposedCommunicationServiceType
———Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection

25.18 Sel.Sel5056.LogicalConnectionValidation.Enums.LogicalConnectionOrigin

key	value	description
ControllerSwitchCommunication	2	TODO
EndUser	0	TODO
L2Learning	1	TODO

Table 25.18: Sel.Sel5056.LogicalConnectionValidation.Enums.LogicalConnectionOrigin

Relationship between this type and other types:
Sel.Sel5056.LogicalConnectionValidation.Enums.LogicalConnectionOrigin
—Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection.connectionOrigin
——Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection

25.19 Sel.Sel5056.OpenFlowPlugin.Enums.OF13MeterBandType

key	value	description
OFPMBT_DROP	1	TODO

OFPMBT_DSCP_REMARK	2	TODO
OFPMBT_EXPERIMENTER	65535	TODO

Table 25.19: Sel.Sel5056.OpenFlowPlugin.Enums.OF13MeterBandType

Relationship between this type and other types:
Sel.Sel5056.OpenFlowPlugin.Enums.OF13MeterBandType
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandHeader.type
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDesc.bands
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions.meterDescriptionsList
————Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures.bandTypes
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures

25.20 Sel.Sel5056.OpenFlowPlugin.Enums.OfpGroupType

key	value	description
All	0	TODO
FastFailover	3	TODO
Indirect	2	TODO
Select	1	TODO

Table 25.20: Sel.Sel5056.OpenFlowPlugin.Enums.OfpGroupType

Relationship between this type and other types:
Sel.Sel5056.OpenFlowPlugin.Enums.OfpGroupType
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.groupType
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group
—Sel.Sel5056.PathPlanner.RestController.GroupReferences.groupType
——Sel.Sel5056.PathPlanner.RestController.GroupWithStats.group
———Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.groupsReferences
——Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.groupsReferences
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.groupType
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc.groups
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc

25.21 Sel.Sel5056.SapphirePlugin.Synchronization.SapphireEndpoint

key	value	description
AlarmContact	8	TODO
ControllerIp	9	TODO
Logging	1	TODO

Networking	2	TODO
None	0	TODO
NtpGlobal	7	TODO
Ptp	4	TODO
Snmp	3	TODO
TrustedCertificateAdd	5	TODO
TrustedCertificateRevoke	6	TODO

Table 25.21: Sel.Sel5056.SapphirePlugin.Synchronization.SapphireEndpoint

Relationship between this type and other types:
Sel.Sel5056.SapphirePlugin.Synchronization.SapphireEndpoint
—Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization.endpoint
——Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization

25.22 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.DeviceFramework

key	value	description
BlueFrame	2	TODO
Omni	1	TODO
Unknown	0	TODO

Table 25.22: Sel.Sel5056.TopologyManager.Attributes.Operational.Node.DeviceFramework

Relationship between this type and other types:
Sel.Sel5056.TopologyManager.Attributes.Operational.Node.DeviceFramework
—Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr.deviceFramework
——Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr.discoveryPortAttribute
——Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.attributes
———Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
——Sel.GetLinkByPortAttribute.attribute
——Sel.GetNodeByAttribute.attribute
——Sel.GetPortByAttribute.attribute
——Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.attributes
———Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
——Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.attributes
———Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

25.23 Sel.Sel5056.TopologyManager.Enums.CommunicationBand

key	value	description
InBand	0	TODO
OutOfBand	1	TODO

Table 25.23: Sel.Sel5056.TopologyManager.Enums.CommunicationBand

Relationship between this type and other types:
Sel.Sel5056.TopologyManager.Enums.CommunicationBand
—Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr.communicationBand
——Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr.discoveryPortAttribute
——Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.attributes
———Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
——Sel.GetLinkByPortAttribute.attribute
——Sel.GetNodeByAttribute.attribute
——Sel.GetPortByAttribute.attribute
——Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.attributes
———Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
——Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.attributes
———Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

25.24 Sel.Sel5056.TopologyManager.Enums.DiscoveryTrustState

key	value	description
Certificate	1	TODO
Internal	2	TODO
Untrusted	0	TODO
User	3	TODO

Table 25.24: Sel.Sel5056.TopologyManager.Enums.DiscoveryTrustState

Relationship between this type and other types:
Sel.Sel5056.TopologyManager.Enums.DiscoveryTrustState
—Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.trustState
——Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
—Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.trustState
——Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
—Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.trustState
——Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

25.25 Sel.Sel5056.TopologyManager.Enums.OfpPortStatus

key	value	description
Blocked	2	TODO
LinkDown	1	TODO
Live	4	TODO
None	0	TODO

Table 25.25: Sel.Sel5056.TopologyManager.Enums.OfpPortStatus

Relationship between this type and other types:
Sel.Sel5056.TopologyManager.Enums.OfpPortStatus
—Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr.ofState
——Sel.Sel5056.TopologyManager.Attributes.Operational.NetworkingAbstractionAttr.mappedOpenFlowPortAttr
———Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr.discoveryPortAttribute
———Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.attributes
————Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
———Sel.GetLinkByPortAttribute.attribute
———Sel.GetNodeByAttribute.attribute
———Sel.GetPortByAttribute.attribute
———Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.attributes
————Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
———Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.attributes
————Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink
—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port.state
——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc.ports
———Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc

25.26 Sel.Sel5056.TopologyManager.Enums.PacketSource

key	value	description
Controller	0	TODO
OpenFlow	1	TODO

Table 25.26: Sel.Sel5056.TopologyManager.Enums.PacketSource

Relationship between this type and other types:
Sel.Sel5056.TopologyManager.Enums.PacketSource
—Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr.discoverySource
——Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr.discoveryPortAttribute
——Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.attributes
———Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
——Sel.GetLinkByPortAttribute.attribute
——Sel.GetNodeByAttribute.attribute
——Sel.GetPortByAttribute.attribute

——Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.attributes
———Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
——Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.attributes
———Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

25.27 Sel.Sel5056.TopologyManager.Enums.State

key	value	description
Adopted	1	TODO
Configured	5	TODO
Disconnected	3	TODO
Established	4	TODO
None	0	TODO
Unadopted	2	TODO

Table 25.27: Sel.Sel5056.TopologyManager.Enums.State

Relationship between this type and other types:
Sel.Sel5056.TopologyManager.Enums.State
—Sel.Sel5056.TopologyManager.Nodes.ConfigNode.state
——Sel.Sel5056.TopologyManager.Nodes.ConfigNode
—Sel.Sel5056.TopologyManager.Ports.ConfigPort.state
——Sel.Sel5056.TopologyManager.Ports.ConfigPort
—Sel.Sel5056.TopologyManager.Links.ConfigLink.state
——Sel.Sel5056.TopologyManager.Links.ConfigLink
—Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.state
——Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
—Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.state
——Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
—Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.state
——Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

25.28 System.Diagnostics.EventLogEntryType

key	value	description
Error	1	TODO
FailureAudit	16	TODO
Information	4	TODO
SuccessAudit	8	TODO
Warning	2	TODO

Table 25.28: System.Diagnostics.EventLogEntryType

Relationship between this type and other types:
System.Diagnostics.EventLogEntryType

—Sel.BlueFrame.EventBus.WindowsEventLog.WindowsEventLogBehavior.emergencyMap
——Sel.BlueFrame.Core.EventBus.EventCategory.behaviors
———Sel.BlueFrame.Core.EventBus.EventCategory
——Sel.BlueFrame.Core.EventBus.EventCategory.behaviors
———Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode.eventCategories
————Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode
—Sel.BlueFrame.EventBus.WindowsEventLog.WindowsEventLogBehavior.alertMap
—Sel.BlueFrame.EventBus.WindowsEventLog.WindowsEventLogBehavior.criticalMap
—Sel.BlueFrame.EventBus.WindowsEventLog.WindowsEventLogBehavior.errorMap
—Sel.BlueFrame.EventBus.WindowsEventLog.WindowsEventLogBehavior.warningMap
—Sel.BlueFrame.EventBus.WindowsEventLog.WindowsEventLogBehavior.noticeMap
—Sel.BlueFrame.EventBus.WindowsEventLog.WindowsEventLogBehavior.informationalMap

25.29 System.DirectoryServices.Protocols.AuthType

key	value	description
Anonymous	0	TODO
Basic	1	TODO
Digest	4	TODO
Dpa	6	TODO
External	8	TODO
Kerberos	9	TODO
Msn	7	TODO
Negotiate	2	TODO
Ntlm	3	TODO
Sicily	5	TODO

Table 25.29: System.DirectoryServices.Protocols.AuthType

Relationship between this type and other types:
System.DirectoryServices.Protocols.AuthType
—Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService.authType
——Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService
——Sel.GetAvailableGroups.bindingParameter
——Sel.TestAuthenticateUser.bindingParameter

Part VI

Appendix

Appendix A

Summary of Changes to this document

This section will be updated once the first version is completed.

Appendix B

Summary of Changes to REST Interface

B.1 v2.2.0.0

Transition to SignalR Core.
Removal of many of the endpoints from the schema that the user could not access.
Removal of actionType and instructionType fields from Actions and Instructions. @odata.type field should be used instead, which works for v2.1 and earlier releases.

B.1.1 Added Endpoints

- api/default/config/LearnAndLockSessions/AcceptAllProposedLearnedLogicalConnections
- api/default/config/LearnAndLockSessions/AcceptLearnedLogicalConnections
- api/default/config/LearnAndLockSessions/AcknowledgeLearnAndLockSession
- api/default/operational/ports('id')/AddOperationalLinkFromHostToSwitch
- api/default/operational/ports('id')/AddSilentHost
- api/default/config/LearnAndLockSessions/BeginLearnAndLock
- api/default/certificate/transactions
- api/default/config/LearnAndLockSessions/ClearLearnedLogicalConnections
- api/default/config/LearnAndLockSessions/DeclineAllProposedLearnedLogicalConnections
- api/default/config/LearnAndLockSessions/DeclineLearnedLogicalConnections
- api/default/config/logicalConnections/DeleteAll
- api/default/config/LearnAndLockSessions/DeleteLearnAndLockSession
- api/default/operational/nodes('id')/DisableControllerArpToNode
- api/default/operational/nodes('id')/EnableControllerArpToNode
- api/default/config/LearnAndLockSessions/GetHardTimeoutTimeRemaining

- api/default/config/LearnAndLockSessions/GetLearnAndLockCounters
- api/default/config/LearnAndLockSessions/GetStateBeginEndTimestamps
- api/default/config/LearnAndLockSessions/GetStatusMessages
- api/default/config/LearnAndLockSessions/LearnAndLockIndividualSessionExportCSV
- api/default/operational/ports('id')/MarkPortAndAddRelayFailoverLink
- api/default/operational/ports('id')/MarkPortAndPerformBlockingDetectSelRelayFailover
- api/default/config/LearnAndLockSessions/ProposeLearnedLogicalConnections
- api/default/config/logicalConnections/ReplanAll
- api/default/settings/learnAndLockSettings
- api/default/config/LearnedLogicalConnections
- api/default/config/LearnAndLockSessions
- api/default/config/LearnAndLockSessions/SetDone
- api/default/config/LearnAndLockSessions/StopLearnAndLock
- api/default/operational/UserInitiatedHostDiscovery

B.1.2 Removed Endpoints

- api/default/security/authCodes
- api/default/security/consents
- api/default/security/oauthClients
- api/default/security/refreshTokens
- api/default/settings/commissioningState
- api/default/settings/monotonicEventId
- api/default/certificate/properties
- api/default/certificate/transactions
- api/default/settings/RestUriContainer
- api/default/certificate/privateKey
- api/default/security/Commission
- api/default/config/allOpenFlowNodesMatch
- api/default/settings/licensing
- api/default/config/policies
- api/default/settings/UpdateLicense

B.1.3 Added Types

- Sel.AcceptAllProposedLearnedLogicalConnections
- Sel.AcceptLearnedLogicalConnections
- Sel.AcknowledgeLearnAndLockSession
- Sel.AddOperationalLinkFromHostToSwitch
- Sel.AddSilentHost
- Sel.BeginLearnAndLock
- Sel.BlueFrame.Core.Networking.NetworkInterfaceAddress
- Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction
- Sel.ClearLearnedLogicalConnections
- Sel.DeclineAllProposedLearnedLogicalConnections
- Sel.DeclineLearnedLogicalConnections
- Sel.DeleteAll
- Sel.DeleteLearnAndLockSession
- Sel.DisableControllerArpToNode
- Sel.EnableControllerArpToNode
- Sel.FileArgument
- Sel.GetHardTimeoutTimeRemaining
- Sel.GetLearnAndLockCounters
- Sel.GetStateBeginEndTimestamps
- Sel.GetStatusMessages
- Sel.LearnAndLockIndividualSessionExportCSV
- Sel.MarkPortAndAddRelayFailoverLink
- Sel.MarkPortAndPerformBlockingDetectSelRelayFailover
- Sel.PathPlanHop
- Sel.ProposeLearnedLogicalConnections
- Sel.ReplanAll
- Sel.Sel5056.LearnAndLock.AutoAdoption.Tags.AutoAdoptionTag
- Sel.Sel5056.LearnAndLock.CSV.AdoptedHostInfo

- Sel.Sel5056.LearnAndLock.CSV.AdoptedLinkInfo
- Sel.Sel5056.LearnAndLock.CSV.AdoptedSwitchInfo
- Sel.Sel5056.LearnAndLock.Configuration.AutoAdoptConfig
- Sel.Sel5056.LearnAndLock.Enums.EndReason
- Sel.Sel5056.LearnAndLock.LearnAndLockSettings
- Sel.Sel5056.LearnAndLock.LearnAndLockStateStatus
- Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.AutoAcceptStatus
- Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.Configuration.LogicalConnectionLearningConfig
- Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection
- Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnectionState
- Sel.Sel5056.LearnAndLock.ProposedCommunicationServiceType
- Sel.Sel5056.LearnAndLock.Sessions.AutoAdoptSession
- Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession
- Sel.Sel5056.LearnAndLock.Sessions.LogicalConnectionLearningSession
- Sel.Sel5056.LearnAndLock.Status.LearnAndLockCounters
- Sel.Sel5056.LearnAndLock.Status.LearnAndLockStateBeginEndTimestamps
- Sel.Sel5056.LogicalConnectionValidation.DataObject.AutoFillSettings
- Sel.Sel5056.LogicalConnectionValidation.Tags.MatchableCst
- Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowData
- Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupData
- Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.FlowModifyDependency
- Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupDeleteDependency
- Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupModifyDependency
- Sel.Sel5056.Tags.MacAddressTag
- Sel.Sel5056.Tags.NoArpTag
- Sel.Sel5056.TopologyManager.Attributes.Operational.Link.UserHintedLinkAttr
- Sel.Sel5056.TopologyManager.Attributes.Operational.Node.NoArpAttr
- Sel.Sel5056.TopologyManager.Attributes.Operational.Node.UserHintedIpAddressAttr

- Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationNodeAttr
- Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedMappedSwitchPortAttr
- Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedNodeAttr
- Sel.Sel5056.TopologyManager.Configurations.Sapphire.AlternateIpConfig
- Sel.SetDone
- Sel.StopLearnAndLock
- Sel.UserInitiatedHostDiscovery

B.1.4 Removed Types

- IdentityServer3.Core.Models.AccessTokenType
- IdentityServer3.Core.Models.Consent
- IdentityServer3.Core.Models.Flows
- IdentityServer3.Core.Models.Secret
- IdentityServer3.Core.Models.TokenExpiration
- IdentityServer3.Core.Models.TokenUsage
- Sel.BlueFrame.AuthBroker.IpClaim
- Sel.BlueFrame.AuthBroker.Models.AuthCode
- Sel.BlueFrame.AuthBroker.Models.ConsentHandle
- Sel.BlueFrame.AuthBroker.Models.OauthClient
- Sel.BlueFrame.AuthBroker.Models.RefreshTokenHandle
- Sel.BlueFrame.CommissioningManager.CommissioningState
- Sel.BlueFrame.EventBus.MonotonicEventIdStore
- Sel.BlueFrame.RestBroker.Models.DataTreeProperties
- Sel.BlueFrame.RestBroker.Models.RestTransaction
- Sel.BlueFrame.RestBroker.UriContainer.RestUriContainer
- Sel.BlueFrame.TrustAuthority.DataTreeObjects.PrivateKeyInformation
- Sel.Commission
- Sel.Sel5056.Common.OpenFlowNodeProgrammer.AllOpenFlowNodesMatch
- Sel.Sel5056.ControllerLicensing.Licensing.LicenseErrorCollection

- Sel.Sel5056.ControllerLicensing.Licensing.LicenseResult
- Sel.Sel5056.ControllerLicensing.LicensingInformation
- Sel.Sel5056.LogicalConnectionValidation.DataObject.CstMatch
- Sel.Sel5056.LogicalConnectionValidation.DataObject.Policy
- Sel.Sel5056.LogicalConnectionValidation.Enums.Directionality
- Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BaseBucket
- Sel.Sel5056.OpenFlowPlugin.Enums.OfpActionType
- Sel.Sel5056.OpenFlowPlugin.Enums.OfpInstructionType
- Sel.Sel5056.PathPlanner.DataObject.PathPlanHop
- Sel.Sel5056.PathPlanner.enumerations.IngressPortActionType
- Sel.UpdateLicense
- System.Collections.Generic.KeyValuePair_2OfString_String
- System.Net.HttpListenerBasicIdentity
- System.Security.Claims.Claim
- System.Security.Claims.ClaimsIdentity
- System.Security.Cryptography.AsymmetricAlgorithm
- System.Security.Cryptography.CngAlgorithm
- System.Security.Cryptography.DSA
- System.Security.Cryptography.DSACng
- System.Security.Cryptography.DSACryptoServiceProvider
- System.Security.Cryptography.ECDiffieHellman
- System.Security.Cryptography.ECDiffieHellmanCng
- System.Security.Cryptography.ECDiffieHellmanKeyDerivationFunction
- System.Security.Cryptography.ECDsa
- System.Security.Cryptography.ECDsaCng
- System.Security.Cryptography.KeySizes
- System.Security.Cryptography.RSA
- System.Security.Cryptography.RSACng

- System.Security.Cryptography.RSACryptoServiceProvider
- System.Security.Cryptography.X509Certificates.X509Certificate2
- System.Security.Principal.GenericIdentity
- System.Security.Principal.IdentityReference
- System.Security.Principal.NTAccount
- System.Security.Principal.SecurityIdentifier
- System.Security.Principal.WindowsIdentity
- System.Web.Security.FormsIdentity

B.1.5 Added Properties

type	property
Sel.BlueFrame.Core.FileArgument	fileName
Sel.BlueFrame.Core.FileReturnValue	contentType
Sel.BlueFrame.Core.FileReturnValue	fileName
Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType	autoFillSettings
Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType	tags
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket	actions
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket	outGroup
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket	outPort
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow	inPort
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow	outGroup
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow	outPort
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat	checkOverlap
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat	noByteCounts
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat	noPacketCounts
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat	resetCounts
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpSpa	translatedValue
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpTpa	translatedValue
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthDst	translatedValue
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthSrc	translatedValue
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.InPort	translatedValue
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Dst	translatedValue
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Src	translatedValue
Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowDeleteSynchronizationRequest	referencedGroupId
Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModifySynchronizationRequest	currentGroupId
Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModifySynchronizationRequest	newGroupId
Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupModifySynchronizationRequest	currentGroupIds
Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupModifySynchronizationRequest	newGroupIds
Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.AmDestination	gettingFirstNodeId
Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.AmDestination	uniqueId

Sel.Sel5056.Southbound.DeviceManagement.CommissioningStateAttr	state
Sel.Sel5056.Tags.GenericArpFlowTag	id
Sel.Sel5056.Tags.InBandTag	id
Sel.Sel5056.Tags.IpAddressTag	id
Sel.Sel5056.Tags.KeyValueTag	id
Sel.Sel5056.Tags.LogicalConnectionTag	id
Sel.Sel5056.TopologyManager.Attributes.Operational.EthernetAttr	id
Sel.Sel5056.TopologyManager.Attributes.Operational.NetworkingAbstractionAttr	mappedSimHostAttr
Sel.Sel5056.TopologyManager.Attributes.Operational.Node.OpenFlowAttr	id
Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr	id
Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationAttr	id
Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr	id
Sel.Sel5056.TopologyManager.Enums.OfpPortStatus	None
Sel.Sel5056.TopologyManager.NetworkSettings	controllerArpSrcNetworkAddresses

Table B.2: Added Properties

B.1.6 Removed Properties

type	property
Sel.BlueFrame.Core.SecurityManager.LocalUser	currentSalt
Sel.BlueFrame.Core.SecurityManager.LocalUser	saltedHashPassword
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Action	actionType
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction	instructionType
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand	type

Table B.4: Removed Properties

B.1.7 Added Property Attributes

type	property	attribute	value
Sel.BlueFrame.CommissioningManager.UserCommissioner.UserCommissioningConcern	password	MaxLength	128
Sel.BlueFrame.CommissioningManager.UserCommissioner.UserCommissioningConcern	password	MinLength	8
Sel.BlueFrame.CommissioningManager.UserCommissioner.UserCommissioningConcern	username	MaxLength	128
Sel.BlueFrame.CommissioningManager.UserCommissioner.UserCommissioningConcern	username	MinLength	1
Sel.Sel5056.Commissioning.OpenFlowNodeProgrammerOptions.FlowProgrammingOptions	displayName	MaxLength	128
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow	displayName	MaxLength	512
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group	displayName	MaxLength	512
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter	displayName	MaxLength	512
Sel.Sel5056.PathPlanner.RestController.DirectionalLink	egressPortId	ForeignKey	Sel.Sel5056.TopologyManager.Ports.Co
Sel.Sel5056.PathPlanner.RestController.DirectionalLink	ingressPortId	ForeignKey	Sel.Sel5056.TopologyManager.Ports.Co
Sel.Sel5056.PathPlanner.RestController.DirectionalLink	linkId	ForeignKey	Sel.Sel5056.TopologyManager.Links.Co

Table B.6: Added Property Attributes

B.1.8 Removed Property Attributes

type	property	attribute	value
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.MatchField	value	ForeignKey	Sel.Sel5056.LogicalConnectionValidatio

Table B.8: Removed Property Attributes

B.1.9 Removed Property Attributes

type	property	attribute	previous	current
Sel.BlueFrame.ApplicationRegistration.Authentication	grantType	Type	IdentityServer3.Core.Models.Flows	EnumString
Sel.ClearTransaction	bindingParameter	Type	['Sel.BlueFrame.RestBroker.Models.RestTransaction	['Sel.BlueFrame.RestBroker.Oda
Sel.ClearTransaction	bindingParameter	Type	['Sel.BlueFrame.RestBroker.Models.RestTransaction	['Sel.BlueFrame.RestBroker.Oda
Sel.ClearTransaction	bindingParameter	Type	['Sel.BlueFrame.RestBroker.Models.RestTransaction	['Sel.BlueFrame.RestBroker.Oda
Sel.ClearTransaction	bindingParameter	Type	['Sel.BlueFrame.RestBroker.Models.RestTransaction	['Sel.BlueFrame.RestBroker.Oda
Sel.ClearTransaction	bindingParameter	Type	['Sel.BlueFrame.RestBroker.Models.RestTransaction	['Sel.BlueFrame.RestBroker.Oda
Sel.Commit	bindingParameter	Type	Sel.BlueFrame.RestBroker.Models.RestTransaction	Sel.BlueFrame.RestBroker.Odata
Sel.Commit	bindingParameter	Type	Sel.BlueFrame.RestBroker.Models.RestTransaction	Sel.BlueFrame.RestBroker.Odata
Sel.Commit	bindingParameter	Type	Sel.BlueFrame.RestBroker.Models.RestTransaction	Sel.BlueFrame.RestBroker.Odata
Sel.Commit	bindingParameter	Type	Sel.BlueFrame.RestBroker.Models.RestTransaction	Sel.BlueFrame.RestBroker.Odata
Sel.Commit	bindingParameter	Type	Sel.BlueFrame.RestBroker.Models.RestTransaction	Sel.BlueFrame.RestBroker.Odata
Sel.FirmwareUpgrade	fileName	Type	Sel.BlueFrame.Core.FileArgument	Sel.Argument
Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType	displayName	MaxLength	32	512
Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group	buckets	Type	['Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bu	['SEL.Sel5056.OpenFlowPlugin.Base.D
Sel.Sel5056.PathPlanner.DataObject.PathPlanHopCollection	pathPlanHops	Type	['Sel.Sel5056.PathPlanner.DataObject.PathPlanHop']	['Sel.DataObject.PathPlanHop']
Sel.Sel5056.PathPlanner.RestController.GroupReferences	buckets	Type	['Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bu	['SEL.Sel5056.OpenFlowPlugin.Base.D
Sel.Sel5056.TopologyManager.LinkConfig	displayName	MaxLength	32	512
Sel.Sel5056.TopologyManager.NodeConfig	displayName	MaxLength	32	512
Sel.Sel5056.TopologyManager.PortConfig	displayName	MaxLength	32	512

Table B.10: Modified Property Attributes

Index

/api/default/certificate/RevokeCertificateByPurpose
POST, 242
/api/default/certificate/RevokeCertificateByThumbprint
POST, 243
/api/default/certificate/UploadCertificate
POST, 241
/api/default/certificate/certificateInfo('id')/AsPem
POST, 240
/api/default/certificate/certificateInfo('id')/RevokeCertificate
POST, 242
/api/default/certificate/certificateInfo/
GET, 239
/api/default/config/communicationServiceTypes('id')/
DELETE, 180
PATCH, 179
PUT, 179
/api/default/config/communicationServiceTypes/
GET, 175
POST, 176
/api/default/config/flows('id')/
DELETE, 193
PUT, 190
/api/default/config/flows('id')/ResetCounters
POST, 192
/api/default/config/flows/
GET, 187
POST, 188
/api/default/config/groups('id')/
DELETE, 206
PATCH, 203
PUT, 202
/api/default/config/groups/
GET, 197
POST, 198
/api/default/config/links('id')/
DELETE, 159
PATCH, 157
PUT, 156

/api/default/config/links/
GET, 153
POST, 155
/api/default/config/logicalConnections('id')/
DELETE, 170
/api/default/config/logicalConnections('id')/GetLogicalConnectionStats
POST, 173
/api/default/config/logicalConnections('id')/GetOpenFlowInformation
POST, 174
/api/default/config/logicalConnections('id')/GetPathPlanInformation
POST, 174
/api/default/config/logicalConnections('id')/ResubmitLogicalConnection
POST, 171
/api/default/config/logicalConnections/
GET, 166
POST, 166
/api/default/config/logicalConnections/DeleteAll
POST, 170
/api/default/config/logicalConnections/ReplanAll
POST, 169
/api/default/config/meters('id')/
DELETE, 212
PATCH, 211
PUT, 210
/api/default/config/meters/
GET, 208
POST, 209
/api/default/config/nodes('id')/
DELETE, 119
PATCH, 117
PUT, 116
/api/default/config/nodes/
GET, 112
POST, 114
/api/default/config/ports('id')/
DELETE, 144
PATCH, 143
PUT, 142
/api/default/config/ports('id')/SendPortMod
POST, 151
/api/default/config/ports/
GET, 139
POST, 141
/api/default/config/vlanVidReservation('id')/
DELETE, 184
/api/default/config/vlanVidReservation/
GET, 181

/api/default/config/vlanVidReservation/ReserveVlanVidRange
POST, 182
/api/default/operational/GetLinkByPortAttribute
POST, 161
/api/default/operational/GetNodeByAttribute
POST, 123
/api/default/operational/GetNodesByIp
POST, 122
/api/default/operational/GetNodesThatRequireSynchronization
POST, 127
/api/default/operational/GetPortByAttribute
POST, 149
/api/default/operational/MergeNodes
POST, 118
/api/default/operational/UserInitiatedHostDiscovery
POST, 123
/api/default/operational/flowStats/
GET, 194
/api/default/operational/groupDesc/
GET, 206
/api/default/operational/groupStats/
GET, 207
/api/default/operational/links('id')/Adopt
POST, 159
/api/default/operational/links('id')/AdoptWithConfig
POST, 160
/api/default/operational/links('id')/ReplaceConfig
POST, 158
/api/default/operational/links('id')/Unadopt
POST, 160
/api/default/operational/links/
GET, 154
/api/default/operational/meterDescriptions/
GET, 212
/api/default/operational/meterFeatures/
GET, 213
/api/default/operational/meterstats/
GET, 214
/api/default/operational/nodes('id')/Adopt
POST, 121
/api/default/operational/nodes('id')/AdoptWithConfig
POST, 121
/api/default/operational/nodes('id')/DisableControllerArpToNode
POST, 125
/api/default/operational/nodes('id')/Disconnect
POST, 120
/api/default/operational/nodes('id')/EnableControllerArpToNode

POST, 124

/api/default/operational/nodes('id')/FactoryDefaultReset
POST, 134

/api/default/operational/nodes('id')/FirmwareUpgrade
POST, 134

/api/default/operational/nodes('id')/PacketOut
POST, 230

/api/default/operational/nodes('id')/Reboot
POST, 135

/api/default/operational/nodes('id')/ReplaceConfig
POST, 119

/api/default/operational/nodes('id')/ReplanInbandPath
POST, 136, 172

/api/default/operational/nodes('id')/Unadopt
POST, 122

/api/default/operational/nodes/
GET, 113

/api/default/operational/packets/
GET, 229

/api/default/operational/portDesc/
GET, 150

/api/default/operational/portStats/
GET, 152

/api/default/operational/ports('id')/AddAbstractNode
POST, 138

/api/default/operational/ports('id')/AddOperationalLinkFromHostToSwitch
POST, 156

/api/default/operational/ports('id')/AddSilentHost
POST, 125

/api/default/operational/ports('id')/Adopt
POST, 145

/api/default/operational/ports('id')/AdoptWithConfig
POST, 145

/api/default/operational/ports('id')/DetectSelRelayFailoverPort
POST, 146

/api/default/operational/ports('id')/MarkPortAndAddRelayFailoverLink
POST, 148

/api/default/operational/ports('id')/MarkPortAndPerformBlockingDetectSelRelayFailover
POST, 148

/api/default/operational/ports('id')/MarkPortAsSelRelayFailover
POST, 147

/api/default/operational/ports('id')/RemoveSelRelayFailoverFromPort
POST, 149

/api/default/operational/ports('id')/ReplaceConfig
POST, 144

/api/default/operational/ports('id')/Unadopt
POST, 146

/api/default/operational/ports/
GET, 140
/api/default/operational/synchronization/
GET, 127
/api/default/operational/tableStats/
GET, 137
/api/default/security/IsCommissioned
POST, 215
/api/default/security/applicationLinks('id')/
DELETE, 236
/api/default/security/applicationLinks('id')/DisableApplicationLink
POST, 237
/api/default/security/applicationLinks('id')/EnableApplicationLink
POST, 237
/api/default/security/applicationLinks/
GET, 236
POST, 235
/api/default/security/applicationLinks/InitiateApplicationRegistration
POST, 234
/api/default/security/authServiceGroups('id')/
DELETE, 104
PATCH, 103
PUT, 102
/api/default/security/authServiceGroups/
GET, 101
POST, 102
/api/default/security/authServices('id')/
DELETE, 99
PATCH, 98
PUT, 98
/api/default/security/authServices('id')/TestAuthenticateUser
POST, 100
/api/default/security/authServices/
GET, 95
POST, 96
/api/default/security/roles/
GET, 94
/api/default/security/users('id')/
DELETE, 94
PATCH, 91
PUT, 91
/api/default/security/users('id')/UpdatePassword
POST, 93
/api/default/security/users/
GET, 87
POST, 89
/api/default/settings/Backup

POST, 216
/api/default/settings/GetUserPreference
POST, 105
/api/default/settings/Restore
POST, 216
/api/default/settings/SetUserPreference
POST, 106
/api/default/settings/eventCategories('id')/
PATCH, 218
PUT, 219
/api/default/settings/eventCategories/
GET, 217
/api/default/settings/eventTypes/
GET, 220
/api/default/settings/networkSettings('id')/
PATCH, 163
PUT, 163
/api/default/settings/networkSettings/
GET, 162
/api/default/settings/preferences/
GET, 105
/api/default/settings/securityManager('id')/
PATCH, 223
PUT, 224
/api/default/settings/securityManager/
GET, 221
/api/default/settings/strings/
GET, 226
/api/default/settings/systemMessages/
GET, 227
/api/default/settings/webSettings('id')/
PATCH, 225
PUT, 226
/api/default/settings/webSettings/
GET, 222