Program Listing for File cuda_runtime_wrapper.h

Return to documentation for file (`include/holoscan/core/system/cuda_runtime_wrapper.h`)

```
/* * SPDX-FileCopyrightText: Copyright (c) 2023 NVIDIA CORPORATION & AFFILIATES.
All rights reserved. * SPDX-License-Identifier: Apache-2.0 * * Licensed under the
Apache License, Version 2.0 (the "License"); * you may not use this file except in
compliance with the License. * You may obtain a copy of the License at * *
http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law
or agreed to in writing, software * distributed under the License is distributed on an
"AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied. * See the License for the specific language governing
permissions and * limitations under the License. */ #ifndef
HOLOSCAN_CORE_SYSTEM_CUDA_RUNTIME_WRAPPER_H #define
HOLOSCAN_CORE_SYSTEM_CUDA_RUNTIME_WRAPPER_H #include <cstddef> // for
size_t namespace holoscan::cuda { // https://docs.nvidia.com/cuda/cuda-runtime-
api/index.html // CUDA definition of UUID #ifndef
HOLOSCAN_CU_UUID_HAS_BEEN_DEFINED #define
HOLOSCAN_CU_UUID_HAS_BEEN_DEFINED struct CUuuid_st { char bytes[16]; };
typedef struct CUuuid_st CUuuid; #endif typedef struct CUuuid_st cudaUUID_t;
struct cudaDeviceProp { char name[256]; cudaUUID_t uuid; char luid[8]; unsigned
int luidDeviceNodeMask; size_t totalGlobalMem; size_t sharedMemPerBlock; int
regsPerBlock; int warpSize; size_t memPitch; int maxThreadsPerBlock; int
maxThreadsDim[3]; int maxGridSize[3]; int clockRate; size_t totalConstMem; int
major; int minor; size_t textureAlignment; size_t texturePitchAlignment; int
deviceOverlap; int multiProcessorCount; int kernelExecTimeoutEnabled; int
integrated; int canMapHostMemory; int computeMode; int maxTexture1D; int
maxTexture1DMipmap; int maxTexture1DLinear; int maxTexture2D[2]; int
maxTexture2DMipmap[2]; int maxTexture2DLinear[3]; int maxTexture2DGather[2];
int maxTexture3D[3]; int maxTexture3DAlt[3]; int maxTextureCubemap; int
maxTexture1DLayered[2]; int maxTexture2DLayered[3]; int
maxTextureCubemapLayered[2]; int maxSurface1D; int maxSurface2D[2]; int
maxSurface3D[3]; int maxSurface1DLayered[2]; int maxSurface2DLayered[3]; int
maxSurfaceCubemap; int maxSurfaceCubemapLayered[2]; size_t surfaceAlignment;
int concurrentKernels; int ECCEnabled; int pciBusID; int pciDeviceID; int
pciDomainID; int tccDriver; int asyncEngineCount; int unifiedAddressing; int
memoryClockRate; int memoryBusWidth; int l2CacheSize; int
persistingL2CacheMaxSize; int maxThreadsPerMultiProcessor; int
streamPrioritiesSupported; int globalL1CacheSupported; int
localL1CacheSupported; size_t sharedMemPerMultiprocessor; int
regsPerMultiprocessor; int managedMemory; int isMultiGpuBoard; int
multiGpuBoardGroupID; int hostNativeAtomicSupported; int
singleToDoublePrecisionPerfRatio; int pageableMemoryAccess; int
concurrentManagedAccess; int computePreemptionSupported; int
canUseHostPointerForRegisteredMem; int cooperativeLaunch; int
cooperativeMultiDeviceLaunch; size_t sharedMemPerBlockOptin; int
pageableMemoryAccessUsesHostPageTables; int
directManagedMemAccessFromHost; int maxBlocksPerMultiProcessor; int
accessPolicyMaxWindowSize; size_t reservedSharedMemPerBlock; }; typedef int
cudaError_t; // __host__ __device__ const char* cudaGetErrorString ( cudaError_t error )
typedef const char* (*cudaGetErrorString_t)(cudaError_t); // __host__ __device__
cudaError_t cudaGetDeviceCount ( int* count ) typedef cudaError_t
(*cudaGetDeviceCount_t)(int*); // __host__ cudaError_t cudaGetDeviceProperties (
cudaDeviceProp* prop, int device ) typedef cudaError_t (*cudaGetDeviceProperties_t)
(cudaDeviceProp*, int); // __host__ cudaError_t cudaDeviceGetPCIBusId ( char* pciBusId,
int len, int device ) typedef cudaError_t (*cudaDeviceGetPCIBusId_t)(char*, int, int); //
__host__ cudaError_t cudaMemGetInfo ( size_t* free, size_t* total ) typedef cudaError_t
(*cudaMemGetInfo_t)(size_t*, size_t*); } // namespace holoscan::cuda #endif/*
HOLOSCAN_CORE_SYSTEM_CUDA_RUNTIME_WRAPPER_H */
```

© Copyright 2022-2024, NVIDIA.. PDF Generated on 06/06/2024