

Copyright (C) 2022, 2666680 *Ontario Inc.*

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of *MERCHANTABILITY* or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 *Franklin* Street, Fifth Floor, *Boston*, MA 02110-1301 *USA*.

EXTENDS

Naturals,
Sequences,

Internal Modules.

RsAccess

CONSTANTS

Nv0000CtrlReserved,
Nv0000CtrlSystem,
Nv0000CtrlGpu,
Nv0000CtrlGsync,
Nv0000CtrlDiag,
Nv0000CtrlEvent,
Nv0000CtrlNvd,
Nv0000CtrlSwInstr,
Nv0000CtrlGspC,
Nv0000CtrlProc,
Nv0000CtrlSyncGpuBoost,
Nv0000CtrlGpuAcct,
Nv0000CtrlVgpu,
Nv0000CtrlClient

Reserved.
Command for system.
Command for *GPU*.
Command for *GSYNC*.
Command for Diagnostics.
Command for Event interactions.
Command for *NVIDIA* Debug Dumps.
Command for *SW* Instruction.
Unknown.
Command for */proc* commands.
Command for *SLI GPU* Boosts.
Command for *GPU* accounting.
Command for *vGPUs*.
Command for Clients.

CLIENTS

CONSTANTS

Nv0000CtrlCmdGetAddrSpaceType,

Nv0000CtrlCmdGetHandleInfo,
Nv0000CtrlCmdClientGetAccessRights,

Nv0000CtrlCmdSetInheritedShare,

Query memory address space type associated with an object.

Query information on a handle.
Gets access rights for an object, the object does not need to be owned by the client requesting the access rights.

Sets an inherited policy list.

<i>Nv0000CtrlCmdGetChildHandle,</i>	Gets the child handle of a given type.
<i>Nv0000CtrlCmdShareObject</i>	Deprecating this command, but it is designed to share an object.
Possible address space types.	
CONSTANTS	
<i>AddrSpaceTypeInvalid,</i> <i>AddrSpaceTypeSysmem,</i> <i>AddrSpaceTypeVidmem,</i> <i>AddrSpaceTypeRegmem,</i> <i>AddrSpaceTypeFabric</i>	
<i>AddrSpaceTypes</i> \triangleq { <i>AddrSpaceTypeInvalid</i> , <i>AddrSpaceTypeSysmem</i> , <i>AddrSpaceTypeVidmem</i> , <i>AddrSpaceTypeRegmem</i> , <i>AddrSpaceTypeFabric</i> }	
Object for getting the address space type.	
<i>NvGetAddrSpaceType</i> \triangleq [<i>object</i> : <i>Nat</i> , <i>mapFlags</i> : <i>Nat</i> , <i>addrSpaceType</i> : <i>AddrSpaceTypes</i>]	Object handle we attempt to look up. [IN] Flags for mapping a space address type. [IN] Type of address space. [OUT]
Possible handle info lookups.	
CONSTANTS	
<i>GetHandleInfoInvalid,</i> <i>GetHandleInfoParent,</i> <i>GetHandleInfoClassId</i>	Invalid lookup. Parent device handle. Class <i>Id</i> of the device.
<i>HandleInfoLookup</i> \triangleq { <i>GetHandleInfoInvalid</i> , <i>GetHandleInfoParent</i> , <i>GetHandleInfoClassId</i> }	
Object for getting the handle information.	
<i>NvGetHandleInfo</i> \triangleq [<i>object</i> : <i>Nat</i> , <i>index</i> : <i>HandleInfoLookup</i>	Object to look up. [IN] Handle info lookup type. [IN]

$, \text{result} : \text{Nat}$ $]$	Result. [OUT]
<p>Object to get the access rights for an object.</p> $\text{NvGetAccessRights} \triangleq$ $[\text{object} : \text{Nat}$ $, \text{client} : \text{Nat}$ $, \text{result} : \text{AccessMask}$ $]$	<p>Object to look up. [IN]</p> <p>Client that owns the object. [IN]</p> <p>Result of the lookup. [OUT]</p>
<p>Object to set inherited share policy.</p> $\text{NvSetInheritedSharePolicy} \triangleq \text{RsSharePolicy}$	
<p>Object to get the child handle.</p> $\text{NvGetChildHandle} \triangleq$ $[\text{parent} : \text{Nat}$ $, \text{classid} : \text{Nat}$ $, \text{object} : \text{Nat}$ $]$	<p>Parent object handle. [IN]</p> <p>Class id of the child. [IN]</p> <p>Object <i>ID</i>. [OUT]</p>
<p>Object to share another object.</p> <p>NOTE: Avoid for releases after <i>R450</i>.</p> $\text{NvShareObject} \triangleq$ $[\text{object} : \text{Nat}$ $, \text{share} : \text{RsSharePolicy}$ $]$	<p>Object to share. [IN]</p> <p>Sharing policy. [IN]</p>

DIAGNOSTICS

CONSTANTS

$\text{Nv0000CtrlCmdGetLockMeter},$	Returns the current lock meter.
$\text{Nv0000CtrlCmdSetLockMeter},$	Sets the current lock meter.
$\text{Nv0000CtrlCmdGetLockMeterEntries},$	Gets a list of lock meter entries.
$\text{Nv0000CtrlCmdProfileRpc},$	Profiles an <i>RPC</i> in <i>VGX</i> mode.
$\text{Nv0000CtrlCmdDumpRpc}$	Dumps <i>RPC</i> runtime information.

Possible lock meter states.

CONSTANTS

$\text{LockMeterDisabled},$	Disables lock metering.
$\text{LockMeterEnabled},$	Enables lock metering.
LockMeterReset	Clears the locks, but requires it is disabled first.

$\text{LockMeterStates} \triangleq$
 $\{ \text{LockMeterDisabled}$
 $, \text{LockMeterEnabled}$

$, LockMeterReset$ $\}$	
Object for getting a meter lock state.	
$NvGetLockMeter \triangleq$ $[state : \{LockMeterDisabled$ $, LockMeterEnabled$ $\}$ $, count : Nat$ $, missedCount : Nat$ $, circularBuffer : BOOLEAN$ $]$	Whether the lock meter is enabled or disabled. Number of entries available. Number of missed entries. If the buffer is circular or sequential.
Object for setting a meter lock state.	
$NvSetMeterLock \triangleq$ $[state : LockMeterStates$ $, circularBuffer : BOOLEAN$ $]$	Possible lock meter states. If the buffer is circular or sequential.
Possible metering tags.	
CONSTANTS	
$LockMeterTagAcquireSema,$ $LockMeterTagAcquireSemaForced,$ $LockMeterTagAcquireSemaCond,$ $LockMeterTagReleaseSema,$ $LockMeterTagAcquireApi,$ $LockMeterTagReleaseApi,$ $LockMeterTagAcquireGpus,$ $LockMeterTagReleaseGpus,$ $LockMeterTagData,$ $LockMeterTagRmCtrl,$ $LockMeterTagCfgGet,$ $LockMeterTagCfgSet,$ $LockMeterTagCfgGetEx,$ $LockMeterTagCfgSetEx,$ $LockMeterTagVidHeap,$ $LockMeterTagMapMem,$ $LockMeterTagUnMapMem,$ $LockMeterTagMapMemDma,$ $LockMeterTagUnMapMemDma,$ $LockMeterTagAlloc,$ $LockMeterTagAllocMem,$ $LockMeterTagDupObject,$ $LockMeterTagFreeClient,$	

LockMeterTagFreeDevice,
LockMeterTagFreeSubDevice,
LockMeterTagFreeSubDeviceDiag,
LockMeterTagFreeDisp,
LockMeterTagFreeDispCmn,
LockMeterTagFreeChannel,
LockMeterTagFreeChannelMpeg,
LockMeterTagFreeChannelDisp,
LockMeterTagIdleChannels,
LockMeterTagBindCtxDma,
LockMeterTagAllocCtxDma,
LockMeterTagIsr,
LockMeterTagDpc

LockMeterTags \triangleq

{ LockMeterTagAcquireSema, LockMeterTagAcquireSemaForced,
LockMeterTagAcquireSemaCond, LockMeterTagReleaseSema,
LockMeterTagAcquireApi, LockMeterTagReleaseApi,
LockMeterTagAcquireGpus, LockMeterTagReleaseGpus,
LockMeterTagData, LockMeterTagRmCtrl,
LockMeterTagCfgGet, LockMeterTagCfgSet,
LockMeterTagCfgGetEx, LockMeterTagCfgSetEx,
LockMeterTagVidHeap, LockMeterTagMapMem,
LockMeterTagUnMapMem, LockMeterTagMapMemDma,
LockMeterTagUnMapMemDma, LockMeterTagAlloc,
LockMeterTagAllocMem, LockMeterTagDupObject,
LockMeterTagFreeClient, LockMeterTagFreeDevice,
LockMeterTagFreeSubDevice, LockMeterTagFreeSubDeviceDiag,
LockMeterTagFreeDisp, LockMeterTagFreeDispCmn,
LockMeterTagFreeChannel, LockMeterTagFreeChannelMpeg,
LockMeterTagFreeChannelDisp, LockMeterTagIdleChannels,
LockMeterTagBindCtxDma, LockMeterTagAllocCtxDma,
LockMeterTagIsr, LockMeterTagDpc}

Lock metering entry.

NOTE: We are ignoring the following for this portion:

- freq
- line
- filename
- cpuNum
- irq
- threadId

NvLockMeterEntry \triangleq

[*counter* : *Nat*

, *tag* : *LockMeterTags*

Nanoseconds since last boot.

Which kind of tag is this meter.

```
, data0 : Nat
, data1 : Nat
, data2 : Nat
]
```

Tag specific information.

This is the NV0000 Allocation Object.
 $Nv0000AllocParams \triangleq [client : Nat$

This is the client id which allows the process to interact with the driver.

```
, processID : Nat
```

This is the *processID* to lock the resources to.

There is also a name parameter which we removed as it serves no purpose in our specs.

```
]
```

$Nv0000Ctrls \triangleq$

```
[Nv0000CtrlReserved : {}] ∪
[Nv0000CtrlClient : {Nv0000CtrlCmdGetAddrSpaceType
, Nv0000CtrlCmdGetHandleInfo
, Nv0000CtrlCmdClientGetAccessRights
, Nv0000CtrlCmdSetInheritedShare
, Nv0000CtrlCmdGetChildHandle
, Nv0000CtrlCmdShareObject
}] ∪
[Nv0000CtrlDiag : {Nv0000CtrlCmdGetLockMeter
, Nv0000CtrlCmdSetLockMeter
, Nv0000CtrlCmdGetLockMeterEntries
, Nv0000CtrlCmdProfileRpc
, Nv0000CtrlCmdDumpRpc
}] ∪
[Nv0000CtrlGspC : {}]
```

```
\ * Modification History
\ * Last modified Mon Jun 20 17:03:41 EDT 2022 by mbuchel
\ * Created Fri Jun 17 11:04:19 EDT 2022 by mbuchel
```