# eeDaemonAPI

## 1.0.2

Generated by Doxygen 1.7.5.1

Wed Jun 27 2012 16:41:51

# Contents

# Chapter 1

# File Index

## 1.1  File List

Here is a list of all files with brief descriptions:

# Chapter 2

# File Documentation

## 2.1 include/defines.h File Reference

**Defines**

- #define CORE_ALL_ID 0
- #define CORE_MIN_ID 1
- #define CORE_MAX_ID 99
- #define NIC_ALL_ID 100
- #define NIC_MIN_ID 101
- #define NIC_MAX_ID 199
- #define DISK_ALL_ID 200
- #define DISK_MIN_ID 201
- #define DISK_MAX_ID 299
- #define MY_CORE_IDS -1
- #define MY_NIC_IDS -2
- #define MY_DISK_IDS -3
- #define RMS_RANK -1
- #define MODE_UNKNOWN 0
- #define MODE_DEFAULT 1
- #define MODE_TURBO 2
- #define MODE_MAX 3
- #define MODE_MED 4
- #define MODE_MIN 5
- #define MODE_UNUSED 6

### 2.1.1 Define Documentation

#### 2.1.1.1 #define CORE_ALL_ID 0

ID for all Cores.

Definition at line 25 of file defines.h.

**2.1.1.2 #define CORE_MIN_ID 1**

Valid Core ID's range between CORE_MIN_ID and CORE_MAX_ID.

Definition at line 29 of file defines.h.

**2.1.1.3 #define CORE_MAX_ID 99**

Valid Core ID's range between CORE_MIN_ID and CORE_MAX_ID.

Definition at line 33 of file defines.h.

**2.1.1.4 #define NIC_ALL_ID 100**

ID for all NICs.

Definition at line 37 of file defines.h.

**2.1.1.5 #define NIC_MIN_ID 101**

Valid Nic ID's range between NIC_MIN_ID and NIC_MAX_ID.

Definition at line 41 of file defines.h.

**2.1.1.6 #define NIC_MAX_ID 199**

Valid Nic ID's range between NIC_MIN_ID and NIC_MAX_ID.

Definition at line 45 of file defines.h.

**2.1.1.7 #define DISK_ALL_ID 200**

ID for all Disks.

Definition at line 49 of file defines.h.

**2.1.1.8 #define DISK_MIN_ID 201**

Valid Disk ID's range between DISK_MIN_ID and DISK_MAX_ID.

Definition at line 53 of file defines.h.

**2.1.1.9 #define DISK_MAX_ID 299**

Valid Disk ID's range between DISK_MIN_ID and DISK_MAX_ID.

Definition at line 57 of file defines.h.

**2.1.1.10    #define MY_CORE_IDS -1**

ID for the registered core ids

Definition at line 62 of file defines.h.

**2.1.1.11    #define MY_NIC_IDS -2**

ID for the registered nic ids

Definition at line 67 of file defines.h.

**2.1.1.12    #define MY_DISK_IDS -3**

ID for the registered core ids

Definition at line 72 of file defines.h.

**2.1.1.13    #define RMS_RANK -1**

Rank identifying the initialization through the RMS

Definition at line 77 of file defines.h.

**2.1.1.14    #define MODE_UNKNOWN 0**

Internal Mode, indicating device in an unknown Mode

Definition at line 107 of file defines.h.

**2.1.1.15    #define MODE_DEFAULT 1**

Mode indicating default (system) mode of devices. For the Cpu, this Mode corresponds
to the ondemand governor. For the Nic, this Mode turns on autonegotiation. For the
Disk, this Mode remounts disk with system defaults. DO NOT USE THIS MODE - It is
only used when finalizing the Daemon.

Definition at line 115 of file defines.h.

**2.1.1.16    #define MODE_TURBO 2**

Mode marking a very high utilization for the device - device must be switched to the
highest performance mode. For the Cpu, this Mode corresponds to the maximal available
frequency (Turboboost frequency if available). For the Nic, this Mode is the same
as MODE_MAX. For the Disk, this Mode is the same as MODE_MAX.

Definition at line 123 of file defines.h.

**2.1.1.17   #define MODE␣MAX 3**

Mode marking a high utilization for the device - device must be switched to the high performance mode. For the Cpu, this Mode corresponds to the maximal available frequency (excluding Turboboost frequency). For the Nic, this Mode corresponds to maximal speed and full duplex. For the Disk, this Mode wakes the Disk if in MODE_MIN or MODE_UNUSED.

Definition at line 131 of file defines.h.

**2.1.1.18   #define MODE␣MED 4**

Mode marking a mid-range utilization for the device - if possible, device can be switched to a mid-range performance mode. For the Cpu, this Mode corresponds to the midrange available frequency. For the Nic, this Mode corresponds to mid-range speed and full duplex. For the Disk, this Mode is the same as MODE_MAX.

Definition at line 139 of file defines.h.

**2.1.1.19   #define MODE␣MIN 5**

Mode marking a low utilization for the device - if possible, device can be switched to a low performance mode. For the Cpu, this Mode corresponds to the minimal available frequency. For the Nic, this Mode corresponds to minimal speed and full duplex. For the Disk, this Mode corresponds to hdparm -y, which force the IDE drive to immediately enter the low power consumption standby mode, usually causing it to spin down.

Definition at line 147 of file defines.h.

**2.1.1.20   #define MODE␣UNUSED 6**

Mode marking the device as unused - which means a device can possibly be switched to sleep. For the CPU, this is Mode is the same as MODE_MIN. For the NIC, this Mode corresponds to minimal speed and half duplex. For the Disk, this Mode corresponds to hdparm -Y, which forces the IDE drive to immediately enter the lowest power consumption sleep mode.

Definition at line 155 of file defines.h.

## 2.2   include/eed.h File Reference

```
#include "defines.h"
```

**Functions**

- void ee_init (int ∗argc, char ∗∗∗argv, int tag, int rank)
- void ee_init_rms (int ∗argc, char ∗∗∗argv, int rank)

- void ee_init_rms_all (int *argc, char ***argv)
- void ee_dev_mode (int device_id, int mode_id)
- void ee_dev_mode_in (int device_id, int mode_id, int secs)
- void ee_finalize ()
- void ee_finalize_all (int tag)

## 2.2.1 Function Documentation

### 2.2.1.1 void ee_init ( int * *argc,* char *** *argv,* int *tag,* int *rank* )

Registers the process at the Daemon.

Registering possibly triggers a mode change corresponding to resource_process_list().

If the connection to the server process fails, all following interface calls will be ignored.

Tag and Rank only needed for instantiation - all future interface calls are identified by the socket connection.

**Parameters**

| | |
|---:|:---|
| *argc* | Pointer to count of commandline args |
| *argv* | Pointer to commandline args |
| *tag* | Tag for this process, must be the same for all processes in one job (e.g. the torque jobid) |
| *rank* | Rank for this process, e.g. the MPI rank |

### 2.2.1.2 void ee_init_rms ( int * *argc,* char *** *argv,* int *rank* )

Distincts the tag by reading the environment variable containing the resource management system jobid. Calls ee_init(). See ee_init() for details.

**Parameters**

| | |
|---:|:---|
| *argc* | Pointer to count of commandline args |
| *argv* | Pointer to commandline args |
| *rank* | Rank for this process, e.g. the MPI rank |

### 2.2.1.3 void ee_init_rms_all ( int * *argc,* char *** *argv* )

Distincts the tag by reading the environment variable containing the resource management system jobid. Calls ee_init() for each possible device with RMS_RANK. See ee_-init() for details.

**Parameters**

| | |
|---:|:---|
| *argc* | Pointer to count of commandline args |
| *argv* | Pointer to commandline args |

**2.2.1.4   void ee_dev_mode ( int *device_id,* int *mode_id* )**

Request Mode for device

**Parameters**

| | |
|---|---|
| *device_id* | Id for this device, e.g. CORE_ALL_ID (for all cores on the node), MY-_CORE_IDS (for the registered core) or a ID between CORE_MIN_ID and CORE_MAX_ID to identify a special device. |
| *mode_id* | Id for the desired mode, one out of MODE_UNUSED, MODE_MIN, M-ODE_MED, MODE_MAX and MODE_TURBO |

**2.2.1.5   void ee_dev_mode_in ( int *device_id,* int *mode_id,* int *secs* )**

Signalizes the device will be needed to be at least in the specified mode in secs seconds

**Parameters**

| | |
|---|---|
| *device_id* | Id for this device, e.g. CORE_ALL_ID (for all cores on the node), MY-_CORE_IDS (for the registered core) or a ID between CORE_MIN_ID and CORE_MAX_ID to identify a special device. |
| *mode_id* | Id for the desired mode, one out of MODE_UNUSED, MODE_MIN, M-ODE_MED, MODE_MAX and MODE_TURBO |
| *secs* | Seconds afterwards this device should be in MODE_MAX |

**2.2.1.6   void ee_finalize ( )**

Unregisters the process at the Daemon.

Unregistering possibly triggers a mode change corresponding to resource_process_-list().

**2.2.1.7   void ee_finalize_all ( int *tag* )**

Unregisters all processes with tag (all processes within a job)

Unregistering possibly triggers a mode change corresponding to resource_process_-list().