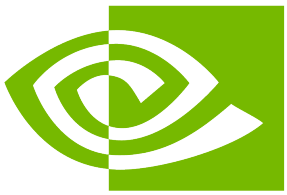

NVIDIA cuObject server v1.0.0 Release Notes

Release r1.16

NVIDIA Corporation

Jan 12, 2026

Contents

1 NVIDIA cuObject server v1.0.0 Release Notes **1**

2 Release information **3**

3 Introduction **5**
3.1 Architecture overview 5

4 Key features and changes **7**

5 Known limitations **9**
5.1 Memory and size constraints 9

6 Getting started **11**
6.1 Hardware requirements 11
6.2 Software requirements 11
6.3 Installation 11
6.3.1 Install DOCA and RDMA libraries 11
6.3.2 Install cuObject server library 11
6.3.3 Verify installation 12

7 Fixed issues **13**

8 Open issues **15**

Chapter 1. NVIDIA cuObject server v1.0.0 Release Notes

Release information for NVIDIA® cuObject server.

Chapter 2. Release information

Version	Date	cuObject server libraries with CUDA Release
v1.0.0	Jan 12, 2026	13.1.1

Chapter 3. Introduction

Release information for NVIDIA® cuObject server for developers and users.

cuObject is a high-performance suite of libraries designed to enable direct data transfers between GPU memory or system memory and an object storage (S3-compatible) solution via RDMA. By relying on RDMA operations, rather than TCP transfer methods, cuObject avoids using CPU kernel code for TCP processing and can bypass the CPU for data payloads. cuObject eliminates the traditional bottleneck of staging data in local scratch file systems, enabling high-throughput data ingestion for AI training and inference at scale.

3.1. Architecture overview

cuObject consists of two primary components:

- **cuObjClient**: Provides client-side APIs for GET and PUT operations with user-defined callbacks for control path communication, while data path operations leverage RDMA for high-performance transfers.
- **cuObjServer**: Implements the RDMA-accelerated server side for S3-compatible object storage services, supporting multi-threaded concurrency, automatic connection management, and Dynamically Connected (DC) transport.

Refer to the following guides for more information about cuObject:

- NVIDIA cuObject Documentation

Chapter 4. Key features and changes

The following features have been added in v1.0.0:

- Multi-threaded, channel-based concurrency
- Scatter-gather I/O support (up to 10 entries)
- Asynchronous operation and polling

Chapter 5. Known limitations

5.1. Memory and size constraints

- Maximum operation size: 1 GiB per `handleGetObject()` or `handlePutObject()` call
- Maximum scatter-gather entries: 10 entries per operation
- Maximum SGE per operation: configurable via `max_sge` parameter (default: 10)

Chapter 6. Getting started

6.1. Hardware requirements

- x86_64 or ARM64 CPU architecture
- Mellanox ConnectX-5 and above InfiniBand HCA, or RoCE-capable Ethernet adapter
- Minimum 16 GB system RAM recommended

6.2. Software requirements

- Linux operating system (tested on Ubuntu 22.04 and 24.04, RHEL 8, 9, and 10)
- InfiniBand and RDMA drivers and libraries (`libibverbs`, `librdmacm`)
- C++14 or later compatible compiler

6.3. Installation

6.3.1. Install DOCA and RDMA libraries

Use the DOCA installation guide to install DOCA.

6.3.2. Install cuObject server library

Follow the instructions from cuObject Server library downloads.

6.3.3. Verify installation

```
# Check library presence
ldconfig -p | grep cuobj

# Verify cuObjServer installation
rpm -qa | grep cuobjserver    # RHEL/CentOS
dpkg -l | grep cuobjserver    # Ubuntu/Debian
```

Chapter 7. Fixed issues

This is the initial release of cuObject server v1.0.0. No prior versions exist.

Chapter 8. Open issues

None reported for the v1.0.0 initial release.

Copyright