VACUS

ASSET TRACKING SOLUTIONS

SNMP Integration Manual

Asset Tracking System

Document Version 1.0

Technical Integration Guide

Table of Contents

1. Overview

This manual describes the SNMP (Simple Network Management Protocol) integration architecture for the Vacus Asset Tracking System. It is intended for system integrators, network administrators, and monitoring teams who need to integrate Vacus asset data into third-party SNMP-based monitoring or management platforms.

The Vacus Asset Tracking System exposes real-time information for every registered asset tag through a set of custom SNMP Object Identifiers (OIDs) under the private enterprise base:

```
1.3.6.1.4.54126.1
```

Each asset tag is uniquely identified by its MAC address. For every tag, the following parameters are available over SNMP:

- Asset MAC ID — the unique hardware identifier of the asset tag
- Rack ID (Reader ID) — the identifier of the rack or reader detecting the tag
- U Location — the rack unit position where the asset is mounted
- Battery — the current battery level of the asset tag
- RSSI — the Received Signal Strength Indicator reported by the reader

All values are returned as SNMP OctetString types. This manual explains the OID structure, the ASCII encoding scheme for MAC addresses, example mappings, and common SNMP operations used to query the system.

2. OID Structure

2.1 Base OID

All Vacus SNMP OIDs are built on the following base OID:

```
1.3.6.1.4.54126.1
```

2.2 OID Format

Every queryable value follows a consistent format:

```
<BASE_OID>.<Field_Identifier>.17.<ASCII_Representation_of_MAC_ID>
```

The numeric literal 17 is a mandatory delimiter placed between the field identifier and the ASCII-encoded MAC suffix. It must be preserved in every query.

2.3 Field Identifiers

The field identifier tells the system which attribute of the asset you are requesting:

Field ID	Attribute	Description
1	Asset MAC Address	Unique hardware ID of the asset tag

Field ID	Attribute	Description
2	Rack ID	Identifier of the detecting rack / reader
3	U Location	Rack unit position of the asset
4	Battery Level	Remaining battery (percentage)
5	RSSI	Received Signal Strength Indicator

3. ASCII Encoding of MAC Address

Because SNMP OIDs can only contain numeric components separated by dots, each MAC address must be converted to an ASCII-encoded numeric form before it is appended to the OID.

Each character in the MAC address string (including hyphens) is replaced with its decimal ASCII value. The resulting numbers are joined with dots.

3.1 Encoding Example

Consider the MAC address:

```
5a-c2-15-02-02-91
```

Character-by-character ASCII conversion:

Character	ASCII Value
5	53
a	97
-	45
c	99
2	50
-	45
1	49
5	53
-	45
0	48
2	50
-	45

Character	ASCII Value
0	48
2	50
-	45
9	57
1	49

Concatenated ASCII representation:

```
53.97.45.99.50.45.49.53.45.48.50.45.48.50.45.57.49
```

Resulting full OID (for Field Identifier 1 — Asset MAC Address):

```
1.3.6.1.4.54126.1.1.17.53.97.45.99.50.45.49.53.45.48.50.45.48.50.45.57.49
```

4. Example Data Mapping

The following table shows a complete example of all five attributes for a single asset tag.

Asset Tag MAC:

```
5a-c2-15-02-02-91
```

Field	OID	Value	Type
MAC ID	1.3.6.1.4.54126.1.1.17.53.97.45.99...57.49	5a-c2-15-02-02-91	OctetString
Rack ID	1.3.6.1.4.54126.1.2.17.53.97.45.99...57.49	5a-c2-15-09-00-01	OctetString
U Location	1.3.6.1.4.54126.1.3.17.53.97.45.99...57.49	26	OctetString
Battery	1.3.6.1.4.54126.1.4.17.53.97.45.99...57.49	90	OctetString
RSSI	1.3.6.1.4.54126.1.5.17.53.97.45.99...57.49	53	OctetString

Note: The OID values above are shown in truncated form for readability. The full OID always contains the complete ASCII-encoded MAC suffix, as described in Section 3.

5. SNMP Operations

This section describes the standard SNMP operations supported by the Vacus Asset Tracking System. All examples use Net-SNMP command-line tools (snmpwalk, snmpget). SNMPv2c is supported.

5.1 Walking All Asset Tags

To retrieve every attribute for every asset tag registered in the system, walk the base OID:

```
snmpwalk -v2c -c <community> <device_ip> 1.3.6.1.4.54126.1
```

Replace the placeholders as follows:

- <community> — the SNMP community string configured on the Vacus device
- <device_ip> — the IP address or hostname of the Vacus asset tracking gateway

5.2 Retrieving a Specific Attribute

To query a single attribute for a specific asset tag, use snmpget with the fully constructed OID. The example below retrieves the battery level (Field ID 4) for an asset tag:

```
snmpget -v2c -c <community> <device_ip> \
1.3.6.1.4.54126.1.4.17.<ascii_encoded_mac>
```

5.3 Parameter Reference

Parameter	Description
`-v2c`	SNMP protocol version (use v2c)
`-c <community>`	Community string for read access
`<device_ip>`	IP address or hostname of the Vacus gateway
`<ascii_encoded_mac>`	ASCII-encoded MAC suffix, produced as described in Section 3

6. Integration Notes and Best Practices

The following points should be observed when implementing integrations against the Vacus SNMP interface:

- All attribute values are returned as SNMP OctetString type, regardless of whether the underlying data is textual (MAC address) or numeric (battery, RSSI, U Location). Integrations should parse values as strings and convert them as needed.
- The MAC ID within each OID is encoded using decimal ASCII character codes separated by dots. The encoding always represents the full MAC string, including the hyphen separators.
- The numeric literal 17 must always be preserved as the delimiter between the field identifier and the ASCII-encoded MAC suffix. Omitting or altering it will result in an invalid OID.
- To map a returned OID back to its source asset, integration systems should extract the ASCII segment after the .17. delimiter and decode it back to the original MAC address string.

- SNMPv2c is the supported protocol version. Ensure that the community string configured on the monitoring system matches the one configured on the Vacus device.
- For large deployments, prefer snmpwalk at scheduled intervals rather than frequent individual snmpget calls, to reduce load on the gateway.

7. Appendix: ASCII Encoder Utility

The following Python helper converts a MAC address string into its ASCII-encoded OID suffix. This is the recommended way to generate the MAC portion of an OID programmatically.

7.1 Python Implementation

```
def mac_to_oid(mac_str):     """     Convert a MAC address string to its ASCII-encoded OID suffix.      Example:         >>> mac_to_oid("5a-c2-15-02-02-91")     '53.97.45.99.50.45.49.53.45.48.50.45.48.50.45.57.49'     """     ascii_parts = [str(ord(c)) for c in mac_str]     return '.'.join(ascii_parts)   # Usage mac = "5a-c2-15-02-02-91" print("OID Suffix:", mac_to_oid(mac))
```

7.2 Expected Output

```
OID Suffix: 53.97.45.99.50.45.49.53.45.48.50.45.48.50.45.57.49
```

7.3 Constructing a Full OID

Once the suffix is generated, prepend the base OID and the desired field identifier. For example, to query the battery level (Field ID 4):

```
BASE_OID = "1.3.6.1.4.54126.1" FIELD_BATTERY = 4 DELIMITER = 17  mac = "5a-c2-15-02-02-91" full_oid = f"{BASE_OID}.{FIELD_BATTERY}.{DELIMITER}.{mac_to_oid(mac)}" print(full_oid)
```

— End of Document —

Vacus Asset Tracking — SNMP Integration Manual v1.0

For technical support, contact your Vacus representative.