



Cisco Crosswork Hierarchical Controller 6.0

NBI および SHQL 参照ガイド

2022 年 10 月

目次

Crosswork Hierarchical Controller API の紹介	6
API アーキテクチャおよびテクノロジー	6
RESTCONF/YANG プロトコルの紹介	6
クエリのフィルタ処理.....	7
転送のセキュリティ	7
データの完全性	7
サーバー認証	7
クライアント認証	7
API エンドポイント	7
オブジェクト ID	8
デバイス管理.....	8
アダプタの入手	9
アダプタ構成スキーマの取得	10
アダプタデバイススキーマの取得	14
すべてのデバイスの取得	15
Crosswork Hierarchical Controller へのデバイスの追加	17
デバイスをアダプタに割り当てる	18
デバイスのアダプタ構成の更新	19
アダプタからのデバイス割り当て解除	20
デバイス名の更新	20
サイトへのデバイスの割り当て	21
デバイスステータスの取得	21
デバイスの取得	22
Crosswork Hierarchical Controller からのデバイスの削除	23
クレデンシャルの取得	23
新しいクレデンシャルの追加	24
クレデンシャルの更新	25
クレデンシャルの削除	26
指定されたクレデンシャルタイプのパラメータスキーマの取得	26
物理インベントリ	27
物理インベントリの取得	27

パフォーマンス カウンタ	29
URI	错误!未定义书签。
パフォーマンスの最新統計の取得	30
期間中のパフォーマンス統計の取得	31
特定のポートのパフォーマンス統計の送信	33
クロスリンク	35
隣接するすべてのクロスリンクの取得	36
すべてのクロスリンクの検証	36
1つのクロスリンクの検証	37
すべての手動クロスリンクの取得	39
クロスリンクの取得	40
Crosswork Hierarchical Controller への手動クロスリンクの追加	41
クロスリンクの削除	42
共有リスク	43
ポリシーの取得	43
ポリシーの取得	45
ポリシーの作成	46
ポリシーの削除	46
ポリシーの共有リスクタイプの更新	47
ポリシータイプの更新	47
ポリシーへのルールの追加	48
ルールの更新	49
ポリシーからのルールの削除	50
SHQL を使用した一般的な情報の検索	51
要求メソッド	51
要求 URL (Request URL)	51
要求の例	51
応答の例	52
SQHL の概要	53
SHQL - 必要性	53
SHQL の概要	53
Crosswork Hierarchical Controller のオブジェクトモデル	54
インベントリタイプ	54
リンク タイプ	55

Crosswork Hierarchical Controller の接続モデル	56
SQL の例	56
SHQL の使用	57
SHQL インターフェースについて	57
クエリの実行	57
クエリの中止	59
クエリの作成	60
クエリの保存	61
クエリの削除	62
列のソート	62
条件の作成	64
例 1 : オペランドによるフィルタ処理	65
例 2 : オペランドによるフィルタ処理	65
例 3 : オペランドによるフィルタ処理	66
例 4 : オペランドによるフィルタ処理	67
オブジェクトタイプの変換/収集	68
例 : オブジェクトタイプの変換	70
例 : オブジェクトタイプのコレクション	70
例 : As 一時変数	71
機能	73
例 : downward 関数	74
例 : retrospective 関数	74
例 : group_by 関数	75
履歴関数	76
例 : history	76
その他の出力関数	77
例 : after	78
例 1 : add_counters	78
例 2 : add_counters	79
例 : view	80
例 2 : view	80
例 3 : view	81
例 4 : view	81

複雑な例.....	82
複雑な例 1	82
複雑な例 2	82
複雑な例 3	83
複雑な例 4	84
実際に試す	85
タイムマシンを使う	86
オブジェクトタイプのプロパティ	87
Inventory_item プロパティ	87
port プロパティ	89
link プロパティ	92
site プロパティ	94
service プロパティ	94
service_intent プロパティ	95
service_intent_resource プロパティ	97
event プロパティ	99
srlg Resource プロパティ	99
path プロパティ	99
site_link プロパティ	100
visual_site プロパティ	100
inventory プロパティ	101
srlg_risk_resource_mtm プロパティ	103
risk_resource プロパティ	103
region プロパティ	103
history プロパティ	104
SHQL ウィジェットの作成	104
SHQL ウィジェットについて	104
SHQL ウィジェットの作成	105
付録	106
用語集	106

Crosswork Hierarchical Controller API の紹介

Cisco Crosswork Hierarchical Controller は、サービスオーケストレータ、OSS、インベントリ システムへのノースバウンド インターフェイスとして、広範な API セットを提供しています。これらの API は、運用ワークフロー、計画シナリオ、管理タスクを自動化するために不可欠なツールです。これらの API を使用して、情報の取得、サービスとリソースのプロビジョニング、ネットワークの構築、通知の登録を行うことができます。

Crosswork Hierarchical Controller は、これらの API に複数のテクノロジーを利用しています。

- RESTCONF - NETCONF 用に定義された YANG モデルに基づく
- 本文に JSON を含む REST
- 本文に SHQL を含む REST

API アーキテクチャおよびテクノロジー

RESTCONF/YANG プロトコルの紹介

RESTCONF プロトコルは、ネットワーク設定プロトコル (NETCONF) [RFC 6241] で定義され、NMDA [RFC 8342] でさらに拡張されたデータストアの概念を使用して、YANG 1.0 [RFC 6020] または YANG 1.1 [RFC 7950] で定義されたデータにアクセスするためのプログラムインターフェイスを提供する HTTP ベースのプロトコルです。

RESTCONF を選択する主な理由は次のとおりです。

- モデル駆動型 API を提供するために HTTP CRUD 操作をマッピングする機能。
- 多くの Web アプリケーションで使用されるユビキタスな REST アーキテクチャを使用する機能。
- 複数の SDO によって生成された YANG モデリング言語で定義されたさまざまな標準データモデルを使用および混合する機能。これにより、実装者は SDN アーキテクチャ (IETF ACTN、ONF TAPI、MEF TAPI) を選択できる可能性があります。

Crosswork Hierarchical Controller の RESTCONF 実装は、次のリソースで構成されています。

- **+restconf}/data (データ API) : YANG データモデルで定義された設定データツリー全体の作成/取得/更新/削除 (CRUD) ベースの API。**
- **+restconf}/operations (操作 API) : YANG で定義された RPC をトリガーする RPC ベースの API エントリ。**
- **+restconf}/data/ietf-restconf-monitoring:restconf-state/streams (通知 API) : RESTCONF プロトコルで定義されている潜在的な通知ストリームを検出する RESTCONF API。詳細については、「[デバイス管理](#)」を参照してください。**
- **+restconf}/yang-library-version : この必須リーフは、このサーバーによって実装された「ietf-yang-library」 YANG モジュールの改訂日を識別する。**

RFC 8040 に従って、RESTCONF 実装は YANG データの JSON エンコーディングをサポートします [RFC 7951]。サーバーは「application/yang-data+json」を受け入れます。

クエリのフィルタ処理

RESTCONF 仕様によると、各操作には、0 個以上のクエリパラメーターをリクエスト URI に含めることができます。具体的には、クエリ操作のパラメータは [\[RFC 8040\] のセクション 4.8](#) で説明されています。次の表で、サポートされるクエリパラメータを示します。

表 1. サポートされるクエリパラメータ

名前	メソッド	説明
content	GET	構成および/または構成以外のデータリソースを選択する
depth	GET	応答コンテンツで制限されたサブツリーの深さをリクエストする
fields	GET	ターゲットのリソースコンテンツのサブセクションをリクエストする

転送のセキュリティ

データの完全性

Crosswork Hierarchical Controller は、TLS [RFC 5246] を介してすべての NBI アクセスデータを処理します。TLS は、データの完全性と機密性を提供するトランスポートレイヤーです。Crosswork Hierarchical Controller にアクセスするすべてのクライアントは、TLS 1.2 以上をサポートしている必要があります。

サーバー認証

Crosswork Hierarchical Controller NBI は、クライアントとの TLS 接続の確立時に X.509v3 証明書を提示します。

クライアントは、RESTCONF [RFC 8040 セクション 2] に従って、証明書の検証をサポートしている必要があります。RESTCONF クライアントは、(1) X.509 証明書パス検証 [RFC5280] を使用して RESTCONF サーバーの TLS 証明書の整合性を検証するか、(2) サーバーの TLS 証明書を信頼できるメカニズム（たとえば、ピン留めされた証明書）と一致させる必要があります。X.509 証明書パスの検証が失敗し、提示された X.509 証明書が信頼できるメカニズムによって取得された証明書と一致しない場合、[RFC5246] のセクション 7.2.1 の規定に従い接続を終了しなければなりません。

注：テストの目的で、クライアントの証明書の検証を無効にしてインターフェイスの機能をチェックできます。

クライアント認証

Crosswork Hierarchical Controller クライアント認証は、HTTP 基本認証 [RFC 7617] に基づいています。クライアントは、確立された TLS 接続を介してユーザー名とパスワードを提示する必要があります。より具体的には、クライアントは、基本認証スキームを使用する認証ヘッダーを使用する必要があります。詳細については、HTTP 認証スキームレジストリ [RFC 7235] を参照してください。

API エンドポイント

Crosswork Hierarchical Controller は、通知、情報、プロビジョニング サービスを受信するための API をサポートします。

- **デバイスマネージャ**：アダプタ、デバイス、クレデンシャルを管理できるさまざまな API です。
- **インベントリの取得**：光デバイスとルータ、それらのカード、ポート、属性を取得します。
- **パフォーマンスカウンタの取得**：特定のポートまたはリンクごとに L2 パフォーマンス（スループット）カウンタを取得します。

- **クロスリンク** : クロスリンクを管理するためのさまざまな API です。
- **共有リスク** : 共有リスクのポリシーとルールを管理できるさまざまな API です。
- **一般的なクエリ** : SHQL コマンドを使用して、Crosswork Hierarchical Controller モデルから任意のデータをクエリします。

オブジェクト ID

API での特定のオブジェクトの参照では、オブジェクト ID を使用します。ID は、ネットワーク、サイト、ノード、TP (ターミネーションポイント)、LSP、サービス、リンク、ネットワークアクセス、VLAN に使用できます。

Crosswork Hierarchical Controller がこれらの ID を実装したものが GUID です。GUID はモデル内のオブジェクトの一意的識別子であり、関連する GET API を使用してオブジェクト ID として取得できます。取得すると、その後はオブジェクト参照として使用できます。

デバイス管理

Crosswork Hierarchical Controller は、デバイス管理のための API を提供します。

Swagger を使用してデバイスマネージャ API にアクセスできます。

• https://<host>/api/v2/apps/device_manager/rest/doc
API には次のものが含まれています。

- すべてのアダプタの取得
- アダプタ構成スキーマの取得
- アダプタデバイススキーマの取得
- すべてのデバイスの取得
- Crosswork Hierarchical Controller へのデバイスの追加
- アダプタへのデバイスの追加
- デバイスのアダプタ構成の更新
- アダプタからのデバイス割り当て解除
- デバイス名の更新
- サイトへのデバイスの割り当て
- デバイスステータスの取得
- デバイスの取得
- Crosswork Hierarchical Controller からのデバイスの削除
- クレデンシャルキーの名前とタイプの取得
- 新しいクレデンシャルの追加
- クレデンシャルの更新
- クレデンシャルの削除
- 指定されたクレデンシャルタイプのパラメータスキーマの取得。

アダプタの入手

この API を使用して、すべてのアダプタのリストを取得します。応答の **guid** は、他のいくつかのデバイスマネージャメソッドで使用される **adapterGuid** です。

要求メソッド

GET

要求 URL

https://example-host/api/v2/apps/device_manager/rest/adapter

要求パラメータ

なし

応答の例

```
[
  {
    "guid": "cisco_ios_xr",
    "enabled": true,
    "config": {
      "polling": 300,
      "concurrency": 1,
      "ssh_config": {
        "enabled": false,
        "connect_timeout": 10,
        "command_timeout": 10
      },
      "file_bringing": {
        "enabled": false,
        "location": "",
        "file_type": "XX"
      },
      "collection_parameters": {
        "enable_igp_isis": true,
        "enable_stats": true,
        "enable_vrf": true,
        "enable_lldp": true,
        "enable_mpls": true,
        "enable_snmp": true,
        "igp_isis_priority": 1,
        "igp_seed_routers": true
      }
    }
  },
]
```

```
{
  "guid": "juniper_os_1",
  "enabled": true,
  "config": {
    "polling": 300,
    "concurrency": 1,
    "ssh_config": {
      "enabled": false,
      "connect_timeout": 10,
      "command_timeout": 10
    },
    "file_bringing": {
      "enabled": false,
      "location": "",
      "file_type": "XX"
    },
    "collection_parameters": {
      "enable_igp_isis": true,
      "enable_igp_ospf": true,
      "enable_stats": true,
      "enable_vrf": true,
      "enable_lldp": true,
      "enable_mpls": true,
      "enable_snmp": true,
      "igp_isis_priority": 1,
      "igp_seed_routers": true
    }
  }
}
```

アダプタ構成スキーマの取得

この API を使用して、アダプタの構成スキーマを取得します。

要求メソッド

GET

要求 URL (Request URL)

[https://example-host/api/v2/apps/device_manager/rest/ adapter/{adapterGuid}/adapter-schema](https://example-host/api/v2/apps/device_manager/rest/adapter/{adapterGuid}/adapter-schema)

要求パラメータ

パラメータ名	データタイプ	説明
adapterGuid	string	アダプタの GUID。アダプタの取得メソッドによって返される guid を使用します。

応答の例

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "polling": {
      "type": "integer",
      "minimum": 0,
      "default": 300,
      "title": "Polling Cycle [sec]",
      "description": "Poll every X seconds"
    },
    "concurrency": {
      "type": "integer",
      "default": 1,
      "title": "Number of concurrent routers collected"
    },
    "ssh_config": {
      "type": "object",
      "title": "SSH Configuration Parameters",
      "properties": {
        "enabled": {
          "type": "boolean",
          "default": false,
          "title": "Enable Tunnel"
        },
        "host": {
          "type": "string",
          "title": "Tunnel Host"
        },
        "port": {
          "type": "integer",
          "title": "Tunnel Port"
        },
        "tunnel_credentials_key": {
          "type": "string",
```

```
"title": "Tunnel Credentials Key",
"credentials-types": [
  "SSH_USER_PASSWORD"
]
},
"connect_timeout": {
  "type": "integer",
  "default": 10,
  "title": "Router Connect timeout"
},
"command_timeout": {
  "type": "integer",
  "default": 10,
  "title": "Router Command timeout"
}
},
"additionalProperties": false
},
"file_bringer": {
  "type": "object",
  "title": "File Bringer Parameters",
  "properties": {
    "enabled": {
      "type": "boolean",
      "default": false,
      "title": "Enable File Bringer"
    },
    "location": {
      "type": "string",
      "title": "File Server Location",
      "description": "\"[sftp|file]://<server>:<port>/<absolute path>\"",
      "default": ""
    },
    "file_type": {
      "title": "File Type",
      "enum": [
        "XX",
        "XX"
      ],
      "default": "XX"
    }
  },
  "authentication": {
```

```
        "type": "string",
        "credentials-types": [
            "SFTP"
        ]
    },
    "additionalProperties": false
},
"collection_parameters": {
    "type": "object",
    "title": "Collection Parameters",
    "properties": {
        "enable_igp_isis": {
            "type": "boolean",
            "default": false,
            "title": "Enable IGP IS-IS Collection"
        },
        "enable_stats": {
            "type": "boolean",
            "default": false,
            "title": "Enable Stats Collection"
        },
        "enable_vrf": {
            "type": "boolean",
            "default": false,
            "title": "Enable VRF Collection"
        },
        "enable_lldp": {
            "type": "boolean",
            "default": false,
            "title": "Enable LLDP Collection"
        },
        "enable_mpls": {
            "type": "boolean",
            "default": false,
            "title": "Enable MPLS Tunnels Collection"
        },
        "enable_snmp": {
            "type": "boolean",
            "default": false,
            "title": "Enable SNMP Collection"
        }
    }
},
```

```

    "igp_isis_priority": {
      "type": "integer",
      "default": 1,
      "title": "IGP IS-IS Priority"
    },
    "igp_seed_routers": {
      "type": "boolean",
      "default": true,
      "title": "Collect only IGP IS-IS seed routers"
    }
  },
  "additionalProperties": false
},
"additionalProperties": false,
"required": [
  "polling"
]
}

```

アダプタデバイススキーマの取得

この API を使用して、特定のアダプタのデバイス構成スキーマを取得します。

要求メソッド

GET

要求 URL (Request URL)

https://example-host/api/v2/apps/device_manager/rest/adapter/{adpaterGuid}/device-schema

要求パラメータ

パラメータ名	データタイプ	説明
adapterGuid	string	アダプタの GUID。アダプタの取得メソッドによって返される guid を使用します。

応答の例

```

{
  "type": "object",
  "properties": {
    "host": {
      "type": "string"
    },
    "port": {
      "type": "integer",

```

```

    "default": 22
  },
  "direct_connect": {
    "type": "boolean",
    "default": false,
    "title": "Direct Connect (avoid tunnel if configured)"
  },
  "authentication": {
    "type": "string",
    "credentials-types": [
      "SSH_USER_PASSWORD"
    ]
  },
  "enabled": {
    "type": "boolean",
    "default": false
  }
},
"additionalProperties": false,
"required": [
  "enabled",
  "host",
  "port"
]
}

```

すべてのデバイスの取得

この API を使用して、特定のアダプタのすべてのデバイスを取得します。応答で返される **device_manager_guid** は、関連するデバイスマネージャ API の **deviceGuid** として使用されます。

要求メソッド

GET

要求 URL (Request URL)

https://example-host/api/v2/apps/device_manager/rest/device?adapter={adpaterGuid}

要求パラメータ

パラメータ名	データタイプ	説明
adapterGuid	string	アダプタの GUID。アダプタの取得メソッドによって返される guid を使用します。

応答の例

```
[
  {
    "device_manager_guid": "0a9af2d2-324a-45dd-895e-9e6edb2f9d39",
    "inventory_type": "ROUTER",
    "device_model_guid": "IN/cisco_ios_xr/ROUTER-xrv-p1",
    "name": "cisco_1",
    "adapters": {
      "cisco_ios_xr": {
        "adapter_guid": "cisco_ios_xr",
        "config": {
          "port": 22,
          "direct_connect": true,
          "enabled": true,
          "host": "10.1.0.58",
          "authentication": "Cisco"
        },
        "enabled": false
      }
    },
    "identifiers": {
      "host_name": "10.1.0.58",
      "tid": null,
      "serial": null,
      "loopback_mac": null,
      "loopback_ip": null,
      "extra": {}
    },
    "site": null,
    "pending_delete": false,
    "enabled": false
  },
  {
    "device_manager_guid": "9988749383",
    "inventory_type": "ROUTER",
    "device_model_guid": "IN/cisco_ios_xr/ROUTER-xrv-p2",
    "name": "cisco_2",
    "adapters": {
      "cisco_ios_xr": {
        "adapter_guid": "cisco_ios_xr",
        "config": {
          "port": 22,
```



```
        "direct_connect": true,
        "enabled": true,
        "host": "10.1.0.71",
        "authentication": "Cisco"
    },
    "enabled": false
}
},
"identifiers": {
    "host_name": "10.1.0.71",
    "tid": null,
    "serial": null,
    "loopback_mac": null,
    "loopback_ip": null,
    "extra": {}
},
"site": null,
"pending_delete": false,
"enabled": false
}
]
```

Crosswork Hierarchical Controller へのデバイスの追加

この API を使用して、デバイスを Crosswork Hierarchical Controller に追加します。

要求メソッド

POST

要求 URL (Request URL)

https://example-host/api/v2/apps/device_manager/rest/device

要求パラメータ

なし

要求本文

パラメータ名	データタイプ	説明
説明	string	デバイス名。
site	string	サイトを指定するか、サイトがない場合は null を指定します。

リクエストボディの例

```
{
  "name": "router-a",
  "site": null
}
```

応答の例

```
{
  "device_manager_guid": "a2087abe-5753-4387-b372-f8cfb571bb1e"
}
```

デバイスをアダプタに割り当てる

この API を使用してデバイスをアダプタに割り当てます。

要求メソッド

POST

要求 URL (Request URL)

https://example-host/api/v2/apps/device_manager/rest/device/{deviceGuid}/adapters/{adapterGuid}

要求パラメータ

パラメータ名	データタイプ	説明
deviceGuid	string	デバイスの GUID。すべてのデバイスの取得メソッドによって返される device_manager_guid を使用します。
アダプタガイド	string	アダプタの GUID。アダプタの取得メソッドによって返される guid を使用します。

要求本文

パラメータ名	データタイプ	説明
説明	string	The adapter's device parameters. The schema can be retrieved with the Get Adapter Device Schema API. Ensure that you add the parameters that are marked as required. In this schema example: <pre>"required": ["enabled", "host", "port"]</pre>

リクエストボディの例

```
{
  "host": "1.1.1.1",
  "port": 22,
  "enabled": false
}
```

応答の例

201 Successful

デバイスのアダプタ構成の更新

この API を使用してデバイスアダプタの構成を更新します。

要求メソッド

PUT

要求 URL (Request URL)

https://example-host/api/v2/apps/device_manager/rest/device/{deviceGuid}/adapters/{adapterGuid}

要求パラメータ

パラメータ名	データタイプ	説明
deviceGuid	string	デバイスの GUID。すべてのデバイスの取得メソッドによって返される device_manager_guid を使用します。
adapterGuid	string	アダプタの GUID。アダプタの取得メソッドによって返される guid を使用します。

要求本文

パラメータ名	データタイプ	説明
説明	string	<p>The adapter's device parameters. The schema can be retrieved with the Get Adapter Device Schema API.</p> <p>Ensure that you add the parameters that are marked as required. In this schema example:</p> <pre>"required": ["enabled", "host", "port"]</pre>

リクエストボディの例

```
{
  "host": "1.1.1.1",
  "port": 22,
  "enabled": true
}
```

応答の例

200 Successful

アダプタからのデバイス割り当て解除

この API を使用して、アダプタからデバイスの割り当てを解除します。デバイスはネットワークから削除されません。

要求メソッド

DELETE

要求 URL (Request URL)

```
https://example-host/api/v2/apps/device_manager/rest/device/{deviceGuid}/adapters/{adapterGuid}
```

要求パラメータ

パラメータ名	データタイプ	説明
deviceGuid	string	デバイスの GUID。すべてのデバイスの取得メソッドによって返される device_manager_guid を使用します。
adapterGuid	string	アダプタの GUID。アダプタの取得メソッドによって返される guid を使用します。

応答の例

200 Successful

デバイス名の更新

この API を使用してデバイス名を更新します。

要求メソッド

PUT

要求 URL (Request URL)

```
https://example-host/api/v2/apps/device_manager/rest/device/{deviceGuid}/name/{name}
```

要求パラメータ

パラメータ名	データタイプ	説明
deviceGuid	string	デバイスの GUID。すべてのデバイスの取得メソッドによって返される device_manager_guid を使用します。
adapterGuid	string	アダプタの GUID。アダプタの取得メソッドによって返される guid を使用します。

応答の例

200 Successful

サイトへのデバイスの割り当て

この API を使用してデバイスをサイトに割り当てます。

要求メソッド

PUT

要求 URL (Request URL)

https://example-host/api/v2/apps/device_manager/rest/device/{deviceGuid}/site

要求パラメータ

パラメータ名	データタイプ	説明
deviceGuid	string	デバイスの GUID。すべてのデバイスの取得メソッドによって返される device_manager_guid を使用します。

要求本文

パラメータ名	データタイプ	説明
guid	string	サイトの GUID。

リクエストボディの例

```
{
  "guid": "ST/e7df76d7a9cb"
}
```

応答の例

200 Successful operation.

デバイスステータスの取得

この API を使用して、デバイスのステータスを取得します。

要求メソッド

GET

要求 URL (Request URL)

https://example-host/api/v2/apps/device_manager/rest/device/{deviceGuid}/status

要求パラメータ

パラメータ名	データタイプ	説明
deviceGuid	string	デバイスの GUID。すべてのデバイスの取得メソッドによって返される device_manager_guid を使用します。

応答の例

```
{
  "adapte-name": {
    "last-successfull-discovery": "2020-05-04T17:50:15.530Z"
  }
}
```

デバイスの取得

この API を使用してデバイスを取得します。

要求メソッド

GET

要求 URL (Request URL)

https://example-host/api/v2/apps/device_manager/rest/ device/{deviceGuid}

要求パラメータ

パラメータ名	データタイプ	説明
deviceGuid	string	デバイスの GUID。すべてのデバイスの取得メソッドによって返される device_manager_guid を使用します。

応答の例

```
{
  "device_manager_guid": "83ce57e9-a82b-488a-994b-f8c4191f0158",
  "inventory_type": null,
  "device_model_guid": null,
  "name": "router-a",
  "adapters": {
    "cisco_ios_xr": {
      "adapter_guid": "cisco_ios_xr",
      "config": {
        "host": "1.1.1.2",
        "port": 22,
        "enabled": false
      }
    }
  }
}
```

```
    },
    "enabled": false
  }
},
"identifiers": {
  "host_name": null,
  "tid": null,
  "serial": null,
  "loopback_mac": null,
  "loopback_ip": null,
  "extra": {}
},
"site": null,
"pending_delete": false,
"enabled": false
}
```

Crosswork Hierarchical Controller からのデバイスの削除

この API を使用して、Crosswork Hierarchical Controller からデバイスを削除します。

要求メソッド

DELETE

要求 URL (Request URL)

https://example-host/api/v2/apps/device_manager/rest/device/{deviceGuid}?force=true

要求パラメータ

パラメータ名	データタイプ	説明
deviceGuid	string	デバイスの GUID。すべてのデバイスの取得メソッドによって返される device_manager_guid を使用します。
force	string	デバイスがアダプタに割り当てられている場合でも、デバイスをシステムから削除するかどうか。

応答の例

200 OK!

クレデンシャルの取得

この API を使用して、構成されたすべてのクレデンシャルキーの名前とタイプを取得します。

要求メソッド

GET

要求 URL (Request URL)

`https://example-host/api/v2/apps/device_manager/rest/credentials?type={type}`

要求パラメータ

パラメータ名	データタイプ	説明
タイプ	array[string]	SSH_USER_PASSWORD、SSH_PUBLIC_KEY、HTTP、SNMP_COMMUNITY、SFTP

応答の例

```
[
  {
    "name": "A",
    "type": "SSH_USER_PASSWORD"
  },
  {
    "name": "Cisco",
    "type": "SSH_USER_PASSWORD"
  },
  {
    "name": "junos",
    "type": "SSH_USER_PASSWORD"
  }
]
```

新しいクレデンシャルの追加

この API を使用して、新しいクレデンシャルを追加します。

要求メソッド

POST

要求 URL (Request URL)

`https://example-host/api/v2/apps/device_manager/rest/credentials/{name}`

要求パラメータ

パラメータ名	データタイプ	説明
name	string	クレデンシャルの名前。

要求本文

パラメータ名	データタイプ	説明
タイプ	array[string]	SSH_USER_PASSWORD、SSH_PUBLIC_KEY、HTTP、SNMP_COMMUNITY、SFTP
config	説明	クレデンシャルの取得 API で説明されているクレデンシャルパラメータの構造。

リクエストボディの例

```
{
  "type": "SSH_USER_PASSWORD",
  "config": {}
}
```

応答の例

201 Successful Operation

クレデンシャルの更新

この API を使用してクレデンシャルを更新します。

要求メソッド

PUT

要求 URL (Request URL)

https://example-host/api/v2/apps/device_manager/rest/credentials/{name}

要求パラメータ

パラメータ名	データタイプ	説明
name	string	クレデンシャルの名前。

要求本文

パラメータ名	データタイプ	説明
タイプ	array[string]	SSH_USER_PASSWORD、SSH_PUBLIC_KEY、HTTP、SNMP_COMMUNITY、SFTP
config	説明	クレデンシャルの取得 API で説明されているクレデンシャルパラメータの構造。

リクエストボディの例

```
{
  "type": "SSH_USER_PASSWORD",
  "config": {}
}
```

応答の例

200 Successful Operation

クレデンシャルの削除

この API を使用してクレデンシャルを削除します。

要求メソッド

DELETE

要求 URL (Request URL)

https://example-host/api/v2/apps/device_manager/rest/credentials/{name}

要求パラメータ

パラメータ名	データタイプ	説明
name	string	クレデンシャルの名前。

応答の例

200 Successful

指定されたクレデンシャルタイプのパラメータスキーマの取得。

この API を使用して、指定されたクレデンシャルタイプのパラメータスキーマを取得します。

要求メソッド

GET

要求 URL (Request URL)

https://example-host/api/v2/apps/device_manager/rest/XXX

要求パラメータ

パラメータ名	データタイプ	説明
タイプ	array[string]	SSH_USER_PASSWORD、SSH_PUBLIC_KEY、HTTP、SNMP_COMMUNITY、SFTP

応答の例

```
{
  "type": "object",
  "properties": {
    "username": {
      "title": "Username",
      "type": "string"
    },
    "password": {
      "title": "Password",
```

```
    "type": "string",
    "encrypted": true
  }
},
"required": [
  "username",
  "password"
]
}
```

物理インベントリ

物理インベントリ API は、デバイスの属性、スロット、カード、ポートに関する情報を提供します。

この API は、光デバイスのみをサポートします。

物理インベントリの取得

API は次の 2 つのクエリを受け入れます。

- デバイス ID - デバイスの完全なリストまたは ID ごとの特定のデバイスを取得します。
- Full - 詳細なデバイスインベントリまたはその ID のみを取得します (full=true/false)。

要求メソッド

GET

要求 URL (Request URL)

/api/v2/apps/network-inventory-app/rest/devices/

要求ヘッダー

Accept: application/yang-data+json

Content-Type: application/yang-data+json

応答の例

```
{
  "id": "XX-01011",
  "type": "ONE",
  "desc": "XX-01011",
  "vendor_model": "ciena_6500",
  "tid": "TID111",
  "serial_number": "[Shelf-1 [1-XX,2-XX]]",
  "SHELF": [
    {
      "id": "1",
      "type": "SHELF",
      "desc": "6500 14-Slot Converged Optical Shelf Assembly, OCP",
      "vendor_model": "ciena_6500",
```

```

"tid": "XX-01011",
"aid": "SHELF-1",
"clei": "WWW",
"serial_number": "SSS",
"FAN": [
  {
    "id": "FAN-1-1",
    "type": "FAN",
    "desc": "XX-01011/MNG-EQPT/FAN-1-1 (FAN), ctype=\"Fan Rear Exhaust High
Flow Cooling\"",
    "vendor_model": "ciena_6500",
    "tid": "XX-01011",
    "aid": "FAN-1-1",
    "clei": "WWW",
    "part_number": "NNN",
    "serial_number": "MMM"
  }
],
"CARD": [
  {
    "id": "1-1",
    "type": "CARD",
    "desc": "XX-01011/1-1 (OTSC), ctype=\"2x10G OTR 4x XFP\"",
    "vendor_model": "ciena_6500",
    "tid": "XX-01011",
    "aid": "000-1-1",
    "clei": "WWW",
    "part_number": "MM",
    "serial_number": "NNN",
    "PORT": [
      {
        "id": "1-1-1",
        "type": "PORT",
        "desc": "XX-01011/1-1-1 (P10GSOEL), ctype=\"OC192 SR1/I64.1
10GBASE-LR/LW OTU2 10GFC 1200SMLLL 1310 nm XFP\"",
        "vendor_model": "ciena_6500",
        "tid": "XX-01011",
        "aid": "UU-1-1-1",
        "clei": "WWW",
        "pluggable": "PLUGGABLE",
        "part_number": "NNNA",
        "serial_number": "MMM"
      }
    ]
  }
]

```

```

    },
  },
  "POWER_SUPPLY": [
    {
      "id": "PWR-1-17-1",
      "type": "POWER_SUPPLY",
      "desc": "XX-01011/MNG-EQPT/PWR-1-17-1 (PWR), ctype=\"Power Card 60A
breakered\"),",
      "vendor_model": "ciena_6500",
      "tid": "XX-01011",
      "aid": "PWR-1-17-1",
      "clei": "WWWD",
      "part_number": "NNNA",
      "serial_number": "MMM"
    },
  ]
}
]
}
}

```

応答パラメータ

応答には、すべてのデバイス インベントリ モジュールに関する情報が含まれています。

- デバイス - すべてのデバイスレベルの属性
- シェルフ - 1 つ以上のシェルフ
- ファン - ファントレイ
- カード - トランシーバーを含む、すべての着脱可能なカード、ラインカード、アグリゲーションカード
- ポート - すべての物理ポート
- 電源 - すべての電源モジュール

パフォーマンス カウンタ

Crosswork Hierarchical Controller パフォーマンス API は、シスコが作成した非 SDO データモデルに基づいています。SDO には、管理対象ネットワークオブジェクトの実パフォーマンスカウンタへの NBI アクセスを可能にするネットワークレベルのパフォーマンスデータモデルがありません。

IETF の現在のパフォーマンス カウンタ データ モデルは、ietf-interfaces などのデバイスレベルに対応していますが、ネットワーク レベル データ モデルは、帯域幅などの取得値に焦点を当てています。シスコは、IETF から利用可能になった時点で標準データモデルを実装します。

Crosswork Hierarchical Controller データモデルは、ポートおよび IP/MPLS LSP から収集された実パフォーマンスカウンタを公開します。これは YANG RPC の形式で実装され、NBI クライアントが RESTCONF NBI を介して操作をトリガーして、レートや平均などを取得するためのパケットおよび/またはオクテットカウンタを取得できます。

URI

最新の PM 間隔または特定の期間のカウンタを取得できます。統計は、特定の LSP/ポート/リンク、すべてのオブジェクト、またはオブジェクトのグループ (POST メッセージで設定) に対して取得できます。

インターフェイスタイプ	インターフェイス操作	要求メソッド	URI
パフォーマンス	ある期間のオブジェクトタイプごとの実パフォーマンスカウンタを取得する RPC。	GET	/api/v2/timeseries/stats/
パフォーマンス	最新の期間のオブジェクトタイプごとの実パフォーマンスカウンタを取得する RPC。	GET	/api/v2/timeseries/stats/latest
パフォーマンス	特定のポート (1 つのポート、またはすべてのポートではない) の実パフォーマンスカウンタを取得する RPC	PUT	/api/v2/timeseries/stats/bulkStatsQuery

パフォーマンスの最新統計の取得

1 つのオブジェクトまたはすべてのオブジェクトの最新の期間のパフォーマンス統計を取得します。特定のポート/LSP またはストリーム ID を使用して、特定の PM タイプを指定できます。

要求メソッド

GET

要求 URL (Request URL)

api/v2/timeseries/stats/latest

要求ヘッダー

Accept: application yang-data+json

Content-Type: application yang-data+json

要求パラメータ

パラメータ名	データタイプ	値	説明
objGuid	string		オブジェクトの GUID。
pmType	array[string]	DELAY_USEC、MIN_RTT_USEC、OCTET_OUT	PM タイプ。
streamId	numeric		ストリーム ID。

要求の例

```
GET /api/v2/timeseries/stats/latest?pmType=DELAY_USEC&objGuid=PO/juniper-northstar/LGC-LabMX960-02:ae5.0
```

```
GET /api/v2/timeseries/stats/latest?pmType=DELAY_USEC&streamId=3
```

要求パラメータ

パラメータ名	データタイプ	値	説明
objGuid	string		オブジェクトの GUID。
timestamp	datetime		時間。
deviceType	string		デバイスタイプ (例 : ROUTER) 。
timePeriodSec	numeric		秒単位の時間
pmType	array[string]	DELAY_USEC、MIN_RTT_USEC、OCTET_OUT	PM タイプ。
value	numeric		パフォーマンス統計情報。
streamId	numeric		ストリーム ID。

応答の例

```
{
  "objGuid":
  "LI/lsp/f66fa3288d396e47/f66fa3288d396e47/f1e815107b715b67/f1e815107b715b67/lsp_1619081766048",
  "timestamp": 1622602613000,
  "deviceType": "ROUTER",
  "timePeriodSec": 0,
  "pmType": "OCTET_OUT",
  "value": 273252081213835,
  "streamId": 6
}
```

期間中のパフォーマンス統計の取得

1 つのオブジェクトまたはすべてのオブジェクトの特定の期間のパフォーマンス統計を取得します。特定のポート/LSP またはストリーム ID を使用して、特定の PM タイプを指定できます。

要求メソッド

GET

要求 URL (Request URL)

api/v2/timeseries/stats

要求ヘッダー

Accept: application/yang-data+json

Content-Type: application/yang-data+json

要求パラメータ

パラメータ名	データタイプ	値	説明
startTimeStamp	datetime		期間の開始時間。
endTimeStamp	datetime		期間の終了時間。
objGuid	string		オブジェクトの GUID。
pmType	array[string]	DELAY_USEC、MIN_RTT_USEC、OCTET_OUT	PM タイプ。
streamId	numeric		ストリーム ID。

要求の例

```
GET /api/v2/timeseries/stats?startTimeStamp=1643793190000&
endTimeStamp=1643878292000&pmTypeDELAY_USEC&objGuid=
PO/r_logical/16f1596a46b13da5/7d82f458ff24bfa3
```

要求パラメータ

パラメータ名	データタイプ	値	説明
objGuid	string		オブジェクトの GUID。
Timestamp	datetime		時間。
deviceType	string		デバイス タイプ (例: ROUTER)。
timePeriodSec	numeric		秒単位の時間
pmType	array[string]	DELAY_USEC、MIN_RTT_USEC、OCTET_OUT	PM タイプ。
値	numeric		パフォーマンス統計情報。
streamId	numeric		ストリーム ID。

応答の例

```
{
  "objGuid": "PO/r_logical/16f1596a46b13da5/7d82f458ff24bfa3",
  "timeStamp": 1643874592000,
  "deviceType": "ROUTER",
  "timePeriodSec": 859,
  "pmType": "DELAY_USEC",
  "value": 649907,
```



```
"streamId": 5
}
{
  "objGuid": "PO/r_logical/16f1596a46b13da5/7d82f458ff24bfa3",
  "timeStamp": 1643875520000,
  "deviceType": "ROUTER",
  "timePeriodSec": 928,
  "pmType": "DELAY_USEC",
  "value": 724091,
  "streamId": 5
}
{
  "objGuid": "PO/r_logical/16f1596a46b13da5/7d82f458ff24bfa3",
  "timeStamp": 1643876484000,
  "deviceType": "ROUTER",
  "timePeriodSec": 927,
  "pmType": "DELAY_USEC",
  "value": 69079,
  "streamId": 5
}
{
  "objGuid": "PO/r_logical/16f1596a46b13da5/7d82f458ff24bfa3",
  "timeStamp": 1643877388000,
  "deviceType": "ROUTER",
  "timePeriodSec": 904,
  "pmType": "DELAY_USEC",
  "value": 665969,
  "streamId": 5
}
```

特定のポートのパフォーマンス統計の送信

特定のポート（1つのポート、またはすべてのポートではない）の特定の期間のパフォーマンス統計を取得します。

要求メソッド

PUT

要求 URL (Request URL)

api/v2/timeseries/stats/bulkStatsQuery

要求ヘッダー

Accept: application/yang-data+json

Content-Type: application/yang-data+json

要求パラメータ

パラメータ名	データタイプ	説明
startTimeStamp	datetime	期間の開始時間。
endTimeStamp	datetime	期間の終了時間。
d@ports.json	JSON 文字列	ここで、ports.json は GUID のリストです。

要求の例

```
POST
/api/v2/timeseries/stats/bulkStatsQuery?startTimeStamp=1659402468000&endTimeStamp=1659802468000' -d@ports.json
```

request.json は GUID のリストです。

```
[
  "PO/r_logical/1d80163d31fa18ee/870a15983eafb41f",
  "PO/r_logical/7b4564a1ccb6e4cb/5a52cb94b8d7dbc4"
]
```

応答パラメータ

パラメータ名	データタイプ	値	説明
objGuid	string		オブジェクトの GUID。
timeStamp	datetime		時間。
deviceType	string		デバイス タイプ (例: ROUTER)。
timePeriodSec	numeric		秒単位の時間
pmType	array[string]	DELAY_USEC、MIN_RTT_USEC、OCTET_OUT	PM タイプ。
value	numeric		パフォーマンス統計情報。
streamId	numeric		ストリーム ID。

応答の例

```
[
  {
    "objGuid": "PO/r_logical/7b4564a1ccb6e4cb/5a52cb94b8d7dbc4",
    "timeStamp": 1659608730000,
    "deviceType": "ROUTER",
    "timePeriodSec": 900,
    "pmType": "MAX_RTT_USEC",
    "value": 481463,
  }
]
```

```
"streamId": 4
},
{
  "objGuid": "PO/r_logical/7b4564a1ccb6e4cb/5a52cb94b8d7dbc4",
  "timeStamp": 1659608730000,
  "deviceType": "ROUTER",
  "timePeriodSec": 900,
  "pmType": "DELAY_USEC",
  "value": 443345,
  "streamId": 4
},
{
  "objGuid": "PO/r_logical/7b4564a1ccb6e4cb/5a52cb94b8d7dbc4",
  "timeStamp": 1659608730000,
  "deviceType": "ROUTER",
  "timePeriodSec": 900,
  "pmType": "MIN_RTT_USEC",
  "value": 441197,
  "streamId": 4
},
{
  "objGuid": "PO/r_logical/1d80163d31fa18ee/870a15983eafb41f",
  "timeStamp": 1659608735000,
  "deviceType": "ROUTER",
  "timePeriodSec": 900,
  "pmType": "OCTET_OUT",
  "value": 17939855810587,
  "streamId": 4
}
}
```

クロスリンク

Crosswork Hierarchical Controller は、クロスリンクを管理するための API を提供します。詳細については、『*Cisco Crosswork Hierarchical Controller Administration Guide*』を参照してください。

API には次のものが含まれています。

- 隣接するすべてのクロスリンクの取得
- すべてのクロスリンクの検証
- 1つのクロスリンクの検証
- すべての手動クロスリンクの取得
- すべての手動クロスリンクの取得
- クロスリンクの取得

- Crosswork Hierarchical Controller への手動クロスリンクの追加
- クロスリンクの削除

注：ユーザーに API を実行する権限がない場合、403 Forbidden が返されます。間違ったクレデンシャルが使用されると、401 Authorization Required が返されます。

隣接するすべてのクロスリンクの取得

この API を使用して、LLDP などの自動検出プロトコルを使用して検出されたポートの隣接関係（アジャセンシー）を返します。このリストの要素は、構成されておらず、手動クロスリンクの一部ではない隣接関係である場合があります。

要求メソッド

GET

要求 URL (Request URL)

`https://example-host/api/v2/crosslinks/adjacencies`

要求パラメータ

なし

応答の例

応答には、隣接するすべてのクロスリンクに関する情報が含まれます。クロスリンクに関する次のものが返されます。

- interfaceA - deviceName、interfaceName、type、guid
- interfaceB - deviceName、interfaceName、type、guid
- technology (ETH または NMC)
- relatedModelObjectGuid

すべてのクロスリンクの検証

この API を使用して、すべてのクロスリンクの検証をトリガーします。これにより、クロスリンク、ルータ物理リンク、eth リンクなどの最新のステータスに基づいてモデル（リンクとパス）を更新するプロセスが実行されます。すべてのクロスリンクを検証（および新しい手動クロスリンクを追加）すると、r-phy リンクのパスが更新される場合があります、まだ存在しない場合は r-phy リンクが一から作成される可能性もあります。新しい手動クロスリンクの結果として、光回線全体がエンドツーエンドで検出される可能性があります。

イーサネットリンクの場合、手動で追加されたクロスリンクとネットワークから検出されたクロスリンクの間に競合がある場合、手動で追加されたリンクは Cisco Crosswork Hierarchical Controller ネットワークモデルから削除されます。手動リンクは手動リンクデータベースに残り、ネットワークで検出されたリンクが消え、競合がなくなった後に手動リンクが肯定的な検証を得た場合に、ネットワークモデルに再度追加されます（これは、肯定的な検証がなくてもネットワークモデルに追加される競合のない手動リンクとは対照的です）。

この API メソッドは空の結果を返します。[すべての手動クロスリンクの取得](#)メソッドを使用して、最新のステータスを表示できます。

要求メソッド

GET

要求 URL (Request URL)

https://example-host/api/v2/crosslinks/validate

要求パラメータ

なし

応答の例

なし

1つのクロスリンクの検証

この API を使用して、特定のクロスリンクの検証をトリガーします。この API は、トポロジモデルに触れることなく、検証済みの手動クロスリンクの関連する検証フィールドのみを更新します。

要求メソッド

GET

要求 URL (Request URL)

https://example-host/api/v2/crosslinks/validate/{guid}

応答パラメータ

パラメータ名	データタイプ	説明
guid	string	クロスリンクの GUID。すべての手動クロスリンクの取得メソッドによって返される guid を使用します。

応答の例

応答は、クロスリンクに関する次のものを返します。

- guid : リンクの GUID
- interfaceA - deviceName、interfaceName、type、guid
- interfaceB - deviceName、interfaceName、type、guid
- technology (ETH または NMC)
- relatedModelObjectGuid
- additionTimeEpochMillis
- 説明
- validated : リンクが検証されているかどうか。
- validationResult : 検証結果 (VALIDATED_BY_PORT_ADJACENCY、VALIDATED_BY_CORRELATION、NOT_CORRELATED、INCONCLUSIVE、UNKNOWN、CONFLICT) 。
- lastValidationCheck : リンクが最後に検証されたときのタイムスタンプ。
- lastConflictTime : 競合が発生しなかった場合、このタイムスタンプは非表示になります。これが **lastValidationCheck** と同じ場合、手動リンクは現在競合しています。 **lastValidationCheck** が

lastConflictTime より後の場合、手動リンクは以前は競合していましたが、現在は競合していません。さらに **validated** が **true** の場合、この手動リンクがネットワークモデルに追加されます。

- **conflicts** : リンクが競合している、または過去に競合していた場合、**conflicts** リストには 1 つまたは 2 つの競合するペアが表示されます (手動リンクポートごとに 1 ペア) 。

```
{
  "guid": "23f6e6c340268f068f817f8f9484aa54a2ba92d5d7072efd7940308f8ecc8fa5",
  "interfaceA": {
    "deviceName": "CR2.ATH",
    "interfaceName": "TenGigE0/0/1/13",
    "type": "R_PHYSICAL",
    "guid": "PO/r_physical/8ff611ed7ebcd772/2c4e6530608a213d"
  },
  "interfaceB": {
    "deviceName": "SD2ATH02",
    "interfaceName": "ETH-1-1-8",
    "type": "ETH",
    "guid": "PO/eth/ecf4e9a6bbb76cbc/86cdb5c6379e5c62"
  },
  "technology": "ETH",
  "relatedModelObjectGuid":
  "LI/eth/8ff611ed7ebcd772/2c4e6530608a213d/ecf4e9a6bbb76cbc/86cdb5c6379e5c62",
  "additionTimeEpochMillis": 1666695054475,
  "description": "cross link description",
  "validated": true,
  "validationResult": "VALIDATED_BY_PORT_ADJACENCY",
  "lastValidationCheck": 1668598747209,
  "conflicts": [
    {
      "a": { "deviceName": "dev-name-of-manual-link-interfaceA", "interfaceName": "iface-
name-of-manual-link-interfaceA" },
      "b": { "deviceName": "dev-name-of-different-than-manual-link-interfaceB",
"interfaceName": "iface-name-different-than-manual-link-interfaceB" }
    },
    {
      "a": { "deviceName": "dev-name-of-manual-link-interfaceB", "interfaceName": "iface-
name-of-manual-link-interfaceB" },
      "b": { "deviceName": "dev-name-of-different-than-manual-link-interfaceA",
"interfaceName": "iface-name-different-than-manual-link-interfaceA" }
    }
  ]
}
```

すべての手動クロスリンクの取得

この API を使用して、一連の手動クロスリンクを取得します。返された guid は、他のクロスリンク API の要求パラメータとして使用できます。

要求メソッド

GET

要求 URL (Request URL)

`https://example-host/api/v2/crosslinks/manual`

要求パラメータ

なし

応答の例

応答は、以下の情報と合わせて一連のクロスリンクを返します。

- guid
- interfaceA - deviceName、interfaceName、type、guid
- interfaceB - deviceName、interfaceName、type、guid
- technology (ETH または NMC)
- relatedModelObjectGuid
- additionTimeEpochMillis
- 説明
- validated
- validationResult (VALIDATED_BY_PORT_ADJACENCY、VALIDATED_BY_CORRELATION、NOT_CORRELATED、INCONCLUSIVE、UNKNOWN、CONFLICT) 。
- lastValidationCheck
- conflicts

```
{
  "guid": "23f6e6c340268f068f817f8f9484aa54a2ba92d5d7072efd7940308f8ecc8fa5",
  "interfaceA": {
    "deviceName": "CR2.ATH",
    "interfaceName": "TenGigE0/0/1/13",
    "type": "R_PHYSICAL",
    "guid": "PO/r_physical/8ff611ed7ebcd772/2c4e6530608a213d"
  },
  "interfaceB": {
    "deviceName": "SD2ATH02",
    "interfaceName": "ETH-1-1-8",
    "type": "ETH",
    "guid": "PO/eth/ecf4e9a6bbb76cbc/86cdb5c6379e5c62"
  }
}
```

```
},
"technology": "ETH",
"relatedModelObjectGuid":
"LI/eth/8ff611ed7ebcd772/2c4e6530608a213d/ecf4e9a6bbb76cbc/86cdb5c6379e5c62",
"additionTimeEpochMillis": 1666695054475,
"description": "cross link description",
"validated": true,
"validationResult": "VALIDATED_BY_PORT_ADJACENCY",
"lastValidationCheck": 1668598747209,
"conflicts": []
}
```

クロスリンクの取得

この API を使用して手動クロスリンクを取得します。

要求メソッド

GET

要求 URL (Request URL)

`https://example-host/api/v2/crosslinks/manual/{guid}`

要求パラメータ

パラメータ名	データタイプ	説明
guid	string	クロスリンクの GUID。すべての手動クロスリンクの取得メソッドによって返される guid を使用します。

応答の例

応答は、クロスリンクに対して次を返します。

- guid
- interfaceA - deviceName、interfaceName、type、guid
- interfaceB - deviceName、interfaceName、type、guid
- technology (ETH または NMC)
- relatedModelObjectGuid
- additionTimeEpochMillis
- 説明
- validated : リンクが検証されているかどうか。
- validationResult : 検証結果 (VALIDATED_BY_PORT_ADJACENCY、VALIDATED_BY_CORRELATION、NOT_CORRELATED、INCONCLUSIVE、UNKNOWN、CONFLICT) 。
- lastValidationCheck : リンクが最後に検証されたときのタイムスタンプ。

- **lastConflictTime** : 競合が発生しなかった場合、このタイムスタンプは非表示になります。これが **lastValidationCheck** と同じ場合、手動リンクは現在競合しています。 **lastValidationCheck** が **lastConflictTime** より後の場合、手動リンクは以前は競合していましたが、現在は競合していません。さらに **validated** が **true** の場合、この手動リンクがネットワークモデルに追加されます。
- **conflicts** : リンクが競合している、または過去に競合していた場合、 **conflicts** リストには 1 つまたは 2 つの競合するペアが表示されます (手動リンクポートごとに 1 ペア) 。

```
{
  "guid": "fa6262a2f1202b4bba1dd699cc78b2cf9cc45c826487de6daad69e3fa0cf0a90",
  "interfaceA": {
    "deviceName": "CR2.BCN",
    "interfaceName": "TenGigE0/0/1/8",
    "type": "R_PHYSICAL",
    "guid": "PO/r_physical/b876eefb0f288974/146956e90f8b5b6d"
  },
  "interfaceB": {
    "deviceName": "OTN1MIL01",
    "interfaceName": "1-2-3",
    "type": "ETH",
    "guid": "PO/eth/5979a210307b1e66/fba4016fb0ebde72"
  },
  "technology": "ETH",
  "relatedModelObjectGuid":
  "LI/CL/PO/r_physical/b876eefb0f288974/146956e90f8b5b6d/PO/eth/5979a210307b1e66/fba4016fb0ebde72",
  "additionTimeEpochMillis": 1668597718579,
  "description": "example",
  "validated": false,
  "validationResult": "UNKNOWN",
  "lastValidationCheck": 1668597744370,
  "conflicts": []
}
```

Crosswork Hierarchical Controller への手動クロスリンクの追加

この API を使用して、Crosswork Hierarchical Controller に手動クロスリンクを追加します。

要求メソッド

POST

要求 URL (Request URL)

<https://example-host/api/v2/apps/crosslinks/manual>

要求パラメータ

なし

要求本文

パラメータ名	データタイプ	説明
interfaceA		ポート A
deviceName	string	デバイス名。
interfaceName	string	インターフェイス名。
interfaceB		ポート B
deviceName	string	デバイス名。
interfaceName	string	インターフェイス名。
テクノロジー	string	テクノロジー。
説明	string	説明。

リクエストボディの例

```
{
  "interfaceA": {
    "deviceName": "CR2.BCN",
    "interfaceName": "TenGigE0/0/1/8"
  },
  "interfaceB": {
    "deviceName": "OTN1MIL01",
    "interfaceName": "1-2-3"
  },
  "technology": "ETH",
  "description": "example"
}
```

応答の例

```
{"guid": "d33ac2ec12c237e3a53bef30aec690e8f1ecff1a9c600c98b406ea9be30e91b7"}
```

クロスリンクの削除

この API を使用して手動クロスリンクを削除します。クロスリンクは削除済みとしてマークされ、次の検証の実行時に削除されます。

要求メソッド

DELETE

要求 URL (Request URL)

`https://example-host/api/v2/crosslinks/manual/{guid}`

要求パラメータ

パラメータ名	データタイプ	説明
guid	string	クロスリンクの GUID。すべての手動クロスリンクの取得メソッドによって返される guid を使用します。

応答の例

200 OK

共有リスク

Crosswork Hierarchical Controller は、共有リスクポリシーとルールを管理するための API を提供します。

Swagger を使用して Shared Risk API にアクセスできます : <https://<host> /api/v2/apps/srlg-app/rest/doc>

API には次のものが含まれています。

- 特定のポリシーの取得
- すべてのポリシーの取得
- ポリシーの作成
- ポリシーの削除
- ポリシーの共有リスクタイプの変更
- ポリシーの種類の変更
- ポリシーへの新しいルールの追加
- ルールリソースの更新
- ポリシーからのルールの削除

ポリシーの取得

この API を使用して、すべてのポリシーのリストを取得します。これにより、すべてのポリシーとそのルールのリストが返されます。

要求メソッド

GET

要求 URL (Request URL)

<https://example-host/api/v2/apps/srlg-app/rest/policy>

要求パラメータ

なし

応答の例

```
{
  "name": "policy-1",
  "shared_risk_types": [
```

```

    "Link",
    "Port",
    "Card",
    "Shelf",
    "Device"
  ],
  "policy_type": "MULTIPLE-LINKS",
  "rules": [
    {
      "name": "rule-1",
      "resources": [
        "LI/eth/000fc44c94a1f2cd/51308dfd752c1574/df753d953c1e1c8f/f8e7b20537ce03b7"
      ]
    },
    {
      "name": "rule99",
      "resources": [
        "inventory[.name=\"CR1.PAR\"]|port|link[.layer=\"R_LOGICAL\"]"
      ]
    }
  ]
},
{
  "name": "test",
  "shared_risk_types": [
    "Link",
    "Device",
    "Shelf",
    "Port",
    "Card"
  ],
  "policy_type": "MULTIPLE-LINKS",
  "rules": [
    {
      "name": "rule001",
      "resources": [
        "inventory[.name=\"ILA-SD1EVO01-SD1SEV01-1\"]|port|link[.layer=\"R_LOGICAL\"]"
      ]
    }
  ]
},
{

```

```

    "name": "policy-3",
    "shared_risk_types": [
      "Link"
    ],
    "policy_type": "SINGLE-PROTECTED",
    "rules": [
      {
        "name": "rule-99",
        "resources": [
          "link[.layer=\"R_LOGICAL\"]"
        ]
      }
    ]
  }
}

```

ポリシーの取得

この API を使用してポリシーを取得します。

要求メソッド

GET

要求 URL (Request URL)

`https:// example-host/api/v2/apps/srlg-app/rest/policy/{policyGuid}`

要求パラメータ

パラメータ名	データタイプ	説明
policyGuid	string	ポリシーの GUID。Get Policies メソッドによって返された guid を使用します。

応答の例

```

{
  "name": "policy-1",
  "shared_risk_types": [
    "Link",
    "Port",
    "Card",
    "Shelf",
    "Device"
  ],
  "policy_type": "MULTIPLE-LINKS",
  "rules": [
    {
      "name": "rule-1",
      "resources": [

```

```

        "LI/eth/000fc44c94a1f2cd/51308dfd752c1574/df753d953c1e1c8f/f8e7b20537ce03b7"
    ]
},
{
    "name": "rule99",
    "resources": [
        "inventory[.name=\"CR1.PAR\"]|port|link[.layer=\"R_LOGICAL\"]"
    ]
}
]
}

```

ポリシーの作成

この API を使用してポリシーを作成します。

要求メソッド

POST

要求 URL (Request URL)

https://example-host/api/v2/apps/srlg-app/rest/policy/{policyGuid}

要求パラメータ

パラメータ名	データタイプ	説明
policyGuid	string	ポリシーの GUID。Get Policies メソッドによって返された guid を使用します。

要求本文

パラメータ名	データタイプ	説明
shared_risk_types	string	リンク、ポート、カード、シェルフ、デバイス
policy_type	string	SINGLE-PROTECTED または MULTIPLE-LINKS。

リクエストボディの例

```

{
    "shared_risk_types": [
        "Link"
    ],
    "policy_type": "SINGLE-PROTECTED"
}

```

応答の例

201 Successful Operation

ポリシーの削除

この API を使用してポリシーを削除します。

要求メソッド

DELETE

要求 URL (Request URL)

https://example-host/api/v2/apps/srlg-app/rest/policy/{policyGuid}

要求パラメータ

パラメータ名	データタイプ	説明
policyGuid	string	ポリシーの GUID。Get Policies メソッドによって返された guid を使用します。

応答の例

200 Successful

ポリシーの共有リスクタイプの更新

この API を使用して、ポリシーの共有リスクタイプを変更します。

要求メソッド

PUT

要求 URL (Request URL)

https://example-host/api/v2/apps/srlg-app/rest/policy/{policyGuid}/shared_risk_types

要求パラメータ

パラメータ名	データタイプ	説明
policyGuid	string	ポリシーの GUID。Get Policies メソッドによって返された guid を使用します。

要求本文

パラメータ名	データタイプ	説明
shared_risk_types	string	リンク、ポート、カード、シェルフ、デバイス

リクエストボディの例

```
{
  "shared_risk_types": [
    "Link"
  ]
}
```

応答の例

200 Successful Operation

ポリシータイプの更新

この API を使用してクレデンシャルを更新します。

要求メソッド

PUT

要求 URL (Request URL)

https://example-host/api/v2/apps/srlg-app/rest/policy/{policyGuid}/policy-type

要求パラメータ

パラメータ名	データタイプ	説明
policyGuid	string	ポリシーの GUID。Get Policies メソッドによって返された guid を使用します。

要求本文

パラメータ名	データタイプ	説明
policy_type	string	SINGLE-PROTECTED または MULTIPLE-LINKS。

リクエストボディの例

```
{
  "policy_type": "SINGLE-PROTECTED"
}
```

応答の例

```
200 Successful Operation
```

ポリシーへのルールの追加

この API を使用して、ルールをポリシーに追加します。GUID の配列や SHQL クエリを使用してルールを作成できます。

要求メソッド

POST

要求 URL (Request URL)

https://example-host/api/v2/apps/srlg-app/rest/policy/{policyGuid}/rules{ruleName}

要求パラメータ

パラメータ名	データタイプ	説明
policyGuid	string	ポリシーの GUID。Get Policies メソッドによって返された guid を使用します。
ruleName	string	規則名。Get Policies メソッドによって返されるルール名の 1 つを使用します。

要求本文

パラメータ名	データタイプ	説明
resources	array(string)	<p>GUID リンクおよび SHQL クエリ両方を含むまたはいずれか一方を含むリスト。</p> <p>SHQL クエリを使用する場合は、式が有効で結果を返すことを確認してください。SHQL ユーザーガイドを参照してください。</p> <p>SHQL クエリを渡すときは、必ず "..." と \s の組み合わせで囲んでください (例: "link[.layer=\\"R_LOGICAL\\"]")。</p>

リクエストボディの例

```
{
  "resources": [
    "link[.layer=\\"R_LOGICAL\\"]"
  ]
}
```

または

```
{
  "resources": [
    "LI/guid1",
    "LI/guid2"
  ]
}
```

または

```
{
  "resources": [
    "inventory[.name=\\"CR1.PAR\\"]|port|link[.layer=\\"R_LOGICAL\\"]"
  ]
}
```

応答の例

```
201 Successful Operation
```

ルールの更新

この API を使用して、ルールのリソースを更新します。GUID の配列や SHQL クエリを使用してルールを作成できます。

要求メソッド

```
PUT
```

要求 URL (Request URL)

```
https://example-host/api/v2/apps/srlg-app/rest/policy/{policyGuid}/rules{ruleName}
```

要求パラメータ

パラメータ名	データタイプ	説明
policyGuid	string	ポリシーの GUID。Get Policies メソッドによって返された guid を使用します。
ruleName	string	規則名。Get Policies メソッドによって返されるルール名の 1 つを使用します。

要求本文

パラメータ名	データタイプ	説明
resources	array(string)	GUID リンクおよび SHQL クエリ両方を含むまたはいずれか一方を含むリスト。 SQL クエリを使用する場合は、式が有効で結果を返すことを確認してください。SQL ユーザーガイドを参照してください。 SQL クエリを渡すときは、必ず "..." と \s の組み合わせで囲んでください (例: "link[.layer=\"R_LOGICAL\"]")。

リクエストボディの例

```
{
  "resources": [
    "link[.layer=\"R_LOGICAL\"]"
  ]
}
```

または

```
{
  "resources": [
    "LI/guid1",
    "LI/guid2"
  ]
}
```

または

```
{
  "resources": [
    "inventory[.name=\"CR1.PAR\"]|port|link[.layer=\"R_LOGICAL\"]"
  ]
}
```

応答の例

```
201 Successful Operation
```

ポリシーからのルールの削除

この API を使用して、ポリシーからルールを削除します。

要求メソッド

```
DELETE
```

要求 URL

```
https://example-host/api/v2/apps/srlg-app/rest/policy/{policyGuid}
/policy/{policyGuid}/rules/{ruleName}
```

要求パラメータ

パラメータ名	データタイプ	説明
policyGuid	string	ポリシーの GUID。Get Policies メソッドによって返された guid を使用します。
ruleName	string	規則名。Get Policies メソッドによって返されるルール名の 1 つを使用します。

応答の例

```
200 Successful
```

SHQL を使用した一般的な情報の検索

Crosswork Hierarchical Controller は、Sedona Hierarchical Query Language (SHQL) と呼ばれる、ネットワークデータ用に設計された特殊なクエリ言語を提供します。SHQL を使用すると、ユーザーは任意クエリを指定して結果をすばやく確認できるだけでなく、他のユーザーのためにクエリを保存することもできます。

SHQL には、ユーザーが単純なキーワードを使用して多層モデルにクエリを実行できるという固有の機能があります。キーワードを使用すると、モデルをナビゲートしたり、あるオブジェクトタイプから別のオブジェクトタイプに遷移したり、複雑な条件の必要性を排除したりできます。

遷移とは、SHQL がオブジェクトのリストを取得し、これを使用して関連するオブジェクトに移動する場合を意味します。たとえば、システム内のすべての SITES を取得してから、言語演算子「|」（パイプ演算子）を使用して、それらのサイトのすべての INVENTORY を取得する場合などのです。

たとえば、特定の LSP によって使用されるすべての OMS は「downward」コマンドを使用するだけで取得できます。これは、指定された LSP について、指定された下位レイヤーのすべてのリンクを取得するコマンドです。

```
link[.name="my_lsp"] | downward ("OMS")
```

SHQL コマンドと構文については、『*Crosswork Hierarchical Controller SHQL Guide*』で説明されています。

SHQL は、Crosswork Hierarchical Controller の UI アプリケーションで、または REST POST コマンドとして使用できます。

要求メソッド

POST

要求 URL (Request URL)

```
https://<server>/api/v2/shql
```

要求の例

この例では、クエリは、動作状態が down のすべての LSP のアンダーレイである OMS のリストを取得します。

```
link[.layer="LSP" and .operStatus="DOWN"] | downward | link[.layer="OMS"
and .operStatus="DOWN"]
```

応答の例

```
{
  "activeProtectionPriority": null,
  "bidi": true,
  "desc": null,
  "distanceMeters": null,
  "extra": null,
  "guid": "LI/oms/af5e85ffc6049e8f/8c290fec341b62da/9bf4b791d3191519/3837d2f977f671bd",
  "inverseLinkId": null,
  "latencyMicros": null,
  "layer": "OMS",
  "name": "SD1BKL01/1-2-5&8 to SD1SLO01/1-3-5&8",
  "operStatus": "DOWN",
  "pathGroupType": "SINGLE_PATH",
  "paths": [
    {
      "guid":
"PA/oms/af5e85ffc6049e8f/8c290fec341b62da/9bf4b791d3191519/3837d2f977f671bd"
    }
  ],
  "portA": {
    "guid": "PO/oms/af5e85ffc6049e8f/8c290fec341b62da",
    "type": "OMS"
  },
  "portB": {
    "guid": "PO/oms/9bf4b791d3191519/3837d2f977f671bd",
    "type": "OMS"
  },
  "protectionStatus": "N_A",
  "provider": "Topogen",
  "role": "REGULAR",
  "srlgs": null,
  "teMetric": null
}
```

SQL の概要

SHQL - 必要性

Cisco Crosswork Hierarchical Controller は、ネットワーク構造に関する有益な情報を提供し、複数の専用アプリケーションを使用してさまざまな条件をシミュレーションできるようにします。

ただし、独自の分析を実行して、結果のレポートを表示することもできます。多面的データモデルのクエリは複雑であり、モデルに関する深い理解と高度な開発スキルが必要です。次に例を示します。

- あるレイヤーを別のレイヤーでフィルタ処理する
- さまざまな時間からデータを取得する
- 異常を見つける
- 複数のディメンションで属性ごとにリンクをグループ化する

モデルデータは多面的であり、次のものが含まれます。

- ベンダー
- トポロジ
- [レイヤ (Layers)]
- ドメイン
- Status (ステータス)
- 時刻

SHQL アプリケーションは、レイヤーテーブルとディメンションテーブルの間にクロスクエリを実装することにより、この問題を解決します。これによってデータの複雑さを隠し、これらすべてのディメンションにわたってモデルを簡単にクエリできる、シンプルかつ強力なクエリ言語を提供します。

SHQL の概要

SHQL アプリケーションを使用すると、複雑なクエリコマンドをすばやく作成し、Crosswork Hierarchical Controller SHQL REST API へのリクエストとして送信できます。SHQL アプリケーションにブラウザからアクセスすると、クエリを実行し、列見出しの下に規則正しく配列された結果を表示できます。列の種類と数は、表示されるオブジェクトタイプのプロパティに基づいています。

このアプリケーションは、クエリ言語のすべての機能をカバーしています。自動補完のコンテキストメニューを使用すると、実行可能なオブジェクトタイプとそれらに関連するプロパティを簡単に選択し、さまざまなオペランドを使用して複数の条件を適用できます。

SHQL を使用すると、同じクエリコマンドライン内でクエリをあるオブジェクトタイプから別のオブジェクトタイプに変換することにより、表示される情報を操作できます。または、すべてのクエリアイテムが関連する一連のオブジェクトタイプを指定できます。

注： このガイドで使用されるオブジェクトタイプという用語は、同じプロパティを共有するアイテムまたはアイテムのグループとして定義されます。

Crosswork Hierarchical Controller のオブジェクトモデル

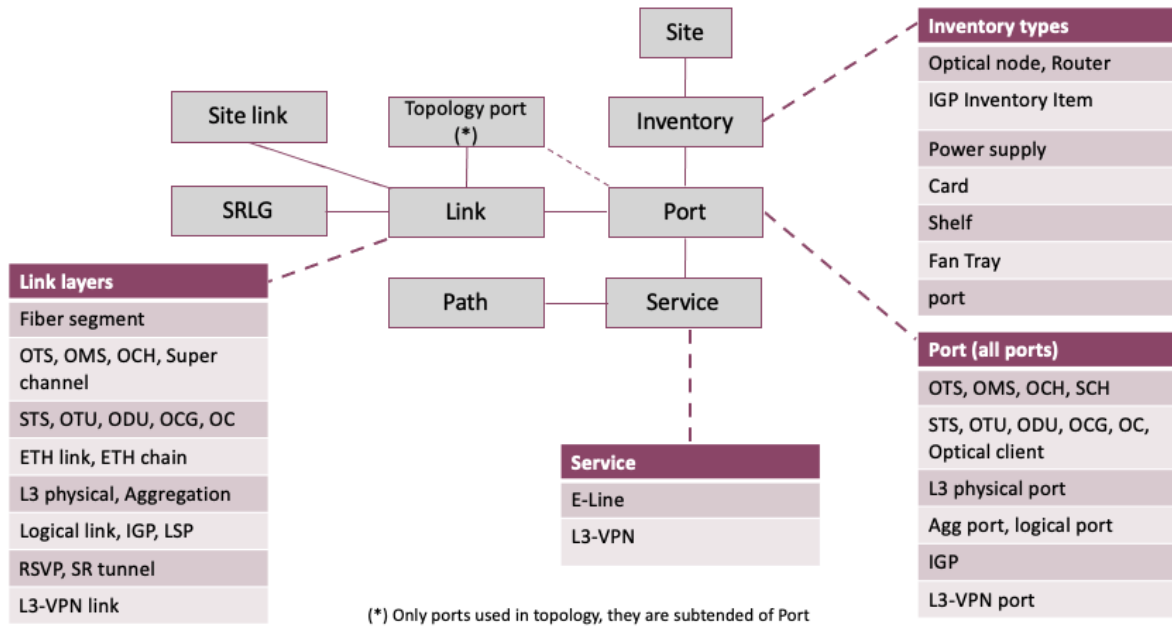


図 1. SHQL オブジェクトモデル

インベントリタイプ

表 2. インベントリタイプ

インベントリ	説明
光ノード	アクティブに管理される光ネットワークの要素。増幅器、ROADM、トランスポンダ、マックスポンダなどが含まれます。
ルータ	ループバック IP アドレスと物理インベントリを持つルータの個別のインスタンス。
IGP インベントリアイテム	ルータのインスタンスとしての論理/仮想ルータ。
電源モジュール	光ノードまたはルータの電源モジュール。
カード	あらゆるタイプの着脱可能なカード。
シェルフ	ルータまたは光ノードのシャーシ。
ファントレイ	光ノードまたはルータの着脱可能または管理可能なファントレイユニット。

リンク タイプ

表 3. リンク タイプ

インベントリ	説明
光ファイバセグメント	1 つの DUCT から別の DUCT にまたがり、光ファイバリンクのセグメントとして使用される物理光ファイバ回線。
ファイバ	光デバイス間にまたがる光ファイバセグメントのチェーン。
OTS	OMS リンクのアンダーレイとして使用される論理レイヤ。 1 つの光ファイバリンク上に 1 つの OTS リンクを作成できます。
OMS	OCH リンクのアンダーレイとして使用される論理レイヤ。 OTS リンクのチェーン上に 1 つの OMS リンクを作成できます。
NMC	1 つ以上の OMS リンクを介した 2 つの ROADM 間の波長接続。マックスポンダのクライアントポートには接続していません。
OCH	いずれかの OEO デバイス (トランスポンダ、マックスポンダ、リジェネレータ) のクライアントポートと別のデバイス間の波長接続。 40 または 80 の OCH リンクを 1 つの OMS リンク上に作成できます。 OCH リンクは 1 つのクライアントポートから別のクライアントポートにまたがり、クライアントポートは TDM または ETH ポートのいずれかです。
SCH	ある ROADM デバイスから別の ROADM デバイスにまたがり、クライアントポートではなくデバイスで終端する OCH。
STS、OCG、OC	1 つの光デバイスから別の光デバイスにまたがる TDM。OCH リンクの上に位置し、TDM クライアントポートで終端する。
OTU	ODU リンクに使用される OTN レイヤのアンダーレイリンク。OCH の上に配置できます。
ODU	OTU リンクのサブ信号。 各 OTU リンクは複数の ODU リンクを伝達することができ、ODU リンクはより細かい ODU リンクに再帰的に分割できます。
ETH リンク	1 つの ETH UNI ポートから別のポートにまたがり、ODU の上位に位置する ETH L2 リンク。
ETH チェーン	ETH リンクのチェーン。
L3 物理	2 つの ETH ポート間のイーサネットリンク (L2 接続)。 これは、2 つのルーター間の IP サブネットリンクのアンダーレイレイヤです。
Agg	複数の ETH リンクがグループ化されて、より高い BW と復元力のあるリンクを作成する Link Aggregation Group (LAG)。
論理リンク	2 つのルーター間の物理リンクを介して 2 つの IP サブネットが接続されるリンク。
IGP	2 つの特定のノード間の最短パスとしてルータによって計算されたパス。 IGP プロトコルメッセージを伝送する 2 つのルータ間のリンク。リンクは IGP の隣接関係を表します。
Tunnel	TE オプションの有無にかかわらず、IGP リンクを介して 2 つのルータ間で作成される MPLS RSVP またはセグメント ルーティング トンネル。
L3-VPN リンク	特定の L3-VPN の 2 つのサイト間の接続 (LSP 接続または IGP パスのチェーンである場合があります)。

Crosswork Hierarchical Controller の接続モデル

例：100G OTN リンク上の 10GE

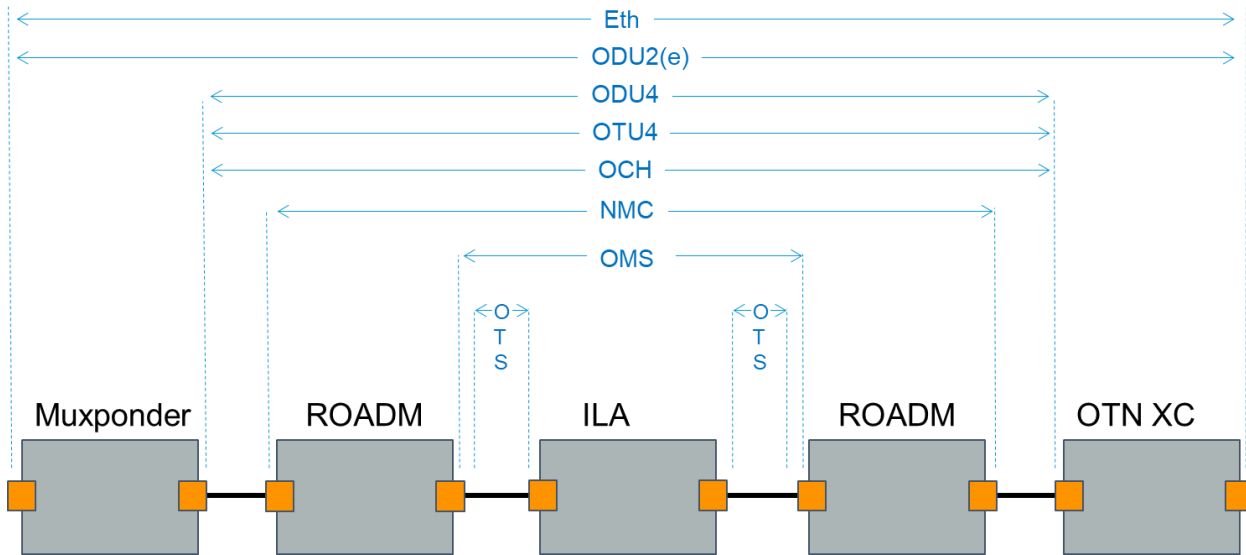


図 2.
接続モデル

SQL の例

この例では、2 つの LSP 間に共有 OMS があるかどうかを検出しています。

The screenshot shows the SHQL interface with a query to find common OMS between two LSPs. The query is: `link[.name="CR2.DUS:CR2.BIL:lsp_0"] | downward | link[.layer="OMS"] as p; link[.name="CR2.DUS:CR1.BIL:lsp_0"] | downward | link[.layer="OMS"] as w; link[.guid in p and .guid in w]`. The results table shows 9 OMS links.

Guid	Layer	Name	OperStatus	PathGroupType	Paths	PortA
9 ITEMS						
Lj/oms/022fa130aca541bb/38...	OMS	SD1GIR01/1-3-5&8 to SD1PAM...	UP	SINGLE_PATH	[[{'guid': 'PA/oms/022fa130aca...	PO/oms/022fa130aca541bb/3...
Lj/oms/b0bd0180e1c79218/8...	OMS	SD1BCN01/1-2-5&8 to SD1MA...	UP	SINGLE_PATH	[[{'guid': 'PA/oms/b0bd0180e1...	PO/oms/b0bd0180e1c79218/...
Lj/oms/b0bd0180e1c79218/0...	OMS	SD1BCN01/1-5-5&8 to SD1GIR...	UP	SINGLE_PATH	[[{'guid': 'PA/oms/b0bd0180e1...	PO/oms/b0bd0180e1c79218/...
Lj/oms/990aae8f0c12944/38...	OMS	SD1STU01/1-3-5&8 to SD1ZU...	UP	SINGLE_PATH	[[{'guid': 'PA/oms/990aae8f0c...	PO/oms/990aae8f0c12944/38...
Lj/oms/738790a2aca5de20/0...	OMS	SD1MAR01/1-5-5&8 to SD1MIL...	UP	SINGLE_PATH	[[{'guid': 'PA/oms/738790a2ac...	PO/oms/738790a2aca5de20/0...

図 3.
SHQL の例

SHQL の使用

SHQL インターフェースについて

ブラウザで、サイドバーのボタンをクリックして SHQL アプリケーションを開きます。

注： Chrome を使用することをお勧めします。

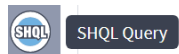


図 4.
SHQL アプリケーションのボタン

SHQL アプリケーション インターフェースが表示されます。クエリを挿入するフィールドは空です。

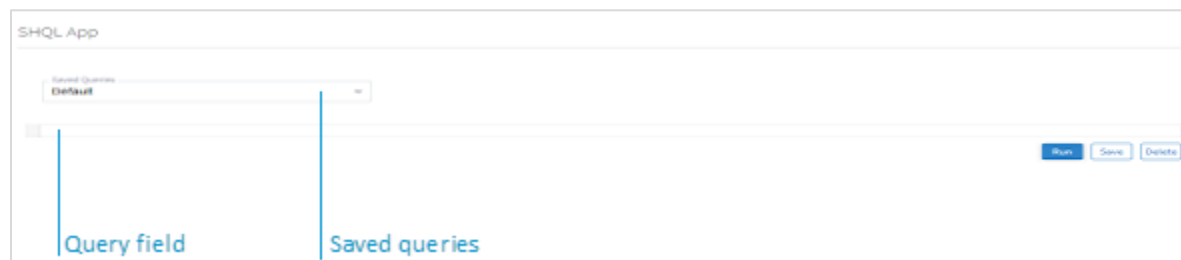


図 5.
SHQL インターフェース

クエリを実行したら、後で使用するために保存できます（クエリの保存を参照）。

クエリーの実行

クエリを実行するには以下の手順に従ってください。

1. [クエリ (Query)] フィールドにクエリコマンドを入力します（クエリの作成を参照）。
2. [実行 (Run)] をクリックします。

取得されたデータは、関連するタブの下に整理されます。

図 6 は、**inventory_item** というクエリを実行した後、2 つのタブの下に表示された結果のリストを示しています。

SHQL

Records fetched at: 12:11:12 06-07-2020

Saved Queries

inventory_item

Run Save Delete Query

RESULTS (546)

Optical Node (463) Router (83)

Guid	Type	Name	Provider	DeviceFamily	DeviceType	ReachabilityStatus	Site	Tags	Vendor	HasRoadm
463 ITEMS										
IN/002d237f16fb...	ONE	ILA-SD1EVO01-S...	Topogen	ILA	ONE	REACHABLE	ST/002d237f16f...	['Vendor': ['Cien...	Ciena	
IN/02539c320f9...	ONE	ILA-SD2BRA01-S...	Topogen	ILA	ONE	REACHABLE	ST/02539c320f9...	['Vendor': ['Cien...	Ciena	
IN/027e3d88f5b...	ONE	ILA-SD1PRA01-S...	Topogen	ILA	ONE	REACHABLE	ST/027e3d88f5b...	['Vendor': ['Cien...	Ciena	
IN/04346445503...	ONE	ILA-SD1BOR01-S...	Topogen	ILA	ONE	REACHABLE	ST/04346445503...	['Vendor': ['Cien...	Ciena	
IN/05717daa222...	ONE	ILA-SD1FRA01-S...	Topogen	ILA	ONE	REACHABLE	ST/05717daa222...	['Vendor': ['Cien...	Ciena	
IN/05b5b6c8c0a...	ONE	ILA-SD1PAM01-S...	Topogen	ILA	ONE	REACHABLE	ST/05b5b6c8c0a...	['Vendor': ['Cien...	Ciena	
IN/0727496c71b...	ONE	ILA-SD2PRA01-S...	Topogen	ILA	ONE	REACHABLE	ST/0727496c71b...	['Vendor': ['Cien...	Ciena	
IN/07dcb20e3f7...	ONE	ILA-SD2CAR01-S...	Topogen	ILA	ONE	REACHABLE	ST/07dcb20e3f7...	['Vendor': ['Cien...	Ciena	
IN/082ba1fca29...	ONE	ILA-SD1EVO01-S...	Topogen	ILA	ONE	REACHABLE	ST/082ba1fca29...	['Vendor': ['Cien...	Ciena	
IN/0975b2257ba...	ONE	ILA-SD1BER01-S...	Topogen	ILA	ONE	REACHABLE	ST/0975b2257ba...	['Vendor': ['Cien...	Ciena	

図 6 :
インベントリアイテムの結果を示すクエリ画面 - 光ノード

SHQL

Records fetched at: 12:11:12 06-07-2020

Saved Queries

inventory_item

Run Save Delete Query

RESULTS (546)

Optical Node (463) Router (83)

Guid	Type	Name	Provider	Extra	ManagementIp	ReachabilityStatus	Site	Tags	Vendor
83 ITEMS									
IN/Router/000fc44...	ROUTER	CR1.BEL	Topogen	['is_core': True]	10.40.0.7	REACHABLE	ST/a9bd23d3ed70	['Vendor': ['Cisco']]	Cisco
IN/Router/03a767f...	ROUTER	ER1.ATH	Topogen	['is_core': True]	10.40.0.46	REACHABLE	ST/7aac1877cd8b	['Vendor': ['Huawei']]	Huawei
IN/Router/0529f67...	ROUTER	CR2.COR	Topogen	['is_core': True]	10.40.0.61	REACHABLE	ST/26a779904306	['Vendor': ['Cisco']]	Cisco
IN/Router/0640f5f...	ROUTER	CR1.BUD	Topogen	['is_core': True]	10.40.0.44	REACHABLE	ST/5960073736d5	['Vendor': ['Cisco']]	Cisco
IN/Router/09ac8d...	ROUTER	CR2.LIV	Topogen	['is_core': True]	10.40.0.4	REACHABLE	ST/9354b4a138dd	['Vendor': ['Cisco']]	Cisco
IN/Router/0a1e09e...	ROUTER	CR2.PRA	Topogen	['is_core': True]	10.40.0.27	REACHABLE	ST/6bdd2d5799e	['Vendor': ['Cisco']]	Cisco
IN/Router/0a22a0...	ROUTER	CR1.TIV	Topogen	['is_core': True]	10.40.0.67	REACHABLE	ST/c8037c4efe39	['Vendor': ['Cisco']]	Cisco
IN/Router/14bc8d...	ROUTER	CR2.MAD	Topogen	['is_core': True]	10.40.0.38	REACHABLE	ST/d5e070604ed5	['Vendor': ['Cisco']]	Cisco
IN/Router/1877a9...	ROUTER	ER1.WAR	Topogen	['is_core': True]	10.40.0.83	REACHABLE	ST/ec803a477b10	['Vendor': ['Huawei']]	Huawei
IN/Router/19694a...	ROUTER	CR1.WAR	Topogen	['is_core': True]	10.40.0.28	REACHABLE	ST/ec803a477b10	['Vendor': ['Cisco']]	Cisco

図 7 .
インベントリアイテムの結果を示すクエリ画面 - ルータ

GUID 列のオブジェクトの隣にある [...] をクリックすると、そのプロパティが JSON 形式で表示されます。JSON には、オブジェクトのプロパティとその値がリストされます。

```
JSON [X]
{
  "accessIdentifier": null,
  "children": null,
  "desc": null,
  "deviceFamily": null,
  "deviceType": null,
  "extra": {
    "is_core": true
  },
  "guid": "IN/Router/000fc44c94a1f2cd",
  "managementIp": "10.40.0.7",
  "name": "CR1.BEL",
  "parent": null,
  "provider": "Topogen",
  "reachabilityStatus": "REACHABLE",
  "serialNumber": null,
  "site": {
    "guid": "ST/a9bd23d3ed70"
  },
  "softwareVersion": null,
  "srlgs": null,
  "tags": {
    "Vendor": [
      "Cisco"
    ]
  },
  "topologyId": null,
  "type": "ROUTER",
  "vendor": "Cisco"
}
```

図 8.
.JSON 形式のインベントリオブジェクト

クエリの中止

結果を返すまでに時間がかかるクエリを実行する場合、クエリを中止できます。

クエリを中止するには以下を実行します。

1. [クエリ (Query)] フィールドにクエリコマンドを入力します (クエリの作成を参照) 。
2. [実行 (Run)] をクリックします。クエリが結果を返すまでに時間がかかる場合は、中止オプションが提示されます。

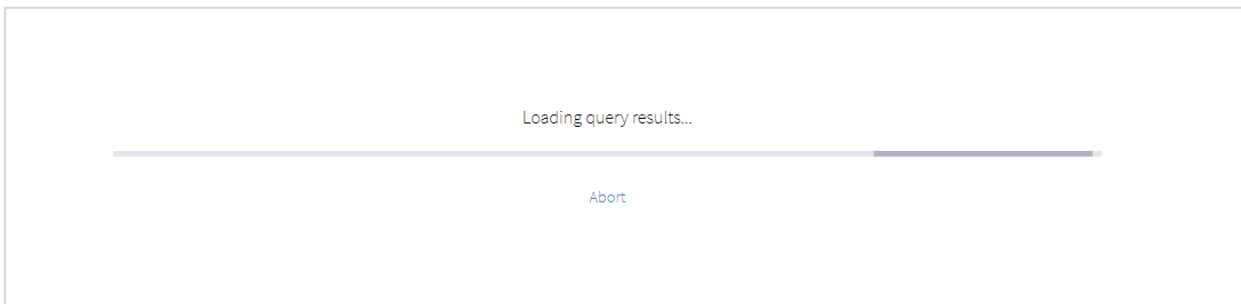


図 9.
クエリの中止

3. クエリを終了する場合は、[中止 (Abort)] をクリックします。

クエリの作成

フリー テキストを使用するか、オートコンプリート コンテキスト メニューから項目を選択して、またはその両方を組み合わせてクエリを作成できます。[クエリ (Query)] フィールドに入力する最初の項目はオブジェクトタイプです。

クエリ構文の任意の場所でオートコンプリート コンテキスト メニューを使用できます。

注： クエリでは大文字と小文字が区別されません。

オートコンプリート コンテキスト メニューを表示するには以下を実行します。

- [クエリ (Query)] フィールド内をクリックし、次のいずれかを実行します。
 - Mac : Command + スペースバーを押します。
 - PC : Ctrl + スペースバーを押します

メニューが表示されます。

図 10 は、オブジェクトタイプのオートコンプリート コンテキスト メニューを示しています。

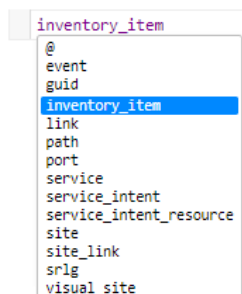



図 10

オートコンプリート コンテキスト メニュー

メニューから項目を選択すると、その項目が [クエリ (Query)] フィールドに表示されます。引き続きクエリを作成し、条件と関数を追加できます。

フリーテキストまたはコンテキストメニューの選択による入力に誤りがある場合、赤いアイコン  で示されません。アイコンにカーソルを合わせると、入力が実行可能でない理由が示されるか、予想される入力が示されます。

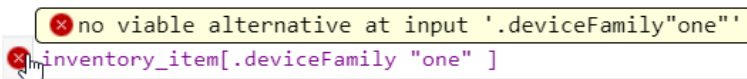


図 11.

入力が実行不可能

エラーは、図 12 : に示すように単純に構文の問題の場合もあります。

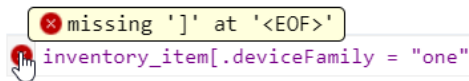


図 12 :
構文エラー

注： クエリ対象のオブジェクトタイプの .JSON ファイルを調べて、実行可能な入力を確認してください。

クエリの保存

クエリを作成したら、それを [保存されたクエリ (Saved Queries)] ドロップダウンメニューに保存して、繰り返し使用することができます。

クエリを保存するには以下を実行します。

1. クエリの作成が完了したら、[名前を付けて保存 (Save As)] をクリックします。
次のダイアログボックスが表示されます。

図 13.
クエリ名のダイアログボックス

2. クエリの名前を入力し、[保存 (Save)] をクリックします。

図 14.
保存されたクエリ

3. 保存するすべてのクエリに対して手順 1 と 2 を繰り返します。

クエリの削除

クエリを削除するには以下を実行します。

1. 保存されたクエリのドロップダウンリストからクエリを選択します。

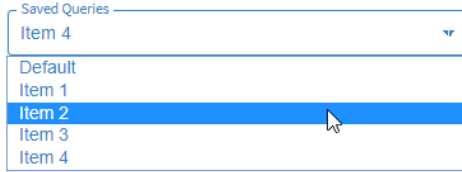


図 15.

クエリの削除

2. [削除 (Delete)] をクリックします。

クエリがリストから削除されます。

列のソート

取得されたデータは、関連する列見出しの下に表示されます。

アイテムを昇順または降順に並べ替え、個々の列にフィルタを適用できます。

注： 列管理は、他の Crosswork Hierarchical Controller テーブルと同様に処理されます。

列をソートするには以下を実行します。

1. 列の見出しをクリックします。

上向きの矢印が表示されます。

↓ Name	☰
ER1.WAR	
ER1.VAL	
ER1.TLV	
ER1.STO	
ER1.SQY	
ER1.PAR	
ER1.OVE	
ER1.MAN	
ER1.MAD	
ER1.LIS	

図 16.

降順に表示された列コンテンツ

2. 矢印をクリックすると、列のアイテムが昇順にソートされます。再度クリックすると、列のアイテムが降順にソートされます。

↓ Name
ER1.WAR
ER1.VAL
ER1.TLV
ER1.STO
ER1.SQY
ER1.PAR
ER1.OVE
ER1.MAN
ER1.MAD
ER1.LIS

図 17.
昇順に表示された列コンテンツ

列のコンテンツをフィルタ処理するには以下を実行します。

1. 列見出しで、 [フィルタ (Filter)] アイコンをクリックします。
列見出しが展開され、フィルタ処理オプションが表示されます。

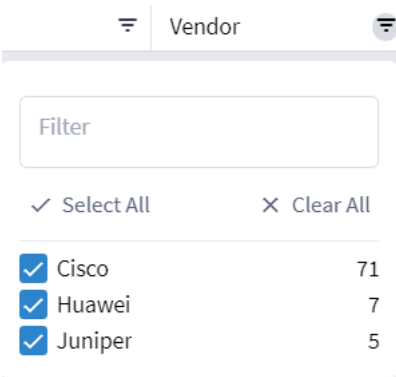


図 18.
列のコンテンツのフィルタ処理


2. オプションの値を選択するか、テキストボックスにパターンを入力して、列のコンテンツをフィルタ処理します。列のフィルタアイコンが  に変わり、フィルタが列に適用されたことを示します。



図 19.

列に適用されたフィルタ

条件の作成

以下の表のオペランドを使用して、オブジェクトタイプに条件を適用できます。

通常、条件では大文字と小文字が区別されません。定義済みの値リストの一部である場合にのみ、大文字と小文字を区別する必要があります。

条件は角カッコ ([]) で囲みます。

表 4. オペランド

オペランド	数字	文字列	説明
!=	✓	✓	次の値と異なる。
<	✓		次の値より小さい。
<=	✓		右辺と比較して小さいか等しい。
=	✓	✓	次の値と等しい。
>	✓		次の値より大きい。
>=	✓		次の値より大きい等しい。
が次の文字列を含む (contains)		✓	一部一致する。
endswith		✓	特定のパターンで終わる。
所有			配列内のアイテム。 フィールドがリストの場合にアイテムを探すために使用します。
in	✓	✓	パターン的一致リスト。 フィールドが 1 つのアイテムであり、フィルタに複数のアイテムが含まれる場合に使用します。
intersect	✓		特定の経度と緯度にあるリージョンの地理的交差点。 次に例を示します。 region[.geometry intersect (4.8945398, 52.3666)]
は		✓	boolean (true / false) および null。
not		✓	contains、endswith、startswith と合わせて使用し、条件を否定します。
startswith		✓	特定のパターンから開始する。
xin	✓	✓	複数のプロパティのパターン的一致リスト。
~		✓	正規表現を使用した文字列。

例 1 : オペランドによるフィルタ処理

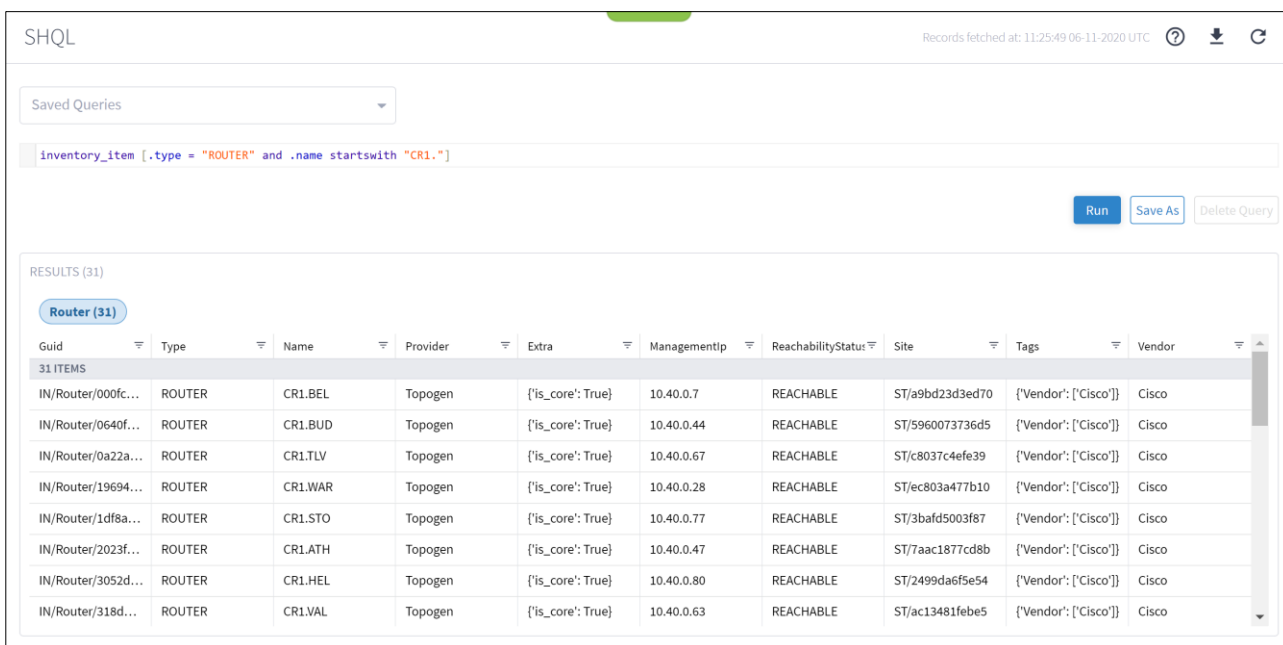
図 20 : は、CR1 で始まる名前のルータのリストを取得します

```
inventory_item [.type = "ROUTER" and .name startswith "CR1."]
```

図 20 :

オペランドを使用したフィルタ処理。

図 21 は結果を示しています。



The screenshot shows the SHQL interface with a query and its results. The query is `inventory_item [.type = "ROUTER" and .name startswith "CR1."]`. The results are displayed in a table with 11 columns: Guid, Type, Name, Provider, Extra, Managementip, ReachabilityStatus, Site, Tags, and Vendor. There are 31 items in total, all of which are Routers from the provider Topogen, with various management IP addresses and sites.

Guid	Type	Name	Provider	Extra	Managementip	ReachabilityStatus	Site	Tags	Vendor
IN/Router/000fc...	ROUTER	CR1.BEL	Topogen	{'is_core': True}	10.40.0.7	REACHABLE	ST/a9bd23d3ed70	{'Vendor': ['Cisco']}	Cisco
IN/Router/0640f...	ROUTER	CR1.BUD	Topogen	{'is_core': True}	10.40.0.44	REACHABLE	ST/5960073736d5	{'Vendor': ['Cisco']}	Cisco
IN/Router/0a22a...	ROUTER	CR1.TLV	Topogen	{'is_core': True}	10.40.0.67	REACHABLE	ST/c8037c4efe39	{'Vendor': ['Cisco']}	Cisco
IN/Router/19694...	ROUTER	CR1.WAR	Topogen	{'is_core': True}	10.40.0.28	REACHABLE	ST/ec803a477b10	{'Vendor': ['Cisco']}	Cisco
IN/Router/1df8a...	ROUTER	CR1.STO	Topogen	{'is_core': True}	10.40.0.77	REACHABLE	ST/3bafd5003f87	{'Vendor': ['Cisco']}	Cisco
IN/Router/2023f...	ROUTER	CR1.ATH	Topogen	{'is_core': True}	10.40.0.47	REACHABLE	ST/7aac1877cd8b	{'Vendor': ['Cisco']}	Cisco
IN/Router/3052d...	ROUTER	CR1.HEL	Topogen	{'is_core': True}	10.40.0.80	REACHABLE	ST/2499da6f5e54	{'Vendor': ['Cisco']}	Cisco
IN/Router/318d...	ROUTER	CR1.VAL	Topogen	{'is_core': True}	10.40.0.63	REACHABLE	ST/ac13481febe5	{'Vendor': ['Cisco']}	Cisco

図 21 :

クエリの結果

例 2 : オペランドによるフィルタ処理

図 22 は、ベンダーが Ciena であるインベントリを取得します。

```
inventory[.tags.Vendor has ("Ciena")]
```

図 22 :

オペランドを使用したフィルタ処理。

図 23 は結果を示しています。

SHQL Records fetched at: 09:39:11 06-11-2020 UTC

Saved Queries

`Inventory[.tags.Vendor has ("Ciena")]`

Run Save As Delete Query

RESULTS (424)

Optical Node (424)

Guid	Type	Name	Provider	DeviceFamily	DeviceType	ReachabilityStat	Site	Tags	Vendor	HasRoadm
424 ITEMS										
IN/002d237f16...	ONE	ILA-SD1EVO01...	Topogen	ILA	ONE	REACHABLE	ST/002d237f1...	['Vendor': ['Cie...	Ciena	
IN/02539c320f...	ONE	ILA-SD2BRA01...	Topogen	ILA	ONE	REACHABLE	ST/02539c320f...	['Vendor': ['Cie...	Ciena	
IN/027e3d88f5...	ONE	ILA-SD1PRA01...	Topogen	ILA	ONE	REACHABLE	ST/027e3d88f...	['Vendor': ['Cie...	Ciena	
IN/043464455...	ONE	ILA-SD1BOR01...	Topogen	ILA	ONE	REACHABLE	ST/043464455...	['Vendor': ['Cie...	Ciena	
IN/05717daa2...	ONE	ILA-SD1FRA01...	Topogen	ILA	ONE	REACHABLE	ST/05717daa2...	['Vendor': ['Cie...	Ciena	
IN/05b5b6c8c...	ONE	ILA-SD1PAM01...	Topogen	ILA	ONE	REACHABLE	ST/05b5b6c8c...	['Vendor': ['Cie...	Ciena	
IN/07774967...	ONF	II A-SD2PRA01...	Tonnoen	II A	ONF	RFACHARI F	ST/07774967...	['Vendor': ['Cie...	Ciena	

図 23 :
クエリの結果

例 3 : オペランドによるフィルタ処理

図 24 は、(port_a.guid = "PO/igp/06c0868cc5601e85/06c0868cc5601e85" and port_b.guid = "PO/igp/c3b0dc2cd3ad6406/c3b0dc2cd3ad6406") または (port_a.guid = "PO/igp/c3b0dc2cd3ad6406/c3b0dc2cd3ad6406" and port_b.guid = "PO/igp/06c0868cc5601e85/06c0868cc5601e85") のリンクを探します。

```
link[.layer="LSP" and (.portA, .portB) xin (port[.guid = "PO/igp/06c0868cc5601e85/06c0868cc5601e85"], port[.guid = "PO/igp/c3b0dc2cd3ad6406/c3b0dc2cd3ad6406"])]
```

図 24 :
オペランドを使用したフィルタ処理。

図 25 は結果を示しています。

SHQL Records fetched at: 09:36:45 06-11-2020 UTC ? ↓ ↻

Saved Queries

```
link[.layer="LSP" and (.portA, .portB) xin (port[.guid = "PO/igp/06c0868cc5601e85/06c0868cc5601e85"], port[.guid = "PO/igp/c3b0dc2cd3ad6406/c3b0dc2cd3ad6406"])]
```

Run Save As Delete Query

RESULTS (1)

LSP Link (1)

Guid	Layer	Name	OperStatu	PathGroup	Paths	PortA	PortB	Protection	Provider	Role	HoldPrior	LspTechno	SetupPrior	SpeedBps
L/isp/c3...	LSP	ER1.WAR...	UP	SINGLE_...	[[guid]:'...	PO/igp/c...	PO/igp/0...	N_A	Topogen	REGULAR	7	MPLS	7	100000000

図 25 :
クエリの結果

例 4 : オペランドによるフィルタ処理

図 26 は、一昨日 (>-2d および <-1d) にキャプチャされたイベントを取得して合計します。timeStamp プロパティは、d (日)、h (時間)、m (月 0)、M (分)、S (秒)、y (年)、または w (週) と合わせて使用できます。

```
event[.timeStamp > -5d and .timeStamp < -1d]
```

図 26 :
オペランドを使用したフィルタ処理。

図 27 は結果を示しています。

RESULTS (3018)

Event (3018)

Count	Data	Guid	LastUpdate	Machineld	Severity	SubType	TimeStamp	Type	Username
3018 ITEMS									
1	{'msg': 'Start ...	EV/11ec0506...	2021-08-24T1...	nir-nachum-t...	DEBUG	calculation	2021-08-24T1...	visual-model	system
1	{'msg': 'Persis...	EV/11ec0506...	2021-08-24T1...	nir-nachum-t...	INFO	logical-link-p...	2021-08-24T1...	persistor	system
1	{'msg': 'Done ...	EV/11ec0506...	2021-08-24T1...	nir-nachum-t...	INFO	calculation	2021-08-24T1...	visual-model	system
1	{'msg': 'Persis...	EV/11ec0506...	2021-08-24T1...	nir-nachum-t...	INFO	otn-persistor-...	2021-08-24T1...	persistor	system
1	{'msg': 'Confi...	EV/11ec0506...	2021-08-24T1...	nir-nachum-t...	WARNING	remove	2021-08-24T1...	dynamic-apps	system
1	{'msg': 'Confi...	EV/11ec0506...	2021-08-24T1...	nir-nachum-t...	WARNING	remove	2021-08-24T1...	dynamic-apps	system

図 27 :
クエリの結果

オブジェクトタイプの変換/収集

オブジェクトタイプをクエリコマンドに追加して、取得するデータが1つのオブジェクトタイプから別のオブジェクトタイプに変換されるかどうか、またはデータが複数のオブジェクトタイプのコレクションとそれらの関連アイテムを反映するかどうかを判断できます。

- **変換**：新しいオブジェクトタイプを追加する前に、クエリコマンドにパイプ (|) を追加します。以前のオブジェクトタイプに関連する結果を、新しいオブジェクトタイプの出力に変換します。
- **コレクション**：新しいオブジェクトタイプを追加する前に、クエリコマンドにアンパサンド (&) を追加します。先行するすべてのオブジェクトタイプのすべての出力を取得します。
- **As**：一時変数を追加します。先行するオブジェクトタイプに関連しないオブジェクトタイプでクエリを作成できます。

以下の表は、オブジェクトタイプと、それらが直接関連するオブジェクトタイプを示しています。

表 5. オブジェクトのタイプとプロパティ

オブジェクトタイプ	オブジェクトタイプのプロパティ
イベント	該当なし
インベントリ	inventory_item インベントリ port site srlg
inventory_item	inventory_item インベントリ port site srlg
link	link path port site link srlg region
path	path link
port	port link inventory_item インベントリ

オブジェクトタイプ	オブジェクトタイプのプロパティ
	service srlg
region	site visual site link
risk_resource	srlg_risk_resource_mtm risk_resource port link inventory_item インベントリ
service	port service intent
service_intent	service_intent service service_intent_resource
service_intent_resource	service_intent_resource service_intent
site	site inventory item インベントリ visual site region
site_link	site_link visual_site link
srlg	srlg port link inventory_item インベントリ
srlg_risk_resource_mtm	srlg_risk_resource_mtm risk_resource port link inventory_item

オブジェクトタイプ	オブジェクトタイプのプロパティ
	インベントリ
visual_site	visual_site site_link site region

例：オブジェクトタイプの変換

図 28：図 28 は、オブジェクトタイプのインベントリアイテムに属するポートを取得するために作成されたクエリを示しています。オブジェクトタイプはパイプ (|) で区切られます。

```
inventory_item | port
```

図 28：
オブジェクトタイプの変換を使用したクエリ

図 29 は結果を示しています。

The screenshot shows a query interface with the query 'inventory_item | port' entered. Below the query, there are buttons for 'Run', 'Save', and 'Delete Query'. The results section shows 4136 results. A filter bar displays various port types with their counts: OTS Port (920), OMS Port (276), OCH Port (178), OTU Port (290), ODU Port (302), Optical Client Port (564), Ethernet Port (166), L3 Physical Port (717), Aggregation Port (6), Logical Port (366), IGP Port (343), and L3 VPN Port (8). The main table displays the following columns: Guid, Type, AdminStatus, Device, Name, OperStatus, Provider, and RelDirection. The first few rows of the table are as follows:

Guid	Type	AdminStatus	Device	Name	OperStatus	Provider	RelDirection
PO/ots/0af713cb7cb832...	OTS	UP	IN/0af713cb7cb8326d	1-1-5&8	UP	Topogen	NONE
PO/ots/0ba52a24245563...	OTS	UP	IN/ROADM/0ba52a24245...	1-3-5&8	UP	Topogen	NONE
PO/ots/0ba52a24245563...	OTS	UP	IN/ROADM/0ba52a24245...	1-4-5&8	UP	Topogen	NONE
PO/ots/0ba52a24245563...	OTS	UP	IN/ROADM/0ba52a24245...	1-2-5&8	UP	Topogen	NONE
PO/ots/0bd548bdeb61b...	OTS	UP	IN/0bd548bdeb61b282	1-2-5&8	UP	Topogen	NONE
PO/ots/0bd548bdeb61b...	OTS	UP	IN/0bd548bdeb61b282	1-1-5&8	UP	Topogen	NONE
PO/ots/0cc638600c3fcd...	OTS	UP	IN/0cc638600c3fcdcf	1-2-5&8	UP	Topogen	NONE
PO/ots/0cc638600c3fcd...	OTS	UP	IN/0cc638600c3fcdcf	1-1-5&8	UP	Topogen	NONE
PO/ots/0e847d73703ce7...	OTS	UP	IN/0e847d73703ce7d4	1-2-5&8	UP	Topogen	NONE

図 29：
変換の結果

例：オブジェクトタイプのコレクション

図 30 は、サイトとサイトが属するインベントリアイテムを取得するために作成されたクエリを示しています。オブジェクトタイプは、アンパサンド (&) で区切られます。

```
inventory_item & site
```

図 30：
オブジェクトタイプコレクションを使用したクエリ

図 31 は結果を示しています。

Guid	Type	Name	Provider	DeviceFamily	DeviceType	Site	Vendor	HasRoadm
IN/002d237f16fb8...	ONE	ILA-SD1EVO01-SD1...	Topogen	ILA	ONE	ST/002d237f16fb8...	Ciena	
IN/02539c320f9a3dff	ONE	ILA-SD2BRA01-SD2...	Topogen	ILA	ONE	ST/02539c320f9a3dff	Ciena	
IN/027e3d88f5b57...	ONE	ILA-SD1PRA01-SD1...	Topogen	ILA	ONE	ST/027e3d88f5b57...	Ciena	
IN/043464455031f...	ONE	ILA-SD1BOR01-SD1...	Topogen	ILA	ONE	ST/043464455031f...	Ciena	
IN/05717daa222ea...	ONE	ILA-SD1FRA01-SD1...	Topogen	ILA	ONE	ST/05717daa222ea...	Ciena	
IN/05b5b6c8c0a02...	ONE	ILA-SD1PAM01-SD1...	Topogen	ILA	ONE	ST/05b5b6c8c0a02...	Ciena	
IN/0727496c71b88...	ONE	ILA-SD2PRA01-SD2...	Topogen	ILA	ONE	ST/0727496c71b88...	Ciena	
IN/07dcb20e3f79b...	ONE	ILA-SD2CAR01-SD2...	Topogen	ILA	ONE	ST/07dcb20e3f79b...	Ciena	
IN/082ba1fca299661	ONE	ILA-SD1EVO01-SD1...	Topogen	ILA	ONE	ST/082ba1fca2996...	Ciena	
IN/0975b2257baf3...	ONE	ILA-SD1BER01-SD1...	Topogen	ILA	ONE	ST/0975b2257baf3...	Ciena	

図 31 :
コレクションの結果

例 : As 一時変数

As を使用すると、クエリの出力をコレクションの入力として使用できます。

次の図は、一時変数を使用して、関連のないオブジェクトタイプをサポートするように構築されたクエリを作成する方法を示しています。

図 32 は、オブジェクトタイプ link で構成されるクエリコマンドを示しています。

```
link[.layer = "R_LOGICAL"]
```

図 32 :
クエリ

図 33 は結果を示しています。

link[.layer = "R_LOGICAL"]

Run Save Delete

RESULTS (118)

RLogicalLink (118)

Guid	Layer	Name	OperStatus	PathGroupType	Paths	PortA	PortB	ProtectionStatus	Provider	Role
Li/r_logical/0...	R_LOGICAL	10.40.0.45 to ...	UP	SINGLE_PATH	[{"guid": "PA/Li...	PO/r_logical/...	PO/r_logical/...	N_A	Topogen	REGULAR
Li/r_logical/0...	R_LOGICAL	10.40.0.41 to ...	UP	SINGLE_PATH	[{"guid": "PA/Li...	PO/r_logical/...	PO/r_logical/...	N_A	Topogen	REGULAR
Li/r_logical/0...	R_LOGICAL	10.40.0.34 to ...	UP	SINGLE_PATH	[{"guid": "PA/Li...	PO/r_logical/...	PO/r_logical/...	N_A	Topogen	REGULAR
Li/r_logical/0...	R_LOGICAL	10.40.1.118 to ...	UP	SINGLE_PATH	[{"guid": "PA/Li...	PO/r_logical/...	PO/r_logical/...	N_A	Topogen	REGULAR
Li/r_logical/0...	R_LOGICAL	10.40.1.30 to ...	UP	SINGLE_PATH	[{"guid": "PA/Li...	PO/r_logical/...	PO/r_logical/...	N_A	Topogen	REGULAR
Li/r_logical/0...	R_LOGICAL	10.40.1.41 to ...	UP	SINGLE_PATH	[{"guid": "PA/Li...	PO/r_logical/...	PO/r_logical/...	N_A	Topogen	REGULAR
Li/r_logical/0...	R_LOGICAL	10.40.1.34 to ...	UP	SINGLE_PATH	[{"guid": "PA/Li...	PO/r_logical/...	PO/r_logical/...	N_A	Topogen	REGULAR
Li/r_logical/0...	R_LOGICAL	10.40.1.153 to ...	UP	SINGLE_PATH	[{"guid": "PA/Li...	PO/r_logical/...	PO/r_logical/...	N_A	Topogen	REGULAR
Li/r_logical/0...	R_LOGICAL	10.40.1.146 to ...	UP	SINGLE_PATH	[{"guid": "PA/Li...	PO/r_logical/...	PO/r_logical/...	N_A	Topogen	REGULAR
Li/r_logical/0...	R_LOGICAL	10.40.1.62 to ...	UP	SINGLE_PATH	[{"guid": "PA/Li...	PO/r_logical/...	PO/r_logical/...	N_A	Topogen	REGULAR

図 33 :
クエリの結果

図 34 は、**As** を使用して一時変数を作成する方法を示しています。変数 L は、有効なオブジェクトタイプとしてコンテキストメニューに表示されます。

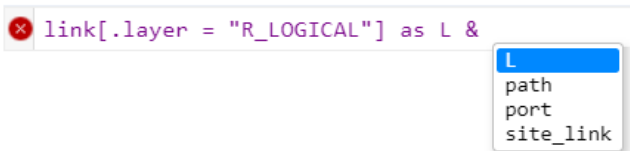


図 34 :
As の追加

図 35 は、一時変数 L を使用して、リンクのリストをポート、パス、サイトリンクに変換する方法を示しています。

```
link[.layer = "R_LOGICAL"] as L & L | port & L | path & L | site_link
```

図 35 :
as を使用したクエリ

図 36 は結果を示しています。

link[.layer = "R_LOGICAL"] as L & L | port & L | path & L |site_link

RESULTS (642)

Site Link (170) Logical Port (236) Logical Link (118) Path (118)

Guid	Depth	Layer	Links	Name	SiteA	SiteB	Status	Tags
SL/R_LOGICAL/VS/596...	3	R_LOGICAL	[{"guid": "L/r_logical/0...	BUD - BUC	VS/5960073736d5_0_1...	VS/a1c737928f90_0_1_2	NORMAL	{"Vendor": ["Cisco"]}
SL/R_LOGICAL/VS/839...	2	R_LOGICAL	[{"guid": "L/r_logical/c...	OVE - MAD	VS/839c2694c73e_0_1	VS/d5e070604ed5_0_1	NORMAL	{"Vendor": ["Cisco"]}
SL/R_LOGICAL/VS/839...	1	R_LOGICAL	[{"guid": "L/r_logical/c...	BCN - MAD	VS/ba7d24f4bc71_0	VS/d5e070604ed5_0	NORMAL	{"Vendor": ["Cisco"]}
SL/R_LOGICAL/VS/3ba...		R_LOGICAL	[{"guid": "L/r_logical/c...	STO - COP	VS/3ba7d5003f87	VS/df8413a1d7a4	NORMAL	{"Vendor": ["Cisco"]}
SL/R_LOGICAL/VS/7aa...	3	R_LOGICAL	[{"guid": "L/r_logical/e...	ATH - BUC	VS/7aac1877cd8b_0_1_2	VS/a1c737928f90_0_1_2	NORMAL	{"Vendor": ["Cisco"]}
SL/R_LOGICAL/VS/26a...	1	R_LOGICAL	[{"guid": "L/r_logical/6...	COR - VAL	VS/26a779904306_0	VS/ac13481febe5_0	NORMAL	{"Vendor": ["Cisco"]}
SL/R_LOGICAL/VS/d5e...		R_LOGICAL	[{"guid": "L/r_logical/5...	MAD - PAR	VS/d5e070604ed5	VS/e7df76d7a9cb	NORMAL	{"Vendor": ["Cisco"]}
SL/R_LOGICAL/VS/56c...	2	R_LOGICAL	[{"guid": "L/r_logical/0...	MAN - BEL	VS/56ceb1686268_0_1	VS/a9bd23d3ed70_0_1	NORMAL	{"Vendor": ["Cisco"]}
SL/R_LOGICAL/VS/297...	1	R_LOGICAL	[{"guid": "L/r_logical/8...	BKL - MAN	VS/2971737bd3ba_0	VS/56ceb1686268_0	DOWN	{"Vendor": ["Cisco"]}
SL/R_LOGICAL/VS/d11...	3	R_LOGICAL	[{"guid": "L/r_logical/d...	LIS - MAD	VS/d11c3fd70e8a_0_1_2	VS/d5e070604ed5_0...	NORMAL	{"Vendor": ["Cisco"]}

図 36 :
一時変数を使用した結果

機能

SHQL には、取得するアイテムの数量、タイプ、ネットワークレイヤを正確に定義できる関数がいくつか用意されています。

クエリコマンドラインでは、関数の前にパイプが付きます。

アイテムを取得してから、関連アイテムをレイヤの上または下のいずれかから取得するか、上下の両方から取得するかを指定できます。これらの再帰操作は、ポート、リンク、サイト、インベントリ、ビジュアルサイトに対して有効です。

レトロスペクティブとタイムスタンプを使用すると、現在より前の状況をクエリできます。たとえば、サイトが現在は稼働しているものの、同じ日のダウンしていたときのステータスを確認できます。

さらに、SHQL 関数により、フリーテキストに基づいてデータを取得できます。

以下の表で SHQL 関数について説明します。SHQL 関数

表 6. SHQL 関数

機能	説明
downward	指定されたアイテムのレイヤの下のレイヤからアイテムを取得する
upward	指定されたアイテムのレイヤの上のレイヤからアイテムを取得する
span	指定されたアイテムのレイヤの上下のレイヤからアイテムを取得する
fts	フリーテキスト検索。入力した検索文字列に従ってアイテムを取得します。
group_by	OperStatus が Up のポートの数など、プロパティごとにグループ化されたクエリ結果を返します。
retrospective(@)	指定されたタイムスタンプに従って、過去のアイテムを取得します。

例 : downward 関数

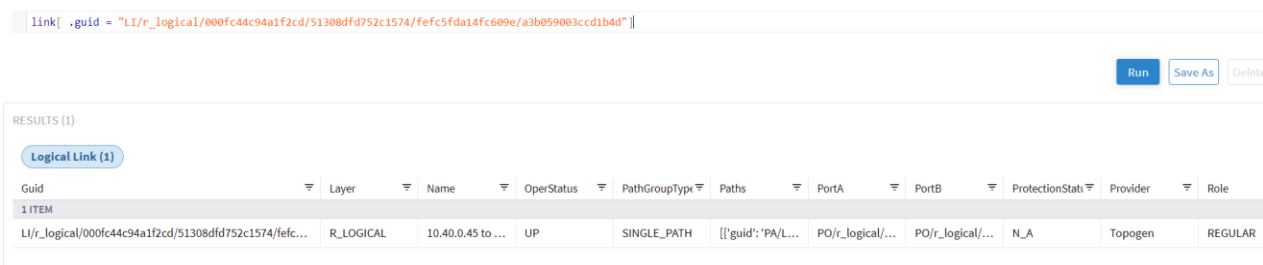
次の図は、GUID に従って、リンクの下にあるすべてのレイヤからアイテムを取得するクエリを作成する方法を示しています。

図 37 は、オブジェクトタイプと後に続く GUID で構成されるクエリコマンドを示しています。

```
link[ .guid = "LI/r_logical/000fc44c94a1f2cd/51308dfd752c1574/fe5fda14fc609e/a3b059003ccd1b4d" ]
```

図 37 :
クエリコマンド

図 38 は結果を示しています。

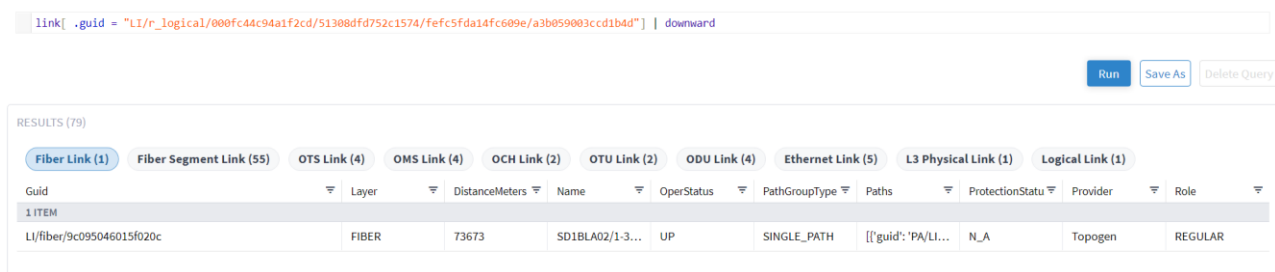


The screenshot shows a query execution interface. At the top, there is a text input field containing the query: `link[.guid = "LI/r_logical/000fc44c94a1f2cd/51308dfd752c1574/fe5fda14fc609e/a3b059003ccd1b4d"]`. To the right of the input field are three buttons: "Run", "Save As", and "Delete". Below the input field, the results are displayed under the heading "RESULTS (1)". There is a single result type "Logical Link (1)". Below this, a table shows the details of the result.

Guid	Layer	Name	OperStatus	PathGroupType	Paths	PortA	PortB	ProtectionStat	Provider	Role
LI/r_logical/000fc44c94a1f2cd/51308dfd752c1574/fe5fda14fc609e/a3b059003ccd1b4d	R_LOGICAL	10.40.0.45 to ...	UP	SINGLE_PATH	[[{'guid': 'PA/L...	PO/r_logical/...	PO/r_logical/...	N_A	Topogen	REGULAR

図 38 :
GUID の結果

図 39 は、コマンドに downward 関数を追加した後に取得された、リンクレイヤの下にある 9 つの追加アイテムを示しています。



The screenshot shows a query execution interface. At the top, there is a text input field containing the query: `link[.guid = "LI/r_logical/000fc44c94a1f2cd/51308dfd752c1574/fe5fda14fc609e/a3b059003ccd1b4d"] | downward`. To the right of the input field are three buttons: "Run", "Save As", and "Delete Query". Below the input field, the results are displayed under the heading "RESULTS (79)". There are multiple result types: "Fiber Link (1)", "Fiber Segment Link (55)", "OTS Link (4)", "OMS Link (4)", "OCH Link (2)", "OTU Link (2)", "ODU Link (4)", "Ethernet Link (5)", "L3 Physical Link (1)", and "Logical Link (1)". Below this, a table shows the details of the first result.

Guid	Layer	DistanceMeters	Name	OperStatus	PathGroupType	Paths	ProtectionStatu	Provider	Role
LI/fiber/9c095046015f020c	FIBER	73673	SD18LA02/1-3...	UP	SINGLE_PATH	[[{'guid': 'PA/LI...	N_A	Topogen	REGULAR

図 39 :
downward 関数の結果

例 : retrospective 関数

図 40 は、過去の特定の時間におけるネットワーク内の状況を取得するために作成されたクエリコマンドを示しています。

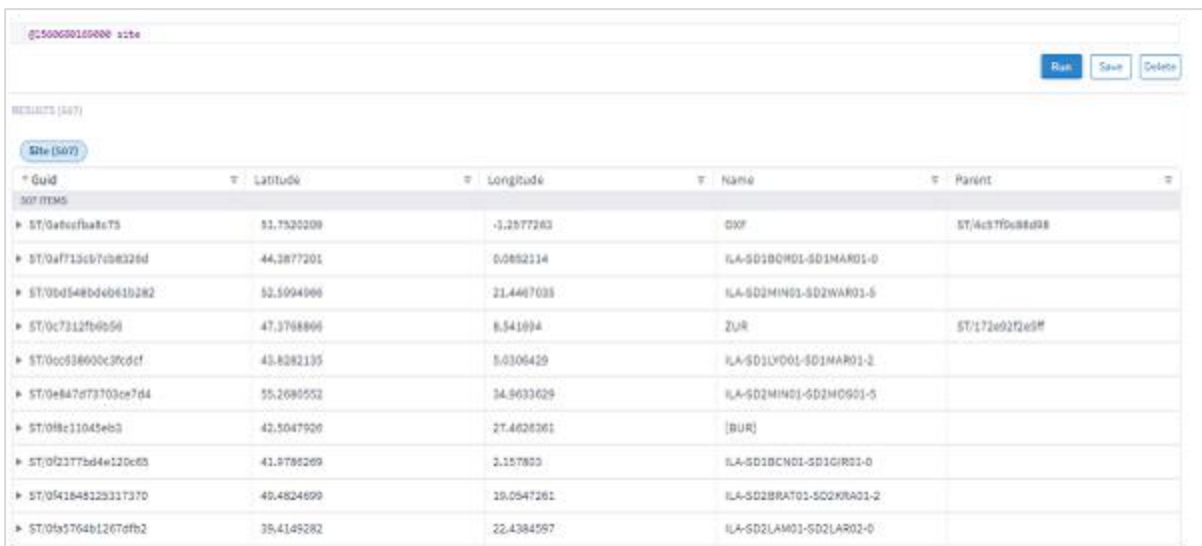
さまざまな時間形式を使用できます。

- 絶対時間 : @2019-05-10 10:00:00
- '-[0-9]+[ymwdHMS]: @-10H の形式の相対時間
- Unix タイムスタンプ (ミリ秒) : @1558610956000

```
@1560680169000 site
```

図 40 :
@ およびエポックタイムスタンプ

図 41 は結果を示しています。



Guid	Latitude	Longitude	Name	Parent
ST/0a6c0fa8c75	51.7520209	-1.2577263	DXF	ST/4c37f0a8d98
ST/0af713c670b8326d	44.3877201	0.0852114	ILA-S01BOR01-S01MAR01-0	
ST/0bd548bd4b610282	52.5094066	21.4467035	ILA-S02MIN01-S02WAR01-5	
ST/0c7312fb0b5d	47.3768866	8.541694	ZUR	ST/172e92f2e9f
ST/0cc53600c3fcdcf	45.8282135	5.0306429	ILA-S01LV001-S01MAR01-2	
ST/0e847d73703ce7d4	55.2680552	14.8633629	ILA-S02MIN01-S02MCG01-5	
ST/088c11045eb3	42.5047500	27.4628361	[BUR]	
ST/02277bd4e120c65	41.8780269	2.157835	ILA-S01BCN01-S01GIR01-0	
ST/041848125317370	49.4824699	19.0547261	ILA-S02BRAT01-S02KRA01-2	
ST/093764b1267dfb2	35.4149282	22.4384597	ILA-S02LAM01-S02LAR02-0	

図 41 :
タイムスタンプの結果

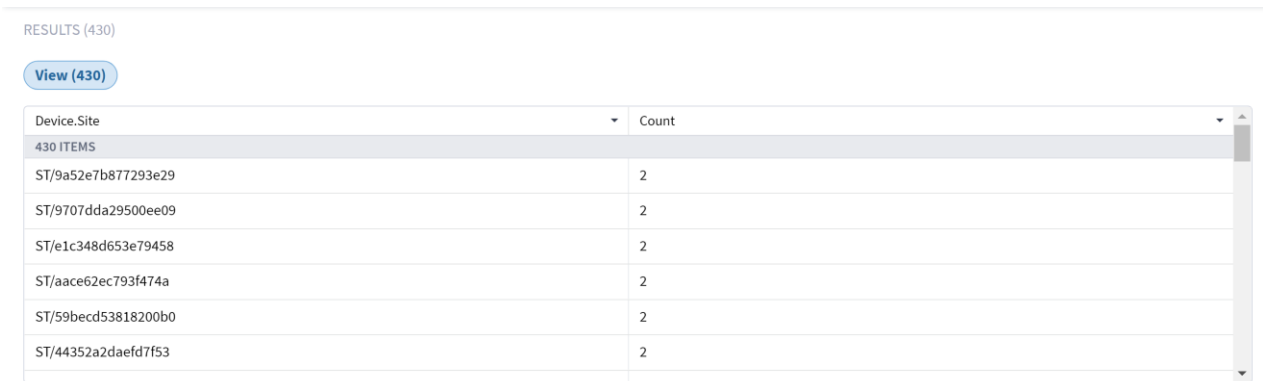
例 : group_by 関数

図 42 は、デバイスをサイトごとにグループ化する（およびそれらをカウントする）クエリコマンドを示しています。

```
port |group_by(.device.site)|asc(.count)
```

図 42 :
group_by

図 23 は結果を示しています。



RESULTS (430)

View (430)

Device.Site	Count
430 ITEMS	
ST/9a52e7b877293e29	2
ST/9707dda29500ee09	2
ST/e1c348d653e79458	2
ST/aace62ec793f474a	2
ST/59becd53818200b0	2
ST/44352a2daefd7f53	2

図 43.
group_by の結果

履歴関数

Crosswork Hierarchical Controller は、ネットワークインベントリとトポロジのすべての変更を記録します。変更は、イベントのリストとしてイベント履歴テーブルに保存されます。イベントは、リソースの追加 (ADD)、削除 (DELETE)、または属性変更 (UPDATE) の記録です。イベントは、モデル内のオブジェクトへの指数として GUID を使用してモデルにリンクされます。イベントテーブルには、時間、リソース ID、オブジェクトタイプ、イベントタイプ、変更前後の値など、変更のすべての詳細が含まれます。

独自の実装が選択されており、返される結果には、選択された時間範囲のある時点でモデル内にあったすべてのリソースが含まれます。そのため、後で削除されたサービスやリンクも検出されることが保証されています。たとえば、特定のタグが付いたすべてのサービスに関する過去 2 週間のイベントを取得するようにテーブルをクエリすると、結果には期間終了時に存在しなくなったサービスを含め、関連サービスが含まれます。詳細については、『*Crosswork Hierarchical Controller Network History Guide*』を参照してください。

標準の SHQL クエリを使用してモデルをフィルタ処理し、パイプ (|) を追加して履歴テーブルをフィルタ処理するクエリを作成できます。履歴テーブルは、すべてのエントリ属性でフィルタ処理できます。

- 変更されたオブジェクト GUID
- 変更されたオブジェクトタイプ
- 変更タイプ (ADD、UPDATE、DELETE)
- 変更前の値
- 変更後の値

結果はイベント履歴テーブルから返されます (一方、通常の SHQL クエリはオブジェクトテーブルから結果を返します)。

例 : history

図 44 は、2 つの日付間のリンクのイベントを取得するために作成されたクエリコマンドを示しています。たとえば、UPDATE イベントのみを表示したり、ダウンしたリンクを検索したりするなど、履歴テーブルのオプションを追加してクエリ結果を絞り込むことができます。

```
@-18d:-17d link | history[.action="UPDATE"]
```

```
@-21d:-13d link | history[.to.operStatus = "DOWN"]
```

図 44 :
@ および history

図 45 は結果を示しています。

Action	Changes	ObjGuid	ObjType	Timestamp
INSERT		LI/r_logical/f38377d02bd88162/69f91c6c31f4ce6e/5d41...	link	1612994572113
INSERT		LI/fiber_segment/f158ea32d21c2f1d	link	1612994572113
INSERT		LI/r_logical/5ed7163adf3b8d5e/7d82458f24bfa3/5d41...	link	1612994572113
INSERT		LI/r_logical/318d05a50856bb78/69f91c6c31f4ce6e/3460...	link	1612994572113
INSERT		LI/fiber_segment/d5ce814095d788f2	link	1612994572113
INSERT		LI/r_logical/3fe1f91b97e054c1/51308df752c1574/09ac...	link	1612994572113
INSERT		LI/fiber_segment/4e58ae4e5eb6e26	link	1612994572113
INSERT		LI/fiber_segment/abc10f48d59c843e	link	1612994572113
INSERT		LI/fiber_segment/3f493e21c7fceb90	link	1612994572113
INSERT		LI/fiber_segment/c06826ea9b8bd8f5	link	1612994572113
INSERT		LI/fiber_segment/24c685e0fe2a140a	link	1612994572113
INSERT		LI/fiber_segment/09678b7b96050d7	link	1612994572113
INSERT		LI/fiber_segment/21dach7f325008be	link	1612994572113
INSERT		LI/fiber_segment/a7c4c8814fd443ef	link	1612994572113
INSERT		LI/fiber_segment/3cd11f8b650f941a	link	1612994572113
INSERT		LI/fiber_segment/99a33043b4518d66	link	1612994572113
INSERT		LI/fiber_segment/cb0a28a1b9cc8e1	link	1612994572113
INSERT		LI/fiber_segment/f81b779794a4a4700	link	1612994572113

図 45
history の結果

その他の出力関数

以下の表の関数を追加して、結果を取得して特定の順序で表示することができます。通常、これらの関数はクエリコマンドの最後に追加されます。クエリ結果の特定のプロパティを表示することもできます。

関数の順序は重要であり、次の順序にする必要があります。

- fibre talk set > view > operands > asc/desc > limit > after

次に例を示します。

- このクエリは無効です : port | limit(10) | asc(.name)
- このクエリは有効です : port | asc(.name) | limit(10)

出力関数の前にはパイプ (|) が付きます。

表 7. 出力関数

機能	説明
add_counters	指定したオブジェクトタイプの属性値ごとの総数を表示します。
after (GUID)	指定された GUID を持つアイテムに続く結果のみを表示します。

機能	説明
asc (column)	結果を自然な昇順で表示します。
desc (column)	結果を自然な降順で表示します。
group_by	
imit(#)	表示される結果の数を制限します。
view	クエリ結果として指定されたプロパティ（および提供されたラベル）を表示します。

例 : after

図 46 は、指定された GUID を持つアイテムに続くアイテムのみを取得するために構築されたクエリを示しています。

```
inventory_item | after("IN/Router/0529f67055b6efe0")
```

図 46 :
after の出力関数

図 47 は結果を示しています。

Router (80)	Guid	Type	Name	Provider	Extra	ManagementIp	ReachabilitySta	Site	Tags	Vendor
80 ITEMS	IN/Router/0a22a0ec157648e1	ROUTER	CR1.TLV	Topogen	{'is_core': True}	10.40.0.67	REACHABLE	ST/c8037c4ef...	{'Vendor': ['Ci...	Cisco
	IN/Router/14bc8dcea3a57f18	ROUTER	CR2.MAD	Topogen	{'is_core': True}	10.40.0.38	REACHABLE	ST/d5e070604...	{'Vendor': ['Ci...	Cisco
	IN/Router/1877a9950429216f	ROUTER	ER1.WAR	Topogen	{'is_core': True}	10.40.0.83	REACHABLE	ST/ec803a477...	{'Vendor': ['H...	Huawei
	IN/Router/19694a81a1d0ebed	ROUTER	CR1.WAR	Topogen	{'is_core': True}	10.40.0.28	REACHABLE	ST/ec803a477...	{'Vendor': ['Ci...	Cisco
	IN/Router/1c8ae4011529613d	ROUTER	ER1.BCN	Topogen	{'is_core': False}	10.40.0.51	REACHABLE	ST/ba7d24f4b...	{'Vendor': ['Ci...	Cisco
	IN/Router/1df8ab6998bf7fc8	ROUTER	CR1.STO	Topogen	{'is_core': True}	10.40.0.77	REACHABLE	ST/3bafd5003...	{'Vendor': ['Ci...	Cisco
	IN/Router/2023fd21fb812871	ROUTER	CR1.ATH	Topogen	{'is_core': True}	10.40.0.47	REACHABLE	ST/7aac1877c...	{'Vendor': ['Ci...	Cisco
	IN/Router/2f29e956db0f0cae	ROUTER	ER1.VAL	Topogen	{'is_core': False}	10.40.0.62	REACHABLE	ST/ac13481fe...	{'Vendor': ['Ci...	Cisco
	IN/Router/3052d71050f39870	ROUTER	CR1.HEL	Topogen	{'is_core': True}	10.40.0.80	REACHABLE	ST/2499da6f5...	{'Vendor': ['Ci...	Cisco
	IN/Router/318dd5a50856bb78	ROUTER	CR1.VAL	Topogen	{'is_core': True}	10.40.0.63	REACHABLE	ST/ac13481fe...	{'Vendor': ['Ci...	Cisco

図 47 :
after の出力結果

例 1 : add_counters

図 48 は、各リンクレイヤの総数を取得するために作成されたクエリを示しています。クエリに | limit (0) を追加することで、応答がカウンタのみに制限されます。

```
link | add_counters (.layer) | limit (0)
```

図 48 :
add_counters 関数

図 49 は結果を示しています。

Attribute Name	Attribute Value	Counter
layer	ODU	151
layer	FIBER_SEGMENT	19380
layer	LSP	1656
layer	OTS	460
layer	FIBER	432
layer	OTU	145
layer	R_LOGICAL	233
layer	IGP	118
layer	OMS	138
layer	OCH	89
layer	L3_VPN	8
layer	ETH	175
layer	R_AGGREGATE	3
layer	R_PHYSICAL	124

図 49 :
add_counters の結果

例 2: add_counters

図 50 は、各リンクレイヤの総数と、各動作ステータス (UP、DOWN、または N_A) のリンクの総数を取得するために作成されたクエリを示しています。クエリに `| limit (0)` を追加することで、応答がカウンタのみに制限されます。

```
link | add_counters (.operStatus, .layer) | limit (0)
```

図 50 :
add_counters 関数

図 51 は結果を示しています。

Attribute Name	Attribute Value	Counter
operStatus	N_A	8
operStatus	DOWN	26
operStatus	UP	23078
layer	ODU	151
layer	FIBER_SEGMENT	19380
layer	LSP	1656
layer	OTS	460
layer	FIBER	432
layer	OTU	145
layer	R_LOGICAL	233
layer	IGP	118
layer	OMS	138
layer	OCH	89
layer	L3_VPN	8
layer	ETH	175
layer	R_AGGREGATE	3
layer	R_PHYSICAL	124

図 51 :
add_counters の結果

例 : view

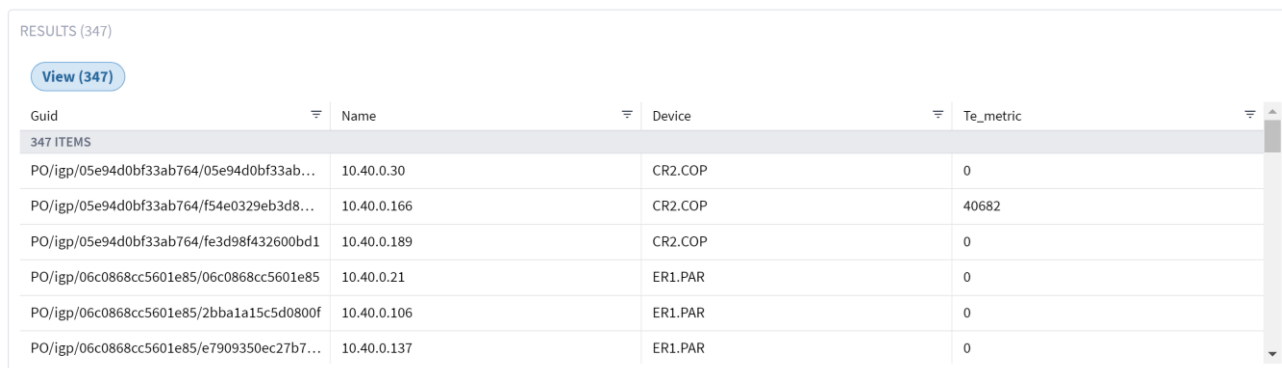
図 52 は、IGP ポートのポート GUID、名前、device.name、teMetric を取得するために作成されたクエリを示しています。

```
port[.type="IGP"]|view ("guid":.guid, "name":.name, "device":.device.name, "te_metric":.teMetric)
```

図 52 :

view 出力関数

図 53 は結果を示しています。



Guid	Name	Device	Te_metric
347 ITEMS			
PO/igp/05e94d0bf33ab764/05e94d0bf33ab...	10.40.0.30	CR2.COP	0
PO/igp/05e94d0bf33ab764/f54e0329eb3d8...	10.40.0.166	CR2.COP	40682
PO/igp/05e94d0bf33ab764/fe3d98f432600bd1	10.40.0.189	CR2.COP	0
PO/igp/06c0868cc5601e85/06c0868cc5601e85	10.40.0.21	ER1.PAR	0
PO/igp/06c0868cc5601e85/2bba1a15c5d0800f	10.40.0.106	ER1.PAR	0
PO/igp/06c0868cc5601e85/e7909350ec27b7...	10.40.0.137	ER1.PAR	0

図 53 :

view の出力結果

例 2 : view

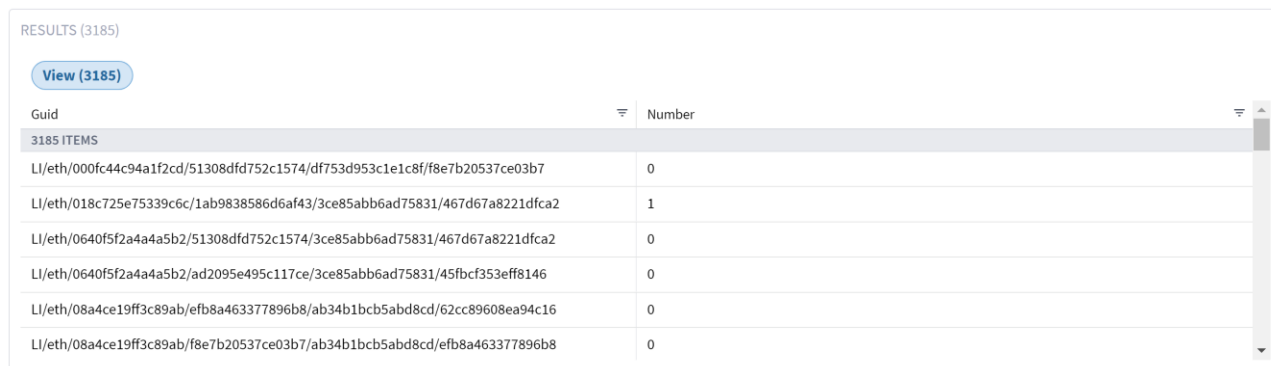
図 54 は、リンクが FIBER または FIBER_SEGMENT レイヤにない場合に、リンクの GUID とリンク内のパス数を取得するために作成されたクエリを示しています。

```
link[.layer not in ("FIBER", "FIBER_SEGMENT")] | view ("guid": .guid, "number": count(.paths))
```

図 54 :

view 出力関数

図 55 は結果を示しています。



Guid	Number
3185 ITEMS	
LI/eth/00fc44c94a1f2cd/51308dfd752c1574/df753d953c1e1c8f/f8e7b20537ce03b7	0
LI/eth/018c725e75339c6c/1ab9838586d6af43/3ce85abb6ad75831/467d67a8221dfca2	1
LI/eth/0640f5f2a4a4a5b2/51308dfd752c1574/3ce85abb6ad75831/467d67a8221dfca2	0
LI/eth/0640f5f2a4a4a5b2/ad2095e495c117ce/3ce85abb6ad75831/45fbcf353eff8146	0
LI/eth/08a4ce19ff3c89ab/efb8a463377896b8/ab34b1bcb5abd8cd/62cc89608ea94c16	0
LI/eth/08a4ce19ff3c89ab/f8e7b20537ce03b7/ab34b1bcb5abd8cd/efb8a463377896b8	0

図 55 :

view の出力結果

例 3 : view

図 56 は、リンクが FIBER または FIBER_SEGMENT レイヤにない場合に、リンクの GUID とポート名を取得するために作成されたクエリを示しています。出力は 20 エントリに制限されています。

```
link[.layer not in ("FIBER", "FIBER_SEGMENT")] | view("g": .guid, "n-a": .portA.name, "n-b": .portB.name)|limit(20)
```

図 56 :

view 出力関数

図 57 は結果を示しています。

G	N-A	N-B
20 ITEMS		
LI/r_logical/f38377d02bd88162/69f91c6c31f4ce6e/5d4172c3965d428...	HundredGigE0/1/0/0	HundredGigE0/1/0/0
LI/r_logical/5ed7163adf3b8d5e/7d82f458ff24bfa3/5d4172c3965d428...	TenGigE0/0/0/8	TenGigE0/0/0/4
LI/r_logical/318dd5a50856bb78/69f91c6c31f4ce6e/3460e659c02aefc...	HundredGigE0/1/0/0	HundredGigE0/1/0/0
LI/r_logical/3fe1f91b97e054c1/51308dfd752c1574/09ac8d52042280e...	TenGigE0/0/0/5	TenGigE0/0/0/4
LI/r_logical/94b82c614c0402dc/51308dfd752c1574/f3b3b56ab45fd1...	TenGigE0/0/0/5	TenGigE0/0/0/4

図 57 :

view の出力結果

例 4 : view

図 58 は、指定されたパスのホップ数をカウントするために作成されたクエリを示しています。

```
path[.guid="PA/LI/fiber/0d9fe6e5f9bad3c5"] | view("g": .guid, "hops": count(.hops))
```

図 58 :

view 出力関数

図 59 は結果を示しています。

G	Hops
1 ITEM	
PA/LI/fiber/0d9fe6e5f9bad3c5	57

図 59 :

view の出力結果

複雑な例

複雑な例 1

次の図は、LSP が失敗する原因となる OMS リンクを取得する方法を示しています。

クエリコマンドには次のものが含まれます。

```
site[.name="DUB"] | inventory_item | port | link[.layer="LSP"] as w; site[.name="BEL"] | inventory_item | port | link[.layer="LSP" and .guid in w] | downward("OMS") | link[.layer="OMS" and .operStatus="DOWN"]
```

サイト DUB と BEL 間のすべての LPS を取得する

- | を使用して、あるオブジェクトタイプから別のオブジェクトタイプに変換します。
- エイリアス と in を使用して、リストを DUB と BEL 間のみに減らします。

OMS リンクまでの downward にし、動作ステータスでフィルタ処理します。

- 「downward」関数を使用します。
- 関数でレイヤ名を使用し (downward ("OMS"))、downward を特定のレベルまでに制限します。

```
site[.name="DUB"] | inventory_item | port | link[.layer="LSP"] as w; site[.name="BEL"] | inventory_item | port | link[.layer="LSP" and .guid in w] | downward("OMS") | link[.layer="OMS" and .operStatus="DOWN"]
```

図 60.

クエリの結果

Guid	Layer	Name	OperStatus	PathGroupType	Paths	PortA	PortB	ProtectionStatus	Provider	Role
LI/oms/af5e85ff...	OMS	SD1BKL01/1-2-5...	DOWN	SINGLE_PATH	[{"guid": "PA/oms...	PO/oms/af5e85...	PO/oms/9bf4b7...	N_A	Topogen	REGULAR

図 61.

クエリの結果

複雑な例 2

次の図は、OMS の現在のリストを取得する方法を示しています。

クエリコマンドには次のものが含まれます。

```
link[.name="CR1.DUB:CR1.BEL:lsp_0"] | downward | link[.layer="OMS"]
```

特定の LSP によって使用されるすべての OMS を取得する

- LSP を名前で取得します。
- LSP によって使用される OMS までの downward にします。

次のものを含むクエリコマンドとなります。

```
| link[.name="CR1.DUB:CR1.BEL:lsp_0"] | downward | link[.layer="OMS"]
```

図 62.
クエリコマンド

Guid	Layer	Name	OperStatus	PathGroupType	Paths	PortA	PortB	ProtectionStatus	Provider	Role
12 ITEMS										
LI/oms/4afa911...	OMS	SD1DUB01/1-2-...	UP	SINGLE_PATH	[{"guid": "PA/om...	PO/oms/4afa91...	PO/oms/08a4ce...	N_A	Topogen	REGULAR
LI/oms/7c54f85...	OMS	SD1LIV01/1-2-5...	UP	SINGLE_PATH	[{"guid": "PA/om...	PO/oms/7c54f8...	PO/oms/ab34b...	N_A	Topogen	REGULAR
LI/oms/af5e85ff...	OMS	SD1BKL01/1-2-...	DOWN	SINGLE_PATH	[{"guid": "PA/om...	PO/oms/af5e85...	PO/oms/9bf4b7...	N_A	Topogen	REGULAR
LI/oms/aaba827...	OMS	SD1OXF01/1-3-...	UP	SINGLE_PATH	[{"guid": "PA/om...	PO/oms/aaba82...	PO/oms/9bf4b7...	N_A	Topogen	REGULAR
LI/oms/9c601df...	OMS	SD1BIR01/1-4-5...	UP	SINGLE_PATH	[{"guid": "PA/om...	PO/oms/9c601d...	PO/oms/aaba82...	N_A	Topogen	REGULAR
LI/oms/9c601df...	OMS	SD1BIR01/1-2-5...	UP	SINGLE_PATH	[{"guid": "PA/om...	PO/oms/9c601d...	PO/oms/7c54f8...	N_A	Topogen	REGULAR
LI/oms/9c601df...	OMS	SD1BIR01/1-3-5...	UP	SINGLE_PATH	[{"guid": "PA/om...	PO/oms/9c601d...	PO/oms/394fa7...	N_A	Topogen	REGULAR
LI/oms/644a90f...	OMS	SD1MAN01/1-3-...	UP	SINGLE_PATH	[{"guid": "PA/om...	PO/oms/644a90...	PO/oms/394fa7...	N_A	Topogen	REGULAR
LI/oms/df753d9...	OMS	SD1BEL01/1-2-5...	UP	SINGLE_PATH	[{"guid": "PA/om...	PO/oms/df753d...	PO/oms/ae10e0...	N_A	Topogen	REGULAR
LI/oms/9e79f8d...	OMS	SD1IOM101/1-3-...	UP	SINGLE_PATH	[{"guid": "PA/om...	PO/oms/9e79f8...	PO/oms/ae10e0...	N_A	Toooeen	REGULAR

図 63.
クエリの結果

時間の経過に伴う LSP パスの変更を追跡する場合は、たとえば @-1d を使用できます。

```
@-1d link[.name="CR1.DUB:CR1.BEL:lsp_0"] | downward | link[.layer="OMS"]
```

図 64.
クエリコマンド

複雑な例 3

次の図は、2 つの IP リンク間の共有リスクを見つける方法を示しています。

クエリコマンドには次のものが含まれます。

```
link[.name="10.40.0.157 to 10.40.0.158"] | downward | link[.layer="OMS"] as p; link[.name="10.40.0.154 to 10.40.0.153"] | downward | link[.layer="OMS"] as w; link[.guid in p and .guid in w]
```

最初の IP 論理リンクの OMS を取得する

- リンク名でフィルタ処理し、その OMS までの downward として、エイリアスを p とします。

2 番目の IP 論理リンクの OMS を取得する

- リンク名でフィルタ処理し、その OMS までの downward として、エイリアスを w とします。

すべての共有 OMS リンクを取得する

- 両方のグループのリンクを取得します。

```
link[.name="10.40.0.157 to 10.40.0.158"] | downward | link[.layer="OMS"] as p; link[.name="10.40.0.154 to 10.40.0.153"] | downward | link[.layer="OMS"] as w; link[.guid in p and .guid in w]
```

図 65.

クエリコマンド

The screenshot shows a SHQL query interface. At the top, it says "SHQL" and "Records fetched at: 20:06:32 06-13-2020 UTC". Below that is a "Saved Queries" dropdown. The query text is: `link[.name="10.40.0.157 to 10.40.0.158"] | downward | link[.layer="OMS"] as p; link[.name="10.40.0.154 to 10.40.0.153"] | downward | link[.layer="OMS"] as w; link[.guid in p and .guid in w]`. There are buttons for "Run", "Save As", and "Delete Query". Below the query is a "RESULTS (1)" section with a table. The table has columns: Guid, Layer, Name, OperStatus, PathGroupType, Paths, PortA, PortB, ProtectionStatus, Provider, Role. The first row is highlighted and contains: LI/oms/1722e5a..., OMS, SD1FRA01/1-7-5..., UP, SINGLE_PATH, [[guid: 'PA/oms...', PO/oms/1722e5..., PO/oms/271610..., N_A, Topogen, REGULAR.

図 66.

クエリの結果

複雑な例 4

次の図は、サイト内のサービスに向けた空きポートを見つける方法を示しています。

次のものを含むクエリコマンドとなります。

```
site[.name contains "MIL"] | inventory_item | port[.type="ETH"] | link | port as p; site[.name contains "MIL"] | inventory_item | port[.type="ETH" and .guid not in p]
```

リンクで使用される特定のサイトのすべての ETH ポートを取得する

- サイトのポートを取得し、リンクに変換してからポートに変換し、エイリアスを p とします。

リンクで使用されていないすべてのポートを取得する。

- サイトのポートを取得し、ポートに変換し、p で表示されるものをフィルタで除外します。

```
site[.name contains "MIL"] | inventory_item | port[.type="ETH"] | link | port as p; site[.name contains "MIL"] | inventory_item | port[.type="ETH" and .guid not in p]
```

図 67.

クエリコマンド

SHQL Records fetched at: 20:33:40 06-12-2020 UTC

Saved Queries

```
site[.name contains "MIL"] | inventory_item | port[.type="ETH"] | link | port as p; site[.name contains "MIL"] | inventory_item | port[.type="ETH" and .guid not in p]
```

Run Save As Delete Query

RESULTS (2)

Ethernet Port (2)

Guid	Type	AdminStatus	Desc	Device	Name	OperStatus	Provider	RelDirection	EthPortType	SpeedBps
PO/eth/5979a21...	ETH	UP	mil-eth-2	IN/ROADM/5979...	1-1-6	UP	Topogen	NONE	ETH_10G	10000000000
PO/eth/5979a21...	ETH	UP	mil-eth-1	IN/ROADM/5979...	1-1-5	UP	Topogen	NONE	ETH_10G	10000000000

図 68. クエリの結果

実際に試す

独自の複雑なクエリを作成してみてください。

図 69 は、以下を含む例を示しています。

- オペランド
- 変換
- コレクション
- 機能

```
site [.guid = "ST/9354b4a138dd_0_1_2"] | downward | inventory_item | port | link | upward | link[.layer in ("LSP", "E_LINE")] | limit(20) & port & inventory_item & site| upward
```

図 69 : 複雑なクエリ

表 8. 複雑なクエリの内容

項目	タイプ	参照先
site	オブジェクトタイプ	オブジェクトのタイプとプロパティ
[.guid = "ST/31105c202abd"]	条件 (オペランド =)	オペランド
downward	機能	SHQL 関数
inventory_item	変換オブジェクトタイプ	オブジェクトタイプの変換/収集
port	変換オブジェクトタイプ	オブジェクトタイプの変換/収集
link	変換オブジェクトタイプ	オブジェクトタイプの変換/収集
upward	機能	SHQL 関数

項目	タイプ	参照先
link[.layer in ("LSP", "E_LINE")]	変換オブジェクトタイプ + 条件 (オペランド in)	オブジェクトタイプの変換/収集
limit(20)	機能	出力関数
& port & inventory_item & site	オブジェクトタイプのコレクション	オブジェクトタイプの変換/収集
upward	機能	SHQL 関数

図 70 は結果を示しています。

site [.guid = "ST/9354b4a138dd_0_1_2"] | downward | inventory_item | port | link | upward | link[.layer in ("LSP", "E_LINE")] | limit(20) & port & inventory_item & site| upward

Run Save As Delete Query

RESULTS (58)

Site (20) Router (9) IGP Port (9) LSP Link (20)

Guid	Latitude	Longitude	Name	Parent
20 ITEMS				
ST/a6da17cd217d	51.2277411	6.7734556	DUS	ST/a6da17cd217d_0
ST/9354b4a138dd	53.4083714	-2.9915726	LIV	ST/9354b4a138dd_0
ST/839c2694c73e	43.3619145	-5.8493887	OVE	ST/839c2694c73e_0
ST/a9bd23d3ed70	54.597285	-5.93012	BEL	ST/a9bd23d3ed70_0
ST/ea4a371e85b7	53.3498053	-6.2603097	DUB	ST/ea4a371e85b7_0
ST/9354b4a138dd_0	53.4083714	-2.9915726	LIV	ST/9354b4a138dd_0_1
ST/a6da17cd217d_0	51.2277411	6.7734556	DUS	ST/a6da17cd217d_0_1
ST/839c2694c73e_0	43.3619145	-5.8493887	OVE	ST/839c2694c73e_0_1
ST/a9bd23d3ed70_0	54.597285	-5.93012	BEL	ST/a9bd23d3ed70_0_1
ST/ea4a371e85b7_0	53.3498053	-6.2603097	DUB	ST/ea4a371e85b7_0_1

図 70 :
複雑なクエリの結果

タイムマシンを使う

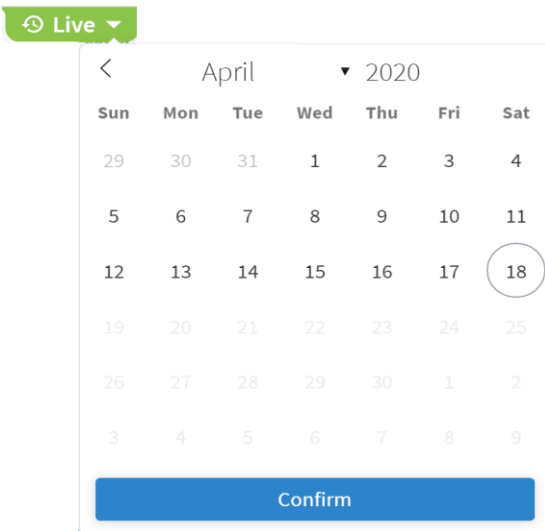
タイムマシンは、過去のある日付のネットワークの状態のスナップショットを提供します。このモードでは、すべてのアプリケーションに、この時点に適用されるデータと分析が反映されます。

タイムマシンを使用して、過去の日付と同じようにモデルをクエリできます。

モデルの日付を変更するには以下を実行します。

1. Crosswork Hierarchical Controller のアプリケーションバーで、[SHQL (SHQL)] を選択します。

2. [ライブ (Live)]をクリックし、日付を選択して [確認 (Confirm)]をクリックします。



3. 必要なクエリを実行します。

オブジェクトタイプのプロパティ

次のセクションでは、異なるオブジェクトタイプに対応するすべてのプロパティの一覧を示します。

Inventory_item プロパティ

表 9. Inventory_item プロパティ

Inventory_item プロパティ
.accessIdentifier
.cardStatus
.cardType
.children
.childrenPorts
.desc
.deviceFamily
.deviceType
domainName
.equipState
.extra
.formFactor
.guid
.hasRoadm

Inventory_item プロパティ

.id
igpType
.isisInfo
.isisInfo.areaAddresses
.isisInfo.instanceIdentifier
.isisInfo.isDiscoveredL1
.isisInfo.isDiscoveredL2
.isisInfo.systemId
.isisInfo.topologyIdentifier
.managementIp
.modelName
.name
.oui
.parent
.partNumber
.pluggability
.provider
.reachabilityStatus
routerId
.serialNumber
.site
.softwareVersion
.supportedLambdas
.supportedPhysicalLayers
.supportedReachabilityMeters
.supportedSpeeds
.tags
.topologyId
.tunableType
.tunedLambda
.type
.vendor

port プロパティ

表 10. 表 1 : port プロパティ

port プロパティ
.adminGroups
.adminGroups.groupNumber
.adminGroups.name
.adminStatus
.aggRateBps
.ceInterfacelp
.cePeRoutingProtocols
.channelNum
.childRole
.containingService
.desc
.device
.egressBandwidthProfile
.egressBandwidthProfile.cbsKb
.egressBandwidthProfile.cir
.egressBandwidthProfile.ebsKb
.egressBandwidthProfile.eir
.egressBandwidthProfile.qosPolicy
.ethPortType
.evcEgressBandwidthProfile
.evcEgressBandwidthProfile.cbsKb
.evcEgressBandwidthProfile.cir
.evcEgressBandwidthProfile.ebsKb
.evcEgressBandwidthProfile.eir
.evcEgressBandwidthProfile.qosPolicy
.evcIngressBandwidthProfilePerEvc
.evcIngressBandwidthProfilePerEvc.cbsKb
.evcIngressBandwidthProfilePerEvc.cir
.evcIngressBandwidthProfilePerEvc.ebsKb

port プロパティ

.evcIngressBandwidthProfilePerEvc.eir
.evcIngressBandwidthProfilePerEvc.qosPolicy
.exportRts
.extra
.guid
.hasServiceEndPointCapability
.id
.ifIndex
.igpInstanceName
.igpMetric
.importRts
.ingressBandwidthProfile
.ingressBandwidthProfile.cbsKb
.ingressBandwidthProfile.cir
.ingressBandwidthProfile.ebsKb
.ingressBandwidthProfile.eir
.ingressBandwidthProfile.qosPolicy
.ipAddress
.isLoopback
.isManagement
.isRouterId
.isStatsDummy
.isisInfo
.isisInfo.isisLevel
.isisInfo.netPrefixMetric
.isisInfo.portType
.lambdaNm
.links
.lowerPorts
.mappedCeTags
.mappedSvlan
mappingType

port プロパティ

.maxLinkBandwidth

.maxReservableLinkBandwidth

.mediaFrameType

.memberCount

.mtuSize

.name

.nonPrimaryIps

.ocType

.oduld

.oduPolarization

.oduType

.operStatus

ospfInfo

.otuType

.packetOpticalDetails

.packetOpticalDetails.availableBandwidth

.parent

.pdhType

.physicalAddress

.portCapabilities

.primaryIp

.protectionType

.provider

.range

.relDirection

.relPort

.rfParams

.routeDistinguisher

.rsvpReservedBandwidthBps

.rsvpStaticBandwidthBps

.spanLoss

.speedBps

port プロパティ

.srPrefixSids

.srlgs

.stsType

.tags

.teMetric

.type

.unreservedLinkBwForPriorityIndex

.upperPorts

.vcatsType

.vlan

.vpnRole

.vrfName

.vtType

.ycableType

link プロパティ

表 11. link プロパティ

link プロパティ

.activeProtectionPriority

.adjacencySegmentInfo

.adminGroupConstraints

.bindingSid

.candidatePaths

.color

.deploymentType

.desc

.distanceMeters

.desc

.evcPerformanceProfile

.extra

.geometry

link プロパティ

.guid

.holdPriority

.id

.inverseLinkId

.isApproximated

.isBidi

.isCeVlanCosPreservation

.latencyMicros

.layer

.lineType

.lspTechnology

.mappingStatus

.name

.numberOfFibers

.operStatus

.owner

.pathGroupType

.paths

.portA

.portB

.prefixSegmentInfo

.protectedLayer

.protectionStatus

.provider

.role

.rxLabel

.segmentType

.setupPriority

.sidIndex

.sidLabel

.signalType

.speedBps

link プロパティ

.srAdjacencySids
.srlgs
.tags
.teMetric
.txLabel
.usedByHops
.vclId

site プロパティ

表 12. site プロパティ

site プロパティ

.extra
.guid
.id
.latitude
.longitude
.name
.parent

service プロパティ

表 13. service プロパティ

service プロパティ

.adminStatus
.anyToAnyOrHubRouteTarget
.containedPorts
.customerDetails
.customerName
.desc
.extra
.guid

service プロパティ
.id
.name
.operStatus
.provider
.serviceIntent
.serviceIntentRelation
.spokeRouteTarget
.tags
.type
.vpnTopology

service_intent プロパティ

表 14. service_intent プロパティ

service_intent プロパティ
.adminState
.anyToAnyDetails
.anyToAnyDetails.minimumSites
.anyToAnyDetails.routeTarget
.baseTemplate
.customerName
.dataForNonMultiplexedService
.dataForNonMultiplexedService.ethPortType
.deploymentInfo.lastStatusChangeTimeStamp
.deploymentInfo.operation
.deploymentInfo.phase
.desc
.extra
.guid
.holdPriority
.hubAndSpokeDetails
.hubAndSpokeDetails.hubRouteTarget
.hubAndSpokeDetails.minimumHubSites

service_intent プロパティ

.hubAndSpokeDetails.minimumSpokeSites
.hubAndSpokeDetails.spokeRouteTarget
.id
.ipAddressAllocationPolicy
.isOamPmCollectionEnabled
.isRemoved
.isTemplate
.name
.oduType
.pathComputationSettings
.pathComputationSettings.backupPathOptimization
.pathComputationSettings.includeLinkStates
.pathComputationSettings.mainPathOptimization
.protectionSettings
.protectionSettings.protectionPolicy
.protectionSettings.resourceDiversityFor1Plus1
.protectionSettings.resourceDiversityFor1Plus1.diversifiedResources
.protectionSettings.resourceDiversityFor1Plus1.diversityPolicy
.provider
.qosSettings
.qosSettings.cbsKb
.qosSettings.cir
.qosSettings.ebsKb
.qosSettings.eir
.qosSettings.qosPolicy
.resourceAllocationPolicy
.routeDistinguisher
.setupPriority
.srData
.tunnelType
.type
.underlayTech

service_intent プロパティ
.underlayTech.allowedNni
.underlayTech.ipMpls
.underlayTech.ipMpls.allowedNni
.underlayTech.ipMpls.tunnelResilliency
.underlayTech.ipMpls.tunnelTypes
.underlayTech.mplsTp
.underlayTech.mplsTp.allowedNni
.underlayTech.mplsTp.tunnelResilliency
.underlayTech.mplsTp.tunnelUsageConstraints
.underlayTech.otn
.underlayTech.otn.allowedNni
.underlayTech.otn.serviceTunnelRate
.underlayTech.otn.serviceTunnelType
.underlayTech.selectedTech
.underlayTech.tunnelResilliency
.underlayTech.tunnelTypes
.underlayTech.wdm
.underlayTech.wdm.allowedNni
.vlanManipulation
.vpnTopology

service_intent_resource プロパティ

表 15. service_intent_resource プロパティ

service_intent_resource プロパティ
.constraintCompliance
.extra
.guid
.id
.includeType
.layer1Info
.layer1Info.tunnelRate

service_intent_resource プロパティ

.layer2Info
.layer2Info.mappedCeTags
.layer2Info.qos
.layer2Info.qos.cbsKb
.layer2Info.qos.cir
.layer2Info.qos.ebsKb
.layer2Info.qos.eir
.layer2Info.qos.qosPolicy
.layer3Info
.layer3Info.cePeSettings
.layer3Info.cePeSettings.bgpRoutingDetails
.layer3Info.cePeSettings.bgpRoutingDetails.autonomousSystem
.layer3Info.cePeSettings.bgpRoutingDetails.peeringIp
.layer3Info.cePeSettings.ospfRoutingDetails
.layer3Info.cePeSettings.ospfRoutingDetails.metric
.layer3Info.cePeSettings.ospfRoutingDetails.ospfArea
.layer3Info.cePeSettings.routingMethod
.layer3Info.cePeSettings.staticRoutingDetails
.layer3Info.cePeSettings.staticRoutingDetails.entries.adminPreference
.layer3Info.cePeSettings.staticRoutingDetails.entries.ipPrefix
.layer3Info.cePeSettings.staticRoutingDetails.entries.nextHop
.layer3Info.ipAddress
.layer3Info.I3VpnRole
.order
.protectionRole
.resource
.resourceDiversity
.serviceIntent
.tag
.type

event プロパティ

表 16. event プロパティ

event プロパティ
.count
.data
.guid
.lastUpdate
.machineld
.severity
.subType
.timeStamp
.type
.username

srlg Resource プロパティ

表 17. srlg プロパティ

srlg プロパティ
.externalId
.extra
.guid
.id
.name
.ordinal
.provider
.srlgs

path プロパティ

表 18. path プロパティ

path プロパティ
.ecmpWeight
.extra
.guid

path プロパティ

.hops

.id

.link

.priority

.provider

.srPathSegments

.tunPortA

.tunPortB

site_link プロパティ

表 19. site_link プロパティ

site_link プロパティ

.depth

.guid

.id

.isCrossLink

.layer

.links

.name

.siteA

.siteB

.status

.tags

.utilization

visual_site プロパティ

表 20. visual_site プロパティ

visual_site プロパティ

.depth

.guid

visual_site プロパティ

.id
.isOnlyAmplifiers
.latitude
.longitude
.name
.oneSize
.parent
.reachabilityStatus
.routerSize
.site
.tags

inventory プロパティ

表 21. inventory プロパティ

inventory プロパティ

.accessIdentifier
.cardStatus
.cardType
.children
.childrenPorts
.desc
.deviceFamily
.deviceType
.equipState
.extra
.formFactor
.guid
.hasRoadm
.isisInfo
.isisInfo.areaAddresses
.isisInfo.instanceIdentifier

inventory プロパティ

.isisInfo.isDiscoveredL1

.isisInfo.isDiscoveredL2

.isisInfo.isisInstanceName

.isisInfo.systemId

.isisInfo.topologyIdentifier

.managementIp

.modelName

.name

.oui

.parent

.partNumber

.pluggability

.ports

.provider

.reachabilityStatus

.routerId

.serialNumber

.site

.softwareVersion

.srCapabilities

.srlgs

.supportedLambdas

.supportedPhysicalLayers

.supportedReachabilityMeters

.supportedSpeeds

.tags

.topologyId

.tunableType

.tunedLambda

.type

.unitType

.vendor

srlg_risk_resource_mtm プロパティ

表 22. srlg_risk_resource_mtm プロパティ

srlg_risk_resource_mtm プロパティ
.guid
.id
.inventoryItem
.link
.port
.provider
.srlg

risk_resource プロパティ

表 23. risk_resource プロパティ

risk_resource プロパティ
.inventoryItem
.guid
.provider
.link
.srlg
.port

region プロパティ

表 24. region プロパティ

region プロパティ
.geometry
.guid
.id
.name
.overlay

history プロパティ

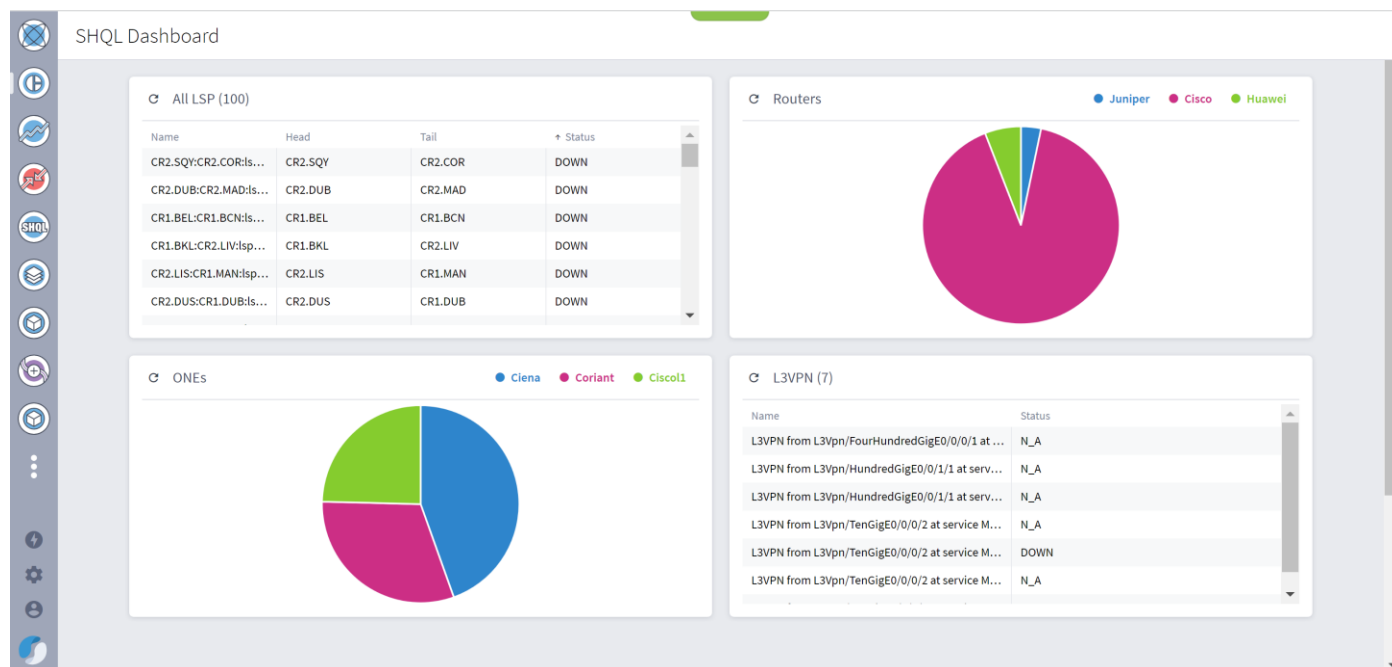
表 25. history プロパティ

history プロパティ
.action
.changes
.objGuid
.objName
.objSubtype
.objType
.timestamp

SHQL ウィジェットの作成

SHQL ウィジェットについて

カスタマイズされたウィジェットをすぐに作成できます。開発の手間やソフトウェアの配布は必要ありません。SHQL ダッシュボードアプリケーションを開くとウィジェットクエリが実行され、ウィジェットが表示されます。ウィジェットには、クエリを手動で実行するための更新ボタンもあります。



SHQL ウィジェットの作成

ウィジェットの属性は次のとおりです。

- **タイトル** : SHQL ダッシュボードに表示される SHQL ウィジェットの名前。
- **クエリ** : 5 ビューまでにする必要があり、制限 (100) があります。
- **ビジュアルモード** :
 - 円/棒 - クエリにカウンタのみが含まれている場合
 - グラフ - クエリにタイムスタンプとカウンタが含まれている場合
 - テーブル - クエリがリストを返す場合
- **「About」** テキスト。
- **オプション** : ウィジェットへのドリルダウンとして使用される追加のクエリ (自動的に生成可能で、カウンタと制限は削除されます) 。

SHQL ウィジェットを作成するには以下を実行します。

1. Crosswork Hierarchical Controller のアプリケーションバーで、[SHQL ウィジェット (SHQL Widgets)] を選択します。
2. [+新しい SHQL ウィジェット (+ New SHQL Widget)] をクリックします。

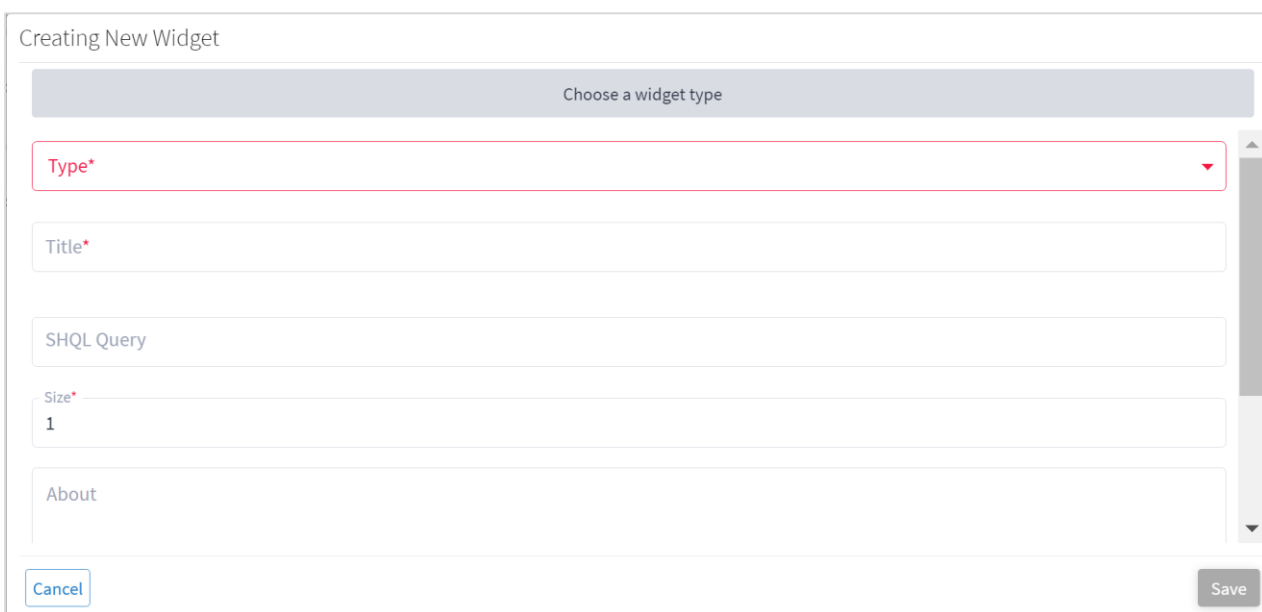


図 71.

新しいウィジェットの作成

3. [タイプ (Type)] を選択します。[テーブル (Table)]、[グラフ (Graph)]、または [円グラフ (Pie)] にすることができます。

選択したウィジェットのタイプに応じて、SHQL で許可される操作に関するガイダンスが提供されます。

付録

用語集

用語	説明
RESTCONF	ネットワーク構成プロトコル (NETCONF) で定義されたデータストアの概念を使用して、YANG バージョン 1 または YANG バージョン 1.1 で定義されたデータを構成するための HTTP に基づくプロトコル

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。

リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動 / 変更されている場合がありますことをご了承ください。

あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。

シスコ コンタクトセンター

自社導入をご検討されているお客様へのお問い合わせ窓口です。

製品に関して | サービスに関して | 各種キャンペーンに関して | お見積依頼 | 一般的なご質問

お問い合わせ先

お電話での問い合わせ

平日 9:00 - 17:00

0120-092-255

お問い合わせウェブフォーム

cisco.com/jp/go/vdc_callback



©2023 Cisco Systems, Inc. All rights reserved.

Cisco, Cisco Systems, および Cisco Systems ロゴは、Cisco Systems, Inc. またはその関連会社の米国およびその他の一定の国における商標登録または商標です。本書類またはウェブサイトに掲載されているその他の商標はそれぞれの権利者の財産です。「パートナー」または「partner」という用語の使用は Cisco と他社との間のパートナーシップ関係を意味するものではありません。(1502R) この資料の記載内容は2023年03月現在のものです。この資料に記載された仕様は予告なく変更する場合があります。



シスコシステムズ合同会社

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
cisco.com/jp