

REST API Documentation SEL-5056 v2.8.0 on Windows

Schweitzer Engineering Laboratories, Inc.

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	14
5.3	Logging In With a Username and Password	18
5.4	Refreshing the Token	20
5.5	Validating a Token	20
5.6	Logging Out	21
5.7	Creating a Permission Level 3 User	23
5.8	Retrieving the Permission Level 3 User	25
5.9	Creating Configuration Nodes	27
5.10	Adopting Alpha	35
5.11	Adopting Charlie	44
5.12	Adopting Links	48
5.13	Adopting the Hosts Including Using Failover Mode	52
5.14	Creating Logical Connections	57
5.15	Creating CSTs	59
5.16	Adding a Virtual Host	63
5.17	Managing Flow Entries	66

5.18	Managing Group Entries	76
5.19	Managing Meter Entries	83
II	REST Overview	89
6	HTTP	91
6.1	HTTPS	91
6.1.1	Headers	91
6.1.2	Response Codes	91
6.1.3	Common Format of HTTP Methods	93
7	OData	95
7.1	OData Version	95
7.2	Service Roots	95
7.2.1	Common Properties	96
7.2.2	Filtering	96
8	Authentication (OAuth)	99
8.1	Including Tokens in a Request	99
8.2	OAuth and Connect ID Parameters	100
8.2.1	Grant Types	100
8.2.2	Client ID and Client Secret	101
8.2.3	Scope	101
8.2.4	Roles	101
8.3	Endpoints	101
8.3.1	Obtaining a Token	102
8.3.2	Resource Owner Password Credentials	102
8.3.3	Client Credentials	104
8.3.4	Inspecting a token	104
8.3.5	Extending the Expiration Time of a Token	106
8.4	Refreshing Token	106
8.5	Managing Permissions	106
8.6	Troubleshooting	107
8.6.1	Registered Applications	107
III	OData Endpoints	109
9	User Management	113
9.1	User Accounts	113
9.1.1	Read Users	113
9.1.2	Create a User	115
9.1.3	Update a User (PUT)	117
9.1.4	Update a User (PATCH)	118
9.1.5	Change the Current User's Password	120

9.1.6	Delete a User	120
9.2	User Roles	121
9.2.1	Read User Roles	121
9.3	Authentication Services	122
9.3.1	Read Authentication Services	122
9.3.2	Create an Authentication Service	123
9.3.3	Update an Authentication Service (PUT)	124
9.3.4	Update an Authentication Service (PATCH)	125
9.3.5	Delete an Authentication Service	126
9.3.6	Test an Authentication Service	127
9.4	Authentication Service Groups	127
9.4.1	Read Authentication Service Groups	128
9.4.2	Create an Authentication Service Group	128
9.4.3	Update an Authentication Service Group (PUT)	129
9.4.4	Update an Authentication Service Group (PATCH)	130
9.4.5	Delete an Authentication Service Group	131
9.5	Preferences	131
9.5.1	Get the Value of a User Preference	131
9.5.2	Read User Preferences	132
9.5.3	Add or Modify a User Preference	133
10	Topology Management	135
10.1	Nodes	137
10.1.1	Read Configuration Nodes	138
10.1.2	Read Operational Nodes	140
10.1.3	Create a Configuration Node	141
10.1.4	Update a Configuration Node (PUT)	145
10.1.5	Update a Configuration Node (PATCH)	146
10.1.6	Merge Two Nodes by Combining the Ports of One into the Other	149
10.1.7	Replace the Configuration Object for an Adopted Node	149
10.1.8	Delete a Configuration Node	150
10.1.9	Remove or Disconnect a Node Entry	150
10.1.10	Adopt a Node With Default Settings	151
10.1.11	Adopt a Node With a Preconfigured Node Object	152
10.1.12	Unadopt a Node	152
10.1.13	Run Host Discovery for a Particular IP Address	153
10.1.14	Allow the SEL-5056 to Send ARPs to a Node	153
10.1.15	Prevent the SEL-5056 From Sending ARPs to a Node	154
10.2	OpenFlow Switches	155
10.2.1	Add a Virtual Host to a Switch Port	155
10.2.2	Read Flow Table Diagnostics	156
10.2.3	Get Nodes that Are Unsynchronized	156
10.2.4	Read Switches to Synchronize	157
10.2.5	Synchronize OpenFlow Switch	164
10.2.6	Synchronize All OpenFlow Switches	165

10.2.7	Synchronize OpenFlow Switch(es)	165
10.2.8	Check Synchronization Status of Switch	166
10.3	SEL-2740S and SEL-2742S Switches	166
10.3.1	Factory-Default Reset an SEL-2740S	167
10.3.2	Firmware Upgrade an SEL-2740S	167
10.3.3	Restart an SEL-2740S or SEL-2742S	169
10.3.4	Replan a Switch IB Management Logical Connection	170
10.3.5	Read SEL-2740S and SEL-2742S SNMP Users	170
10.3.6	Update a SEL-2740S and SEL-2742S SNMP User (PATCH)	171
10.4	Traditional Switches	171
10.4.1	Add a Traditional Switch	172
10.5	Ports	172
10.5.1	Read Configuration Ports	173
10.5.2	Read Operational Ports	174
10.5.3	Create a Configuration Port	175
10.5.4	Update a Configuration Port (PUT)	176
10.5.5	Update a Configuration Port (PATCH)	177
10.5.6	Replace the Configuration Object for an Adopted Port	178
10.5.7	Delete a Configuration Port	178
10.5.8	Get List of Nodes that are Unsynchronized	179
10.5.9	Adopt a Port With a Preconfigured Port Object	179
10.5.10	Unadopt a Port	180
10.5.11	Detect the Other Port of a Device in Failover Mode	180
10.5.12	Add SEL Relay Failover Mode to a Port	181
10.5.13	Add SEL Relay Failover Mode to a Port and Create an Adopted Operational Link From a Node to a Switch	182
10.5.14	Detect the Other Port of a Device in Failover Mode and Block Until the Link Is Discovered or Timed Out	182
10.5.15	Remove SEL Relay Failover Mode from a Port	183
10.6	OpenFlow Ports	184
10.6.1	Read OpenFlow Port Descriptions	184
10.6.2	Read OpenFlow Port Diagnostics	185
10.6.3	Modify the OpenFlow Settings of a Port	186
10.6.4	Modify the OpenFlow Settings of a Port	186
10.7	Links	187
10.7.1	Read Configuration Links	188
10.7.2	Read Operational Links	188
10.7.3	Create a Configuration Link	189
10.7.4	Create an Adopted Operational Link From a Node to a Switch	190
10.7.5	Update a Configuration Link (PUT)	191
10.7.6	Update a Configuration Link (PATCH)	192
10.7.7	Replace the Configuration Object for a Link	193
10.7.8	Delete a Configuration Link	193
10.7.9	Adopt a Link with Default Settings	194
10.7.10	Adopt a Link With a Predefined Link Object	194

10.7.11 Unadopt a Link	195
10.8 Default Adoption Settings	195
10.8.1 Read Default Adoption Settings and Default ARP SPA Addresses	195
10.8.2 Update Default Adoption Settings and Default ARP SPA Addresses (PUT)	196
10.8.3 Update Default Adoption Settings and Default ARP SPA Addresses (PATCH)	197
11 Logical Programming	199
11.1 Logical Connections	199
11.1.1 Read Logical Connections	200
11.1.2 Create a Logical Connection	201
11.1.3 Update a Logical Connection (PATCH)	203
11.1.4 Update a Logical Connection (PUT)	204
11.1.5 Delete a Logical Connection	205
11.1.6 Delete all Logical Connections	206
11.1.7 Resubmit a Logical Connection	206
11.1.8 Replan a Switch IB Management Logical Connection	207
11.1.9 Replan all Logical Connections	208
11.1.10 Get the OpenFlow Counters for a Logical Connection	209
11.1.11 Get the OpenFlow Entries for a Logical Connection	209
11.1.12 Get the Path Plan Information for a Logical Connection	210
11.2 Communication Service Types	211
11.2.1 Read Communication Service Types	211
11.2.2 Create a Communication Service Type	212
11.2.3 Update a Communication Service Type (PUT)	216
11.2.4 Update a Communication Service Type (PATCH)	217
11.2.5 Delete a Communication Service Type	218
11.3 VID Reservation	218
11.3.1 Read Reserved VIDs	219
11.3.2 Reserve a VID or a Range of VIDs	220
11.3.3 Delete a Reserved VID	222
12 OpenFlow Programming	223
12.1 Flow Entries	223
12.1.1 Read Flow Entries	225
12.1.2 Create a Flow Entry	226
12.1.3 Update a Flow Entry (PUT)	230
12.1.4 Update a Flow Entry (PATCH)	230
12.1.5 Reset the Counters for a Flow Entry	233
12.1.6 Delete a Flow Entry	234
12.1.7 Read Flow Entry Diagnostics	234
12.2 Group Entries	235
12.2.1 Read Group Entries	236
12.2.2 Create a Group Entry	237
12.2.3 Update a Group Entry (PUT)	243
12.2.4 Update a Group Entry (PATCH)	247

12.2.5	Delete a Group Entry	250
12.2.6	Read Group Entry Descriptions	251
12.2.7	Read Group Entry Diagnostics	251
12.3	Meter Entries	252
12.3.1	Read Meter Entries	253
12.3.2	Create a Meter Entry	254
12.3.3	Update a Meter Entry (PUT)	255
12.3.4	Update a Meter Entry (PATCH)	256
12.3.5	Delete a Meter Entry	257
12.3.6	Read Meter Descriptions	257
12.3.7	Read Meter Features	258
12.3.8	Read Meter Diagnostics	259
13	Controller Management	261
13.1	Commissioning	261
13.1.1	Commissioned Status of SEL-5056	261
13.1.2	Commission the SEL-5056	262
13.2	Database Management	262
13.2.1	Export the SEL-5056 Database	262
13.2.2	Import an SEL-5056 Database	263
13.3	Event Categories	264
13.3.1	Read Event Categories	264
13.3.2	Update an Event Category (PATCH)	265
13.3.3	Update an Event Category (PUT)	266
13.4	Event Types	266
13.4.1	Read Event Types	267
13.4.2	Update an Event Type (PUT)	268
13.4.3	Update an Event Type (PATCH)	268
13.5	Login and Password Management	269
13.5.1	Read Password and Login Management	269
13.5.2	Update Password and Login Management (PATCH)	270
13.5.3	Update Password and Login Management (PUT)	271
13.6	Usage Policy and Session Management	272
13.6.1	Read Usage Policy and Session Management	272
13.6.2	Update Usage Policy and Session Management (PATCH)	273
13.6.3	Update Usage Policy and Session Management (PUT)	274
13.7	Strings	274
13.7.1	Read Localization Strings	275
14	Packet In/Out	277
14.1	Packet In/Out Management	277
14.1.1	Read Packets Sent to the Controller Using the OpenFlow Controller Port	277
14.1.2	Inject a Packet Out an OpenFlow Port	278
14.1.3	Inject a Packet into an OpenFlow Switch With the InPort Set	279

15 Registered Applications	281
15.1 Registered Application Management	281
15.1.1 Request Application Registration Information From an External URL	282
15.1.2 Create a Registered Application	283
15.1.3 Delete a Registered Application	283
15.1.4 Read Registered Applications	284
15.1.5 Enable an Application	285
15.1.6 Disable an Application	285
16 X.509 Certificates	287
16.1 X.509 Certificate Management	287
16.1.1 Read X.509 Certificates	288
16.1.2 Export a Certificate in the PEM Format	289
16.1.3 Upload an X.509 Certificate	290
16.1.4 Delete a X.509 Certificate	291
16.1.5 Revoke an X.509 Certificate	291
16.1.6 Revoke a X.509 Certificate by its Purpose	293
16.1.7 Revoke an X.509 Certificate by its Thumbprint	294
16.1.8 Unrevoke a Revoke Certificate	295
16.2 Certificate Revocation List	296
16.2.1 Read Certificate Revocation List (CRL) Settings	297
16.2.2 Update a Certificate Revocation List (CRL) Setting (PATCH)	297
16.2.3 Update a Certificate Revocation List (CRL) Setting (PUT)	298
IV Additional Topics	301
17 Situational Awareness, Diagnostics, and Counters	305
17.1 Counters	305
17.1.1 Direct (OpenFlow) Counters	305
17.1.2 Indirect Counters	306
17.2 Diagnostics	306
17.2.1 OpenFlow	306
17.3 Situational Awareness	306
17.3.1 Logical Connections	306
18 Asynchronous Data Exchange With SignalR	307
18.1 Hubs	307
18.2 Targets	307
18.3 Authorization	308
18.4 Permissions	308
18.5 WebSocket	308
18.5.1 Creating a WebSocket Connection	308
18.5.2 Subscribing to all Flow Entry Changes by Using WebSocket	309
18.5.3 Subscribing to a Specific Flow Entry Change by Using WebSocket	310

19 Permissions and Roles	311
V OData Definitions	313
20 Native OData Types	317
21 Primary OData Types	319
21.1 Sel.BlueFrame.ApplicationRegistration.ApplicationLink	319
21.2 Sel.BlueFrame.Core.EventBus.EventCategory	320
21.3 Sel.BlueFrame.Core.EventBus.EventType	320
21.3.1 Sel.BlueFrame.Core.EventBus.ClearedEventType	320
21.4 Sel.BlueFrame.Core.EventBus.LocaleString	321
21.5 Sel.BlueFrame.Core.SecurityManager.AuthService	321
21.5.1 Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService	321
21.6 Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup	322
21.7 Sel.BlueFrame.Core.SecurityManager.Role	323
21.8 Sel.BlueFrame.Core.SecurityManager.User	323
21.8.1 Sel.BlueFrame.Core.SecurityManager.LocalUser	324
21.8.2 Sel.BlueFrame.Core.SecurityManager.AuthServiceUser	324
21.9 Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction	324
21.10 Sel.BlueFrame.RestBroker.Plumbing.RestSettings	325
21.11 Sel.BlueFrame.SecurityManager.SecurityManagerSettings	325
21.12 Sel.BlueFrame.TrustAuthority.CrlSettings	326
21.13 Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation	327
21.14 Sel.BlueFrame.UserPreferences.Preference	327
21.15 Sel.Sel5056.LearnAndLock.LearnAndLockSettings	328
21.16 Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection	328
21.17 Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession	329
21.18 Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType	330
21.18.1 Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType- PreDefined	331
21.19 Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection	331
21.20 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow	332
21.21 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats	333
21.22 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group	333
21.23 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc	334
21.24 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats	334
21.25 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter	335
21.26 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions	335
21.27 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures	336
21.28 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats	336
21.29 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc	337
21.30 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats	337
21.31 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ReceivedPacket	338

21.32	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats	338
21.33	Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation	338
21.33.1	Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidDestinationReservation	339
21.34	Sel.Sel5056.Snmp.SnmpUser	339
21.35	Sel.Sel5056.SystemMessaging.SystemMessage	340
21.36	Sel.Sel5056.TopologyManager.Links.ConfigLink	340
21.37	Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink	341
21.38	Sel.Sel5056.TopologyManager.NetworkSettings	341
21.39	Sel.Sel5056.TopologyManager.Nodes.ConfigNode	342
21.39.1	Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode	343
21.39.2	Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode	343
21.40	Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode	344
21.41	Sel.Sel5056.TopologyManager.Ports.ConfigPort	344
21.42	Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort	345
21.43	Sel.Sel5056.TopologyManager.Synchronization.SynchronizationRequest	346
21.43.1	Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowAdd-SynchronizationRequest	346
21.43.2	Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowDelete-SynchronizationRequest	347
21.43.3	Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModify-SynchronizationRequest	347
21.43.4	Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupAdd-SynchronizationRequest	347
21.43.5	Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupDelete-SynchronizationRequest	348
21.43.6	Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupModify-SynchronizationRequest	348
21.43.7	Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterAdd-SynchronizationRequest	349
21.43.8	Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterDelete-SynchronizationRequest	349
21.43.9	Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterModify-SynchronizationRequest	349
21.43.10	Sel.Sel5056.OpenFlowPlugin.PortModController.PortModSynchronization	350
21.43.11	Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization	350
21.43.12	Sel.Sel5056.SapphirePlugin.Synchronization.FirmwareUpgradeRequired-Synchronization	350
22	Secondary OData Types	351
22.1	Sel.BlueFrame.ApplicationRegistration.AuthenticationType	351
22.2	Sel.BlueFrame.CommissioningManager.CommissioningConcern	351
22.2.1	Sel.BlueFrame.CommissioningManager.UserCommissioner.UserCommissioning-Concern	352
22.3	Sel.BlueFrame.Core.EventBus.Behaviors.Behavior	352
22.3.1	Sel.BlueFrame.EventBus.Syslog.SyslogBehavior	352

22.3.2	Sel.BlueFrame.EventBus.WindowsEventLog.WindowsEventLogBehavior	353
22.3.3	Sel.BlueFrame.Particle.AlarmContact.AlarmContactBehavior	353
22.3.4	Sel.BlueFrame.Particle.LocalEventStore.LocalEventStoreBehavior	353
22.4	Sel.BlueFrame.Core.EventBus.LocaleRecord	353
22.5	Sel.BlueFrame.Core.FileArgument	354
22.6	Sel.BlueFrame.Core.FileReturnValue	354
22.7	Sel.BlueFrame.Core.Networking.NetworkInterfaceAddress	354
22.8	Sel.BlueFrame.Core.SecurityManager.Password	354
22.9	Sel.BlueFrame.Core.SecurityManager.Permission	355
22.9.1	Sel.BlueFrame.Core.DataBrokerPermission	355
22.9.2	Sel.BlueFrame.Core.SecurityAttributePermission	355
22.9.3	Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission	356
22.9.4	Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission	356
22.10	Sel.BlueFrame.Core.SecurityManager.UserFailedLogin	356
22.11	Sel.BlueFrame.Core.TrustAuthority.CertificatePurpose	356
22.11.1	Sel.BlueFrame.TrustAuthority.TrustedCertificates.TrustedCertificatePurpose	357
22.11.2	Sel.BlueFrame.TrustAuthority.WebCertificateManager.WebCertificatePurpose	357
22.11.3	Sel.BlueFrame.TrustAuthority.InternalCertificateAuthority.InternalCaPurpose	357
22.11.4	Sel.Sel5056.Drivers.OpenFlow.Trust.OpenFlowDeviceCertificate	357
22.11.5	Sel.Sel5056.Drivers.OpenFlow.Trust.OpenFlowServerCertificate	357
22.11.6	Sel.Sel5056.SapphirePlugin.Trust.SapphireDeviceCertificate	357
22.11.7	Sel.Sel5056.SapphirePlugin.Trust.SapphireServerCertificate	357
22.12	Sel.BlueFrame.SecurityManager.AuthServiceGroupDescription	358
22.13	Sel.BlueFrame.SecurityManager.AuthServiceResult	358
22.14	Sel.BlueFrame.SecurityManager.AuthServiceResultCollection	358
22.15	Sel.BlueFrame.SecurityManager.GroupCollection	358
22.16	Sel.BlueFrame.UserPreferences.Setting	359
22.17	Sel.FileArgument	359
22.18	Sel.PathPlanHop	359
22.19	Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions	359
22.20	Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.MeterMetadata	360
22.21	Sel.Sel5056.LearnAndLock.CSV.AdoptedHostInfo	360
22.22	Sel.Sel5056.LearnAndLock.CSV.AdoptedLinkInfo	361
22.23	Sel.Sel5056.LearnAndLock.CSV.AdoptedSwitchInfo	361
22.24	Sel.Sel5056.LearnAndLock.Configuration.AutoAdoptConfig	361
22.25	Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.Configuration.LogicalConnection-LearningConfig	362
22.26	Sel.Sel5056.LearnAndLock.ProposedCommunicationServiceType	363
22.27	Sel.Sel5056.LearnAndLock.Sessions.AutoAdoptSession	363
22.28	Sel.Sel5056.LearnAndLock.Sessions.LogicalConnectionLearningSession	363
22.29	Sel.Sel5056.LearnAndLock.Status.LearnAndLockCounters	364
22.30	Sel.Sel5056.LearnAndLock.Status.LearnAndLockStateBeginEndTimestamps	364
22.31	Sel.Sel5056.LogicalConnectionValidation.DataObject.AutoFillSettings	365
22.32	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Action	365
22.32.1	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetQueueAction	367

22.32.2	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupAction	367
22.32.3	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputAction	367
22.32.4	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PopVlanAction	367
22.32.5	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PushVlanAction	367
22.32.6	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetFieldAction	368
22.32.7	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupActionByAlias	368
22.32.8	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputActionByAlias	368
22.33	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BandStat	368
22.34	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket	368
22.34.1	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupBucketByAlias	369
22.34.2	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortBucketByAlias	369
22.35	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BucketStats	369
22.36	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowData	370
22.37	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat	370
22.38	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupData	371
22.39	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat	371
22.40	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction	372
22.40.1	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ApplyActions	372
22.40.2	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ClearActions	372
22.40.3	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GoToTable	372
22.40.4	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterInstruction	373
22.40.5	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions	373
22.40.6	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterInstructionByAlias	373
22.41	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match	373
22.42	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.MatchField	374
22.42.1	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpOp	377
22.42.2	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpSpa	377
22.42.3	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpTpa	377
22.42.4	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthDst	377
22.42.5	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthSrc	378
22.42.6	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthType	378
22.42.7	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.InPort	378
22.42.8	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.IpProto	378
22.42.9	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Dst	378
22.42.10	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Src	378
22.42.11	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.TcpDst	379
22.42.12	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.TcpSrc	379
22.42.13	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.UdpDst	379
22.42.14	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.UdpSrc	379
22.42.15	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.VlanPcp	379
22.42.16	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.VlanVid	379
22.42.17	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpSpaByAlias	379
22.42.18	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpTpaByAlias	379
22.42.19	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthDstByAlias	380
22.42.20	Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthSrcByAlias	380

22.42.21 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthernetApplication-ByAlias	380
22.42.22 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.InPortByAlias	380
22.42.23 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4DstByAlias	380
22.42.24 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4SrcByAlias	380
22.43 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand	380
22.43.1 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandDrop	381
22.44 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandHeader	381
22.45 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDesc	381
22.46 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStat	382
22.47 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port	382
22.48 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStat	383
22.49 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStat	383
22.50 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency	383
22.50.1 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.FlowDeleteDependency	384
22.50.2 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupDeleteDependency	385
22.50.3 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupDependency	385
22.50.4 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.MeterDependency	385
22.50.5 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.FlowModifyDependency	385
22.50.6 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupModifyDependency	385
22.51 Sel.Sel5056.PathPlanner.DataObject.PathPlanHopCollection	386
22.52 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.Am-Destination	386
22.52.1 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.-RelayFailoverDestination	387
22.52.2 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.-SimpleDestination	387
22.52.3 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.-TraditionalTiePointDestination	387
22.52.4 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.-ProgrammableDestination	387
22.52.5 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.-InbandDestination	388
22.53 Sel.Sel5056.PathPlanner.RestController.DirectionLink	388
22.54 Sel.Sel5056.PathPlanner.RestController.FlowReferences	388
22.55 Sel.Sel5056.PathPlanner.RestController.FlowWithStats	389
22.56 Sel.Sel5056.PathPlanner.RestController.GroupReferences	389
22.57 Sel.Sel5056.PathPlanner.RestController.GroupWithStats	389
22.58 Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats	390
22.59 Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo	390
22.60 Sel.Sel5056.PathPlanner.RestController.PathPlanInfo	390
22.61 Sel.Sel5056.Tags.AmTag	391
22.61.1 Sel.Sel5056.Tags.LogicalConnectionTag	392
22.61.2 Sel.Sel5056.Tags.InBandPortIdTag	392
22.61.3 Sel.Sel5056.Tags.GenericArpFlowTag	392

22.61.4	Sel.Sel5056.Tags.SystemTag	392
22.61.5	Sel.Sel5056.Tags.OpenFlowPortIdTag	392
22.61.6	Sel.Sel5056.Tags.KeyValueTag	393
22.61.7	Sel.Sel5056.Tags.InBandTag	393
22.61.8	Sel.Sel5056.Tags.ControllerTag	393
22.61.9	Sel.Sel5056.Tags.IpAddressTag	393
22.61.10	Sel.Sel5056.Tags.MacAddressTag	393
22.61.11	Sel.Sel5056.Tags.ReadoptTag	394
22.61.12	Sel.Sel5056.Tags.ControllerRouteTag	394
22.61.13	Sel.Sel5056.Tags.NoArpTag	394
22.61.14	Sel.Sel5056.LogicalConnectionValidation.Tags.MatchableCst	394
22.61.15	Sel.Sel5056.LearnAndLock.AutoAdoption.Tags.AutoAdoptionTag	394
22.62	Sel.Sel5056.TopologyManager.Attributes.AmAttribute	394
22.62.1	Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr	395
22.62.2	Sel.Sel5056.TopologyManager.Attributes.Operational.Port.Sel2740SPortAttr	396
22.62.3	Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationAttr	396
22.62.4	Sel.Sel5056.TopologyManager.Attributes.Operational.Port.SelRelayFailoverMode-Attr	396
22.62.5	Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplication-NodeAttr	396
22.62.6	Sel.Sel5056.TopologyManager.Attributes.Operational.Node.OpenFlowAttr	397
22.62.7	Sel.Sel5056.TopologyManager.Attributes.Operational.Node.OpenFlowTable-FeaturesAttr	397
22.62.8	Sel.Sel5056.TopologyManager.Attributes.Operational.Node.IpAttr	397
22.62.9	Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr	397
22.62.10	Sel.Sel5056.TopologyManager.Attributes.Operational.Node.FirmwareUpdateAttr	398
22.62.11	Sel.Sel5056.TopologyManager.Attributes.Operational.Node.UserHintedIpAddress-Attr	399
22.62.12	Sel.Sel5056.TopologyManager.Attributes.Operational.Node.NoArpAttr	399
22.62.13	Sel.Sel5056.TopologyManager.Attributes.Operational.Link.OfdpDiscovered	399
22.62.14	Sel.Sel5056.TopologyManager.Attributes.Operational.Link.OpenFlowVerified	399
22.62.15	Sel.Sel5056.TopologyManager.Attributes.Operational.Link.ArpVerified	399
22.62.16	Sel.Sel5056.TopologyManager.Attributes.Operational.Link.UserHintedLinkAttr	399
22.62.17	Sel.Sel5056.TopologyManager.Attributes.Operational.ControllerAttr	399
22.62.18	Sel.Sel5056.TopologyManager.Attributes.Operational.EthernetAttr	400
22.62.19	Sel.Sel5056.TopologyManager.Attributes.Operational.DoNotExpireAttr	400
22.62.20	Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedMappedSwitch-PortAttr	400
22.62.21	Sel.Sel5056.TopologyManager.Attributes.Operational.NetworkingAbstractionAttr	400
22.62.22	Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedNodeAttr	400
22.62.23	Sel.Sel5056.Southbound.DeviceManagement.CommissioningStateAttr	401
22.63	Sel.Sel5056.TopologyManager.Attributes.Operational.Node.TableFeature	401
22.64	Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationProvider-Information	401
22.65	Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractSettingBase	402

22.65.1	Sel.Sel5056.TopologyManager.Configurations.Abstraction.TraditionalSwitch	403
22.66	Sel.Sel5056.TopologyManager.Configurations.AmConfig	403
22.66.1	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.ConfigBase .	404
22.66.2	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.FeatureBase	404
22.66.3	Sel.Sel5056.TopologyManager.Configurations.ControllerConfig	404
22.66.4	Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriority	404
22.66.5	Sel.Sel5056.TopologyManager.Configurations.OpenFlow.OpenFlowConfig	404
22.66.6	Sel.Sel5056.TopologyManager.Configurations.FailOver.SelRelayFailoverModeConfig	404
22.66.7	Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractionConfig	404
22.66.8	Sel.Sel5056.TopologyManager.Configurations.Sapphire.AlternateIpConfig	405
22.66.9	Sel.Sel5056.Snmp.SnmpConfig	405
22.66.10	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.NoForward .	405
22.66.11	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.NoPacketIn .	405
22.66.12	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.NoReceive .	406
22.66.13	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.PortDown . .	406
22.66.14	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.Auto-Negotiation	406
22.66.15	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.Copper . . .	406
22.66.16	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.Fiber	406
22.66.17	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.FortyGiga-BitFullDuplex	406
22.66.18	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneGiga-BitFullDuplex	406
22.66.19	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneGiga-BitHalfDuplex	406
22.66.20	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneHundredGigaBitFullDuplex	406
22.66.21	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneHundredMegaBitFullDuplex	407
22.66.22	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneHundredMegaBitHalfDuplex	407
22.66.23	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OneTera-BitFullDuplex	407
22.66.24	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.OtherBitRate	407
22.66.25	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.Pause	407
22.66.26	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.Pause-Asymmetric	407
22.66.27	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.TenGiga-BitFullDuplex	407
22.66.28	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.TenMega-BitFullDuplex	407
22.66.29	Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.TenMega-BitHalfDuplex	407
22.67	Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.AmSystemPriority	408

22.67.1	Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.System-Critical	409
22.67.2	Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.System-Required	409
22.68	Sel.Sel5056.TopologyManager.Nodes.PortLinkEventConfiguration	409
22.69	Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.RateTypeBase	409
22.69.1	Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.Burst	410
22.69.2	Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.Kilobites-PerSecond	410
22.69.3	Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.PacketsPer-Second	410
22.69.4	Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.Stats	410
23	OData Actions	411
23.1	Sel.AcceptAllProposedLearnedLogicalConnections	411
23.2	Sel.AcceptLearnedLogicalConnections	411
23.3	Sel.AcknowledgeLearnAndLockSession	411
23.4	Sel.AddAbstractNode	412
23.5	Sel.AddOperationalLinkFromHostToSwitch	412
23.6	Sel.AddSilentHost	412
23.7	Sel.Adopt	412
23.7.1	Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink	413
23.7.2	Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode	413
23.7.3	Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort	413
23.8	Sel.AdoptWithConfig	413
23.8.1	Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink	413
23.8.2	Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode	414
23.8.3	Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort	414
23.9	Sel.AsPem	414
23.10	Sel.AuditDeviceConfiguration	414
23.11	Sel.Backup	414
23.12	Sel.BeginLearnAndLock	415
23.13	Sel.ClearLearnedLogicalConnections	415
23.14	Sel.ClearTransaction	415
23.15	Sel.Commission	415
23.16	Sel.Commit	416
23.17	Sel.DeclineAllProposedLearnedLogicalConnections	416
23.18	Sel.DeclineLearnedLogicalConnections	416
23.19	Sel.DeleteAll	417
23.20	Sel.DeleteLearnAndLockSession	417
23.21	Sel.DetectSelRelayFailoverPort	417
23.22	Sel.DisableApplicationLink	417
23.23	Sel.DisableControllerArpToNode	417
23.24	Sel.Disconnect	418
23.25	Sel.EnableApplicationLink	418

23.26 Sel.EnableControllerArpToNode	418
23.27 Sel.FactoryDefaultReset	418
23.28 Sel.FirmwareUpgrade	418
23.29 Sel.GetAvailableGroups	419
23.30 Sel.GetHardTimeoutTimeRemaining	419
23.31 Sel.GetLearnAndLockCounters	419
23.32 Sel.GetLinkByPortAttribute	419
23.33 Sel.GetLogicalConnectionStats	419
23.34 Sel.GetNodeByAttribute	420
23.35 Sel.GetNodesByIp	420
23.36 Sel.GetNodesThatRequireSynchronization	420
23.37 Sel.GetOpenFlowInformation	420
23.38 Sel.GetPathPlanInformation	420
23.39 Sel.GetPortByAttribute	421
23.40 Sel.GetStateBeginEndTimestamps	421
23.41 Sel.GetStatusMessages	421
23.42 Sel.GetUserPreference	421
23.43 Sel.Heartbeat	422
23.44 Sel.InitiateApplicationRegistration	422
23.45 Sel.IsCommissioned	422
23.46 Sel.LearnAndLockIndividualSessionExportCSV	422
23.47 Sel.MarkPortAndAddRelayFailoverLink	422
23.48 Sel.MarkPortAndPerformBlockingDetectSelRelayFailover	423
23.49 Sel.MarkPortAsSelRelayFailover	423
23.50 Sel.MergeNodes	423
23.51 Sel.PacketOut	423
23.52 Sel.PacketOutToTable	423
23.53 Sel.ProposeLearnedLogicalConnections	424
23.54 Sel.Reboot	424
23.55 Sel.RemoveSelRelayFailoverFromPort	424
23.56 Sel.ReplaceConfig	424
23.56.1 Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink	425
23.56.2 Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode	425
23.56.3 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort	425
23.57 Sel.ReplanAll	425
23.58 Sel.ReplanInbandPath	426
23.59 Sel.ReserveVlanVidRange	426
23.60 Sel.ResetCounters	426
23.61 Sel.Restore	426
23.62 Sel.ResubmitLogicalConnection	426
23.63 Sel.RevokeCertificate	427
23.64 Sel.RevokeCertificateByPurpose	427
23.65 Sel.RevokeCertificateByThumbprint	427
23.66 Sel.SendPortMod	427
23.66.1 Sel.Sel5056.TopologyManager.Ports.ConfigPort	428

23.66.2 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort	428
23.67 Sel.SetDone	428
23.68 Sel.SetUserPreference	428
23.69 Sel.StopLearnAndLock	429
23.70 Sel.SynchronizeAllConfiguration	429
23.71 Sel.SynchronizeConfiguration	429
23.72 Sel.SynchronizeOpenFlowConfiguration	429
23.73 Sel.SynchronizeSelectedConfiguration	429
23.74 Sel.TestAuthenticateUser	430
23.75 Sel.TrustCertificate	430
23.76 Sel.Unadopt	430
23.76.1 Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink	430
23.76.2 Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode	430
23.76.3 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort	431
23.77 Sel.UpdatePassword	431
23.78 Sel.UploadCertificate	431
23.79 Sel.UserInitiatedHostDiscovery	431
24 OData Enums	433
24.1 Sel.BlueFrame.Core.ErrorState	433
24.2 Sel.BlueFrame.Core.EventBus.Enums.DurationType	434
24.3 Sel.BlueFrame.Core.EventBus.Enums.FacilityCode	434
24.4 Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel	435
24.5 Sel.BlueFrame.Core.TrustAuthority.Enums.CertificateStatus	436
24.6 Sel.BlueFrame.Core.TrustAuthority.Enums.UserInteraction	436
24.7 Sel.BlueFrame.EventBus.Syslog.RFCType	437
24.8 Sel.BlueFrame.EventBus.Syslog.TransportType	437
24.9 Sel.BlueFrame.SecurityManager.AuthServiceResultState	437
24.10 Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.NodesConfiguration	438
24.11 Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.OutputConfiguration	438
24.12 Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.TableConfiguration	438
24.13 Sel.Sel5056.LearnAndLock.Enums.EndReason	439
24.14 Sel.Sel5056.LearnAndLock.LearnAndLockStateStatus	439
24.15 Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.AutoAcceptStatus	440
24.16 Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnectionState	440
24.17 Sel.Sel5056.LogicalConnectionValidation.Enums.CstCommunicationType	441
24.18 Sel.Sel5056.LogicalConnectionValidation.Enums.LogicalConnectionOrigin	441
24.19 Sel.Sel5056.OpenFlowPlugin.Enums.Of13MeterBandType	441
24.20 Sel.Sel5056.OpenFlowPlugin.Enums.OfpGroupType	442
24.21 Sel.Sel5056.SapphirePlugin.Synchronization.SapphireEndpoint	442
24.22 Sel.Sel5056.Snmp.SnmpAuthenticationMethod	443
24.23 Sel.Sel5056.Snmp.SnmpEncryptionMethod	444
24.24 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.DeviceFramework	444
24.25 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.Sel274xNodeType	444
24.26 Sel.Sel5056.TopologyManager.Enums.CommunicationBand	445

24.27	Sel.Sel5056.TopologyManager.Enums.DiscoveryTrustState	445
24.28	Sel.Sel5056.TopologyManager.Enums.OfpPortStatus	446
24.29	Sel.Sel5056.TopologyManager.Enums.PacketSource	447
24.30	Sel.Sel5056.TopologyManager.Enums.Sel2742SDigitalInputFunctions	447
24.31	Sel.Sel5056.TopologyManager.Enums.Sel274xTimeSource	447
24.32	Sel.Sel5056.TopologyManager.Enums.State	448
24.33	System.Diagnostics.EventLogEntryType	448

VI Appendix 451

A Manual Versions 453

List of Figures

5.1	Tutorial Network Topology	12
-----	---------------------------	----

List of Tables

5.1	Header Values for /api/default/security/Commission	13
5.2	Body Values for Commissioning the SEL-5056	13
5.3	Header Values for /configuration/settings.json	14
5.4	Header Values for /identity/connect/token	18
5.5	Body Values for Logging In With a Username and Password	19
5.6	Header Values for /api/default/settings/Heartbeat	20
5.7	Header Values for /identity/connect/accesstokenvalidation	20
5.8	Parameter Values for /identity/connect/accesstokenvalidation	21
5.9	Header Values for /identity/connect/revocation	22
5.10	Body Values for Logging Out	22
5.11	Header Values for /identity/connect/accesstokenvalidation	22
5.12	Parameter Values for /identity/connect/accesstokenvalidation	23
5.13	Header Values for /api/default/security/users	23
5.14	Body Values for Creating a Permission Level 3 User	24
5.15	Header Values for /api/default/security/users/	25
5.16	Parameter Values for /api/default/security/users/	25

5.17	Header Values for /api/default/config/nodes	27
5.18	Body Values for Creating Configuration Nodes	27
5.19	Header Values for /api/default/config/nodes	34
5.20	Body Values for Creating Configuration Nodes	34
5.21	Header Values for /api/default/operational/nodes	35
5.22	Header Values for /api/default/config/nodes	37
5.23	Parameter Values for /api/default/config/nodes	38
5.24	Header Values for /api/default/operational/nodes	44
5.25	Body Values for Adopting Alpha	44
5.26	Header Values for /api/default/operational/nodes	45
5.27	Parameter Values for /api/default/operational/nodes	45
5.28	Header Values for /api/default/operational/nodes	47
5.29	Body Values for Adopting Charlie	47
5.30	Header Values for /api/default/operational/ports	48
5.31	Parameter Values for /api/default/operational/ports	49
5.32	Header Values for /api/default/operational/links	52
5.33	Header Values for /api/default/operational/nodes	52
5.34	Body Values for Adopting the Hosts Including Using Failover Mode	53
5.35	Header Values for /api/default/operational/ports	53
5.36	Header Values for /api/default/operational/links	55
5.37	Header Values for /api/default/operational/ports	55
5.38	Header Values for /api/default/operational/ports	56
5.39	Header Values for /api/default/config/logicalConnections	57
5.40	Body Values for Creating Logical Connections	57
5.41	Header Values for /api/default/config/logicalConnections	58
5.42	Body Values for Creating Logical Connections	58
5.43	Header Values for /api/default/config/communicationServiceTypes	59
5.44	Body Values for Creating CSTs	60
5.45	Header Values for /api/default/config/logicalConnections	62
5.46	Body Values for Creating CSTs	62
5.47	Header Values for /api/default/config/nodes	63
5.48	Body Values for Adding a Virtual Host	64
5.49	Header Values for /api/default/operational/ports	65
5.50	Body Values for Adding a Virtual Host	65
5.51	Header Values for /api/default/config/flows	66
5.52	Body Values for Managing Flow Entries	66
5.53	Header Values for /api/default/config/flows	68
5.54	Header Values for /api/default/config/flows	69
5.55	Body Values for Managing Flow Entries	69
5.56	Header Values for /api/default/config/flows	71
5.57	Body Values for Managing Flow Entries	72
5.58	Header Values for /api/default/config/flows	75
5.59	Body Values for Managing Flow Entries	75
5.60	Header Values for /api/default/config/groups	76
5.61	Body Values for Managing Group Entries	77

5.62	Header Values for /api/default/config/flows	80
5.63	Body Values for Managing Group Entries	80
5.64	Header Values for /api/default/config/meters	83
5.65	Body Values for Managing Meter Entries	84
5.66	Header Values for /api/default/config/flows	85
5.67	Body Values for Managing Meter Entries	86
7.1	OData Service Roots	95
8.1	Supported OAuth Grant Types	100
8.3	OAuth Grant Types Requirements	100
8.5	OAuth Grant Types Client ID and Secret	101
8.7	OAuth/Token Endpoints	102
8.10	Resource Owner Password Credentials Token Request Parameters	103
8.11	Client Credentials Token Request Parameters	104
9.2	Default Values for Sel.BlueFrame.Core.SecurityManager.LocalUser	115
9.3	Default Values for Sel.BlueFrame.Core.SecurityManager.AuthServiceUser	116
9.4	Header Values for /api/default/security/users/	116
9.5	Body Values for Create a Permission Level 3 User	117
9.6	Header Values for /api/default/security/users	119
9.7	Body Values for Disable a User	119
9.10	Default Values for Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService	124
10.4	Default Values for Sel.Sel5056.TopologyManager.Nodes.ConfigNode	141
10.5	Default Values for Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode	144
10.6	Default Values for Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode	144
10.9	Header Values for /api/default/operational/nodes	168
10.12	Default Values for Sel.Sel5056.TopologyManager.Ports.ConfigPort	176
10.15	Default Values for Sel.Sel5056.TopologyManager.Links.ConfigLink	190
11.2	Default Values for Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection	201
11.3	Header Values for /api/default/config/logicalConnections	202
11.4	Body Values for Create a Unicast LC	202
11.5	Header Values for /api/default/config/logicalConnections	203
11.6	Body Values for Create a Multicast LC	203
11.8	Default Values for Sel.Sel5056.LogicalConnectionValidation.DataObject.Communication-ServiceType	213
11.9	Header Values for /api/default/config/communicationServiceTypes	214
11.10	Body Values for Create a Unicast CST	214
11.11	Header Values for /api/default/config/communicationServiceTypes	215
11.12	Body Values for Create a Multicast CST	215
11.14	Header Values for /api/default/config/vlanVidReservation/ReserveVlanVidRange	221
11.15	Body Values for Reserve a Range of Values	221
11.16	Header Values for /api/default/config/vlanVidReservation/ReserveVlanVidRange	221
11.17	Body Values for Reserve a Single VID	222

12.2	Default Values for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow	227
12.3	Header Values for /api/default/config/flows	228
12.4	Body Values for Create a Simple Flow Entry	229
12.5	Header Values for /api/default/config/flows	232
12.6	Body Values for Update a Simple Flow Entry by Changing the Output	232
12.8	Default Values for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group	237
12.9	Header Values for /api/default/config/groups	238
12.10	Body Values for Create a Simple All Group	239
12.11	Header Values for /api/default/config/groups	240
12.12	Body Values for Create a Simple Fast Failover Group	242
12.13	Header Values for /api/default/config/groups	244
12.14	Body Values for Modify (PUT) the Action Buckets of an All Group	245
12.15	Header Values for /api/default/config/groups	248
12.16	Body Values for Modify (PATCH) the Action Buckets of an All Group	249
12.18	Default Values for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter	254
13.3	Header Values for /api/default/settings/Restore	263
15.2	Default Values for Sel.BlueFrame.ApplicationRegistration.ApplicationLink	283
18.1	Client SignalR Targets	308
20.2	Edm.* Types	317
21.1	Properties for Sel.BlueFrame.ApplicationRegistration.ApplicationLink	319
21.2	Properties for Sel.BlueFrame.Core.EventBus.EventCategory	320
21.3	Properties for Sel.BlueFrame.Core.EventBus.EventType	320
21.4	Property for Sel.BlueFrame.Core.EventBus.ClearedEventType	321
21.5	Properties for Sel.BlueFrame.Core.EventBus.LocaleString	321
21.6	Properties for Sel.BlueFrame.Core.SecurityManager.AuthService	321
21.7	Properties for Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService	322
21.8	Properties for Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup	323
21.9	Properties for Sel.BlueFrame.Core.SecurityManager.Role	323
21.10	Properties for Sel.BlueFrame.Core.SecurityManager.User	324
21.11	Properties for Sel.BlueFrame.Core.SecurityManager.LocalUser	324
21.12	Property for Sel.BlueFrame.Core.SecurityManager.AuthServiceUser	324
21.13	Property for Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction	324
21.14	Properties for Sel.BlueFrame.RestBroker.Plumbing.RestSettings	325
21.15	Properties for Sel.BlueFrame.SecurityManager.SecurityManagerSettings	326
21.16	Properties for Sel.BlueFrame.TrustAuthority.CrlSettings	326
21.17	Properties for Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation	327
21.18	Properties for Sel.BlueFrame.UserPreferences.Preference	327
21.19	Properties for Sel.Sel5056.LearnAndLock.LearnAndLockSettings	328
21.20	Properties for Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogical- Connection	329
21.21	Properties for Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession	330

21.22	Properties for Sel.Sel5056.LogicalConnectionValidation.DataObject.Communication-ServiceType	331
21.23	Properties for Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection . .	332
21.24	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow	333
21.25	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats	333
21.26	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group	334
21.27	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc	334
21.28	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats	335
21.29	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter	335
21.30	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions	336
21.31	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures	336
21.32	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats	337
21.33	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc	337
21.34	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats	337
21.35	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ReceivedPacket	338
21.36	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats	338
21.37	Properties for Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation	339
21.38	Property for Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidDestinationReservation	339
21.39	Properties for Sel.Sel5056.Snmp.SnmpUser	340
21.40	Properties for Sel.Sel5056.SystemMessaging.SystemMessage	340
21.41	Properties for Sel.Sel5056.TopologyManager.Links.ConfigLink	341
21.42	Properties for Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink	341
21.43	Properties for Sel.Sel5056.TopologyManager.NetworkSettings	342
21.44	Properties for Sel.Sel5056.TopologyManager.Nodes.ConfigNode	342
21.45	Properties for Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode	343
21.46	Properties for Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode	343
21.47	Properties for Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode	344
21.48	Properties for Sel.Sel5056.TopologyManager.Ports.ConfigPort	345
21.49	Properties for Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort	345
21.50	Properties for Sel.Sel5056.TopologyManager.Synchronization.SynchronizationRequest . .	346
21.51	Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowAdd-SynchronizationRequest	347
21.52	Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowDelete-SynchronizationRequest	347
21.53	Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModify-SynchronizationRequest	347
21.54	Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupAdd-SynchronizationRequest	348
21.55	Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupDelete-SynchronizationRequest	348
21.56	Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupModifySynchronizationRequest	348
21.57	Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterAdd-SynchronizationRequest	349

21.58	Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterDelete-SynchronizationRequest	349
21.59	Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterModify-SynchronizationRequest	349
21.60	Properties for Sel.Sel5056.OpenFlowPlugin.PortModController.PortModSynchronization	350
21.61	Properties for Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization	350
22.1	Properties for Sel.BlueFrame.ApplicationRegistration.AuthenticationType	351
22.2	Properties for Sel.BlueFrame.CommissioningManager.UserCommissioner.User-CommissioningConcern	352
22.3	Properties for Sel.BlueFrame.Core.EventBus.Behaviors.Behavior	352
22.4	Properties for Sel.BlueFrame.EventBus.Syslog.SyslogBehavior	353
22.5	Properties for Sel.BlueFrame.EventBus.WindowsEventLog.WindowsEventLogBehavior	353
22.6	Properties for Sel.BlueFrame.Core.EventBus.LocaleRecord	353
22.7	Property for Sel.BlueFrame.Core.FileArgument	354
22.8	Properties for Sel.BlueFrame.Core.FileReturnValue	354
22.9	Property for Sel.BlueFrame.Core.Networking.NetworkInterfaceAddress	354
22.10	Property for Sel.BlueFrame.Core.SecurityManager.Password	354
22.11	Properties for Sel.BlueFrame.Core.SecurityManager.Permission	355
22.12	Properties for Sel.BlueFrame.Core.DataBrokerPermission	355
22.13	Property for Sel.BlueFrame.Core.SecurityAttributePermission	355
22.14	Property for Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission	356
22.15	Property for Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission	356
22.16	Properties for Sel.BlueFrame.Core.SecurityManager.UserFailedLogin	356
22.17	Property for Sel.Sel5056.Drivers.OpenFlow.Trust.OpenFlowDeviceCertificate	357
22.18	Properties for Sel.Sel5056.SapphirePlugin.Trust.SapphireDeviceCertificate	357
22.19	Properties for Sel.BlueFrame.SecurityManager.AuthServiceGroupDescription	358
22.20	Properties for Sel.BlueFrame.SecurityManager.AuthServiceResult	358
22.21	Property for Sel.BlueFrame.SecurityManager.AuthServiceResultCollection	358
22.22	Property for Sel.BlueFrame.SecurityManager.GroupCollection	359
22.23	Properties for Sel.BlueFrame.UserPreferences.Setting	359
22.24	Properties for Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.Flow-ProgrammingOptions	360
22.25	Properties for Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.MeterMetadata	360
22.26	Properties for Sel.Sel5056.LearnAndLock.CSV.AdoptedHostInfo	361
22.27	Properties for Sel.Sel5056.LearnAndLock.CSV.AdoptedLinkInfo	361
22.28	Properties for Sel.Sel5056.LearnAndLock.CSV.AdoptedSwitchInfo	361
22.29	Properties for Sel.Sel5056.LearnAndLock.Configuration.AutoAdoptConfig	362
22.30	Properties for Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.Configuration.-LogicalConnectionLearningConfig	362
22.31	Properties for Sel.Sel5056.LearnAndLock.ProposedCommunicationServiceType	363
22.32	Property for Sel.Sel5056.LearnAndLock.Sessions.AutoAdoptSession	363
22.33	Property for Sel.Sel5056.LearnAndLock.Sessions.LogicalConnectionLearningSession	363
22.34	Properties for Sel.Sel5056.LearnAndLock.Status.LearnAndLockCounters	364
22.35	Properties for Sel.Sel5056.LearnAndLock.Status.LearnAndLockStateBeginEndTimestamps	365

22.36	Properties for Sel.Sel5056.LogicalConnectionValidation.DataObject.AutoFillSettings . . .	365
22.37	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetQueueAction	367
22.38	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupAction	367
22.39	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputAction	367
22.40	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PushVlanAction	367
22.41	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetFieldAction	368
22.42	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupActionByAlias	368
22.43	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputActionByAlias	368
22.44	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BandStat	368
22.45	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket	369
22.46	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupBucketByAlias	369
22.47	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortBucketByAlias	369
22.48	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BucketStats	369
22.49	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowData	370
22.50	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat	371
22.51	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupData	371
22.52	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat	371
22.53	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction	372
22.54	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GoToTable	373
22.55	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterInstruction	373
22.56	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterInstructionByAlias . . .	373
22.57	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match	373
22.58	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.MatchField . . .	374
22.59	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpSpa	377
22.60	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpTpa	377
22.61	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthDst	377
22.62	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthSrc	378
22.63	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.InPort	378
22.64	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Dst	378
22.65	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Src	378
22.66	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.VlanVid	379
22.67	Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ethernet- ApplicationByAlias	380
22.68	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand	380
22.69	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandHeader	381
22.70	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDesc	381
22.71	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStat	382
22.72	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port	382
22.73	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStat	383
22.74	Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStat	383
22.75	Property for Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.FlowDelete- Dependency	385
22.76	Property for Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupDelete- Dependency	385
22.77	Property for Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupDependency	385

22.78	Property for Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.MeterDependency	385
22.79	Property for Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.FlowModify- Dependency	385
22.80	Property for Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupModify- Dependency	386
22.81	Properties for Sel.Sel5056.PathPlanner.DataObject.PathPlanHopCollection	386
22.82	Properties for Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.- Types.AmDestination	386
22.83	Properties for Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.- Types.RelayFailoverDestination	387
22.84	Property for Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.- Types.SimpleDestination	387
22.85	Properties for Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.- Types.TraditionalTiePointDestination	387
22.86	Property for Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.- Types.ProgrammableDestination	387
22.87	Properties for Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.- Types.InbandDestination	388
22.88	Properties for Sel.Sel5056.PathPlanner.RestController.DirectionLink	388
22.89	Properties for Sel.Sel5056.PathPlanner.RestController.FlowReferences	388
22.90	Properties for Sel.Sel5056.PathPlanner.RestController.FlowWithStats	389
22.91	Properties for Sel.Sel5056.PathPlanner.RestController.GroupReferences	389
22.92	Properties for Sel.Sel5056.PathPlanner.RestController.GroupWithStats	389
22.93	Properties for Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats	390
22.94	Properties for Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo	390
22.95	Properties for Sel.Sel5056.PathPlanner.RestController.PathPlanInfo	390
22.96	Properties for Sel.Sel5056.Tags.LogicalConnectionTag	392
22.97	Property for Sel.Sel5056.Tags.InBandPortIdTag	392
22.98	Properties for Sel.Sel5056.Tags.GenericArpFlowTag	392
22.99	Property for Sel.Sel5056.Tags.OpenFlowPortIdTag	392
22.100	Properties for Sel.Sel5056.Tags.KeyValueTag	393
22.101	Properties for Sel.Sel5056.Tags.InBandTag	393
22.102	Property for Sel.Sel5056.Tags.ControllerTag	393
22.103	Properties for Sel.Sel5056.Tags.IpAddressTag	393
22.104	Properties for Sel.Sel5056.Tags.MacAddressTag	393
22.105	Property for Sel.Sel5056.Tags.ControllerRouteTag	394
22.106	Property for Sel.Sel5056.LearnAndLock.AutoAdoption.Tags.AutoAdoptionTag	394
22.107	Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr	396
22.108	Property for Sel.Sel5056.TopologyManager.Attributes.Operational.Port.Sel2740SPortAttr	396
22.109	Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Port.Ethernet- ApplicationAttr	396
22.110	Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Port.Ethernet- ApplicationNodeAttr	397
22.111	Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Node.OpenFlowAttr	397

22.112	Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Node.OpenFlow-TableFeaturesAttr	397
22.113	Property for Sel.Sel5056.TopologyManager.Attributes.Operational.Node.IpAttr	397
22.114	Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr	398
22.115	Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Node.Firmware-UpdateAttr	398
22.116	Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Node.UserHintedIp-AddressAttr	399
22.117	Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.EthernetAttr	400
22.118	Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedMapped-SwitchPortAttr	400
22.119	Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Networking-AbstractionAttr	400
22.120	Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedNodeAttr	401
22.121	Properties for Sel.Sel5056.Southbound.DeviceManagement.CommissioningStateAttr	401
22.122	Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Node.TableFeature	401
22.123	Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Port.Ethernet-ApplicationProviderInformation	402
22.124	Property for Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriority	404
22.125	Property for Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractionConfig	404
22.126	Properties for Sel.Sel5056.TopologyManager.Configurations.Sapphire.AlternateIpConfig	405
22.127	Properties for Sel.Sel5056.Snmp.SnmpConfig	405
22.128	Properties for Sel.Sel5056.TopologyManager.Nodes.PortLinkEventConfiguration	409
23.1	Property for Sel.AcceptLearnedLogicalConnections	411
23.2	Property for Sel.AcknowledgeLearnAndLockSession	411
23.3	Property for Sel.AddAbstractNode	412
23.4	Property for Sel.AddOperationalLinkFromHostToSwitch	412
23.5	Property for Sel.AddSilentHost	412
23.6	Property for Sel.AdoptWithConfig	413
23.7	Property for Sel.AdoptWithConfig	414
23.8	Property for Sel.AdoptWithConfig	414
23.9	Property for Sel.AuditDeviceConfiguration	414
23.10	Property for Sel.Backup	414
23.11	Properties for Sel.BeginLearnAndLock	415
23.12	Property for Sel.ClearLearnedLogicalConnections	415
23.13	Property for Sel.Commission	416
23.14	Property for Sel.DeclineLearnedLogicalConnections	416
23.15	Property for Sel.DeleteLearnAndLockSession	417
23.16	Property for Sel.FirmwareUpgrade	418
23.17	Property for Sel.GetHardTimeoutTimeRemaining	419
23.18	Property for Sel.GetLearnAndLockCounters	419
23.19	Property for Sel.GetLinkByPortAttribute	419
23.20	Property for Sel.GetNodeByAttribute	420
23.21	Property for Sel.GetNodesByIp	420

23.22	Property for Sel.GetPortByAttribute	421
23.23	Property for Sel.GetStateBeginEndTimestamps	421
23.24	Property for Sel.GetStatusMessages	421
23.25	Properties for Sel.GetUserPreference	421
23.26	Property for Sel.InitiateApplicationRegistration	422
23.27	Property for Sel.LearnAndLockIndividualSessionExportCSV	422
23.28	Property for Sel.MarkPortAndAddRelayFailoverLink	422
23.29	Properties for Sel.MergeNodes	423
23.30	Properties for Sel.PacketOut	423
23.31	Properties for Sel.PacketOutToTable	424
23.32	Property for Sel.ProposeLearnedLogicalConnections	424
23.33	Property for Sel.ReplaceConfig	425
23.34	Property for Sel.ReplaceConfig	425
23.35	Property for Sel.ReplaceConfig	425
23.36	Properties for Sel.ReserveVlanVidRange	426
23.37	Properties for Sel.Restore	426
23.38	Property for Sel.RevokeCertificateByPurpose	427
23.39	Property for Sel.RevokeCertificateByThumbprint	427
23.40	Properties for Sel.SendPortMod	428
23.41	Properties for Sel.SendPortMod	428
23.42	Properties for Sel.SetUserPreference	428
23.43	Property for Sel.SynchronizeConfiguration	429
23.44	Property for Sel.SynchronizeOpenFlowConfiguration	429
23.45	Property for Sel.SynchronizeSelectedConfiguration	429
23.46	Properties for Sel.TestAuthenticateUser	430
23.47	Properties for Sel.UpdatePassword	431
23.48	Properties for Sel.UploadCertificate	431
23.49	Property for Sel.UserInitiatedHostDiscovery	431
24.1	Sel.BlueFrame.Core.ErrorState	433
24.2	Sel.BlueFrame.Core.EventBus.Enums.DurationType	434
24.3	Sel.BlueFrame.Core.EventBus.Enums.FacilityCode	435
24.4	Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel	435
24.5	Sel.BlueFrame.Core.TrustAuthority.Enums.CertificateStatus	436
24.6	Sel.BlueFrame.Core.TrustAuthority.Enums.UserInteraction	436
24.7	Sel.BlueFrame.EventBus.Syslog.RFCType	437
24.8	Sel.BlueFrame.EventBus.Syslog.TransportType	437
24.9	Sel.BlueFrame.SecurityManager.AuthServiceResultState	437
24.10	Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.NodesConfiguration	438
24.11	Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.OutputConfiguration	438
24.12	Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.TableConfiguration	438
24.13	Sel.Sel5056.LearnAndLock.Enums.EndReason	439
24.14	Sel.Sel5056.LearnAndLock.LearnAndLockStateStatus	439
24.15	Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.AutoAcceptStatus	440
24.16	Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnectionState	440

24.17	Sel.Sel5056.LogicalConnectionValidation.Enums.CstCommunicationType	441
24.18	Sel.Sel5056.LogicalConnectionValidation.Enums.LogicalConnectionOrigin	441
24.19	Sel.Sel5056.OpenFlowPlugin.Enums.OF13MeterBandType	442
24.20	Sel.Sel5056.OpenFlowPlugin.Enums.OfpGroupType	442
24.21	Sel.Sel5056.SapphirePlugin.Synchronization.SapphireEndpoint	443
24.22	Sel.Sel5056.Snmp.SnmpAuthenticationMethod	443
24.23	Sel.Sel5056.Snmp.SnmpEncryptionMethod	444
24.24	Sel.Sel5056.TopologyManager.Attributes.Operational.Node.DeviceFramework	444
24.25	Sel.Sel5056.TopologyManager.Attributes.Operational.Node.Sel274xNodeType	444
24.26	Sel.Sel5056.TopologyManager.Enums.CommunicationBand	445
24.27	Sel.Sel5056.TopologyManager.Enums.DiscoveryTrustState	446
24.28	Sel.Sel5056.TopologyManager.Enums.OfpPortStatus	446
24.29	Sel.Sel5056.TopologyManager.Enums.PacketSource	447
24.30	Sel.Sel5056.TopologyManager.Enums.Sel2742SDigitalInputFunctions	447
24.31	Sel.Sel5056.TopologyManager.Enums.Sel274xTimeSource	448
24.32	Sel.Sel5056.TopologyManager.Enums.State	448
24.33	System.Diagnostics.EventLogEntryType	449
A.1	Manual Revision History	453

List of Examples

8.1	Resource Owner Password Credentials - Obtaining a Token	103
8.2	Client Credentials - Obtaining a Token	104
8.3	Inspecting a token	104
8.4	Resource Owner Password Credentials - Revoking a Token	105
8.5	Client Credentials - Revoking a Token	105
8.6	Extending the expiration time of a Token	106
9.7	Create a Permission Level 3 User	116
9.8	Disable a User	119
10.9	Upgrade an SEL-2740S or SEL-2742S	168
11.10	Create a Unicast LC	202
11.11	Create a Multicast LC	203
11.12	Create a Unicast CST	213
11.13	Create a Multicast CST	215
11.14	Reserve a Range of Values	221
11.15	Reserve a Single VID	221
12.16	Create a Simple Flow Entry	228
12.17	Update a Simple Flow Entry by Changing the Outport	232
12.18	Create a Simple All Group	238

12.19 Create a Simple Fast Failover Group 240

12.20 Modify (PUT) the Action Buckets of an All Group 244

12.21 Modify (PATCH) the Action Buckets of an All Group 248

13.22 Import an SEL-5056 Database 263

18.23 Subscribing to all Flow entry changes using Websocket 309

18.24 Subscribing to a specific Flow entry change using Websocket 310

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

Chapter 1

Introduction

This document describes the SEL-5056 REST interface by providing a list of endpoints, a tutorial that uses the endpoints to achieve useful results, a description of the data model, and other material. Documentation for the SEL-2740S and SEL-2742S REST interfaces, including accessing the information found on the Device view, will be in another document.

This document focuses 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 and are familiar with using JSON, OpenFlow, and operating the SEL-5056. Familiarity with OData is not assumed but will be helpful.

Chapter 2

Conventions

2.1 URIs

To conserve space, this document only displays part of Uniform Resource Locators (URLs) (schema and authority such as `https://`). Assuming the fully qualified domain name (FQDN) setting 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 SignalR) include the `/api/default/` prefix in this version of the SEL-5056. Occasionally when context is clear outside of HTTP messages, the `/api/default` prefix can be replaced with the character `~`, so the URI `~/config/nodes/` corresponds to `https://localhost/api/default/config/nodes/`.

2.2 Displaying HTTP Messages

To simplify this document and the included HTTP messages, HTTP messages are formatted in the following manner:

- Most responses from the SEL-5056 are chunked, so have the transfer encoding HTTP header set to the value of `chunked`. In these examples, the body is shown unchunked but the transfer encoding header shows the original chunked transfer encoding header.
- To simplify this document, some of the code has been modified to fit the line length limits, such as the insertion of new lines and indentation. Therefore, the Content-Length will not match what is shown. If the Content-Type is JSON, the body is displayed in a modified "pretty print" style to be more readable.
- The following headers have been removed from the SEL-5056 HTTP responses:
 - Cache-Control

- Date
 - OData-Version
 - Pragma
 - Location
 - Server
- OAuth tokens has been modified to be 000...000. An example of a token is 929f5cf4d7a5863d8a665bbb0ac19194a6fdf25a346b1b1384eba4df83886577.

Chapter 3

Getting Started

There are two 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 **End-point** sections for specific endpoints.

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 FQDN setting. See the SEL-5056 instruction manual for further information on configuring this setting.

4.1 API Onboarding

You can access the API 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 by 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 the credentials covered in [Authentication \(OAuth\)](#).

4.1.1 API Discovery

OData exposes the metadata 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 non-user account required to access the REST interface.

4.1.3 API Applications

If you want to create an external application that uses the [client credentials or authorization code grants](#), you first must understand [OAuth token management](#) and then [register an application](#).

4.2 Accessing the REST Interface

The SEL-5056 only supports HTTPS. A user, application, or script must first obtain an **OAuth** access token by using the appropriate OAuth endpoints before making REST calls to endpoints that require a valid endpoint.

The following summarizes the overall process:

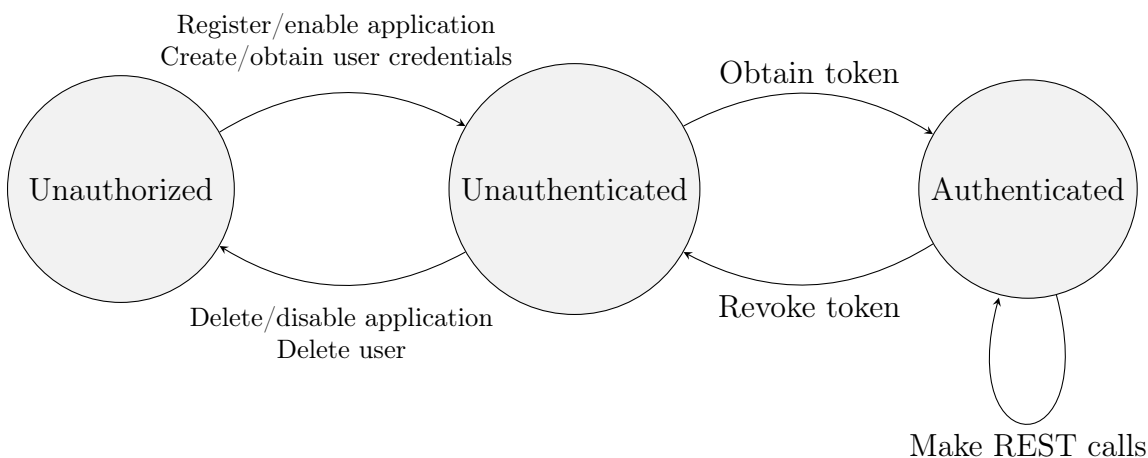
- Obtain credentials with the appropriate permissions or use the Implicit grant type with the appropriate role.
- Obtain an OAuth token by using a supported grant type.
- Make REST calls.
- (Optional) Revoke the token (i.e., log out).

The following are the two paths for obtaining credentials:

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

After obtaining credentials, you can request a token from the SEL-5056 by using a supported OAuth type. **OAuth and ID Connect** handle token management. After you have a token, you can make REST calls by using the token, assuming the token has the appropriate permission(s) for the endpoint(s).

The following shows a summary of the process:



Chapter 5

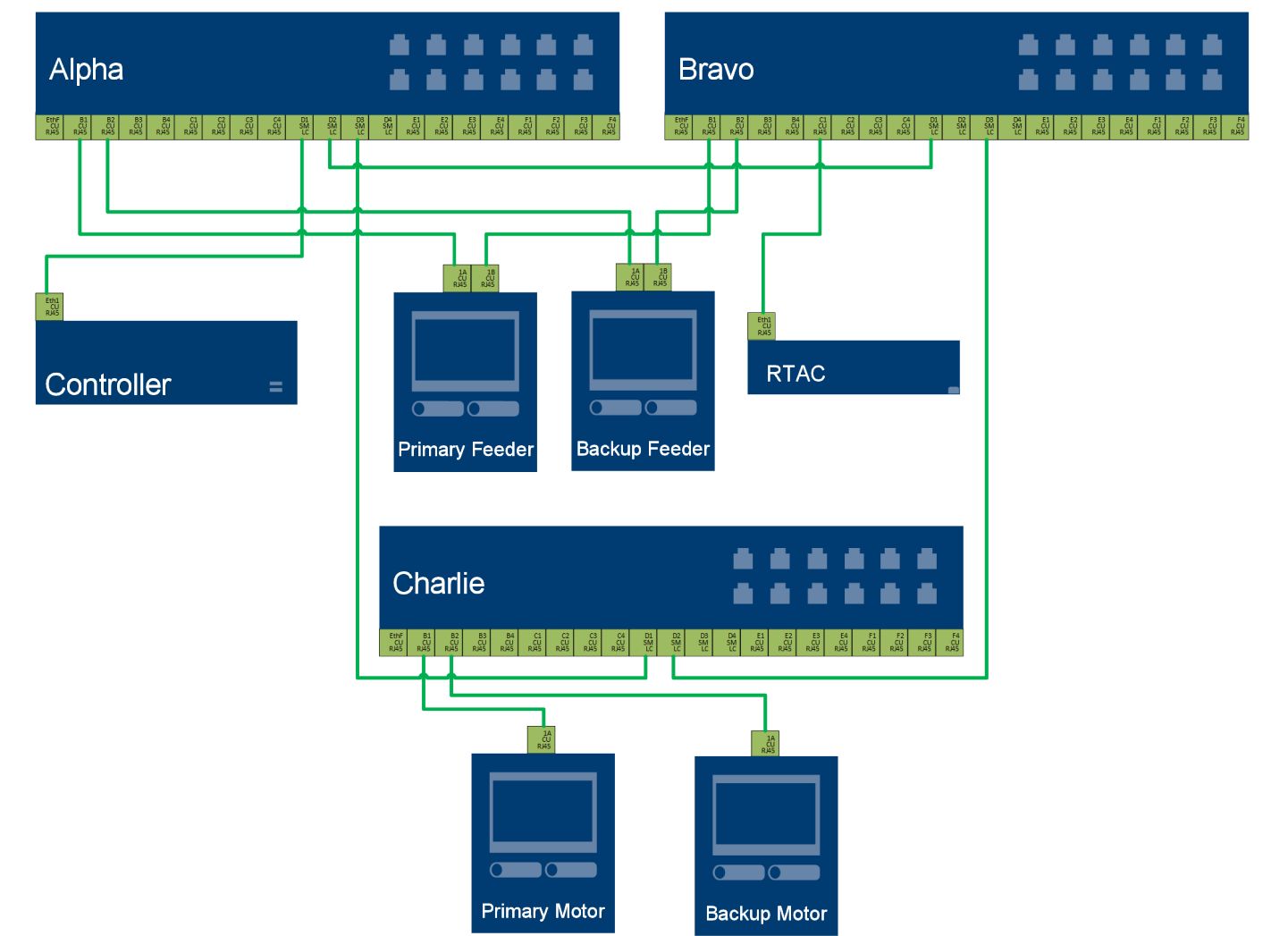
Tutorial

See **Conventions** for further information about the format of HTTP messages shown in this chapter.

This chapter covers making several common SEL-5056 REST calls, such as those to used setup the network shown in Figure 5.1.

For each HTTP call that is shown, this document includes a table listing the HTTP headers, a table listing the body object properties (if applicable), and the HTTP request message along with the HTTP reply (formatted according to **Conventions**).

Figure 5.1: Tutorial Network Topology



5.1 Commissioning the SEL-5056

Begin by commissioning the database with an initial user with the Security Administrator role. You can do this by using the commissioning and settings tool or using a simple call that requires no token to make. The following shows the headers, body, and resulting request.

Key	Value
Endpoint	/api/default/security/Commission
Method	POST

Content-Type	application/json
--------------	------------------

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

Request

You need one property 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.User-
      CommissioningConcern",
      "username": "admin",
      "password": "Example1!"
    }
  ]
}
```

Reply

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

The SEL-5056 is now commissioned.

5.2 Using the OData Schema

The SEL-5056 REST interface, in addition to the endpoints themselves, hosts endpoint metadata information. This section covers looking at the REST interface metadata to find out which OData endpoints and their corresponding properties are defined. You can obtain a list of OData service roots (which are endpoints that contain the endpoints that you will be using) from the config.json endpoint. In addition to the OData service roots, obtain both the SEL-5056 version (controllerVersion) and the time according to the SEL-5056 (currentTime).

Key	Value
Endpoint	/configuration/settings.json
Method	GET
Content-Type	Not Present

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

Request

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

Reply

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 330

{
  "webUri": "https://localhost:443",
  "controllerVersion": "2.8.0",
  "buildDate": "2021-10-29T22:07:04.000-04:00",
  "trees": [
    "api/default/settings",
    "api/default/security",
    "api/default/certificate",
    "api/default/config",
    "api/default/operational"
  ],
  "systemCommissioned": true
  "remoteIp": "127.0.0.1",
  "currentTime": "2021-11-23T14:36:36.996-05:00"
}
```

The trees property contains the list of service roots available. This allows you to find all the roots that

contain all of the OData endpoints, which include the endpoints except for 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

Navigating to any of these endpoints shows the object and action endpoints that belong to each service root. For example, if you navigate to /api/default/config, you obtain 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"
}
]
}

```

Logical connections are listed as logicalConnections, so the endpoint is located at /api/default/config/logicalConnections. This is the endpoint used to retrieve or post logical connections, as listed in the [Logical Connection OData Endpoint section](#).

To figure out how to commission the SEL-5056 based on the OData information obtained from the REST API, look at the /api/default/security endpoint. The JSON that is returned shows how each function is mapped to an endpoint. The /api/default/security endpoint returns the following:

```

{
  "@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"
}
]
}

```

You can obtain the XML document containing the metadata that contains a description of the endpoints and objects, such as the properties or constraints on the values for those properties, that are available on this branch by going to [https://localhost/api/default/security/\\$metadata](https://localhost/api/default/security/$metadata).

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

One of the actions has the attribute `Name="Commission"`, which includes the endpoint used previously. The endpoints have one parameter: `commissioningList` of Type `Collection(Sel.BlueFrame.CommissioningManager.CommissioningConcern)`. This means that this parameter is a list (from `Collection`) and the type of elements in the list are of the `Sel.BlueFrame.CommissioningManager.CommissioningConcern` type.

If you search for `Sel.BlueFrame.CommissioningManager.CommissioningConcern` in the XML document, you 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, obtain 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 name space is `Sel.BlueFrame.CommissioningManager.UserCommissioner` and the name of the `ComplexType` used is `UserCommissioningConcern`. You must place a `#` character before the `odata.type` value and a `@` character before `odata.type` in the request. The two properties are `username` and `password`. From this, the HTTP body would be the following:

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

where `X` and `Y` are the values for the `username` and `password`, respectively. Because a list is required, create a list of one by using this data structure that matches the body of POST used to commission the SEL-5056 previously.

5.3 Logging In With a Username and Password

Next, log in by using the credentials of the commissioned user. Logging in to the SEL-5056 means obtaining an OAuth token. These tokens have an implicit set of permissions that allow you to access endpoints. You obtain a token by using credentials and a grant type. For now, only focus on the `password` grant type, which requires a username and password.

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

Table 5.4: Header Values for `/identity/connect/token`

Request

You need eight 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": 600,
  "token_type": "Bearer",
  "scope": "Rest"
}
```

In the response is the `access_token` property that contains the access token. The token is valid for 600 seconds.

5.4 Refreshing the Token

To extend the duration of the token, use the Heartbeat endpoint or use the token with an OData endpoint; either option resets the token expiration back to 600 seconds.

Key	Value
Endpoint	/api/default/settings/Heartbeat
Method	POST
Content-Type	Not Present

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

Request

```
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

Use the token validation endpoint to check that a token is valid, that is, verify you can access any set of endpoints and methods (GET, POST, etc.) to which Security Administrator has access. Here the token must be passed as a URL parameter that is not in a body.

Key	Value
Endpoint	/identity/connect/accesstokenvalidation
Method	POST
Content-Type	Not Present

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

Key	Value
token	000...000

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

Request

```
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: 316

{
  "aud": [],
  "iss": "https://localhost/identity",
  "nbf": 1637696686,
  "exp": 1637697286,
  "client_id": "password-client",
  "scope": ["Rest"],
  "sub": "admin",
  "name": "admin",
  "amr": "local",
  "idp": "local",
  "auth_time": "1637696686",
  "role": "SecurityAdministrator",
  "ipAddress": "127.0.0.1",
  "lastSuccessfulLogin": "2021-11-23T19:42:22.8432416Z"
}
```

The expiration date is displayed in the exp property of the response. This is in number of seconds since epoch, which is January 1, 1970 at 00:00:00 (UTC). As mentioned previously, the current system time is available at the /configuration/settings.json endpoint. If the token is not valid, the response is code 400.

5.6 Logging Out

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

Key	Value
Endpoint	/identity/connect/revocation

Method	POST
Content-Type	application/x-www-form-urlencoded

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

Request

You need four properties for the URL encoded body.

Property	Value
token	000...000
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=000...000&
token_type_hint=access_token&
client_id=password-client&
client_secret=Rest%20Interface
```

Reply

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

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

Key	Value
Endpoint	/identity/connect/accesstokenvalidation
Method	POST
Content-Type	Not Present

Table 5.11: Header Values for /identity/connect/accesstokenvalidation

Key	Value
token	000...000

Table 5.12: Parameter Values for /identity/connect/accesstokenvalidation

Request

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

Reply

```
HTTP/1.1 400 Bad Request
Content-Length: 0
```

5.7 Creating a Permission Level 3 User

First, log back in with the Security Administrator user by using the same request as **before** to obtain a token.

The next step is to add a Permission Level 3 (PL3) user. For example, if you look at the listed available roles for **adding an LC**, the PermissionLevel3 role is the only one in the Available User Roles with Required Permissions list.

Conversely, you can see from the Available User Roles with Required Permissions section of the **Add users endpoint** that the required role is SecurityAdministrator. Also give the PL3 user the monitor role.

Refer to **Create Users** for more information on user creation. The URL is listed as /api/default/security/users/ and the method is POST.

Looking at the request structure, you need at least three properties (the properties that are not in italics are required): the OData type (@odata.type), the password (clearTextPass), and the username. Also set the optional (because it is listed in italics) roles property to add in the PL3 and monitor roles.

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

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

Request

You need four properties for the JSON body.

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

Table 5.14: 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: 603

{
  "@odata.context": "https://localhost/api/default/security/$metadata#users/Sel-
.BlueFrame.Core.SecurityManager.LocalUser/$entity",
  "@odata.type":   "#Sel.BlueFrame.Core.SecurityManager.LocalUser",
  "id":            "a28fa3c5217014410a8879df5eb1abdd",
  "displayName":   "pl3",
  "enabled":       true
}
```



```

"isLockedOut": false,
"lastSuccessfulLogin": "0001-01-01T00:00:00Z",
"lastSuccessfulLoginIp": null,
"lockedOutUntil": "0001-01-01T00:00:00-05:00",
"lockoutTime": "0001-01-01T00:00:00-05:00",
"roles": [
  "PermissionLevel3",
  "Monitor"
],
"username": "pl3",
"clearTextPass": null,
"passwordCreationTime": "0001-01-01T00:00:00Z",
"failedLogins": []
"passwordStore": []
}

```

For this POST call, you receive the new user object back.

5.8 Retrieving the Permission Level 3 User

You have received the created user object back with the POST request, but what if you need to retrieve the user at a later time, for example, to see if you are locked out and that is why you cannot to log back in?

Because you posted to the `/api/default/security/users` endpoint, you can retrieve the user from the `/api/default/security/users` endpoint. There are two types of GET calls, individual GET (`/api/default/security/users(id')`) or a GET all (`/api/default/security/users/`). To use the individual GET, you must know the ID (the value of the `id` property). If you do not have the ID, retrieve the complete list of users and then iterate to find the user with the username `pl3`.

You can also use **OData filtering** to retrieve the user by using the value of an attribute, in this case, the username (another option is using the `displayName`). The filtering is done by the SEL-5056 and the return value contains only the objects that match the filter or an empty list if no objects match.

Key	Value
Endpoint	<code>/api/default/security/users/</code>
Method	GET
Content-Type	Not Present

Table 5.15: Header Values for `/api/default/security/users/`

Key	Value
<code>\$filter</code>	<code>username eq 'pl3'</code>

Table 5.16: Parameter Values for `/api/default/security/users/`

Request

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

Reply

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 562

{
  "@odata.context": "https://localhost/api/default/security/$metadata#users",
  "value": [
    {
      "@odata.type": "#Sel.BlueFrame.Core.SecurityManager.LocalUser",
      "id": "af0d746dd67f449d99baafdd66f05d59",
      "displayName": "pl3",
      "enabled": true
      "isLockedOut": false,
      "lastSuccessfulLogin": "0001-01-01T00:00:00Z",
      "lastSuccessLoginIp": null,
      "lockedOutUntil": "0001-01-01T00:00:00-05:00",
      "lockoutTime": "0001-01-01T00:00:00-05:00",
      "roles": [
        "PermissionLevel3",
        "Monitor"
      ],
      "username": "pl3",
      "clearTextPass": null,
      "passwordCreationTime": "0001-01-01T00:00:00Z",
      "failedLogins": []
      "passwordStore": []
    }
  ]
}
```

Note that only one user, the one with the displayName of pl3, is returned.

5.9 Creating Configuration Nodes

First, create configuration nodes for the SEL-2740S switches. The SEL-5056 requires at least the IPv4 address of the SEL-2740S, the subnet of the switch, the default gateway of the switch, and the IPv4 address of the SEL-5056 to create a configuration node. The other settings will be covered later in this document.

The following table shows the settings for each switch:

Name	Switch 1	Switch 2	Switch 3	Switch 4
Subnet	255.255.0.0	255.255.0.0	255.255.0.0	255.255.0.0
Default Gateway	192.168.1.1	192.168.1.1	192.168.1.1	192.168.1.1
Controller IP Address	192.168.1.1	192.168.1.1	192.168.1.1	192.168.1.1

Configuration nodes come in two types: SEL-2740S and generic. If you are creating an SEL-2740S node, the @odata.type parameter must be included in the POST request.

The following table includes information for the endpoint for creating the first switch.

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

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

Request

You need six 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

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: 219

{
  "@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"
}
```

Reply

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 4026

{
  "@odata.context": "https://localhost/api/default/config/$metadata#nodes/Sel-
.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode/$entity",
  "@odata.type": "#Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode",
  "id": "a97287576ee4243fdbcccce172a9d067",
  "state": "Configured",
  "displayName": "Alpha",
  "linkedKey": null,
  "ports": []
  "additionalTrustedCertificates": []
  "alarmMinimumDuration": 1,
  "gigabitCopperFastFailover": false,
  "controllerIp": "192.168.1.1",
  "defaultGateway": "192.168.1.1",
  "enablePtp": false,
  "ipAddress": "192.168.1.100",
  "ntpServers": []
  "subnetMask": "255.255.0.0",
  "timeSourceSelection": "None",
  "configurations": []
  "tags": []
  "portLinkEventConfigurations": [
    {
      "portId": "F",
      "linkDownAutoClear": false,
      "linkDownSeverity": "Notice",

```

```
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "1",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "2",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "3",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "4",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "5",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "6",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
```

```
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "7",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "8",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "9",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "10",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "11",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "12",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
```

```
}
{
  "portId": "13",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
  "portId": "14",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
  "portId": "15",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
  "portId": "16",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
  "portId": "17",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
  "portId": "18",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
```

```

{
  "portId": "19",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
  "portId": "20",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
],
"eventCategories": [
  {
    "key": "default",
    "id": "default",
    "behaviors": [
      {
        "@odata.type": "#Sel.BlueFrame.Particle.AlarmContact.AlarmContact-
Behavior",
        "name": "default",
        "severity": "Error"
      }
      {
        "@odata.type": "#Sel.BlueFrame.Particle.LocalEventStore.LocalEventStore-
Behavior",
        "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": []
}
]
}

```

Create the other two switches by using similar requests.

The following five hosts only require the displayName value to be configured.

1. Primary Feeder
2. Backup Feeder

3. Primary Motor
4. Backup Motor
5. RTAC

The following table includes the information for the request for creating the first host (Primary Feeder).

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

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

Request

You need one property for the JSON body.

Property	Value
displayName	Primary Feeder

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: 33

{
  "displayName": "Primary Feeder"
}
```

Reply

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

{
  "@odata.context": "https://localhost/api/default/config/$metadata#nodes/-
  $entity",
  "id": "ab7364fc3e25b46a28ef1bfccfba9f8f",
  "state": "Configured",
```

```

    "displayName": "Primary Feeder",
    "linkedKey": null,
    "ports": []
    "configurations": []
    "tags": []
  }

```

Create the other hosts by using similar requests.

5.10 Adopting Alpha

Now that the configuration objects are set up, 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**.

You cannot adopt a switch by using a default profile because there are required settings like an IP address that must be set, so use the **an adopt with an previously created configuration node** endpoint. For this action, we need two parameters, the binding parameter, which is the operational ID, and the configuration ID.

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

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

In this tutorial, adopt Alpha by using the first method and Charlie by using the second method.

First, look for Alpha, the switch directly connected to the SEL-5056 and therefore the only detected switch. The SEL-2740S operational nodes have an **SelSapphireAttr (Sel.Sel5056.TopologyManager.Attributes-Operational.Node.SelSapphireAttr)** attribute that indicate the adopted status.

Therefore, find switches by retrieving all operational node objects and filtering out those without the SelSapphireAttr attribute. You can also use **OData filtering** to filtering on this attribute.

Key	Value
Endpoint	/api/default/operational/nodes
Method	GET
Content-Type	Not Present

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

Request

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

Reply

(This example shows only the item with the SelSapphireAttr attribute.)

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 4234

{
  "@odata.context": "https://localhost/api/default/operational/$metadata#nodes",
  "value": [
    {
      "id": "a2eae1505cc88479ba36f1c510ab09db",
      "displayName": "OpenFlow:00000030A744EB65",
      "state": "Unadopted",
      "trustState": "Untrusted",
      "linkedKey": null,
      "ports": ["a979c8a0e086a4b0f96cc2964c1004f5"],
      "attributes": [
        {
          "@odata.type": "#Sel.Sel5056.TopologyManager.Attributes.Operational.-
EthernetAttr",
          "macAddress": "0030A744EB65",
          "id": "0030A744EB65"
        }
      ]
    },
    {
      "@odata.type": "#Sel.Sel5056.TopologyManager.Attributes.Operational.-
EthernetAttr",
      "macAddress": "0030A744EBD3",
      "id": "0030A744EBD3"
    },
    {
      "@odata.type": "#Sel.Sel5056.TopologyManager.Attributes.Operational.-
Node.SelSapphireAttr",
      "nodeType": "Sel2740S",
      "deviceFramework": "BlueFrame",
      "autoDiscoveryVersion": "0.2",
```

```

    "communicationBand": "InBand",
    "dataPathId": 208964741989,
    "discoveryMac": "0030A744EB65",
    "discoveryPort": 12,
    "discoverySource": "Controller",
    "ethFrontIp": "169.254.188.75",
    "ethFrontMac": "0030A744EBD3",
    "ethRearIp": "192.168.1.100",
    "ethRearMac": "0030A744EB65",
    "failedEventQueryAttempts": 0,
    "id": "1234543210",
    "isCommissioned": false,
    "lastAutoDiscoveryReceived": "2021-11-23T15:27:49.4711074-05:00",
    "lastQueriedEventId": 0,
    "modelName": "SEL-2740S",
    "name": "SEL1234543210",
    "omniPort": 443,
    "prettyDataPathId": "00000030A744EB65",
    "serialNumber": "1234543210",
    "x509Fingerprint": "8AD36F8EA5EB24CE6E92D341AE72DD226074CFBE",
    "discoveryPortAttribute": {
      "@odata.type": "#Sel.Sel5056.TopologyManager.Attributes.Operational.-
EthernetAttr",
      "macAddress": "24F5A2F15003",
      "id": "24F5A2F15003"
    }
  }
]
}
]
}

```

Now that you have the switch operational ID, obtain the configuration ID. Use OData filters to quickly find the object with the displayName of Alpha.

Key	Value
Endpoint	/api/default/config/nodes
Method	GET
Content-Type	Not Present

Table 5.22: Header Values for /api/default/config/nodes

Key	Value
\$filter	displayName eq 'Alpha'

Table 5.23: Parameter Values for `/api/default/config/nodes`**Request**

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

Reply

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 4096

{
  "@odata.context": "https://localhost/api/default/config/$metadata#nodes",
  "value": [
    {
      "@odata.type": "#Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode",
      "id": "aa060ccb70b2b407aadca8f90f120ac0",
      "state": "Configured",
      "displayName": "Alpha",
      "linkedKey": null,
      "ports": ["a47065dc4f6804c6083a8b7b6ebd9e34"],
      "additionalTrustedCertificates": [],
      "alarmMinimumDuration": 1,
      "gigabitCopperFastFailover": false,
      "controllerIp": "192.168.1.1",
      "defaultGateway": "192.168.1.1",
      "enablePtp": false,
      "ipAddress": "192.168.1.100",
      "ntpServers": [],
      "subnetMask": "255.255.0.0",
      "timeSourceSelection": "None",
      "configurations": [
        {
          "@odata.type": "#Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-
OpenFlowConfig"
        }
      ],
      "tags": []
    }
  ]
}
```

```
"portLinkEventConfigurations":      [
  {
    "portId": "F",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "1",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "2",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "3",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "4",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
  {
    "portId": "5",
    "linkDownAutoClear": false,
    "linkDownSeverity": "Notice",
    "linkUpAutoClear": false,
    "linkUpSeverity": "Notice"
  }
}
```

```
{
  "portId": "6",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
  "portId": "7",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
  "portId": "8",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
  "portId": "9",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
  "portId": "10",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
  "portId": "11",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
```



```
"portId": "12",
"linkDownAutoClear": false,
"linkDownSeverity": "Notice",
"linkUpAutoClear": false,
"linkUpSeverity": "Notice"
}
{
  "portId": "13",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
  "portId": "14",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
  "portId": "15",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
  "portId": "16",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
  "portId": "17",
  "linkDownAutoClear": false,
  "linkDownSeverity": "Notice",
  "linkUpAutoClear": false,
  "linkUpSeverity": "Notice"
}
{
  "portId": "18",
```

```

        "linkDownAutoClear": false,
        "linkDownSeverity": "Notice",
        "linkUpAutoClear": false,
        "linkUpSeverity": "Notice"
    }
    {
        "portId": "19",
        "linkDownAutoClear": false,
        "linkDownSeverity": "Notice",
        "linkUpAutoClear": false,
        "linkUpSeverity": "Notice"
    }
    {
        "portId": "20",
        "linkDownAutoClear": false,
        "linkDownSeverity": "Notice",
        "linkUpAutoClear": false,
        "linkUpSeverity": "Notice"
    }
],
"eventCategories": [
    {
        "key": "default",
        "id": "default",
        "behaviors": [
            {
                "@odata.type": "#Sel.BlueFrame.Particle.AlarmContact.AlarmContact-
Behavior",
                "name": "default",
                "severity": "Error"
            }
            {
                "@odata.type": "#Sel.BlueFrame.Particle.LocalEventStore.LocalEvent-
StoreBehavior",
                "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": []
}
    ]
}
]
```

```
}

```

Now that you have both the operational ID and the configuration ID, adopt the switch.

Key	Value
Endpoint	/api/default/operational/nodes
Object id	a2eae1505cc88479ba36f1c510ab09db
Action	AdoptWithConfig
Method	POST
Content-Type	application/json

Table 5.24: Header Values for /api/default/operational/nodes

Request

You need one property for the JSON body.

Property	Value
configKey	aa060ccb70b2b407aadca8f90f120ac0

Table 5.25: Body Values for Adopting Alpha

```
POST /api/default/operational/nodes('a2eae1505cc88479ba36f1c510ab09db') -
/AdoptWithConfig HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 49

{
  "configKey": "aa060ccb70b2b407aadca8f90f120ac0"
}
```

Reply

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

5.11 Adopting Charlie

To adopt Charlie, you need the configuration ID of configuration node with the name Charlie. Use the same method that you used to find the configuration node with displayName Alpha. However, you cannot

adopt Charlie until Alpha is adopted and the SEL-5056 detects the LLDP packet from the switch. If you pull the Alpha configuration object, you can examine the SelSapphireAttr attribute to continuously check the state property until the value becomes Adopted.

Key	Value
Endpoint	/api/default/operational/nodes
Method	GET
Content-Type	Not Present

Table 5.26: Header Values for /api/default/operational/nodes

Key	Value
\$filter	attributes/any(t: isof(t, 'Sel.Sel5056.TopologyManager.Attributes-Operational.Node.SelSapphireAttr') and t/Sel.Sel5056.TopologyManager-Attributes.Operational.Node.SelSapphireAttr/prettyDataPathId eq '00000030A7141316')

Table 5.27: Parameter Values for /api/default/operational/nodes

Request

```
GET /api/default/operational/nodes?$filter=attributes/any(t%3A%20isof(t%2C%20-
'Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr')%20and-
%20t/Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr-
/prettyDataPathId%20eq%20'00000030A7141316') HTTP/1.1
HOST: localhost
Authorization: Bearer 000...000
```

Reply

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1701

{
  "@odata.context": "https://localhost/api/default/operational/$metadata#nodes",
  "value": [
    {
      "id": "a1ab9a6129922438fb008a0185ba75b9",
      "displayName": "OpenFlow:00000030A7141316",
      "state": "Unadopted",
      "trustState": "Untrusted",
      "linkedKey": null,
```

```

"ports":      ["a5fc9a3ab78d04ee1a523dcdbf8484ba"],
"attributes": [
  {
    "@odata.type": "#Sel.Sel5056.TopologyManager.Attributes.Operational.-
EthernetAttr",
    "macAddress": "0030A7141315",
    "id": "0030A7141315"
  }
  {
    "@odata.type": "#Sel.Sel5056.TopologyManager.Attributes.Operational.-
EthernetAttr",
    "macAddress": "0030A7141316",
    "id": "0030A7141316"
  }
  {
    "@odata.type": "#Sel.Sel5056.TopologyManager.Attributes.Operational.-
Node.SelSapphireAttr",
    "nodeType": "Sel2740S",
    "deviceFramework": "BlueFrame",
    "autoDiscoveryVersion": "0.2",
    "communicationBand": "InBand",
    "dataPathId": 208961540886,
    "discoveryMac": "0030A7141316",
    "discoveryPort": 12,
    "discoverySource": "OpenFlow",
    "ethFrontIp": "169.254.124.121",
    "ethFrontMac": "0030A7141315",
    "ethRearIp": "192.168.114.237",
    "ethRearMac": "0030A7141316",
    "failedEventQueryAttempts": 0,
    "id": "1170950626",
    "isCommissioned": false,
    "lastAutoDiscoveryReceived": "2021-11-23T15:47:55.3781606-05:00",
    "lastQueriedEventId": 0,
    "modelName": "SEL-2740S",
    "name": "SEL1170950626",
    "omniPort": 443,
    "prettyDataPathId": "00000030A7141316",
    "serialNumber": "1170950626",
    "x509Fingerprint": "65B3233288D8F27EE792B0D7F5D381DBA934EDCE",
    "discoveryPortAttribute": {
      "@odata.type": "#Sel.Sel5056.TopologyManager.Attributes.Operational.-
Port.OpenFlowPortAttr",

```

```

    "portId": 11,
    "dataPathId": 208964741989,
    "prettyDataPathId": "00000030A744EB65",
    "currentFeatures": 0,
    "ofState": "None",
    "currentSpeed": 0,
    "maxSpeed": 0,
    "name": null,
    "hardwareAddress": null,
    "peer": 0,
    "id": "00000030A744EB65:11",
    "advertised": []
    "config": []
    "supported": []
  }
}
]
}

```

For Charlie, use the datapath ID to first find the operational node that uses the **complex type filter**. Now, you can adopt Charlie.

Key	Value
Endpoint	/api/default/operational/nodes
Object id	a1ab9a6129922438fb008a0185ba75b9
Action	AdoptWithConfig
Method	POST
Content-Type	application/json

Table 5.28: Header Values for /api/default/operational/nodes

Request

You need one property for the JSON body.

Property	Value
configKey	a65eba3c53cff47c7a43468e368a34de

Table 5.29: Body Values for Adopting Charlie

```

POST /api/default/operational/nodes('a1ab9a6129922438fb008a0185ba75b9') -
/AdoptWithConfig HTTP/1.1

```

```

Content-Type:  application/json
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 49

{
  "configKey":  "a65eba3c53cff47c7a43468e368a34de"
}

```

Reply

```

HTTP/1.1 200 OK
Content-Length: 0

```

Now Charlie is also adopted. Finally adopt Bravo using one of the methods presented. Now all three switches are adopted.

5.12 Adopting Links

Adopting switches automatically adopts the links to form a path between the controller node and the adopted switch. The link between Bravo and Charlie is still unadopted. Because you can adopt a link with a default profile and you do not need to create a pre-defined profile, use **an adopt with an auto-generated configuration node** endpoint. However, first retrieve the ID of the operational link to adopt the link. The following are two of several methods for obtaining the ID of the operational link:

1. Start at Bravo or Charlie and then retrieve the port object D2 (which is the port to which that link is attached) and then get the object ID of the attached link.
2. Retrieve all of the links and then determine if the attached ports are D2 on Bravo or Charlie.

Perform the first method starting with Bravo. Use OData filtering to obtain the port in one query and then use the first entry of the attachedLinks array to obtain the link ID that you need.

Key	Value
Endpoint	/api/default/operational/ports
Method	GET
Content-Type	Not Present

Table 5.30: 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 '00000030A71306F1' and t/Sel.Sel5056- .TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr- /portId eq 11)
----------	---

Table 5.31: Parameter Values for /api/default/operational/ports

Request

```
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%20'00000030A71306F1'%20and%20t/Sel.Sel5056.TopologyManager-
.Attributes.Operational.Port.OpenFlowPortAttr/portId%20eq%2011) HTTP/1.1
HOST: localhost
Authorization: Bearer 000...000
```

Reply

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 2340

{
  "@odata.context": "https://localhost/api/default/operational/$metadata#ports",
  "value": [
    {
      "id": "a8fbf100a08924179b33e24f2d1a6a54",
      "displayName": "OpenFlow:00000030A71306F1:D3(11)",
      "isConnected": true
      "state": "Adopted",
      "trustState": "Internal",
      "linkedKey": "ae8724d432bca462fb4aa7c2f033329b",
      "parentNode": "aa5645d29b09a40a4aa2c3e4a1d6bf4f",
      "attachedLinks": ["a95f22885c9ff4d1ea5aaf39bde502ef"],
      "attributes": [
        {
          "@odata.type": "#Sel.Sel5056.TopologyManager.Attributes.Operational.-
Port.OpenFlowPortAttr",
          "portId": 11,
```

```

"dataPathId": 208961472241,
"prettyDataPathId": "00000030A71306F1",
"currentFeatures": 10272,
"ofState": "Live",
"currentSpeed": 1000000,
"maxSpeed": 1000000,
"name": "ethd-3",
"hardwareAddress": "0030A71306F1",
"peer": 0,
"id": "00000030A71306F1:11",
"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"
  }
  {
    "@odata.type": "#Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.OneHundredMegaBitFullDuplex"
  }
  {
    "@odata.type": "#Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.OneHundredMegaBitHalfDuplex"
  }
  {
    "@odata.type": "#Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.TenMegaBitFullDuplex"
  }
  {
    "@odata.type": "#Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.TenMegaBitHalfDuplex"
  }
],
"config": []
"supported": [
  {

```

```

        "@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"
    }
    {
        "@odata.type": "#Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.OneHundredMegaBitFullDuplex"
    }
    {
        "@odata.type": "#Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.OneHundredMegaBitHalfDuplex"
    }
    {
        "@odata.type": "#Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.TenMegaBitFullDuplex"
    }
    {
        "@odata.type": "#Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.TenMegaBitHalfDuplex"
    }
    ]
}
{
    "@odata.type": "#Sel.Sel5056.TopologyManager.Attributes.Operational.-
Port.Sel2740SPortAttr",
    "portType": "Sel2740S"
}
]
}

```

Now, you can adopt the link.

Key	Value
Endpoint	/api/default/operational/links
Object id	a95f22885c9ff4d1ea5aaf39bde502ef

Action	Adopt
Method	POST
Content-Type	Not Present

Table 5.32: Header Values for /api/default/operational/links

Request

```
POST /api/default/operational/links('a95f22885c9ff4d1ea5aaf39bde502ef')/Adopt-
HTTP/1.1
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 0
```

Reply

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

This completes the steps for adopting the switches and all the links between the switches. Follow the next step to adopt the hosts.

5.13 Adopting the Hosts Including Using Failover Mode

The network contains six devices. The controller device, representing the SEL-5056 host computer, is already adopted. Of the remaining five devices, two (Primary and Backup Feeder) have ports in Failover mode and the other three (RTAC, Primary Motor, and Backup Motor) are connected by a single link.

Use the AdoptWithConfig action to adopt a specific configuration node to an operational node. Assume that we have obtained the operational and the configuration node ID of the Primary Feeder.

Key	Value
Endpoint	/api/default/operational/nodes
Object id	ac739bf1888d04c0abd354325dac299b
Action	AdoptWithConfig
Method	POST
Content-Type	application/json

Table 5.33: Header Values for /api/default/operational/nodes

Request

You need one property for the JSON body.

Property	Value
configKey	a5938361690964371980261cdffa9e5d

Table 5.34: Body Values for Adopting the Hosts Including Using Failover Mode

```
POST /api/default/operational/nodes('ac739bf1888d04c0abd354325dac299b') -
/AdoptWithConfig HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 49

{
  "configKey": "a5938361690964371980261cdffa9e5d"
}
```

Reply

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

Next, adopt the link between the Primary Feeder and the switch. First, obtain the ID of the operational port from the returned ports property of the operational node definition.

Key	Value
Endpoint	/api/default/operational/ports
Object id	a1510b5efdf5149569f174e50ff12ee5
Method	GET
Content-Type	Not Present

Table 5.35: Header Values for /api/default/operational/ports

Request

```
GET /api/default/operational/ports('a1510b5efdf5149569f174e50ff12ee5') 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": "a1510b5efdf5149569f174e50ff12ee5",
  "displayName": "IP:192.168.1.150",
  "isConnected": true
  "state": "Unadopted",
  "trustState": "Untrusted",
  "linkedKey": null,
  "parentNode": "ac739bf1888d04c0abd354325dac299b",
  "attachedLinks": ["a0c8ad73d4eb6488ebea3151dbe87225"],
  "attributes": [
    {
      "@odata.type": "#Sel.Sel5056.TopologyManager.Attributes.Operational.Port.-
EthernetApplicationAttr",
      "ethType": 35000,
      "macAddress": "0030A714BBAC",
      "id": "35000:0030A714BBAC",
      "providerInformation": [
        {
          "vlan": 901,
          "destinationMac": "010CCD010011"
        }
      ]
    }
  ]
  {
    "@odata.type": "#Sel.Sel5056.TopologyManager.Attributes.Operational.-
EthernetAttr",
    "macAddress": "0030A714BBAB",
    "id": "0030A714BBAB"
  }
  {
    "@odata.type": "#Sel.Sel5056.TopologyManager.Attributes.Operational.Node.Ip-
Attr",
    "ipAddress": "192.168.1.150"
  }
]
}

```

Using the ID from the operational port, obtain the ID of the link from the attachedLinks property. Using

the operational link ID, adopt the link by using the Adopt action because you do not need to predefine a configuration link beforehand.

Key	Value
Endpoint	/api/default/operational/links
Object id	a0c8ad73d4eb6488ebea3151dbe87225
Action	Adopt
Method	POST
Content-Type	Not Present

Table 5.36: Header Values for /api/default/operational/links

Request

```
POST /api/default/operational/links('a0c8ad73d4eb6488ebea3151dbe87225')/Adopt-
HTTP/1.1
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 0
```

Reply

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

Using the operational port ID, mark the port as a failover port by using the MarkPortAsSelRelayFailover action.

Key	Value
Endpoint	/api/default/operational/ports
Object id	a1510b5efdf5149569f174e50ff12ee5
Action	MarkPortAsSelRelayFailover
Method	POST
Content-Type	Not Present

Table 5.37: Header Values for /api/default/operational/ports

Request

```
POST /api/default/operational/ports('a1510b5efdf5149569f174e50ff12ee5')-
/MarkPortAsSelRelayFailover HTTP/1.1
HOST: localhost
```

```
Authorization: Bearer 000...000
Content-Length: 0
```

Reply

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

To detect the other port, use the DetectSelRelayFailoverPort action. This is the same as if you had used the OpenFlow port modification feature to disable to the port, adopted the additional link that appeared, and then re-enabled the port by using the OpenFlow port modification feature.

Key	Value
Endpoint	/api/default/operational/ports
Object id	a1510b5efdf5149569f174e50ff12ee5
Action	DetectSelRelayFailoverPort
Method	POST
Content-Type	Not Present

Table 5.38: Header Values for /api/default/operational/ports

Request

```
POST /api/default/operational/ports('a1510b5efdf5149569f174e50ff12ee5') -
/DetectSelRelayFailoverPort HTTP/1.1
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 0
```

Reply

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

Using similar procedures, adopt the remaining hosts. After that, the network (switches, hosts, ports, and links) is completely adopted.

5.14 Creating Logical Connections

Now that the network is completely adopted, create the logical connections (LC) to program the switches to pass traffic. Start with the Network Time Protocol (NTP) LC between the Primary Feeder and RTAC. Because this is a unicast IP LC, add an Address Resolution Protocol (ARP) LC.

Like the other calls, you need the IDs of the nodes. How to find those IDs is not shown in this section. The `sourceEndPoints` list contains only the ID of the Primary Feeder configuration node, and the `destinationEndPoints` list contains the RTAC configuration node ID.

We will use one of the default CSTs with an ID of NTP Client.

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

Table 5.39: Header Values for /api/default/config/logicalConnections

Request

You need three properties for the JSON body.

Property	Value
sourceEndPoints	["a5938361690964371980261cdffa9e5d"]
destinationEndPoints	["acc99fa5d1d7a4259bca00bf9750b28f"]
communicationService- TypeId	NTP Client

Table 5.40: Body Values for Creating Logical Connections

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

{
  "sourceEndPoints": ["a5938361690964371980261cdffa9e5d"],
  "destinationEndPoints": ["acc99fa5d1d7a4259bca00bf9750b28f"],
  "communicationServiceTypeId": "NTP Client"
}
```

Reply

```

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 440

{
  "@odata.context": "https://localhost/api/default/config/$metadata-
#logicalConnections/$entity",
  "id": "a40bd19d2ef144db6ba00a9e5aad1d30",
  "enabled": true
  "lastEnabled": "2021-11-23T21:08:52.3902924Z",
  "sourceEndPoints": ["a5938361690964371980261cdffa9e5d"],
  "destinationEndPoints": ["acc99fa5d1d7a4259bca00bf9750b28f"],
  "communicationServiceTypeId": "NTP Client",
  "connectionOrigin": "EndUser",
  "errorState": "InProgress",
  "errors": []
  "version": 1,
  "tags": []
}

```

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

Table 5.41: Header Values for /api/default/config/logicalConnections

Request

You need three properties for the JSON body.

Property	Value
sourceEndPoints	["a5938361690964371980261cdffa9e5d"]
destinationEndPoints	["acc99fa5d1d7a4259bca00bf9750b28f"]
communicationService- TypeId	Bidirectional ARP

Table 5.42: Body Values for Creating Logical Connections

```

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

```

```
Content-Length: 170

{
  "sourceEndpoints": ["a5938361690964371980261cdffa9e5d"],
  "destinationEndpoints": ["acc99fa5d1d7a4259bca00bf9750b28f"],
  "communicationServiceTypeId": "Bidirectional ARP"
}
```

Reply

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 447

{
  "@odata.context": "https://localhost/api/default/config/$metadata-#logicalConnections/$entity",
  "id": "ab6e44d9ee21b428581f1928ca3aa742",
  "enabled": true
  "lastEnabled": "2021-11-23T21:08:52.6507977Z",
  "sourceEndpoints": ["a5938361690964371980261cdffa9e5d"],
  "destinationEndpoints": ["acc99fa5d1d7a4259bca00bf9750b28f"],
  "communicationServiceTypeId": "Bidirectional ARP",
  "connectionOrigin": "EndUser",
  "errorState": "InProgress",
  "errors": []
  "version": 1,
  "tags": []
}
```

Follow the same process for the other IP LCs.

5.15 Creating CSTs

For the GOOSE LCs, you cannot use the default CSTs. Instead, create your own specific CSTs. Create a CST with the match fields to match against the VID (VlanVid of 901) and the Ethertype (EthType of GOOSE).

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

Table 5.43: Header Values for /api/default/config/communicationServiceTypes

Request

You need four properties for the JSON body.

Property	Value
cstCommunicationType	Multicast
displayName	GOOSE 901
trafficMatch	<pre>{ "matchFields": [{ "@odata.type": "Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.MatchFields.VlanVid", "value": "901" } { "@odata.type": "Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.MatchFields.EthType", "value": "GOOSE" }] }</pre>
initialEnabledState	true

Table 5.44: 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: 329

{
  "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"
    }
]
}
"initialEnabledState": true
}

```

Reply

```

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 646

{
  "@odata.context": "https://localhost/api/default/config/$metadata-
#communicationServiceTypes/$entity",
  "id": "ac90e48385f0348cab6aebfde40619c6",
  "priority": 2000,
  "initialEnabledState": true
  "initialHardTimeout": 0,
  "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": null
      }
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.-
EthType",
        "value": "GOOSE"
      }
    ]
  }
  "setQueue": null,

```

```

    "autoFillSettings": null,
    "tags": []
  }

```

With this CST, create the corresponding LC.

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

Table 5.45: Header Values for /api/default/config/logicalConnections

Request

You need three properties for the JSON body.

Property	Value
sourceEndpoints	["a5938361690964371980261cdffa9e5d"]
destinationEndpoints	["a535d0f7a111b468d92a3020bc47cb8a"]
communicationService-TypeId	ac90e48385f0348cab6aebfde40619c6

Table 5.46: Body Values for Creating CSTs

```

POST /api/default/config/logicalConnections HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 185

{
  "sourceEndpoints": ["a5938361690964371980261cdffa9e5d"],
  "destinationEndpoints": ["a535d0f7a111b468d92a3020bc47cb8a"],
  "communicationServiceTypeId": "ac90e48385f0348cab6aebfde40619c6"
}

```

Reply

```

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 462

{

```

```

"@odata.context": "https://localhost/api/default/config/$metadata-
#logicalConnections/$entity",
"id": "ab219fc624c994c10ad730b9a60e107a",
"enabled": true
"lastEnabled": "2021-11-23T21:10:48.1684179Z",
"sourceEndPoints": ["a5938361690964371980261cdffa9e5d"],
"destinationEndPoints": ["a535d0f7a111b468d92a3020bc47cb8a"],
"communicationServiceTypeId": "ac90e48385f0348cab6aebfde40619c6",
"connectionOrigin": "EndUser",
"errorState": "InProgress",
"errors": []
"version": 1,
"tags": []
}

```

5.16 Adding a Virtual Host

Create virtual hosts by using the following three steps:

1. Create the configuration node object with the appropriate IP and MAC address attributes, as needed.
2. Obtain the operational ID of the switch port to which the virtual host will be attached.
3. Use the AddSilentHost action to bind the configuration node to the operational node.

This document has already covered obtaining the switch operational port IDs, so Step 2 will not be shown, but assume that the presented operational ID is for port C1(5). The IP address of the virtual host is 192.168.1.2.

First, create the configuration node with an ipAddress tag.

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

Table 5.47: Header Values for /api/default/config/nodes

Request

You need two properties for the JSON body.

Property	Value
displayName	Tap

tags	<pre>[{ "@odata.type": "#Sel.Sel5056.Tags.IpAddressTag", "ipAddress": "192.168.1.2" }]</pre>
------	--

Table 5.48: Body Values for Adding a Virtual 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": "a1cb2d9260f32484a96bc6a4cec2f471",
  "state": "Configured",
  "displayName": "Tap",
  "linkedKey": null,
  "ports": []
  "configurations": []
  "tags": [
    {
      "@odata.type": "#Sel.Sel5056.Tags.IpAddressTag",
```



```
    "ipAddress": "192.168.1.2",  
    "id": "192.168.1.2"  
  }  
]  
}
```

Second, obtain the ID of the operational port of the switch to which you want to add the virtual host (not shown) and then use the **AddSilentHost** action.

Key	Value
Endpoint	/api/default/operational/ports
Object id	a17d940ef60d84dec924eacd442c751d
Action	AddSilentHost
Method	POST
Content-Type	application/json

Table 5.49: Header Values for /api/default/operational/ports

Request

You need one property for the JSON body.

Property	Value
configNodeId	a1cb2d9260f32484a96bc6a4cec2f471

Table 5.50: Body Values for Adding a Virtual Host

```
POST /api/default/operational/ports('a17d940ef60d84dec924eacd442c751d') -  
/AddSilentHost HTTP/1.1  
Content-Type: application/json  
HOST: localhost  
Authorization: Bearer 000...000  
Content-Length: 52  
  
{  
  "configNodeId": "a1cb2d9260f32484a96bc6a4cec2f471"  
}
```

Reply

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

The virtual host is now present.

5.17 Managing Flow Entries

Now that you have explored using the automation provided by LCs and CSTs, proceed with managing OpenFlow entries directly. First, add a simple entry to send table-miss traffic from Alpha to the virtual device you added. To prevent the flow entry from matching normal traffic, put this flow entry into Table 3 with a priority of 100. This flow entry contains a single action (OutputAction within a write-actions instruction) to send the traffic out of Port 5 and no match fields.

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

Table 5.51: Header Values for /api/default/config/flows

Request

You need five properties for the JSON body.

Property	Value
displayName	Alpha Tap
node	aa060ccb70b2b407aadca8f90f120ac0
priority	100
tableId	3
instructions	[<pre> { "actions": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.OutputAction", "outPort": "5" }], "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.WriteActions" }]</pre>

Table 5.52: 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": "aa060ccb70b2b407aadca8f90f120ac0",
  "priority": 100,
  "tableId": 3,
  "instructions": [
    {
      "actions": [
        {
          "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
          "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": "a1a0290aa910d4133a5fb4625492f764",
  "outPort": 5,
  "outGroup": null,
  "inPort": null,
  "tableId": 3,
  "bufferId": 0,
  "hardTimeout": 0,
  "idleTimeout": 0,
  "noPacketCounts": false,
  "noByteCounts": false,
  "resetCounts": false,
  "checkOverlap": false,
  "priority": 100,
}
```

```

"lastEdit": "2021-11-23T16:13:53.7089984-05:00",
"errorState": "InProgress",
"errors": []
"enabled": true
"node": "aa060ccb70b2b407aadca8f90f120ac0",
"cookie": 1221,
"displayName": "Alpha Tap",
"version": 1,
"match": {
  "matchFields": []
}
"instructions": [
  {
    "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions",
    "actions": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
        "maxLength": 65535,
        "outPort": "5"
      }
    ]
  }
],
"tags": []
}

```

To delete the flow entry, use the DELETE method with the ID of the flow entry you want to delete.

Key	Value
Endpoint	/api/default/config/flows
Object id	a1a0290aa910d4133a5fb4625492f764
Method	DELETE
Content-Type	Not Present

Table 5.53: Header Values for /api/default/config/flows

Request

```

DELETE /api/default/config/flows('a1a0290aa910d4133a5fb4625492f764') HTTP/1.1
HOST: localhost
Authorization: Bearer 000...000

```

Reply

HTTP/1.1 204 No Content

Instead of using the `OutputAction`, use the `OutputActionByAlias` so that the output is tied to a configuration object instead of hard coded to a number. In this case, you need the ID of the configuration switch port (obtaining this ID is not shown).

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

Table 5.54: Header Values for /api/default/config/flows

Request

You need five properties for the JSON body.

Property	Value
displayName	Alpha Tap
node	aa060ccb70b2b407aadca8f90f120ac0
priority	100
tableId	3
instructions	[<pre> { "actions": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.OutputActionByAlias", "configPortId": "a323795aeec3c47668ee4d64012095b3" }], "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.WriteActions" }]</pre>

Table 5.55: 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: 347
```

```

{
  "displayName": "Alpha Tap",
  "node": "aa060ccb70b2b407aadca8f90f120ac0",
  "priority": 100,
  "tableId": 3,
  "instructions": [
    {
      "actions": [
        {
          "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
ActionByAlias",
          "configPortId": "a323795aeec3c47668ee4d64012095b3"
        }
      ],
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions"
    }
  ]
}

```

Reply

```

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 819

{
  "@odata.context": "https://localhost/api/default/config/$metadata#flows/-
$entity",
  "id": "aa2d542f3653543c1812fac9a146dfdb",
  "outPort": 5,
  "outGroup": null,
  "inPort": null,
  "tableId": 3,
  "bufferId": 0,
  "hardTimeout": 0,
  "idleTimeout": 0,
  "noPacketCounts": false,
  "noByteCounts": false,
  "resetCounts": false,
  "checkOverlap": false,
  "priority": 100,
  "lastEdit": "2021-11-23T16:13:53.8845265-05:00",
  "errorState": "InProgress",

```

```

"errors": [],
"enabled": true
"node": "aa060ccb70b2b407aadca8f90f120ac0",
"cookie": 1222,
"displayName": "Alpha Tap",
"version": 1,
"match": {
  "matchFields": []
}
"instructions": [
  {
    "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions",
    "actions": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
ActionByAlias",
        "maxLength": 65535,
        "outPort": "5",
        "configPortId": "a323795aeec3c47668ee4d64012095b3"
      }
    ]
  }
],
"tags": []
}

```

If you want to add a flow entry to allow a specific IP subnet to the controller for host discovery, add a flow entry with an Ipv4Src match field with the appropriate value and mask at a higher priority than the flow entry you just added.

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

Table 5.56: Header Values for /api/default/config/flows

Request

You need six properties for the JSON body.

Property	Value
displayName	192.168.0.0/16 to Controller
node	aa060ccb70b2b407aadca8f90f120ac0
priority	101

tableId	3
match	<pre>{ "matchFields": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.MatchFields.Ipv4Src", "value": "192.168.0.0", "mask": "255.255.0.0" }] }</pre>
instructions	<pre>[{ "actions": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.OutputAction", "outPort": "Controller" }], "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.WriteActions" }]</pre>

Table 5.57: 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: 491

{
  "displayName": "192.168.0.0/16 to Controller",
  "node": "aa060ccb70b2b407aadca8f90f120ac0",
  "priority": 101,
  "tableId": 3,
  "match": {
    "matchFields": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.-
Ipv4Src",
```



```

        "value": "192.168.0.0",
        "mask": "255.255.0.0"
    }
]
}
"instructions": [
    {
        "actions": [
            {
                "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
                "outPort": "Controller"
            }
        ],
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions"
    }
]
}

```

Reply

```

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 1052

{
  "@odata.context": "https://localhost/api/default/config/$metadata#flows/-
$entity",
  "id": "a4acb6ce1fe604ce6acfb5e3522e742f",
  "outPort": 4294967293,
  "outGroup": null,
  "inPort": null,
  "tableId": 3,
  "bufferId": 0,
  "hardTimeout": 0,
  "idleTimeout": 0,
  "noPacketCounts": false,
  "noByteCounts": false,
  "resetCounts": false,
  "checkOverlap": false,
  "priority": 101,
  "lastEdit": "2021-11-23T16:13:53.9913789-05:00",
  "errorState": "InProgress",

```

```

"errors": []
"enabled": true
"node": "aa060ccb70b2b407aadca8f90f120ac0",
"cookie": 1223,
"displayname": "192.168.0.0/16 to Controller",
"version": 1,
"match": {
  "matchFields": [
    {
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.-
Ipv4Src",
      "value": "192.168.0.0",
      "mask": "255.255.0.0",
      "translatedValue": "192.168.0.0"
    }
    {
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.-
EthType",
      "value": "IPv4"
    }
  ]
}
"instructions": [
  {
    "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions",
    "actions": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
        "maxLength": 65535,
        "outPort": "Controller"
      }
    ]
  }
],
"tags": []
}

```

If you want to disable the tap flow entry, use the PATCH method on the flow entry and set the enabled property to false.

Key	Value
Endpoint	/api/default/config/flows
Object id	aa2d542f3653543c1812fac9a146dfdb

Method	PATCH
Content-Type	application/json

Table 5.58: Header Values for `/api/default/config/flows`

Request

You need one property for the JSON body.

Property	Value
enabled	false

Table 5.59: Body Values for Managing Flow Entries

```
PATCH /api/default/config/flows('aa2d542f3653543c1812fac9a146dfdb') HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 18

{
  "enabled": false
}
```

Reply

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 820

{
  "@odata.context": "https://localhost/api/default/config/$metadata#flows/-
$entity",
  "id": "aa2d542f3653543c1812fac9a146dfdb",
  "outPort": 5,
  "outGroup": null,
  "inPort": null,
  "tableId": 3,
  "bufferId": 0,
  "hardTimeout": 0,
  "idleTimeout": 0,
  "noPacketCounts": false,
  "noByteCounts": false,
  "resetCounts": false,
```

```

"checkOverlap": false,
"priority": 100,
"lastEdit": "2021-11-23T16:13:54.0977748-05:00",
"errorState": "InProgress",
"errors": []
"enabled": false,
"node": "aa060ccb70b2b407aadca8f90f120ac0",
"cookie": 1222,
"displayName": "Alpha Tap",
"version": 1,
"match": {
  "matchFields": []
}
"instructions": [
  {
    "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions",
    "actions": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
ActionByAlias",
        "maxLength": 65535,
        "outPort": "5",
        "configPortId": "a323795aeec3c47668ee4d64012095b3"
      }
    ]
  }
],
"tags": []
}

```

5.18 Managing Group Entries

For groups, create a fast failover group that outputs to Port D2 if Port D2 is live, otherwise, output to Port D3. Also, create a flow entry to handle Sampled Values (SV) traffic ingressing into Port B1 and then egress the traffic by using the logic defined in the fast failover group you just created.

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

Table 5.60: Header Values for /api/default/config/groups

Request

You need five properties for the JSON body.

Property	Value
displayName	D2 then D3
node	aa060ccb70b2b407aadca8f90f120ac0
groupId	0
groupType	FastFailover
buckets	<pre>[{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.Bucket", "watchPort": "10", "actions": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.OutputAction", "outPort": "10" }] } { "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.Bucket", "watchPort": "11", "actions": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.OutputAction", "outPort": "11" }] }]</pre>

Table 5.61: 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: 523
```

```

{
  "displayName": "D2 then D3",
  "node": "aa060ccb70b2b407aadca8f90f120ac0",
  "groupId": 0,
  "groupType": "FastFailover",
  "buckets": [
    {
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket",
      "watchPort": "10",
      "actions": [
        {
          "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
          "outPort": "10"
        }
      ]
    }
    {
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket",
      "watchPort": "11",
      "actions": [
        {
          "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
          "outPort": "11"
        }
      ]
    }
  ]
}

```

Reply

```

HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 771

{
  "@odata.context": "https://localhost/api/default/config/$metadata#groups/-
$entity",
  "id": "adba39a8e291e48b3bb2ecce788c9ad4",
  "groupType": "FastFailover",
  "groupId": 0,

```

```

"errorState": "InProgress",
"errors": []
"node": "aa060ccb70b2b407aadca8f90f120ac0",
"enabled": true
"displayName": "D2 then D3",
"version": 1,
"buckets": [
  {
    "id": "ab38f9053d275467f95123ba7ffd3bf0",
    "outGroup": null,
    "outPort": 10,
    "watchGroup": "Any",
    "watchPort": "10",
    "actions": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
        "maxLength": 65535,
        "outPort": "10"
      }
    ]
  }
  {
    "id": "aadd061c8315545308144804015f9684",
    "outGroup": null,
    "outPort": 11,
    "watchGroup": "Any",
    "watchPort": "11",
    "actions": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
        "maxLength": 65535,
        "outPort": "11"
      }
    ]
  }
],
"tags": []
}

```

Now that the group is created, create a flow entry with the appropriate group action in the flow entry.

Key	Value
-----	-------

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

Table 5.62: Header Values for /api/default/config/flows

Request

You need six properties for the JSON body.

Property	Value
displayName	To Bravo then to Charlie
node	aa060ccb70b2b407aadca8f90f120ac0
priority	1000
tableId	0
match	<pre>{ "matchFields": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.MatchFields.InPort", "value": "1" } { "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.MatchFields.EthType", "value": "SV" }] }</pre>
instructions	<pre>[{ "actions": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.GroupAction", "groupId": 0 }], "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.WriteActions" }]</pre>

Table 5.63: Body Values for Managing Group Entries


```
POST /api/default/config/flows HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 541

{
  "displayName": "To Bravo then to Charlie",
  "node": "aa060ccb70b2b407aadca8f90f120ac0",
  "priority": 1000,
  "tableId": 0,
  "match": {
    "matchFields": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.-
InPort",
        "value": "1"
      }
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.-
EthType",
        "value": "SV"
      }
    ]
  }
  "instructions": [
    {
      "actions": [
        {
          "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group-
Action",
          "groupId": 0
        }
      ],
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions"
    }
  ]
}
```

Reply

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

Content-Length: 961

```
{
  "@odata.context": "https://localhost/api/default/config/$metadata#flows/-
$entity",
  "id": "a03da3572c31941549cff4c0610860b8",
  "outPort": null,
  "outGroup": 0,
  "inPort": 1,
  "tableId": 0,
  "bufferId": 0,
  "hardTimeout": 0,
  "idleTimeout": 0,
  "noPacketCounts": false,
  "noByteCounts": false,
  "resetCounts": false,
  "checkOverlap": false,
  "priority": 1000,
  "lastEdit": "2021-11-23T16:17:34.6123609-05:00",
  "errorState": "InProgress",
  "errors": []
  "enabled": true
  "node": "aa060ccb70b2b407aadca8f90f120ac0",
  "cookie": 1224,
  "displayName": "To Bravo then to Charlie",
  "version": 1,
  "match": {
    "matchFields": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.-
InPort",
        "value": "1",
        "translatedValue": 1
      }
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.-
EthType",
        "value": "SV"
      }
    ]
  }
  "instructions": [
    {
```

```

    "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions",
    "actions": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group-
Action",
        "groupId": 0
      }
    ]
  },
  "tags": []
}

```

5.19 Managing Meter Entries

For meters, create a meter for the tap connection with a rate of 100 and a burst size of 1000.

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

Table 5.64: Header Values for /api/default/config/meters

Request

You need five properties for the JSON body.

Property	Value
displayName	SV Meter
node	aa060ccb70b2b407aadca8f90f120ac0
meterId	1
flags	[{ "@odata.type": "#Sel.Sel5056.TopologyManager.OpenFlow.- Meter.RateTypeFlagObjects.PacketsPerSecond" }]

meterBands	<pre>[{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.MeterBandDrop", "rate": 100, "burstSize": 1000 }]</pre>
------------	---

Table 5.65: 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": "aa060ccb70b2b407aadca8f90f120ac0",
  "meterId": 1,
  "flags": [
    {
      "@odata.type": "#Sel.Sel5056.TopologyManager.OpenFlow.Meter.-
RateTypeFlagObjects.PacketsPerSecond"
    }
  ],
  "meterBands": [
    {
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandDrop",
      "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": "a6ee6bd9b6b994a0595c15976c89fc58",
"meterId": 1,
"errorState": "InProgress",
"errors": []
"enabled": true
"node": "aa060ccb70b2b407aadca8f90f120ac0",
"displayName": "SV Meter",
"version": 1,
"flags": [
{
"@odata.type": "#Sel.Sel5056.TopologyManager.OpenFlow.Meter.-
RateTypeFlagObjects.PacketsPerSecond"
}
],
"meterBands": [
{
"@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandDrop",
"burstSize": 1000,
"rate": 100
}
],
"tags": []
}

```

To attach the meter to flow entry, add in a meter instruction. Because you are editing the instructions property, include the current WriteActions definition in addition to the meter instruction.

Key	Value
Endpoint	/api/default/config/flows
Object id	aa2d542f3653543c1812fac9a146dfdb
Method	PATCH
Content-Type	application/json

Table 5.66: Header Values for /api/default/config/flows

Request

You need one property for the JSON body.

Property	Value
----------	-------

instructions	<pre>[{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.WriteActions", "actions": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.OutputActionByAlias", "maxLength": 65535, "outPort": "5", "configPortId": "a323795aee3c47668ee4d64012095b3" }] } { "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.MeterInstruction", "meterId": 1 }]</pre>
--------------	--

Table 5.67: Body Values for Managing Meter Entries

```
PATCH /api/default/config/flows('aa2d542f3653543c1812fac9a146dfdb') HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 375

{
  "instructions": [
    {
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions",
      "actions": [
        {
          "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
ActionByAlias",
          "maxLength": 65535,
          "outPort": "5",
          "configPortId": "a323795aee3c47668ee4d64012095b3"
        }
      ]
    }
  ]
}
```

```
{
  "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter-
Instruction",
  "meterId": 1
}
]
```

Reply

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 924

{
  "@odata.context": "https://localhost/api/default/config/$metadata#flows/-
$entity",
  "id": "aa2d542f3653543c1812fac9a146dfdb",
  "outPort": 5,
  "outGroup": null,
  "inPort": null,
  "tableId": 3,
  "bufferId": 0,
  "hardTimeout": 0,
  "idleTimeout": 0,
  "noPacketCounts": false,
  "noByteCounts": false,
  "resetCounts": false,
  "checkOverlap": false,
  "priority": 100,
  "lastEdit": "2021-11-23T16:23:46.4182713-05:00",
  "errorState": "InProgress",
  "errors": []
  "enabled": false,
  "node": "aa060ccb70b2b407aadca8f90f120ac0",
  "cookie": 1222,
  "displayName": "Alpha Tap",
  "version": 2,
  "match": {
    "matchFields": []
  }
  "instructions": [
    {
```

```
"@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions",
"actions": [
  {
    "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
ActionByAlias",
    "maxLength": 65535,
    "outPort": "5",
    "configPortId": "a323795aeec3c47668ee4d64012095b3"
  }
]
}
{
  "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter-
Instruction",
  "meterId": 1,
  "actions": []
}
],
"tags": []
}
```

This concludes the tutorial section.

Part II

REST Overview

Chapter 6

HTTP

6.1 HTTPS

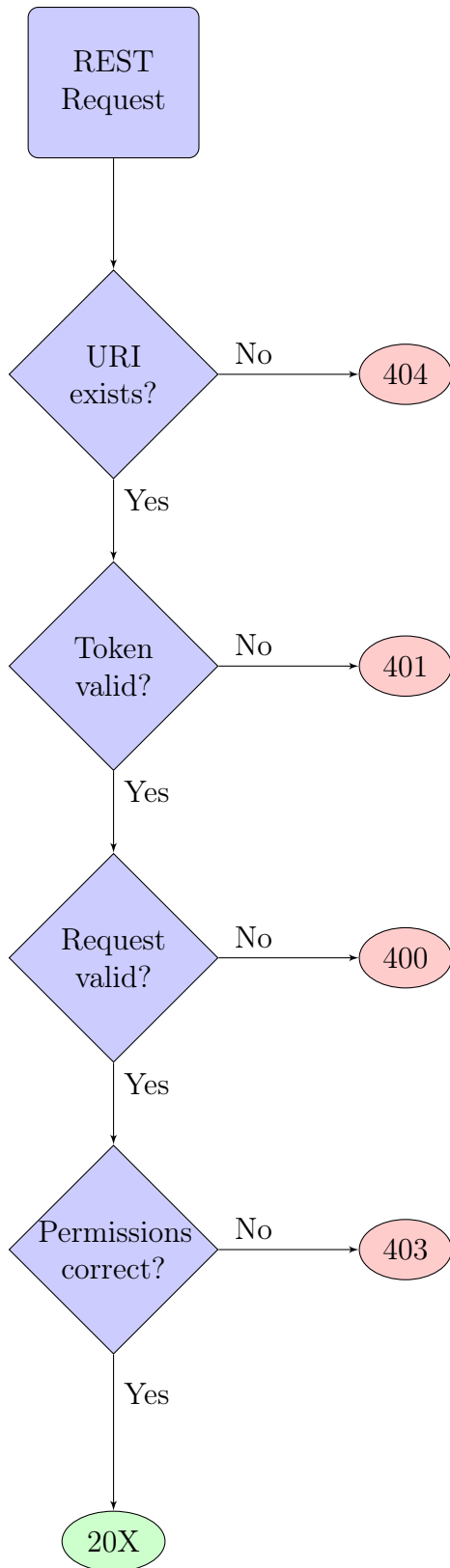
The SEL-5056 requires HTTP over TLS (HTTPS) connections to the REST interface.

6.1.1 Headers

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

6.1.2 Response Codes

There is a precedence in which the request is validated (relevant ones shown in the following figure) that might affect the HTTP code returned.



Deleting an object that does not exist returns 204.

6.1.3 Common Format of HTTP Methods

GET

The GET method has two types and three forms, as shown in the following table:

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

*You may include **OData filters** to filter the returned values.

Example

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

POST

Use the POST method for both adding objects to an entity set and calling an action. The POST method has four types and two forms, as shown in the following table:

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 the following 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/GetUserPreference	

PATCH or PUT

The following are the several methods used to modify an object, depending on the object:

- Use either the PATCH or PUT method on the object
- Use PUT only the object
- Use an action

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

DELETE

The DELETE method has two 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 7

OData

7.1 OData Version

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

OData-Version: 4.0

Other values return a 404 status code.

7.2 Service Roots

The SEL-5056 has five service roots:

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

Table 7.1: OData Service Roots

Refer to `/configuration/settings.json` on the REST interface for the list of available roots and the API version.

You can find the entities at each root by accessing `~/root/`, such as `/api/default/certificate/`.

Append `/$metadata` to the service root, such as `/api/default/config/$metadata`, to access the OData schema metadata that describes the endpoints.

7.2.1 Common Properties

Many of the endpoints share common properties, such as those listed in the following table.

Property	Description
displayName	Friendly name often used by the SEL-5056 UI
id	Unique key for the object, usually a UUID that is used to reference individual objects
errorState	State of the item, such as success or failure
errors	List of errors of the object, if in the error state

7.2.2 Filtering

The SEL-5056 through OData supports returning a subset 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.

Single Property With a Specific Value

Many endpoints contain **primitive type** properties (strings, numbers, boolean, or null) in the outermost level. These are filterable using the following form:

$\$filter = (property)(operator)(value)$

where $\langle property \rangle$ is the name of the property, $\langle operator \rangle$ is a valid OData filter operator, and $\langle value \rangle$ is the value to match.

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

To find all flows that are enabled (boolean), you can use the following:
`/api/default/config/flows?$filter=enabled eq true`

To find all flows that have an errorState of Success (string), you can use the following:
`/api/default/config/flows?$filter=errorState eq 'Success'`

Complex Type With a Specific Value

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 following form:

$\$filter = < complextype > /any(t : isof(t, ' < odatatype > ') and t / < odatatype > / < property > < operator > < value >)$

where $<complex type>$ is the OData type, $<property>$ is the name of the property within the complex type, and $<value>$ is the value to match against.

For example, to find all of the flows that belong to a logical connection with the ID abc...123, you can use the following:

`/api/default/config/flows?$filter=tags/any(t: isof(t, 'Sel.Sel5056.Tags.LogicalConnectionTag') and t/Sel.Sel5056.Tags.LogicalConnectionTag/logicalConnectionId eq 'abc...123')`

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

Chapter 8

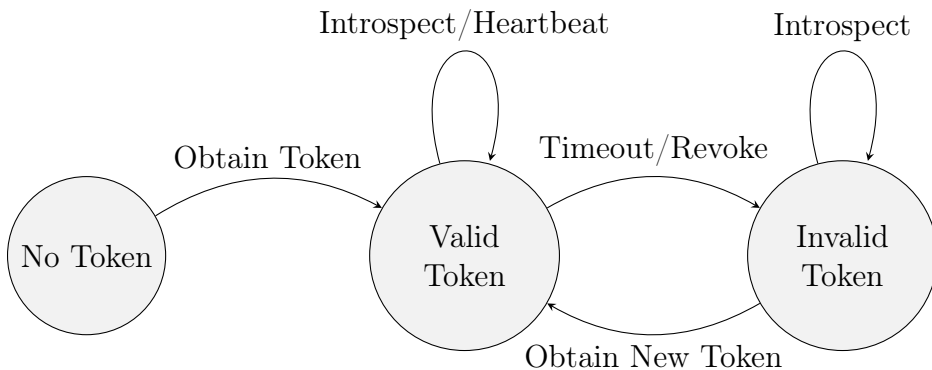
Authentication (OAuth)

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

This section assumes that you have the user 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.

The following figure shows an 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](#)).

8.1 Including Tokens in a Request

Tokens are of the Bearer type as defined in RFC 6750.

Pass the token to OData or SignalR endpoints by using the Authorization header if the endpoint requires permissions to access. For example, if the token received by the SEL-5056 is 929f5cf4d7a5863d8a665bbb0ac19194a6fdf25a346b1b1384eba4df83886577, then to make a call to get the PermissionLevel3 role, 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 929
f5cf4d7a5863d8a665bbb0ac19194a6fdf25a346b1b1384eba4df83886577
```

Listing 8.1: Using an OAuth Token

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

8.2 OAuth and Connect ID Parameters

8.2.1 Grant Types

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

Grant Type	Applicable Clients	Registered Application Required
Authorization Code	Applications	Yes
Implicit	(SEL-5056) UI	No
Resource Owner Password Credentials	Any	No
Client Credentials	Applications	Yes

Table 8.1: 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 8.3: 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

8.2.2 Client ID and Client Secret

Each grant type has a corresponding client ID and 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 8.5: OAuth Grant Types Client ID and Secret

8.2.3 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 8.5 for further information.

8.2.4 Roles

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

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

Encode the roles in the token request by using the `acr_values` parameter, such as `acr_values=role:PermissionLevel3`.

Applications do not have roles; instead, they have permissions.

8.3 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 8.7: OAuth/Token Endpoints

The Refresh token OAuth functionality is not supported. Use the [Heartbeat](#) functionality instead.

8.3.1 Obtaining a Token

URL: [/identity/connect/token](#)

Method: POST

Request Structure

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

Name	Valid Value(s)	Description
grant_type	(One of the available Grant Types)	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	Valid Value(s)	Description
access_token	Token	Bearer token used in HTTP requests that require a token
expires_in	600	Number of seconds that token is valid from point of issue
token_type	Bearer	OAuth token type

8.3.2 Resource Owner Password Credentials

Request Structure

Including the [general parameters](#), the following table lists the required parameters for this grant type:

Parameter	Valid Value(s)	OAuth Name
grant_type	(One of the available Grant Types)	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)	-
client_secret	Rest Interface	OAuth Client Secret

Table 8.10: Resource Owner Password Credentials Token Request Parameters

Resource Owner Password Credentials Example

In this example, the user is example, the password is "!qazxsw@", the role is Permission Level 3, and the token is c9e4fc629059422f25a63864003e185b.

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 8.2: Password Credentials Token (Normalized With Line Wrapping)

8.3.3 Client Credentials

Request Structure

Including the **general parameters**, the following table lists the required parameters for this grant type:

grant_type	(One of the available Grant Types)	OAuth Grant Type
client_id	password-client	OAuth Client ID
scope	Rest	OAuth Scope
state	(Any opaque string)	OAuth State

Table 8.11: 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 8.3: Client Credentials Token Request (Normalized With Line Wrapping)

8.3.4 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 8.4: Inspecting a Token

Revoking a Token

Example 4.

To revoke your token (i.e., logout), see the following example:

```
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 8.5: 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 8.6: Client Credentials Token Revocation (Normalized With Line Wrapping)

Client credential sessions do not count against the session limit.

8.3.5 Extending the Expiration Time of a Token

The following example 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 8.7: Client Credentials Token Revocation (Normalized With Line Wrapping)

8.4 Refreshing Token

The SEL-5056 does not support refresh tokens. You can use the Heartbeat endpoint to extend the expiration time of the token. This also resets the expiration time of the token to the original value from whenever the token is used.

8.5 Managing Permissions

The permissions of a client who uses the authorization grant are the intersection of the requested permissions at registration and the permissions of the role of the user authorizing the client. The per-

missions of a client who uses the client credentials grant are all of the permissions requested at registration.

User permissions are covered in the [User Permissions](#) section.

8.6 Troubleshooting

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

```
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
```

8.6.1 Registered Applications

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

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

If you receive a 401 Unauthorized response when making any REST call after obtaining a token, ensure the following:

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 exchange when using the `/security/applicationLinks/InitiateApplicationRegistration/` endpoint to register is uploaded and not revoked.
3. You did not register the application with an `ipAddress` of 127.0.0.1.
4. The IP address that the SEL-5056 uses 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 general features. The OAuth endpoints are covered in [Authentication \(OAuth\)](#) and the SignalR endpoints are covered [Asynchronous Data Exchange With SignalR](#).

Each chapter contains an introduction followed by sections for each group of endpoints. For example, [User Management](#) covers user management, which includes the `/security/users` endpoints.

The section divides the endpoints for each group into different purposes; the most basic purpose relates to CRUD (create, read, update, and delete). Other purposes include diagnostics or other actions.

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 type(s) and the JSON body of each return type are listed. Refer to [OData Definitions](#) for more details regarding each property of the JSON.

Methods are carried by HTTP. The following are the five methods covered in this document.

- GET
- POST
- PATCH
- PUT
- DELETE

See [HTTP](#) for further information on formatting URIs.

Chapter 9

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)

Manage the first two users through **User Accounts**. Manage the latter through **Authentication Services**. LDAP is the only supported authentication service.

9.1 User Accounts

Purpose	Endpoints	Methods
Read	<code>~/security/users/</code>	GET
	<code>~/security/users('id')/</code>	GET
Create	<code>~/security/users/</code>	POST
Update	<code>~/security/users('id')/</code>	PUT
	<code>~/security/users('id')/</code>	PATCH
	<code>~/security/users('id')/UpdatePassword</code>	POST
Delete	<code>~/security/users('id')/</code>	DELETE

9.1.1 Read Users

URL: `/api/default/security/users/`

Method: GET

OData Type: `Sel.BlueFrame.Core.SecurityManager.User`

There are two types that can be returned:

- `Sel.BlueFrame.Core.SecurityManager.LocalUser`
- `Sel.BlueFrame.Core.SecurityManager.AuthServiceUser`

The response for this request contains a list of these types.

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

```
{
  "@odata.type": "#Sel.BlueFrame.Core.SecurityManager.LocalUser,
  "clearTextPass": Edm.String,
  "passwordCreationTime": Edm.DateTimeOffset,
  "passwordStore": [ Sel.BlueFrame.Core.SecurityManager.Password, ],
  "displayName": Edm.String,
  "enabled": Edm.Boolean,
  "failedLogins": [ Sel.BlueFrame.Core.SecurityManager.UserFailedLogin, ],
  "id": Edm.String,
  "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,
  "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,
}
```

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

9.1.2 Create a User

URL: /api/default/security/users/

Method: POST

OData Type: `Sel.BlueFrame.Core.SecurityManager.User`

There are two types that support the POST command:

- `Sel.BlueFrame.Core.SecurityManager.LocalUser`
- `Sel.BlueFrame.Core.SecurityManager.AuthServiceUser`

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

Property	Default Value
enabled	true
passwordCreationTime	0001-01-01T00:00:00Z
passwordStore	[]
roles	[]

Table 9.2: Default Values for `Sel.BlueFrame.Core.SecurityManager.LocalUser`

```
{
  "@odata.type": "#Sel.BlueFrame.Core.SecurityManager.LocalUser",
  "clearTextPass": Edm.String,
  "passwordCreationTime": Edm.DateTimeOffset,
  "passwordStore": [ Sel.BlueFrame.Core.SecurityManager.Password, ],
  "enabled": Edm.Boolean,
  "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
roles	[]

Table 9.3: 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,
  "roles": [ Edm.String, ] (Sel.BlueFrame.Core.SecurityManager.Role.id),
  "username": Edm.String,
}
```

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

Examples

Create a Permission Level 3 User Example 7.

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

Table 9.4: Header Values for /api/default/security/users/

Request

You need four properties for the JSON body.

Property	Value
@odata.type	#Sel.BlueFrame.Core.SecurityManager.LocalUser

roles	["PermissionLevel3"]
username	pl3
clearTextPass	1Example\$

Table 9.5: Body Values for Create 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: 144

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

```

9.1.3 Update a User (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 two types 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 previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their **default value**.

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

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their **default value**.

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

9.1.4 Update a User (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 two types that can be modified:

- `Sel.BlueFrame.Core.SecurityManager.LocalUser`
- `Sel.BlueFrame.Core.SecurityManager.AuthServiceUser`

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

- enabled
- isLockedOut
- lastSuccessLoginIp
- passwordCreationTime
- passwordStore
- roles
- username

The following properties are patchable for `Sel.BlueFrame.Core.SecurityManager.AuthServiceUser`:

- authService
- displayName
- enabled
- isLockedOut
- lastSuccessLoginIp
- roles
- username

Available User Role 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"
}
```

Examples**Disable a User Example 8.**

Key	Value
Endpoint	/api/default/security/users
Object id	ab2b727602f24461a96d726ee6b9abf5
Method	PATCH
Content-Type	application/json

Table 9.6: Header Values for /api/default/security/users

Request

You need one property for the JSON body.

Property	Value
enabled	False

Table 9.7: Body Values for Disable a User

```
PATCH /api/default/security/users('ab2b727602f24461a96d726ee6b9abf5') HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 20
```

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

9.1.5 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 is one type that supports the POST command:

- `Sel.UpdatePassword`

Request Structure for `Sel.UpdatePassword`

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

Available User Roles With Required Permissions

- Monitor
- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

This endpoint is for users only.

9.1.6 Delete a User

URL: `/api/default/security/users('id')/`

Method: DELETE

odata Type: `Sel.BlueFrame.Core.SecurityManager.User`

Binding parameter (id): `Sel.BlueFrame.Core.SecurityManager.User.id`

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

9.2 User Roles

Purpose	Endpoints	Methods
Read	~/security/roles/	GET
	~/security/roles('id')/	GET
Create	Not supported	
Update	Not supported	
Delete	Not supported	

9.2.1 Read User Roles

URL: [/api/default/security/roles/](#)

Method: GET

OData Type: [Sel.BlueFrame.Core.SecurityManager.Role](#)

There is one type that can be returned:

- [Sel.BlueFrame.Core.SecurityManager.Role](#)

The response for this request contains a list of this type.

Response Structure for [Sel.BlueFrame.Core.SecurityManager.Role](#)

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

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

9.3 Authentication Services

Purpose	Endpoints	Methods
Read	<code>~/security/authServices/</code>	GET
	<code>~/security/authServices('id')/</code>	GET
Create	<code>~/security/authServices/</code>	POST
Update	<code>~/security/authServices('id')/</code>	PUT
	<code>~/security/authServices('id')/</code>	PATCH
Delete	<code>~/security/authServices('id')/</code>	DELETE
Test	<code>~/security/authServices('id')/TestAuthenticateUser</code>	POST

9.3.1 Read Authentication Services

URL: `/api/default/security/authServices/`

Method: GET

OData Type: `Sel.BlueFrame.Core.SecurityManager.AuthService`

There is one type that can be returned:

- `Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService`

The response for this request contains a list of this type.

Response Structure 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,
}
```

```

"protocolVersion": Edm.Int32,
"searchBase": Edm.String,
"userIdFilter": Edm.String,
"workPhone": Edm.String,
"errorState": Sel.BlueFrame.Core.ErrorState,
"errors": [ Edm.String, ],
"id": Edm.String,
}

```

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```

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

```

9.3.2 Create an Authentication Service

URL: /api/default/security/authServices/

Method: POST

OData Type: **Sel.BlueFrame.Core.SecurityManager.AuthService**

There is one type that supports the POST command:

- **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 9.10: 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,
}
```

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

9.3.3 Update an Authentication Service (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 is one type 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 previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their **default value**.

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

9.3.4 Update an Authentication Service (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 is one type that can be modified:

- **Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService**

The following properties are patchable for Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService:

- bindDN
- bindPassword
- email
- firstName
- groupMemberAttribute
- hostname
- lastName
- name
- portNumber

- searchBase
- userIdFilter
- workPhone

Available User Role 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"
}
```

9.3.5 Delete an Authentication Service

URL: /api/default/security/authServices('id')/

Method: DELETE

OData Type: [Sel.BlueFrame.Core.SecurityManager.AuthService](#)

Binding parameter (id): [Sel.BlueFrame.Core.SecurityManager.AuthService.id](#)

Available User Role 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"
}
```

9.3.6 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 is one type that supports the POST command:

- **Sel.TestAuthenticateUser**

Request Structure for Sel.TestAuthenticateUser

No request body

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

9.4 Authentication Service Groups

Purpose	Endpoints	Methods
Read	~/security/authServiceGroups/	GET
	~/security/authServiceGroups('id')/	GET
Create	~/security/authServiceGroups/	POST
Update	~/security/authServiceGroups('id')/	PUT
	~/security/authServiceGroups('id')/	PATCH
Delete	~/security/authServiceGroups('id')/	DELETE
	~/security/authServiceGroups('id')/	DELETE

9.4.1 Read Authentication Service Groups

URL: /api/default/security/authServiceGroups/

Method: GET

OData Type: `Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup`

There is one type that can be returned:

- `Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup`

The response for this request contains a list of this type.

Response Structure 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),
  "id":           Edm.String,
}
```

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

9.4.2 Create an Authentication Service Group

URL: /api/default/security/authServiceGroups/

Method: POST

OData Type: `Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup`

There is one type that supports the POST command:

- `Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup`

Request Structure 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),
  "id":  Edm.String,
}
```

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

9.4.3 Update an Authentication Service Group (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 is one type that can be modified:

- Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup

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

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their default value.

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

9.4.4 Update an Authentication Service Group (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 is one type that can be modified:

- `Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup`

The following properties are patchable for `Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup`:

- `displayName`
- `groupRoles`

Available User Role 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"
}
```

9.4.5 Delete an Authentication Service Group

URL: /api/default/security/authServiceGroups('id')/

Method: DELETE

OData Type: **Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup**

Binding parameter (id): **Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup.id**

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

9.5 Preferences

Purpose	Endpoints	Methods
Read	~/settings/GetUserPreference	POST
	~/settings/preferences/	GET
Create	~/settings/SetUserPreference	POST
Delete	Not supported	
Update	~/settings/SetUserPreference	POST

9.5.1 Get the Value of a User Preference

URL: /api/default/settings/GetUserPreference

Method: POST

OData Type: **Sel.GetUserPreference**

There is one type that supports the POST command:

- **Sel.GetUserPreference**

Request Structure for Sel.GetUserPreference

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

Available User Roles With Required Permissions

- Monitor
- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

This endpoint is for users only.

9.5.2 Read User Preferences

URL: /api/default/settings/preferences/

Method: GET

OData Type: `Sel.BlueFrame.UserPreferences.Preference`

There is one type that can be returned:

- `Sel.BlueFrame.UserPreferences.Preference`

The response for this request contains a list of this type.

Response Structure for `Sel.BlueFrame.UserPreferences.Preference`

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

Available User Roles With Required Permissions

- Monitor
- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

This endpoint is for users only.

9.5.3 Add or Modify a User Preference

URL: /api/default/settings/SetUserPreference

Method: POST

OData Type: **Sel.SetUserPreference**

There is one type that supports the POST command:

- **Sel.SetUserPreference**

Request Structure for Sel.SetUserPreference

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

Available User Roles With Required Permissions

- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

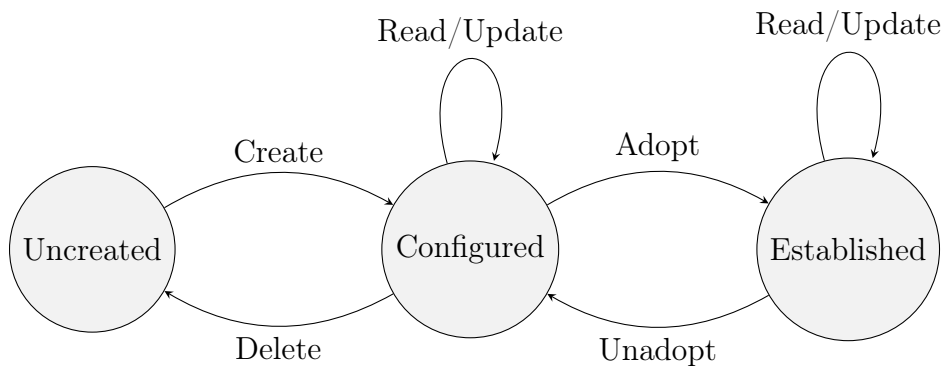
This endpoint is for users only.

Chapter 10

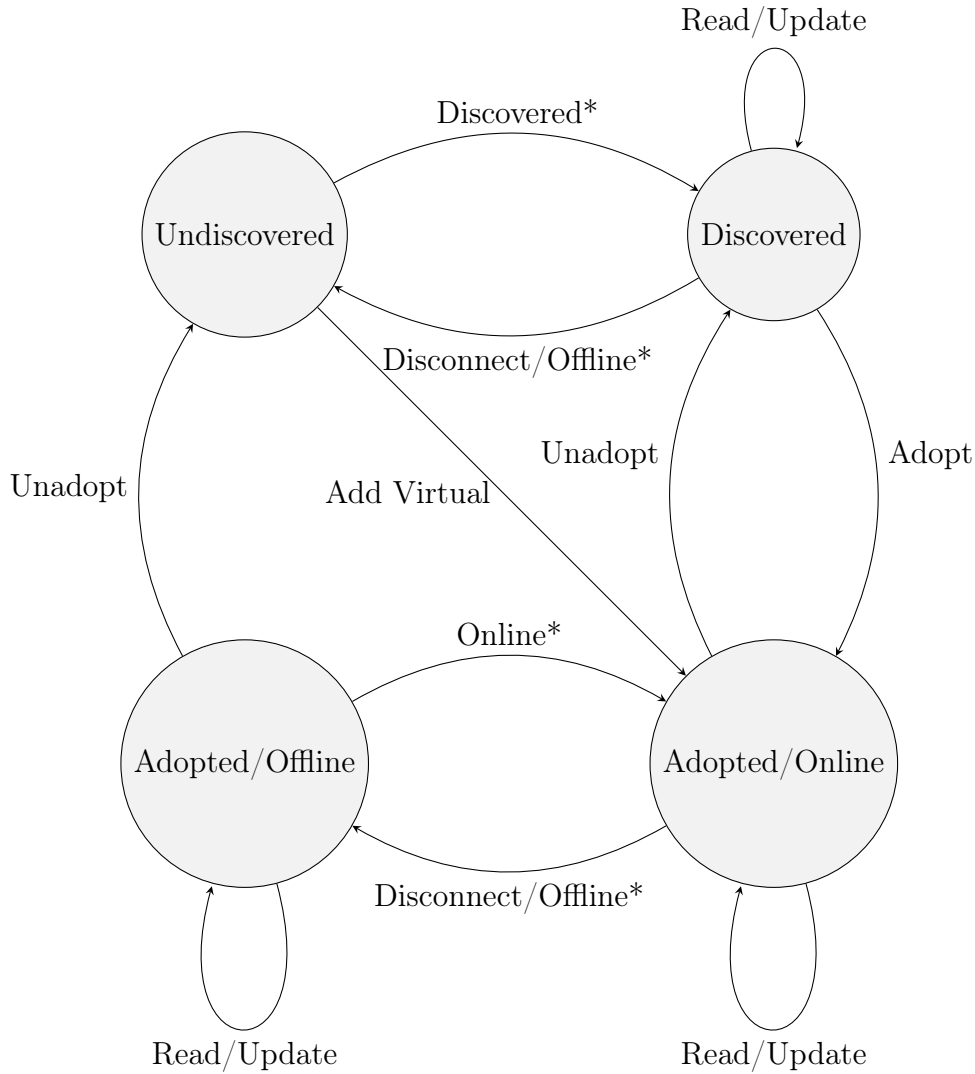
Topology Management

There are two types of objects in a topology: operational and configuration. Operational objects are detected by the SEL-5056 through one of the discovery processes. Configuration objects contain the abstract settings that are independent of the operational object. Configuration objects apply 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 in one of three states.



There are three types of objects:

- Nodes
- Ports
- Links

A node may have one or more ports, which can be connected to one or more links. These links are then connected to the port on the other side, which belongs to a node. Therefore, nodes are connected by the following chain of objects: a node to its port to its link to the other port to the other 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

10.1 Nodes

Nodes may be one of four types:

- Controller, type of Generic node representing the SEL-5056 host machine
- NetworkAbstraction, type of Generic node representing switches not controlled by SEL-5056
- SEL-2740S, representing the SEL-2740S,
- SEL-2742S, representing the SEL-2742S,
- Other Generic, representing all other nodes, including end devices and other types of OpenFlow switches.

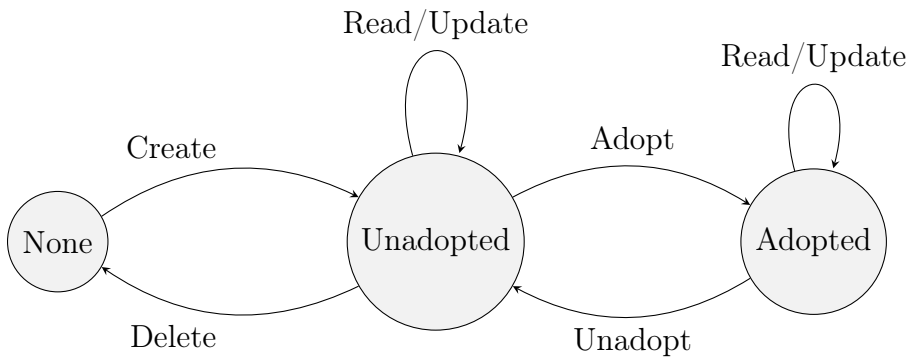
Controller and NetworkAbstraction are both subtypes of Generic, determined by a tag within the data structure. SEL-2740S is a derived type of Generic node.

Identifying the Node Type

Type	Method to Identify
SEL-2740S	@odata.type is Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode
SEL-2742S	@odata.type is Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode
NetworkAbstraction	Configurations property contains { "@odata.type": "#Sel.Sel5056.-TopologyManager.Configurations.Abstraction.AbstractionConfig", "abstraction-Setting": { "@odata.type": "#Sel.Sel5056.TopologyManager.Configurations.-Abstraction.TraditionalSwitch" } }
Controller	Configurations property contains { "@odata.type": "#Sel.Sel5056.TopologyManager.Configurations.ControllerConfig" }
Other Generic	All other nodes

Purpose	Endpoints	Methods
Read	~/config/nodes/	GET
	~/config/nodes('id')/	GET
	~/operational/nodes/	GET
	~/operational/nodes('id')/	GET
Create	~/config/nodes/	POST
Update	~/config/nodes('id')/	PUT
	~/config/nodes('id')/	PATCH
	~/operational/MergeNodes	POST
	~/operational/nodes('id')/ReplaceConfig	POST
Delete	~/config/nodes('id')/	DELETE
Disconnect or Remove	~/operational/nodes('id')/Disconnect	POST
Adopt	~/operational/nodes('id')/Adopt	POST

	<code>~/operational/nodes('id')/AdoptWithConfig</code>	POST
Unadopt	<code>~/operational/nodes('id')/Unadopt</code>	POST
Search	<code>~/operational/UserInitiatedHostDiscovery</code>	POST
Manage ARP	<code>~/operational/nodes('id')/EnableControllerArpToNode</code>	POST
	<code>~/operational/nodes('id')/DisableControllerArpToNode</code>	POST



10.1.1 Read Configuration Nodes

URL: `/api/default/config/nodes/`

Method: GET

odata Type: `Sel.Sel5056.TopologyManager.Nodes.ConfigNode`

There are three types that can be returned:

- `Sel.Sel5056.TopologyManager.Nodes.ConfigNode`
- `Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode`
- `Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode`

The response for this request contains a list of these types.

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.OperationalNetwork-
Node.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 Thumbprint),
  "alarmMinimumDuration": Edm.Int64,
  "controllerIp": Edm.String (IpAddress),
  "defaultGateway": Edm.String (IpAddress),
  "enablePtp": Edm.Boolean,
  "eventCategories": [ Sel.BlueFrame.Core.EventBus.EventCategory, ],
  "gigabitCopperFastFailover": Edm.Boolean,
  "ipAddress": Edm.String,
  "ntpServers": [ Edm.String, ] (ipv4address),
  "portLinkEventConfigurations": [ Sel.Sel5056.TopologyManager.Nodes.PortLinkEvent-
Configuration, ],
  "subnetMask": Edm.String (IpAddress),
  "timeSourceSelection": Sel.Sel5056.TopologyManager.Enums.Sel274xTimeSource,
  "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, ],
}
```

Response Structure for Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode

```
{
  "@odata.type": "#Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode,
  "digitalInput1Function": Sel.Sel5056.TopologyManager.Enums.Sel2742SDigitalInput-
Functions,
  "digitalInput2Function": Sel.Sel5056.TopologyManager.Enums.Sel2742SDigitalInput-
Functions,
  "enablePinHoleReset": Edm.Boolean,
  "enablePoePort5": Edm.Boolean,
  "enablePoePort6": Edm.Boolean,
  "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"
}

```

10.1.2 Read Operational Nodes

URL: /api/default/operational/nodes/

Method: GET

OData Type: `Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode`

There is one type that can be returned:

- `Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode`

The response for this request contains a list of this type.

Response Structure for `Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode`

```

{
  "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"
}
```

10.1.3 Create a Configuration Node

URL: /api/default/config/nodes/

Method: POST

OData Type: *Sel.Sel5056.TopologyManager.Nodes.ConfigNode*

There are three types that support the POST command:

- *Sel.Sel5056.TopologyManager.Nodes.ConfigNode*
- *Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode*
- *Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode*

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

Property	Default Value
configurations	[]
displayName	null
ports	[]
tags	[]

Table 10.4: 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
additionalTrusted-Certificates	[]
alarmMinimumDuration	1
configurations	[]
displayName	null
enablePtp	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" }]
gigabitCopperFastFailover	false
ntpServers	[]

tags	[]
timeSourceSelection	None

Table 10.5: Default Values for Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

```
{
  "@odata.type": "#Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode,
  "additionalTrustedCertificates": [ Edm.String, ] (X.509 Thumbprint),
  "alarmMinimumDuration": Edm.Int64,
  "controllerIp": Edm.String (IpAddress),
  "defaultGateway": Edm.String (IpAddress),
  "enablePtp": Edm.Boolean,
  "eventCategories": [ Sel.BlueFrame.Core.EventBus.EventCategory, ],
  "gigabitCopperFastFailover": Edm.Boolean,
  "ipAddress": Edm.String,
  "ntpServers": [ Edm.String, ] (ipv4address),
  "portLinkEventConfigurations": [ Sel.Sel5056.TopologyManager.Nodes.PortLinkEvent-
  Configuration, ],
  "subnetMask": Edm.String (IpAddress),
  "timeSourceSelection": Sel.Sel5056.TopologyManager.Enums.Sel274xTimeSource,
  "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.Sel2742SConfigNode

Property	Default Value
configurations	[]
digitalInput1Function	Disabled
digitalInput2Function	Disabled
displayName	null
enablePinHoleReset	true
enablePoePort5	false
enablePoePort6	false
ports	[]
tags	[]

Table 10.6: Default Values for Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode

```
{
  "@odata.type": "#Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode,
```



```

    "digitalInput1Function": Sel.Sel5056.TopologyManager.Enums.Sel2742SDigitalInput-
    Functions,
    "digitalInput2Function": Sel.Sel5056.TopologyManager.Enums.Sel2742SDigitalInput-
    Functions,
    "enablePinHoleReset": Edm.Boolean,
    "enablePoePort5": Edm.Boolean,
    "enablePoePort6": Edm.Boolean,
    "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 Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```

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

```

10.1.4 Update a Configuration Node (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 three types that can be modified:

- Sel.Sel5056.TopologyManager.Nodes.ConfigNode
- Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode
- Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode

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

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their **default value**.

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

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their **default value**.

Request Structure for Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their **default value**.

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.1.5 Update a Configuration Node (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 three types that can be modified:

- **Sel.Sel5056.TopologyManager.Nodes.ConfigNode**
- **Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode**
- **Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode**

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

- configurations
- displayName
- ports
- tags

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

- additionalTrustedCertificates
- alarmMinimumDuration
- configurations
- controllerIp
- defaultGateway
- displayName
- enablePtp
- eventCategories
- gigabitCopperFastFailover
- ipAddress
- ntpServers
- portLinkEventConfigurations
- ports
- subnetMask
- tags
- timeSourceSelection

The following properties are patchable for Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode:

- additionalTrustedCertificates
- alarmMinimumDuration
- configurations
- controllerIp
- defaultGateway
- digitalInput1Function
- digitalInput2Function
- displayName
- enablePinHoleReset

- enablePoePort5
- enablePoePort6
- enablePtp
- eventCategories
- gigabitCopperFastFailover
- ipAddress
- ntpServers
- portLinkEventConfigurations
- ports
- subnetMask
- tags
- timeSourceSelection

Available User Role 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"
}
```

10.1.6 Merge Two Nodes by Combining the Ports of One into the Other

URL: /api/default/operational/MergeNodes

Method: POST

OData Type: [Sel.MergeNodes](#)

There is one type that supports the POST command:

- [Sel.MergeNodes](#)

Request Structure for Sel.MergeNodes

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

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.1.7 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 is one type that supports the POST command:

- [Sel.ReplaceConfig](#)

Request Structure for Sel.ReplaceConfig

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

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.1.8 Delete a Configuration Node

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 Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.1.9 Remove or Disconnect 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 is one type that supports the POST command:

- [Sel.Disconnect](#)

Request Structure for Sel.Disconnect

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.1.10 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 is one type that supports the POST command:

- **Sel.Adopt**

Request Structure for Sel.Adopt

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.1.11 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 is one type that supports the POST command:

- [Sel.AdoptWithConfig](#)

Request Structure for Sel.AdoptWithConfig

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

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.1.12 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 is one type that supports the POST command:

- [Sel.Unadopt](#)

Request Structure for Sel.Unadopt

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.1.13 Run Host Discovery for a Particular IP Address

URL: /api/default/operational/UserInitiatedHostDiscovery

Method: POST

OData Type: [Sel.UserInitiatedHostDiscovery](#)

There is one type that supports the POST command:

- [Sel.UserInitiatedHostDiscovery](#)

Request Structure for Sel.UserInitiatedHostDiscovery

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

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.1.14 Allow the SEL-5056 to Send ARPs to a Node

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

Method: POST

OData Type: [Sel.EnableControllerArpToNode](#)

Binding parameter (id): [Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id](#)

There is one type that supports the POST command:

- [Sel.EnableControllerArpToNode](#)

Request Structure for Sel.EnableControllerArpToNode

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.1.15 Prevent the SEL-5056 From Sending ARPs to a Node

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

Method: POST

OData Type: [Sel.DisableControllerArpToNode](#)

Binding parameter (id): [Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id](#)

There is one type that supports the POST command:

- [Sel.DisableControllerArpToNode](#)

Request Structure for Sel.DisableControllerArpToNode

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

```

```

    "tree": "OperationalTree"
}

```

10.2 OpenFlow Switches

Purpose	Endpoints	Methods
Create Virtual Host	~/operational/ports('id')/AddSilentHost	POST
Get Diagnostics	~/operational/tableStats/	GET
	~/operational/tableStats('id')/	GET
Synchronize	~/operational/GetNodesThatRequireSynchronization	POST
	~/operational/synchronization/	GET
	~/operational/SynchronizeConfiguration	POST
	~/operational/SynchronizeAllConfiguration	POST
	~/operational/SynchronizeSelectedConfiguration	POST
	~/operational/AuditDeviceConfiguration	POST

10.2.1 Add a Virtual Host to a Switch Port

URL: [/api/default/operational/ports\('id'\)/AddSilentHost](#)

Method: POST

OData Type: [Sel.AddSilentHost](#)

Binding parameter (id): [Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id](#)

There is one type that supports the POST command:

- [Sel.AddSilentHost](#)

Request Structure for Sel.AddSilentHost

```

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

```

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```

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

```

```

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

```

10.2.2 Read Flow Table Diagnostics

URL: /api/default/operational/tableStats/

Method: GET

OData Type: `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats`

There is one type that can be returned:

- `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats`

The response for this request contains a list of this type.

Response Structure for `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats`

```

{
  "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"
}

```

10.2.3 Get Nodes that Are Unsynchronized

URL: /api/default/operational/GetNodesThatRequireSynchronization

Method: POST

OData Type: [Sel.GetNodesThatRequireSynchronization](#)

There is one type that supports the POST command:

- [Sel.GetNodesThatRequireSynchronization](#)

Request Structure for Sel.GetNodesThatRequireSynchronization

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"
}
```

10.2.4 Read Switches to Synchronize

URL: /api/default/operational/synchronization/

Method: GET

OData Type: [Sel.Sel5056.TopologyManager.Synchronization.SynchronizationRequest](#)

There are thirteen types 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.OpenFlowPlugin.PortModController.PortModSynchronization`
- `Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization`
- `Sel.Sel5056.SapphirePlugin.Synchronization.FirmwareUpgradeRequiredSynchronization`

The response for this request contains a list of these types.

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.Flow.cookie),
  "dependencies":  [ Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlow-
Dependency, ],
  "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.Flow.cookie),
  "dependencies": [ Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlow-
Dependency, ],
  "priorityConfiguration": Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-
SystemPriorities.AmSystemPriority,
  "referencedGroupId": Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
Group.groupId),
  "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.Flow.cookie),
  "currentGroupId": Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
Group.groupId),
  "dependencies": [ Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlow-
Dependency, ],
  "newGroupId": Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.groupId),
  "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.OpenFlow-
Dependency, ],
  "groupId": Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.groupId),
  "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.OpenFlow-
Dependency, ],
  "groupId": Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.groupId),
  "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.-
Group.groupId),
  "dependencies": [ Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlow-
Dependency, ],
  "groupId": Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.groupId),
  "newGroupIds": [ Edm.Int64, ] (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
Group.groupId),
  "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.OpenFlow-
Dependency, ],
  "meterId": Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.meterId),
  "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.OpenFlow-
Dependency, ],
  "meterId": Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.meterId),
  "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.OpenFlow-
Dependency, ],
  "meterId": Edm.Int64 (Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.meterId),
  "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.PortModController.PortModSynchronization

```
{
  "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.PortModController.-
PortModSynchronization,
  "configSettings": [ Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.-
ConfigBase, ],
  "featureSettings": [ Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.FeatureBase, ],
  "operationalPortId": 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.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"
}

```

10.2.5 Synchronize OpenFlow Switch

URL: /api/default/operational/SynchronizeConfiguration

Method: POST

OData Type: [Sel.SynchronizeConfiguration](#)

There is one type that supports the POST command:

- [Sel.SynchronizeConfiguration](#)

Request Structure for Sel.SynchronizeConfiguration

```

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

```

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```

{
  "module": "Synchronization",
  "permissionTag": "PerformSynchronization"
}

```

10.2.6 Synchronize All OpenFlow Switches

URL: /api/default/operational/SynchronizeAllConfiguration

Method: POST

OData Type: [Sel.SynchronizeAllConfiguration](#)

There is one type that supports the POST command:

- [Sel.SynchronizeAllConfiguration](#)

Request Structure for Sel.SynchronizeAllConfiguration

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{  
  "module": "Synchronization",  
  "permissionTag": "PerformSynchronization"  
}
```

10.2.7 Synchronize OpenFlow Switch(es)

URL: /api/default/operational/SynchronizeSelectedConfiguration

Method: POST

OData Type: [Sel.SynchronizeSelectedConfiguration](#)

There is one type that supports the POST command:

- [Sel.SynchronizeSelectedConfiguration](#)

Request Structure for Sel.SynchronizeSelectedConfiguration

```
{  
  "configNodeIds": [ Edm.String, ],  
}
```

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{  
  "module": "Synchronization",
```

```
"permissionTag": "PerformSynchronization"
}
```

10.2.8 Check Synchronization Status of Switch

URL: /api/default/operational/AuditDeviceConfiguration

Method: POST

OData Type: **Sel.AuditDeviceConfiguration**

There is one type that supports the POST command:

- **Sel.AuditDeviceConfiguration**

Request Structure for Sel.AuditDeviceConfiguration

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

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
  "module": "Synchronization",
  "permissionTag": "PerformSynchronization"
}
```

10.3 SEL-2740S and SEL-2742S Switches

Purpose	Endpoints	Methods
Factory Default Reset	~/operational/nodes('id')/FactoryDefaultReset	POST
Firmware Up-date	~/operational/nodes('id')/FirmwareUpgrade	POST
Reboot	~/operational/nodes('id')/Reboot	POST
Replan In-band LC	~/operational/nodes('id')/ReplanInbandPath	POST
Get SNMP Users	~/security/snmpUsers/	GET
	~/security/snmpUsers('id')/	GET
Update SNMP Users	~/security/snmpUsers('id')/	PATCH

10.3.1 Factory-Default Reset an 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 is one type that supports the POST command:

- [Sel.FactoryDefaultReset](#)

Request Structure for Sel.FactoryDefaultReset

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.3.2 Firmware Upgrade an 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 is one type that supports the POST command:

- [Sel.FirmwareUpgrade](#)


```

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

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

```

10.3.3 Restart an SEL-2740S or SEL-2742S

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

Method: POST

OData Type: **Sel.Reboot**

Binding parameter (id): **Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id**

There is one type that supports the POST command:

- **Sel.Reboot**

Request Structure for Sel.Reboot

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```

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

```

```

    "module": "DeviceManagement",
    "permissionTag": "Reboot"
}

```

10.3.4 Replan a Switch IB Management Logical Connection

See [Replan a Switch IB Management Logical Connection](#).

10.3.5 Read SEL-2740S and SEL-2742S SNMP Users

URL: /api/default/security/snmpUsers/

Method: GET

odata Type: [Sel.Sel5056.Snmp.SnmpUser](#)

There is one type that can be returned:

- [Sel.Sel5056.Snmp.SnmpUser](#)

The response for this request contains a list of this type.

Response Structure for Sel.Sel5056.Snmp.SnmpUser

```

{
  "authenticationMethod": Sel.Sel5056.Snmp.SnmpAuthenticationMethod,
  "authenticationPassword": Edm.String,
  "encryptionMethod": Sel.Sel5056.Snmp.SnmpEncryptionMethod,
  "encryptionPassword": Edm.String,
  "id": Edm.String,
  "username": Edm.String,
}

```

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```

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

```

10.3.6 Update a SEL-2740S and SEL-2742S SNMP User (PATCH)

URL: /api/default/security/snmpUsers('id')/

Method: PATCH

OData Type: **Sel.Sel5056.Snmp.SnmpUser**

Binding parameter (id): **Sel.Sel5056.Snmp.SnmpUser.id**

There is one type that can be modified:

- **Sel.Sel5056.Snmp.SnmpUser**

The following properties are patchable for Sel.Sel5056.Snmp.SnmpUser:

- authenticationMethod
- authenticationPassword
- encryptionPassword

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

10.4 Traditional Switches

Purpose	Endpoints	Methods
Create	~/operational/ports('id')/AddAbstractNode	POST

10.4.1 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 is one type that supports the POST command:

- `Sel.AddAbstractNode`

Request Structure for `Sel.AddAbstractNode`

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

Available User Role With Required Permissions

- `PermissionLevel3`

Permissions for Registered Applications

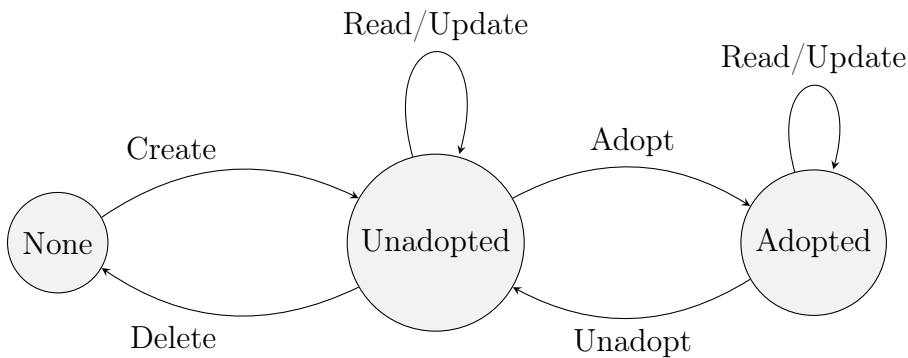
```
{
  "module": "TopologyManager",
  "permissionTag": "TopologyDiscoveryHinting"
},
{
  "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
  "dataType": "OperationalNetworkPort",
  "module": "DataBroker",
  "permissionTag": "Read",
  "tree": "OperationalTree"
}
```

10.5 Ports

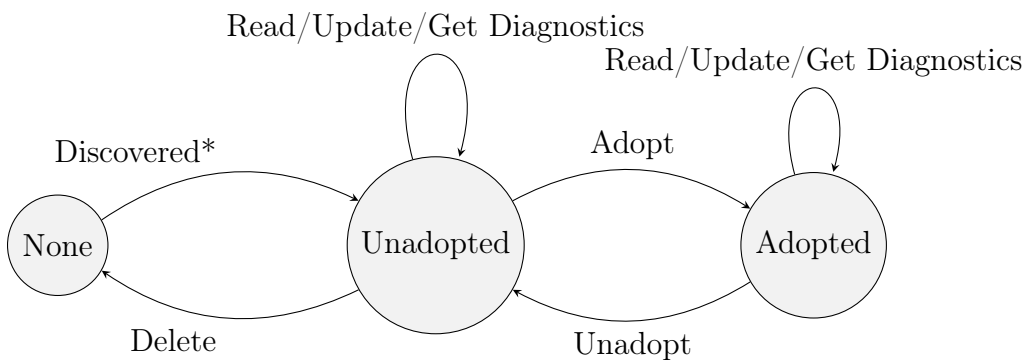
Purpose	Endpoints	Methods
Read	<code>~/config/ports/</code>	GET
	<code>~/config/ports('id')/</code>	GET
	<code>~/operational/ports/</code>	GET
	<code>~/operational/ports('id')/</code>	GET
Create	<code>~/config/ports/</code>	POST

Update	~/config/ports('id')/	PUT
	~/config/ports('id')/	PATCH
	~/operational/ports('id')/ReplaceConfig	POST
Delete	~/config/ports('id')/	DELETE
Adopt	~/operational/ports('id')/Adopt	POST
	~/operational/ports('id')/AdoptWithConfig	POST
Unadopt	~/operational/ports('id')/Unadopt	POST
Manage Failover Mode	~/operational/ports('id')/DetectSelRelayFailoverPort	POST
	~/operational/ports('id')/MarkPortAsSelRelayFailover	POST
	~/operational/ports('id')/MarkPortAndAddRelayFailoverLink	POST
	~/operational/ports('id')/MarkPortAndPerformBlockingDetectSelRelay-Failover	POST
	~/operational/ports('id')/RemoveSelRelayFailoverFromPort	POST

Configuration Ports



Operational Ports



10.5.1 Read Configuration Ports

URL: [/api/default/config/ports/](#)

Method: GET

odata Type: [Sel.Sel5056.TopologyManager.Ports.ConfigPort](#)

There is one type that can be returned:

- [Sel.Sel5056.TopologyManager.Ports.ConfigPort](#)

The response for this request contains a list of this type.

Response Structure for `Sel.Sel5056.TopologyManager.Ports.ConfigPort`

```
{
  "attachedLinks": [ Edm.String, ] (Sel.Sel5056.TopologyManager.Links.ConfigLink),
  "configurations": [ Sel.Sel5056.TopologyManager.Configurations.AmConfig, ],
  "displayName": Edm.String,
  "displayNameSetByUser": Edm.Boolean,
  "id": Edm.String,
  "linkedKey": Edm.String (Sel.Sel5056.TopologyManager.Ports.OperationalNetwork-
Port.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"
}
```

10.5.2 Read Operational Ports

URL: `/api/default/operational/ports/`

Method: GET

OData Type: `Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort`

There is one type that can be returned:

- `Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort`

The response for this request contains a list of this type.

Response Structure for Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

```
{
  "attachedLinks": [ Edm.String, ] (Sel.Sel5056.TopologyManager.Links.Operational-
NetworkLink),
  "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.OperationalNetworkNode),
  "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"
}
```

10.5.3 Create a Configuration Port

URL: /api/default/config/ports/

Method: POST

OData Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort

There is one type that supports the POST command:

- Sel.Sel5056.TopologyManager.Ports.ConfigPort

Request Structure for Sel.Sel5056.TopologyManager.Ports.ConfigPort

Property	Default Value
attachedLinks	[]
configurations	[]

displayName	null
tags	[]

Table 10.12: Default Values for Sel.Sel5056.TopologyManager.Ports.ConfigPort

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

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.5.4 Update a Configuration Port (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 is one type that can be modified:

- Sel.Sel5056.TopologyManager.Ports.ConfigPort

Request Structure for Sel.Sel5056.TopologyManager.Ports.ConfigPort

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their **default value**.

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.5.5 Update a Configuration Port (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 is one type that can be modified:

- [Sel.Sel5056.TopologyManager.Ports.ConfigPort](#)

The following properties are patchable for [Sel.Sel5056.TopologyManager.Ports.ConfigPort](#):

- attachedLinks
- configurations
- displayName
- displayNameSetByUser
- parentNode
- tags

Available User Role 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"
}

```

10.5.6 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 is one type that supports the POST command:

- [Sel.ReplaceConfig](#)

Request Structure for Sel.ReplaceConfig

```

{
  "configKey": Edm.String,
}

```

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```

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

```

10.5.7 Delete a Configuration Port

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 Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.5.8 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 is one type that supports the POST command:

- [Sel.Adopt](#)

Request Structure for Sel.Adopt

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.5.9 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 is one type that supports the POST command:

- [Sel.AdoptWithConfig](#)

Request Structure for Sel.AdoptWithConfig

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

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.5.10 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 is one type that supports the POST command:

- **Sel.Unadopt**

Request Structure for Sel.Unadopt

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.5.11 Detect the Other Port of a Device in Failover Mode

URL: /api/default/operational/ports('id')/DetectSelRelayFailoverPort

Method: POST

OData Type: [Sel.DetectSelRelayFailoverPort](#)

Binding parameter (id): [Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id](#)

There is one type that supports the POST command:

- [Sel.DetectSelRelayFailoverPort](#)

Request Structure for Sel.DetectSelRelayFailoverPort

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.5.12 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 is one type that supports the POST command:

- [Sel.MarkPortAsSelRelayFailover](#)

Request Structure for Sel.MarkPortAsSelRelayFailover

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.5.13 Add SEL Relay Failover Mode to a Port and Create an Adopted Operational Link From a Node to a Switch

URL: /api/default/operational/ports('id')/MarkPortAndAddRelayFailoverLink

Method: POST

OData Type: [Sel.MarkPortAndAddRelayFailoverLink](#)

Binding parameter (id): [Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id](#)

There is one type that supports the POST command:

- [Sel.MarkPortAndAddRelayFailoverLink](#)

Request Structure for Sel.MarkPortAndAddRelayFailoverLink

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

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.5.14 Detect the Other Port of a Device in Failover Mode and Block 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 is one type that supports the POST command:

- [Sel.MarkPortAndPerformBlockingDetectSelRelayFailover](#)

Request Structure for Sel.MarkPortAndPerformBlockingDetectSelRelayFailover

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.5.15 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 is one type that supports the POST command:

- [Sel.RemoveSelRelayFailoverFromPort](#)

Request Structure for Sel.RemoveSelRelayFailoverFromPort

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.6 OpenFlow Ports

Purpose	Endpoints	Methods
Get Descriptions	~/operational/portDesc/	GET
	~/operational/portDesc('id')/	GET
Get Diagnostics	~/operational/portStats/	GET
	~/operational/portStats('id')/	GET
Update	~/config/ports('id')/SendPortMod	POST
	~/operational/ports('id')/SendPortMod	POST

10.6.1 Read OpenFlow Port Descriptions

URL: [/api/default/operational/portDesc/](#)

Method: GET

OData Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc](#)

There is one type that can be returned:

- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc](#)

The response for this request contains a list of this type.

Response Structure for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc

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


Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.6.2 Read OpenFlow Port Diagnostics

URL: /api/default/operational/portStats/

Method: GET

OData Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats](#)

There is one type that can be returned:

- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats](#)

The response for this request contains a list of this type.

Response Structure for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats

```
{
  "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"
}
```

10.6.3 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 is one type that supports the POST command:

- [Sel.SendPortMod](#)

Request Structure for Sel.SendPortMod

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

Available User Role With Required Permissions

- [PermissionLevel3](#)

Permissions for Registered Applications

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

10.6.4 Modify the OpenFlow Settings of a Port

URL: /api/default/operational/ports('id')/SendPortMod

Method: POST

OData Type: [Sel.SendPortMod](#)

Binding parameter (id): [Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id](#)

There is one type that supports the POST command:

- [Sel.SendPortMod](#)

Request Structure for Sel.SendPortMod

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

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.7 Links

Purpose	Endpoints	Methods
Read	~/config/links/	GET
	~/config/links('id')/	GET
	~/operational/links/	GET
	~/operational/links('id')/	GET
Create	~/config/links/	POST
	~/operational/ports('id')/AddOperationalLinkFromHostToSwitch	POST
Update	~/config/links('id')/	PUT
	~/config/links('id')/	PATCH
	~/operational/links('id')/ReplaceConfig	POST
Delete	~/config/links('id')/	DELETE
Adopt	~/operational/links('id')/Adopt	POST
	~/operational/links('id')/AdoptWithConfig	POST
Unadopt	~/operational/links('id')/Unadopt	POST

10.7.1 Read Configuration Links

URL: /api/default/config/links/

Method: GET

OData Type: `Sel.Sel5056.TopologyManager.Links.ConfigLink`

There is one type that can be returned:

- `Sel.Sel5056.TopologyManager.Links.ConfigLink`

The response for this request contains a list of this type.

Response Structure 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),
  "id": Edm.String,
  "last": Edm.String (Sel.Sel5056.TopologyManager.Ports.ConfigPort),
  "linkedKey": Edm.String (Sel.Sel5056.TopologyManager.Links.OperationalNetwork-
Link.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"
}
```

10.7.2 Read Operational Links

URL: /api/default/operational/links/

Method: GET

OData Type: `Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink`

There is one type that can be returned:

- `Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink`

The response for this request contains a list of this type.

Response Structure for `Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink`

```
{
  "attributes": [ Sel.Sel5056.TopologyManager.Attributes.AmAttribute, ],
  "displayName": Edm.String,
  "endPoints": [ Edm.String, ] (Sel.Sel5056.TopologyManager.Ports.OperationalNetwork-
Port),
  "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"
}
```

10.7.3 Create a Configuration Link

URL: `/api/default/config/links/`

Method: POST

OData Type: `Sel.Sel5056.TopologyManager.Links.ConfigLink`

There is one type that supports the POST command:

- `Sel.Sel5056.TopologyManager.Links.ConfigLink`

Request Structure for Sel.Sel5056.TopologyManager.Links.ConfigLink

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

Table 10.15: 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 Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.7.4 Create an Adopted Operational Link From a Node to a Switch

URL: /api/default/operational/ports('id')/AddOperationalLinkFromHostToSwitch

Method: POST

OData Type: Sel.AddOperationalLinkFromHostToSwitch

Binding parameter (id): Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.id

There is one type that supports the POST command:

- Sel.AddOperationalLinkFromHostToSwitch

Request Structure for Sel.AddOperationalLinkFromHostToSwitch

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

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.7.5 Update a Configuration Link (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 is one type that can be modified:

- [Sel.Sel5056.TopologyManager.Links.ConfigLink](#)

Request Structure for Sel.Sel5056.TopologyManager.Links.ConfigLink

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their **default value**.

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.7.6 Update a Configuration Link (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 is one type that can be modified:

- `Sel.Sel5056.TopologyManager.Links.ConfigLink`

The following properties are patchable for `Sel.Sel5056.TopologyManager.Links.ConfigLink`:

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

Available User Role 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"
}
```


10.7.7 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 is one type that supports the POST command:

- [Sel.ReplaceConfig](#)

Request Structure for Sel.ReplaceConfig

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

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.7.8 Delete a Configuration Link

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 Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.7.9 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 is one type that supports the POST command:

- **Sel.Adopt**

Request Structure for Sel.Adopt

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.7.10 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 is one type that supports the POST command:

- **Sel.AdoptWithConfig**

Request Structure for Sel.AdoptWithConfig

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

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.7.11 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 is one type that supports the POST command:

- **Sel.Unadopt**

Request Structure for Sel.Unadopt

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.8 Default Adoption Settings

Purpose	Endpoints	Methods
Read	~/settings/networkSettings/	GET
	~/settings/networkSettings('id')/	GET
Update	~/settings/networkSettings('id')/	PUT
	~/settings/networkSettings('id')/	PATCH

10.8.1 Read Default Adoption Settings and Default ARP SPA Addresses

URL: /api/default/settings/networkSettings/

Method: GET

OData Type: **Sel.Sel5056.TopologyManager.NetworkSettings**

There is one type that can be returned:

- `Sel.Sel5056.TopologyManager.NetworkSettings`

The response for this request contains a list of this type.

Response Structure for `Sel.Sel5056.TopologyManager.NetworkSettings`

```
{
  "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"
}
```

10.8.2 Update Default Adoption Settings and Default ARP SPA Addresses (PUT)

URL: `/api/default/settings/networkSettings('id')/`

Method: PUT

OData Type: `Sel.Sel5056.TopologyManager.NetworkSettings`

Binding parameter (id): `Sel.Sel5056.TopologyManager.NetworkSettings.id`

There is one type that can be modified:

- `Sel.Sel5056.TopologyManager.NetworkSettings`

Request Structure for Sel.Sel5056.TopologyManager.NetworkSettings

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their default value.

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

10.8.3 Update Default Adoption Settings and Default ARP SPA Addresses (PATCH)

URL: /api/default/settings/networkSettings('id')/

Method: PATCH

OData Type: [Sel.Sel5056.TopologyManager.NetworkSettings](#)

Binding parameter (id): [Sel.Sel5056.TopologyManager.NetworkSettings.id](#)

There is one type that can be modified:

- [Sel.Sel5056.TopologyManager.NetworkSettings](#)

The following properties are patchable for Sel.Sel5056.TopologyManager.NetworkSettings:

- controllerArpSrcIpAddress
- controllerArpSrcNetworkAddresses
- controllerIp
- defaultGateway
- ntpServers

Available User Role 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"
}
```

Chapter 11

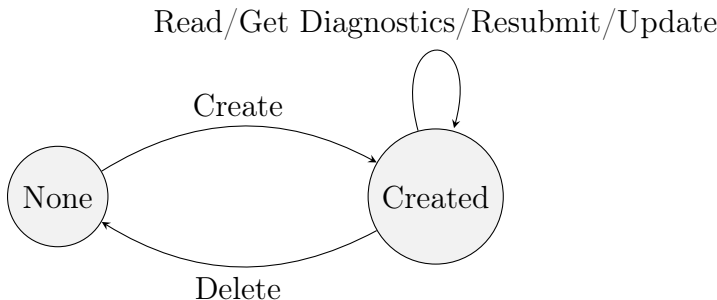
Logical Programming

The logical programming component provides an automated method to generating the OpenFlow programming necessary for an OpenFlow switch to forward traffic. Logical programming is composed of two parts: communication service types (CSTs) and logical connections (LCs).

11.1 Logical Connections

LCs require a CST, an adopted source node, adopted destination node(s), and at least one adopted path.

Purpose	Endpoints	Methods
Read	<code>~/config/logicalConnections/</code>	GET
	<code>~/config/logicalConnections('id')/</code>	GET
Create	<code>~/config/logicalConnections/</code>	POST
Update	<code>~/config/logicalConnections('id')/</code>	PATCH
	<code>~/config/logicalConnections('id')/</code>	PUT
Delete	<code>~/config/logicalConnections('id')/</code>	DELETE
	<code>~/config/logicalConnections/DeleteAll</code>	POST
Resubmit	<code>~/config/logicalConnections('id')/ResubmitLogicalConnection</code>	POST
	<code>~/operational/nodes('id')/ReplanInbandPath</code>	POST
	<code>~/config/logicalConnections/ReplanAll</code>	POST
Get Diagnostics	<code>~/config/logicalConnections('id')/GetLogicalConnectionStats</code>	POST
	<code>~/config/logicalConnections('id')/GetOpenFlowInformation</code>	POST
	<code>~/config/logicalConnections('id')/GetPathPlanInformation</code>	POST



11.1.1 Read Logical Connections

URL: /api/default/config/logicalConnections/

Method: GET

OData Type: `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection`

There is one type that can be returned:

- `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection`

The response for this request contains a list of this type.

Response Structure for `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection`

```
{
  "communicationServiceTypeId": Edm.String (Sel.Sel5056.LogicalConnectionValidation.-
DataObject.CommunicationServiceType.id),
  "connectionOrigin": Sel.Sel5056.LogicalConnectionValidation.Enums.LogicalConnection-
Origin,
  "destinationEndPoints": [ Edm.String, ],
  "enabled": Edm.Boolean,
  "errorState": Sel.BlueFrame.Core.ErrorState,
  "errors": [ Edm.String, ],
  "id": Edm.String,
  "lastEnabled": Edm.DateTimeOffset,
  "sourceEndPoints": [ Edm.String, ],
  "tags": [ Sel.Sel5056.Tags.AmTag, ],
  "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"
}
```


11.1.2 Create a Logical Connection

URL: /api/default/config/logicalConnections/

Method: POST

OData Type: `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection`

There is one type that supports the POST command:

- `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection`

Request Structure for `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection`

Property	Default Value
enabled	true
tags	[]

Table 11.2: Default Values for `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection`

```
{
  "communicationServiceTypeId": Edm.String (Sel.Sel5056.LogicalConnectionValidation.-
DataObject.CommunicationServiceType.id),
  "destinationEndpoints": [ Edm.String, ],
  "enabled": Edm.Boolean,
  "sourceEndpoints": [ Edm.String, ],
  "tags": [ Sel.Sel5056.Tags.AmTag, ],
}
```

Available User Role 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": "ConfigNode",
  "module": "DataBroker",
  "permissionTag": "Read",
}
```

```

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

```

Examples

Create a Unicast LC Example 10.

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

Table 11.3: Header Values for /api/default/config/logicalConnections

Request

You need three properties for the JSON body.

Property	Value
communicationService- TypeId	NTP Client
destinationEndPoints	["ad721fff6b7bc4445995efba2f9f214f"]
sourceEndPoints	["ae1bb68b3b1a94a5894504c6ba015dd2"]

Table 11.4: Body Values for Create a Unicast LC

```

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

{
  "communicationServiceTypeId": "NTP Client",
  "destinationEndPoints": ["ad721fff6b7bc4445995efba2f9f214f"],

```

```

"sourceEndpoints":    ["ae1bb68b3b1a94a5894504c6ba015dd2"]
}

```

Create a Multicast LC Example 11.

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

Table 11.5: Header Values for /api/default/config/logicalConnections

Request

You need three properties for the JSON body.

Property	Value
communicationService- TypeId	PTP Power Profile
destinationEndpoints	["aabc86b4b408b4e7d8135a89af7480c1", "ae9fc26e02e8a49d7bf3bdcf818e3dae"]
sourceEndpoints	["ad721fff6b7bc4445995efba2f9f214f"]

Table 11.6: Body Values for Create a Multicast LC

```

POST /api/default/config/logicalConnections HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 206

{
  "communicationServiceTypeId": "PTP Power Profile",
  "destinationEndpoints":    [
    "aabc86b4b408b4e7d8135a89af7480c1",
    "ae9fc26e02e8a49d7bf3bdcf818e3dae"
  ],
  "sourceEndpoints":    ["ad721fff6b7bc4445995efba2f9f214f"]
}

```

11.1.3 Update a Logical Connection (PATCH)

URL: /api/default/config/logicalConnections('id')/

Method: PATCH

OData Type: `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection`

Binding parameter (id): `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection.id`

There is one type that can be modified:

- `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection`

The following properties are patchable for `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection`:

- enabled
- lastEnabled
- tags

Available User Role 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": "Update",
  "tree": "ConfigTree"
}
```

11.1.4 Update a Logical Connection (PUT)

URL: `/api/default/config/logicalConnections('id')/`

Method: PUT

OData Type: `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection`

Binding parameter (id): `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection.id`

There is one type that can be modified:

- `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection`

Request Structure for Sel.Sel5056.LogicalConnectionValidation.DataObject.Logical-Connection

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their **default value**.

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

11.1.5 Delete a Logical Connection

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 Role 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"
}
```

11.1.6 Delete all Logical Connections

URL: /api/default/config/logicalConnections/DeleteAll

Method: POST

OData Type: **Sel.DeleteAll**

There is one type that supports the POST command:

- **Sel.DeleteAll**

Request Structure for Sel.DeleteAll

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

11.1.7 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 is one type that supports the POST command:

- **Sel.ResubmitLogicalConnection**

Request Structure for Sel.ResubmitLogicalConnection

No request body

Available User Role With Required Permissions

- `PermissionLevel3`

Permissions for Registered Applications

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

11.1.8 Replan a Switch IB Management Logical Connection

URL: `/api/default/operational/nodes('id')/ReplanInbandPath`

Method: POST

OData Type: `Sel.ReplanInbandPath`

Binding parameter (id): `Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id`

There is one type that supports the POST command:

- `Sel.ReplanInbandPath`

Request Structure for Sel.ReplanInbandPath

No request body

Available User Role With Required Permissions

- `PermissionLevel3`

Permissions for Registered Applications

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

11.1.9 Replan all Logical Connections

URL: /api/default/config/logicalConnections/ReplanAll

Method: POST

OData Type: **Sel.ReplanAll**

There is one type that supports the POST command:

- **Sel.ReplanAll**

Request Structure for Sel.ReplanAll

No request body

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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



```
"tree": "ConfigTree"
}
```

11.1.10 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 is one type that supports the POST command:

- [Sel.GetLogicalConnectionStats](#)

Request Structure for Sel.GetLogicalConnectionStats

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"
}
```

11.1.11 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 is one type that supports the POST command:

- [Sel.GetOpenFlowInformation](#)

Request Structure for Sel.GetOpenFlowInformation

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"
}
```

11.1.12 Get the Path Plan Information 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 is one type that supports the POST command:

- [Sel.GetPathPlanInformation](#)

Request Structure for Sel.GetPathPlanInformation

No request body

Available User Roles With Required Permissions

- Monitor
- PermissionLevel3

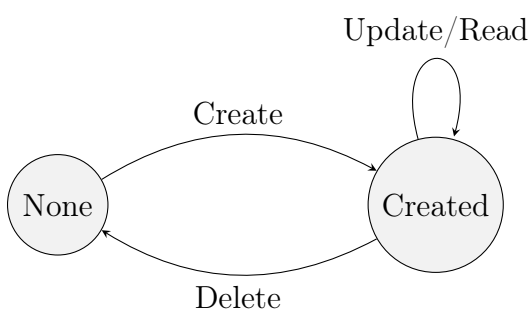
Permissions for Registered Applications

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

11.2 Communication Service Types

A communication service type (CST) is a blueprint for the OpenFlow entries that are created for LCs. All of the settings for LCs are contained in the CST and may be reused for multiple LCs.

Purpose	Endpoints	Methods
Read	<code>~/config/communicationServiceTypes/</code>	GET
	<code>~/config/communicationServiceTypes('id')/</code>	GET
Create	<code>~/config/communicationServiceTypes/</code>	POST
Update	<code>~/config/communicationServiceTypes('id')/</code>	PUT
	<code>~/config/communicationServiceTypes('id')/</code>	PATCH
Delete	<code>~/config/communicationServiceTypes('id')/</code>	DELETE



11.2.1 Read Communication Service Types

URL: `/api/default/config/communicationServiceTypes/`

Method: GET

OData Type: `Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType`

There is one type that can be returned:

- `Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType`

The response for this request contains a list of this type.

Response Structure for `Sel.Sel5056.LogicalConnectionValidation.DataObject.-CommunicationServiceType`

```

{
  "autoFillSettings": Sel.Sel5056.LogicalConnectionValidation.DataObject.AutoFill-
Settings,
  "cstCommunicationType": Sel.Sel5056.LogicalConnectionValidation.Enums.Cst-
CommunicationType,
  "displayName": Edm.String,
  "errorState": Sel.BlueFrame.Core.ErrorState,
  "errors": [ Edm.String, ],
  "id": Edm.String,
  "initialEnabledState": Edm.Boolean,

```

```

"initialHardTimeout":  Edm.Int32,
"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"
}

```

11.2.2 Create a Communication Service Type

URL: /api/default/config/communicationServiceTypes/

Method: POST

OData Type: `Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType`

There is one type that supports the POST command:

- `Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType`

Request Structure for `Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType`

Property	Default Value
autoFillSettings	null
cstCommunicationType	UnicastBidirectional
initialEnabledState	true
initialHardTimeout	0
numFailableLinks	0

numFailableNodes	0
priority	2000
setQueue	null
tags	[]
trafficMatch	{ "matchFields": [] }

Table 11.8: Default Values for Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType

```
{
  "autoFillSettings": Sel.Sel5056.LogicalConnectionValidation.DataObject.AutoFill-
    Settings,
  "cstCommunicationType": Sel.Sel5056.LogicalConnectionValidation.Enums.Cst-
    CommunicationType,
  "displayName": Edm.String,
  "initialEnabledState": Edm.Boolean,
  "initialHardTimeout": Edm.Int32,
  "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 Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

Examples

Create a Unicast CST Example 12.

Key	Value
-----	-------

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

Table 11.9: Header Values for /api/default/config/communicationServiceTypes

Request

You need four properties for the JSON body.

Property	Value
cstCommunicationType	UnicastBidirectional
displayName	DNP
setQueue	{ "queueId": 3 }
trafficMatch	{ "matchFields": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.MatchFields.UdpDst", "value": "20001" }] }

Table 11.10: Body Values for Create a Unicast CST

```
POST /api/default/config/communicationServiceTypes HTTP/1.1
```

```
Content-Type: application/json
```

```
Host: localhost
```

```
Authorization: Bearer 000...000
```

```
Content-Length: 234
```

```
{
  "cstCommunicationType": "UnicastBidirectional",
  "displayName": "DNP",
  "setQueue": {
    "queueId": 3
  }
  "trafficMatch": {
    "matchFields": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.-
UdpDst",
```

```

        "value": "20001"
      }
    ]
  }
}

```

Create a Multicast CST Example 13.

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

Table 11.11: Header Values for /api/default/config/communicationServiceTypes

Request

You need four properties for the JSON body.

Property	Value
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" }] }

Table 11.12: Body Values for Create a Multicast CST

```

POST /api/default/config/communicationServiceTypes HTTP/1.1
Content-Type: application/json
Host: localhost
Authorization: Bearer 000...000
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"
        }
      ]
    }
  }
}

```

11.2.3 Update a Communication Service Type (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 is one type that can be modified:

- `Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType`

Request Structure for `Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType`

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their **default value**.

Available User Role With Required Permissions

- `PermissionLevel3`

Permissions for Registered Applications

```

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

```


11.2.4 Update a Communication Service Type (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 is one type that can be modified:

- `Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType`

The following properties are patchable for `Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType`:

- `autoFillSettings`
- `cstCommunicationType`
- `displayName`
- `errors`
- `initialEnabledState`
- `initialHardTimeout`
- `numFailableLinks`
- `numFailableNodes`
- `priority`
- `setQueue`
- `tags`
- `trafficMatch`

Available User Role 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"
}

```

11.2.5 Delete a Communication Service Type

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 Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```

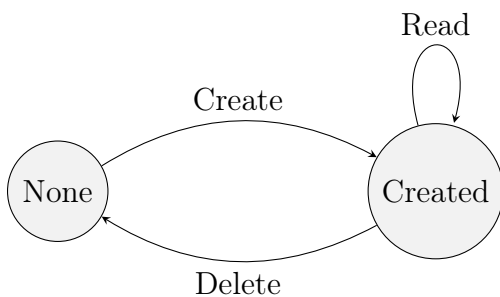
{
  "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
  "dataType": "CommunicationServiceType",
  "module": "DataBroker",
  "permissionTag": "Delete",
  "tree": "ConfigTree"
},
{
  "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",
  "dataType": "LogicalConnection",
  "module": "DataBroker",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

```

11.3 VID Reservation

Purpose	Endpoints	Methods
Read	~/config/vlanVidReservation/	GET
	~/config/vlanVidReservation('id')/	GET
Create	~/config/vlanVidReservation/ReserveVlanVidRange	POST

Update	Not supported	
Delete	~/config/vlanVidReservation('id')/	DELETE



11.3.1 Read Reserved VIDs

URL: [/api/default/config/vlanVidReservation/](#)

Method: GET

OData Type: [Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation](#)

There are two types that can be returned:

- [Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation](#)
- [Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidDestinationReservation](#)

The response for this request contains a list of these types.

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"
}
```

11.3.2 Reserve a VID or a Range of VIDs

URL: /api/default/config/vlanVidReservation/ReserveVlanVidRange

Method: POST

OData Type: [Sel.ReserveVlanVidRange](#)

There is one type that supports the POST command:

- [Sel.ReserveVlanVidRange](#)

Request Structure for Sel.ReserveVlanVidRange

```
{
  "end":  Edm.Int32,
  "start": Edm.Int32,
}
```

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
  "module": "PathProgrammer",
  "permissionTag": "ReserveVlanVidPermission"
}
```

Examples**Reserve a Range of Values Example 14.**

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

Table 11.14: Header Values for /api/default/config/vlanVidReservation/ReserveVlanVidRange

Request

You need two properties for the JSON body.

Property	Value
start	5
end	10

Table 11.15: Body Values for Reserve a Range of Values

```
POST /api/default/config/vlanVidReservation/ReserveVlanVidRange HTTP/1.1
Content-Type: application/json
Host: localhost
Authorization: Bearer 000...000
Content-Length: 23

{
  "start": 5,
  "end": 10
}
```

Reserve a Single VID Example 15.

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

Table 11.16: Header Values for /api/default/config/vlanVidReservation/ReserveVlanVidRange

Request

You need two properties for the JSON body.

Property	Value
start	5
end	5

Table 11.17: Body Values for Reserve a Single VID

```
POST /api/default/config/vlanVidReservation/ReserveVlanVidRange HTTP/1.1
Content-Type: application/json
Host: localhost
Authorization: Bearer 000...000
Content-Length: 22

{
  "start": 5,
  "end": 5
}
```

11.3.3 Delete a Reserved VID

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 Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

Chapter 12

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 represent the entry as found on the OpenFlow switches and contains additional information such the statistics returned by the switches. You can sort and filter entries through OData.

A flow entry may contain a group action that references a group entry and a meter instruction that references a meter entry. Group entries may also contain group actions to reference other groups through chaining.

OpenFlow entry diagnostics are contained in the `flowStats`, `groupStats`, and `meterStats` endpoints. These contain the OpenFlow statistics (or diagnostics) of the OpenFlow entries. Descriptions are contained in the `flowStats`, `groupDesc`, and `meterDescriptions` endpoints. These sets of data represent the information the controller retrieved from the switches.

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 informs the SEL-5056 to try and resolve the difference by modifying the switch configuration to match the SEL-5056 configuration.

12.1 Flow Entries

This chapter describes the flow entry data structure and its related endpoints and methods. The flow entry model of the SEL-5056 differs from the OpenFlow data model in the following three ways:

- Use of friendly names for values

- Use of aliased match fields, actions, and instructions
- Additional settings

Friendly names are strings that stand for integers. Fields that can have friendly names can have multiple values that map to the same OpenFlow value that is programmed on the switch. The translated value is normalized. Refer to the SEL-5056 instruction manual for more information.

Aliased values provide dynamic updates to flow entries by making 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, the SEL-5056 automatically updates the flow entry configuration in the SEL-5056.

Flow entries are made of the following three parts:

- General settings
- Match fields
- Instructions (including Actions)

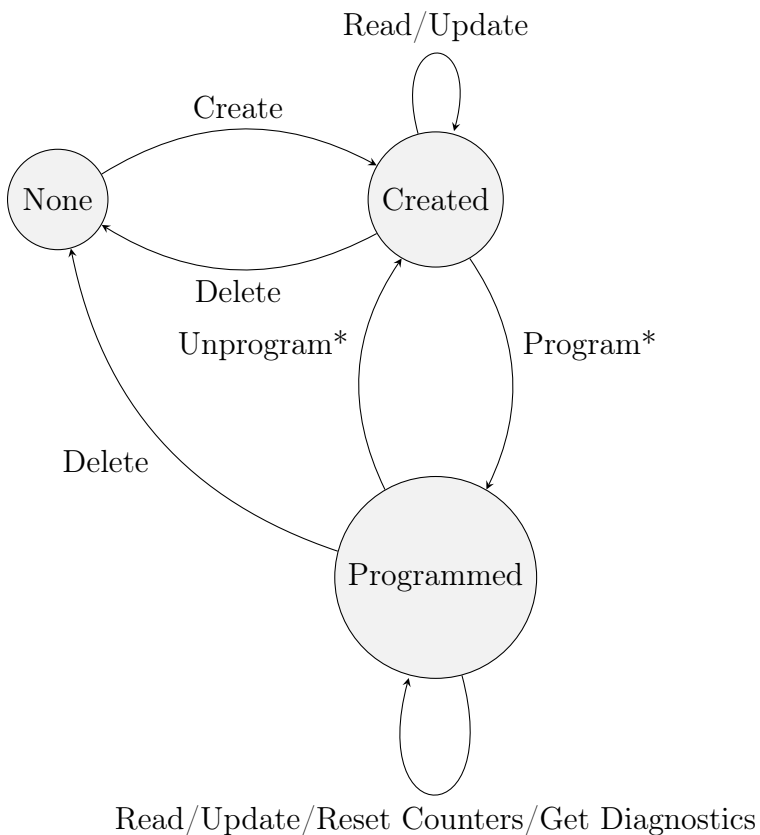
General settings can be further divided into two sets: those that are and are not defined by OpenFlow. The following settings are not defined by OpenFlow:

- `displayName`
- `enabled`
- `errorState`
- `errors`
- `id`
- `lastEdit`
- `node`
- `tags`
- `version`

The following table describes the settings defined by OpenFlow. The rest are defined in the OpenFlow 1.3.4 specification.

Purpose	Endpoints	Methods
Read	<code>~/config/flows/</code>	GET
	<code>~/config/flows('id')/</code>	GET
Create	<code>~/config/flows/</code>	POST

Update	~/config/flows('id')/	PUT
	~/config/flows('id')/	PATCH
Reset Counters	~/config/flows('id')/ResetCounters	POST
Delete	~/config/flows('id')/	DELETE
Get Diagnostics	~/operational/flowStats/	GET
	~/operational/flowStats('id')/	GET



12.1.1 Read Flow Entries

URL: [/api/default/config/flows/](#)

Method: GET

OData Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow](#)

There is one type that can be returned:

- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow](#)

The response for this request contains a list of this type.

Response Structure for [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow](#)

```
{
  "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,
}

```

Available 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"
}

```

12.1.2 Create a Flow Entry

URL: /api/default/config/flows/
 Method: POST

OData Type: `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow`

There is one type that supports the POST command:

- `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow`

Request Structure for `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow`

Property	Default Value
bufferId	0
checkOverlap	false
displayName	null
enabled	true
hardTimeout	0
idleTimeout	0
instructions	[]
match	{ "matchFields": [] }
noByteCounts	false
noPacketCounts	false
priority	1000
resetCounts	false
tableId	0
tags	[]

Table 12.2: 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, ],
}
```

Available User Role 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.

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

Table 12.3: Header Values for /api/default/config/flows

Request

You need four properties for the JSON body.

Property	Value
displayName	Simple Flow
node	ae1bb68b3b1a94a5894504c6ba015dd2
match	{ "matchFields": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.MatchFields.EthDst", "value": "012300000001" }] }

instructions	<pre>[{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.WriteActions", "actions": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.OutputAction", "outPort": "9" }] }]</pre>
--------------	--

Table 12.4: Body Values for Create a Simple Flow Entry

```
POST /api/default/config/flows HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 411

{
  "displayName": "Simple Flow",
  "node": "ae1bb68b3b1a94a5894504c6ba015dd2",
  "match": {
    "matchFields": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.-
EthDst",
        "value": "012300000001"
      }
    ]
  }
  "instructions": [
    {
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions",
      "actions": [
        {
          "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
          "outPort": "9"
        }
      ]
    }
  ]
}
```

```

    ]
  }
]
}

```

12.1.3 Update a Flow Entry (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 is one type that can be modified:

- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow](#)

Request Structure for [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow](#)

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their **default value**.

Available User Role With Required Permissions

- `PermissionLevel3`

Permissions for Registered Applications

```

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

```

12.1.4 Update a Flow Entry (PATCH)

URL: /api/default/config/flows('id')/

Method: PATCH

OData Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow](#)

Binding parameter (id): [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow.id](#)

There is one type that can be modified:

- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow](#)

The following properties are patchable for [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow](#):

- bufferId
- checkOverlap
- displayName
- enabled
- hardTimeout
- idleTimeout
- instructions
- match
- noByteCounts
- noPacketCounts
- node
- priority
- resetCounts
- tableId
- tags

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

Examples

Update a Simple Flow Entry by Changing the Outport Example 17.

Key	Value
Endpoint	/api/default/config/flows
Object id	a32521248118440eaabc014eaa82a13a
Method	PATCH
Content-Type	application/json

Table 12.5: Header Values for /api/default/config/flows

Request

You need one property for the JSON body.

Property	Value
instructions	[<pre> { "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.WriteActions", "actions": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.OutputAction", "maxLength": 65535, "outPort": "10" }] }]]</pre>

Table 12.6: Body Values for Update a Simple Flow Entry by Changing the Outport

```

PATCH /api/default/config/flows('a32521248118440eaabc014eaa82a13a') HTTP/1.1
Content-Type: application/json
Host: localhost
Authorization: Bearer 000...000
Content-Length: 222

{
  "instructions": [
    {
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions",
      "actions": [
        {

```



```

        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
        "maxLength": 65535,
        "outPort": "10"
    }
  ]
}
]
}

```

12.1.5 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 is one type that supports the POST command:

- **Sel.ResetCounters**

Request Structure for Sel.ResetCounters

No request body

Available User Role 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": "Flow",
  "module": "DataBroker",
  "permissionTag": "Read",
  "tree": "ConfigTree"
},
{

```

```

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

```

12.1.6 Delete a Flow Entry

URL: /api/default/config/flows('id')/

Method: DELETE

OData Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow](#)

Binding parameter (id): [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow.id](#)

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```

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

```

12.1.7 Read Flow Entry Diagnostics

URL: /api/default/operational/flowStats/

Method: GET

OData Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats](#)

There is one type that can be returned:

- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats](#)

The response for this request contains a list of this type.

Response Structure for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats

```

{
  "dataPathId": Edm.Int64,

```

```

    "id":  Edm.String,
    "stats":  [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat, ],
    "transactionId":  Edm.Int64,
}

```

Available User Roles With Required Permissions

- Monitor
- PermissionLevel3

Permissions for Registered Applications

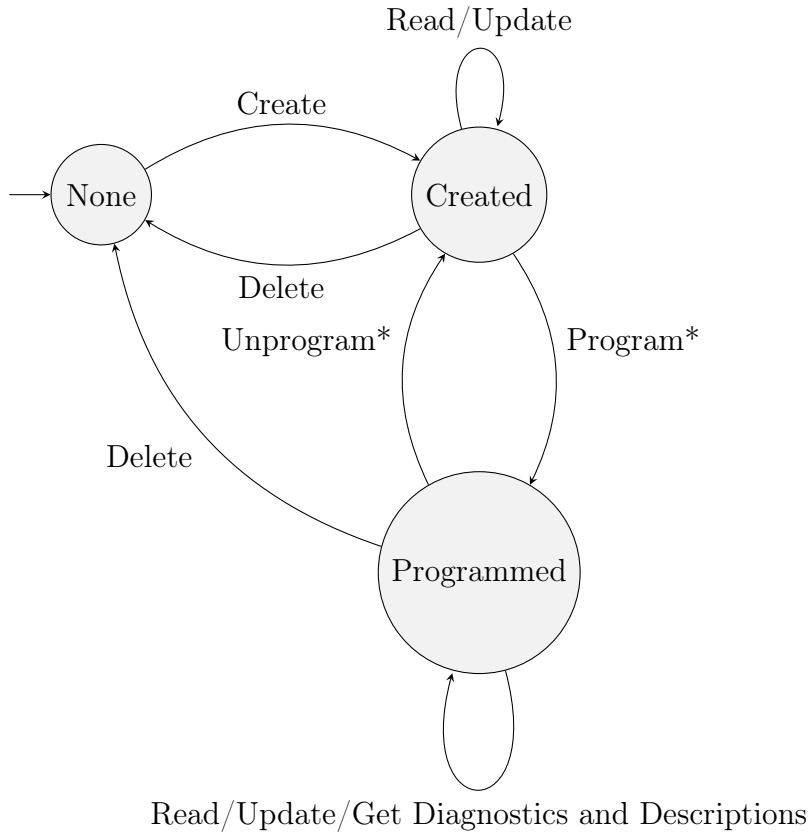
```

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

```

12.2 Group Entries

Purpose	Endpoints	Methods
Read	~/config/groups/	GET
	~/config/groups('id')/	GET
Create	~/config/groups/	POST
Update	~/config/groups('id')/	PUT
	~/config/groups('id')/	PATCH
Delete	~/config/groups('id')/	DELETE
Get Descriptions	~/operational/groupDesc/	GET
	~/operational/groupDesc('id')/	GET
Get Diagnostics	~/operational/groupStats/	GET
	~/operational/groupStats('id')/	GET



12.2.1 Read Group Entries

URL: `/api/default/config/groups/`

Method: GET

OData Type: `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group`

There is one type that can be returned:

- `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group`

The response for this request contains a list of this type.

Response Structure for `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group`

```

{
  "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),

```

```

"tags": [ Sel.Sel5056.Tags.AmTag, ],
"version": Edm.Int32,
}

```

Available 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"
}

```

12.2.2 Create a Group Entry

URL: /api/default/config/groups/

Method: POST

OData Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group](#)

There is one type that supports the POST command:

- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group](#)

Request Structure for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group

Property	Default Value
buckets	[]
displayName	null
enabled	true
groupId	0
groupType	FastFailover
tags	[]

Table 12.8: Default Values for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group

```

{
  "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,
"node": Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode),
"tags": [ Sel.Sel5056.Tags.AmTag, ],
}

```

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```

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

```

Examples

Create a Simple All Group Example 18.

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

Table 12.9: Header Values for /api/default/config/groups

Request

You need five properties for the JSON body.

Property	Value
displayName	Simple All Group
node	ae1bb68b3b1a94a5894504c6ba015dd2
groupId	100
groupType	All

buckets	<pre>[{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.Bucket", "watchPort": "Any", "watchGroup": "Any", "actions": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.OutputAction", "outPort": "1" }] } { "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.Bucket", "watchPort": "Any", "watchGroup": "Any", "actions": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.OutputAction", "outPort": "2" }] }]</pre>
---------	--

Table 12.10: Body Values for Create a Simple All Group

```
POST /api/default/config/groups HTTP/1.1
```

```
Content-Type: application/json
```

```
HOST: localhost
```

```
Authorization: Bearer 000...000
```

```
Content-Length: 564
```

```
{
  "displayName": "Simple All Group",
  "node": "ae1bb68b3b1a94a5894504c6ba015dd2",
  "groupId": 100,
  "groupType": "All",
  "buckets": [
```

```

{
  "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket",
  "watchPort": "Any",
  "watchGroup": "Any",
  "actions": [
    {
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
      "outPort": "1"
    }
  ]
}
{
  "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket",
  "watchPort": "Any",
  "watchGroup": "Any",
  "actions": [
    {
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
      "outPort": "2"
    }
  ]
}
]
}

```

Create a Simple Fast Failover Group Example 19.

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

Table 12.11: Header Values for /api/default/config/groups

Request

You need five properties for the JSON body.

Property	Value
displayName	Simple FF Group
node	a97287576ee4243fdbcccce172a9d067
groupId	101
groupType	FastFailover

buckets

```
[
  {
    "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.-
DataTreeObjects.Bucket",
    "watchPort": "B1",
    "watchGroup": "ANY",
    "actions": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.-
DataTreeObjects.GroupAction",
        "groupId": 100
      }
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.-
DataTreeObjects.SetFieldAction",
        "field": {
          "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.-
DataTreeObjects.MatchFields.VlanVid",
          "value": "4001"
        }
      }
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.-
DataTreeObjects.PushVlanAction",
        "etherType": "33024"
      }
    ]
  }
  {
    "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.-
DataTreeObjects.Bucket",
    "watchPort": "B2",
    "watchGroup": "ANY",
    "actions": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.-
DataTreeObjects.OutputAction",
        "outPort": "B2"
      }
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.-
DataTreeObjects.PopVlanAction"
      }
    ]
  }
]
```

Table 12.12: Body Values for Create a Simple Fast Failover Group

```

POST /api/default/config/groups HTTP/1.1
Content-Type: application/json
Host: localhost
Authorization: Bearer 000...000
Content-Length: 938

{
  "displayName": "Simple FF Group",
  "node": "a97287576ee4243fdbcccce172a9d067",
  "groupId": 101,
  "groupType": "FastFailover",
  "buckets": [
    {
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket",
      "watchPort": "B1",
      "watchGroup": "ANY",
      "actions": [
        {
          "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group-
Action",
          "groupId": 100
        }
        {
          "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetField-
Action",
          "field": {
            "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-
MatchFields.VlanVid",
            "value": "4001"
          }
        }
        {
          "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PushVlan-
Action",
          "etherType": "33024"
        }
      ]
    }
  ]
  {
    "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket",

```

```

    "watchPort": "B2",
    "watchGroup": "ANY",
    "actions": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
        "outPort": "B2"
      }
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PopVlan-
Action"
      }
    ]
  }
]
}

```

12.2.3 Update a Group Entry (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 is one type that can be modified:

- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group](#)

Request Structure for [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group](#)

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their **default value**.

Available User Role 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.

Key	Value
Endpoint	/api/default/config/groups
Object id	a926c96ef3545433dac018399e17a24d
Method	PUT
Content-Type	application/json

Table 12.13: Header Values for /api/default/config/groups

Request

You need five properties for the JSON body.

Property	Value
displayName	Simple All Group
node	ae1bb68b3b1a94a5894504c6ba015dd2
groupId	100
groupType	All

buckets	<pre>[{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.Bucket", "watchPort": "Any", "watchGroup": "Any", "actions": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.OutputAction", "outPort": "1" }] } { "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.Bucket", "watchPort": "Any", "watchGroup": "Any", "actions": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.OutputAction", "outPort": "2" }] } { "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.Bucket", "watchPort": "Any", "watchGroup": "Any", "actions": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.OutputAction", "outPort": "3" }] }]</pre>
---------	---

Table 12.14: Body Values for Modify (PUT) the Action Buckets of an All Group

```

PUT /api/default/config/groups('a926c96ef3545433dac018399e17a24d') HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 782

{
  "displayName": "Simple All Group",
  "node": "ae1bb68b3b1a94a5894504c6ba015dd2",
  "groupId": 100,
  "groupType": "All",
  "buckets": [
    {
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket",
      "watchPort": "Any",
      "watchGroup": "Any",
      "actions": [
        {
          "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
          "outPort": "1"
        }
      ]
    }
  ]
  {
    "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket",
    "watchPort": "Any",
    "watchGroup": "Any",
    "actions": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
        "outPort": "2"
      }
    ]
  }
  {
    "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket",
    "watchPort": "Any",
    "watchGroup": "Any",
    "actions": [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-

```

```

    Action",
      "outPort":  "3"
    }
  ]
}
]
}

```

12.2.4 Update a Group Entry (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), include the entire set of action buckets after making all modifications, such as reordering, adding, deleting, or modifying.

There is one type that can be modified:

- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group](#)

The following properties are patchable for [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group](#):

- buckets
- displayName
- enabled
- groupId
- groupType
- node
- tags

Available User Role 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"
  }
}

```

Examples

Modify (PATCH) the Action Buckets of an All Group Example 21.

Key	Value
Endpoint	/api/default/config/groups
Object id	a926c96ef3545433dac018399e17a24d
Method	PATCH
Content-Type	application/json

Table 12.15: Header Values for /api/default/config/groups

Request

You need one property for the JSON body.

Property	Value
----------	-------

buckets	<pre>[{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.Bucket", "watchPort": "Any", "watchGroup": "Any", "actions": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.OutputAction", "outPort": "3" }] } { "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.Bucket", "watchPort": "Any", "watchGroup": "Any", "actions": [{ "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.- DataTreeObjects.OutputAction", "outPort": "4" }] }]</pre>
---------	--

Table 12.16: Body Values for Modify (PATCH) the Action Buckets of an All Group

```
PATCH /api/default/config/groups('a926c96ef3545433dac018399e17a24d') HTTP/1.1
Content-Type: application/json
HOST: localhost
Authorization: Bearer 000...000
Content-Length: 449
```

```
{
  "buckets": [
    {
      "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket",
      "watchPort": "Any",
      "watchGroup": "Any",
```

```

    "actions":      [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
        "outPort":   "3"
      }
    ]
  }
  {
    "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket",
    "watchPort":   "Any",
    "watchGroup":  "Any",
    "actions":      [
      {
        "@odata.type": "#Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Output-
Action",
        "outPort":   "4"
      }
    ]
  }
]
}

```

12.2.5 Delete a Group Entry

URL: /api/default/config/groups('id')/

Method: DELETE

OData Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group

Binding parameter (id): Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group.id

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```

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

```

12.2.6 Read Group Entry Descriptions

URL: /api/default/operational/groupDesc/

Method: GET

OData Type: `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc`

There is one type that can be returned:

- `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc`

The response for this request contains a list of this type.

Response Structure for `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc`

```
{
  "dataPathId":  Edm.Int64,
  "groups":  [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group, ],
  "id":  Edm.String,
  "transactionId":  Edm.Int64,
}
```

Available 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"
}
```

12.2.7 Read Group Entry Diagnostics

URL: /api/default/operational/groupStats/

Method: GET

OData Type: `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats`

There is one type that can be returned:

- `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats`

The response for this request contains a list of this type.

Response Structure for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats

```
{
  "dataPathId":  Edm.Int64,
  "id":  Edm.String,
  "stats":  [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat, ],
  "transactionId":  Edm.Int64,
}
```

Available User Roles With Required Permissions

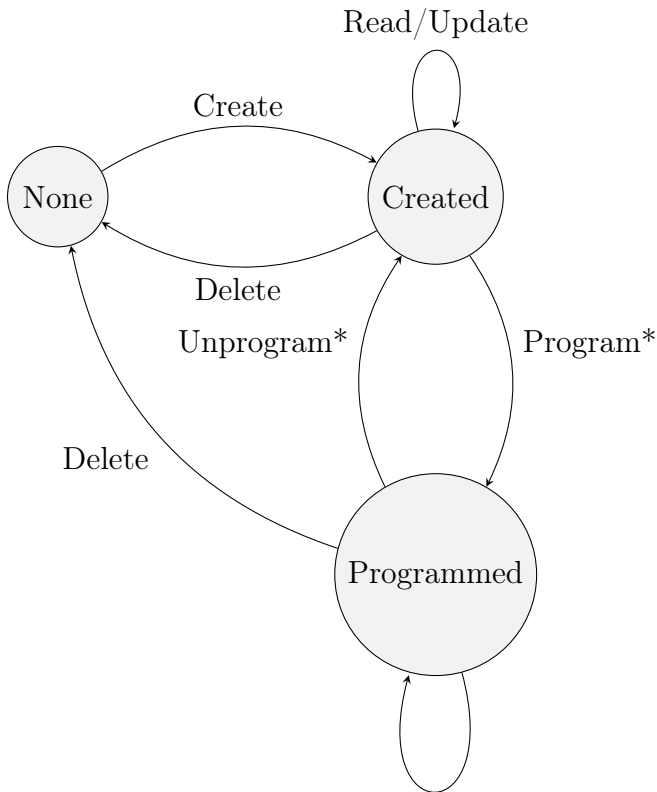
- Monitor
- PermissionLevel3

Permissions for Registered Applications

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

12.3 Meter Entries

Purpose	Endpoints	Methods
Read	~/config/meters/	GET
	~/config/meters('id')/	GET
Create	~/config/meters/	POST
Update	~/config/meters('id')/	PUT
	~/config/meters('id')/	PATCH
Delete	~/config/meters('id')/	DELETE
Get Descriptions	~/operational/meterDescriptions/	GET
	~/operational/meterDescriptions('id')/	GET
Get Features	~/operational/meterFeatures/	GET
	~/operational/meterFeatures('id')/	GET
Get Diagnostics	~/operational/meterstats/	GET
	~/operational/meterstats('id')/	GET



Read/Update/Get Diagnostics, Features, and Diagnostics

12.3.1 Read Meter Entries

URL: /api/default/config/meters/

Method: GET

OData Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter](#)

There is one type that can be returned:

- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter](#)

The response for this request contains a list of this type.

Response Structure for [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter](#)

```

{
  "displayName": Edm.String,
  "enabled": Edm.Boolean,
  "errorState": Sel.BlueFrame.Core.ErrorState,
  "errors": [ Edm.String, ],
  "flags": [ Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.RateType-Base, ],
  "id": Edm.String,
  "meterBands": [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand, ],
  "meterId": Edm.Int64,

```

```

"node":  Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode),
"tags":  [ Sel.Sel5056.Tags.AmTag, ],
"version":  Edm.Int32,
}

```

Available 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"
}

```

12.3.2 Create a Meter Entry

URL: /api/default/config/meters/

Method: POST

OData Type: `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter`

There is one type that supports the POST command:

- `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter`

Request Structure for `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter`

Property	Default Value
displayName	null
enabled	true
flags	[]
meterBands	[]
meterId	0
tags	[]

Table 12.18: Default Values for `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter`

```

{
  "displayName":  Edm.String,
  "enabled":  Edm.Boolean,

```

```

"flags": [ Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.RateType-
Base, ],
"meterBands": [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand, ],
"meterId": Edm.Int64,
"node": Edm.String (Sel.Sel5056.TopologyManager.Nodes.ConfigNode),
"tags": [ Sel.Sel5056.Tags.AmTag, ],
}

```

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```

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

```

12.3.3 Update a Meter Entry (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 is one type that can be modified:

- Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter

Request Structure for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their **default value**.

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```

{
  "@odata.type": "#Sel.BlueFrame.Core.DataBrokerPermission",

```

```

    "dataType": "Meter",
    "module": "DataBroker",
    "permissionTag": "Update",
    "tree": "ConfigTree"
}

```

12.3.4 Update a Meter Entry (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 is one type that can be modified:

- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter](#)

The following properties are patchable for [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter](#):

- displayName
- enabled
- flags
- meterBands
- meterId
- node
- tags

Available User Role 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"
}

```

12.3.5 Delete a Meter Entry

URL: /api/default/config/meters('id')/

Method: DELETE

OData Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter](#)

Binding parameter (id): [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.id](#)

Available User Role With Required Permissions

- [PermissionLevel3](#)

Permissions for Registered Applications

```

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

```

12.3.6 Read Meter Descriptions

URL: /api/default/operational/meterDescriptions/

Method: GET

OData Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions](#)

There is one type that can be returned:

- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions](#)

The response for this request contains a list of this type.

Response Structure for [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions](#)

```

{
  "dataPathId": Edm.Int64,
  "id": Edm.String,
  "meterDescriptionsList": [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDesc, ],
  "transactionId": Edm.Int64,
}

```

Available 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"
}
```

12.3.7 Read Meter Features

URL: /api/default/operational/meterFeatures/

Method: GET

OData Type: `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures`

There is one type that can be returned:

- `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures`

The response for this request contains a list of this type.

Response Structure for `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures`

```
{
  "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,
}
```

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

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

12.3.8 Read Meter Diagnostics

URL: /api/default/operational/meterstats/

Method: GET

OData Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats](#)

There is one type that can be returned:

- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats](#)

The response for this request contains a list of this type.

Response Structure for [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats](#)

```
{
  "dataPathId": Edm.Int64,
  "id": Edm.String,
  "meterStatList": [ Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStat, ],
  "transactionId": Edm.Int64,
}
```

Available 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 13

Controller Management

This chapter covers endpoints related to the controller service outside of the network and X.509 certificate configuration, such as database and session management.

13.1 Commissioning

Purpose	Endpoints	Methods
Read	~/security/IsCommissioned	POST
Commission	~/security/Commission	POST

13.1.1 Commissioned Status of SEL-5056

URL: /api/default/security/IsCommissioned

Method: POST

odata Type: **Sel.IsCommissioned**

There is one type that supports the POST command:

- **Sel.IsCommissioned**

Request Structure for Sel.IsCommissioned

No request body

Available User Roles With Required Permissions

- Monitor
- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

None required

13.1.2 Commission the SEL-5056

URL: /api/default/security/Commission

Method: POST

OData Type: **Sel.Commission**

There is one type that supports the POST command:

- **Sel.Commission**

Request Structure for Sel.Commission

```
{
  "commissioningList": [ Sel.BlueFrame.CommissioningManager.CommissioningConcern, ],
}
```

Available User Roles With Required Permissions

No token required.

Permissions for Registered Applications

This endpoint is not available for registered applications.

13.2 Database Management

Purpose	Endpoints	Methods
Backup	~/settings/Backup	POST
Restore	~/settings/Restore	POST

13.2.1 Export the SEL-5056 Database

URL: /api/default/settings/Backup

Method: POST

OData Type: **Sel.Backup**

There is one type that supports the POST command:

- **Sel.Backup**

Request Structure for Sel.Backup

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

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
  "module": "TrustAuthority",
  "permissionTag": "Backup"
}
```

13.2.2 Import an SEL-5056 Database

URL: /api/default/settings/Restore

Method: POST

odata Type: **Sel.Restore**

There is one type that supports the POST command:

- **Sel.Restore**

Request Structure for Sel.Restore

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

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```
{
  "module": "TrustAuthority",
  "permissionTag": "Restore"
}
```

Examples**Import an SEL-5056 Database Example 22.**

Key	Value
Endpoint	/api/default/settings/Restore
Method	POST
Content-Type	multipart/form-data; boundary=----- 226827793616512233304021929274

Table 13.3: Header Values for /api/default/settings/Restore

Request

```

POST /api/default/settings/Restore HTTP/1.1
Content-Type: multipart/form-data;
               boundary=-----226827793616512233304021929274
HOST: 192.168.56.1
Authorization: Bearer 000...000
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--

```

13.3 Event Categories

Purpose	Endpoints	Methods
Read	~/settings/eventCategories/	GET
	~/settings/eventCategories('id')/	GET
Update	~/settings/eventCategories('id')/	PATCH
	~/settings/eventCategories('id')/	PUT
Delete	Not supported	
Create	Not supported	

13.3.1 Read Event Categories

URL: /api/default/settings/eventCategories/

Method: GET

OData Type: `Sel.BlueFrame.Core.EventBus.EventCategory`

There is one type that can be returned:

- `Sel.BlueFrame.Core.EventBus.EventCategory`

The response for this request contains a list of this type.

Response Structure for `Sel.BlueFrame.Core.EventBus.EventCategory`

```

{
  "behaviors": [ Sel.BlueFrame.Core.EventBus.Behaviors.Behavior, ],

```



```

    "id":  Edm.String,
    "key":  Edm.String,
  }

```

Available 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"
}

```

13.3.2 Update an Event Category (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 is one type that can be modified:

- [Sel.BlueFrame.Core.EventBus.EventCategory](#)

The following property is patchable for [Sel.BlueFrame.Core.EventBus.EventCategory](#):

- behaviors

Available User Role 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"
  }
}

```

13.3.3 Update an Event Category (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 is one type that can be modified:

- `Sel.BlueFrame.Core.EventBus.EventCategory`

Request Structure for `Sel.BlueFrame.Core.EventBus.EventCategory`

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their default value.

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```

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

```

13.4 Event Types

Purpose	Endpoints	Methods
Read	<code>~/settings/eventTypes/</code>	GET
	<code>~/settings/eventTypes('id')/</code>	GET

Update	<code>~/settings/eventTypes('id')/</code>	PUT
	<code>~/settings/eventTypes('id')/</code>	PATCH
Delete	Not supported	
Create	Not supported	

13.4.1 Read Event Types

URL: `/api/default/settings/eventTypes/`

Method: GET

OData Type: `Sel.BlueFrame.Core.EventBus.EventType`

There is one type that can be returned:

- `Sel.BlueFrame.Core.EventBus.EventType`

The response for this request contains a list of this type.

Response Structure for `Sel.BlueFrame.Core.EventBus.EventType`

```
{
  "canAutoClear": Edm.Boolean,
  "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,
}
```

Available 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"
}
```

13.4.2 Update an Event Type (PUT)

URL: /api/default/settings/eventTypes('id')/

Method: PUT

OData Type: `Sel.BlueFrame.Core.EventBus.EventType`

Binding parameter (id): `Sel.BlueFrame.Core.EventBus.EventType.id`

There is one type that can be modified:

- `Sel.BlueFrame.Core.EventBus.EventType`

Request Structure for `Sel.BlueFrame.Core.EventBus.EventType`

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their default value.

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

13.4.3 Update an Event Type (PATCH)

URL: /api/default/settings/eventTypes('id')/

Method: PATCH

OData Type: `Sel.BlueFrame.Core.EventBus.EventType`

Binding parameter (id): `Sel.BlueFrame.Core.EventBus.EventType.id`

There is one type that can be modified:

- `Sel.BlueFrame.Core.EventBus.EventType`

The following properties are patchable for `Sel.BlueFrame.Core.EventBus.EventType`:

- canAutoClear
- facility
- severity

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

13.5 Login and Password Management

Purpose	Endpoints	Methods
Read	~/settings/securityManager/	GET
	~/settings/securityManager('id')/	GET
Update	~/settings/securityManager('id')/	PATCH
	~/settings/securityManager('id')/	PUT

13.5.1 Read Password and Login Management

URL: /api/default/settings/securityManager/

Method: GET

OData Type: **Sel.BlueFrame.SecurityManager.SecurityManagerSettings**

There is one type that can be returned:

- **Sel.BlueFrame.SecurityManager.SecurityManagerSettings**

The response for this request contains a list of this type.

Response Structure for Sel.BlueFrame.SecurityManager.SecurityManagerSettings

```
{
  "enforceUniquePasswordChanges": Edm.Boolean,
  "id": Edm.String,
  "isAdminInterventionRequiredOnLockout": Edm.Boolean,
```

```

"isUsagePolicyAcknowledgmentRequired": Edm.Boolean,
"lockoutSeconds": Edm.Int32,
"loginAttemptWindow": Edm.Int32,
"minimumPasswordLength": Edm.Int32,
"passwordHistorySize": Edm.Int32,
"passwordUpdateFrequencyLimitInHours": Edm.Int32,
"preventPasswordReuse": Edm.Boolean,
}

```

Available User Roles With Required Permissions

- Monitor
- PermissionLevel3
- SecurityAdministrator

Permissions for Registered Applications

```

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

```

13.5.2 Update Password and Login Management (PATCH)

URL: /api/default/settings/securityManager('id')/

Method: PATCH

OData Type: `Sel.BlueFrame.SecurityManager.SecurityManagerSettings`

Binding parameter (id): `Sel.BlueFrame.SecurityManager.SecurityManagerSettings.id`

There is one type that can be modified:

- `Sel.BlueFrame.SecurityManager.SecurityManagerSettings`

The following properties are patchable for `Sel.BlueFrame.SecurityManager.SecurityManagerSettings`:

- `enforceUniquePasswordChanges`
- `isAdminInterventionRequiredOnLockout`
- `isUsagePolicyAcknowledgmentRequired`
- `lockoutSeconds`

- loginAttemptWindow
- minimumPasswordLength
- passwordHistorySize
- passwordUpdateFrequencyLimitInHours
- preventPasswordReuse

Available User Role 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"
}
```

13.5.3 Update Password and Login Management (PUT)

URL: /api/default/settings/securityManager('id')/

Method: PUT

OData Type: **Sel.BlueFrame.SecurityManager.SecurityManagerSettings**

Binding parameter (id): **Sel.BlueFrame.SecurityManager.SecurityManagerSettings.id**

There is one type that can be modified:

- **Sel.BlueFrame.SecurityManager.SecurityManagerSettings**

Request Structure for Sel.BlueFrame.SecurityManager.SecurityManagerSettings

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their default value.

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

13.6 Usage Policy and Session Management

Purpose	Endpoints	Methods
Read	~/settings/webSettings/	GET
	~/settings/webSettings('id')/	GET
Update	~/settings/webSettings('id')/	PATCH
	~/settings/webSettings('id')/	PUT

13.6.1 Read Usage Policy and Session Management

URL: /api/default/settings/webSettings/

Method: GET

OData Type: `Sel.BlueFrame.RestBroker.Plumbing.RestSettings`

There is one type that can be returned:

- `Sel.BlueFrame.RestBroker.Plumbing.RestSettings`

The response for this request contains a list of this type.

Response Structure for `Sel.BlueFrame.RestBroker.Plumbing.RestSettings`

```
{
  "id": Edm.String,
  "loginBanner": Edm.String,
  "maxSessions": Edm.Int64,
  "maxSessionsPerUser": Edm.Int64,
  "sessionTimeoutMinutes": Edm.Int64,
}
```

Available 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"
}
```

13.6.2 Update Usage Policy and Session Management (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 is one type that can be modified:

- [Sel.BlueFrame.RestBroker.Plumbing.RestSettings](#)

The following properties are patchable for [Sel.BlueFrame.RestBroker.Plumbing.RestSettings](#):

- loginBanner
- maxSessions
- maxSessionsPerUser
- sessionTimeoutMinutes

Available User Role 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"
}

```

13.6.3 Update Usage Policy and Session Management (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 is one type that can be modified:

- **Sel.BlueFrame.RestBroker.Plumbing.RestSettings**

Request Structure for Sel.BlueFrame.RestBroker.Plumbing.RestSettings

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their default value.

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

```

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

```

13.7 Strings

Purpose	Endpoints	Methods
Read	~/settings/strings/	GET
	~/settings/strings('id')/	GET
Update	Not supported	
Delete	Not supported	
Create	Not supported	

13.7.1 Read Localization Strings

URL: /api/default/settings/strings/

Method: GET

OData Type: `Sel.BlueFrame.Core.EventBus.LocaleString`

There is one type that can be returned:

- `Sel.BlueFrame.Core.EventBus.LocaleString`

The response for this request contains a list of this type.

Response Structure for `Sel.BlueFrame.Core.EventBus.LocaleString`

```
{
  "id":  Edm.String,
  "key":  Edm.String,
  "records":  [ Sel.BlueFrame.Core.EventBus.LocaleRecord, ],
  "variables":  [ Edm.String, ],
}
```

Available 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"
}
```


Chapter 14

Packet In/Out

This chapter covers endpoints related to the PACKET OUT and PACKET IN OpenFlow functionality.

14.1 Packet In/Out Management

Purpose	Endpoints	Methods
Read	<code>~/operational/packets/</code>	GET
Create	<code>~/operational/nodes('id')/PacketOut</code>	POST
	<code>~/operational/nodes('id')/PacketOutToTable</code>	POST
Update	Not supported	
Delete	Not supported	

14.1.1 Read Packets Sent to the Controller Using the OpenFlow Controller Port

URL: `/api/default/operational/packets/`

Method: GET

OData Type: `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ReceivedPacket`

There is one type that can be returned:

- `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ReceivedPacket`

The response for this request contains a list of this type.

Response Structure for `Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ReceivedPacket`

```
{
  "contents":  Edm.String,
  "cookie":    Edm.Int64,
  "dataPathId": Edm.Int64,
  "id":        Edm.String,
  "monotonicId": Edm.Int64,
  "port":      Edm.Int64,
```

```
"prettyDataPathId": Edm.String,
"receivedAt": Edm.Int64,
}
```

Available 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"
}
```

14.1.2 Inject a Packet Out an OpenFlow Port

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

Method: POST

OData Type: **Sel.PacketOut**

Binding parameter (id): **Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id**

There is one type that supports the POST command:

- **Sel.PacketOut**

Request Structure for Sel.PacketOut

```
{
  "destinationPortId": Edm.Int64,
  "packetData": Edm.String,
}
```

Available User Role With Required Permissions

- PermissionLevel3

Permissions for Registered Applications

```
{
  "module": "OpenFlowPlugin",
```

```

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

```

14.1.3 Inject a Packet into an OpenFlow Switch With the InPort Set

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

Method: POST

OData Type: [Sel.PacketOutToTable](#)

Binding parameter (id): [Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.id](#)

There is one type that supports the POST command:

- [Sel.PacketOutToTable](#)

Request Structure for Sel.PacketOutToTable

```

{
  "inPortId": Edm.Int64,
  "packetData": Edm.String,
}

```

Available User Role With Required Permissions

- [PermissionLevel3](#)

Permissions for Registered Applications

```

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

```


Chapter 15

Registered Applications

Registered applications provide additional capabilities for authenticating and managing permissions with the SEL-5056.

Generally, registered applications provide the following four benefits:

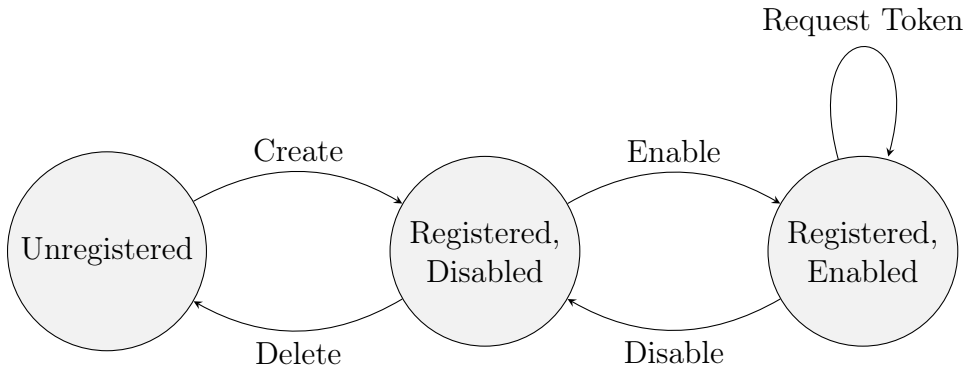
- Use of the **Client Credentials and Authorization Code** grant types
- Finer control over **permissions**
- A profile listed on the Application Management page
- Client-side X.509 certificates

Once an application or user obtains a token, the only difference between a registered application and other grant types are what permissions are allowed for the token.

There are two methods for registering an application. You can **use the GET method to retrieve the information from the application** or you can **use the POST method to send the information to the SEL-5056**.

15.1 Registered Application Management

Purpose	Endpoints	Methods
Create	~/security/applicationLinks/InitiateApplicationRegistration	POST
	~/security/applicationLinks/	POST
Delete	~/security/applicationLinks('id')/	DELETE
Read	~/security/applicationLinks/	GET
	~/security/applicationLinks('id')/	GET
Update	Not supported	
Enable	~/security/applicationLinks('id')/EnableApplicationLink	POST
Disable	~/security/applicationLinks('id')/DisableApplicationLink	POST



15.1.1 Request Application Registration Information From an External URL

URL: /api/default/security/applicationLinks/InitiateApplicationRegistration

Method: POST

OData Type: **Sel.InitiateApplicationRegistration**

There is one type that supports the POST command:

- **Sel.InitiateApplicationRegistration**

Request Structure for Sel.InitiateApplicationRegistration

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

Available User Role 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 website serving the registration information is not present on the SEL-5056 or is revoked.

If the certification has been previously revoked and subsequently reuploaded, manually remove the revoked versions from the database.

15.1.2 Create a Registered Application

URL: /api/default/security/applicationLinks/

Method: POST

OData Type: `Sel.BlueFrame.ApplicationRegistration.ApplicationLink`

There is one type that supports the POST command:

- `Sel.BlueFrame.ApplicationRegistration.ApplicationLink`

Request Structure for `Sel.BlueFrame.ApplicationRegistration.ApplicationLink`

Property	Default Value
applicationIcon	null
description	null

Table 15.2: Default Values for `Sel.BlueFrame.ApplicationRegistration.ApplicationLink`

```
{
  "applicationIcon":  Edm.String,
  "applicationIp":    Edm.String (IpAddress),
  "applicationName":  Edm.String,
  "authenticationTypes": [ Sel.BlueFrame.ApplicationRegistration.AuthenticationType,
],
  "certificateThumbprint":  Edm.String,
  "description":           Edm.String,
  "versions": [ Edm.String, ] (SEL-5056 version (X.X.X)),
}
```

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

15.1.3 Delete a Registered Application

URL: /api/default/security/applicationLinks('id')/

Method: DELETE

OData Type: `Sel.BlueFrame.ApplicationRegistration.ApplicationLink`

Binding parameter (id): `Sel.BlueFrame.ApplicationRegistration.ApplicationLink.id`

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

15.1.4 Read Registered Applications

URL: `/api/default/security/applicationLinks/`

Method: GET

OData Type: `Sel.BlueFrame.ApplicationRegistration.ApplicationLink`

There is one type that can be returned:

- `Sel.BlueFrame.ApplicationRegistration.ApplicationLink`

The response for this request contains a list of this type.

Response Structure for `Sel.BlueFrame.ApplicationRegistration.ApplicationLink`

```
{
  "applicationIcon": Edm.String,
  "applicationIp": Edm.String (IpAddress),
  "applicationName": Edm.String,
  "authenticationTypes": [ Sel.BlueFrame.ApplicationRegistration.AuthenticationType,
],
  "certificateThumbprint": Edm.String,
  "description": Edm.String,
  "enabled": Edm.Boolean,
  "id": Edm.String,
  "versions": [ Edm.String, ] (SEL-5056 version (X.X.X)),
}
```

Available 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"
}
```

15.1.5 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 is one type that supports the POST command:

- `Sel.EnableApplicationLink`

Request Structure for `Sel.EnableApplicationLink`

No request body

Available User Roles With Required Permissions

- `PermissionLevel3`
- `SecurityAdministrator`

Permissions for Registered Applications

```
{
  "module": "ApplicationRegistration",
  "permissionTag": "EnableApplicationLink"
}
```

15.1.6 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 is one type that supports the POST command:

- `Sel.DisableApplicationLink`

Request Structure for `Sel.DisableApplicationLink`

No request body

Available User Roles With Required Permissions

- `PermissionLevel3`
- `SecurityAdministrator`

Permissions for Registered Applications

```
{  
  "module": "ApplicationRegistration",  
  "permissionTag": "DisableApplicationLink"  
}
```

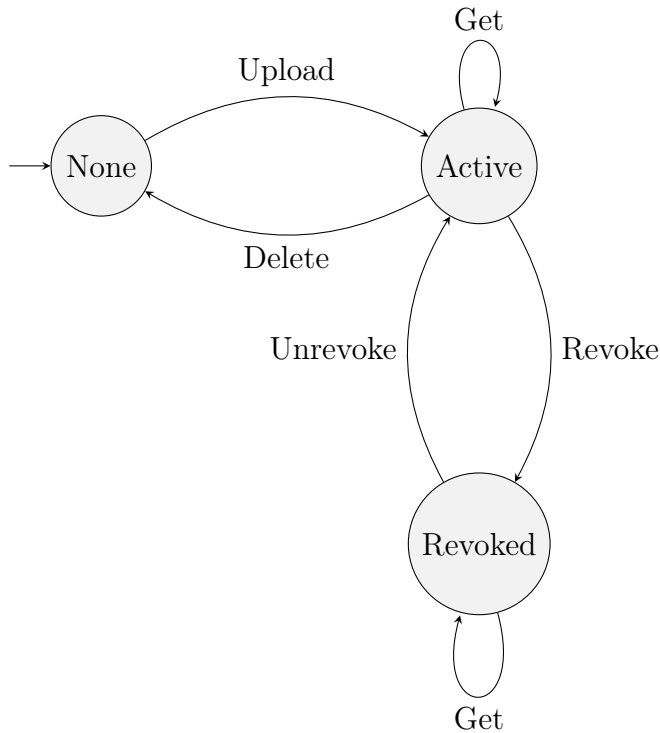
Chapter 16

X.509 Certificates

This chapter covers managing X.509 certifications for the SEL-5056 and SEL-2740S and SEL-2742S devices.

16.1 X.509 Certificate Management

Purpose	Endpoints	Methods
Get	~/certificate/certificateInfo/	GET
	~/certificate/certificateInfo('id')/AsPem	POST
	~/certificate/certificateInfo('id')/	GET
Create	Not supported	
Upload	~/certificate/UploadCertificate	POST
Delete	~/certificate/certificateInfo('id')/	DELETE
Revoke	~/certificate/certificateInfo('id')/RevokeCertificate	POST
	~/certificate/RevokeCertificateByPurpose	POST
	~/certificate/RevokeCertificateByThumbprint	POST
Unrevoke	~/certificate/certificateInfo('id')/TrustCertificate	POST
Update	Not supported	



16.1.1 Read X.509 Certificates

URL: `/api/default/certificate/certificateInfo/`

Method: GET

OData Type: `Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation`

There is one type that can be returned:

- `Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation`

The response for this request contains a list of this type.

Response Structure for `Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation`

```

{
  "certificateVersion": Edm.Int32,
  "commonName": Edm.String,
  "displayName": Edm.String,
  "hasPrivateKey": Edm.Boolean,
  "id": Edm.String,
  "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,
}

```

Available 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"
}

```

16.1.2 Export a Certificate in the 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 is one type that supports the POST command:

- [Sel.AsPem](#)

Request Structure for Sel.AsPem

No request body

Available 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"
}
```

16.1.3 Upload an X.509 Certificate

URL: /api/default/certificate/UploadCertificate

Method: POST

OData Type: *Sel.UploadCertificate*

There is one type that supports the POST command:

- *Sel.UploadCertificate*

Request Structure for Sel.UploadCertificate

```
{
  "base64Certificate": Edm.String,
  "certificatePassword": Edm.String,
  "purpose": Sel.BlueFrame.Core.TrustAuthority.CertificatePurpose,
}
```

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

If the purpose is *Sel.BlueFrame.TrustAuthority.TrustedCertificates.TrustedCertificatePurpose*, the required permission is:

```
{
  "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
  "module": "TrustAuthority",
  "permissionTag": "Replace",
  "purpose": "TrustedCertificatePurpose"
}
```

If the purpose is *Sel.BlueFrame.TrustAuthority.WebCertificateManager.WebCertificatePurpose*, the required permission is:

```
{
  "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
  "module": "TrustAuthority",
  "permissionTag": "Replace",
  "purpose": "WebCertificatePurpose"
}
```

If the purpose is `Sel.BlueFrame.TrustAuthority.InternalCertificateAuthority.InternalCaPurpose`, the required permission is:

```
{
  "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
  "module": "TrustAuthority",
  "permissionTag": "Replace",
  "purpose": "InternalCaPurpose"
}
```

16.1.4 Delete a X.509 Certificate

URL: `/api/default/certificate/certificateInfo('id')/`

Method: DELETE

OData Type: `Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation`

Binding parameter (id): `Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation.id`

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

16.1.5 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 is one type that supports the POST command:

- `Sel.RevokeCertificate`

Request Structure for Sel.RevokeCertificate

No request body

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

There are two permissions required. One regardless of purpose is the following:

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

And one of the following permissions depending on purpose.

If the certificate has a purpose of Sel.BlueFrame.TrustAuthority.TrustedCertificates.-TrustedCertificatePurpose, the other required permission is the following:

```
{
  "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
  "module": "TrustAuthority",
  "permissionTag": "Revoke",
  "purpose": "TrustedCertificatePurpose"
}
```

If the certificate has a purpose of Sel.BlueFrame.TrustAuthority.WebCertificateManager.-WebCertificatePurpose, the other required permission is the following:

```
{
  "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
  "module": "TrustAuthority",
  "permissionTag": "Revoke",
  "purpose": "WebCertificatePurpose"
}
```

If the certificate has a purpose of Sel.BlueFrame.TrustAuthority.InternalCertificateAuthority.-InternalCaPurpose, the other required permission is the following:

```
{
  "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
  "module": "TrustAuthority",
  "permissionTag": "Revoke",
  "purpose": "InternalCaPurpose"
}
```

16.1.6 Revoke a X.509 Certificate by its Purpose

URL: /api/default/certificate/RevokeCertificateByPurpose

Method: POST

OData Type: **Sel.RevokeCertificateByPurpose**

There is one type that supports the POST command:

- **Sel.RevokeCertificateByPurpose**

Request Structure for Sel.RevokeCertificateByPurpose

```
{
  "purpose":  Sel.BlueFrame.Core.TrustAuthority.CertificatePurpose,
}
```

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

There are two permissions required. One regardless of purpose is the following:

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

And one of the following permissions depending on purpose.

If the purpose is **Sel.BlueFrame.TrustAuthority.TrustedCertificates.TrustedCertificatePurpose**, the other required permission is the following:

```
{
  "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
  "module": "TrustAuthority",
  "permissionTag": "Revoke",
  "purpose": "TrustedCertificatePurpose"
}
```

If the purpose is **Sel.BlueFrame.TrustAuthority.WebCertificateManager.WebCertificatePurpose**, the other required permission is the following:

```
{
  "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
  "module": "TrustAuthority",
  "permissionTag": "Revoke",
  "purpose": "WebCertificatePurpose"
}
```

If the purpose is `Sel.BlueFrame.TrustAuthority.InternalCertificateAuthority.InternalCaPurpose`, the other required permission is the following:

```
{
  "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
  "module": "TrustAuthority",
  "permissionTag": "Revoke",
  "purpose": "InternalCaPurpose"
}
```

16.1.7 Revoke an X.509 Certificate by its Thumbprint

URL: `/api/default/certificate/RevokeCertificateByThumbprint`

Method: POST

odata Type: `Sel.RevokeCertificateByThumbprint`

There is one type that supports the POST command:

- `Sel.RevokeCertificateByThumbprint`

Request Structure for `Sel.RevokeCertificateByThumbprint`

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

Available User Role With Required Permissions

- `SecurityAdministrator`

Permissions for Registered Applications

There are two certificates required. One regardless of purpose is the following:

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

And one of the following permissions depending on purpose.

If the certificate has a purpose of `Sel.BlueFrame.TrustAuthority.TrustedCertificates.-TrustedCertificatePurpose`, the other required permission is the following:

```
{
  "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
  "module": "TrustAuthority",
  "permissionTag": "Revoke",
  "purpose": "TrustedCertificatePurpose"
}
```

If the certificate has a purpose of `Sel.BlueFrame.TrustAuthority.WebCertificateManager.-WebCertificatePurpose`, the other required permission is the following:

```
{
  "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
  "module": "TrustAuthority",
  "permissionTag": "Revoke",
  "purpose": "WebCertificatePurpose"
}
```

If the certificate has a purpose of `Sel.BlueFrame.TrustAuthority.InternalCertificateAuthority.-InternalCaPurpose`, the other required permission is the following:

```
{
  "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
  "module": "TrustAuthority",
  "permissionTag": "Revoke",
  "purpose": "InternalCaPurpose"
}
```

16.1.8 Unrevoke a Revoke Certificate

URL: `/api/default/certificate/certificateInfo('id')/TrustCertificate`

Method: POST

OData Type: `Sel.TrustCertificate`

Binding parameter (id): `Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation.id`

There is one type that supports the POST command:

- `Sel.TrustCertificate`

Request Structure for `Sel.TrustCertificate`

No request body

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

There are two certificates required. One regardless of purpose is the following:

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

And one of the following permissions depending on purpose.

If the certificate has a purpose of `Sel.BlueFrame.TrustAuthority.TrustedCertificates.-TrustedCertificatePurpose`, the other required permission is the following:

```
{
  "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
  "module": "TrustAuthority",
  "permissionTag": "Trust",
  "purpose": "TrustedCertificatePurpose"
}
```

If the certificate has a purpose of `Sel.BlueFrame.TrustAuthority.WebCertificateManager.-WebCertificatePurpose`, the other required permission is the following:

```
{
  "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
  "module": "TrustAuthority",
  "permissionTag": "Trust",
  "purpose": "WebCertificatePurpose"
}
```

If the certificate has a purpose of `Sel.BlueFrame.TrustAuthority.InternalCertificateAuthority.-InternalCaPurpose`, the other required permission is the following:

```
{
  "@odata.type": "#Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission",
  "module": "TrustAuthority",
  "permissionTag": "Trust",
  "purpose": "InternalCaPurpose"
}
```

16.2 Certificate Revocation List

Purpose	Endpoints	Methods
Get	~/settings/crlSettings/	GET
Update	~/settings/crlSettings('id')/	PATCH
	~/settings/crlSettings('id')/	PUT

16.2.1 Read Certificate Revocation List (CRL) Settings

URL: [/api/default/settings/crlSettings/](#)

Method: GET

OData Type: [Sel.BlueFrame.TrustAuthority.CrlSettings](#)

There is one type that can be returned:

- [Sel.BlueFrame.TrustAuthority.CrlSettings](#)

The response for this request contains a list of this type.

Response Structure for [Sel.BlueFrame.TrustAuthority.CrlSettings](#)

```
{
  "crlEnabled": Edm.Boolean,
  "id": Edm.String,
  "protectInternalCA": Edm.Boolean,
}
```

Available User Role With Required Permissions

- SecurityAdministrator

Permissions for Registered Applications

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

16.2.2 Update a Certificate Revocation List (CRL) Setting (PATCH)

URL: [/api/default/settings/crlSettings\('id'\)/](#)

Method: PATCH

OData Type: [Sel.BlueFrame.TrustAuthority.CrlSettings](#)

Binding parameter (id): [Sel.BlueFrame.TrustAuthority.CrlSettings.id](#)

There is one type that can be modified:

- `Sel.BlueFrame.TrustAuthority.CrlSettings`

The following properties are patchable for `Sel.BlueFrame.TrustAuthority.CrlSettings`:

- `crlEnabled`
- `protectInternalCA`

Available User Role With Required Permissions

- `SecurityAdministrator`

Permissions for Registered Applications

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

16.2.3 Update a Certificate Revocation List (CRL) Setting (PUT)

URL: `/api/default/settings/crlSettings('id')/`

Method: PUT

OData Type: `Sel.BlueFrame.TrustAuthority.CrlSettings`

Binding parameter (id): `Sel.BlueFrame.TrustAuthority.CrlSettings.id`

There is one type that can be modified:

- `Sel.BlueFrame.TrustAuthority.CrlSettings`

Request Structure for `Sel.BlueFrame.TrustAuthority.CrlSettings`

The request structure is the same as the previous GET response. Some fields cannot be included in a POST request but can be included in a PUT request. Fields that can be included in a POST message but are not included in a PUT request are reset to their default value.

Available User Role With Required Permissions

- `SecurityAdministrator`

Permissions for Registered Applications

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


Part IV

Additional Topics

This part covers additional functionality.

Chapter 17

Situational Awareness, Diagnostics, and Counters

This chapter summarizes the situational function of the SEL-5056. The situational awareness functionality can be divided into several parts, which include diagnostics and counters.

17.1 Counters

Counters represent measurements of some network functions. Counters can be divided into the following two types:

1. OpenFlow
2. Non-OpenFlow

OpenFlow counters can be divided into two types:

1. Those that are defined in the OpenFlow specification (Direct)
2. Those that are defined by the SEL-5056 (Indirect)

17.1.1 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-2740S and SEL-2742S switches

17.1.2 Indirect Counters

Logical Connections

You can obtain the flow and group diagnostics for LCs 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 show those used by the LC.

17.2 Diagnostics

Objects are classified into the following two types:

1. Separate operational information for associated configuration object, such as flow stats (operational) for flow entries (configuration)
2. Operational information included in the object itself (other trees)

17.2.1 OpenFlow

OpenFlow reports group and meter descriptions separately from the group and meter entry statistics and they are combined in flow entries. The SEL-5056 expresses 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/

17.3 Situational Awareness

17.3.1 Logical Connections

The path plans used to the program the flow and group entries for an LC are available from the [GetPathPlanInformation](#) endpoint.

Chapter 18

Asynchronous Data Exchange With SignalR

SignalR is an API abstraction for support for an asynchronous data exchange. Use SignalR to receive a notification when an object has been created, modified, or deleted. SignalR does not replace the need to retrieve objects but rather replaces the need to poll the SEL-5056 when those objects are added, removed, or modified.

The supported protocols are the following:

1. WebSocket (RFC 6455)
2. SSE (server side events)
3. Long polling
4. Forever Frame

This section also covers a basic example of setting up a WebSocket connection with the SEL-5056.

18.1 Hubs

Subscriptions to object types are handled by hubs, each with a name derived from the URI in the form:

default-X-Y

where X is the service root (certificate, config, etc.) and Y is the object type.

If you want to access an object at the URI `/api/default/X/Y`, the corresponding hub name is default-X-Y.

For example, the flow entry URI is `/api/default/config/flows`, so the corresponding hub name is default-config-flows.

18.2 Targets

For each endpoint, there are several client targets (methods) available, as shown in [Table 18.1](#).

Target	Description	Parameters
SubscribeAll	Receive notifications for all object changes	-
UnsubscribeAll	Stop receiving notifications for all object changes	-
SubscribeAllCreates	Receive notifications for newly created objects	-
UnsubscribeAllCreates	Stop receiving notifications for newly created objects	-
SubscribeTo	Receive notifications for specific changes	ids
UnsubscribeTo	Stop receiving notifications for specific changes	ids

Table 18.1: Client SignalR Targets

The SubscribeTo and UnsubscribeTo targets take object IDs as parameters, which are a list of object IDs, such as ["a138faec7a61843209f422b67aaa6c53"].

The server sends updates with one of the following three targets:

- OnCreate
- OnUpdate
- OnDelete

The arguments field contains a list that includes the hub and the ID of the object affected by the target. For example, if a flow object with an ID of a138faec7a61843209f422b67aaa6c53 is created, the argument would be ["default", "config", "flows", "a138faec7a61843209f422b67aaa6c53"].

18.3 Authorization

For authorization, a valid token obtained with the required permissions must be passed into the connect HTTP call by using the same form as OData.

18.4 Permissions

There are a set of permissions specific to SignalR. If the user role has access to an endpoint, the user also has permission to subscribe to the endpoint over SignalR if a hub exists for it.

18.5 WebSocket

18.5.1 Creating a WebSocket Connection

The following turns the HTTP socket into a WebSocket socket for subscribing to all flow entry changes.

First, negotiate SignalR by using the POST method to the /signalr/api/negotiate endpoint.

```
POST /signalr/api/negotiate HTTP/1.1
HOST: localhost
Authorization: Bearer 000.000
Content-Length: 0
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Server: Kestrel
Content-Length: 252
```

```
{"connectionId":"zM13TOTGc7Ycr-FShRLosA","availableTransports":[{"transport":"WebSockets","transferFormats":["Text","Binary"]},{"transport":"ServerSentEvents","transferFormats":["Text"]},{"transport":"LongPolling","transferFormats":["Text","Binary"]}]}
```

Listing 18.1: Negotiating SignalR

Then select WebSocket by using the upgrade header and passing in the OAuth token as an `access_token` parameter.

```
GET /signalr/api?id=zM13TOTGc7Ycr-FShRLosA&access_token=000...000 HTTP
/1.1
Connection: Upgrade
Origin: https://localhost
Sec-WebSocket-Key: oDwoWeKbFWhQ8jjeRC/VQQ==
Sec-WebSocket-Version: 13
Upgrade: websocket
HOST: localhost
Authorization: Bearer 000.000
```

```
HTTP/1.1 101 Switching Protocols
Connection: Upgrade
Server: Kestrel
Upgrade: websocket
Access-Control-Allow-Origin: *
Sec-WebSocket-Accept: 21zNErwhWrRp/n0sgheD5yoUuEU=
```

Listing 18.2: Selecting WebSocket

At this point, the socket is now operating under the WebSocket protocol. The next step is to send a subscribe message.

18.5.2 Subscribing to all Flow Entry Changes by Using WebSocket

Example 23.

In this example, subscribe to all flow entry changes. Therefore, the hub is default-config-flows and the target is SubscribeAll, which does not have any parameters. Send the following WebSocket message:

```
{"arguments":["default-operational-flowstats"],"invocationId":"0","target":"subscribeAll","type":1}
```

Listing 18.3: WebSocket Message to Subscribe to All Flow Entry Changes

WebSocket messages to the SEL-5056 require a header and masking. Assuming that the masking value is 0xdf9e7221, the request would be the following octets:

```
0x81 0xdb 0xdf 0x9e 0x72 0x21 0xa4 0xbc 0x13 0x53
0xb8 0xeb 0x1f 0x44 0xb1 0xea 0x01 0x03 0xe5 0xc5
0x50 0x45 0xba 0xf8 0x13 0x54 0xb3 0xea 0x5f 0x42
0xb0 0xf0 0x14 0x48 0xb8 0xb3 0x14 0x4d 0xb0 0xe9
0x01 0x03 0x82 0xb2 0x50 0x48 0xb1 0xe8 0x1d 0x42
0xbe 0xea 0x1b 0x4e 0xb1 0xd7 0x16 0x03 0xe5 0xbc
0x42 0x03 0xf3 0xbc 0x06 0x40 0xad 0xf9 0x17 0x55
0xfd 0xa4 0x50 0x52 0xaa 0xfc 0x01 0x42 0xad 0xf7
0x10 0x44 0x9e 0xf2 0x1e 0x03 0xf3 0xbc 0x06 0x58
0xaf 0xfb 0x50 0x1b 0xee 0xe3 0x6c
```

Listing 18.4: Masked WebSocket Message to Subscribe to All Flow Entry Changes

Ignoring the header, when a flow entry with the ID of a138faec7a61843209f422b67aaa6c53 has been updated, the SEL-5056 sends the following message:

```
{"type":1,"target":"OnUpdate","arguments":["default","config","flows","a138faec7a61843209f422b67aaa6c53"]}
```

Listing 18.5: Notification of Flow Update

The ID appears at the last index of the arguments array.

18.5.3 Subscribing to a Specific Flow Entry Change by Using WebSocket

Example 24.

To subscribe to a specific flow entry change, use the same negotiation process for subscribing to all flow entries except that the target is SubscribeTo and the IDs are added as an array to the arguments array. To subscribe to only flow objects changes with an ID of af19b0c47bd404dbd90ab7529f72d01e, use the following message:

```
{'arguments': ['default-config-flows', ['af19b0c47bd404dbd90ab7529f72d01e']], 'invocationId': '0', 'target': 'subscribeTo', 'type': 1}
```

Listing 18.6: WebSocket Message to Subscribe to One Specific Flow Entry Change

Chapter 19

Permissions and Roles

Permissions are objects that represent the ability to read, modify, delete, or create objects for specific endpoints and methods (such as the permission to POST to `/api/default/config/flows`) and, in some cases, also by 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 use the GET method to read users from the REST interface at the endpoint `/api/default/security/users`, as shown in the [read users](#) section.

The SEL-5056 assigns a fix set of permissions to each of the following three roles:

- SecurityAdministrator
- PermissionLevel3
- Monitor

You can retrieve the list of permissions for each role by using the [read user roles](#) endpoint. Use the [read registered applications](#) endpoint to view the permissions available to a registered application.

Some endpoint and method combinations do not require permissions and some require multiple permissions. Each endpoint described in [endpoints section](#) under Permissions for Registered Applications contains the permissions required to use the endpoints and method combination for registered applications that may have a variable set of permissions. The user roles with required permissions contain roles that can use the endpoint and method combination.

Part V

OData Definitions

This section lists the OData types defined by the SEL-5056. Those that begin with Sel. are defined specifically by the SEL-5056 and those that do not are defined elsewhere.

This section is divided into the following four parts:

Name	Description
Native	Types defined by the OData standard and referenced by this document
Primary	Types that define the OData type that define the request or response object for endpoints
Secondary	Types that are referenced by the primary types
Actions	Types defined for OData actions
Enums	Types defined for OData enums

The primary types include the URL at which the type is hosted. Each type includes a list of the properties for each type along with additional attributes.

The primary and secondary sections also include lists of the relationships between the type and other types that reference it. From this information, you can trace how this type is referenced down to the primary types. When the property listed has a type ending in `bindingParameter`, this represents a reference to a specific instance of the type by the value of the `id` property.

Chapter 20

Native OData Types

This section covers the types defined by the OData standard that are also used by the SEL-5056. All of these types begin with Edm. Refer to the OData standard for more information.

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.

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 20.2: Edm.* Types

Chapter 21

Primary OData Types

This section describes the OData types that are base types for the endpoints. **Secondary OData types** describes the OData types are used by the endpoints in the individual properties.

21.1 Sel.BlueFrame.ApplicationRegistration.ApplicationLink

URL: /api/default/security/applicationLinks/

Name	Attributes
applicationIcon	Type: Edm.String
applicationIp	DataClass: IPAddress
	Type: Edm.String
applicationName	CharacterSet: Printable7BitAscii
	MaxLength: 48
	Type: Edm.String
authenticationTypes	MinLength: 1
	Type: [Sel.BlueFrame.ApplicationRegistration.AuthenticationType]
certificateThumbprint	Type: Edm.String
description	MaxLength: 4096
	Type: Edm.String
enabled	Type: Edm.Boolean
id	Type: Edm.String
versions	MinLength: 1
	Type: [Edm.String]

Table 21.1: Properties for Sel.BlueFrame.ApplicationRegistration.ApplicationLink

Relationship of Sel.BlueFrame.ApplicationRegistration.ApplicationLink to other types:

- Sel.DisableApplicationLink** (binding parameter)
- Sel.EnableApplicationLink** (binding parameter)
- Sel.InitiateApplicationRegistration** (binding parameter)

21.2 Sel.BlueFrame.Core.EventBus.EventCategory

URL: </api/default/settings/eventCategories/>

Name	Attributes
behaviors	Type: [Sel.BlueFrame.Core.EventBus.Behaviors.Behavior]
id	Type: Edm.String
key	Type: Edm.String

Table 21.2: Properties for Sel.BlueFrame.Core.EventBus.EventCategory

Relationship of Sel.BlueFrame.Core.EventBus.EventCategory to other types:

—Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode.eventCategories

——[Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode](#)

21.3 Sel.BlueFrame.Core.EventBus.EventType

URL: </api/default/settings/eventTypes/>

Name	Attributes
canAutoClear	Type: Edm.Boolean
categoryKey	Type: Edm.String
durationSetting	Type: Sel.BlueFrame.Core.EventBus.Enums.DurationType
eventKey	Type: Edm.String
facility	Type: Sel.BlueFrame.Core.EventBus.Enums.FacilityCode
id	Type: Edm.String
messageId	Type: Edm.String
module	Type: Edm.String
severity	Type: Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel

Table 21.3: Properties for Sel.BlueFrame.Core.EventBus.EventType

The following types inherit from this type:

-[Sel.BlueFrame.Core.EventBus.ClearedEventType](#)

Relationship of Sel.BlueFrame.Core.EventBus.EventType to other types:

No other data types reference this type.

21.3.1 Sel.BlueFrame.Core.EventBus.ClearedEventType

Name	Attributes
------	------------

linkedEventKey	Type: Edm.String
----------------	-------------------------

Table 21.4: Property for Sel.BlueFrame.Core.EventBus.ClearedEventType

21.4 Sel.BlueFrame.Core.EventBus.LocaleString

URL: </api/default/settings/strings/>

Name	Attributes
id	Type: Edm.String
key	Type: Edm.String
	WriteOnce: true
records	Type: [Sel.BlueFrame.Core.EventBus.LocaleRecord]
variables	Type: [Edm.String]

Table 21.5: Properties for Sel.BlueFrame.Core.EventBus.LocaleString

Relationship of Sel.BlueFrame.Core.EventBus.LocaleString to other types:

No other data types reference this type.

21.5 Sel.BlueFrame.Core.SecurityManager.AuthService

URL: </api/default/security/authServices/>

Name	Attributes
errorState	Type: Sel.BlueFrame.Core.ErrorState
errors	Type: [Edm.String]
id	Type: Edm.String

Table 21.6: Properties for Sel.BlueFrame.Core.SecurityManager.AuthService

The following types inherit from this type:

-**Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService**

Relationship of Sel.BlueFrame.Core.SecurityManager.AuthService to other types:

—**Sel.GetAvailableGroups** (binding parameter)

—**Sel.TestAuthenticateUser** (binding parameter)

21.5.1 Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService

Name	Attributes
bindDN	MaxLength: 256
	Type: Edm.String
bindPassword	Type: Edm.String
email	MaxLength: 256
	Type: Edm.String
firstName	MaxLength: 256
	Type: Edm.String
groupMemberAttribute	MaxLength: 256
	Type: Edm.String
hostname	Type: Edm.String
lastName	MaxLength: 256
	Type: Edm.String
name	MaxLength: 32
	MinLength: 1
	Type: Edm.String
	Unique: True
portNumber	Max: 65535
	Min: 1
	Step: 1
	Type: Edm.Int32
protocolVersion	Type: Edm.Int32
searchBase	MaxLength: 256
	Type: Edm.String
userIdFilter	MaxLength: 256
	Type: Edm.String
workPhone	MaxLength: 256
	Type: Edm.String

Table 21.7: Properties for Sel.BlueFrame.SecurityManager.LdapPlugin.LDAPAuthService

21.6 Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup

URL: </api/default/security/authServiceGroups/>

Name	Attributes
authService	MaxLength: 256
	Type: Sel.BlueFrame.Core.SecurityManager.AuthService
	WriteOnce: true
displayName	MaxLength: 32
	Type: Edm.String
	Unique: True
dn	MaxLength: 256

	Type: Edm.String
	WriteOnce: true
groupRoles	Type: Sel.BlueFrame.Core.SecurityManager.Role
id	Type: Edm.String

Table 21.8: Properties for Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup

Relationship of Sel.BlueFrame.Core.SecurityManager.AuthServiceGroup to other types:
No other data types reference this type.

21.7 Sel.BlueFrame.Core.SecurityManager.Role

URL: </api/default/security/roles/>

Name	Attributes
displayName	Type: Edm.String
id	Type: Edm.String
permissions	Type: [Sel.BlueFrame.Core.SecurityManager.Permission]
roleName	Type: Edm.String

Table 21.9: Properties for Sel.BlueFrame.Core.SecurityManager.Role

Relationship of Sel.BlueFrame.Core.SecurityManager.Role to other types:
No other data types reference this type.

21.8 Sel.BlueFrame.Core.SecurityManager.User

URL: </api/default/security/users/>

Name	Attributes
displayName	Type: Edm.String
enabled	Type: Edm.Boolean
failedLogins	Type: [Sel.BlueFrame.Core.SecurityManager.UserFailedLogin]
id	Type: Edm.String
isLockedOut	Type: Edm.Boolean
lastSuccessLoginIp	Type: Edm.String
lastSuccessfulLogin	Type: Edm.DateTimeOffset
lockedOutUntil	Type: Edm.DateTimeOffset
lockoutTime	Type: Edm.DateTimeOffset
roles	Type: Sel.BlueFrame.Core.SecurityManager.Role

username	CaseInsensitive: true
	CharacterSet: Printable7BitAscii
	MaxLength: 128
	MinLength: 1
	Type: Edm.String
	Unique: True

Table 21.10: Properties for Sel.BlueFrame.Core.SecurityManager.User

The following types inherit from this type:

- Sel.BlueFrame.Core.SecurityManager.LocalUser
- Sel.BlueFrame.Core.SecurityManager.AuthServiceUser

Relationship of Sel.BlueFrame.Core.SecurityManager.User to other types:

- Sel.UpdatePassword (binding parameter)

21.8.1 Sel.BlueFrame.Core.SecurityManager.LocalUser

Name	Attributes
clearTextPass	CharacterSet: Printable7BitAscii
	MaxLength: 128
	Type: Edm.String
passwordCreationTime	Type: Edm.DateTimeOffset
passwordStore	Type: [Sel.BlueFrame.Core.SecurityManager.Password]

Table 21.11: Properties for Sel.BlueFrame.Core.SecurityManager.LocalUser

21.8.2 Sel.BlueFrame.Core.SecurityManager.AuthServiceUser

Name	Attributes
authService	Type: Sel.BlueFrame.Core.SecurityManager.AuthService

Table 21.12: Property for Sel.BlueFrame.Core.SecurityManager.AuthServiceUser

21.9 Sel.BlueFrame.RestBroker.OdataPlugin.Models.Rest-Transaction

URL: **/api/default/operational/transactions/**

Name	Attributes
id	Type: Edm.String

Table 21.13: Property for Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction

Relationship of Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction to other types:

- Sel.Commit (binding parameter)
- Sel.ClearTransaction (binding parameter)
- Sel.Commit (binding parameter)
- Sel.ClearTransaction (binding parameter)
- Sel.Commit (binding parameter)
- Sel.ClearTransaction (binding parameter)
- Sel.Commit (binding parameter)
- Sel.ClearTransaction (binding parameter)
- Sel.Commit (binding parameter)
- Sel.ClearTransaction (binding parameter)

21.10 Sel.BlueFrame.RestBroker.Plumbing.RestSettings

URL: /api/default/settings/webSettings/

Name	Attributes
id	Type: Edm.String
loginBanner	MaxLength: 4095
	MinLength: 0
	Type: Edm.String
maxSessions	Type: Edm.Int64
maxSessionsPerUser	Type: Edm.Int64
sessionTimeoutMinutes	Type: Edm.Int64

Table 21.14: Properties for Sel.BlueFrame.RestBroker.Plumbing.RestSettings

Relationship of Sel.BlueFrame.RestBroker.Plumbing.RestSettings to other types:

No other data types reference this type.

21.11 Sel.BlueFrame.SecurityManager.SecurityManagerSettings

URL: /api/default/settings/securityManager/

Name	Attributes
enforceUniquePassword-Changes	Type: Edm.Boolean
id	Type: Edm.String
isAdminIntervention-RequiredOnLockout	Type: Edm.Boolean

isUsagePolicy-AcknowledgmentRequired	Type: Edm.Boolean
lockoutSeconds	Type: Edm.Int32
loginAttemptWindow	Type: Edm.Int32
minimumPasswordLength	Max: 32
	Min: 8
	Step: 1
	Type: Edm.Int32
passwordHistorySize	Max: 10
	Min: 1
	Step: 1
	Type: Edm.Int32
passwordUpdate-FrequencyLimitInHours	Max: 168
	Min: 0
	Step: 1
	Type: Edm.Int32
preventPasswordReuse	Type: Edm.Boolean

Table 21.15: Properties for Sel.BlueFrame.SecurityManager.SecurityManagerSettings

Relationship of Sel.BlueFrame.SecurityManager.SecurityManagerSettings to other types:
No other data types reference this type.

21.12 Sel.BlueFrame.TrustAuthority.CrlSettings

URL: </api/default/settings/crlSettings/>

Name	Attributes
crlEnabled	Type: Edm.Boolean
id	Type: Edm.String
protectInternalCA	Type: Edm.Boolean

Table 21.16: Properties for Sel.BlueFrame.TrustAuthority.CrlSettings

Relationship of Sel.BlueFrame.TrustAuthority.CrlSettings to other types:
No other data types reference this type.

21.13 Sel.BlueFrame.TrustAuthority.DataTreeObjects.-CertificateInformation

URL: </api/default/certificate/certificateInfo/>

Name	Attributes
certificateVersion	Type: Edm.Int32
commonName	Type: Edm.String
displayName	Type: Edm.String
hasPrivateKey	Type: Edm.Boolean
id	Type: Edm.String
issuer	Type: Edm.String
purpose	Type: Sel.BlueFrame.Core.TrustAuthority.CertificatePurpose
purposeId	Type: Edm.String
serialNumber	Type: Edm.String
signatureAlgorithm	Type: Edm.String
status	Type: Sel.BlueFrame.Core.TrustAuthority.Enums.CertificateStatus
subject	Type: Edm.String
thumbprint	Type: Edm.String
userInteraction	Type: Sel.BlueFrame.Core.TrustAuthority.Enums.UserInteraction
validEndDate	Type: Edm.DateTimeOffset
validStartDate	Type: Edm.DateTimeOffset

Table 21.17: Properties for Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation

Relationship of Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation to other types:

- [Sel.AsPem](#) (binding parameter)
- [Sel.TrustCertificate](#) (binding parameter)
- [Sel.RevokeCertificate](#) (binding parameter)

21.14 Sel.BlueFrame.UserPreferences.Preference

URL: </api/default/settings/preferences/>

Name	Attributes
id	Type: Edm.String
registry	Type: [Sel.BlueFrame.UserPreferences.Setting]
username	Type: Edm.String
	WriteOnce: true

Table 21.18: Properties for Sel.BlueFrame.UserPreferences.Preference

Relationship of Sel.BlueFrame.UserPreferences.Preference to other types:

No other data types reference this type.

21.15 Sel.Sel5056.LearnAndLock.LearnAndLockSettings

URL: </api/default/settings/learnAndLockSettings/>

Name	Attributes
id	Type: Edm.String
includeControllerIn-LearningRegion	Type: Edm.Boolean
performAutoAdoption	Type: Edm.Boolean
performLogical-ConnectionLearning	Type: Edm.Boolean
performNetworkReset	Type: Edm.Boolean

Table 21.19: Properties for Sel.Sel5056.LearnAndLock.LearnAndLockSettings

Relationship of Sel.Sel5056.LearnAndLock.LearnAndLockSettings to other types:

No other data types reference this type.

21.16 Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.- LearnedLogicalConnection

URL: </api/default/config/LearnedLogicalConnections/>

Name	Attributes
destinationConfigNode-Keys	Type: [Edm.String]
destinationIps	Type: [Edm.String]
dontProposeReverse	Type: Edm.Boolean
id	Type: Edm.String
lastModifiedBy	Type: Edm.String
linkedCSTKey	Type: Edm.String
linkedLCKey	Type: Edm.String
linkedLLSessionKey	Type: Edm.String
proposedCommunication-ServiceType	Type: Sel.Sel5056.LearnAndLock.ProposedCommunicationServiceType
sourceConfigNodeKeys	Type: [Edm.String]
sourceIps	Type: [Edm.String]

state	Type: Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnectionState
-------	--

Table 21.20: Properties for [Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection](#)

Relationship of [Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection](#) to other types:

No other data types reference this type.

21.17 Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

URL: [/api/default/config/LearnAndLockSessions/](#)

Name	Attributes
adoptedHostsInfo	Type: [Sel.Sel5056.LearnAndLock.CSV.AdoptedHostInfo]
adoptedLinksInfo	Type: [Sel.Sel5056.LearnAndLock.CSV.AdoptedLinkInfo]
adoptedSwitchesInfo	Type: [Sel.Sel5056.LearnAndLock.CSV.AdoptedSwitchInfo]
autoAdoptConfig	Type: Sel.Sel5056.LearnAndLock.Configuration.AutoAdoptConfig
autoAdoptSession	Type: Sel.Sel5056.LearnAndLock.Sessions.AutoAdoptSession
autoAdoptionEndReason	Type: Sel.Sel5056.LearnAndLock.Enums.EndReason
autoAdoptionEndedBy-UserOrForReason	Type: Edm.String
autoAdoptionWasRun	Type: Edm.Boolean
beginSessionTime	Type: Edm.DateTimeOffset
endSessionTime	Type: Edm.DateTimeOffset
hardTimeoutRemaining	Type: Edm.Int32
id	Type: Edm.String
isActive	Type: Edm.Boolean
learnAndLockSessionEnd-Reason	Type: Edm.String
logicalConnection-LearningConfig	Type: Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.-Configuration.LogicalConnectionLearningConfig
logicalConnection-LearningEndReason	Type: Sel.Sel5056.LearnAndLock.Enums.EndReason
logicalConnection-LearningEndedByUser	Type: Edm.String
logicalConnection-LearningSession	Type: Sel.Sel5056.LearnAndLock.Sessions.LogicalConnectionLearning-Session
logicalConnection-LearningWasRun	Type: Edm.Boolean

networkResetWasRun	Type: Edm.Boolean
sessionAcknowledged	Type: Edm.Boolean
sessionAcknowledgedTime	Type: Edm.DateTimeOffset
stateBeginEnd-Timestamps	Type: Sel.Sel5056.LearnAndLock.Status.LearnAndLockStateBeginEnd-Timestamps
stateStatus	Type: Sel.Sel5056.LearnAndLock.LearnAndLockStateStatus
statusCounters	Type: Sel.Sel5056.LearnAndLock.Status.LearnAndLockCounters
statusMessages	Type: [Edm.String]
usernameWho-Acknowledged	Type: Edm.String
usernameWhoStarted	Type: Edm.String

Table 21.21: Properties for Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

Relationship of Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession to other types:

- **Sel.BeginLearnAndLock** (binding parameter)
- **Sel.SetDone** (binding parameter)
- **Sel.StopLearnAndLock** (binding parameter)
- **Sel.AcknowledgeLearnAndLockSession** (binding parameter)
- **Sel.AcceptLearnedLogicalConnections** (binding parameter)
- **Sel.ClearLearnedLogicalConnections** (binding parameter)
- **Sel.ProposeLearnedLogicalConnections** (binding parameter)
- **Sel.D declineLearnedLogicalConnections** (binding parameter)
- **Sel.AcceptAllProposedLearnedLogicalConnections** (binding parameter)
- **Sel.D declineAllProposedLearnedLogicalConnections** (binding parameter)
- **Sel.DeleteLearnAndLockSession** (binding parameter)
- **Sel.LearnAndLockIndividualSessionExportCSV** (binding parameter)
- **Sel.GetLearnAndLockCounters** (binding parameter)
- **Sel.GetHardTimeoutTimeRemaining** (binding parameter)
- **Sel.GetStatusMessages** (binding parameter)
- **Sel.GetStateBeginEndTimestamps** (binding parameter)

21.18 Sel.Sel5056.LogicalConnectionValidation.DataObject.-CommunicationServiceType

URL: </api/default/config/communicationServiceTypes/>

Name	Attributes
autoFillSettings	Type: Sel.Sel5056.LogicalConnectionValidation.DataObject.AutoFill-Settings
cstCommunicationType	Type: Sel.Sel5056.LogicalConnectionValidation.Enums.Cst-CommunicationType

displayName	MaxLength: 512
	Type: Edm.String
	Unique: True
errorState	Deprecated: This is not implemented and is only set to success
	Type: Sel.BlueFrame.Core.ErrorState
errors	Deprecated: Errors are never set
	Type: [Edm.String]
id	Type: Edm.String
initialEnabledState	Type: Edm.Boolean
initialHardTimeout	Type: Edm.Int32
numFailableLinks	Type: Edm.Int64
numFailableNodes	Type: Edm.Int64
priority	Type: Edm.Int32
setQueue	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetQueueAction
tags	Type: [Sel.Sel5056.Tags.AmTag]
trafficMatch	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match
version	Type: Edm.Int32

Table 21.22: Properties for Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType

The following types inherit from this type:

-[Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceTypePreDefined](#)

Relationship of Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType to other types:

No other data types reference this type.

21.18.1 Sel.Sel5056.LogicalConnectionValidation.DataObject.-CommunicationServiceTypePreDefined

This type has no properties.

21.19 Sel.Sel5056.LogicalConnectionValidation.DataObject.-LogicalConnection

URL: </api/default/config/logicalConnections/>

Name	Attributes
communicationService-TypeId	Type: Sel.Sel5056.LogicalConnectionValidation.DataObject.-CommunicationServiceType

connectionOrigin	Type: Sel.Sel5056.LogicalConnectionValidation.Enums.Logical-ConnectionOrigin
destinationEndPoints	Type: [Edm.String]
enabled	Type: Edm.Boolean
errorState	Type: Sel.BlueFrame.Core.ErrorState
errors	Type: [Edm.String]
id	Type: Edm.String
lastEnabled	Type: Edm.DateTimeOffset
sourceEndPoints	Type: [Edm.String]
tags	Type: [Sel.Sel5056.Tags.AmTag]
version	Type: Edm.Int32

Table 21.23: Properties for [Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection](#)

Relationship of [Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection](#) to other types:

- [Sel.GetLogicalConnectionStats](#) (binding parameter)
- [Sel.GetOpenFlowInformation](#) (binding parameter)
- [Sel.GetPathPlanInformation](#) (binding parameter)
- [Sel.ResubmitLogicalConnection](#) (binding parameter)
- [Sel.DeleteAll](#) (binding parameter)
- [Sel.ReplanAll](#) (binding parameter)

21.20 [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow](#)

URL: [/api/default/config/flows/](#)

Name	Attributes
bufferId	Type: Edm.Int64
checkOverlap	Type: Edm.Boolean
cookie	Type: Edm.Int64
displayName	MaxLength: 512
	Type: Edm.String
enabled	Type: Edm.Boolean
errorState	Type: Sel.BlueFrame.Core.ErrorState
errors	Type: [Edm.String]
hardTimeout	Type: Edm.Int32
id	Type: Edm.String
idleTimeout	Type: Edm.Int32
inPort	Type: Edm.Int64
instructions	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction]
lastEdit	Type: Edm.DateTimeOffset
match	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match

noByteCounts	Type: Edm.Boolean
noPacketCounts	Type: Edm.Boolean
node	Type: Sel.Sel5056.TopologyManager.Nodes.ConfigNode
outGroup	Type: Edm.Int64
outPort	Type: Edm.Int64
priority	Max: 65535
	Min: 0
	Step: 1
	Type: Edm.Int32
resetCounts	Type: Edm.Boolean
tableId	Max: 255
	Min: 0
	Step: 1
	Type: Edm.Byte
tags	Type: [Sel.Sel5056.Tags.AmTag]
version	Type: Edm.Int32

Table 21.24: Properties for [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow](#)

Relationship of [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow](#) to other types:

—[Sel.ResetCounters](#) (binding parameter)

21.21 [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats](#)

URL: [/api/default/operational/flowStats/](#)

Name	Attributes
dataPathId	Type: Edm.Int64
id	Type: Edm.String
stats	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat]
transactionId	Type: Edm.Int64

Table 21.25: Properties for [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats](#)

Relationship of [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats](#) to other types:

No other data types reference this type.

21.22 [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group](#)

URL: [/api/default/config/groups/](#)

Name	Attributes
buckets	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket]
displayName	MaxLength: 512 Type: Edm.String
enabled	Type: Edm.Boolean
errorState	Type: Sel.BlueFrame.Core.ErrorState
errors	Type: [Edm.String]
groupId	Type: Edm.Int64
groupType	Type: Sel.Sel5056.OpenFlowPlugin.Enums.OfpGroupType
id	Type: Edm.String
node	Type: Sel.Sel5056.TopologyManager.Nodes.ConfigNode
tags	Type: [Sel.Sel5056.Tags.AmTag]
version	Type: Edm.Int32

Table 21.26: Properties for [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group](#)

Relationship of [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group](#) to other types:

—[Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc](#).groups

—[Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc](#)

21.23 [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc](#)

URL: [/api/default/operational/groupDesc/](#)

Name	Attributes
dataPathId	Type: Edm.Int64
groups	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Group]
id	Type: Edm.String
transactionId	Type: Edm.Int64

Table 21.27: Properties for [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc](#)

Relationship of [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupDesc](#) to other types:

No other data types reference this type.

21.24 [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats](#)

URL: [/api/default/operational/groupStats/](#)

Name	Attributes
------	------------

dataPathId	Type: Edm.Int64
id	Type: Edm.String
stats	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat]
transactionId	Type: Edm.Int64

Table 21.28: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats to other types:
No other data types reference this type.

21.25 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter

URL: </api/default/config/meters/>

Name	Attributes
displayName	MaxLength: 512
	Type: Edm.String
enabled	Type: Edm.Boolean
errorState	Type: Sel.BlueFrame.Core.ErrorState
errors	Type: [Edm.String]
flags	Type: [Sel.Sel5056.TopologyManager.OpenFlow.Meter.-RateTypeFlagObjects.RateTypeBase]
id	Type: Edm.String
meterBands	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand]
meterId	Type: Edm.Int64
node	Type: Sel.Sel5056.TopologyManager.Nodes.ConfigNode
tags	Type: [Sel.Sel5056.Tags.AmTag]
version	Type: Edm.Int32

Table 21.29: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter to other types:
No other data types reference this type.

21.26 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter-Descriptions

URL: </api/default/operational/meterDescriptions/>

Name	Attributes
dataPathId	Type: Edm.Int64
id	Type: Edm.String
meterDescriptionsList	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDesc]
transactionId	Type: Edm.Int64

Table 21.30: Properties for [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions](#)

Relationship of [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions](#) to other types:
No other data types reference this type.

21.27 [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures](#)

URL: [/api/default/operational/meterFeatures/](#)

Name	Attributes
bandTypes	Type: Sel.Sel5056.OpenFlowPlugin.Enums.OF13MeterBandType
capabilities	Type: [Sel.Sel5056.TopologyManager.OpenFlow.Meter.-RateTypeFlagObjects.RateTypeBase]
dataPathId	Type: Edm.Int64
id	Type: Edm.String
maxBands	Type: Edm.Byte
maxColor	Type: Edm.Byte
maxMeters	Type: Edm.Int64

Table 21.31: Properties for [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures](#)

Relationship of [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures](#) to other types:
No other data types reference this type.

21.28 [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats](#)

URL: [/api/default/operational/meterstats/](#)

Name	Attributes
dataPathId	Type: Edm.Int64
id	Type: Edm.String
meterStatList	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStat]

transactionId	Type: Edm.Int64
---------------	---------------------------------

Table 21.32: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats to other types:
No other data types reference this type.

21.29 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc

URL: </api/default/operational/portDesc/>

Name	Attributes
dataPathId	Type: Edm.Int64
id	Type: Edm.String
ports	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port]
transactionId	Type: Edm.Int64

Table 21.33: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc to other types:
No other data types reference this type.

21.30 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats

URL: </api/default/operational/portStats/>

Name	Attributes
dataPathId	Type: Edm.Int64
id	Type: Edm.String
portStatList	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStat]

Table 21.34: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats to other types:
No other data types reference this type.

21.31 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ReceivedPacket

URL: </api/default/operational/packets/>

Name	Attributes
contents	Type: Edm.String
cookie	Type: Edm.Int64
dataPathId	Type: Edm.Int64
id	Type: Edm.String
monotonicId	Type: Edm.Int64
port	Type: Edm.Int64
prettyDataPathId	Type: Edm.String
receivedAt	Type: Edm.Int64

Table 21.35: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ReceivedPacket

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ReceivedPacket to other types:
No other data types reference this type.

21.32 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats

URL: </api/default/operational/tableStats/>

Name	Attributes
dataPathId	Type: Edm.Int64
id	Type: Edm.String
tableStatList	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStat]

Table 21.36: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats to other types:
No other data types reference this type.

21.33 Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation

URL: </api/default/config/vlanVidReservation/>

Name	Attributes
creator	Type: Edm.String
id	Type: Edm.String
uniqueId	Type: Edm.String
vlanVid	Max: 4094
	Min: 1
	Step: 1
	Type: Edm.Int32

Table 21.37: Properties for Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation

The following types inherit from this type:

-[Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidDestinationReservation](#)

Relationship of Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation to other types:

—[Sel.ReserveVlanVidRange](#) (binding parameter)

21.33.1 Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVid-DestinationReservation

Name	Attributes
amDestination	Type: Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.-DestinationDescriptions.Types.AmDestination

Table 21.38: Property for Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidDestinationReservation

21.34 Sel.Sel5056.Snmp.SnmpUser

URL: [/api/default/security/snmpUsers/](#)

Name	Attributes
authenticationMethod	Type: Sel.Sel5056.Snmp.SnmpAuthenticationMethod
authenticationPassword	CharacterSet: Printable7BitAscii
	MaxLength: 128
	MinLength: 8
	Type: Edm.String
encryptionMethod	Type: Sel.Sel5056.Snmp.SnmpEncryptionMethod
encryptionPassword	CharacterSet: Printable7BitAscii
	MaxLength: 128
	MinLength: 8
	Type: Edm.String
id	Type: Edm.String

username	CharacterSet: Printable7BitAscii
	MaxLength: 256
	MinLength: 6
	Type: Edm.String

Table 21.39: Properties for Sel.Sel5056.Snmp.SnmpUser

Relationship of Sel.Sel5056.Snmp.SnmpUser to other types:
No other data types reference this type.

21.35 Sel.Sel5056.SystemMessaging.SystemMessage

URL: </api/default/settings/systemMessages/>

Name	Attributes
id	Type: Edm.String
message	Type: Edm.String
module	Type: Edm.String
severity	Type: Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel
type	Type: Edm.String

Table 21.40: Properties for Sel.Sel5056.SystemMessaging.SystemMessage

Relationship of Sel.Sel5056.SystemMessaging.SystemMessage to other types:
No other data types reference this type.

21.36 Sel.Sel5056.TopologyManager.Links.ConfigLink

URL: </api/default/config/links/>

Name	Attributes
configurations	Type: [Sel.Sel5056.TopologyManager.Configurations.AmConfig]
displayName	AcceptServerChanges: true
	MaxLength: 512
	MinLength: 0
	Type: Edm.String
endPoints	Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort
first	Type: Edm.String
id	Type: Edm.String
last	Type: Edm.String

linkedKey	Type: Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink
state	Type: Sel.Sel5056.TopologyManager.Enums.State
tags	Type: [Sel.Sel5056.Tags.AmTag]
weight	Max: 2147483647
	Min: 1
	Step: 1
	Type: Edm.Int64

Table 21.41: Properties for [Sel.Sel5056.TopologyManager.Links.ConfigLink](#)

Relationship of [Sel.Sel5056.TopologyManager.Links.ConfigLink](#) to other types:

No other data types reference this type.

21.37 [Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink](#)

URL: [/api/default/operational/links/](#)

Name	Attributes
attributes	Type: [Sel.Sel5056.TopologyManager.Attributes.AmAttribute]
displayName	Type: Edm.String
endPoints	Type: [Edm.String]
first	Type: Edm.String
id	Type: Edm.String
last	Type: Edm.String
linkedKey	Type: Sel.Sel5056.TopologyManager.Links.ConfigLink
state	Type: Sel.Sel5056.TopologyManager.Enums.State
trustState	Type: Sel.Sel5056.TopologyManager.Enums.DiscoveryTrustState

Table 21.42: Properties for [Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink](#)

Relationship of [Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink](#) to other types:

- [Sel.Adopt](#) (binding parameter)
- [Sel.AdoptWithConfig](#) (binding parameter)
- [Sel.Unadopt](#) (binding parameter)
- [Sel.ReplaceConfig](#) (binding parameter)

21.38 [Sel.Sel5056.TopologyManager.NetworkSettings](#)

URL: [/api/default/settings/networkSettings/](#)

Name	Attributes
controllerArpSrcIp-Address	DataClass: IPAddress Type: Edm.String
controllerArpSrcNetwork-Addresses	Type: [Sel.BlueFrame.Core.Networking.NetworkInterfaceAddress]
controllerIp	DataClass: IPAddress Type: Edm.String
defaultGateway	DataClass: IPAddress Type: Edm.String
id	Type: Edm.String
ntpServers	Type: [Edm.String]

Table 21.43: Properties for Sel.Sel5056.TopologyManager.NetworkSettings

Relationship of Sel.Sel5056.TopologyManager.NetworkSettings to other types:
No other data types reference this type.

21.39 Sel.Sel5056.TopologyManager.Nodes.ConfigNode

URL: [/api/default/config/nodes/](#)

Name	Attributes
configurations	Type: [Sel.Sel5056.TopologyManager.Configurations.AmConfig]
displayName	AcceptServerChanges: true MaxLength: 512 MinLength: 0 Type: Edm.String
id	Type: Edm.String
linkedKey	Type: Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
ports	Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort
state	Type: Sel.Sel5056.TopologyManager.Enums.State
tags	Type: [Sel.Sel5056.Tags.AmTag]

Table 21.44: Properties for Sel.Sel5056.TopologyManager.Nodes.ConfigNode

The following types inherit from this type:

- [Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode](#)
- [Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode](#)

Relationship of Sel.Sel5056.TopologyManager.Nodes.ConfigNode to other types:
No other data types reference this type.

21.39.1 Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

Name	Attributes
additionalTrusted-Certificates	Type: [Edm.String]
alarmMinimumDuration	Max: 30
	Min: 1
	Step: 1
	Type: Edm.Int64
controllerIp	DataClass: IPAddress
	Type: Edm.String
defaultGateway	DataClass: IPAddress
	Type: Edm.String
enablePtp	Type: Edm.Boolean
eventCategories	Type: [Sel.BlueFrame.Core.EventBus.EventCategory]
gigabitCopperFastFailover	Type: Edm.Boolean
ipAddress	Type: Edm.String
	Unique: True
ntpServers	Type: [Edm.String]
portLinkEvent-Configurations	Type: [Sel.Sel5056.TopologyManager.Nodes.PortLinkEventConfiguration]
subnetMask	DataClass: IPAddress
	Type: Edm.String
timeSourceSelection	Type: Sel.Sel5056.TopologyManager.Enums.Sel274xTimeSource

Table 21.45: Properties for Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

21.39.2 Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode

Name	Attributes
digitalInput1Function	Type: Sel.Sel5056.TopologyManager.Enums.Sel2742SDigitalInput-Functions
digitalInput2Function	Type: Sel.Sel5056.TopologyManager.Enums.Sel2742SDigitalInput-Functions
enablePinHoleReset	Type: Edm.Boolean
enablePoePort5	Type: Edm.Boolean
enablePoePort6	Type: Edm.Boolean

Table 21.46: Properties for Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode

21.40 Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

URL: </api/default/operational/nodes/>

Name	Attributes
attributes	Type: [Sel.Sel5056.TopologyManager.Attributes.AmAttribute]
displayName	Type: Edm.String
id	Type: Edm.String
linkedKey	Type: Sel.Sel5056.TopologyManager.Nodes.ConfigNode
ports	Type: [Edm.String]
state	Type: Sel.Sel5056.TopologyManager.Enums.State
trustState	Type: Sel.Sel5056.TopologyManager.Enums.DiscoveryTrustState

Table 21.47: Properties for Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

Relationship of Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode to other types:

- [Sel.Disconnect](#) (binding parameter)
- [Sel.DisableControllerArpToNode](#) (binding parameter)
- [Sel.EnableControllerArpToNode](#) (binding parameter)
- [Sel.PacketOut](#) (binding parameter)
- [Sel.PacketOutToTable](#) (binding parameter)
- [Sel.FactoryDefaultReset](#) (binding parameter)
- [Sel.FirmwareUpgrade](#) (binding parameter)
- [Sel.Reboot](#) (binding parameter)
- [Sel.ReplanInbandPath](#) (binding parameter)

21.41 Sel.Sel5056.TopologyManager.Ports.ConfigPort

URL: </api/default/config/ports/>

Name	Attributes
attachedLinks	Type: [Edm.String]
configurations	Type: [Sel.Sel5056.TopologyManager.Configurations.AmConfig]
displayName	AcceptServerChanges: true
	MaxLength: 512
	MinLength: 0
	Type: Edm.String
displayNameSetByUser	Type: Edm.Boolean
id	Type: Edm.String
linkedKey	Type: Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort
parentNode	Type: Sel.Sel5056.TopologyManager.Nodes.ConfigNode

state	Type: Sel.Sel5056.TopologyManager.Enums.State
tags	Type: [Sel.Sel5056.Tags.AmTag]

Table 21.48: Properties for [Sel.Sel5056.TopologyManager.Ports.ConfigPort](#)

Relationship of [Sel.Sel5056.TopologyManager.Ports.ConfigPort](#) to other types:

—[Sel.SendPortMod](#) (binding parameter)

21.42 [Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort](#)

URL: [/api/default/operational/ports/](#)

Name	Attributes
attachedLinks	Type: [Edm.String]
attributes	Type: [Sel.Sel5056.TopologyManager.Attributes.AmAttribute]
displayName	Type: Edm.String
id	Type: Edm.String
isConnected	Type: Edm.Boolean
linkedKey	Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort
parentNode	Type: Edm.String
state	Type: Sel.Sel5056.TopologyManager.Enums.State
trustState	Type: Sel.Sel5056.TopologyManager.Enums.DiscoveryTrustState

Table 21.49: Properties for [Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort](#)

Relationship of [Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort](#) to other types:

—[Sel.MarkPortAsSelRelayFailover](#) (binding parameter)
 —[Sel.RemoveSelRelayFailoverFromPort](#) (binding parameter)
 —[Sel.DetectSelRelayFailoverPort](#) (binding parameter)
 —[Sel.MarkPortAndPerformBlockingDetectSelRelayFailover](#) (binding parameter)
 —[Sel.AddSilentHost](#) (binding parameter)
 —[Sel.AddOperationalLinkFromHostToSwitch](#) (binding parameter)
 —[Sel.MarkPortAndAddRelayFailoverLink](#) (binding parameter)
 —[Sel.AddAbstractNode](#) (binding parameter)
 —[Sel.SendPortMod](#) (binding parameter)

21.43 Sel.Sel5056.TopologyManager.Synchronization.-SynchronizationRequest

URL: </api/default/operational/synchronization/>

Name	Attributes
configNodeId	Type: Edm.String
description	Type: Edm.String
errorState	Type: Sel.BlueFrame.Core.ErrorState
errors	Type: [Edm.String]
id	Type: Edm.String
version	Type: Edm.Int32

Table 21.50: 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.OpenFlowPlugin.PortModController.PortModSynchronization](#)
- [Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization](#)
- [Sel.Sel5056.SapphirePlugin.Synchronization.FirmwareUpgradeRequiredSynchronization](#)

Relationship of Sel.Sel5056.TopologyManager.Synchronization.SynchronizationRequest to other types:
No other data types reference this type.

21.43.1 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.Flow-AddSynchronizationRequest

Name	Attributes
alias	Type: Edm.String
configFlowId	Type: Edm.String
cookie	Type: Edm.Int64
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.Open-FlowDependency]

priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-SystemPriorities.AmSystemPriority
-----------------------	--

Table 21.51: Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowAddSynchronizationRequest

21.43.2 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowDeleteSynchronizationRequest

Name	Attributes
alias	Type: Edm.String
cookie	Type: Edm.Int64
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]
priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-SystemPriorities.AmSystemPriority
referencedGroupId	Type: Edm.Int64

Table 21.52: Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowDeleteSynchronizationRequest

21.43.3 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModifySynchronizationRequest

Name	Attributes
alias	Type: Edm.String
configFlowId	Type: Edm.String
cookie	Type: Edm.Int64
currentGroupId	Type: Edm.Int64
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]
newGroupId	Type: Edm.Int64
priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-SystemPriorities.AmSystemPriority

Table 21.53: Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.FlowModifySynchronizationRequest

21.43.4 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.-GroupAddSynchronizationRequest

Name	Attributes
------	------------

alias	Type: Edm.String
configGroupId	Type: Edm.String
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]
groupId	Type: Edm.Int64
priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-SystemPriorities.AmSystemPriority

Table 21.54: Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupAddSynchronizationRequest

21.43.5 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.-GroupDeleteSynchronizationRequest

Name	Attributes
alias	Type: Edm.String
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]
groupId	Type: Edm.Int64
priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-SystemPriorities.AmSystemPriority

Table 21.55: Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupDeleteSynchronizationRequest

21.43.6 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.-GroupModifySynchronizationRequest

Name	Attributes
alias	Type: Edm.String
configGroupId	Type: Edm.String
currentGroupIds	Type: [Edm.Int64]
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]
groupId	Type: Edm.Int64
newGroupIds	Type: [Edm.Int64]
priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-SystemPriorities.AmSystemPriority

Table 21.56: Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.GroupModifySynchronizationRequest

21.43.7 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.-MeterAddSynchronizationRequest

Name	Attributes
alias	Type: Edm.String
configMeterId	Type: Edm.String
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]
meterId	Type: Edm.Int64
priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-SystemPriorities.AmSystemPriority

Table 21.57: Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterAddSynchronizationRequest

21.43.8 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.-MeterDeleteSynchronizationRequest

Name	Attributes
alias	Type: Edm.String
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]
meterId	Type: Edm.Int64
priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-SystemPriorities.AmSystemPriority

Table 21.58: Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterDeleteSynchronizationRequest

21.43.9 Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.-MeterModifySynchronizationRequest

Name	Attributes
alias	Type: Edm.String
configMeterId	Type: Edm.String
dependencies	Type: [Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency]
meterId	Type: Edm.Int64
priorityConfiguration	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.-SystemPriorities.AmSystemPriority

Table 21.59: Properties for Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterModifySynchronizationRequest

21.43.10 Sel.Sel5056.OpenFlowPlugin.PortModController.PortModSynchronization

Name	Attributes
configSettings	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-ConfigFlagObjects.ConfigBase]
featureSettings	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.FeatureBase]
operationalPortId	Type: Edm.String

Table 21.60: Properties for Sel.Sel5056.OpenFlowPlugin.PortModController.PortModSynchronization

21.43.11 Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization

Name	Attributes
endpoint	Type: Sel.Sel5056.SapphirePlugin.Synchronization.SapphireEndpoint
key	Type: Edm.String

Table 21.61: Properties for Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization

21.43.12 Sel.Sel5056.SapphirePlugin.Synchronization.FirmwareUpgradeRequiredSynchronization

This type has no properties.

Chapter 22

Secondary OData Types

This section describes the OData types that are properties of the base types described in the **Primary OData types** chapter.

22.1 Sel.BlueFrame.ApplicationRegistration.AuthenticationType

Name	Attributes
clientId	Type: Edm.String
grantType	Type: Edm.String
permissions	MinLength: 1
	Type: [Sel.BlueFrame.Core.SecurityManager.Permission]
redirectUri	Type: [Edm.String]

Table 22.1: Properties for Sel.BlueFrame.ApplicationRegistration.AuthenticationType

Relationship of Sel.BlueFrame.ApplicationRegistration.AuthenticationType to other types:

—Sel.BlueFrame.ApplicationRegistration.ApplicationLink.authenticationTypes

——**Sel.BlueFrame.ApplicationRegistration.ApplicationLink**

22.2 Sel.BlueFrame.CommissioningManager.CommissioningConcern

This type has no properties.

The following types inherit from this type:

-**Sel.BlueFrame.CommissioningManager.UserCommissioner.UserCommissioningConcern**

Relationship of Sel.BlueFrame.CommissioningManager.CommissioningConcern to other types:

—Sel.Commission.commissioningList

22.2.1 Sel.BlueFrame.CommissioningManager.UserCommissioner.User-CommissioningConcern

Name	Attributes
password	CharacterSet: Printable7BitAscii
	MaxLength: 128
	MinLength: 8
	Type: Edm.String
username	CharacterSet: Printable7BitAscii
	MaxLength: 128
	MinLength: 1
	Type: Edm.String

Table 22.2: Properties for Sel.BlueFrame.CommissioningManager.UserCommissioner.User-CommissioningConcern

22.3 Sel.BlueFrame.Core.EventBus.Behaviors.Behavior

Name	Attributes
name	Type: Edm.String
severity	Type: Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel

Table 22.3: 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

Relationship of Sel.BlueFrame.Core.EventBus.Behaviors.Behavior to other types:

- 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**

22.3.1 Sel.BlueFrame.EventBus.Syslog.SyslogBehavior

Name	Attributes
ipAddress	DataClass: IPAddress
	Type: Edm.String
	Max: 65535

port

	Min: 1
	Step: 1
	Type: Edm.Int32
rfcType	Type: Sel.BlueFrame.EventBus.Syslog.RFCType
transportType	Type: Sel.BlueFrame.EventBus.Syslog.TransportType

Table 22.4: Properties for Sel.BlueFrame.EventBus.Syslog.SyslogBehavior

22.3.2 Sel.BlueFrame.EventBus.WindowsEventLog.WindowsEventLogBehavior

Name	Attributes
alertMap	Type: System.Diagnostics.EventLogEntryType
criticalMap	Type: System.Diagnostics.EventLogEntryType
emergencyMap	Type: System.Diagnostics.EventLogEntryType
errorMap	Type: System.Diagnostics.EventLogEntryType
informationalMap	Type: System.Diagnostics.EventLogEntryType
noticeMap	Type: System.Diagnostics.EventLogEntryType
warningMap	Type: System.Diagnostics.EventLogEntryType

Table 22.5: Properties for Sel.BlueFrame.EventBus.WindowsEventLog.WindowsEventLogBehavior

22.3.3 Sel.BlueFrame.Particle.AlarmContact.AlarmContactBehavior

This type has no properties.

22.3.4 Sel.BlueFrame.Particle.LocalEventStore.LocalEventStoreBehavior

This type has no properties.

22.4 Sel.BlueFrame.Core.EventBus.LocaleRecord

Name	Attributes
locale	Type: Edm.String
value	Type: Edm.String

Table 22.6: Properties for Sel.BlueFrame.Core.EventBus.LocaleRecord

Relationship of Sel.BlueFrame.Core.EventBus.LocaleRecord to other types:

—Sel.BlueFrame.Core.EventBus.LocaleString.records

—[Sel.BlueFrame.Core.EventBus.LocaleString](#)

22.5 Sel.BlueFrame.Core.FileArgument

Name	Attributes
fileName	Type: Edm.String

Table 22.7: Property for Sel.BlueFrame.Core.FileArgument

Relationship of Sel.BlueFrame.Core.FileArgument to other types:

—Sel.Restore.fileName

22.6 Sel.BlueFrame.Core.FileReturnValue

Name	Attributes
contentType	Type: Edm.String
fileName	Type: Edm.String

Table 22.8: Properties for Sel.BlueFrame.Core.FileReturnValue

Relationship of Sel.BlueFrame.Core.FileReturnValue to other types:

No other data types reference this type.

22.7 Sel.BlueFrame.Core.Networking.NetworkInterfaceAddress

Name	Attributes
fullAddress	Type: Edm.String

Table 22.9: Property for Sel.BlueFrame.Core.Networking.NetworkInterfaceAddress

Relationship of Sel.BlueFrame.Core.Networking.NetworkInterfaceAddress to other types:

—Sel.Sel5056.TopologyManager.NetworkSettings.controllerArpSrcNetworkAddresses

—[Sel.Sel5056.TopologyManager.NetworkSettings](#)

22.8 Sel.BlueFrame.Core.SecurityManager.Password

Name	Attributes
passwordCreationTime	Type: Edm.DateTimeOffset

Table 22.10: Property for Sel.BlueFrame.Core.SecurityManager.Password

Relationship of Sel.BlueFrame.Core.SecurityManager.Password to other types:

- Sel.BlueFrame.Core.SecurityManager.LocalUser.passwordStore
- Sel.BlueFrame.Core.SecurityManager.LocalUser
- Sel.UpdatePassword (binding parameter)

22.9 Sel.BlueFrame.Core.SecurityManager.Permission

Name	Attributes
module	Type: Edm.String
permissionTag	Type: Edm.String

Table 22.11: 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

Relationship of Sel.BlueFrame.Core.SecurityManager.Permission to other types:

- 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](#)

22.9.1 Sel.BlueFrame.Core.DataBrokerPermission

Name	Attributes
dataType	Type: Edm.String
tree	Type: Edm.String

Table 22.12: Properties for Sel.BlueFrame.Core.DataBrokerPermission

22.9.2 Sel.BlueFrame.Core.SecurityAttributePermission

Name	Attributes
securityAttribute	Type: Edm.String

Table 22.13: Property for Sel.BlueFrame.Core.SecurityAttributePermission

22.9.3 Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission

Name	Attributes
purpose	Type: Edm.String

Table 22.14: Property for Sel.BlueFrame.Core.TrustAuthority.TrustAuthorityPermission

22.9.4 Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission

Name	Attributes
deviceType	Type: Edm.String

Table 22.15: Property for Sel.Sel5056.Southbound.DeviceManagement.DeviceManagementPermission

22.10 Sel.BlueFrame.Core.SecurityManager.UserFailedLogin

Name	Attributes
ipAddress	Type: Edm.String
loginTime	Type: Edm.DateTimeOffset

Table 22.16: Properties for Sel.BlueFrame.Core.SecurityManager.UserFailedLogin

Relationship of Sel.BlueFrame.Core.SecurityManager.UserFailedLogin to other types:

—Sel.BlueFrame.Core.SecurityManager.User.failedLogins

——**Sel.BlueFrame.Core.SecurityManager.User**

22.11 Sel.BlueFrame.Core.TrustAuthority.CertificatePurpose

This type has no properties.

The following types inherit from this type:

- Sel.BlueFrame.TrustAuthority.TrustedCertificates.TrustedCertificatePurpose**
- Sel.BlueFrame.TrustAuthority.WebCertificateManager.WebCertificatePurpose**
- Sel.BlueFrame.TrustAuthority.InternalCertificateAuthority.InternalCaPurpose**
- Sel.Sel5056.Drivers.OpenFlow.Trust.OpenFlowDeviceCertificate**
- Sel.Sel5056.Drivers.OpenFlow.Trust.OpenFlowServerCertificate**
- Sel.Sel5056.SapphirePlugin.Trust.SapphireDeviceCertificate**
- Sel.Sel5056.SapphirePlugin.Trust.SapphireServerCertificate**

Relationship of Sel.BlueFrame.Core.TrustAuthority.CertificatePurpose to other types:

—Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation.purpose

——**Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation**

—Sel.RevokeCertificateByPurpose.purpose

—Sel.UploadCertificate.purpose

22.11.1 Sel.BlueFrame.TrustAuthority.TrustedCertificates.TrustedCertificatePurpose

This type has no properties.

22.11.2 Sel.BlueFrame.TrustAuthority.WebCertificateManager.WebCertificatePurpose

This type has no properties.

22.11.3 Sel.BlueFrame.TrustAuthority.InternalCertificateAuthority.-InternalCaPurpose

This type has no properties.

22.11.4 Sel.Sel5056.Drivers.OpenFlow.Trust.OpenFlowDeviceCertificate

Name	Attributes
dataPathId	Type: Edm.Int64

Table 22.17: Property for Sel.Sel5056.Drivers.OpenFlow.Trust.OpenFlowDeviceCertificate

22.11.5 Sel.Sel5056.Drivers.OpenFlow.Trust.OpenFlowServerCertificate

This type has no properties.

22.11.6 Sel.Sel5056.SapphirePlugin.Trust.SapphireDeviceCertificate

Name	Attributes
ipAddress	DataClass: IPAddress
	Type: Edm.String
serialNumber	Type: Edm.String

Table 22.18: Properties for Sel.Sel5056.SapphirePlugin.Trust.SapphireDeviceCertificate

22.11.7 Sel.Sel5056.SapphirePlugin.Trust.SapphireServerCertificate

This type has no properties.

22.12 Sel.BlueFrame.SecurityManager.AuthServiceGroupDescription

Name	Attributes
displayName	Type: Edm.String
name	Type: Edm.String

Table 22.19: Properties for Sel.BlueFrame.SecurityManager.AuthServiceGroupDescription

Relationship of Sel.BlueFrame.SecurityManager.AuthServiceGroupDescription to other types:
 —Sel.BlueFrame.SecurityManager.GroupCollection.groups

22.13 Sel.BlueFrame.SecurityManager.AuthServiceResult

Name	Attributes
attemptedTest	Type: Edm.String
message	Type: Edm.String
state	Type: Sel.BlueFrame.SecurityManager.AuthServiceResultState

Table 22.20: Properties for Sel.BlueFrame.SecurityManager.AuthServiceResult

Relationship of Sel.BlueFrame.SecurityManager.AuthServiceResult to other types:
 —Sel.BlueFrame.SecurityManager.AuthServiceResultCollection.results

22.14 Sel.BlueFrame.SecurityManager.AuthServiceResultCollection

Name	Attributes
results	Type: [Sel.BlueFrame.SecurityManager.AuthServiceResult]

Table 22.21: Property for Sel.BlueFrame.SecurityManager.AuthServiceResultCollection

Relationship of Sel.BlueFrame.SecurityManager.AuthServiceResultCollection to other types:
 No other data types reference this type.

22.15 Sel.BlueFrame.SecurityManager.GroupCollection

Name	Attributes
groups	Type: [Sel.BlueFrame.SecurityManager.AuthServiceGroupDescription]

Table 22.22: Property for Sel.BlueFrame.SecurityManager.GroupCollection

Relationship of Sel.BlueFrame.SecurityManager.GroupCollection to other types:
No other data types reference this type.

22.16 Sel.BlueFrame.UserPreferences.Setting

Name	Attributes
key	Type: Edm.String
value	Type: Edm.String

Table 22.23: Properties for Sel.BlueFrame.UserPreferences.Setting

Relationship of Sel.BlueFrame.UserPreferences.Setting to other types:
—Sel.BlueFrame.UserPreferences.Preference.registry
——[Sel.BlueFrame.UserPreferences.Preference](#)

22.17 Sel.FileArgument

This type has no properties.

Relationship of Sel.FileArgument to other types:
—Sel.FirmwareUpgrade.fileName

22.18 Sel.PathPlanHop

This type has no properties.

Relationship of Sel.PathPlanHop to other types:
—Sel.Sel5056.PathPlanner.DataObject.PathPlanHopCollection.pathPlanHops

22.19 Sel.Sel5056.Common.OpenFlowNodeProgrammer.-Options.FlowProgrammingOptions

Name	Attributes
displayName	MaxLength: 128
	Type: Edm.String
match	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match
meterMetadata	Type: Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.Meter-Metadata

nodesSetting	Deprecated: This property has no behavior associated with it, do not use it
	Type: <code>Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.Nodes-Configuration</code>
outputSetting	Type: <code>Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.Output-Configuration</code>
priority	Type: <code>Edm.Int32</code>
tableId	Type: <code>Edm.Byte</code>
tableSetting	Type: <code>Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.Table-Configuration</code>

Table 22.24: Properties for `Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions`

Relationship of `Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions` to other types:

No other data types reference this type.

22.20 Sel.Sel5056.Common.OpenFlowNodeProgrammer.-Options.MeterMetadata

Name	Attributes
flags	Type: [<code>Sel.Sel5056.TopologyManager.OpenFlow.Meter.-RateTypeFlagObjects.RateTypeBase</code>]
meterBands	Type: [<code>Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand</code>]
meterId	Type: <code>Edm.Int64</code>

Table 22.25: Properties for `Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.MeterMetadata`

Relationship of `Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.MeterMetadata` to other types:
—`Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions.meterMetadata`

22.21 Sel.Sel5056.LearnAndLock.CSV.AdoptedHostInfo

Name	Attributes
connectedSwitchPort1	Type: <code>Edm.String</code>
connectedSwitchPort2	Type: <code>Edm.String</code>
displayName	Type: <code>Edm.String</code>
id	Type: <code>Edm.String</code>
ipAddress	Type: <code>Edm.String</code>

redundancyDetected	Type: Edm.Boolean
--------------------	-----------------------------------

Table 22.26: Properties for Sel.Sel5056.LearnAndLock.CSV.AdoptedHostInfo

Relationship of Sel.Sel5056.LearnAndLock.CSV.AdoptedHostInfo to other types:

—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.adoptedHostsInfo

——[Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession](#)

22.22 Sel.Sel5056.LearnAndLock.CSV.AdoptedLinkInfo

Name	Attributes
endpoint_1_Node	Type: Edm.String
endpoint_1_Port	Type: Edm.String
endpoint_2_Node	Type: Edm.String
endpoint_2_Port	Type: Edm.String

Table 22.27: Properties for Sel.Sel5056.LearnAndLock.CSV.AdoptedLinkInfo

Relationship of Sel.Sel5056.LearnAndLock.CSV.AdoptedLinkInfo to other types:

—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.adoptedLinksInfo

——[Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession](#)

22.23 Sel.Sel5056.LearnAndLock.CSV.AdoptedSwitchInfo

Name	Attributes
dataPathId	Type: Edm.Int64
displayName	Type: Edm.String
ipAddress	Type: Edm.String
type	Type: Edm.String

Table 22.28: Properties for Sel.Sel5056.LearnAndLock.CSV.AdoptedSwitchInfo

Relationship of Sel.Sel5056.LearnAndLock.CSV.AdoptedSwitchInfo to other types:

—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.adoptedSwitchesInfo

——[Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession](#)

22.24 Sel.Sel5056.LearnAndLock.Configuration.AutoAdopt-Config

Name	Attributes
------	------------

configNodes	Type: [Edm.String]
endingIpAddress	Type: Edm.String
maxNumberOfHostsToAdopt	Type: Edm.Int64
maxNumberOfOpenFlowSwitchesToAdopt	Type: Edm.Int64
startingIpAddress	Type: Edm.String
unlimitedNumberOfHostsToAdopt	Type: Edm.Boolean
unlimitedNumberOfOpenFlowSwitchesToAdopt	Type: Edm.Boolean

Table 22.29: Properties for [Sel.Sel5056.LearnAndLock.Configuration.AutoAdoptConfig](#)

Relationship of [Sel.Sel5056.LearnAndLock.Configuration.AutoAdoptConfig](#) to other types:

- [Sel.BeginLearnAndLock.autoAdoptConfig](#)
- [Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.autoAdoptConfig](#)
- [Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession](#)

22.25 [Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.Configuration.LogicalConnectionLearningConfig](#)

Name	Attributes
additionalLearningRegionIps	Type: [Edm.String]
autoAcceptStatus	Type: Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.AutoAcceptStatus
enableMulticastLearning	Type: Edm.Boolean
hardTimeoutSeconds	Max: 604800
	Min: 0
	Step: 1
	Type: Edm.Int32

Table 22.30: Properties for [Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.Configuration.LogicalConnectionLearningConfig](#)

Relationship of [Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.Configuration.LogicalConnectionLearningConfig](#) to other types:

- [Sel.BeginLearnAndLock.logicalConnectionLearningConfig](#)
- [Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.logicalConnectionLearningConfig](#)
- [Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession](#)

22.26 Sel.Sel5056.LearnAndLock.ProposedCommunication-ServiceType

Name	Attributes
cstCommunicationType	Type: Sel.Sel5056.LogicalConnectionValidation.Enums.Cst-CommunicationType
displayName	MaxLength: 128
	Type: Edm.String
	Unique: True
matchableCst	Type: Edm.Boolean
numFailableLinks	Type: Edm.Int64
numFailableNodes	Type: Edm.Int64
priority	Type: Edm.Int32
setQueue	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetQueueAction
trafficMatch	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match

Table 22.31: Properties for Sel.Sel5056.LearnAndLock.ProposedCommunicationServiceType

Relationship of Sel.Sel5056.LearnAndLock.ProposedCommunicationServiceType to other types:

—Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection.proposed-CommunicationServiceType

——[Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection](#)

22.27 Sel.Sel5056.LearnAndLock.Sessions.AutoAdoptSession

Name	Attributes
status	Type: Sel.Sel5056.LearnAndLock.LearnAndLockStateStatus

Table 22.32: Property for Sel.Sel5056.LearnAndLock.Sessions.AutoAdoptSession

Relationship of Sel.Sel5056.LearnAndLock.Sessions.AutoAdoptSession to other types:

—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.autoAdoptSession

——[Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession](#)

22.28 Sel.Sel5056.LearnAndLock.Sessions.LogicalConnection-LearningSession

Name	Attributes
status	Type: Sel.Sel5056.LearnAndLock.LearnAndLockStateStatus

Table 22.33: Property for Sel.Sel5056.LearnAndLock.Sessions.LogicalConnectionLearningSession

Relationship of Sel.Sel5056.LearnAndLock.Sessions.LogicalConnectionLearningSession to other types:
 —Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.logicalConnectionLearningSession
 —[Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession](#)

22.29 Sel.Sel5056.LearnAndLock.Status.LearnAndLockCounters

Name	Attributes
hostsAdopted	Type: Edm.Int64
logicalConnections-Accepted	Type: Edm.Int64
logicalConnections-Declined	Type: Edm.Int64
logicalConnections-Programmed	Type: Edm.Int64
logicalConnections-Proposed	Type: Edm.Int64
switchesAdopted	Type: Edm.Int64

Table 22.34: Properties for Sel.Sel5056.LearnAndLock.Status.LearnAndLockCounters

Relationship of Sel.Sel5056.LearnAndLock.Status.LearnAndLockCounters to other types:
 —Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.statusCounters
 —[Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession](#)

22.30 Sel.Sel5056.LearnAndLock.Status.LearnAndLockState-BeginEndTimestamps

Name	Attributes
beginAutoAdoption	Type: Edm.DateTimeOffset
beginInbandManagement-Replanning	Type: Edm.DateTimeOffset
beginLogicalConnection-Learning	Type: Edm.DateTimeOffset
beginNetworkReset	Type: Edm.DateTimeOffset
beginRelayFailover-Detection	Type: Edm.DateTimeOffset
endAutoAdoption	Type: Edm.DateTimeOffset
endInbandManagement-Replanning	Type: Edm.DateTimeOffset
endLogicalConnection-Learning	Type: Edm.DateTimeOffset

endNetworkReset	Type: Edm.DateTimeOffset
endRelayFailover-Detection	Type: Edm.DateTimeOffset

Table 22.35: Properties for Sel.Sel5056.LearnAndLock.Status.LearnAndLockStateBeginEndTimestamps

Relationship of Sel.Sel5056.LearnAndLock.Status.LearnAndLockStateBeginEndTimestamps to other types:

—Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.stateBeginEndTimestamps

——[Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession](#)

22.31 Sel.Sel5056.LogicalConnectionValidation.DataObject.-AutoFillSettings

Name	Attributes
autoFillMacWhenAuto-FillingIp	Type: Edm.Boolean
disableSourceAutoFill	Type: Edm.Boolean

Table 22.36: Properties for Sel.Sel5056.LogicalConnectionValidation.DataObject.AutoFillSettings

Relationship of Sel.Sel5056.LogicalConnectionValidation.DataObject.AutoFillSettings to other types:

—Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType.autoFillSettings

——[Sel.Sel5056.LogicalConnectionValidation.DataObject.CommunicationServiceType](#)

22.32 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](#)

-[Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupActionByAlias](#)

-[Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputActionByAlias](#)

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Action to other types:

—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**

22.32.1 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetQueueAction

Name	Attributes
queueId	Type: Edm.Int64

Table 22.37: Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetQueueAction

22.32.2 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupAction

Name	Attributes
groupId	Type: Edm.Int64

Table 22.38: Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupAction

22.32.3 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputAction

Name	Attributes
maxLength	Max: 65535
	Min: 0
	Step: 1
	Type: Edm.Int32
outPort	Type: Edm.String

Table 22.39: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputAction

22.32.4 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PopVlanAction

This type has no properties.

22.32.5 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PushVlanAction

Name	Attributes
etherType	Type: Edm.String

Table 22.40: Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PushVlanAction

22.32.6 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetFieldAction

Name	Attributes
field	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.MatchField

Table 22.41: Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.SetFieldAction

22.32.7 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupActionByAlias

Name	Attributes
configGroupId	Type: Edm.String

Table 22.42: Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupActionByAlias

22.32.8 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputActionByAlias

Name	Attributes
configPortId	Type: Edm.String

Table 22.43: Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.OutputActionByAlias

22.33 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BandStat

Name	Attributes
byteBandCount	Type: Edm.Int64
packetBandCount	Type: Edm.Int64

Table 22.44: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BandStat

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BandStat to other types:

- Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStat.bandStats
- Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats.meterStatList
- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats](#)

22.34 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket

Name	Attributes
actions	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Action]
id	Type: Edm.String
outGroup	Type: Edm.Int64

outPort	Type: Edm.Int64
watchGroup	Type: Edm.String
watchPort	Type: Edm.String

Table 22.45: Properties for *Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket*

The following types inherit from this type:

-[Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupBucketByAlias](#)

-[Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortBucketByAlias](#)

Relationship of *Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket* to other types:

—*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](#)

22.34.1 *Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupBucketByAlias*

Name	Attributes
watchGroupConfigId	Type: Edm.String

Table 22.46: Property for *Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupBucketByAlias*

22.34.2 *Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortBucketByAlias*

Name	Attributes
watchPortConfigId	Type: Edm.String

Table 22.47: Property for *Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortBucketByAlias*

22.35 *Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BucketStats*

Name	Attributes
byteCount	Type: Edm.Int64
packetCount	Type: Edm.Int64

Table 22.48: Properties for *Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BucketStats*

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BucketStats to other types:

- 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**

22.36 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowData

Name	Attributes
cookie	Type: Edm.Int64
nodeId	Type: Edm.String
priority	Type: Edm.Int64
tableId	Type: Edm.Byte

Table 22.49: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowData

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowData to other types:

No other data types reference this type.

22.37 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat

Name	Attributes
byteCount	Type: Edm.Int64
checkOverlap	Type: Edm.Boolean
cookie	Type: Edm.Int64
duration	Type: Edm.Duration
flags	Type: Edm.Int32
hardTimeout	Type: Edm.Int32
idleTimeout	Type: Edm.Int32
instructions	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction]
match	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match
noByteCounts	Type: Edm.Boolean
noPacketCounts	Type: Edm.Boolean
packetCount	Type: Edm.Int64
priority	Type: Edm.Int32

resetCounts	Type: Edm.Boolean
tableId	Type: Edm.Byte

Table 22.50: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat to other types:

- Sel.Sel5056.PathPlanner.RestController.FlowWithStats.statistics
- Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
- Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats.stats
- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStats](#)

22.38 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupData

Name	Attributes
groupId	Type: Edm.Int64
nodeId	Type: Edm.String

Table 22.51: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupData

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupData to other types:

No other data types reference this type.

22.39 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat

Name	Attributes
bucketStatsList	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BucketStats]
byteCount	Type: Edm.Int64
duration	Type: Edm.Duration
groupId	Type: Edm.Int64
packetCount	Type: Edm.Int64
refCount	Type: Edm.Int64

Table 22.52: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat to other types:

- Sel.Sel5056.PathPlanner.RestController.GroupWithStats.statistics
- Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.groupsReferences
- Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats.stats
- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStats](#)

22.40 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction

Name	Attributes
actions	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Action]

Table 22.53: Property 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](#)
- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterInstructionByAlias](#)

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction to other types:

- 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](#)

22.40.1 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ApplyActions

This type has no properties.

22.40.2 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.ClearActions

This type has no properties.

22.40.3 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GoToTable

Name	Attributes
tableId	Max: 255
	Min: 1
	Step: 1
	Type: Edm.Byte

Table 22.54: Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GoToTable

22.40.4 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterInstruction

Name	Attributes
meterId	Type: Edm.Int64

Table 22.55: Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterInstruction

22.40.5 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.WriteActions

This type has no properties.

22.40.6 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterInstructionByAlias

Name	Attributes
configMeterId	Type: Edm.String

Table 22.56: Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterInstructionByAlias

22.41 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match

Name	Attributes
matchFields	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.-MatchField]

Table 22.57: Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Match to other types:

- 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.proposed-CommunicationServiceType
 —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

22.42 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.-MatchFields.MatchField

Name	Attributes
value	Type: Edm.String

Table 22.58: Property 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
 -Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpSpaByAlias
 -Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpTpaByAlias
 -Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthDstByAlias
 -Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthSrcByAlias
 -Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthernetApplicationByAlias

-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.InPortByAlias
 -Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4DstByAlias
 -Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4SrcByAlias

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.MatchField to other types:

- 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.proposed-CommunicationServiceType
- 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.proposed-CommunicationServiceType
 —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

22.42.1 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpOp

This type has no properties.

22.42.2 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpSpa

Name	Attributes
mask	DataClass: IPAddress
	Type: Edm.String
translatedValue	Type: Edm.String

Table 22.59: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpSpa

22.42.3 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpTpa

Name	Attributes
mask	DataClass: IPAddress
	Type: Edm.String
translatedValue	Type: Edm.String

Table 22.60: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.ArpTpa

22.42.4 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthDst

Name	Attributes
mask	DataClass: MacAddress
	Type: Edm.String
translatedValue	Type: Edm.String

Table 22.61: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthDst

22.42.5 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthSrc

Name	Attributes
mask	DataClass: MacAddress
	Type: Edm.String
translatedValue	Type: Edm.String

Table 22.62: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthSrc

22.42.6 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Eth-Type

This type has no properties.

22.42.7 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.InPort

Name	Attributes
translatedValue	Type: Edm.Int64

Table 22.63: Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.InPort

22.42.8 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.IpProto

This type has no properties.

22.42.9 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Dst

Name	Attributes
mask	DataClass: IpAddress
	Type: Edm.String
translatedValue	Type: Edm.String

Table 22.64: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Dst

22.42.10 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4-Src

Name	Attributes
mask	DataClass: IpAddress
	Type: Edm.String
translatedValue	Type: Edm.String

Table 22.65: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4Src

22.42.11 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Tcp-Dst

This type has no properties.

22.42.12 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.TcpSrc

This type has no properties.

22.42.13 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Udp-Dst

This type has no properties.

22.42.14 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Udp-Src

This type has no properties.

22.42.15 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Vlan-Pcp

This type has no properties.

22.42.16 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Vlan-Vid

Name	Attributes
mask	Type: Edm.String

Table 22.66: Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.VlanVid

22.42.17 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Arp-SpaByAlias

This type has no properties.

22.42.18 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Arp-TpaByAlias

This type has no properties.

22.42.19 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Eth-DstByAlias

This type has no properties.

22.42.20 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Eth-SrcByAlias

This type has no properties.

22.42.21 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.-EthernetApplicationByAlias

Name	Attributes
applicationEthType	Type: Edm.Int32

Table 22.67: Property for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.EthernetApplicationByAlias

22.42.22 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.In-PortByAlias

This type has no properties.

22.42.23 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4-DstByAlias

This type has no properties.

22.42.24 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MatchFields.Ipv4-SrcByAlias

This type has no properties.

22.43 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand

Name	Attributes
burstSize	Type: Edm.Int64
rate	Type: Edm.Int64

Table 22.68: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand

The following types inherit from this type:

-Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandDrop

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBand to other types:

—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

22.43.1 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandDrop

This type has no properties.

22.44 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandHeader

Name	Attributes
burstSize	Type: Edm.Int64
length	Type: Edm.Int32
rate	Type: Edm.Int64
type	Type: Sel.Sel5056.OpenFlowPlugin.Enums.OF13MeterBandType

Table 22.69: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandHeader

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandHeader to other types:

—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDesc.bands

——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions.meterDescriptionsList

——**Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions**

22.45 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDesc

Name	Attributes
bands	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterBandHeader]
flags	Type: Edm.Int32
length	Type: Edm.Int32
meterId	Type: Edm.Int64

Table 22.70: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDesc

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDesc to other types:

—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions.meterDescriptionsList

——[Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterDescriptions](#)

22.46 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStat

Name	Attributes
bandStats	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.BandStat]
byteInCount	Type: Edm.Int64
duration	Type: Edm.Duration
flowCount	Type: Edm.Int64
meterId	Type: Edm.Int64
packetInCount	Type: Edm.Int64

Table 22.71: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStat

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStat to other types:

——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats.meterStatList

——[Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterStats](#)

22.47 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port

Name	Attributes
advertised	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.FeatureBase]
config	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-ConfigFlagObjects.ConfigBase]
curr	Type: Edm.Int64
currentSpeed	Type: Edm.Int64
hardwareAddress	Type: Edm.String
maxSpeed	Type: Edm.Int64
name	Type: Edm.String
peer	Type: Edm.Int64
portId	Type: Edm.Int64
state	Type: Sel.Sel5056.TopologyManager.Enums.OfpPortStatus
supported	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.FeatureBase]

Table 22.72: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Port to other types:

——Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc.ports

——[Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortDesc](#)

22.48 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStat

Name	Attributes
collisions	Type: Edm.Int64
duration	Type: Edm.Duration
portNo	Type: Edm.Int64
rxBytes	Type: Edm.Int64
rxCrcErr	Type: Edm.Int64
rxDropped	Type: Edm.Int64
rxErrors	Type: Edm.Int64
rxFrameErr	Type: Edm.Int64
rxOverError	Type: Edm.Int64
rxPackets	Type: Edm.Int64
txBytes	Type: Edm.Int64
txDropped	Type: Edm.Int64
txErrors	Type: Edm.Int64
txPackets	Type: Edm.Int64

Table 22.73: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStat

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStat to other types:

—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats.portStatList

—[Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.PortStats](#)

22.49 Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStat

Name	Attributes
activeCount	Type: Edm.Int64
lookupCount	Type: Edm.Int64
matchCount	Type: Edm.Int64
tableId	Type: Edm.Byte

Table 22.74: Properties for Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStat

Relationship of Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStat to other types:

—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats.tableStatList

—[Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.TableStats](#)

22.50 Sel.Sel5056.OpenFlowPlugin.Synchronization.- Dependency.OpenFlowDependency

This type has no properties.

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

Relationship of Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.OpenFlowDependency to other types:

- 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

22.50.1 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.FlowDeleteDependency

Name	Attributes
cookie	Type: Edm.Int64

Table 22.75: Property for Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.FlowDelete-Dependency

22.50.2 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.Group-DeleteDependency

Name	Attributes
groupId	Type: Edm.Int64

Table 22.76: Property for Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupDelete-Dependency

22.50.3 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.Group-Dependency

Name	Attributes
groupId	Type: Edm.Int64

Table 22.77: Property for Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupDependency

22.50.4 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.Meter-Dependency

Name	Attributes
meterId	Type: Edm.Int64

Table 22.78: Property for Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.MeterDependency

22.50.5 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.Flow-ModifyDependency

Name	Attributes
cookie	Type: Edm.Int64

Table 22.79: Property for Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.FlowModify-Dependency

22.50.6 Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.Group-ModifyDependency

Name	Attributes
groupId	Type: Edm.Int64

Table 22.80: Property for Sel.Sel5056.OpenFlowPlugin.Synchronization.Dependency.GroupModify-Dependency

22.51 Sel.Sel5056.PathPlanner.DataObject.PathPlanHop-Collection

Name	Attributes
destinationNodeId	Type: Edm.String
pathPlanHops	Type: [Sel.PathPlanHop]
sourceNodeId	Type: Edm.String

Table 22.81: Properties for Sel.Sel5056.PathPlanner.DataObject.PathPlanHopCollection

Relationship of Sel.Sel5056.PathPlanner.DataObject.PathPlanHopCollection to other types:
No other data types reference this type.

22.52 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.-DestinationDescriptions.Types.AmDestination

Name	Attributes
getFirstNodeId	Type: Edm.String
uniqueId	Type: Edm.String

Table 22.82: Properties for Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.-Types.AmDestination

The following types inherit from this type:

- Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.RelayFailover-Destination
- Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.SimpleDestination
- Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.TraditionalTiePoint-Destination
- Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.Programmable-Destination
- Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.InbandDestination

Relationship of Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.AmDestination to other types:

- Sel.Sel5056.PathPlanner.RestController.PathPlanInfo.pathDestination
- Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidDestinationReservation.amDestination
- Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidDestinationReservation
- Sel.ReserveVlanVidRange (binding parameter)

22.52.1 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.- DestinationDescriptions.Types.RelayFailoverDestination

Name	Attributes
egressPortId	Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort
nodeId	Type: Edm.String

Table 22.83: Properties for Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.-Types.RelayFailoverDestination

22.52.2 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.- DestinationDescriptions.Types.SimpleDestination

Name	Attributes
nodeId	Type: Edm.String

Table 22.84: Property for Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.-Types.SimpleDestination

22.52.3 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.- DestinationDescriptions.Types.TraditionalTiePointDestination

Name	Attributes
nodeIds	Type: [Edm.String]
reachableOpenFlowPortId	Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort
sourcePortId	Type: Edm.String

Table 22.85: Properties for Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.-Types.TraditionalTiePointDestination

22.52.4 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.- DestinationDescriptions.Types.ProgrammableDestination

Name	Attributes
nodeId	Type: Edm.String

Table 22.86: Property for Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.-Types.ProgrammableDestination

22.52.5 Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.- DestinationDescriptions.Types.InbandDestination

Name	Attributes
inBandPort	Type: Edm.String
nodeId	Type: Edm.String

Table 22.87: Properties for Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.DestinationDescriptions.Types.InbandDestination

22.53 Sel.Sel5056.PathPlanner.RestController.Directionallink

Name	Attributes
egressPortId	Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort
ingressPortId	Type: Sel.Sel5056.TopologyManager.Ports.ConfigPort
linkId	Type: Sel.Sel5056.TopologyManager.Links.ConfigLink

Table 22.88: Properties for Sel.Sel5056.PathPlanner.RestController.Directionallink

Relationship of Sel.Sel5056.PathPlanner.RestController.Directionallink to other types:

- Sel.Sel5056.PathPlanner.RestController.PathPlanInfo.primaryLinks
- Sel.Sel5056.PathPlanner.RestController.PathPlanInfo.failOverLinks
- Sel.Sel5056.PathPlanner.RestController.PathPlanInfo.notFailedAroundLinks

22.54 Sel.Sel5056.PathPlanner.RestController.FlowReferences

Name	Attributes
cookie	Type: Edm.Int64
displayName	Type: Edm.String
id	Type: Edm.String
instructions	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Instruction]
node	Type: Edm.String
tags	Type: [Sel.Sel5056.Tags.AmTag]

Table 22.89: Properties for Sel.Sel5056.PathPlanner.RestController.FlowReferences

Relationship of Sel.Sel5056.PathPlanner.RestController.FlowReferences to other types:

- Sel.Sel5056.PathPlanner.RestController.FlowWithStats.flow
- Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences
- Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.flowReferences

22.55 Sel.Sel5056.PathPlanner.RestController.FlowWithStats

Name	Attributes
dataPathId	Type: Edm.Int64
flow	Type: Sel.Sel5056.PathPlanner.RestController.FlowReferences
outputs	Type: Edm.String
statistics	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.FlowStat
switchDisplayName	Type: Edm.String

Table 22.90: Properties for Sel.Sel5056.PathPlanner.RestController.FlowWithStats

Relationship of Sel.Sel5056.PathPlanner.RestController.FlowWithStats to other types:

—Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.flowReferences

22.56 Sel.Sel5056.PathPlanner.RestController.GroupReferences

Name	Attributes
buckets	Type: [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Bucket]
displayName	Type: Edm.String
groupId	Type: Edm.Int64
groupType	Type: Sel.Sel5056.OpenFlowPlugin.Enums.OfpGroupType
id	Type: Edm.String
node	Type: Edm.String
tags	Type: [Sel.Sel5056.Tags.AmTag]

Table 22.91: Properties for Sel.Sel5056.PathPlanner.RestController.GroupReferences

Relationship of Sel.Sel5056.PathPlanner.RestController.GroupReferences to other types:

—Sel.Sel5056.PathPlanner.RestController.GroupWithStats.group

—Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.groupsReferences

—Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo.groupsReferences

22.57 Sel.Sel5056.PathPlanner.RestController.GroupWithStats

Name	Attributes
dataPathId	Type: Edm.Int64
group	Type: Sel.Sel5056.PathPlanner.RestController.GroupReferences
outputs	Type: Edm.String
statistics	Type: Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.GroupStat
switchDisplayName	Type: Edm.String

Table 22.92: Properties for Sel.Sel5056.PathPlanner.RestController.GroupWithStats

Relationship of Sel.Sel5056.PathPlanner.RestController.GroupWithStats to other types:
 —Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats.groupsReferences

22.58 Sel.Sel5056.PathPlanner.RestController.Logical-ConnectionStats

Name	Attributes
flowReferences	Type: [Sel.Sel5056.PathPlanner.RestController.FlowWithStats]
groupsReferences	Type: [Sel.Sel5056.PathPlanner.RestController.GroupWithStats]

Table 22.93: Properties for Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats

Relationship of Sel.Sel5056.PathPlanner.RestController.LogicalConnectionStats to other types:
 No other data types reference this type.

22.59 Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo

Name	Attributes
flowReferences	Type: [Sel.Sel5056.PathPlanner.RestController.FlowReferences]
groupsReferences	Type: [Sel.Sel5056.PathPlanner.RestController.GroupReferences]

Table 22.94: Properties for Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo

Relationship of Sel.Sel5056.PathPlanner.RestController.OpenFlowInfo to other types:
 No other data types reference this type.

22.60 Sel.Sel5056.PathPlanner.RestController.PathPlanInfo

Name	Attributes
failOverLinks	Type: [Sel.Sel5056.PathPlanner.RestController.DirectionLink]
failOverNodes	Type: [Edm.String]
notFailedAroundLinks	Type: [Sel.Sel5056.PathPlanner.RestController.DirectionLink]
pathDestination	Type: Sel.Sel5056.PathPlanner.PathFinding.PrefixTree.-DestinationDescriptions.Types.AmDestination
primaryLinks	Type: [Sel.Sel5056.PathPlanner.RestController.DirectionLink]
primaryNodes	Type: [Edm.String]

Table 22.95: Properties for Sel.Sel5056.PathPlanner.RestController.PathPlanInfo

Relationship of *Sel.Sel5056.PathPlanner.RestController.PathPlanInfo* to other types:
 No other data types reference this type.

22.61 *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*

Relationship of *Sel.Sel5056.Tags.AmTag* to other types:

- Sel.Sel5056.TopologyManager.Nodes.ConfigNode.tags*
- Sel.Sel5056.TopologyManager.Nodes.ConfigNode*
- 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.LogicalConnection.tags*
- Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection*
- 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.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

22.61.1 Sel.Sel5056.Tags.LogicalConnectionTag

Name	Attributes
id	Type: Edm.String
logicalConnectionId	Type: Edm.String

Table 22.96: Properties for Sel.Sel5056.Tags.LogicalConnectionTag

22.61.2 Sel.Sel5056.Tags.InBandPortIdTag

Name	Attributes
inbandPortId	Type: Edm.Int64

Table 22.97: Property for Sel.Sel5056.Tags.InBandPortIdTag

22.61.3 Sel.Sel5056.Tags.GenericArpFlowTag

Name	Attributes
configNodeId	Type: Edm.String
id	Type: Edm.String

Table 22.98: Properties for Sel.Sel5056.Tags.GenericArpFlowTag

22.61.4 Sel.Sel5056.Tags.SystemTag

This type has no properties.

22.61.5 Sel.Sel5056.Tags.OpenFlowPortIdTag

Name	Attributes
portId	Type: Edm.Int64

Table 22.99: Property for Sel.Sel5056.Tags.OpenFlowPortIdTag

22.61.6 Sel.Sel5056.Tags.KeyValueTag

Name	Attributes
id	Type: Edm.String
key	Type: Edm.String
value	Type: Edm.String

Table 22.100: Properties for Sel.Sel5056.Tags.KeyValueTag

22.61.7 Sel.Sel5056.Tags.InBandTag

Name	Attributes
configNodeId	Type: Edm.String
id	Type: Edm.String

Table 22.101: Properties for Sel.Sel5056.Tags.InBandTag

22.61.8 Sel.Sel5056.Tags.ControllerTag

Name	Attributes
id	Type: Edm.String

Table 22.102: Property for Sel.Sel5056.Tags.ControllerTag

22.61.9 Sel.Sel5056.Tags.IpAddressTag

Name	Attributes
id	Type: Edm.String
ipAddress	DataClass: IPAddress
	Type: Edm.String

Table 22.103: Properties for Sel.Sel5056.Tags.IpAddressTag

22.61.10 Sel.Sel5056.Tags.MacAddressTag

Name	Attributes
id	Type: Edm.String
macAddress	DataClass: MacAddress
	Type: Edm.String

Table 22.104: Properties for Sel.Sel5056.Tags.MacAddressTag

22.61.11 Sel.Sel5056.Tags.ReadoptTag

This type has no properties.

22.61.12 Sel.Sel5056.Tags.ControllerRouteTag

Name	Attributes
id	Type: Edm.String

Table 22.105: Property for Sel.Sel5056.Tags.ControllerRouteTag

22.61.13 Sel.Sel5056.Tags.NoArpTag

This type has no properties.

22.61.14 Sel.Sel5056.LogicalConnectionValidation.Tags.MatchableCst

This type has no properties.

22.61.15 Sel.Sel5056.LearnAndLock.AutoAdoption.Tags.AutoAdoptionTag

Name	Attributes
learnAndLockSessionId	Type: Edm.String

Table 22.106: Property for Sel.Sel5056.LearnAndLock.AutoAdoption.Tags.AutoAdoptionTag

22.62 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.OpenFlowTableFeaturesAttr
- 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

Relationship of Sel.Sel5056.TopologyManager.Attributes.AmAttribute to other types:

—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

22.62.1 Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr

Name	Attributes
advertised	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.FeatureBase]
config	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-ConfigFlagObjects.ConfigBase]
currentFeatures	Type: Edm.Int64
currentSpeed	Type: Edm.Int64
dataPathId	Type: Edm.Int64
hardwareAddress	DataClass: MacAddress
	Type: Edm.String

id	Type: Edm.String
maxSpeed	Type: Edm.Int64
name	Type: Edm.String
ofState	Type: Sel.Sel5056.TopologyManager.Enums.OfpPortStatus
peer	Type: Edm.Int64
portId	Type: Edm.Int64
prettyDataPathId	Type: Edm.String
supported	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.FeatureBase]

Table 22.107: Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Port.OpenFlowPortAttr

22.62.2 Sel.Sel5056.TopologyManager.Attributes.Operational.Port.Sel2740SPortAttr

Name	Attributes
portType	Type: Sel.Sel5056.TopologyManager.Attributes.Operational.Node.Sel2740NodePortType

Table 22.108: Property for Sel.Sel5056.TopologyManager.Attributes.Operational.Port.Sel2740SPortAttr

22.62.3 Sel.Sel5056.TopologyManager.Attributes.Operational.Port.-EthernetApplicationAttr

Name	Attributes
ethType	Type: Edm.Int32
id	Type: Edm.String
macAddress	Type: Edm.String
providerInformation	Type: [Sel.Sel5056.TopologyManager.Attributes.Operational.Port.-EthernetApplicationProviderInformation]

Table 22.109: Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationAttr

22.62.4 Sel.Sel5056.TopologyManager.Attributes.Operational.Port.SelRelayFailoverModeAttr

This type has no properties.

22.62.5 Sel.Sel5056.TopologyManager.Attributes.Operational.Port.-EthernetApplicationNodeAttr

Name	Attributes
id	Type: Edm.String
macAddress	Type: Edm.String

Table 22.110: Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Port.Ethernet-ApplicationNodeAttr

22.62.6 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.Open-FlowAttr

Name	Attributes
dataPathId	Type: Edm.Int64
id	Type: Edm.String
prettyDataPathId	Type: Edm.String

Table 22.111: Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Node.OpenFlowAttr

22.62.7 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.Open-FlowTableFeaturesAttr

Name	Attributes
dataPathId	Type: Edm.Int64
id	Type: Edm.String
tables	Type: [Sel.Sel5056.TopologyManager.Attributes.Operational.Node.Table-Feature]

Table 22.112: Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Node.OpenFlowTableFeaturesAttr

22.62.8 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.IpAttr

Name	Attributes
ipAddress	DataClass: IPAddress
	Type: Edm.String

Table 22.113: Property for Sel.Sel5056.TopologyManager.Attributes.Operational.Node.IpAttr

22.62.9 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.Sel-SapphireAttr

Name	Attributes
autoDiscoveryVersion	Type: Edm.String

communicationBand	Type: Sel.Sel5056.TopologyManager.Enums.CommunicationBand
dataPathId	Type: Edm.Int64
deviceFramework	Type: Sel.Sel5056.TopologyManager.Attributes.Operational.Node.DeviceFramework
discoveryMac	Type: Edm.String
discoveryPort	Type: Edm.Int64
discoveryPortAttribute	Type: Sel.Sel5056.TopologyManager.Attributes.AmAttribute
discoverySource	Type: Sel.Sel5056.TopologyManager.Enums.PacketSource
ethFrontIp	Type: Edm.String
ethFrontMac	Type: Edm.String
ethRearIp	Type: Edm.String
ethRearMac	Type: Edm.String
failedEventQuery-Attempts	Type: Edm.Int32
id	Type: Edm.String
isCommissioned	Type: Edm.Boolean
lastAutoDiscovery-Received	Type: Edm.DateTimeOffset
lastQueriedEventId	Type: Edm.Int64
modelName	Type: Edm.String
name	Type: Edm.String
nodeType	Type: Sel.Sel5056.TopologyManager.Attributes.Operational.Node.Sel27-4xNodeType
omniPort	Type: Edm.Int32
prettyDataPathId	Type: Edm.String
serialNumber	Type: Edm.String
x509Fingerprint	Type: Edm.String

Table 22.114: Properties for [Sel.Sel5056.TopologyManager.Attributes.Operational.Node.SelSapphireAttr](#)

22.62.10 [Sel.Sel5056.TopologyManager.Attributes.Operational.Node.-FirmwareUpdateAttr](#)

Name	Attributes
firmwareFileUpload-PercentComplete	Type: Edm.Int32
firmwareUploadDevice-Status	Type: Edm.String
lastUpdate	Type: Edm.DateTimeOffset

Table 22.115: Properties for [Sel.Sel5056.TopologyManager.Attributes.Operational.Node.FirmwareUpdateAttr](#)

22.62.11 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.UserHintedIpAddressAttr

Name	Attributes
id	Type: Edm.String
ipAddress	DataClass: IPAddress
	Type: Edm.String
openFlowConfigPortId	Type: Edm.String

Table 22.116: Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Node.UserHintedIpAddressAttr

22.62.12 Sel.Sel5056.TopologyManager.Attributes.Operational.Node.NoArpAttr

This type has no properties.

22.62.13 Sel.Sel5056.TopologyManager.Attributes.Operational.Link.OfdpDiscovered

This type has no properties.

22.62.14 Sel.Sel5056.TopologyManager.Attributes.Operational.Link.OpenFlowVerified

This type has no properties.

22.62.15 Sel.Sel5056.TopologyManager.Attributes.Operational.Link.ArpVerified

This type has no properties.

22.62.16 Sel.Sel5056.TopologyManager.Attributes.Operational.Link.UserHintedLinkAttr

This type has no properties.

22.62.17 Sel.Sel5056.TopologyManager.Attributes.Operational.ControllerAttr

This type has no properties.

22.62.18 Sel.Sel5056.TopologyManager.Attributes.Operational.Ethernet-Attr

Name	Attributes
id	Type: Edm.String
macAddress	Type: Edm.String

Table 22.117: Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.EthernetAttr

22.62.19 Sel.Sel5056.TopologyManager.Attributes.Operational.DoNot-ExpireAttr

This type has no properties.

22.62.20 Sel.Sel5056.TopologyManager.Attributes.Operational.UserHinted-MappedSwitchPortAttr

Name	Attributes
configPortId	Type: Edm.String
id	Type: Edm.String

Table 22.118: Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedMappedSwitchPortAttr

22.62.21 Sel.Sel5056.TopologyManager.Attributes.Operational.Networking-AbstractionAttr

Name	Attributes
id	Type: Edm.String
mappedEthernet-ApplicationPortAttr	Type: Sel.Sel5056.TopologyManager.Attributes.Operational.Port.-EthernetApplicationAttr
mappedEthernetPortAttr	Type: Sel.Sel5056.TopologyManager.Attributes.Operational.EthernetAttr
mappedOpenFlowPort-Attr	Type: Sel.Sel5056.TopologyManager.Attributes.Operational.Port.Open-FlowPortAttr
mappedSilentHostAttr	Type: Sel.Sel5056.TopologyManager.Attributes.Operational.UserHinted-MappedSwitchPortAttr

Table 22.119: Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Networking-AbstractionAttr

22.62.22 Sel.Sel5056.TopologyManager.Attributes.Operational.UserHinted-NodeAttr

Name	Attributes
configNodeId	Type: Edm.String
id	Type: Edm.String

Table 22.120: Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.UserHintedNodeAttr

22.62.23 Sel.Sel5056.Southbound.DeviceManagement.CommissioningStateAttr

Name	Attributes
state	Type: Sel.BlueFrame.Core.ErrorState
status	Type: Edm.String

Table 22.121: Properties for Sel.Sel5056.Southbound.DeviceManagement.CommissioningStateAttr

22.63 Sel.Sel5056.TopologyManager.Attributes.Operational.-Node.TableFeature

Name	Attributes
maxEntries	Type: Edm.Int64
name	Type: Edm.String
tableId	Type: Edm.Byte

Table 22.122: Properties for Sel.Sel5056.TopologyManager.Attributes.Operational.Node.TableFeature

Relationship of Sel.Sel5056.TopologyManager.Attributes.Operational.Node.TableFeature to other types:

- Sel.Sel5056.TopologyManager.Attributes.Operational.Node.OpenFlowTableFeaturesAttr.tables
- 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](#)

22.64 Sel.Sel5056.TopologyManager.Attributes.Operational.-Port.EthernetApplicationProviderInformation

Name	Attributes
destinationMac	Type: Edm.String
vlan	Type: Edm.Int32

Table 22.123: Properties for `Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationProviderInformation`

Relationship of `Sel.Sel5056.TopologyManager.Attributes.Operational.Port.EthernetApplicationProviderInformation` to other types:

—`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](#)

22.65 Sel.Sel5056.TopologyManager.Configurations.Abstraction.-AbstractSettingBase

This type has no properties.

The following types inherit from this type:

-[Sel.Sel5056.TopologyManager.Configurations.Abstraction.TraditionalSwitch](#)

Relationship of `Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractSettingBase` to other types:

—`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`

22.65.1 Sel.Sel5056.TopologyManager.Configurations.Abstraction.-TraditionalSwitch

This type has no properties.

22.66 Sel.Sel5056.TopologyManager.Configurations.AmConfig

This type has no properties.

The following types inherit from this type:

- Sel.Sel5056.TopologyManager.OpenFlow.PortMod.ConfigFlagObjects.ConfigBase
- Sel.Sel5056.TopologyManager.OpenFlow.PortMod.FeatureFlagObjects.FeatureBase
- 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
- Sel.Sel5056.Snmp.SnmpConfig
- 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
- 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

Relationship of Sel.Sel5056.TopologyManager.Configurations.AmConfig to other types:

- 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](#)

**22.66.1 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
ConfigFlagObjects.ConfigBase**

This type has no properties.

**22.66.2 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.FeatureBase**

This type has no properties.

22.66.3 Sel.Sel5056.TopologyManager.Configurations.ControllerConfig

This type has no properties.

**22.66.4 Sel.Sel5056.TopologyManager.Configurations.OpenFlow.System-
Priority**

Name	Attributes
priority	Type: Sel.Sel5056.TopologyManager.Configurations.OpenFlow.- SystemPriorities.AmSystemPriority

Table 22.124: Property for Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriority

**22.66.5 Sel.Sel5056.TopologyManager.Configurations.OpenFlow.OpenFlow-
Config**

This type has no properties.

**22.66.6 Sel.Sel5056.TopologyManager.Configurations.FailOver.SelRelay-
FailoverModeConfig**

This type has no properties.

**22.66.7 Sel.Sel5056.TopologyManager.Configurations.Abstraction.-
AbstractionConfig**

Name	Attributes
abstractionSetting	Type: Sel.Sel5056.TopologyManager.Configurations.Abstraction.- AbstractSettingBase

Table 22.125: Property for Sel.Sel5056.TopologyManager.Configurations.Abstraction.AbstractionConfig

22.66.8 Sel.Sel5056.TopologyManager.Configurations.Sapphire.AlternateIpConfig

Name	Attributes
alternateIp	Type: Edm.String
alternateSubnetMask	DataClass: IPAddress
	Type: Edm.String

Table 22.126: Properties for Sel.Sel5056.TopologyManager.Configurations.Sapphire.AlternateIpConfig

22.66.9 Sel.Sel5056.Snmp.SnmpConfig

Name	Attributes
engineId	CharacterSet: Printable7BitAscii
	MaxLength: 64
	MinLength: 10
	Type: Edm.String
snmpV2cCommunity-String	CharacterSet: Printable7BitAscii
	MaxLength: 128
	MinLength: 1
	Type: Edm.String
snmpV2cEnabled	Type: Edm.Boolean
snmpV3Enabled	Type: Edm.Boolean
systemDescription	CharacterSet: Printable7BitAscii
	MaxLength: 256
	Type: Edm.String
systemLocation	CharacterSet: Printable7BitAscii
	MaxLength: 256
	Type: Edm.String
systemName	CharacterSet: Printable7BitAscii
	MaxLength: 256
	Type: Edm.String

Table 22.127: Properties for Sel.Sel5056.Snmp.SnmpConfig

22.66.10 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-ConfigFlagObjects.NoForward

This type has no properties.

22.66.11 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-ConfigFlagObjects.NoPacketIn

This type has no properties.

**22.66.12 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
ConfigFlagObjects.NoReceive**

This type has no properties.

**22.66.13 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
ConfigFlagObjects.PortDown**

This type has no properties.

**22.66.14 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.AutoNegotiation**

This type has no properties.

**22.66.15 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.Copper**

This type has no properties.

**22.66.16 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.Fiber**

This type has no properties.

**22.66.17 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.FortyGigaBitFullDuplex**

This type has no properties.

**22.66.18 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.OneGigaBitFullDuplex**

This type has no properties.

**22.66.19 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.OneGigaBitHalfDuplex**

This type has no properties.

**22.66.20 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.OneHundredGigaBitFullDuplex**

This type has no properties.

**22.66.21 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.OneHundredMegaBitFullDuplex**

This type has no properties.

**22.66.22 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.OneHundredMegaBitHalfDuplex**

This type has no properties.

**22.66.23 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.OneTeraBitFullDuplex**

This type has no properties.

**22.66.24 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.OtherBitRate**

This type has no properties.

**22.66.25 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.Pause**

This type has no properties.

**22.66.26 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.PauseAsymmetric**

This type has no properties.

**22.66.27 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.TenGigaBitFullDuplex**

This type has no properties.

**22.66.28 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.TenMegaBitFullDuplex**

This type has no properties.

**22.66.29 Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-
FeatureFlagObjects.TenMegaBitHalfDuplex**

This type has no properties.

22.67 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

Relationship of Sel.Sel5056.TopologyManager.Configurations.OpenFlow.SystemPriorities.AmSystemPriority to other types:

- 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.MeterDeleteSynchronizationRequest
- Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterModifySynchronizationRequest.-priorityConfiguration
- Sel.Sel5056.OpenFlowPlugin.Synchronization.Types.Requests.MeterModifySynchronizationRequest

22.67.1 Sel.Sel5056.TopologyManager.Configurations.OpenFlow.- SystemPriorities.SystemCritical

This type has no properties.

22.67.2 Sel.Sel5056.TopologyManager.Configurations.OpenFlow.- SystemPriorities.SystemRequired

This type has no properties.

22.68 Sel.Sel5056.TopologyManager.Nodes.PortLinkEvent- Configuration

Name	Attributes
linkDownAutoClear	Type: Edm.Boolean
linkDownSeverity	Type: Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel
linkUpAutoClear	Type: Edm.Boolean
linkUpSeverity	Type: Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel
portId	Type: Edm.String
	Unique: True
	WriteOnce: true

Table 22.128: Properties for Sel.Sel5056.TopologyManager.Nodes.PortLinkEventConfiguration

Relationship of Sel.Sel5056.TopologyManager.Nodes.PortLinkEventConfiguration to other types:

—Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode.portLinkEventConfigurations

—[Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode](#)

22.69 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](#)

Relationship of Sel.Sel5056.TopologyManager.OpenFlow.Meter.RateTypeFlagObjects.RateTypeBase to other types:

—Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.flags

—[Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter](#)

- Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.MeterMetadata.flags
- Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions.meterMetadata
- Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Meter.flags
- Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures.capabilities
- [Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.MeterFeatures](#)

22.69.1 Sel.Sel5056.TopologyManager.OpenFlow.Meter.- RateTypeFlagObjects.Burst

This type has no properties.

22.69.2 Sel.Sel5056.TopologyManager.OpenFlow.Meter.- RateTypeFlagObjects.KilobitesPerSecond

This type has no properties.

22.69.3 Sel.Sel5056.TopologyManager.OpenFlow.Meter.- RateTypeFlagObjects.PacketsPerSecond

This type has no properties.

22.69.4 Sel.Sel5056.TopologyManager.OpenFlow.Meter.- RateTypeFlagObjects.Stats

This type has no properties.

Chapter 23

OData Actions

23.1 Sel.AcceptAllProposedLearnedLogicalConnections

URL: /api/default/config/LearnAndLockSessions/AcceptAllProposedLearnedLogicalConnections

The binding parameter type is Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.
This type has no properties.

23.2 Sel.AcceptLearnedLogicalConnections

URL: /api/default/config/LearnAndLockSessions/AcceptLearnedLogicalConnections

The binding parameter type is Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.

Name	Attributes
learnedLogical-ConnectionIds	Type: [Edm.String]

Table 23.1: Property for Sel.AcceptLearnedLogicalConnections

23.3 Sel.AcknowledgeLearnAndLockSession

URL: /api/default/config/LearnAndLockSessions/AcknowledgeLearnAndLockSession

The binding parameter type is Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.

Name	Attributes
sessionId	Type: Edm.String

Table 23.2: Property for Sel.AcknowledgeLearnAndLockSession

23.4 Sel.AddAbstractNode

URL: [/api/default/operational/ports\('id'\)/AddAbstractNode](/api/default/operational/ports('id')/AddAbstractNode)

The binding parameter type is Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.

Name	Attributes
configSetting	Type: Sel.Sel5056.TopologyManager.Configurations.Abstraction.-AbstractSettingBase

Table 23.3: Property for Sel.AddAbstractNode

23.5 Sel.AddOperationalLinkFromHostToSwitch

URL: [/api/default/operational/ports\('id'\)/AddOperationalLinkFromHostToSwitch](/api/default/operational/ports('id')/AddOperationalLinkFromHostToSwitch)

The binding parameter type is Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.

Name	Attributes
switchPortId	Type: Edm.String

Table 23.4: Property for Sel.AddOperationalLinkFromHostToSwitch

23.6 Sel.AddSilentHost

URL: [/api/default/operational/ports\('id'\)/AddSilentHost](/api/default/operational/ports('id')/AddSilentHost)

The binding parameter type is Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.

Name	Attributes
configNodeId	Type: Edm.String

Table 23.5: Property for Sel.AddSilentHost

23.7 Sel.Adopt

URLs:

- [/api/default/operational/links\('id'\)/Adopt](/api/default/operational/links('id')/Adopt)
- [/api/default/operational/nodes\('id'\)/Adopt](/api/default/operational/nodes('id')/Adopt)
- [/api/default/operational/ports\('id'\)/Adopt](/api/default/operational/ports('id')/Adopt)

There are three instances of this action:

-[Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink](#)

-Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
 -Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

23.7.1 Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

The binding parameter type is Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.
 This type has no properties.

23.7.2 Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

The binding parameter type is Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.
 This type has no properties.

23.7.3 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

The binding parameter type is Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.
 This type has no properties.

23.8 Sel.AdoptWithConfig

URLs:

- /api/default/operational/links('id')/AdoptWithConfig
- /api/default/operational/nodes('id')/AdoptWithConfig
- /api/default/operational/ports('id')/AdoptWithConfig

There are three instances of this action:

-Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink
 -Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode
 -Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

23.8.1 Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

The binding parameter type is Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink.

Name	Attributes
configKey	Type: Edm.String

Table 23.6: Property for Sel.AdoptWithConfig

23.8.2 Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

The binding parameter type is Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.

Name	Attributes
configKey	Type: Edm.String

Table 23.7: Property for Sel.AdoptWithConfig

23.8.3 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

The binding parameter type is Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.

Name	Attributes
configKey	Type: Edm.String

Table 23.8: Property for Sel.AdoptWithConfig

23.9 Sel.AsPem

URL: [/api/default/certificate/certificateInfo\('id'\)/AsPem](#)

The binding parameter type is Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation. This type has no properties.

23.10 Sel.AuditDeviceConfiguration

URL: [/api/default/operational/AuditDeviceConfiguration](#)

Name	Attributes
configNodeId	Type: Edm.String

Table 23.9: Property for Sel.AuditDeviceConfiguration

23.11 Sel.Backup

URL: [/api/default/settings/Backup](#)

Name	Attributes
password	Type: Edm.String

Table 23.10: Property for Sel.Backup

23.12 Sel.BeginLearnAndLock

URL: `/api/default/config/LearnAndLockSessions/BeginLearnAndLock`

The binding parameter type is `Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession`.

Name	Attributes
autoAdoptConfig	Type: <code>Sel.Sel5056.LearnAndLock.Configuration.AutoAdoptConfig</code>
logicalConnection-LearningConfig	Type: <code>Sel.Sel5056.LearnAndLock.LogicalConnectionLearning-Configuration.LogicalConnectionLearningConfig</code>

Table 23.11: Properties for `Sel.BeginLearnAndLock`

23.13 Sel.ClearLearnedLogicalConnections

URL: `/api/default/config/LearnAndLockSessions/ClearLearnedLogicalConnections`

The binding parameter type is `Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession`.

Name	Attributes
learnedLogical-ConnectionIds	Type: [<code>Edm.String</code>]

Table 23.12: Property for `Sel.ClearLearnedLogicalConnections`

23.14 Sel.ClearTransaction

URLs:

- `/api/default/certificate/transactions/ClearTransaction`
- `/api/default/config/transactions/ClearTransaction`
- `/api/default/operational/transactions/ClearTransaction`
- `/api/default/security/transactions/ClearTransaction`
- `/api/default/settings/transactions/ClearTransaction`

The binding parameter type is `Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction`. This type has no properties.

23.15 Sel.Commission

URL: `/api/default/security/Commission`

Name	Attributes
commissioningList	Type: [Sel.BlueFrame.CommissioningManager.CommissioningConcern]

Table 23.13: Property for Sel.Commission

23.16 Sel.Commit

URLs:

- /api/default/certificate/transactions('id')/Commit
- /api/default/config/transactions('id')/Commit
- /api/default/operational/transactions('id')/Commit
- /api/default/security/transactions('id')/Commit
- /api/default/settings/transactions('id')/Commit

The binding parameter type is Sel.BlueFrame.RestBroker.OdataPlugin.Models.RestTransaction. This type has no properties.

23.17 Sel.DeclineAllProposedLearnedLogicalConnections

URL: /api/default/config/LearnAndLockSessions/DeclineAllProposedLearnedLogicalConnections

The binding parameter type is Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession. This type has no properties.

23.18 Sel.DeclineLearnedLogicalConnections

URL: /api/default/config/LearnAndLockSessions/DeclineLearnedLogicalConnections

The binding parameter type is Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.

Name	Attributes
learnedLogical-ConnectionIds	Type: [Edm.String]

Table 23.14: Property for Sel.DeclineLearnedLogicalConnections

23.19 Sel.DeleteAll

URL: </api/default/config/logicalConnections/DeleteAll>

The binding parameter type is Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection.
This type has no properties.

23.20 Sel.DeleteLearnAndLockSession

URL: </api/default/config/LearnAndLockSessions/DeleteLearnAndLockSession>

The binding parameter type is Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.

Name	Attributes
sessionId	Type: Edm.String

Table 23.15: Property for Sel.DeleteLearnAndLockSession

23.21 Sel.DetectSelRelayFailoverPort

URL: [/api/default/operational/ports\('id'\)/DetectSelRelayFailoverPort](/api/default/operational/ports('id')/DetectSelRelayFailoverPort)

The binding parameter type is Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.
This type has no properties.

23.22 Sel.DisableApplicationLink

URL: [/api/default/security/applicationLinks\('id'\)/DisableApplicationLink](/api/default/security/applicationLinks('id')/DisableApplicationLink)

The binding parameter type is Sel.BlueFrame.ApplicationRegistration.ApplicationLink.
This type has no properties.

23.23 Sel.DisableControllerArpToNode

URL: [/api/default/operational/nodes\('id'\)/DisableControllerArpToNode](/api/default/operational/nodes('id')/DisableControllerArpToNode)

The binding parameter type is Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.
This type has no properties.

23.24 Sel.Disconnect

URL: [/api/default/operational/nodes\('id'\)/Disconnect](/api/default/operational/nodes('id')/Disconnect)

The binding parameter type is Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.
This type has no properties.

23.25 Sel.EnableApplicationLink

URL: [/api/default/security/applicationLinks\('id'\)/EnableApplicationLink](/api/default/security/applicationLinks('id')/EnableApplicationLink)

The binding parameter type is Sel.BlueFrame.ApplicationRegistration.ApplicationLink.
This type has no properties.

23.26 Sel.EnableControllerArpToNode

URL: [/api/default/operational/nodes\('id'\)/EnableControllerArpToNode](/api/default/operational/nodes('id')/EnableControllerArpToNode)

The binding parameter type is Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.
This type has no properties.

23.27 Sel.FactoryDefaultReset

URL: [/api/default/operational/nodes\('id'\)/FactoryDefaultReset](/api/default/operational/nodes('id')/FactoryDefaultReset)

The binding parameter type is Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.
This type has no properties.

23.28 Sel.FirmwareUpgrade

URL: [/api/default/operational/nodes\('id'\)/FirmwareUpgrade](/api/default/operational/nodes('id')/FirmwareUpgrade)

The binding parameter type is Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.

Name	Attributes
fileName	Type: Sel.FileArgument

Table 23.16: Property for Sel.FirmwareUpgrade

23.29 Sel.GetAvailableGroups

URL: `/api/default/security/authServices('id')/GetAvailableGroups`

The binding parameter type is `Sel.BlueFrame.Core.SecurityManager.AuthService`.
This type has no properties.

23.30 Sel.GetHardTimeoutTimeRemaining

URL: `/api/default/config/LearnAndLockSessions/GetHardTimeoutTimeRemaining`

The binding parameter type is `Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession`.

Name	Attributes
sessionId	Type: <code>Edm.String</code>

Table 23.17: Property for `Sel.GetHardTimeoutTimeRemaining`

23.31 Sel.GetLearnAndLockCounters

URL: `/api/default/config/LearnAndLockSessions/GetLearnAndLockCounters`

The binding parameter type is `Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession`.

Name	Attributes
sessionId	Type: <code>Edm.String</code>

Table 23.18: Property for `Sel.GetLearnAndLockCounters`

23.32 Sel.GetLinkByPortAttribute

URL: `/api/default/operational/GetLinkByPortAttribute`

Name	Attributes
attribute	Type: <code>Sel.Sel5056.TopologyManager.Attributes.AmAttribute</code>

Table 23.19: Property for `Sel.GetLinkByPortAttribute`

23.33 Sel.GetLogicalConnectionStats

URL: `/api/default/config/logicalConnections('id')/GetLogicalConnectionStats`

The binding parameter type is `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection`. This type has no properties.

23.34 Sel.GetNodeByAttribute

URL: `/api/default/operational/GetNodeByAttribute`

Name	Attributes
attribute	Type: <code>Sel.Sel5056.TopologyManager.Attributes.AmAttribute</code>

Table 23.20: Property for Sel.GetNodeByAttribute

23.35 Sel.GetNodesByIp

URL: `/api/default/operational/GetNodesByIp`

Name	Attributes
ip	Type: <code>Edm.String</code>

Table 23.21: Property for Sel.GetNodesByIp

23.36 Sel.GetNodesThatRequireSynchronization

URL: `/api/default/operational/GetNodesThatRequireSynchronization`

This type has no properties.

23.37 Sel.GetOpenFlowInformation

URL: `/api/default/config/logicalConnections('id')/GetOpenFlowInformation`

The binding parameter type is `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection`. This type has no properties.

23.38 Sel.GetPathPlanInformation

URL: `/api/default/config/logicalConnections('id')/GetPathPlanInformation`

The binding parameter type is `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection`.

This type has no properties.

23.39 Sel.GetPortByAttribute

URL: </api/default/operational/GetPortByAttribute>

Name	Attributes
attribute	Type: Sel.Sel5056.TopologyManager.Attributes.AmAttribute

Table 23.22: Property for Sel.GetPortByAttribute

23.40 Sel.GetStateBeginEndTimestamps

URL: </api/default/config/LearnAndLockSessions/GetStateBeginEndTimestamps>

The binding parameter type is Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.

Name	Attributes
sessionId	Type: Edm.String

Table 23.23: Property for Sel.GetStateBeginEndTimestamps

23.41 Sel.GetStatusMessages

URL: </api/default/config/LearnAndLockSessions/GetStatusMessages>

The binding parameter type is Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.

Name	Attributes
sessionId	Type: Edm.String

Table 23.24: Property for Sel.GetStatusMessages

23.42 Sel.GetUserPreference

URL: </api/default/settings/GetUserPreference>

Name	Attributes
defaultValue	Type: Edm.String
key	Type: Edm.String

Table 23.25: Properties for Sel.GetUserPreference

23.43 Sel.Heartbeat

URL: `/api/default/settings/Heartbeat`

This type has no properties.

23.44 Sel.InitiateApplicationRegistration

URL: `/api/default/security/applicationLinks/InitiateApplicationRegistration`

The binding parameter type is `Sel.BlueFrame.ApplicationRegistration.ApplicationLink`.

Name	Attributes
url	Type: <code>Edm.String</code>

Table 23.26: Property for `Sel.InitiateApplicationRegistration`

23.45 Sel.IsCommissioned

URL: `/api/default/security/IsCommissioned`

This type has no properties.

23.46 Sel.LearnAndLockIndividualSessionExportCSV

URL: `/api/default/config/LearnAndLockSessions/LearnAndLockIndividualSessionExportCSV`

The binding parameter type is `Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession`.

Name	Attributes
sessionId	Type: <code>Edm.String</code>

Table 23.27: Property for `Sel.LearnAndLockIndividualSessionExportCSV`

23.47 Sel.MarkPortAndAddRelayFailoverLink

URL: `/api/default/operational/ports('id')/MarkPortAndAddRelayFailoverLink`

The binding parameter type is `Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort`.

Name	Attributes
switchPortId	Type: <code>Edm.String</code>

Table 23.28: Property for `Sel.MarkPortAndAddRelayFailoverLink`

23.48 Sel.MarkPortAndPerformBlockingDetectSelRelayFailover

URL: [/api/default/operational/ports\('id'\)/MarkPortAndPerformBlockingDetectSelRelayFailover](/api/default/operational/ports('id')/MarkPortAndPerformBlockingDetectSelRelayFailover)

The binding parameter type is Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.
This type has no properties.

23.49 Sel.MarkPortAsSelRelayFailover

URL: [/api/default/operational/ports\('id'\)/MarkPortAsSelRelayFailover](/api/default/operational/ports('id')/MarkPortAsSelRelayFailover)

The binding parameter type is Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.
This type has no properties.

23.50 Sel.MergeNodes

URL: </api/default/operational/MergeNodes>

Name	Attributes
firstOperationalNodeId	Type: Edm.String
secondOperationalNodeId	Type: Edm.String

Table 23.29: Properties for Sel.MergeNodes

23.51 Sel.PacketOut

URL: [/api/default/operational/nodes\('id'\)/PacketOut](/api/default/operational/nodes('id')/PacketOut)

The binding parameter type is Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.

Name	Attributes
destinationPortId	Type: Edm.Int64
packetData	Type: Edm.String

Table 23.30: Properties for Sel.PacketOut

23.52 Sel.PacketOutToTable

URL: [/api/default/operational/nodes\('id'\)/PacketOutToTable](/api/default/operational/nodes('id')/PacketOutToTable)

The binding parameter type is `Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode`.

Name	Attributes
inPortId	Type: <code>Edm.Int64</code>
packetData	Type: <code>Edm.String</code>

Table 23.31: Properties for `Sel.PacketOutToTable`

23.53 Sel.ProposeLearnedLogicalConnections

URL: `/api/default/config/LearnAndLockSessions/ProposeLearnedLogicalConnections`

The binding parameter type is `Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession`.

Name	Attributes
learnedLogical-ConnectionIds	Type: [<code>Edm.String</code>]

Table 23.32: Property for `Sel.ProposeLearnedLogicalConnections`

23.54 Sel.Reboot

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

The binding parameter type is `Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode`.
This type has no properties.

23.55 Sel.RemoveSelRelayFailoverFromPort

URL: `/api/default/operational/ports('id')/RemoveSelRelayFailoverFromPort`

The binding parameter type is `Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort`.
This type has no properties.

23.56 Sel.ReplaceConfig

URLs:

- `/api/default/operational/links('id')/ReplaceConfig`
- `/api/default/operational/nodes('id')/ReplaceConfig`

- [/api/default/operational/ports\('id'\)/ReplaceConfig](#)

There are three instances of this action:

-[Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink](#)
 -[Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode](#)
 -[Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort](#)

23.56.1 **Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink**

The binding parameter type is [Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink](#).

Name	Attributes
configKey	Type: Edm.String

Table 23.33: Property for [Sel.ReplaceConfig](#)

23.56.2 **Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode**

The binding parameter type is [Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode](#).

Name	Attributes
configKey	Type: Edm.String

Table 23.34: Property for [Sel.ReplaceConfig](#)

23.56.3 **Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort**

The binding parameter type is [Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort](#).

Name	Attributes
configKey	Type: Edm.String

Table 23.35: Property for [Sel.ReplaceConfig](#)

23.57 **Sel.ReplanAll**

URL: [/api/default/config/logicalConnections/ReplanAll](#)

The binding parameter type is [Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection](#).
 This type has no properties.

23.58 Sel.ReplanInbandPath

URL: [/api/default/operational/nodes\('id'\)/ReplanInbandPath](/api/default/operational/nodes('id')/ReplanInbandPath)

The binding parameter type is Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode.
This type has no properties.

23.59 Sel.ReserveVlanVidRange

URL: </api/default/config/vlanVidReservation/ReserveVlanVidRange>

The binding parameter type is Sel.Sel5056.PathProgrammer.VlanVidManager.VlanVidReservation.

Name	Attributes
end	Type: Edm.Int32
start	Type: Edm.Int32

Table 23.36: Properties for Sel.ReserveVlanVidRange

23.60 Sel.ResetCounters

URL: [/api/default/config/flows\('id'\)/ResetCounters](/api/default/config/flows('id')/ResetCounters)

The binding parameter type is Sel.Sel5056.OpenFlowPlugin.DataTreeObjects.Flow.
This type has no properties.

23.61 Sel.Restore

URL: </api/default/settings/Restore>

Name	Attributes
fileName	Type: Sel.BlueFrame.Core.FileArgument
password	Type: Edm.String

Table 23.37: Properties for Sel.Restore

23.62 Sel.ResubmitLogicalConnection

URL: [/api/default/config/logicalConnections\('id'\)/ResubmitLogicalConnection](/api/default/config/logicalConnections('id')/ResubmitLogicalConnection)

The binding parameter type is `Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection`. This type has no properties.

23.63 Sel.RevokeCertificate

URL: [/api/default/certificate/certificateInfo\('id'\)/RevokeCertificate](/api/default/certificate/certificateInfo('id')/RevokeCertificate)

The binding parameter type is `Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation`. This type has no properties.

23.64 Sel.RevokeCertificateByPurpose

URL: </api/default/certificate/RevokeCertificateByPurpose>

Name	Attributes
purpose	Type: Sel.BlueFrame.Core.TrustAuthority.CertificatePurpose

Table 23.38: Property for `Sel.RevokeCertificateByPurpose`

23.65 Sel.RevokeCertificateByThumbprint

URL: </api/default/certificate/RevokeCertificateByThumbprint>

Name	Attributes
thumbprint	Type: Edm.String

Table 23.39: Property for `Sel.RevokeCertificateByThumbprint`

23.66 Sel.SendPortMod

URLs:

- [/api/default/config/ports\('id'\)/SendPortMod](/api/default/config/ports('id')/SendPortMod)
- [/api/default/operational/ports\('id'\)/SendPortMod](/api/default/operational/ports('id')/SendPortMod)

There are two instances of this action:

- [Sel.Sel5056.TopologyManager.Ports.ConfigPort](#)
- [Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort](#)

23.66.1 Sel.Sel5056.TopologyManager.Ports.ConfigPort

The binding parameter type is Sel.Sel5056.TopologyManager.Ports.ConfigPort.

Name	Attributes
configFlags	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-ConfigFlagObjects.ConfigBase]
featuresFlags	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.FeatureBase]

Table 23.40: Properties for Sel.SendPortMod

23.66.2 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

The binding parameter type is Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.

Name	Attributes
configFlags	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-ConfigFlagObjects.ConfigBase]
featuresFlags	Type: [Sel.Sel5056.TopologyManager.OpenFlow.PortMod.-FeatureFlagObjects.FeatureBase]

Table 23.41: Properties for Sel.SendPortMod

23.67 Sel.SetDone

URL: /api/default/config/LearnAndLockSessions/SetDone

The binding parameter type is Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.

This type has no properties.

23.68 Sel.SetUserPreference

URL: /api/default/settings/SetUserPreference

Name	Attributes
key	Type: Edm.String
value	Type: Edm.String

Table 23.42: Properties for Sel.SetUserPreference

23.69 Sel.StopLearnAndLock

URL: </api/default/config/LearnAndLockSessions/StopLearnAndLock>

The binding parameter type is Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.
This type has no properties.

23.70 Sel.SynchronizeAllConfiguration

URL: </api/default/operational/SynchronizeAllConfiguration>

This type has no properties.

23.71 Sel.SynchronizeConfiguration

URL: </api/default/operational/SynchronizeConfiguration>

Name	Attributes
configNodeId	Type: Edm.String

Table 23.43: Property for Sel.SynchronizeConfiguration

23.72 Sel.SynchronizeOpenFlowConfiguration

URL: </api/default/operational/SynchronizeOpenFlowConfiguration>

Name	Attributes
configNodeId	Type: Edm.String

Table 23.44: Property for Sel.SynchronizeOpenFlowConfiguration

23.73 Sel.SynchronizeSelectedConfiguration

URL: </api/default/operational/SynchronizeSelectedConfiguration>

Name	Attributes
configNodeIds	Type: [Edm.String]

Table 23.45: Property for Sel.SynchronizeSelectedConfiguration

23.74 Sel.TestAuthenticateUser

URL: [/api/default/security/authServices\('id'\)/TestAuthenticateUser](/api/default/security/authServices('id')/TestAuthenticateUser)

The binding parameter type is Sel.BlueFrame.Core.SecurityManager.AuthService.

Name	Attributes
password	Type: Edm.String
username	Type: Edm.String

Table 23.46: Properties for Sel.TestAuthenticateUser

23.75 Sel.TrustCertificate

URL: [/api/default/certificate/certificateInfo\('id'\)/TrustCertificate](/api/default/certificate/certificateInfo('id')/TrustCertificate)

The binding parameter type is Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation. This type has no properties.

23.76 Sel.Unadopt

URLs:

- [/api/default/operational/links\('id'\)/Unadopt](/api/default/operational/links('id')/Unadopt)
- [/api/default/operational/nodes\('id'\)/Unadopt](/api/default/operational/nodes('id')/Unadopt)
- [/api/default/operational/ports\('id'\)/Unadopt](/api/default/operational/ports('id')/Unadopt)

There are three instances of this action:

- [Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink](#)
- [Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode](#)
- [Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort](#)

23.76.1 Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink

The binding parameter type is Sel.Sel5056.TopologyManager.Links.OperationalNetworkLink. This type has no properties.

23.76.2 Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode

The binding parameter type is Sel.Sel5056.TopologyManager.Nodes.OperationalNetworkNode. This type has no properties.

23.76.3 Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort

The binding parameter type is Sel.Sel5056.TopologyManager.Ports.OperationalNetworkPort.
This type has no properties.

23.77 Sel.UpdatePassword

URL: [/api/default/security/users\('id'\)/UpdatePassword](/api/default/security/users('id')/UpdatePassword)

The binding parameter type is Sel.BlueFrame.Core.SecurityManager.User.

Name	Attributes
newPassword	Type: Edm.String
oldPassword	Type: Edm.String

Table 23.47: Properties for Sel.UpdatePassword

23.78 Sel.UploadCertificate

URL: </api/default/certificate/UploadCertificate>

Name	Attributes
base64Certificate	Type: Edm.String
certificatePassword	Type: Edm.String
purpose	Type: Sel.BlueFrame.Core.TrustAuthority.CertificatePurpose

Table 23.48: Properties for Sel.UploadCertificate

23.79 Sel.UserInitiatedHostDiscovery

URL: </api/default/operational/UserInitiatedHostDiscovery>

Name	Attributes
ip	Type: Edm.String

Table 23.49: Property for Sel.UserInitiatedHostDiscovery

Chapter 24

OData Enums

24.1 Sel.BlueFrame.Core.ErrorState

Key	Value
Failure	2
FlowStatTransaction-Failure	4
InProgress	0
Success	1
ValidationError	3

Table 24.1: Sel.BlueFrame.Core.ErrorState

Relationship between this type and other types:

- 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

24.2 Sel.BlueFrame.Core.EventBus.Enums.DurationType

Key	Value
Momentary	0
Persistent	1

Table 24.2: Sel.BlueFrame.Core.EventBus.Enums.DurationType

Relationship between this type and other types:

- Sel.BlueFrame.Core.EventBus.EventType.durationSetting
- Sel.BlueFrame.Core.EventBus.EventType

24.3 Sel.BlueFrame.Core.EventBus.Enums.FacilityCode

Key	Value
ClockDaemon1	9
ClockDaemon2	15
FtpDaemon	11
InternalMessages	5
KernelMessages	0
LinePrinterSubsystem	6
LocalUse0	16
LocalUse1	17
LocalUse2	18
LocalUse3	19
LocalUse4	20
LocalUse5	21
LocalUse6	22
LocalUse7	23
LogAlert	14

LogAudit	13
MailSystem	2
NetworkNewsSubsystem	7
NtpSubsystem	12
SecurityOrAuthorization-Messages1	4
SecurityOrAuthorization-Messages2	10
SystemDaemons	3
UserLevelMessages	1
UucpSubsystem	8

Table 24.3: *Sel.BlueFrame.Core.EventBus.Enums.FacilityCode*

Relationship between this type and other types:

— *Sel.BlueFrame.Core.EventBus.EventType.facility*

—— *Sel.BlueFrame.Core.EventBus.EventType*

24.4 *Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel*

Key	Value
Alert	1
Critical	2
Debug	7
Emergency	0
Error	3
Informational	6
Notice	5
Warning	4

Table 24.4: *Sel.BlueFrame.Core.EventBus.Enums.SeverityLevel*

Relationship between this type and other types:

— *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
 —Sel.Sel5056.TopologyManager.Nodes.PortLinkEventConfiguration.linkDownSeverity
 —Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode.portLinkEventConfigurations
 —Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode
 —Sel.Sel5056.TopologyManager.Nodes.PortLinkEventConfiguration.linkUpSeverity
 —Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode.portLinkEventConfigurations
 —Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode

24.5 Sel.BlueFrame.Core.TrustAuthority.Enums.Certificate-Status

Key	Value
Expired	2
ExternallyRevoked	3
InternallyRevoked	1
Valid	0

Table 24.5: Sel.BlueFrame.Core.TrustAuthority.Enums.CertificateStatus

Relationship between this type and other types:

—Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation.status
 —Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation

24.6 Sel.BlueFrame.Core.TrustAuthority.Enums.User-Interaction

Key	Value
NotUserUploaded	0
UserUploaded	1

Table 24.6: Sel.BlueFrame.Core.TrustAuthority.Enums.UserInteraction

Relationship between this type and other types:

—Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation.userInteraction
 —Sel.BlueFrame.TrustAuthority.DataTreeObjects.CertificateInformation

24.7 Sel.BlueFrame.EventBus.Syslog.RFCType

Key	Value
Rfc3164	0
Rfc5424	1

Table 24.7: Sel.BlueFrame.EventBus.Syslog.RFCType

Relationship between this type and other types:

- 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

24.8 Sel.BlueFrame.EventBus.Syslog.TransportType

Key	Value
SecureTcp	1
Udp	0

Table 24.8: Sel.BlueFrame.EventBus.Syslog.TransportType

Relationship between this type and other types:

- 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

24.9 Sel.BlueFrame.SecurityManager.AuthServiceResultState

Key	Value
Error	2
Info	0
Success	3
Warning	1

Table 24.9: Sel.BlueFrame.SecurityManager.AuthServiceResultState

Relationship between this type and other types:

- Sel.BlueFrame.SecurityManager.AuthServiceResult.state
- Sel.BlueFrame.SecurityManager.AuthServiceResultCollection.results

24.10 Sel.Sel5056.Common.OpenFlowNodeProgrammer.- Options.NodesConfiguration

Key	Value
All	0
Only2740S	1

Table 24.10: Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.NodesConfiguration

Relationship between this type and other types:

- Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions.nodesSetting

24.11 Sel.Sel5056.Common.OpenFlowNodeProgrammer.- Options.OutputConfiguration

Key	Value
Controller	0
Local	1
NextTable	2

Table 24.11: Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.OutputConfiguration

Relationship between this type and other types:

- Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions.outputSetting

24.12 Sel.Sel5056.Common.OpenFlowNodeProgrammer.- Options.TableConfiguration

Key	Value
AllTables	1
SpecificTable	0

Table 24.12: Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.TableConfiguration

Relationship between this type and other types:

- Sel.Sel5056.Common.OpenFlowNodeProgrammer.Options.FlowProgrammingOptions.tableSetting

24.13 Sel.Sel5056.LearnAndLock.Enums.EndReason

Key	Value
Automatic	1
ByUser	2
ForReason	3
NA	0
Timeout	4

Table 24.13: Sel.Sel5056.LearnAndLock.Enums.EndReason

Relationship between this type and other types:

- Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.autoAdoptionEndReason
- [Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession](#)
- Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.logicalConnectionLearningEndReason
- [Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession](#)

24.14 Sel.Sel5056.LearnAndLock.LearnAndLockStateStatus

Key	Value
AutoAdoptDone	3
AutoAdoptNotStarted	1
AutoAdoptRunning	2
Failure	11
LearnAndLockDone	9
LearnAndLockNotStarted	7
LearnAndLockRunning	8
LogicalConnection-LearningDone	6
LogicalConnection-LearningNotStarted	4
LogicalConnection-LearningRunning	5
None	0
Success	10

Table 24.14: Sel.Sel5056.LearnAndLock.LearnAndLockStateStatus

Relationship between this type and other types:

- 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

24.15 Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.-AutoAcceptStatus

Key	Value
All	0
ArpOnly	1
None	2

Table 24.15: Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.AutoAcceptStatus

Relationship between this type and other types:

—Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.Configuration.LogicalConnectionLearningConfig.-autoAcceptStatus
 —Sel.BeginLearnAndLock.logicalConnectionLearningConfig
 —Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession.logicalConnectionLearningConfig
 —Sel.Sel5056.LearnAndLock.Sessions.LearnAndLockSession

24.16 Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.-LearnedLogicalConnectionState

Key	Value
Accepted	2
Cleared	6
Declined	5
Disregarded	4
Failed	7
Filtered	3
None	0
Programmed	1
Proposed	8

Table 24.16: Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnectionState

Relationship between this type and other types:

—Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection.state
 —Sel.Sel5056.LearnAndLock.LogicalConnectionLearning.LearnedLogicalConnection

24.17 Sel.Sel5056.LogicalConnectionValidation.Enums.CstCommunicationType

Key	Value
ControllerRoute	4
Multicast	2
Undefined	0
Unicast	1
UnicastBidirectional	3

Table 24.17: Sel.Sel5056.LogicalConnectionValidation.Enums.CstCommunicationType

Relationship between this type and other types:

—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](#)

24.18 Sel.Sel5056.LogicalConnectionValidation.Enums.LogicalConnectionOrigin

Key	Value
ControllerSwitch-Communication	2
EndUser	0
L2Learning	1

Table 24.18: Sel.Sel5056.LogicalConnectionValidation.Enums.LogicalConnectionOrigin

Relationship between this type and other types:

—Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection.connectionOrigin

—[Sel.Sel5056.LogicalConnectionValidation.DataObject.LogicalConnection](#)

24.19 Sel.Sel5056.OpenFlowPlugin.Enums.OF13MeterBandType

Key	Value
-----	-------

OFPMBT_DROP	1
OFPMBT_DSCP_REMARK	2
OFPMBT_EXPERIMENTER	65535

Table 24.19: Sel.Sel5056.OpenFlowPlugin.Enums.OF13MeterBandType

Relationship between this type and other types:

- 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](#)

24.20 Sel.Sel5056.OpenFlowPlugin.Enums.OfpGroupType

Key	Value
All	0
FastFailover	3
Indirect	2
Select	1

Table 24.20: Sel.Sel5056.OpenFlowPlugin.Enums.OfpGroupType

Relationship between this type and other types:

- 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](#)

24.21 Sel.Sel5056.SapphirePlugin.Synchronization.Sapphire-Endpoint

Key	Value
-----	-------

AlarmContact	8
BasicSnmp	3
ControllerIp	9
DigitalInputs	12
FirmwareUpgrade-Required	16
GigabitFastMode	10
Logging	1
Networking	2
None	0
NtpGlobal	7
ParticleSnmp	13
ParticleSnmpSwitch-Config	14
PortEventConfiguration	17
PowerOverEthernet	11
Ptp	4
SnmpUsers	15
TimeSourceSelection	18
TrustedCertificateAdd	5
TrustedCertificateRevoke	6

Table 24.21: Sel.Sel5056.SapphirePlugin.Synchronization.SapphireEndpoint

Relationship between this type and other types:

—Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization.endpoint

——[Sel.Sel5056.SapphirePlugin.Synchronization.SapphireConfigSynchronization](#)

24.22 Sel.Sel5056.Snmp.SnmpAuthenticationMethod

Key	Value
MD5	0
SHA1	1
SHA256	2
SHA512	3

Table 24.22: Sel.Sel5056.Snmp.SnmpAuthenticationMethod

Relationship between this type and other types:

—Sel.Sel5056.Snmp.SnmpUser.authenticationMethod

——[Sel.Sel5056.Snmp.SnmpUser](#)

24.23 Sel.Sel5056.Snmp.SnmpEncryptionMethod

Key	Value
AES	0

Table 24.23: Sel.Sel5056.Snmp.SnmpEncryptionMethod

Relationship between this type and other types:

—Sel.Sel5056.Snmp.SnmpUser.encryptionMethod

——[Sel.Sel5056.Snmp.SnmpUser](#)

24.24 Sel.Sel5056.TopologyManager.Attributes.Operational.-Node.DeviceFramework

Key	Value
BlueFrame	2
Omni	1
Unknown	0

Table 24.24: Sel.Sel5056.TopologyManager.Attributes.Operational.Node.DeviceFramework

Relationship between this type and other types:

—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](#)

24.25 Sel.Sel5056.TopologyManager.Attributes.Operational.-Node.Sel274xNodeType

Key	Value
Sel2740S	0
Sel2742S	1

Table 24.25: Sel.Sel5056.TopologyManager.Attributes.Operational.Node.Sel274xNodeType

Relationship between this type and other types:

- Sel.Sel5056.TopologyManager.Attributes.Operational.Port.Sel2740SPortAttr.portType
- 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.Node.SelSapphireAttr.nodeType

24.26 Sel.Sel5056.TopologyManager.Enums.Communication-Band

Key	Value
InBand	0
OutOfBand	1

Table 24.26: Sel.Sel5056.TopologyManager.Enums.CommunicationBand

Relationship between this type and other types:

- 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](#)

24.27 Sel.Sel5056.TopologyManager.Enums.DiscoveryTrust-State

Key	Value
Certificate	1

Internal	2
Untrusted	0
User	3

Table 24.27: Sel.Sel5056.TopologyManager.Enums.DiscoveryTrustState

Relationship between this type and other types:

- 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](#)

24.28 Sel.Sel5056.TopologyManager.Enums.OfpPortStatus

Key	Value
Blocked	2
LinkDown	1
Live	4
None	0

Table 24.28: Sel.Sel5056.TopologyManager.Enums.OfpPortStatus

Relationship between this type and other types:

- 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](#)

24.29 Sel.Sel5056.TopologyManager.Enums.PacketSource

Key	Value
Controller	0
OpenFlow	1

Table 24.29: Sel.Sel5056.TopologyManager.Enums.PacketSource

Relationship between this type and other types:

- 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](#)

24.30 Sel.Sel5056.TopologyManager.Enums.Sel2742SDigitalInputFunctions

Key	Value
Disabled	0
EnableSettingsLock	2
FactoryDefaultReset	1

Table 24.30: Sel.Sel5056.TopologyManager.Enums.Sel2742SDigitalInputFunctions

Relationship between this type and other types:

- Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode.digitalInput1Function
- [Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode](#)
- Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode.digitalInput2Function
- [Sel.Sel5056.TopologyManager.Nodes.Sel2742SConfigNode](#)

24.31 Sel.Sel5056.TopologyManager.Enums.Sel274xTimeSource

Key	Value
Acts	2

None	0
Ntp	1

Table 24.31: Sel.Sel5056.TopologyManager.Enums.Sel274xTimeSource

Relationship between this type and other types:

—Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode.timeSourceSelection

——[Sel.Sel5056.TopologyManager.Nodes.Sel2740SConfigNode](#)

24.32 Sel.Sel5056.TopologyManager.Enums.State

Key	Value
Adopted	1
Configured	5
Disconnected	3
Established	4
None	0
Unadopted	2

Table 24.32: Sel.Sel5056.TopologyManager.Enums.State

Relationship between this type and other types:

—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](#)

24.33 System.Diagnostics.EventLogEntryType

Key	Value
Error	1
FailureAudit	16
Information	4
SuccessAudit	8

Warning	2
---------	---

Table 24.33: *System.Diagnostics.EventLogEntryType*

Relationship between this type and other types:

- 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

Part VI

Appendix

Appendix A

Manual Versions

Table A.1 lists the manual versions and revision descriptions. The most recent manual version is listed first.

Date Code	Controller Version	Summary of Revisions
20220119	2.8.0	Initial version

Table A.1: Manual Revision History

Index

/api/default/certificate/RevokeCertificateByPurpose
POST, 293

/api/default/certificate/RevokeCertificateByThumbprint
POST, 294

/api/default/certificate/UploadCertificate
POST, 290

/api/default/certificate/certificateInfo('id')/
DELETE, 291

/api/default/certificate/certificateInfo('id')/AsPem
POST, 289

/api/default/certificate/certificateInfo('id')/RevokeCertificate
POST, 291

/api/default/certificate/certificateInfo('id')/TrustCertificate
POST, 295

/api/default/certificate/certificateInfo/
GET, 288

/api/default/config/communicationServiceTypes('id')/
DELETE, 218
PATCH, 217
PUT, 216

/api/default/config/communicationServiceTypes/
GET, 211
POST, 212

/api/default/config/flows('id')/
DELETE, 234
PATCH, 230
PUT, 230

/api/default/config/flows('id')/ResetCounters
POST, 233

/api/default/config/flows/
GET, 225
POST, 226

/api/default/config/groups('id')/
DELETE, 250
PATCH, 247

- PUT, 243
- /api/default/config/groups/
 - GET, 236
 - POST, 237
- /api/default/config/links('id')/
 - DELETE, 193
 - PATCH, 192
 - PUT, 191
- /api/default/config/links/
 - GET, 188
 - POST, 189
- /api/default/config/logicalConnections('id')/
 - DELETE, 205
 - PATCH, 203
 - PUT, 204
- /api/default/config/logicalConnections('id')/GetLogicalConnectionStats
 - POST, 209
- /api/default/config/logicalConnections('id')/GetOpenFlowInformation
 - POST, 209
- /api/default/config/logicalConnections('id')/GetPathPlanInformation
 - POST, 210
- /api/default/config/logicalConnections('id')/ResubmitLogicalConnection
 - POST, 206
- /api/default/config/logicalConnections/
 - GET, 200
 - POST, 201
- /api/default/config/logicalConnections/DeleteAll
 - POST, 206
- /api/default/config/logicalConnections/ReplanAll
 - POST, 208
- /api/default/config/meters('id')/
 - DELETE, 257
 - PATCH, 256
 - PUT, 255
- /api/default/config/meters/
 - GET, 253
 - POST, 254
- /api/default/config/nodes('id')/
 - DELETE, 150
 - PATCH, 146
 - PUT, 145
- /api/default/config/nodes/
 - GET, 138
 - POST, 141
- /api/default/config/ports('id')/

- DELETE, 178
- PATCH, 177
- PUT, 176
- /api/default/config/ports('id')/SendPortMod
 - POST, 186
- /api/default/config/ports/
 - GET, 173
 - POST, 175
- /api/default/config/vlanVidReservation('id')/
 - DELETE, 222
- /api/default/config/vlanVidReservation/
 - GET, 219
- /api/default/config/vlanVidReservation/ReserveVlanVidRange
 - POST, 220
- /api/default/operational/AuditDeviceConfiguration
 - POST, 166
- /api/default/operational/GetNodesThatRequireSynchronization
 - POST, 156
- /api/default/operational/MergeNodes
 - POST, 149
- /api/default/operational/SynchronizeAllConfiguration
 - POST, 165
- /api/default/operational/SynchronizeConfiguration
 - POST, 164
- /api/default/operational/SynchronizeSelectedConfiguration
 - POST, 165
- /api/default/operational/UserInitiatedHostDiscovery
 - POST, 153
- /api/default/operational/flowStats/
 - GET, 234
- /api/default/operational/groupDesc/
 - GET, 251
- /api/default/operational/groupStats/
 - GET, 251
- /api/default/operational/links('id')/Adopt
 - POST, 194
- /api/default/operational/links('id')/AdoptWithConfig
 - POST, 194
- /api/default/operational/links('id')/ReplaceConfig
 - POST, 193
- /api/default/operational/links('id')/Unadopt
 - POST, 195
- /api/default/operational/links/
 - GET, 188
- /api/default/operational/meterDescriptions/

- GET, 257
- /api/default/operational/meterFeatures/
 - GET, 258
- /api/default/operational/meterstats/
 - GET, 259
- /api/default/operational/nodes('id')/Adopt
 - POST, 151
- /api/default/operational/nodes('id')/AdoptWithConfig
 - POST, 152
- /api/default/operational/nodes('id')/DisableControllerArpToNode
 - POST, 154
- /api/default/operational/nodes('id')/Disconnect
 - POST, 150
- /api/default/operational/nodes('id')/EnableControllerArpToNode
 - POST, 153
- /api/default/operational/nodes('id')/FactoryDefaultReset
 - POST, 167
- /api/default/operational/nodes('id')/FirmwareUpgrade
 - POST, 167
- /api/default/operational/nodes('id')/PacketOut
 - POST, 278
- /api/default/operational/nodes('id')/PacketOutToTable
 - POST, 279
- /api/default/operational/nodes('id')/Reboot
 - POST, 169
- /api/default/operational/nodes('id')/ReplaceConfig
 - POST, 149
- /api/default/operational/nodes('id')/ReplanInbandPath
 - POST, 207
- /api/default/operational/nodes('id')/Unadopt
 - POST, 152
- /api/default/operational/nodes/
 - GET, 140
- /api/default/operational/packets/
 - GET, 277
- /api/default/operational/portDesc/
 - GET, 184
- /api/default/operational/portStats/
 - GET, 185
- /api/default/operational/ports('id')/AddAbstractNode
 - POST, 172
- /api/default/operational/ports('id')/AddOperationalLinkFromHostToSwitch
 - POST, 190
- /api/default/operational/ports('id')/AddSilentHost
 - POST, 155

/api/default/operational/ports('id')/Adopt
POST, 179

/api/default/operational/ports('id')/AdoptWithConfig
POST, 179

/api/default/operational/ports('id')/DetectSelRelayFailoverPort
POST, 180

/api/default/operational/ports('id')/MarkPortAndAddRelayFailoverLink
POST, 182

/api/default/operational/ports('id')/MarkPortAndPerformBlockingDetectSelRelayFailover
POST, 182

/api/default/operational/ports('id')/MarkPortAsSelRelayFailover
POST, 181

/api/default/operational/ports('id')/RemoveSelRelayFailoverFromPort
POST, 183

/api/default/operational/ports('id')/ReplaceConfig
POST, 178

/api/default/operational/ports('id')/SendPortMod
POST, 186

/api/default/operational/ports('id')/Unadopt
POST, 180

/api/default/operational/ports/
GET, 174

/api/default/operational/synchronization/
GET, 157

/api/default/operational/tableStats/
GET, 156

/api/default/security/Commission
POST, 262

/api/default/security/IsCommissioned
POST, 261

/api/default/security/applicationLinks('id')/
DELETE, 283

/api/default/security/applicationLinks('id')/DisableApplicationLink
POST, 285

/api/default/security/applicationLinks('id')/EnableApplicationLink
POST, 285

/api/default/security/applicationLinks/
GET, 284
POST, 283

/api/default/security/applicationLinks/InitiateApplicationRegistration
POST, 282

/api/default/security/authServiceGroups('id')/
DELETE, 131
PATCH, 130
PUT, 129

- /api/default/security/authServiceGroups/
 - GET, 128
 - POST, 128
- /api/default/security/authServices('id')/
 - DELETE, 126
 - PATCH, 125
 - PUT, 124
- /api/default/security/authServices('id')/TestAuthenticateUser
 - POST, 127
- /api/default/security/authServices/
 - GET, 122
 - POST, 123
- /api/default/security/roles/
 - GET, 121
- /api/default/security/snmpUsers('id')/
 - PATCH, 171
- /api/default/security/snmpUsers/
 - GET, 170
- /api/default/security/users('id')/
 - DELETE, 120
 - PATCH, 118
 - PUT, 117
- /api/default/security/users('id')/UpdatePassword
 - POST, 120
- /api/default/security/users/
 - GET, 113
 - POST, 115
- /api/default/settings/Backup
 - POST, 262
- /api/default/settings/GetUserPreference
 - POST, 131
- /api/default/settings/Restore
 - POST, 263
- /api/default/settings/SetUserPreference
 - POST, 133
- /api/default/settings/crlSettings('id')/
 - PATCH, 297
 - PUT, 298
- /api/default/settings/crlSettings/
 - GET, 297
- /api/default/settings/eventCategories('id')/
 - PATCH, 265
 - PUT, 266
- /api/default/settings/eventCategories/
 - GET, 264

- `/api/default/settings/eventTypes('id')/`
 - PATCH, [268](#)
 - PUT, [268](#)
 - `/api/default/settings/eventTypes/`
 - GET, [267](#)
 - `/api/default/settings/networkSettings('id')/`
 - PATCH, [197](#)
 - PUT, [196](#)
 - `/api/default/settings/networkSettings/`
 - GET, [195](#)
 - `/api/default/settings/preferences/`
 - GET, [132](#)
 - `/api/default/settings/securityManager('id')/`
 - PATCH, [270](#)
 - PUT, [271](#)
 - `/api/default/settings/securityManager/`
 - GET, [269](#)
 - `/api/default/settings/strings/`
 - GET, [275](#)
 - `/api/default/settings/webSettings('id')/`
 - PATCH, [273](#)
 - PUT, [274](#)
 - `/api/default/settings/webSettings/`
 - GET, [272](#)

