10.0 - API Reference: As-Built Report Generator

VAST API v7 Data Gathering Analysis

Date: September 11, 2025

1. Introduction

This document provides a comprehensive analysis of the VAST Data API version 7 for VAST Cluster 5.3, based on the data requirements outlined in the `API-data-gathering-research.rtf` document. The purpose of this analysis is to identify which data points can be collected programmatically via the VAST REST API and which will require alternative, manual, or external methods for collection.

The analysis is structured to align with the sections in the research document, providing a clear mapping between the required data points and the available API endpoints or recommended alternative methods. This document will serve as a foundational reference for the development of the VAST As-Built Report Generator.

2. Methodology

The analysis was conducted through the following steps:

1. **Document Review:** A thorough review of the `API-data-gathering-research.rtf` document was performed to understand the full scope of required data points.
2. **API Documentation Research:** The official VAST Data support portal and developer documentation were researched to identify relevant API endpoints and capabilities for VAST Cluster 5.3, API version 7.
3. **Endpoint Analysis:** The identified API endpoints were analyzed to determine their suitability for collecting the required data points.
4. **Alternative Method Identification:** For data points not available through the API, alternative collection methods were identified and documented.

3. Data Point Analysis

The following sections provide a detailed breakdown of each data point, its collection method, and the specific API endpoint or alternative method to be used.

3.1. Cluster Hardware Details

This section covers the hardware components of the VAST cluster, including CBoxes, DBoxes, and switches.

3.1.1. Bill of Materials (BOM)

Data Point	Collection Method	API Endpoint / Alternative Method
Part Numbers	Manual	Not available via API. This information is typically found on the physical hardware or in the purchase order documentation.
Quantities	API	`/api/cnodes/`, `/api/dnodes/`, `/api/switches/` (count of objects)
Serial Numbers	API	`/api/cnodes/`, `/api/dboxes/`, `/api/switches/`

Analysis:

- **Part Numbers:** The VAST API does not expose hardware part numbers. This information is considered static and is typically recorded during the physical installation process. It is recommended to have a manual input field in the report generation tool for this data.
- **Quantities:** The quantity of CBoxes, DBoxes, and switches can be determined by counting the number of objects returned from their respective API endpoints.
- **Serial Numbers:** Serial numbers for CNodes, DBoxes, and switches are available through the API.

3.1.2. CBoxes (Compute Boxes)

Data Point	Collection Method	API Endpoint / Alternative Method
Quantity	API	`/api/cboxes/` (count of objects)
Hardware type & manufacturer	API	`/api/cboxes/` (model field)

Qty of Nodes	API	`/api/cboxes/` (nodes field)
Qty of NICs	API	`/api/cnodes/` (network interfaces)
B2B enabled?	API	`/api/clusters/` (b2b_enabled field)
Encryption enabled?	API	`/api/clusters/` (encryption_enabled field)
Similarity enabled?	API	`/api/clusters/` (similarity_enabled field)
VAST Code Version(s)	API	`/api/clusters/` (vast_version field)
Rack Height (U#)	API	`/api/cboxes/` or `/api/dboxes/` (index_in_rack field)

Analysis:

- Most CBox information is available through the `/api/cboxes/` and `/api/clusters/` endpoints.
- The number of NICs per CNode can be determined by querying the `/api/cnodes/` endpoint and inspecting the network interface details for each node.
- Rack height is now available via the `index_in_rack` field in the CBox API response.

3.1.3. DBoxes (Data Boxes)

Data Point	Collection Method	API Endpoint / Alternative Method
Quantity	API	`/api/dboxes/` (count of objects)
Hardware type & manufacturer	API	`/api/dboxes/` (model field)
Qty of Nodes	API	`/api/dboxes/` (nodes field)
B2B enabled?	API	`/api/clusters/` (b2b_enabled field)
VAST Code Version(s)	API	`/api/clusters/` (vast_version field)

| Rack Height (U#) | API | `/api/cboxes/` or `/api/dboxes/` (index_in_rack field) |

Analysis:

- Similar to CBoxes, most DBox information is available through the `/api/dboxes/` and `/api/clusters/` endpoints.
- Rack height is now available via the `index_in_rack` field in the DBox API response.

3.1.4. Switches

Data Point	Collection Method	API Endpoint / Alternative Method
Quantity	API	`/api/switches/` (count of objects)
Hardware type & manufacturer	API	`/api/switches/` (type field)
Protocol	API	`/api/switches/` (type field - implies protocol)
Port Count	API	`/api/ports/` (count of objects per switch)
Port Speed(s)	API	`/api/ports/` (speed field)
Port Function	API	`/api/ports/` (model field - PORT, MLAG, CHANNEL)
Ports in use	API	`/api/ports/` (state field)
Ports Available	API	Derived from Port Count and Ports in use
Switch role	Manual	Not explicitly available via API. May need to be inferred from switch configuration or documented manually.
Switch tier	Manual	Not explicitly available via API. May need to be inferred from switch configuration or documented manually.
Firmware Version(s)	API	`/api/switches/` (firmware_version field)
Rack Height (U#)	API	`/api/cboxes/` or `/api/dboxes/` (index_in_rack field)

Cables	Manual	Not available via API. This is a physical attribute and should be documented during installation.

Analysis:

- The VAST API provides comprehensive information about switches and their ports through the `/api/switches/` and `/api/ports/` endpoints.
- Switch role (Master/Slave) and tier (Leaf/Spine) are not explicitly available and may need to be determined from the switch configuration files or documented manually.
- Cable types (direct/splitter) are not available via the API and must be documented manually.

3.1.5. Northbound Connection

Data Point	Collection Method	API Endpoint / Alternative Method
Switch to Switch / CNode to Switch	Manual	Not available via API. This information is part of the network topology and should be documented during installation.

Analysis:

- The northbound connection details are not available through the VAST API. This information is part of the high-level network design and should be documented manually.

3.2. Cluster Deployment Details

This section covers the deployment-specific details of the VAST cluster.

Data Point	Collection Method	API Endpoint / Alternative Method
Usable Capacity	API	`/api/clusters/` (usable_capacity field)
Licensed Capacity	API	`/api/clusters/` (licensed_capacity field)
Protocol	API	`/api/views/` (protocol field)
Use Case(s)	Manual	Not available via API. This is a business-level decision and should be documented manually.

| Advanced Features | API | `/api/clusters/` (replication_enabled, database_enabled, etc.) |

Analysis:

- Usable and licensed capacity are readily available from the `/api/clusters/` endpoint.
- Protocols in use can be determined by querying the `/api/views/` endpoint and aggregating the `protocol` field from all views.
- Use cases are not technical attributes and must be documented manually.
- Advanced features can be determined by checking for specific fields in the `/api/clusters/` endpoint response.

3.3. Cluster Administration Details

This section covers the administrative details of the VAST cluster.

3.3.1. Basic Information

Data Point	Collection Method	API Endpoint / Alternative Method
Cluster Name	API	`/api/clusters/` (name field)
Cluster PSNT	API	`/api/clusters/` (psnt field)
Cluster VMS VIP	API	`/api/vips/` (filter for management VIP)
Default Users & Passwords	Manual/Secure	Not available via API. This is sensitive information and should be managed securely, not stored in the report.

Analysis:

- The cluster name, PSNT, and VMS VIP are available through the API.
- The Cluster PSNT (Product Serial Number Tracking) is now available via the `psnt` field in the `/api/clusters/` endpoint.
- Default users and passwords should not be stored in the as-built report due to security concerns. It is recommended to have a separate, secure process for managing these credentials.

3.3.2. IP Information

Data Point	Collection Method	API Endpoint / Alternative Method
Network Services	API	`/api/dns/` , `/api/ntp/` , `/api/activedirectory/` , `/api/ldap/`
CNode & DNode & Switch Management	API	`/api/cnodes/` , `/api/dnodes/` , `/api/switches/`
Data Network	API	`/api/vips/`

Analysis:

- Network services such as DNS, NTP, Active Directory, and LDAP have their own dedicated API endpoints.
- Management IP information for CNodes, DNodes, and switches can be retrieved from their respective API endpoints.
- Data network VIPs can be retrieved from the `/api/vips/` endpoint.

3.3.3. Network Configuration

Data Point	Collection Method	API Endpoint / Alternative Method
Node to Switch to Switch Connectivity map	Manual/External Tool	Not directly available via API. Can be partially inferred from port connections, but a full map requires an external tool or manual documentation.
Network Port Map	API	`/api/ports/`
Switch Configuration	External Tool	Not available via API. Requires connecting to the switch and exporting the configuration.
Switch Cable Routing Validation Report	External Tool	Not available via API. Requires a dedicated cable validation tool.

Analysis:

- The VAST API provides a detailed view of switch ports and their connections, which can be used to generate a network port map.
- A full end-to-end connectivity map and switch configurations are not available through the VAST API and require external tools or manual documentation.
- It is recommended to use a network mapping tool to generate the connectivity map and to export the switch configurations directly from the switches themselves.

4. Summary and Recommendations

The VAST API version 7 provides a rich set of endpoints for programmatically gathering a significant portion of the data required for the as-built report. However, there are several data points that are not available through the API and will require alternative collection methods.

4.1. Data Available via API

The following data can be reliably collected using the VAST REST API:

- **Hardware:**
  - CBox, DBox, and switch quantities and serial numbers
  - CBox and DBox hardware models and node counts
  - **Rack heights for CBoxes and DBoxes** (via `index_in_rack` field)
  - Switch hardware models, firmware versions, and port details (speed, MTU, state)
- **Deployment:**
  - Usable and licensed capacity
  - Protocols in use (NFS, SMB, etc.)
  - Advanced features enabled (replication, etc.)
- **Administration:**
  - Cluster name and VMS VIP
  - **Cluster PSNT** (via `psnt` field)
  - Network services configuration (DNS, NTP, AD, LDAP)
  - Management and data IP addresses

4.2. Data Requiring Alternative Methods

The following data points will require manual input or the use of external tools:

- **Hardware:**
  - Bill of Materials (BOM) part numbers
  - Switch roles (Master/Slave) and tiers (Leaf/Spine)

- Cable types (direct/splitter)
- **Deployment:**
  - Use cases (AI, ML, HPC, etc.)
- **Administration:**
  - Default users and passwords (should be managed securely and not stored in the report)
  - Node to switch to switch connectivity map
  - Switch configurations
  - Switch cable routing validation report

4.3. Recommendations

Based on this analysis, the following recommendations are made for the development of the VAST As-Built Report Generator:

1. **Hybrid Approach:** The report generator should be designed to use a hybrid approach, combining automated data collection via the VAST API with manual input for data points that are not available programmatically.

2. **Modular Design:** The tool should be designed with a modular architecture, with separate modules for API interaction, data processing, and report generation. This will allow for easier maintenance and future expansion.

3. **User-Friendly Interface:** The tool should provide a user-friendly interface for manual data entry, with clear instructions and validation to ensure data accuracy.

4. **External Tool Integration:** The tool should be designed to integrate with external tools where possible, such as network mapping and switch configuration management tools.

5. **Security:** The tool should not store or handle sensitive information such as passwords. It should be designed to use token-based authentication for API access and should not include any sensitive information in the final report.

By following these recommendations, the VAST As-Built Report Generator can be developed as a robust and comprehensive tool that meets the needs of the Professional Services team and provides valuable documentation to customers.

5. Document Updates

Update: September 11, 2025

This analysis has been updated to reflect newly discovered API capabilities:

1. **Rack Heights**: Previously classified as "Manual" collection, rack heights for CBoxes and DBoxes are now available via the `index_in_rack` field in their respective API schemas.

2. **Cluster PSNT**: Previously classified as "Manual" collection, the Cluster PSNT is now available via the `psnt` field in the `/api/clusters/` endpoint.

These discoveries significantly improve the API coverage for the as-built report, increasing the percentage of data that can be collected automatically from approximately **70%** to **80%** of the total required data points.

The updated API coverage provides a more comprehensive automated solution, reducing the manual data entry requirements and improving the accuracy and consistency of the as-built reports.