[NCCL](index.html)

[2.25](https://docs.nvidia.com/deeplearning/sdk/nccl-archived/index.html)

    * [ncclGetLastError](api/comms.html#ncclgetlasterror)
    * [ncclGetErrorString](api/comms.html#ncclgeterrorstring)
    * [ncclGetVersion](api/comms.html#ncclgetversion)
    * [ncclGetUniqueId](api/comms.html#ncclgetuniqueid)
    * [ncclCommInitRank](api/comms.html#ncclcomminitrank)
    * [ncclCommInitAll](api/comms.html#ncclcomminitall)
    * [ncclCommInitRankConfig](api/comms.html#ncclcomminitrankconfig)
    * [ncclCommInitRankScalable](api/comms.html#ncclcomminitrankscalable)
    * [ncclCommSplit](api/comms.html#ncclcommsplit)
    * [ncclCommFinalize](api/comms.html#ncclcommfinalize)
    * [ncclCommDestroy](api/comms.html#ncclcommdestroy)
    * [ncclCommAbort](api/comms.html#ncclcommabort)
    * [ncclCommGetAsyncError](api/comms.html#ncclcommgetasyncerror)
    * [ncclCommCount](api/comms.html#ncclcommcount)
    * [ncclCommCuDevice](api/comms.html#ncclcommcudevice)
    * [ncclCommUserRank](api/comms.html#ncclcommuserrank)
    * [ncclCommRegister](api/comms.html#ncclcommregister)
    * [ncclCommDeregister](api/comms.html#ncclcommderegister)
    * [ncclMemAlloc](api/comms.html#ncclmemalloc)
    * [ncclMemFree](api/comms.html#ncclmemfree)
  * [Collective Communication Functions](api/colls.html)
    * [ncclAllReduce](api/colls.html#ncclallreduce)
    * [ncclBroadcast](api/colls.html#ncclbroadcast)
    * [ncclReduce](api/colls.html#ncclreduce)
    * [ncclAllGather](api/colls.html#ncclallgather)
    * [ncclReduceScatter](api/colls.html#ncclreducescatter)
  * [Group Calls](api/group.html)

* [System configuration](env.html#system-configuration)
  * [NCCL_SOCKET_IFNAME](env.html#nccl-socket-ifname)
    * [Values accepted](env.html#values-accepted)
  * [NCCL_SOCKET_FAMILY](env.html#nccl-socket-family)
    * [Values accepted](env.html#id2)
  * [NCCL_SOCKET_RETRY_CNT](env.html#nccl-socket-retry-cnt)
    * [Values accepted](env.html#id3)
  * [NCCL_SOCKET_RETRY_SLEEP_MSEC](env.html#nccl-socket-retry-sleep-msec)
    * [Values accepted](env.html#id4)
  * [NCCL_SOCKET_NTHREADS](env.html#nccl-socket-nthreads)
    * [Values accepted](env.html#id5)
  * [NCCL_NSOCKS_PERTHREAD](env.html#nccl-nsocks-perthread)
    * [Values accepted](env.html#id6)
  * [NCCL_CROSS_NIC](env.html#nccl-cross-nic)
    * [Values accepted](env.html#id7)
  * [NCCL_IB_HCA](env.html#nccl-ib-hca)
    * [Values accepted](env.html#id8)
  * [NCCL_IB_TIMEOUT](env.html#nccl-ib-timeout)
    * [Values accepted](env.html#id9)
  * [NCCL_IB_RETRY_CNT](env.html#nccl-ib-retry-cnt)
    * [Values accepted](env.html#id10)
  * [NCCL_IB_GID_INDEX](env.html#nccl-ib-gid-index)
    * [Values accepted](env.html#id11)
  * [NCCL_IB_ADDR_FAMILY](env.html#nccl-ib-addr-family)
    * [Values accepted](env.html#id12)
  * [NCCL_IB_ADDR_RANGE](env.html#nccl-ib-addr-range)
    * [Values accepted](env.html#id13)

* [NCCL_IB_ROCE_VERSION_NUM](env.html#nccl-ib-roce-version-num)
  * [Values accepted](env.html#id14)
* [NCCL_IB_SL](env.html#nccl-ib-sl)
  * [Values accepted](env.html#id15)
* [NCCL_IB_TC](env.html#nccl-ib-tc)
  * [Values accepted](env.html#id16)
* [NCCL_IB_FIFO_TC](env.html#nccl-ib-fifo-tc)
  * [Values accepted](env.html#id17)
* [NCCL_IB_RETURN_ASYNC_EVENTS](env.html#nccl-ib-return-async-events)
  * [Values accepted](env.html#id18)
* [NCCL_OOB_NET_ENABLE](env.html#nccl-oob-net-enable)
  * [Values accepted](env.html#id19)
* [NCCL_OOB_NET_IFNAME](env.html#nccl-oob-net-ifname)
  * [Values accepted](env.html#id20)
* [NCCL_UID_STAGGER_THRESHOLD](env.html#nccl-uid-stagger-threshold)
  * [Values accepted](env.html#id21)
* [NCCL_UID_STAGGER_RATE](env.html#nccl-uid-stagger-rate)
  * [Values accepted](env.html#id22)
* [NCCL_NET](env.html#nccl-net)
  * [Values accepted](env.html#id23)
* [NCCL_NET_PLUGIN](env.html#nccl-net-plugin)
  * [Values accepted](env.html#id24)
* [NCCL_TUNER_PLUGIN](env.html#nccl-tuner-plugin)
  * [Values accepted](env.html#id25)
* [NCCL_PROFILER_PLUGIN](env.html#nccl-profiler-plugin)
  * [Values accepted](env.html#id26)
* [NCCL_IGNORE_CPU_AFFINITY](env.html#nccl-ignore-cpu-affinity)

* [NCCL_SHM_DISABLE](env.html#nccl-shm-disable)
  * [Values accepted](env.html#id40)
* [NCCL_BUFFSIZE](env.html#nccl-buffsize)
  * [Values accepted](env.html#id41)
* [NCCL_NTHREADS](env.html#nccl-nthreads)
  * [Values accepted](env.html#id42)
* [NCCL_MAX_NCHANNELS](env.html#nccl-max-nchannels)
  * [Values accepted](env.html#id43)
* [NCCL_MIN_NCHANNELS](env.html#nccl-min-nchannels)
  * [Values accepted](env.html#id44)
* [NCCL_CHECKS_DISABLE](env.html#nccl-checks-disable)
  * [Values accepted](env.html#id45)
* [NCCL_CHECK_POINTERS](env.html#nccl-check-pointers)
  * [Values accepted](env.html#id46)
* [NCCL_LAUNCH_MODE](env.html#nccl-launch-mode)
  * [Values accepted](env.html#id47)
* [NCCL_IB_DISABLE](env.html#nccl-ib-disable)
  * [Values accepted](env.html#id48)
* [NCCL_IB_AR_THRESHOLD](env.html#nccl-ib-ar-threshold)
  * [Values accepted](env.html#id49)
* [NCCL_IB_QPS_PER_CONNECTION](env.html#nccl-ib-qps-per-connection)
  * [Values accepted](env.html#id50)
* [NCCL_IB_SPLIT_DATA_ON_QPS](env.html#nccl-ib-split-data-on-qps)
  * [Values accepted](env.html#id51)
* [NCCL_IB_CUDA_SUPPORT](env.html#nccl-ib-cuda-support)
  * [Values accepted](env.html#id52)
* [NCCL_IB_PCI_RELAXED_ORDERING](env.html#nccl-ib-pci-relaxed-ordering)

  * [Values accepted](env.html#id53)
* [NCCL_IB_ADAPTIVE_ROUTING](env.html#nccl-ib-adaptive-routing)
  * [Values accepted](env.html#id54)
* [NCCL_IB_ECE_ENABLE](env.html#nccl-ib-ece-enable)
  * [Values accepted](env.html#id55)
* [NCCL_MEM_SYNC_DOMAIN](env.html#nccl-mem-sync-domain)
  * [Values accepted](env.html#id56)
* [NCCL_CUMEM_ENABLE](env.html#nccl-cumem-enable)
  * [Values accepted](env.html#id57)
* [NCCL_CUMEM_HOST_ENABLE](env.html#nccl-cumem-host-enable)
  * [Values accepted](env.html#id58)
* [NCCL_NET_GDR_LEVEL (formerly NCCL_IB_GDR_LEVEL)](env.html#nccl-net-gdr-level-formerly-nccl-ib-gdr-level)
  * [Values accepted](env.html#id59)
  * [Integer Values (Legacy)](env.html#id60)
* [NCCL_NET_GDR_READ](env.html#nccl-net-gdr-read)
  * [Values accepted](env.html#id61)
* [NCCL_NET_SHARED_BUFFERS](env.html#nccl-net-shared-buffers)
  * [Value accepted](env.html#id62)
* [NCCL_NET_SHARED_COMMS](env.html#nccl-net-shared-comms)
  * [Value accepted](env.html#id63)
* [NCCL_SINGLE_RING_THRESHOLD](env.html#nccl-single-ring-threshold)
  * [Values accepted](env.html#id64)
* [NCCL_LL_THRESHOLD](env.html#nccl-ll-threshold)
  * [Values accepted](env.html#id65)
* [NCCL_TREE_THRESHOLD](env.html#nccl-tree-threshold)
  * [Values accepted](env.html#id66)

* [NCCL_ALGO](env.html#nccl-algo)
  * [Values accepted](env.html#id67)
* [NCCL_PROTO](env.html#nccl-proto)
  * [Values accepted](env.html#id68)
* [NCCL_NVB_DISABLE](env.html#nccl-nvb-disable)
  * [Value accepted](env.html#id69)
* [NCCL_PXN_DISABLE](env.html#nccl-pxn-disable)
  * [Value accepted](env.html#id70)
* [NCCL_P2P_PXN_LEVEL](env.html#nccl-p2p-pxn-level)
  * [Value accepted](env.html#id71)
* [NCCL_RUNTIME_CONNECT](env.html#nccl-runtime-connect)
  * [Value accepted](env.html#id72)
* [NCCL_GRAPH_REGISTER](env.html#nccl-graph-register)
  * [Value accepted](env.html#id74)
* [NCCL_LOCAL_REGISTER](env.html#nccl-local-register)
  * [Value accepted](env.html#id75)
* [NCCL_LEGACY_CUDA_REGISTER](env.html#nccl-legacy-cuda-register)
  * [Value accepted](env.html#id76)
* [NCCL_SET_STACK_SIZE](env.html#nccl-set-stack-size)
  * [Value accepted](env.html#id77)
* [NCCL_GRAPH_MIXING_SUPPORT](env.html#nccl-graph-mixing-support)
  * [Value accepted](env.html#id79)
* [NCCL_DMABUF_ENABLE](env.html#nccl-dmabuf-enable)
  * [Value accepted](env.html#id80)
* [NCCL_P2P_NET_CHUNKSIZE](env.html#nccl-p2p-net-chunksize)
  * [Values accepted](env.html#id81)
* [NCCL_P2P_LL_THRESHOLD](env.html#nccl-p2p-ll-threshold)

__[NCCL](index.html)

* * *

Migrating from NCCL 1 to NCCL 2Â¶

If you are using NCCL 1.x and want to move to NCCL 2.x, be aware that the APIs have changed slightly. NCCL 2.x supports all of the collectives that NCCL 1.x supports, but with slight modifications to the API.

In addition, NCCL 2.x also requires the usage of the â€œGroup APIâ€• when a single thread manages NCCL calls for multiple GPUs.

The following list summarizes the changes that may be required in usage of NCCL API when using an application that has a single thread that manages NCCL calls for multiple GPUs, and is ported from NCCL 1.x to 2.x:

InitializationÂ¶

In versions 1.x, NCCL had to be initialized using ncclCommInitAll at a single thread or having one thread per GPU concurrently call ncclCommInitRank. NCCL 2.x retains these two modes of initialization. It adds a new mode with the Group API where ncclCommInitRank can be called in a loop, like a communication call, as shown below. The loop has to be guarded by the Group start and end API.

```
ncclGroupStart();
for (int i=0; i<ngpus; i++) {
  cudaSetDevice(i);
  ncclCommInitRank(comms+i, ngpus, id, i);
```

```
}
ncclGroupEnd();
```

CommunicationÂ¶

In NCCL 2.x, the collective operation can be initiated for different devices by making calls in a loop, on a single thread. This is similar to the usage in NCCL 1.x. However, this loop has to be guarded by the Group API in 2.x. Unlike in 1.x, the application does not have to select the relevant CUDA device before making the communication API call. NCCL runtime internally selects the device associated with the NCCL communicator handle. For example:

```
ncclGroupStart();
for (int i=0; i<nLocalDevs; i++) {
  ncclAllReduce(..., comm[i], stream[i]);
}
ncclGroupEnd();
```

When using only one device per thread or one device per process, the general usage of the API remains unchanged from NCCL 1.x to 2.x. The usage of the group API is not required in this case.

CountsÂ¶

Counts provided as arguments are now of type size_t instead of integer.

In-place usage for AllGather and ReduceScatterÂ¶

For more information, see â€œIn-place Operationsâ€•.

AllGather arguments orderÂ¶

The AllGather function had its arguments reordered. The prototype changed

from:

```
ncclResult_t  ncclAllGather(const void* sendbuff, int count, ncclDataType_t datatype,
    void* recvbuff, ncclComm_t comm, cudaStream_t stream);
```

to:

```
ncclResult_t  ncclAllGather(const void* sendbuff, void* recvbuff, size_t sendcount,
    ncclDataType_t datatype, ncclComm_t comm, cudaStream_t stream);
```

The recvbuff argument has been moved after sendbuff to be consistent with all

the other operations.

DatatypesÂ¶

New datatypes have been added in NCCL 2.x. The ones present in NCCL 1.x did not change and are still usable in NCCL 2.x.

Error codesÂ¶

Error codes have been merged into the ncclInvalidArgument category and have been simplified. A new ncclInvalidUsage code has been created to cover new programming errors.

[Next](examples.html "Examples") [Previous](api/ops.html "User Defined Reduction Operators")

* * *

(C) Copyright 2020, NVIDIA Corporation

Built with [Sphinx](http://sphinx-doc.org/) using a [theme](https://github.com/rtfd/sphinx_rtd_theme) provided by [Read the Docs](https://readthedocs.org).