



# **SNMP** 통합 데이터를 수집하는 중입니다

## OnCommand Insight

NetApp  
October 24, 2024

# 목차

SNMP 통합 데이터를 수집하는 중입니다.....	1
통합 팩 .....	1
통합 파일 형식.....	1
SNMP 통합 팩을 가져오는 중입니다.....	2
SNMP 통합 데이터 소스 생성.....	3
integration.json 파일 정보 .....	5

# SNMP 통합 데이터를 수집하는 중입니다

통합 SNMP 데이터 소스를 사용하면 OnCommand Insight에서 일반 SNMP 데이터를 수집할 수 있습니다.

## 통합 팩

SNMP Integration 데이터 소스는 "통합 팩"을 사용하여 수집되는 통합 값과 이러한 값을 제공하는 SNMP 개체를 정의합니다.

통합 팩은 다음으로 구성됩니다.

- 특정 장치 유형(스위치, 라우터 등)의 SNMP 개체 측면에서 통합 페이로드 콘텐츠를 정의하는 JSON 구성 파일(integration.json)입니다.
- 통합 팩이 의존하는 MIB 파일 목록입니다.

통합 팩은 여러 데이터 유형을 정의할 수 있습니다. 예를 들어 RHEL 호스트를 통합할 때 가동 시간, 사용자 수, 실행 중인 프로세스 수와 같은 일반 시스템 정보에 대해 데이터 유형을 정의할 수 있으며, 메모리 및 파일 시스템 사용 데이터에 대해 두 번째 데이터 유형을 정의할 수 있습니다. 일반적으로 각 데이터 형식은 "플랫"이어야 하며 중첩된 데이터를 포함할 수 없습니다.

단일 통합 팩에서 24개 이상의 데이터 유형을 정의해서는 안 됩니다. Insight는 수집되는 통합 데이터의 양을 제한합니다. 1분 동안 24개 이상의 보고서를 수집하려고 시도하면 속도 오류가 발생합니다.

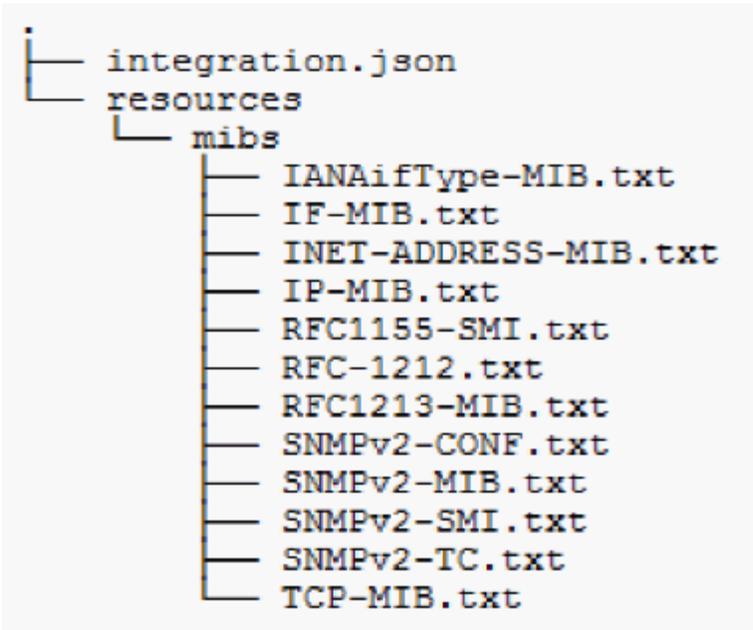
통합 유형의 이름은 다음 규칙을 준수해야 합니다.

- 이름은 \_, - 또는, + 문자로 시작할 수 없습니다
- 이름에는 #, \, /, \*, ?, ", <, >, |, ', ;
- UTF-8로 인코딩된 바이트 수가 100개를 초과할 수 없습니다
- 이름을 지정할 수 없습니다. 또는...

## 통합 파일 형식

통합 팩은 SNMP 개체 측면에서 통합 페이로드 콘텐츠를 정의하는 JSON 구성 파일(integration.json)이 포함된 ZIP 파일입니다. 또한 모든 MIB 파일과 해당 MIB 종속성이 포함된 MIBS 폴더도 포함되어 있습니다.

를 클릭합니다 integration.json 파일은 ZIP 파일의 최상위 수준에 있어야 하며 MIB 파일은 ZIP 내의 "resources/mibs" 하위 디렉토리에 있어야 합니다. 필요한 경우 ZIP 파일에는 "readme.txt"와 같은 파일도 포함될 수 있습니다. 통합 ZIP 구조의 예는 다음과 같습니다.



## SNMP 통합 팩을 가져오는 중입니다

웹 UI를 사용하여 SNMP 통합 팩을 OnCommand Insight로 가져옵니다. 통합 팩은 에 정의된 "통합 팩 이름" 값으로 식별됩니다 integration.json ZIP 파일에 포함된 구성 파일입니다.

### 시작하기 전에

OnCommand Insight 서버로 가져올 통합 팩이 포함된 올바른 형식의 ZIP 파일을 만들어야 합니다.

### 이 작업에 대해

다음 단계를 사용하여 SNMP 통합 팩을 Insight 서버로 가져옵니다.

### 단계

1. Admin \* > \* Setup \* > \* SNMP Integration \* 을 클릭합니다

SNMP 패키지 가져오기 화면이 표시됩니다.

#### Import SNMP package

⤵ Select file    No file selected    ⬇ Import

**Warning:** This will overwrite any conflicting package from existing database.

2. SNMP 패키지가 포함된 로컬 파일을 선택하려면 \* 파일 선택 \* 을 클릭합니다.

선택한 파일이 파일 상자에 표시됩니다.



같은 이름의 기존 통합 팩을 덮어씁니다.

3. 가져오기 \* 를 클릭합니다

파일을 Insight 서버로 가져옵니다.

## SNMP 통합 데이터 소스 생성

통합 SNMP 데이터 소스는 Brocade 및 Cisco용 OnCommand Insight 데이터 소스에 포함된 다른 SNMP 기반 데이터 소스와 유사한 공통 SNMP 구성 속성을 제공합니다.

### 시작하기 전에

Integration SNMP 데이터 소스를 성공적으로 사용하여 수집하려면 다음 조건이 참이어야 합니다.

- 이 SNMP 데이터 소스에 사용할 통합 팩을 이미 가져와야 합니다.
- 모든 타겟 디바이스는 동일한 자격 증명을 공유합니다.
- 모든 타겟 디바이스는 구성된 통합 팩이 참조하는 SNMP 객체를 구현합니다.

### 이 작업에 대해

SNMP Integration 데이터 소스를 생성하려면 데이터 소스 생성 마법사에서 공급업체 "통합"과 모델 "SNMP"를 선택합니다.

### 단계

1. OnCommand Insight 웹 UI에서 \* 관리자 \* > \* 데이터 소스 \* 를 클릭합니다
2. 추가 \* 를 클릭합니다
3. 데이터 원본의 이름을 입력합니다
4. 공급 업체의 경우 \* 통합 \* 을 선택합니다
5. 모델 에서 \* SNMP \* 를 선택합니다

## Add data source ✕

**Settings**

\*Name

Vendor Integration

Model SNMP

Where to run local

What to collect  Integration (BETA)

[Configure](#) ↓

**Configuration**

**Advanced configuration**

**Test**

Cancel
Save

6. 수집 대상 에 대해서는 \* 통합 \* 을 확인하십시오

이 데이터 원본의 유일한 패키지이며 기본적으로 선택되어 있습니다.

7. 구성 \* 을 클릭합니다

8. SNMP 데이터를 수집할 시스템의 IP 주소를 입력합니다

9. 가져온 SNMP 통합 팩을 선택합니다

10. 통합 폴링 간격을 설정합니다

11. SNMP 버전을 선택합니다

12. SNMP 커뮤니티 문자열을 입력합니다

SNMP V1 및 V2의 경우

13. 데이터를 수집할 시스템의 사용자 이름과 암호를 추가합니다.

SNMP V3의 경우

14. 고급 구성 \* 을 클릭합니다

고급 구성 기본 설정이 표시됩니다. 필요한 설정을 변경합니다.

## integration.json 파일 정보

integration.json 파일은 페이로드를 식별합니다.

다음 그림에서는 간단한 통합 .json 파일의 컬러 코딩 표현을 제공합니다. 첨부 테이블은 파일에 있는 오브젝트의 기능을 식별합니다.

```
{
  "integrationPacName": "WindowsSnmp",
  "description": "Generic integration for mibs supported by the default
SNMP Agent for Windows 2012, including HOST-RESOURCES",
  "acquisitionType": "SNMP",
  "integrationTypes": [
    {
      "integrationType": "snmp_win2012_host",
      "name": {
        "mibModuleName": "RFC1213-MIB",
        "objectName": "sysName"
      },
      "identifiers": {
        "hostname": {
          "mibModuleName": "RFC1213-MIB",
        }
      },
      "attributes": {
        "description": {
          "mibModuleName": "RFC1213-MIB",
          "objectName": "sysDescr"
        },
        "snmp_sys_obj_id": {
          "mibModuleName": "RFC1213-MIB",
          "objectName": "sysObjectID"
        }
      },
      "dataPoints": {
        "uptime": {
          "num": {
            "mibModuleName": "RFC1213-MIB",
            "objectName": "sysUpTime"
          }
        }
      }
    }
  ]
}
```

Blue	Reserved
Red	User customizable strings and IDs
Green	MIB names
Purple	MIB object
Black	JSON structure

## integration.json 파일 정보

각 필드에는 다음과 같은 특성이 있습니다.

- "식별자" 섹션은 Insight에서 새 "개체"를 만들기 위해 고유한 복합 키를 형성합니다
- "속성"은 객체에 대한 메타 데이터를 지원합니다.

이 두 경우 모두 해당 개체에 대한 최신 보고서 값(식별자로 식별)만 보존됩니다.

- "데이터 요소"는 시계열 데이터이며 숫자 값이어야 합니다. Insight는 여기에 보고된 각 값과 모든 값을 기본적으로 90일 동안 유지하고 타임 시리즈를 식별된 개체에 연결합니다.

## 수치 표현식

기본적으로 모든 값 식은 통합 페이로드에서 문자열로 보고됩니다. "식별자" 및 "특성"은 문자열 값만 정의할 수 있습니다. "datapoints"는 문자열 또는 숫자 값을 정의할 수 있습니다. 숫자 값은 다음 한정자 키 중 하나를 사용하여 정의됩니다.

- Num - 카운터가 마지막으로 초기화된 이후 수신된 총 바이트 수입니다
- Delta - 폴링 간격 동안 수신된 바이트 수입니다
- 속도 - 폴링 간격 동안의 평균 수신 속도(바이트/초)입니다

폴링 간격 동안 초당 메가바이트의 평균 수신 속도는 속도 및 수학 연산의 조합을 사용하여 수행할 수 있습니다

## 수학 연산

를 클릭합니다 `integration.json` 파일에서 지원되는 수학 연산은 더하기, 빼기, 곱하기, 나누기입니다. 다음 예제에서는 JSON 파일의 곱하기, 나누기 및 합계 작업을 보여 줍니다.

```

"network_utilization":
{
  "mult": [
    {
      "div": [
        {
          "sum": [
            "rate": {
              "mibModuleName": "IF-MIB",
              "objectName": "ifHCOutOctets",
              "comment": "bytes per second out"
            },
            "rate": {
              "mibModuleName": "IF-MIB",
              "objectName": "ifHCInOctets",
              "comment": "bytes per second in"
            }
          ]
        },
        {
          "num": {
            "mibModuleName": "IF-MIB",
            "objectName": "ifSpeed",
            "comment": "1,000,000 bits per second"
          }
        }
      ]
    },
    {
      "const": 0.0008,
      "comment": "normalize to ratio of bits and convert to percent:
8 * 100 / 1,000,000 = 0.0008"
    }
  ]
}

```

## 키워드

통합 팩 키워드 string은 일반적으로 16진수 형식으로 렌더링되어 ASCII 문자로 렌더링되는 옥텟 문자열이나 옥텟 문자열에서 파생된 독점 형식을 강제로 사용하도록 구현됩니다.

옥텟 문자열에는 대개 바이너리 데이터가 포함됩니다(예: MAC 주소 및 WWN:

```

"interface_mac": {
  "mibModuleName": "IF-MIB",
  "objectName": "ifPhysAddress"
}

```

ifPhysAddress 는 물리주소 유형으로, 옥텟 문자열일 뿐입니다:

```

PhysAddress ::= TEXTUAL-CONVENTION
                DISPLAY-HINT "1x:"
                STATUS         current
                DESCRIPTION
                    "Represents media- or physical-level
addresses."
                SYNTAX         OCTET STRING

```

ifPhysAddress 가 기본적으로 16진수로 렌더링되면 결과는 다음과 같습니다.

```
"interface_mac": "00:50:56:A2:07:E7"
```

그러나 ASCII로 해석하려는 옥텟 문자열에서 파생된 옥텟 문자열 또는 독점 유형이 있는 경우 "string" 키워드를 사용할 수 있습니다.

```

"string_test_1": {
  "string": {
    "mibModuleName":      "IF-MIB",
    "objectName":         "ifPhysAddress"
  }
},

"string_test_2": {
  "string": [
    {
      "mibModuleName":      "IF-MIB",
      "objectName":         "ifPhysAddress"
    },
    {
      "const": "JSD"
    },
    {
      "mibModuleName":      "IF-MIB",
      "objectName":         "ifPhysAddress"
    }
  ]
}

```

키워드는 기존 문자열 연결 규칙을 따르며 다음 예제에서 용어 사이에 공백을 하나만 삽입합니다.

```

"string_test_1": "PVϕϕ",
  "string_test_2": "PVϕϕ JSD PVϕϕ"

```

"string" 키워드는 단일 용어 또는 용어 목록에 작동하지만 중첩된 식에는 적용되지 않습니다. 중첩 식은 dataPoint 표현식에만 지원됩니다. dataPoint 식에서 "string" 식을 사용하려고 하면 다음과 유사한 오류가 발생합니다.

```
_java.lang.IllegalArgumentException: 통합 팩 'GenericSwitch32' 인덱스 'snmp_generic_interface_32' 섹션  
'ataPoints' 키 'tring_test_3'은 지원되지 않는 JSON 수치 표현식 '{"string":{"mibModuleName":"if-MIB",  
"ifjectName"}}' _
```

DisplayString, SnmpAdminString 등의 일부 파생 옥텟 문자열 형식은 "string" 키워드보다 하드 코딩된 우선 순위를 가집니다. 이는 특히 SnmpAdminString이 UTF-8로 인코딩되어 올바르게 처리하려고 하는 반면 "string" 키워드는 문자당 1바이트 ASCII 코드 포인트를 사용하는 snmp\_Framework에서 반환하는 기본 문자열 표현을 강제로 사용하기 때문입니다.

## 저작권 정보

Copyright © 2024 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

## 상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.