CaaS Xen Hypervisor

Generated by Doxygen 1.8.4

Tue Aug 13 2013 15:17:21

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 _SHA256_CTX Struct Reference

```
#include <sha2.h>
```

Data Fields

- uint32_t state [8]
- uint64_t bitcount
- uint8_t buffer [SHA256_BLOCK_LENGTH]

3.1.1 Detailed Description

NOTE: If your architecture does not define either u_intXX_t types or uintXX_t (from inttypes.h), you may need to define things by hand for your system: Most BSD systems already define u_intXX_t types, as does Linux. Some systems, however, like Compaq's Tru64 Unix instead can use uintXX_t types defined by very recent ANSI C standards and included in the file:

#include <inttypes.h>

If you choose to use <inttypes.h> then please define:

#define SHA2_USE_INTTYPES_H

Or on the command line during compile:

cc -DSHA2_USE_INTTYPES_H ...

3.1.2 Field Documentation

3.1.2.1 uint64_t bitcount

3.1.2.2 uint8_t buffer[SHA256_BLOCK_LENGTH]

3.1.2.3 uint32_t state[8]

The documentation for this struct was generated from the following file:

- xen/stubdom/domc/crypto/sha2.h

3.2 _SHA512_CTX Struct Reference

```
#include <sha2.h>
```

Data Fields

- uint64_t state [8]
- uint64_t bitcount [2]
- uint8_t buffer [SHA512_BLOCK_LENGTH]

3.2.1 Field Documentation

3.2.1.1 uint64_t bitcount[2]

3.2.1.2 uint8_t buffer[SHA512_BLOCK_LENGTH]

3.2.1.3 uint64_t state[8]

The documentation for this struct was generated from the following file:

- xen/stubdom/domc/crypto/sha2.h

3.3 blk_buffer Struct Reference

Data Fields

- void * page
- grant_ref_t gref

3.3.1 Field Documentation

3.3.1.1 grant_ref_t gref

3.3.1.2 void* page

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/blkfront.c

3.4 blk_req Struct Reference

Data Fields

- struct blkfront_aiocb aiocb
- int rand_value
- struct blk_req * next

3.4.1 Field Documentation

3.4.1.1 struct blkfront_aiocb aiocb

3.4.1.2 struct blk_req* next

3.4.1.3 int rand_value

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/kernel.c

3.5 blkback_aiocb Struct Reference

```
#include <blkback.h>
```

Data Fields

- struct blkback_dev * b_dev
- blkif_request_t * req
- struct blkfront_aiocb * f_aiocb
- uint8_t * pages_orig
- uint8_t * pages_new
- int pages_order

3.5.1 Detailed Description

Completely different that frontend aiocb

3.5.2 Field Documentation

3.5.2.1 struct blkback_dev* b_dev

3.5.2.2 struct blkfront_aiocb* f_aiocb

3.5.2.3 uint8_t* pages_new

3.5.2.4 int pages_order

3.5.2.5 uint8_t* pages_orig

3.5.2.6 blkif_request_t* req

The documentation for this struct was generated from the following file:

- xen/stubdom/domc/blkback.h

3.6 blkback_dev Struct Reference

```
#include <blkback.h>
```

Data Fields

- domid_t dom
- struct blkif_back_ring ring
- grant_ref_t ring_ref
- evtchn_port_t evtchn
- blkif_vdev_t handle
- char * nodename
- char * frontend
- struct blkback_info info
- xenbus_event_queue events
- struct thread * b_thread
- struct thread * f_thread
- struct blkfront_dev * f_dev
- struct gntmap gmap
- domc_info_t * domc_info
- evtchn_port_t local_port
- uint64_t count_read
- uint64_t count_write
- uint32_t outstanding

3.6.1 Field Documentation

3.6.1.1 struct thread* b_thread

3.6.1.2 uint64_t count_read

3.6.1.3 uint64_t count_write

3.6.1.4 domid_t dom

3.6.1.5 domc_info_t* domc_info

3.6.1.6 xenbus_event_queue events

3.6.1.7 evtchn_port_t evtchn

3.6.1.8 struct blkfront_dev* f_dev

3.6.1.9 struct thread* f_thread

3.6.1.10 char* frontend

3.6.1.11 struct gntmap gmap

3.6.1.12 blkif_vdev_t handle

3.6.1.13 struct blkback_info info

3.6.1.14 evtchn_port_t local_port

3.6.1.15 char* nodename

3.6.1.16 uint32_t outstanding

3.6.1.17 struct blkif_back_ring ring

3.6.1.18 grant_ref_t ring_ref

The documentation for this struct was generated from the following file:

- xen/stubdom/domc/blkback.h

3.7 blkback_info Struct Reference

```
#include <blkback.h>
```

Data Fields

- uint64_t sectors
- unsigned sector_size
- int mode
- int info
- int barrier
- int flush
- char * dev
- char * type
- char * hotplug_status
- int online
- int bootable
- int encrypt

3.7.1 Field Documentation

3.7.1.1 int barrier

3.7.1.2 int bootable

3.7.1.3 char* dev

3.7.1.4 int encrypt

3.7.1.5 int flush

3.7.1.6 char* hotplug_status

3.7.1.7 int info

3.7.1.8 int mode

3.7.1.9 int online

3.7.1.10 unsigned sector_size

3.7.1.11 uint64_t sectors

3.7.1.12 char* type

The documentation for this struct was generated from the following file:

- xen/stubdom/domc/blkback.h

3.8 blkfront_aiocb Struct Reference

```
#include <blkfront.h>
```

Data Fields

- struct blkfront_dev ∗ aio_dev
- uint8_t ∗ aio_buf
- size_t aio_nbytes
- off_t aio_offset
- size_t total_bytes
- uint8_t is_write
- void ∗ data
- grant_ref_t gref [BLKIF_MAX_SEGMENTS_PER_REQUEST]
- int n
- void(∗ aio_cb)(struct blkfront_aiocb ∗aiocb, int ret)

3.8.1 Field Documentation

3.8.1.1 uint8_t∗ aio_buf

3.8.1.2 void(∗ aio_cb)(struct blkfront_aiocb ∗aiocb, int ret)

3.8.1.3 struct blkfront_dev∗ aio_dev

3.8.1.4 size_t aio_nbytes

3.8.1.5 off_t aio_offset

3.8.1.6 void∗ data

3.8.1.7 grant_ref_t gref[BLKIF_MAX_SEGMENTS_PER_REQUEST]

3.8.1.8 uint8_t is_write

3.8.1.9 int n

3.8.1.10 size_t total_bytes

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/include/blkfront.h

3.9 blkfront_dev Struct Reference

Data Fields

- domid_t dom
- struct blkif_front_ring ring

- grant_ref_t ring_ref
- evtchn_port_t evtchn
- blkif_vdev_t handle
- char * nodename
- char * backend
- struct blkfront_info info
- xenbus_event_queue events

3.9.1 Field Documentation

3.9.1.1 char* backend

3.9.1.2 domid_t dom

3.9.1.3 xenbus_event_queue events

3.9.1.4 evtchn_port_t evtchn

3.9.1.5 blkif_vdev_t handle

3.9.1.6 struct blkfront_info info

3.9.1.7 char* nodename

3.9.1.8 struct blkif_front_ring ring

3.9.1.9 grant_ref_t ring_ref

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/blkfront.c

3.10 blkfront_info Struct Reference

```
#include <blkfront.h>
```

Data Fields

- uint64_t sectors
- unsigned sector_size
- int mode
- int info
- int barrier
- int flush

3.10.1 Field Documentation

3.10.1.1 int barrier

3.10.1.2 int flush

3.10.1.3 int info

3.10.1.4 int mode

3.10.1.5 unsigned sector_size

3.10.1.6 uint64_t sectors

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/include/blkfront.h

3.11 boot_video_info Struct Reference

Data Fields

- u8 orig_x
- u8 orig_y
- u8 orig_video_mode
- u8 orig_video_cols
- u8 orig_video_lines
- u8 orig_video_isVGA
- u16 orig_video_points
- u32 capabilities
- u16 lfb_linelength
- u16 lfb_width
- u16 lfb_height
- u16 lfb_depth
- u32 lfb_base
- u32 lfb_size
- u8 red_size
- u8 red_pos
- u8 green_size
- u8 green_pos
- u8 blue_size
- u8 blue_pos
- u8 rsvd_size
- u8 rsvd_pos
- u16 vesapm_seg
- u16 vesapm_off
- u16 vesa_attrib

3.11.1 Field Documentation

3.11.1.1 u8 blue_pos

3.11.1.2 u8 blue_size

3.11.1.3 u32 capabilities

3.11.1.4 u8 green_pos

3.11.1.5 u8 green_size

3.11.1.6 u32 lfb_base

3.11.1.7 u16 lfb_depth

3.11.1.8 u16 lfb_height

3.11.1.9 u16 lfb_linelength

3.11.1.10 u32 lfb_size

3.11.1.11 u16 lfb_width

3.11.1.12 u8 orig_video_cols

3.11.1.13 u8 orig_video_isVGA

3.11.1.14 u8 orig_video_lines

3.11.1.15 u8 orig_video_mode

3.11.1.16 u16 orig_video_points

3.11.1.17 u8 orig_x

3.11.1.18 u8 orig_y

3.11.1.19 u8 red_pos

3.11.1.20 u8 red_size

3.11.1.21 u8 rsvd_pos

3.11.1.22 u8 rsvd_size

3.11.1.23 u16 vesa_attrib

3.11.1.24 u16 vesapm_off

3.11.1.25 u16 vesapm_seg

The documentation for this struct was generated from the following file:

- xen/xen/arch/x86/setup.c

3.12 buffer Struct Reference

Data Fields

- char * data
- size_t consumed
- size_t size
- size_t capacity
- size_t max_capacity

3.12.1 Field Documentation

3.12.1.1 size_t capacity

3.12.1.2 size_t consumed

3.12.1.3 char* data

3.12.1.4 size_t max_capacity

3.12.1.5 size_t size

The documentation for this struct was generated from the following file:

- xen/tools/console/daemon/io.c

3.13 buffer_t Struct Reference

```
#include <buffer.h>
```

Data Fields

- tpm_size_t size
- tpm_size_t alloc_size
- BYTE * bytes
- BOOL is_owner

3.13.1 Field Documentation

3.13.1.1 tpm_size_t alloc_size

3.13.1.2 BYTE* bytes

3.13.1.3 BOOL is_owner

3.13.1.4 tpm_size_t size

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/util/buffer.h

3.14 caas_op Struct Reference

```
#include <caas_op.h>
```

Data Fields

- uint32_t cmd
- uint32_t size
- char * buf

3.14.1 Field Documentation

3.14.1.1 char* buf

3.14.1.2 uint32_t cmd

3.14.1.3 uint32_t size

The documentation for this struct was generated from the following file:

- xen/xen/include/public/xsm/caas_op.h

3.15 cap_t Union Reference

Data Fields

- struct permanent_flags_t perm_flags
- struct stclear_flags_t stclear_flags
- bool owned
- uint32_t num_pcrs
- struct tpm_version_t tpm_version
- struct tpm_version_1_2_t tpm_version_1_2
- uint32_t manufacturer_id
- struct timeout_t timeout
- struct duration_t duration

3.15.1 Field Documentation

3.15.1.1 struct duration_t duration

3.15.1.2 uint32_t manufacturer_id

3.15.1.3 uint32_t num_pcrs

3.15.1.4 bool owned

3.15.1.5 struct permanent_flags_t perm_flags

3.15.1.6 struct stclear_flags_t stclear_flags

3.15.1.7 struct timeout_t timeout

3.15.1.8 struct tpm_version_t tpm_version

3.15.1.9 struct tpm_version_1_2_t tpm_version_1_2

The documentation for this union was generated from the following file:

- xen/extras/mini-os/tpm_tis.c

3.16 changed_domain Struct Reference

Data Fields

- struct list_head list
- unsigned int domid
- int nbentry

3.16.1 Field Documentation

3.16.1.1 unsigned int domid

3.16.1.2 struct list_head list

3.16.1.3 int nbentry

The documentation for this struct was generated from the following file:

- xen/tools/xenstore/xenstored_transaction.c

3.17 changed_node Struct Reference

Data Fields

- struct list_head list
- char * node
- bool recurse

3.17.1 Field Documentation

3.17.1.1 struct list_head list

3.17.1.2 char* node

3.17.1.3 bool recurse

The documentation for this struct was generated from the following file:

- xen/tools/xenstore/xenstored_transaction.c

3.18 cipherInstance Struct Reference

```
#include <rijndael-api-fst.h>
```

Data Fields

- BYTE mode
- BYTE IV_old [MAX_IV_SIZE]
- BYTE * IV

3.18.1 Field Documentation

3.18.1.1 BYTE* IV

3.18.1.2 BYTE IV_old[MAX_IV_SIZE]

3.18.1.3 BYTE mode

The documentation for this struct was generated from the following file:

- xen/stubdom/domc/crypto/rijndael-api-fst.h

3.19 CRYPTO_INFO Struct Reference

```
#include <crypto.h>
```

Data Fields

- void * keyInfo
- UINT32 algorithmID
- UINT32 encScheme
- UINT32 sigScheme

3.19.1 Field Documentation

3.19.1.1 UINT32 algorithmID

3.19.1.2 UINT32 encScheme

3.19.1.3 void* keyInfo

3.19.1.4 UINT32 sigScheme

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/crypto/crypto.h

3.20 cspipe Struct Reference

```
#include <cspipe_intf.h>
```

Data Fields

- uint32_t magic_nr
- uint32_t domid_domu
- uint32_t domid_domc
- char s_key [128/8]

3.20.1 Detailed Description

CaaS pipe info, placed in the DomC memory by the domain builder.

3.20.2 Field Documentation

3.20.2.1 uint32_t domid_domc

3.20.2.2 uint32_t domid_domu

3.20.2.3 uint32_t magic_nr

3.20.2.4 char s_key[128/8]

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/cspipe_intf.h

3.21 dom0_vga_console_info Struct Reference

```
#include <xen.h>
```

Data Fields

- uint8_t video_type
- union {
  - struct {
    - uint16_t font_height
    - uint16_t cursor_x
    - uint16_t cursor_y
    - uint16_t rows
    - uint16_t columns
  - } text_mode_3
  - struct {
    - uint16_t width
    - uint16_t height
    - uint16_t bytes_per_line
    - uint16_t bits_per_pixel
    - uint32_t lfb_base
    - uint32_t lfb_size
    - uint8_t red_pos
    - uint8_t red_size
    - uint8_t green_pos
    - uint8_t green_size
    - uint8_t blue_pos
    - uint8_t blue_size
    - uint8_t rsvd_pos
    - uint8_t rsvd_size
  - } vesa_lfb
- } u

3.21.1 Field Documentation

3.21.1.1 uint16_t bits_per_pixel

3.21.1.2 uint8_t blue_pos

3.21.1.3 uint8_t blue_size

3.21.1.4 uint16_t bytes_per_line

3.21.1.5 uint16_t columns

3.21.1.6 uint16_t cursor_x

3.21.1.7 uint16_t cursor_y

3.21.1.8 uint16_t font_height

3.21.1.9 uint8_t green_pos

3.21.1.10 uint8_t green_size

3.21.1.11 uint16_t height

3.21.1.12 uint32_t lfb_base

3.21.1.13 uint32_t lfb_size

3.21.1.14 uint8_t red_pos

3.21.1.15 uint8_t red_size

3.21.1.16 uint16_t rows

3.21.1.17 uint8_t rsvd_pos

3.21.1.18 uint8_t rsvd_size

3.21.1.19 struct { ... } text_mode_3

3.21.1.20 union { ... } u

3.21.1.21 struct { ... } vesa_lfb

3.21.1.22 uint8_t video_type

3.21.1.23 uint16_t width

The documentation for this struct was generated from the following file:

- xen/xen/include/public/xen.h

3.22 domain Struct Reference

```
#include <sched.h>
```

Public Types

- enum { DOMDYING_alive, DOMDYING_dying, DOMDYING_dead }

Public Member Functions

- DECLARE_BITMAP (poll_mask, MAX_VIRT_CPUS)

Data Fields

- int domid
- int master_fd
- int slave_fd
- int log_fd
- bool is_dead
- struct buffer buffer
- struct domain ∗ next
- char ∗ conspath
- int ring_ref
- evtchn_port_or_error_t local_port
- evtchn_port_or_error_t remote_port
- xc_evtchn ∗ xce_handle
- struct xencons_interface ∗ interface
- int event_count
- long long next_period
- struct list_head list
- unsigned int domid
- evtchn_port_t port
- evtchn_port_t remote_port
- unsigned long mfn
- char ∗ path
- struct xenstore_domain_interface ∗ interface
- struct connection ∗ conn
- int shutdown
- int nbentry
- int nbwatch
- domid_t domain_id
- shared_info_t ∗ shared_info
- spinlock_t domain_lock
- spinlock_t page_alloc_lock
- struct page_list_head page_list
- struct page_list_head xenpage_list
- unsigned int tot_pages
- unsigned int max_pages
- atomic_t shr_pages
- unsigned int xenheap_pages
- unsigned int max_vcpus
- void ∗ sched_priv
- struct cpupool ∗ cpupool
- struct domain ∗ next_in_list
- struct domain ∗ next_in_hashbucket
- struct list_head rangesets
- spinlock_t rangesets_lock
- struct evtchn ∗ evtchn [NR_EVTCHN_BUCKETS]
- spinlock_t event_lock
- struct grant_table ∗ grant_table
- unsigned int nr_pirqs
- u16 ∗ pirq_to_evtchn

- unsigned long * pirq_mask
- struct rangeset * iomem_caps
- struct rangeset * irq_caps
- bool_t is_hvm
- bool_t need_iommu
- bool_t is_privileged
- struct domain * target
- bool_t debugger_attached
- enum domain:: { ... } is_dying
- bool_t is_paused_by_controller
- bool_t is_pinned
- spinlock_t shutdown_lock
- bool_t is_shutting_down
- bool_t is_shut_down
- int shutdown_code
- int suspend_evtchn
- atomic_t pause_count
- unsigned long vm_assist
- atomic_t refcnt
- struct vcpu ** vcpu
- cpumask_t domain_dirty_cpumask
- struct arch_domain arch
- void * ssid
- xen_domain_handle_t handle
- struct xenoprof * xenoprof
- int32_t time_offset_seconds
- spinlock_t watchdog_lock
- uint32_t watchdog_inuse_map
- struct timer watchdog_timer [NR_DOMAIN_WATCHDOG_TIMERS]
- struct rcu_head rcu
- spinlock_t hypercall_deadlock_mutex
- void * tmem
- struct lock_profile_qhead profile_head
- bool_t disable_migrate
- struct mem_event_domain mem_event
- nodemask_t node_affinity
- unsigned int last_alloc_node
- spinlock_t node_affinity_lock

3.22.1 Member Enumeration Documentation

3.22.1.1 anonymous enum

Enumerator

DOMDYING_alive

DOMDYING_dying

DOMDYING_dead

3.22.2 Member Function Documentation

3.22.2.1 DECLARE_BITMAP (poll_mask , MAX_VIRT_CPUS)

3.22.3 Field Documentation

3.22.3.1 struct arch_domain arch

3.22.3.2 struct buffer buffer

3.22.3.3 struct connection* conn

3.22.3.4 char* conspath

3.22.3.5 struct cpupool* cpupool

3.22.3.6 bool_t debugger_attached

3.22.3.7 bool_t disable_migrate

3.22.3.8 cpumask_t domain_dirty_cpumask

3.22.3.9 domid_t domain_id

3.22.3.10 spinlock_t domain_lock

3.22.3.11 unsigned int domid

3.22.3.12 int domid

3.22.3.13 int event_count

3.22.3.14 spinlock_t event_lock

3.22.3.15 struct evtchn* evtchn[NR_EVTCHN_BUCKETS]

3.22.3.16 struct grant_table* grant_table

3.22.3.17 xen_domain_handle_t handle

3.22.3.18 spinlock_t hypercall_deadlock_mutex

3.22.3.19 struct xenstore_domain_interface* interface

3.22.3.20 struct xencons_interface* interface

3.22.3.21 struct rangeset* iomem_caps

3.22.3.22 struct rangeset* irq_caps

3.22.3.23 bool is_dead

3.22.3.24 enum { ... } is_dying

3.22.3.25 bool_t is_hvm

3.22.3.26 bool_t is_paused_by_controller

3.22.3.27 bool_t is_pinned

3.22.3.28 bool_t is_privileged

3.22.3.29 bool_t is_shut_down

3.22.3.30 bool_t is_shutting_down

3.22.3.31 unsigned int last_alloc_node

3.22.3.32 struct list_head list

3.22.3.33 evtchn_port_or_error_t local_port

3.22.3.34 int log_fd

3.22.3.35 int master_fd

3.22.3.36 unsigned int max_pages

3.22.3.37 unsigned int max_vcpus

3.22.3.38 struct mem_event_domain mem_event

3.22.3.39 unsigned long mfn

3.22.3.40 int nbentry

3.22.3.41 int nbwatch

3.22.3.42 bool_t need_iommu

3.22.3.43 struct domain* next

3.22.3.44 struct domain* next_in_hashbucket

3.22.3.45 struct domain* next_in_list

3.22.3.46 long long next_period

3.22.3.47 nodemask_t node_affinity

3.22.3.48 spinlock_t node_affinity_lock

3.22.3.49 unsigned int nr_pirqs

3.22.3.50 spinlock_t page_alloc_lock

3.22.3.51 struct page_list_head page_list

3.22.3.52 char* path

3.22.3.53 atomic_t pause_count

3.22.3.54 unsigned long* pirq_mask

3.22.3.55 u16* pirq_to_evtchn

3.22.3.56 evtchn_port_t port

3.22.3.57 struct lock_profile_qhead profile_head

3.22.3.58 struct list_head rangesets

3.22.3.59 spinlock_t rangesets_lock

3.22.3.60 struct rcu_head rcu

3.22.3.61 atomic_t refcnt

3.22.3.62 evtchn_port_t remote_port

3.22.3.63 evtchn_port_or_error_t remote_port

3.22.3.64 int ring_ref

3.22.3.65 void* sched_priv

3.22.3.66 shared_info_t* shared_info

3.22.3.67 atomic_t shr_pages

3.22.3.68 int shutdown

3.22.3.69 int shutdown_code

3.22.3.70 spinlock_t shutdown_lock

3.22.3.71 int slave_fd

3.22.3.72 void* ssid

3.22.3.73 int suspend_evtchn

3.22.3.74 struct domain* target

3.22.3.75 int32_t time_offset_seconds

3.22.3.76 void* tmem

3.22.3.77 unsigned int tot_pages

3.22.3.78 struct vcpu** vcpu

3.22.3.79 unsigned long vm_assist

3.22.3.80 uint32_t watchdog_inuse_map

3.22.3.81 spinlock_t watchdog_lock

3.22.3.82 struct timer watchdog_timer[NR_DOMAIN_WATCHDOG_TIMERS]

3.22.3.83 xc_evtchn* xce_handle

3.22.3.84 unsigned int xenheap_pages

3.22.3.85 struct xenoprof* xenoprof

3.22.3.86 struct page_list_head xenpage_list

The documentation for this struct was generated from the following files:

- xen/tools/console/daemon/io.c
- xen/xen/include/xen/sched.h
- xen/tools/xenstore/xenstored_domain.c

3.23 domain_create Struct Reference

Data Fields

- int debug
- int daemonize
- int paused
- int dryrun
- int quiet
- int console_autoconnect
- const char * config_file
- const char * extra_config
- const char * restore_file
- int migrate_fd
- char ** migration_domname_r

3.23.1 Field Documentation

3.23.1.1 const char* config_file

3.23.1.2 int console_autoconnect

3.23.1.3 int daemonize

3.23.1.4 int debug

3.23.1.5 int dryrun

3.23.1.6 const char* extra_config

3.23.1.7 int migrate_fd

3.23.1.8 char** migration_domname_r

3.23.1.9 int paused

3.23.1.10 int quiet

3.23.1.11 const char* restore_file

The documentation for this struct was generated from the following file:

- xen/tools/libxl/xl_cmdimpl.c

3.24 domain_setup_info Struct Reference

```
#include <sched.h>
```

Data Fields

- unsigned long image_addr
- unsigned long image_len
- unsigned long v_start
- unsigned long v_end
- unsigned long v_kernstart
- unsigned long v_kernend
- unsigned long v_kernentry
- unsigned int pae_kernel
- unsigned long elf_paddr_offset
- unsigned int load_symtab
- unsigned long symtab_addr
- unsigned long symtab_len

3.24.1 Field Documentation

3.24.1.1 unsigned long elf_paddr_offset

3.24.1.2 unsigned long image_addr

3.24.1.3 unsigned long image_len

3.24.1.4 unsigned int load_symtab

3.24.1.5 unsigned int pae_kernel

3.24.1.6 unsigned long symtab_addr

3.24.1.7 unsigned long symtab_len

3.24.1.8 unsigned long v_end

3.24.1.9 unsigned long v_kernend

3.24.1.10 unsigned long v_kernentry

3.24.1.11 unsigned long v_kernstart

3.24.1.12 unsigned long v_start

The documentation for this struct was generated from the following file:

- xen/xen/include/xen/sched.h

3.25 domc_info Struct Reference

```
#include <main.h>
```

Data Fields

- domid_t self
- domid_t target

3.25.1 Detailed Description

Custom types. Use domc_ prefixes to avoid clashes.

3.25.2 Field Documentation

3.25.2.1 domid_t self

3.25.2.2 domid_t target

The documentation for this struct was generated from the following file:

- xen/stubdom/domc/main.h

3.26 duration_t Struct Reference

Data Fields

- uint32_t tpm_short
- uint32_t tpm_medium
- uint32_t tpm_long

3.26.1 Field Documentation

3.26.1.1 uint32_t tpm_long

3.26.1.2 uint32_t tpm_medium

3.26.1.3 uint32_t tpm_short

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/tpm_tis.c

3.27 evtchn Struct Reference

```
#include <sched.h>
```

Data Fields

- u8 state
- u8 consumer_is_xen
- u16 notify_vcpu_id
- union {
  - struct {
    - domid_t remote_domid
  - } unbound
  - struct {
    - u16 remote_port
    - struct domain * remote_dom
  - } interdomain
  - struct {
    - u16 irq
    - u16 next_port
    - u16 prev_port
  - } pirq
  - u16 virq
- } u

3.27.1 Field Documentation

3.27.1.1 u8 consumer_is_xen

3.27.1.2 struct { ... } interdomain

3.27.1.3 u16 irq

3.27.1.4 u16 next_port

3.27.1.5 u16 notify_vcpu_id

3.27.1.6 struct { ... } pirq

3.27.1.7 u16 prev_port

3.27.1.8 struct domain* remote_dom

3.27.1.9 domid_t remote_domid

3.27.1.10 u16 remote_port

3.27.1.11 u8 state

3.27.1.12 union { ... } u

3.27.1.13 struct { ... } unbound

3.27.1.14 u16 virq

The documentation for this struct was generated from the following file:

- xen/xen/include/xen/sched.h

3.28 fe_gref_map Struct Reference

Data Fields

- struct vfs_gref_map ∗ gmap
- int gref
- int order
- void ∗ pgmap
- void ∗ pbuf

3.28.1 Detailed Description

tracks some administration for the frontend for read/write

3.28.2 Field Documentation

3.28.2.1 struct vfs_gref_map∗ gmap

3.28.2.2 int gref

3.28.2.3 int order

3.28.2.4 void ∗ pbuf

3.28.2.5 void∗ pgmap

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/integrated/vfs.c

3.29 keyInstance Struct Reference

```
#include <rijndael-api-fst.h>
```

Data Fields

- BYTE direction
- int keyLen
- char keyMaterial [MAX_KEY_SIZE+1]
- int Nr
- u32 rk [4 ∗(MAXNR+1)]
- u32 ek [4 ∗(MAXNR+1)]

3.29.1 Field Documentation

3.29.1.1 BYTE direction

3.29.1.2 u32 ek[4 ∗(MAXNR+1)]

3.29.1.3 int keyLen

3.29.1.4 char keyMaterial[MAX_KEY_SIZE+1]

3.29.1.5 int Nr

3.29.1.6 u32 rk[4 ∗(MAXNR+1)]

The documentation for this struct was generated from the following file:

- xen/stubdom/domc/crypto/rijndael-api-fst.h

3.30 kvec Struct Reference

```
#include <lib.h>
```

Data Fields

- void ∗ iov_base
- size_t iov_len

3.30.1 Field Documentation

3.30.1.1 void∗ iov_base

3.30.1.2 size_t iov_len

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/include/lib.h

3.31 libxl__cpuid_policy Struct Reference

```
#include <libxl_internal.h>
```

Data Fields

- uint32_t input [2]
- char ∗ policy [4]

3.31.1 Field Documentation

3.31.1.1 uint32_t input[2]

3.31.1.2 char∗ policy[4]

The documentation for this struct was generated from the following file:

- xen/tools/libxl/libxl_internal.h

3.32 libxl__device Struct Reference

```
#include <libxl_internal.h>
```

Data Fields

- uint32_t backend_devid
- uint32_t backend_domid
- uint32_t devid
- uint32_t domid
- libxl__device_kinds backend_kind
- libxl__device_kinds kind

3.32.1 Field Documentation

3.32.1.1 uint32_t backend_devid

3.32.1.2 uint32_t backend_domid

3.32.1.3 libxl__device_kinds backend_kind

3.32.1.4 uint32_t devid

3.32.1.5 uint32_t domid

3.32.1.6 libxl__device_kinds kind

The documentation for this struct was generated from the following file:

- xen/tools/libxl/libxl_internal.h

3.33 libxl__device_model_starting Struct Reference

```
#include <libxl_internal.h>
```

Data Fields

- libxl__spawn_starting * for_spawn
- char * dom_path
- int domid

3.33.1 Field Documentation

3.33.1.1 char* dom_path

3.33.1.2 int domid

3.33.1.3 libxl__spawn_starting* for_spawn

The documentation for this struct was generated from the following file:

- xen/tools/libxl/libxl_internal.h

3.34 libxl__gc Struct Reference

```
#include <libxl_internal.h>
```

Data Fields

- int alloc_maxsize
- void ** alloc_ptrs
- libxl_ctx * owner

3.34.1 Field Documentation

3.34.1.1 int alloc_maxsize

3.34.1.2 void** alloc_ptrs

3.34.1.3 libxl_ctx* owner

The documentation for this struct was generated from the following file:

- xen/tools/libxl/libxl_internal.h

3.35 libxl__spawn_starting Struct Reference

```
#include <libxl_internal.h>
```

Data Fields

- pid_t intermediate
- char * what

3.35.1 Field Documentation

3.35.1.1 pid_t intermediate

3.35.1.2 char* what

The documentation for this struct was generated from the following file:

- xen/tools/libxl/libxl_internal.h

3.36 libxl__xen_console_reader Struct Reference

```
#include <libxl_internal.h>
```

Data Fields

- char * buffer
- unsigned int size

- unsigned int count
- unsigned int clear
- unsigned int incremental
- unsigned int index

3.36.1 Field Documentation

3.36.1.1 char* buffer

3.36.1.2 unsigned int clear

3.36.1.3 unsigned int count

3.36.1.4 unsigned int incremental

3.36.1.5 unsigned int index

3.36.1.6 unsigned int size

The documentation for this struct was generated from the following file:

- xen/tools/libxl/libxl_internal.h

3.37 libxl_cpuarray Struct Reference

```
#include <libxl.h>
```

Data Fields

- uint32_t entries
- uint32_t * array

3.37.1 Field Documentation

3.37.1.1 uint32_t* array

3.37.1.2 uint32_t entries

The documentation for this struct was generated from the following file:

- xen/tools/libxl/libxl.h

3.38 libxl_cpumap Struct Reference

```
#include <libxl.h>
```

Data Fields

- uint32_t size
- uint8_t * map

3.38.1 Field Documentation

3.38.1.1 uint8_t* map

3.38.1.2 uint32_t size

The documentation for this struct was generated from the following file:

- xen/tools/libxl/libxl.h

3.39 libxl_ctx Struct Reference

```
#include <libxl.h>
```

Data Fields

- xentoollog_logger * lg
- xc_interface * xch
- struct xs_handle * xsh
- int(* waitpid_instead)(pid_t pid, int *status, int flags)
- libxl_version_info version_info

3.39.1 Field Documentation

3.39.1.1 xentoollog_logger* lg

3.39.1.2 libxl_version_info version_info

3.39.1.3 int(* waitpid_instead)(pid_t pid, int *status, int flags)

3.39.1.4 xc_interface* xch

3.39.1.5 struct xs_handle* xsh

The documentation for this struct was generated from the following file:

- xen/tools/libxl/libxl.h

3.40 libxl_domain_config Struct Reference

```
#include <libxl.h>
```

Data Fields

- libxl_domain_create_info c_info
- libxl_domain_build_info b_info
- libxl_device_model_info dm_info
- int num_disks
- int num_vifs
- int num_vif2s
- int num_pcidevs

- int num_vfbs
- int num_vkbs
- int num_domc_disks
- int num_domc_vifs
- libxl_device_disk * disks
- libxl_device_nic * vifs
- libxl_device_net2 * vif2s
- libxl_device_pci * pcidevs
- libxl_device_vfb * vfbs
- libxl_device_vkb * vkbs
- libxl_device_disk * domc_disks
- libxl_device_nic * domc_vifs
- enum libxl_action_on_shutdown on_poweroff
- enum libxl_action_on_shutdown on_reboot
- enum libxl_action_on_shutdown on_watchdog
- enum libxl_action_on_shutdown on_crash

3.40.1 Field Documentation

3.40.1.1 libxl_domain_build_info b_info

3.40.1.2 libxl_domain_create_info c_info

3.40.1.3 libxl_device_disk* disks

3.40.1.4 libxl_device_model_info dm_info

3.40.1.5 libxl_device_disk* domc_disks

3.40.1.6 libxl_device_nic* domc_vifs

3.40.1.7 int num_disks

3.40.1.8 int num_domc_disks

3.40.1.9 int num_domc_vifs

3.40.1.10 int num_pcidevs

3.40.1.11 int num_vfbs

3.40.1.12 int num_vif2s

3.40.1.13 int num_vifs

3.40.1.14 int num_vkbs

3.40.1.15 enum libxl_action_on_shutdown on_crash

3.40.1.16 enum libxl_action_on_shutdown on_poweroff

3.40.1.17 enum libxl_action_on_shutdown on_reboot

3.40.1.18 enum libxl_action_on_shutdown on_watchdog

3.40.1.19 libxl_device_pci* pcidevs

3.40.1.20 libxl_device_vfb* vfbs

3.40.1.21 libxl_device_net2* vif2s

3.40.1.22 libxl_device_nic* vifs

3.40.1.23 libxl_device_vkb* vkbs

The documentation for this struct was generated from the following file:

- xen/tools/libxl/libxl.h

3.41 libxl_domain_suspend_info Struct Reference

```
#include <libxl.h>
```

Data Fields

- int flags
- int(* suspend_callback)(void *, int)

3.41.1 Field Documentation

3.41.1.1 int flags

3.41.1.2 int(* suspend_callback)(void *, int)

The documentation for this struct was generated from the following file:

- xen/tools/libxl/libxl.h

3.42 libxl_event Struct Reference

```
#include <libxl.h>
```

Data Fields

- libxl_event_type type
- char * path
- char * token

3.42.1 Field Documentation

3.42.1.1 char* path

3.42.1.2 char* token

3.42.1.3 libxl_event_type type

The documentation for this struct was generated from the following file:

- xen/tools/libxl/libxl.h

3.43 libxl_file_reference Struct Reference

```
#include <libxl.h>
```

Data Fields

- char * path
- int mapped
- void * data
- size_t size

3.43.1 Field Documentation

3.43.1.1 void* data

3.43.1.2 int mapped

3.43.1.3 char* path

3.43.1.4 size_t size

The documentation for this struct was generated from the following file:

- xen/tools/libxl/libxl.h

3.44 libxl_waiter Struct Reference

```
#include <libxl.h>
```

Data Fields

- char * path
- char * token

3.44.1 Field Documentation

3.44.1.1 char* path

3.44.1.2 char* token

The documentation for this struct was generated from the following file:

- xen/tools/libxl/libxl.h

3.45 mem_event_domain Struct Reference

```
#include <sched.h>
```

Data Fields

- spinlock_t ring_lock
- mem_event_shared_page_t * shared_page
- void * ring_page
- mem_event_front_ring_t front_ring
- int xen_port

3.45.1 Field Documentation

3.45.1.1 mem_event_front_ring_t front_ring

3.45.1.2 spinlock_t ring_lock

3.45.1.3 void* ring_page

3.45.1.4 mem_event_shared_page_t* shared_page

3.45.1.5 int xen_port

The documentation for this struct was generated from the following file:

- xen/xen/include/xen/sched.h

3.46 memop_args Struct Reference

Data Fields

- struct domain * domain

3.46.1 Field Documentation

3.46.1.1 struct domain* domain

The documentation for this struct was generated from the following file:

- xen/xen/common/memory.c

3.47 memory_map_context Struct Reference

Data Fields

- unsigned int n
- unsigned long s
- struct xen_memory_map map

3.47.1 Field Documentation

3.47.1.1 struct xen_memory_map map

3.47.1.2 unsigned int n

3.47.1.3 unsigned long s

The documentation for this struct was generated from the following file:

- xen/xen/arch/x86/mm.c

3.48 migrate_info Struct Reference

Data Fields

- long(∗ func)(void ∗data)
- void ∗ data
- struct vcpu ∗ vcpu
- unsigned int cpu
- unsigned int nest

3.48.1 Field Documentation

3.48.1.1 unsigned int cpu

3.48.1.2 void∗ data

3.48.1.3 long(∗ func)(void ∗data)

3.48.1.4 unsigned int nest

3.48.1.5 struct vcpu∗ vcpu

The documentation for this struct was generated from the following file:

- xen/xen/common/domain.c

3.49 mmu_update Struct Reference

```
#include <xen.h>
```

Data Fields

- uint64_t ptr
- uint64_t val

3.49.1 Field Documentation

3.49.1.1 uint64_t ptr

3.49.1.2 uint64_t val

The documentation for this struct was generated from the following file:

- xen/xen/include/public/xen.h

3.50 mmuext_op Struct Reference

```
#include <xen.h>
```

Data Fields

- unsigned int cmd
- union {
  xen_pfn_t mfn
  unsigned long linear_addr
  } arg1

- union {
  unsigned int nr_ents
  const void * vcpumask
  xen_pfn_t src_mfn
  } arg2

3.50.1 Field Documentation

3.50.1.1 union { ... } arg1

3.50.1.2 union { ... } arg2

3.50.1.3 unsigned int cmd

3.50.1.4 unsigned long linear_addr

3.50.1.5 xen_pfn_t mfn

3.50.1.6 unsigned int nr_ents

3.50.1.7 xen_pfn_t src_mfn

3.50.1.8 const void* vcpumask

The documentation for this struct was generated from the following file:

- xen/xen/include/public/xen.h

3.51 multicall_entry Struct Reference

```
#include <xen.h>
```

Data Fields

- unsigned long op
- unsigned long result
- unsigned long args [6]

3.51.1 Field Documentation

3.51.1.1 unsigned long args[6]

3.51.1.2 unsigned long op

3.51.1.3 unsigned long result

The documentation for this struct was generated from the following file:

- xen/xen/include/public/xen.h

3.52 oiap_sess Struct Reference

```
#include <tc.h>
```

Data Fields

- uint32_t handle
- unsigned char enonce [TPM_DIGEST_SIZE]

3.52.1 Field Documentation

3.52.1.1 unsigned char enonce[TPM_DIGEST_SIZE]

3.52.1.2 uint32_t handle

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/tc.h

3.53 Opt_args Struct Reference

```
#include <vtpm.h>
```

Public Types

- enum Mode { MODE_INVALID, MODE_PVM, MODE_HVM }
- enum StartUp { ST_CLEAR = 1, ST_SAVE = 2, ST_DEACTIVATED = 3 }
- enum EntropySource { ENT_TPM, ENT_RAND }

Data Fields

- enum Opt_args::Mode mode
- enum Opt_args::StartUp startup
- enum Opt_args::EntropySource entropysrc
- int loglevel

3.53.1 Detailed Description

Global commandline options

3.53.2 Member Enumeration Documentation

3.53.2.1 enum EntropySource

Enumerator

ENT_TPM

ENT_RAND

3.53.2.2 enum Mode

Enumerator

MODE_INVALID

MODE_PVM

MODE_HVM

3.53.2.3 enum StartUp

Enumerator

ST_CLEAR

ST_SAVE

ST_DEACTIVATED

3.53.3 Field Documentation

3.53.3.1 enum Opt_args::EntropySource entropysrc

3.53.3.2 int loglevel

3.53.3.3 enum Opt_args::Mode mode

3.53.3.4 enum Opt_args::StartUp startup

The documentation for this struct was generated from the following file:

- xen/stubdom/vtpm/vtpm.h

3.54 osap_sess Struct Reference

```
#include <tc.h>
```

Data Fields

- uint32_t handle
- unsigned char secret [TPM_DIGEST_SIZE]
- unsigned char enonce [TPM_DIGEST_SIZE]

3.54.1 Field Documentation

3.54.1.1 unsigned char enonce[TPM_DIGEST_SIZE]

3.54.1.2 uint32_t handle

3.54.1.3 unsigned char secret[TPM_DIGEST_SIZE]

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/tc.h

3.55 pack_buf_t Struct Reference

```
#include <tcg.h>
```

Data Fields

- UINT32 size
- BYTE * data

3.55.1 Field Documentation

3.55.1.1 BYTE* data

3.55.1.2 UINT32 size

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/util/tcg.h

3.56 pack_constbuf_t Struct Reference

```
#include <tcg.h>
```

Data Fields

- UINT32 size
- const BYTE * data

3.56.1 Field Documentation

3.56.1.1 const BYTE* data

3.56.1.2 UINT32 size

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/util/tcg.h

3.57 permanent_flags_t Struct Reference

Data Fields

- uint16_t tag
- uint8_t disable
- uint8_t ownership
- uint8_t deactivated
- uint8_t readPubek
- uint8_t disableOwnerClear
- uint8_t allowMaintenance
- uint8_t physicalPresenceLifetimeLock
- uint8_t physicalPresenceHWEnable
- uint8_t physicalPresenceCMDEnable
- uint8_t CEKPUsed
- uint8_t TPMpost
- uint8_t TPMpostLock
- uint8_t FIPS
- uint8_t operator
- uint8_t enableRevokeEK
- uint8_t nvLocked
- uint8_t readSRKPub
- uint8_t tpmEstablished
- uint8_t maintenanceDone
- uint8_t disableFullDALogicInfo

3.57.1 Field Documentation

3.57.1.1 uint8_t allowMaintenance

3.57.1.2 uint8_t CEKPUsed

3.57.1.3 uint8_t deactivated

3.57.1.4 uint8_t disable

3.57.1.5 uint8_t disableFullDALogicInfo

3.57.1.6 uint8_t disableOwnerClear

3.57.1.7 uint8_t enableRevokeEK

3.57.1.8 uint8_t FIPS

3.57.1.9 uint8_t maintenanceDone

3.57.1.10 uint8_t nvLocked

3.57.1.11 uint8_t operator

3.57.1.12 uint8_t ownership

3.57.1.13 uint8_t physicalPresenceCMDEnable

3.57.1.14 uint8_t physicalPresenceHWEnable

3.57.1.15 uint8_t physicalPresenceLifetimeLock

3.57.1.16 uint8_t readPubek

3.57.1.17 uint8_t readSRKPub

3.57.1.18 uint16_t tag

3.57.1.19 uint8_t tpmEstablished

3.57.1.20 uint8_t TPMpost

3.57.1.21 uint8_t TPMpostLock

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/tpm_tis.c

3.58 ptwr_emulate_ctxt Struct Reference

Data Fields

- struct x86_emulate_ctxt ctxt
- unsigned long cr2
- l1_pgentry_t pte

3.58.1 Field Documentation

3.58.1.1 unsigned long cr2

3.58.1.2 struct x86_emulate_ctxt ctxt

3.58.1.3 l1_pgentry_t pte

The documentation for this struct was generated from the following file:

- xen/xen/arch/x86/mm.c

3.59 save_file_header Struct Reference

Data Fields

- char magic [32]
- uint32_t byteorder
- uint32_t mandatory_flags
- uint32_t optional_flags
- uint32_t optional_data_len

3.59.1 Field Documentation

3.59.1.1 uint32_t byteorder

3.59.1.2 char magic[32]

3.59.1.3 uint32_t mandatory_flags

3.59.1.4 uint32_t optional_data_len

3.59.1.5 uint32_t optional_flags

The documentation for this struct was generated from the following file:

- xen/tools/libxl/xl_cmdimpl.c

3.60 schedid_name Struct Reference

Data Fields

- char * name
- int id

3.60.1 Field Documentation

3.60.1.1 int id

3.60.1.2 char* name

The documentation for this struct was generated from the following file:

- xen/tools/libxl/libxl_utils.c

3.61 shared_info Struct Reference

```
#include <xen.h>
```

Data Fields

- struct vcpu_info vcpu_info [XEN_LEGACY_MAX_VCPUS]
- unsigned long evtchn_pending [sizeof(unsigned long)*8]
- unsigned long evtchn_mask [sizeof(unsigned long)*8]
- uint32_t wc_version

- uint32_t wc_sec
- uint32_t wc_nsec
- struct arch_shared_info arch

3.61.1 Field Documentation

3.61.1.1 struct arch_shared_info arch

3.61.1.2 unsigned long evtchn_mask[sizeof(unsigned long)∗8]

3.61.1.3 unsigned long evtchn_pending[sizeof(unsigned long)∗8]

3.61.1.4 struct vcpu_info vcpu_info[XEN_LEGACY_MAX_VCPUS]

3.61.1.5 uint32_t wc_nsec

3.61.1.6 uint32_t wc_sec

3.61.1.7 uint32_t wc_version

The documentation for this struct was generated from the following file:

- xen/xen/include/public/xen.h

3.62 shared_info_any_t Union Reference

```
#include <xenctrl.h>
```

Data Fields

- shared_info_t s

3.62.1 Field Documentation

3.62.1.1 shared_info_t s

The documentation for this union was generated from the following file:

- xen/tools/libxc/xenctrl.h

3.63 shpage Struct Reference

Data Fields

- void ∗ page
- grant_ref_t grantref

3.63.1 Field Documentation

3.63.1.1 grant_ref_t grantref

3.63.1.2 void* page

The documentation for this struct was generated from the following file:

- xen/stubdom/vtpm/vtpm_cmd.c

3.64 start_info Struct Reference

```
#include <xen.h>
```

Data Fields

- char magic [32]
- unsigned long nr_pages
- unsigned long shared_info
- uint32_t flags
- xen_pfn_t store_mfn
- uint32_t store_evtchn
- union {
  - struct {
    - xen_pfn_t mfn
    - uint32_t evtchn
  - } domU
  - struct {
    - uint32_t info_off
    - uint32_t info_size
  - } dom0
- } console

- unsigned long pt_base
- unsigned long nr_pt_frames
- unsigned long mfn_list
- unsigned long mod_start
- unsigned long mod_len
- int8_t cmd_line [MAX_GUEST_CMDLINE]
- unsigned long first_p2m_pfn
- unsigned long nr_p2m_frames
- unsigned long caas_mfn

3.64.1 Field Documentation

3.64.1.1 unsigned long caas_mfn

3.64.1.2 int8_t cmd_line[MAX_GUEST_CMDLINE]

3.64.1.3 union { ... } console

3.64.1.4 struct { ... } dom0

3.64.1.5 struct { ... } domU

3.64.1.6 uint32_t evtchn

3.64.1.7 unsigned long first_p2m_pfn

3.64.1.8 uint32_t flags

3.64.1.9 uint32_t info_off

3.64.1.10 uint32_t info_size

3.64.1.11 char magic[32]

3.64.1.12 xen_pfn_t mfn

3.64.1.13 unsigned long mfn_list

3.64.1.14 unsigned long mod_len

3.64.1.15 unsigned long mod_start

3.64.1.16 unsigned long nr_p2m_frames

3.64.1.17 unsigned long nr_pages

3.64.1.18 unsigned long nr_pt_frames

3.64.1.19 unsigned long pt_base

3.64.1.20 unsigned long shared_info

3.64.1.21 uint32_t store_evtchn

3.64.1.22 xen_pfn_t store_mfn

The documentation for this struct was generated from the following file:

- xen/xen/include/public/xen.h

3.65 start_info_any_t Union Reference

```
#include <xenctrl.h>
```

Data Fields

- start_info_t s

3.65.1 Field Documentation

3.65.1.1 start_info_t s

The documentation for this union was generated from the following file:

- xen/tools/libxc/xenctrl.h

3.66 stclear_flags_t Struct Reference

Data Fields

- uint16_t tag
- uint8_t deactivated
- uint8_t disableForceClear
- uint8_t physicalPresence
- uint8_t physicalPresenceLock
- uint8_t bGlobalLock

3.66.1 Field Documentation

3.66.1.1 uint8_t bGlobalLock

3.66.1.2 uint8_t deactivated

3.66.1.3 uint8_t disableForceClear

3.66.1.4 uint8_t physicalPresence

3.66.1.5 uint8_t physicalPresenceLock

3.66.1.6 uint16_t tag

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/tpm_tis.c

3.67 suspendinfo Struct Reference

Data Fields

- libxl__gc * gc
- xc_evtchn * xce
- int suspend_eventchn
- int domid
- int hvm
- unsigned int flags
- int guest_responded

3.67.1 Field Documentation

3.67.1.1 int domid

3.67.1.2 unsigned int flags

3.67.1.3 libxl__gc* gc

3.67.1.4 int guest_responded

3.67.1.5 int hvm

3.67.1.6 int suspend_eventchn

3.67.1.7 xc_evtchn* xce

The documentation for this struct was generated from the following file:

- xen/tools/libxl/libxl_dom.c

3.68 tc_info Struct Reference

```
#include <tc.h>
```

Data Fields

- struct tpm_chip * chip
- tpm_hmac_ctx_t * hmh

3.68.1 Detailed Description

build information used in nearly all TC funcs

3.68.2 Field Documentation

3.68.2.1 struct tpm_chip* chip

3.68.2.2 tpm_hmac_ctx_t* hmh

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/tc.h

3.69 tc_state Struct Reference

```
#include <tc.h>
```

Data Fields

- struct tc_info info
- struct oiap_sess * oi
- unsigned char srkpw [TPM_DIGEST_SIZE]
- unsigned char keypw [TPM_DIGEST_SIZE]
- struct TPM_KEY12 * key
- uint32_t loadedKeyHandle

3.69.1 Field Documentation

3.69.1.1 struct tc_info info

3.69.1.2 struct TPM_KEY12* key

3.69.1.3 unsigned char keypw[TPM_DIGEST_SIZE]

3.69.1.4 uint32_t loadedKeyHandle

3.69.1.5 struct oiap_sess* oi

3.69.1.6 unsigned char srkpw[TPM_DIGEST_SIZE]

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/tc.h

3.70 TCS_AUTH Struct Reference

```
#include <tcg.h>
```

Data Fields

- TCS_AUTHHANDLE AuthHandle
- TPM_NONCE NonceOdd
- TPM_NONCE NonceEven
- BOOL fContinueAuthSession
- TPM_AUTHDATA HMAC

3.70.1 Field Documentation

3.70.1.1 TCS_AUTHHANDLE AuthHandle

3.70.1.2 BOOL fContinueAuthSession

3.70.1.3 TPM_AUTHDATA HMAC

3.70.1.4 TPM_NONCE NonceEven

3.70.1.5 TPM_NONCE NonceOdd

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/util/tcg.h

3.71 tdVTPM_GLOBALS Struct Reference

```
#include <vtpmpriv.h>
```

Data Fields

- pthread_t master_thread
- pthread_t be_thread
- pthread_t hp_thread
- pthread_t dmi_thread
- int connected_dmis
- struct hashtable * dmi_map
- VTPM_MIGKEY_LIST * mig_keys
- TCS_CONTEXT_HANDLE manager_tcs_handle
- TPM_HANDLE storageKeyHandle
- CRYPTO_INFO storageKey
- CRYPTO_INFO bootKey
- TCS_AUTH keyAuth
- TPM_AUTHDATA owner_usage_auth
- buffer_t storageKeyWrap
- TPM_AUTHDATA srk_usage_auth
- TPM_AUTHDATA storage_key_usage_auth
- buffer_t bootKeyWrap

3.71.1 Field Documentation

3.71.1.1 pthread_t be_thread

3.71.1.2 CRYPTO_INFO bootKey

3.71.1.3 buffer_t bootKeyWrap

3.71.1.4 int connected_dmis

3.71.1.5 struct hashtable* dmi_map

3.71.1.6 pthread_t dmi_thread

3.71.1.7 pthread_t hp_thread

3.71.1.8 TCS_AUTH keyAuth

3.71.1.9 TCS_CONTEXT_HANDLE manager_tcs_handle

3.71.1.10 pthread_t master_thread

3.71.1.11 VTPM_MIGKEY_LIST* mig_keys

3.71.1.12 TPM_AUTHDATA owner_usage_auth

3.71.1.13 TPM_AUTHDATA srk_usage_auth

3.71.1.14 TPM_AUTHDATA storage_key_usage_auth

3.71.1.15 CRYPTO_INFO storageKey

3.71.1.16 TPM_HANDLE storageKeyHandle

3.71.1.17 buffer_t storageKeyWrap

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/manager/vtpmpriv.h

3.72 tdVTPM_MIGKEY_LIST Struct Reference

```
#include <vtpmpriv.h>
```

Data Fields

- UINT32 name_size
- BYTE * name
- CRYPTO_INFO key
- struct tdVTPM_MIGKEY_LIST * next

3.72.1 Field Documentation

3.72.1.1 CRYPTO_INFO key

3.72.1.2 BYTE* name

3.72.1.3 UINT32 name_size

3.72.1.4 struct tdVTPM_MIGKEY_LIST* next

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/manager/vtpmpriv.h

3.73 timeout_t Struct Reference

Data Fields

- uint32_t a
- uint32_t b
- uint32_t c
- uint32_t d

3.73.1 Field Documentation

3.73.1.1 uint32_t a

3.73.1.2 uint32_t b

3.73.1.3 uint32_t c

3.73.1.4 uint32_t d

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/tpm_tis.c

3.74 tmem_oid Struct Reference

```
#include <xenctrl.h>
```

Data Fields

- uint64_t oid [3]

3.74.1 Detailed Description

tmem operations

3.74.2 Field Documentation

3.74.2.1 uint64_t oid[3]

The documentation for this struct was generated from the following file:

- xen/tools/libxc/xenctrl.h

3.75 TPM_BOUND_DATA Struct Reference

```
#include <tcg.h>
```

Data Fields

- TPM_VERSION ver
- TPM_PAYLOAD_TYPE payload
- BYTE * payloadData

3.75.1 Field Documentation

3.75.1.1 TPM_PAYLOAD_TYPE payload

3.75.1.2 BYTE* payloadData

3.75.1.3 TPM_VERSION ver

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/util/tcg.h

3.76 tpm_chip Struct Reference

Data Fields

- struct minios_list_head list

- int enabled_localities
- int locality
- unsigned long baseaddr
- uint8_t * pages [5]
- int did
- int vid
- int rid
- uint8_t data_buffer [TPM_BUFSIZE]
- int data_len
- s_time_t timeout_a
- s_time_t timeout_b
- s_time_t timeout_c
- s_time_t timeout_d
- s_time_t duration [3]
- unsigned int irq
- struct wait_queue_head read_queue
- struct wait_queue_head int_queue

3.76.1 Field Documentation

3.76.1.1 unsigned long baseaddr

3.76.1.2 uint8_t data_buffer[TPM_BUFSIZE]

3.76.1.3 int data_len

3.76.1.4 int did

3.76.1.5 s_time_t duration[3]

3.76.1.6 int enabled_localities

3.76.1.7 struct wait_queue_head int_queue

3.76.1.8 unsigned int irq

3.76.1.9 struct minios_list_head list

3.76.1.10 int locality

3.76.1.11 uint8_t* pages[5]

3.76.1.12 struct wait_queue_head read_queue

3.76.1.13 int rid

3.76.1.14 s_time_t timeout_a

3.76.1.15 s_time_t timeout_b

3.76.1.16 s_time_t timeout_c

3.76.1.17 s_time_t timeout_d

3.76.1.18 int vid

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/tpm_tis.c

3.77 tpm_cmd Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- tpm_cmd_header header
- tpm_cmd_params params

3.77.1 Field Documentation

3.77.1.1 tpm_cmd_header header

3.77.1.2 tpm_cmd_params params

The documentation for this struct was generated from the following files:

- xen/extras/mini-os/tpm_tis.c
- xen/stubdom/domt/tpm_defs.h

3.78 tpm_cmd_header Union Reference

```
#include <tpm_defs.h>
```

Data Fields

- struct tpm_input_header in
- struct tpm_output_header out

3.78.1 Field Documentation

3.78.1.1 struct tpm_input_header in

3.78.1.2 struct tpm_output_header out

The documentation for this union was generated from the following files:

- xen/extras/mini-os/tpm_tis.c
- xen/stubdom/domt/tpm_defs.h

3.79 tpm_cmd_params Union Reference

```
#include <tpm_defs.h>
```

Data Fields

- struct tpm_getcap_params_out getcap_out
- struct tpm_readpubek_params_out readpubek_out
- uint8_t readpubek_out_buffer [sizeof(struct tpm_readpubek_params_out)]
- struct tpm_getcap_params_in getcap_in
- struct tpm_pcrread_in pcrread_in
- struct tpm_pcrread_out pcrread_out
- struct tpm_pcrextend_in pcrextend_in
- struct tpm_oiap_out oiap_out
- struct tpm_osap_in osap_in
- struct tpm_osap_out osap_out
- struct tpm_getrandom_in getrandom_in
- struct tpm_getrandom_out getrandom_out
- struct tpm_flushspecific_in flushspecific_in

3.79.1 Field Documentation

3.79.1.1 struct tpm_flushspecific_in flushspecific_in

3.79.1.2 struct tpm_getcap_params_in getcap_in

3.79.1.3 struct tpm_getcap_params_out getcap_out

3.79.1.4 struct tpm_getrandom_in getrandom_in

3.79.1.5 struct tpm_getrandom_out getrandom_out

3.79.1.6 struct tpm_oiap_out oiap_out

3.79.1.7 struct tpm_osap_in osap_in

3.79.1.8 struct tpm_osap_out osap_out

3.79.1.9 struct tpm_pcrextend_in pcrextend_in

3.79.1.10 struct tpm_pcrread_in pcrread_in

3.79.1.11 struct tpm_pcrread_out pcrread_out

3.79.1.12 struct tpm_readpubek_params_out readpubek_out

3.79.1.13 uint8_t readpubek_out_buffer

The documentation for this union was generated from the following files:

- xen/extras/mini-os/tpm_tis.c
- xen/stubdom/domt/tpm_defs.h

3.80 TPM_DIGEST Struct Reference

```
#include <tcg.h>
```

Data Fields

- BYTE digest [TPM_DIGEST_SIZE]

3.80.1 Field Documentation

3.80.1.1 BYTE digest[TPM_DIGEST_SIZE]

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/util/tcg.h

3.81 tpm_flushspecific_in Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- uint32_t handle
- uint32_t resourceType

3.81.1 Field Documentation

3.81.1.1 uint32_t handle

3.81.1.2 uint32_t resourceType

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/tpm_defs.h

3.82 tpm_getcap_params_in Struct Reference

Data Fields

- uint32_t cap
- uint32_t subcap_size
- uint32_t subcap

3.82.1 Field Documentation

3.82.1.1 uint32_t cap

3.82.1.2 uint32_t subcap

3.82.1.3 uint32_t subcap_size

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/tpm_tis.c

3.83 tpm_getcap_params_out Struct Reference

Data Fields

- uint32_t cap_size
- cap_t cap

3.83.1 Field Documentation

3.83.1.1 cap_t cap

3.83.1.2 uint32_t cap_size

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/tpm_tis.c

3.84 tpm_getrandom_in Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- uint32_t numbytes

3.84.1 Field Documentation

3.84.1.1 uint32_t numbytes

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/tpm_defs.h

3.85 tpm_getrandom_out Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- uint32_t numrandbytes
- uint8_t randbytes [MAX_RAND_SIZE]

3.85.1 Field Documentation

3.85.1.1 uint32_t numrandbytes

3.85.1.2 uint8_t randbytes[MAX_RAND_SIZE]

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/tpm_defs.h

3.86 tpm_hmac_ctx_t Struct Reference

```
#include <hmac.h>
```

Data Fields

- tpm_sha1_ctx_t ctx
- uint8_t k_opad [HMAC_PAD_LENGTH]

3.86.1 Field Documentation

3.86.1.1 tpm_sha1_ctx_t ctx

3.86.1.2 uint8_t k_opad[HMAC_PAD_LENGTH]

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/crypto/hmac.h

3.87 tpm_input_header Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- uint16_t tag
- uint32_t length
- uint32_t ordinal

3.87.1 Field Documentation

3.87.1.1 uint32_t length

3.87.1.2 uint32_t ordinal

3.87.1.3 uint16_t tag

The documentation for this struct was generated from the following files:

- xen/extras/mini-os/tpm_tis.c
- xen/stubdom/domt/tpm_defs.h

3.88 TPM_KEY Struct Reference

```
#include <tcg.h>
```

Data Fields

- TPM_VERSION ver
- TPM_KEY_USAGE keyUsage
- TPM_KEY_FLAGS keyFlags
- TPM_AUTH_DATA_USAGE authDataUsage
- TPM_KEY_PARMS algorithmParms
- UINT32 PCRInfoSize
- BYTE * PCRInfo
- TPM_STORE_PUBKEY pubKey
- UINT32 encDataSize
- BYTE * encData

3.88.1 Field Documentation

3.88.1.1 TPM_KEY_PARMS algorithmParms

3.88.1.2 TPM_AUTH_DATA_USAGE authDataUsage

3.88.1.3 BYTE* encData

3.88.1.4 UINT32 encDataSize

3.88.1.5 TPM_KEY_FLAGS keyFlags

3.88.1.6 TPM_KEY_USAGE keyUsage

3.88.1.7 BYTE* PCRInfo

3.88.1.8 UINT32 PCRInfoSize

3.88.1.9 TPM_STORE_PUBKEY pubKey

3.88.1.10 TPM_VERSION ver

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/util/tcg.h

3.89 TPM_KEY12 Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- uint16_t tag
- uint16_t fill
- uint16_t keyUsage
- uint32_t keyFlags
- uint8_t authDataUsage
- struct TPM_KEY_PARMS algorithmParms
- uint32_t PCRInfoSize
- uint8_t * PCRInfo

- struct TPM_STORE_PUBKEY pubKey
- uint32_t encDataSize
- uint8_t * encData

3.89.1 Field Documentation

3.89.1.1 struct TPM_KEY_PARMS algorithmParms

3.89.1.2 uint8_t authDataUsage

3.89.1.3 uint8_t* encData

3.89.1.4 uint32_t encDataSize

3.89.1.5 uint16_t fill

3.89.1.6 uint32_t keyFlags

3.89.1.7 uint16_t keyUsage

3.89.1.8 uint8_t* PCRInfo

3.89.1.9 uint32_t PCRInfoSize

3.89.1.10 struct TPM_STORE_PUBKEY pubKey

3.89.1.11 uint16_t tag

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/tpm_defs.h

3.90 TPM_KEY_PARMS Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- uint32_t algorithmID
- uint16_t encScheme
- uint16_t sigScheme
- uint32_t parmSize
- struct TPM_RSA_KEY_PARMS * parms
- TPM_ALGORITHM_ID algorithmID
- TPM_ENC_SCHEME encScheme
- TPM_SIG_SCHEME sigScheme
- UINT32 parmSize
- BYTE * parms

3.90.1 Field Documentation

3.90.1.1 TPM_ALGORITHM_ID algorithmID

3.90.1.2 uint32_t algorithmID

3.90.1.3 TPM_ENC_SCHEME encScheme

3.90.1.4 uint16_t encScheme

3.90.1.5 BYTE* parms

3.90.1.6 struct TPM_RSA_KEY_PARMS* parms

3.90.1.7 UINT32 parmSize

3.90.1.8 uint32_t parmSize

3.90.1.9 TPM_SIG_SCHEME sigScheme

3.90.1.10 uint16_t sigScheme

The documentation for this struct was generated from the following files:

- xen/stubdom/domt/tpm_defs.h
- xen/tools/vtpm_manager/util/tcg.h

3.91 TPM_NONCE Struct Reference

```
#include <tcg.h>
```

Data Fields

- BYTE nonce [TPM_DIGEST_SIZE]

3.91.1 Field Documentation

3.91.1.1 BYTE nonce[TPM_DIGEST_SIZE]

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/util/tcg.h

3.92 tpm_oiap_out Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- uint32_t authHandle
- uint8_t nonceEven [TPM_DIGEST_SIZE]

3.92.1 Field Documentation

3.92.1.1 uint32_t authHandle

3.92.1.2 uint8_t nonceEven[TPM_DIGEST_SIZE]

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/tpm_defs.h

3.93 tpm_osap_in Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- uint16_t entity_type
- uint32_t entity_value
- uint8_t nonce_odd_osap [TPM_DIGEST_SIZE]

3.93.1 Field Documentation

3.93.1.1 uint16_t entity_type

3.93.1.2 uint32_t entity_value

3.93.1.3 uint8_t nonce_odd_osap[TPM_DIGEST_SIZE]

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/tpm_defs.h

3.94 tpm_osap_out Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- uint32_t authhandle
- uint8_t nonce_even [TPM_DIGEST_SIZE]
- uint8_t nonce_even_osap [TPM_DIGEST_SIZE]

3.94.1 Field Documentation

3.94.1.1 uint32_t authhandle

3.94.1.2 uint8_t nonce_even[TPM_DIGEST_SIZE]

3.94.1.3 uint8_t nonce_even_osap[TPM_DIGEST_SIZE]

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/tpm_defs.h

3.95 tpm_output_header Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- uint16_t tag
- uint32_t length
- uint32_t return_code

3.95.1 Field Documentation

3.95.1.1 uint32_t length

3.95.1.2 uint32_t return_code

3.95.1.3 uint16_t tag

The documentation for this struct was generated from the following files:

- xen/extras/mini-os/tpm_tis.c
- xen/stubdom/domt/tpm_defs.h

3.96 TPM_PCR_COMPOSITE Struct Reference

```
#include <tcg.h>
```

Data Fields

- TPM_PCR_SELECTION select
- UINT32 valueSize
- TPM_PCRVALUE * pcrValue

3.96.1 Field Documentation

3.96.1.1 TPM_PCRVALUE* pcrValue

3.96.1.2 TPM_PCR_SELECTION select

3.96.1.3 UINT32 valueSize

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/util/tcg.h

3.97 TPM_PCR_INFO Struct Reference

```
#include <tcg.h>
```

Data Fields

- TPM_PCR_SELECTION pcrSelection
- TPM_COMPOSITE_HASH digestAtRelease
- TPM_COMPOSITE_HASH digestAtCreation

3.97.1 Field Documentation

3.97.1.1 TPM_COMPOSITE_HASH digestAtCreation

3.97.1.2 TPM_COMPOSITE_HASH digestAtRelease

3.97.1.3 TPM_PCR_SELECTION pcrSelection

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/util/tcg.h

3.98 TPM_PCR_SELECTION Struct Reference

```
#include <tcg.h>
```

Data Fields

- UINT16 sizeOfSelect
- BYTE * pcrSelect

  in bytes

3.98.1 Field Documentation

3.98.1.1 BYTE* pcrSelect

in bytes

3.98.1.2 UINT16 sizeOfSelect

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/util/tcg.h

3.99 tpm_pcrextend_in Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- uint32_t pcr_idx
- uint8_t hash [TPM_DIGEST_SIZE]

3.99.1 Field Documentation

3.99.1.1 uint8_t hash

3.99.1.2 uint32_t pcr_idx

The documentation for this struct was generated from the following files:

- xen/extras/mini-os/tpm_tis.c
- xen/stubdom/domt/tpm_defs.h

3.100 tpm_pcrread_in Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- uint32_t pcr_idx

3.100.1 Field Documentation

3.100.1.1 uint32_t pcr_idx

The documentation for this struct was generated from the following files:

- xen/extras/mini-os/tpm_tis.c
- xen/stubdom/domt/tpm_defs.h

3.101 tpm_pcrread_out Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- uint8_t pcr_result [TPM_DIGEST_SIZE]

3.101.1 Field Documentation

3.101.1.1 uint8_t pcr_result

The documentation for this struct was generated from the following files:

- xen/extras/mini-os/tpm_tis.c
- xen/stubdom/domt/tpm_defs.h

3.102 TPM_PUBKEY Struct Reference

```
#include <tcg.h>
```

Data Fields

- TPM_KEY_PARMS algorithmParms
- TPM_STORE_PUBKEY pubKey

3.102.1 Field Documentation

3.102.1.1 TPM_KEY_PARMS algorithmParms

3.102.1.2 TPM_STORE_PUBKEY pubKey

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/util/tcg.h

3.103 tpm_readpubek_params_out Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- uint8_t algorithm [4]
- uint8_t encscheme [2]
- uint8_t sigscheme [2]
- uint32_t paramsize
- uint8_t parameters [12]
- uint32_t keysize
- uint8_t modulus [256]
- uint8_t checksum [20]

3.103.1 Field Documentation

3.103.1.1 uint8_t algorithm

3.103.1.2 uint8_t checksum

3.103.1.3 uint8_t encscheme

3.103.1.4 uint32_t keysize

3.103.1.5 uint8_t modulus

3.103.1.6 uint8_t parameters

3.103.1.7 uint32_t paramsize

3.103.1.8 uint8_t sigscheme

The documentation for this struct was generated from the following files:

- xen/extras/mini-os/tpm_tis.c
- xen/stubdom/domt/tpm_defs.h

3.104 TPM_RSA_KEY_PARMS Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- uint32_t keyLength
- uint32_t numPrimes
- uint32_t exponentSize
- uint8_t * exponent
- UINT32 keyLength
- UINT32 numPrimes
- UINT32 exponentSize
- BYTE * exponent

3.104.1 Field Documentation

3.104.1.1 BYTE* exponent

3.104.1.2 uint8_t* exponent

3.104.1.3 UINT32 exponentSize

3.104.1.4 uint32_t exponentSize

3.104.1.5 UINT32 keyLength

3.104.1.6 uint32_t keyLength

3.104.1.7 UINT32 numPrimes

3.104.1.8 uint32_t numPrimes

The documentation for this struct was generated from the following files:

- xen/stubdom/domt/tpm_defs.h
- xen/tools/vtpm_manager/util/tcg.h

3.105 tpm_sha1_ctx_t Struct Reference

```
#include <sha1.h>
```

Data Fields

- uint32_t h [5]
- uint32_t count_lo
- uint32_t count_hi
- uint8_t buf [64]

3.105.1 Field Documentation

3.105.1.1 uint8_t buf[64]

3.105.1.2 uint32_t count_hi

3.105.1.3 uint32_t count_lo

3.105.1.4 uint32_t h[5]

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/crypto/sha1.h

3.106 TPM_STORE_PUBKEY Struct Reference

```
#include <tpm_defs.h>
```

Data Fields

- uint32_t keyLength
- uint8_t * key
- UINT32 keyLength
- BYTE * key

3.106.1 Field Documentation

3.106.1.1 BYTE* key

3.106.1.2 uint8_t* key

3.106.1.3 UINT32 keyLength

3.106.1.4 uint32_t keyLength

The documentation for this struct was generated from the following files:

- xen/stubdom/domt/tpm_defs.h
- xen/tools/vtpm_manager/util/tcg.h

3.107 TPM_STORED_DATA Struct Reference

```
#include <tcg.h>
```

Data Fields

- TPM_VERSION ver
- UINT32 sealInfoSize
- BYTE * sealInfo
- UINT32 encDataSize
- BYTE * encData

3.107.1 Field Documentation

3.107.1.1 BYTE* encData

3.107.1.2 UINT32 encDataSize

3.107.1.3 BYTE* sealInfo

3.107.1.4 UINT32 sealInfoSize

3.107.1.5 TPM_VERSION ver

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/util/tcg.h

3.108 TPM_VERSION Struct Reference

```
#include <tcg.h>
```

Data Fields

- BYTE major
- BYTE minor
- BYTE revMajor
- BYTE revMinor

3.108.1 Field Documentation

3.108.1.1 BYTE major

3.108.1.2 BYTE minor

3.108.1.3 BYTE revMajor

3.108.1.4 BYTE revMinor

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/util/tcg.h

3.109 tpm_version_1_2_t Struct Reference

Data Fields

- uint16_t tag
- uint8_t Major
- uint8_t Minor
- uint8_t revMajor
- uint8_t revMinor

3.109.1 Field Documentation

3.109.1.1 uint8_t Major

3.109.1.2 uint8_t Minor

3.109.1.3 uint8_t revMajor

3.109.1.4 uint8_t revMinor

3.109.1.5 uint16_t tag

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/tpm_tis.c

3.110 tpm_version_t Struct Reference

Data Fields

- uint8_t Major
- uint8_t Minor
- uint8_t revMajor
- uint8_t revMinor

3.110.1 Field Documentation

3.110.1.1 uint8_t Major

3.110.1.2 uint8_t Minor

3.110.1.3 uint8_t revMajor

3.110.1.4 uint8_t revMinor

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/tpm_tis.c

3.111 tpmback_dev Struct Reference

Data Fields

- tpmif_t ** tpmlist
- unsigned long num_tpms
- unsigned long num_alloc
- struct gntmap map
- int flags
- char ** exclusive_uuids
- xenbus_event_queue events
- void(* open_callback)(domid_t, unsigned int)
- void(* close_callback)(domid_t, unsigned int)
- void(* suspend_callback)(domid_t, unsigned int)
- void(* resume_callback)(domid_t, unsigned int)

3.111.1 Field Documentation

3.111.1.1 void(∗ close_callback)(domid_t, unsigned int)

3.111.1.2 xenbus_event_queue events

3.111.1.3 char∗∗ exclusive_uuids

3.111.1.4 int flags

3.111.1.5 struct gntmap map

3.111.1.6 unsigned long num_alloc

3.111.1.7 unsigned long num_tpms

3.111.1.8 void(∗ open_callback)(domid_t, unsigned int)

3.111.1.9 void(∗ resume_callback)(domid_t, unsigned int)

3.111.1.10 void(∗ suspend_callback)(domid_t, unsigned int)

3.111.1.11 tpmif_t∗∗ tpmlist

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/tpmback.c

3.112 tpmcmd Struct Reference

```
#include <tpmback.h>
```

Data Fields

- domid_t domid
- unsigned int handle
- char ∗ uuid
- unsigned int req_len
- uint8_t ∗ req
- unsigned int resp_len
- uint8_t ∗ resp

3.112.1 Field Documentation

3.112.1.1 domid_t domid

3.112.1.2 unsigned int handle

3.112.1.3 uint8_t∗ req

3.112.1.4 unsigned int req_len

3.112.1.5 uint8_t∗ resp

3.112.1.6 unsigned int resp_len

3.112.1.7 char∗ uuid

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/include/tpmback.h

3.113 tpmfront_dev Struct Reference

```
#include <tpmfront.h>
```

Data Fields

- grant_ref_t ring_ref
- evtchn_port_t evtchn
- tpmif_tx_interface_t ∗ tx
- void ∗∗ pages
- domid_t bedomid
- char ∗ nodename
- char ∗ bepath
- XenbusState state
- uint8_t waiting
- struct wait_queue_head waitq
- uint8_t ∗ respbuf
- size_t resplen

3.113.1 Field Documentation

3.113.1.1 domid_t bedomid

3.113.1.2 char∗ bepath

3.113.1.3 evtchn_port_t evtchn

3.113.1.4 char∗ nodename

3.113.1.5 void∗∗ pages

3.113.1.6 uint8_t∗ respbuf

3.113.1.7 size_t resplen

3.113.1.8 grant_ref_t ring_ref

3.113.1.9 XenbusState state

3.113.1.10 tpmif_tx_interface_t∗ tx

3.113.1.11 uint8_t waiting

3.113.1.12 struct wait_queue_head waitq

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/include/tpmfront.h

3.114 tpmif Struct Reference

Public Types

- enum { DISCONNECTED, DISCONNECTING, CONNECTED }

Data Fields

- domid_t domid
- unsigned int handle
- char * fe_path
- char * fe_state_path
- evtchn_port_t evtchn
- tpmif_tx_interface_t * tx
- void ** pages
- enum xenbus_state state
- enum tpmif:: { ... } status
- char * uuid
- int flags

3.114.1 Member Enumeration Documentation

3.114.1.1 anonymous enum

Enumerator

DISCONNECTED

DISCONNECTING

CONNECTED

3.114.2 Field Documentation

3.114.2.1 domid_t domid

3.114.2.2 evtchn_port_t evtchn

3.114.2.3 char* fe_path

3.114.2.4 char* fe_state_path

3.114.2.5 int flags

3.114.2.6 unsigned int handle

3.114.2.7 void** pages

3.114.2.8 enum xenbus_state state

3.114.2.9 enum { ... } status

3.114.2.10 tpmif_tx_interface_t* tx

3.114.2.11 char* uuid

The documentation for this struct was generated from the following file:

- xen/extras/mini-os/tpmback.c

3.115 transaction Struct Reference

Data Fields

- struct list_head list
- uint32_t id
- unsigned int generation
- TDB_CONTEXT * tdb
- char * tdb_name
- struct list_head changes
- struct list_head changed_domains

3.115.1 Field Documentation

3.115.1.1 struct list_head changed_domains

3.115.1.2 struct list_head changes

3.115.1.3 unsigned int generation

3.115.1.4 uint32_t id

3.115.1.5 struct list_head list

3.115.1.6 TDB_CONTEXT* tdb

3.115.1.7 char* tdb_name

The documentation for this struct was generated from the following file:

- xen/tools/xenstore/xenstored_transaction.c

3.116 vcpu Struct Reference

```
#include <sched.h>
```

Public Member Functions

- XEN_GUEST_HANDLE (vcpu_runstate_info_t) runstate_guest

Data Fields

- int vcpu_id
- int processor
- vcpu_info_t ∗ vcpu_info
- struct domain ∗ domain
- struct vcpu ∗ next_in_list
- s_time_t periodic_period
- s_time_t periodic_last_event
- struct timer periodic_timer
- struct timer singleshot_timer
- struct timer poll_timer
- void ∗ sched_priv
- struct vcpu_runstate_info runstate
- uint64_t last_run_time
- bool_t fpu_initialised
- bool_t fpu_dirtied
- bool_t is_initialised
- bool_t is_running
- bool_t is_urgent
- bool_t defer_shutdown
- bool_t paused_for_shutdown
- int poll_evtchn
- int pirq_evtchn_head
- unsigned long pause_flags
- atomic_t pause_count
- u16 virq_to_evtchn [NR_VIRQS]
- spinlock_t virq_lock
- cpumask_t cpu_affinity
- cpumask_t cpu_affinity_tmp
- cpumask_t vcpu_dirty_cpumask
- struct tasklet continue_hypercall_tasklet
- struct mc_state mc_state
- struct waitqueue_vcpu ∗ waitqueue_vcpu
- struct arch_vcpu arch

3.116.1 Member Function Documentation

3.116.1.1 XEN_GUEST_HANDLE (vcpu_runstate_info_t)

3.116.2 Field Documentation

3.116.2.1 struct arch_vcpu arch

3.116.2.2 struct tasklet continue_hypercall_tasklet

3.116.2.3 cpumask_t cpu_affinity

3.116.2.4 cpumask_t cpu_affinity_tmp

3.116.2.5 bool_t defer_shutdown

3.116.2.6 struct domain∗ domain

3.116.2.7 bool_t fpu_dirtied

3.116.2.8 bool_t fpu_initialised

3.116.2.9 bool_t is_initialised

3.116.2.10 bool_t is_running

3.116.2.11 bool_t is_urgent

3.116.2.12 uint64_t last_run_time

3.116.2.13 struct mc_state mc_state

3.116.2.14 struct vcpu* next_in_list

3.116.2.15 atomic_t pause_count

3.116.2.16 unsigned long pause_flags

3.116.2.17 bool_t paused_for_shutdown

3.116.2.18 s_time_t periodic_last_event

3.116.2.19 s_time_t periodic_period

3.116.2.20 struct timer periodic_timer

3.116.2.21 int pirq_evtchn_head

3.116.2.22 int poll_evtchn

3.116.2.23 struct timer poll_timer

3.116.2.24 int processor

3.116.2.25 struct vcpu_runstate_info runstate

3.116.2.26 void* sched_priv

3.116.2.27 struct timer singleshot_timer

3.116.2.28 cpumask_t vcpu_dirty_cpumask

3.116.2.29 int vcpu_id

3.116.2.30 vcpu_info_t* vcpu_info

3.116.2.31 spinlock_t virq_lock

3.116.2.32 u16 virq_to_evtchn[NR_VIRQS]

3.116.2.33 struct waitqueue_vcpu* waitqueue_vcpu

The documentation for this struct was generated from the following file:

- xen/xen/include/xen/sched.h

3.117 vcpu_guest_context_any_t Union Reference

```
#include <xenctrl.h>
```

Data Fields

- vcpu_guest_context_t c

3.117.1 Field Documentation

3.117.1.1 vcpu_guest_context_t c

The documentation for this union was generated from the following file:

- xen/tools/libxc/xenctrl.h

3.118 vcpu_guest_context_u Union Reference

```
#include <domain.h>
```

Data Fields

- struct vcpu_guest_context * nat
- struct compat_vcpu_guest_context * cmp

3.118.1 Field Documentation

3.118.1.1 struct compat_vcpu_guest_context* cmp

3.118.1.2 struct vcpu_guest_context* nat

The documentation for this union was generated from the following file:

- xen/xen/include/xen/domain.h

3.119 vcpu_info Struct Reference

```
#include <xen.h>
```

Data Fields

- uint8_t evtchn_upcall_pending
- uint8_t evtchn_upcall_mask
- unsigned long evtchn_pending_sel
- struct arch_vcpu_info arch
- struct vcpu_time_info time

3.119.1 Field Documentation

3.119.1.1 struct arch_vcpu_info arch

3.119.1.2 unsigned long evtchn_pending_sel

3.119.1.3 uint8_t evtchn_upcall_mask

3.119.1.4 uint8_t evtchn_upcall_pending

3.119.1.5 struct vcpu_time_info time

The documentation for this struct was generated from the following file:

- xen/xen/include/public/xen.h

3.120 vcpu_time_info Struct Reference

```
#include <xen.h>
```

Data Fields

- uint32_t version
- uint32_t pad0
- uint64_t tsc_timestamp
- uint64_t system_time
- uint32_t tsc_to_system_mul
- int8_t tsc_shift
- int8_t pad1 [3]

3.120.1 Field Documentation

3.120.1.1 uint32_t pad0

3.120.1.2 int8_t pad1[3]

3.120.1.3 uint64_t system_time

3.120.1.4 int8_t tsc_shift

3.120.1.5 uint64_t tsc_timestamp

3.120.1.6 uint32_t tsc_to_system_mul

3.120.1.7 uint32_t version

The documentation for this struct was generated from the following file:

- xen/xen/include/public/xen.h

3.121 vfs_cmd Struct Reference

```
#include <vfs_intf.h>
```

Data Fields

- enum vfs_cmdtype type
- union {
  struct vfs_cmd_init init
  struct vfs_cmd_open open
  struct vfs_cmd_close close
  struct vfs_cmd_stat stat
  struct vfs_cmd_lseek lseek
  struct vfs_cmd_read read
  struct vfs_cmd_write write
  struct vfs_cmd_majorminor majorminor
  } u

3.121.1 Field Documentation

3.121.1.1 struct vfs_cmd_close close

3.121.1.2 struct vfs_cmd_init init

3.121.1.3 struct vfs_cmd_lseek lseek

3.121.1.4 struct vfs_cmd_majorminor majorminor

3.121.1.5 struct vfs_cmd_open open

3.121.1.6 struct vfs_cmd_read read

3.121.1.7 struct vfs_cmd_stat stat

3.121.1.8 enum vfs_cmdtype type

3.121.1.9 union { ... } u

3.121.1.10 struct vfs_cmd_write write

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/integrated/vfs_intf.h

3.122 vfs_cmd_close Struct Reference

```
#include <vfs_intf.h>
```

Data Fields

- int fd
- int ret
- int errn

3.122.1 Field Documentation

3.122.1.1 int errn

3.122.1.2 int fd

3.122.1.3 int ret

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/integrated/vfs_intf.h

3.123 vfs_cmd_init Struct Reference

```
#include <vfs_intf.h>
```

Data Fields

- int ret

3.123.1 Detailed Description

A) no fixed-width definitions are used for types going through the XenStore. They are used, however, for types placed in a shared page buffer.

B) The value of errno will only be used by the frontend if the return field indicates an error.

3.123.2 Field Documentation

3.123.2.1 int ret

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/integrated/vfs_intf.h

3.124 vfs_cmd_lseek Struct Reference

```
#include <vfs_intf.h>
```

Data Fields

- int fd
- int offset
- int whence
- int ret
- int errn

3.124.1 Field Documentation

3.124.1.1 int errn

3.124.1.2 int fd

3.124.1.3 int offset

3.124.1.4 int ret

3.124.1.5 int whence

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/integrated/vfs_intf.h

3.125 vfs_cmd_majorminor Struct Reference

```
#include <vfs_intf.h>
```

Data Fields

- char * path
- int major
- int minor

3.125.1 Field Documentation

3.125.1.1 int major

3.125.1.2 int minor

3.125.1.3 char* path

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/integrated/vfs_intf.h

3.126 vfs_cmd_open Struct Reference

```
#include <vfs_intf.h>
```

Data Fields

- char * path
- int flags
- int mode
- int ret
- int errn

3.126.1 Field Documentation

3.126.1.1 int errn

3.126.1.2 int flags

3.126.1.3 int mode

3.126.1.4 char* path

3.126.1.5 int ret

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/integrated/vfs_intf.h

3.127 vfs_cmd_read Struct Reference

```
#include <vfs_intf.h>
```

Data Fields

- int fd
- int gref
- int nbyte
- int ret
- int errn

3.127.1 Field Documentation

3.127.1.1 int errn

3.127.1.2 int fd

3.127.1.3 int gref

3.127.1.4 int nbyte

3.127.1.5 int ret

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/integrated/vfs_intf.h

3.128 vfs_cmd_stat Struct Reference

```
#include <vfs_intf.h>
```

Data Fields

- char * path
- int gref
- int ret
- int errn

3.128.1 Field Documentation

3.128.1.1 int errn

3.128.1.2 int gref

3.128.1.3 char* path

3.128.1.4 int ret

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/integrated/vfs_intf.h

3.129 vfs_cmd_write Struct Reference

```
#include <vfs_intf.h>
```

Data Fields

- int fd
- int gref
- int nbyte
- int ret
- int errn

3.129.1 Field Documentation

3.129.1.1 int errn

3.129.1.2 int fd

3.129.1.3 int gref

3.129.1.4 int nbyte

3.129.1.5 int ret

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/integrated/vfs_intf.h

3.130 vfs_gref_map Struct Reference

```
#include <vfs_intf.h>
```

Data Fields

- uint32_t num
- uint32_t grefs [GREFS_PER_GREFMAP]

3.130.1 Field Documentation

3.130.1.1 uint32_t grefs[GREFS_PER_GREFMAP]

3.130.1.2 uint32_t num

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/integrated/vfs_intf.h

3.131 vmcb Struct Reference

```
#include <caas.h>
```

Data Fields

- char s_key [128/8]

3.131.1 Detailed Description

VM Control Blob. The vmcb and cspipe are strongly related, and before all functionality including the HMACs is fully implemented, they will look the same.

3.131.2 Field Documentation

3.131.2.1 char s_key[128/8]

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/caas.h

3.132 VTPM_DMI_RESOURCE_T Struct Reference

```
#include <vtpmpriv.h>
```

Data Fields

- vtpm_ipc_handle_t * tx_vtpm_ipc_h
- vtpm_ipc_handle_t * rx_vtpm_ipc_h
- vtpm_ipc_handle_t * tx_tpm_ipc_h
- vtpm_ipc_handle_t * rx_tpm_ipc_h
- pid_t dmi_pid
- bool connected
- UINT32 dmi_domain_id

- TCS_CONTEXT_HANDLE TCSContext
- char * NVMLocation
- UINT32 dmi_id
- BYTE dmi_type
- TPM_DIGEST NVM_measurement
- TPM_DIGEST DMI_measurement

3.132.1 Field Documentation

3.132.1.1 bool connected

3.132.1.2 UINT32 dmi_domain_id

3.132.1.3 UINT32 dmi_id

3.132.1.4 TPM_DIGEST DMI_measurement

3.132.1.5 pid_t dmi_pid

3.132.1.6 BYTE dmi_type

3.132.1.7 TPM_DIGEST NVM_measurement

3.132.1.8 char* NVMLocation

3.132.1.9 vtpm_ipc_handle_t* rx_tpm_ipc_h

3.132.1.10 vtpm_ipc_handle_t* rx_vtpm_ipc_h

3.132.1.11 TCS_CONTEXT_HANDLE TCSContext

3.132.1.12 vtpm_ipc_handle_t* tx_tpm_ipc_h

3.132.1.13 vtpm_ipc_handle_t* tx_vtpm_ipc_h

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/manager/vtpmpriv.h

3.133 vtpm_ipc_handle_t Struct Reference

```
#include <vtpm_ipc.h>
```

Data Fields

- int fh
- int flags
- char * name

3.133.1 Field Documentation

3.133.1.1 int fh

3.133.1.2 int flags

3.133.1.3 char∗ name

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/manager/vtpm_ipc.h

3.134 vtpm_thread_params_s Struct Reference

Data Fields

- vtpm_ipc_handle_t ∗ tx_ipc_h
- vtpm_ipc_handle_t ∗ rx_ipc_h
- BOOL fw_tpm
- vtpm_ipc_handle_t ∗ fw_tx_ipc_h
- vtpm_ipc_handle_t ∗ fw_rx_ipc_h
- BOOL is_priv
- char ∗ thread_name

3.134.1 Field Documentation

3.134.1.1 vtpm_ipc_handle_t∗ fw_rx_ipc_h

3.134.1.2 BOOL fw_tpm

3.134.1.3 vtpm_ipc_handle_t∗ fw_tx_ipc_h

3.134.1.4 BOOL is_priv

3.134.1.5 vtpm_ipc_handle_t∗ rx_ipc_h

3.134.1.6 char∗ thread_name

3.134.1.7 vtpm_ipc_handle_t∗ tx_ipc_h

The documentation for this struct was generated from the following file:

- xen/tools/vtpm_manager/manager/vtpmd.c

3.135 xc_core_header Struct Reference

```
#include <xenctrl.h>
```

Data Fields

- unsigned int xch_magic
- unsigned int xch_nr_vcpus

- unsigned int xch_nr_pages
- unsigned int xch_ctxt_offset
- unsigned int xch_index_offset
- unsigned int xch_pages_offset

3.135.1 Field Documentation

3.135.1.1 unsigned int xch_ctxt_offset

3.135.1.2 unsigned int xch_index_offset

3.135.1.3 unsigned int xch_magic

3.135.1.4 unsigned int xch_nr_pages

3.135.1.5 unsigned int xch_nr_vcpus

3.135.1.6 unsigned int xch_pages_offset

The documentation for this struct was generated from the following file:

- xen/tools/libxc/xenctrl.h

3.136 xc_cpupoolinfo Struct Reference

```
#include <xenctrl.h>
```

Data Fields

- uint32_t cpupool_id
- uint32_t sched_id
- uint32_t n_dom
- xc_cpumap_t cpumap

3.136.1 Field Documentation

3.136.1.1 xc_cpumap_t cpumap

3.136.1.2 uint32_t cpupool_id

3.136.1.3 uint32_t n_dom

3.136.1.4 uint32_t sched_id

The documentation for this struct was generated from the following file:

- xen/tools/libxc/xenctrl.h

3.137 xc_cx_stat Struct Reference

```
#include <xenctrl.h>
```

Data Fields

- uint32_t nr
- uint32_t last
- uint64_t idle_time
- uint64_t * triggers
- uint64_t * residencies
- uint64_t pc3
- uint64_t pc6
- uint64_t pc7
- uint64_t cc3
- uint64_t cc6

3.137.1 Field Documentation

3.137.1.1 uint64_t cc3

3.137.1.2 uint64_t cc6

3.137.1.3 uint64_t idle_time

3.137.1.4 uint32_t last

3.137.1.5 uint32_t nr

3.137.1.6 uint64_t pc3

3.137.1.7 uint64_t pc6

3.137.1.8 uint64_t pc7

3.137.1.9 uint64_t* residencies

3.137.1.10 uint64_t* triggers

The documentation for this struct was generated from the following file:

- xen/tools/libxc/xenctrl.h

3.138 xc_dom_arch Struct Reference

```
#include <xc_dom.h>
```

Data Fields

- int(* alloc_magic_pages)(struct xc_dom_image *dom)
- int(* count_pgtables)(struct xc_dom_image *dom)
- int(* setup_pgtables)(struct xc_dom_image *dom)
- int(* start_info)(struct xc_dom_image *dom)
- int(* shared_info)(struct xc_dom_image *dom, void *shared_info)
- int(* vcpu)(struct xc_dom_image *dom, void *vcpu_ctxt)
- char * guest_type
- char * native_protocol

- int page_shift
- int sizeof_pfn
- struct xc_dom_arch * next

3.138.1 Field Documentation

3.138.1.1 int(* alloc_magic_pages)(struct xc_dom_image *dom)

3.138.1.2 int(* count_pgtables)(struct xc_dom_image *dom)

3.138.1.3 char* guest_type

3.138.1.4 char* native_protocol

3.138.1.5 struct xc_dom_arch* next

3.138.1.6 int page_shift

3.138.1.7 int(* setup_pgtables)(struct xc_dom_image *dom)

3.138.1.8 int(* shared_info)(struct xc_dom_image *dom, void *shared_info)

3.138.1.9 int sizeof_pfn

3.138.1.10 int(* start_info)(struct xc_dom_image *dom)

3.138.1.11 int(* vcpu)(struct xc_dom_image *dom, void *vcpu_ctxt)

The documentation for this struct was generated from the following file:

- xen/tools/libxc/xc_dom.h

3.139 xc_dom_image Struct Reference

```
#include <xc_dom.h>
```

Data Fields

- void * kernel_blob
- size_t kernel_size
- void * ramdisk_blob
- size_t ramdisk_size
- char * cmdline
- uint32_t f_requested [XENFEAT_NR_SUBMAPS]
- struct elf_dom_parms parms
- char * guest_type
- struct xc_dom_seg kernel_seg
- struct xc_dom_seg ramdisk_seg
- struct xc_dom_seg p2m_seg
- struct xc_dom_seg pgtables_seg
- struct xc_dom_seg devicetree_seg
- xen_pfn_t start_info_pfn
- xen_pfn_t console_pfn

- xen_pfn_t xenstore_pfn
- xen_pfn_t shared_info_pfn
- xen_pfn_t bootstack_pfn
- xen_pfn_t caas_pfn
- xen_vaddr_t virt_alloc_end
- xen_vaddr_t bsd_symtab_start
- unsigned int pgtables
- unsigned int pg_l4
- unsigned int pg_l3
- unsigned int pg_l2
- unsigned int pg_l1
- unsigned int alloc_bootstack
- unsigned int extra_pages
- xen_vaddr_t virt_pgtab_end
- uint32_t f_active [XENFEAT_NR_SUBMAPS]
- xen_pfn_t ∗ p2m_host
- void ∗ p2m_guest
- xen_pfn_t total_pages
- struct xc_dom_phys ∗ phys_pages
- int realmodearea_log
- struct xc_dom_mem ∗ memblocks
- size_t alloc_malloc
- size_t alloc_mem_map
- size_t alloc_file_map
- size_t alloc_domU_map
- unsigned long flags
- unsigned int console_evtchn
- unsigned int xenstore_evtchn
- xen_pfn_t shared_info_mfn
- xc_interface ∗ xch
- domid_t guest_domid
- int8_t vhpt_size_log2
- int8_t superpages
- int shadow_enabled
- int xen_version
- xen_capabilities_info_t xen_caps
- struct xc_dom_loader ∗ kernel_loader
- void ∗ private_loader
- struct xc_dom_arch ∗ arch_hooks
- int(∗ allocate)(struct xc_dom_image ∗dom, xen_vaddr_t up_to)

3.139.1 Field Documentation

3.139.1.1 unsigned int alloc_bootstack

3.139.1.2 size_t alloc_domU_map

3.139.1.3 size_t alloc_file_map

3.139.1.4 size_t alloc_malloc

3.139.1.5 size_t alloc_mem_map

3.139.1.6 int(∗ allocate)(struct xc_dom_image ∗dom, xen_vaddr_t up_to)

3.139.1.7 struct xc_dom_arch* arch_hooks

3.139.1.8 xen_pfn_t bootstack_pfn

3.139.1.9 xen_vaddr_t bsd_symtab_start

3.139.1.10 xen_pfn_t caas_pfn

3.139.1.11 char* cmdline

3.139.1.12 unsigned int console_evtchn

3.139.1.13 xen_pfn_t console_pfn

3.139.1.14 struct xc_dom_seg devicetree_seg

3.139.1.15 unsigned int extra_pages

3.139.1.16 uint32_t f_active[XENFEAT_NR_SUBMAPS]

3.139.1.17 uint32_t f_requested[XENFEAT_NR_SUBMAPS]

3.139.1.18 unsigned long flags

3.139.1.19 domid_t guest_domid

3.139.1.20 char* guest_type

3.139.1.21 void* kernel_blob

3.139.1.22 struct xc_dom_loader* kernel_loader

3.139.1.23 struct xc_dom_seg kernel_seg

3.139.1.24 size_t kernel_size

3.139.1.25 struct xc_dom_mem* memblocks

3.139.1.26 void* p2m_guest

3.139.1.27 xen_pfn_t* p2m_host

3.139.1.28 struct xc_dom_seg p2m_seg

3.139.1.29 struct elf_dom_parms parms

3.139.1.30 unsigned int pg_l1

3.139.1.31 unsigned int pg_l2

3.139.1.32 unsigned int pg_l3

3.139.1.33 unsigned int pg_l4

3.139.1.34 unsigned int pgtables

3.139.1.35 struct xc_dom_seg pgtables_seg

3.139.1.36 struct xc_dom_phys* phys_pages

3.139.1.37 void* private_loader

3.139.1.38 void* ramdisk_blob

3.139.1.39 struct xc_dom_seg ramdisk_seg

3.139.1.40 size_t ramdisk_size

3.139.1.41 int realmodearea_log

3.139.1.42 int shadow_enabled

3.139.1.43 xen_pfn_t shared_info_mfn

3.139.1.44 xen_pfn_t shared_info_pfn

3.139.1.45 xen_pfn_t start_info_pfn

3.139.1.46 int8_t superpages

3.139.1.47 xen_pfn_t total_pages

3.139.1.48 int8_t vhpt_size_log2

3.139.1.49 xen_vaddr_t virt_alloc_end

3.139.1.50 xen_vaddr_t virt_pgtab_end

3.139.1.51 xc_interface* xch

3.139.1.52 xen_capabilities_info_t xen_caps

3.139.1.53 int xen_version

3.139.1.54 unsigned int xenstore_evtchn

3.139.1.55 xen_pfn_t xenstore_pfn

The documentation for this struct was generated from the following file:

- xen/tools/libxc/xc_dom.h

3.140 xc_dom_loader Struct Reference

```
#include <xc_dom.h>
```

Data Fields

- char * name
- int(* probe)(struct xc_dom_image *dom)

- int(∗ parser)(struct xc_dom_image ∗dom)
- int(∗ loader)(struct xc_dom_image ∗dom)
- struct xc_dom_loader ∗ next

3.140.1 Field Documentation

3.140.1.1 int(∗ loader)(struct xc_dom_image ∗dom)

3.140.1.2 char∗ name

3.140.1.3 struct xc_dom_loader∗ next

3.140.1.4 int(∗ parser)(struct xc_dom_image ∗dom)

3.140.1.5 int(∗ probe)(struct xc_dom_image ∗dom)

The documentation for this struct was generated from the following file:

- xen/tools/libxc/xc_dom.h

3.141 xc_dom_mem Struct Reference

```
#include <xc_dom.h>
```

Data Fields

- struct xc_dom_mem ∗ next
- void ∗ mmap_ptr
- size_t mmap_len
- unsigned char memory [0]

3.141.1 Field Documentation

3.141.1.1 unsigned char memory[0]

3.141.1.2 size_t mmap_len

3.141.1.3 void∗ mmap_ptr

3.141.1.4 struct xc_dom_mem∗ next

The documentation for this struct was generated from the following file:

- xen/tools/libxc/xc_dom.h

3.142 xc_dom_phys Struct Reference

```
#include <xc_dom.h>
```

Data Fields

- struct xc_dom_phys * next
- void * ptr
- xen_pfn_t first
- xen_pfn_t count

3.142.1 Field Documentation

3.142.1.1 xen_pfn_t count

3.142.1.2 xen_pfn_t first

3.142.1.3 struct xc_dom_phys* next

3.142.1.4 void* ptr

The documentation for this struct was generated from the following file:

- xen/tools/libxc/xc_dom.h

3.143 xc_dom_seg Struct Reference

```
#include <xc_dom.h>
```

Data Fields

- xen_vaddr_t vstart
- xen_vaddr_t vend
- xen_pfn_t pfn

3.143.1 Field Documentation

3.143.1.1 xen_pfn_t pfn

3.143.1.2 xen_vaddr_t vend

3.143.1.3 xen_vaddr_t vstart

The documentation for this struct was generated from the following file:

- xen/tools/libxc/xc_dom.h

3.144 xc_dominfo Struct Reference

```
#include <xenctrl.h>
```

Data Fields

- uint32_t domid
- uint32_t ssidref
- unsigned int dying:1
- unsigned int crashed:1
- unsigned int shutdown:1
- unsigned int paused:1
- unsigned int blocked:1
- unsigned int running:1
- unsigned int hvm:1
- unsigned int debugged:1
- unsigned int shutdown_reason
- unsigned long nr_pages
- unsigned long nr_shared_pages
- unsigned long shared_info_frame
- uint64_t cpu_time
- unsigned long max_memkb
- unsigned int nr_online_vcpus
- unsigned int max_vcpu_id
- xen_domain_handle_t handle
- unsigned int cpupool

3.144.1 Field Documentation

3.144.1.1 unsigned int blocked

3.144.1.2 uint64_t cpu_time

3.144.1.3 unsigned int cpupool

3.144.1.4 unsigned int crashed

3.144.1.5 unsigned int debugged

3.144.1.6 uint32_t domid

3.144.1.7 unsigned int dying

3.144.1.8 xen_domain_handle_t handle

3.144.1.9 unsigned int hvm

3.144.1.10 unsigned long max_memkb

3.144.1.11 unsigned int max_vcpu_id

3.144.1.12 unsigned int nr_online_vcpus

3.144.1.13 unsigned long nr_pages

3.144.1.14 unsigned long nr_shared_pages

3.144.1.15 unsigned int paused

3.144.1.16 unsigned int running

3.144.1.17 unsigned long shared_info_frame

3.144.1.18 unsigned int shutdown

3.144.1.19 unsigned int shutdown_reason

3.144.1.20 uint32_t ssidref

The documentation for this struct was generated from the following file:

- xen/tools/libxc/xenctrl.h

3.145 xc_error Struct Reference

```
#include <xenctrl.h>
```

Data Fields

- enum xc_error_code code
- char message [XC_MAX_ERROR_MSG_LEN]

3.145.1 Field Documentation

3.145.1.1 enum xc_error_code code

3.145.1.2 char message[XC_MAX_ERROR_MSG_LEN]

The documentation for this struct was generated from the following file:

- xen/tools/libxc/xenctrl.h

3.146 xc_get_cpufreq_para Struct Reference

```
#include <xenctrl.h>
```

Data Fields

- uint32_t cpu_num
- uint32_t freq_num
- uint32_t gov_num
- uint32_t * affected_cpus
- uint32_t * scaling_available_frequencies
- char * scaling_available_governors
- char scaling_driver [CPUFREQ_NAME_LEN]
- uint32_t cpuinfo_cur_freq
- uint32_t cpuinfo_max_freq
- uint32_t cpuinfo_min_freq
- uint32_t scaling_cur_freq
- char scaling_governor [CPUFREQ_NAME_LEN]

- uint32_t scaling_max_freq
- uint32_t scaling_min_freq
- union {
  xc_userspace_t userspace
  xc_ondemand_t ondemand
  } u

- int32_t turbo_enabled

3.146.1 Field Documentation

3.146.1.1 uint32_t* affected_cpus

3.146.1.2 uint32_t cpu_num

3.146.1.3 uint32_t cpuinfo_cur_freq

3.146.1.4 uint32_t cpuinfo_max_freq

3.146.1.5 uint32_t cpuinfo_min_freq

3.146.1.6 uint32_t freq_num

3.146.1.7 uint32_t gov_num

3.146.1.8 xc_ondemand_t ondemand

3.146.1.9 uint32_t* scaling_available_frequencies

3.146.1.10 char* scaling_available_governors

3.146.1.11 uint32_t scaling_cur_freq

3.146.1.12 char scaling_driver[CPUFREQ_NAME_LEN]

3.146.1.13 char scaling_governor[CPUFREQ_NAME_LEN]

3.146.1.14 uint32_t scaling_max_freq

3.146.1.15 uint32_t scaling_min_freq

3.146.1.16 int32_t turbo_enabled

3.146.1.17 union { ... } u

3.146.1.18 xc_userspace_t userspace

The documentation for this struct was generated from the following file:

- xen/tools/libxc/xenctrl.h

3.147 xc_hypercall_buffer Struct Reference

```
#include <xenctrl.h>
```

Data Fields

- void * hbuf
- struct xc_hypercall_buffer * param_shadow
- int dir
- void * ubuf
- size_t sz

3.147.1 Field Documentation

3.147.1.1 int dir

3.147.1.2 void* hbuf

3.147.1.3 struct xc_hypercall_buffer* param_shadow

3.147.1.4 size_t sz

3.147.1.5 void* ubuf

The documentation for this struct was generated from the following file:

- xen/tools/libxc/xenctrl.h

3.148 xc_px_stat Struct Reference

```
#include <xenctrl.h>
```

Data Fields

- uint8_t total
- uint8_t usable
- uint8_t last
- uint8_t cur
- uint64_t * trans_pt
- struct xc_px_val * pt

3.148.1 Field Documentation

3.148.1.1 uint8_t cur

3.148.1.2 uint8_t last

3.148.1.3 struct xc_px_val* pt

3.148.1.4 uint8_t total

3.148.1.5 uint64_t* trans_pt

3.148.1.6 uint8_t usable

The documentation for this struct was generated from the following file:

- xen/tools/libxc/xenctrl.h

3.149 xc_px_val Struct Reference

```
#include <xenctrl.h>
```

Data Fields

- uint64_t freq
- uint64_t residency
- uint64_t count

3.149.1 Field Documentation

3.149.1.1 uint64_t count

3.149.1.2 uint64_t freq

3.149.1.3 uint64_t residency

The documentation for this struct was generated from the following file:

- xen/tools/libxc/xenctrl.h

3.150 xen_domctl Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t cmd
- uint32_t interface_version
- domid_t domain
- union {
  struct xen_domctl_createdomain createdomain
  struct xen_domctl_getdomaininfo getdomaininfo
  struct xen_domctl_getmemlist getmemlist
  struct xen_domctl_getpageframeinfo getpageframeinfo
  struct xen_domctl_getpageframeinfo2 getpageframeinfo2
  struct xen_domctl_getpageframeinfo3 getpageframeinfo3
  struct xen_domctl_vcpuaffinity vcpuaffinity
  struct xen_domctl_shadow_op shadow_op
  struct xen_domctl_max_mem max_mem
  struct xen_domctl_vcpucontext vcpucontext
  struct xen_domctl_getvcpuinfo getvcpuinfo
  struct xen_domctl_max_vcpus max_vcpus
  struct xen_domctl_scheduler_op scheduler_op
  struct xen_domctl_setdomainhandle setdomainhandle
  struct xen_domctl_setdebugging setdebugging
  struct xen_domctl_irq_permission irq_permission
  struct xen_domctl_iomem_permission iomem_permission
  struct xen_domctl_ioport_permission ioport_permission
  struct xen_domctl_hypercall_init hypercall_init
  struct xen_domctl_arch_setup arch_setup
  struct xen_domctl_settimeoffset settimeoffset

struct xen_domctl_disable_migrate disable_migrate
struct xen_domctl_tsc_info tsc_info
struct xen_domctl_real_mode_area real_mode_area
struct xen_domctl_hvmcontext hvmcontext
struct
xen_domctl_hvmcontext_partial hvmcontext_partial
struct xen_domctl_address_size address_size
struct xen_domctl_sendtrigger sendtrigger
struct xen_domctl_get_device_group get_device_group
struct xen_domctl_assign_device assign_device
struct xen_domctl_bind_pt_irq bind_pt_irq
struct xen_domctl_memory_mapping memory_mapping
struct xen_domctl_ioport_mapping ioport_mapping
struct xen_domctl_pin_mem_cacheattr pin_mem_cacheattr
struct xen_domctl_ext_vcpucontext ext_vcpucontext
struct xen_domctl_set_opt_feature set_opt_feature
struct xen_domctl_set_target set_target
struct xen_domctl_subscribe subscribe
struct xen_domctl_debug_op debug_op
struct xen_domctl_mem_event_op mem_event_op
struct xen_domctl_mem_sharing_op mem_sharing_op
struct
xen_domctl_set_access_required access_required
struct xen_domctl_gdbsx_memio gdbsx_guest_memio
struct
xen_domctl_gdbsx_pauseunp_vcpu gdbsx_pauseunp_vcpu
struct xen_domctl_gdbsx_domstatus gdbsx_domstatus
uint8_t pad [128]
} u

3.150.1 Field Documentation

3.150.1.1 struct xen_domctl_set_access_required access_required

3.150.1.2 struct xen_domctl_address_size address_size

3.150.1.3 struct xen_domctl_arch_setup arch_setup

3.150.1.4 struct xen_domctl_assign_device assign_device

3.150.1.5 struct xen_domctl_bind_pt_irq bind_pt_irq

3.150.1.6 uint32_t cmd

3.150.1.7 struct xen_domctl_createdomain createdomain

3.150.1.8 struct xen_domctl_debug_op debug_op

3.150.1.9 struct xen_domctl_disable_migrate disable_migrate

3.150.1.10 domid_t domain

3.150.1.11 struct xen_domctl_ext_vcpucontext ext_vcpucontext

3.150.1.12 struct xen_domctl_gdbsx_domstatus gdbsx_domstatus

3.150.1.13 struct **xen_domctl_gdbsx_memio** gdbsx_guest_memio

3.150.1.14 struct **xen_domctl_gdbsx_pauseunp_vcpu** gdbsx_pauseunp_vcpu

3.150.1.15 struct **xen_domctl_get_device_group** get_device_group

3.150.1.16 struct **xen_domctl_getdomaininfo** getdomaininfo

3.150.1.17 struct **xen_domctl_getmemlist** getmemlist

3.150.1.18 struct **xen_domctl_getpageframeinfo** getpageframeinfo

3.150.1.19 struct **xen_domctl_getpageframeinfo2** getpageframeinfo2

3.150.1.20 struct **xen_domctl_getpageframeinfo3** getpageframeinfo3

3.150.1.21 struct **xen_domctl_getvcpuinfo** getvcpuinfo

3.150.1.22 struct **xen_domctl_hvmcontext** hvmcontext

3.150.1.23 struct **xen_domctl_hvmcontext_partial** hvmcontext_partial

3.150.1.24 struct **xen_domctl_hypercall_init** hypercall_init

3.150.1.25 uint32_t interface_version

3.150.1.26 struct **xen_domctl_iomem_permission** iomem_permission

3.150.1.27 struct **xen_domctl_ioport_mapping** ioport_mapping

3.150.1.28 struct **xen_domctl_ioport_permission** ioport_permission

3.150.1.29 struct **xen_domctl_irq_permission** irq_permission

3.150.1.30 struct **xen_domctl_max_mem** max_mem

3.150.1.31 struct **xen_domctl_max_vcpus** max_vcpus

3.150.1.32 struct **xen_domctl_mem_event_op** mem_event_op

3.150.1.33 struct **xen_domctl_mem_sharing_op** mem_sharing_op

3.150.1.34 struct **xen_domctl_memory_mapping** memory_mapping

3.150.1.35 uint8_t pad[128]

3.150.1.36 struct **xen_domctl_pin_mem_cacheattr** pin_mem_cacheattr

3.150.1.37 struct **xen_domctl_real_mode_area** real_mode_area

3.150.1.38 struct **xen_domctl_scheduler_op** scheduler_op

3.150.1.39 struct **xen_domctl_sendtrigger** sendtrigger

3.150.1.40 struct **xen_domctl_set_opt_feature** set_opt_feature

3.150.1.41 struct xen_domctl_set_target set_target

3.150.1.42 struct xen_domctl_setdebugging setdebugging

3.150.1.43 struct xen_domctl_setdomainhandle setdomainhandle

3.150.1.44 struct xen_domctl_settimeoffset settimeoffset

3.150.1.45 struct xen_domctl_shadow_op shadow_op

3.150.1.46 struct xen_domctl_subscribe subscribe

3.150.1.47 struct xen_domctl_tsc_info tsc_info

3.150.1.48 union { ... } u

3.150.1.49 struct xen_domctl_vcpuaffinity vcpuaffinity

3.150.1.50 struct xen_domctl_vcpucontext vcpucontext

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.151 xen_domctl_address_size Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t size

3.151.1 Field Documentation

3.151.1.1 uint32_t size

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.152 xen_domctl_arch_setup Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint64_aligned_t flags

3.152.1 Field Documentation

3.152.1.1 uint64_aligned_t flags

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.153 xen_domctl_assign_device Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t machine_bdf

3.153.1 Field Documentation

3.153.1.1 uint32_t machine_bdf

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.154 xen_domctl_bind_pt_irq Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t machine_irq
- pt_irq_type_t irq_type
- uint32_t hvm_domid
- union {
  - struct {
    - uint8_t isa_irq
  - } isa
  - struct {
    - uint8_t bus
    - uint8_t device
    - uint8_t intx
  - } pci
  - struct {
    - uint8_t gvec
    - uint32_t gflags
    - uint64_aligned_t gtable
  - } msi
  - } u

3.154.1 Field Documentation

3.154.1.1 uint8_t bus

3.154.1.2 uint8_t device

3.154.1.3 uint32_t gflags

3.154.1.4 uint64_aligned_t gtable

3.154.1.5 uint8_t gvec

3.154.1.6 uint32_t hvm_domid

3.154.1.7 uint8_t intx

3.154.1.8 pt_irq_type_t irq_type

3.154.1.9 struct { ... } isa

3.154.1.10 uint8_t isa_irq

3.154.1.11 uint32_t machine_irq

3.154.1.12 struct { ... } msi

3.154.1.13 struct { ... } pci

3.154.1.14 union { ... } u

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.155 xen_domctl_createdomain Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t ssidref
- xen_domain_handle_t handle
- uint32_t flags

3.155.1 Field Documentation

3.155.1.1 uint32_t flags

3.155.1.2 xen_domain_handle_t handle

3.155.1.3 uint32_t ssidref

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.156 xen_domctl_debug_op Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t op
- uint32_t vcpu

3.156.1 Field Documentation

3.156.1.1 uint32_t op

3.156.1.2 uint32_t vcpu

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.157 xen_domctl_disable_migrate Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t disable

3.157.1 Field Documentation

3.157.1.1 uint32_t disable

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.158 xen_domctl_ext_vcpucontext Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t vcpu
- uint32_t size

3.158.1 Field Documentation

3.158.1.1 uint32_t size

3.158.1.2 uint32_t vcpu

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.159 xen_domctl_gdbsx_domstatus Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint8_t paused
- uint32_t vcpu_id
- uint32_t vcpu_ev

3.159.1 Field Documentation

3.159.1.1 uint8_t paused

3.159.1.2 uint32_t vcpu_ev

3.159.1.3 uint32_t vcpu_id

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.160 xen_domctl_gdbsx_memio Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint64_aligned_t pgd3val
- uint64_aligned_t gva
- uint64_aligned_t uva
- uint32_t len
- uint8_t gwr
- uint32_t remain

3.160.1 Field Documentation

3.160.1.1 uint64_aligned_t gva

3.160.1.2 uint8_t gwr

3.160.1.3 uint32_t len

3.160.1.4 uint64_aligned_t pgd3val

3.160.1.5 uint32_t remain

3.160.1.6 uint64_aligned_t uva

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.161 xen_domctl_gdbsx_pauseunp_vcpu Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t vcpu

3.161.1 Field Documentation

3.161.1.1 uint32_t vcpu

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.162 xen_domctl_get_device_group Struct Reference

```
#include <domctl.h>
```

Public Member Functions

- XEN_GUEST_HANDLE_64 (uint32) sdev_array

Data Fields

- uint32_t machine_bdf
- uint32_t max_sdevs
- uint32_t num_sdevs

3.162.1 Member Function Documentation

3.162.1.1 XEN_GUEST_HANDLE_64 (uint32)

3.162.2 Field Documentation

3.162.2.1 uint32_t machine_bdf

3.162.2.2 uint32_t max_sdevs

3.162.2.3 uint32_t num_sdevs

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.163 xen_domctl_getdomaininfo Struct Reference

```
#include <domctl.h>
```

Data Fields

- domid_t domain
- uint32_t flags
- uint64_aligned_t tot_pages
- uint64_aligned_t max_pages
- uint64_aligned_t shr_pages
- uint64_aligned_t shared_info_frame
- uint64_aligned_t cpu_time
- uint32_t nr_online_vcpus
- uint32_t max_vcpu_id
- uint32_t ssidref
- xen_domain_handle_t handle
- uint32_t cpupool

3.163.1 Field Documentation

3.163.1.1 uint64_aligned_t cpu_time

3.163.1.2 uint32_t cpupool

3.163.1.3 domid_t domain

3.163.1.4 uint32_t flags

3.163.1.5 xen_domain_handle_t handle

3.163.1.6 uint64_aligned_t max_pages

3.163.1.7 uint32_t max_vcpu_id

3.163.1.8 uint32_t nr_online_vcpus

3.163.1.9 uint64_aligned_t shared_info_frame

3.163.1.10 uint64_aligned_t shr_pages

3.163.1.11 uint32_t ssidref

3.163.1.12 uint64_aligned_t tot_pages

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.164 xen_domctl_getmemlist Struct Reference

```
#include <domctl.h>
```

Public Member Functions

- XEN_GUEST_HANDLE_64 (uint64) buffer

Data Fields

- uint64_aligned_t max_pfns
- uint64_aligned_t start_pfn
- uint64_aligned_t num_pfns

3.164.1 Member Function Documentation

3.164.1.1 XEN_GUEST_HANDLE_64 (uint64)

3.164.2 Field Documentation

3.164.2.1 uint64_aligned_t max_pfns

3.164.2.2 uint64_aligned_t num_pfns

3.164.2.3 uint64_aligned_t start_pfn

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.165 xen_domctl_getpageframeinfo Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint64_aligned_t gmfn
- uint32_t type

3.165.1 Field Documentation

3.165.1.1 uint64_aligned_t gmfn

3.165.1.2 uint32_t type

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.166 xen_domctl_getpageframeinfo2 Struct Reference

```
#include <domctl.h>
```

Public Member Functions

- XEN_GUEST_HANDLE_64 (uint32) array

Data Fields

- uint64_aligned_t num

3.166.1 Member Function Documentation

3.166.1.1 XEN_GUEST_HANDLE_64 (uint32)

3.166.2 Field Documentation

3.166.2.1 uint64_aligned_t num

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.167 xen_domctl_getpageframeinfo3 Struct Reference

```
#include <domctl.h>
```

Public Member Functions

- XEN_GUEST_HANDLE_64 (xen_pfn_t) array

Data Fields

- uint64_aligned_t num

3.167.1 Member Function Documentation

3.167.1.1 XEN_GUEST_HANDLE_64 (xen_pfn_t)

3.167.2 Field Documentation

3.167.2.1 uint64_aligned_t num

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.168 xen_domctl_getvcpuinfo Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t vcpu
- uint8_t online
- uint8_t blocked
- uint8_t running
- uint64_aligned_t cpu_time
- uint32_t cpu

3.168.1 Field Documentation

3.168.1.1 uint8_t blocked

3.168.1.2 uint32_t cpu

3.168.1.3 uint64_aligned_t cpu_time

3.168.1.4 uint8_t online

3.168.1.5 uint8_t running

3.168.1.6 uint32_t vcpu

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.169 xen_domctl_hvmcontext Struct Reference

```
#include <domctl.h>
```

Public Member Functions

- XEN_GUEST_HANDLE_64 (uint8) buffer

Data Fields

- uint32_t size

3.169.1 Member Function Documentation

3.169.1.1 XEN_GUEST_HANDLE_64 (uint8)

3.169.2 Field Documentation

3.169.2.1 uint32_t size

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.170 xen_domctl_hvmcontext_partial Struct Reference

```
#include <domctl.h>
```

Public Member Functions

- XEN_GUEST_HANDLE_64 (uint8) buffer

Data Fields

- uint32_t type
- uint32_t instance

3.170.1 Member Function Documentation

3.170.1.1 XEN_GUEST_HANDLE_64 (uint8)

3.170.2 Field Documentation

3.170.2.1 uint32_t instance

3.170.2.2 uint32_t type

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.171 xen_domctl_hypercall_init Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint64_aligned_t gmfn

3.171.1 Field Documentation

3.171.1.1 uint64_aligned_t gmfn

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.172 xen_domctl_iomem_permission Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint64_aligned_t first_mfn
- uint64_aligned_t nr_mfns
- uint8_t allow_access

3.172.1 Field Documentation

3.172.1.1 uint8_t allow_access

3.172.1.2 uint64_aligned_t first_mfn

3.172.1.3 uint64_aligned_t nr_mfns

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.173 xen_domctl_ioport_mapping Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t first_gport
- uint32_t first_mport
- uint32_t nr_ports
- uint32_t add_mapping

3.173.1 Field Documentation

3.173.1.1 uint32_t add_mapping

3.173.1.2 uint32_t first_gport

3.173.1.3 uint32_t first_mport

3.173.1.4 uint32_t nr_ports

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.174 xen_domctl_ioport_permission Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t first_port
- uint32_t nr_ports
- uint8_t allow_access

3.174.1 Field Documentation

3.174.1.1 uint8_t allow_access

3.174.1.2 uint32_t first_port

3.174.1.3 uint32_t nr_ports

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.175 xen_domctl_irq_permission Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint8_t pirq
- uint8_t allow_access

3.175.1 Field Documentation

3.175.1.1 uint8_t allow_access

3.175.1.2 uint8_t pirq

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.176 xen_domctl_max_mem Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint64_aligned_t max_memkb

3.176.1 Field Documentation

3.176.1.1 uint64_aligned_t max_memkb

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.177 xen_domctl_max_vcpus Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t max

3.177.1 Field Documentation

3.177.1.1 uint32_t max

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.178 xen_domctl_mem_event_op Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t op
- uint32_t mode
- uint64_aligned_t shared_addr
- uint64_aligned_t ring_addr
- uint64_aligned_t gfn

3.178.1 Field Documentation

3.178.1.1 uint64_aligned_t gfn

3.178.1.2 uint32_t mode

3.178.1.3 uint32_t op

3.178.1.4 uint64_aligned_t ring_addr

3.178.1.5 uint64_aligned_t shared_addr

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.179 xen_domctl_mem_sharing_op Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint8_t op

- union {
  uint8_t enable
  struct **mem_sharing_op_nominate** {
    union {
      uint64_aligned_t gfn
      uint32_t grant_ref
    } u
    uint64_aligned_t handle
  } nominate
  struct **mem_sharing_op_share** {
    uint64_aligned_t source_handle
    uint64_aligned_t client_handle
  } share
  struct **mem_sharing_op_debug** {
    union {
      uint64_aligned_t gfn
      uint64_aligned_t mfn
      grant_ref_t gref
    } u
  } debug
  } u

3.179.1 Field Documentation

3.179.1.1 uint64_aligned_t client_handle

3.179.1.2 struct { ... } ::mem_sharing_op_debug debug

3.179.1.3 uint8_t enable

3.179.1.4 uint64_aligned_t gfn

3.179.1.5 uint32_t grant_ref

3.179.1.6 grant_ref_t gref

3.179.1.7 uint64_aligned_t handle

3.179.1.8 uint64_aligned_t mfn

3.179.1.9 struct { ... } ::mem_sharing_op_nominate nominate

3.179.1.10 uint8_t op

3.179.1.11 struct { ... } ::mem_sharing_op_share share

3.179.1.12 uint64_aligned_t source_handle

3.179.1.13 union { ... } u

3.179.1.14 union { ... } u

3.179.1.15 union { ... } u

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.180 xen_domctl_memory_mapping Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint64_aligned_t first_gfn
- uint64_aligned_t first_mfn
- uint64_aligned_t nr_mfns
- uint32_t add_mapping
- uint32_t padding

3.180.1 Field Documentation

3.180.1.1 uint32_t add_mapping

3.180.1.2 uint64_aligned_t first_gfn

3.180.1.3 uint64_aligned_t first_mfn

3.180.1.4 uint64_aligned_t nr_mfns

3.180.1.5 uint32_t padding

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.181 xen_domctl_pin_mem_cacheattr Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint64_aligned_t start
- uint64_aligned_t end
- uint32_t type

3.181.1 Field Documentation

3.181.1.1 uint64_aligned_t end

3.181.1.2 uint64_aligned_t start

3.181.1.3 uint32_t type

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.182 xen_domctl_real_mode_area Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t log

3.182.1 Field Documentation

3.182.1.1 uint32_t log

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.183 xen_domctl_scheduler_op Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t sched_id
- uint32_t cmd
- union {
  - struct **xen_domctl_sched_sedf** {
    - uint64_aligned_t period
    - uint64_aligned_t slice
    - uint64_aligned_t latency
    - uint32_t extratime
    - uint32_t weight
  - } sedf
  - struct **xen_domctl_sched_credit** {
    - uint16_t weight
    - uint16_t cap
  - } credit
  - struct **xen_domctl_sched_credit2** {
    - uint16_t weight
  - } credit2
- } u

3.183.1 Field Documentation

3.183.1.1 uint16_t cap

3.183.1.2 uint32_t cmd

3.183.1.3 struct { ... } ::xen_domctl_sched_credit credit

3.183.1.4 struct { ... } ::xen_domctl_sched_credit2 credit2

3.183.1.5 uint32_t extratime

3.183.1.6 uint64_aligned_t latency

3.183.1.7 uint64_aligned_t period

3.183.1.8 uint32_t sched_id

3.183.1.9 struct { ... } ::xen_domctl_sched_sedf sedf

3.183.1.10 uint64_aligned_t slice

3.183.1.11 union { ... } u

3.183.1.12 uint32_t weight

3.183.1.13 uint16_t weight

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.184 xen_domctl_sendtrigger Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t trigger
- uint32_t vcpu

3.184.1 Field Documentation

3.184.1.1 uint32_t trigger

3.184.1.2 uint32_t vcpu

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.185 xen_domctl_set_access_required Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint8_t access_required

3.185.1 Field Documentation

3.185.1.1 uint8_t access_required

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.186 xen_domctl_set_opt_feature Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint64_t dummy

3.186.1 Field Documentation

3.186.1.1 uint64_t dummy

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.187 xen_domctl_set_target Struct Reference

```
#include <domctl.h>
```

Data Fields

- domid_t target

3.187.1 Field Documentation

3.187.1.1 domid_t target

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.188 xen_domctl_setdebugging Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint8_t enable

3.188.1 Field Documentation

3.188.1.1 uint8_t enable

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.189 xen_domctl_setdomainhandle Struct Reference

```
#include <domctl.h>
```

Data Fields

- xen_domain_handle_t handle

3.189.1 Field Documentation

3.189.1.1 xen_domain_handle_t handle

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.190 xen_domctl_settimeoffset Struct Reference

```
#include <domctl.h>
```

Data Fields

- int32_t time_offset_seconds

3.190.1 Field Documentation

3.190.1.1 int32_t time_offset_seconds

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.191 xen_domctl_shadow_op Struct Reference

```
#include <domctl.h>
```

Public Member Functions

- XEN_GUEST_HANDLE_64 (uint8) dirty_bitmap

Data Fields

- uint32_t op
- uint32_t mode
- uint32_t mb
- uint64_aligned_t pages
- struct xen_domctl_shadow_op_stats stats

3.191.1 Member Function Documentation

3.191.1.1 XEN_GUEST_HANDLE_64 (uint8)

3.191.2 Field Documentation

3.191.2.1 uint32_t mb

3.191.2.2 uint32_t mode

3.191.2.3 uint32_t op

3.191.2.4 uint64_aligned_t pages

3.191.2.5 struct xen_domctl_shadow_op_stats stats

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.192 xen_domctl_shadow_op_stats Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t fault_count
- uint32_t dirty_count

3.192.1 Field Documentation

3.192.1.1 uint32_t dirty_count

3.192.1.2 uint32_t fault_count

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.193 xen_domctl_subscribe Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t port

3.193.1 Field Documentation

3.193.1.1 uint32_t port

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.194 xen_domctl_tsc_info Struct Reference

```
#include <domctl.h>
```

Public Member Functions

- XEN_GUEST_HANDLE_64 (xen_guest_tsc_info_t) out_info

Data Fields

- xen_guest_tsc_info_t info

3.194.1 Member Function Documentation

3.194.1.1 XEN_GUEST_HANDLE_64 (xen_guest_tsc_info_t)

3.194.2 Field Documentation

3.194.2.1 xen_guest_tsc_info_t info

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.195 xen_domctl_vcpuaffinity Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t vcpu
- struct xenctl_cpumap cpumap

3.195.1 Field Documentation

3.195.1.1 struct xenctl_cpumap cpumap

3.195.1.2 uint32_t vcpu

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.196 xen_domctl_vcpucontext Struct Reference

```
#include <domctl.h>
```

Public Member Functions

- XEN_GUEST_HANDLE_64 (vcpu_guest_context_t) ctxt

Data Fields

- uint32_t vcpu

3.196.1 Member Function Documentation

3.196.1.1 XEN_GUEST_HANDLE_64 (vcpu_guest_context_t)

3.196.2 Field Documentation

3.196.2.1 uint32_t vcpu

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.197 xen_guest_tsc_info Struct Reference

```
#include <domctl.h>
```

Data Fields

- uint32_t tsc_mode
- uint32_t gtsc_khz
- uint32_t incarnation
- uint32_t pad
- uint64_aligned_t elapsed_nsec

3.197.1 Field Documentation

3.197.1.1 uint64_aligned_t elapsed_nsec

3.197.1.2 uint32_t gtsc_khz

3.197.1.3 uint32_t incarnation

3.197.1.4 uint32_t pad

3.197.1.5 uint32_t tsc_mode

The documentation for this struct was generated from the following file:

- xen/xen/include/public/domctl.h

3.198 xen_multiboot_mod_list Struct Reference

```
#include <xen.h>
```

Data Fields

- uint32_t mod_start
- uint32_t mod_end
- uint32_t cmdline
- uint32_t pad

3.198.1 Field Documentation

3.198.1.1 uint32_t cmdline

3.198.1.2 uint32_t mod_end

3.198.1.3 uint32_t mod_start

3.198.1.4 uint32_t pad

The documentation for this struct was generated from the following file:

- xen/xen/include/public/xen.h

3.199 xlc_cmd Struct Reference

```
#include <xlc_intf.h>
```

Data Fields

- enum xlc_cmdtype type
- union {
  struct xlc_cmd_ping ping
  struct xlc_cmd_getpage getpage

```
    struct xlc_cmd_retpage retpage
    struct xlc_cmd_dommake dommake
    struct xlc_cmd_dombuild dombuild
    struct xlc_cmd_domcreate domcreate
  } u
```

3.199.1 Field Documentation

3.199.1.1 struct xlc_cmd_dombuild dombuild

3.199.1.2 struct xlc_cmd_domcreate domcreate

3.199.1.3 struct xlc_cmd_dommake dommake

3.199.1.4 struct xlc_cmd_getpage getpage

3.199.1.5 struct xlc_cmd_ping ping

3.199.1.6 struct xlc_cmd_retpage retpage

3.199.1.7 enum xlc_cmdtype type

3.199.1.8 union { ... } u

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/xlc_intf.h

3.200 xlc_cmd_dombuild Struct Reference

```
#include <xlc_intf.h>
```

Data Fields

- uint32_t in_gref
- uint32_t out_gref
- uint32_t domid
- int ret

3.200.1 Field Documentation

3.200.1.1 uint32_t domid

3.200.1.2 uint32_t in_gref

3.200.1.3 uint32_t out_gref

3.200.1.4 int ret

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/xlc_intf.h

3.201 xlc_cmd_domcreate Struct Reference

```
#include <xlc_intf.h>
```

Data Fields

- uint32_t in_gref
- uint32_t out_gref
- int restore_fd
- int ret
- uint32_t domid
- uint32_t domid_dc

3.201.1 Field Documentation

3.201.1.1 uint32_t domid

3.201.1.2 uint32_t domid_dc

3.201.1.3 uint32_t in_gref

3.201.1.4 uint32_t out_gref

3.201.1.5 int restore_fd

3.201.1.6 int ret

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/xlc_intf.h

3.202 xlc_cmd_dommake Struct Reference

```
#include <xlc_intf.h>
```

Data Fields

- uint32_t in_gref
- uint32_t out_gref
- int ret
- uint32_t domid

3.202.1 Field Documentation

3.202.1.1 uint32_t domid

3.202.1.2 uint32_t in_gref

3.202.1.3 uint32_t out_gref

3.202.1.4 int ret

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/xlc_intf.h

3.203 xlc_cmd_getpage Struct Reference

```
#include <xlc_intf.h>
```

Data Fields

- uint32_t in_gref
- uint32_t out_gref

3.203.1 Detailed Description

Open a page used for sending commands

3.203.2 Field Documentation

3.203.2.1 uint32_t in_gref

3.203.2.2 uint32_t out_gref

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/xlc_intf.h

3.204 xlc_cmd_ping Struct Reference

```
#include <xlc_intf.h>
```

Data Fields

- int ret

3.204.1 Detailed Description

Test if a DomT is running

3.204.2 Field Documentation

3.204.2.1 int ret

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/xlc_intf.h

3.205 xlc_cmd_retpage Struct Reference

```
#include <xlc_intf.h>
```

Data Fields

- uint32_t out_gref

3.205.1 Field Documentation

3.205.1.1 uint32_t out_gref

The documentation for this struct was generated from the following file:

- xen/stubdom/domt/xlc_intf.h

3.206 XlObject Struct Reference

Data Fields

- PyObject_HEAD
- libxl_ctx ctx
- xentoollog_logger_stdiostream * logger
- xentoollog_level minmsglevel

3.206.1 Field Documentation

3.206.1.1 libxl_ctx ctx

3.206.1.2 xentoollog_logger_stdiostream* logger

3.206.1.3 xentoollog_level minmsglevel

3.206.1.4 PyObject_HEAD

The documentation for this struct was generated from the following file:

- xen/tools/python/xen/lowlevel/xl/xl.c

3.207 yyalloc Union Reference

Data Fields

- yytype_int16 yyss_alloc
- YYSTYPE yyvs_alloc
- YYLTYPE yyls_alloc

3.207.1 Field Documentation

3.207.1.1 YYLTYPE yyls_alloc

3.207.1.2 yytype_int16 yyss_alloc

3.207.1.3 YYSTYPE yyvs_alloc

The documentation for this union was generated from the following file:

- xen/tools/libxl/libxlu_cfg_y.c

3.208 YYLTYPE Struct Reference

```
#include <libxlu_cfg_y.h>
```

Data Fields

- int first_line
- int first_column
- int last_line
- int last_column

3.208.1 Field Documentation

3.208.1.1 int first_column

3.208.1.2 int first_line

3.208.1.3 int last_column

3.208.1.4 int last_line

The documentation for this struct was generated from the following files:

- xen/tools/libxl/libxlu_cfg_y.c
- xen/tools/libxl/libxlu_cfg_y.h

3.209 YYSTYPE Union Reference

```
#include <libxlu_cfg_y.h>
```

Data Fields

- char * string
- XLU_ConfigSetting * setting

3.209.1 Field Documentation

3.209.1.1 XLU_ConfigSetting * setting

3.209.1.2 char * string

The documentation for this union was generated from the following files:

- xen/tools/libxl/libxlu_cfg_y.c
- xen/tools/libxl/libxlu_cfg_y.h

Chapter 4

File Documentation

4.1 xen/extras/mini-os/arch/ia64/iorw.c File Reference

```
#include <mini-os/iorw.h>
#include <mini-os/console.h>
```

Functions

- void iowrite8 (volatile void ∗addr, uint8_t val)
- void iowrite32 (volatile void ∗addr, uint32_t val)
- uint8_t ioread8 (volatile void ∗addr)
- uint32_t ioread32 (volatile void ∗addr)

4.1.1 Function Documentation

4.1.1.1 uint32_t ioread32 (volatile void ∗ *addr*)

4.1.1.2 uint8_t ioread8 (volatile void ∗ *addr*)

4.1.1.3 void iowrite32 (volatile void ∗ *addr,* uint32_t *val*)

4.1.1.4 void iowrite8 (volatile void ∗ *addr,* uint8_t *val*)

4.2 xen/extras/mini-os/arch/x86/iorw.c File Reference

```
#include <mini-os/iorw.h>
```

Functions

- void iowrite8 (volatile void ∗addr, uint8_t val)
- void iowrite32 (volatile void ∗addr, uint32_t val)
- uint8_t ioread8 (volatile void ∗addr)
- uint32_t ioread32 (volatile void ∗addr)

4.2.1 Function Documentation

4.2.1.1 uint32_t ioread32 (volatile void * *addr*)

4.2.1.2 uint8_t ioread8 (volatile void * *addr*)

4.2.1.3 void iowrite32 (volatile void * *addr,* uint32_t *val*)

4.2.1.4 void iowrite8 (volatile void * *addr,* uint8_t *val*)

4.3 xen/extras/mini-os/arch/x86/mm.c File Reference

```
#include <mini-os/os.h>
#include <mini-os/hypervisor.h>
#include <mini-os/mm.h>
#include <mini-os/types.h>
#include <mini-os/lib.h>
#include <mini-os/xmalloc.h>
#include <xen/memory.h>
```

Macros

- #define DEBUG(_f, _a...) ((void)0)
- #define MEM_TEST_MAX_ERRORS 10
- #define DEMAND_MAP_PAGES ((2ULL << 30) / PAGE_SIZE)
- #define HEAP_PAGES 0
- #define MAP_BATCH ((STACK_SIZE / 2) / sizeof(mmu_update_t))
- #define UNMAP_BATCH ((STACK_SIZE / 2) / sizeof(multicall_entry_t))
- #define MAX_CONTIG_ORDER 9 /* 2MB */
- #define L1_P2M_SHIFT 10
- #define L2_P2M_SHIFT 20
- #define L3_P2M_SHIFT 30
- #define L1_P2M_ENTRIES (1 << L1_P2M_SHIFT)
- #define L2_P2M_ENTRIES (1 << (L2_P2M_SHIFT - L1_P2M_SHIFT))
- #define L3_P2M_ENTRIES (1 << (L3_P2M_SHIFT - L2_P2M_SHIFT))
- #define L1_P2M_MASK (L1_P2M_ENTRIES - 1)
- #define L2_P2M_MASK (L2_P2M_ENTRIES - 1)
- #define L3_P2M_MASK (L3_P2M_ENTRIES - 1)

Functions

- void page_walk (unsigned long va)
- int mem_test (unsigned long *start_va, unsigned long *end_va, int verbose)
- pgentry_t * need_pgt (unsigned long va)
- void arch_init_demand_mapping_area (unsigned long cur_pfn)
- unsigned long allocate_ondemand (unsigned long n, unsigned long alignment)
- void do_map_frames (unsigned long va, const unsigned long *mfns, unsigned long n, unsigned long stride, unsigned long incr, domid_t id, int *err, unsigned long prot)
- void * map_frames_ex (const unsigned long *mfns, unsigned long n, unsigned long stride, unsigned long incr, unsigned long alignment, domid_t id, int *err, unsigned long prot)
- int unmap_frames (unsigned long va, unsigned long num_frames)
- unsigned long alloc_contig_pages (int order, unsigned int addr_bits)
- int mfn_is_ram (unsigned long mfn)
- void arch_init_p2m (unsigned long max_pfn)
- void arch_init_mm (unsigned long *start_pfn_p, unsigned long *max_pfn_p)

Variables

- unsigned long * phys_to_machine_mapping
- unsigned long mfn_zero
- char stack []
- struct shared_info shared_info

4.3.1 Macro Definition Documentation

4.3.1.1 #define DEBUG(_f, _a...) ((void)0)

4.3.1.2 #define DEMAND_MAP_PAGES ((2ULL << 30) / PAGE_SIZE)

4.3.1.3 #define HEAP_PAGES 0

4.3.1.4 #define L1_P2M_ENTRIES (1 << L1_P2M_SHIFT)

4.3.1.5 #define L1_P2M_MASK (L1_P2M_ENTRIES - 1)

4.3.1.6 #define L1_P2M_SHIFT 10

4.3.1.7 #define L2_P2M_ENTRIES (1 << (L2_P2M_SHIFT - L1_P2M_SHIFT))

4.3.1.8 #define L2_P2M_MASK (L2_P2M_ENTRIES - 1)

4.3.1.9 #define L2_P2M_SHIFT 20

4.3.1.10 #define L3_P2M_ENTRIES (1 << (L3_P2M_SHIFT - L2_P2M_SHIFT))

4.3.1.11 #define L3_P2M_MASK (L3_P2M_ENTRIES - 1)

4.3.1.12 #define L3_P2M_SHIFT 30

4.3.1.13 #define MAP_BATCH ((STACK_SIZE / 2) / sizeof(mmu_update_t))

Map an array of MFNs contiguously into virtual address space starting at va. map f[i*stride]+i*increment for i in 0..n-1.

4.3.1.14 #define MAX_CONTIG_ORDER 9 /* 2MB */

Allocate pages which are contiguous in machine memory. Returns a VA to where they are mapped or 0 on failure.

addr_bits indicates if the region has restrictions on where it is located. Typical values are 32 (if for example PCI devices can't access 64bit memory) or 0 for no restrictions.

Allocated pages can be freed using the page allocators free_pages() function.

based on Linux function xen_create_contiguous_region()

4.3.1.15 #define MEM_TEST_MAX_ERRORS 10

A useful mem testing function. Write the address to every address in the range provided and read back the value. If verbose, print page walk to some VA

If we get MEM_TEST_MAX_ERRORS we might as well stop

4.3.1.16 #define UNMAP_BATCH ((STACK_SIZE / 2) / sizeof(multicall_entry_t))

Unmap nun_frames frames mapped at virtual address va.

4.3.2 Function Documentation

4.3.2.1 unsigned long alloc_contig_pages (int *order,* unsigned int *addr_bits*)

4.3.2.2 unsigned long allocate_ondemand (unsigned long *n,* unsigned long *alignment*)

4.3.2.3 void arch_init_demand_mapping_area (unsigned long *cur_pfn*)

4.3.2.4 void arch_init_mm (unsigned long * *start_pfn_p,* unsigned long * *max_pfn_p*)

4.3.2.5 void arch_init_p2m (unsigned long *max_pfn*)

4.3.2.6 void do_map_frames (unsigned long *va,* const unsigned long * *mfns,* unsigned long *n,* unsigned long *stride,* unsigned long *incr,* domid_t *id,* int * *err,* unsigned long *prot*)

4.3.2.7 void* map_frames_ex (const unsigned long * *mfns,* unsigned long *n,* unsigned long *stride,* unsigned long *incr,* unsigned long *alignment,* domid_t *id,* int * *err,* unsigned long *prot*)

Map an array of MFNs contiguous into virtual address space. Virtual addresses are allocated from the on demand area.

4.3.2.8 int mem_test (unsigned long * *start_va,* unsigned long * *end_va,* int *verbose*)

4.3.2.9 int mfn_is_ram (unsigned long *mfn*)

4.3.2.10 pgentry_t* need_pgt (unsigned long *va*)

return a valid PTE for a given virtual address. If PTE does not exist, allocate page-table pages.

4.3.2.11 void page_walk (unsigned long *va*)

4.3.2.12 int unmap_frames (unsigned long *va,* unsigned long *num_frames*)

4.3.3 Variable Documentation

4.3.3.1 unsigned long mfn_zero

4.3.3.2 unsigned long* phys_to_machine_mapping

4.3.3.3 struct shared_info shared_info

Mark portion of the address space read only.

4.3.3.4 char stack[]

4.4 xen/xen/arch/x86/mm.c File Reference

```
#include <xen/config.h>
#include <xen/init.h>
#include <xen/kernel.h>
#include <xen/lib.h>
#include <xen/mm.h>
#include <xen/domain.h>
#include <xen/sched.h>
#include <xen/errno.h>
#include <xen/perfc.h>
#include <xen/irq.h>
#include <xen/softirq.h>
#include <xen/domain_page.h>
#include <xen/event.h>
#include <xen/iocap.h>
#include <xen/guest_access.h>
#include <asm/paging.h>
#include <asm/shadow.h>
#include <asm/page.h>
#include <asm/flushtlb.h>
#include <asm/io.h>
#include <asm/ldt.h>
#include <asm/x86_emulate.h>
#include <asm/e820.h>
#include <asm/hypercall.h>
#include <asm/shared.h>
#include <public/memory.h>
#include <public/sched.h>
#include <xsm/xsm.h>
#include <xen/trace.h>
#include <asm/setup.h>
#include <asm/fixmap.h>
#include <asm/mem_sharing.h>
```

Data Structures

- struct memory_map_context
- struct ptwr_emulate_ctxt

Macros

- #define MEM_LOG(_f, _a...) gdprintk(XENLOG_WARNING , _f "\n" , ## _a)
- #define PTE_UPDATE_WITH_CMPXCHG
- #define PAGE_CACHE_ATTRS (_PAGE_PAT|_PAGE_PCD|_PAGE_PWT)
- #define l1_disallow_mask(d)
- #define l3_disallow_mask(d) L3_DISALLOW_MASK
- #define define_get_linear_pagetable(level)
- #define adjust_guest_l1e(_p, _d) ((void)(_d))
- #define adjust_guest_l2e(_p, _d) ((void)(_d))
- #define adjust_guest_l3e(_p, _d) ((void)(_d))
- #define unadjust_guest_l3e(_p, _d) ((void)(_d))
- #define pae_flush_pgd(mfn, idx, nl3e) ((void)0)
- #define alloc_l4_table(page, preemptible) (-EINVAL)

- #define free_l4_table(page, preemptible) (-EINVAL)
- #define UPDATE_ENTRY(_t, _p, _o, _n, _m, _v, _ad)
- #define fixmap_domain_page(mfn) mfn_to_virt(mfn)
- #define fixunmap_domain_page(ptr) ((void)(ptr))
- #define l1f_to_lNf(f) (((f) & _PAGE_PRESENT) ? ((f) | _PAGE_PSE) : (f))
- #define lNf_to_l1f(f) (((f) & _PAGE_PRESENT) ? ((f) & ~_PAGE_PSE) : (f))
- #define flush_area(v, f)

Typedefs

- typedef struct e820entry e820entry_t

Functions

- l1_pgentry_t __attribute__ ((__section__(".bss.page_aligned")))
- boolean_param ("allowsuperpage", opt_allow_superpage)
- void __init init_frametable (void)
- void __init arch_init_memory (void)
- int page_is_ram_type (unsigned long mfn, unsigned long mem_type)
- unsigned long domain_get_maximum_gpfn (struct domain *d)
- void share_xen_page_with_guest (struct page_info *page, struct domain *d, int readonly)
- void share_xen_page_with_privileged_guests (struct page_info *page, int readonly)
- void make_cr3 (struct vcpu *v, unsigned long mfn)
- void write_ptbase (struct vcpu *v)
- void update_cr3 (struct vcpu *v)
- int map_ldt_shadow_page (unsigned int off)
- int is_iomem_page (unsigned long mfn)
- int get_page_from_l1e (l1_pgentry_t l1e, struct domain *l1e_owner, struct domain *pg_owner)
- define_get_linear_pagetable (l2)
- define_get_linear_pagetable (l3)
- void put_page_from_l1e (l1_pgentry_t l1e, struct domain *l1e_owner)
- void put_page (struct page_info *page)
- struct domain * page_get_owner_and_reference (struct page_info *page)
- int get_page (struct page_info *page, struct domain *domain)
- int free_page_type (struct page_info *page, unsigned long type, int preemptible)
- void put_page_type (struct page_info *page)
- int get_page_type (struct page_info *page, unsigned long type)
- int put_page_type_preemptible (struct page_info *page)
- int get_page_type_preemptible (struct page_info *page, unsigned long type)
- void clear_superpage_mark (struct page_info *page)
- int new_guest_cr3 (unsigned long mfn)
- int do_mmuext_op (XEN_GUEST_HANDLE(mmuext_op_t) uops, unsigned int count, XEN_GUEST_HANDLE(uint) pdone, unsigned int foreigndom)
- int do_mmu_update (XEN_GUEST_HANDLE(mmu_update_t) ureqs, unsigned int count, XEN_GUEST_HANDLE(uint) pdone, unsigned int foreigndom)
- int create_grant_host_mapping (uint64_t addr, unsigned long frame, unsigned int flags, unsigned int cache_flags)
- int replace_grant_host_mapping (uint64_t addr, unsigned long frame, uint64_t new_addr, unsigned int flags)
- int donate_page (struct domain *d, struct page_info *page, unsigned int memflags)
- int steal_page (struct domain *d, struct page_info *page, unsigned int memflags)
- int page_make_sharable (struct domain *d, struct page_info *page, int expected_refcnt)
- int page_make_private (struct domain *d, struct page_info *page)
- int do_update_va_mapping (unsigned long va, u64 val64, unsigned long flags)
- int do_update_va_mapping_otherdomain (unsigned long va, u64 val64, unsigned long flags, domid_t domid)

- void destroy_gdt (struct vcpu *v)
- long set_gdt (struct vcpu *v, unsigned long *frames, unsigned int entries)
- long do_set_gdt (XEN_GUEST_HANDLE(ulong) frame_list, unsigned int entries)
- long do_update_descriptor (u64 pa, u64 desc)
- DEFINE_XEN_GUEST_HANDLE (e820entry_t)
- long arch_memory_op (int op, XEN_GUEST_HANDLE(void) arg)
- int ptwr_do_page_fault (struct vcpu *v, unsigned long addr, struct cpu_user_regs *regs)
- void free_xen_pagetable (void *v)
- int map_pages_to_xen (unsigned long virt, unsigned long mfn, unsigned long nr_mfns, unsigned int flags)
- void destroy_xen_mappings (unsigned long s, unsigned long e)
- void __set_fixmap (enum fixed_addresses idx, unsigned long mfn, unsigned long flags)
- void memguard_guard_stack (void *p)
- void memguard_unguard_stack (void *p)

Variables

- bool_t __read_mostly opt_allow_superpage

4.4.1 Macro Definition Documentation

4.4.1.1 #define adjust_guest_l1e(_p, _d) ((void)(_d))

4.4.1.2 #define adjust_guest_l2e(_p, _d) ((void)(_d))

4.4.1.3 #define adjust_guest_l3e(_p, _d) ((void)(_d))

4.4.1.4 #define alloc_l4_table(*page, preemptible*) (-EINVAL)

4.4.1.5 #define define_get_linear_pagetable(*level*)

4.4.1.6 #define fixmap_domain_page(*mfn*) mfn_to_virt(mfn)

4.4.1.7 #define fixunmap_domain_page(*ptr*) ((void)(ptr))

4.4.1.8 #define flush_area(*v, f*)

Value:

```
(!local_irq_is_enabled() ?                \
                         flush_area_local((const void *)v, f) : \
                         flush_area_all((const void *)v, f))
```

4.4.1.9 #define free_l4_table(*page, preemptible*) (-EINVAL)

4.4.1.10 #define l1_disallow_mask(d)

Value:

```
((d != dom_io) &&                                          \
     (rangeset_is_empty((d)->iomem_caps) &&                    \
      rangeset_is_empty((d)->arch.ioport_caps) &&              \
      !has_arch_pdevs(d) &&                                    \
      !is_hvm_domain(d)) ?                                     \
     L1_DISALLOW_MASK : (L1_DISALLOW_MASK & ~PAGE_CACHE_ATTRS))
```

4.4.1.11 #define l1f_to_lNf(f) (((f) & _PAGE_PRESENT) ? ((f) | _PAGE_PSE) : (f))

4.4.1.12 #define l3_disallow_mask(d) L3_DISALLOW_MASK

4.4.1.13 #define lNf_to_l1f(f) (((f) & _PAGE_PRESENT) ? ((f) & ~_PAGE_PSE) : (f))

4.4.1.14 #define MEM_LOG(_f, _a...) gdprintk(XENLOG_WARNING , _f "\n" , ## _a)

4.4.1.15 #define pae_flush_pgd(mfn, idx, nl3e) ((void)0)

4.4.1.16 #define PAGE_CACHE_ATTRS (_PAGE_PAT|_PAGE_PCD|_PAGE_PWT)

4.4.1.17 #define PTE_UPDATE_WITH_CMPXCHG

4.4.1.18 #define unadjust_guest_l3e(_p, _d) ((void)(_d))

4.4.1.19 #define UPDATE_ENTRY(_t, _p, _o, _n, _m, _v, _ad)

Value:

```
update_intpte(&_t ## e_get_intpte(*(_p)),                        \
                 _t ## e_get_intpte(_o), _t ## e_get_intpte(_n),   \
                 (_m), (_v), (_ad))
```

4.4.2 Typedef Documentation

4.4.2.1 typedef struct e820entry e820entry_t

4.4.3 Function Documentation

4.4.3.1 l1_pgentry_t __attribute__ ((__section__(".bss.page_aligned")))

4.4.3.2 void __set_fixmap (enum fixed_addresses *idx,* unsigned long *mfn,* unsigned long *flags*)

4.4.3.3 void __init arch_init_memory (void)

4.4.3.4 long arch_memory_op (int *op,* XEN_GUEST_HANDLE(void) *arg*)

4.4.3.5 boolean_param ("allowsuperpage" , opt_allow_superpage)

4.4.3.6 void clear_superpage_mark (struct page_info * *page*)

4.4.3.7 int create_grant_host_mapping (uint64_t *addr,* unsigned long *frame,* unsigned int *flags,* unsigned int *cache_flags*)

4.4.3.8 define_get_linear_pagetable (l2)

NB. Virtual address 'l2e' maps to a machine address within frame 'pfn'.

4.4.3.9 define_get_linear_pagetable (l3)

4.4.3.10 DEFINE_XEN_GUEST_HANDLE (e820entry_t)

4.4.3.11 void destroy_gdt (struct vcpu * *v*)

4.4.3.12 void destroy_xen_mappings (unsigned long *s,* unsigned long *e*)

4.4.3.13 int do_mmu_update (XEN_GUEST_HANDLE(mmu_update_t) *ureqs,* unsigned int *count,* XEN_GUEST_HANDLE(uint) *pdone,* unsigned int *foreigndom*)

4.4.3.14 int do_mmuext_op (XEN_GUEST_HANDLE(mmuext_op_t) *uops,* unsigned int *count,* XEN_GUEST_HANDLE(uint) *pdone,* unsigned int *foreigndom*)

4.4.3.15 long do_set_gdt (XEN_GUEST_HANDLE(ulong) *frame_list,* unsigned int *entries*)

4.4.3.16 long do_update_descriptor (u64 *pa,* u64 *desc*)

4.4.3.17 int do_update_va_mapping (unsigned long *va,* u64 *val64,* unsigned long *flags*)

4.4.3.18 int do_update_va_mapping_otherdomain (unsigned long *va,* u64 *val64,* unsigned long *flags,* domid_t *domid*)

4.4.3.19 unsigned long domain_get_maximum_gpfn (struct domain ∗ *d*)

4.4.3.20 int donate_page (struct domain ∗ *d,* struct page_info ∗ *page,* unsigned int *memflags*)

4.4.3.21 int free_page_type (struct page_info ∗ *page,* unsigned long *type,* int *preemptible*)

4.4.3.22 void free_xen_pagetable (void ∗ *v*)

4.4.3.23 int get_page (struct page_info ∗ *page,* struct domain ∗ *domain*)

4.4.3.24 int get_page_from_l1e (l1_pgentry_t *l1e,* struct domain ∗ *l1e_owner,* struct domain ∗ *pg_owner*)

4.4.3.25 int get_page_type (struct page_info ∗ *page,* unsigned long *type*)

4.4.3.26 int get_page_type_preemptible (struct page_info ∗ *page,* unsigned long *type*)

4.4.3.27 void __init init_frametable (void)

4.4.3.28 int is_iomem_page (unsigned long *mfn*)

4.4.3.29 void make_cr3 (struct vcpu ∗ *v,* unsigned long *mfn*)

4.4.3.30 int map_ldt_shadow_page (unsigned int *off*)

Map shadow page at offset .

4.4.3.31 int map_pages_to_xen (unsigned long *virt,* unsigned long *mfn,* unsigned long *nr_mfns,* unsigned int *flags*)

4.4.3.32 void memguard_guard_stack (void ∗ *p*)

4.4.3.33 void memguard_unguard_stack (void ∗ *p*)

4.4.3.34 int new_guest_cr3 (unsigned long *mfn*)

4.4.3.35 struct domain∗ page_get_owner_and_reference (struct page_info ∗ *page*)

4.4.3.36 int page_is_ram_type (unsigned long *mfn,* unsigned long *mem_type*)

4.4.3.37 int page_make_private (struct domain ∗ *d,* struct page_info ∗ *page*)

4.4.3.38 int page_make_sharable (struct domain ∗ *d,* struct page_info ∗ *page,* int *expected_refcnt*)

4.4.3.39 int ptwr_do_page_fault (struct vcpu * *v*, unsigned long *addr*, struct cpu_user_regs * *regs*)

Write page fault handler: check if guest is trying to modify a PTE.

4.4.3.40 void put_page (struct page_info * *page*)

4.4.3.41 void put_page_from_l1e (l1_pgentry_t *l1e*, struct domain * *l1e_owner*)

4.4.3.42 void put_page_type (struct page_info * *page*)

4.4.3.43 int put_page_type_preemptible (struct page_info * *page*)

4.4.3.44 int replace_grant_host_mapping (uint64_t *addr*, unsigned long *frame*, uint64_t *new_addr*, unsigned int *flags*)

4.4.3.45 long set_gdt (struct vcpu * *v*, unsigned long * *frames*, unsigned int *entries*)

4.4.3.46 void share_xen_page_with_guest (struct page_info * *page*, struct domain * *d*, int *readonly*)

4.4.3.47 void share_xen_page_with_privileged_guests (struct page_info * *page*, int *readonly*)

4.4.3.48 int steal_page (struct domain * *d*, struct page_info * *page*, unsigned int *memflags*)

4.4.3.49 void update_cr3 (struct vcpu * *v*)

Should be called after CR3 is updated.

Uses values found in vcpu->arch.(guest_table and guest_table_user), and for HVM guests, arch.monitor_table and hvm's guest CR3.

Update ref counts to shadow tables appropriately.

4.4.3.50 void write_ptbase (struct vcpu * *v*)

4.4.4 Variable Documentation

4.4.4.1 bool_t __read_mostly opt_allow_superpage

4.5 xen/extras/mini-os/blkfront.c File Reference

```
#include <stdint.h>
#include <mini-os/os.h>
#include <mini-os/xenbus.h>
#include <mini-os/events.h>
#include <errno.h>
#include <xen/io/blkif.h>
#include <xen/io/protocols.h>
#include <mini-os/gnttab.h>
#include <mini-os/xmalloc.h>
#include <time.h>
#include <mini-os/blkfront.h>
#include <mini-os/lib.h>
#include <fcntl.h>
```

Data Structures

- struct blk_buffer
- struct blkfront_dev

Macros

- #define strtoul simple_strtoul
- #define BLK_RING_SIZE __RING_SIZE((struct blkif_sring ∗)0, PAGE_SIZE)
- #define GRANT_INVALID_REF 0

Functions

- DECLARE_WAIT_QUEUE_HEAD (blkfront_queue)
- void blkfront_handler (evtchn_port_t port, struct pt_regs ∗regs, void ∗data)
- struct blkfront_dev ∗ init_blkfront (char ∗_nodename, struct blkfront_info ∗info)
- void shutdown_blkfront (struct blkfront_dev ∗dev)
- void blkfront_aio (struct blkfront_aiocb ∗aiocbp, int write)
- void blkfront_io (struct blkfront_aiocb ∗aiocbp, int write)
- void blkfront_aio_push_operation (struct blkfront_aiocb ∗aiocbp, uint8_t op)
- void blkfront_sync (struct blkfront_dev ∗dev)
- int blkfront_aio_poll (struct blkfront_dev ∗dev)

4.5.1 Macro Definition Documentation

4.5.1.1 #define BLK_RING_SIZE __RING_SIZE((struct blkif_sring ∗)0, PAGE_SIZE)

4.5.1.2 #define GRANT_INVALID_REF 0

4.5.1.3 #define strtoul simple_strtoul

4.5.2 Function Documentation

4.5.2.1 void blkfront_aio (struct blkfront_aiocb ∗ *aiocbp*, int *write*)

Issue an aio

4.5.2.2 int blkfront_aio_poll (struct blkfront_dev ∗ *dev*)

4.5.2.3 void blkfront_aio_push_operation (struct blkfront_aiocb ∗ *aiocbp*, uint8_t *op*)

4.5.2.4 void blkfront_handler (evtchn_port_t *port*, struct pt_regs ∗ *regs*, void ∗ *data*)

4.5.2.5 void blkfront_io (struct blkfront_aiocb ∗ *aiocbp*, int *write*)

4.5.2.6 void blkfront_sync (struct blkfront_dev ∗ *dev*)

4.5.2.7 DECLARE_WAIT_QUEUE_HEAD (blkfront_queue)

4.5.2.8 struct blkfront_dev∗ init_blkfront (char ∗ *_nodename*, struct blkfront_info ∗ *info*)

4.5.2.9 void shutdown_blkfront (struct blkfront_dev ∗ *dev*)

4.6 xen/extras/mini-os/console/console.c File Reference

```
#include <mini-os/types.h>
#include <mini-os/wait.h>
#include <mini-os/mm.h>
#include <mini-os/hypervisor.h>
#include <mini-os/events.h>
#include <mini-os/os.h>
#include <mini-os/lib.h>
#include <mini-os/xenbus.h>
#include <xen/io/console.h>
```

Macros

- #define USE_XEN_CONSOLE

Functions

- void xencons_rx (char ∗buf, unsigned len, struct pt_regs ∗regs)
- void xencons_tx (void)
- void console_print (struct consfront_dev ∗dev, char ∗data, int length)
- void print (int direct, const char ∗fmt, va_list args)
- void printk (const char ∗fmt,...)
- void xprintk (const char ∗fmt,...)
- void init_console (void)
- void fini_console (struct consfront_dev ∗dev)

4.6.1 Macro Definition Documentation

4.6.1.1 #define USE_XEN_CONSOLE

4.6.2 Function Documentation

4.6.2.1 void console_print (struct consfront_dev ∗ *dev,* char ∗ *data,* int *length*)

4.6.2.2 void fini_console (struct consfront_dev ∗ *dev*)

4.6.2.3 void init_console (void)

4.6.2.4 void print (int *direct,* const char ∗ *fmt,* va_list *args*)

4.6.2.5 void printk (const char ∗ *fmt,* ...)

4.6.2.6 void xencons_rx (char ∗ *buf,* unsigned *len,* struct pt_regs ∗ *regs*)

4.6.2.7 void xencons_tx (void)

4.6.2.8 void xprintk (const char ∗ *fmt,* ...)

4.7 xen/extras/mini-os/discovery.c File Reference

4.8 xen/extras/mini-os/include/blkfront.h File Reference

```
#include <mini-os/wait.h>
#include <xen/io/blkif.h>
#include <mini-os/types.h>
```

Data Structures

- struct blkfront_aiocb
- struct blkfront_info

Macros

- #define blkfront_aio_read(aiocbp) blkfront_aio(aiocbp, 0)
- #define blkfront_aio_write(aiocbp) blkfront_aio(aiocbp, 1)
- #define blkfront_read(aiocbp) blkfront_io(aiocbp, 0)
- #define blkfront_write(aiocbp) blkfront_io(aiocbp, 1)

Functions

- struct blkfront_dev ∗ init_blkfront (char ∗nodename, struct blkfront_info ∗info)
- void blkfront_aio (struct blkfront_aiocb ∗aiocbp, int write)
- void blkfront_io (struct blkfront_aiocb ∗aiocbp, int write)
- void blkfront_aio_push_operation (struct blkfront_aiocb ∗aiocbp, uint8_t op)
- int blkfront_aio_poll (struct blkfront_dev ∗dev)
- void blkfront_sync (struct blkfront_dev ∗dev)
- void shutdown_blkfront (struct blkfront_dev ∗dev)

Variables

- struct wait_queue_head blkfront_queue

4.8.1 Macro Definition Documentation

4.8.1.1 #define blkfront_aio_read(*aiocbp*) blkfront_aio(aiocbp, 0)

4.8.1.2 #define blkfront_aio_write(*aiocbp*) blkfront_aio(aiocbp, 1)

4.8.1.3 #define blkfront_read(*aiocbp*) blkfront_io(aiocbp, 0)

4.8.1.4 #define blkfront_write(*aiocbp*) blkfront_io(aiocbp, 1)

4.8.2 Function Documentation

4.8.2.1 void blkfront_aio (struct blkfront_aiocb ∗ *aiocbp,* int *write*)

Issue an aio

4.8.2.2 int blkfront_aio_poll (struct blkfront_dev * *dev*)

4.8.2.3 void blkfront_aio_push_operation (struct blkfront_aiocb * *aiocbp*, uint8_t *op*)

4.8.2.4 void blkfront_io (struct blkfront_aiocb * *aiocbp*, int *write*)

4.8.2.5 void blkfront_sync (struct blkfront_dev * *dev*)

4.8.2.6 struct blkfront_dev* init_blkfront (char * *nodename*, struct blkfront_info * *info*)

4.8.2.7 void shutdown_blkfront (struct blkfront_dev * *dev*)

4.8.3 Variable Documentation

4.8.3.1 struct wait_queue_head blkfront_queue

4.9 xen/extras/mini-os/include/byteorder.h File Reference

```
#include <mini-os/byteswap.h>
#include <mini-os/endian.h>
```

Macros

- #define be16_to_cpu(v) bswap_16(v)
- #define be32_to_cpu(v) bswap_32(v)
- #define be64_to_cpu(v) bswap_64(v)
- #define le16_to_cpu(v) (v)
- #define le32_to_cpu(v) (v)
- #define le64_to_cpu(v) (v)
- #define cpu_to_be16(v) be16_to_cpu(v)
- #define cpu_to_be32(v) be32_to_cpu(v)
- #define cpu_to_be64(v) be64_to_cpu(v)
- #define cpu_to_le16(v) le16_to_cpu(v)
- #define cpu_to_le32(v) le32_to_cpu(v)
- #define cpu_to_le64(v) le64_to_cpu(v)

4.9.1 Macro Definition Documentation

4.9.1.1 #define be16_to_cpu(*v*) bswap_16(v)

4.9.1.2 #define be32_to_cpu(*v*) bswap_32(v)

4.9.1.3 #define be64_to_cpu(*v*) bswap_64(v)

4.9.1.4 #define cpu_to_be16(*v*) be16_to_cpu(v)

4.9.1.5 #define cpu_to_be32(*v*) be32_to_cpu(v)

4.9.1.6 #define cpu_to_be64(*v*) be64_to_cpu(v)

4.9.1.7 #define cpu_to_le16(*v*) le16_to_cpu(v)

4.9.1.8 #define cpu_to_le32(*v*) le32_to_cpu(v)

4.9.1.9 #define cpu_to_le64(*v*) le64_to_cpu(v)

4.9.1.10 #define le16_to_cpu(*v*) (v)

4.9.1.11 #define le32_to_cpu(*v*) (v)

4.9.1.12 #define le64_to_cpu(*v*) (v)

4.10 xen/extras/mini-os/include/byteswap.h File Reference

```
#include <mini-os/types.h>
```

Macros

- #define bswap_16(x)
- #define bswap_32(x)
- #define bswap_64(x)

4.10.1 Macro Definition Documentation

4.10.1.1 #define bswap_16(*x*)

Value:

```
((uint16_t)(                          \
             (((uint16_t)(x) & (uint16_t)0x00ffU) << 8) |                  \
             (((uint16_t)(x) & (uint16_t)0xff00U) >> 8)))
```

4.10.1.2 #define bswap_32(*x*)

Value:

```
((uint32_t)(                          \
             (((uint32_t)(x) & (uint32_t)0x000000ffUL) << 24) |            \
             (((uint32_t)(x) & (uint32_t)0x0000ff00UL) <<  8) |            \
             (((uint32_t)(x) & (uint32_t)0x00ff0000UL) >>  8) |            \
             (((uint32_t)(x) & (uint32_t)0xff000000UL) >> 24)))
```

4.10.1.3 #define bswap_64(*x*)

Value:

```
((uint64_t)(                          \
             (((uint64_t)(x) & (uint64_t)0x00000000000000ffULL) << 56) |   \
             (((uint64_t)(x) & (uint64_t)0x000000000000ff00ULL) << 40) |   \
             (((uint64_t)(x) & (uint64_t)0x0000000000ff0000ULL) << 24) |   \
             (((uint64_t)(x) & (uint64_t)0x00000000ff000000ULL) <<  8) |   \
             (((uint64_t)(x) & (uint64_t)0x000000ff00000000ULL) >>  8) |   \
             (((uint64_t)(x) & (uint64_t)0x0000ff0000000000ULL) >> 24) |   \
             (((uint64_t)(x) & (uint64_t)0x00ff000000000000ULL) >> 40) |   \
             (((uint64_t)(x) & (uint64_t)0xff00000000000000ULL) >> 56)))
```

4.11 xen/extras/mini-os/include/discovery.h File Reference

Functions

- void discovery (void)
- void discovery_start_kernel (void *p)

4.11.1 Function Documentation

4.11.1.1 void discovery (void)

4.11.1.2 void discovery_start_kernel (void * *p*)

4.12 xen/extras/mini-os/include/endian.h File Reference

```
#include <arch_endian.h>
#include <arch_wordsize.h>
```

Macros

- #define __LITTLE_ENDIAN 1234
- #define __BIG_ENDIAN 4321
- #define __PDP_ENDIAN 3412
- #define ARCH_ENDIAN_H
- #define __FLOAT_WORD_ORDER __BYTE_ORDER

4.12.1 Macro Definition Documentation

4.12.1.1 #define __BIG_ENDIAN 4321

4.12.1.2 #define __FLOAT_WORD_ORDER __BYTE_ORDER

4.12.1.3 #define __LITTLE_ENDIAN 1234

4.12.1.4 #define __PDP_ENDIAN 3412

4.12.1.5 #define ARCH_ENDIAN_H

4.13 xen/extras/mini-os/include/ia64/arch_endian.h File Reference

4.14 xen/extras/mini-os/include/x86/arch_endian.h File Reference

4.15 xen/extras/mini-os/include/ia64/arch_wordsize.h File Reference

Macros

- #define __WORDSIZE 64

4.15.1 Macro Definition Documentation

4.15.1.1 #define __WORDSIZE 64

4.16 xen/extras/mini-os/include/x86/x86_32/arch_wordsize.h File Reference

Macros

- #define __WORDSIZE 32

4.16.1 Macro Definition Documentation

4.16.1.1 #define __WORDSIZE 32

4.17 xen/extras/mini-os/include/x86/x86_64/arch_wordsize.h File Reference

Macros

- #define __WORDSIZE 64
- #define __WORDSIZE_COMPAT32 1

4.17.1 Macro Definition Documentation

4.17.1.1 #define __WORDSIZE 64

4.17.1.2 #define __WORDSIZE_COMPAT32 1

4.18 xen/extras/mini-os/include/iorw.h File Reference

```
#include <mini-os/types.h>
```

Functions

- void iowrite8 (volatile void *addr, uint8_t val)
- void iowrite32 (volatile void *addr, uint32_t val)
- uint8_t ioread8 (volatile void *addr)
- uint32_t ioread32 (volatile void *addr)

4.18.1 Function Documentation

4.18.1.1 uint32_t ioread32 (volatile void * *addr*)

4.18.1.2 uint8_t ioread8 (volatile void * *addr*)

4.18.1.3 void iowrite32 (volatile void * *addr*, uint32_t *val*)

4.18.1.4 void iowrite8 (volatile void * *addr*, uint8_t *val*)

4.19 xen/extras/mini-os/include/lib.h File Reference

```
#include <stdarg.h>
#include <stddef.h>
#include <xen/xen.h>
#include <xen/event_channel.h>
#include "gntmap.h"
#include <mini-os/console.h>
#include <mini-os/xenbus.h>
```

Data Structures

- struct kvec

Macros

- #define _p(_x) ((void *)(unsigned long)(_x))
- #define RAND_MIX 2654435769U
- #define ARRAY_SIZE(x) (sizeof(x) / sizeof((x)[0]))
- #define ASSERT(x)
- #define BUG_ON(x) ASSERT(!(x))

Functions

- int vsnprintf (char *buf, size_t size, const char *fmt, va_list args)
- int vscnprintf (char *buf, size_t size, const char *fmt, va_list args)
- int snprintf (char *buf, size_t size, const char *fmt,...)
- int scnprintf (char *buf, size_t size, const char *fmt,...)
- int vsprintf (char *buf, const char *fmt, va_list args)
- int sprintf (char *buf, const char *fmt,...)
- int vsscanf (const char *buf, const char *fmt, va_list args)
- int sscanf (const char *buf, const char *fmt,...)
- long simple_strtol (const char *cp, char **endp, unsigned int base)
- unsigned long simple_strtoul (const char *cp, char **endp, unsigned int base)
- long long simple_strtoll (const char *cp, char **endp, unsigned int base)
- unsigned long long simple_strtoull (const char *cp, char **endp, unsigned int base)
- int memcmp (const void *cs, const void *ct, size_t count)
- void * memcpy (void *dest, const void *src, size_t count)
- int strncmp (const char *cs, const char *ct, size_t count)
- int strcmp (const char *cs, const char *ct)
- char * strcpy (char *dest, const char *src)
- char * strncpy (char *dest, const char *src, size_t count)
- void * memset (void *s, int c, size_t count)
- size_t strnlen (const char *s, size_t count)
- size_t strlen (const char *s)
- char * strchr (const char *s, int c)
- char * strrchr (const char *s, int c)
- char * strstr (const char *s1, const char *s2)
- char * strcat (char *dest, const char *src)
- char * strdup (const char *s)
- int rand (void)
- void sanity_check (void)

4.19.1 Macro Definition Documentation

4.19.1.1 #define _p(_x) ((void *)(unsigned long)(_x))

4.19.1.2 #define ARRAY_SIZE(x) (sizeof(x) / sizeof((x)[0]))

4.19.1.3 #define ASSERT(x)

Value:

```
do {                                                        \
    if (!(x)) {                                             \
        printk("ASSERTION FAILED: %s at %s:%d.\n",          \
               # x ,                                       \
               __FILE__,                                    \
               __LINE__);                                   \
        BUG();                                              \
    }                                                       \
} while(0)
```

4.19.1.4 #define BUG_ON(x) ASSERT(!(x))

4.19.1.5 #define RAND_MIX 2654435769U

4.19.2 Function Documentation

4.19.2.1 int memcmp (const void * *cs*, const void * *ct*, size_t *count*)

4.19.2.2 void* memcpy (void * *dest*, const void * *src*, size_t *count*)

4.19.2.3 void* memset (void * *s*, int *c*, size_t *count*)

4.19.2.4 int rand (void)

4.19.2.5 void sanity_check (void)

4.19.2.6 int scnprintf (char * *buf*, size_t *size*, const char * *fmt*, ...)

4.19.2.7 long simple_strtol (const char * *cp*, char ** *endp*, unsigned int *base*)

4.19.2.8 long long simple_strtoll (const char * *cp*, char ** *endp*, unsigned int *base*)

4.19.2.9 unsigned long simple_strtoul (const char * *cp*, char ** *endp*, unsigned int *base*)

4.19.2.10 unsigned long long simple_strtoull (const char * *cp*, char ** *endp*, unsigned int *base*)

4.19.2.11 int snprintf (char * *buf*, size_t *size*, const char * *fmt*, ...)

4.19.2.12 int sprintf (char * *buf*, const char * *fmt*, ...)

4.19.2.13 int sscanf (const char * *buf*, const char * *fmt*, ...)

4.19.2.14 char* strcat (char * *dest*, const char * *src*)

4.19.2.15 char* strchr (const char * *s*, int *c*)

4.19.2.16 int strcmp (const char * *cs*, const char * *ct*)

4.19.2.17 char* strcpy (char * *dest*, const char * *src*)

4.19.2.18 char* strdup (const char * *s*)

4.19.2.19 size_t strlen (const char * *s*)

4.19.2.20 int strncmp (const char * *cs*, const char * *ct*, size_t *count*)

4.19.2.21 char* strncpy (char * *dest*, const char * *src*, size_t *count*)

4.19.2.22 size_t strnlen (const char * *s*, size_t *count*)

4.19.2.23 char* strrchr (const char * *s*, int *c*)

4.19.2.24 char* strstr (const char * *s1*, const char * *s2*)

4.19.2.25 int vscnprintf (char * *buf*, size_t *size*, const char * *fmt*, va_list *args*)

4.19.2.26 int vsnprintf (char * *buf*, size_t *size*, const char * *fmt*, va_list *args*)

4.19.2.27 int vsprintf (char * *buf*, const char * *fmt*, va_list *args*)

4.19.2.28 int vsscanf (const char * *buf*, const char * *fmt*, va_list *args*)

4.20 xen/extras/mini-os/include/minios_macros.h File Reference

4.21 xen/stubdom/domt/integrated/minios_macros.h File Reference

Macros

- #define DEBUG_ON
- #define DEBUG_TEST 1
- #define PANIC(...) { ERROR("PANIC() " __VA_ARGS__); while (1); }
- #define DEBUG(fmt,...)
- #define ERROR(fmt,...)
- #define XENBUS_XWRITE(path, value)
- #define XENBUS_XWRITE_INT(path, value)
- #define XENBUS_XPRINTF(path, key, value,...)
- #define XENBUS_XSET_PERMS(path, domid, perm)
- #define XENBUS_XREAD(path, value)
- #define XENBUS_XREAD2(path1, path2, value)
- #define XENBUS_READ2(path1, path2, value)
- #define XENBUS_XREAD_INTEGER2(path1, path2, value)

4.21.1 Macro Definition Documentation

4.21.1.1 #define DEBUG(*fmt*, ...)

Value:

```
do { if (DEBUG_TEST) printk("%s:%d:%s(): " fmt, __FILE__,                    \
        __LINE__, __func__, ##__VA_ARGS__); } while (0)
```

4.21.1.2 #define DEBUG_ON

Supplemental print and debug macros.

These are somewhat more verbose than the standard printk() offered by Mini-OS.

4.21.1.3 #define DEBUG_TEST 1

4.21.1.4 #define ERROR(*fmt, ...*)

Value:

```
printk("***ERROR %s:%d:%s(): " fmt, __FILE__, __LINE__,                    \
           __func__, ##__VA_ARGS__)
```

4.21.1.5 #define PANIC(...) { ERROR("PANIC() " __VA_ARGS__); while (1); }

4.21.1.6 #define XENBUS_READ2(*path1, path2, value*)

Value:

```
do { char *path;                                                        \
        asprintf(&path, "%s/%s", path1, path2);                             \
        xenbus_read(XBT_NIL, path, value);                                  \
        free(path);                                                         \
    } while (0)
```

4.21.1.7 #define XENBUS_XPRINTF(*path, key, value, ...*)

Value:

```
do { char *msg = xenbus_printf(XBT_NIL, path, key, value, ##__VA_ARGS__);  \
        if (msg) {                                                          \
            PANIC("Error %s while writing xenstore at %s/%s\n",             \
                  msg, path, key);                                          \
        } } while (0)
```

4.21.1.8 #define XENBUS_XREAD(*path, value*)

Value:

```
do { char *msg = xenbus_read(XBT_NIL, path, value);                     \
        if (msg) {                                                          \
            PANIC("Error %s while reading xenstore at %s\n",                \
                  msg, path);                                               \
        } } while (0)
```

4.21.1.9 #define XENBUS_XREAD2(*path1, path2, value*)

Value:

```
do { char *msg, *path;                                                  \
        asprintf(&path, "%s/%s", path1, path2);                             \
        msg = xenbus_read(XBT_NIL, path, value);                            \
        if (msg) {                                                          \
            PANIC("Error %s while reading xenstore at %s\n", msg, path);    \
        }                                                                   \
        free(path);                                                         \
    } while (0)
```

4.21.1.10 #define XENBUS_XREAD_INTEGER2(*path1, path2, value*)

Value:

```
do { char *path;                                                        \
        asprintf(&path, "%s/%s", path1, path2);                         \
        *value = xenbus_read_integer(path);                             \
        if (*value == -1) {                                             \
            PANIC("Error while reading xenstore at %s\n", path);        \
        }                                                               \
        free(path);                                                     \
    } while (0)
```

4.21.1.11 #define XENBUS_XSET_PERMS(*path, domid, perm*)

Value:

```
do { char *msg = xenbus_set_perms(XBT_NIL, path, domid, perm);          \
        if (msg) {                                                      \
            PANIC("Error %s while setting permissions in xenstore at %s\n",  \
                  msg, path);                                           \
        } } while (0)
```

4.21.1.12 #define XENBUS_XWRITE(*path, value*)

Value:

```
do { char *msg = xenbus_write(XBT_NIL, path, value);                    \
        if (msg) { \
            PANIC("Error %s while writing xenstore at %s\n", msg, path);  \
        } } while (0)
```

4.21.1.13 #define XENBUS_XWRITE_INT(*path, value*)

Value:

```
do { char *msg, *tmp;                                                   \
        asprintf(&tmp, "%d", value);                                    \
        msg = xenbus_write(XBT_NIL, path, tmp);                         \
        if (msg) {                                                      \
            PANIC("Error %s while writing xenstore at %s\n", msg, path);  \
        }                                                               \
        free(tmp);                                                      \
    } while (0)
```

4.22 xen/extras/mini-os/include/tpm_tis.h File Reference

```
#include <mini-os/os.h>
#include <mini-os/types.h>
#include <mini-os/byteorder.h>
```

Macros

- #define TPM_TIS_EN_LOCL0 1
- #define TPM_TIS_EN_LOCL1 (1 << 1)
- #define TPM_TIS_EN_LOCL2 (1 << 2)
- #define TPM_TIS_EN_LOCL3 (1 << 3)

- #define TPM_TIS_EN_LOCL4 (1 << 4)
- #define TPM_TIS_EN_LOCLALL (TPM_TIS_EN_LOCL0 | TPM_TIS_EN_LOCL1 | TPM_TIS_EN_LOCL2 | TPM_TIS_EN_LOCL3 | TPM_TIS_EN_LOCL4)
- #define TPM_TIS_LOCL_INT_TO_FLAG(x) (1 << x)
- #define TPM_BASEADDR 0xFED40000
- #define TPM_PROBE_IRQ 0xFFFF

Functions

- struct tpm_chip * init_tpm_tis (unsigned long baseaddr, int localities, unsigned int irq)
- void shutdown_tpm_tis (struct tpm_chip *tpm)
- int tpm_tis_request_locality (struct tpm_chip *tpm, int locality)
- int tpm_tis_cmd (struct tpm_chip *tpm, uint8_t *req, size_t reqlen, uint8_t **resp, size_t *resplen)
- int tpm_tis_transmit (struct tpm_chip *chip, struct tpm_cmd *cmd, int len, const char *desc)

4.22.1 Macro Definition Documentation

4.22.1.1 #define TPM_BASEADDR 0xFED40000

4.22.1.2 #define TPM_PROBE_IRQ 0xFFFF

4.22.1.3 #define TPM_TIS_EN_LOCL0 1

4.22.1.4 #define TPM_TIS_EN_LOCL1 (1 << 1)

4.22.1.5 #define TPM_TIS_EN_LOCL2 (1 << 2)

4.22.1.6 #define TPM_TIS_EN_LOCL3 (1 << 3)

4.22.1.7 #define TPM_TIS_EN_LOCL4 (1 << 4)

4.22.1.8 #define TPM_TIS_EN_LOCLALL (TPM_TIS_EN_LOCL0 | TPM_TIS_EN_LOCL1 | TPM_TIS_EN_LOCL2 | TPM_TIS_EN_LOCL3 | TPM_TIS_EN_LOCL4)

4.22.1.9 #define TPM_TIS_LOCL_INT_TO_FLAG(*x*) (1 << x)

4.22.2 Function Documentation

4.22.2.1 struct tpm_chip* init_tpm_tis (unsigned long *baseaddr,* int *localities,* unsigned int *irq*)

4.22.2.2 void shutdown_tpm_tis (struct tpm_chip * *tpm*)

4.22.2.3 int tpm_tis_cmd (struct tpm_chip * *tpm,* uint8_t * *req,* size_t *reqlen,* uint8_t ** *resp,* size_t * *resplen*)

4.22.2.4 int tpm_tis_request_locality (struct tpm_chip * *tpm,* int *locality*)

4.22.2.5 int tpm_tis_transmit (struct tpm_chip * *chip,* struct tpm_cmd * *cmd,* int *len,* const char * *desc*)

4.23 xen/extras/mini-os/include/tpmback.h File Reference

```
#include <xen/io/tpmif.h>
#include <xen/io/xenbus.h>
#include <mini-os/types.h>
#include <xen/xen.h>
```

Data Structures

- struct tpmcmd

Typedefs

- typedef struct tpmcmd tpmcmd_t

Functions

- void init_tpmback (char **exclusive_uuids)
- void shutdown_tpmback (void)
- tpmcmd_t * tpmback_req_any (void)
- tpmcmd_t * tpmback_req (domid_t domid, unsigned int handle)
- void tpmback_resp (tpmcmd_t *tpmcmd)
- int tpmback_wait_for_frontend_connect (domid_t *domid, unsigned int *handle)
- int tpmback_num_frontends (void)
- char * tpmback_get_uuid (domid_t domid, unsigned int handle)
- void tpmback_set_open_callback (void(*cb)(domid_t, unsigned int))
- void tpmback_set_close_callback (void(*cb)(domid_t, unsigned int))
- void tpmback_set_suspend_callback (void(*cb)(domid_t, unsigned int))
- void tpmback_set_resume_callback (void(*cb)(domid_t, unsigned int))

4.23.1 Typedef Documentation

4.23.1.1 typedef struct tpmcmd tpmcmd_t

4.23.2 Function Documentation

4.23.2.1 void init_tpmback (char ** *exclusive_uuids*)

Initialize the tpm backend driver - This is NULL terminated list of vtpm uuid strings. If this list is non-empty, then only frontend domains with vtpm uuid's matching entries in this list will be allowed to connect. Other connections will be immediatly closed. Set this argument to NULL to allow any vtpm to connect.

4.23.2.2 void shutdown_tpmback (void)

Shutdown tpm backend driver

4.23.2.3 char* tpmback_get_uuid (domid_t *domid,* unsigned int *handle*)

Returns the uuid of the specified frontend, NULL on error

4.23.2.4 int tpmback_num_frontends (void)

returns the number of frontends connected

4.23.2.5 tpmcmd_t* tpmback_req (domid_t *domid,* unsigned int *handle*)

Blocks until a tpm command from the frontend at domid/handle is sent. Returns NULL if domid/handle is not connected, tpmback is shutdown or shutting down, or if there is an error

4.23.2.6 tpmcmd_t* tpmback_req_any (void)

Blocks until a tpm command is sent from any front end. Returns a pointer to the tpm command to handle. Do not try to free this pointer or the req buffer This function will return NULL if the tpm backend driver is shutdown or any other error occurs

4.23.2.7 void tpmback_resp (tpmcmd_t * *tpmcmd*)

Send the response to the tpm command back to the frontend This function will free the tpmcmd object, but you must free the resp buffer yourself

4.23.2.8 void tpmback_set_close_callback (void(*)(domid_t, unsigned int) *cb*)

Specify a function to call when a tpm device disconnects

4.23.2.9 void tpmback_set_open_callback (void(*)(domid_t, unsigned int) *cb*)

Specify a function to call when a new tpm device connects

4.23.2.10 void tpmback_set_resume_callback (void(*)(domid_t, unsigned int) *cb*)

4.23.2.11 void tpmback_set_suspend_callback (void(*)(domid_t, unsigned int) *cb*)

4.23.2.12 int tpmback_wait_for_frontend_connect (domid_t * *domid,* unsigned int * *handle*)

Waits for the first frontend to connect and then sets domid and handle appropriately. If one or more frontends are already connected, this will set domid and handle to one of them arbitrarily. The main use for this function is to wait until a single frontend connection has occured. returns 0 on success, non-zero on failure

4.24 xen/extras/mini-os/include/tpmfront.h File Reference

```
#include <mini-os/types.h>
#include <mini-os/os.h>
#include <mini-os/events.h>
#include <mini-os/wait.h>
#include <xen/xen.h>
#include <xen/io/xenbus.h>
#include <xen/io/tpmif.h>
```

Data Structures

- struct tpmfront_dev

Functions

- struct tpmfront_dev * init_tpmfront (const char *nodename)
- void shutdown_tpmfront (struct tpmfront_dev *dev)
- int tpmfront_cmd (struct tpmfront_dev *dev, uint8_t *req, size_t reqlen, uint8_t **resp, size_t *resplen)

4.24.1 Function Documentation

4.24.1.1 struct tpmfront_dev* init_tpmfront (const char * *nodename*)

4.24.1.2 void shutdown_tpmfront (struct tpmfront_dev * *dev*)

4.24.1.3 int tpmfront_cmd (struct tpmfront_dev * *dev,* uint8_t * *req,* size_t *reqlen,* uint8_t ** *resp,* size_t * *resplen*)

4.25 xen/extras/mini-os/include/uuid.h File Reference

Typedefs

- typedef unsigned char uuid_t

Functions

- void uuid_create (uuid_t *uuid, uint32_t *status)
- void uuid_to_string (uuid_t *uuid, char **str, uint32_t *status)
- int uuid_is_nil (uuid_t *uuid, unsigned int *status)

4.25.1 Typedef Documentation

4.25.1.1 typedef unsigned char uuid_t

4.25.2 Function Documentation

4.25.2.1 void uuid_create (uuid_t * *uuid,* uint32_t * *status*)

just fill the struct with random values for now

4.25.2.2 int uuid_is_nil (uuid_t * *uuid,* unsigned int * *status*)

4.25.2.3 void uuid_to_string (uuid_t * *uuid,* char ** *str,* uint32_t * *status*)

convert the struct to a printable string

4.26 xen/extras/mini-os/include/vfs.h File Reference

4.27 xen/stubdom/domt/integrated/vfs.h File Reference

```
#include <minios_macros.h>
#include <vfs_intf.h>
```

Functions

- int vfs_init (void)
- int vfs_command (struct vfs_cmd *cmd)

4.27.1 Function Documentation

4.27.1.1 int vfs_command (struct vfs_cmd * *cmd*)

4.27.1.2 int vfs_init (void)

See vfs_hooks.h for the bulk of public functions.

The init command has become deprecated. Maybe it can be useful to test if VFS is alive at boottime. It might also play a role in detection of xenvfsd restart/crashing; since then we need to invalidate all vfs file descriptors. Not a priority task, however.

4.28 xen/extras/mini-os/include/vfs_hooks.h File Reference

4.29 xen/stubdom/domt/integrated/vfs_hooks.h File Reference

```
#include <vfs_intf.h>
```

Functions

- int vfs_open (const char *pathname, int flags, mode_t mode)
- int vfs_close (int fd)
- int vfs_stat (const char *path, struct stat *buf)
- off_t vfs_lseek (int fd, off_t offset, int whence)
- ssize_t vfs_read (int fd, void *buf, size_t nbyte)
- ssize_t vfs_write (int fd, const void *buf, size_t nbyte)
- void vfs_major_minor (const char *physpath, int *major, int *minor)
- void * vfs_mmap (void *addr, size_t length, int prot, int flags, int fd, off_t offset)

4.29.1 Function Documentation

4.29.1.1 int vfs_close (int *fd*)

4.29.1.2 off_t vfs_lseek (int *fd*, off_t *offset*, int *whence*)

4.29.1.3 void vfs_major_minor (const char * *physpath*, int * *major*, int * *minor*)

combines the major and minor calls into one

4.29.1.4 void* vfs_mmap (void * *addr*, size_t *length*, int *prot*, int *flags*, int *fd*, off_t *offset*)

For now, mmap is synthetic. it is a TODO to build a native solution via grant tables. We probably want to share a writable gref_map with Dom0 then. Until then, this function does not sync back changes. Since this call is only used in the domainbuilder for reading in the images, priority for this fix is low currently.

4.29.1.5 int vfs_open (const char * *pathname*, int *flags*, mode_t *mode*)

NOTE: munmap, close, are not hooked since minios default implementation suffices. NOTE: newlib implements for us: fread() fwrite() fopen() fclose().

4.29.1.6 ssize_t vfs_read (int *fd*, void * *buf*, size_t *nbyte*)

4.29.1.7 int vfs_stat (const char * *path*, struct stat * *buf*)

4.29.1.8 ssize_t vfs_write (int *fd*, const void * *buf*, size_t *nbyte*)

4.30 xen/extras/mini-os/include/vfs_intf.h File Reference

4.31 xen/stubdom/domt/integrated/vfs_intf.h File Reference

```
#include <sys/stat.h>
#include <fcntl.h>
```

Data Structures

- struct vfs_cmd_init
- struct vfs_cmd_open
- struct vfs_cmd_close
- struct vfs_cmd_stat
- struct vfs_cmd_lseek
- struct vfs_cmd_read
- struct vfs_cmd_write
- struct vfs_cmd_majorminor
- struct vfs_cmd
- struct vfs_gref_map

Macros

- #define VFS_PATH "/local/domain/0/domt/vfs"
- #define PAGE_SIZE 4096
- #define GREFS_PER_GREFMAP ((PAGE_SIZE - sizeof(unsigned int)) / sizeof(unsigned int))
- #define VFS_O_RDONLY 1UL << 1
- #define VFS_O_WRONLY 1UL << 2
- #define VFS_O_RDWR 1UL << 2
- #define VFS_O_CREAT 1UL << 4
- #define VFS_O_TRUNC 1UL << 5

Enumerations

- enum vfs_cmdtype {
  VFS_INIT = 1, VFS_OPEN = 2, VFS_CLOSE = 3, VFS_STAT = 4,
  VFS_LSEEK = 5, VFS_READ = 6, VFS_WRITE = 7, VFS_MAJORMINOR = 8 }

4.31.1 Macro Definition Documentation

4.31.1.1 #define GREFS_PER_GREFMAP ((PAGE_SIZE - sizeof(unsigned int)) / sizeof(unsigned int))

4.31.1.2 #define PAGE_SIZE 4096

4.31.1.3 #define VFS_O_CREAT 1UL << 4

4.31.1.4 #define VFS_O_RDONLY 1UL << 1

Mini-OS (newlib) and Linux don't agree on the values of certain macros, therefore we introduce platform independent conversions.

4.31.1.5 #define VFS_O_RDWR 1UL << 2

4.31.1.6 #define VFS_O_TRUNC 1UL << 5

4.31.1.7 #define VFS_O_WRONLY 1UL << 2

4.31.1.8 #define VFS_PATH "/local/domain/0/domt/vfs"

VFS structures shared between Mini-OS and tools.

4.31.2 Enumeration Type Documentation

4.31.2.1 enum vfs_cmdtype

Enumerator

VFS_INIT

VFS_OPEN

VFS_CLOSE

VFS_STAT

VFS_LSEEK

VFS_READ

VFS_WRITE

VFS_MAJORMINOR

4.32 xen/tools/xenvfsd/include/vfs_intf.h File Reference

4.33 xen/extras/mini-os/include/x86/arch_limits.h File Reference

Macros

- #define __PAGE_SHIFT 12
- #define __PAGE_SIZE (1ULL << __PAGE_SHIFT)
- #define __STACK_SIZE_PAGE_ORDER 6
- #define __STACK_SIZE (__PAGE_SIZE * (1 << __STACK_SIZE_PAGE_ORDER))

4.33.1 Macro Definition Documentation

4.33.1.1 #define __PAGE_SHIFT 12

4.33.1.2 #define __PAGE_SIZE (1ULL << __PAGE_SHIFT)

4.33.1.3 #define __STACK_SIZE (__PAGE_SIZE * (1 << __STACK_SIZE_PAGE_ORDER))

4.33.1.4 #define __STACK_SIZE_PAGE_ORDER 6

4.34 xen/extras/mini-os/kernel.c File Reference

```
#include <mini-os/os.h>
#include <mini-os/hypervisor.h>
#include <mini-os/mm.h>
#include <mini-os/events.h>
#include <mini-os/time.h>
#include <mini-os/types.h>
#include <mini-os/lib.h>
#include <mini-os/sched.h>
#include <mini-os/xenbus.h>
#include <mini-os/gnttab.h>
#include <mini-os/netfront.h>
#include <mini-os/blkfront.h>
#include <mini-os/fbfront.h>
#include <mini-os/pcifront.h>
#include <mini-os/xmalloc.h>
#include <fcntl.h>
#include <xen/features.h>
#include <xen/version.h>
```

Data Structures

- struct blk_req

Macros

- #define WIDTH 800
- #define HEIGHT 600
- #define DEPTH 32

Functions

- void setup_xen_features (void)
- void test_xenbus (void)
- __attribute__ ((weak))
- void start_kernel (start_info_t *si)
- void stop_kernel (void)
- void do_exit (void)

Variables

- uint8_t xen_features [XENFEAT_NR_SUBMAPS *32]

4.34.1 Macro Definition Documentation

4.34.1.1 #define DEPTH 32

4.34.1.2 #define HEIGHT 600

4.34.1.3 #define WIDTH 800

4.34.2 Function Documentation

4.34.2.1 __attribute__ ((weak))

4.34.2.2 void do_exit (void)

This is called whenever an IRET fails in entry.S. This will generally be because an application has got itself into a really bad state (probably a bad CS or SS). It must be killed. Of course, minimal OS doesn't have applications :-)

4.34.2.3 void setup_xen_features (void)

4.34.2.4 void start_kernel (start_info_t * *si*)

4.34.2.5 void stop_kernel (void)

4.34.2.6 void test_xenbus (void)

4.34.3 Variable Documentation

4.34.3.1 uint8_t xen_features[XENFEAT_NR_SUBMAPS *32]

4.35 xen/extras/mini-os/lib/sys.c File Reference

Macros

- #define DEBUG(fmt,...)

4.35.1 Macro Definition Documentation

4.35.1.1 #define DEBUG(*fmt, ...*)

4.36 xen/extras/mini-os/lib/xs.c File Reference

4.37 xen/extras/mini-os/main.c File Reference

4.38 xen/stubdom/domc/main.c File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <xenctrl.h>
#include <os.h>
#include <time.h>
#include <mini-os/gnttab.h>
#include <mini-os/events.h>
#include <sys/select.h>
#include <inttypes.h>
#include <stdio.h>
#include <string.h>
#include <byteorder.h>
#include <mini-os/blkfront.h>
#include "blkback.h"
#include "version.h"
#include "main.h"
#include "crypt.h"
#include "cspipe_intf.h"
```

Macros

- #define DIM(x) (sizeof(x)/sizeof(∗(x)))

Functions

- int wrap_notify_remote_via_evtchn (evtchn_port_t port)
- int main (int argc, char ∗argv[])

Variables

- grant_entry_v1_t ∗ grant_table
- grant_entry_v2_t ∗ grant_table2
- gnttab_setup_table_t setup_op

4.38.1 Macro Definition Documentation

4.38.1.1 #define DIM(*x*) (sizeof(x)/sizeof(∗(x)))

4.38.2 Function Documentation

4.38.2.1 int main (int *argc*, char ∗ *argv[]*)

4.38.2.2 int wrap_notify_remote_via_evtchn (evtchn_port_t *port*)

4.38.3 Variable Documentation

4.38.3.1 grant_entry_v1_t∗ grant_table

4.38.3.2 grant_entry_v2_t∗ grant_table2

4.38.3.3 gnttab_setup_table_t setup_op

4.39 xen/stubdom/domt/main.c File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <xenctrl.h>
#include <os.h>
#include <time.h>
#include <mini-os/gnttab.h>
#include <mini-os/events.h>
#include <sys/select.h>
#include <inttypes.h>
#include <stdio.h>
#include <string.h>
#include <libxl.h>
#include "version.h"
#include "main.h"
#include "caas.h"
#include "xlc.h"
#include "tc.h"
```

Functions

- int main (int argc, char *argv[])

4.39.1 Function Documentation

4.39.1.1 int main (int *argc*, char * *argv[]*)

4.40 xen/extras/mini-os/tpm_tis.c File Reference

```
#include <mini-os/ioremap.h>
#include <mini-os/iorw.h>
#include <mini-os/tpm_tis.h>
#include <mini-os/os.h>
#include <mini-os/sched.h>
#include <mini-os/list.h>
#include <mini-os/byteorder.h>
#include <mini-os/events.h>
#include <mini-os/wait.h>
#include <mini-os/xmalloc.h>
#include <errno.h>
#include <stdbool.h>
```

Data Structures

- struct tpm_input_header
- struct tpm_output_header
- struct stclear_flags_t
- struct tpm_version_t
- struct tpm_version_1_2_t

- struct timeout_t
- struct duration_t
- struct permanent_flags_t
- union cap_t
- struct tpm_getcap_params_in
- struct tpm_getcap_params_out
- struct tpm_readpubek_params_out
- union tpm_cmd_header
- struct tpm_pcrread_out
- struct tpm_pcrread_in
- struct tpm_pcrextend_in
- union tpm_cmd_params
- struct tpm_cmd
- struct tpm_chip

Macros

- #define min(a, b) (((a) < (b)) ? (a) : (b))
- #define TPM_HEADER_SIZE 10
- #define TPM_BUFSIZE 2048
- #define TPM_DIGEST_SIZE 20
- #define TPM_MAX_ORDINAL 243
- #define TPM_MAX_PROTECTED_ORDINAL 12
- #define TPM_PROTECTED_ORDINAL_MASK 0xFF
- #define TPM_DIGEST_SIZE 20
- #define TPM_ERROR_SIZE 10
- #define TPM_RET_CODE_IDX 6
- #define TPM_CAP_FLAG cpu_to_be32(4)
- #define TPM_CAP_PROP cpu_to_be32(5)
- #define CAP_VERSION_1_1 cpu_to_be32(0x06)
- #define CAP_VERSION_1_2 cpu_to_be32(0x1A)
- #define TPM_CAP_PROP_PCR cpu_to_be32(0x101)
- #define TPM_CAP_PROP_MANUFACTURER cpu_to_be32(0x103)
- #define TPM_CAP_FLAG_PERM cpu_to_be32(0x108)
- #define TPM_CAP_FLAG_VOL cpu_to_be32(0x109)
- #define TPM_CAP_PROP_OWNER cpu_to_be32(0x111)
- #define TPM_CAP_PROP_TIS_TIMEOUT cpu_to_be32(0x115)
- #define TPM_CAP_PROP_TIS_DURATION cpu_to_be32(0x120)
- #define TPM_INTERNAL_RESULT_SIZE 200
- #define TPM_TAG_RQU_COMMAND cpu_to_be16(193)
- #define TPM_ORD_GET_CAP cpu_to_be32(101)
- #define TPM_TIMEOUT 5
- #define TPM_ACCESS(t, l) (((uint8_t*)t->pages[l]) + 0x0000)
- #define TPM_INT_ENABLE(t, l) ((uint32_t*)(((uint8_t*)t->pages[l]) + 0x0008))
- #define TPM_INT_VECTOR(t, l) (((uint8_t*)t->pages[l]) + 0x000C)
- #define TPM_INT_STATUS(t, l) (((uint8_t*)t->pages[l]) + 0x0010)
- #define TPM_INTF_CAPS(t, l) ((uint32_t*)(((uint8_t*)t->pages[l]) + 0x0014))
- #define TPM_STS(t, l) ((uint8_t*)(((uint8_t*)t->pages[l]) + 0x0018))
- #define TPM_DATA_FIFO(t, l) (((uint8_t*)t->pages[l]) + 0x0024)
- #define TPM_DID_VID(t, l) ((uint32_t*)(((uint8_t*)t->pages[l]) + 0x0F00))
- #define TPM_RID(t, l) (((uint8_t*)t->pages[l]) + 0x0F04)
- #define tpm_tis_cancel_cmd(v) tpm_tis_ready(v)

Enumerations

- enum tpm_duration {
  TPM_SHORT = 0, TPM_MEDIUM = 1, TPM_LONG = 2, TPM_UNDEFINED,
  TPM_SHORT = 0, TPM_MEDIUM = 1, TPM_LONG = 2, TPM_UNDEFINED }
- enum tis_access {
  TPM_ACCESS_VALID = 0x80, TPM_ACCESS_ACTIVE_LOCALITY = 0x20, TPM_ACCESS_RELINQUISH_LOCALITY = 0x20, TPM_ACCESS_REQUEST_PENDING = 0x04,
  TPM_ACCESS_REQUEST_USE = 0x02 }
- enum tis_status {
  TPM_STS_VALID = 0x80, TPM_STS_COMMAND_READY = 0x40, TPM_STS_DATA_AVAIL = 0x10, TPM_STS_DATA_EXPECT = 0x08,
  TPM_STS_GO = 0x20 }
- enum tis_int_flags {
  TPM_GLOBAL_INT_ENABLE = 0x80000000, TPM_INTF_BURST_COUNT_STATIC = 0x100, TPM_INTF_CMD_READY_INT = 0x080, TPM_INTF_INT_EDGE_FALLING = 0x040,
  TPM_INTF_INT_EDGE_RISING = 0x020, TPM_INTF_INT_LEVEL_LOW = 0x010, TPM_INTF_INT_LEVEL_HIGH = 0x008, TPM_INTF_LOCALITY_CHANGE_INT = 0x004,
  TPM_INTF_STS_VALID_INT = 0x002, TPM_INTF_DATA_AVAIL_INT = 0x001 }
- enum tis_defaults { TIS_MEM_BASE = 0xFED40000, TIS_MEM_LEN = 0x5000, TIS_SHORT_TIMEOUT = 750, TIS_LONG_TIMEOUT = 2000 }

Functions

- struct tpm_input_header __attribute__ ((packed))
- MINIOS_LIST_HEAD (tpm_list)
- s_time_t tpm_calc_ordinal_duration (struct tpm_chip *chip, uint32_t ordinal)
- void release_locality (struct tpm_chip *tpm, int l, int force)
- int tpm_tis_request_locality (struct tpm_chip *tpm, int l)
- int tpm_tis_recv (struct tpm_chip *tpm, uint8_t *buf, size_t count)
- int tpm_tis_send (struct tpm_chip *tpm, uint8_t *buf, size_t len)
- int tpm_tis_transmit (struct tpm_chip *chip, struct tpm_cmd *cmd, int len, const char *desc)
- void tpm_get_timeouts (struct tpm_chip *chip)
- void tpm_continue_selftest (struct tpm_chip *chip)
- ssize_t tpm_getcap (struct tpm_chip *chip, uint32_t subcap_id, cap_t *cap, const char *desc)
- struct tpm_chip * init_tpm_tis (unsigned long baseaddr, int localities, unsigned int irq)
- void shutdown_tpm_tis (struct tpm_chip *tpm)
- int tpm_tis_cmd (struct tpm_chip *tpm, uint8_t *req, size_t reqlen, uint8_t **resp, size_t *resplen)

Variables

- typedef __attribute__
- uint16_t tag
- uint32_t length
- uint32_t ordinal
- uint32_t return_code
- uint8_t deactivated
- uint8_t disableForceClear
- uint8_t physicalPresence
- uint8_t physicalPresenceLock
- uint8_t bGlobalLock
- uint8_t Major
- uint8_t Minor
- uint8_t revMajor
- uint8_t revMinor

- uint32_t a
- uint32_t b
- uint32_t c
- uint32_t d
- uint32_t tpm_short
- uint32_t tpm_medium
- uint32_t tpm_long
- uint8_t disable
- uint8_t ownership
- uint8_t readPubek
- uint8_t disableOwnerClear
- uint8_t allowMaintenance
- uint8_t physicalPresenceLifetimeLock
- uint8_t physicalPresenceHWEnable
- uint8_t physicalPresenceCMDEnable
- uint8_t CEKPUsed
- uint8_t TPMpost
- uint8_t TPMpostLock
- uint8_t FIPS
- uint8_t operator
- uint8_t enableRevokeEK
- uint8_t nvLocked
- uint8_t readSRKPub
- uint8_t tpmEstablished
- uint8_t maintenanceDone
- uint8_t disableFullDALogicInfo
- uint32_t cap
- uint32_t subcap_size
- uint32_t subcap
- uint32_t cap_size
- uint8_t algorithm [4]
- uint8_t encscheme [2]
- uint8_t sigscheme [2]
- uint32_t paramsize
- uint8_t parameters [12]
- uint32_t keysize
- uint8_t modulus [256]
- uint8_t checksum [20]
- uint8_t pcr_result [TPM_DIGEST_SIZE]
- uint32_t pcr_idx
- uint8_t hash [TPM_DIGEST_SIZE]
- tpm_cmd_header header
- tpm_cmd_params params
- const uint8_t tpm_protected_ordinal_duration [TPM_MAX_PROTECTED_ORDINAL]
- const uint8_t tpm_ordinal_duration [TPM_MAX_ORDINAL]
- struct tpm_input_header tpm_getcap_header

4.40.1 Macro Definition Documentation

4.40.1.1 #define CAP_VERSION_1_1 cpu_to_be32(0x06)

4.40.1.2 #define CAP_VERSION_1_2 cpu_to_be32(0x1A)

4.40.1.3 #define min(a, b) (((a) < (b)) ? (a) : (b))

4.40.1.4 #define TPM_ACCESS(t, l) (((uint8_t*)t->pages[l]) + 0x0000)

4.40.1.5 #define TPM_BUFSIZE 2048

4.40.1.6 #define TPM_CAP_FLAG cpu_to_be32(4)

4.40.1.7 #define TPM_CAP_FLAG_PERM cpu_to_be32(0x108)

4.40.1.8 #define TPM_CAP_FLAG_VOL cpu_to_be32(0x109)

4.40.1.9 #define TPM_CAP_PROP cpu_to_be32(5)

4.40.1.10 #define TPM_CAP_PROP_MANUFACTURER cpu_to_be32(0x103)

4.40.1.11 #define TPM_CAP_PROP_OWNER cpu_to_be32(0x111)

4.40.1.12 #define TPM_CAP_PROP_PCR cpu_to_be32(0x101)

4.40.1.13 #define TPM_CAP_PROP_TIS_DURATION cpu_to_be32(0x120)

4.40.1.14 #define TPM_CAP_PROP_TIS_TIMEOUT cpu_to_be32(0x115)

4.40.1.15 #define TPM_DATA_FIFO(t, l) (((uint8_t*)t->pages[l]) + 0x0024)

4.40.1.16 #define TPM_DID_VID(t, l) ((uint32_t*)(((uint8_t*)t->pages[l]) + 0x0F00))

4.40.1.17 #define TPM_DIGEST_SIZE 20

4.40.1.18 #define TPM_DIGEST_SIZE 20

4.40.1.19 #define TPM_ERROR_SIZE 10

4.40.1.20 #define TPM_HEADER_SIZE 10

4.40.1.21 #define TPM_INT_ENABLE(t, l) ((uint32_t*)(((uint8_t*)t->pages[l]) + 0x0008))

4.40.1.22 #define TPM_INT_STATUS(t, l) (((uint8_t*)t->pages[l]) + 0x0010)

4.40.1.23 #define TPM_INT_VECTOR(t, l) (((uint8_t*)t->pages[l]) + 0x000C)

4.40.1.24 #define TPM_INTERNAL_RESULT_SIZE 200

4.40.1.25 #define TPM_INTF_CAPS(t, l) ((uint32_t*)(((uint8_t*)t->pages[l]) + 0x0014))

4.40.1.26 #define TPM_MAX_ORDINAL 243

4.40.1.27 #define TPM_MAX_PROTECTED_ORDINAL 12

4.40.1.28 #define TPM_ORD_GET_CAP cpu_to_be32(101)

4.40.1.29 #define TPM_PROTECTED_ORDINAL_MASK 0xFF

4.40.1.30 #define TPM_RET_CODE_IDX 6

4.40.1.31 #define TPM_RID(*t*, *l*) (((uint8_t∗)t->pages[l]) + 0x0F04)

4.40.1.32 #define TPM_STS(*t*, *l*) ((uint8_t∗)(((uint8_t∗)t->pages[l]) + 0x0018))

4.40.1.33 #define TPM_TAG_RQU_COMMAND cpu_to_be16(193)

4.40.1.34 #define TPM_TIMEOUT 5

4.40.1.35 #define tpm_tis_cancel_cmd(*v*) tpm_tis_ready(v)

4.40.2 Enumeration Type Documentation

4.40.2.1 enum tis_access

Enumerator

TPM_ACCESS_VALID
TPM_ACCESS_ACTIVE_LOCALITY
TPM_ACCESS_RELINQUISH_LOCALITY
TPM_ACCESS_REQUEST_PENDING
TPM_ACCESS_REQUEST_USE

4.40.2.2 enum tis_defaults

Enumerator

TIS_MEM_BASE
TIS_MEM_LEN
TIS_SHORT_TIMEOUT
TIS_LONG_TIMEOUT

4.40.2.3 enum tis_int_flags

Enumerator

TPM_GLOBAL_INT_ENABLE
TPM_INTF_BURST_COUNT_STATIC
TPM_INTF_CMD_READY_INT
TPM_INTF_INT_EDGE_FALLING
TPM_INTF_INT_EDGE_RISING
TPM_INTF_INT_LEVEL_LOW
TPM_INTF_INT_LEVEL_HIGH
TPM_INTF_LOCALITY_CHANGE_INT
TPM_INTF_STS_VALID_INT
TPM_INTF_DATA_AVAIL_INT

4.40.2.4 enum tis_status

Enumerator

TPM_STS_VALID

TPM_STS_COMMAND_READY

TPM_STS_DATA_AVAIL

TPM_STS_DATA_EXPECT

TPM_STS_GO

4.40.2.5 enum tpm_duration

Enumerator

TPM_SHORT

TPM_MEDIUM

TPM_LONG

TPM_UNDEFINED

TPM_SHORT

TPM_MEDIUM

TPM_LONG

TPM_UNDEFINED

4.40.3 Function Documentation

4.40.3.1 struct tpm_input_header __attribute__ ((packed))

4.40.3.2 struct tpm_chip* init_tpm_tis (unsigned long *baseaddr,* int *localities,* unsigned int *irq*)

4.40.3.3 MINIOS_LIST_HEAD (tpm_list)

4.40.3.4 void release_locality (struct tpm_chip * *tpm,* int *l,* int *force*)

4.40.3.5 void shutdown_tpm_tis (struct tpm_chip * *tpm*)

4.40.3.6 s_time_t tpm_calc_ordinal_duration (struct tpm_chip * *chip,* uint32_t *ordinal*)

Returns max number of nsecs to wait

4.40.3.7 void tpm_continue_selftest (struct tpm_chip * *chip*)

4.40.3.8 void tpm_get_timeouts (struct tpm_chip * *chip*)

4.40.3.9 ssize_t tpm_getcap (struct tpm_chip * *chip,* uint32_t *subcap_id,* cap_t * *cap,* const char * *desc*)

4.40.3.10 int tpm_tis_cmd (struct tpm_chip * *tpm,* uint8_t * *req,* size_t *reqlen,* uint8_t ** *resp,* size_t * *resplen*)

4.40.3.11 int tpm_tis_recv (struct tpm_chip * *tpm,* uint8_t * *buf,* size_t *count*)

4.40.3.12 int tpm_tis_request_locality (struct tpm_chip * *tpm,* int *l*)

4.40.3.13 int tpm_tis_send (struct tpm_chip * *tpm,* uint8_t * *buf,* size_t *len*)

4.40.3.14 int tpm_tis_transmit (struct tpm_chip * *chip,* struct tpm_cmd * *cmd,* int *len,* const char * *desc*)

4.40.4 Variable Documentation

4.40.4.1 enum tpm_duration __attribute__

4.40.4.2 uint32_t a

4.40.4.3 uint8_t algorithm[4]

4.40.4.4 uint8_t allowMaintenance

4.40.4.5 uint32_t b

4.40.4.6 uint8_t bGlobalLock

4.40.4.7 uint32_t c

4.40.4.8 cap_t cap

4.40.4.9 uint32_t cap_size

4.40.4.10 uint8_t CEKPUsed

4.40.4.11 uint8_t checksum[20]

4.40.4.12 uint32_t d

4.40.4.13 uint8_t deactivated

4.40.4.14 uint8_t disable

4.40.4.15 uint8_t disableForceClear

4.40.4.16 uint8_t disableFullDALogicInfo

4.40.4.17 uint8_t disableOwnerClear

4.40.4.18 uint8_t enableRevokeEK

4.40.4.19 uint8_t encscheme[2]

4.40.4.20 uint8_t FIPS

4.40.4.21 uint8_t hash[TPM_DIGEST_SIZE]

4.40.4.22 tpm_cmd_header header

4.40.4.23 uint32_t keysize

4.40.4.24 uint32_t length

4.40.4.25 uint8_t maintenanceDone

4.40.4.26 uint8_t Major

4.40.4.27 uint8_t Minor

4.40.4.28 uint8_t modulus[256]

4.40.4.29 uint8_t nvLocked

4.40.4.30 uint8_t operator

4.40.4.31 uint32_t ordinal

4.40.4.32 uint8_t ownership

4.40.4.33 uint8_t parameters[12]

4.40.4.34 tpm_cmd_params params

4.40.4.35 uint32_t paramsize

4.40.4.36 uint32_t pcr_idx

4.40.4.37 uint8_t pcr_result[TPM_DIGEST_SIZE]

4.40.4.38 uint8_t physicalPresence

4.40.4.39 uint8_t physicalPresenceCMDEnable

4.40.4.40 uint8_t physicalPresenceHWEnable

4.40.4.41 uint8_t physicalPresenceLifetimeLock

4.40.4.42 uint8_t physicalPresenceLock

4.40.4.43 uint8_t readPubek

4.40.4.44 uint8_t readSRKPub

4.40.4.45 uint32_t return_code

4.40.4.46 uint8_t revMajor

4.40.4.47 uint8_t revMinor

4.40.4.48 uint8_t sigscheme[2]

4.40.4.49 uint32_t subcap

4.40.4.50 uint32_t subcap_size

4.40.4.51 uint16_t tag

4.40.4.52 struct tpm_input_header tpm_getcap_header

Initial value:

```
= {
        .tag = TPM_TAG_RQU_COMMAND,
        .length = cpu_to_be32(22),
        .ordinal = TPM_ORD_GET_CAP
```

```
}
```

4.40.4.53 uint32_t tpm_long

4.40.4.54 uint32_t tpm_medium

4.40.4.55 const uint8_t tpm_ordinal_duration

4.40.4.56 const uint8_t tpm_protected_ordinal_duration

Initial value:

```
= {
   TPM_UNDEFINED,
   TPM_UNDEFINED,
   TPM_UNDEFINED,
   TPM_UNDEFINED,
   TPM_UNDEFINED,
   TPM_UNDEFINED,
   TPM_UNDEFINED,
   TPM_UNDEFINED,
   TPM_UNDEFINED,
   TPM_UNDEFINED,
   TPM_SHORT,
   TPM_SHORT,
}
```

4.40.4.57 uint32_t tpm_short

4.40.4.58 uint8_t tpmEstablished

4.40.4.59 uint8_t TPMpost

4.40.4.60 uint8_t TPMpostLock

4.41 xen/extras/mini-os/tpmback.c File Reference

```
#include <mini-os/os.h>
#include <mini-os/xenbus.h>
#include <mini-os/events.h>
#include <errno.h>
#include <mini-os/gnttab.h>
#include <xen/io/xenbus.h>
#include <xen/io/tpmif.h>
#include <xen/io/protocols.h>
#include <mini-os/xmalloc.h>
#include <time.h>
#include <mini-os/tpmback.h>
#include <mini-os/lib.h>
#include <fcntl.h>
#include <mini-os/mm.h>
#include <mini-os/posix/sys/mman.h>
#include <mini-os/semaphore.h>
#include <mini-os/wait.h>
```

Data Structures

- struct tpmif

- struct tpmback_dev

Macros

- #define strtoul simple_strtoul
- #define TPMBACK_DEBUG(fmt,...)
- #define TPMBACK_ERR(fmt,...) printk("Tpmback:Error " fmt, ##__VA_ARGS__)
- #define TPMBACK_LOG(fmt,...) printk("Tpmback:Info " fmt, ##__VA_ARGS__)
- #define min(a, b) (((a) < (b)) ? (a) : (b))
- #define DEF_ARRAY_SIZE 1
- #define TPMIF_CLOSED 1
- #define TPMIF_REQ_READY 2

Typedefs

- typedef struct tpmif tpmif_t
- typedef struct tpmback_dev tpmback_dev_t

Enumerations

- enum { EV_NONE, EV_NEWFE, EV_STCHNG }

Functions

- void tpmif_req_ready (tpmif_t *tpmif)
- void tpmdev_check_req (void)
- void tpmif_req_finished (tpmif_t *tpmif)
- int __get_tpmif_index (int st, int n, domid_t domid, unsigned int handle)
- int get_tpmif_index (domid_t domid, unsigned int handle)
- tpmif_t * get_tpmif (domid_t domid, unsigned int handle)
- int remove_tpmif (tpmif_t *tpmif)
- int insert_tpmif (tpmif_t *tpmif)
- int tpmif_change_state (tpmif_t *tpmif, enum xenbus_state state)
- tpmif_t * __init_tpmif (domid_t domid, unsigned int handle)
- void __free_tpmif (tpmif_t *tpmif)
- tpmif_t * new_tpmif (domid_t domid, unsigned int handle)
- void free_tpmif (tpmif_t *tpmif)
- void tpmback_handler (evtchn_port_t port, struct pt_regs *regs, void *data)
- int connect_fe (tpmif_t *tpmif)
- void handle_backend_event (char *evstr)
- char * tpmback_get_uuid (domid_t domid, unsigned int handle)
- void tpmback_set_open_callback (void(*cb)(domid_t, unsigned int))
- void tpmback_set_close_callback (void(*cb)(domid_t, unsigned int))
- void tpmback_set_suspend_callback (void(*cb)(domid_t, unsigned int))
- void tpmback_set_resume_callback (void(*cb)(domid_t, unsigned int))
- void event_thread (void *p)
- void init_tpmback (char **exclusive_uuids)
- void shutdown_tpmback (void)
- void init_tpmcmd (tpmcmd_t *tpmcmd, domid_t domid, unsigned int handle, char *uuid)
- tpmcmd_t * get_request (tpmif_t *tpmif)
- void send_response (tpmcmd_t *cmd, tpmif_t *tpmif)
- tpmcmd_t * tpmback_req_any (void)
- tpmcmd_t * tpmback_req (domid_t domid, unsigned int handle)

- void tpmback_resp (tpmcmd_t *tpmcmd)
- int tpmback_wait_for_frontend_connect (domid_t *domid, unsigned int *handle)
- int tpmback_num_frontends (void)

Variables

- enum { ... } tpm_ev_enum
- struct wait_queue_head waitq
- int globalinit = 0

4.41.1 Macro Definition Documentation

4.41.1.1 #define DEF_ARRAY_SIZE 1

4.41.1.2 #define min(a, b) (((a) < (b)) ? (a) : (b))

4.41.1.3 #define strtoul simple_strtoul

4.41.1.4 #define TPMBACK_DEBUG(*fmt, ...*)

4.41.1.5 #define TPMBACK_ERR(*fmt, ...*) printk("Tpmback:Error " fmt, ##__VA_ARGS__)

4.41.1.6 #define TPMBACK_LOG(*fmt, ...*) printk("Tpmback:Info " fmt, ##__VA_ARGS__)

4.41.1.7 #define TPMIF_CLOSED 1

4.41.1.8 #define TPMIF_REQ_READY 2

4.41.2 Typedef Documentation

4.41.2.1 typedef struct tpmback_dev tpmback_dev_t

4.41.2.2 typedef struct tpmif tpmif_t

4.41.3 Enumeration Type Documentation

4.41.3.1 anonymous enum

Enumerator

EV_NONE
EV_NEWFE
EV_STCHNG

4.41.4 Function Documentation

4.41.4.1 void __free_tpmif (tpmif_t * *tpmif*)

4.41.4.2 int __get_tpmif_index (int *st*, int *n*, domid_t *domid*, unsigned int *handle*)

4.41.4.3 tpmif_t* __init_tpmif (domid_t *domid*, unsigned int *handle*) [inline]

4.41.4.4 int connect_fe (tpmif_t * *tpmif*)

Connect to frontend

4.41.4.5 void event_thread (void * *p*)

4.41.4.6 void free_tpmif (tpmif_t * *tpmif*)

Removes tpmif from dev->tpmlist and frees it's memory usage

4.41.4.7 tpmcmd_t* get_request (tpmif_t * *tpmif*)

4.41.4.8 tpmif_t* get_tpmif (domid_t *domid,* unsigned int *handle*)

Returns the tpmif domid/handle or NULL if none exists

4.41.4.9 int get_tpmif_index (domid_t *domid,* unsigned int *handle*)

Returns the array index of the tpmif domid/handle. Returns -1 if no such tpmif exists

4.41.4.10 void handle_backend_event (char * *evstr*)

4.41.4.11 void init_tpmback (char ** *exclusive_uuids*)

Initialize the tpm backend driver - This is NULL terminated list of vtpm uuid strings. If this list is non-empty, then only frontend domains with vtpm uuid's matching entries in this list will be allowed to connect. Other connections will be immediatly closed. Set this argument to NULL to allow any vtpm to connect.

4.41.4.12 void init_tpmcmd (tpmcmd_t * *tpmcmd,* domid_t *domid,* unsigned int *handle,* char * *uuid*) [inline]

4.41.4.13 int insert_tpmif (tpmif_t * *tpmif*)

Insert tpmif into dev->tpmlist. Returns 0 on success and non zero on error. It is an error to insert a tpmif with the same domid and handle number as something already in the list

4.41.4.14 tpmif_t* new_tpmif (domid_t *domid,* unsigned int *handle*)

Creates a new tpm interface, adds it to the sorted array and returns it. returns NULL on error If the tpm interface already exists, it is returned

4.41.4.15 int remove_tpmif (tpmif_t * *tpmif*)

Remove the given tpmif. Returns 0 if it was removed, -1 if it was not removed

4.41.4.16 void send_response (tpmcmd_t * *cmd,* tpmif_t * *tpmif*)

4.41.4.17 void shutdown_tpmback (void)

Shutdown tpm backend driver

4.41.4.18 char* tpmback_get_uuid (domid_t *domid,* unsigned int *handle*)

Returns the uuid of the specified frontend, NULL on error

4.41.4.19 void tpmback_handler (evtchn_port_t *port,* struct pt_regs * *regs,* void * *data*)

Event channel handler

4.41.4.20 int tpmback_num_frontends (void)

returns the number of frontends connected

4.41.4.21 tpmcmd_t* tpmback_req (domid_t *domid,* unsigned int *handle*)

Blocks until a tpm command from the frontend at domid/handle is sent. Returns NULL if domid/handle is not connected, tpmback is shutdown or shutting down, or if there is an error

4.41.4.22 tpmcmd_t* tpmback_req_any (void)

Blocks until a tpm command is sent from any front end. Returns a pointer to the tpm command to handle. Do not try to free this pointer or the req buffer This function will return NULL if the tpm backend driver is shutdown or any other error occurs

4.41.4.23 void tpmback_resp (tpmcmd_t * *tpmcmd*)

Send the response to the tpm command back to the frontend This function will free the tpmcmd object, but you must free the resp buffer yourself

4.41.4.24 void tpmback_set_close_callback (void(*)(domid_t, unsigned int) *cb*)

Specify a function to call when a tpm device disconnects

4.41.4.25 void tpmback_set_open_callback (void(*)(domid_t, unsigned int) *cb*)

Specify a function to call when a new tpm device connects

4.41.4.26 void tpmback_set_resume_callback (void(*)(domid_t, unsigned int) *cb*)

4.41.4.27 void tpmback_set_suspend_callback (void(*)(domid_t, unsigned int) *cb*)

4.41.4.28 int tpmback_wait_for_frontend_connect (domid_t * *domid,* unsigned int * *handle*)

Waits for the first frontend to connect and then sets domid and handle appropriately. If one or more frontends are already connected, this will set domid and handle to one of them arbitrarily. The main use for this function is to wait until a single frontend connection has occured. returns 0 on success, non-zero on failure

4.41.4.29 void tpmdev_check_req (void) [inline]

4.41.4.30 int tpmif_change_state (tpmif_t * *tpmif,* enum xenbus_state *state*)

Attempts to change the backend state in xenstore returns 0 on success and non-zero on error

4.41.4.31 void tpmif_req_finished (tpmif_t * *tpmif*) [inline]

4.41.4.32 void tpmif_req_ready (tpmif_t * *tpmif*) [inline]

4.41.5 Variable Documentation

4.41.5.1 int globalinit = 0

4.41.5.2 enum { ... } tpm_ev_enum

4.41.5.3 struct wait_queue_head waitq

4.42 xen/extras/mini-os/tpmfront.c File Reference

```
#include <mini-os/os.h>
#include <mini-os/xenbus.h>
#include <mini-os/xmalloc.h>
#include <mini-os/events.h>
#include <mini-os/wait.h>
#include <mini-os/gnttab.h>
#include <xen/io/xenbus.h>
#include <xen/io/tpmif.h>
#include <mini-os/tpmfront.h>
#include <fcntl.h>
```

Macros

- #define TPMFRONT_DEBUG(fmt,...)
- #define TPMFRONT_ERR(fmt,...) printk("Tpmfront:Error " fmt, ##__VA_ARGS__)
- #define TPMFRONT_LOG(fmt,...) printk("Tpmfront:Info " fmt, ##__VA_ARGS__)
- #define min(a, b) (((a) < (b)) ? (a) : (b))

Functions

- void tpmfront_handler (evtchn_port_t port, struct pt_regs *regs, void *data)
- struct tpmfront_dev * init_tpmfront (const char *_nodename)
- void shutdown_tpmfront (struct tpmfront_dev *dev)
- int tpmfront_send (struct tpmfront_dev *dev, const uint8_t *msg, size_t length)
- int tpmfront_recv (struct tpmfront_dev *dev, uint8_t **msg, size_t *length)
- int tpmfront_cmd (struct tpmfront_dev *dev, uint8_t *req, size_t reqlen, uint8_t **resp, size_t *resplen)

4.42.1 Macro Definition Documentation

4.42.1.1 #define min(a, b) (((a) < (b)) ? (a) : (b))

4.42.1.2 #define TPMFRONT_DEBUG(*fmt, ...*)

4.42.1.3 #define TPMFRONT_ERR(*fmt, ...*) printk("Tpmfront:Error " fmt, ##__VA_ARGS__)

4.42.1.4 #define TPMFRONT_LOG(*fmt, ...*) printk("Tpmfront:Info " fmt, ##__VA_ARGS__)

4.42.2 Function Documentation

4.42.2.1 struct tpmfront_dev* init_tpmfront (const char * _nodename)

4.42.2.2 void shutdown_tpmfront (struct tpmfront_dev * dev)

4.42.2.3 int tpmfront_cmd (struct tpmfront_dev * dev, uint8_t * req, size_t reqlen, uint8_t ** resp, size_t * resplen)

4.42.2.4 void tpmfront_handler (evtchn_port_t port, struct pt_regs * regs, void * data)

4.42.2.5 int tpmfront_recv (struct tpmfront_dev * dev, uint8_t ** msg, size_t * length)

4.42.2.6 int tpmfront_send (struct tpmfront_dev * dev, const uint8_t * msg, size_t length)

4.43 xen/extras/mini-os/uuid.c File Reference

```
#include <inttypes.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <uuid.h>
```

Functions

- void uuid_create (uuid_t *uuid, uint32_t *status)
- void uuid_to_string (uuid_t *uuid, char **str, uint32_t *status)
- int uuid_is_nil (uuid_t *uuid, unsigned int *status)

4.43.1 Function Documentation

4.43.1.1 void uuid_create (uuid_t * uuid, uint32_t * status)

just fill the struct with random values for now

4.43.1.2 int uuid_is_nil (uuid_t * uuid, unsigned int * status)

4.43.1.3 void uuid_to_string (uuid_t * uuid, char ** str, uint32_t * status)

convert the struct to a printable string

4.44 xen/extras/mini-os/vfs.c File Reference

4.45 xen/stubdom/domt/integrated/vfs.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <stdint.h>
#include <unistd.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <errno.h>
#include <mini-os/os.h>
#include <mini-os/xenbus.h>
#include <mini-os/events.h>
#include <mini-os/gnttab.h>
#include <vfs.h>
#include <vfs_hooks.h>
```

Data Structures

- struct fe_gref_map

Macros

- #define INVALID_GREF(ref) (ref >= NR_GRANT_ENTRIES || ref < NR_RESERVED_ENTRIES)
- #define NR_RESERVED_ENTRIES 8
- #define NR_GRANT_FRAMES 4
- #define NR_GRANT_ENTRIES (NR_GRANT_FRAMES * PAGE_SIZE / sizeof(grant_entry_t))
- #define MIN(a, b) (((a) < (b)) ? (a) : (b))

Functions

- int vfs_init (void)
- int vfs_command (struct vfs_cmd *cmd)
- int vfs_open (const char *pathname, int flags, mode_t mode)
- int vfs_close (int fd)
- int vfs_stat (const char *path, struct stat *buf)
- off_t vfs_lseek (int fd, off_t offset, int whence)
- ssize_t vfs_read (int fd, void *buf, size_t nbyte)
- ssize_t vfs_write (int fd, const void *buf, size_t nbyte)
- void vfs_major_minor (const char *physpath, int *major, int *minor)
- void * vfs_mmap (void *addr, size_t length, int prot, int flags, int fd, off_t offset)

Variables

- struct fe_gref_map __attribute__

4.45.1 Macro Definition Documentation

4.45.1.1 #define INVALID_GREF(*ref*) (ref >= NR_GRANT_ENTRIES || ref < NR_RESERVED_ENTRIES)

4.45.1.2 #define MIN(a, b) (((a) < (b)) ? (a) : (b))

4.45.1.3 #define NR_GRANT_ENTRIES (NR_GRANT_FRAMES ∗ PAGE_SIZE / sizeof(grant_entry_t))

4.45.1.4 #define NR_GRANT_FRAMES 4

4.45.1.5 #define NR_RESERVED_ENTRIES 8

4.45.2 Function Documentation

4.45.2.1 int vfs_close (int *fd*)

4.45.2.2 int vfs_command (struct vfs_cmd ∗ *cmd*)

4.45.2.3 int vfs_init (void)

The init command has become deprecated. Maybe it can be useful to test if VFS is alive at boottime. It might also play a role in detection of xenvfsd restart/crashing; since then we need to invalidate all vfs file descriptors. Not a priority task, however.

4.45.2.4 off_t vfs_lseek (int *fd,* off_t *offset,* int *whence*)

4.45.2.5 void vfs_major_minor (const char ∗ *physpath,* int ∗ *major,* int ∗ *minor*)

combines the major and minor calls into one

4.45.2.6 void∗ vfs_mmap (void ∗ *addr,* size_t *length,* int *prot,* int *flags,* int *fd,* off_t *offset*)

For now, mmap is synthetic. it is a TODO to build a native solution via grant tables. We probably want to share a writable gref_map with Dom0 then. Until then, this function does not sync back changes. Since this call is only used in the domainbuilder for reading in the images, priority for this fix is low currently.

4.45.2.7 int vfs_open (const char ∗ *pathname,* int *flags,* mode_t *mode*)

NOTE: munmap, close, are not hooked since minios default implementation suffices. NOTE: newlib implements for us: fread() fwrite() fopen() fclose().

4.45.2.8 ssize_t vfs_read (int *fd,* void ∗ *buf,* size_t *nbyte*)

4.45.2.9 int vfs_stat (const char ∗ *path,* struct stat ∗ *buf*)

4.45.2.10 ssize_t vfs_write (int *fd,* const void ∗ *buf,* size_t *nbyte*)

4.45.3 Variable Documentation

4.45.3.1 struct fe_gref_map __attribute__

4.46 xen/stubdom/domc/blkback.c File Reference

```
#include <stdint.h>
#include <mini-os/os.h>
#include <mini-os/xenbus.h>
#include <mini-os/events.h>
#include <errno.h>
#include <xen/io/blkif.h>
#include <xen/io/protocols.h>
#include <mini-os/gnttab.h>
#include <mini-os/xmalloc.h>
#include <time.h>
#include <mini-os/blkfront.h>
#include <mini-os/lib.h>
#include <fcntl.h>
#include <unistd.h>
#include "blkback.h"
#include "main.h"
#include "crypt.h"
```

Macros

- #define SECTOR_SIZE 512
- #define SECTS_PER_PAGE PAGE_SIZE / SECTOR_SIZE
- #define SECTS_IN_SEG(req, i)
- #define BLK_RING_SIZE __RING_SIZE((struct blkif_sring ∗)0, PAGE_SIZE)
- #define MAX_GNTMAP BLKIF_MAX_SEGMENTS_PER_REQUEST ∗ BLK_RING_SIZE + 1

Enumerations

- enum bridging { OLDNEW, NEWOLD }

Functions

- void blkback_cb_back (evtchn_port_t port, struct pt_regs ∗regs, void ∗data)
- struct blkback_dev ∗ derive_blkback (char ∗_nodename, domc_info_t ∗domc_info, struct blkfront_dev ∗f_dev, struct blkfront_info ∗f_dev_info)
- int init_blkback (struct blkback_dev ∗b_dev)
- void shutdown_blkback (struct blkback_dev ∗b_dev)
- void blkback_monitor (struct blkback_dev ∗b_dev)
- struct blkback_aiocb ∗ blkback_build_fwd (struct blkback_dev ∗b_dev, blkif_request_t ∗req)
- int blkback_aio_poll (struct blkback_dev ∗b_dev)

4.46.1 Macro Definition Documentation

4.46.1.1 #define BLK_RING_SIZE __RING_SIZE((struct blkif_sring ∗)0, PAGE_SIZE)

4.46.1.2 #define MAX_GNTMAP BLKIF_MAX_SEGMENTS_PER_REQUEST ∗ BLK_RING_SIZE + 1

the one slot extra is for the sring page

4.46.1.3 #define SECTOR_SIZE 512

The assumptions about IRQs and pre-emptiveness for the blkfront hold here as well. No wait queues are used, since we cannot trap multiple threads in our functions here. Our design assumes a single thread which never leaves _monitor().

4.46.1.4 #define SECTS_IN_SEG(*req, i*)

Value:

```
(1+(req->seg[i].last_sect - \
                                req->seg[i].first_sect))
```

4.46.1.5 #define SECTS_PER_PAGE PAGE_SIZE / SECTOR_SIZE

4.46.2 Enumeration Type Documentation

4.46.2.1 enum bridging

Enumerator

OLDNEW

NEWOLD

4.46.3 Function Documentation

4.46.3.1 int blkback_aio_poll (struct blkback_dev ∗ *b_dev*)

4.46.3.2 struct blkback_aiocb∗ blkback_build_fwd (struct blkback_dev ∗ *b_dev,* blkif_request_t ∗ *req*)

Builds an aio to forward from blkback to blkfront

4.46.3.3 void blkback_cb_back (evtchn_port_t *port,* struct pt_regs ∗ *regs,* void ∗ *data*)

4.46.3.4 void blkback_monitor (struct blkback_dev ∗ *b_dev*)

We placed the blkfront_sync() calls in a separate thread and not after the blkback_poll() here. If we'd wait with writing new requests until blkfront_sync() returns, we would not be making optimal use of the ring buffer with Dom0.

4.46.3.5 struct blkback_dev∗ derive_blkback (char ∗ *_nodename,* domc_info_t ∗ *domc_info,* struct blkfront_dev ∗ *f_dev,* struct blkfront_info ∗ *f_dev_info*)

4.46.3.6 int init_blkback (struct blkback_dev ∗ *b_dev*)

4.46.3.7 void shutdown_blkback (struct blkback_dev ∗ *b_dev*)

4.47 xen/stubdom/domc/blkback.h File Reference

```
#include <mini-os/blkfront.h>
#include "main.h"
```

Data Structures

- struct blkback_info
- struct blkback_dev
- struct blkback_aiocb

Functions

- struct blkback_dev ∗ derive_blkback (char ∗_nodename, domc_info_t ∗domc_info, struct blkfront_dev ∗f_dev, struct blkfront_info ∗f_dev_info)
- int init_blkback (struct blkback_dev ∗b_dev)
- int blkback_aio_poll (struct blkback_dev ∗b_dev)
- void blkback_monitor (struct blkback_dev ∗b_dev)
- void shutdown_blkback (struct blkback_dev ∗b_dev)

4.47.1 Function Documentation

4.47.1.1 int blkback_aio_poll (struct blkback_dev ∗ *b_dev*)

4.47.1.2 void blkback_monitor (struct blkback_dev ∗ *b_dev*)

We placed the blkfront_sync() calls in a separate thread and not after the blkback_poll() here. If we'd wait with writing new requests until blkfront_sync() returns, we would not be making optimal use of the ring buffer with Dom0.

4.47.1.3 struct blkback_dev∗ derive_blkback (char ∗ *_nodename,* domc_info_t ∗ *domc_info,* struct blkfront_dev ∗ *f_dev,* struct blkfront_info ∗ *f_dev_info*)

4.47.1.4 int init_blkback (struct blkback_dev ∗ *b_dev*)

4.47.1.5 void shutdown_blkback (struct blkback_dev ∗ *b_dev*)

4.48 xen/stubdom/domc/crypt.c File Reference

```
#include <string.h>
#include "rijndael-api-fst.h"
#include "sha2.h"
#include "main.h"
#include "crypt.h"
```

Macros

- #define SECTOR_SIZE 512

Enumerations

- enum crypt_dir { CRYPT_EN, CRYPT_DEC }

Functions

- int startup_crypt (char *key)
- int encrypt_sectors (unsigned int sector, BYTE *in, BYTE *out, size_t len)
- int decrypt_sectors (unsigned int sector, BYTE *in, BYTE *out, size_t len)
- void shutdown_crypt (void)
- void test_crypt (void)

Variables

- keyInstance keyInst_enc
- keyInstance keyInst_dec
- keyInstance keyInst_iv
- cipherInstance cipherInst_iv

4.48.1 Macro Definition Documentation

4.48.1.1 #define SECTOR_SIZE 512

4.48.2 Enumeration Type Documentation

4.48.2.1 enum crypt_dir

Enumerator

CRYPT_EN

CRYPT_DEC

4.48.3 Function Documentation

4.48.3.1 int decrypt_sectors (unsigned int *sector*, BYTE * *in*, BYTE * *out*, size_t *len*)

4.48.3.2 int encrypt_sectors (unsigned int *sector*, BYTE * *in*, BYTE * *out*, size_t *len*)

4.48.3.3 void shutdown_crypt (void)

4.48.3.4 int startup_crypt (char * *key*)

4.48.3.5 void test_crypt (void)

4.48.4 Variable Documentation

4.48.4.1 cipherInstance cipherInst_iv

4.48.4.2 keyInstance keyInst_dec

4.48.4.3 keyInstance keyInst_enc

4.48.4.4 keyInstance keyInst_iv

4.49 xen/stubdom/domc/crypt.h File Reference

```
#include "rijndael-api-fst.h"
```

Functions

- int startup_crypt (char *key)
- int encrypt_sectors (unsigned int sector, BYTE *in, BYTE *out, size_t len)
- int decrypt_sectors (unsigned int sector, BYTE *in, BYTE *out, size_t len)
- void shutdown_crypt (void)
- void test_crypt (void)

4.49.1 Function Documentation

4.49.1.1 int decrypt_sectors (unsigned int *sector*, BYTE * *in*, BYTE * *out*, size_t *len*)

4.49.1.2 int encrypt_sectors (unsigned int *sector*, BYTE * *in*, BYTE * *out*, size_t *len*)

4.49.1.3 void shutdown_crypt (void)

4.49.1.4 int startup_crypt (char * *key*)

4.49.1.5 void test_crypt (void)

4.50 xen/stubdom/domc/crypto/rijndael-alg-fst.c File Reference

```
#include <assert.h>
#include <stdlib.h>
#include "rijndael-alg-fst.h"
```

Macros

- #define FULL_UNROLL
- #define SWAP(x) (_lrotl(x, 8) & 0x00ff00ff | _lrotr(x, 8) & 0xff00ff00)
- #define GETU32(pt) (((u32)(pt)[0] << 24) ^ ((u32)(pt)[1] << 16) ^ ((u32)(pt)[2] << 8) ^ ((u32)(pt)[3]))
- #define PUTU32(ct, st) { (ct)[0] = (u8)((st) >> 24); (ct)[1] = (u8)((st) >> 16); (ct)[2] = (u8)((st) >> 8); (ct)[3] = (u8)(st); }

Functions

- int rijndaelKeySetupEnc (u32 rk[], const u8 cipherKey[], int keyBits)
- int rijndaelKeySetupDec (u32 rk[], const u8 cipherKey[], int keyBits)
- void rijndaelEncrypt (const u32 rk[], int Nr, const u8 pt[16], u8 ct[16])
- void rijndaelDecrypt (const u32 rk[], int Nr, const u8 ct[16], u8 pt[16])

4.50.1 Macro Definition Documentation

4.50.1.1 #define FULL_UNROLL

rijndael-alg-fst.c

Version

3.0 (December 2000)

Optimised ANSI C code for the Rijndael cipher (now AES)

Author

Vincent Rijmen vincent.rijmen@esat.kuleuven.ac.be
Antoon Bosselaers antoon.bosselaers@esat.kuleuven.ac.be
Paulo Barreto paulo.barreto@terra.com.br

4.50.1.2 #define GETU32(*pt*) (((u32)(pt)[0] << 24) ^ ((u32)(pt)[1] << 16) ^ ((u32)(pt)[2] << 8) ^ ((u32)(pt)[3]))

4.50.1.3 #define PUTU32(*ct*, *st*) { (ct)[0] = (u8)((st) >> 24); (ct)[1] = (u8)((st) >> 16); (ct)[2] = (u8)((st) >> 8); (ct)[3] = (u8)(st); }

4.50.1.4 #define SWAP(*x*) (_lrotl(x, 8) & 0x00ff00ff | _lrotr(x, 8) & 0xff00ff00)

4.50.2 Function Documentation

4.50.2.1 void rijndaelDecrypt (const u32 *rk[]*, int *Nr*, const u8 *ct[16]*, u8 *pt[16]*)

4.50.2.2 void rijndaelEncrypt (const u32 *rk[]*, int *Nr*, const u8 *pt[16]*, u8 *ct[16]*)

4.50.2.3 int rijndaelKeySetupDec (u32 *rk[]*, const u8 *cipherKey[]*, int *keyBits*)

Expand the cipher key into the decryption key schedule.

Returns

the number of rounds for the given cipher key size.

4.50.2.4 int rijndaelKeySetupEnc (u32 *rk[]*, const u8 *cipherKey[]*, int *keyBits*)

Expand the cipher key into the encryption key schedule.

Returns

the number of rounds for the given cipher key size.

4.51 xen/stubdom/domc/crypto/rijndael-alg-fst.h File Reference

Macros

- #define MAXKC (256/32)
- #define MAXKB (256/8)
- #define MAXNR 14

Typedefs

- typedef unsigned char u8
- typedef unsigned short u16
- typedef unsigned int u32

Functions

- int rijndaelKeySetupEnc (u32 rk[], const u8 cipherKey[], int keyBits)
- int rijndaelKeySetupDec (u32 rk[], const u8 cipherKey[], int keyBits)
- void rijndaelEncrypt (const u32 rk[], int Nr, const u8 pt[16], u8 ct[16])
- void rijndaelDecrypt (const u32 rk[], int Nr, const u8 ct[16], u8 pt[16])

4.51.1 Macro Definition Documentation

4.51.1.1 #define MAXKB (256/8)

4.51.1.2 #define MAXKC (256/32)

rijndael-alg-fst.h

Version

3.0 (December 2000)

Optimised ANSI C code for the Rijndael cipher (now AES)

Author

Vincent Rijmen vincent.rijmen@esat.kuleuven.ac.be
Antoon Bosselaers antoon.bosselaers@esat.kuleuven.ac.be
Paulo Barreto paulo.barreto@terra.com.br

This code is hereby placed in the public domain.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

4.51.1.3 #define MAXNR 14

4.51.2 Typedef Documentation

4.51.2.1 typedef unsigned short u16

4.51.2.2 typedef unsigned int u32

4.51.2.3 typedef unsigned char u8

4.51.3 Function Documentation

4.51.3.1 void rijndaelDecrypt (const u32 *rk[],* int *Nr,* const u8 *ct[16],* u8 *pt[16]*)

4.51.3.2 void rijndaelEncrypt (const u32 *rk[],* int *Nr,* const u8 *pt[16],* u8 *ct[16]*)

4.51.3.3 int rijndaelKeySetupDec (u32 *rk[],* const u8 *cipherKey[],* int *keyBits*)

Expand the cipher key into the decryption key schedule.

Returns

the number of rounds for the given cipher key size.

4.51.3.4 int rijndaelKeySetupEnc (u32 *rk[],* const u8 *cipherKey[],* int *keyBits*)

Expand the cipher key into the encryption key schedule.

Returns

the number of rounds for the given cipher key size.

4.52 xen/stubdom/domc/crypto/rijndael-api-fst.c File Reference

```
#include <assert.h>
#include <stdlib.h>
#include <string.h>
#include "rijndael-alg-fst.h"
#include "rijndael-api-fst.h"
```

Functions

- int makeKey (keyInstance *key, BYTE direction, int keyLen, char *keyMaterial)
- int cipherInit (cipherInstance *cipher, BYTE mode, BYTE *IV)
- int blockEncrypt (cipherInstance *cipher, keyInstance *key, BYTE *input, int inputLen, BYTE *outBuffer)
- int padEncrypt (cipherInstance *cipher, keyInstance *key, BYTE *input, int inputOctets, BYTE *outBuffer)
- int blockDecrypt (cipherInstance *cipher, keyInstance *key, BYTE *input, int inputLen, BYTE *outBuffer)
- int padDecrypt (cipherInstance *cipher, keyInstance *key, BYTE *input, int inputOctets, BYTE *outBuffer)

4.52.1 Function Documentation

4.52.1.1 int blockDecrypt (cipherInstance * *cipher,* keyInstance * *key,* BYTE * *input,* int *inputLen,* BYTE * *outBuffer*)

4.52.1.2 int blockEncrypt (cipherInstance * *cipher,* keyInstance * *key,* BYTE * *input,* int *inputLen,* BYTE * *outBuffer*)

4.52.1.3 int cipherInit (cipherInstance * *cipher,* BYTE *mode,* BYTE * *IV*)

4.52.1.4 int makeKey (keyInstance * *key,* BYTE *direction,* int *keyLen,* char * *keyMaterial*)

rijndael-api-fst.c

Version

2.9 (December 2000)

Optimised ANSI C code for the Rijndael cipher (now AES)

Author

Vincent Rijmen vincent.rijmen@esat.kuleuven.ac.be
Antoon Bosselaers antoon.bosselaers@esat.kuleuven.ac.be
Paulo Barreto paulo.barreto@terra.com.br

Acknowledgements:

We are deeply indebted to the following people for their bug reports, fixes, and improvement suggestions to this implementation. Though we tried to list all contributions, we apologise in advance for any missing reference.

Andrew Bales Andrew.Bales@Honeywell.com Markus Friedl markus.friedl@informatik.uni-erlangen.de John Skodon skodonj@webquill.com

4.52.1.5 int padDecrypt (cipherInstance * *cipher*, keyInstance * *key*, BYTE * *input*, int *inputOctets*, BYTE * *outBuffer*)

4.52.1.6 int padEncrypt (cipherInstance * *cipher*, keyInstance * *key*, BYTE * *input*, int *inputOctets*, BYTE * *outBuffer*)

Encrypt data partitioned in octets, using RFC 2040-like padding.

Parameters

input	data to be encrypted (octet sequence)
inputOctets	input length in octets (not bits)
outBuffer	encrypted output data

Returns

length in octets (not bits) of the encrypted output buffer.

4.53 xen/stubdom/domc/crypto/rijndael-api-fst.h File Reference

```
#include <stdio.h>
#include "rijndael-alg-fst.h"
```

Data Structures

- struct keyInstance
- struct cipherInstance

Macros

- #define DIR_ENCRYPT 0 /∗ Are we encrpyting? ∗/
- #define DIR_DECRYPT 1 /∗ Are we decrpyting? ∗/
- #define MODE_ECB 1 /∗ Are we ciphering in ECB mode? ∗/
- #define MODE_CBC 2 /∗ Are we ciphering in CBC mode? ∗/
- #define MODE_CFB1 3 /∗ Are we ciphering in 1-bit CFB mode? ∗/
- #define TRUE 1
- #define FALSE 0
- #define BITSPERBLOCK 128 /∗ Default number of bits in a cipher block ∗/
- #define BAD_KEY_DIR -1 /∗ Key direction is invalid, e.g., unknown value ∗/
- #define BAD_KEY_MAT -2 /∗ Key material not of correct length ∗/
- #define BAD_KEY_INSTANCE -3 /∗ Key passed is not valid ∗/
- #define BAD_CIPHER_MODE -4 /∗ Params struct passed to cipherInit invalid ∗/
- #define BAD_CIPHER_STATE -5 /∗ Cipher in wrong state (e.g., not initialized) ∗/
- #define BAD_BLOCK_LENGTH -6
- #define BAD_CIPHER_INSTANCE -7
- #define BAD_DATA -8 /∗ Data contents are invalid, e.g., invalid padding ∗/
- #define BAD_OTHER -9 /∗ Unknown error ∗/
- #define MAX_KEY_SIZE 64 /∗ # of ASCII char's needed to represent a key ∗/
- #define MAX_IV_SIZE 16 /∗ # bytes needed to represent an IV ∗/

Typedefs

- typedef unsigned char BYTE

Functions

- int makeKey (keyInstance ∗key, BYTE direction, int keyLen, char ∗keyMaterial)
- int cipherInit (cipherInstance ∗cipher, BYTE mode, BYTE ∗IV)
- int blockEncrypt (cipherInstance ∗cipher, keyInstance ∗key, BYTE ∗input, int inputLen, BYTE ∗outBuffer)
- int padEncrypt (cipherInstance ∗cipher, keyInstance ∗key, BYTE ∗input, int inputOctets, BYTE ∗outBuffer)
- int blockDecrypt (cipherInstance ∗cipher, keyInstance ∗key, BYTE ∗input, int inputLen, BYTE ∗outBuffer)
- int padDecrypt (cipherInstance ∗cipher, keyInstance ∗key, BYTE ∗input, int inputOctets, BYTE ∗outBuffer)

4.53.1 Macro Definition Documentation

4.53.1.1 #define BAD_BLOCK_LENGTH -6

4.53.1.2 #define BAD_CIPHER_INSTANCE -7

4.53.1.3 #define BAD_CIPHER_MODE -4 /∗ Params struct passed to cipherInit invalid ∗/

4.53.1.4 #define BAD_CIPHER_STATE -5 /∗ Cipher in wrong state (e.g., not initialized) ∗/

4.53.1.5 #define BAD_DATA -8 /∗ Data contents are invalid, e.g., invalid padding ∗/

4.53.1.6 #define BAD_KEY_DIR -1 /∗ Key direction is invalid, e.g., unknown value ∗/

4.53.1.7 #define BAD_KEY_INSTANCE -3 /∗ Key passed is not valid ∗/

4.53.1.8 #define BAD_KEY_MAT -2 /∗ Key material not of correct length ∗/

4.53.1.9 #define BAD_OTHER -9 /∗ Unknown error ∗/

4.53.1.10 #define BITSPERBLOCK 128 /∗ Default number of bits in a cipher block ∗/

4.53.1.11 #define DIR_DECRYPT 1 /∗ Are we decrpyting? ∗/

4.53.1.12 #define DIR_ENCRYPT 0 /∗ Are we encrpyting? ∗/

rijndael-api-fst.h

Version

2.9 (December 2000)

Optimised ANSI C code for the Rijndael cipher (now AES)

Author

Vincent Rijmen vincent.rijmen@esat.kuleuven.ac.be
Antoon Bosselaers antoon.bosselaers@esat.kuleuven.ac.be
Paulo Barreto paulo.barreto@terra.com.br

This code is hereby placed in the public domain.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Acknowledgements:

We are deeply indebted to the following people for their bug reports, fixes, and improvement suggestions to this implementation. Though we tried to list all contributions, we apologise in advance for any missing reference.

Andrew Bales Andrew.Bales@Honeywell.com Markus Friedl markus.friedl@informatik.uni-erlangen.de John Skodon skodonj@webquill.com

4.53.1.13 #define FALSE 0

4.53.1.14 #define MAX_IV_SIZE 16 /∗ # bytes needed to represent an IV ∗/

4.53.1.15 #define MAX_KEY_SIZE 64 /∗ # of ASCII char's needed to represent a key ∗/

4.53.1.16 #define MODE_CBC 2 /∗ Are we ciphering in CBC mode? ∗/

4.53.1.17 #define MODE_CFB1 3 /∗ Are we ciphering in 1-bit CFB mode? ∗/

4.53.1.18 #define MODE_ECB 1 /∗ Are we ciphering in ECB mode? ∗/

4.53.1.19 #define TRUE 1

4.53.2 Typedef Documentation

4.53.2.1 typedef unsigned char BYTE

4.53.3 Function Documentation

4.53.3.1 int blockDecrypt (cipherInstance * *cipher,* keyInstance * *key,* BYTE * *input,* int *inputLen,* BYTE * *outBuffer*)

4.53.3.2 int blockEncrypt (cipherInstance * *cipher,* keyInstance * *key,* BYTE * *input,* int *inputLen,* BYTE * *outBuffer*)

4.53.3.3 int cipherInit (cipherInstance * *cipher,* BYTE *mode,* BYTE * *IV*)

4.53.3.4 int makeKey (keyInstance * *key,* BYTE *direction,* int *keyLen,* char * *keyMaterial*)

rijndael-api-fst.c

Version

2.9 (December 2000)

Optimised ANSI C code for the Rijndael cipher (now AES)

Author

Vincent Rijmen vincent.rijmen@esat.kuleuven.ac.be
Antoon Bosselaers antoon.bosselaers@esat.kuleuven.ac.be
Paulo Barreto paulo.barreto@terra.com.br

This code is hereby placed in the public domain.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Acknowledgements:

We are deeply indebted to the following people for their bug reports, fixes, and improvement suggestions to this implementation. Though we tried to list all contributions, we apologise in advance for any missing reference.

Andrew Bales Andrew.Bales@Honeywell.com Markus Friedl markus.friedl@informatik.-uni-erlangen.de John Skodon skodonj@webquill.com

4.53.3.5 int padDecrypt (cipherInstance * *cipher,* keyInstance * *key,* BYTE * *input,* int *inputOctets,* BYTE * *outBuffer*)

4.53.3.6 int padEncrypt (cipherInstance * *cipher,* keyInstance * *key,* BYTE * *input,* int *inputOctets,* BYTE * *outBuffer*)

Encrypt data partitioned in octets, using RFC 2040-like padding.

Parameters

input	data to be encrypted (octet sequence)
inputOctets	input length in octets (not bits)

outBuffer	encrypted output data

Returns

length in octets (not bits) of the encrypted output buffer.

4.54 xen/stubdom/domc/crypto/sha2.c File Reference

```
#include <string.h>
#include <assert.h>
#include "sha2.h"
```

Macros

- #define SHA256_SHORT_BLOCK_LENGTH (SHA256_BLOCK_LENGTH - 8)
- #define SHA384_SHORT_BLOCK_LENGTH (SHA384_BLOCK_LENGTH - 16)
- #define SHA512_SHORT_BLOCK_LENGTH (SHA512_BLOCK_LENGTH - 16)
- #define REVERSE32(w, x)
- #define REVERSE64(w, x)
- #define ADDINC128(w, n)
- #define SHA2_USE_MEMSET_MEMCPY 1
- #define MEMSET_BZERO(p, l) memset((p), 0, (l))
- #define MEMCPY_BCOPY(d, s, l) memcpy((d), (s), (l))
- #define R(b, x) ((x) >> (b))
- #define S32(b, x) (((x) >> (b)) | ((x) << (32 - (b))))
- #define S64(b, x) (((x) >> (b)) | ((x) << (64 - (b))))
- #define Ch(x, y, z) (((x) & (y)) ^ ((~(x)) & (z)))
- #define Maj(x, y, z) (((x) & (y)) ^ ((x) & (z)) ^ ((y) & (z)))
- #define Sigma0_256(x) (S32(2, (x)) ^ S32(13, (x)) ^ S32(22, (x)))
- #define Sigma1_256(x) (S32(6, (x)) ^ S32(11, (x)) ^ S32(25, (x)))
- #define sigma0_256(x) (S32(7, (x)) ^ S32(18, (x)) ^ R(3 , (x)))
- #define sigma1_256(x) (S32(17, (x)) ^ S32(19, (x)) ^ R(10, (x)))
- #define Sigma0_512(x) (S64(28, (x)) ^ S64(34, (x)) ^ S64(39, (x)))
- #define Sigma1_512(x) (S64(14, (x)) ^ S64(18, (x)) ^ S64(41, (x)))
- #define sigma0_512(x) (S64(1, (x)) ^ S64(8, (x)) ^ R(7, (x)))
- #define sigma1_512(x) (S64(19, (x)) ^ S64(61, (x)) ^ R(6, (x)))

Typedefs

- typedef uint8_t sha2_byte
- typedef uint32_t sha2_word32
- typedef uint64_t sha2_word64

Functions

- void SHA512_Last (SHA512_CTX *)
- void SHA256_Transform (SHA256_CTX *, const sha2_word32 *)
- void SHA512_Transform (SHA512_CTX *, const sha2_word64 *)
- void SHA256_Init (SHA256_CTX *context)
- void SHA256_Update (SHA256_CTX *context, const sha2_byte *data, size_t len)
- void SHA256_Final (sha2_byte digest[], SHA256_CTX *context)

- char * SHA256_End (SHA256_CTX *context, char buffer[])
- char * SHA256_Data (const sha2_byte *data, size_t len, char digest[SHA256_DIGEST_STRING_LENGTH])
- void SHA512_Init (SHA512_CTX *context)
- void SHA512_Update (SHA512_CTX *context, const sha2_byte *data, size_t len)
- void SHA512_Final (sha2_byte digest[], SHA512_CTX *context)
- char * SHA512_End (SHA512_CTX *context, char buffer[])
- char * SHA512_Data (const sha2_byte *data, size_t len, char digest[SHA512_DIGEST_STRING_LENGTH])
- void SHA384_Init (SHA384_CTX *context)
- void SHA384_Update (SHA384_CTX *context, const sha2_byte *data, size_t len)
- void SHA384_Final (sha2_byte digest[], SHA384_CTX *context)
- char * SHA384_End (SHA384_CTX *context, char buffer[])
- char * SHA384_Data (const sha2_byte *data, size_t len, char digest[SHA384_DIGEST_STRING_LENGTH])

4.54.1 Macro Definition Documentation

4.54.1.1 #define ADDINC128(*w, n*)

Value:

```
{ \
    (w)[0] += (sha2_word64)(n); \
    if ((w)[0] < (n)) { \
        (w)[1]++; \
    } \
}
```

4.54.1.2 #define Ch(*x, y, z*) (((x) & (y)) ^ ((~(x)) & (z)))

4.54.1.3 #define Maj(*x, y, z*) (((x) & (y)) ^ ((x) & (z)) ^ ((y) & (z)))

4.54.1.4 #define MEMCPY_BCOPY(*d, s, l*) memcpy((d), (s), (l))

4.54.1.5 #define MEMSET_BZERO(*p, l*) memset((p), 0, (l))

4.54.1.6 #define R(*b, x*) ((x) >> (b))

4.54.1.7 #define REVERSE32(*w, x*)

Value:

```
{ \
    sha2_word32 tmp = (w); \
    tmp = (tmp >> 16) | (tmp << 16); \
    (x) = ((tmp & 0xff00ff00UL) >> 8) | ((tmp & 0x00ff00ffUL) << 8); \
}
```

4.54.1.8 #define REVERSE64(*w, x*)

Value:

```
{ \
    sha2_word64 tmp = (w); \
    tmp = (tmp >> 32) | (tmp << 32); \
    tmp = ((tmp & 0xff00ff00ff00ff00ULL) >> 8) | \
          ((tmp & 0x00ff00ff00ff00ffULL) << 8); \
    (x) = ((tmp & 0xffff0000ffff0000ULL) >> 16) | \
          ((tmp & 0x0000ffff0000ffffULL) << 16); \
}
```

4.54.1.9 #define S32(b, x) (((x) >> (b)) | ((x) << (32 - (b))))

4.54.1.10 #define S64(b, x) (((x) >> (b)) | ((x) << (64 - (b))))

4.54.1.11 #define SHA256_SHORT_BLOCK_LENGTH (SHA256_BLOCK_LENGTH - 8)

4.54.1.12 #define SHA2_USE_MEMSET_MEMCPY 1

4.54.1.13 #define SHA384_SHORT_BLOCK_LENGTH (SHA384_BLOCK_LENGTH - 16)

4.54.1.14 #define SHA512_SHORT_BLOCK_LENGTH (SHA512_BLOCK_LENGTH - 16)

4.54.1.15 #define Sigma0_256(x) (S32(2, (x)) ^ S32(13, (x)) ^ S32(22, (x)))

4.54.1.16 #define sigma0_256(x) (S32(7, (x)) ^ S32(18, (x)) ^ R(3 , (x)))

4.54.1.17 #define Sigma0_512(x) (S64(28, (x)) ^ S64(34, (x)) ^ S64(39, (x)))

4.54.1.18 #define sigma0_512(x) (S64(1, (x)) ^ S64(8, (x)) ^ R(7, (x)))

4.54.1.19 #define Sigma1_256(x) (S32(6, (x)) ^ S32(11, (x)) ^ S32(25, (x)))

4.54.1.20 #define sigma1_256(x) (S32(17, (x)) ^ S32(19, (x)) ^ R(10, (x)))

4.54.1.21 #define Sigma1_512(x) (S64(14, (x)) ^ S64(18, (x)) ^ S64(41, (x)))

4.54.1.22 #define sigma1_512(x) (S64(19, (x)) ^ S64(61, (x)) ^ R(6, (x)))

4.54.2 Typedef Documentation

4.54.2.1 typedef uint8_t sha2_byte

4.54.2.2 typedef uint32_t sha2_word32

4.54.2.3 typedef uint64_t sha2_word64

4.54.3 Function Documentation

4.54.3.1 char* SHA256_Data (const sha2_byte * *data,* size_t *len,* char *digest[SHA256_DIGEST_STRING_LENGTH]*)

4.54.3.2 char* SHA256_End (SHA256_CTX * *context,* char *buffer[]*)

4.54.3.3 void SHA256_Final (sha2_byte *digest[],* SHA256_CTX * *context*)

4.54.3.4 void SHA256_Init (SHA256_CTX * *context*)

4.54.3.5 void SHA256_Transform (SHA256_CTX * *context,* const sha2_word32 * *data*)

4.54.3.6 void SHA256_Update (SHA256_CTX * *context,* const sha2_byte * *data,* size_t *len*)

4.54.3.7 char* SHA384_Data (const sha2_byte * *data,* size_t *len,* char *digest[SHA384_DIGEST_STRING_LENGTH]*)

4.54.3.8 char* SHA384_End (SHA384_CTX * *context,* char *buffer[]*)

4.54.3.9 void SHA384_Final (sha2_byte *digest[],* SHA384_CTX * *context*)

4.54.3.10 void SHA384_Init (SHA384_CTX * *context*)

4.54.3.11 void SHA384_Update (SHA384_CTX * *context*, const sha2_byte * *data*, size_t *len*)

4.54.3.12 char* SHA512_Data (const sha2_byte * *data*, size_t *len*, char *digest[SHA512_DIGEST_STRING_LENGTH]*)

4.54.3.13 char* SHA512_End (SHA512_CTX * *context*, char *buffer[]*)

4.54.3.14 void SHA512_Final (sha2_byte *digest[]*, SHA512_CTX * *context*)

4.54.3.15 void SHA512_Init (SHA512_CTX * *context*)

4.54.3.16 void SHA512_Last (SHA512_CTX * *context*)

4.54.3.17 void SHA512_Transform (SHA512_CTX * *context*, const sha2_word64 * *data*)

4.54.3.18 void SHA512_Update (SHA512_CTX * *context*, const sha2_byte * *data*, size_t *len*)

4.55 xen/stubdom/domc/crypto/sha2.h File Reference

```
#include <sys/types.h>
#include <inttypes.h>
```

Data Structures

- struct _SHA256_CTX
- struct _SHA512_CTX

Macros

- #define SHA2_USE_INTTYPES_H
- #define LITTLE_ENDIAN 1234
- #define BYTE_ORDER LITTLE_ENDIAN
- #define SHA256_BLOCK_LENGTH 64
- #define SHA256_DIGEST_LENGTH 32
- #define SHA256_DIGEST_STRING_LENGTH (SHA256_DIGEST_LENGTH * 2 + 1)
- #define SHA384_BLOCK_LENGTH 128
- #define SHA384_DIGEST_LENGTH 48
- #define SHA384_DIGEST_STRING_LENGTH (SHA384_DIGEST_LENGTH * 2 + 1)
- #define SHA512_BLOCK_LENGTH 128
- #define SHA512_DIGEST_LENGTH 64
- #define SHA512_DIGEST_STRING_LENGTH (SHA512_DIGEST_LENGTH * 2 + 1)

Typedefs

- typedef struct _SHA256_CTX SHA256_CTX
- typedef struct _SHA512_CTX SHA512_CTX
- typedef SHA512_CTX SHA384_CTX

Functions

- void SHA256_Init (SHA256_CTX ∗)
- void SHA256_Update (SHA256_CTX ∗, const uint8_t ∗, size_t)
- void SHA256_Final (uint8_t[SHA256_DIGEST_LENGTH], SHA256_CTX ∗)
- char ∗ SHA256_End (SHA256_CTX ∗, char[SHA256_DIGEST_STRING_LENGTH])
- char ∗ SHA256_Data (const uint8_t ∗, size_t, char[SHA256_DIGEST_STRING_LENGTH])
- void SHA384_Init (SHA384_CTX ∗)
- void SHA384_Update (SHA384_CTX ∗, const uint8_t ∗, size_t)
- void SHA384_Final (uint8_t[SHA384_DIGEST_LENGTH], SHA384_CTX ∗)
- char ∗ SHA384_End (SHA384_CTX ∗, char[SHA384_DIGEST_STRING_LENGTH])
- char ∗ SHA384_Data (const uint8_t ∗, size_t, char[SHA384_DIGEST_STRING_LENGTH])
- void SHA512_Init (SHA512_CTX ∗)
- void SHA512_Update (SHA512_CTX ∗, const uint8_t ∗, size_t)
- void SHA512_Final (uint8_t[SHA512_DIGEST_LENGTH], SHA512_CTX ∗)
- char ∗ SHA512_End (SHA512_CTX ∗, char[SHA512_DIGEST_STRING_LENGTH])
- char ∗ SHA512_Data (const uint8_t ∗, size_t, char[SHA512_DIGEST_STRING_LENGTH])

4.55.1 Macro Definition Documentation

4.55.1.1 #define BYTE_ORDER LITTLE_ENDIAN

4.55.1.2 #define LITTLE_ENDIAN 1234

4.55.1.3 #define SHA256_BLOCK_LENGTH 64

Import u_intXX_t size_t type definitions from system headers. You may need to change this, or define these things yourself in this file.

4.55.1.4 #define SHA256_DIGEST_LENGTH 32

4.55.1.5 #define SHA256_DIGEST_STRING_LENGTH (SHA256_DIGEST_LENGTH ∗ 2 + 1)

4.55.1.6 #define SHA2_USE_INTTYPES_H

4.55.1.7 #define SHA384_BLOCK_LENGTH 128

4.55.1.8 #define SHA384_DIGEST_LENGTH 48

4.55.1.9 #define SHA384_DIGEST_STRING_LENGTH (SHA384_DIGEST_LENGTH ∗ 2 + 1)

4.55.1.10 #define SHA512_BLOCK_LENGTH 128

4.55.1.11 #define SHA512_DIGEST_LENGTH 64

4.55.1.12 #define SHA512_DIGEST_STRING_LENGTH (SHA512_DIGEST_LENGTH ∗ 2 + 1)

4.55.2 Typedef Documentation

4.55.2.1 typedef struct _SHA256_CTX SHA256_CTX

NOTE: If your architecture does not define either u_intXX_t types or uintXX_t (from inttypes.h), you may need to define things by hand for your system: Most BSD systems already define u_intXX_t types, as does Linux. Some systems, however, like Compaq's Tru64 Unix instead can use uintXX_t types defined by very recent ANSI C standards and included in the file:

#include <inttypes.h>

If you choose to use <inttypes.h> then please define:

#define SHA2_USE_INTTYPES_H

Or on the command line during compile:

cc -DSHA2_USE_INTTYPES_H ...

4.55.2.2 typedef SHA512_CTX SHA384_CTX

4.55.2.3 typedef struct _SHA512_CTX SHA512_CTX

4.55.3 Function Documentation

4.55.3.1 char* SHA256_Data (const uint8_t * , size_t , char *[SHA256_DIGEST_STRING_LENGTH]*)

4.55.3.2 char* SHA256_End (SHA256_CTX * , char *[SHA256_DIGEST_STRING_LENGTH]*)

4.55.3.3 void SHA256_Final (uint8_t *[SHA256_DIGEST_LENGTH]*, SHA256_CTX *)

4.55.3.4 void SHA256_Init (SHA256_CTX *)

4.55.3.5 void SHA256_Update (SHA256_CTX * , const uint8_t * , size_t)

4.55.3.6 char* SHA384_Data (const uint8_t * , size_t , char *[SHA384_DIGEST_STRING_LENGTH]*)

4.55.3.7 char* SHA384_End (SHA384_CTX * , char *[SHA384_DIGEST_STRING_LENGTH]*)

4.55.3.8 void SHA384_Final (uint8_t *[SHA384_DIGEST_LENGTH]*, SHA384_CTX *)

4.55.3.9 void SHA384_Init (SHA384_CTX *)

4.55.3.10 void SHA384_Update (SHA384_CTX * , const uint8_t * , size_t)

4.55.3.11 char* SHA512_Data (const uint8_t * , size_t , char *[SHA512_DIGEST_STRING_LENGTH]*)

4.55.3.12 char* SHA512_End (SHA512_CTX * , char *[SHA512_DIGEST_STRING_LENGTH]*)

4.55.3.13 void SHA512_Final (uint8_t *[SHA512_DIGEST_LENGTH]*, SHA512_CTX *)

4.55.3.14 void SHA512_Init (SHA512_CTX *)

4.55.3.15 void SHA512_Update (SHA512_CTX * , const uint8_t * , size_t)

4.56 xen/stubdom/domc/cspipe_intf.h File Reference

4.57 xen/stubdom/domt/cspipe_intf.h File Reference

```
#include <inttypes.h>
```

Data Structures

- struct cspipe

Functions

- struct cspipe __attribute__ ((__packed__))

Variables

- uint32_t magic_nr
- uint32_t domid_domu
- uint32_t domid_domc
- char s_key [128/8]

4.57.1 Function Documentation

4.57.1.1 struct cspipe __attribute__ ((__packed__))

4.57.2 Variable Documentation

4.57.2.1 uint32_t domid_domc

4.57.2.2 uint32_t domid_domu

4.57.2.3 uint32_t magic_nr

4.57.2.4 char s_key[128/8]

4.58 xen/stubdom/domc/main.h File Reference

```
#include <mini-os/blkfront.h>
#include <minios_macros.h>
```

Data Structures

- struct domc_info

Typedefs

- typedef struct domc_info domc_info_t

Functions

- int wrap_notify_remote_via_evtchn (evtchn_port_t port)

4.58.1 Typedef Documentation

4.58.1.1 typedef struct domc_info domc_info_t

4.58.2 Function Documentation

4.58.2.1 int wrap_notify_remote_via_evtchn (evtchn_port_t *port*)

4.59 xen/stubdom/domt/main.h File Reference

```
#include <minios_macros.h>
```

4.60 xen/stubdom/domt/caas.c File Reference

```
#include <xenctrl.h>
#include <xc_dom.h>
#include <os.h>
#include <sys/mman.h>
#include "main.h"
#include "xlc.h"
#include "tc.h"
#include "caas.h"
```

Functions

- int caas_init (struct tc_state *state)
- int caas_domain_create (libxl_ctx *ctx, libxl_domain_config *d_config, uint32_t *domid_out, uint32_t *domid_out_dc, int restore_fd)
- int caas_setup_cspipe (void *info_p, void *dom_p, unsigned int mfn)

Variables

- struct tc_state * tc

4.60.1 Function Documentation

4.60.1.1 int caas_domain_create (libxl_ctx * *ctx*, libxl_domain_config * *d_config*, uint32_t * *domid_out*, uint32_t * *domid_out_dc*, int *restore_fd*)

The domain creation function specifically finetuned for crypto domain booting.

We create a new domain config which takes the domc_disks / domc_vifs from the original domain config as its disks and vifs.

The supplied kernel/ramdisk information is not used for either domain; the DomU will boot a PVGRUB, while the DomC clone will boot a DomC template.

TODO: it would probably be best if the customer in the encrypted payload can embed a policy that a) indicates the uuid of the primary boot device and b) expresses whether any unencrypted disks should even be attached at all. (Right now the sysadmin can attach those as well.)

4.60.1.2 int caas_init (struct tc_state * *state*)

4.60.1.3 int caas_setup_cspipe (void * *info_p*, void * *dom_p*, unsigned int *mfn*)

Set-up the CaaS pipe between DomT <-> DomC. Initially, this will contain the AES key; later on it should contain also the vTPM NV, perhaps in the form as a pipe (therefore this naming) in similar form as the vTPM pipes normally in Dom0. Right now, this is a callback from inside libxl; in the future it would be cleaner to use a ring buffer and event channels so that we can set it up after the domain has been built, making the callback unnecessary.

4.60.2 Variable Documentation

4.60.2.1 struct tc_state* tc

Since the module is a singleton, this is OK.

4.61 xen/stubdom/domt/caas.h File Reference

```
#include <libxl.h>
#include "tc.h"
#include "cspipe_intf.h"
```

Data Structures

- struct vmcb

Functions

- struct vmcb __attribute__ ((__packed__))
- int caas_init (struct tc_state *state)
- int caas_domain_create (libxl_ctx *ctx, libxl_domain_config *d_config, uint32_t *domid_out, uint32_t *domid_out_dc, int restore_fd)
- int caas_setup_cspipe (void *info_p, void *dom_p, unsigned int mfn)

Variables

- char s_key [128/8]

4.61.1 Function Documentation

4.61.1.1 struct vmcb __attribute__ ((__packed__))

4.61.1.2 int caas_domain_create (libxl_ctx * *ctx*, libxl_domain_config * *d_config*, uint32_t * *domid_out*, uint32_t * *domid_out_dc*, int *restore_fd*)

The domain creation function specifically finetuned for crypto domain booting.

We create a new domain config which takes the domc_disks / domc_vifs from the original domain config as its disks and vifs.

The supplied kernel/ramdisk information is not used for either domain; the DomU will boot a PVGRUB, while the DomC clone will boot a DomC template.

TODO: it would probably be best if the customer in the encrypted payload can embed a policy that a) indicates the uuid of the primary boot device and b) expresses whether any unencrypted disks should even be attached at all. (Right now the sysadmin can attach those as well.)

4.61.1.3 int caas_init (struct tc_state * *state*)

4.61.1.4 int caas_setup_cspipe (void * *info_p*, void * *dom_p*, unsigned int *mfn*)

Set-up the CaaS pipe between DomT <-> DomC. Initially, this will contain the AES key; later on it should contain also the vTPM NV, perhaps in the form as a pipe (therefore this naming) in similar form as the vTPM pipes normally

in Dom0. Right now, this is a callback from inside libxl; in the future it would be cleaner to use a ring buffer and event channels so that we can set it up after the domain has been built, making the callback unnecessary.

4.61.2 Variable Documentation

4.61.2.1 char s_key[128/8]

4.62 xen/stubdom/domt/crypto/hmac.c File Reference

```
#include "hmac.h"
```

Functions

- void tpm_hmac_init (tpm_hmac_ctx_t *ctx, const uint8_t *key, size_t key_len)
- void tpm_hmac_update (tpm_hmac_ctx_t *ctx, const uint8_t *data, size_t length)
- void tpm_hmac_final (tpm_hmac_ctx_t *ctx, uint8_t *digest)

4.62.1 Function Documentation

4.62.1.1 void tpm_hmac_final (tpm_hmac_ctx_t * *ctx*, uint8_t * *digest*)

4.62.1.2 void tpm_hmac_init (tpm_hmac_ctx_t * *ctx*, const uint8_t * *key*, size_t *key_len*)

4.62.1.3 void tpm_hmac_update (tpm_hmac_ctx_t * *ctx*, const uint8_t * *data*, size_t *length*)

4.63 xen/stubdom/domt/crypto/hmac.h File Reference

```
#include "sha1.h"
```

Data Structures

- struct tpm_hmac_ctx_t

Macros

- #define HMAC_PAD_LENGTH 64

Functions

- void tpm_hmac_init (tpm_hmac_ctx_t *ctx, const uint8_t *key, size_t key_len)
- void tpm_hmac_update (tpm_hmac_ctx_t *ctx, const uint8_t *data, size_t length)
- void tpm_hmac_final (tpm_hmac_ctx_t *ctx, uint8_t *digest)

Variables

- typedef __attribute__

4.63.1 Macro Definition Documentation

4.63.1.1 #define HMAC_PAD_LENGTH 64

4.63.2 Function Documentation

4.63.2.1 void tpm_hmac_final (tpm_hmac_ctx_t * *ctx*, uint8_t * *digest*)

4.63.2.2 void tpm_hmac_init (tpm_hmac_ctx_t * *ctx*, const uint8_t * *key*, size_t *key_len*)

4.63.2.3 void tpm_hmac_update (tpm_hmac_ctx_t * *ctx*, const uint8_t * *data*, size_t *length*)

4.63.3 Variable Documentation

4.63.3.1 typedef __attribute__

4.64 xen/stubdom/domt/crypto/sha1.c File Reference

```
#include "sha1.h"
```

Macros

- #define rol(v, b) (((v) << (b)) | ((v) >> (32 - (b))))
- #define B0(i) (buf[i] = CPU_TO_BE32(buf[i]))
- #define B1(i)
- #define F0(x, y, z) ((x & (y ^ z)) ^ z)
- #define F1(x, y, z) (x ^ y ^ z)
- #define F2(x, y, z) (((x | y) & z) | (x & y))
- #define R0(a, b, c, d, e, i) e += F0(b,c,d) + B0(i) + 0x5A827999 + rol(a,5); b = rol(b,30);
- #define R1(a, b, c, d, e, i) e += F0(b,c,d) + B1(i) + 0x5A827999 + rol(a,5); b = rol(b,30);
- #define R2(a, b, c, d, e, i) e += F1(b,c,d) + B1(i) + 0x6ED9EBA1 + rol(a,5); b = rol(b,30);
- #define R3(a, b, c, d, e, i) e += F2(b,c,d) + B1(i) + 0x8F1BBCDC + rol(a,5); b = rol(b,30);
- #define R4(a, b, c, d, e, i) e += F1(b,c,d) + B1(i) + 0xCA62C1D6 + rol(a,5); b = rol(b,30);

Functions

- void tpm_sha1_init (tpm_sha1_ctx_t *ctx)
- void tpm_sha1_update (tpm_sha1_ctx_t *ctx, const uint8_t *data, size_t length)
- void tpm_sha1_final (tpm_sha1_ctx_t *ctx, uint8_t digest[SHA1_DIGEST_LENGTH])

4.64.1 Macro Definition Documentation

4.64.1.1 #define B0(*i*) (buf[i] = CPU_TO_BE32(buf[i]))

4.64.1.2 #define B1(*i*)

Value:

```
(buf[i & 15] = rol(buf[i & 15] ^ buf[(i-14) & 15] \
                                ^ buf[(i-8) & 15] ^ buf[(i-3) & 15], 1))
```

4.64.1.3 #define F0(x, y, z) ((x & (y ^ z)) ^ z)

4.64.1.4 #define F1(x, y, z) (x ^ y ^ z)

4.64.1.5 #define F2(x, y, z) (((x | y) & z) | (x & y))

4.64.1.6 #define R0(a, b, c, d, e, i) e += F0(b,c,d) + B0(i) + 0x5A827999 + rol(a,5); b = rol(b,30);

4.64.1.7 #define R1(a, b, c, d, e, i) e += F0(b,c,d) + B1(i) + 0x5A827999 + rol(a,5); b = rol(b,30);

4.64.1.8 #define R2(a, b, c, d, e, i) e += F1(b,c,d) + B1(i) + 0x6ED9EBA1 + rol(a,5); b = rol(b,30);

4.64.1.9 #define R3(a, b, c, d, e, i) e += F2(b,c,d) + B1(i) + 0x8F1BBCDC + rol(a,5); b = rol(b,30);

4.64.1.10 #define R4(a, b, c, d, e, i) e += F1(b,c,d) + B1(i) + 0xCA62C1D6 + rol(a,5); b = rol(b,30);

4.64.1.11 #define rol(v, b) (((v) << (b)) | ((v) >> (32 - (b))))

4.64.2 Function Documentation

4.64.2.1 void tpm_sha1_final (tpm_sha1_ctx_t * *ctx*, uint8_t *digest[SHA1_DIGEST_LENGTH]*)

4.64.2.2 void tpm_sha1_init (tpm_sha1_ctx_t * *ctx*)

4.64.2.3 void tpm_sha1_update (tpm_sha1_ctx_t * *ctx*, const uint8_t * *data*, size_t *length*)

4.65 xen/stubdom/domt/crypto/sha1.h File Reference

```
#include <inttypes.h>
#include <os.h>
#include <byteorder.h>
#include <arpa/inet.h>
#include <mini-os/byteorder.h>
#include <unistd.h>
#include <stdbool.h>
```

Data Structures

- struct tpm_sha1_ctx_t

Macros

- #define CPU_TO_BE32 cpu_to_be32
- #define SHA1_DIGEST_LENGTH 20

Functions

- void tpm_sha1_init (tpm_sha1_ctx_t *ctx)
- void tpm_sha1_update (tpm_sha1_ctx_t *ctx, const uint8_t *data, size_t length)
- void tpm_sha1_final (tpm_sha1_ctx_t *ctx, uint8_t digest[SHA1_DIGEST_LENGTH])

4.65.1 Macro Definition Documentation

4.65.1.1 #define CPU_TO_BE32 cpu_to_be32

Necessary includes for the port of the library to Mini-OS.

4.65.1.2 #define SHA1_DIGEST_LENGTH 20

4.65.2 Function Documentation

4.65.2.1 void tpm_sha1_final (tpm_sha1_ctx_t * *ctx*, uint8_t *digest[SHA1_DIGEST_LENGTH]*)

4.65.2.2 void tpm_sha1_init (tpm_sha1_ctx_t * *ctx*)

4.65.2.3 void tpm_sha1_update (tpm_sha1_ctx_t * *ctx*, const uint8_t * *data*, size_t *length*)

4.66 xen/stubdom/domt/tc.c File Reference

```
#include <unistd.h>
#include <os.h>
#include <malloc.h>
#include <string.h>
#include <mini-os/ioremap.h>
#include <mini-os/tpm_tis.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdlib.h>
#include "main.h"
#include "tc.h"
#include "tpm_defs.h"
#include "hmac.h"
```

Macros

- #define TPM_DEFAULT_ENCSIZE 256
- #define TPM_DEFAULT_PKEYLEN 256
- #define TC_CHECK_BUF_OVERFLOW(buf, len)
- #define TC_CHECK_RC_ERROR(ret, rc, rsplen)
- #define TC_CHECK_IF_ALL_READ(p, rsp, rsplen)
- #define TC_CHECK_DIGESTS(resAuth, verifyAuth)
- #define TC_OPT_INVALIDATE_OIAP(s, contAuthSes)
- #define TC_OPT_INVALIDATE_OSAP(s, contAuthSes)
- #define TC_CHECK_TPM_TIS_CMD(chip, buf, len, rsp, rsplen)
- #define FILL_CONVERT 1
- #define TC_PLACE_SZ(SZ)
- #define tc_place_(a, b, c, d) tc_place(a, b, c, d, NULL, NULL)
- #define tc_place_s(a, b, c, d, e) tc_place(a, b, c, d, e, NULL)
- #define tc_place_h(a, b, c, d, e) tc_place(a, b, c, d, NULL, e)
- #define tc_place16_(a, b, c) tc_place16(a, b, c, NULL, NULL)
- #define tc_place16_s(a, b, c, d) tc_place16(a, b, c, d, NULL)
- #define tc_place16_h(a, b, c, d) tc_place16(a, b, c, NULL, d)
- #define tc_place32_(a, b, c) tc_place32(a, b, c, NULL, NULL)
- #define tc_place32_s(a, b, c, d) tc_place32(a, b, c, d, NULL)
- #define tc_place32_h(a, b, c, d) tc_place32(a, b, c, NULL, d)

Functions

- int tc_transmit_cmd (struct tc_info *info, struct tpm_cmd *cmd)
- int tc_createwrapkey (struct tc_info *info, struct osap_sess *s, uint32_t handle, uint8_t *dataUsageAuth, size_t dataUsageAuthLen, struct TPM_KEY12 *key, struct TPM_KEY12 *key2)
- int tc_loadkey2 (struct tc_info *info, struct oiap_sess *s, uint32_t parentHandle, uint8_t *parentAuth, size_t parentAuthLen, struct TPM_KEY12 *key, uint32_t *inkeyHandle, uint8_t contAuthSes)
- int tc_chk_loadkey2 (struct tc_info *info, struct oiap_sess *s, uint32_t parentHandle, uint8_t *parentAuth, size_t parentAuthLen, struct TPM_KEY12 *key, uint32_t *inkeyHandle, uint8_t contAuthSes)
- int tc_unbind (struct tc_info *info, struct oiap_sess *s, uint32_t keyHandle, uint8_t *keyAuth, size_t keyAuthLen, uint8_t *inData, uint32_t inDataSize, uint8_t **outData, uint32_t *outDataSize, uint8_t contAuthSes)
- int tc_listkeys (struct tc_info *info, uint32_t **keylist, uint16_t *keylistSize)
- void tc_flush_all_keys (struct tc_info *info)
- void tc_print_pcrs (struct tc_info *info)
- int tpm_pcr_extend (struct tc_info *info, int pcr_idx, const uint8_t *hash)
- int tpm_get_random (struct tc_info *info, unsigned char *buf, uint32_t len)
- int init_oiap (struct tc_info *info, struct oiap_sess *s)
- int init_osap (struct tc_info *info, struct osap_sess *s, const unsigned char *key, uint16_t type, uint32_t handle)
- int tc_create_key (struct tc_info *info, struct osap_sess *s, struct TPM_KEY12 **in_key, uint8_t *dataUsageAuth, size_t len)
- void tc_free_key (struct TPM_KEY12 *key)
- int tc_reset (struct tc_info *info)
- int tc_flushspecific (struct tc_info *info, uint32_t handle, uint32_t resourceType)
- struct tc_state * init_tc (void)
- void shutdown_tc (struct tc_state *state)
- void tc_test (struct tc_state *state)
- void tc_hexdump (void *p, size_t len)

4.66.1 Macro Definition Documentation

4.66.1.1 #define FILL_CONVERT 1

NOTE: we *must* hash everything in big endian to match TPM. However, if dumping to a place in cpu ordering, one can set as a non-zero value != FILL_CONVERT (e.g., 2). Secondly, setting == NULL is an 'update hash only' modus.

4.66.1.2 #define TC_CHECK_BUF_OVERFLOW(*buf, len*)

Value:

```
DEBUG("[cmd length: %d]\n", len);                                   \
    if (len > sizeof(buf)) {                                            \
        PANIC("FATAL: TPM buffer overflowed!\n");                       \
    }
```

Helper macros to make the code with TC commands less cluttered by removing boilerplate code. These should not be used standalone, due to hidden assumptions on context.

4.66.1.3 #define TC_CHECK_DIGESTS(*resAuth, verifyAuth*)

Value:

```
if (memcmp(resAuth, verifyAuth, TPM_DIGEST_SIZE)) {                 \
        PANIC("TPM return digest did not match!\n");                    \
    }                                                                   \
```

4.66.1.4 #define TC_CHECK_IF_ALL_READ(*p, rsp, rsplen*)

Value:

```
if ((p - rsp) != rsplen) {                                          \
        DEBUG("Error: walking pointer has read != rsplen\n");           \
        goto out;                                                       \
    }
```

4.66.1.5 #define TC_CHECK_RC_ERROR(*ret, rc, rsplen*)

Value:

```
if (rsplen == TPM_ERROR_SIZE) {                                     \
        DEBUG("warning: resplen (%d) indicates an error\n", rsplen);    \
        if ((ret = rc)) {                                               \
            DEBUG("TPM error: %d (0x%x)\n", rc, rc);                    \
            goto out;                                                   \
        }   }
```

4.66.1.6 #define TC_CHECK_TPM_TIS_CMD(*chip, buf, len, rsp, rsplen*)

Value:

```
if ((ret = tpm_tis_cmd(chip, buf, len, &rsp, &rsplen))) {           \
        DEBUG("warning: non-zero return from tpm_tis driver: %d\n", ret);   \
        /*goto out;*/                                                   \
    }                                                                   \
    if (rsplen < sizeof(struct tpm_input_header)) {                     \
        DEBUG("Error: TPM cmd returned an inconsistent resplen."        \
              " (Usually caused by bad formatted input.)\n");           \
        ret = rsplen;                                                   \
        goto out;                                                       \
    }                                                                   \
```

4.66.1.7 #define TC_OPT_INVALIDATE_OIAP(*s, contAuthSes*)

Value:

```
if (contAuthSes == 0) {                                             \
        s->handle = 0;                                                  \
        memset(s->enonce, 0, sizeof(s->enonce));                        \
    }
```

4.66.1.8 #define TC_OPT_INVALIDATE_OSAP(*s, contAuthSes*)

Value:

```
if (contAuthSes == 0) {                                             \
        s->handle = 0;                                                  \
        memset(s->secret, 0, sizeof(s->secret));                        \
        memset(s->enonce, 0, sizeof(s->enonce));                        \
    }
```

4.66.1.9 #define tc_place16_(a, b, c) tc_place16(a, b, c, NULL, NULL)

4.66.1.10 #define tc_place16_h(a, b, c, d) tc_place16(a, b, c, NULL, d)

4.66.1.11 #define tc_place16_s(a, b, c, d) tc_place16(a, b, c, d, NULL)

4.66.1.12 #define tc_place32_(a, b, c) tc_place32(a, b, c, NULL, NULL)

4.66.1.13 #define tc_place32_h(a, b, c, d) tc_place32(a, b, c, NULL, d)

4.66.1.14 #define tc_place32_s(a, b, c, d) tc_place32(a, b, c, d, NULL)

4.66.1.15 #define tc_place_(a, b, c, d) tc_place(a, b, c, d, NULL, NULL)

4.66.1.16 #define tc_place_h(a, b, c, d, e) tc_place(a, b, c, d, NULL, e)

4.66.1.17 #define tc_place_s(a, b, c, d, e) tc_place(a, b, c, d, e, NULL)

4.66.1.18 #define TC_PLACE_SZ(SZ)

Value:

```
static void tc_place ## SZ (int fill, uint8_t **p, uint ## SZ ## _t *in,     \
                                tpm_sha1_ctx_t *shh, tpm_hmac_ctx_t *hmh)     \
{                                                                              \
    uint ## SZ ## _t tmp;                                                      \
    int len = sizeof(*in);                                                     \
    BUG_ON(!in);                                                               \
    if (fill == FILL_CONVERT) {                                                \
        tmp = cpu_to_be ## SZ (*in);                                           \
        in = &tmp;                                                             \
    }                                                                          \
    if (!fill)                                                                 \
        memcpy(in, *p, len);                                                   \
    if (shh != NULL)                                                           \
        tpm_sha1_update(shh, (uint8_t*)in, len);                               \
    if (hmh != NULL)                                                           \
        tpm_hmac_update(hmh, (uint8_t*)in, len);                               \
    if (!p || !*p)                                                             \
        return;                                                                \
    if (fill)                                                                  \
        memcpy(*p, in, len);                                                   \
    if (!fill)                                                                 \
        *in = cpu_to_be ## SZ (*in);                                           \
    *p += len;                                                                 \
}
```

4.66.1.19 #define TPM_DEFAULT_ENCSIZE 256

Code comments are sparse, refer to TPM spec for details.

4.66.1.20 #define TPM_DEFAULT_PKEYLEN 256

4.66.2 Function Documentation

4.66.2.1 int init_oiap (struct tc_info * *info*, struct oiap_sess * *s*)

4.66.2.2 int init_osap (struct tc_info * *info*, struct osap_sess * *s*, const unsigned char * *key*, uint16_t *type*, uint32_t *handle*)

Create an object specific authorisation protocol (OSAP) session.

4.66.2.3 struct tc_state∗ init_tc (void)

4.66.2.4 void shutdown_tc (struct tc_state ∗ *state*)

4.66.2.5 int tc_chk_loadkey2 (struct tc_info ∗ *info,* struct oiap_sess ∗ *s,* uint32_t *parentHandle,* uint8_t ∗ *parentAuth,* size_t *parentAuthLen,* struct TPM_KEY12 ∗ *key,* uint32_t ∗ *inkeyHandle,* uint8_t *contAuthSes*)

4.66.2.6 int tc_create_key (struct tc_info ∗ *info,* struct osap_sess ∗ *s,* struct TPM_KEY12 ∗∗ *in_key,* uint8_t ∗ *dataUsageAuth,* size_t *len*)

4.66.2.7 int tc_createwrapkey (struct tc_info ∗ *info,* struct osap_sess ∗ *s,* uint32_t *handle,* uint8_t ∗ *dataUsageAuth,* size_t *dataUsageAuthLen,* struct TPM_KEY12 ∗ *key,* struct TPM_KEY12 ∗ *key2*)

Because of the variable length arguments, we do not use the tpm_input_∗ structs approach. Passing == is allowed.

4.66.2.8 void tc_flush_all_keys (struct tc_info ∗ *info*)

4.66.2.9 int tc_flushspecific (struct tc_info ∗ *info,* uint32_t *handle,* uint32_t *resourceType*)

4.66.2.10 void tc_free_key (struct TPM_KEY12 ∗ *key*)

4.66.2.11 void tc_hexdump (void ∗ *p,* size_t *len*)

Not a particular TC related function, but heavily used in all modules for debugging.

4.66.2.12 int tc_listkeys (struct tc_info ∗ *info,* uint32_t ∗∗ *keylist,* uint16_t ∗ *keylistSize*)

If necessary, this could be generalized to a get_capability function.

4.66.2.13 int tc_loadkey2 (struct tc_info ∗ *info,* struct oiap_sess ∗ *s,* uint32_t *parentHandle,* uint8_t ∗ *parentAuth,* size_t *parentAuthLen,* struct TPM_KEY12 ∗ *key,* uint32_t ∗ *inkeyHandle,* uint8_t *contAuthSes*)

Caller is responsible for being properly initialized.

4.66.2.14 void tc_print_pcrs (struct tc_info ∗ *info*)

4.66.2.15 int tc_reset (struct tc_info ∗ *info*)

RESET is deprecated in 1.2 of spec.

4.66.2.16 void tc_test (struct tc_state ∗ *state*)

4.66.2.17 int tc_transmit_cmd (struct tc_info ∗ *info,* struct tpm_cmd ∗ *cmd*)

Note: not to be used with variable-length commands such as createwrapkey, loadkey2, etc.

4.66.2.18 int tc_unbind (struct tc_info ∗ *info,* struct oiap_sess ∗ *s,* uint32_t *keyHandle,* uint8_t ∗ *keyAuth,* size_t *keyAuthLen,* uint8_t ∗ *inData,* uint32_t *inDataSize,* uint8_t ∗∗ *outData,* uint32_t ∗ *outDataSize,* uint8_t *contAuthSes*)

4.66.2.19 int tpm_get_random (struct tc_info ∗ *info,* unsigned char ∗ *buf,* uint32_t *len*)

4.66.2.20 int tpm_pcr_extend (struct tc_info * *info,* int *pcr_idx,* const uint8_t * *hash*)

4.67 xen/stubdom/domt/tc.h File Reference

```
#include <tpm_tis.h>
#include "tpm_defs.h"
#include "hmac.h"
```

Data Structures

- struct tc_info
- struct tc_state
- struct osap_sess
- struct oiap_sess

Macros

- #define TPM_MFN (TPM_BASEADDR >> PAGE_SHIFT)
- #define TPM_LEN 5
- #define TPM_LOCS TPM_TIS_EN_LOCL0
- #define TPM_NUM_PCR 24
- #define TC_PATH_PREFIX "/var/"
- #define TC_PATH_KEYDATA TC_PATH_PREFIX "domt_keydata.bin"
- #define TC_PATH_PUBKEY TC_PATH_PREFIX "domt_pubkey.bin"

Functions

- struct tc_state * init_tc (void)
- void shutdown_tc (struct tc_state *state)
- int tc_transmit_cmd (struct tc_info *info, struct tpm_cmd *in)
- int tc_createwrapkey (struct tc_info *info, struct osap_sess *s, uint32_t handle, uint8_t *dataUsageAuth, size_t dataUsageAuthLen, struct TPM_KEY12 *key, struct TPM_KEY12 *key2)
- int tc_loadkey2 (struct tc_info *info, struct oiap_sess *s, uint32_t parentHandle, uint8_t *parentAuth, size_t parentAuthLen, struct TPM_KEY12 *key, uint32_t *inkeyHandle, uint8_t contAuthSes)
- int tc_chk_loadkey2 (struct tc_info *info, struct oiap_sess *s, uint32_t parentHandle, uint8_t *parentAuth, size_t parentAuthLen, struct TPM_KEY12 *key, uint32_t *inkeyHandle, uint8_t contAuthSes)
- int tc_unbind (struct tc_info *info, struct oiap_sess *s, uint32_t keyHandle, uint8_t *keyAuth, size_t keyAuthLen, uint8_t *inData, uint32_t inDataSize, uint8_t **outData, uint32_t *outDataSize, uint8_t contAuthSes)
- int tc_listkeys (struct tc_info *info, uint32_t **keylist, uint16_t *keylistSize)
- void tc_flush_all_keys (struct tc_info *info)
- void tc_print_pcrs (struct tc_info *info)
- int tpm_pcr_extend (struct tc_info *info, int pcr_idx, const uint8_t *hash)
- int tpm_get_random (struct tc_info *info, unsigned char *buf, uint32_t len)
- int init_oiap (struct tc_info *info, struct oiap_sess *s)
- int init_osap (struct tc_info *info, struct osap_sess *s, const unsigned char *key, uint16_t type, uint32_t handle)
- int tc_create_key (struct tc_info *info, struct osap_sess *s, struct TPM_KEY12 **in_key, uint8_t *dataUsageAuth, size_t len)
- void tc_free_key (struct TPM_KEY12 *key)
- int tc_reset (struct tc_info *info)
- int tc_flushspecific (struct tc_info *info, uint32_t handle, uint32_t resourceType)
- void tc_test (struct tc_state *state)
- void tc_hexdump (void *p, size_t len)

4.67.1 Macro Definition Documentation

4.67.1.1 #define TC_PATH_KEYDATA TC_PATH_PREFIX "domt_keydata.bin"

4.67.1.2 #define TC_PATH_PREFIX "/var/"

4.67.1.3 #define TC_PATH_PUBKEY TC_PATH_PREFIX "domt_pubkey.bin"

4.67.1.4 #define TPM_LEN 5

4.67.1.5 #define TPM_LOCS TPM_TIS_EN_LOCL0

4.67.1.6 #define TPM_MFN (TPM_BASEADDR >> PAGE_SHIFT)

4.67.1.7 #define TPM_NUM_PCR 24

4.67.2 Function Documentation

4.67.2.1 int init_oiap (struct tc_info * *info*, struct oiap_sess * *s*)

4.67.2.2 int init_osap (struct tc_info * *info*, struct osap_sess * *s*, const unsigned char * *key*, uint16_t *type*, uint32_t *handle*)

Create an object specific authorisation protocol (OSAP) session.

4.67.2.3 struct tc_state* init_tc (void)

4.67.2.4 void shutdown_tc (struct tc_state * *state*)

4.67.2.5 int tc_chk_loadkey2 (struct tc_info * *info*, struct oiap_sess * *s*, uint32_t *parentHandle*, uint8_t * *parentAuth*, size_t *parentAuthLen*, struct TPM_KEY12 * *key*, uint32_t * *inkeyHandle*, uint8_t *contAuthSes*)

4.67.2.6 int tc_create_key (struct tc_info * *info*, struct osap_sess * *s*, struct TPM_KEY12 ** *in_key*, uint8_t * *dataUsageAuth*, size_t *len*)

4.67.2.7 int tc_createwrapkey (struct tc_info * *info*, struct osap_sess * *s*, uint32_t *handle*, uint8_t * *dataUsageAuth*, size_t *dataUsageAuthLen*, struct TPM_KEY12 * *key*, struct TPM_KEY12 * *key2*)

Because of the variable length arguments, we do not use the tpm_input_* structs approach. Passing == is allowed.

4.67.2.8 void tc_flush_all_keys (struct tc_info * *info*)

4.67.2.9 int tc_flushspecific (struct tc_info * *info*, uint32_t *handle*, uint32_t *resourceType*)

4.67.2.10 void tc_free_key (struct TPM_KEY12 * *key*)

4.67.2.11 void tc_hexdump (void * *p*, size_t *len*)

Not a particular TC related function, but heavily used in all modules for debugging.

4.67.2.12 int tc_listkeys (struct tc_info * *info*, uint32_t ** *keylist*, uint16_t * *keylistSize*)

If necessary, this could be generalized to a get_capability function.

4.67.2.13 int tc_loadkey2 (struct tc_info * *info*, struct oiap_sess * *s*, uint32_t *parentHandle*, uint8_t * *parentAuth*, size_t *parentAuthLen*, struct TPM_KEY12 * *key*, uint32_t * *inkeyHandle*, uint8_t *contAuthSes*)

Caller is responsible for being properly initialized.

4.67.2.14 void tc_print_pcrs (struct tc_info * *info*)

4.67.2.15 int tc_reset (struct tc_info * *info*)

RESET is deprecated in 1.2 of spec.

4.67.2.16 void tc_test (struct tc_state * *state*)

4.67.2.17 int tc_transmit_cmd (struct tc_info * *info*, struct tpm_cmd * *cmd*)

Note: not to be used with variable-length commands such as createwrapkey, loadkey2, etc.

4.67.2.18 int tc_unbind (struct tc_info * *info*, struct oiap_sess * *s*, uint32_t *keyHandle*, uint8_t * *keyAuth*, size_t *keyAuthLen*, uint8_t * *inData*, uint32_t *inDataSize*, uint8_t ** *outData*, uint32_t * *outDataSize*, uint8_t *contAuthSes*)

4.67.2.19 int tpm_get_random (struct tc_info * *info*, unsigned char * *buf*, uint32_t *len*)

4.67.2.20 int tpm_pcr_extend (struct tc_info * *info*, int *pcr_idx*, const uint8_t * *hash*)

4.68 xen/stubdom/domt/tpm_defs.c File Reference

```
#include <mini-os/byteorder.h>
#include "tpm_defs.h"
```

Variables

- struct tpm_input_header tpm_readpubek_header
- struct tpm_input_header pcrread_header
- struct tpm_input_header pcrextend_header
- struct tpm_input_header oiap_header
- struct tpm_input_header osap_header
- struct tpm_input_header getrandom_header
- struct tpm_input_header reset_header
- struct tpm_input_header flushspecific_header

4.68.1 Variable Documentation

4.68.1.1 struct tpm_input_header flushspecific_header

Initial value:

```
= {
    .tag = _TPM_TAG_RQU_COMMAND,
    .length = cpu_to_be32(18),
    .ordinal = _TPM_ORD_FLUSHSPECIFIC,
}
```

4.68.1.2 struct tpm_input_header getrandom_header

Initial value:

```
= {
    .tag = _TPM_TAG_RQU_COMMAND,
    .length = cpu_to_be32(14),
    .ordinal = _TPM_ORD_GETRANDOM,
}
```

4.68.1.3 struct tpm_input_header oiap_header

Initial value:

```
= {
    .tag = _TPM_TAG_RQU_COMMAND,
    .length = cpu_to_be32(10),
    .ordinal = _TPM_ORD_OIAP,
}
```

4.68.1.4 struct tpm_input_header osap_header

Initial value:

```
= {
    .tag = _TPM_TAG_RQU_COMMAND,
    .length = cpu_to_be32(36),
    .ordinal = _TPM_ORD_OSAP,
}
```

4.68.1.5 struct tpm_input_header pcrextend_header

Initial value:

```
= {
    .tag = _TPM_TAG_RQU_COMMAND,
    .length = cpu_to_be32(34),
    .ordinal = _TPM_ORD_PCR_EXTEND,
}
```

4.68.1.6 struct tpm_input_header pcrread_header

Initial value:

```
= {
    .tag = _TPM_TAG_RQU_COMMAND,
    .length = cpu_to_be32(14),
    .ordinal = _TPM_ORDINAL_PCRREAD,
}
```

4.68.1.7 struct tpm_input_header reset_header

Initial value:

```
= {
    .tag = _TPM_TAG_RQU_COMMAND,
    .length = cpu_to_be32(10),
    .ordinal = _TPM_ORD_RESET,
}
```

4.68.1.8 struct tpm_input_header tpm_readpubek_header

Initial value:

```
= {
    .tag = _TPM_TAG_RQU_COMMAND,
    .length = cpu_to_be32(30),
    .ordinal = _TPM_ORD_READPUBEK,
}
```

4.69 xen/stubdom/domt/tpm_defs.h File Reference

```
#include <stdbool.h>
#include <inttypes.h>
#include <unistd.h>
#include <os.h>
```

Data Structures

- struct tpm_input_header
- struct tpm_output_header
- struct tpm_readpubek_params_out
- struct tpm_pcrread_in
- struct tpm_pcrread_out
- struct tpm_pcrextend_in
- struct tpm_oiap_out
- struct tpm_osap_in
- struct tpm_osap_out
- struct tpm_getrandom_in
- struct tpm_getrandom_out
- struct tpm_flushspecific_in
- union tpm_cmd_header
- union tpm_cmd_params
- struct tpm_cmd
- struct TPM_KEY_PARMS
- struct TPM_RSA_KEY_PARMS
- struct TPM_STORE_PUBKEY
- struct TPM_KEY12

Macros

- #define TPM_HEADER_SIZE 10
- #define TPM_BUFSIZE 2048
- #define TPM_DIGEST_SIZE 20
- #define TPM_ERROR_SIZE 10
- #define TPM_RET_CODE_IDX 6
- #define TPM_INTERNAL_RESULT_SIZE 200
- #define MAX_RAND_SIZE TPM_DIGEST_SIZE
- #define TPM_CAP_KEY_HANDLE 0x00000007
- #define TPM_CAP_HANDLE 0x00000014
- #define TPM_TAG_RQU_COMMAND 0x00C1
- #define TPM_TAG_RQU_AUTH1_COMMAND 0x00C2
- #define TPM_TAG_RQU_AUTH2_COMMAND 0x00C3

- #define TPM_TAG_RSP_COMMAND 0x00C4
- #define TPM_TAG_RSP_AUTH1_COMMAND 0x00C5
- #define TPM_TAG_RSP_AUTH2_COMMAND 0x00C6
- #define _TPM_TAG_RQU_COMMAND cpu_to_be16(0x00C1)
- #define _TPM_TAG_RQU_AUTH1_COMMAND cpu_to_be16(0x00C2)
- #define _TPM_TAG_RQU_AUTH2_COMMAND cpu_to_be16(0x00C3)
- #define _TPM_ORD_GET_CAP cpu_to_be32(101)
- #define _TPM_ORD_READPUBEK cpu_to_be32(124)
- #define _TPM_ORDINAL_PCRREAD cpu_to_be32(21)
- #define _TPM_ORD_PCR_EXTEND cpu_to_be32(20)
- #define _TPM_ORD_OIAP cpu_to_be32(10)
- #define _TPM_ORD_OSAP cpu_to_be32(11)
- #define _TPM_ORD_GETRANDOM cpu_to_be32(70)
- #define _TPM_ORD_RESET cpu_to_be32(90)
- #define _TPM_ORD_FLUSHSPECIFIC cpu_to_be32(186)
- #define TPM_ORD_GETCAPABILITY 101
- #define TPM_ORD_CREATEWRAPKEY 31
- #define TPM_ORD_LOADKEY2 65
- #define TPM_ORD_UNBIND 30
- #define TPM_RT_KEY 0x00000001
- #define TPM_RT_AUTH 0x00000002
- #define TPM_RT_HASH 0X00000003
- #define TPM_RT_TRANS 0x00000004
- #define TPM_RT_CONTEXT 0x00000005
- #define TPM_RT_COUNTER 0x00000006
- #define TPM_RT_DELEGATE 0x00000007
- #define TPM_RT_DAA_TPM 0x00000008
- #define TPM_RT_DAA_V0 0x00000009
- #define TPM_RT_DAA_V1 0x0000000A
- #define TPM_NOSPACE 17
- #define TPM_KEY_PARMS_EMPTYSIZE (4 + 2 + 2 + 4)
- #define TPM_RSA_KEY_EMPTYSIZE (4 + 4 + 4)
- #define TPM_STORE_PUBKEY_EMPTYSIZE 4
- #define TPM_KEY12_EMPTYSIZE (2 + 2 + 2 + 2 + 1 + 4 + 4)
- #define TPM_KH_SRK 0x40000000
- #define _TPM_KH_SRK cpu_to_be32(0x40000000)
- #define TPM_ET_SRK 0x04
- #define _TPM_ET_SRK cpu_to_be16(0x04)
- #define TPM_KEY_STORAGE 0x0011
- #define TPM_KEY_BIND 0x0014
- #define _TPM_KEY_BIND cpu_to_be16(0x0014)
- #define TPM_TAG_KEY12 0x0028
- #define _TPM_TAG_KEY12 cpu_to_be16(0x0028)
- #define TPM_AUTH_ALWAYS 0x01
- #define TPM_ALG_RSA 0x01
- #define _TPM_ALG_RSA cpu_to_be32(0x01)
- #define TPM_ES_RSAESOAEP_SHA1_MGF1 0x0003
- #define _TPM_ES_RSAESOAEP_SHA1_MGF1 cpu_to_be16(0x0003)
- #define TPM_SS_NONE 0x0001
- #define _TPM_SS_NONE cpu_to_be16(0x0001)

Enumerations

- enum tpm_duration {
  TPM_SHORT = 0, TPM_MEDIUM = 1, TPM_LONG = 2, TPM_UNDEFINED,
  TPM_SHORT = 0, TPM_MEDIUM = 1, TPM_LONG = 2, TPM_UNDEFINED }

Functions

- struct tpm_input_header __attribute__ ((packed))

Variables

- typedef __attribute__
- uint16_t tag
- uint32_t length
- uint32_t ordinal
- uint32_t return_code
- uint8_t algorithm [4]
- uint8_t encscheme [2]
- uint8_t sigscheme [2]
- uint32_t paramsize
- uint8_t parameters [12]
- uint32_t keysize
- uint8_t modulus [256]
- uint8_t checksum [20]
- uint32_t pcr_idx
- uint8_t pcr_result [TPM_DIGEST_SIZE]
- uint8_t hash [TPM_DIGEST_SIZE]
- uint32_t authHandle
- uint8_t nonceEven [TPM_DIGEST_SIZE]
- uint16_t entity_type
- uint32_t entity_value
- uint8_t nonce_odd_osap [TPM_DIGEST_SIZE]
- uint32_t authhandle
- uint8_t nonce_even [TPM_DIGEST_SIZE]
- uint8_t nonce_even_osap [TPM_DIGEST_SIZE]
- uint32_t numbytes
- uint32_t numrandbytes
- uint8_t randbytes [MAX_RAND_SIZE]
- uint32_t handle
- uint32_t resourceType
- tpm_cmd_header header
- tpm_cmd_params params
- struct tpm_input_header tpm_readpubek_header
- struct tpm_input_header pcrread_header
- struct tpm_input_header pcrextend_header
- struct tpm_input_header oiap_header
- struct tpm_input_header osap_header
- struct tpm_input_header getrandom_header
- struct tpm_input_header reset_header
- struct tpm_input_header flushspecific_header

4.69.1 Macro Definition Documentation

4.69.1.1 #define _TPM_ALG_RSA cpu_to_be32(0x01)

4.69.1.2 #define _TPM_ES_RSAESOAEP_SHA1_MGF1 cpu_to_be16(0x0003)

4.69.1.3 #define _TPM_ET_SRK cpu_to_be16(0x04)

4.69.1.4 #define _TPM_KEY_BIND cpu_to_be16(0x0014)

4.69.1.5 #define _TPM_KH_SRK cpu_to_be32(0x40000000)

4.69.1.6 #define _TPM_ORD_FLUSHSPECIFIC cpu_to_be32(186)

4.69.1.7 #define _TPM_ORD_GET_CAP cpu_to_be32(101)

4.69.1.8 #define _TPM_ORD_GETRANDOM cpu_to_be32(70)

4.69.1.9 #define _TPM_ORD_OIAP cpu_to_be32(10)

4.69.1.10 #define _TPM_ORD_OSAP cpu_to_be32(11)

4.69.1.11 #define _TPM_ORD_PCR_EXTEND cpu_to_be32(20)

4.69.1.12 #define _TPM_ORD_READPUBEK cpu_to_be32(124)

4.69.1.13 #define _TPM_ORD_RESET cpu_to_be32(90)

4.69.1.14 #define _TPM_ORDINAL_PCRREAD cpu_to_be32(21)

4.69.1.15 #define _TPM_SS_NONE cpu_to_be16(0x0001)

4.69.1.16 #define _TPM_TAG_KEY12 cpu_to_be16(0x0028)

4.69.1.17 #define _TPM_TAG_RQU_AUTH1_COMMAND cpu_to_be16(0x00C2)

4.69.1.18 #define _TPM_TAG_RQU_AUTH2_COMMAND cpu_to_be16(0x00C3)

4.69.1.19 #define _TPM_TAG_RQU_COMMAND cpu_to_be16(0x00C1)

4.69.1.20 #define MAX_RAND_SIZE TPM_DIGEST_SIZE

4.69.1.21 #define TPM_ALG_RSA 0x01

4.69.1.22 #define TPM_AUTH_ALWAYS 0x01

4.69.1.23 #define TPM_BUFSIZE 2048

4.69.1.24 #define TPM_CAP_HANDLE 0x00000014

4.69.1.25 #define TPM_CAP_KEY_HANDLE 0x00000007

4.69.1.26 #define TPM_DIGEST_SIZE 20

4.69.1.27 #define TPM_ERROR_SIZE 10

4.69.1.28 #define TPM_ES_RSAESOAEP_SHA1_MGF1 0x0003

4.69.1.29 #define TPM_ET_SRK 0x04

4.69.1.30 #define TPM_HEADER_SIZE 10

4.69.1.31 #define TPM_INTERNAL_RESULT_SIZE 200

4.69.1.32 #define TPM_KEY12_EMPTYSIZE (2 + 2 + 2 + 2 + 1 + 4 + 4)

4.69.1.33 #define TPM_KEY_BIND 0x0014

4.69.1.34 #define TPM_KEY_PARMS_EMPTYSIZE (4 + 2 + 2 + 4)

4.69.1.35 #define TPM_KEY_STORAGE 0x0011

4.69.1.36 #define TPM_KH_SRK 0x40000000

4.69.1.37 #define TPM_NOSPACE 17

4.69.1.38 #define TPM_ORD_CREATEWRAPKEY 31

4.69.1.39 #define TPM_ORD_GETCAPABILITY 101

4.69.1.40 #define TPM_ORD_LOADKEY2 65

4.69.1.41 #define TPM_ORD_UNBIND 30

4.69.1.42 #define TPM_RET_CODE_IDX 6

4.69.1.43 #define TPM_RSA_KEY_EMPTYSIZE (4 + 4 + 4)

4.69.1.44 #define TPM_RT_AUTH 0x00000002

4.69.1.45 #define TPM_RT_CONTEXT 0x00000005

4.69.1.46 #define TPM_RT_COUNTER 0x00000006

4.69.1.47 #define TPM_RT_DAA_TPM 0x00000008

4.69.1.48 #define TPM_RT_DAA_V0 0x00000009

4.69.1.49 #define TPM_RT_DAA_V1 0x0000000A

4.69.1.50 #define TPM_RT_DELEGATE 0x00000007

4.69.1.51 #define TPM_RT_HASH 0X00000003

4.69.1.52 #define TPM_RT_KEY 0x00000001

4.69.1.53 #define TPM_RT_TRANS 0x00000004

4.69.1.54 #define TPM_SS_NONE 0x0001

4.69.1.55 #define TPM_STORE_PUBKEY_EMPTYSIZE 4

4.69.1.56 #define TPM_TAG_KEY12 0x0028

4.69.1.57 #define TPM_TAG_RQU_AUTH1_COMMAND 0x00C2

4.69.1.58 #define TPM_TAG_RQU_AUTH2_COMMAND 0x00C3

4.69.1.59 #define TPM_TAG_RQU_COMMAND 0x00C1

4.69.1.60 #define TPM_TAG_RSP_AUTH1_COMMAND 0x00C5

4.69.1.61 #define TPM_TAG_RSP_AUTH2_COMMAND 0x00C6

4.69.1.62 #define TPM_TAG_RSP_COMMAND 0x00C4

4.69.2 Enumeration Type Documentation

4.69.2.1 enum tpm_duration

Enumerator

TPM_SHORT
TPM_MEDIUM
TPM_LONG
TPM_UNDEFINED
TPM_SHORT
TPM_MEDIUM
TPM_LONG
TPM_UNDEFINED

4.69.3 Function Documentation

4.69.3.1 struct tpm_input_header __attribute__ ((packed))

4.69.4 Variable Documentation

4.69.4.1 enum tpm_duration __attribute__

4.69.4.2 uint8_t algorithm[4]

4.69.4.3 uint32_t authHandle

4.69.4.4 uint32_t authhandle

4.69.4.5 uint8_t checksum[20]

4.69.4.6 uint8_t encscheme[2]

4.69.4.7 uint16_t entity_type

4.69.4.8 uint32_t entity_value

4.69.4.9 struct tpm_input_header flushspecific_header

4.69.4.10 struct tpm_input_header getrandom_header

4.69.4.11 uint32_t handle

4.69.4.12 uint8_t hash[TPM_DIGEST_SIZE]

4.69.4.13 tpm_cmd_header header

4.69.4.14 uint32_t keysize

4.69.4.15 uint32_t length

4.69.4.16 uint8_t modulus[256]

4.69.4.17 uint8_t nonce_even[TPM_DIGEST_SIZE]

4.69.4.18 uint8_t nonce_even_osap[TPM_DIGEST_SIZE]

4.69.4.19 uint8_t nonce_odd_osap[TPM_DIGEST_SIZE]

4.69.4.20 uint8_t nonceEven[TPM_DIGEST_SIZE]

4.69.4.21 uint32_t numbytes

4.69.4.22 uint32_t numrandbytes

4.69.4.23 struct tpm_input_header oiap_header

4.69.4.24 uint32_t ordinal

4.69.4.25 struct tpm_input_header osap_header

4.69.4.26 uint8_t parameters[12]

4.69.4.27 tpm_cmd_params params

4.69.4.28 uint32_t paramsize

4.69.4.29 uint32_t pcr_idx

4.69.4.30 uint8_t pcr_result[TPM_DIGEST_SIZE]

4.69.4.31 struct tpm_input_header pcrextend_header

4.69.4.32 struct tpm_input_header pcrread_header

4.69.4.33 uint8_t randbytes[MAX_RAND_SIZE]

4.69.4.34 struct tpm_input_header reset_header

4.69.4.35 uint32_t resourceType

4.69.4.36 uint32_t return_code

4.69.4.37 uint8_t sigscheme[2]

4.69.4.38 uint16_t tag

4.69.4.39 struct tpm_input_header tpm_readpubek_header

4.70 xen/stubdom/domt/xlc.c File Reference

```
#include <xenctrl.h>
#include <os.h>
#include <mini-os/gnttab.h>
#include <sys/mman.h>
#include <unistd.h>
#include <vfs.h>
#include "xlc_intf.h"
#include "main.h"
#include "xlc.h"
#include "caas.h"
```

Functions

- libxl_ctx ∗ xlc_init (void)
- void xlc_listen (libxl_ctx ∗ctx)
- int do_create (libxl_ctx ∗ctx, libxl_domain_config ∗d_config, uint32_t ∗domid)

4.70.1 Function Documentation

4.70.1.1 int do_create (libxl_ctx ∗ *ctx*, libxl_domain_config ∗ *d_config*, uint32_t ∗ *domid*)

Modeled after libxl_create.c:do_domain_create(), with the logic removed which cannot function (yet) with a stubdom domain builder.

4.70.1.2 libxl_ctx∗ xlc_init (void)

4.70.1.3 void xlc_listen (libxl_ctx ∗ *ctx*)

4.71 xen/tools/libxlc/xlc.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <malloc.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/select.h>
#include <sys/mman.h>
#include <assert.h>
#include <xlc_intf.h>
#include "xlc_util.h"
#include "xlc.h"
#include <libxl_internal.h>
```

Macros

- #define TIMEOUT_SEC_INIT 30
- #define TIMEOUT_SEC 100 /∗ normal timeout is higher, due to TPM delay involved! ∗/

- #define DOMID_T 1

Functions

- domid_t read_dyn_domid (void)
- void print_libxl_domain_config (int domid, libxl_domain_config ∗d_config, libxl_device_model_info ∗dm_info)
- int xlc_command (struct xs_handle ∗xsh, struct xlc_cmd ∗cmd)
- int xlc_domain_make (libxl_ctx ∗ctx, libxl_domain_create_info ∗info, uint32_t ∗domid)
- int xlc_domain_build (libxl_ctx ∗ctx, libxl_domain_build_info ∗info, uint32_t domid, libxl_domain_build_state ∗state)
- int xlc_do_domain_create (libxl_ctx ∗ctx, libxl_domain_config ∗d_config, libxl_console_ready cb, void ∗priv, uint32_t ∗domid_out, int restore_fd)

4.71.1 Macro Definition Documentation

4.71.1.1 #define DOMID_T 1

4.71.1.2 #define TIMEOUT_SEC 100 /∗ normal timeout is higher, due to TPM delay involved! ∗/

4.71.1.3 #define TIMEOUT_SEC_INIT 30

4.71.2 Function Documentation

4.71.2.1 void print_libxl_domain_config (int *domid,* **libxl_domain_config** ∗ *d_config,* libxl_device_model_info ∗ *dm_info*)

4.71.2.2 **domid_t** read_dyn_domid (void)

FIXME: the logging and erroring here should use the libxl-offered logger.

4.71.2.3 int xlc_command (struct xs_handle ∗ *xsh,* struct **xlc_cmd** ∗ *cmd*)

4.71.2.4 int xlc_do_domain_create (**libxl_ctx** ∗ *ctx,* **libxl_domain_config** ∗ *d_config,* **libxl_console_ready** *cb,* void ∗ *priv,* uint32_t ∗ *domid_out,* int *restore_fd*)

4.71.2.5 int xlc_domain_build (**libxl_ctx** ∗ *ctx,* libxl_domain_build_info ∗ *info,* uint32_t *domid,* libxl_domain_build_state ∗ *state*)

4.71.2.6 int xlc_domain_make (**libxl_ctx** ∗ *ctx,* libxl_domain_create_info ∗ *info,* uint32_t ∗ *domid*)

Implementation of stub functions which libxl will call.

4.72 xen/stubdom/domt/xlc.h File Reference

```
#include <xenctrl.h>
#include <os.h>
#include <libxl.h>
#include <libxl_internal.h>
#include <xlc_util.h>
```

Functions

- libxl_ctx ∗ xlc_init (void)
- void xlc_listen (libxl_ctx ∗ctx)
- int do_create (libxl_ctx ∗ctx, libxl_domain_config ∗d_config, uint32_t ∗domid)

4.72.1 Function Documentation

4.72.1.1 int do_create (libxl_ctx ∗ *ctx,* libxl_domain_config ∗ *d_config,* uint32_t ∗ *domid*)

Modeled after libxl_create.c:do_domain_create(), with the logic removed which cannot function (yet) with a stubdom domain builder.

4.72.1.2 libxl_ctx∗ xlc_init (void)

4.72.1.3 void xlc_listen (libxl_ctx ∗ *ctx*)

4.73 xen/tools/libxlc/xlc.h File Reference

```
#include <errno.h>
#include <libxl.h>
#include <libxl_utils.h>
```

Macros

- #define XLC_FAILED -EINVAL

Functions

- int xlc_domain_make (libxl_ctx ∗ctx, libxl_domain_create_info ∗info, uint32_t ∗domid)
- int xlc_domain_build (libxl_ctx ∗ctx, libxl_domain_build_info ∗info, uint32_t domid, libxl_domain_build_state ∗state)
- int xlc_do_domain_create (libxl_ctx ∗ctx, libxl_domain_config ∗d_config, libxl_console_ready cb, void ∗priv, uint32_t ∗domid_out, int restore_fd)

4.73.1 Macro Definition Documentation

4.73.1.1 #define XLC_FAILED -EINVAL

4.73.2 Function Documentation

4.73.2.1 int xlc_do_domain_create (libxl_ctx ∗ *ctx,* libxl_domain_config ∗ *d_config,* libxl_console_ready *cb,* void ∗ *priv,* uint32_t ∗ *domid_out,* int *restore_fd*)

4.73.2.2 int xlc_domain_build (libxl_ctx ∗ *ctx,* libxl_domain_build_info ∗ *info,* uint32_t *domid,* libxl_domain_build_state ∗ *state*)

4.73.2.3 int xlc_domain_make (libxl_ctx ∗ *ctx,* libxl_domain_create_info ∗ *info,* uint32_t ∗ *domid*)

Implementation of stub functions which libxl will call.

4.74 xen/stubdom/domt/xlc_intf.h File Reference

```
#include <inttypes.h>
```

Data Structures

- struct xlc_cmd_ping
- struct xlc_cmd_getpage
- struct xlc_cmd_retpage
- struct xlc_cmd_dommake
- struct xlc_cmd_dombuild
- struct xlc_cmd_domcreate
- struct xlc_cmd

Macros

- #define XLC_PATH "/local/domain/0/domt/xlc"

Enumerations

- enum xlc_cmdtype {
  XLC_PING = 0, XLC_GETPAGE = 1, XLC_RETPAGE = 2, XLC_DOMMAKE = 3,
  XLC_DOMBUILD = 4, XLC_DOMCREATE = 5 }

4.74.1 Macro Definition Documentation

4.74.1.1 #define XLC_PATH "/local/domain/0/domt/xlc"

4.74.2 Enumeration Type Documentation

4.74.2.1 enum xlc_cmdtype

Enumerator

XLC_PING

XLC_GETPAGE

XLC_RETPAGE

XLC_DOMMAKE

XLC_DOMBUILD

XLC_DOMCREATE

4.75 xen/tools/libxlc/include/xlc_intf.h File Reference

4.76 xen/stubdom/domt/xlc_util.c File Reference

```
#include <string.h>
#include <errno.h>
#include <libxl_internal.h>
#include "xlc_util.h"
```

Macros

- #define CHK_XLC_MEMCPY_OUT(a, b, c, d) {if ((rc=xlc_memcpy_out(a,b,c,d))) goto error;}
- #define CHK_XLC_MEMCPY_IN(a, b, c, d) {if ((rc=xlc_memcpy_in(a,b,c,d))) goto error;}
- #define CHK_XLC_STRCPY_OUT(a, b, c) {if ((rc=xlc_strcpy_out(a,b,c))) goto error;}
- #define CHK_XLC_STRACPY_IN(a, b, c) {if ((rc=xlc_stracpy_in(a,b,c))) goto error;}
- #define CHK_XLC_DARRAY_OUT(a, b, c) {if ((rc=xlc_darray_out(a,b,c))) goto error;}
- #define CHK_XLC_DARRAY_IN(a, b, c) {if ((rc=xlc_darray_in(a,b,c))) goto error;}
- #define CHK_XLC_SERIALIZE_DISKS(a, b, c, d) {if ((rc=xlc_serialize_disks(a,b,c,d))) goto error;}
- #define CHK_XLC_SERIALIZE_VIFS(a, b, c, d) {if ((rc=xlc_serialize_vifs(a,b,c,d))) goto error;}
- #define CHK_XLC_DESERIALIZE_DISKS(a, b, c, d) {if ((rc=xlc_deserialize_disks(a,b,c,d))) goto error;}
- #define CHK_XLC_DESERIALIZE_VIFS(a, b, c, d) {if ((rc=xlc_deserialize_vifs(a,b,c,d))) goto error;}
- #define CHK_MEMCMP(a, b) {if (memcmp(&a, &b, sizeof(a))) goto error;}
- #define CHK_STRCMP(a, b) {if (a && b && strcmp(a, b)) goto error;}

Functions

- int serialize_domain_create_info (void *buf, size_t len, libxl_domain_create_info *info)
- int deserialize_domain_create_info (void *buf, size_t len, libxl_domain_create_info *info)
- int serialize_domain_build_info (void *buf, size_t len, libxl_domain_build_info *info, libxl_domain_build_state *state)
- int deserialize_domain_build_info (void *buf, size_t len, libxl_domain_build_info *info, libxl_domain_build_state *state)
- int serialize_domain_config (void *buf, size_t len, libxl_domain_config *info)
- int deserialize_domain_config (void *buf, size_t len, libxl_domain_config *info)

4.76.1 Macro Definition Documentation

4.76.1.1 #define CHK_MEMCMP(a, b) {if (memcmp(&a, &b, sizeof(a))) goto error;}

4.76.1.2 #define CHK_STRCMP(a, b) {if (a && b && strcmp(a, b)) goto error;}

4.76.1.3 #define CHK_XLC_DARRAY_IN(a, b, c) {if ((rc=xlc_darray_in(a,b,c))) goto error;}

4.76.1.4 #define CHK_XLC_DARRAY_OUT(a, b, c) {if ((rc=xlc_darray_out(a,b,c))) goto error;}

4.76.1.5 #define CHK_XLC_DESERIALIZE_DISKS(a, b, c, d) {if ((rc=xlc_deserialize_disks(a,b,c,d))) goto error;}

4.76.1.6 #define CHK_XLC_DESERIALIZE_VIFS(a, b, c, d) {if ((rc=xlc_deserialize_vifs(a,b,c,d))) goto error;}

4.76.1.7 #define CHK_XLC_MEMCPY_IN(a, b, c, d) {if ((rc=xlc_memcpy_in(a,b,c,d))) goto error;}

4.76.1.8 #define CHK_XLC_MEMCPY_OUT(a, b, c, d) {if ((rc=xlc_memcpy_out(a,b,c,d))) goto error;}

4.76.1.9 #define CHK_XLC_SERIALIZE_DISKS(a, b, c, d) {if ((rc=xlc_serialize_disks(a,b,c,d))) goto error;}

4.76.1.10 #define CHK_XLC_SERIALIZE_VIFS(a, b, c, d) {if ((rc=xlc_serialize_vifs(a,b,c,d))) goto error;}

4.76.1.11 #define CHK_XLC_STRACPY_IN(a, b, c) {if ((rc=xlc_stracpy_in(a,b,c))) goto error;}

4.76.1.12 #define CHK_XLC_STRCPY_OUT(a, b, c) {if ((rc=xlc_strcpy_out(a,b,c))) goto error;}

4.76.2 Function Documentation

4.76.2.1 int deserialize_domain_build_info (void * *buf,* size_t *len,* libxl_domain_build_info * *info,* libxl_domain_build_state * *state*)

4.76.2.2 int deserialize_domain_config (void * *buf,* size_t *len,* **libxl_domain_config** * *info*)

4.76.2.3 int deserialize_domain_create_info (void * *buf,* size_t *len,* libxl_domain_create_info * *info*)

NB: strings are strdup()'ed instead of setting the pointer to the string in the shared page (which would waste a lot of memory). caller must be remember to free these strings struct on destruction.

4.76.2.4 int serialize_domain_build_info (void * *buf,* size_t *len,* libxl_domain_build_info * *info,* libxl_domain_build_state * *state*)

hvm not supported, nor is a pre-mapped image. (i.e., kernel.data and ramdisk.data are set to null!)

4.76.2.5 int serialize_domain_config (void * *buf,* size_t *len,* **libxl_domain_config** * *info*)

4.76.2.6 int serialize_domain_create_info (void * *buf,* size_t *len,* libxl_domain_create_info * *info*)

4.77 xen/tools/libxlc/xlc_util.c File Reference

4.78 xen/stubdom/domt/xlc_util.h File Reference

```
#include <stdlib.h>
#include <libxl.h>
#include <libxl_utils.h>
```

Functions

- int serialize_domain_create_info (void *buf, size_t len, libxl_domain_create_info *info)
- int deserialize_domain_create_info (void *buf, size_t len, libxl_domain_create_info *info)
- int serialize_domain_build_info (void *buf, size_t len, libxl_domain_build_info *info, libxl_domain_build_state *state)
- int deserialize_domain_build_info (void *buf, size_t len, libxl_domain_build_info *info, libxl_domain_build_state *state)
- int deserialize_domain_config (void *buf, size_t len, libxl_domain_config *info)
- int serialize_domain_config (void *buf, size_t len, libxl_domain_config *info)

4.78.1 Function Documentation

4.78.1.1 int deserialize_domain_build_info (void * *buf,* size_t *len,* libxl_domain_build_info * *info,* libxl_domain_build_state * *state*)

4.78.1.2 int deserialize_domain_config (void * *buf,* size_t *len,* **libxl_domain_config** * *info*)

4.78.1.3 int deserialize_domain_create_info (void * *buf,* size_t *len,* libxl_domain_create_info * *info*)

NB: strings are strdup()'ed instead of setting the pointer to the string in the shared page (which would waste a lot of memory). caller must be remember to free these strings struct on destruction.

4.78.1.4 int serialize_domain_build_info (void * *buf,* size_t *len,* libxl_domain_build_info * *info,* libxl_domain_build_state * *state*)

hvm not supported, nor is a pre-mapped image. (i.e., kernel.data and ramdisk.data are set to null!)

4.78.1.5 int serialize_domain_config (void * *buf,* size_t *len,* libxl_domain_config * *info*)

4.78.1.6 int serialize_domain_create_info (void * *buf,* size_t *len,* libxl_domain_create_info * *info*)

4.79 xen/tools/libxlc/xlc_util.h File Reference

4.80 xen/stubdom/vtpm/compat/big_endian.h File Reference

Macros

- #define __cpu_to_le64(x) (__swab64((x)))
- #define __le64_to_cpu(x) (__swab64((uint64_t)(x)))
- #define __cpu_to_le32(x) (__swab32((x)))
- #define __le32_to_cpu(x) (__swab32((uint32_t)(x)))
- #define __cpu_to_le16(x) (__swab16((x)))
- #define __le16_to_cpu(x) (__swab16((uint16_t)(x)))
- #define __cpu_to_be64(x) ((uint64_t)(x))
- #define __be64_to_cpu(x) ((uint64_t)(x))
- #define __cpu_to_be32(x) ((uint32_t)(x))
- #define __be32_to_cpu(x) ((uint32_t)(x))
- #define __cpu_to_be16(x) ((uint16_t)(x))
- #define __be16_to_cpu(x) ((uint16_t)(x))

4.80.1 Macro Definition Documentation

4.80.1.1 #define __be16_to_cpu(*x*) ((uint16_t)(x))

4.80.1.2 #define __be32_to_cpu(*x*) ((uint32_t)(x))

4.80.1.3 #define __be64_to_cpu(*x*) ((uint64_t)(x))

4.80.1.4 #define __cpu_to_be16(*x*) ((uint16_t)(x))

4.80.1.5 #define __cpu_to_be32(*x*) ((uint32_t)(x))

4.80.1.6 #define __cpu_to_be64(*x*) ((uint64_t)(x))

4.80.1.7 #define __cpu_to_le16(*x*) (__swab16((x)))

4.80.1.8 #define __cpu_to_le32(*x*) (__swab32((x)))

4.80.1.9 #define __cpu_to_le64(*x*) (__swab64((x)))

4.80.1.10 #define __le16_to_cpu(*x*) (__swab16((uint16_t)(x)))

4.80.1.11 #define __le32_to_cpu(*x*) (__swab32((uint32_t)(x)))

4.80.1.12 #define __le64_to_cpu(*x*) (__swab64((uint64_t)(x)))

4.81 xen/stubdom/vtpm/compat/endian_test.c File Reference

```
#include <stdio.h>
```

Functions

- int main (void)

4.81.1 Function Documentation

4.81.1.1 int main (void)

4.82 xen/stubdom/vtpm/compat/little_endian.h File Reference

Macros

- #define __cpu_to_le64(x) ((uint64_t)(x))
- #define __le64_to_cpu(x) ((uint64_t)(x))
- #define __cpu_to_le32(x) ((uint32_t)(x))
- #define __le32_to_cpu(x) ((uint32_t)(x))
- #define __cpu_to_le16(x) ((uint16_t)(x))
- #define __le16_to_cpu(x) ((uint16_t)(x))
- #define __cpu_to_be64(x) (__swab64((x)))
- #define __be64_to_cpu(x) (__swab64((uint64_t)(x)))
- #define __cpu_to_be32(x) (__swab32((x)))
- #define __be32_to_cpu(x) (__swab32((uint32_t)(x)))
- #define __cpu_to_be16(x) (__swab16((x)))
- #define __be16_to_cpu(x) (__swab16((uint16_t)(x)))

4.82.1 Macro Definition Documentation

4.82.1.1 #define __be16_to_cpu(x) (__swab16((uint16_t)(x)))

4.82.1.2 #define __be32_to_cpu(x) (__swab32((uint32_t)(x)))

4.82.1.3 #define __be64_to_cpu(x) (__swab64((uint64_t)(x)))

4.82.1.4 #define __cpu_to_be16(x) (__swab16((x)))

4.82.1.5 #define __cpu_to_be32(x) (__swab32((x)))

4.82.1.6 #define __cpu_to_be64(x) (__swab64((x)))

4.82.1.7 #define __cpu_to_le16(x) ((uint16_t)(x))

4.82.1.8 #define __cpu_to_le32(x) ((uint32_t)(x))

4.82.1.9 #define __cpu_to_le64(x) ((uint64_t)(x))

4.82.1.10 #define __le16_to_cpu(x) ((uint16_t)(x))

4.82.1.11 #define __le32_to_cpu(x) ((uint32_t)(x))

4.82.1.12 #define __le64_to_cpu(x) ((uint64_t)(x))

4.83 xen/stubdom/vtpm/compat/swab.h File Reference

```
#include <stdint.h>
```

Macros

- #define __swab16(x)
- #define __swab32(x)
- #define __swab64(x)

4.83.1 Macro Definition Documentation

4.83.1.1 #define __swab16(x)

Value:

```
((uint16_t)(                \
    (((uint16_t)(x) & (uint16_t)0x00ffU) << 8) |            \
    (((uint16_t)(x) & (uint16_t)0xff00U) >> 8)))
```

casts are necessary for constants, because we never know how for sure how U/UL/ULL map to __u16, __u32, __u64. At least not in a portable way.

4.83.1.2 #define __swab32(x)

Value:

```
((uint32_t)(                \
    (((uint32_t)(x) & (uint32_t)0x000000ffUL) << 24) |      \
    (((uint32_t)(x) & (uint32_t)0x0000ff00UL) <<  8) |      \
    (((uint32_t)(x) & (uint32_t)0x00ff0000UL) >>  8) |      \
    (((uint32_t)(x) & (uint32_t)0xff000000UL) >> 24)))
```

4.83.1.3 #define __swab64(x)

Value:

```
((uint64_t)(                \
    (((uint64_t)(x) & (uint64_t)0x00000000000000ffULL) << 56) | \
    (((uint64_t)(x) & (uint64_t)0x000000000000ff00ULL) << 40) | \
    (((uint64_t)(x) & (uint64_t)0x0000000000ff0000ULL) << 24) | \
    (((uint64_t)(x) & (uint64_t)0x00000000ff000000ULL) <<  8) | \
    (((uint64_t)(x) & (uint64_t)0x000000ff00000000ULL) >>  8) | \
    (((uint64_t)(x) & (uint64_t)0x0000ff0000000000ULL) >> 24) | \
    (((uint64_t)(x) & (uint64_t)0x00ff000000000000ULL) >> 40) | \
    (((uint64_t)(x) & (uint64_t)0xff00000000000000ULL) >> 56)))
```

4.84 xen/stubdom/vtpm/vtpm.c File Reference

```
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <syslog.h>
#include <errno.h>
#include <sys/time.h>
#include <xen/xen.h>
#include <tpmback.h>
#include <tpmfront.h>
#include "tpm_emulator_config.h"
#include "tpm/tpm_emulator.h"
#include "vtpm.h"
#include "vtpm_cmd.h"
#include "vtpmblk.h"
```

Functions

- void tpm_log (int priority, const char ∗fmt,...)
- void get_rand_bytes_rand (uint8_t ∗buf, size_t nbytes)
- void tpm_get_extern_random_bytes (void ∗buf, size_t nbytes)
- uint64_t tpm_get_ticks (void)
- int tpm_read_from_file (uint8_t ∗∗data, size_t ∗data_length)
- int tpm_write_to_file (uint8_t ∗data, size_t data_length)
- int init_random (void)
- int parse_cmd_line (int argc, char ∗∗argv)
- int main (int argc, char ∗∗argv)

Variables

- struct Opt_args opt_args
- struct tpmfront_dev ∗ tpmfront_dev

4.84.1 Function Documentation

4.84.1.1 void get_rand_bytes_rand (uint8_t ∗ *buf,* size_t *nbytes*)

4.84.1.2 int init_random (void)

4.84.1.3 int main (int *argc,* char ∗∗ *argv*)

4.84.1.4 int parse_cmd_line (int *argc,* char ∗∗ *argv*)

4.84.1.5 void tpm_get_extern_random_bytes (void ∗ *buf,* size_t *nbytes*)

4.84.1.6 uint64_t tpm_get_ticks (void)

4.84.1.7 void tpm_log (int *priority,* const char ∗ *fmt,* ...)

4.84.1.8 int tpm_read_from_file (uint8_t ∗∗ *data,* size_t ∗ *data_length*)

4.84.1.9 int tpm_write_to_file (uint8_t * *data*, size_t *data_length*)

4.84.2 Variable Documentation

4.84.2.1 struct Opt_args opt_args

Initial value:

```
= {
   .mode = MODE_PVM,
   .startup = ST_CLEAR,
   .loglevel = LOG_INFO,
   .entropysrc = ENT_TPM,
}
```

Global commandline options - default values

4.84.2.2 struct tpmfront_dev* tpmfront_dev

4.85 xen/stubdom/vtpm/vtpm.h File Reference

Data Structures

- struct Opt_args

Macros

- #define VERS_CMD "\x00\xC1\x00\x00\x00\x16\x00\x00\x00\x65\x00\x00\x00\x05\x00\x00\x00\x04\x00\x00\x01\x03"
- #define VERS_CMD_LEN 22

Variables

- struct Opt_args opt_args

4.85.1 Macro Definition Documentation

4.85.1.1 #define VERS_CMD "\x00\xC1\x00\x00\x00\x16\x00\x00\x00\x65\x00\x00\x00\x05\x00\x00\x00\x04\x00\x00\x01\x03"

For testing

4.85.1.2 #define VERS_CMD_LEN 22

4.85.2 Variable Documentation

4.85.2.1 struct Opt_args opt_args

Global commandline options - default values

4.86 xen/stubdom/vtpm/vtpm_cmd.c File Reference

```
#include <types.h>
#include <xen/xen.h>
#include <mm.h>
#include <gnttab.h>
#include "tpm/tpm_marshalling.h"
#include "vtpm_manager.h"
#include "vtpm_cmd.h"
```

Data Structures

- struct shpage

Macros

- #define TRYFAILGOTO(C)
- #define TRYFAILGOTOMSG(C, msg)
- #define CHECKSTATUSGOTO(ret, fname)
- #define ERR_MALFORMED "Malformed response from backend"
- #define ERR_TPMFRONT "Error sending command through frontend device"

Typedefs

- typedef struct shpage shpage_t

Functions

- TPM_RESULT VTPM_GetRandom (struct tpmfront_dev *tpmfront_dev, BYTE *bytes, UINT32 numbytes)
- TPM_RESULT VTPM_LoadKey (struct tpmfront_dev *tpmfront_dev, uint8_t **data, size_t *data_length)
- TPM_RESULT VTPM_SaveKey (struct tpmfront_dev *tpmfront_dev, uint8_t *data, size_t data_length)

4.86.1 Macro Definition Documentation

4.86.1.1 #define CHECKSTATUSGOTO(*ret, fname*)

Value:

```
if((ret) != TPM_SUCCESS) { \
      error("%s failed with error code (%lu)", fname, (unsigned long) ret); \
      status = ord; \
      goto abort_egress; \
   }
```

4.86.1.2 #define ERR_MALFORMED "Malformed response from backend"

4.86.1.3 #define ERR_TPMFRONT "Error sending command through frontend device"

4.86.1.4 #define TRYFAILGOTO(*C*)

Value:

```
if((C)) { \
      status = TPM_FAIL; \
      goto abort_egress; \
   }
```

4.86.1.5 #define TRYFAILGOTOMSG(*C, msg*)

Value:

```
if((C)) { \
      status = TPM_FAIL; \
      error(msg); \
      goto abort_egress; \
   }
```

4.86.2 Typedef Documentation

4.86.2.1 typedef struct shpage shpage_t

4.86.3 Function Documentation

4.86.3.1 TPM_RESULT VTPM_GetRandom (struct tpmfront_dev * *tpmfront_dev,* BYTE * *bytes,* UINT32 *numbytes*)

Request random bytes from hardware tpm, returns 0 on success

4.86.3.2 TPM_RESULT VTPM_LoadKey (struct tpmfront_dev * *tpmfront_dev,* uint8_t ** *data,* size_t * *data_length*)

Retreive 256 bit AES encryption key from manager

4.86.3.3 TPM_RESULT VTPM_SaveKey (struct tpmfront_dev * *tpmfront_dev,* uint8_t * *data,* size_t *data_length*)

Manager securely saves our 256 bit AES encryption key

4.87 xen/stubdom/vtpm/vtpm_cmd.h File Reference

```
#include <tpmfront.h>
#include "tpm_emulator_config.h"
#include "tpm/tpm_emulator.h"
#include "tpm/tpm_structures.h"
```

Functions

- TPM_RESULT VTPM_GetRandom (struct tpmfront_dev *tpmfront_dev, BYTE *bytes, UINT32 numbytes)
- TPM_RESULT VTPM_SaveNVM (struct tpmfront_dev *tpmfront_dev, BYTE *data, size_t data_length)
- TPM_RESULT VTPM_LoadNVM (struct tpmfront_dev *tpmfront_dev, BYTE **data, size_t *data_length)
- TPM_RESULT VTPM_GetNVMSize (struct tpmfront_dev *tpmfront_dev, UINT32 *size)
- TPM_RESULT VTPM_LoadKey (struct tpmfront_dev *tpmfront_dev, uint8_t **data, size_t *data_length)
- TPM_RESULT VTPM_SaveKey (struct tpmfront_dev *tpmfront_dev, uint8_t *data, size_t data_length)

4.87.1 Function Documentation

4.87.1.1 TPM_RESULT VTPM_GetNVMSize (struct tpmfront_dev * *tpmfront_dev*, UINT32 * *size*)

Retreive the size of the NVM data

4.87.1.2 TPM_RESULT VTPM_GetRandom (struct tpmfront_dev * *tpmfront_dev*, BYTE * *bytes*, UINT32 *numbytes*)

Request random bytes from hardware tpm, returns 0 on success

4.87.1.3 TPM_RESULT VTPM_LoadKey (struct tpmfront_dev * *tpmfront_dev*, uint8_t ** *data*, size_t * *data_length*)

Retreive 256 bit AES encryption key from manager

4.87.1.4 TPM_RESULT VTPM_LoadNVM (struct tpmfront_dev * *tpmfront_dev*, BYTE ** *data*, size_t * *data_length*)

Retreive NVM data from manager

4.87.1.5 TPM_RESULT VTPM_SaveKey (struct tpmfront_dev * *tpmfront_dev*, uint8_t * *data*, size_t *data_length*)

Manager securely saves our 256 bit AES encryption key

4.87.1.6 TPM_RESULT VTPM_SaveNVM (struct tpmfront_dev * *tpmfront_dev*, BYTE * *data*, size_t *data_length*)

Send NVM data to manager for secure storage

4.88 xen/stubdom/vtpm/vtpmblk.c File Reference

```
#include <blkfront.h>
#include <fcntl.h>
#include <xmalloc.h>
#include <asm/byteorder.h>
#include "vtpmblk.h"
#include "tpm_emulator_config.h"
#include "tpm/tpm_marshalling.h"
#include "vtpm_cmd.h"
#include "polarssl/aes.h"
```

Macros

- #define SECTORS_PER_BLOCK 8
- #define BLKSZ 16
- #define KEYSZ 32
- #define write_finish() rwop_finish(1)
- #define read_finish() rwop_finish(0)
- #define write_data(data, len) rw_data(data, len, 1)
- #define read_data(data, len) rw_data(data, len, 0)

Functions

- int init_vtpmblk (struct tpmfront_dev *tpmfront_dev)
- void shutdown_vtpmblk (void)
- int write_vtpmblk_raw (uint8_t *data, size_t data_length)
- int read_vtpmblk_raw (uint8_t **data, size_t *data_length)
- int encrypt_vtpmblk (uint8_t *clear, size_t clear_len, uint8_t **cipher, size_t *cipher_len, uint8_t **symkey)
- int decrypt_vtpmblk (uint8_t *cipher, size_t cipher_len, uint8_t **clear, size_t *clear_len, uint8_t *symkey)
- int write_vtpmblk (struct tpmfront_dev *tpmfront_dev, uint8_t *data, size_t data_length)
- int read_vtpmblk (struct tpmfront_dev *tpmfront_dev, uint8_t **data, size_t *data_length)

Variables

- struct blkfront_dev * blkdev = NULL
- struct blkfront_info blkinfo
- struct blkfront_aiocb aiocb
- unsigned long long disksize = 0
- unsigned int blocksize
- off_t blkoff

4.88.1 Macro Definition Documentation

4.88.1.1 #define BLKSZ 16

Encryption key and block sizes

4.88.1.2 #define KEYSZ 32

4.88.1.3 #define read_data(*data, len*) rw_data(data, len, 0)

4.88.1.4 #define read_finish( ) rwop_finish(0)

4.88.1.5 #define SECTORS_PER_BLOCK 8

4.88.1.6 #define write_data(*data, len*) rw_data(data, len, 1)

4.88.1.7 #define write_finish( ) rwop_finish(1)

4.88.2 Function Documentation

4.88.2.1 int decrypt_vtpmblk (uint8_t * *cipher,* size_t *cipher_len,* uint8_t ** *clear,* size_t * *clear_len,* uint8_t * *symkey*)

4.88.2.2 int encrypt_vtpmblk (uint8_t * *clear,* size_t *clear_len,* uint8_t ** *cipher,* size_t * *cipher_len,* uint8_t ** *symkey*)

4.88.2.3 int init_vtpmblk (struct tpmfront_dev * *tpmfront_dev*)

4.88.2.4 int read_vtpmblk (struct tpmfront_dev * *tpmfront_dev,* uint8_t ** *data,* size_t * *data_length*)

Reads, Decrypts, and returns data from blk device

4.88.2.5 int read_vtpmblk_raw (uint8_t ** *data*, size_t * *data_length*)

4.88.2.6 void shutdown_vtpmblk (void)

4.88.2.7 int write_vtpmblk (struct **tpmfront_dev** * *tpmfront_dev*, uint8_t * *data*, size_t *data_length*)

Encrypts and writes data to blk device

4.88.2.8 int write_vtpmblk_raw (uint8_t * *data*, size_t *data_length*)

4.88.3 Variable Documentation

4.88.3.1 struct **blkfront_aiocb** aiocb

4.88.3.2 struct **blkfront_dev*** blkdev = NULL

4.88.3.3 struct **blkfront_info** blkinfo

4.88.3.4 off_t blkoff

4.88.3.5 unsigned int blocksize

4.88.3.6 unsigned long long disksize = 0

4.89 xen/stubdom/vtpm/vtpmblk.h File Reference

```
#include <mini-os/types.h>
#include <tpmfront.h>
```

Functions

- int init_vtpmblk (struct tpmfront_dev *tpmfront_dev)
- void shutdown_vtpmblk (void)
- int write_vtpmblk (struct tpmfront_dev *tpmfront_dev, uint8_t *data, size_t data_length)
- int read_vtpmblk (struct tpmfront_dev *tpmfront_dev, uint8_t **data, size_t *data_length)

4.89.1 Function Documentation

4.89.1.1 int init_vtpmblk (struct **tpmfront_dev** * *tpmfront_dev*)

4.89.1.2 int read_vtpmblk (struct **tpmfront_dev** * *tpmfront_dev*, uint8_t ** *data*, size_t * *data_length*)

Reads, Decrypts, and returns data from blk device

4.89.1.3 void shutdown_vtpmblk (void)

4.89.1.4 int write_vtpmblk (struct **tpmfront_dev** * *tpmfront_dev*, uint8_t * *data*, size_t *data_length*)

Encrypts and writes data to blk device

4.90 xen/tools/console/daemon/io.c File Reference

```
#include "utils.h"
#include "io.h"
#include <xs.h>
#include <xen/io/console.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <sys/select.h>
#include <fcntl.h>
#include <unistd.h>
#include <termios.h>
#include <stdarg.h>
#include <sys/mman.h>
#include <time.h>
#include <assert.h>
```

Data Structures

- struct buffer
- struct domain

Macros

- #define _GNU_SOURCE
- #define MAX(a, b) (((a) > (b)) ? (a) : (b))
- #define MIN(a, b) (((a) < (b)) ? (a) : (b))
- #define MAX_STRLEN(x) ((sizeof(x) ∗ CHAR_BIT + CHAR_BIT-1) / 10 ∗ 3 + 2)
- #define RATE_LIMIT_ALLOWANCE 30
- #define RATE_LIMIT_PERIOD 200
- #define DOMID_DOMT 1

Functions

- void enum_domains (void)
- void handle_io (void)

Variables

- int log_reload
- int log_guest
- int log_hv
- int log_time_hv
- int log_time_guest
- char ∗ log_dir
- int discard_overflowed_data

4.90.1 Macro Definition Documentation

4.90.1.1 #define _GNU_SOURCE

4.90.1.2 #define DOMID_DOMT 1

4.90.1.3 #define MAX(a, b) (((a) > (b)) ? (a) : (b))

4.90.1.4 #define MAX_STRLEN(x) ((sizeof(x) * CHAR_BIT + CHAR_BIT-1) / 10 * 3 + 2)

4.90.1.5 #define MIN(a, b) (((a) < (b)) ? (a) : (b))

4.90.1.6 #define RATE_LIMIT_ALLOWANCE 30

4.90.1.7 #define RATE_LIMIT_PERIOD 200

4.90.2 Function Documentation

4.90.2.1 void enum_domains (void)

4.90.2.2 void handle_io (void)

4.90.3 Variable Documentation

4.90.3.1 int discard_overflowed_data

4.90.3.2 char* log_dir

4.90.3.3 int log_guest

4.90.3.4 int log_hv

4.90.3.5 int log_reload

4.90.3.6 int log_time_guest

4.90.3.7 int log_time_hv

4.91 xen/tools/debugger/kdd/.kdd-xen.o.d File Reference

4.92 xen/tools/debugger/kdd/.kdd.o.d File Reference

4.93 xen/tools/ioemu-dir/alpha-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-alpha"
- #define TARGET_ARCH "alpha"
- #define TARGET_ALPHA 1
- #define CONFIG_USER_ONLY 1

- #define CONFIG_LINUX_USER 1

4.93.1 Macro Definition Documentation

4.93.1.1 #define CONFIG_LINUX_USER 1

4.93.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-alpha"

4.93.1.3 #define CONFIG_USER_ONLY 1

4.93.1.4 #define TARGET_ALPHA 1

4.93.1.5 #define TARGET_ARCH "alpha"

4.94 xen/tools/ioemu-dir/arm-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-arm"
- #define TARGET_ARCH "arm"
- #define TARGET_ARM 1
- #define CONFIG_USER_ONLY 1
- #define CONFIG_LINUX_USER 1
- #define CONFIG_SOFTFLOAT 1
- #define TARGET_HAS_BFLT 1

4.94.1 Macro Definition Documentation

4.94.1.1 #define CONFIG_LINUX_USER 1

4.94.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-arm"

4.94.1.3 #define CONFIG_SOFTFLOAT 1

4.94.1.4 #define CONFIG_USER_ONLY 1

4.94.1.5 #define TARGET_ARCH "arm"

4.94.1.6 #define TARGET_ARM 1

4.94.1.7 #define TARGET_HAS_BFLT 1

4.95 xen/tools/ioemu-dir/arm-softmmu/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-arm"
- #define TARGET_ARCH "arm"
- #define TARGET_ARM 1
- #define CONFIG_SOFTMMU 1
- #define CONFIG_SOFTFLOAT 1

4.95.1 Macro Definition Documentation

4.95.1.1 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-arm"

4.95.1.2 #define CONFIG_SOFTFLOAT 1

4.95.1.3 #define CONFIG_SOFTMMU 1

4.95.1.4 #define TARGET_ARCH "arm"

4.95.1.5 #define TARGET_ARM 1

4.96 xen/tools/ioemu-dir/armeb-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-armeb"
- #define TARGET_ARCH "arm"
- #define TARGET_ARM 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_USER_ONLY 1
- #define CONFIG_LINUX_USER 1
- #define CONFIG_SOFTFLOAT 1
- #define TARGET_HAS_BFLT 1

4.96.1 Macro Definition Documentation

4.96.1.1 #define CONFIG_LINUX_USER 1

4.96.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-armeb"

4.96.1.3 #define CONFIG_SOFTFLOAT 1

4.96.1.4 #define CONFIG_USER_ONLY 1

4.96.1.5 #define TARGET_ARCH "arm"

4.96.1.6 #define TARGET_ARM 1

4.96.1.7 #define TARGET_HAS_BFLT 1

4.96.1.8 #define TARGET_WORDS_BIGENDIAN 1

4.97 xen/tools/ioemu-dir/cris-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-cris"
- #define TARGET_ARCH "cris"
- #define TARGET_CRIS 1
- #define CONFIG_USER_ONLY 1
- #define CONFIG_LINUX_USER 1

4.97.1 Macro Definition Documentation

4.97.1.1 #define CONFIG_LINUX_USER 1

4.97.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-cris"

4.97.1.3 #define CONFIG_USER_ONLY 1

4.97.1.4 #define TARGET_ARCH "cris"

4.97.1.5 #define TARGET_CRIS 1

4.98 xen/tools/ioemu-dir/cris-softmmu/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-cris"
- #define TARGET_ARCH "cris"
- #define TARGET_CRIS 1
- #define CONFIG_SOFTMMU 1

4.98.1 Macro Definition Documentation

4.98.1.1 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-cris"

4.98.1.2 #define CONFIG_SOFTMMU 1

4.98.1.3 #define TARGET_ARCH "cris"

4.98.1.4 #define TARGET_CRIS 1

4.99 xen/tools/ioemu-dir/i386-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-i386"
- #define TARGET_ARCH "i386"
- #define TARGET_I386 1
- #define CONFIG_USER_ONLY 1
- #define CONFIG_LINUX_USER 1

4.99.1 Macro Definition Documentation

4.99.1.1 #define CONFIG_LINUX_USER 1

4.99.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-i386"

4.99.1.3 #define CONFIG_USER_ONLY 1

4.99.1.4 #define TARGET_ARCH "i386"

4.99.1.5 #define TARGET_I386 1

4.100 xen/tools/ioemu-dir/i386-softmmu/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-i386"
- #define TARGET_ARCH "i386"
- #define TARGET_I386 1
- #define USE_KQEMU 1
- #define CONFIG_KVM 1
- #define CONFIG_SOFTMMU 1

4.100.1 Macro Definition Documentation

4.100.1.1 #define CONFIG_KVM 1

4.100.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-i386"

4.100.1.3 #define CONFIG_SOFTMMU 1

4.100.1.4 #define TARGET_ARCH "i386"

4.100.1.5 #define TARGET_I386 1

4.100.1.6 #define USE_KQEMU 1

4.101 xen/tools/ioemu-dir/m68k-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-m68k"
- #define TARGET_ARCH "m68k"
- #define TARGET_M68K 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_USER_ONLY 1
- #define CONFIG_LINUX_USER 1
- #define CONFIG_SOFTFLOAT 1
- #define TARGET_HAS_BFLT 1

4.101.1 Macro Definition Documentation

4.101.1.1 #define CONFIG_LINUX_USER 1

4.101.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-m68k"

4.101.1.3 #define CONFIG_SOFTFLOAT 1

4.101.1.4 #define CONFIG_USER_ONLY 1

4.101.1.5 #define TARGET_ARCH "m68k"

4.101.1.6 #define TARGET_HAS_BFLT 1

4.101.1.7 #define TARGET_M68K 1

4.101.1.8 #define TARGET_WORDS_BIGENDIAN 1

4.102 xen/tools/ioemu-dir/m68k-softmmu/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-m68k"
- #define TARGET_ARCH "m68k"
- #define TARGET_M68K 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_SOFTMMU 1
- #define CONFIG_SOFTFLOAT 1

4.102.1 Macro Definition Documentation

4.102.1.1 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-m68k"

4.102.1.2 #define CONFIG_SOFTFLOAT 1

4.102.1.3 #define CONFIG_SOFTMMU 1

4.102.1.4 #define TARGET_ARCH "m68k"

4.102.1.5 #define TARGET_M68K 1

4.102.1.6 #define TARGET_WORDS_BIGENDIAN 1

4.103 xen/tools/ioemu-dir/mips-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-mips"
- #define TARGET_ARCH "mips"
- #define TARGET_MIPS 1
- #define TARGET_ABI_MIPSO32 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_USER_ONLY 1
- #define CONFIG_LINUX_USER 1
- #define CONFIG_SOFTFLOAT 1

4.103.1 Macro Definition Documentation

4.103.1.1 #define CONFIG_LINUX_USER 1

4.103.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-mips"

4.103.1.3 #define CONFIG_SOFTFLOAT 1

4.103.1.4 #define CONFIG_USER_ONLY 1

4.103.1.5 #define TARGET_ABI_MIPSO32 1

4.103.1.6 #define TARGET_ARCH "mips"

4.103.1.7 #define TARGET_MIPS 1

4.103.1.8 #define TARGET_WORDS_BIGENDIAN 1

4.104 xen/tools/ioemu-dir/mips-softmmu/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-mips"
- #define TARGET_ARCH "mips"
- #define TARGET_MIPS 1
- #define TARGET_ABI_MIPSO32 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_SOFTMMU 1
- #define CONFIG_SOFTFLOAT 1

4.104.1 Macro Definition Documentation

4.104.1.1 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-mips"

4.104.1.2 #define CONFIG_SOFTFLOAT 1

4.104.1.3 #define CONFIG_SOFTMMU 1

4.104.1.4 #define TARGET_ABI_MIPSO32 1

4.104.1.5 #define TARGET_ARCH "mips"

4.104.1.6 #define TARGET_MIPS 1

4.104.1.7 #define TARGET_WORDS_BIGENDIAN 1

4.105 xen/tools/ioemu-dir/mips64-softmmu/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-mips64"
- #define TARGET_ARCH "mips64"
- #define TARGET_MIPS 1
- #define TARGET_MIPS64 1
- #define TARGET_ABI_MIPSN64 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_SOFTMMU 1
- #define CONFIG_SOFTFLOAT 1

4.105.1 Macro Definition Documentation

4.105.1.1 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-mips64"

4.105.1.2 #define CONFIG_SOFTFLOAT 1

4.105.1.3 #define CONFIG_SOFTMMU 1

4.105.1.4 #define TARGET_ABI_MIPSN64 1

4.105.1.5 #define TARGET_ARCH "mips64"

4.105.1.6 #define TARGET_MIPS 1

4.105.1.7 #define TARGET_MIPS64 1

4.105.1.8 #define TARGET_WORDS_BIGENDIAN 1

4.106 xen/tools/ioemu-dir/mips64el-softmmu/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-mips64el"
- #define TARGET_ARCH "mips64"
- #define TARGET_MIPS 1
- #define TARGET_MIPS64 1
- #define TARGET_ABI_MIPSN64 1
- #define CONFIG_SOFTMMU 1
- #define CONFIG_SOFTFLOAT 1

4.106.1 Macro Definition Documentation

4.106.1.1 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-mips64el"

4.106.1.2 #define CONFIG_SOFTFLOAT 1

4.106.1.3 #define CONFIG_SOFTMMU 1

4.106.1.4 #define TARGET_ABI_MIPSN64 1

4.106.1.5 #define TARGET_ARCH "mips64"

4.106.1.6 #define TARGET_MIPS 1

4.106.1.7 #define TARGET_MIPS64 1

4.107 xen/tools/ioemu-dir/mipsel-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-mipsel"
- #define TARGET_ARCH "mips"
- #define TARGET_MIPS 1
- #define TARGET_ABI_MIPSO32 1
- #define CONFIG_USER_ONLY 1
- #define CONFIG_LINUX_USER 1
- #define CONFIG_SOFTFLOAT 1

4.107.1 Macro Definition Documentation

4.107.1.1 #define CONFIG_LINUX_USER 1

4.107.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-mipsel"

4.107.1.3 #define CONFIG_SOFTFLOAT 1

4.107.1.4 #define CONFIG_USER_ONLY 1

4.107.1.5 #define TARGET_ABI_MIPSO32 1

4.107.1.6 #define TARGET_ARCH "mips"

4.107.1.7 #define TARGET_MIPS 1

4.108 xen/tools/ioemu-dir/mipsel-softmmu/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-mipsel"
- #define TARGET_ARCH "mips"
- #define TARGET_MIPS 1
- #define TARGET_ABI_MIPSO32 1
- #define CONFIG_SOFTMMU 1
- #define CONFIG_SOFTFLOAT 1

4.108.1 Macro Definition Documentation

4.108.1.1 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-mipsel"

4.108.1.2 #define CONFIG_SOFTFLOAT 1

4.108.1.3 #define CONFIG_SOFTMMU 1

4.108.1.4 #define TARGET_ABI_MIPSO32 1

4.108.1.5 #define TARGET_ARCH "mips"

4.108.1.6 #define TARGET_MIPS 1

4.109 xen/tools/ioemu-dir/ppc-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-ppc"
- #define TARGET_ARCH "ppc"
- #define TARGET_PPC 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_USER_ONLY 1
- #define CONFIG_LINUX_USER 1
- #define CONFIG_SOFTFLOAT 1

4.109.1 Macro Definition Documentation

4.109.1.1 #define CONFIG_LINUX_USER 1

4.109.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-ppc"

4.109.1.3 #define CONFIG_SOFTFLOAT 1

4.109.1.4 #define CONFIG_USER_ONLY 1

4.109.1.5 #define TARGET_ARCH "ppc"

4.109.1.6 #define TARGET_PPC 1

4.109.1.7 #define TARGET_WORDS_BIGENDIAN 1

4.110 xen/tools/ioemu-dir/ppc-softmmu/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-ppc"
- #define TARGET_ARCH "ppc"
- #define TARGET_PPC 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_SOFTMMU 1
- #define CONFIG_SOFTFLOAT 1

4.110.1 Macro Definition Documentation

4.110.1.1 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-ppc"

4.110.1.2 #define CONFIG_SOFTFLOAT 1

4.110.1.3 #define CONFIG_SOFTMMU 1

4.110.1.4 #define TARGET_ARCH "ppc"

4.110.1.5 #define TARGET_PPC 1

4.110.1.6 #define TARGET_WORDS_BIGENDIAN 1

4.111 xen/tools/ioemu-dir/ppc64-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-ppc64"
- #define TARGET_ARCH "ppc64"
- #define TARGET_PPC 1
- #define TARGET_PPC64 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_USER_ONLY 1

- #define CONFIG_LINUX_USER 1
- #define CONFIG_SOFTFLOAT 1

4.111.1 Macro Definition Documentation

4.111.1.1 #define CONFIG_LINUX_USER 1

4.111.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-ppc64"

4.111.1.3 #define CONFIG_SOFTFLOAT 1

4.111.1.4 #define CONFIG_USER_ONLY 1

4.111.1.5 #define TARGET_ARCH "ppc64"

4.111.1.6 #define TARGET_PPC 1

4.111.1.7 #define TARGET_PPC64 1

4.111.1.8 #define TARGET_WORDS_BIGENDIAN 1

4.112 xen/tools/ioemu-dir/ppc64-softmmu/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-ppc64"
- #define TARGET_ARCH "ppc64"
- #define TARGET_PPC 1
- #define TARGET_PPC64 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_SOFTMMU 1
- #define CONFIG_SOFTFLOAT 1

4.112.1 Macro Definition Documentation

4.112.1.1 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-ppc64"

4.112.1.2 #define CONFIG_SOFTFLOAT 1

4.112.1.3 #define CONFIG_SOFTMMU 1

4.112.1.4 #define TARGET_ARCH "ppc64"

4.112.1.5 #define TARGET_PPC 1

4.112.1.6 #define TARGET_PPC64 1

4.112.1.7 #define TARGET_WORDS_BIGENDIAN 1

4.113 xen/tools/ioemu-dir/ppc64abi32-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-ppc64abi32"
- #define TARGET_ARCH "ppc64"
- #define TARGET_PPC 1
- #define TARGET_PPC64 1
- #define TARGET_ABI32 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_USER_ONLY 1
- #define CONFIG_LINUX_USER 1
- #define CONFIG_SOFTFLOAT 1

4.113.1 Macro Definition Documentation

4.113.1.1 #define CONFIG_LINUX_USER 1

4.113.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-ppc64abi32"

4.113.1.3 #define CONFIG_SOFTFLOAT 1

4.113.1.4 #define CONFIG_USER_ONLY 1

4.113.1.5 #define TARGET_ABI32 1

4.113.1.6 #define TARGET_ARCH "ppc64"

4.113.1.7 #define TARGET_PPC 1

4.113.1.8 #define TARGET_PPC64 1

4.113.1.9 #define TARGET_WORDS_BIGENDIAN 1

4.114 xen/tools/ioemu-dir/ppcemb-softmmu/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-ppcemb"
- #define TARGET_ARCH "ppcemb"
- #define TARGET_PPC 1
- #define TARGET_PPCEMB 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_SOFTMMU 1
- #define CONFIG_SOFTFLOAT 1

4.114.1 Macro Definition Documentation

4.114.1.1 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-ppcemb"

4.114.1.2 #define CONFIG_SOFTFLOAT 1

4.114.1.3 #define CONFIG_SOFTMMU 1

4.114.1.4 #define TARGET_ARCH "ppcemb"

4.114.1.5 #define TARGET_PPC 1

4.114.1.6 #define TARGET_PPCEMB 1

4.114.1.7 #define TARGET_WORDS_BIGENDIAN 1

4.115 xen/tools/ioemu-dir/sh4-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-sh4"
- #define TARGET_ARCH "sh4"
- #define TARGET_SH4 1
- #define CONFIG_USER_ONLY 1
- #define CONFIG_LINUX_USER 1
- #define TARGET_HAS_BFLT 1

4.115.1 Macro Definition Documentation

4.115.1.1 #define CONFIG_LINUX_USER 1

4.115.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-sh4"

4.115.1.3 #define CONFIG_USER_ONLY 1

4.115.1.4 #define TARGET_ARCH "sh4"

4.115.1.5 #define TARGET_HAS_BFLT 1

4.115.1.6 #define TARGET_SH4 1

4.116 xen/tools/ioemu-dir/sh4-softmmu/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-sh4"
- #define TARGET_ARCH "sh4"

- #define TARGET_SH4 1
- #define CONFIG_SOFTMMU 1

4.116.1 Macro Definition Documentation

4.116.1.1 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-sh4"

4.116.1.2 #define CONFIG_SOFTMMU 1

4.116.1.3 #define TARGET_ARCH "sh4"

4.116.1.4 #define TARGET_SH4 1

4.117 xen/tools/ioemu-dir/sh4eb-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-sh4eb"
- #define TARGET_ARCH "sh4"
- #define TARGET_SH4 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_USER_ONLY 1
- #define CONFIG_LINUX_USER 1
- #define TARGET_HAS_BFLT 1

4.117.1 Macro Definition Documentation

4.117.1.1 #define CONFIG_LINUX_USER 1

4.117.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-sh4eb"

4.117.1.3 #define CONFIG_USER_ONLY 1

4.117.1.4 #define TARGET_ARCH "sh4"

4.117.1.5 #define TARGET_HAS_BFLT 1

4.117.1.6 #define TARGET_SH4 1

4.117.1.7 #define TARGET_WORDS_BIGENDIAN 1

4.118 xen/tools/ioemu-dir/sh4eb-softmmu/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-sh4eb"
- #define TARGET_ARCH "sh4"
- #define TARGET_SH4 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_SOFTMMU 1

4.118.1 Macro Definition Documentation

4.118.1.1 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-sh4eb"

4.118.1.2 #define CONFIG_SOFTMMU 1

4.118.1.3 #define TARGET_ARCH "sh4"

4.118.1.4 #define TARGET_SH4 1

4.118.1.5 #define TARGET_WORDS_BIGENDIAN 1

4.119 xen/tools/ioemu-dir/sparc-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-sparc"
- #define TARGET_ARCH "sparc"
- #define TARGET_SPARC 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_USER_ONLY 1
- #define CONFIG_LINUX_USER 1
- #define CONFIG_SOFTFLOAT 1

4.119.1 Macro Definition Documentation

4.119.1.1 #define CONFIG_LINUX_USER 1

4.119.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-sparc"

4.119.1.3 #define CONFIG_SOFTFLOAT 1

4.119.1.4 #define CONFIG_USER_ONLY 1

4.119.1.5 #define TARGET_ARCH "sparc"

4.119.1.6 #define TARGET_SPARC 1

4.119.1.7 #define TARGET_WORDS_BIGENDIAN 1

4.120 xen/tools/ioemu-dir/sparc-softmmu/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-sparc"
- #define TARGET_ARCH "sparc"
- #define TARGET_SPARC 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_SOFTMMU 1
- #define CONFIG_SOFTFLOAT 1

4.120.1 Macro Definition Documentation

4.120.1.1 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-sparc"

4.120.1.2 #define CONFIG_SOFTFLOAT 1

4.120.1.3 #define CONFIG_SOFTMMU 1

4.120.1.4 #define TARGET_ARCH "sparc"

4.120.1.5 #define TARGET_SPARC 1

4.120.1.6 #define TARGET_WORDS_BIGENDIAN 1

4.121 xen/tools/ioemu-dir/sparc32plus-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-sparc32plus"
- #define TARGET_ARCH "sparc64"
- #define TARGET_SPARC 1
- #define TARGET_SPARC64 1
- #define TARGET_ABI32 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_USER_ONLY 1
- #define CONFIG_LINUX_USER 1
- #define CONFIG_SOFTFLOAT 1

4.121.1 Macro Definition Documentation

4.121.1.1 #define CONFIG_LINUX_USER 1

4.121.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-sparc32plus"

4.121.1.3 #define CONFIG_SOFTFLOAT 1

4.121.1.4 #define CONFIG_USER_ONLY 1

4.121.1.5 #define TARGET_ABI32 1

4.121.1.6 #define TARGET_ARCH "sparc64"

4.121.1.7 #define TARGET_SPARC 1

4.121.1.8 #define TARGET_SPARC64 1

4.121.1.9 #define TARGET_WORDS_BIGENDIAN 1

4.122 xen/tools/ioemu-dir/sparc64-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-sparc64"
- #define TARGET_ARCH "sparc64"
- #define TARGET_SPARC 1
- #define TARGET_SPARC64 1
- #define TARGET_WORDS_BIGENDIAN 1
- #define CONFIG_USER_ONLY 1
- #define CONFIG_LINUX_USER 1
- #define CONFIG_SOFTFLOAT 1
- #define TARGET_HAS_ELFLOAD32 1

4.122.1 Macro Definition Documentation

4.122.1.1 #define CONFIG_LINUX_USER 1

4.122.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-sparc64"

4.122.1.3 #define CONFIG_SOFTFLOAT 1

4.122.1.4 #define CONFIG_USER_ONLY 1

4.122.1.5 #define TARGET_ARCH "sparc64"

4.122.1.6 #define TARGET_HAS_ELFLOAD32 1

4.122.1.7 #define TARGET_SPARC 1

4.122.1.8 #define TARGET_SPARC64 1

4.122.1.9 #define TARGET_WORDS_BIGENDIAN 1

4.123 xen/tools/ioemu-dir/x86_64-linux-user/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-x86_64"
- #define TARGET_ARCH "x86_64"
- #define TARGET_I386 1
- #define TARGET_X86_64 1
- #define CONFIG_USER_ONLY 1
- #define CONFIG_LINUX_USER 1

4.123.1 Macro Definition Documentation

4.123.1.1 #define CONFIG_LINUX_USER 1

4.123.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-x86_64"

4.123.1.3 #define CONFIG_USER_ONLY 1

4.123.1.4 #define TARGET_ARCH "x86_64"

4.123.1.5 #define TARGET_I386 1

4.123.1.6 #define TARGET_X86_64 1

4.124 xen/tools/ioemu-dir/x86_64-softmmu/config.h File Reference

```
#include "../config-host.h"
```

Macros

- #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-x86_64"
- #define TARGET_ARCH "x86_64"
- #define TARGET_I386 1
- #define TARGET_X86_64 1
- #define USE_KQEMU 1
- #define CONFIG_KVM 1
- #define CONFIG_SOFTMMU 1

4.124.1 Macro Definition Documentation

4.124.1.1 #define CONFIG_KVM 1

4.124.1.2 #define CONFIG_QEMU_PREFIX "/usr/gnemul/qemu-x86_64"

4.124.1.3 #define CONFIG_SOFTMMU 1

4.124.1.4 #define TARGET_ARCH "x86_64"

4.124.1.5 #define TARGET_I386 1

4.124.1.6 #define TARGET_X86_64 1

4.124.1.7 #define USE_KQEMU 1

4.125 xen/tools/ioemu-dir/config-host.h File Reference

```
#include "xen-config-host.h"
```

Macros

- #define CONFIG_QEMU_SHAREDIR "/usr/share/xen/qemu"
- #define HOST_X86_64 1
- #define HOST_LONG_BITS 64
- #define HAVE_BYTESWAP_H 1
- #define CONFIG_AC97 1
- #define CONFIG_ES1370 1
- #define CONFIG_SB16 1
- #define AUDIO_DRIVERS &oss_audio_driver, \
- #define CONFIG_VNC_TLS 1
- #define QEMU_VERSION "0.10.2"
- #define CONFIG_SDL 1
- #define CONFIG_OPENGL 1
- #define CONFIG_AIO 1
- #define HAVE_IOVEC 1
- #define CONFIG_UNAME_RELEASE ""
- #define DEFAULT_NETWORK_SCRIPT "/etc/xen/scripts/qemu-ifup"
- #define DEFAULT_NETWORK_DOWN_SCRIPT "/etc/xen/scripts/qemu-ifdown"

4.125.1 Macro Definition Documentation

4.125.1.1 #define AUDIO_DRIVERS &oss_audio_driver, \

4.125.1.2 #define CONFIG_AC97 1

4.125.1.3 #define CONFIG_AIO 1

4.125.1.4 #define CONFIG_ES1370 1

4.125.1.5 #define CONFIG_OPENGL 1

4.125.1.6 #define CONFIG_QEMU_SHAREDIR "/usr/share/xen/qemu"

4.125.1.7 #define CONFIG_SB16 1

4.125.1.8 #define CONFIG_SDL 1

4.125.1.9 #define CONFIG_UNAME_RELEASE ""

4.125.1.10 #define CONFIG_VNC_TLS 1

4.125.1.11 #define DEFAULT_NETWORK_DOWN_SCRIPT "/etc/xen/scripts/qemu-ifdown"

4.125.1.12 #define DEFAULT_NETWORK_SCRIPT "/etc/xen/scripts/qemu-ifup"

4.125.1.13 #define HAVE_BYTESWAP_H 1

4.125.1.14 #define HAVE_IOVEC 1

4.125.1.15 #define HOST_LONG_BITS 64

4.125.1.16 #define HOST_X86_64 1

4.125.1.17 #define QEMU_VERSION "0.10.2"

4.126 xen/tools/ioemu-dir/tests/test-mmap.c File Reference

4.127 xen/tools/libxc/xc_caas.c File Reference

```
#include "xc_private.h"
#include <unistd.h>
#include <stdio.h>
#include <errno.h>
#include <fcntl.h>
#include <string.h>
#include <sys/mman.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <stdlib.h>
#include <stdint.h>
#include <sys/ioctl.h>
```

Functions

- int xc_caas_op (xc_interface *xch, caas_op_t *op)
- int xc_caas_exempt (xc_interface *xc_handle, uint64_t mfn)

4.127.1 Function Documentation

4.127.1.1 int xc_caas_exempt (xc_interface * *xc_handle*, uint64_t *mfn*)

4.127.1.2 int xc_caas_op (xc_interface * *xch*, caas_op_t * *op*)

4.128 xen/tools/libxc/xc_dom.h File Reference

```
#include <xen/libelf/libelf.h>
```

Data Structures

- struct xc_dom_seg
- struct xc_dom_mem
- struct xc_dom_phys
- struct xc_dom_image
- struct xc_dom_loader
- struct xc_dom_arch

Macros

- #define INVALID_P2M_ENTRY ((xen_pfn_t)-1)
- #define PRIpfn PRI_xen_pfn
- #define __init __attribute__ ((constructor))
- #define XC_DOM_PAGE_SHIFT(dom) ((dom)->arch_hooks->page_shift)
- #define XC_DOM_PAGE_SIZE(dom) (1 << (dom)->arch_hooks->page_shift)
- #define xc_dom_panic(xch, err, fmt, args...) xc_dom_panic_func(xch, __FILE__, __LINE__, err, fmt, ## args)
- #define xc_dom_trace(mark) xc_dom_printf("%s:%d: trace %s\n", __FILE__, __LINE__, mark)

Typedefs

- typedef uint64_t xen_vaddr_t
- typedef uint64_t xen_paddr_t

Functions

- void xc_dom_register_loader (struct xc_dom_loader *loader)
- void xc_dom_register_arch_hooks (struct xc_dom_arch *hooks)
- struct xc_dom_image * xc_dom_allocate (xc_interface *xch, const char *cmdline, const char *features)
- void xc_dom_release_phys (struct xc_dom_image *dom)
- void xc_dom_release (struct xc_dom_image *dom)
- int xc_dom_mem_init (struct xc_dom_image *dom, unsigned int mem_mb)
- size_t xc_dom_check_gzip (xc_interface *xch, void *blob, size_t ziplen)
- int xc_dom_do_gunzip (xc_interface *xch, void *src, size_t srclen, void *dst, size_t dstlen)
- int xc_dom_try_gunzip (struct xc_dom_image *dom, void **blob, size_t *size)
- int xc_dom_kernel_file (struct xc_dom_image *dom, const char *filename)
- int xc_dom_ramdisk_file (struct xc_dom_image *dom, const char *filename)
- int xc_dom_kernel_mem (struct xc_dom_image *dom, const void *mem, size_t memsize)
- int xc_dom_ramdisk_mem (struct xc_dom_image *dom, const void *mem, size_t memsize)
- int xc_dom_parse_image (struct xc_dom_image *dom)
- struct xc_dom_arch * xc_dom_find_arch_hooks (xc_interface *xch, char *guest_type)
- int xc_dom_build_image (struct xc_dom_image *dom)
- int xc_dom_update_guest_p2m (struct xc_dom_image *dom)
- int xc_dom_boot_xen_init (struct xc_dom_image *dom, xc_interface *xch, domid_t domid)
- int xc_dom_boot_mem_init (struct xc_dom_image *dom)
- void * xc_dom_boot_domU_map (struct xc_dom_image *dom, xen_pfn_t pfn, xen_pfn_t count)
- int xc_dom_boot_image (struct xc_dom_image *dom)
- int xc_dom_compat_check (struct xc_dom_image *dom)
- int xc_dom_loginit (xc_interface *xch)
- void xc_dom_printf (xc_interface *xch, const char *fmt,...) __attribute__((format(printf
- void void xc_dom_panic_func (xc_interface *xch, const char *file, int line, xc_error_code err, const char *fmt,...) __attribute__((format(printf
- void xc_dom_log_memory_footprint (struct xc_dom_image *dom)
- void * xc_dom_malloc (struct xc_dom_image *dom, size_t size)
- void * xc_dom_malloc_page_aligned (struct xc_dom_image *dom, size_t size)
- void * xc_dom_malloc_filemap (struct xc_dom_image *dom, const char *filename, size_t *size)
- char * xc_dom_strdup (struct xc_dom_image *dom, const char *str)
- int xc_dom_alloc_page (struct xc_dom_image *dom, char *name)
- int xc_dom_alloc_segment (struct xc_dom_image *dom, struct xc_dom_seg *seg, char *name, xen_vaddr_t start, xen_vaddr_t size)
- void * xc_dom_pfn_to_ptr (struct xc_dom_image *dom, xen_pfn_t first, xen_pfn_t count)
- void xc_dom_unmap_one (struct xc_dom_image *dom, xen_pfn_t pfn)
- void xc_dom_unmap_all (struct xc_dom_image *dom)
- int arch_setup_meminit (struct xc_dom_image *dom)
- int arch_setup_bootearly (struct xc_dom_image *dom)
- int arch_setup_bootlate (struct xc_dom_image *dom)

4.128.1 Macro Definition Documentation

4.128.1.1 #define __init __attribute__ ((constructor))

4.128.1.2 #define INVALID_P2M_ENTRY ((xen_pfn_t)-1)

4.128.1.3 #define PRIpfn PRI_xen_pfn

4.128.1.4 #define XC_DOM_PAGE_SHIFT(*dom*) ((dom)->arch_hooks->page_shift)

4.128.1.5 #define XC_DOM_PAGE_SIZE(*dom*) (1 << (dom)->arch_hooks->page_shift)

4.128.1.6 #define xc_dom_panic(*xch, err, fmt, args...*) xc_dom_panic_func(xch, __FILE__, __LINE__, err, fmt, ## args)

4.128.1.7 #define xc_dom_trace(*mark*) xc_dom_printf("%s:%d: trace %s\n", __FILE__, __LINE__, mark)

4.128.2 Typedef Documentation

4.128.2.1 typedef uint64_t xen_paddr_t

4.128.2.2 typedef uint64_t xen_vaddr_t

4.128.3 Function Documentation

4.128.3.1 int arch_setup_bootearly (struct xc_dom_image * *dom*)

4.128.3.2 int arch_setup_bootlate (struct xc_dom_image * *dom*)

4.128.3.3 int arch_setup_meminit (struct xc_dom_image * *dom*)

4.128.3.4 int xc_dom_alloc_page (struct xc_dom_image * *dom,* char * *name*)

4.128.3.5 int xc_dom_alloc_segment (struct xc_dom_image * *dom,* struct xc_dom_seg * *seg,* char * *name,* xen_vaddr_t *start,* xen_vaddr_t *size*)

4.128.3.6 struct xc_dom_image* xc_dom_allocate (xc_interface * *xch,* const char * *cmdline,* const char * *features*)

4.128.3.7 void* xc_dom_boot_domU_map (struct xc_dom_image * *dom,* xen_pfn_t *pfn,* xen_pfn_t *count*)

4.128.3.8 int xc_dom_boot_image (struct xc_dom_image * *dom*)

4.128.3.9 int xc_dom_boot_mem_init (struct xc_dom_image * *dom*)

4.128.3.10 int xc_dom_boot_xen_init (struct xc_dom_image * *dom,* xc_interface * *xch,* domid_t *domid*)

4.128.3.11 int xc_dom_build_image (struct xc_dom_image * *dom*)

4.128.3.12 size_t xc_dom_check_gzip (xc_interface * *xch,* void * *blob,* size_t *ziplen*)

4.128.3.13 int xc_dom_compat_check (struct xc_dom_image * *dom*)

4.128.3.14 int xc_dom_do_gunzip (xc_interface * *xch,* void * *src,* size_t *srclen,* void * *dst,* size_t *dstlen*)

4.128.3.15 struct xc_dom_arch* xc_dom_find_arch_hooks (xc_interface * *xch,* char * *guest_type*)

4.128.3.16 int xc_dom_kernel_file (struct **xc_dom_image** ∗ *dom,* const char ∗ *filename*)

4.128.3.17 int xc_dom_kernel_mem (struct **xc_dom_image** ∗ *dom,* const void ∗ *mem,* size_t *memsize*)

4.128.3.18 void xc_dom_log_memory_footprint (struct **xc_dom_image** ∗ *dom*)

4.128.3.19 int xc_dom_loginit (**xc_interface** ∗ *xch*)

4.128.3.20 void∗ xc_dom_malloc (struct **xc_dom_image** ∗ *dom,* size_t *size*)

4.128.3.21 void∗ xc_dom_malloc_filemap (struct **xc_dom_image** ∗ *dom,* const char ∗ *filename,* size_t ∗ *size*)

4.128.3.22 void∗ xc_dom_malloc_page_aligned (struct **xc_dom_image** ∗ *dom,* size_t *size*)

4.128.3.23 int xc_dom_mem_init (struct **xc_dom_image** ∗ *dom,* unsigned int *mem_mb*)

4.128.3.24 void void xc_dom_panic_func (**xc_interface** ∗ *xch,* const char ∗ *file,* int *line,* **xc_error_code** *err,* const char ∗ *fmt,* ...)

4.128.3.25 int xc_dom_parse_image (struct **xc_dom_image** ∗ *dom*)

4.128.3.26 void∗ xc_dom_pfn_to_ptr (struct **xc_dom_image** ∗ *dom,* xen_pfn_t *first,* xen_pfn_t *count*)

4.128.3.27 void xc_dom_printf (**xc_interface** ∗ *xch,* const char ∗ *fmt,* ...)

4.128.3.28 int xc_dom_ramdisk_file (struct **xc_dom_image** ∗ *dom,* const char ∗ *filename*)

4.128.3.29 int xc_dom_ramdisk_mem (struct **xc_dom_image** ∗ *dom,* const void ∗ *mem,* size_t *memsize*)

4.128.3.30 void xc_dom_register_arch_hooks (struct **xc_dom_arch** ∗ *hooks*)

4.128.3.31 void xc_dom_register_loader (struct **xc_dom_loader** ∗ *loader*)

4.128.3.32 void xc_dom_release (struct **xc_dom_image** ∗ *dom*)

4.128.3.33 void xc_dom_release_phys (struct **xc_dom_image** ∗ *dom*)

4.128.3.34 char∗ xc_dom_strdup (struct **xc_dom_image** ∗ *dom,* const char ∗ *str*)

4.128.3.35 int xc_dom_try_gunzip (struct **xc_dom_image** ∗ *dom,* void ∗∗ *blob,* size_t ∗ *size*)

4.128.3.36 void xc_dom_unmap_all (struct **xc_dom_image** ∗ *dom*)

4.128.3.37 void xc_dom_unmap_one (struct **xc_dom_image** ∗ *dom,* xen_pfn_t *pfn*)

4.128.3.38 int xc_dom_update_guest_p2m (struct **xc_dom_image** ∗ *dom*)

4.129 xen/tools/libxc/xc_dom_core.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <inttypes.h>
#include <zlib.h>
#include <assert.h>
#include "xg_private.h"
#include "xc_dom.h"
```

Functions

- int xc_dom_loginit (xc_interface *xch)
- void xc_dom_printf (xc_interface *xch, const char *fmt,...)
- void xc_dom_panic_func (xc_interface *xch, const char *file, int line, xc_error_code err, const char *fmt,...)
- void xc_dom_log_memory_footprint (struct xc_dom_image *dom)
- void * xc_dom_malloc (struct xc_dom_image *dom, size_t size)
- void * xc_dom_malloc_page_aligned (struct xc_dom_image *dom, size_t size)
- void * xc_dom_malloc_filemap (struct xc_dom_image *dom, const char *filename, size_t *size)
- char * xc_dom_strdup (struct xc_dom_image *dom, const char *str)
- size_t xc_dom_check_gzip (xc_interface *xch, void *blob, size_t ziplen)
- int xc_dom_do_gunzip (xc_interface *xch, void *src, size_t srclen, void *dst, size_t dstlen)
- int xc_dom_try_gunzip (struct xc_dom_image *dom, void **blob, size_t *size)
- void * xc_dom_pfn_to_ptr (struct xc_dom_image *dom, xen_pfn_t pfn, xen_pfn_t count)
- int xc_dom_alloc_segment (struct xc_dom_image *dom, struct xc_dom_seg *seg, char *name, xen_vaddr_t start, xen_vaddr_t size)
- int xc_dom_alloc_page (struct xc_dom_image *dom, char *name)
- void xc_dom_unmap_one (struct xc_dom_image *dom, xen_pfn_t pfn)
- void xc_dom_unmap_all (struct xc_dom_image *dom)
- void xc_dom_register_loader (struct xc_dom_loader *loader)
- void xc_dom_register_arch_hooks (struct xc_dom_arch *hooks)
- struct xc_dom_arch * xc_dom_find_arch_hooks (xc_interface *xch, char *guest_type)
- void xc_dom_release (struct xc_dom_image *dom)
- struct xc_dom_image * xc_dom_allocate (xc_interface *xch, const char *cmdline, const char *features)
- int xc_dom_kernel_file (struct xc_dom_image *dom, const char *filename)
- int xc_dom_ramdisk_file (struct xc_dom_image *dom, const char *filename)
- int xc_dom_kernel_mem (struct xc_dom_image *dom, const void *mem, size_t memsize)
- int xc_dom_ramdisk_mem (struct xc_dom_image *dom, const void *mem, size_t memsize)
- int xc_dom_parse_image (struct xc_dom_image *dom)
- int xc_dom_mem_init (struct xc_dom_image *dom, unsigned int mem_mb)
- int xc_dom_update_guest_p2m (struct xc_dom_image *dom)
- int xc_dom_build_image (struct xc_dom_image *dom)

4.129.1 Function Documentation

4.129.1.1 int xc_dom_alloc_page (struct xc_dom_image * *dom*, char * *name*)

4.129.1.2 int xc_dom_alloc_segment (struct xc_dom_image * *dom*, struct xc_dom_seg * *seg*, char * *name*, xen_vaddr_t *start*, xen_vaddr_t *size*)

4.129.1.3 struct **xc_dom_image**∗ xc_dom_allocate (**xc_interface** ∗ *xch,* const char ∗ *cmdline,* const char ∗ *features*)

4.129.1.4 int xc_dom_build_image (struct **xc_dom_image** ∗ *dom*)

4.129.1.5 size_t xc_dom_check_gzip (**xc_interface** ∗ *xch,* void ∗ *blob,* size_t *ziplen*)

4.129.1.6 int xc_dom_do_gunzip (**xc_interface** ∗ *xch,* void ∗ *src,* size_t *srclen,* void ∗ *dst,* size_t *dstlen*)

4.129.1.7 struct **xc_dom_arch**∗ xc_dom_find_arch_hooks (**xc_interface** ∗ *xch,* char ∗ *guest_type*)

4.129.1.8 int xc_dom_kernel_file (struct **xc_dom_image** ∗ *dom,* const char ∗ *filename*)

4.129.1.9 int xc_dom_kernel_mem (struct **xc_dom_image** ∗ *dom,* const void ∗ *mem,* size_t *memsize*)

4.129.1.10 void xc_dom_log_memory_footprint (struct **xc_dom_image** ∗ *dom*)

4.129.1.11 int xc_dom_loginit (**xc_interface** ∗ *xch*)

4.129.1.12 void∗ xc_dom_malloc (struct **xc_dom_image** ∗ *dom,* size_t *size*)

4.129.1.13 void∗ xc_dom_malloc_filemap (struct **xc_dom_image** ∗ *dom,* const char ∗ *filename,* size_t ∗ *size*)

4.129.1.14 void∗ xc_dom_malloc_page_aligned (struct **xc_dom_image** ∗ *dom,* size_t *size*)

4.129.1.15 int xc_dom_mem_init (struct **xc_dom_image** ∗ *dom,* unsigned int *mem_mb*)

4.129.1.16 void xc_dom_panic_func (**xc_interface** ∗ *xch,* const char ∗ *file,* int *line,* **xc_error_code** *err,* const char ∗ *fmt,* ...)

4.129.1.17 int xc_dom_parse_image (struct **xc_dom_image** ∗ *dom*)

4.129.1.18 void∗ xc_dom_pfn_to_ptr (struct **xc_dom_image** ∗ *dom,* xen_pfn_t *pfn,* xen_pfn_t *count*)

4.129.1.19 void xc_dom_printf (**xc_interface** ∗ *xch,* const char ∗ *fmt,* ...)

4.129.1.20 int xc_dom_ramdisk_file (struct **xc_dom_image** ∗ *dom,* const char ∗ *filename*)

4.129.1.21 int xc_dom_ramdisk_mem (struct **xc_dom_image** ∗ *dom,* const void ∗ *mem,* size_t *memsize*)

4.129.1.22 void xc_dom_register_arch_hooks (struct **xc_dom_arch** ∗ *hooks*)

4.129.1.23 void xc_dom_register_loader (struct **xc_dom_loader** ∗ *loader*)

4.129.1.24 void xc_dom_release (struct **xc_dom_image** ∗ *dom*)

4.129.1.25 char∗ xc_dom_strdup (struct **xc_dom_image** ∗ *dom,* const char ∗ *str*)

4.129.1.26 int xc_dom_try_gunzip (struct **xc_dom_image** ∗ *dom,* void ∗∗ *blob,* size_t ∗ *size*)

4.129.1.27 void xc_dom_unmap_all (struct **xc_dom_image** ∗ *dom*)

4.129.1.28 void xc_dom_unmap_one (struct **xc_dom_image** ∗ *dom,* xen_pfn_t *pfn*)

4.129.1.29 int xc_dom_update_guest_p2m (struct **xc_dom_image** ∗ *dom*)

4.130 xen/tools/libxc/xc_dom_x86.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <inttypes.h>
#include <xen/xen.h>
#include <xen/foreign/x86_32.h>
#include <xen/foreign/x86_64.h>
#include <xen/hvm/hvm_info_table.h>
#include <xen/io/protocols.h>
#include "xg_private.h"
#include "xc_dom.h"
#include "xenctrl.h"
```

Macros

- #define SUPERPAGE_PFN_SHIFT 9
- #define SUPERPAGE_NR_PFNS (1UL << SUPERPAGE_PFN_SHIFT)
- #define bits_to_mask(bits) (((xen_vaddr_t)1 << (bits))-1)
- #define round_down(addr, mask) ((addr) & ~(mask))
- #define round_up(addr, mask) ((addr) | (mask))
- #define L1_PROT (_PAGE_PRESENT|_PAGE_RW|_PAGE_ACCESSED)
- #define L2_PROT (_PAGE_PRESENT|_PAGE_RW|_PAGE_ACCESSED|_PAGE_DIRTY|_PAGE_USER)
- #define L3_PROT (_PAGE_PRESENT)
- #define pfn_to_paddr(pfn) ((xen_paddr_t)(pfn) << PAGE_SHIFT_X86)
- #define L1_PROT (_PAGE_PRESENT|_PAGE_RW|_PAGE_ACCESSED)
- #define L2_PROT (_PAGE_PRESENT|_PAGE_RW|_PAGE_ACCESSED|_PAGE_DIRTY|_PAGE_USER)
- #define L3_PROT (_PAGE_PRESENT|_PAGE_RW|_PAGE_ACCESSED|_PAGE_DIRTY|_PAGE_USER)
- #define L4_PROT (_PAGE_PRESENT|_PAGE_RW|_PAGE_ACCESSED|_PAGE_DIRTY|_PAGE_USER)

Functions

- int arch_setup_meminit (struct xc_dom_image *dom)
- int arch_setup_bootearly (struct xc_dom_image *dom)
- int arch_setup_bootlate (struct xc_dom_image *dom)

4.130.1 Macro Definition Documentation

4.130.1.1 #define bits_to_mask(*bits*) (((xen_vaddr_t)1 << (bits))-1)

4.130.1.2 #define L1_PROT (_PAGE_PRESENT|_PAGE_RW|_PAGE_ACCESSED)

4.130.1.3 #define L1_PROT (_PAGE_PRESENT|_PAGE_RW|_PAGE_ACCESSED)

4.130.1.4 #define L2_PROT (_PAGE_PRESENT|_PAGE_RW|_PAGE_ACCESSED|_PAGE_DIRTY|_PAGE_USER)

4.130.1.5 #define L2_PROT (_PAGE_PRESENT|_PAGE_RW|_PAGE_ACCESSED|_PAGE_DIRTY|_PAGE_USER)

4.130.1.6 #define L3_PROT (_PAGE_PRESENT)

4.130.1.7 #define L3_PROT (_PAGE_PRESENT|_PAGE_RW|_PAGE_ACCESSED|_PAGE_DIRTY|_PAGE_USER)

4.130.1.8 #define L4_PROT (_PAGE_PRESENT|_PAGE_RW|_PAGE_ACCESSED|_PAGE_DIRTY|_PAGE_USER)

4.130.1.9 #define pfn_to_paddr(pfn) ((xen_paddr_t)(pfn) << PAGE_SHIFT_X86)

4.130.1.10 #define round_down(*addr, mask*) ((addr) & ~(mask))

4.130.1.11 #define round_up(*addr, mask*) ((addr) | (mask))

4.130.1.12 #define SUPERPAGE_NR_PFNS (1UL << SUPERPAGE_PFN_SHIFT)

4.130.1.13 #define SUPERPAGE_PFN_SHIFT 9

4.130.2 Function Documentation

4.130.2.1 int arch_setup_bootearly (struct xc_dom_image * *dom*)

4.130.2.2 int arch_setup_bootlate (struct xc_dom_image * *dom*)

4.130.2.3 int arch_setup_meminit (struct xc_dom_image * *dom*)

4.131 xen/tools/libxc/xenctrl.h File Reference

```
#include <stddef.h>
#include <stdint.h>
#include <stdio.h>
#include <xen/xen.h>
#include <xen/domctl.h>
#include <xen/physdev.h>
#include <xen/sysctl.h>
#include <xen/version.h>
#include <xen/event_channel.h>
#include <xen/sched.h>
#include <xen/memory.h>
#include <xen/grant_table.h>
#include <xen/hvm/params.h>
#include <xen/xsm/acm.h>
#include <xen/xsm/acm_ops.h>
#include <xen/xsm/flask_op.h>
#include <xen/xsm/caas_op.h>
#include <xen/tmem.h>
#include "xentoollog.h"
```

Data Structures

- struct xc_hypercall_buffer
- struct xc_core_header
- struct xc_dominfo
- union vcpu_guest_context_any_t
- union shared_info_any_t
- union start_info_any_t
- struct xc_cpupoolinfo
- struct xc_error
- struct xc_px_val
- struct xc_px_stat

- struct xc_cx_stat
- struct xc_get_cpufreq_para
- struct tmem_oid

Macros

- #define __XEN_TOOLS__ 1
- #define XC_PAGE_SHIFT 12
- #define XC_PAGE_SIZE (1UL << XC_PAGE_SHIFT)
- #define XC_PAGE_MASK (~(XC_PAGE_SIZE-1))
- #define INVALID_MFN (~0UL)
- #define XENCTRL_HAS_XC_INTERFACE 1
- #define XC__HYPERCALL_BUFFER_NAME(_name) xc__hypercall_buffer_##_name
- #define HYPERCALL_BUFFER(_name)
- #define HYPERCALL_BUFFER_INIT_NO_BOUNCE .dir = 0, .sz = 0, .ubuf = (void *)-1
- #define DECLARE_HYPERCALL_BUFFER(_type, _name)
- #define DECLARE_HYPERCALL_BUFFER_ARGUMENT(_name)
- #define HYPERCALL_BUFFER_AS_ARG(_name)
- #define set_xen_guest_handle(_hnd, _val)
- #define xc_hypercall_buffer_alloc(_xch, _name, _size) xc__hypercall_buffer_alloc(_xch, HYPERCALL_BUFFER(_name), _size)
- #define xc_hypercall_buffer_free(_xch, _name) xc__hypercall_buffer_free(_xch, HYPERCALL_BUFFER(_name))
- #define xc_hypercall_buffer_alloc_pages(_xch, _name, _nr) xc__hypercall_buffer_alloc_pages(_xch, HYPERCALL_BUFFER(_name), _nr)
- #define xc_hypercall_buffer_free_pages(_xch, _name, _nr) xc__hypercall_buffer_free_pages(_xch, HYPERCALL_BUFFER(_name), _nr)
- #define XC_CORE_MAGIC 0xF00FEBED
- #define XC_CORE_MAGIC_HVM 0xF00FEBEE
- #define XC_MAX_ERROR_MSG_LEN 1024

Typedefs

- typedef struct xc_interface_core xc_interface
- typedef struct xc_interface_core xc_evtchn
- typedef struct xc_interface_core xc_gnttab
- typedef enum xc_error_code xc_error_code
- typedef struct xc_hypercall_buffer xc_hypercall_buffer_t
- typedef uint8_t * xc_cpumap_t
- typedef struct xc_core_header xc_core_header_t
- typedef struct xc_dominfo xc_dominfo_t
- typedef xen_domctl_getdomaininfo_t xc_domaininfo_t
- typedef int(dumpcore_rtn_t)(xc_interface *xch, void *arg, char *buffer, unsigned int length)
- typedef xen_domctl_getvcpuinfo_t xc_vcpuinfo_t
- typedef xen_domctl_shadow_op_stats_t xc_shadow_op_stats_t
- typedef struct xc_cpupoolinfo xc_cpupoolinfo_t
- typedef int evtchn_port_or_error_t
- typedef struct evtchn_status xc_evtchn_status_t
- typedef xen_sysctl_physinfo_t xc_physinfo_t
- typedef xen_sysctl_topologyinfo_t xc_topologyinfo_t
- typedef xen_sysctl_numainfo_t xc_numainfo_t
- typedef uint32_t xc_cpu_to_node_t
- typedef uint32_t xc_cpu_to_socket_t

- typedef uint32_t xc_cpu_to_core_t
- typedef uint64_t xc_node_to_memsize_t
- typedef uint64_t xc_node_to_memfree_t
- typedef uint32_t xc_node_to_node_dist_t
- typedef xen_sysctl_cpuinfo_t xc_cpuinfo_t
- typedef xen_sysctl_perfc_desc_t xc_perfc_desc_t
- typedef xen_sysctl_perfc_val_t xc_perfc_val_t
- typedef xen_sysctl_lockprof_data_t xc_lockprof_data_t
- typedef struct xc_error xc_error
- typedef struct xc_cx_stat xc_cx_stat_t
- typedef xen_userspace_t xc_userspace_t
- typedef xen_ondemand_t xc_ondemand_t

Enumerations

- enum xc_open_flags { XC_OPENFLAG_DUMMY = 1<<0, XC_OPENFLAG_NON_REENTRANT = 1<<1 }
- enum xc_error_code {
  XC_ERROR_NONE = 0, XC_INTERNAL_ERROR = 1, XC_INVALID_KERNEL = 2, XC_INVALID_PARAM = 3,
  XC_OUT_OF_MEMORY = 4 }

Functions

- xc_interface * xc_interface_open (xentoollog_logger *logger, xentoollog_logger *dombuild_logger, unsigned open_flags)
- int xc_interface_close (xc_interface *xch)
- int xc_interface_is_fake (void)
- xc_hypercall_buffer_t XC__HYPERCALL_BUFFER_NAME (HYPERCALL_BUFFER_NULL)
- void * xc__hypercall_buffer_alloc (xc_interface *xch, xc_hypercall_buffer_t *b, size_t size)
- void xc__hypercall_buffer_free (xc_interface *xch, xc_hypercall_buffer_t *b)
- void * xc__hypercall_buffer_alloc_pages (xc_interface *xch, xc_hypercall_buffer_t *b, int nr_pages)
- void xc__hypercall_buffer_free_pages (xc_interface *xch, xc_hypercall_buffer_t *b, int nr_pages)
- int xc_get_max_cpus (xc_interface *xch)
- int xc_get_cpumap_size (xc_interface *xch)
- xc_cpumap_t xc_cpumap_alloc (xc_interface *xch)
- int xc_domain_create (xc_interface *xch, uint32_t ssidref, xen_domain_handle_t handle, uint32_t flags, uint32_t *pdomid)
- int xc_domain_dumpcore (xc_interface *xch, uint32_t domid, const char *corename)
- int xc_domain_dumpcore_via_callback (xc_interface *xch, uint32_t domid, void *arg, dumpcore_rtn_t dump_rtn)
- int xc_domain_max_vcpus (xc_interface *xch, uint32_t domid, unsigned int max)
- int xc_domain_pause (xc_interface *xch, uint32_t domid)
- int xc_domain_unpause (xc_interface *xch, uint32_t domid)
- int xc_domain_destroy (xc_interface *xch, uint32_t domid)
- int xc_domain_resume (xc_interface *xch, uint32_t domid, int fast)
- int xc_domain_shutdown (xc_interface *xch, uint32_t domid, int reason)
- int xc_watchdog (xc_interface *xch, uint32_t id, uint32_t timeout)
- int xc_vcpu_setaffinity (xc_interface *xch, uint32_t domid, int vcpu, xc_cpumap_t cpumap)
- int xc_vcpu_getaffinity (xc_interface *xch, uint32_t domid, int vcpu, xc_cpumap_t cpumap)
- int xc_domain_getinfo (xc_interface *xch, uint32_t first_domid, unsigned int max_doms, xc_dominfo_t *info)
- int xc_vcpu_setcontext (xc_interface *xch, uint32_t domid, uint32_t vcpu, vcpu_guest_context_any_t *ctxt)
- int xc_domain_getinfolist (xc_interface *xch, uint32_t first_domain, unsigned int max_domains, xc_domaininfo_t *info)
- int xc_domain_hvm_getcontext (xc_interface *xch, uint32_t domid, uint8_t *ctxt_buf, uint32_t size)

- int xc_domain_hvm_getcontext_partial (xc_interface *xch, uint32_t domid, uint16_t typecode, uint16_t instance, void *ctxt_buf, uint32_t size)
- int xc_domain_hvm_setcontext (xc_interface *xch, uint32_t domid, uint8_t *hvm_ctxt, uint32_t size)
- int xc_vcpu_getcontext (xc_interface *xch, uint32_t domid, uint32_t vcpu, vcpu_guest_context_any_t *ctxt)
- int xc_vcpu_getinfo (xc_interface *xch, uint32_t domid, uint32_t vcpu, xc_vcpuinfo_t *info)
- long long xc_domain_get_cpu_usage (xc_interface *xch, domid_t domid, int vcpu)
- int xc_domain_sethandle (xc_interface *xch, uint32_t domid, xen_domain_handle_t handle)
- int xc_shadow_control (xc_interface *xch, uint32_t domid, unsigned int sop, xc_hypercall_buffer_t *dirty_bitmap, unsigned long pages, unsigned long *mb, uint32_t mode, xc_shadow_op_stats_t *stats)
- int xc_sedf_domain_set (xc_interface *xch, uint32_t domid, uint64_t period, uint64_t slice, uint64_t latency, uint16_t extratime, uint16_t weight)
- int xc_sedf_domain_get (xc_interface *xch, uint32_t domid, uint64_t *period, uint64_t *slice, uint64_t *latency, uint16_t *extratime, uint16_t *weight)
- int xc_sched_credit_domain_set (xc_interface *xch, uint32_t domid, struct xen_domctl_sched_credit *sdom)
- int xc_sched_credit_domain_get (xc_interface *xch, uint32_t domid, struct xen_domctl_sched_credit *sdom)
- int xc_sched_credit2_domain_set (xc_interface *xch, uint32_t domid, struct xen_domctl_sched_credit2 *sdom)
- int xc_sched_credit2_domain_get (xc_interface *xch, uint32_t domid, struct xen_domctl_sched_credit2 *sdom)
- int xc_sched_arinc653_schedule_set (xc_interface *xch, struct xen_sysctl_arinc653_schedule *schedule)
- int xc_sched_arinc653_schedule_get (xc_interface *xch, struct xen_sysctl_arinc653_schedule *schedule)
- int xc_domain_send_trigger (xc_interface *xch, uint32_t domid, uint32_t trigger, uint32_t vcpu)
- int xc_domain_setdebugging (xc_interface *xch, uint32_t domid, unsigned int enable)
- int xc_domain_set_access_required (xc_interface *xch, uint32_t domid, unsigned int required)
- int xc_cpupool_create (xc_interface *xch, uint32_t *ppoolid, uint32_t sched_id)
- int xc_cpupool_destroy (xc_interface *xch, uint32_t poolid)
- xc_cpupoolinfo_t * xc_cpupool_getinfo (xc_interface *xch, uint32_t poolid)
- void xc_cpupool_infofree (xc_interface *xch, xc_cpupoolinfo_t *info)
- int xc_cpupool_addcpu (xc_interface *xch, uint32_t poolid, int cpu)
- int xc_cpupool_removecpu (xc_interface *xch, uint32_t poolid, int cpu)
- int xc_cpupool_movedomain (xc_interface *xch, uint32_t poolid, uint32_t domid)
- xc_cpumap_t xc_cpupool_freeinfo (xc_interface *xch)
- evtchn_port_or_error_t xc_evtchn_alloc_unbound (xc_interface *xch, uint32_t dom, uint32_t remote_dom)
- int xc_evtchn_reset (xc_interface *xch, uint32_t dom)
- int xc_evtchn_status (xc_interface *xch, xc_evtchn_status_t *status)
- xc_evtchn * xc_evtchn_open (xentoollog_logger *logger, unsigned open_flags)
- int xc_evtchn_close (xc_evtchn *xce)
- int xc_evtchn_fd (xc_evtchn *xce)
- int xc_evtchn_notify (xc_evtchn *xce, evtchn_port_t port)
- evtchn_port_or_error_t xc_evtchn_bind_unbound_port (xc_evtchn *xce, int domid)
- evtchn_port_or_error_t xc_evtchn_bind_interdomain (xc_evtchn *xce, int domid, evtchn_port_t remote_port)
- evtchn_port_or_error_t xc_evtchn_bind_virq (xc_evtchn *xce, unsigned int virq)
- int xc_evtchn_unbind (xc_evtchn *xce, evtchn_port_t port)
- evtchn_port_or_error_t xc_evtchn_pending (xc_evtchn *xce)
- int xc_evtchn_unmask (xc_evtchn *xce, evtchn_port_t port)
- int xc_physdev_pci_access_modify (xc_interface *xch, uint32_t domid, int bus, int dev, int func, int enable)
- int xc_readconsolering (xc_interface *xch, char *buffer, unsigned int *pnr_chars, int clear, int incremental, uint32_t *pindex)
- int xc_send_debug_keys (xc_interface *xch, char *keys)
- int xc_physinfo (xc_interface *xch, xc_physinfo_t *info)
- int xc_topologyinfo (xc_interface *xch, xc_topologyinfo_t *info)
- int xc_numainfo (xc_interface *xch, xc_numainfo_t *info)
- int xc_sched_id (xc_interface *xch, int *sched_id)
- int xc_machphys_mfn_list (xc_interface *xch, unsigned long max_extents, xen_pfn_t *extent_start)
- int xc_getcpuinfo (xc_interface *xch, int max_cpus, xc_cpuinfo_t *info, int *nr_cpus)

- int xc_domain_setmaxmem (xc_interface *xch, uint32_t domid, unsigned int max_memkb)
- int xc_domain_set_memmap_limit (xc_interface *xch, uint32_t domid, unsigned long map_limitkb)
- int xc_domain_set_time_offset (xc_interface *xch, uint32_t domid, int32_t time_offset_seconds)
- int xc_domain_set_tsc_info (xc_interface *xch, uint32_t domid, uint32_t tsc_mode, uint64_t elapsed_nsec, uint32_t gtsc_khz, uint32_t incarnation)
- int xc_domain_get_tsc_info (xc_interface *xch, uint32_t domid, uint32_t *tsc_mode, uint64_t *elapsed_nsec, uint32_t *gtsc_khz, uint32_t *incarnation)
- int xc_domain_disable_migrate (xc_interface *xch, uint32_t domid)
- int xc_domain_maximum_gpfn (xc_interface *xch, domid_t domid)
- int xc_domain_increase_reservation (xc_interface *xch, uint32_t domid, unsigned long nr_extents, unsigned int extent_order, unsigned int mem_flags, xen_pfn_t *extent_start)
- int xc_domain_increase_reservation_exact (xc_interface *xch, uint32_t domid, unsigned long nr_extents, unsigned int extent_order, unsigned int mem_flags, xen_pfn_t *extent_start)
- int xc_domain_decrease_reservation (xc_interface *xch, uint32_t domid, unsigned long nr_extents, unsigned int extent_order, xen_pfn_t *extent_start)
- int xc_domain_decrease_reservation_exact (xc_interface *xch, uint32_t domid, unsigned long nr_extents, unsigned int extent_order, xen_pfn_t *extent_start)
- int xc_domain_add_to_physmap (xc_interface *xch, uint32_t domid, unsigned int space, unsigned long idx, xen_pfn_t gpfn)
- int xc_domain_populate_physmap (xc_interface *xch, uint32_t domid, unsigned long nr_extents, unsigned int extent_order, unsigned int mem_flags, xen_pfn_t *extent_start)
- int xc_domain_populate_physmap_exact (xc_interface *xch, uint32_t domid, unsigned long nr_extents, unsigned int extent_order, unsigned int mem_flags, xen_pfn_t *extent_start)
- int xc_domain_memory_exchange_pages (xc_interface *xch, int domid, unsigned long nr_in_extents, unsigned int in_order, xen_pfn_t *in_extents, unsigned long nr_out_extents, unsigned int out_order, xen_pfn_t *out_extents)
- int xc_domain_set_pod_target (xc_interface *xch, uint32_t domid, uint64_t target_pages, uint64_t *tot_pages, uint64_t *pod_cache_pages, uint64_t *pod_entries)
- int xc_domain_get_pod_target (xc_interface *xch, uint32_t domid, uint64_t *tot_pages, uint64_t *pod_cache_pages, uint64_t *pod_entries)
- int xc_domain_ioport_permission (xc_interface *xch, uint32_t domid, uint32_t first_port, uint32_t nr_ports, uint32_t allow_access)
- int xc_domain_irq_permission (xc_interface *xch, uint32_t domid, uint8_t pirq, uint8_t allow_access)
- int xc_domain_iomem_permission (xc_interface *xch, uint32_t domid, unsigned long first_mfn, unsigned long nr_mfns, uint8_t allow_access)
- int xc_domain_pin_memory_cacheattr (xc_interface *xch, uint32_t domid, uint64_t start, uint64_t end, uint32_t type)
- unsigned long xc_make_page_below_4G (xc_interface *xch, uint32_t domid, unsigned long mfn)
- int xc_perfc_reset (xc_interface *xch)
- int xc_perfc_query_number (xc_interface *xch, int *nbr_desc, int *nbr_val)
- int xc_perfc_query (xc_interface *xch, xc_hypercall_buffer_t *desc, xc_hypercall_buffer_t *val)
- int xc_lockprof_reset (xc_interface *xch)
- int xc_lockprof_query_number (xc_interface *xch, uint32_t *n_elems)
- int xc_lockprof_query (xc_interface *xch, uint32_t *n_elems, uint64_t *time, xc_hypercall_buffer_t *data)
- void * xc_map_foreign_range (xc_interface *xch, uint32_t dom, int size, int prot, unsigned long mfn)
- void * xc_map_foreign_pages (xc_interface *xch, uint32_t dom, int prot, const xen_pfn_t *arr, int num)
- void * xc_map_foreign_batch (xc_interface *xch, uint32_t dom, int prot, xen_pfn_t *arr, int num)
- void * xc_map_foreign_bulk (xc_interface *xch, uint32_t dom, int prot, const xen_pfn_t *arr, int *err, unsigned int num)
- unsigned long xc_translate_foreign_address (xc_interface *xch, uint32_t dom, int vcpu, unsigned long long virt)
- int xc_get_pfn_list (xc_interface *xch, uint32_t domid, uint64_t *pfn_buf, unsigned long max_pfns)
- unsigned long xc_ia64_fpsr_default (void)
- int xc_copy_to_domain_page (xc_interface *xch, uint32_t domid, unsigned long dst_pfn, const char *src_page)
- int xc_clear_domain_page (xc_interface *xch, uint32_t domid, unsigned long dst_pfn)

- int xc_mmuext_op (xc_interface *xch, struct mmuext_op *op, unsigned int nr_ops, domid_t dom)
- long xc_maximum_ram_page (xc_interface *xch)
- long xc_get_tot_pages (xc_interface *xch, uint32_t domid)
- int xc_availheap (xc_interface *xch, int min_width, int max_width, int node, uint64_t *bytes)
- int xc_tbuf_enable (xc_interface *xch, unsigned long pages, unsigned long *mfn, unsigned long *size)
- int xc_tbuf_disable (xc_interface *xch)
- int xc_tbuf_set_size (xc_interface *xch, unsigned long size)
- int xc_tbuf_get_size (xc_interface *xch, unsigned long *size)
- int xc_tbuf_set_cpu_mask (xc_interface *xch, uint32_t mask)
- int xc_tbuf_set_evt_mask (xc_interface *xch, uint32_t mask)
- int xc_domctl (xc_interface *xch, struct xen_domctl *domctl)
- int xc_sysctl (xc_interface *xch, struct xen_sysctl *sysctl)
- int xc_version (xc_interface *xch, int cmd, void *arg)
- int xc_acm_op (xc_interface *xch, int cmd, void *arg, unsigned long arg_size)
- int xc_flask_op (xc_interface *xch, flask_op_t *op)
- int xc_caas_op (xc_interface *xch, caas_op_t *op)
- int xc_domain_subscribe_for_suspend (xc_interface *xch, domid_t domid, evtchn_port_t port)
- xc_gnttab * xc_gnttab_open (xentoollog_logger *logger, unsigned open_flags)
- int xc_gnttab_close (xc_gnttab *xcg)
- void * xc_gnttab_map_grant_ref (xc_gnttab *xcg, uint32_t domid, uint32_t ref, int prot)
- void * xc_gnttab_map_grant_refs (xc_gnttab *xcg, uint32_t count, uint32_t *domids, uint32_t *refs, int prot)
- void * xc_gnttab_map_domain_grant_refs (xc_gnttab *xcg, uint32_t count, uint32_t domid, uint32_t *refs, int prot)
- int xc_gnttab_munmap (xc_gnttab *xcg, void *start_address, uint32_t count)
- int xc_gnttab_set_max_grants (xc_gnttab *xcg, uint32_t count)
- int xc_gnttab_op (xc_interface *xch, int cmd, void *op, int op_size, int count)
- int xc_gnttab_get_version (xc_interface *xch, int domid)
- grant_entry_v1_t * xc_gnttab_map_table_v1 (xc_interface *xch, int domid, int *gnt_num)
- grant_entry_v2_t * xc_gnttab_map_table_v2 (xc_interface *xch, int domid, int *gnt_num)
- int xc_physdev_map_pirq (xc_interface *xch, int domid, int index, int *pirq)
- int xc_physdev_map_pirq_msi (xc_interface *xch, int domid, int index, int *pirq, int devfn, int bus, int entry_nr, uint64_t table_base)
- int xc_physdev_unmap_pirq (xc_interface *xch, int domid, int pirq)
- int xc_hvm_set_pci_intx_level (xc_interface *xch, domid_t dom, uint8_t domain, uint8_t bus, uint8_t device, uint8_t intx, unsigned int level)
- int xc_hvm_set_isa_irq_level (xc_interface *xch, domid_t dom, uint8_t isa_irq, unsigned int level)
- int xc_hvm_set_pci_link_route (xc_interface *xch, domid_t dom, uint8_t link, uint8_t isa_irq)
- int xc_hvm_track_dirty_vram (xc_interface *xch, domid_t dom, uint64_t first_pfn, uint64_t nr, unsigned long *bitmap)
- int xc_hvm_modified_memory (xc_interface *xch, domid_t dom, uint64_t first_pfn, uint64_t nr)
- int xc_hvm_set_mem_type (xc_interface *xch, domid_t dom, hvmmem_type_t memtype, uint64_t first_pfn, uint64_t nr)
- int xc_hvm_set_mem_access (xc_interface *xch, domid_t dom, hvmmem_access_t memaccess, uint64_t first_pfn, uint64_t nr)
- int xc_hvm_get_mem_access (xc_interface *xch, domid_t dom, uint64_t pfn, hvmmem_access_t *memaccess)
- int xc_hvm_inject_trap (xc_interface *xch, domid_t dom, int vcpu, uint32_t trap, uint32_t error_code, uint64_t cr2)
- const char * xc_error_code_to_desc (int code)
- const char * xc_strerror (xc_interface *xch, int errcode)
- const xc_error * xc_get_last_error (xc_interface *handle)
- void xc_clear_last_error (xc_interface *xch)
- int xc_set_hvm_param (xc_interface *handle, domid_t dom, int param, unsigned long value)
- int xc_get_hvm_param (xc_interface *handle, domid_t dom, int param, unsigned long *value)
- int xc_ia64_save_to_nvram (xc_interface *xch, uint32_t dom)

- int xc_ia64_nvram_init (xc_interface *xch, char *dom_name, uint32_t dom)
- int xc_ia64_set_os_type (xc_interface *xch, char *guest_os_type, uint32_t dom)
- int xc_assign_device (xc_interface *xch, uint32_t domid, uint32_t machine_bdf)
- int xc_get_device_group (xc_interface *xch, uint32_t domid, uint32_t machine_bdf, uint32_t max_sdevs, uint32_t *num_sdevs, uint32_t *sdev_array)
- int xc_test_assign_device (xc_interface *xch, uint32_t domid, uint32_t machine_bdf)
- int xc_deassign_device (xc_interface *xch, uint32_t domid, uint32_t machine_bdf)
- int xc_domain_memory_mapping (xc_interface *xch, uint32_t domid, unsigned long first_gfn, unsigned long first_mfn, unsigned long nr_mfns, uint32_t add_mapping)
- int xc_domain_ioport_mapping (xc_interface *xch, uint32_t domid, uint32_t first_gport, uint32_t first_mport, uint32_t nr_ports, uint32_t add_mapping)
- int xc_domain_update_msi_irq (xc_interface *xch, uint32_t domid, uint32_t gvec, uint32_t pirq, uint32_t gflags, uint64_t gtable)
- int xc_domain_unbind_msi_irq (xc_interface *xch, uint32_t domid, uint32_t gvec, uint32_t pirq, uint32_t gflags)
- int xc_domain_bind_pt_irq (xc_interface *xch, uint32_t domid, uint8_t machine_irq, uint8_t irq_type, uint8_t bus, uint8_t device, uint8_t intx, uint8_t isa_irq)
- int xc_domain_unbind_pt_irq (xc_interface *xch, uint32_t domid, uint8_t machine_irq, uint8_t irq_type, uint8_t bus, uint8_t device, uint8_t intx, uint8_t isa_irq)
- int xc_domain_bind_pt_pci_irq (xc_interface *xch, uint32_t domid, uint8_t machine_irq, uint8_t bus, uint8_t device, uint8_t intx)
- int xc_domain_bind_pt_isa_irq (xc_interface *xch, uint32_t domid, uint8_t machine_irq)
- int xc_domain_set_machine_address_size (xc_interface *xch, uint32_t domid, unsigned int width)
- int xc_domain_get_machine_address_size (xc_interface *xch, uint32_t domid)
- int xc_domain_suppress_spurious_page_faults (xc_interface *xch, uint32_t domid)
- int xc_domain_set_target (xc_interface *xch, uint32_t domid, uint32_t target)
- int xc_domain_debug_control (xc_interface *xch, uint32_t domid, uint32_t sop, uint32_t vcpu)
- int xc_pm_get_max_px (xc_interface *xch, int cpuid, int *max_px)
- int xc_pm_get_pxstat (xc_interface *xch, int cpuid, struct xc_px_stat *pxpt)
- int xc_pm_reset_pxstat (xc_interface *xch, int cpuid)
- int xc_pm_get_max_cx (xc_interface *xch, int cpuid, int *max_cx)
- int xc_pm_get_cxstat (xc_interface *xch, int cpuid, struct xc_cx_stat *cxpt)
- int xc_pm_reset_cxstat (xc_interface *xch, int cpuid)
- int xc_cpu_online (xc_interface *xch, int cpu)
- int xc_cpu_offline (xc_interface *xch, int cpu)
- int xc_get_cpufreq_para (xc_interface *xch, int cpuid, struct xc_get_cpufreq_para *user_para)
- int xc_set_cpufreq_gov (xc_interface *xch, int cpuid, char *govname)
- int xc_set_cpufreq_para (xc_interface *xch, int cpuid, int ctrl_type, int ctrl_value)
- int xc_get_cpufreq_avgfreq (xc_interface *xch, int cpuid, int *avg_freq)
- int xc_set_sched_opt_smt (xc_interface *xch, uint32_t value)
- int xc_set_vcpu_migration_delay (xc_interface *xch, uint32_t value)
- int xc_get_vcpu_migration_delay (xc_interface *xch, uint32_t *value)
- int xc_get_cpuidle_max_cstate (xc_interface *xch, uint32_t *value)
- int xc_set_cpuidle_max_cstate (xc_interface *xch, uint32_t value)
- int xc_enable_turbo (xc_interface *xch, int cpuid)
- int xc_disable_turbo (xc_interface *xch, int cpuid)
- int xc_tmem_control_oid (xc_interface *xch, int32_t pool_id, uint32_t subop, uint32_t cli_id, uint32_t arg1, uint32_t arg2, struct tmem_oid oid, void *buf)
- int xc_tmem_control (xc_interface *xch, int32_t pool_id, uint32_t subop, uint32_t cli_id, uint32_t arg1, uint32_t arg2, uint64_t arg3, void *buf)
- int xc_tmem_auth (xc_interface *xch, int cli_id, char *uuid_str, int arg1)
- int xc_tmem_save (xc_interface *xch, int dom, int live, int fd, int field_marker)
- int xc_tmem_save_extra (xc_interface *xch, int dom, int fd, int field_marker)
- void xc_tmem_save_done (xc_interface *xch, int dom)
- int xc_tmem_restore (xc_interface *xch, int dom, int fd)

- int xc_tmem_restore_extra (xc_interface *xch, int dom, int fd)
- int xc_mem_event_control (xc_interface *xch, domid_t domain_id, unsigned int op, unsigned int mode, void *shared_page, void *ring_page, unsigned long gfn)
- int xc_mem_event_enable (xc_interface *xch, domid_t domain_id, void *shared_page, void *ring_page)
- int xc_mem_event_disable (xc_interface *xch, domid_t domain_id)
- int xc_mem_paging_nominate (xc_interface *xch, domid_t domain_id, unsigned long gfn)
- int xc_mem_paging_evict (xc_interface *xch, domid_t domain_id, unsigned long gfn)
- int xc_mem_paging_prep (xc_interface *xch, domid_t domain_id, unsigned long gfn)
- int xc_mem_paging_resume (xc_interface *xch, domid_t domain_id, unsigned long gfn)
- int xc_mem_access_resume (xc_interface *xch, domid_t domain_id, unsigned long gfn)
- int xc_memshr_control (xc_interface *xch, uint32_t domid, int enable)
- int xc_memshr_nominate_gfn (xc_interface *xch, uint32_t domid, unsigned long gfn, uint64_t *handle)
- int xc_memshr_nominate_gref (xc_interface *xch, uint32_t domid, grant_ref_t gref, uint64_t *handle)
- int xc_memshr_share (xc_interface *xch, uint64_t source_handle, uint64_t client_handle)
- int xc_memshr_domain_resume (xc_interface *xch, uint32_t domid)
- int xc_memshr_debug_gfn (xc_interface *xch, uint32_t domid, unsigned long gfn)
- int xc_memshr_debug_mfn (xc_interface *xch, uint32_t domid, unsigned long mfn)
- int xc_memshr_debug_gref (xc_interface *xch, uint32_t domid, grant_ref_t gref)
- int xc_flask_load (xc_interface *xc_handle, char *buf, uint32_t size)
- int xc_flask_context_to_sid (xc_interface *xc_handle, char *buf, uint32_t size, uint32_t *sid)
- int xc_flask_sid_to_context (xc_interface *xc_handle, int sid, char *buf, uint32_t size)
- int xc_flask_getenforce (xc_interface *xc_handle)
- int xc_flask_setenforce (xc_interface *xc_handle, int mode)
- int xc_flask_add_pirq (xc_interface *xc_handle, unsigned int pirq, char *scontext)
- int xc_flask_add_ioport (xc_interface *xc_handle, unsigned long low, unsigned long high, char *scontext)
- int xc_flask_add_iomem (xc_interface *xc_handle, unsigned long low, unsigned long high, char *scontext)
- int xc_flask_add_device (xc_interface *xc_handle, unsigned long device, char *scontext)
- int xc_flask_del_pirq (xc_interface *xc_handle, unsigned int pirq)
- int xc_flask_del_ioport (xc_interface *xc_handle, unsigned long low, unsigned long high)
- int xc_flask_del_iomem (xc_interface *xc_handle, unsigned long low, unsigned long high)
- int xc_flask_del_device (xc_interface *xc_handle, unsigned long device)
- int xc_flask_access (xc_interface *xc_handle, const char *scon, const char *tcon, uint16_t tclass, uint32_t req, uint32_t *allowed, uint32_t *decided, uint32_t *auditallow, uint32_t *auditdeny, uint32_t *seqno)
- int xc_flask_avc_cachestats (xc_interface *xc_handle, char *buf, int size)
- int xc_flask_policyvers (xc_interface *xc_handle, char *buf, int size)
- int xc_flask_avc_hashstats (xc_interface *xc_handle, char *buf, int size)
- int xc_flask_getavc_threshold (xc_interface *xc_handle)
- int xc_flask_setavc_threshold (xc_interface *xc_handle, int threshold)
- int xc_caas_exempt (xc_interface *xc_handle, uint64_t mfn)
- void xc_elf_set_logfile (xc_interface *xch, struct elf_binary *elf, int verbose)

4.131.1 Macro Definition Documentation

4.131.1.1 #define __XEN_TOOLS__ 1

4.131.1.2 #define DECLARE_HYPERCALL_BUFFER(_type, _name)

Value:

```
_type *_name = NULL;                                          \
    xc_hypercall_buffer_t XC__HYPERCALL_BUFFER_NAME(_name
      ) = { \
        .hbuf = NULL,                                         \
        .param_shadow = NULL,                                 \
        HYPERCALL_BUFFER_INIT_NO_BOUNCE                       \
    }
```

4.131.1.3 #define DECLARE_HYPERCALL_BUFFER_ARGUMENT(_name)

Value:

```
xc_hypercall_buffer_t XC__HYPERCALL_BUFFER_NAME(_name) = { \
        .hbuf = (void *)-1,                                \
        .param_shadow = _name,                             \
        HYPERCALL_BUFFER_INIT_NO_BOUNCE                    \
    }
```

4.131.1.4 #define HYPERCALL_BUFFER(_name)

Value:

```
({  xc_hypercall_buffer_t _val1;
        \
        typeof(XC__HYPERCALL_BUFFER_NAME(_name)) *_val2 = &
      XC__HYPERCALL_BUFFER_NAME(_name); \
        (void)(&_val1 == _val2);                                                    \
        (_val2)->param_shadow ? (_val2)->param_shadow : (_val2);                    \
     })
```

4.131.1.5 #define HYPERCALL_BUFFER_AS_ARG(_name)

Value:

```
({  xc_hypercall_buffer_t _val1;                                        \
        typeof(XC__HYPERCALL_BUFFER_NAME(_name)) *_val2 =
      HYPERCALL_BUFFER(_name); \
        (void)(&_val1 == _val2);                                          \
        (unsigned long)(_val2)->hbuf;                                     \
     })
```

4.131.1.6 #define HYPERCALL_BUFFER_INIT_NO_BOUNCE .dir = 0, .sz = 0, .ubuf = (void ∗)-1

4.131.1.7 #define INVALID_MFN (∼0UL)

4.131.1.8 #define set_xen_guest_handle(_hnd, _val)

Value:

```
do {                                                                  \
        xc_hypercall_buffer_t _val1;
         \
        typeof(XC__HYPERCALL_BUFFER_NAME(_val)) *_val2 =
      HYPERCALL_BUFFER(_val); \
        (void) (&_val1 == _val2);                                       \
        set_xen_guest_handle_raw(_hnd, (_val2)->hbuf);                  \
    } while (0)
```

4.131.1.9 #define XC__HYPERCALL_BUFFER_NAME(_name) xc__hypercall_buffer_##_name

4.131.1.10 #define XC_CORE_MAGIC 0xF00FEBED

4.131.1.11 #define XC_CORE_MAGIC_HVM 0xF00FEBEE

4.131.1.12 #define xc_hypercall_buffer_alloc(_xch, _name, _size) xc__hypercall_buffer_alloc(_xch, HYPERCALL_BUFFER(_name), _size)

4.131.1.13 #define xc_hypercall_buffer_alloc_pages(*_xch, _name, _nr*) **xc__hypercall_buffer_alloc_pages**(_xch, **HYPERCALL_BUFFER**(_name), _nr)

4.131.1.14 #define xc_hypercall_buffer_free(*_xch, _name*) **xc__hypercall_buffer_free**(_xch, **HYPERCALL_BUFFER**(_name))

4.131.1.15 #define xc_hypercall_buffer_free_pages(*_xch, _name, _nr*) **xc__hypercall_buffer_free_pages**(_xch, **HYPERCALL_BUFFER**(_name), _nr)

4.131.1.16 #define XC_MAX_ERROR_MSG_LEN 1024

4.131.1.17 #define XC_PAGE_MASK (~(**XC_PAGE_SIZE**-1))

4.131.1.18 #define XC_PAGE_SHIFT 12

4.131.1.19 #define XC_PAGE_SIZE (1UL << XC_PAGE_SHIFT)

4.131.1.20 #define XENCTRL_HAS_XC_INTERFACE 1

4.131.2 Typedef Documentation

4.131.2.1 typedef int(dumpcore_rtn_t)(**xc_interface** *xch, void *arg, char ***buffer**, unsigned int **length**)

4.131.2.2 typedef int **evtchn_port_or_error_t**

4.131.2.3 typedef struct **xc_core_header xc_core_header_t**

4.131.2.4 typedef uint32_t **xc_cpu_to_core_t**

4.131.2.5 typedef uint32_t **xc_cpu_to_node_t**

4.131.2.6 typedef uint32_t **xc_cpu_to_socket_t**

4.131.2.7 typedef xen_sysctl_cpuinfo_t **xc_cpuinfo_t**

4.131.2.8 typedef uint8_t* **xc_cpumap_t**

4.131.2.9 typedef struct **xc_cpupoolinfo xc_cpupoolinfo_t**

4.131.2.10 typedef struct **xc_cx_stat xc_cx_stat_t**

4.131.2.11 typedef **xen_domctl_getdomaininfo_t xc_domaininfo_t**

4.131.2.12 typedef struct **xc_dominfo xc_dominfo_t**

4.131.2.13 typedef struct **xc_error xc_error**

4.131.2.14 typedef enum **xc_error_code xc_error_code**

4.131.2.15 typedef struct xc_interface_core **xc_evtchn**

4.131.2.16 typedef struct evtchn_status **xc_evtchn_status_t**

4.131.2.17 typedef struct xc_interface_core **xc_gnttab**

4.131.2.18 typedef struct **xc_hypercall_buffer xc_hypercall_buffer_t**

4.131.2.19 typedef struct xc_interface_core xc_interface

4.131.2.20 typedef xen_sysctl_lockprof_data_t xc_lockprof_data_t

4.131.2.21 typedef uint64_t xc_node_to_memfree_t

4.131.2.22 typedef uint64_t xc_node_to_memsize_t

4.131.2.23 typedef uint32_t xc_node_to_node_dist_t

4.131.2.24 typedef xen_sysctl_numainfo_t xc_numainfo_t

4.131.2.25 typedef xen_ondemand_t xc_ondemand_t

4.131.2.26 typedef xen_sysctl_perfc_desc_t xc_perfc_desc_t

4.131.2.27 typedef xen_sysctl_perfc_val_t xc_perfc_val_t

4.131.2.28 typedef xen_sysctl_physinfo_t xc_physinfo_t

4.131.2.29 typedef xen_domctl_shadow_op_stats_t xc_shadow_op_stats_t

4.131.2.30 typedef xen_sysctl_topologyinfo_t xc_topologyinfo_t

4.131.2.31 typedef xen_userspace_t xc_userspace_t

4.131.2.32 typedef xen_domctl_getvcpuinfo_t xc_vcpuinfo_t

4.131.3 Enumeration Type Documentation

4.131.3.1 enum xc_error_code

Enumerator

XC_ERROR_NONE
XC_INTERNAL_ERROR
XC_INVALID_KERNEL
XC_INVALID_PARAM
XC_OUT_OF_MEMORY

4.131.3.2 enum xc_open_flags

Enumerator

XC_OPENFLAG_DUMMY
XC_OPENFLAG_NON_REENTRANT

4.131.4 Function Documentation

4.131.4.1 void* xc__hypercall_buffer_alloc (xc_interface * *xch*, xc_hypercall_buffer_t * *b*, size_t *size*)

4.131.4.2 void* xc__hypercall_buffer_alloc_pages (xc_interface * *xch*, xc_hypercall_buffer_t * *b*, int *nr_pages*)

4.131.4.3 void xc__hypercall_buffer_free (xc_interface * *xch*, xc_hypercall_buffer_t * *b*)

4.131.4.4 void xc__hypercall_buffer_free_pages (xc_interface * *xch,* xc_hypercall_buffer_t * *b,* int *nr_pages*)

4.131.4.5 xc_hypercall_buffer_t XC__HYPERCALL_BUFFER_NAME (HYPERCALL_BUFFER_NULL)

4.131.4.6 int xc_acm_op (xc_interface * *xch,* int *cmd,* void * *arg,* unsigned long *arg_size*)

4.131.4.7 int xc_assign_device (xc_interface * *xch,* uint32_t *domid,* uint32_t *machine_bdf*)

4.131.4.8 int xc_availheap (xc_interface * *xch,* int *min_width,* int *max_width,* int *node,* uint64_t * *bytes*)

This function retrieves the the number of bytes available in the heap in a specific range of address-widths and nodes.

xch a handle to an open hypervisor interface domid the domain to query min_width the smallest address width to query (0 if don't care) max_width the largest address width to query (0 if don't care) node the node to query (-1 for all) *bytes caller variable to put total bytes counted

Returns

0 on success, <0 on failure.

4.131.4.9 int xc_caas_exempt (xc_interface * *xc_handle,* uint64_t *mfn*)

4.131.4.10 int xc_caas_op (xc_interface * *xch,* caas_op_t * *op*)

4.131.4.11 int xc_clear_domain_page (xc_interface * *xch,* uint32_t *domid,* unsigned long *dst_pfn*)

4.131.4.12 void xc_clear_last_error (xc_interface * *xch*)

4.131.4.13 int xc_copy_to_domain_page (xc_interface * *xch,* uint32_t *domid,* unsigned long *dst_pfn,* const char * *src_page*)

4.131.4.14 int xc_cpu_offline (xc_interface * *xch,* int *cpu*)

4.131.4.15 int xc_cpu_online (xc_interface * *xch,* int *cpu*)

4.131.4.16 xc_cpumap_t xc_cpumap_alloc (xc_interface * *xch*)

4.131.4.17 int xc_cpupool_addcpu (xc_interface * *xch,* uint32_t *poolid,* int *cpu*)

Add cpu to a cpupool. cpu may be -1 indicating the first unassigned.

xc_handle a handle to an open hypervisor interface poolid id of the cpupool cpu cpu number to add return 0 on success, -1 on failure

4.131.4.18 int xc_cpupool_create (xc_interface * *xch,* uint32_t * *ppoolid,* uint32_t *sched_id*)

Create a new cpupool.

xc_handle a handle to an open hypervisor interface ppoolid pointer to the new cpupool id (in/out) sched_id id of scheduler to use for pool return 0 on success, -1 on failure

4.131.4.19 int xc_cpupool_destroy (xc_interface * *xch,* uint32_t *poolid*)

Destroy a cpupool. Pool must be unused and have no cpu assigned.

xc_handle a handle to an open hypervisor interface poolid id of the cpupool to destroy return 0 on success, -1 on failure

4.131.4.20 xc_cpumap_t xc_cpupool_freeinfo (xc_interface * *xch*)

Return map of cpus not in any cpupool.

xc_handle a handle to an open hypervisor interface return cpumap array on success, NULL else

4.131.4.21 xc_cpupoolinfo_t* xc_cpupool_getinfo (xc_interface * *xch,* uint32_t *poolid*)

Get cpupool info. Returns info for up to the specified number of cpupools starting at the given id. xc_handle a handle to an open hypervisor interface poolid lowest id for which info is returned return cpupool info ptr (to be freed via xc_cpupool_infofree)

4.131.4.22 void xc_cpupool_infofree (xc_interface * *xch,* xc_cpupoolinfo_t * *info*)

Free cpupool info. Used to free info obtained via xc_cpupool_getinfo. xc_handle a handle to an open hypervisor interface info area to free

4.131.4.23 int xc_cpupool_movedomain (xc_interface * *xch,* uint32_t *poolid,* uint32_t *domid*)

Move domain to another cpupool.

xc_handle a handle to an open hypervisor interface poolid id of the destination cpupool domid id of the domain to move return 0 on success, -1 on failure

4.131.4.24 int xc_cpupool_removecpu (xc_interface * *xch,* uint32_t *poolid,* int *cpu*)

Remove cpu from cpupool. cpu may be -1 indicating the last cpu of the pool.

xc_handle a handle to an open hypervisor interface poolid id of the cpupool cpu cpu number to remove return 0 on success, -1 on failure

4.131.4.25 int xc_deassign_device (xc_interface * *xch,* uint32_t *domid,* uint32_t *machine_bdf*)

4.131.4.26 int xc_disable_turbo (xc_interface * *xch,* int *cpuid*)

4.131.4.27 int xc_domain_add_to_physmap (xc_interface * *xch,* uint32_t *domid,* unsigned int *space,* unsigned long *idx,* xen_pfn_t *gpfn*)

4.131.4.28 int xc_domain_bind_pt_irq (xc_interface * *xch,* uint32_t *domid,* uint8_t *machine_irq,* uint8_t *irq_type,* uint8_t *bus,* uint8_t *device,* uint8_t *intx,* uint8_t *isa_irq*)

4.131.4.29 int xc_domain_bind_pt_isa_irq (xc_interface * *xch,* uint32_t *domid,* uint8_t *machine_irq*)

4.131.4.30 int xc_domain_bind_pt_pci_irq (xc_interface * *xch,* uint32_t *domid,* uint8_t *machine_irq,* uint8_t *bus,* uint8_t *device,* uint8_t *intx*)

4.131.4.31 int xc_domain_create (xc_interface * *xch,* uint32_t *ssidref,* xen_domain_handle_t *handle,* uint32_t *flags,* uint32_t * *pdomid*)

4.131.4.32 int xc_domain_debug_control (xc_interface * *xch,* uint32_t *domid,* uint32_t *sop,* uint32_t *vcpu*)

4.131.4.33 int xc_domain_decrease_reservation (xc_interface * *xch,* uint32_t *domid,* unsigned long *nr_extents,* unsigned int *extent_order,* xen_pfn_t * *extent_start*)

4.131.4.34 int xc_domain_decrease_reservation_exact (xc_interface * *xch,* uint32_t *domid,* unsigned long *nr_extents,* unsigned int *extent_order,* xen_pfn_t * *extent_start*)

4.131.4.35 int xc_domain_destroy (xc_interface * *xch,* uint32_t *domid*)

This function will destroy a domain. Destroying a domain removes the domain completely from memory. This function should be called after sending the domain a SHUTDOWN control message to free up the domain resources.

xch a handle to an open hypervisor interface domid the domain id to destroy

Returns

0 on success, -1 on failure

4.131.4.36 int xc_domain_disable_migrate (xc_interface * *xch,* uint32_t *domid*)

4.131.4.37 int xc_domain_dumpcore (xc_interface * *xch,* uint32_t *domid,* const char * *corename*)

4.131.4.38 int xc_domain_dumpcore_via_callback (xc_interface * *xch,* uint32_t *domid,* void * *arg,* dumpcore_rtn_t *dump_rtn*)

4.131.4.39 long long xc_domain_get_cpu_usage (xc_interface * *xch,* domid_t *domid,* int *vcpu*)

4.131.4.40 int xc_domain_get_machine_address_size (xc_interface * *xch,* uint32_t *domid*)

4.131.4.41 int xc_domain_get_pod_target (xc_interface * *xch,* uint32_t *domid,* uint64_t * *tot_pages,* uint64_t * *pod_cache_pages,* uint64_t * *pod_entries*)

4.131.4.42 int xc_domain_get_tsc_info (xc_interface * *xch,* uint32_t *domid,* uint32_t * *tsc_mode,* uint64_t * *elapsed_nsec,* uint32_t * *gtsc_khz,* uint32_t * *incarnation*)

4.131.4.43 int xc_domain_getinfo (xc_interface * *xch,* uint32_t *first_domid,* unsigned int *max_doms,* xc_dominfo_t * *info*)

This function will return information about one or more domains. It is designed to iterate over the list of domains. If a single domain is requested, this function will return the next domain in the list - if one exists. It is, therefore, important in this case to make sure the domain requested was the one returned.

xch a handle to an open hypervisor interface first_domid the first domain to enumerate information from. Domains are currently enumerate in order of creation. max_doms the number of elements in info info an array of max_doms size that will contain the information for the enumerated domains.

Returns

the number of domains enumerated or -1 on error

4.131.4.44 int xc_domain_getinfolist (xc_interface * *xch,* uint32_t *first_domain,* unsigned int *max_domains,* xc_domaininfo_t * *info*)

This function will return information about one or more domains, using a single hypercall. The domain information will be stored into the supplied array of xc_domaininfo_t structures.

xch a handle to an open hypervisor interface first_domain the first domain to enumerate information from. Domains are currently enumerate in order of creation. max_domains the number of elements in info info an array of max_doms size that will contain the information for the enumerated domains.

Returns

the number of domains enumerated or -1 on error

4.131.4.45 int xc_domain_hvm_getcontext (xc_interface * *xch,* uint32_t *domid,* uint8_t * *ctxt_buf,* uint32_t *size*)

This function returns information about the context of a hvm domain xch a handle to an open hypervisor interface domid the domain to get information from ctxt_buf a pointer to a structure to store the execution context of the hvm domain size the size of ctxt_buf in bytes

Returns

0 on success, -1 on failure

4.131.4.46 int xc_domain_hvm_getcontext_partial (xc_interface * *xch,* uint32_t *domid,* uint16_t *typecode,* uint16_t *instance,* void * *ctxt_buf,* uint32_t *size*)

This function returns one element of the context of a hvm domain xch a handle to an open hypervisor interface domid the domain to get information from typecode which type of elemnt required instance which instance of the type ctxt_buf a pointer to a structure to store the execution context of the hvm domain size the size of ctxt_buf (must be >= HVM_SAVE_LENGTH(typecode))

Returns

0 on success, -1 on failure

4.131.4.47 int xc_domain_hvm_setcontext (xc_interface * *xch,* uint32_t *domid,* uint8_t * *hvm_ctxt,* uint32_t *size*)

This function will set the context for hvm domain

xch a handle to an open hypervisor interface domid the domain to set the hvm domain context for hvm_ctxt pointer to the the hvm context with the values to set size the size of hvm_ctxt in bytes

Returns

0 on success, -1 on failure

4.131.4.48 int xc_domain_increase_reservation (xc_interface * *xch,* uint32_t *domid,* unsigned long *nr_extents,* unsigned int *extent_order,* unsigned int *mem_flags,* xen_pfn_t * *extent_start*)

4.131.4.49 int xc_domain_increase_reservation_exact (xc_interface * *xch,* uint32_t *domid,* unsigned long *nr_extents,* unsigned int *extent_order,* unsigned int *mem_flags,* xen_pfn_t * *extent_start*)

4.131.4.50 int xc_domain_iomem_permission (xc_interface * *xch,* uint32_t *domid,* unsigned long *first_mfn,* unsigned long *nr_mfns,* uint8_t *allow_access*)

4.131.4.51 int xc_domain_ioport_mapping (xc_interface * *xch,* uint32_t *domid,* uint32_t *first_gport,* uint32_t *first_mport,* uint32_t *nr_ports,* uint32_t *add_mapping*)

4.131.4.52 int xc_domain_ioport_permission (xc_interface * *xch,* uint32_t *domid,* uint32_t *first_port,* uint32_t *nr_ports,* uint32_t *allow_access*)

4.131.4.53 int xc_domain_irq_permission (xc_interface * *xch,* uint32_t *domid,* uint8_t *pirq,* uint8_t *allow_access*)

4.131.4.54 int xc_domain_max_vcpus (xc_interface * *xch,* uint32_t *domid,* unsigned int *max*)

4.131.4.55 int xc_domain_maximum_gpfn (xc_interface * *xch,* domid_t *domid*)

4.131.4.56 int xc_domain_memory_exchange_pages (xc_interface * *xch,* int *domid,* unsigned long *nr_in_extents,* unsigned int *in_order,* xen_pfn_t * *in_extents,* unsigned long *nr_out_extents,* unsigned int *out_order,* xen_pfn_t * *out_extents*)

4.131.4.57 int xc_domain_memory_mapping (xc_interface * *xch,* uint32_t *domid,* unsigned long *first_gfn,* unsigned long *first_mfn,* unsigned long *nr_mfns,* uint32_t *add_mapping*)

4.131.4.58 int xc_domain_pause (xc_interface * *xch,* uint32_t *domid*)

This function pauses a domain. A paused domain still exists in memory however it does not receive any timeslices from the hypervisor.

xch a handle to an open hypervisor interface domid the domain id to pause

Returns

0 on success, -1 on failure.

4.131.4.59 int xc_domain_pin_memory_cacheattr (xc_interface * *xch,* uint32_t *domid,* uint64_t *start,* uint64_t *end,* uint32_t *type*)

4.131.4.60 int xc_domain_populate_physmap (xc_interface * *xch,* uint32_t *domid,* unsigned long *nr_extents,* unsigned int *extent_order,* unsigned int *mem_flags,* xen_pfn_t * *extent_start*)

4.131.4.61 int xc_domain_populate_physmap_exact (xc_interface * *xch,* uint32_t *domid,* unsigned long *nr_extents,* unsigned int *extent_order,* unsigned int *mem_flags,* xen_pfn_t * *extent_start*)

4.131.4.62 int xc_domain_resume (xc_interface * *xch,* uint32_t *domid,* int *fast*)

This function resumes a suspended domain. The domain should have been previously suspended.

xch a handle to an open hypervisor interface domid the domain id to resume fast use cooperative resume (guest must support this) return 0 on success, -1 on failure

4.131.4.63 int xc_domain_send_trigger (xc_interface * *xch,* uint32_t *domid,* uint32_t *trigger,* uint32_t *vcpu*)

This function sends a trigger to a domain.

xch a handle to an open hypervisor interface domid the domain id to send trigger trigger the trigger type vcpu the vcpu number to send trigger return 0 on success, -1 on failure

4.131.4.64 int xc_domain_set_access_required (xc_interface * *xch,* uint32_t *domid,* unsigned int *required*)

This function sets or clears the requirement that an access memory event listener is required on the domain.

xch a handle to an open hypervisor interface domid the domain id to send trigger enable true to require a listener return 0 on success, -1 on failure

4.131.4.65 int xc_domain_set_machine_address_size (xc_interface * *xch,* uint32_t *domid,* unsigned int *width*)

4.131.4.66 int xc_domain_set_memmap_limit (xc_interface * *xch,* uint32_t *domid,* unsigned long *map_limitkb*)

4.131.4.67 int xc_domain_set_pod_target (xc_interface * *xch,* uint32_t *domid,* uint64_t *target_pages,* uint64_t * *tot_pages,* uint64_t * *pod_cache_pages,* uint64_t * *pod_entries*)

4.131.4.68 int xc_domain_set_target (xc_interface * *xch,* uint32_t *domid,* uint32_t *target*)

4.131.4.69 int xc_domain_set_time_offset (xc_interface * *xch,* uint32_t *domid,* int32_t *time_offset_seconds*)

4.131.4.70 int xc_domain_set_tsc_info (xc_interface * *xch,* uint32_t *domid,* uint32_t *tsc_mode,* uint64_t *elapsed_nsec,* uint32_t *gtsc_khz,* uint32_t *incarnation*)

4.131.4.71 int xc_domain_setdebugging (xc_interface * *xch,* uint32_t *domid,* unsigned int *enable*)

This function enables or disable debugging of a domain.

xch a handle to an open hypervisor interface domid the domain id to send trigger enable true to enable debugging return 0 on success, -1 on failure

4.131.4.72 int xc_domain_sethandle (xc_interface * *xch,* uint32_t *domid,* xen_domain_handle_t *handle*)

4.131.4.73 int xc_domain_setmaxmem (xc_interface * *xch,* uint32_t *domid,* unsigned int *max_memkb*)

4.131.4.74 int xc_domain_shutdown (xc_interface * *xch,* uint32_t *domid,* int *reason*)

This function will shutdown a domain. This is intended for use in fully-virtualized domains where this operation is analogous to the sched_op operations in a paravirtualized domain. The caller is expected to give the reason for the shutdown.

xch a handle to an open hypervisor interface domid the domain id to destroy reason is the reason (SHUTDOWN_-xxx) for the shutdown

Returns

0 on success, -1 on failure

4.131.4.75 int xc_domain_subscribe_for_suspend (xc_interface * *xch,* domid_t *domid,* evtchn_port_t *port*)

4.131.4.76 int xc_domain_suppress_spurious_page_faults (xc_interface * *xch,* uint32_t *domid*)

4.131.4.77 int xc_domain_unbind_msi_irq (xc_interface * *xch,* uint32_t *domid,* uint32_t *gvec,* uint32_t *pirq,* uint32_t *gflags*)

4.131.4.78 int xc_domain_unbind_pt_irq (xc_interface * *xch,* uint32_t *domid,* uint8_t *machine_irq,* uint8_t *irq_type,* uint8_t *bus,* uint8_t *device,* uint8_t *intx,* uint8_t *isa_irq*)

4.131.4.79 int xc_domain_unpause (xc_interface * *xch,* uint32_t *domid*)

This function unpauses a domain. The domain should have been previously paused.

xch a handle to an open hypervisor interface domid the domain id to unpause return 0 on success, -1 on failure

4.131.4.80 int xc_domain_update_msi_irq (xc_interface * *xch,* uint32_t *domid,* uint32_t *gvec,* uint32_t *pirq,* uint32_t *gflags,* uint64_t *gtable*)

4.131.4.81 int xc_domctl (xc_interface * *xch,* struct xen_domctl * *domctl*)

4.131.4.82 void xc_elf_set_logfile (xc_interface * *xch,* struct elf_binary * *elf,* int *verbose*)

4.131.4.83 int xc_enable_turbo (xc_interface * *xch,* int *cpuid*)

4.131.4.84 const char* xc_error_code_to_desc (int *code*)

4.131.4.85 evtchn_port_or_error_t xc_evtchn_alloc_unbound (xc_interface * *xch*, uint32_t *dom*, uint32_t *remote_dom*)

This function allocates an unbound port. Ports are named endpoints used for interdomain communication. This function is most useful in opening a well-known port within a domain to receive events on.

NOTE: If you are allocating a *local* unbound port, you probably want to use xc_evtchn_bind_unbound_port(). This function is intended for allocating ports *only* during domain creation.

xch a handle to an open hypervisor interface dom the ID of the local domain (the 'allocatee') remote_dom the ID of the domain who will later bind

Returns

allocated port (in) on success, -1 on failure

4.131.4.86 evtchn_port_or_error_t xc_evtchn_bind_interdomain (xc_evtchn * *xce*, int *domid*, evtchn_port_t *remote_port*)

4.131.4.87 evtchn_port_or_error_t xc_evtchn_bind_unbound_port (xc_evtchn * *xce*, int *domid*)

4.131.4.88 evtchn_port_or_error_t xc_evtchn_bind_virq (xc_evtchn * *xce*, unsigned int *virq*)

4.131.4.89 int xc_evtchn_close (xc_evtchn * *xce*)

4.131.4.90 int xc_evtchn_fd (xc_evtchn * *xce*)

4.131.4.91 int xc_evtchn_notify (xc_evtchn * *xce*, evtchn_port_t *port*)

4.131.4.92 xc_evtchn* xc_evtchn_open (xentoollog_logger * *logger*, unsigned *open_flags*)

4.131.4.93 evtchn_port_or_error_t xc_evtchn_pending (xc_evtchn * *xce*)

4.131.4.94 int xc_evtchn_reset (xc_interface * *xch*, uint32_t *dom*)

4.131.4.95 int xc_evtchn_status (xc_interface * *xch*, xc_evtchn_status_t * *status*)

4.131.4.96 int xc_evtchn_unbind (xc_evtchn * *xce*, evtchn_port_t *port*)

4.131.4.97 int xc_evtchn_unmask (xc_evtchn * *xce*, evtchn_port_t *port*)

4.131.4.98 int xc_flask_access (xc_interface * *xc_handle*, const char * *scon*, const char * *tcon*, uint16_t *tclass*, uint32_t *req*, uint32_t * *allowed*, uint32_t * *decided*, uint32_t * *auditallow*, uint32_t * *auditdeny*, uint32_t * *seqno*)

4.131.4.99 int xc_flask_add_device (xc_interface * *xc_handle*, unsigned long *device*, char * *scontext*)

4.131.4.100 int xc_flask_add_iomem (xc_interface * *xc_handle*, unsigned long *low*, unsigned long *high*, char * *scontext*)

4.131.4.101 int xc_flask_add_ioport (xc_interface * *xc_handle*, unsigned long *low*, unsigned long *high*, char * *scontext*)

4.131.4.102 int xc_flask_add_pirq (xc_interface * *xc_handle*, unsigned int *pirq*, char * *scontext*)

4.131.4.103 int xc_flask_avc_cachestats (xc_interface * *xc_handle*, char * *buf*, int *size*)

4.131.4.104 int xc_flask_avc_hashstats (xc_interface * *xc_handle*, char * *buf*, int *size*)

4.131.4.105 int xc_flask_context_to_sid (xc_interface * *xc_handle*, char * *buf*, uint32_t *size*, uint32_t * *sid*)

4.131.4.106 int xc_flask_del_device (xc_interface * *xc_handle,* unsigned long *device*)

4.131.4.107 int xc_flask_del_iomem (xc_interface * *xc_handle,* unsigned long *low,* unsigned long *high*)

4.131.4.108 int xc_flask_del_ioport (xc_interface * *xc_handle,* unsigned long *low,* unsigned long *high*)

4.131.4.109 int xc_flask_del_pirq (xc_interface * *xc_handle,* unsigned int *pirq*)

4.131.4.110 int xc_flask_getavc_threshold (xc_interface * *xc_handle*)

4.131.4.111 int xc_flask_getenforce (xc_interface * *xc_handle*)

4.131.4.112 int xc_flask_load (xc_interface * *xc_handle,* char * *buf,* uint32_t *size*)

4.131.4.113 int xc_flask_op (xc_interface * *xch,* flask_op_t * *op*)

4.131.4.114 int xc_flask_policyvers (xc_interface * *xc_handle,* char * *buf,* int *size*)

4.131.4.115 int xc_flask_setavc_threshold (xc_interface * *xc_handle,* int *threshold*)

4.131.4.116 int xc_flask_setenforce (xc_interface * *xc_handle,* int *mode*)

4.131.4.117 int xc_flask_sid_to_context (xc_interface * *xc_handle,* int *sid,* char * *buf,* uint32_t *size*)

4.131.4.118 int xc_get_cpufreq_avgfreq (xc_interface * *xch,* int *cpuid,* int * *avg_freq*)

4.131.4.119 int xc_get_cpufreq_para (xc_interface * *xch,* int *cpuid,* struct xc_get_cpufreq_para * *user_para*)

4.131.4.120 int xc_get_cpuidle_max_cstate (xc_interface * *xch,* uint32_t * *value*)

4.131.4.121 int xc_get_cpumap_size (xc_interface * *xch*)

4.131.4.122 int xc_get_device_group (xc_interface * *xch,* uint32_t *domid,* uint32_t *machine_bdf,* uint32_t *max_sdevs,* uint32_t * *num_sdevs,* uint32_t * *sdev_array*)

4.131.4.123 int xc_get_hvm_param (xc_interface * *handle,* domid_t *dom,* int *param,* unsigned long * *value*)

4.131.4.124 const xc_error* xc_get_last_error (xc_interface * *handle*)

4.131.4.125 int xc_get_max_cpus (xc_interface * *xch*)

4.131.4.126 int xc_get_pfn_list (xc_interface * *xch,* uint32_t *domid,* uint64_t * *pfn_buf,* unsigned long *max_pfns*)

DEPRECATED. Avoid using this, as it does not correctly account for PFNs without a backing MFN.

4.131.4.127 long xc_get_tot_pages (xc_interface * *xch,* uint32_t *domid*)

4.131.4.128 int xc_get_vcpu_migration_delay (xc_interface * *xch,* uint32_t * *value*)

4.131.4.129 int xc_getcpuinfo (xc_interface * *xch,* int *max_cpus,* xc_cpuinfo_t * *info,* int * *nr_cpus*)

4.131.4.130 int xc_gnttab_close (xc_gnttab * *xcg*)

4.131.4.131 int xc_gnttab_get_version (xc_interface * *xch,* int *domid*)

4.131.4.132 void* xc_gnttab_map_domain_grant_refs (xc_gnttab * *xcg,* uint32_t *count,* uint32_t *domid,* uint32_t * *refs,* int *prot*)

Memory maps one or more grant references from one domain to a contiguous local address range. Mappings should be unmapped with xc_gnttab_munmap. Logs errors.

xcg a handle on an open grant table interface count the number of grant references to be mapped domid the domain to map memory from refs an array of grant references to be mapped prot same flag as in mmap()

4.131.4.133 void* xc_gnttab_map_grant_ref (xc_gnttab * *xcg,* uint32_t *domid,* uint32_t *ref,* int *prot*)

4.131.4.134 void* xc_gnttab_map_grant_refs (xc_gnttab * *xcg,* uint32_t *count,* uint32_t * *domids,* uint32_t * *refs,* int *prot*)

Memory maps one or more grant references from one or more domains to a contiguous local address range. Mappings should be unmapped with xc_gnttab_munmap. Logs errors.

xcg a handle on an open grant table interface count the number of grant references to be mapped domids an array of domain IDs by which the corresponding were granted refs an array of grant references to be mapped prot same flag as in mmap()

4.131.4.135 grant_entry_v1_t* xc_gnttab_map_table_v1 (xc_interface * *xch,* int *domid,* int * *gnt_num*)

4.131.4.136 grant_entry_v2_t* xc_gnttab_map_table_v2 (xc_interface * *xch,* int *domid,* int * *gnt_num*)

4.131.4.137 int xc_gnttab_munmap (xc_gnttab * *xcg,* void * *start_address,* uint32_t *count*)

4.131.4.138 int xc_gnttab_op (xc_interface * *xch,* int *cmd,* void * *op,* int *op_size,* int *count*)

4.131.4.139 xc_gnttab* xc_gnttab_open (xentoollog_logger * *logger,* unsigned *open_flags*)

4.131.4.140 int xc_gnttab_set_max_grants (xc_gnttab * *xcg,* uint32_t *count*)

4.131.4.141 int xc_hvm_get_mem_access (xc_interface * *xch,* domid_t *dom,* uint64_t *pfn,* hvmmem_access_t * *memaccess*)

4.131.4.142 int xc_hvm_inject_trap (xc_interface * *xch,* domid_t *dom,* int *vcpu,* uint32_t *trap,* uint32_t *error_code,* uint64_t *cr2*)

4.131.4.143 int xc_hvm_modified_memory (xc_interface * *xch,* domid_t *dom,* uint64_t *first_pfn,* uint64_t *nr*)

4.131.4.144 int xc_hvm_set_isa_irq_level (xc_interface * *xch,* domid_t *dom,* uint8_t *isa_irq,* unsigned int *level*)

4.131.4.145 int xc_hvm_set_mem_access (xc_interface * *xch,* domid_t *dom,* hvmmem_access_t *memaccess,* uint64_t *first_pfn,* uint64_t *nr*)

4.131.4.146 int xc_hvm_set_mem_type (xc_interface * *xch,* domid_t *dom,* hvmmem_type_t *memtype,* uint64_t *first_pfn,* uint64_t *nr*)

4.131.4.147 int xc_hvm_set_pci_intx_level (xc_interface * *xch,* domid_t *dom,* uint8_t *domain,* uint8_t *bus,* uint8_t *device,* uint8_t *intx,* unsigned int *level*)

4.131.4.148 int xc_hvm_set_pci_link_route (xc_interface * *xch,* domid_t *dom,* uint8_t *link,* uint8_t *isa_irq*)

4.131.4.149 int xc_hvm_track_dirty_vram (xc_interface * *xch,* domid_t *dom,* uint64_t *first_pfn,* uint64_t *nr,* unsigned long * *bitmap*)

4.131.4.150 unsigned long xc_ia64_fpsr_default (void)

4.131.4.151 int xc_ia64_nvram_init (xc_interface * *xch,* char * *dom_name,* uint32_t *dom*)

4.131.4.152 int xc_ia64_save_to_nvram (xc_interface * *xch,* uint32_t *dom*)

4.131.4.153 int xc_ia64_set_os_type (xc_interface * *xch,* char * *guest_os_type,* uint32_t *dom*)

4.131.4.154 int xc_interface_close (xc_interface * *xch*)

This function closes an open hypervisor interface.

This function can fail if the handle does not represent an open interface or if there were problems closing the interface. In the latter case the interface is still closed.

xch a handle to an open hypervisor interface

Returns

0 on success, -1 otherwise.

4.131.4.155 int xc_interface_is_fake (void)

Query the active OS interface (i.e. that which would be returned by xc_interface_open) to find out if it is fake (i.e. backends onto something other than an actual Xen hypervisor).

Returns

0 is "real", >0 if fake, -1 on error.

4.131.4.156 xc_interface* xc_interface_open (xentoollog_logger * *logger,* xentoollog_logger * *dombuild_logger,* unsigned *open_flags*)

This function opens a handle to the hypervisor interface. This function can be called multiple times within a single process. Multiple processes can have an open hypervisor interface at the same time.

Each call to this function should have a corresponding call to xc_interface_close().

This function can fail if the caller does not have superuser permission or if a Xen-enabled kernel is not currently running.

Returns

a handle to the hypervisor interface

4.131.4.157 int xc_lockprof_query (xc_interface * *xch,* uint32_t * *n_elems,* uint64_t * *time,* xc_hypercall_buffer_t * *data*)

4.131.4.158 int xc_lockprof_query_number (xc_interface * *xch,* uint32_t * *n_elems*)

4.131.4.159 int xc_lockprof_reset (xc_interface * *xch*)

4.131.4.160 int xc_machphys_mfn_list (xc_interface * *xch,* unsigned long *max_extents,* xen_pfn_t * *extent_start*)

4.131.4.161 unsigned long xc_make_page_below_4G (xc_interface * *xch,* uint32_t *domid,* unsigned long *mfn*)

4.131.4.162 void* xc_map_foreign_batch (xc_interface * *xch*, uint32_t *dom*, int *prot*, xen_pfn_t * *arr*, int *num*)

DEPRECATED - use xc_map_foreign_bulk() instead.

Like xc_map_foreign_pages(), except it can succeeed partially. When a page cannot be mapped, its PFN in is or'ed with 0xF0000000 to indicate the error.

4.131.4.163 void* xc_map_foreign_bulk (xc_interface * *xch*, uint32_t *dom*, int *prot*, const xen_pfn_t * *arr*, int * *err*, unsigned int *num*)

Like xc_map_foreign_pages(), except it can succeed partially. When a page cannot be mapped, its respective field in is set to the corresponding errno value.

4.131.4.164 void* xc_map_foreign_pages (xc_interface * *xch*, uint32_t *dom*, int *prot*, const xen_pfn_t * *arr*, int *num*)

4.131.4.165 void* xc_map_foreign_range (xc_interface * *xch*, uint32_t *dom*, int *size*, int *prot*, unsigned long *mfn*)

Memory maps a range within one domain to a local address range. Mappings should be unmapped with munmap and should follow the same rules as mmap regarding page alignment. Returns NULL on failure.

xch a handle on an open hypervisor interface dom the domain to map memory from size the amount of memory to map (in multiples of page size) prot same flag as in mmap(). mfn the frame address to map.

4.131.4.166 long xc_maximum_ram_page (xc_interface * *xch*)

4.131.4.167 int xc_mem_access_resume (xc_interface * *xch*, domid_t *domain_id*, unsigned long *gfn*)

4.131.4.168 int xc_mem_event_control (xc_interface * *xch*, domid_t *domain_id*, unsigned int *op*, unsigned int *mode*, void * *shared_page*, void * *ring_page*, unsigned long *gfn*)

mem_event operations

4.131.4.169 int xc_mem_event_disable (xc_interface * *xch*, domid_t *domain_id*)

4.131.4.170 int xc_mem_event_enable (xc_interface * *xch*, domid_t *domain_id*, void * *shared_page*, void * *ring_page*)

4.131.4.171 int xc_mem_paging_evict (xc_interface * *xch*, domid_t *domain_id*, unsigned long *gfn*)

4.131.4.172 int xc_mem_paging_nominate (xc_interface * *xch*, domid_t *domain_id*, unsigned long *gfn*)

4.131.4.173 int xc_mem_paging_prep (xc_interface * *xch*, domid_t *domain_id*, unsigned long *gfn*)

4.131.4.174 int xc_mem_paging_resume (xc_interface * *xch*, domid_t *domain_id*, unsigned long *gfn*)

4.131.4.175 int xc_memshr_control (xc_interface * *xch*, uint32_t *domid*, int *enable*)

memshr operations

4.131.4.176 int xc_memshr_debug_gfn (xc_interface * *xch*, uint32_t *domid*, unsigned long *gfn*)

4.131.4.177 int xc_memshr_debug_gref (xc_interface * *xch*, uint32_t *domid*, grant_ref_t *gref*)

4.131.4.178 int xc_memshr_debug_mfn (xc_interface * *xch*, uint32_t *domid*, unsigned long *mfn*)

4.131.4.179 int xc_memshr_domain_resume (**xc_interface** ∗ *xch,* uint32_t *domid*)

4.131.4.180 int xc_memshr_nominate_gfn (**xc_interface** ∗ *xch,* uint32_t *domid,* unsigned long *gfn,* uint64_t ∗ *handle*)

4.131.4.181 int xc_memshr_nominate_gref (**xc_interface** ∗ *xch,* uint32_t *domid,* grant_ref_t *gref,* uint64_t ∗ *handle*)

4.131.4.182 int xc_memshr_share (**xc_interface** ∗ *xch,* uint64_t *source_handle,* uint64_t *client_handle*)

4.131.4.183 int xc_mmuext_op (**xc_interface** ∗ *xch,* struct **mmuext_op** ∗ *op,* unsigned int *nr_ops,* **domid_t** *dom*)

4.131.4.184 int xc_numainfo (**xc_interface** ∗ *xch,* **xc_numainfo_t** ∗ *info*)

4.131.4.185 int xc_perfc_query (**xc_interface** ∗ *xch,* **xc_hypercall_buffer_t** ∗ *desc,* **xc_hypercall_buffer_t** ∗ *val*)

4.131.4.186 int xc_perfc_query_number (**xc_interface** ∗ *xch,* int ∗ *nbr_desc,* int ∗ *nbr_val*)

4.131.4.187 int xc_perfc_reset (**xc_interface** ∗ *xch*)

4.131.4.188 int xc_physdev_map_pirq (**xc_interface** ∗ *xch,* int *domid,* int *index,* int ∗ *pirq*)

4.131.4.189 int xc_physdev_map_pirq_msi (**xc_interface** ∗ *xch,* int *domid,* int *index,* int ∗ *pirq,* int *devfn,* int *bus,* int *entry_nr,* uint64_t *table_base*)

4.131.4.190 int xc_physdev_pci_access_modify (**xc_interface** ∗ *xch,* uint32_t *domid,* int *bus,* int *dev,* int *func,* int *enable*)

4.131.4.191 int xc_physdev_unmap_pirq (**xc_interface** ∗ *xch,* int *domid,* int *pirq*)

4.131.4.192 int xc_physinfo (**xc_interface** ∗ *xch,* **xc_physinfo_t** ∗ *info*)

4.131.4.193 int xc_pm_get_cxstat (**xc_interface** ∗ *xch,* int *cpuid,* struct **xc_cx_stat** ∗ *cxpt*)

4.131.4.194 int xc_pm_get_max_cx (**xc_interface** ∗ *xch,* int *cpuid,* int ∗ *max_cx*)

4.131.4.195 int xc_pm_get_max_px (**xc_interface** ∗ *xch,* int *cpuid,* int ∗ *max_px*)

4.131.4.196 int xc_pm_get_pxstat (**xc_interface** ∗ *xch,* int *cpuid,* struct **xc_px_stat** ∗ *pxpt*)

4.131.4.197 int xc_pm_reset_cxstat (**xc_interface** ∗ *xch,* int *cpuid*)

4.131.4.198 int xc_pm_reset_pxstat (**xc_interface** ∗ *xch,* int *cpuid*)

4.131.4.199 int xc_readconsolering (**xc_interface** ∗ *xch,* char ∗ *buffer,* unsigned int ∗ *pnr_chars,* int *clear,* int *incremental,* uint32_t ∗ *pindex*)

4.131.4.200 int xc_sched_arinc653_schedule_get (**xc_interface** ∗ *xch,* struct xen_sysctl_arinc653_schedule ∗ *schedule*)

4.131.4.201 int xc_sched_arinc653_schedule_set (**xc_interface** ∗ *xch,* struct xen_sysctl_arinc653_schedule ∗ *schedule*)

4.131.4.202 int xc_sched_credit2_domain_get (**xc_interface** ∗ *xch,* uint32_t *domid,* struct xen_domctl_sched_credit2 ∗ *sdom*)

4.131.4.203 int xc_sched_credit2_domain_set (**xc_interface** ∗ *xch,* uint32_t *domid,* struct xen_domctl_sched_credit2 ∗ *sdom*)

4.131.4.204 int xc_sched_credit_domain_get (**xc_interface** ∗ *xch,* uint32_t *domid,* struct xen_domctl_sched_credit ∗ *sdom*)

4.131.4.205 int xc_sched_credit_domain_set (xc_interface * *xch,* uint32_t *domid,* struct xen_domctl_sched_credit * *sdom*)

4.131.4.206 int xc_sched_id (xc_interface * *xch,* int * *sched_id*)

4.131.4.207 int xc_sedf_domain_get (xc_interface * *xch,* uint32_t *domid,* uint64_t * *period,* uint64_t * *slice,* uint64_t * *latency,* uint16_t * *extratime,* uint16_t * *weight*)

4.131.4.208 int xc_sedf_domain_set (xc_interface * *xch,* uint32_t *domid,* uint64_t *period,* uint64_t *slice,* uint64_t *latency,* uint16_t *extratime,* uint16_t *weight*)

4.131.4.209 int xc_send_debug_keys (xc_interface * *xch,* char * *keys*)

4.131.4.210 int xc_set_cpufreq_gov (xc_interface * *xch,* int *cpuid,* char * *govname*)

4.131.4.211 int xc_set_cpufreq_para (xc_interface * *xch,* int *cpuid,* int *ctrl_type,* int *ctrl_value*)

4.131.4.212 int xc_set_cpuidle_max_cstate (xc_interface * *xch,* uint32_t *value*)

4.131.4.213 int xc_set_hvm_param (xc_interface * *handle,* domid_t *dom,* int *param,* unsigned long *value*)

4.131.4.214 int xc_set_sched_opt_smt (xc_interface * *xch,* uint32_t *value*)

4.131.4.215 int xc_set_vcpu_migration_delay (xc_interface * *xch,* uint32_t *value*)

4.131.4.216 int xc_shadow_control (xc_interface * *xch,* uint32_t *domid,* unsigned int *sop,* xc_hypercall_buffer_t * *dirty_bitmap,* unsigned long *pages,* unsigned long * *mb,* uint32_t *mode,* xc_shadow_op_stats_t * *stats*)

4.131.4.217 const char* xc_strerror (xc_interface * *xch,* int *errcode*)

4.131.4.218 int xc_sysctl (xc_interface * *xch,* struct xen_sysctl * *sysctl*)

4.131.4.219 int xc_tbuf_disable (xc_interface * *xch*)

4.131.4.220 int xc_tbuf_enable (xc_interface * *xch,* unsigned long *pages,* unsigned long * *mfn,* unsigned long * *size*)

xc_tbuf_enable - enable tracing buffers

xch a handle to an open hypervisor interface cnt size of tracing buffers to create (in pages) mfn location to store mfn of the trace buffers to size location to store the size (in bytes) of a trace buffer to

Gets the machine address of the trace pointer area and the size of the per CPU buffers.

4.131.4.221 int xc_tbuf_get_size (xc_interface * *xch,* unsigned long * *size*)

This function retrieves the current size of the trace buffers. Note that the size returned is in terms of bytes, not pages.

xch a handle to an open hypervisor interface size will contain the size in bytes for the trace buffers

Returns

0 on success, -1 on failure.

4.131.4.222 int xc_tbuf_set_cpu_mask (xc_interface * *xch,* uint32_t *mask*)

4.131.4.223 int xc_tbuf_set_evt_mask (xc_interface * *xch,* uint32_t *mask*)

4.131.4.224 int xc_tbuf_set_size (xc_interface * *xch,* unsigned long *size*)

This function sets the size of the trace buffers. Setting the size is currently a one-shot operation that may be performed either at boot time or via this interface, not both. The buffer size must be set before enabling tracing.

xch a handle to an open hypervisor interface size the size in pages per cpu for the trace buffers

Returns

0 on success, -1 on failure.

4.131.4.225 int xc_test_assign_device (xc_interface * *xch,* uint32_t *domid,* uint32_t *machine_bdf*)

4.131.4.226 int xc_tmem_auth (xc_interface * *xch,* int *cli_id,* char * *uuid_str,* int *arg1*)

4.131.4.227 int xc_tmem_control (xc_interface * *xch,* int32_t *pool_id,* uint32_t *subop,* uint32_t *cli_id,* uint32_t *arg1,* uint32_t *arg2,* uint64_t *arg3,* void * *buf*)

4.131.4.228 int xc_tmem_control_oid (xc_interface * *xch,* int32_t *pool_id,* uint32_t *subop,* uint32_t *cli_id,* uint32_t *arg1,* uint32_t *arg2,* struct tmem_oid *oid,* void * *buf*)

4.131.4.229 int xc_tmem_restore (xc_interface * *xch,* int *dom,* int *fd*)

4.131.4.230 int xc_tmem_restore_extra (xc_interface * *xch,* int *dom,* int *fd*)

4.131.4.231 int xc_tmem_save (xc_interface * *xch,* int *dom,* int *live,* int *fd,* int *field_marker*)

4.131.4.232 void xc_tmem_save_done (xc_interface * *xch,* int *dom*)

4.131.4.233 int xc_tmem_save_extra (xc_interface * *xch,* int *dom,* int *fd,* int *field_marker*)

4.131.4.234 int xc_topologyinfo (xc_interface * *xch,* xc_topologyinfo_t * *info*)

4.131.4.235 unsigned long xc_translate_foreign_address (xc_interface * *xch,* uint32_t *dom,* int *vcpu,* unsigned long long *virt*)

Translates a virtual address in the context of a given domain and vcpu returning the GFN containing the address (that is, an MFN for PV guests, a PFN for HVM guests). Returns 0 for failure.

xch a handle on an open hypervisor interface dom the domain to perform the translation in vcpu the vcpu to perform the translation on virt the virtual address to translate

4.131.4.236 int xc_vcpu_getaffinity (xc_interface * *xch,* uint32_t *domid,* int *vcpu,* xc_cpumap_t *cpumap*)

4.131.4.237 int xc_vcpu_getcontext (xc_interface * *xch,* uint32_t *domid,* uint32_t *vcpu,* vcpu_guest_context_any_t * *ctxt*)

This function returns information about the execution context of a particular vcpu of a domain.

xch a handle to an open hypervisor interface domid the domain to get information from vcpu the vcpu number ctxt a pointer to a structure to store the execution context of the domain

Returns

0 on success, -1 on failure

4.131.4.238 int xc_vcpu_getinfo (xc_interface * *xch*, uint32_t *domid*, uint32_t *vcpu*, xc_vcpuinfo_t * *info*)

4.131.4.239 int xc_vcpu_setaffinity (xc_interface * *xch*, uint32_t *domid*, int *vcpu*, xc_cpumap_t *cpumap*)

4.131.4.240 int xc_vcpu_setcontext (xc_interface * *xch*, uint32_t *domid*, uint32_t *vcpu*, vcpu_guest_context_any_t * *ctxt*)

This function will set the execution context for the specified vcpu.

xch a handle to an open hypervisor interface domid the domain to set the vcpu context for vcpu the vcpu number for the context ctxt pointer to the the cpu context with the values to set

Returns

the number of domains enumerated or -1 on error

4.131.4.241 int xc_version (xc_interface * *xch*, int *cmd*, void * *arg*)

4.131.4.242 int xc_watchdog (xc_interface * *xch*, uint32_t *id*, uint32_t *timeout*)

4.132 xen/tools/libxl/libxl.c File Reference

```
#include "libxl_osdeps.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/select.h>
#include <sys/wait.h>
#include <sys/time.h>
#include <signal.h>
#include <unistd.h>
#include <stdint.h>
#include <inttypes.h>
#include <assert.h>
#include "libxl.h"
#include "libxl_utils.h"
#include "libxl_internal.h"
#include "flexarray.h"
```

Macros

- #define PAGE_TO_MEMKB(pages) ((pages) * 4)
- #define BACKEND_STRING_SIZE 5
- #define STRINGIFY(x) #x
- #define TOSTRING(x) STRINGIFY(x)

Functions

- int libxl_ctx_init (libxl_ctx *ctx, int version, xentoollog_logger *lg)
- int libxl_ctx_free (libxl_ctx *ctx)
- void libxl_string_list_destroy (libxl_string_list *psl)

- void libxl_key_value_list_destroy (libxl_key_value_list *pkvl)
- int libxl_domain_rename (libxl_ctx *ctx, uint32_t domid, const char *old_name, const char *new_name, xs_transaction_t trans)
- int libxl_domain_resume (libxl_ctx *ctx, uint32_t domid)
- int libxl_domain_preserve (libxl_ctx *ctx, uint32_t domid, libxl_domain_create_info *info, const char *name_suffix, libxl_uuid new_uuid)
- libxl_dominfo * libxl_list_domain (libxl_ctx *ctx, int *nb_domain)
- int libxl_domain_info (libxl_ctx *ctx, libxl_dominfo *info_r, uint32_t domid)
- libxl_cpupoolinfo * libxl_list_cpupool (libxl_ctx *ctx, int *nb_pool)
- libxl_vminfo * libxl_list_vm (libxl_ctx *ctx, int *nb_vm)
- int libxl_domain_suspend (libxl_ctx *ctx, libxl_domain_suspend_info *info, uint32_t domid, int fd)
- int libxl_domain_pause (libxl_ctx *ctx, uint32_t domid)
- int libxl_domain_core_dump (libxl_ctx *ctx, uint32_t domid, const char *filename)
- int libxl_domain_unpause (libxl_ctx *ctx, uint32_t domid)
- int libxl_domain_shutdown (libxl_ctx *ctx, uint32_t domid, int req)
- int libxl_get_wait_fd (libxl_ctx *ctx, int *fd)
- int libxl_wait_for_domain_death (libxl_ctx *ctx, uint32_t domid, libxl_waiter *waiter)
- int libxl_wait_for_disk_ejects (libxl_ctx *ctx, uint32_t guest_domid, libxl_device_disk *disks, int num_disks, libxl_waiter *waiter)
- int libxl_get_event (libxl_ctx *ctx, libxl_event *event)
- int libxl_stop_waiting (libxl_ctx *ctx, libxl_waiter *waiter)
- int libxl_free_event (libxl_event *event)
- int libxl_free_waiter (libxl_waiter *waiter)
- int libxl_event_get_domain_death_info (libxl_ctx *ctx, uint32_t domid, libxl_event *event, libxl_dominfo *info)
- int libxl_event_get_disk_eject_info (libxl_ctx *ctx, uint32_t domid, libxl_event *event, libxl_device_disk *disk)
- int libxl_domain_destroy (libxl_ctx *ctx, uint32_t domid, int force)
- int libxl_console_exec (libxl_ctx *ctx, uint32_t domid, int cons_num, libxl_console_constype type)
- int libxl_primary_console_exec (libxl_ctx *ctx, uint32_t domid_vm)
- int libxl_vncviewer_exec (libxl_ctx *ctx, uint32_t domid, int autopass)
- int libxl_device_disk_add (libxl_ctx *ctx, uint32_t domid, libxl_device_disk *disk)
- int libxl_device_disk_del (libxl_ctx *ctx, libxl_device_disk *disk, int wait)
- char * libxl_device_disk_local_attach (libxl_ctx *ctx, libxl_device_disk *disk)
- int libxl_device_disk_local_detach (libxl_ctx *ctx, libxl_device_disk *disk)
- int libxl_device_nic_init (libxl_device_nic *nic_info, int devnum)
- int libxl_device_nic_add (libxl_ctx *ctx, uint32_t domid, libxl_device_nic *nic)
- int libxl_device_nic_del (libxl_ctx *ctx, libxl_device_nic *nic, int wait)
- libxl_nicinfo * libxl_list_nics (libxl_ctx *ctx, uint32_t domid, unsigned int *nb)
- void libxl_device_net2_init (libxl_device_net2 *net2_info, int devnum)
- int libxl_device_net2_add (libxl_ctx *ctx, uint32_t domid, libxl_device_net2 *net2)
- libxl_net2info * libxl_device_net2_list (libxl_ctx *ctx, uint32_t domid, unsigned int *nb)
- int libxl_device_net2_del (libxl_ctx *ctx, libxl_device_net2 *net2, int wait)
- int libxl_device_console_add (libxl_ctx *ctx, uint32_t domid, libxl_device_console *console)
- void libxl_device_vkb_init (libxl_device_vkb *vkb, int dev_num)
- int libxl_device_vkb_add (libxl_ctx *ctx, uint32_t domid, libxl_device_vkb *vkb)
- int libxl_device_vkb_clean_shutdown (libxl_ctx *ctx, uint32_t domid)
- int libxl_device_vkb_hard_shutdown (libxl_ctx *ctx, uint32_t domid)
- libxl_device_disk * libxl_device_disk_list (libxl_ctx *ctx, uint32_t domid, int *num)
- int libxl_device_disk_getinfo (libxl_ctx *ctx, uint32_t domid, libxl_device_disk *disk, libxl_diskinfo *diskinfo)
- int libxl_cdrom_insert (libxl_ctx *ctx, uint32_t domid, libxl_device_disk *disk)
- void libxl_device_vfb_init (libxl_device_vfb *vfb, int dev_num)
- int libxl_device_vfb_add (libxl_ctx *ctx, uint32_t domid, libxl_device_vfb *vfb)
- int libxl_device_vfb_clean_shutdown (libxl_ctx *ctx, uint32_t domid)
- int libxl_device_vfb_hard_shutdown (libxl_ctx *ctx, uint32_t domid)
- int libxl_domain_setmaxmem (libxl_ctx *ctx, uint32_t domid, uint32_t max_memkb)
- int libxl_set_memory_target (libxl_ctx *ctx, uint32_t domid, int32_t target_memkb, int relative, int enforce)

- int libxl_get_memory_target (libxl_ctx *ctx, uint32_t domid, uint32_t *out_target)
- int libxl_domain_need_memory (libxl_ctx *ctx, libxl_domain_build_info *b_info, libxl_device_model_info *dm-_info, int domc, uint32_t *need_memkb)
- int libxl_get_free_memory (libxl_ctx *ctx, uint32_t *memkb)
- int libxl_wait_for_free_memory (libxl_ctx *ctx, uint32_t domid, uint32_t memory_kb, int wait_secs)
- int libxl_wait_for_memory_target (libxl_ctx *ctx, uint32_t domid, int wait_secs)
- int libxl_button_press (libxl_ctx *ctx, uint32_t domid, libxl_button button)
- int libxl_get_physinfo (libxl_ctx *ctx, libxl_physinfo *physinfo)
- int libxl_get_topologyinfo (libxl_ctx *ctx, libxl_topologyinfo *info)
- const libxl_version_info * libxl_get_version_info (libxl_ctx *ctx)
- libxl_vcpuinfo * libxl_list_vcpu (libxl_ctx *ctx, uint32_t domid, int *nb_vcpu, int *nrcpus)
- int libxl_set_vcpuaffinity (libxl_ctx *ctx, uint32_t domid, uint32_t vcpuid, libxl_cpumap *cpumap)
- int libxl_set_vcpuonline (libxl_ctx *ctx, uint32_t domid, libxl_cpumap *cpumap)
- int libxl_get_sched_id (libxl_ctx *ctx)
- int libxl_sched_credit_domain_get (libxl_ctx *ctx, uint32_t domid, libxl_sched_credit *scinfo)
- int libxl_sched_credit_domain_set (libxl_ctx *ctx, uint32_t domid, libxl_sched_credit *scinfo)
- int libxl_send_trigger (libxl_ctx *ctx, uint32_t domid, char *trigger_name, uint32_t vcpuid)
- int libxl_send_sysrq (libxl_ctx *ctx, uint32_t domid, char sysrq)
- int libxl_send_debug_keys (libxl_ctx *ctx, char *keys)
- libxl_xen_console_reader * libxl_xen_console_read_start (libxl_ctx *ctx, int clear)
- int libxl_xen_console_read_line (libxl_ctx *ctx, libxl_xen_console_reader *cr, char **line_r)
- void libxl_xen_console_read_finish (libxl_ctx *ctx, libxl_xen_console_reader *cr)
- uint32_t libxl_vm_get_start_time (libxl_ctx *ctx, uint32_t domid)
- char * libxl_tmem_list (libxl_ctx *ctx, uint32_t domid, int use_long)
- int libxl_tmem_freeze (libxl_ctx *ctx, uint32_t domid)
- int libxl_tmem_destroy (libxl_ctx *ctx, uint32_t domid)
- int libxl_tmem_thaw (libxl_ctx *ctx, uint32_t domid)
- int libxl_tmem_set (libxl_ctx *ctx, uint32_t domid, char *name, uint32_t set)
- int libxl_tmem_shared_auth (libxl_ctx *ctx, uint32_t domid, char *uuid, int auth)
- int libxl_tmem_freeable (libxl_ctx *ctx)
- void libxl_file_reference_destroy (libxl_file_reference *f)
- int libxl_get_freecpus (libxl_ctx *ctx, libxl_cpumap *cpumap)
- int libxl_create_cpupool (libxl_ctx *ctx, const char *name, int schedid, libxl_cpumap cpumap, libxl_uuid *uuid, uint32_t *poolid)
- int libxl_destroy_cpupool (libxl_ctx *ctx, uint32_t poolid)
- int libxl_cpupool_rename (libxl_ctx *ctx, const char *name, uint32_t poolid)
- int libxl_cpupool_cpuadd (libxl_ctx *ctx, uint32_t poolid, int cpu)
- int libxl_cpupool_cpuadd_node (libxl_ctx *ctx, uint32_t poolid, int node, int *cpus)
- int libxl_cpupool_cpuremove (libxl_ctx *ctx, uint32_t poolid, int cpu)
- int libxl_cpupool_cpuremove_node (libxl_ctx *ctx, uint32_t poolid, int node, int *cpus)
- int libxl_cpupool_movedomain (libxl_ctx *ctx, uint32_t poolid, uint32_t domid)

4.132.1 Macro Definition Documentation

4.132.1.1 #define BACKEND_STRING_SIZE 5

4.132.1.2 #define PAGE_TO_MEMKB(*pages*) ((pages) * 4)

4.132.1.3 #define STRINGIFY(*x*) #x

4.132.1.4 #define TOSTRING(*x*) STRINGIFY(x)

4.132.2 Function Documentation

4.132.2.1 int libxl_button_press (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* **libxl_button** *button*)

4.132.2.2 int libxl_cdrom_insert (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_device_disk ∗ *disk*)

4.132.2.3 int libxl_console_exec (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* int *cons_num,* **libxl_console_constype** *type*)

4.132.2.4 int libxl_cpupool_cpuadd (**libxl_ctx** ∗ *ctx,* uint32_t *poolid,* int *cpu*)

4.132.2.5 int libxl_cpupool_cpuadd_node (**libxl_ctx** ∗ *ctx,* uint32_t *poolid,* int *node,* int ∗ *cpus*)

4.132.2.6 int libxl_cpupool_cpuremove (**libxl_ctx** ∗ *ctx,* uint32_t *poolid,* int *cpu*)

4.132.2.7 int libxl_cpupool_cpuremove_node (**libxl_ctx** ∗ *ctx,* uint32_t *poolid,* int *node,* int ∗ *cpus*)

4.132.2.8 int libxl_cpupool_movedomain (**libxl_ctx** ∗ *ctx,* uint32_t *poolid,* uint32_t *domid*)

4.132.2.9 int libxl_cpupool_rename (**libxl_ctx** ∗ *ctx,* const char ∗ *name,* uint32_t *poolid*)

4.132.2.10 int libxl_create_cpupool (**libxl_ctx** ∗ *ctx,* const char ∗ *name,* int *schedid,* **libxl_cpumap** *cpumap,* libxl_uuid ∗ *uuid,* uint32_t ∗ *poolid*)

4.132.2.11 int libxl_ctx_free (**libxl_ctx** ∗ *ctx*)

4.132.2.12 int libxl_ctx_init (**libxl_ctx** ∗ *ctx,* int *version,* xentoollog_logger ∗ *lg*)

4.132.2.13 int libxl_destroy_cpupool (**libxl_ctx** ∗ *ctx,* uint32_t *poolid*)

4.132.2.14 int libxl_device_console_add (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_device_console ∗ *console*)

4.132.2.15 int libxl_device_disk_add (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_device_disk ∗ *disk*)

4.132.2.16 int libxl_device_disk_del (**libxl_ctx** ∗ *ctx,* libxl_device_disk ∗ *disk,* int *wait*)

4.132.2.17 int libxl_device_disk_getinfo (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_device_disk ∗ *disk,* libxl_diskinfo ∗ *diskinfo*)

4.132.2.18 libxl_device_disk∗ libxl_device_disk_list (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* int ∗ *num*)

4.132.2.19 char∗ libxl_device_disk_local_attach (**libxl_ctx** ∗ *ctx,* libxl_device_disk ∗ *disk*)

Make a disk available in this domain. Returns path to a device.

4.132.2.20 int libxl_device_disk_local_detach (**libxl_ctx** ∗ *ctx,* libxl_device_disk ∗ *disk*)

4.132.2.21 int libxl_device_net2_add (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_device_net2 ∗ *net2*)

4.132.2.22 int libxl_device_net2_del (**libxl_ctx** ∗ *ctx,* libxl_device_net2 ∗ *net2,* int *wait*)

4.132.2.23 void libxl_device_net2_init (libxl_device_net2 ∗ *net2_info,* int *devnum*)

4.132.2.24 libxl_net2info∗ libxl_device_net2_list (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* unsigned int ∗ *nb*)

4.132.2.25 int libxl_device_nic_add (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_device_nic ∗ *nic*)

4.132.2.26 int libxl_device_nic_del (**libxl_ctx** ∗ *ctx,* libxl_device_nic ∗ *nic,* int *wait*)

4.132.2.27 int libxl_device_nic_init (libxl_device_nic * *nic_info,* int *devnum*)

4.132.2.28 int libxl_device_vfb_add (**libxl_ctx** * *ctx,* uint32_t *domid,* libxl_device_vfb * *vfb*)

4.132.2.29 int libxl_device_vfb_clean_shutdown (**libxl_ctx** * *ctx,* uint32_t *domid*)

4.132.2.30 int libxl_device_vfb_hard_shutdown (**libxl_ctx** * *ctx,* uint32_t *domid*)

4.132.2.31 void libxl_device_vfb_init (libxl_device_vfb * *vfb,* int *dev_num*)

4.132.2.32 int libxl_device_vkb_add (**libxl_ctx** * *ctx,* uint32_t *domid,* libxl_device_vkb * *vkb*)

4.132.2.33 int libxl_device_vkb_clean_shutdown (**libxl_ctx** * *ctx,* uint32_t *domid*)

4.132.2.34 int libxl_device_vkb_hard_shutdown (**libxl_ctx** * *ctx,* uint32_t *domid*)

4.132.2.35 void libxl_device_vkb_init (libxl_device_vkb * *vkb,* int *dev_num*)

4.132.2.36 int libxl_domain_core_dump (**libxl_ctx** * *ctx,* uint32_t *domid,* const char * *filename*)

4.132.2.37 int libxl_domain_destroy (**libxl_ctx** * *ctx,* uint32_t *domid,* int *force*)

4.132.2.38 int libxl_domain_info (**libxl_ctx** * *ctx,* libxl_dominfo * *info_r,* uint32_t *domid*)

4.132.2.39 int libxl_domain_need_memory (**libxl_ctx** * *ctx,* libxl_domain_build_info * *b_info,* libxl_device_model_info * *dm_info,* int *domc,* uint32_t * *need_memkb*)

how much free memory in the system a domain needs to be built

4.132.2.40 int libxl_domain_pause (**libxl_ctx** * *ctx,* uint32_t *domid*)

4.132.2.41 int libxl_domain_preserve (**libxl_ctx** * *ctx,* uint32_t *domid,* libxl_domain_create_info * *info,* const char * *name_suffix,* libxl_uuid *new_uuid*)

Preserves a domain but rewrites xenstore etc to make it unique so that the domain can be restarted.

Does not modify info so that it may be reused.

4.132.2.42 int libxl_domain_rename (**libxl_ctx** * *ctx,* uint32_t *domid,* const char * *old_name,* const char * *new_name,* xs_transaction_t *trans*)

4.132.2.43 int libxl_domain_resume (**libxl_ctx** * *ctx,* uint32_t *domid*)

4.132.2.44 int libxl_domain_setmaxmem (**libxl_ctx** * *ctx,* uint32_t *domid,* uint32_t *max_memkb*)

4.132.2.45 int libxl_domain_shutdown (**libxl_ctx** * *ctx,* uint32_t *domid,* int *req*)

4.132.2.46 int libxl_domain_suspend (**libxl_ctx** * *ctx,* **libxl_domain_suspend_info** * *info,* uint32_t *domid,* int *fd*)

4.132.2.47 int libxl_domain_unpause (**libxl_ctx** * *ctx,* uint32_t *domid*)

4.132.2.48 int libxl_event_get_disk_eject_info (**libxl_ctx** * *ctx,* uint32_t *domid,* **libxl_event** * *event,* libxl_device_disk * *disk*)

Returns true and fills *disk if the caller should eject the disk

4.132.2.49 int libxl_event_get_domain_death_info (libxl_ctx * *ctx*, uint32_t *domid*, libxl_event * *event*, libxl_dominfo * *info*)

Returns:

- 0 if the domain is dead but there is no cleanup to be done. e.g because someone else has already done it.
- 1 if the domain is dead and there is cleanup to be done.

Can return error if the domain exists and is still running.

*info will contain valid domain state iff 1 is returned. In particular if 1 is returned then info->shutdown_reason is guaranteed to be valid since by definition the domain is (shutdown||dying))

4.132.2.50 void libxl_file_reference_destroy (libxl_file_reference * *f*)

4.132.2.51 int libxl_free_event (libxl_event * *event*)

4.132.2.52 int libxl_free_waiter (libxl_waiter * *waiter*)

4.132.2.53 int libxl_get_event (libxl_ctx * *ctx*, libxl_event * *event*)

4.132.2.54 int libxl_get_free_memory (libxl_ctx * *ctx*, uint32_t * *memkb*)

how much free memory is available in the system

4.132.2.55 int libxl_get_freecpus (libxl_ctx * *ctx*, libxl_cpumap * *cpumap*)

4.132.2.56 int libxl_get_memory_target (libxl_ctx * *ctx*, uint32_t *domid*, uint32_t * *out_target*)

4.132.2.57 int libxl_get_physinfo (libxl_ctx * *ctx*, libxl_physinfo * *physinfo*)

4.132.2.58 int libxl_get_sched_id (libxl_ctx * *ctx*)

returns one of the XEN_SCHEDULER_* constants from public/domctl.h

4.132.2.59 int libxl_get_topologyinfo (libxl_ctx * *ctx*, libxl_topologyinfo * *info*)

4.132.2.60 const libxl_version_info* libxl_get_version_info (libxl_ctx * *ctx*)

4.132.2.61 int libxl_get_wait_fd (libxl_ctx * *ctx*, int * *fd*)

4.132.2.62 void libxl_key_value_list_destroy (libxl_key_value_list * *pkvl*)

4.132.2.63 libxl_cpupoolinfo* libxl_list_cpupool (libxl_ctx * *ctx*, int * *nb_pool*)

4.132.2.64 libxl_dominfo* libxl_list_domain (libxl_ctx * *ctx*, int * *nb_domain*)

4.132.2.65 libxl_nicinfo* libxl_list_nics (libxl_ctx * *ctx*, uint32_t *domid*, unsigned int * *nb*)

4.132.2.66 libxl_vcpuinfo* libxl_list_vcpu (libxl_ctx * *ctx*, uint32_t *domid*, int * *nb_vcpu*, int * *nrcpus*)

4.132.2.67 libxl_vminfo* libxl_list_vm (libxl_ctx * *ctx*, int * *nb_vm*)

this API call only list VM running on this host. a VM can be an aggregate of multiple domains.

4.132.2.68 int libxl_primary_console_exec (libxl_ctx * *ctx,* uint32_t *domid_vm*)

libxl_primary_console_exec finds the domid and console number corresponding to the primary console of the given vm, then calls libxl_console_exec with the right arguments (domid might be different if the guest is using stubdoms). This function can be called after creating the device model, in case of HVM guests, and before libxl_run_bootloader in case of PV guests using pygrub.

4.132.2.69 int libxl_sched_credit_domain_get (libxl_ctx * *ctx,* uint32_t *domid,* libxl_sched_credit * *scinfo*)

4.132.2.70 int libxl_sched_credit_domain_set (libxl_ctx * *ctx,* uint32_t *domid,* libxl_sched_credit * *scinfo*)

4.132.2.71 int libxl_send_debug_keys (libxl_ctx * *ctx,* char * *keys*)

4.132.2.72 int libxl_send_sysrq (libxl_ctx * *ctx,* uint32_t *domid,* char *sysrq*)

4.132.2.73 int libxl_send_trigger (libxl_ctx * *ctx,* uint32_t *domid,* char * *trigger_name,* uint32_t *vcpuid*)

4.132.2.74 int libxl_set_memory_target (libxl_ctx * *ctx,* uint32_t *domid,* int32_t *target_memkb,* int *relative,* int *enforce*)

4.132.2.75 int libxl_set_vcpuaffinity (libxl_ctx * *ctx,* uint32_t *domid,* uint32_t *vcpuid,* libxl_cpumap * *cpumap*)

4.132.2.76 int libxl_set_vcpuonline (libxl_ctx * *ctx,* uint32_t *domid,* libxl_cpumap * *cpumap*)

4.132.2.77 int libxl_stop_waiting (libxl_ctx * *ctx,* libxl_waiter * *waiter*)

4.132.2.78 void libxl_string_list_destroy (libxl_string_list * *psl*)

4.132.2.79 int libxl_tmem_destroy (libxl_ctx * *ctx,* uint32_t *domid*)

4.132.2.80 int libxl_tmem_freeable (libxl_ctx * *ctx*)

4.132.2.81 int libxl_tmem_freeze (libxl_ctx * *ctx,* uint32_t *domid*)

4.132.2.82 char* libxl_tmem_list (libxl_ctx * *ctx,* uint32_t *domid,* int *use_long*)

4.132.2.83 int libxl_tmem_set (libxl_ctx * *ctx,* uint32_t *domid,* char * *name,* uint32_t *set*)

4.132.2.84 int libxl_tmem_shared_auth (libxl_ctx * *ctx,* uint32_t *domid,* char * *uuid,* int *auth*)

4.132.2.85 int libxl_tmem_thaw (libxl_ctx * *ctx,* uint32_t *domid*)

4.132.2.86 uint32_t libxl_vm_get_start_time (libxl_ctx * *ctx,* uint32_t *domid*)

4.132.2.87 int libxl_vncviewer_exec (libxl_ctx * *ctx,* uint32_t *domid,* int *autopass*)

4.132.2.88 int libxl_wait_for_disk_ejects (libxl_ctx * *ctx,* uint32_t *guest_domid,* libxl_device_disk * *disks,* int *num_disks,* libxl_waiter * *waiter*)

4.132.2.89 int libxl_wait_for_domain_death (libxl_ctx * *ctx,* uint32_t *domid,* libxl_waiter * *waiter*)

4.132.2.90 int libxl_wait_for_free_memory (libxl_ctx * *ctx,* uint32_t *domid,* uint32_t *memory_kb,* int *wait_secs*)

wait for a given amount of memory to be free in the system

4.132.2.91 int libxl_wait_for_memory_target (libxl_ctx * *ctx*, uint32_t *domid*, int *wait_secs*)

wait for the memory target of a domain to be reached

4.132.2.92 void libxl_xen_console_read_finish (libxl_ctx * *ctx*, libxl_xen_console_reader * *cr*)

4.132.2.93 int libxl_xen_console_read_line (libxl_ctx * *ctx*, libxl_xen_console_reader * *cr*, char ** *line_r*)

return values: *line_r 1 success, whole line obtained from buffer non-0 0 no more lines available right now 0 negative error code ERROR_* 0 On success *line_r is updated to point to a nul-terminated string which is valid until the next call on the same console reader. The libxl caller may overwrite parts of the string if it wishes.

4.132.2.94 libxl_xen_console_reader* libxl_xen_console_read_start (libxl_ctx * *ctx*, int *clear*)

4.133 xen/tools/libxl/libxl.h File Reference

```
#include <stdint.h>
#include <stdarg.h>
#include <netinet/in.h>
#include <xenctrl.h>
#include <xs.h>
#include <sys/wait.h>
#include "libxl_uuid.h"
#include "_libxl_types.h"
```

Data Structures

- struct libxl_cpumap
- struct libxl_cpuarray
- struct libxl_file_reference
- struct libxl_ctx
- struct libxl_domain_suspend_info
- struct libxl_domain_config
- struct libxl_event
- struct libxl_waiter

Macros

- #define LIBXL_CPUARRAY_INVALID_ENTRY ~0
- #define LIBXL_PCI_FUNC_ALL (~0U)
- #define XL_SUSPEND_DEBUG 1
- #define XL_SUSPEND_LIVE 2
- #define LIBXL_VERSION 0

Typedefs

- typedef uint8_t libxl_mac [6]
- typedef char ** libxl_string_list
- typedef char ** libxl_key_value_list
- typedef uint32_t libxl_hwcap [8]
- typedef struct libxl__cpuid_policy libxl_cpuid_policy

- typedef libxl_cpuid_policy * libxl_cpuid_policy_list
- typedef int(* libxl_console_ready)(libxl_ctx *ctx, uint32_t domid, void *priv)
- typedef struct libxl__xen_console_reader libxl_xen_console_reader

Enumerations

- enum libxl_qemu_machine_type { XENFV = 1, XENPV }
- enum libxl_console_constype { LIBXL_CONSTYPE_SERIAL = 1, LIBXL_CONSTYPE_PV }
- enum libxl_console_consback { LIBXL_CONSBACK_XENCONSOLED, LIBXL_CONSBACK_IOEMU }
- enum libxl_disk_format { DISK_FORMAT_UNKNOWN = 0, DISK_FORMAT_QCOW, DISK_FORMAT_QCOW2, DISK_FORMAT_VHD, DISK_FORMAT_RAW, DISK_FORMAT_EMPTY }
- enum libxl_disk_backend { DISK_BACKEND_UNKNOWN = 0, DISK_BACKEND_PHY, DISK_BACKEND_TAP, DISK_BACKEND_QDISK }
- enum libxl_nic_type { NICTYPE_IOEMU = 1, NICTYPE_VIF }
- enum { ERROR_NONSPECIFIC = -1, ERROR_VERSION = -2, ERROR_FAIL = -3, ERROR_NI = -4, ERROR_NOMEM = -5, ERROR_INVAL = -6, ERROR_BADFAIL = -7, ERROR_GUEST_TIMEDOUT = -8 }
- enum libxl_action_on_shutdown { LIBXL_ACTION_DESTROY, LIBXL_ACTION_RESTART, LIBXL_ACTION_RESTART_RENAME, LIBXL_ACTION_PRESERVE, LIBXL_ACTION_COREDUMP_DESTROY, LIBXL_ACTION_COREDUMP_RESTART }
- enum libxl_event_type { LIBXL_EVENT_DOMAIN_DEATH, LIBXL_EVENT_DISK_EJECT }
- enum libxl_button { POWER_BUTTON, SLEEP_BUTTON }

Functions

- void libxl_string_list_destroy (libxl_string_list *sl)
- void libxl_key_value_list_destroy (libxl_key_value_list *kvl)
- void libxl_cpumap_destroy (libxl_cpumap *map)
- void libxl_cpuarray_destroy (libxl_cpuarray *array)
- void libxl_file_reference_destroy (libxl_file_reference *p)
- void libxl_cpuid_destroy (libxl_cpuid_policy_list *cpuid_list)
- const libxl_version_info * libxl_get_version_info (libxl_ctx *ctx)
- int libxl_ctx_init (libxl_ctx *ctx, int version, xentoollog_logger *)
- int libxl_ctx_free (libxl_ctx *ctx)
- int libxl_ctx_set_log (libxl_ctx *ctx, xentoollog_logger *)
- int libxl_ctx_postfork (libxl_ctx *ctx)
- void libxl_init_create_info (libxl_domain_create_info *c_info)
- void libxl_init_build_info (libxl_domain_build_info *b_info, libxl_domain_create_info *c_info)
- void libxl_init_dm_info (libxl_device_model_info *dm_info, libxl_domain_create_info *c_info, libxl_domain_build_info *b_info)
- int libxl_domain_create_new (libxl_ctx *ctx, libxl_domain_config *d_config, libxl_console_ready cb, void *priv, uint32_t *domid)
- int libxl_domain_create_restore (libxl_ctx *ctx, libxl_domain_config *d_config, libxl_console_ready cb, void *priv, uint32_t *domid, int restore_fd)
- void libxl_domain_config_destroy (libxl_domain_config *d_config)
- int libxl_domain_suspend (libxl_ctx *ctx, libxl_domain_suspend_info *info, uint32_t domid, int fd)
- int libxl_domain_resume (libxl_ctx *ctx, uint32_t domid)
- int libxl_domain_shutdown (libxl_ctx *ctx, uint32_t domid, int req)
- int libxl_domain_destroy (libxl_ctx *ctx, uint32_t domid, int force)
- int libxl_domain_preserve (libxl_ctx *ctx, uint32_t domid, libxl_domain_create_info *info, const char *name_suffix, libxl_uuid new_uuid)

- int libxl_get_max_cpus (libxl_ctx *ctx)
- int libxl_run_bootloader (libxl_ctx *ctx, libxl_domain_build_info *info, libxl_device_disk *disk, uint32_t domid)
- int libxl_get_wait_fd (libxl_ctx *ctx, int *fd)
- int libxl_wait_for_domain_death (libxl_ctx *ctx, uint32_t domid, libxl_waiter *waiter)
- int libxl_wait_for_disk_ejects (libxl_ctx *ctx, uint32_t domid, libxl_device_disk *disks, int num_disks, libxl_-waiter *waiter)
- int libxl_get_event (libxl_ctx *ctx, libxl_event *event)
- int libxl_stop_waiting (libxl_ctx *ctx, libxl_waiter *waiter)
- int libxl_free_event (libxl_event *event)
- int libxl_free_waiter (libxl_waiter *waiter)
- int libxl_event_get_domain_death_info (libxl_ctx *ctx, uint32_t domid, libxl_event *event, libxl_dominfo *info)
- int libxl_event_get_disk_eject_info (libxl_ctx *ctx, uint32_t domid, libxl_event *event, libxl_device_disk *disk)
- int libxl_domain_rename (libxl_ctx *ctx, uint32_t domid, const char *old_name, const char *new_name, xs_-transaction_t trans)
- int libxl_domain_pause (libxl_ctx *ctx, uint32_t domid)
- int libxl_domain_unpause (libxl_ctx *ctx, uint32_t domid)
- int libxl_domain_core_dump (libxl_ctx *ctx, uint32_t domid, const char *filename)
- int libxl_domain_setmaxmem (libxl_ctx *ctx, uint32_t domid, uint32_t target_memkb)
- int libxl_set_memory_target (libxl_ctx *ctx, uint32_t domid, int32_t target_memkb, int relative, int enforce)
- int libxl_get_memory_target (libxl_ctx *ctx, uint32_t domid, uint32_t *out_target)
- int libxl_domain_need_memory (libxl_ctx *ctx, libxl_domain_build_info *b_info, libxl_device_model_info *dm-_info, int domc, uint32_t *need_memkb)
- int libxl_get_free_memory (libxl_ctx *ctx, uint32_t *memkb)
- int libxl_wait_for_free_memory (libxl_ctx *ctx, uint32_t domid, uint32_t memory_kb, int wait_secs)
- int libxl_wait_for_memory_target (libxl_ctx *ctx, uint32_t domid, int wait_secs)
- int libxl_vncviewer_exec (libxl_ctx *ctx, uint32_t domid, int autopass)
- int libxl_console_exec (libxl_ctx *ctx, uint32_t domid, int cons_num, libxl_console_constype type)
- int libxl_primary_console_exec (libxl_ctx *ctx, uint32_t domid_vm)
- int libxl_domain_info (libxl_ctx *, libxl_dominfo *info_r, uint32_t domid)
- libxl_dominfo * libxl_list_domain (libxl_ctx *, int *nb_domain)
- libxl_cpupoolinfo * libxl_list_cpupool (libxl_ctx *, int *nb_pool)
- libxl_vminfo * libxl_list_vm (libxl_ctx *ctx, int *nb_vm)
- int libxl_device_disk_add (libxl_ctx *ctx, uint32_t domid, libxl_device_disk *disk)
- int libxl_device_disk_del (libxl_ctx *ctx, libxl_device_disk *disk, int wait)
- libxl_device_disk * libxl_device_disk_list (libxl_ctx *ctx, uint32_t domid, int *num)
- int libxl_device_disk_getinfo (libxl_ctx *ctx, uint32_t domid, libxl_device_disk *disk, libxl_diskinfo *diskinfo)
- int libxl_cdrom_insert (libxl_ctx *ctx, uint32_t domid, libxl_device_disk *disk)
- char * libxl_device_disk_local_attach (libxl_ctx *ctx, libxl_device_disk *disk)
- int libxl_device_disk_local_detach (libxl_ctx *ctx, libxl_device_disk *disk)
- int libxl_device_nic_init (libxl_device_nic *nic, int dev_num)
- int libxl_device_nic_add (libxl_ctx *ctx, uint32_t domid, libxl_device_nic *nic)
- int libxl_device_nic_del (libxl_ctx *ctx, libxl_device_nic *nic, int wait)
- libxl_nicinfo * libxl_list_nics (libxl_ctx *ctx, uint32_t domid, unsigned int *nb)
- int libxl_device_console_add (libxl_ctx *ctx, uint32_t domid, libxl_device_console *console)
- void libxl_device_vkb_init (libxl_device_vkb *vkb, int dev_num)
- int libxl_device_vkb_add (libxl_ctx *ctx, uint32_t domid, libxl_device_vkb *vkb)
- int libxl_device_vkb_clean_shutdown (libxl_ctx *ctx, uint32_t domid)
- int libxl_device_vkb_hard_shutdown (libxl_ctx *ctx, uint32_t domid)
- void libxl_device_vfb_init (libxl_device_vfb *vfb, int dev_num)
- int libxl_device_vfb_add (libxl_ctx *ctx, uint32_t domid, libxl_device_vfb *vfb)
- int libxl_device_vfb_clean_shutdown (libxl_ctx *ctx, uint32_t domid)
- int libxl_device_vfb_hard_shutdown (libxl_ctx *ctx, uint32_t domid)
- int libxl_device_pci_add (libxl_ctx *ctx, uint32_t domid, libxl_device_pci *pcidev)
- int libxl_device_pci_remove (libxl_ctx *ctx, uint32_t domid, libxl_device_pci *pcidev, int force)
- int libxl_device_pci_shutdown (libxl_ctx *ctx, uint32_t domid)

- int libxl_device_pci_list_assigned (libxl_ctx *ctx, libxl_device_pci **list, uint32_t domid, int *num)
- int libxl_device_pci_list_assignable (libxl_ctx *ctx, libxl_device_pci **list, int *num)
- int libxl_device_pci_parse_bdf (libxl_ctx *ctx, libxl_device_pci *pcidev, const char *str)
- int libxl_cpuid_parse_config (libxl_cpuid_policy_list *cpuid, const char *str)
- int libxl_cpuid_parse_config_xend (libxl_cpuid_policy_list *cpuid, const char *str)
- void libxl_cpuid_apply_policy (libxl_ctx *ctx, uint32_t domid)
- void libxl_cpuid_set (libxl_ctx *ctx, uint32_t domid, libxl_cpuid_policy_list cpuid)
- int libxl_userdata_store (libxl_ctx *ctx, uint32_t domid, const char *userdata_userid, const uint8_t *data, int datalen)
- int libxl_userdata_retrieve (libxl_ctx *ctx, uint32_t domid, const char *userdata_userid, uint8_t **data_r, int *datalen_r)
- int libxl_button_press (libxl_ctx *ctx, uint32_t domid, libxl_button button)
- int libxl_get_physinfo (libxl_ctx *ctx, libxl_physinfo *physinfo)
- int libxl_get_topologyinfo (libxl_ctx *ctx, libxl_topologyinfo *info)
- libxl_vcpuinfo * libxl_list_vcpu (libxl_ctx *ctx, uint32_t domid, int *nb_vcpu, int *nrcpus)
- int libxl_set_vcpuaffinity (libxl_ctx *ctx, uint32_t domid, uint32_t vcpuid, libxl_cpumap *cpumap)
- int libxl_set_vcpuonline (libxl_ctx *ctx, uint32_t domid, libxl_cpumap *cpumap)
- int libxl_get_sched_id (libxl_ctx *ctx)
- int libxl_sched_credit_domain_get (libxl_ctx *ctx, uint32_t domid, libxl_sched_credit *scinfo)
- int libxl_sched_credit_domain_set (libxl_ctx *ctx, uint32_t domid, libxl_sched_credit *scinfo)
- int libxl_send_trigger (libxl_ctx *ctx, uint32_t domid, char *trigger_name, uint32_t vcpuid)
- int libxl_send_sysrq (libxl_ctx *ctx, uint32_t domid, char sysrq)
- int libxl_send_debug_keys (libxl_ctx *ctx, char *keys)
- libxl_xen_console_reader * libxl_xen_console_read_start (libxl_ctx *ctx, int clear)
- int libxl_xen_console_read_line (libxl_ctx *ctx, libxl_xen_console_reader *cr, char **line_r)
- void libxl_xen_console_read_finish (libxl_ctx *ctx, libxl_xen_console_reader *cr)
- uint32_t libxl_vm_get_start_time (libxl_ctx *ctx, uint32_t domid)
- char * libxl_tmem_list (libxl_ctx *ctx, uint32_t domid, int use_long)
- int libxl_tmem_freeze (libxl_ctx *ctx, uint32_t domid)
- int libxl_tmem_destroy (libxl_ctx *ctx, uint32_t domid)
- int libxl_tmem_thaw (libxl_ctx *ctx, uint32_t domid)
- int libxl_tmem_set (libxl_ctx *ctx, uint32_t domid, char *name, uint32_t set)
- int libxl_tmem_shared_auth (libxl_ctx *ctx, uint32_t domid, char *uuid, int auth)
- int libxl_tmem_freeable (libxl_ctx *ctx)
- void libxl_device_net2_init (libxl_device_net2 *net2, int dev_num)
- int libxl_device_net2_add (libxl_ctx *ctx, uint32_t domid, libxl_device_net2 *net2)
- libxl_net2info * libxl_device_net2_list (libxl_ctx *ctx, uint32_t domid, unsigned int *nb)
- int libxl_device_net2_del (libxl_ctx *ctx, libxl_device_net2 *net2, int wait)
- int libxl_get_freecpus (libxl_ctx *ctx, libxl_cpumap *cpumap)
- int libxl_create_cpupool (libxl_ctx *ctx, const char *name, int schedid, libxl_cpumap cpumap, libxl_uuid *uuid, uint32_t *poolid)
- int libxl_destroy_cpupool (libxl_ctx *ctx, uint32_t poolid)
- int libxl_cpupool_rename (libxl_ctx *ctx, const char *name, uint32_t poolid)
- int libxl_cpupool_cpuadd (libxl_ctx *ctx, uint32_t poolid, int cpu)
- int libxl_cpupool_cpuadd_node (libxl_ctx *ctx, uint32_t poolid, int node, int *cpus)
- int libxl_cpupool_cpuremove (libxl_ctx *ctx, uint32_t poolid, int cpu)
- int libxl_cpupool_cpuremove_node (libxl_ctx *ctx, uint32_t poolid, int node, int *cpus)
- int libxl_cpupool_movedomain (libxl_ctx *ctx, uint32_t poolid, uint32_t domid)
- int init_console_info (libxl_device_console *console, int dev_num, libxl_domain_build_state *state)
- const char * libxl_sbindir_path (void)
- const char * libxl_bindir_path (void)
- const char * libxl_libexec_path (void)
- const char * libxl_libdir_path (void)
- const char * libxl_sharedir_path (void)
- const char * libxl_private_bindir_path (void)

- const char * libxl_xenfirmwaredir_path (void)
- const char * libxl_xen_config_dir_path (void)
- const char * libxl_xen_script_dir_path (void)
- const char * libxl_lock_dir_path (void)

4.133.1 Macro Definition Documentation

4.133.1.1 #define LIBXL_CPUARRAY_INVALID_ENTRY ~0

4.133.1.2 #define LIBXL_PCI_FUNC_ALL (~0U)

4.133.1.3 #define LIBXL_VERSION 0

4.133.1.4 #define XL_SUSPEND_DEBUG 1

4.133.1.5 #define XL_SUSPEND_LIVE 2

4.133.2 Typedef Documentation

4.133.2.1 typedef int(* libxl_console_ready)(libxl_ctx *ctx, uint32_t domid, void *priv)

4.133.2.2 typedef struct libxl__cpuid_policy libxl_cpuid_policy

4.133.2.3 typedef libxl_cpuid_policy* libxl_cpuid_policy_list

4.133.2.4 typedef uint32_t libxl_hwcap[8]

4.133.2.5 typedef char** libxl_key_value_list

4.133.2.6 typedef uint8_t libxl_mac[6]

4.133.2.7 typedef char** libxl_string_list

4.133.2.8 typedef struct libxl__xen_console_reader libxl_xen_console_reader

4.133.3 Enumeration Type Documentation

4.133.3.1 anonymous enum

Enumerator

ERROR_NONSPECIFIC

ERROR_VERSION

ERROR_FAIL

ERROR_NI

ERROR_NOMEM

ERROR_INVAL

ERROR_BADFAIL

ERROR_GUEST_TIMEDOUT

4.133.3.2 enum libxl_action_on_shutdown

Enumerator

LIBXL_ACTION_DESTROY
LIBXL_ACTION_RESTART
LIBXL_ACTION_RESTART_RENAME
LIBXL_ACTION_PRESERVE
LIBXL_ACTION_COREDUMP_DESTROY
LIBXL_ACTION_COREDUMP_RESTART

4.133.3.3 enum libxl_button

Enumerator

POWER_BUTTON
SLEEP_BUTTON

4.133.3.4 enum libxl_console_consback

Enumerator

LIBXL_CONSBACK_XENCONSOLED
LIBXL_CONSBACK_IOEMU

4.133.3.5 enum libxl_console_constype

Enumerator

LIBXL_CONSTYPE_SERIAL
LIBXL_CONSTYPE_PV

4.133.3.6 enum libxl_disk_backend

Enumerator

DISK_BACKEND_UNKNOWN
DISK_BACKEND_PHY
DISK_BACKEND_TAP
DISK_BACKEND_QDISK

4.133.3.7 enum libxl_disk_format

Enumerator

DISK_FORMAT_UNKNOWN
DISK_FORMAT_QCOW
DISK_FORMAT_QCOW2
DISK_FORMAT_VHD
DISK_FORMAT_RAW
DISK_FORMAT_EMPTY

4.133.3.8 enum libxl_event_type

Enumerator

LIBXL_EVENT_DOMAIN_DEATH

LIBXL_EVENT_DISK_EJECT

4.133.3.9 enum libxl_nic_type

Enumerator

NICTYPE_IOEMU

NICTYPE_VIF

4.133.3.10 enum libxl_qemu_machine_type

Enumerator

XENFV

XENPV

4.133.4 Function Documentation

4.133.4.1 int init_console_info (libxl_device_console ∗ *console,* int *dev_num,* libxl_domain_build_state ∗ *state*)

4.133.4.2 const char∗ libxl_bindir_path (void)

4.133.4.3 int libxl_button_press (libxl_ctx ∗ *ctx,* uint32_t *domid,* libxl_button *button*)

4.133.4.4 int libxl_cdrom_insert (libxl_ctx ∗ *ctx,* uint32_t *domid,* libxl_device_disk ∗ *disk*)

4.133.4.5 int libxl_console_exec (libxl_ctx ∗ *ctx,* uint32_t *domid,* int *cons_num,* libxl_console_constype *type*)

4.133.4.6 void libxl_cpuarray_destroy (libxl_cpuarray ∗ *array*)

4.133.4.7 void libxl_cpuid_apply_policy (libxl_ctx ∗ *ctx,* uint32_t *domid*)

4.133.4.8 void libxl_cpuid_destroy (libxl_cpuid_policy_list ∗ *cpuid_list*)

4.133.4.9 int libxl_cpuid_parse_config (libxl_cpuid_policy_list ∗ *cpuid,* const char ∗ *str*)

4.133.4.10 int libxl_cpuid_parse_config_xend (libxl_cpuid_policy_list ∗ *cpuid,* const char ∗ *str*)

4.133.4.11 void libxl_cpuid_set (libxl_ctx ∗ *ctx,* uint32_t *domid,* libxl_cpuid_policy_list *cpuid*)

4.133.4.12 void libxl_cpumap_destroy (libxl_cpumap ∗ *map*)

4.133.4.13 int libxl_cpupool_cpuadd (libxl_ctx ∗ *ctx,* uint32_t *poolid,* int *cpu*)

4.133.4.14 int libxl_cpupool_cpuadd_node (libxl_ctx ∗ *ctx,* uint32_t *poolid,* int *node,* int ∗ *cpus*)

4.133.4.15 int libxl_cpupool_cpuremove (libxl_ctx ∗ *ctx,* uint32_t *poolid,* int *cpu*)

4.133.4.16 int libxl_cpupool_cpuremove_node (libxl_ctx ∗ *ctx,* uint32_t *poolid,* int *node,* int ∗ *cpus*)

4.133.4.17 int libxl_cpupool_movedomain (**libxl_ctx** ∗ *ctx,* uint32_t *poolid,* uint32_t *domid*)

4.133.4.18 int libxl_cpupool_rename (**libxl_ctx** ∗ *ctx,* const char ∗ *name,* uint32_t *poolid*)

4.133.4.19 int libxl_create_cpupool (**libxl_ctx** ∗ *ctx,* const char ∗ *name,* int *schedid,* **libxl_cpumap** *cpumap,* libxl_uuid ∗ *uuid,* uint32_t ∗ *poolid*)

4.133.4.20 int libxl_ctx_free (**libxl_ctx** ∗ *ctx*)

4.133.4.21 int libxl_ctx_init (**libxl_ctx** ∗ *ctx,* int *version,* xentoollog_logger ∗)

4.133.4.22 int libxl_ctx_postfork (**libxl_ctx** ∗ *ctx*)

4.133.4.23 int libxl_ctx_set_log (**libxl_ctx** ∗ *ctx,* xentoollog_logger ∗)

4.133.4.24 int libxl_destroy_cpupool (**libxl_ctx** ∗ *ctx,* uint32_t *poolid*)

4.133.4.25 int libxl_device_console_add (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_device_console ∗ *console*)

4.133.4.26 int libxl_device_disk_add (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_device_disk ∗ *disk*)

4.133.4.27 int libxl_device_disk_del (**libxl_ctx** ∗ *ctx,* libxl_device_disk ∗ *disk,* int *wait*)

4.133.4.28 int libxl_device_disk_getinfo (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_device_disk ∗ *disk,* libxl_diskinfo ∗ *diskinfo*)

4.133.4.29 libxl_device_disk∗ libxl_device_disk_list (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* int ∗ *num*)

4.133.4.30 char∗ libxl_device_disk_local_attach (**libxl_ctx** ∗ *ctx,* libxl_device_disk ∗ *disk*)

Make a disk available in this domain. Returns path to a device.

4.133.4.31 int libxl_device_disk_local_detach (**libxl_ctx** ∗ *ctx,* libxl_device_disk ∗ *disk*)

4.133.4.32 int libxl_device_net2_add (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_device_net2 ∗ *net2*)

4.133.4.33 int libxl_device_net2_del (**libxl_ctx** ∗ *ctx,* libxl_device_net2 ∗ *net2,* int *wait*)

4.133.4.34 void libxl_device_net2_init (libxl_device_net2 ∗ *net2,* int *dev_num*)

4.133.4.35 libxl_net2info∗ libxl_device_net2_list (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* unsigned int ∗ *nb*)

4.133.4.36 int libxl_device_nic_add (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_device_nic ∗ *nic*)

4.133.4.37 int libxl_device_nic_del (**libxl_ctx** ∗ *ctx,* libxl_device_nic ∗ *nic,* int *wait*)

4.133.4.38 int libxl_device_nic_init (libxl_device_nic ∗ *nic,* int *dev_num*)

4.133.4.39 int libxl_device_pci_add (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_device_pci ∗ *pcidev*)

4.133.4.40 int libxl_device_pci_list_assignable (**libxl_ctx** ∗ *ctx,* libxl_device_pci ∗∗ *list,* int ∗ *num*)

4.133.4.41 int libxl_device_pci_list_assigned (**libxl_ctx** ∗ *ctx,* libxl_device_pci ∗∗ *list,* uint32_t *domid,* int ∗ *num*)

4.133.4.42 int libxl_device_pci_parse_bdf (**libxl_ctx** ∗ *ctx,* libxl_device_pci ∗ *pcidev,* const char ∗ *str*)

4.133.4.43 int libxl_device_pci_remove (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_device_pci ∗ *pcidev,* int *force*)

4.133.4.44 int libxl_device_pci_shutdown (**libxl_ctx** ∗ *ctx,* uint32_t *domid*)

4.133.4.45 int libxl_device_vfb_add (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_device_vfb ∗ *vfb*)

4.133.4.46 int libxl_device_vfb_clean_shutdown (**libxl_ctx** ∗ *ctx,* uint32_t *domid*)

4.133.4.47 int libxl_device_vfb_hard_shutdown (**libxl_ctx** ∗ *ctx,* uint32_t *domid*)

4.133.4.48 void libxl_device_vfb_init (libxl_device_vfb ∗ *vfb,* int *dev_num*)

4.133.4.49 int libxl_device_vkb_add (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_device_vkb ∗ *vkb*)

4.133.4.50 int libxl_device_vkb_clean_shutdown (**libxl_ctx** ∗ *ctx,* uint32_t *domid*)

4.133.4.51 int libxl_device_vkb_hard_shutdown (**libxl_ctx** ∗ *ctx,* uint32_t *domid*)

4.133.4.52 void libxl_device_vkb_init (libxl_device_vkb ∗ *vkb,* int *dev_num*)

4.133.4.53 void libxl_domain_config_destroy (**libxl_domain_config** ∗ *d_config*)

4.133.4.54 int libxl_domain_core_dump (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* const char ∗ *filename*)

4.133.4.55 int libxl_domain_create_new (**libxl_ctx** ∗ *ctx,* **libxl_domain_config** ∗ *d_config,* **libxl_console_ready** *cb,* void ∗ *priv,* uint32_t ∗ *domid*)

4.133.4.56 int libxl_domain_create_restore (**libxl_ctx** ∗ *ctx,* **libxl_domain_config** ∗ *d_config,* **libxl_console_ready** *cb,* void ∗ *priv,* uint32_t ∗ *domid,* int *restore_fd*)

4.133.4.57 int libxl_domain_destroy (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* int *force*)

4.133.4.58 int libxl_domain_info (**libxl_ctx** ∗ , libxl_dominfo ∗ *info_r,* uint32_t *domid*)

4.133.4.59 int libxl_domain_need_memory (**libxl_ctx** ∗ *ctx,* libxl_domain_build_info ∗ *b_info,* libxl_device_model_info ∗ *dm_info,* int *domc,* uint32_t ∗ *need_memkb*)

how much free memory in the system a domain needs to be built

4.133.4.60 int libxl_domain_pause (**libxl_ctx** ∗ *ctx,* uint32_t *domid*)

4.133.4.61 int libxl_domain_preserve (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_domain_create_info ∗ *info,* const char ∗ *name_suffix,* libxl_uuid *new_uuid*)

Preserves a domain but rewrites xenstore etc to make it unique so that the domain can be restarted.

Does not modify info so that it may be reused.

4.133.4.62 int libxl_domain_rename (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* const char ∗ *old_name,* const char ∗ *new_name,* xs_transaction_t *trans*)

4.133.4.63 int libxl_domain_resume (**libxl_ctx** ∗ *ctx,* uint32_t *domid*)

4.133.4.64 int libxl_domain_setmaxmem (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* uint32_t *target_memkb*)

4.133.4.65 int libxl_domain_shutdown (libxl_ctx * *ctx*, uint32_t *domid*, int *req*)

4.133.4.66 int libxl_domain_suspend (libxl_ctx * *ctx*, libxl_domain_suspend_info * *info*, uint32_t *domid*, int *fd*)

4.133.4.67 int libxl_domain_unpause (libxl_ctx * *ctx*, uint32_t *domid*)

4.133.4.68 int libxl_event_get_disk_eject_info (libxl_ctx * *ctx*, uint32_t *domid*, libxl_event * *event*, libxl_device_disk * *disk*)

Returns true and fills *disk if the caller should eject the disk

4.133.4.69 int libxl_event_get_domain_death_info (libxl_ctx * *ctx*, uint32_t *domid*, libxl_event * *event*, libxl_dominfo * *info*)

Returns:

- 0 if the domain is dead but there is no cleanup to be done. e.g because someone else has already done it.
- 1 if the domain is dead and there is cleanup to be done.

Can return error if the domain exists and is still running.

*info will contain valid domain state iff 1 is returned. In particular if 1 is returned then info->shutdown_reason is guaranteed to be valid since by definition the domain is (shutdown||dying))

4.133.4.70 void libxl_file_reference_destroy (libxl_file_reference * *p*)

4.133.4.71 int libxl_free_event (libxl_event * *event*)

4.133.4.72 int libxl_free_waiter (libxl_waiter * *waiter*)

4.133.4.73 int libxl_get_event (libxl_ctx * *ctx*, libxl_event * *event*)

4.133.4.74 int libxl_get_free_memory (libxl_ctx * *ctx*, uint32_t * *memkb*)

how much free memory is available in the system

4.133.4.75 int libxl_get_freecpus (libxl_ctx * *ctx*, libxl_cpumap * *cpumap*)

4.133.4.76 int libxl_get_max_cpus (libxl_ctx * *ctx*)

4.133.4.77 int libxl_get_memory_target (libxl_ctx * *ctx*, uint32_t *domid*, uint32_t * *out_target*)

4.133.4.78 int libxl_get_physinfo (libxl_ctx * *ctx*, libxl_physinfo * *physinfo*)

4.133.4.79 int libxl_get_sched_id (libxl_ctx * *ctx*)

returns one of the XEN_SCHEDULER_* constants from public/domctl.h

4.133.4.80 int libxl_get_topologyinfo (libxl_ctx * *ctx*, libxl_topologyinfo * *info*)

4.133.4.81 const libxl_version_info* libxl_get_version_info (libxl_ctx * *ctx*)

4.133.4.82 int libxl_get_wait_fd (libxl_ctx * *ctx*, int * *fd*)

4.133.4.83 void libxl_init_build_info (libxl_domain_build_info * *b_info,* libxl_domain_create_info * *c_info*)

4.133.4.84 void libxl_init_create_info (libxl_domain_create_info * *c_info*)

4.133.4.85 void libxl_init_dm_info (libxl_device_model_info * *dm_info,* libxl_domain_create_info * *c_info,* libxl_domain_build_info * *b_info*)

4.133.4.86 void libxl_key_value_list_destroy (**libxl_key_value_list** * *kvl*)

4.133.4.87 const char* libxl_libdir_path (void)

4.133.4.88 const char* libxl_libexec_path (void)

4.133.4.89 libxl_cpupoolinfo* libxl_list_cpupool (**libxl_ctx** * , int * *nb_pool*)

4.133.4.90 libxl_dominfo* libxl_list_domain (**libxl_ctx** * , int * *nb_domain*)

4.133.4.91 libxl_nicinfo* libxl_list_nics (**libxl_ctx** * *ctx,* uint32_t *domid,* unsigned int * *nb*)

4.133.4.92 libxl_vcpuinfo* libxl_list_vcpu (**libxl_ctx** * *ctx,* uint32_t *domid,* int * *nb_vcpu,* int * *nrcpus*)

4.133.4.93 libxl_vminfo* libxl_list_vm (**libxl_ctx** * *ctx,* int * *nb_vm*)

this API call only list VM running on this host. a VM can be an aggregate of multiple domains.

4.133.4.94 const char* libxl_lock_dir_path (void)

4.133.4.95 int libxl_primary_console_exec (**libxl_ctx** * *ctx,* uint32_t *domid_vm*)

libxl_primary_console_exec finds the domid and console number corresponding to the primary console of the given vm, then calls libxl_console_exec with the right arguments (domid might be different if the guest is using stubdoms). This function can be called after creating the device model, in case of HVM guests, and before libxl_run_bootloader in case of PV guests using pygrub.

4.133.4.96 const char* libxl_private_bindir_path (void)

4.133.4.97 int libxl_run_bootloader (**libxl_ctx** * *ctx,* libxl_domain_build_info * *info,* libxl_device_disk * *disk,* uint32_t *domid*)

Run the configured bootloader for a PV domain and update info->kernel, info->u.pv.ramdisk and info->u.pv.cmdline as appropriate (any initial values present in these fields must have been allocated with malloc).

Is a NOP on non-PV domains or those with no bootloader configured.

Users should call libxl_file_reference_unmap on the kernel and ramdisk to cleanup or rely on libxl_domain_-{build,restore} to do it.

4.133.4.98 const char* libxl_sbindir_path (void)

4.133.4.99 int libxl_sched_credit_domain_get (**libxl_ctx** * *ctx,* uint32_t *domid,* libxl_sched_credit * *scinfo*)

4.133.4.100 int libxl_sched_credit_domain_set (**libxl_ctx** * *ctx,* uint32_t *domid,* libxl_sched_credit * *scinfo*)

4.133.4.101 int libxl_send_debug_keys (**libxl_ctx** * *ctx,* char * *keys*)

4.133.4.102 int libxl_send_sysrq (libxl_ctx * *ctx,* uint32_t *domid,* char *sysrq*)

4.133.4.103 int libxl_send_trigger (libxl_ctx * *ctx,* uint32_t *domid,* char * *trigger_name,* uint32_t *vcpuid*)

4.133.4.104 int libxl_set_memory_target (libxl_ctx * *ctx,* uint32_t *domid,* int32_t *target_memkb,* int *relative,* int *enforce*)

4.133.4.105 int libxl_set_vcpuaffinity (libxl_ctx * *ctx,* uint32_t *domid,* uint32_t *vcpuid,* libxl_cpumap * *cpumap*)

4.133.4.106 int libxl_set_vcpuonline (libxl_ctx * *ctx,* uint32_t *domid,* libxl_cpumap * *cpumap*)

4.133.4.107 const char* libxl_sharedir_path (void)

4.133.4.108 int libxl_stop_waiting (libxl_ctx * *ctx,* libxl_waiter * *waiter*)

4.133.4.109 void libxl_string_list_destroy (libxl_string_list * *sl*)

4.133.4.110 int libxl_tmem_destroy (libxl_ctx * *ctx,* uint32_t *domid*)

4.133.4.111 int libxl_tmem_freeable (libxl_ctx * *ctx*)

4.133.4.112 int libxl_tmem_freeze (libxl_ctx * *ctx,* uint32_t *domid*)

4.133.4.113 char* libxl_tmem_list (libxl_ctx * *ctx,* uint32_t *domid,* int *use_long*)

4.133.4.114 int libxl_tmem_set (libxl_ctx * *ctx,* uint32_t *domid,* char * *name,* uint32_t *set*)

4.133.4.115 int libxl_tmem_shared_auth (libxl_ctx * *ctx,* uint32_t *domid,* char * *uuid,* int *auth*)

4.133.4.116 int libxl_tmem_thaw (libxl_ctx * *ctx,* uint32_t *domid*)

4.133.4.117 int libxl_userdata_retrieve (libxl_ctx * *ctx,* uint32_t *domid,* const char * *userdata_userid,* uint8_t ** *data_r,* int * *datalen_r*)

4.133.4.118 int libxl_userdata_store (libxl_ctx * *ctx,* uint32_t *domid,* const char * *userdata_userid,* const uint8_t * *data,* int *datalen*)

4.133.4.119 uint32_t libxl_vm_get_start_time (libxl_ctx * *ctx,* uint32_t *domid*)

4.133.4.120 int libxl_vncviewer_exec (libxl_ctx * *ctx,* uint32_t *domid,* int *autopass*)

4.133.4.121 int libxl_wait_for_disk_ejects (libxl_ctx * *ctx,* uint32_t *domid,* libxl_device_disk * *disks,* int *num_disks,* libxl_waiter * *waiter*)

4.133.4.122 int libxl_wait_for_domain_death (libxl_ctx * *ctx,* uint32_t *domid,* libxl_waiter * *waiter*)

4.133.4.123 int libxl_wait_for_free_memory (libxl_ctx * *ctx,* uint32_t *domid,* uint32_t *memory_kb,* int *wait_secs*)

wait for a given amount of memory to be free in the system

4.133.4.124 int libxl_wait_for_memory_target (libxl_ctx * *ctx,* uint32_t *domid,* int *wait_secs*)

wait for the memory target of a domain to be reached

4.133.4.125 const char* libxl_xen_config_dir_path (void)

4.133.4.126 void libxl_xen_console_read_finish (libxl_ctx * *ctx*, libxl_xen_console_reader * *cr*)

4.133.4.127 int libxl_xen_console_read_line (libxl_ctx * *ctx*, libxl_xen_console_reader * *cr*, char ** *line_r*)

return values: *line_r 1 success, whole line obtained from buffer non-0 0 no more lines available right now 0 negative error code ERROR_* 0 On success *line_r is updated to point to a nul-terminated string which is valid until the next call on the same console reader. The libxl caller may overwrite parts of the string if it wishes.

4.133.4.128 libxl_xen_console_reader* libxl_xen_console_read_start (libxl_ctx * *ctx*, int *clear*)

4.133.4.129 const char* libxl_xen_script_dir_path (void)

4.133.4.130 const char* libxl_xenfirmwaredir_path (void)

4.134 xen/tools/libxl/libxl.idl File Reference

4.135 xen/tools/libxl/libxl_blktap2.c File Reference

```
#include "libxl.h"
#include "libxl_osdeps.h"
#include "libxl_internal.h"
#include "tap-ctl.h"
```

Functions

- int libxl__blktap_enabled (libxl__gc *gc)
- const char * libxl__blktap_devpath (libxl__gc *gc, const char *disk, libxl_disk_format format)

4.135.1 Function Documentation

4.135.1.1 const char* libxl__blktap_devpath (libxl__gc * *gc*, const char * *disk*, libxl_disk_format *format*)

4.135.1.2 int libxl__blktap_enabled (libxl__gc * *gc*)

4.136 xen/tools/libxl/libxl_create.c File Reference

```
#include "libxl_osdeps.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <xenctrl.h>
#include <xc_dom.h>
#include <xenguest.h>
#include <assert.h>
#include "libxl.h"
#include "libxl_utils.h"
#include "libxl_internal.h"
#include "flexarray.h"
```

Functions

- void libxl_domain_config_destroy (libxl_domain_config ∗d_config)
- void libxl_init_create_info (libxl_domain_create_info ∗c_info)
- void libxl_init_build_info (libxl_domain_build_info ∗b_info, libxl_domain_create_info ∗c_info)
- void libxl_init_dm_info (libxl_device_model_info ∗dm_info, libxl_domain_create_info ∗c_info, libxl_domain_-build_info ∗b_info)
- int init_console_info (libxl_device_console ∗console, int dev_num, libxl_domain_build_state ∗state)
- int libxl__domain_build (libxl_ctx ∗ctx, libxl_domain_build_info ∗info, uint32_t domid, libxl_domain_build_state ∗state)
- int libxl__domain_make (libxl_ctx ∗ctx, libxl_domain_create_info ∗info, uint32_t ∗domid)
- int libxl_domain_create_new (libxl_ctx ∗ctx, libxl_domain_config ∗d_config, libxl_console_ready cb, void ∗priv, uint32_t ∗domid)
- int libxl_domain_create_restore (libxl_ctx ∗ctx, libxl_domain_config ∗d_config, libxl_console_ready cb, void ∗priv, uint32_t ∗domid, int restore_fd)

4.136.1 Function Documentation

4.136.1.1 int init_console_info (libxl_device_console ∗ *console,* int *dev_num,* libxl_domain_build_state ∗ *state*)

4.136.1.2 int libxl__domain_build (libxl_ctx ∗ *ctx,* libxl_domain_build_info ∗ *info,* uint32_t *domid,* libxl_domain_build_state ∗ *state*)

4.136.1.3 int libxl__domain_make (libxl_ctx ∗ *ctx,* libxl_domain_create_info ∗ *info,* uint32_t ∗ *domid*)

4.136.1.4 void libxl_domain_config_destroy (libxl_domain_config ∗ *d_config*)

4.136.1.5 int libxl_domain_create_new (libxl_ctx ∗ *ctx,* libxl_domain_config ∗ *d_config,* libxl_console_ready *cb,* void ∗ *priv,* uint32_t ∗ *domid*)

4.136.1.6 int libxl_domain_create_restore (libxl_ctx ∗ *ctx,* libxl_domain_config ∗ *d_config,* libxl_console_ready *cb,* void ∗ *priv,* uint32_t ∗ *domid,* int *restore_fd*)

4.136.1.7 void libxl_init_build_info (libxl_domain_build_info ∗ *b_info,* libxl_domain_create_info ∗ *c_info*)

4.136.1.8 void libxl_init_create_info (libxl_domain_create_info ∗ *c_info*)

4.136.1.9 void libxl_init_dm_info (libxl_device_model_info ∗ *dm_info,* libxl_domain_create_info ∗ *c_info,* libxl_domain_build_info ∗ *b_info*)

4.137 xen/tools/libxl/libxl_device.c File Reference

```
#include "libxl_osdeps.h"
#include <string.h>
#include <stdio.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <fcntl.h>
#include "libxl.h"
#include "libxl_internal.h"
```

Functions

- char * libxl__device_frontend_path (libxl__gc *gc, libxl__device *device)
- char * libxl__device_backend_path (libxl__gc *gc, libxl__device *device)
- int libxl__device_generic_add (libxl_ctx *ctx, libxl__device *device, char **bents, char **fents)
- char * libxl__device_disk_string_of_format (libxl_disk_format format)
- char * libxl__device_disk_string_of_backend (libxl_disk_backend backend)
- int libxl__device_physdisk_major_minor (const char *physpath, int *major, int *minor)
- int libxl__device_disk_dev_number (const char *virtpath)
- int libxl__device_destroy (libxl_ctx *ctx, char *be_path, int force)
- int libxl__devices_destroy (libxl_ctx *ctx, uint32_t domid, int force)
- int libxl__device_del (libxl_ctx *ctx, libxl__device *dev, int wait)
- int libxl__wait_for_device_model (libxl_ctx *ctx, uint32_t domid, char *state, int(*check_callback)(libxl_ctx *ctx, uint32_t domid, const char *state, void *userdata), void *check_callback_userdata)
- int libxl__wait_for_backend (libxl_ctx *ctx, char *be_path, char *state)

4.137.1 Function Documentation

4.137.1.1 char* libxl__device_backend_path (libxl__gc * *gc*, libxl__device * *device*)

4.137.1.2 int libxl__device_del (libxl_ctx * *ctx*, libxl__device * *dev*, int *wait*)

4.137.1.3 int libxl__device_destroy (libxl_ctx * *ctx*, char * *be_path*, int *force*)

4.137.1.4 int libxl__device_disk_dev_number (const char * *virtpath*)

4.137.1.5 char* libxl__device_disk_string_of_backend (libxl_disk_backend *backend*)

4.137.1.6 char* libxl__device_disk_string_of_format (libxl_disk_format *format*)

4.137.1.7 char* libxl__device_frontend_path (libxl__gc * *gc*, libxl__device * *device*)

4.137.1.8 int libxl__device_generic_add (libxl_ctx * *ctx*, libxl__device * *device*, char ** *bents*, char ** *fents*)

4.137.1.9 int libxl__device_physdisk_major_minor (const char * *physpath*, int * *major*, int * *minor*)

4.137.1.10 int libxl__devices_destroy (libxl_ctx * *ctx*, uint32_t *domid*, int *force*)

4.137.1.11 int libxl__wait_for_backend (libxl_ctx * *ctx*, char * *be_path*, char * *state*)

4.137.1.12 int libxl__wait_for_device_model (libxl_ctx * *ctx*, uint32_t *domid*, char * *state*, int(*)(libxl_ctx *ctx, uint32_t domid, const char *state, void *userdata) *check_callback*, void * *check_callback_userdata*)

4.138 xen/tools/libxl/libxl_dm.c File Reference

```
#include "libxl_osdeps.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>
#include <fcntl.h>
#include <assert.h>
#include "libxl_utils.h"
#include "libxl_internal.h"
#include "libxl.h"
#include "flexarray.h"
```

Functions

- int libxl__create_device_model (libxl_ctx *ctx, libxl_device_model_info *info, libxl_device_disk *disks, int num_disks, libxl_device_nic *vifs, int num_vifs, libxl__device_model_starting **starting_r)
- int libxl__confirm_device_model_startup (libxl_ctx *ctx, libxl__device_model_starting *starting)
- int libxl__destroy_device_model (libxl_ctx *ctx, uint32_t domid)
- int libxl__need_xenpv_qemu (libxl_ctx *ctx, int nr_consoles, libxl_device_console *consoles, int nr_vfbs, libxl_device_vfb *vfbs, int nr_disks, libxl_device_disk *disks)
- int libxl__create_xenpv_qemu (libxl_ctx *ctx, uint32_t domid, libxl_device_vfb *vfb, libxl__device_model_starting **starting_r)

4.138.1 Function Documentation

4.138.1.1 int libxl__confirm_device_model_startup (**libxl_ctx** * *ctx,* **libxl__device_model_starting** * *starting*)

4.138.1.2 int libxl__create_device_model (**libxl_ctx** * *ctx,* libxl_device_model_info * *info,* libxl_device_disk * *disks,* int *num_disks,* libxl_device_nic * *vifs,* int *num_vifs,* **libxl__device_model_starting** ** *starting_r*)

4.138.1.3 int libxl__create_xenpv_qemu (**libxl_ctx** * *ctx,* uint32_t *domid,* libxl_device_vfb * *vfb,* **libxl__device_model_starting** ** *starting_r*)

4.138.1.4 int libxl__destroy_device_model (**libxl_ctx** * *ctx,* uint32_t *domid*)

4.138.1.5 int libxl__need_xenpv_qemu (**libxl_ctx** * *ctx,* int *nr_consoles,* libxl_device_console * *consoles,* int *nr_vfbs,* libxl_device_vfb * *vfbs,* int *nr_disks,* libxl_device_disk * *disks*)

4.139 xen/tools/libxl/libxl_dom.c File Reference

```
#include "libxl_osdeps.h"
#include <stdio.h>
#include <assert.h>
#include <glob.h>
#include <inttypes.h>
#include <string.h>
#include <sys/mman.h>
#include <sys/time.h>
#include <sys/stat.h>
#include <unistd.h>
#include <xenctrl.h>
#include <xc_dom.h>
#include <xenguest.h>
#include <fcntl.h>
#include <xen/hvm/hvm_info_table.h>
#include "libxl.h"
#include "libxl_internal.h"
```

Data Structures

- struct suspendinfo

Functions

- int libxl__domain_is_hvm (libxl_ctx *ctx, uint32_t domid)
- int libxl__domain_shutdown_reason (libxl_ctx *ctx, uint32_t domid)
- int libxl__build_pre (libxl_ctx *ctx, uint32_t domid, libxl_domain_build_info *info, libxl_domain_build_state *state)
- int libxl__build_post (libxl_ctx *ctx, uint32_t domid, libxl_domain_build_info *info, libxl_domain_build_state *state, char **vms_ents, char **local_ents)
- int libxl__build_pv (libxl_ctx *ctx, uint32_t domid, libxl_domain_build_info *info, libxl_domain_build_state *state)
- int libxl__build_hvm (libxl_ctx *ctx, uint32_t domid, libxl_domain_build_info *info, libxl_domain_build_state *state)
- int libxl__domain_restore_common (libxl_ctx *ctx, uint32_t domid, libxl_domain_build_info *info, libxl_domain_build_state *state, int fd)
- int libxl__domain_suspend_common (libxl_ctx *ctx, uint32_t domid, int fd, int hvm, int live, int debug)
- int libxl__domain_save_device_model (libxl_ctx *ctx, uint32_t domid, int fd)
- char * libxl__uuid2string (libxl__gc *gc, const libxl_uuid uuid)
- void libxl__userdata_destroyall (libxl_ctx *ctx, uint32_t domid)
- int libxl_userdata_store (libxl_ctx *ctx, uint32_t domid, const char *userdata_userid, const uint8_t *data, int datalen)
- int libxl_userdata_retrieve (libxl_ctx *ctx, uint32_t domid, const char *userdata_userid, uint8_t **data_r, int *datalen_r)

4.139.1 Function Documentation

4.139.1.1 int libxl__build_hvm (libxl_ctx * *ctx*, uint32_t *domid*, libxl_domain_build_info * *info*, libxl_domain_build_state * *state*)

4.139.1.2 int libxl__build_post (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_domain_build_info ∗ *info,* libxl_domain_build_state ∗ *state,* char ∗∗ *vms_ents,* char ∗∗ *local_ents*)

4.139.1.3 int libxl__build_pre (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_domain_build_info ∗ *info,* libxl_domain_build_state ∗ *state*)

4.139.1.4 int libxl__build_pv (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_domain_build_info ∗ *info,* libxl_domain_build_state ∗ *state*)

4.139.1.5 int libxl__domain_is_hvm (**libxl_ctx** ∗ *ctx,* uint32_t *domid*)

4.139.1.6 int libxl__domain_restore_common (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_domain_build_info ∗ *info,* libxl_domain_build_state ∗ *state,* int *fd*)

4.139.1.7 int libxl__domain_save_device_model (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* int *fd*)

4.139.1.8 int libxl__domain_shutdown_reason (**libxl_ctx** ∗ *ctx,* uint32_t *domid*)

4.139.1.9 int libxl__domain_suspend_common (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* int *fd,* int *hvm,* int *live,* int *debug*)

4.139.1.10 void libxl__userdata_destroyall (**libxl_ctx** ∗ *ctx,* uint32_t *domid*)

4.139.1.11 char∗ libxl__uuid2string (**libxl__gc** ∗ *gc,* const libxl_uuid *uuid*)

4.139.1.12 int libxl_userdata_retrieve (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* const char ∗ *userdata_userid,* uint8_t ∗∗ *data_r,* int ∗ *datalen_r*)

4.139.1.13 int libxl_userdata_store (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* const char ∗ *userdata_userid,* const uint8_t ∗ *data,* int *datalen*)

4.140 xen/tools/libxl/libxl_internal.c File Reference

```
#include "libxl_osdeps.h"
#include <stdio.h>
#include <stdarg.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/mman.h>
#include <unistd.h>
#include "libxl.h"
#include "libxl_internal.h"
#include "libxl_utils.h"
```

Functions

- int libxl__error_set (libxl_ctx ∗ctx, int code)
- int libxl__ptr_add (libxl__gc ∗gc, void ∗ptr)
- void libxl__free_all (libxl__gc ∗gc)
- void ∗ libxl__zalloc (libxl__gc ∗gc, int bytes)
- void ∗ libxl__calloc (libxl__gc ∗gc, size_t nmemb, size_t size)
- char ∗ libxl__sprintf (libxl__gc ∗gc, const char ∗fmt,...)
- char ∗ libxl__strdup (libxl__gc ∗gc, const char ∗c)

- char * libxl__dirname (libxl__gc *gc, const char *s)
- void libxl__logv (libxl_ctx *ctx, xentoollog_level msglevel, int errnoval, const char *file, int line, const char *func, char *fmt, va_list ap)
- void libxl__log (libxl_ctx *ctx, xentoollog_level msglevel, int errnoval, const char *file, int line, const char *func, char *fmt,...)
- char * libxl__abs_path (libxl__gc *gc, char *s, const char *path)
- int libxl__file_reference_map (libxl_file_reference *f)
- int libxl__file_reference_unmap (libxl_file_reference *f)

4.140.1 Function Documentation

4.140.1.1 char* libxl__abs_path (libxl__gc * *gc*, char * *s*, const char * *path*)

4.140.1.2 void* libxl__calloc (libxl__gc * *gc*, size_t *nmemb*, size_t *size*)

4.140.1.3 char* libxl__dirname (libxl__gc * *gc*, const char * *s*)

4.140.1.4 int libxl__error_set (libxl_ctx * *ctx*, int *code*)

4.140.1.5 int libxl__file_reference_map (libxl_file_reference * *f*)

4.140.1.6 int libxl__file_reference_unmap (libxl_file_reference * *f*)

4.140.1.7 void libxl__free_all (libxl__gc * *gc*)

4.140.1.8 void libxl__log (libxl_ctx * *ctx*, xentoollog_level *msglevel*, int *errnoval*, const char * *file*, int *line*, const char * *func*, char * *fmt*, ...)

4.140.1.9 void libxl__logv (libxl_ctx * *ctx*, xentoollog_level *msglevel*, int *errnoval*, const char * *file*, int *line*, const char * *func*, char * *fmt*, va_list *ap*)

4.140.1.10 int libxl__ptr_add (libxl__gc * *gc*, void * *ptr*)

4.140.1.11 char* libxl__sprintf (libxl__gc * *gc*, const char * *fmt*, ...)

4.140.1.12 char* libxl__strdup (libxl__gc * *gc*, const char * *c*)

4.140.1.13 void* libxl__zalloc (libxl__gc * *gc*, int *bytes*)

4.141 xen/tools/libxl/libxl_internal.h File Reference

```
#include <stdint.h>
#include <stdarg.h>
#include <stdlib.h>
#include <xs.h>
#include <xenctrl.h>
#include "xentoollog.h"
#include "flexarray.h"
#include "libxl_utils.h"
```

Data Structures

- struct libxl__device

- struct libxl__gc
- struct libxl__spawn_starting
- struct libxl__device_model_starting
- struct libxl__cpuid_policy
- struct libxl__xen_console_reader

Macros

- #define _hidden
- #define _protected
- #define LIBXL_DESTROY_TIMEOUT 10
- #define LIBXL_DEVICE_MODEL_START_TIMEOUT 10
- #define LIBXL_XENCONSOLE_LIMIT 1048576
- #define LIBXL_XENCONSOLE_PROTOCOL "vt100"
- #define LIBXL_MAXMEM_CONSTANT 1024
- #define LIBXL_PV_EXTRA_MEMORY 1024
- #define LIBXL_HVM_EXTRA_MEMORY 2048
- #define LIBXL_MIN_DOM0_MEM (128∗1024)
- #define QEMU_SIGNATURE "DeviceModelRecord0002"
- #define STUBDOM_CONSOLE_LOGGING 0
- #define STUBDOM_CONSOLE_SAVE 1
- #define STUBDOM_CONSOLE_RESTORE 2
- #define STUBDOM_CONSOLE_SERIAL 3
- #define STUBDOM_SPECIAL_CONSOLES 3
- #define SAVEFILE "/var/lib/xen/qemu-save"
- #define ARRAY_SIZE(a) (sizeof(a) / sizeof(a[0]))
- #define LIBXL__LOGGING_ENABLED
- #define LIBXL__LOG(ctx, loglevel, _f, _a...) libxl__log(ctx, loglevel, -1, __FILE__, __LINE__, __func__, _f, ##_a)
- #define LIBXL__LOG_ERRNO(ctx, loglevel, _f, _a...) libxl__log(ctx, loglevel, errno, __FILE__, __LINE__, __func__, _f, ##_a)
- #define LIBXL__LOG_ERRNOVAL(ctx, loglevel, errnoval, _f, _a...) libxl__log(ctx, loglevel, errnoval, __FILE__, __LINE__, __func__, _f, ##_a)
- #define is_valid_device_kind(kind) (((kind) >= DEVICE_VIF) && ((kind) <= DEVICE_CONSOLE))
- #define XC_PCI_BDF "0x%x, 0x%x, 0x%x, 0x%x"
- #define AUTO_PHP_SLOT 0x100
- #define SYSFS_PCI_DEV "/sys/bus/pci/devices"
- #define SYSFS_PCIBACK_DRIVER "/sys/bus/pci/drivers/pciback"
- #define XENSTORE_PID_FILE "/var/run/xenstored.pid"
- #define PROC_PCI_NUM_RESOURCES 7
- #define PCI_BAR_IO 0x01
- #define PRINTF_ATTRIBUTE(x, y) __attribute__((format(printf, x, y)))
- #define LIBXL_INIT_GC(ctx) (libxl__gc){ .alloc_maxsize = 0, .alloc_ptrs = 0, .owner = ctx }
- #define LIBXL__LOG_DEBUG XTL_DEBUG
- #define LIBXL__LOG_INFO XTL_INFO
- #define LIBXL__LOG_WARNING XTL_WARN
- #define LIBXL__LOG_ERROR XTL_ERROR

Enumerations

- enum libxl__device_kinds {
  DEVICE_VIF = 1, DEVICE_VIF2, DEVICE_VBD, DEVICE_TAP,
  DEVICE_QDISK, DEVICE_PCI, DEVICE_VFB, DEVICE_VKBD,
  DEVICE_CONSOLE }

Functions

- _hidden void libxl__logv (libxl_ctx *ctx, xentoollog_level msglevel, int errnoval, const char *file, int line, const char *func, char *fmt, va_list al) __attribute__((format(printf
- _hidden void _hidden void libxl__log (libxl_ctx *ctx, xentoollog_level msglevel, int errnoval, const char *file, int line, const char *func, char *fmt,...) __attribute__((format(printf
- _hidden int libxl__ptr_add (libxl__gc *gc, void *ptr)
- _hidden void libxl__free_all (libxl__gc *gc)
- _hidden void * libxl__zalloc (libxl__gc *gc, int bytes)
- _hidden void * libxl__calloc (libxl__gc *gc, size_t nmemb, size_t size)
- _hidden char * libxl__sprintf (libxl__gc *gc, const char *fmt,...) PRINTF_ATTRIBUTE(2
- _hidden char _hidden char * libxl__strdup (libxl__gc *gc, const char *c)
- _hidden char * libxl__dirname (libxl__gc *gc, const char *s)
- _hidden char ** libxl__xs_kvs_of_flexarray (libxl__gc *gc, flexarray_t *array, int length)
- _hidden int libxl__xs_writev (libxl__gc *gc, xs_transaction_t t, char *dir, char **kvs)
- _hidden int libxl__xs_write (libxl__gc *gc, xs_transaction_t t, char *path, char *fmt,...) PRINTF_ATTRIBUTE(4
- _hidden int _hidden char * libxl__xs_get_dompath (libxl__gc *gc, uint32_t domid)
- _hidden char * libxl__xs_read (libxl__gc *gc, xs_transaction_t t, char *path)
- _hidden char ** libxl__xs_directory (libxl__gc *gc, xs_transaction_t t, char *path, unsigned int *nb)
- _hidden int libxl__domain_is_hvm (libxl_ctx *ctx, uint32_t domid)
- _hidden int libxl__domain_shutdown_reason (libxl_ctx *ctx, uint32_t domid)
- _hidden int libxl__build_pre (libxl_ctx *ctx, uint32_t domid, libxl_domain_build_info *info, libxl_domain_build_state *state)
- _hidden int libxl__build_post (libxl_ctx *ctx, uint32_t domid, libxl_domain_build_info *info, libxl_domain_build_state *state, char **vms_ents, char **local_ents)
- _hidden int libxl__build_pv (libxl_ctx *ctx, uint32_t domid, libxl_domain_build_info *info, libxl_domain_build_state *state)
- _hidden int libxl__build_hvm (libxl_ctx *ctx, uint32_t domid, libxl_domain_build_info *info, libxl_domain_build_state *state)
- _hidden int libxl__domain_restore_common (libxl_ctx *ctx, uint32_t domid, libxl_domain_build_info *info, libxl_domain_build_state *state, int fd)
- _hidden int libxl__domain_suspend_common (libxl_ctx *ctx, uint32_t domid, int fd, int hvm, int live, int debug)
- _hidden int libxl__domain_save_device_model (libxl_ctx *ctx, uint32_t domid, int fd)
- _hidden void libxl__userdata_destroyall (libxl_ctx *ctx, uint32_t domid)
- _hidden char * libxl__device_disk_string_of_backend (libxl_disk_backend backend)
- _hidden char * libxl__device_disk_string_of_format (libxl_disk_format format)
- _hidden int libxl__device_physdisk_major_minor (const char *physpath, int *major, int *minor)
- _hidden int libxl__device_disk_dev_number (const char *virtpath)
- _hidden int libxl__device_generic_add (libxl_ctx *ctx, libxl__device *device, char **bents, char **fents)
- _hidden char * libxl__device_backend_path (libxl__gc *gc, libxl__device *device)
- _hidden char * libxl__device_frontend_path (libxl__gc *gc, libxl__device *device)
- _hidden int libxl__device_del (libxl_ctx *ctx, libxl__device *dev, int wait)
- _hidden int libxl__device_destroy (libxl_ctx *ctx, char *be_path, int force)
- _hidden int libxl__devices_destroy (libxl_ctx *ctx, uint32_t domid, int force)
- _hidden int libxl__wait_for_device_model (libxl_ctx *ctx, uint32_t domid, char *state, int(*check_callback)(libxl_ctx *ctx, uint32_t domid, const char *state, void *userdata), void *check_callback_userdata)
- _hidden int libxl__wait_for_backend (libxl_ctx *ctx, char *be_path, char *state)
- _hidden int libxl__device_pci_add (libxl_ctx *ctx, uint32_t domid, libxl_device_pci *pcidev, int starting)
- int libxl__domain_make (libxl_ctx *ctx, libxl_domain_create_info *info, uint32_t *domid)
- int libxl__domain_build (libxl_ctx *ctx, libxl_domain_build_info *info, uint32_t domid, libxl_domain_build_state *state)
- _hidden int libxl__create_device_model (libxl_ctx *ctx, libxl_device_model_info *info, libxl_device_disk *disk, int num_disks, libxl_device_nic *vifs, int num_vifs, libxl__device_model_starting **starting_r)
- int libxl__create_xenpv_qemu (libxl_ctx *ctx, uint32_t domid, libxl_device_vfb *vfb, libxl__device_model_starting **starting_r)

- _hidden int libxl__need_xenpv_qemu (libxl_ctx *ctx, int nr_consoles, libxl_device_console *consoles, int nr_vfbs, libxl_device_vfb *vfbs, int nr_disks, libxl_device_disk *disks)
- int libxl__confirm_device_model_startup (libxl_ctx *ctx, libxl__device_model_starting *starting)
- _hidden int libxl__spawn_spawn (libxl_ctx *ctx, libxl__device_model_starting *starting, const char *what, void(*intermediate_hook)(void *for_spawn, pid_t innerchild))
- _hidden int libxl__destroy_device_model (libxl_ctx *ctx, uint32_t domid)
- _hidden int libxl__spawn_detach (libxl_ctx *ctx, libxl__spawn_starting *for_spawn)
- _hidden int libxl__spawn_check (libxl_ctx *ctx, void *for_spawn)
- _hidden void libxl__exec (int stdinfd, int stdoutfd, int stderrfd, const char *arg0, char **args)
- _hidden void libxl__log_child_exitstatus (libxl__gc *gc, const char *what, pid_t pid, int status)
- _hidden char * libxl__abs_path (libxl__gc *gc, char *s, const char *path)
- _hidden char * libxl__domid_to_name (libxl__gc *gc, uint32_t domid)
- _hidden char * libxl__cpupoolid_to_name (libxl__gc *gc, uint32_t poolid)
- _hidden int libxl__blktap_enabled (libxl__gc *gc)
- _hidden const char * libxl__blktap_devpath (libxl__gc *gc, const char *disk, libxl_disk_format format)
- _hidden char * libxl__uuid2string (libxl__gc *gc, const libxl_uuid uuid)
- _hidden int libxl__error_set (libxl_ctx *ctx, int code)
- _hidden int libxl__file_reference_map (libxl_file_reference *f)
- _hidden int libxl__file_reference_unmap (libxl_file_reference *f)

4.141.1 Macro Definition Documentation

4.141.1.1 #define _hidden

4.141.1.2 #define _protected

4.141.1.3 #define ARRAY_SIZE(a) (sizeof(a) / sizeof(a[0]))

4.141.1.4 #define AUTO_PHP_SLOT 0x100

4.141.1.5 #define is_valid_device_kind(*kind*) (((kind) >= DEVICE_VIF) && ((kind) <= DEVICE_CONSOLE))

4.141.1.6 #define LIBXL__LOG(ctx, *loglevel, _f, _a...*) libxl__log(ctx, loglevel, -1, __FILE__, __LINE__, __func__, _f, ##_a)

4.141.1.7 #define LIBXL__LOG_DEBUG XTL_DEBUG

4.141.1.8 #define LIBXL__LOG_ERRNO(ctx, *loglevel, _f, _a...*) libxl__log(ctx, loglevel, errno, __FILE__, __LINE__, __func__, _f, ##_a)

4.141.1.9 #define LIBXL__LOG_ERRNOVAL(ctx, *loglevel, errnoval, _f, _a...*) libxl__log(ctx, loglevel, errnoval, __FILE__, __LINE__, __func__, _f, ##_a)

4.141.1.10 #define LIBXL__LOG_ERROR XTL_ERROR

4.141.1.11 #define LIBXL__LOG_INFO XTL_INFO

4.141.1.12 #define LIBXL__LOG_WARNING XTL_WARN

4.141.1.13 #define LIBXL__LOGGING_ENABLED

4.141.1.14 #define LIBXL_DESTROY_TIMEOUT 10

4.141.1.15 #define LIBXL_DEVICE_MODEL_START_TIMEOUT 10

4.141.1.16 #define LIBXL_HVM_EXTRA_MEMORY 2048

4.141.1.17 #define LIBXL_INIT_GC(ctx) (libxl__gc){ .alloc_maxsize = 0, .alloc_ptrs = 0, .owner = ctx }

4.141.1.18 #define LIBXL_MAXMEM_CONSTANT 1024

4.141.1.19 #define LIBXL_MIN_DOM0_MEM (128∗1024)

4.141.1.20 #define LIBXL_PV_EXTRA_MEMORY 1024

4.141.1.21 #define LIBXL_XENCONSOLE_LIMIT 1048576

4.141.1.22 #define LIBXL_XENCONSOLE_PROTOCOL "vt100"

4.141.1.23 #define PCI_BAR_IO 0x01

4.141.1.24 #define PRINTF_ATTRIBUTE(*x, y*) __attribute__((format(printf, x, y)))

4.141.1.25 #define PROC_PCI_NUM_RESOURCES 7

4.141.1.26 #define QEMU_SIGNATURE "DeviceModelRecord0002"

4.141.1.27 #define SAVEFILE "/var/lib/xen/qemu-save"

4.141.1.28 #define STUBDOM_CONSOLE_LOGGING 0

4.141.1.29 #define STUBDOM_CONSOLE_RESTORE 2

4.141.1.30 #define STUBDOM_CONSOLE_SAVE 1

4.141.1.31 #define STUBDOM_CONSOLE_SERIAL 3

4.141.1.32 #define STUBDOM_SPECIAL_CONSOLES 3

4.141.1.33 #define SYSFS_PCI_DEV "/sys/bus/pci/devices"

4.141.1.34 #define SYSFS_PCIBACK_DRIVER "/sys/bus/pci/drivers/pciback"

4.141.1.35 #define XC_PCI_BDF "0x%x, 0x%x, 0x%x, 0x%x"

4.141.1.36 #define XENSTORE_PID_FILE "/var/run/xenstored.pid"

4.141.2 Enumeration Type Documentation

4.141.2.1 enum libxl__device_kinds

Enumerator

DEVICE_VIF

DEVICE_VIF2

DEVICE_VBD

DEVICE_TAP

DEVICE_QDISK

DEVICE_PCI

DEVICE_VFB

DEVICE_VKBD

DEVICE_CONSOLE

4.141.3 Function Documentation

4.141.3.1 **_hidden** char* libxl__abs_path (**libxl__gc** * *gc,* char * *s,* const char * *path*)

4.141.3.2 **_hidden** const char* libxl__blktap_devpath (**libxl__gc** * *gc,* const char * *disk,* **libxl_disk_format** *format*)

4.141.3.3 **_hidden** int libxl__blktap_enabled (**libxl__gc** * *gc*)

4.141.3.4 **_hidden** int libxl__build_hvm (**libxl_ctx** * *ctx,* uint32_t *domid,* libxl_domain_build_info * *info,* libxl_domain_build_state * *state*)

4.141.3.5 **_hidden** int libxl__build_post (**libxl_ctx** * *ctx,* uint32_t *domid,* libxl_domain_build_info * *info,* libxl_domain_build_state * *state,* char ** *vms_ents,* char ** *local_ents*)

4.141.3.6 **_hidden** int libxl__build_pre (**libxl_ctx** * *ctx,* uint32_t *domid,* libxl_domain_build_info * *info,* libxl_domain_build_state * *state*)

4.141.3.7 **_hidden** int libxl__build_pv (**libxl_ctx** * *ctx,* uint32_t *domid,* libxl_domain_build_info * *info,* libxl_domain_build_state * *state*)

4.141.3.8 **_hidden** void* libxl__calloc (**libxl__gc** * *gc,* size_t *nmemb,* size_t *size*)

4.141.3.9 int libxl__confirm_device_model_startup (**libxl_ctx** * *ctx,* **libxl__device_model_starting** * *starting*)

4.141.3.10 **_hidden** char* libxl__cpupoolid_to_name (**libxl__gc** * *gc,* uint32_t *poolid*)

4.141.3.11 **_hidden** int libxl__create_device_model (**libxl_ctx** * *ctx,* libxl_device_model_info * *info,* libxl_device_disk * *disk,* int *num_disks,* libxl_device_nic * *vifs,* int *num_vifs,* **libxl__device_model_starting** ** *starting_r*)

4.141.3.12 int libxl__create_xenpv_qemu (**libxl_ctx** * *ctx,* uint32_t *domid,* libxl_device_vfb * *vfb,* **libxl__device_model_starting** ** *starting_r*)

4.141.3.13 **_hidden** int libxl__destroy_device_model (**libxl_ctx** * *ctx,* uint32_t *domid*)

4.141.3.14 **_hidden** char* libxl__device_backend_path (**libxl__gc** * *gc,* **libxl__device** * *device*)

4.141.3.15 **_hidden** int libxl__device_del (**libxl_ctx** * *ctx,* **libxl__device** * *dev,* int *wait*)

4.141.3.16 **_hidden** int libxl__device_destroy (**libxl_ctx** * *ctx,* char * *be_path,* int *force*)

4.141.3.17 **_hidden** int libxl__device_disk_dev_number (const char * *virtpath*)

4.141.3.18 **_hidden** char* libxl__device_disk_string_of_backend (**libxl_disk_backend** *backend*)

4.141.3.19 **_hidden** char* libxl__device_disk_string_of_format (**libxl_disk_format** *format*)

4.141.3.20 **_hidden** char* libxl__device_frontend_path (**libxl__gc** * *gc,* **libxl__device** * *device*)

4.141.3.21 **_hidden** int libxl__device_generic_add (**libxl_ctx** * *ctx,* **libxl__device** * *device,* char ** *bents,* char ** *fents*)

4.141.3.22 **_hidden** int libxl__device_pci_add (**libxl_ctx** * *ctx,* uint32_t *domid,* libxl_device_pci * *pcidev,* int *starting*)

4.141.3.23 **_hidden** int libxl__device_physdisk_major_minor (const char ∗ *physpath,* int ∗ *major,* int ∗ *minor*)

4.141.3.24 **_hidden** int libxl__devices_destroy (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* int *force*)

4.141.3.25 **_hidden** char∗ libxl__dirname (**libxl__gc** ∗ *gc,* const char ∗ *s*)

4.141.3.26 int libxl__domain_build (**libxl_ctx** ∗ *ctx,* libxl_domain_build_info ∗ *info,* uint32_t *domid,* libxl_domain_build_state ∗ *state*)

4.141.3.27 **_hidden** int libxl__domain_is_hvm (**libxl_ctx** ∗ *ctx,* uint32_t *domid*)

4.141.3.28 int libxl__domain_make (**libxl_ctx** ∗ *ctx,* libxl_domain_create_info ∗ *info,* uint32_t ∗ *domid*)

4.141.3.29 **_hidden** int libxl__domain_restore_common (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* libxl_domain_build_info ∗ *info,* libxl_domain_build_state ∗ *state,* int *fd*)

4.141.3.30 **_hidden** int libxl__domain_save_device_model (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* int *fd*)

4.141.3.31 **_hidden** int libxl__domain_shutdown_reason (**libxl_ctx** ∗ *ctx,* uint32_t *domid*)

4.141.3.32 **_hidden** int libxl__domain_suspend_common (**libxl_ctx** ∗ *ctx,* uint32_t *domid,* int *fd,* int *hvm,* int *live,* int *debug*)

4.141.3.33 **_hidden** char∗ libxl__domid_to_name (**libxl__gc** ∗ *gc,* uint32_t *domid*)

4.141.3.34 **_hidden** int libxl__error_set (**libxl_ctx** ∗ *ctx,* int *code*)

4.141.3.35 **_hidden** void libxl__exec (int *stdinfd,* int *stdoutfd,* int *stderrfd,* const char ∗ *arg0,* char ∗∗ *args*)

4.141.3.36 **_hidden** int libxl__file_reference_map (**libxl_file_reference** ∗ *f*)

4.141.3.37 **_hidden** int libxl__file_reference_unmap (**libxl_file_reference** ∗ *f*)

4.141.3.38 **_hidden** void libxl__free_all (**libxl__gc** ∗ *gc*)

4.141.3.39 **_hidden** void **_hidden** void libxl__log (**libxl_ctx** ∗ *ctx,* xentoollog_level *msglevel,* int *errnoval,* const char ∗ *file,* int *line,* const char ∗ *func,* char ∗ *fmt,* ...)

4.141.3.40 **_hidden** void libxl__log_child_exitstatus (**libxl__gc** ∗ *gc,* const char ∗ *what,* pid_t *pid,* int *status*)

4.141.3.41 **_hidden** void libxl__logv (**libxl_ctx** ∗ *ctx,* xentoollog_level *msglevel,* int *errnoval,* const char ∗ *file,* int *line,* const char ∗ *func,* char ∗ *fmt,* va_list *al*)

4.141.3.42 **_hidden** int libxl__need_xenpv_qemu (**libxl_ctx** ∗ *ctx,* int *nr_consoles,* libxl_device_console ∗ *consoles,* int *nr_vfbs,* libxl_device_vfb ∗ *vfbs,* int *nr_disks,* libxl_device_disk ∗ *disks*)

4.141.3.43 **_hidden** int libxl__ptr_add (**libxl__gc** ∗ *gc,* void ∗ *ptr*)

4.141.3.44 **_hidden** int libxl__spawn_check (**libxl_ctx** ∗ *ctx,* void ∗ *for_spawn*)

4.141.3.45 **_hidden** int libxl__spawn_detach (**libxl_ctx** ∗ *ctx,* **libxl__spawn_starting** ∗ *for_spawn*)

4.141.3.46 **_hidden** int libxl__spawn_spawn (**libxl_ctx** ∗ *ctx,* **libxl__device_model_starting** ∗ *starting,* const char ∗ *what,* void(∗)(void ∗for_spawn, pid_t innerchild) *intermediate_hook*)

4.141.3.47 **_hidden** char∗ libxl__sprintf (**libxl__gc** ∗ *gc,* const char ∗ *fmt,* ...)

4.141.3.48 _hidden char _hidden char* libxl__strdup (libxl__gc * *gc*, const char * *c*)

4.141.3.49 _hidden void libxl__userdata_destroyall (libxl_ctx * *ctx*, uint32_t *domid*)

4.141.3.50 _hidden char* libxl__uuid2string (libxl__gc * *gc*, const libxl_uuid *uuid*)

4.141.3.51 _hidden int libxl__wait_for_backend (libxl_ctx * *ctx*, char * *be_path*, char * *state*)

4.141.3.52 _hidden int libxl__wait_for_device_model (libxl_ctx * *ctx*, uint32_t *domid*, char * *state*, int(*)(libxl_ctx *ctx, uint32_t domid, const char *state, void *userdata) *check_callback*, void * *check_callback_userdata*)

4.141.3.53 _hidden char** libxl__xs_directory (libxl__gc * *gc*, xs_transaction_t *t*, char * *path*, unsigned int * *nb*)

4.141.3.54 _hidden int _hidden char* libxl__xs_get_dompath (libxl__gc * *gc*, uint32_t *domid*)

4.141.3.55 _hidden char** libxl__xs_kvs_of_flexarray (libxl__gc * *gc*, flexarray_t * *array*, int *length*)

4.141.3.56 _hidden char* libxl__xs_read (libxl__gc * *gc*, xs_transaction_t *t*, char * *path*)

4.141.3.57 _hidden int libxl__xs_write (libxl__gc * *gc*, xs_transaction_t *t*, char * *path*, char * *fmt*, ...)

4.141.3.58 _hidden int libxl__xs_writev (libxl__gc * *gc*, xs_transaction_t *t*, char * *dir*, char ** *kvs*)

4.141.3.59 _hidden void* libxl__zalloc (libxl__gc * *gc*, int *bytes*)

4.142 xen/tools/libxl/libxl_utils.c File Reference

```
#include "libxl_osdeps.h"
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <string.h>
#include <xs.h>
#include <xenctrl.h>
#include <ctype.h>
#include <errno.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include <assert.h>
#include "libxl_utils.h"
#include "libxl_internal.h"
```

Data Structures

- struct schedid_name

Macros

- #define READ_WRITE_EXACTLY(rw, zero_is_eof, constdata)
- #define QEMU_VERSION_STR "QEMU emulator version "

Functions

- const char * libxl_basename (const char *name)
- unsigned long libxl_get_required_shadow_memory (unsigned long maxmem_kb, unsigned int smp_cpus)
- char * libxl_domid_to_name (libxl_ctx *ctx, uint32_t domid)
- char * libxl__domid_to_name (libxl__gc *gc, uint32_t domid)
- int libxl_name_to_domid (libxl_ctx *ctx, const char *name, uint32_t *domid)
- char * libxl_cpupoolid_to_name (libxl_ctx *ctx, uint32_t poolid)
- char * libxl__cpupoolid_to_name (libxl__gc *gc, uint32_t poolid)
- int libxl_name_to_cpupoolid (libxl_ctx *ctx, const char *name, uint32_t *poolid)
- int libxl_name_to_schedid (libxl_ctx *ctx, const char *name)
- char * libxl_schedid_to_name (libxl_ctx *ctx, int schedid)
- int libxl_get_stubdom_id (libxl_ctx *ctx, int guest_domid)
- int libxl_is_stubdom (libxl_ctx *ctx, uint32_t domid, uint32_t *target_domid)
- int libxl_create_logfile (libxl_ctx *ctx, char *name, char **full_name)
- int libxl_string_to_backend (libxl_ctx *ctx, char *s, libxl_disk_backend *backend)
- int libxl_read_file_contents (libxl_ctx *ctx, const char *filename, void **data_r, int *datalen_r)
- int libxl_ctx_postfork (libxl_ctx *ctx)
- pid_t libxl_fork (libxl_ctx *ctx)
- int libxl_pipe (libxl_ctx *ctx, int pipes[2])
- int libxl_mac_to_device_nic (libxl_ctx *ctx, uint32_t domid, const char *mac, libxl_device_nic *nic)
- int libxl_devid_to_device_nic (libxl_ctx *ctx, uint32_t domid, const char *devid, libxl_device_nic *nic)
- int libxl_devid_to_device_disk (libxl_ctx *ctx, uint32_t domid, const char *devid, libxl_device_disk *disk)
- int libxl_devid_to_device_net2 (libxl_ctx *ctx, uint32_t domid, const char *devid, libxl_device_net2 *net2)
- int libxl_strtomac (const char *mac_s, uint8_t *mac)
- int libxl_check_device_model_version (libxl_ctx *ctx, char *path)
- int libxl_cpumap_alloc (libxl_ctx *ctx, libxl_cpumap *cpumap)
- void libxl_cpumap_destroy (libxl_cpumap *map)
- int libxl_cpumap_test (libxl_cpumap *cpumap, int cpu)
- void libxl_cpumap_set (libxl_cpumap *cpumap, int cpu)
- void libxl_cpumap_reset (libxl_cpumap *cpumap, int cpu)
- int libxl_cpuarray_alloc (libxl_ctx *ctx, libxl_cpuarray *cpuarray)
- void libxl_cpuarray_destroy (libxl_cpuarray *array)
- int libxl_get_max_cpus (libxl_ctx *ctx)

4.142.1 Macro Definition Documentation

4.142.1.1 #define QEMU_VERSION_STR "QEMU emulator version "

4.142.1.2 #define READ_WRITE_EXACTLY(*rw, zero_is_eof, constdata*)

Value:

```
\
  int libxl_##rw##_exactly(libxl_ctx *ctx, int fd,                  \
                           constdata void *data, ssize_t sz,                \
                           const char *filename, const char *what) {        \
      ssize_t got;                                                          \
                                                                            \
      while (sz > 0) {                                                      \
          got = rw(fd, data, sz);                                           \
          if (got == -1) {                                                  \
              if (errno == EINTR) continue;                                 \
              if (!ctx) return errno;                                       \
              LIBXL__LOG_ERRNO(ctx, LIBXL__LOG_ERROR, "failed to " #rw "
       %s%s%s", \
                           what?what:"", what?" from ":"", filename);       \
              return errno;                                                 \
          }                                                                 \
          if (got == 0) {                                                   \
              if (!ctx) return EPROTO;                                      \
```

```
            LIBXL__LOG(ctx, LIBXL__LOG_ERROR,                                   \
                   zero_is_eof                                                  \
                   ? "file/stream truncated reading %s%s%s"                     \
                   : "file/stream write returned 0! writing %s%s%s",            \
                   what?what:"", what?" from ":"", filename);                   \
            return EPROTO;                                                      \
        }                                                                       \
        sz -= got;                                                              \
        data = (char*)data + got;                                               \
    }                                                                           \
    return 0;                                                                   \
}
```

4.142.2 Function Documentation

4.142.2.1 char* libxl__cpupoolid_to_name (libxl__gc * gc, uint32_t poolid)

4.142.2.2 char* libxl__domid_to_name (libxl__gc * gc, uint32_t domid)

4.142.2.3 const char* libxl_basename (const char * name)

4.142.2.4 int libxl_check_device_model_version (libxl_ctx * ctx, char * path)

4.142.2.5 int libxl_cpuarray_alloc (libxl_ctx * ctx, libxl_cpuarray * cpuarray)

4.142.2.6 void libxl_cpuarray_destroy (libxl_cpuarray * array)

4.142.2.7 int libxl_cpumap_alloc (libxl_ctx * ctx, libxl_cpumap * cpumap)

4.142.2.8 void libxl_cpumap_destroy (libxl_cpumap * map)

4.142.2.9 void libxl_cpumap_reset (libxl_cpumap * cpumap, int cpu)

4.142.2.10 void libxl_cpumap_set (libxl_cpumap * cpumap, int cpu)

4.142.2.11 int libxl_cpumap_test (libxl_cpumap * cpumap, int cpu)

4.142.2.12 char* libxl_cpupoolid_to_name (libxl_ctx * ctx, uint32_t poolid)

4.142.2.13 int libxl_create_logfile (libxl_ctx * ctx, char * name, char ** full_name)

4.142.2.14 int libxl_ctx_postfork (libxl_ctx * ctx)

4.142.2.15 int libxl_devid_to_device_disk (libxl_ctx * ctx, uint32_t domid, const char * devid, libxl_device_disk * disk)

4.142.2.16 int libxl_devid_to_device_net2 (libxl_ctx * ctx, uint32_t domid, const char * devid, libxl_device_net2 * net2)

4.142.2.17 int libxl_devid_to_device_nic (libxl_ctx * ctx, uint32_t domid, const char * devid, libxl_device_nic * nic)

4.142.2.18 char* libxl_domid_to_name (libxl_ctx * ctx, uint32_t domid)

4.142.2.19 pid_t libxl_fork (libxl_ctx * ctx)

4.142.2.20 int libxl_get_max_cpus (libxl_ctx * ctx)

4.142.2.21 unsigned long libxl_get_required_shadow_memory (unsigned long maxmem_kb, unsigned int smp_cpus)

4.142.2.22 int libxl_get_stubdom_id (libxl_ctx * ctx, int guest_domid)

4.142.2.23 int libxl_is_stubdom (libxl_ctx * *ctx*, uint32_t *domid*, uint32_t * *target_domid*)

4.142.2.24 int libxl_mac_to_device_nic (libxl_ctx * *ctx*, uint32_t *domid*, const char * *mac*, libxl_device_nic * *nic*)

4.142.2.25 int libxl_name_to_cpupoolid (libxl_ctx * *ctx*, const char * *name*, uint32_t * *poolid*)

4.142.2.26 int libxl_name_to_domid (libxl_ctx * *ctx*, const char * *name*, uint32_t * *domid*)

4.142.2.27 int libxl_name_to_schedid (libxl_ctx * *ctx*, const char * *name*)

4.142.2.28 int libxl_pipe (libxl_ctx * *ctx*, int *pipes[2]*)

4.142.2.29 int libxl_read_file_contents (libxl_ctx * *ctx*, const char * *filename*, void ** *data_r*, int * *datalen_r*)

4.142.2.30 char* libxl_schedid_to_name (libxl_ctx * *ctx*, int *schedid*)

4.142.2.31 int libxl_string_to_backend (libxl_ctx * *ctx*, char * *s*, libxl_disk_backend * *backend*)

4.142.2.32 int libxl_strtomac (const char * *mac_s*, uint8_t * *mac*)

4.143 xen/tools/libxl/libxl_uuid.h File Reference

Macros

- #define LIBXL_UUID_FMT "%02hhx%02hhx%02hhx%02hhx-%02hhx%02hhx-%02hhx%02hhx-%02hhx%02hhx-%02hhx%02hhx%02hhx%02hhx%02hhx%02hhx"
- #define LIBXL__UUID_BYTES(uuid)

4.143.1 Macro Definition Documentation

4.143.1.1 #define LIBXL__UUID_BYTES(*uuid*)

Value:

```
uuid[0], uuid[1], uuid[2], uuid[3], \
                                uuid[4], uuid[5], uuid[6], uuid[7], \
                                uuid[8], uuid[9], uuid[10], uuid[11], \
                                uuid[12], uuid[13], uuid[14], uuid[15]
```

4.143.1.2 #define LIBXL_UUID_FMT "%02hhx%02hhx%02hhx%02hhx-%02hhx%02hhx-%02hhx%02hhx-%02hhx%02hhx-%02hhx%02hhx%02hhx%02hhx%02hhx%02hhx"

4.144 xen/tools/libxl/libxlu_cfg_y.c File Reference

```
#include "libxlu_cfg_i.h"
#include "libxlu_cfg_l.h"
```

Data Structures

- union YYSTYPE
- struct YYLTYPE
- union yyalloc

Macros

- #define YYBISON 1
- #define YYBISON_VERSION "2.5"
- #define YYSKELETON_NAME "yacc.c"
- #define YYPURE 1
- #define YYPUSH 0
- #define YYPULL 1
- #define YYLSP_NEEDED 1
- #define yyparse xlu__cfg_yyparse
- #define yylex xlu__cfg_yylex
- #define yyerror xlu__cfg_yyerror
- #define yylval xlu__cfg_yylval
- #define yychar xlu__cfg_yychar
- #define yydebug xlu__cfg_yydebug
- #define yynerrs xlu__cfg_yynerrs
- #define yylloc xlu__cfg_yylloc
- #define YYLEX_PARAM ctx->scanner
- #define YYDEBUG 0
- #define YYERROR_VERBOSE 1
- #define YYTOKEN_TABLE 0
- #define YYTOKENTYPE
- #define YYSTYPE_IS_TRIVIAL 1
- #define yystype YYSTYPE /* obsolescent; will be withdrawn */
- #define YYSTYPE_IS_DECLARED 1
- #define yyltype YYLTYPE /* obsolescent; will be withdrawn */
- #define YYLTYPE_IS_DECLARED 1
- #define YYLTYPE_IS_TRIVIAL 1
- #define YYSIZE_T unsigned int
- #define YYSIZE_MAXIMUM ((YYSIZE_T) -1)
- #define YY_(msgid) msgid
- #define YYUSE(e) ((void) (e))
- #define YYID(n) (n)
- #define YYSTACK_ALLOC YYMALLOC
- #define YYSTACK_FREE YYFREE
- #define YYSTACK_ALLOC_MAXIMUM YYSIZE_MAXIMUM
- #define YYMALLOC malloc
- #define YYFREE free
- #define YYSTACK_GAP_MAXIMUM (sizeof (union yyalloc) - 1)
- #define YYSTACK_BYTES(N)
- #define YYCOPY_NEEDED 1
- #define YYSTACK_RELOCATE(Stack_alloc, Stack)
- #define YYCOPY(To, From, Count)
- #define YYFINAL 2
- #define YYLAST 23
- #define YYNTOKENS 12
- #define YYNNTS 10
- #define YYNRULES 20
- #define YYNSTATES 29
- #define YYUNDEFTOK 2
- #define YYMAXUTOK 261
- #define YYTRANSLATE(YYX) ((unsigned int) (YYX) <= YYMAXUTOK ? yytranslate[YYX] : YYUNDEFTOK)
- #define YYPACT_NINF -17
- #define YYTABLE_NINF -1
- #define yypact_value_is_default(yystate) ((yystate) == (-17))

- #define yytable_value_is_error(yytable_value) YYID (0)
- #define yyerrok (yyerrstatus = 0)
- #define yyclearin (yychar = YYEMPTY)
- #define YYEMPTY (-2)
- #define YYEOF 0
- #define YYACCEPT goto yyacceptlab
- #define YYABORT goto yyabortlab
- #define YYERROR goto yyerrorlab
- #define YYFAIL goto yyerrlab
- #define YYRECOVERING() (!!yyerrstatus)
- #define YYBACKUP(Token, Value)
- #define YYTERROR 1
- #define YYERRCODE 256
- #define YYRHSLOC(Rhs, K) ((Rhs)[K])
- #define YYLLOC_DEFAULT(Current, Rhs, N)
- #define YY_LOCATION_PRINT(File, Loc)
- #define YYLEX yylex (&yylval, &yylloc, YYLEX_PARAM)
- #define YYDPRINTF(Args)
- #define YY_SYMBOL_PRINT(Title, Type, Value, Location)
- #define YY_STACK_PRINT(Bottom, Top)
- #define YY_REDUCE_PRINT(Rule)
- #define YYINITDEPTH 200
- #define YYMAXDEPTH 10000
- #define YYCASE_(N, S)
- #define YYPOPSTACK(N) (yyvsp -= (N), yyssp -= (N), yylsp -= (N))
- #define YYSYNTAX_ERROR

Typedefs

- typedef union YYSTYPE YYSTYPE
- typedef struct YYLTYPE YYLTYPE
- typedef unsigned char yytype_uint8
- typedef short int yytype_int8
- typedef unsigned short int yytype_uint16
- typedef short int yytype_int16

Enumerations

- enum yytokentype {
  IDENT = 258, STRING = 259, NUMBER = 260, NEWLINE = 261,
  IDENT = 258, STRING = 259, NUMBER = 260, NEWLINE = 261 }

Functions

- int yyparse ()
- int yyparse (CfgParseContext *ctx)

4.144.1 Macro Definition Documentation

4.144.1.1 #define YY_(*msgid*) msgid

4.144.1.2 #define YY_LOCATION_PRINT(*File, Loc*)

Value:

```
fprintf (File, "%d.%d-%d.%d",            \
          (Loc).first_line, (Loc).first_column, \
          (Loc).last_line,  (Loc).last_column)
```

4.144.1.3 #define YY_REDUCE_PRINT(*Rule*)

4.144.1.4 #define YY_STACK_PRINT(*Bottom, Top*)

4.144.1.5 #define YY_SYMBOL_PRINT(*Title, Type, Value, Location*)

4.144.1.6 #define YYABORT goto yyabortlab

4.144.1.7 #define YYACCEPT goto yyacceptlab

4.144.1.8 #define YYBACKUP(*Token, Value*)

Value:

```
do                                  \
  if (yychar == YYEMPTY && yylen == 1)              \
    {                                       \
      yychar = (Token);                             \
      yylval = (Value);                             \
      YYPOPSTACK (1);                               \
      goto yybackup;                                \
    }                                       \
  else                                      \
    {                                       \
      yyerror (&yylloc, ctx, YY_("syntax error: cannot back up")); \
      YYERROR;                                  \
    }                                       \
while (YYID (0))
```

4.144.1.9 #define YYBISON 1

4.144.1.10 #define YYBISON_VERSION "2.5"

4.144.1.11 #define YYCASE_(*N, S*)

Value:

```
case N:                                     \
        yyformat = S;                               \
      break
```

4.144.1.12 #define yychar xlu__cfg_yychar

4.144.1.13 #define yyclearin (yychar = YYEMPTY)

4.144.1.14 #define YYCOPY(*To, From, Count*)

Value:

```
do                    \
    {                     \
      YYSIZE_T yyi;               \
      for (yyi = 0; yyi < (Count); yyi++)    \
        (To)[yyi] = (From)[yyi];         \
    }                     \
      while (YYID (0))
```

4.144.1.15 #define YYCOPY_NEEDED 1

4.144.1.16 #define yydebug xlu__cfg_yydebug

4.144.1.17 #define YYDEBUG 0

4.144.1.18 #define YYDPRINTF(*Args*)

4.144.1.19 #define YYEMPTY (-2)

4.144.1.20 #define YYEOF 0

4.144.1.21 #define YYERRCODE 256

4.144.1.22 #define yyerrok (yyerrstatus = 0)

4.144.1.23 #define yyerror xlu__cfg_yyerror

4.144.1.24 #define YYERROR goto yyerrorlab

4.144.1.25 #define YYERROR_VERBOSE 1

4.144.1.26 #define YYFAIL goto yyerrlab

4.144.1.27 #define YYFINAL 2

4.144.1.28 #define YYFREE free

4.144.1.29 #define YYID(*n*) (n)

4.144.1.30 #define YYINITDEPTH 200

4.144.1.31 #define YYLAST 23

4.144.1.32 #define yylex xlu__cfg_yylex

4.144.1.33 #define YYLEX yylex (&yylval, &yylloc, YYLEX_PARAM)

4.144.1.34 #define YYLEX_PARAM ctx->scanner

4.144.1.35 #define yylloc xlu__cfg_yylloc

4.144.1.36 #define YYLLOC_DEFAULT(*Current, Rhs, N*)

Value:

```
do                                    \
      if (YYID (N))                                                   \
    {                                 \
      (Current).first_line   = YYRHSLOC (Rhs, 1).first_line;    \
      (Current).first_column = YYRHSLOC (Rhs, 1).first_column;  \
```

```
    (Current).last_line    = YYRHSLOC (Rhs, N).last_line;      \
    (Current).last_column  = YYRHSLOC (Rhs, N).last_column;    \
  }                                   \
    else                                  \
  {                                   \
    (Current).first_line   = (Current).last_line   =      \
      YYRHSLOC (Rhs, 0).last_line;                 \
    (Current).first_column = (Current).last_column =      \
      YYRHSLOC (Rhs, 0).last_column;               \
  }                                   \
  while (YYID (0))
```

4.144.1.37 #define YYLSP_NEEDED 1

4.144.1.38 #define yyltype YYLTYPE /∗ obsolescent; will be withdrawn ∗/

4.144.1.39 #define YYLTYPE_IS_DECLARED 1

4.144.1.40 #define YYLTYPE_IS_TRIVIAL 1

4.144.1.41 #define yylval xlu__cfg_yylval

4.144.1.42 #define YYMALLOC malloc

4.144.1.43 #define YYMAXDEPTH 10000

4.144.1.44 #define YYMAXUTOK 261

4.144.1.45 #define yynerrs xlu__cfg_yynerrs

4.144.1.46 #define YYNNTS 10

4.144.1.47 #define YYNRULES 20

4.144.1.48 #define YYNSTATES 29

4.144.1.49 #define YYNTOKENS 12

4.144.1.50 #define YYPACT_NINF -17

4.144.1.51 #define yypact_value_is_default(*yystate*) ((yystate) == (-17))

4.144.1.52 #define yyparse xlu__cfg_yyparse

4.144.1.53 #define YYPOPSTACK(*N*) (yyvsp -= (N), yyssp -= (N), yylsp -= (N))

4.144.1.54 #define YYPULL 1

4.144.1.55 #define YYPURE 1

4.144.1.56 #define YYPUSH 0

4.144.1.57 #define YYRECOVERING( ) (!!yyerrstatus)

4.144.1.58 #define YYRHSLOC(*Rhs*, *K*) ((Rhs)[K])

4.144.1.59 #define YYSIZE_MAXIMUM ((YYSIZE_T) -1)

4.144.1.60 #define YYSIZE_T unsigned int

4.144.1.61 #define YYSKELETON_NAME "yacc.c"

4.144.1.62 #define YYSTACK_ALLOC YYMALLOC

4.144.1.63 #define YYSTACK_ALLOC_MAXIMUM YYSIZE_MAXIMUM

4.144.1.64 #define YYSTACK_BYTES(*N*)

Value:

```
((N) * (sizeof (yytype_int16) + sizeof (YYSTYPE) + sizeof (
      YYLTYPE)) \
      + 2 * YYSTACK_GAP_MAXIMUM)
```

4.144.1.65 #define YYSTACK_FREE YYFREE

4.144.1.66 #define YYSTACK_GAP_MAXIMUM (sizeof (union yyalloc) - 1)

4.144.1.67 #define YYSTACK_RELOCATE(*Stack_alloc, Stack*)

Value:

```
do                                      \
    {                                       \
    YYSIZE_T yynewbytes;                        \
    YYCOPY (&yyptr->Stack_alloc, Stack, yysize);            \
    Stack = &yyptr->Stack_alloc;                    \
    yynewbytes = yystacksize * sizeof (*Stack) + YYSTACK_GAP_MAXIMUM; \
    yyptr += yynewbytes / sizeof (*yyptr);              \
    }                                       \
    while (YYID (0))
```

4.144.1.68 #define yystype YYSTYPE /∗ obsolescent; will be withdrawn ∗/

4.144.1.69 #define YYSTYPE_IS_DECLARED 1

4.144.1.70 #define YYSTYPE_IS_TRIVIAL 1

4.144.1.71 #define YYSYNTAX_ERROR

Value:

```
yysyntax_error (&yymsg_alloc, &yymsg, \
                                        yyssp, yytoken)
```

4.144.1.72 #define YYTABLE_NINF -1

4.144.1.73 #define yytable_value_is_error(*yytable_value*) YYID (0)

4.144.1.74 #define YYTERROR 1

4.144.1.75 #define YYTOKEN_TABLE 0

4.144.1.76 #define YYTOKENTYPE

4.144.1.77 #define YYTRANSLATE(*YYX*) ((unsigned int) (YYX) <= YYMAXUTOK ? yytranslate[YYX] : YYUNDEFTOK)

4.144.1.78 #define YYUNDEFTOK 2

4.144.1.79 #define YYUSE(*e*) ((void) (e))

4.144.2 Typedef Documentation

4.144.2.1 typedef struct YYLTYPE YYLTYPE

4.144.2.2 typedef union YYSTYPE YYSTYPE

4.144.2.3 typedef short int yytype_int16

4.144.2.4 typedef short int yytype_int8

4.144.2.5 typedef unsigned short int yytype_uint16

4.144.2.6 typedef unsigned char yytype_uint8

4.144.3 Enumeration Type Documentation

4.144.3.1 enum yytokentype

Enumerator

IDENT
STRING
NUMBER
NEWLINE
IDENT
STRING
NUMBER
NEWLINE

4.144.4 Function Documentation

4.144.4.1 int yyparse ( )

4.144.4.2 int yyparse (CfgParseContext * *ctx*)

4.145 xen/tools/libxl/libxlu_cfg_y.h File Reference

Data Structures

- union YYSTYPE
- struct YYLTYPE

Macros

- #define YYSTYPE_IS_TRIVIAL 1
- #define yystype YYSTYPE /* obsolescent; will be withdrawn */
- #define YYSTYPE_IS_DECLARED 1

- #define yyltype YYLTYPE /∗ obsolescent; will be withdrawn ∗/
- #define YYLTYPE_IS_DECLARED 1
- #define YYLTYPE_IS_TRIVIAL 1

Typedefs

- typedef union YYSTYPE YYSTYPE
- typedef struct YYLTYPE YYLTYPE

Enumerations

- enum yytokentype {
  IDENT = 258, STRING = 259, NUMBER = 260, NEWLINE = 261,
  IDENT = 258, STRING = 259, NUMBER = 260, NEWLINE = 261 }

4.145.1 Macro Definition Documentation

4.145.1.1 #define yyltype YYLTYPE /∗ obsolescent; will be withdrawn ∗/

4.145.1.2 #define YYLTYPE_IS_DECLARED 1

4.145.1.3 #define YYLTYPE_IS_TRIVIAL 1

4.145.1.4 #define yystype YYSTYPE /∗ obsolescent; will be withdrawn ∗/

4.145.1.5 #define YYSTYPE_IS_DECLARED 1

4.145.1.6 #define YYSTYPE_IS_TRIVIAL 1

4.145.2 Typedef Documentation

4.145.2.1 typedef struct YYLTYPE YYLTYPE

4.145.2.2 typedef union YYSTYPE YYSTYPE

4.145.3 Enumeration Type Documentation

4.145.3.1 enum yytokentype

Enumerator

IDENT

STRING

NUMBER

NEWLINE

IDENT

STRING

NUMBER

NEWLINE

4.146 xen/tools/libxl/xl_cmdimpl.c File Reference

```
#include "libxl_osdeps.h"
#include <stdio.h>
#include <assert.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <time.h>
#include <getopt.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/time.h>
#include <fcntl.h>
#include <signal.h>
#include <sys/socket.h>
#include <sys/select.h>
#include <sys/utsname.h>
#include <xenctrl.h>
#include <ctype.h>
#include <inttypes.h>
#include "libxl.h"
#include "libxl_utils.h"
#include "libxlutil.h"
#include "xl.h"
```

Data Structures

- struct save_file_header
- struct domain_create

Macros

- #define CHK_ERRNO(call)
- #define MUST(call)
- #define SAVEFILE_BYTEORDER_VALUE ((uint32_t)0x01020304UL)
- #define LOG(_f, _a...) dolog(__FILE__, __LINE__, __func__, _f "\n", ##_a)
- #define DSTATE_INITIAL 0
- #define DSTATE_TAP 1
- #define DSTATE_PHYSPATH 2
- #define DSTATE_VIRTPATH 3
- #define DSTATE_VIRTTYPE 4
- #define DSTATE_RW 5
- #define DSTATE_TERMINAL 6
- #define OPTDATA_LEFT (hdr.optional_data_len - (optdata_here - optdata_begin))
- #define WITH_OPTDATA(amt, body)
- #define ADD_OPTDATA(ptr, len)

Functions

- void help (const char *command)
- int main_memmax (int argc, char **argv)
- int main_memset (int argc, char **argv)

- int main_cd_eject (int argc, char **argv)
- int main_cd_insert (int argc, char **argv)
- int main_console (int argc, char **argv)
- int main_vncviewer (int argc, char **argv)
- int main_pcilist_assignable (int argc, char **argv)
- int main_pcilist (int argc, char **argv)
- int main_pcidetach (int argc, char **argv)
- int main_pciattach (int argc, char **argv)
- int main_restore (int argc, char **argv)
- int main_migrate_receive (int argc, char **argv)
- int main_save (int argc, char **argv)
- int main_migrate (int argc, char **argv)
- int main_dump_core (int argc, char **argv)
- int main_pause (int argc, char **argv)
- int main_unpause (int argc, char **argv)
- int main_destroy (int argc, char **argv)
- int main_shutdown (int argc, char **argv)
- int main_reboot (int argc, char **argv)
- int main_list (int argc, char **argv)
- int main_list_vm (int argc, char **argv)
- int main_create (int argc, char **argv)
- int main_button_press (int argc, char **argv)
- int main_vcpulist (int argc, char **argv)
- int main_vcpupin (int argc, char **argv)
- int main_vcpuset (int argc, char **argv)
- int main_info (int argc, char **argv)
- int main_sched_credit (int argc, char **argv)
- int main_domid (int argc, char **argv)
- int main_domname (int argc, char **argv)
- int main_rename (int argc, char **argv)
- int main_trigger (int argc, char **argv)
- int main_sysrq (int argc, char **argv)
- int main_debug_keys (int argc, char **argv)
- int main_dmesg (int argc, char **argv)
- int main_top (int argc, char **argv)
- int main_networkattach (int argc, char **argv)
- int main_networklist (int argc, char **argv)
- int main_networkdetach (int argc, char **argv)
- int main_blockattach (int argc, char **argv)
- int main_blocklist (int argc, char **argv)
- int main_blockdetach (int argc, char **argv)
- int main_network2attach (int argc, char **argv)
- int main_network2list (int argc, char **argv)
- int main_network2detach (int argc, char **argv)
- int main_uptime (int argc, char **argv)
- int main_tmem_list (int argc, char **argv)
- int main_tmem_freeze (int argc, char **argv)
- int main_tmem_destroy (int argc, char **argv)
- int main_tmem_thaw (int argc, char **argv)
- int main_tmem_set (int argc, char **argv)
- int main_tmem_shared_auth (int argc, char **argv)
- int main_tmem_freeable (int argc, char **argv)
- int main_cpupoolcreate (int argc, char **argv)
- int main_cpupoollist (int argc, char **argv)
- int main_cpupooldestroy (int argc, char **argv)

- int main_cpupoolrename (int argc, char **argv)
- int main_cpupoolcpuadd (int argc, char **argv)
- int main_cpupoolcpuremove (int argc, char **argv)
- int main_cpupoolmigrate (int argc, char **argv)
- int main_cpupoolnumasplit (int argc, char **argv)

Variables

- int logfile = 2
- libxl_ctx ctx

4.146.1 Macro Definition Documentation

4.146.1.1 #define ADD_OPTDATA(*ptr, len*)

Value:

```
({                                                  \
    if ((len)) {                                                          \
        hdr.optional_data_len += (len);                                   \
        optdata_begin = xrealloc(optdata_begin, hdr.optional_data_len); \
        memcpy(optdata_begin + hdr.optional_data_len - (len),             \
               (ptr), (len));                                             \
    }                                                                     \
                        })
```

4.146.1.2 #define CHK_ERRNO(*call*)

Value:

```
({                                                  \
        int chk_errno = (call);                                           \
        if (chk_errno < 0) {                                                  \
            fprintf(stderr,"xl: fatal error: %s:%d: %s: %s\n",            \
                    __FILE__,__LINE__, strerror(chk_errno), #call);       \
            exit(-ERROR_FAIL);                                            \
        }                                                                 \
    })
```

4.146.1.3 #define DSTATE_INITIAL 0

4.146.1.4 #define DSTATE_PHYSPATH 2

4.146.1.5 #define DSTATE_RW 5

4.146.1.6 #define DSTATE_TAP 1

4.146.1.7 #define DSTATE_TERMINAL 6

4.146.1.8 #define DSTATE_VIRTPATH 3

4.146.1.9 #define DSTATE_VIRTTYPE 4

4.146.1.10 #define LOG(*_f, _a...*) dolog(__FILE__, __LINE__, __func__, _f "\n", ##_a)

4.146.1.11 #define MUST(*call*)

Value:

```
({                                                      \
        int must_rc = (call);                                                   \
        if (must_rc < 0) {                                                          \
            fprintf(stderr,"xl: fatal error: %s:%d, rc=%d: %s\n",               \
                    __FILE__,__LINE__, must_rc, #call);                         \
            exit(-must_rc);                                                     \
        }                                                                       \
    })
```

4.146.1.12 #define OPTDATA_LEFT (hdr.optional_data_len - (optdata_here - optdata_begin))

4.146.1.13 #define SAVEFILE_BYTEORDER_VALUE ((uint32_t)0x01020304UL)

4.146.1.14 #define WITH_OPTDATA(*amt*, *body*)

Value:

```
if (OPTDATA_LEFT < (amt)) {                             \
                fprintf(stderr, "Savefile truncated.\n");       \
                return ERROR_INVAL;                             \
            } else {                                            \
                body;                                           \
                optdata_here += (amt);                          \
            }
```

4.146.2 Function Documentation

4.146.2.1 void help (const char * *command*)

4.146.2.2 int main_blockattach (int *argc*, char ** *argv*)

4.146.2.3 int main_blockdetach (int *argc*, char ** *argv*)

4.146.2.4 int main_blocklist (int *argc*, char ** *argv*)

4.146.2.5 int main_button_press (int *argc*, char ** *argv*)

4.146.2.6 int main_cd_eject (int *argc*, char ** *argv*)

4.146.2.7 int main_cd_insert (int *argc*, char ** *argv*)

4.146.2.8 int main_console (int *argc*, char ** *argv*)

4.146.2.9 int main_cpupoolcpuadd (int *argc*, char ** *argv*)

4.146.2.10 int main_cpupoolcpuremove (int *argc*, char ** *argv*)

4.146.2.11 int main_cpupoolcreate (int *argc*, char ** *argv*)

4.146.2.12 int main_cpupooldestroy (int *argc*, char ** *argv*)

4.146.2.13 int main_cpupoollist (int *argc*, char ** *argv*)

4.146.2.14 int main_cpupoolmigrate (int *argc*, char ** *argv*)

4.146.2.15 int main_cpupoolnumasplit (int *argc*, char ** *argv*)

4.146.2.16 int main_cpupoolrename (int *argc*, char ** *argv*)

4.146.2.17 int main_create (int *argc*, char ** *argv*)

4.146.2.18 int main_debug_keys (int *argc*, char ** *argv*)

4.146.2.19 int main_destroy (int *argc*, char ** *argv*)

4.146.2.20 int main_dmesg (int *argc*, char ** *argv*)

4.146.2.21 int main_domid (int *argc*, char ** *argv*)

4.146.2.22 int main_domname (int *argc*, char ** *argv*)

4.146.2.23 int main_dump_core (int *argc*, char ** *argv*)

4.146.2.24 int main_info (int *argc*, char ** *argv*)

4.146.2.25 int main_list (int *argc*, char ** *argv*)

4.146.2.26 int main_list_vm (int *argc*, char ** *argv*)

4.146.2.27 int main_memmax (int *argc*, char ** *argv*)

4.146.2.28 int main_memset (int *argc*, char ** *argv*)

4.146.2.29 int main_migrate (int *argc*, char ** *argv*)

4.146.2.30 int main_migrate_receive (int *argc*, char ** *argv*)

4.146.2.31 int main_network2attach (int *argc*, char ** *argv*)

4.146.2.32 int main_network2detach (int *argc*, char ** *argv*)

4.146.2.33 int main_network2list (int *argc*, char ** *argv*)

4.146.2.34 int main_networkattach (int *argc*, char ** *argv*)

4.146.2.35 int main_networkdetach (int *argc*, char ** *argv*)

4.146.2.36 int main_networklist (int *argc*, char ** *argv*)

4.146.2.37 int main_pause (int *argc*, char ** *argv*)

4.146.2.38 int main_pciattach (int *argc*, char ** *argv*)

4.146.2.39 int main_pcidetach (int *argc*, char ** *argv*)

4.146.2.40 int main_pcilist (int *argc*, char ** *argv*)

4.146.2.41 int main_pcilist_assignable (int *argc*, char ** *argv*)

4.146.2.42 int main_reboot (int *argc*, char ** *argv*)

4.146.2.43 int main_rename (int *argc*, char ** *argv*)

4.146.2.44 int main_restore (int *argc*, char ** *argv*)

4.146.2.45 int main_save (int *argc*, char ** *argv*)

4.146.2.46 int main_sched_credit (int *argc*, char ** *argv*)

4.146.2.47 int main_shutdown (int *argc*, char ** *argv*)

4.146.2.48 int main_sysrq (int *argc*, char ** *argv*)

4.146.2.49 int main_tmem_destroy (int *argc*, char ** *argv*)

4.146.2.50 int main_tmem_freeable (int *argc*, char ** *argv*)

4.146.2.51 int main_tmem_freeze (int *argc*, char ** *argv*)

4.146.2.52 int main_tmem_list (int *argc*, char ** *argv*)

4.146.2.53 int main_tmem_set (int *argc*, char ** *argv*)

4.146.2.54 int main_tmem_shared_auth (int *argc*, char ** *argv*)

4.146.2.55 int main_tmem_thaw (int *argc*, char ** *argv*)

4.146.2.56 int main_top (int *argc*, char ** *argv*)

4.146.2.57 int main_trigger (int *argc*, char ** *argv*)

4.146.2.58 int main_unpause (int *argc*, char ** *argv*)

4.146.2.59 int main_uptime (int *argc*, char ** *argv*)

4.146.2.60 int main_vcpulist (int *argc*, char ** *argv*)

4.146.2.61 int main_vcpupin (int *argc*, char ** *argv*)

4.146.2.62 int main_vcpuset (int *argc*, char ** *argv*)

4.146.2.63 int main_vncviewer (int *argc*, char ** *argv*)

4.146.3 Variable Documentation

4.146.3.1 libxl_ctx ctx

4.146.3.2 int logfile = 2

4.147 xen/tools/python/xen/lowlevel/xl/xl.c File Reference

```
#include <Python.h>
#include <stdio.h>
#include <assert.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <time.h>
#include <getopt.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/time.h>
#include <fcntl.h>
#include <signal.h>
#include <sys/socket.h>
#include <sys/select.h>
#include <arpa/inet.h>
#include <xenctrl.h>
#include <ctype.h>
#include <inttypes.h>
#include <libxl.h>
#include <libxl_utils.h>
#include "_pyxl_types.h"
```

Data Structures

- struct XlObject

Macros

- #define ARRAY_SIZE(x) (sizeof(x) / sizeof((x)[0]))
- #define PyMODINIT_FUNC DL_EXPORT(void)
- #define CLS "ctx"
- #define Py_ssize_t int
- #define _INT_CONST(m, c) PyModule_AddIntConstant(m, #c, c)
- #define _INT_CONST_LIBXL(m, c) PyModule_AddIntConstant(m, #c, LIBXL_ ## c)

Functions

- int genwrap__obj_init (PyObject *self, PyObject *args, PyObject *kwds)
- int genwrap__string_set (PyObject *v, char **str)
- PyObject * genwrap__string_get (char **str)
- PyObject * genwrap__ull_get (unsigned long long val)
- int genwrap__ull_set (PyObject *v, unsigned long long *val, unsigned long long mask)
- PyObject * genwrap__ll_get (long long val)
- int genwrap__ll_set (PyObject *v, long long *val, long long mask)
- int attrib__libxl_cpuid_policy_list_set (PyObject *v, libxl_cpuid_policy_list *pptr)
- int attrib__void_ptr_set (PyObject *v, void **caas_info)
- int attrib__libxl_cpumap_set (PyObject *v, libxl_cpumap *pptr)
- int attrib__libxl_cpuarray_set (PyObject *v, libxl_cpuarray *pptr)
- int attrib__libxl_domain_build_state_ptr_set (PyObject *v, libxl_domain_build_state **pptr)
- int attrib__libxl_file_reference_set (PyObject *v, libxl_file_reference *pptr)

- int attrib__libxl_hwcap_set (PyObject *v, libxl_hwcap *pptr)
- int attrib__libxl_key_value_list_set (PyObject *v, libxl_key_value_list *pptr)
- int attrib__libxl_mac_set (PyObject *v, libxl_mac *pptr)
- int attrib__libxl_string_list_set (PyObject *v, libxl_string_list *pptr)
- int attrib__libxl_uuid_set (PyObject *v, libxl_uuid *pptr)
- int attrib__struct_in_addr_set (PyObject *v, struct in_addr *pptr)
- PyObject * attrib__libxl_cpuid_policy_list_get (libxl_cpuid_policy_list *pptr)
- PyObject * attrib__void_ptr_get (void **caas_info)
- PyObject * attrib__libxl_cpumap_get (libxl_cpumap *pptr)
- PyObject * attrib__libxl_cpuarray_get (libxl_cpuarray *pptr)
- PyObject * attrib__libxl_domain_build_state_ptr_get (libxl_domain_build_state **pptr)
- PyObject * attrib__libxl_file_reference_get (libxl_file_reference *pptr)
- PyObject * attrib__libxl_hwcap_get (libxl_hwcap *pptr)
- PyObject * attrib__libxl_key_value_list_get (libxl_key_value_list *pptr)
- PyObject * attrib__libxl_mac_get (libxl_mac *pptr)
- PyObject * attrib__libxl_string_list_get (libxl_string_list *pptr)
- PyObject * attrib__libxl_uuid_get (libxl_uuid *pptr)
- PyObject * attrib__struct_in_addr_get (struct in_addr *pptr)
- PyMODINIT_FUNC initxl (void)

4.147.1 Macro Definition Documentation

4.147.1.1 #define _INT_CONST(*m,* c) PyModule_AddIntConstant(m, #c, c)

4.147.1.2 #define _INT_CONST_LIBXL(*m,* c) PyModule_AddIntConstant(m, #c, LIBXL_ ## c)

4.147.1.3 #define ARRAY_SIZE(*x*) (sizeof(x) / sizeof((x)[0]))

4.147.1.4 #define CLS "ctx"

4.147.1.5 #define Py_ssize_t int

4.147.1.6 #define PyMODINIT_FUNC DL_EXPORT(void)

4.147.2 Function Documentation

4.147.2.1 PyObject* attrib__libxl_cpuarray_get (libxl_cpuarray * *pptr*)

4.147.2.2 int attrib__libxl_cpuarray_set (PyObject * *v,* libxl_cpuarray * *pptr*)

4.147.2.3 PyObject* attrib__libxl_cpuid_policy_list_get (libxl_cpuid_policy_list * *pptr*)

4.147.2.4 int attrib__libxl_cpuid_policy_list_set (PyObject * *v,* libxl_cpuid_policy_list * *pptr*)

4.147.2.5 PyObject* attrib__libxl_cpumap_get (libxl_cpumap * *pptr*)

4.147.2.6 int attrib__libxl_cpumap_set (PyObject * *v,* libxl_cpumap * *pptr*)

4.147.2.7 PyObject* attrib__libxl_domain_build_state_ptr_get (libxl_domain_build_state ** *pptr*)

4.147.2.8 int attrib__libxl_domain_build_state_ptr_set (PyObject * *v,* libxl_domain_build_state ** *pptr*)

4.147.2.9 PyObject* attrib__libxl_file_reference_get (libxl_file_reference * *pptr*)

4.147.2.10 int attrib__libxl_file_reference_set (PyObject * *v,* **libxl_file_reference** * *pptr*)

4.147.2.11 PyObject* attrib__libxl_hwcap_get (**libxl_hwcap** * *pptr*)

4.147.2.12 int attrib__libxl_hwcap_set (PyObject * *v,* **libxl_hwcap** * *pptr*)

4.147.2.13 PyObject* attrib__libxl_key_value_list_get (**libxl_key_value_list** * *pptr*)

4.147.2.14 int attrib__libxl_key_value_list_set (PyObject * *v,* **libxl_key_value_list** * *pptr*)

4.147.2.15 PyObject* attrib__libxl_mac_get (**libxl_mac** * *pptr*)

4.147.2.16 int attrib__libxl_mac_set (PyObject * *v,* **libxl_mac** * *pptr*)

4.147.2.17 PyObject* attrib__libxl_string_list_get (**libxl_string_list** * *pptr*)

4.147.2.18 int attrib__libxl_string_list_set (PyObject * *v,* **libxl_string_list** * *pptr*)

4.147.2.19 PyObject* attrib__libxl_uuid_get (libxl_uuid * *pptr*)

4.147.2.20 int attrib__libxl_uuid_set (PyObject * *v,* libxl_uuid * *pptr*)

4.147.2.21 PyObject* attrib__struct_in_addr_get (struct in_addr * *pptr*)

4.147.2.22 int attrib__struct_in_addr_set (PyObject * *v,* struct in_addr * *pptr*)

4.147.2.23 PyObject* attrib__void_ptr_get (void ** *caas_info*)

4.147.2.24 int attrib__void_ptr_set (PyObject * *v,* void ** *caas_info*)

4.147.2.25 PyObject* genwrap__ll_get (long long *val*)

4.147.2.26 int genwrap__ll_set (PyObject * *v,* long long * *val,* long long *mask*)

4.147.2.27 int genwrap__obj_init (PyObject * *self,* PyObject * *args,* PyObject * *kwds*)

4.147.2.28 PyObject* genwrap__string_get (char ** *str*)

4.147.2.29 int genwrap__string_set (PyObject * *v,* char ** *str*)

4.147.2.30 PyObject* genwrap__ull_get (unsigned long long *val*)

4.147.2.31 int genwrap__ull_set (PyObject * *v,* unsigned long long * *val,* unsigned long long *mask*)

4.147.2.32 **PyMODINIT_FUNC** initxl (void)

4.148 xen/tools/vtpm_manager/crypto/crypto.h File Reference

```
#include <stddef.h>
#include <stdint.h>
#include <stdbool.h>
#include "tcg.h"
#include "sym_crypto.h"
```

Data Structures

- struct CRYPTO_INFO

Macros

- #define CRYPTO_MAX_SIG_SIZE (2048 / 8)
- #define CRYPTO_MAX_RSA_KEY_SIZE (4096 / 8)
- #define OAEP_P "TCPA"
- #define OAEP_P_SIZE 4
- #define CRYPTO_ALGORITH_RSA 0x01
- #define CRYPTO_ES_NONE 0x0001
- #define CRYPTO_ES_RSAESPKCSv15 0x0002
- #define CRYPTO_ES_RSAESOAEP_SHA1_MGF1 0x0003
- #define CRYPTO_SS_NONE 0x0001
- #define CRYPTO_SS_RSASSAPKCS1v15_SHA1 0x0002
- #define CRYPTO_SS_RSASSAPKCS1v15_DER 0x0003

Typedefs

- typedef struct CRYPTO_INFO CRYPTO_INFO

Functions

- void Crypto_Init (const BYTE *rand, int size)
- void Crypto_Exit ()
- void Crypto_GetRandom (void *data, int size)
- void Crypto_HMAC (const BYTE *text, int text_len, const BYTE *key, int key_len, BYTE *digest)
- TPM_RESULT Crypto_HMAC_buf (const buffer_t *text, const buffer_t *key, BYTE *o_digest)
- void Crypto_SHA1Full (const BYTE *text, UINT32 size, BYTE *hash)
- TPM_RESULT Crypto_SHA1Full_buf (const buffer_t *buf, BYTE *o_hash)
- void Crypto_SHA1Start (UINT32 *maxNumBytes)
- void Crypto_SHA1Update (int numBytes, const BYTE *hashData)
- void Crypto_SHA1Complete (int hashDataSize, const BYTE *hashData, BYTE *hashValue)
- void Crypto_RSACreateKey (UINT32 keySize, UINT32 pubExpSize, BYTE *pubExp, UINT32 *privExpSize, BYTE *privExp, UINT32 *modulusSize, BYTE *modulus, CRYPTO_INFO *keys)
- void Crypto_RSABuildCryptoInfo (UINT32 pubExpSize, BYTE *pubExp, UINT32 privExpSize, BYTE *privExp, UINT32 modulusSize, BYTE *modulus, CRYPTO_INFO *cryptoInfo)
- void Crypto_RSABuildCryptoInfoPublic (UINT32 pubExpSize, BYTE *pubExp, UINT32 modulusSize, BYTE *modulus, CRYPTO_INFO *cryptoInfo)
- void Crypto_RSACryptoInfoFree (CRYPTO_INFO *cryptoInfo)
- TPM_RESULT Crypto_RSAPackCryptoInfo (const CRYPTO_INFO *cryptoInfo, BYTE **io_buf, UINT32 *io_buflen)
- TPM_RESULT Crypto_RSAUnpackCryptoInfo (CRYPTO_INFO *ci, BYTE *in, UINT32 len, UINT32 *o_lenread)
- int Crypto_RSAEnc (CRYPTO_INFO *keys, UINT32 inDataSize, BYTE *inData, UINT32 *outDataSize, BYTE *outData)
- int Crypto_RSADec (CRYPTO_INFO *keys, UINT32 inDataSize, BYTE *inData, UINT32 *outDataSize, BYTE *outData)
- int Crypto_RSASign (CRYPTO_INFO *keys, UINT32 inDataSize, BYTE *inData, UINT32 *sigSize, BYTE *sig)
- bool Crypto_RSAVerify (CRYPTO_INFO *keys, UINT32 inDataSize, BYTE *inData, UINT32 sigSize, BYTE *sig)

- int RSA_verify_DER (int dtype, unsigned char ∗m, unsigned int m_len, unsigned char ∗sigbuf, unsigned int siglen, CRYPTO_INFO ∗key)
- int RSA_sign_DER (int type, unsigned char ∗m, unsigned int m_len, unsigned char ∗sigret, unsigned int ∗siglen, CRYPTO_INFO ∗key)

4.148.1 Macro Definition Documentation

4.148.1.1 #define CRYPTO_ALGORITH_RSA 0x01

4.148.1.2 #define CRYPTO_ES_NONE 0x0001

4.148.1.3 #define CRYPTO_ES_RSAESOAEP_SHA1_MGF1 0x0003

4.148.1.4 #define CRYPTO_ES_RSAESPKCSv15 0x0002

4.148.1.5 #define CRYPTO_MAX_RSA_KEY_SIZE (4096 / 8)

4.148.1.6 #define CRYPTO_MAX_SIG_SIZE (2048 / 8)

4.148.1.7 #define CRYPTO_SS_NONE 0x0001

4.148.1.8 #define CRYPTO_SS_RSASSAPKCS1v15_DER 0x0003

4.148.1.9 #define CRYPTO_SS_RSASSAPKCS1v15_SHA1 0x0002

4.148.1.10 #define OAEP_P "TCPA"

4.148.1.11 #define OAEP_P_SIZE 4

4.148.2 Typedef Documentation

4.148.2.1 typedef struct CRYPTO_INFO CRYPTO_INFO

4.148.3 Function Documentation

4.148.3.1 void Crypto_Exit ( )

4.148.3.2 void Crypto_GetRandom (void ∗ *data,* int *size*)

4.148.3.3 void Crypto_HMAC (const BYTE ∗ *text,* int *text_len,* const BYTE ∗ *key,* int *key_len,* BYTE ∗ *digest*)

4.148.3.4 TPM_RESULT Crypto_HMAC_buf (const buffer_t ∗ *text,* const buffer_t ∗ *key,* BYTE ∗ *o_digest*)

4.148.3.5 void Crypto_Init (const BYTE ∗ *rand,* int *size*)

4.148.3.6 void Crypto_RSABuildCryptoInfo (UINT32 *pubExpSize,* BYTE ∗ *pubExp,* UINT32 *privExpSize,* BYTE ∗ *privExp,* UINT32 *modulusSize,* BYTE ∗ *modulus,* CRYPTO_INFO ∗ *cryptoInfo*)

4.148.3.7 void Crypto_RSABuildCryptoInfoPublic (UINT32 *pubExpSize,* BYTE ∗ *pubExp,* UINT32 *modulusSize,* BYTE ∗ *modulus,* CRYPTO_INFO ∗ *cryptoInfo*)

4.148.3.8 void Crypto_RSACreateKey (UINT32 *keySize,* UINT32 *pubExpSize,* BYTE ∗ *pubExp,* UINT32 ∗ *privExpSize,* BYTE ∗ *privExp,* UINT32 ∗ *modulusSize,* BYTE ∗ *modulus,* CRYPTO_INFO ∗ *keys*)

4.148.3.9 void Crypto_RSACryptoInfoFree (CRYPTO_INFO ∗ *cryptoInfo*)

4.148.3.10 int Crypto_RSADec (CRYPTO_INFO * *keys*, UINT32 *inDataSize*, BYTE * *inData*, UINT32 * *outDataSize*, BYTE * *outData*)

4.148.3.11 int Crypto_RSAEnc (CRYPTO_INFO * *keys*, UINT32 *inDataSize*, BYTE * *inData*, UINT32 * *outDataSize*, BYTE * *outData*)

4.148.3.12 TPM_RESULT Crypto_RSAPackCryptoInfo (const CRYPTO_INFO * *cryptoInfo*, BYTE ** *io_buf*, UINT32 * *io_buflen*)

4.148.3.13 int Crypto_RSASign (CRYPTO_INFO * *keys*, UINT32 *inDataSize*, BYTE * *inData*, UINT32 * *sigSize*, BYTE * *sig*)

4.148.3.14 TPM_RESULT Crypto_RSAUnpackCryptoInfo (CRYPTO_INFO * *ci*, BYTE * *in*, UINT32 *len*, UINT32 * *o_lenread*)

4.148.3.15 bool Crypto_RSAVerify (CRYPTO_INFO * *keys*, UINT32 *inDataSize*, BYTE * *inData*, UINT32 *sigSize*, BYTE * *sig*)

4.148.3.16 void Crypto_SHA1Complete (int *hashDataSize*, const BYTE * *hashData*, BYTE * *hashValue*)

4.148.3.17 void Crypto_SHA1Full (const BYTE * *text*, UINT32 *size*, BYTE * *hash*)

4.148.3.18 TPM_RESULT Crypto_SHA1Full_buf (const buffer_t * *buf*, BYTE * *o_hash*)

4.148.3.19 void Crypto_SHA1Start (UINT32 * *maxNumBytes*)

4.148.3.20 void Crypto_SHA1Update (int *numBytes*, const BYTE * *hashData*)

4.148.3.21 int RSA_sign_DER (int *type*, unsigned char * *m*, unsigned int *m_len*, unsigned char * *sigret*, unsigned int * *siglen*, CRYPTO_INFO * *key*)

4.148.3.22 int RSA_verify_DER (int *dtype*, unsigned char * *m*, unsigned int *m_len*, unsigned char * *sigbuf*, unsigned int *siglen*, CRYPTO_INFO * *key*)

4.149 xen/tools/vtpm_manager/crypto/rsa.c File Reference

```
#include <string.h>
#include <openssl/crypto.h>
#include <openssl/evp.h>
#include <openssl/bn.h>
#include <openssl/rsa.h>
#include <openssl/rand.h>
#include <openssl/x509.h>
#include <openssl/err.h>
#include <stdio.h>
#include "tcg.h"
#include "buffer.h"
#include "crypto.h"
#include "log.h"
```

Functions

- void Crypto_RSACreateKey (UINT32 keySize, UINT32 pubExpSize, BYTE *pubExp, UINT32 *privExpSize, BYTE *privExp, UINT32 *modulusSize, BYTE *modulus, CRYPTO_INFO *keys)

- void Crypto_RSABuildCryptoInfo (UINT32 pubExpSize, BYTE *pubExp, UINT32 privExpSize, BYTE *privExp, UINT32 modulusSize, BYTE *modulus, CRYPTO_INFO *cryptoInfo)
- void Crypto_RSABuildCryptoInfoPublic (UINT32 pubExpSize, BYTE *pubExp, UINT32 modulusSize, BYTE *modulus, CRYPTO_INFO *cryptoInfo)
- void Crypto_RSACryptoInfoFree (CRYPTO_INFO *cryptoInfo)
- int Crypto_RSAEnc (CRYPTO_INFO *key, UINT32 inDataSize, BYTE *inData, UINT32 *outDataSize, BYTE *outData)
- int Crypto_RSADec (CRYPTO_INFO *key, UINT32 inDataSize, BYTE *inData, UINT32 *outDataSize, BYTE *outData)
- int Crypto_RSASign (CRYPTO_INFO *key, UINT32 inDataSize, BYTE *inData, UINT32 *sigSize, BYTE *sig)
- bool Crypto_RSAVerify (CRYPTO_INFO *key, UINT32 inDataSize, BYTE *inData, UINT32 sigSize, BYTE *sig)
- TPM_RESULT Crypto_RSAPackCryptoInfo (const CRYPTO_INFO *cryptoInfo, BYTE **io_buf, UINT32 *io_buflen)
- TPM_RESULT Crypto_RSAUnpackCryptoInfo (CRYPTO_INFO *ci, BYTE *in, UINT32 len, UINT32 *o_lenread)

4.149.1 Function Documentation

4.149.1.1 void Crypto_RSABuildCryptoInfo (UINT32 *pubExpSize*, BYTE * *pubExp*, UINT32 *privExpSize*, BYTE * *privExp*, UINT32 *modulusSize*, BYTE * *modulus*, CRYPTO_INFO * *cryptoInfo*)

4.149.1.2 void Crypto_RSABuildCryptoInfoPublic (UINT32 *pubExpSize*, BYTE * *pubExp*, UINT32 *modulusSize*, BYTE * *modulus*, CRYPTO_INFO * *cryptoInfo*)

4.149.1.3 void Crypto_RSACreateKey (UINT32 *keySize*, UINT32 *pubExpSize*, BYTE * *pubExp*, UINT32 * *privExpSize*, BYTE * *privExp*, UINT32 * *modulusSize*, BYTE * *modulus*, CRYPTO_INFO * *keys*)

4.149.1.4 void Crypto_RSACryptoInfoFree (CRYPTO_INFO * *cryptoInfo*)

4.149.1.5 int Crypto_RSADec (CRYPTO_INFO * *key*, UINT32 *inDataSize*, BYTE * *inData*, UINT32 * *outDataSize*, BYTE * *outData*)

4.149.1.6 int Crypto_RSAEnc (CRYPTO_INFO * *key*, UINT32 *inDataSize*, BYTE * *inData*, UINT32 * *outDataSize*, BYTE * *outData*)

4.149.1.7 TPM_RESULT Crypto_RSAPackCryptoInfo (const CRYPTO_INFO * *cryptoInfo*, BYTE ** *io_buf*, UINT32 * *io_buflen*)

4.149.1.8 int Crypto_RSASign (CRYPTO_INFO * *key*, UINT32 *inDataSize*, BYTE * *inData*, UINT32 * *sigSize*, BYTE * *sig*)

4.149.1.9 TPM_RESULT Crypto_RSAUnpackCryptoInfo (CRYPTO_INFO * *ci*, BYTE * *in*, UINT32 *len*, UINT32 * *o_lenread*)

4.149.1.10 bool Crypto_RSAVerify (CRYPTO_INFO * *key*, UINT32 *inDataSize*, BYTE * *inData*, UINT32 *sigSize*, BYTE * *sig*)

4.150 xen/tools/vtpm_manager/crypto/sym_crypto.c File Reference

```
#include <openssl/evp.h>
#include <openssl/rand.h>
#include "tcg.h"
#include "log.h"
#include "sym_crypto.h"
```

Typedefs

- typedef enum crypt_op_type_t crypt_op_type_t

Enumerations

- enum crypt_op_type_t { CRYPT_ENCRYPT, CRYPT_DECRYPT }

Functions

- TPM_RESULT ossl_symcrypto_op (symkey_t *key, const buffer_t *in, const buffer_t *iv, buffer_t *out, crypt_op_type_t optype)
- TPM_RESULT Crypto_symcrypto_initkey (symkey_t *key, const buffer_t *keybits)
- TPM_RESULT Crypto_symcrypto_genkey (symkey_t *key)
- TPM_RESULT Crypto_symcrypto_encrypt (symkey_t *key, const buffer_t *clear, buffer_t *o_cipher)
- TPM_RESULT Crypto_symcrypto_decrypt (symkey_t *key, const buffer_t *cipher, buffer_t *o_clear)
- TPM_RESULT Crypto_symcrypto_freekey (symkey_t *key)

Variables

- const EVP_CIPHER * SYM_CIPHER = NULL
- const BYTE ZERO_IV [EVP_MAX_IV_LENGTH] = {0}

4.150.1 Typedef Documentation

4.150.1.1 typedef enum crypt_op_type_t crypt_op_type_t

4.150.2 Enumeration Type Documentation

4.150.2.1 enum crypt_op_type_t

Enumerator

CRYPT_ENCRYPT

CRYPT_DECRYPT

4.150.3 Function Documentation

4.150.3.1 TPM_RESULT Crypto_symcrypto_decrypt (symkey_t * *key*, const buffer_t * *cipher*, buffer_t * *o_clear*)

4.150.3.2 TPM_RESULT Crypto_symcrypto_encrypt (symkey_t * *key*, const buffer_t * *clear*, buffer_t * *o_cipher*)

4.150.3.3 TPM_RESULT Crypto_symcrypto_freekey (symkey_t * *key*)

4.150.3.4 TPM_RESULT Crypto_symcrypto_genkey (symkey_t * *key*)

4.150.3.5 TPM_RESULT Crypto_symcrypto_initkey (symkey_t * *key*, const buffer_t * *keybits*)

4.150.3.6 TPM_RESULT ossl_symcrypto_op (symkey_t * *key*, const buffer_t * *in*, const buffer_t * *iv*, buffer_t * *out*, crypt_op_type_t *optype*)

4.150.4 Variable Documentation

4.150.4.1 const EVP_CIPHER* SYM_CIPHER = NULL

4.150.4.2 const BYTE ZERO_IV[EVP_MAX_IV_LENGTH] = {0}

4.151 xen/tools/vtpm_manager/manager/dmictl.c File Reference

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include "vtpmpriv.h"
#include "bsg.h"
#include "buffer.h"
#include "log.h"
#include "hashtable.h"
#include "hashtable_itr.h"
#include "vtpm_ipc.h"
```

Macros

- #define TPM_EMULATOR_PATH "/usr/bin/vtpmd"

Functions

- TPM_RESULT init_dmi (UINT32 dmi_id, BYTE dmi_type, VTPM_DMI_RESOURCE **dmi_res)
- TPM_RESULT close_dmi (VTPM_DMI_RESOURCE *dmi_res)
- void free_dmi (VTPM_DMI_RESOURCE *dmi_res)
- TPM_RESULT VTPM_Handle_New_DMI (const buffer_t *param_buf)
- TPM_RESULT VTPM_Handle_Close_DMI (const buffer_t *param_buf)
- TPM_RESULT VTPM_Handle_Delete_DMI (const buffer_t *param_buf)

4.151.1 Macro Definition Documentation

4.151.1.1 #define TPM_EMULATOR_PATH "/usr/bin/vtpmd"

4.151.2 Function Documentation

4.151.2.1 TPM_RESULT close_dmi (VTPM_DMI_RESOURCE * *dmi_res*)

4.151.2.2 void free_dmi (VTPM_DMI_RESOURCE * *dmi_res*)

4.151.2.3 TPM_RESULT init_dmi (UINT32 *dmi_id,* BYTE *dmi_type,* VTPM_DMI_RESOURCE ** *dmi_res*)

4.151.2.4 TPM_RESULT VTPM_Handle_Close_DMI (const buffer_t * *param_buf*)

4.151.2.5 TPM_RESULT VTPM_Handle_Delete_DMI (const buffer_t * *param_buf*)

4.151.2.6 TPM_RESULT VTPM_Handle_New_DMI (const buffer_t * *param_buf*)

4.152 xen/tools/vtpm_manager/manager/securestorage.c File Reference

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h>
#include "tcg.h"
#include "vtpm_manager.h"
#include "vtpmpriv.h"
#include "vtsp.h"
#include "bsg.h"
#include "crypto.h"
#include "hashtable.h"
#include "hashtable_itr.h"
#include "buffer.h"
#include "log.h"
```

Functions

- TPM_RESULT envelope_encrypt (const buffer_t *inbuf, CRYPTO_INFO *asymkey, buffer_t *sealed_data)
- TPM_RESULT symkey_encrypt (const buffer_t *inbuf, CRYPTO_INFO *asymkey, buffer_t *sealed_key)
- TPM_RESULT envelope_decrypt (const buffer_t *cipher, TCS_CONTEXT_HANDLE TCSContext, TPM_HANDLE keyHandle, const TPM_AUTHDATA *key_usage_auth, buffer_t *unsealed_data)
- TPM_RESULT symkey_decrypt (const buffer_t *cipher, TCS_CONTEXT_HANDLE TCSContext, TPM_HANDLE keyHandle, const TPM_AUTHDATA *key_usage_auth, buffer_t *symkey_clear)
- TPM_RESULT VTPM_Handle_Save_NVM (VTPM_DMI_RESOURCE *myDMI, const buffer_t *inbuf, buffer_t *outbuf)
- TPM_RESULT VTPM_Handle_Load_NVM (VTPM_DMI_RESOURCE *myDMI, const buffer_t *inbuf, buffer_t *outbuf)
- TPM_RESULT VTPM_Handle_Get_NVM_Size (VTPM_DMI_RESOURCE *myDMI, const buffer_t *inbuf, buffer_t *outbuf)
- TPM_RESULT VTPM_Handle_Load_Key (VTPM_DMI_RESOURCE *myDMI, const buffer_t *inbuf, buffer_t *outbuf)
- TPM_RESULT VTPM_Handle_Save_Key (VTPM_DMI_RESOURCE *myDMI, const buffer_t *inbuf, buffer_t *outbuf)
- TPM_RESULT VTPM_SaveManagerData (void)
- TPM_RESULT VTPM_LoadManagerData (void)

4.152.1 Function Documentation

4.152.1.1 TPM_RESULT envelope_decrypt (const buffer_t * *cipher*, TCS_CONTEXT_HANDLE *TCSContext*, TPM_HANDLE *keyHandle*, const TPM_AUTHDATA * *key_usage_auth*, buffer_t * *unsealed_data*)

4.152.1.2 TPM_RESULT envelope_encrypt (const buffer_t * *inbuf*, CRYPTO_INFO * *asymkey*, buffer_t * *sealed_data*)

4.152.1.3 TPM_RESULT symkey_decrypt (const buffer_t * *cipher*, TCS_CONTEXT_HANDLE *TCSContext*, TPM_HANDLE *keyHandle*, const TPM_AUTHDATA * *key_usage_auth*, buffer_t * *symkey_clear*)

4.152.1.4 TPM_RESULT symkey_encrypt (const buffer_t * *inbuf*, CRYPTO_INFO * *asymkey*, buffer_t * *sealed_key*)

4.152.1.5 TPM_RESULT VTPM_Handle_Get_NVM_Size (VTPM_DMI_RESOURCE * *myDMI*, const buffer_t * *inbuf*, buffer_t * *outbuf*)

4.152.1.6 TPM_RESULT VTPM_Handle_Load_Key (VTPM_DMI_RESOURCE * *myDMI*, const buffer_t * *inbuf*, buffer_t * *outbuf*)

4.152.1.7 TPM_RESULT VTPM_Handle_Load_NVM (VTPM_DMI_RESOURCE * *myDMI*, const buffer_t * *inbuf*, buffer_t * *outbuf*)

4.152.1.8 TPM_RESULT VTPM_Handle_Save_Key (VTPM_DMI_RESOURCE * *myDMI*, const buffer_t * *inbuf*, buffer_t * *outbuf*)

4.152.1.9 TPM_RESULT VTPM_Handle_Save_NVM (VTPM_DMI_RESOURCE * *myDMI*, const buffer_t * *inbuf*, buffer_t * *outbuf*)

4.152.1.10 TPM_RESULT VTPM_LoadManagerData (void)

4.152.1.11 TPM_RESULT VTPM_SaveManagerData (void)

4.153 xen/tools/vtpm_manager/manager/vtpm_ipc.c File Reference

```
#include <sys/stat.h>
#include "vtpm_ipc.h"
#include "vtpmpriv.h"
#include "log.h"
```

Functions

- int vtpm_ipc_init (vtpm_ipc_handle_t *ipc_h, char *name, int flags, BOOL create, BOOL preopen)
- int vtpm_ipc_create (vtpm_ipc_handle_t *ipc_h)
- int vtpm_ipc_read (vtpm_ipc_handle_t *ipc_h, vtpm_ipc_handle_t *alt_ipc_h, BYTE *bytes, UINT32 size)
- int vtpm_ipc_write (vtpm_ipc_handle_t *ipc_h, vtpm_ipc_handle_t *alt_ipc_h, BYTE *bytes, UINT32 size)
- void vtpm_ipc_close (vtpm_ipc_handle_t *ipc_h)

Variables

- volatile sig_atomic_t IPC_QUIT_FLAG = 0

4.153.1 Function Documentation

4.153.1.1 void vtpm_ipc_close (vtpm_ipc_handle_t * *ipc_h*)

4.153.1.2 int vtpm_ipc_create (vtpm_ipc_handle_t * *ipc_h*)

4.153.1.3 int vtpm_ipc_init (vtpm_ipc_handle_t * *ipc_h*, char * *name*, int *flags*, BOOL *create*, BOOL *preopen*)

4.153.1.4 int vtpm_ipc_read (vtpm_ipc_handle_t * *ipc_h*, vtpm_ipc_handle_t * *alt_ipc_h*, BYTE * *bytes*, UINT32 *size*)

4.153.1.5 int vtpm_ipc_write (vtpm_ipc_handle_t * *ipc_h*, vtpm_ipc_handle_t * *alt_ipc_h*, BYTE * *bytes*, UINT32 *size*)

4.153.2 Variable Documentation

4.153.2.1 volatile sig_atomic_t IPC_QUIT_FLAG = 0

4.154 xen/tools/vtpm_manager/manager/vtpm_ipc.h File Reference

```
#include <signal.h>
#include "tcg.h"
```

Data Structures

- struct vtpm_ipc_handle_t

Macros

- #define VTPM_IPC_CLOSED -1

Typedefs

- typedef struct vtpm_ipc_handle_t vtpm_ipc_handle_t

Functions

- int vtpm_ipc_init (vtpm_ipc_handle_t ∗ioh, char ∗name, int flags, BOOL create, BOOL preopen)
- int vtpm_ipc_create (vtpm_ipc_handle_t ∗ioh)
- int vtpm_ipc_read (vtpm_ipc_handle_t ∗ioh, vtpm_ipc_handle_t ∗alt_ioh, BYTE ∗bytes, UINT32 size)
- int vtpm_ipc_write (vtpm_ipc_handle_t ∗ioh, vtpm_ipc_handle_t ∗alt_ioh, BYTE ∗bytes, UINT32 size)
- void vtpm_ipc_close (vtpm_ipc_handle_t ∗ioh)

Variables

- volatile sig_atomic_t IPC_QUIT_FLAG

4.154.1 Macro Definition Documentation

4.154.1.1 #define VTPM_IPC_CLOSED -1

4.154.2 Typedef Documentation

4.154.2.1 typedef struct vtpm_ipc_handle_t vtpm_ipc_handle_t

4.154.3 Function Documentation

4.154.3.1 void vtpm_ipc_close (vtpm_ipc_handle_t ∗ *ioh*)

4.154.3.2 int vtpm_ipc_create (vtpm_ipc_handle_t ∗ *ioh*)

4.154.3.3 int vtpm_ipc_init (vtpm_ipc_handle_t ∗ *ioh,* char ∗ *name,* int *flags,* BOOL *create,* BOOL *preopen*)

4.154.3.4 int vtpm_ipc_read (vtpm_ipc_handle_t ∗ *ioh,* vtpm_ipc_handle_t ∗ *alt_ioh,* BYTE ∗ *bytes,* UINT32 *size*)

4.154.3.5 int vtpm_ipc_write (vtpm_ipc_handle_t ∗ *ioh,* vtpm_ipc_handle_t ∗ *alt_ioh,* BYTE ∗ *bytes,* UINT32 *size*)

4.154.4 Variable Documentation

4.154.4.1 volatile sig_atomic_t IPC_QUIT_FLAG

4.155 xen/tools/vtpm_manager/manager/vtpm_manager.c File Reference

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include "vtpm_manager.h"
#include "vtpmpriv.h"
#include "vtsp.h"
#include "tpmddl.h"
#include "bsg.h"
#include "hashtable.h"
#include "hashtable_itr.h"
#include "log.h"
#include "buffer.h"
```

Macros

- #define RLISTSZ 6

Functions

- TPM_RESULT VTPM_FlushResources ()
- TPM_RESULT VTPM_Create_Manager ()
- TPM_RESULT VTPM_Init_Manager ()
- void VTPM_Stop_Manager ()

Variables

- VTPM_GLOBALS * vtpm_globals =NULL
- const TPM_AUTHDATA SRK_AUTH

4.155.1 Macro Definition Documentation

4.155.1.1 #define RLISTSZ 6

4.155.2 Function Documentation

4.155.2.1 TPM_RESULT VTPM_Create_Manager ( )

4.155.2.2 TPM_RESULT VTPM_FlushResources ( )

4.155.2.3 TPM_RESULT VTPM_Init_Manager ( )

4.155.2.4 void VTPM_Stop_Manager ( )

4.155.3 Variable Documentation

4.155.3.1 const TPM_AUTHDATA SRK_AUTH

Initial value:

```
= {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}
```

4.155.3.2 VTPM_GLOBALS* vtpm_globals =NULL

4.156 xen/tools/vtpm_manager/manager/vtpm_manager.h File Reference

Macros

- #define VTPM_TAG_REQ 0x01c1
- #define VTPM_TAG_RSP 0x01c4
- #define COMMAND_BUFFER_SIZE 4096
- #define VTPM_COMMAND_HEADER_SIZE_CLT (2 + 4 + 4)
- #define VTPM_COMMAND_HEADER_SIZE_SRV (4 + VTPM_COMMAND_HEADER_SIZE_CLT)
- #define VTPM_ORD_BASE 0x0000
- #define VTPM_PRIV_MASK 0x01000000
- #define VTPM_PRIV_BASE (VTPM_ORD_BASE | VTPM_PRIV_MASK)
- #define VTPM_ORD_SAVENVM (VTPM_ORD_BASE + 1)
- #define VTPM_ORD_LOADNVM (VTPM_ORD_BASE + 2)
- #define VTPM_ORD_TPMCOMMAND (VTPM_ORD_BASE + 3)
- #define VTPM_ORD_GET_MIG_KEY (VTPM_ORD_BASE + 4)
- #define VTPM_ORD_LOAD_MIG_KEY (VTPM_ORD_BASE + 5)
- #define VTPM_ORD_GETNVMSIZE (VTPM_ORD_BASE + 6)
- #define VTPM_ORD_SAVEKEY (VTPM_ORD_BASE + 7)
- #define VTPM_ORD_LOADKEY (VTPM_ORD_BASE + 8)
- #define VTPM_ORD_OPEN (VTPM_PRIV_BASE + 1)
- #define VTPM_ORD_CLOSE (VTPM_PRIV_BASE + 2)
- #define VTPM_ORD_DELETE (VTPM_PRIV_BASE + 3)
- #define VTPM_ORD_MIGRATE_IN (VTPM_PRIV_BASE + 4)
- #define VTPM_ORD_MIGRATE_OUT (VTPM_PRIV_BASE + 5)
- #define VTPM_TYPE_PVM 1
- #define VTPM_TYPE_HVM 2
- #define VTPM_SUCCESS 0
- #define VTPM_FAIL 1
- #define VTPM_UNSUPPORTED 2
- #define VTPM_FORBIDDEN 3
- #define VTPM_RESTORE_CONTEXT_FAILED 4
- #define VTPM_INVALID_REQUEST 5
- #define VTPM_TYPE_NON_MIGRATABLE 0x00
- #define VTPM_TYPE_MIGRATABLE 0x01
- #define VTPM_TYPE_MIGRATED 0xFF
- #define TPM_EMULATOR_PATH "/usr/bin/vtpmd"
- #define VTPM_BE_FNAME "/dev/vtpm"
- #define VTPM_DUMMY_TX_BE_FNAME "/var/vtpm/fifos/dummy_out.fifo"
- #define VTPM_DUMMY_RX_BE_FNAME "/var/vtpm/fifos/dummy_in.fifo"
- #define VTPM_TX_TPM_FNAME "/var/vtpm/fifos/tpm_cmd_to_%d.fifo"
- #define VTPM_RX_TPM_FNAME "/var/vtpm/fifos/tpm_rsp_from_all.fifo"
- #define VTPM_TX_VTPM_FNAME "/var/vtpm/fifos/vtpm_rsp_to_%d.fifo"
- #define VTPM_RX_VTPM_FNAME "/var/vtpm/fifos/vtpm_cmd_from_all.fifo"
- #define VTPM_TX_HP_FNAME "/var/vtpm/fifos/to_console.fifo"
- #define VTPM_RX_HP_FNAME "/var/vtpm/fifos/from_console.fifo"
- #define VTPM_TYPE_PVM_STRING "pvm"
- #define VTPM_TYPE_HVM_STRING "hvm"

4.156.1 Macro Definition Documentation

4.156.1.1 #define COMMAND_BUFFER_SIZE 4096

4.156.1.2 #define TPM_EMULATOR_PATH "/usr/bin/vtpmd"

4.156.1.3 #define VTPM_BE_FNAME "/dev/vtpm"

4.156.1.4 #define VTPM_COMMAND_HEADER_SIZE_CLT (2 + 4 + 4)

4.156.1.5 #define VTPM_COMMAND_HEADER_SIZE_SRV (4 + VTPM_COMMAND_HEADER_SIZE_CLT)

4.156.1.6 #define VTPM_DUMMY_RX_BE_FNAME "/var/vtpm/fifos/dummy_in.fifo"

4.156.1.7 #define VTPM_DUMMY_TX_BE_FNAME "/var/vtpm/fifos/dummy_out.fifo"

4.156.1.8 #define VTPM_FAIL 1

4.156.1.9 #define VTPM_FORBIDDEN 3

4.156.1.10 #define VTPM_INVALID_REQUEST 5

4.156.1.11 #define VTPM_ORD_BASE 0x0000

4.156.1.12 #define VTPM_ORD_CLOSE (VTPM_PRIV_BASE + 2)

4.156.1.13 #define VTPM_ORD_DELETE (VTPM_PRIV_BASE + 3)

4.156.1.14 #define VTPM_ORD_GET_MIG_KEY (VTPM_ORD_BASE + 4)

4.156.1.15 #define VTPM_ORD_GETNVMSIZE (VTPM_ORD_BASE + 6)

4.156.1.16 #define VTPM_ORD_LOAD_MIG_KEY (VTPM_ORD_BASE + 5)

4.156.1.17 #define VTPM_ORD_LOADKEY (VTPM_ORD_BASE + 8)

4.156.1.18 #define VTPM_ORD_LOADNVM (VTPM_ORD_BASE + 2)

4.156.1.19 #define VTPM_ORD_MIGRATE_IN (VTPM_PRIV_BASE + 4)

4.156.1.20 #define VTPM_ORD_MIGRATE_OUT (VTPM_PRIV_BASE + 5)

4.156.1.21 #define VTPM_ORD_OPEN (VTPM_PRIV_BASE + 1)

4.156.1.22 #define VTPM_ORD_SAVEKEY (VTPM_ORD_BASE + 7)

4.156.1.23 #define VTPM_ORD_SAVENVM (VTPM_ORD_BASE + 1)

4.156.1.24 #define VTPM_ORD_TPMCOMMAND (VTPM_ORD_BASE + 3)

4.156.1.25 #define VTPM_PRIV_BASE (VTPM_ORD_BASE | VTPM_PRIV_MASK)

4.156.1.26 #define VTPM_PRIV_MASK 0x01000000

4.156.1.27 #define VTPM_RESTORE_CONTEXT_FAILED 4

4.156.1.28 #define VTPM_RX_HP_FNAME "/var/vtpm/fifos/from_console.fifo"

4.156.1.29 #define VTPM_RX_TPM_FNAME "/var/vtpm/fifos/tpm_rsp_from_all.fifo"

4.156.1.30 #define VTPM_RX_VTPM_FNAME "/var/vtpm/fifos/vtpm_cmd_from_all.fifo"

4.156.1.31 #define VTPM_SUCCESS 0

4.156.1.32 #define VTPM_TAG_REQ 0x01c1

4.156.1.33 #define VTPM_TAG_RSP 0x01c4

4.156.1.34 #define VTPM_TX_HP_FNAME "/var/vtpm/fifos/to_console.fifo"

4.156.1.35 #define VTPM_TX_TPM_FNAME "/var/vtpm/fifos/tpm_cmd_to_%d.fifo"

4.156.1.36 #define VTPM_TX_VTPM_FNAME "/var/vtpm/fifos/vtpm_rsp_to_%d.fifo"

4.156.1.37 #define VTPM_TYPE_HVM 2

4.156.1.38 #define VTPM_TYPE_HVM_STRING "hvm"

4.156.1.39 #define VTPM_TYPE_MIGRATABLE 0x01

4.156.1.40 #define VTPM_TYPE_MIGRATED 0xFF

4.156.1.41 #define VTPM_TYPE_NON_MIGRATABLE 0x00

4.156.1.42 #define VTPM_TYPE_PVM 1

4.156.1.43 #define VTPM_TYPE_PVM_STRING "pvm"

4.156.1.44 #define VTPM_UNSUPPORTED 2

4.157 xen/tools/vtpm_manager/manager/vtpm_manager_handler.c File Reference

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <errno.h>
#include <signal.h>
#include "vtpm_manager.h"
#include "vtpmpriv.h"
#include "vtsp.h"
#include "bsg.h"
#include "hashtable.h"
#include "hashtable_itr.h"
#include "log.h"
#include "buffer.h"
```

Macros

- #define vtpmhandlerloginfo(module, fmt, args...) vtpmloginfo (module, "[%s]: " fmt, thread_name, ##args);
- #define vtpmhandlerloginfomore(module, fmt, args...) vtpmloginfomore (module, fmt, ##args);

- #define vtpmhandlerlogerror(module, fmt, args...) vtpmlogerror (module, "[%s]: " fmt, thread_name, ##args);

Functions

- TPM_RESULT vtpm_manager_handle_vtpm_cmd (VTPM_DMI_RESOURCE *dmi_res, TPM_COMMAND_CODE ord, buffer_t *command_buf, buffer_t *result_buf, BOOL is_priv, char *thread_name)
- TPM_RESULT vtpm_manager_handle_tpm_cmd (vtpm_ipc_handle_t *tx_ipc_h, vtpm_ipc_handle_t *rx_ipc_h, VTPM_DMI_RESOURCE *dmi_res, BYTE *cmd_header, buffer_t *param_buf, buffer_t *result_buf, char *thread_name)
- TPM_RESULT VTPM_Manager_Handler (vtpm_ipc_handle_t *tx_ipc_h, vtpm_ipc_handle_t *rx_ipc_h, BOOL fw_tpm, vtpm_ipc_handle_t *fw_tx_ipc_h, vtpm_ipc_handle_t *fw_rx_ipc_h, BOOL is_priv, char *thread_name)

Variables

- volatile sig_atomic_t HANDLER_QUIT_FLAG = 0

4.157.1 Macro Definition Documentation

4.157.1.1 #define vtpmhandlerlogerror(*module, fmt, args...*) vtpmlogerror (module, "[%s]: " fmt, thread_name, ##args);

4.157.1.2 #define vtpmhandlerloginfo(*module, fmt, args...*) vtpmloginfo (module, "[%s]: " fmt, thread_name, ##args);

4.157.1.3 #define vtpmhandlerloginfomore(*module, fmt, args...*) vtpmloginfomore (module, fmt, ##args);

4.157.2 Function Documentation

4.157.2.1 TPM_RESULT vtpm_manager_handle_tpm_cmd (vtpm_ipc_handle_t * *tx_ipc_h,* vtpm_ipc_handle_t * *rx_ipc_h,* VTPM_DMI_RESOURCE * *dmi_res,* BYTE * *cmd_header,* buffer_t * *param_buf,* buffer_t * *result_buf,* char * *thread_name*)

4.157.2.2 TPM_RESULT vtpm_manager_handle_vtpm_cmd (VTPM_DMI_RESOURCE * *dmi_res,* TPM_COMMAND_CODE *ord,* buffer_t * *command_buf,* buffer_t * *result_buf,* BOOL *is_priv,* char * *thread_name*)

4.157.2.3 TPM_RESULT VTPM_Manager_Handler (vtpm_ipc_handle_t * *tx_ipc_h,* vtpm_ipc_handle_t * *rx_ipc_h,* BOOL *fw_tpm,* vtpm_ipc_handle_t * *fw_tx_ipc_h,* vtpm_ipc_handle_t * *fw_rx_ipc_h,* BOOL *is_priv,* char * *thread_name*)

4.157.3 Variable Documentation

4.157.3.1 volatile sig_atomic_t HANDLER_QUIT_FLAG = 0

4.158 xen/tools/vtpm_manager/manager/vtpmd.c File Reference

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <signal.h>
#include <string.h>
#include <pthread.h>
#include "vtpm_manager.h"
#include "vtpmpriv.h"
#include "tcg.h"
#include "log.h"
#include "vtpm_ipc.h"
```

Data Structures

- struct vtpm_thread_params_s

Functions

- void ∗ vtpm_manager_thread (void ∗arg_void)
- void signal_handler (int signal)
- TPM_RESULT VTPM_New_DMI_Extra (VTPM_DMI_RESOURCE ∗dmi_res, BYTE vm_type, BYTE startup_mode)
- TPM_RESULT VTPM_Close_DMI_Extra (VTPM_DMI_RESOURCE ∗dmi_res)
- int main (int argc, char ∗∗argv)

Variables

- struct sigaction ctl_c_handler

4.158.1 Function Documentation

4.158.1.1 int main (int *argc*, char ∗∗ *argv*)

4.158.1.2 void signal_handler (int *signal*)

4.158.1.3 TPM_RESULT VTPM_Close_DMI_Extra (VTPM_DMI_RESOURCE ∗ *dmi_res*)

4.158.1.4 void∗ vtpm_manager_thread (void ∗ *arg_void*)

4.158.1.5 TPM_RESULT VTPM_New_DMI_Extra (VTPM_DMI_RESOURCE ∗ *dmi_res*, BYTE *vm_type*, BYTE *startup_mode*)

4.158.2 Variable Documentation

4.158.2.1 struct sigaction ctl_c_handler

4.159 xen/tools/vtpm_manager/manager/vtpmpriv.h File Reference

```
#include <signal.h>
#include "vtpm_manager.h"
#include "tcg.h"
#include "tcs.h"
#include "buffer.h"
#include "crypto.h"
#include "vtpm_ipc.h"
```

Data Structures

- struct VTPM_DMI_RESOURCE_T
- struct tdVTPM_MIGKEY_LIST
- struct tdVTPM_GLOBALS

Macros

- #define VTPM_MANAGER_GEN 2
- #define STATE_FILE "/var/vtpm/VTPM"
- #define DMI_NVM_FILE "/var/vtpm/vtpm_dm_%d.data"
- #define VTPM_CTL_DM 0

Typedefs

- typedef struct VTPM_DMI_RESOURCE_T VTPM_DMI_RESOURCE
- typedef struct tdVTPM_MIGKEY_LIST VTPM_MIGKEY_LIST
- typedef struct tdVTPM_GLOBALS VTPM_GLOBALS

Functions

- TPM_RESULT VTPM_Init_Manager ()
- void VTPM_Stop_Manager ()
- TPM_RESULT VTPM_Manager_Handler (vtpm_ipc_handle_t *tx_ipc_h, vtpm_ipc_handle_t *rx_ipc_h, BOOL fw_tpm, vtpm_ipc_handle_t *fw_tx_ipc_h, vtpm_ipc_handle_t *fw_rx_ipc_h, BOOL is_priv, char *client_name)
- TPM_RESULT VTPM_Handle_Load_NVM (VTPM_DMI_RESOURCE *myDMI, const buffer_t *inbuf, buffer_t *outbuf)
- TPM_RESULT VTPM_Handle_Save_NVM (VTPM_DMI_RESOURCE *myDMI, const buffer_t *inbuf, buffer_t *outbuf)
- TPM_RESULT VTPM_Handle_Get_NVM_Size (VTPM_DMI_RESOURCE *myDMI, const buffer_t *inbuf, buffer_t *outbuf)
- TPM_RESULT VTPM_Handle_TPM_Command (VTPM_DMI_RESOURCE *dmi, buffer_t *inbuf, buffer_t *outbuf)
- TPM_RESULT VTPM_Handle_New_DMI (const buffer_t *param_buf)
- TPM_RESULT VTPM_Handle_Close_DMI (const buffer_t *param_buf)
- TPM_RESULT VTPM_Handle_Delete_DMI (const buffer_t *param_buf)
- TPM_RESULT VTPM_Handle_Migrate_In (const buffer_t *param_buf, buffer_t *result_buf)
- TPM_RESULT VTPM_Handle_Migrate_Out (const buffer_t *param_buf, buffer_t *result_buf)
- TPM_RESULT VTPM_Handle_Get_Migration_key (const buffer_t *param_buf, buffer_t *result_buf)
- TPM_RESULT VTPM_SaveManagerData (void)
- TPM_RESULT VTPM_LoadManagerData (void)

- TPM_RESULT VTPM_New_DMI_Extra (VTPM_DMI_RESOURCE *dmi_res, BYTE vm_type, BYTE startup_mode)
- TPM_RESULT VTPM_Close_DMI_Extra (VTPM_DMI_RESOURCE *dmi_res)
- TPM_RESULT close_dmi (VTPM_DMI_RESOURCE *dmi_res)
- TPM_RESULT init_dmi (UINT32 dmi_id, BYTE type, VTPM_DMI_RESOURCE **dmi_res)
- void free_dmi (VTPM_DMI_RESOURCE *dmi_res)
- TPM_RESULT envelope_encrypt (const buffer_t *inbuf, CRYPTO_INFO *asymkey, buffer_t *sealed_data)
- TPM_RESULT envelope_decrypt (const buffer_t *cipher, TCS_CONTEXT_HANDLE TCSContext, TPM_HANDLE keyHandle, const TPM_AUTHDATA *key_usage_auth, buffer_t *unsealed_data)

Variables

- volatile sig_atomic_t HANDLER_QUIT_FLAG
- VTPM_GLOBALS * vtpm_globals
- const TPM_AUTHDATA SRK_AUTH

4.159.1 Macro Definition Documentation

4.159.1.1 #define DMI_NVM_FILE "/var/vtpm/vtpm_dm_%d.data"

4.159.1.2 #define STATE_FILE "/var/vtpm/VTPM"

4.159.1.3 #define VTPM_CTL_DM 0

4.159.1.4 #define VTPM_MANAGER_GEN 2

4.159.2 Typedef Documentation

4.159.2.1 typedef struct VTPM_DMI_RESOURCE_T VTPM_DMI_RESOURCE

4.159.2.2 typedef struct tdVTPM_GLOBALS VTPM_GLOBALS

4.159.2.3 typedef struct tdVTPM_MIGKEY_LIST VTPM_MIGKEY_LIST

4.159.3 Function Documentation

4.159.3.1 TPM_RESULT close_dmi (VTPM_DMI_RESOURCE * *dmi_res*)

4.159.3.2 TPM_RESULT envelope_decrypt (const buffer_t * *cipher*, TCS_CONTEXT_HANDLE *TCSContext*, TPM_HANDLE *keyHandle*, const TPM_AUTHDATA * *key_usage_auth*, buffer_t * *unsealed_data*)

4.159.3.3 TPM_RESULT envelope_encrypt (const buffer_t * *inbuf*, CRYPTO_INFO * *asymkey*, buffer_t * *sealed_data*)

4.159.3.4 void free_dmi (VTPM_DMI_RESOURCE * *dmi_res*)

4.159.3.5 TPM_RESULT init_dmi (UINT32 *dmi_id*, BYTE *type*, VTPM_DMI_RESOURCE ** *dmi_res*)

4.159.3.6 TPM_RESULT VTPM_Close_DMI_Extra (VTPM_DMI_RESOURCE * *dmi_res*)

4.159.3.7 TPM_RESULT VTPM_Handle_Close_DMI (const buffer_t * *param_buf*)

4.159.3.8 TPM_RESULT VTPM_Handle_Delete_DMI (const buffer_t * *param_buf*)

4.159.3.9 TPM_RESULT VTPM_Handle_Get_Migration_key (const buffer_t * *param_buf,* buffer_t * *result_buf*)

4.159.3.10 TPM_RESULT VTPM_Handle_Get_NVM_Size (VTPM_DMI_RESOURCE * *myDMI,* const buffer_t * *inbuf,* buffer_t * *outbuf*)

4.159.3.11 TPM_RESULT VTPM_Handle_Load_NVM (VTPM_DMI_RESOURCE * *myDMI,* const buffer_t * *inbuf,* buffer_t * *outbuf*)

4.159.3.12 TPM_RESULT VTPM_Handle_Migrate_In (const buffer_t * *param_buf,* buffer_t * *result_buf*)

4.159.3.13 TPM_RESULT VTPM_Handle_Migrate_Out (const buffer_t * *param_buf,* buffer_t * *result_buf*)

4.159.3.14 TPM_RESULT VTPM_Handle_New_DMI (const buffer_t * *param_buf*)

4.159.3.15 TPM_RESULT VTPM_Handle_Save_NVM (VTPM_DMI_RESOURCE * *myDMI,* const buffer_t * *inbuf,* buffer_t * *outbuf*)

4.159.3.16 TPM_RESULT VTPM_Handle_TPM_Command (VTPM_DMI_RESOURCE * *dmi,* buffer_t * *inbuf,* buffer_t * *outbuf*)

4.159.3.17 TPM_RESULT VTPM_Init_Manager ( )

4.159.3.18 TPM_RESULT VTPM_LoadManagerData (void)

4.159.3.19 TPM_RESULT VTPM_Manager_Handler (vtpm_ipc_handle_t * *tx_ipc_h,* vtpm_ipc_handle_t * *rx_ipc_h,* BOOL *fw_tpm,* vtpm_ipc_handle_t * *fw_tx_ipc_h,* vtpm_ipc_handle_t * *fw_rx_ipc_h,* BOOL *is_priv,* char * *client_name*)

4.159.3.20 TPM_RESULT VTPM_New_DMI_Extra (VTPM_DMI_RESOURCE * *dmi_res,* BYTE *vm_type,* BYTE *startup_mode*)

4.159.3.21 TPM_RESULT VTPM_SaveManagerData (void)

4.159.3.22 void VTPM_Stop_Manager ( )

4.159.4 Variable Documentation

4.159.4.1 volatile sig_atomic_t HANDLER_QUIT_FLAG

4.159.4.2 const TPM_AUTHDATA SRK_AUTH

4.159.4.3 VTPM_GLOBALS* vtpm_globals

4.160 xen/tools/vtpm_manager/manager/vtsp.c File Reference

```
#include <string.h>
#include "tcg.h"
#include "tcs.h"
#include "bsg.h"
#include "log.h"
#include "crypto.h"
#include "vtsp.h"
#include "buffer.h"
```

Macros

- #define RSA_KEY_SIZE 0x0800

Functions

- TPM_RESULT GenerateAuth (const BYTE *inParamDigestText, UINT32 inParamDigestTextSize, const TPM_SECRET *HMACkey, TCS_AUTH *auth)
- TPM_RESULT VerifyAuth (const BYTE *outParamDigestText, UINT32 outParamDigestTextSize, const TPM_SECRET *HMACkey, TCS_AUTH *auth, TCS_CONTEXT_HANDLE hContext)
- TPM_RESULT VTSP_OIAP (const TCS_CONTEXT_HANDLE hContext, TCS_AUTH *auth)
- TPM_RESULT VTSP_OSAP (const TCS_CONTEXT_HANDLE hContext, const TPM_ENTITY_TYPE entityType, const UINT32 entityValue, const TPM_AUTHDATA *usageAuth, TPM_SECRET *sharedSecret, TCS_AUTH *auth)
- TPM_RESULT VTSP_TerminateHandle (const TCS_CONTEXT_HANDLE hContext, const TCS_AUTH *auth)
- TPM_RESULT VTSP_ReadPubek (const TCS_CONTEXT_HANDLE hContext, CRYPTO_INFO *crypto_info)
- TPM_RESULT VTSP_TakeOwnership (const TCS_CONTEXT_HANDLE hContext, const TPM_AUTHDATA *ownerAuth, const TPM_AUTHDATA *srkAuth, CRYPTO_INFO *ek_cryptoInfo, TCS_AUTH *auth)
- TPM_RESULT VTSP_DisablePubekRead (const TCS_CONTEXT_HANDLE hContext, const TPM_AUTHDATA *ownerAuth, TCS_AUTH *auth)
- TPM_RESULT VTSP_CreateWrapKey (const TCS_CONTEXT_HANDLE hContext, const TPM_KEY_USAGE usage, const TPM_AUTHDATA *newKeyAuth, const TCS_KEY_HANDLE parentHandle, const TPM_AUTHDATA *osapSharedSecret, buffer_t *pubKeyBuf, TCS_AUTH *auth)
- TPM_RESULT VTSP_LoadKey (const TCS_CONTEXT_HANDLE hContext, const TCS_KEY_HANDLE hUnwrappingKey, const buffer_t *rgbWrappedKeyBlob, const TPM_AUTHDATA *parentAuth, TPM_HANDLE *newKeyHandle, TCS_AUTH *auth, CRYPTO_INFO *cryptoinfo, const BOOL skipTPMLoad)
- TPM_RESULT VTSP_Unbind (const TCS_CONTEXT_HANDLE hContext, const TPM_KEY_HANDLE key_handle, const buffer_t *bound_data, const TPM_AUTHDATA *usage_auth, buffer_t *clear_data, TCS_AUTH *auth)
- TPM_RESULT VTSP_Bind (CRYPTO_INFO *cryptoInfo, const buffer_t *inData, buffer_t *outData)
- TPM_RESULT VTSP_Seal (const TCS_CONTEXT_HANDLE hContext, const TPM_KEY_HANDLE keyHandle, const TPM_AUTHDATA *sealDataAuth, const TPM_PCR_COMPOSITE *pcrComp, const buffer_t *inData, TPM_STORED_DATA *sealedData, const TPM_SECRET *osapSharedSecret, TCS_AUTH *auth)
- TPM_RESULT VTSP_Unseal (const TCS_CONTEXT_HANDLE hContext, const TPM_KEY_HANDLE keyHandle, const TPM_STORED_DATA *sealedData, const TPM_AUTHDATA *key_usage_auth, const TPM_AUTHDATA *data_usage_auth, buffer_t *outData, TCS_AUTH *auth, TCS_AUTH *dataAuth)
- TPM_RESULT VTSP_SaveState (const TCS_CONTEXT_HANDLE hContext)
- TPM_RESULT VTSP_RawTransmit (const TCS_CONTEXT_HANDLE hContext, const buffer_t *inbuf, buffer_t *outbuf)

4.160.1 Macro Definition Documentation

4.160.1.1 #define RSA_KEY_SIZE 0x0800

4.160.2 Function Documentation

4.160.2.1 TPM_RESULT GenerateAuth (const BYTE * *inParamDigestText,* UINT32 *inParamDigestTextSize,* const TPM_SECRET * *HMACkey,* TCS_AUTH * *auth*)

4.160.2.2 TPM_RESULT VerifyAuth (const BYTE * *outParamDigestText,* UINT32 *outParamDigestTextSize,* const TPM_SECRET * *HMACkey,* TCS_AUTH * *auth,* TCS_CONTEXT_HANDLE *hContext*)

4.160.2.3 TPM_RESULT VTSP_Bind (CRYPTO_INFO * *cryptoInfo,* const buffer_t * *inData,* buffer_t * *outData*)

4.160.2.4 **TPM_RESULT** VTSP_CreateWrapKey (const **TCS_CONTEXT_HANDLE** *hContext,* const **TPM_KEY_USAGE** *usage,* const **TPM_AUTHDATA** ∗ *newKeyAuth,* const **TCS_KEY_HANDLE** *parentHandle,* const **TPM_AUTHDATA** ∗ *osapSharedSecret,* **buffer_t** ∗ *pubKeyBuf,* **TCS_AUTH** ∗ *auth*)

4.160.2.5 **TPM_RESULT** VTSP_DisablePubekRead (const **TCS_CONTEXT_HANDLE** *hContext,* const **TPM_AUTHDATA** ∗ *ownerAuth,* **TCS_AUTH** ∗ *auth*)

4.160.2.6 **TPM_RESULT** VTSP_LoadKey (const **TCS_CONTEXT_HANDLE** *hContext,* const **TCS_KEY_HANDLE** *hUnwrappingKey,* const **buffer_t** ∗ *rgbWrappedKeyBlob,* const **TPM_AUTHDATA** ∗ *parentAuth,* **TPM_HANDLE** ∗ *newKeyHandle,* **TCS_AUTH** ∗ *auth,* **CRYPTO_INFO** ∗ *cryptoinfo,* const **BOOL** *skipTPMLoad*)

4.160.2.7 **TPM_RESULT** VTSP_OIAP (const **TCS_CONTEXT_HANDLE** *hContext,* **TCS_AUTH** ∗ *auth*)

4.160.2.8 **TPM_RESULT** VTSP_OSAP (const **TCS_CONTEXT_HANDLE** *hContext,* const **TPM_ENTITY_TYPE** *entityType,* const **UINT32** *entityValue,* const **TPM_AUTHDATA** ∗ *usageAuth,* **TPM_SECRET** ∗ *sharedSecret,* **TCS_AUTH** ∗ *auth*)

4.160.2.9 **TPM_RESULT** VTSP_RawTransmit (const **TCS_CONTEXT_HANDLE** *hContext,* const **buffer_t** ∗ *inbuf,* **buffer_t** ∗ *outbuf*)

4.160.2.10 **TPM_RESULT** VTSP_ReadPubek (const **TCS_CONTEXT_HANDLE** *hContext,* **CRYPTO_INFO** ∗ *crypto_info*)

4.160.2.11 **TPM_RESULT** VTSP_SaveState (const **TCS_CONTEXT_HANDLE** *hContext*)

4.160.2.12 **TPM_RESULT** VTSP_Seal (const **TCS_CONTEXT_HANDLE** *hContext,* const **TPM_KEY_HANDLE** *keyHandle,* const **TPM_AUTHDATA** ∗ *sealDataAuth,* const **TPM_PCR_COMPOSITE** ∗ *pcrComp,* const **buffer_t** ∗ *inData,* **TPM_STORED_DATA** ∗ *sealedData,* const **TPM_SECRET** ∗ *osapSharedSecret,* **TCS_AUTH** ∗ *auth*)

4.160.2.13 **TPM_RESULT** VTSP_TakeOwnership (const **TCS_CONTEXT_HANDLE** *hContext,* const **TPM_AUTHDATA** ∗ *ownerAuth,* const **TPM_AUTHDATA** ∗ *srkAuth,* **CRYPTO_INFO** ∗ *ek_cryptoInfo,* **TCS_AUTH** ∗ *auth*)

4.160.2.14 **TPM_RESULT** VTSP_TerminateHandle (const **TCS_CONTEXT_HANDLE** *hContext,* const **TCS_AUTH** ∗ *auth*)

4.160.2.15 **TPM_RESULT** VTSP_Unbind (const **TCS_CONTEXT_HANDLE** *hContext,* const **TPM_KEY_HANDLE** *key_handle,* const **buffer_t** ∗ *bound_data,* const **TPM_AUTHDATA** ∗ *usage_auth,* **buffer_t** ∗ *clear_data,* **TCS_AUTH** ∗ *auth*)

4.160.2.16 **TPM_RESULT** VTSP_Unseal (const **TCS_CONTEXT_HANDLE** *hContext,* const **TPM_KEY_HANDLE** *keyHandle,* const **TPM_STORED_DATA** ∗ *sealedData,* const **TPM_AUTHDATA** ∗ *key_usage_auth,* const **TPM_AUTHDATA** ∗ *data_usage_auth,* **buffer_t** ∗ *outData,* **TCS_AUTH** ∗ *auth,* **TCS_AUTH** ∗ *dataAuth*)

4.161 xen/tools/vtpm_manager/migration/vtpm_manager_if.c File Reference

```
#include <stdio.h>
#include <fcntl.h>
#include <malloc.h>
#include <string.h>
#include "tcg.h"
#include "buffer.h"
#include "log.h"
#include "vtpm_ipc.h"
#include "bsg.h"
#include "vtpm_migrator.h"
#include "vtpm_manager.h"
```

Functions

- TPM_RESULT vtpm_manager_open ()
- void vtpm_manager_close ()
- TPM_RESULT vtpm_manager_command (TPM_COMMAND_CODE ord, buffer_t *command_param_buf, TPM_RESULT *cmd_status, buffer_t *result_param_buf)

4.161.1 Function Documentation

4.161.1.1 void vtpm_manager_close ( )

4.161.1.2 TPM_RESULT vtpm_manager_command (TPM_COMMAND_CODE *ord*, buffer_t * *command_param_buf*, TPM_RESULT * *cmd_status*, buffer_t * *result_param_buf*)

4.161.1.3 TPM_RESULT vtpm_manager_open ( )

4.162 xen/tools/vtpm_manager/tcs/contextmgr.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <malloc.h>
#include "tcs.h"
#include "contextmgr.h"
#include "log.h"
#include "hashtable.h"
```

Functions

- BYTE * AddMemBlock (CONTEXT_HANDLE *pContextHandle, int BlockSize)
- BOOL DeleteMemBlock (CONTEXT_HANDLE *pContextHandle, BYTE *pTCPA_BYTEs)
- BOOL AddHandleToList (TCS_CONTEXT_HANDLE hContext, TPM_RESOURCE_TYPE type, TPM_HANDLE handle)
- BOOL DeleteHandleFromList (TCS_CONTEXT_HANDLE hContext, TPM_HANDLE handle)
- BOOL FreeHandleList (CONTEXT_HANDLE *pContextHandle)

4.162.1 Function Documentation

4.162.1.1 BOOL AddHandleToList (TCS_CONTEXT_HANDLE *hContext*, TPM_RESOURCE_TYPE *type*, TPM_HANDLE *handle*)

4.162.1.2 BYTE* AddMemBlock (CONTEXT_HANDLE * *pContextHandle*, int *BlockSize*)

4.162.1.3 BOOL DeleteHandleFromList (TCS_CONTEXT_HANDLE *hContext*, TPM_HANDLE *handle*)

4.162.1.4 BOOL DeleteMemBlock (CONTEXT_HANDLE * *pContextHandle*, BYTE * *pTCPA_BYTEs*)

4.162.1.5 BOOL FreeHandleList (CONTEXT_HANDLE * *pContextHandle*)

4.163 xen/tools/vtpm_manager/tcs/tcs.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <malloc.h>
#include "tcg.h"
#include "bsg.h"
#include "tcs.h"
#include "contextmgr.h"
#include "tpmddl.h"
#include "log.h"
#include "hashtable.h"
#include "hashtable_itr.h"
```

Macros

- #define TCPA_MAX_BUFFER_LENGTH 0x2000

Functions

- CONTEXT_HANDLE * LookupContext (TCS_CONTEXT_HANDLE hContext)
- TPM_RESULT TCS_create ()
- void TCS_destroy ()
- TPM_RESULT TCS_Malloc (TCS_CONTEXT_HANDLE hContext, UINT32 MemSize, BYTE **ppMemPtr)
- TPM_RESULT TCS_FreeMemory (TCS_CONTEXT_HANDLE hContext, BYTE *pMemory)
- TPM_RESULT TCS_OpenContext (TCS_CONTEXT_HANDLE *hContext)
- TPM_RESULT TCS_CloseContext (TCS_CONTEXT_HANDLE hContext)
- int packAuth (BYTE *dst, TCS_AUTH *auth)
- int unpackAuth (TCS_AUTH *auth, BYTE *src)
- TPM_RESULT TCSP_OIAP (TCS_CONTEXT_HANDLE hContext, TCS_AUTHHANDLE *authHandle, TPM_NONCE *nonce0)
- TPM_RESULT TCSP_OSAP (TCS_CONTEXT_HANDLE hContext, TPM_ENTITY_TYPE entityType, UINT32 entityValue, TPM_NONCE nonceOddOSAP, TCS_AUTHHANDLE *authHandle, TPM_NONCE *nonceEven, TPM_NONCE *nonceEvenOSAP)
- TPM_RESULT TCSP_TakeOwnership (TCS_CONTEXT_HANDLE hContext, UINT16 protocolID, UINT32 encOwnerAuthSize, BYTE *encOwnerAuth, UINT32 encSrkAuthSize, BYTE *encSrkAuth, UINT32 *SrkSize, BYTE **Srk, TCS_AUTH *ownerAuth)
- TPM_RESULT TCSP_DisablePubekRead (TCS_CONTEXT_HANDLE hContext, TCS_AUTH *ownerAuth)
- TPM_RESULT TCSP_TerminateHandle (TCS_CONTEXT_HANDLE hContext, TCS_AUTHHANDLE handle)
- TPM_RESULT TCSP_Extend (TCS_CONTEXT_HANDLE hContext, TPM_PCRINDEX pcrNum, TPM_DIGEST inDigest, TPM_PCRVALUE *outDigest)

- TPM_RESULT TCSP_Seal (TCS_CONTEXT_HANDLE hContext, TCS_KEY_HANDLE keyHandle, TPM_-ENCAUTH encAuth, UINT32 pcrInfoSize, BYTE *PcrInfo, UINT32 inDataSize, BYTE *inData, TCS_AUTH *pubAuth, UINT32 *SealedDataSize, BYTE **SealedData)
- TPM_RESULT TCSP_Unseal (TCS_CONTEXT_HANDLE hContext, TCS_KEY_HANDLE parentHandle, U-INT32 SealedDataSize, BYTE *SealedData, TCS_AUTH *parentAuth, TCS_AUTH *dataAuth, UINT32 *DataSize, BYTE **Data)
- TPM_RESULT TCSP_UnBind (TCS_CONTEXT_HANDLE hContext, TCS_KEY_HANDLE keyHandle, UIN-T32 inDataSize, BYTE *inData, TCS_AUTH *privAuth, UINT32 *outDataSize, BYTE **outData)
- TPM_RESULT TCSP_CreateWrapKey (TCS_CONTEXT_HANDLE hContext, TCS_KEY_HANDLE h-WrappingKey, TPM_ENCAUTH KeyUsageAuth, TPM_ENCAUTH KeyMigrationAuth, UINT32 *pcKeySize, BYTE **prgbKey, TCS_AUTH *pAuth)
- TPM_RESULT TCSP_LoadKeyByBlob (TCS_CONTEXT_HANDLE hContext, TCS_KEY_HANDLE h-UnwrappingKey, UINT32 cWrappedKeyBlobSize, BYTE *rgbWrappedKeyBlob, TCS_AUTH *pAuth, TC-S_KEY_HANDLE *phKeyTCSI, TCS_KEY_HANDLE *phKeyHMAC)
- TPM_RESULT TCSP_EvictKey (TCS_CONTEXT_HANDLE hContext, TCS_KEY_HANDLE hKey)
- TPM_RESULT TCSP_GetRandom (TCS_CONTEXT_HANDLE hContext, UINT32 *bytesRequested, BYTE **randomBytes)
- TPM_RESULT TCSP_ReadPubek (TCS_CONTEXT_HANDLE hContext, TPM_NONCE antiReplay, UINT32 *pubEndorsementKeySize, BYTE **pubEndorsementKey, TPM_DIGEST *checksum)
- TPM_RESULT TCSP_SaveState (TCS_CONTEXT_HANDLE hContext)
- TPM_RESULT TCSP_RawTransmitData (UINT32 inDataSize, BYTE *inData, UINT32 *outDataSize, BYTE *outData)

Variables

- struct hashtable * context_ht

4.163.1 Macro Definition Documentation

4.163.1.1 #define TCPA_MAX_BUFFER_LENGTH 0x2000

4.163.2 Function Documentation

4.163.2.1 CONTEXT_HANDLE* LookupContext (TCS_CONTEXT_HANDLE *hContext*)

4.163.2.2 int packAuth (BYTE * *dst*, TCS_AUTH * *auth*)

4.163.2.3 TPM_RESULT TCS_CloseContext (TCS_CONTEXT_HANDLE *hContext*)

4.163.2.4 TPM_RESULT TCS_create ( )

4.163.2.5 void TCS_destroy ( )

4.163.2.6 TPM_RESULT TCS_FreeMemory (TCS_CONTEXT_HANDLE *hContext*, BYTE * *pMemory*)

4.163.2.7 TPM_RESULT TCS_Malloc (TCS_CONTEXT_HANDLE *hContext*, UINT32 *MemSize*, BYTE ** *ppMemPtr*)

4.163.2.8 TPM_RESULT TCS_OpenContext (TCS_CONTEXT_HANDLE * *hContext*)

4.163.2.9 TPM_RESULT TCSP_CreateWrapKey (TCS_CONTEXT_HANDLE *hContext*, TCS_KEY_HANDLE *hWrappingKey*, TPM_ENCAUTH *KeyUsageAuth*, TPM_ENCAUTH *KeyMigrationAuth*, UINT32 * *pcKeySize*, BYTE ** *prgbKey*, TCS_AUTH * *pAuth*)

4.163.2.10 TPM_RESULT TCSP_DisablePubekRead (TCS_CONTEXT_HANDLE *hContext*, TCS_AUTH * *ownerAuth*)

4.163.2.11 **TPM_RESULT** TCSP_EvictKey (**TCS_CONTEXT_HANDLE** *hContext,* **TCS_KEY_HANDLE** *hKey*)

4.163.2.12 **TPM_RESULT** TCSP_Extend (**TCS_CONTEXT_HANDLE** *hContext,* **TPM_PCRINDEX** *pcrNum,* **TPM_DIGEST** *inDigest,* **TPM_PCRVALUE** ∗ *outDigest*)

4.163.2.13 **TPM_RESULT** TCSP_GetRandom (**TCS_CONTEXT_HANDLE** *hContext,* **UINT32** ∗ *bytesRequested,* **BYTE** ∗∗ *randomBytes*)

4.163.2.14 **TPM_RESULT** TCSP_LoadKeyByBlob (**TCS_CONTEXT_HANDLE** *hContext,* **TCS_KEY_HANDLE** *hUnwrappingKey,* **UINT32** *cWrappedKeyBlobSize,* **BYTE** ∗ *rgbWrappedKeyBlob,* **TCS_AUTH** ∗ *pAuth,* **TCS_KEY_HANDLE** ∗ *phKeyTCSI,* **TCS_KEY_HANDLE** ∗ *phKeyHMAC*)

4.163.2.15 **TPM_RESULT** TCSP_OIAP (**TCS_CONTEXT_HANDLE** *hContext,* **TCS_AUTHHANDLE** ∗ *authHandle,* **TPM_NONCE** ∗ *nonce0*)

4.163.2.16 **TPM_RESULT** TCSP_OSAP (**TCS_CONTEXT_HANDLE** *hContext,* **TPM_ENTITY_TYPE** *entityType,* **UINT32** *entityValue,* **TPM_NONCE** *nonceOddOSAP,* **TCS_AUTHHANDLE** ∗ *authHandle,* **TPM_NONCE** ∗ *nonceEven,* **TPM_NONCE** ∗ *nonceEvenOSAP*)

4.163.2.17 **TPM_RESULT** TCSP_RawTransmitData (**UINT32** *inDataSize,* **BYTE** ∗ *inData,* **UINT32** ∗ *outDataSize,* **BYTE** ∗ *outData*)

4.163.2.18 **TPM_RESULT** TCSP_ReadPubek (**TCS_CONTEXT_HANDLE** *hContext,* **TPM_NONCE** *antiReplay,* **UINT32** ∗ *pubEndorsementKeySize,* **BYTE** ∗∗ *pubEndorsementKey,* **TPM_DIGEST** ∗ *checksum*)

4.163.2.19 **TPM_RESULT** TCSP_SaveState (**TCS_CONTEXT_HANDLE** *hContext*)

4.163.2.20 **TPM_RESULT** TCSP_Seal (**TCS_CONTEXT_HANDLE** *hContext,* **TCS_KEY_HANDLE** *keyHandle,* **TPM_ENCAUTH** *encAuth,* **UINT32** *pcrInfoSize,* **BYTE** ∗ *PcrInfo,* **UINT32** *inDataSize,* **BYTE** ∗ *inData,* **TCS_AUTH** ∗ *pubAuth,* **UINT32** ∗ *SealedDataSize,* **BYTE** ∗∗ *SealedData*)

4.163.2.21 **TPM_RESULT** TCSP_TakeOwnership (**TCS_CONTEXT_HANDLE** *hContext,* **UINT16** *protocolID,* **UINT32** *encOwnerAuthSize,* **BYTE** ∗ *encOwnerAuth,* **UINT32** *encSrkAuthSize,* **BYTE** ∗ *encSrkAuth,* **UINT32** ∗ *SrkSize,* **BYTE** ∗∗ *Srk,* **TCS_AUTH** ∗ *ownerAuth*)

4.163.2.22 **TPM_RESULT** TCSP_TerminateHandle (**TCS_CONTEXT_HANDLE** *hContext,* **TCS_AUTHHANDLE** *handle*)

4.163.2.23 **TPM_RESULT** TCSP_UnBind (**TCS_CONTEXT_HANDLE** *hContext,* **TCS_KEY_HANDLE** *keyHandle,* **UINT32** *inDataSize,* **BYTE** ∗ *inData,* **TCS_AUTH** ∗ *privAuth,* **UINT32** ∗ *outDataSize,* **BYTE** ∗∗ *outData*)

4.163.2.24 **TPM_RESULT** TCSP_Unseal (**TCS_CONTEXT_HANDLE** *hContext,* **TCS_KEY_HANDLE** *parentHandle,* **UINT32** *SealedDataSize,* **BYTE** ∗ *SealedData,* **TCS_AUTH** ∗ *parentAuth,* **TCS_AUTH** ∗ *dataAuth,* **UINT32** ∗ *DataSize,* **BYTE** ∗∗ *Data*)

4.163.2.25 int unpackAuth (**TCS_AUTH** ∗ *auth,* **BYTE** ∗ *src*)

4.163.3 Variable Documentation

4.163.3.1 struct hashtable∗ context_ht

4.164 xen/tools/vtpm_manager/tcs/tpmddl.c File Reference

```
#include <string.h>
#include "tpmddl.h"
#include "tcs.h"
#include "bsg.h"
#include "log.h"
```

Macros

- #define MIN(X, Y) ((X) < (Y) ? (X) : (Y))

Functions

- TDDL_RESULT TDDL_GetCapability (TDDL_UINT32 cap, TDDL_UINT32 sub, TDDL_BYTE *buffer, TDDL_UINT32 *size)
- TDDL_RESULT TDDL_FlushSpecific (TDDL_UINT32 handle, TDDL_UINT32 res)

4.164.1 Macro Definition Documentation

4.164.1.1 #define MIN(*X, Y*) ((X) < (Y) ? (X) : (Y))

4.164.2 Function Documentation

4.164.2.1 TDDL_RESULT TDDL_FlushSpecific (TDDL_UINT32 *handle,* TDDL_UINT32 *res*)

4.164.2.2 TDDL_RESULT TDDL_GetCapability (TDDL_UINT32 *cap,* TDDL_UINT32 *sub,* TDDL_BYTE * *buffer,* TDDL_UINT32 * *size*)

4.165 xen/tools/vtpm_manager/tcs/tpmddl.h File Reference

Macros

- #define TDDL_CAP_PROP_MANUFACTURER 0x0001
- #define TDDL_E_FAIL 1
- #define TDDL_E_SUCCESS 0
- #define TDDL_SUCCESS 0

Typedefs

- typedef unsigned int TDDL_UINT32
- typedef TDDL_UINT32 TDDL_RESULT
- typedef unsigned char TDDL_BYTE

Functions

- TDDL_RESULT TDDL_Open ()
- void TDDL_Close ()
- TDDL_RESULT TDDL_TransmitData (TDDL_BYTE *in, TDDL_UINT32 insize, TDDL_BYTE *out, TDDL_UINT32 *outsize)
- TDDL_RESULT TDDL_GetStatus ()

- TDDL_RESULT TDDL_GetCapability (TDDL_UINT32 cap, TDDL_UINT32 sub, TDDL_BYTE *buffer, TDDL_UINT32 *size)
- TDDL_RESULT TDDL_SetCapability (TDDL_UINT32 cap, TDDL_UINT32 sub, TDDL_BYTE *buffer, TDDL_UINT32 *size)
- TDDL_RESULT TDDL_FlushSpecific (TDDL_UINT32 handle, TDDL_UINT32 res)

4.165.1 Macro Definition Documentation

4.165.1.1 #define TDDL_CAP_PROP_MANUFACTURER 0x0001

4.165.1.2 #define TDDL_E_FAIL 1

4.165.1.3 #define TDDL_E_SUCCESS 0

4.165.1.4 #define TDDL_SUCCESS 0

4.165.2 Typedef Documentation

4.165.2.1 typedef unsigned char TDDL_BYTE

4.165.2.2 typedef TDDL_UINT32 TDDL_RESULT

4.165.2.3 typedef unsigned int TDDL_UINT32

4.165.3 Function Documentation

4.165.3.1 void TDDL_Close ( )

4.165.3.2 TDDL_RESULT TDDL_FlushSpecific (TDDL_UINT32 *handle*, TDDL_UINT32 *res*)

4.165.3.3 TDDL_RESULT TDDL_GetCapability (TDDL_UINT32 *cap*, TDDL_UINT32 *sub*, TDDL_BYTE * *buffer*, TDDL_UINT32 * *size*)

4.165.3.4 TDDL_RESULT TDDL_GetStatus ( )

4.165.3.5 TDDL_RESULT TDDL_Open ( )

4.165.3.6 TDDL_RESULT TDDL_SetCapability (TDDL_UINT32 *cap*, TDDL_UINT32 *sub*, TDDL_BYTE * *buffer*, TDDL_UINT32 * *size*)

4.165.3.7 TDDL_RESULT TDDL_TransmitData (TDDL_BYTE * *in*, TDDL_UINT32 *insize*, TDDL_BYTE * *out*, TDDL_UINT32 * *outsize*)

4.166 xen/tools/vtpm_manager/tcs/transmit.c File Reference

```
#include <unistd.h>
#include <stdio.h>
#include <errno.h>
#include <fcntl.h>
#include "tcg.h"
#include "buffer.h"
#include "log.h"
#include "tpmddl.h"
```

Macros

- #define TPM_TX_FNAME "/dev/tpm0"

Functions

- TPM_RESULT TDDL_TransmitData (TDDL_BYTE *in, TDDL_UINT32 insize, TDDL_BYTE *out, TDDL_UINT32 *outsize)
- TPM_RESULT TDDL_Open ()
- void TDDL_Close ()

4.166.1 Macro Definition Documentation

4.166.1.1 #define TPM_TX_FNAME "/dev/tpm0"

4.166.2 Function Documentation

4.166.2.1 void TDDL_Close ( )

4.166.2.2 TPM_RESULT TDDL_Open ( )

4.166.2.3 TPM_RESULT TDDL_TransmitData (TDDL_BYTE * *in*, TDDL_UINT32 *insize*, TDDL_BYTE * *out*, TDDL_UINT32 * *outsize*)

4.167 xen/tools/vtpm_manager/util/buffer.c File Reference

```
#include <stdarg.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <sys/param.h>
#include "tcg.h"
#include "log.h"
#include "bsg.h"
#include "buffer.h"
```

Functions

- TPM_RESULT buffer_init (buffer_t *buf, tpm_size_t initsize, const BYTE *initval)
- TPM_RESULT buffer_init_convert (buffer_t *buf, tpm_size_t initsize, BYTE *initval)
- TPM_RESULT buffer_init_copy (buffer_t *buf, const buffer_t *src)
- TPM_RESULT buffer_init_const (buffer_t *buf, tpm_size_t size, const BYTE *val)
- TPM_RESULT buffer_init_alias (buffer_t *buf, const buffer_t *b, tpm_size_t offset, tpm_size_t len)
- TPM_RESULT buffer_init_alias_convert (buffer_t *buf, tpm_size_t size, BYTE *val)
- TPM_RESULT buffer_copy (buffer_t *dest, const buffer_t *src)
- BOOL buffer_eq (const buffer_t *a, const buffer_t *b)
- void buffer_memset (buffer_t *buf, BYTE b)
- TPM_RESULT buffer_append_raw (buffer_t *buf, tpm_size_t len, const BYTE *bytes)
- TPM_RESULT buffer_prepend_raw (buffer_t *buf, tpm_size_t len, const BYTE *bytes)
- tpm_size_t buffer_len (const buffer_t *buf)
- TPM_RESULT buffer_free (buffer_t *buf)

4.167.1 Function Documentation

4.167.1.1 TPM_RESULT buffer_append_raw (buffer_t * *buf*, tpm_size_t *len*, const BYTE * *bytes*)

4.167.1.2 TPM_RESULT buffer_copy (buffer_t * *dest*, const buffer_t * *src*)

4.167.1.3 BOOL buffer_eq (const buffer_t * *a*, const buffer_t * *b*)

4.167.1.4 TPM_RESULT buffer_free (buffer_t * *buf*)

4.167.1.5 TPM_RESULT buffer_init (buffer_t * *buf*, tpm_size_t *initsize*, const BYTE * *initval*)

4.167.1.6 TPM_RESULT buffer_init_alias (buffer_t * *buf*, const buffer_t * *b*, tpm_size_t *offset*, tpm_size_t *len*)

4.167.1.7 TPM_RESULT buffer_init_alias_convert (buffer_t * *buf*, tpm_size_t *size*, BYTE * *val*)

4.167.1.8 TPM_RESULT buffer_init_const (buffer_t * *buf*, tpm_size_t *size*, const BYTE * *val*)

4.167.1.9 TPM_RESULT buffer_init_convert (buffer_t * *buf*, tpm_size_t *initsize*, BYTE * *initval*)

4.167.1.10 TPM_RESULT buffer_init_copy (buffer_t * *buf*, const buffer_t * *src*)

4.167.1.11 tpm_size_t buffer_len (const buffer_t * *buf*)

4.167.1.12 void buffer_memset (buffer_t * *buf*, BYTE *b*)

4.167.1.13 TPM_RESULT buffer_prepend_raw (buffer_t * *buf*, tpm_size_t *len*, const BYTE * *bytes*)

4.168 xen/tools/vtpm_manager/util/buffer.h File Reference

```
#include <stddef.h>
#include "tcg.h"
```

Data Structures

- struct buffer_t

Macros

- #define NULL_BUF {0,0,0,0}

Typedefs

- typedef UINT32 tpm_size_t

Functions

- TPM_RESULT buffer_init (buffer_t *buf, tpm_size_t initsize, const BYTE *initval)
- TPM_RESULT buffer_init_convert (buffer_t *buf, tpm_size_t initsize, BYTE *initval)
- TPM_RESULT buffer_init_const (buffer_t *buf, tpm_size_t size, const BYTE *val)
- TPM_RESULT buffer_init_alias (buffer_t *buf, const buffer_t *b, tpm_size_t offset, tpm_size_t)
- TPM_RESULT buffer_init_alias_convert (buffer_t *buf, tpm_size_t size, BYTE *val)

- TPM_RESULT buffer_init_copy (buffer_t *buf, const buffer_t *src)
- TPM_RESULT buffer_copy (buffer_t *dest, const buffer_t *src)
- BOOL buffer_eq (const buffer_t *a, const buffer_t *b)
- void buffer_memset (buffer_t *buf, BYTE b)
- tpm_size_t buffer_len (const buffer_t *buf)
- TPM_RESULT buffer_free (buffer_t *buf)
- TPM_RESULT buffer_append_raw (buffer_t *buf, tpm_size_t len, const BYTE *bytes)
- TPM_RESULT buffer_prepend_raw (buffer_t *buf, tpm_size_t len, const BYTE *bytes)

4.168.1 Macro Definition Documentation

4.168.1.1 #define NULL_BUF {0,0,0,0}

4.168.2 Typedef Documentation

4.168.2.1 typedef UINT32 tpm_size_t

4.168.3 Function Documentation

4.168.3.1 TPM_RESULT buffer_append_raw (buffer_t * *buf,* tpm_size_t *len,* const BYTE * *bytes*)

4.168.3.2 TPM_RESULT buffer_copy (buffer_t * *dest,* const buffer_t * *src*)

4.168.3.3 BOOL buffer_eq (const buffer_t * *a,* const buffer_t * *b*)

4.168.3.4 TPM_RESULT buffer_free (buffer_t * *buf*)

4.168.3.5 TPM_RESULT buffer_init (buffer_t * *buf,* tpm_size_t *initsize,* const BYTE * *initval*)

4.168.3.6 TPM_RESULT buffer_init_alias (buffer_t * *buf,* const buffer_t * *b,* tpm_size_t *offset,* tpm_size_t)

4.168.3.7 TPM_RESULT buffer_init_alias_convert (buffer_t * *buf,* tpm_size_t *size,* BYTE * *val*)

4.168.3.8 TPM_RESULT buffer_init_const (buffer_t * *buf,* tpm_size_t *size,* const BYTE * *val*)

4.168.3.9 TPM_RESULT buffer_init_convert (buffer_t * *buf,* tpm_size_t *initsize,* BYTE * *initval*)

4.168.3.10 TPM_RESULT buffer_init_copy (buffer_t * *buf,* const buffer_t * *src*)

4.168.3.11 tpm_size_t buffer_len (const buffer_t * *buf*)

4.168.3.12 void buffer_memset (buffer_t * *buf,* BYTE *b*)

4.168.3.13 TPM_RESULT buffer_prepend_raw (buffer_t * *buf,* tpm_size_t *len,* const BYTE * *bytes*)

4.169 xen/tools/vtpm_manager/util/tcg.h File Reference

```
#include <stdio.h>
```

Data Structures

- struct TPM_VERSION
- struct TPM_DIGEST

- struct TPM_NONCE
- struct TPM_KEY_PARMS
- struct TPM_RSA_KEY_PARMS
- struct TPM_STORE_PUBKEY
- struct TPM_PUBKEY
- struct TPM_KEY
- struct TPM_PCR_SELECTION
- struct TPM_PCR_COMPOSITE
- struct TPM_PCR_INFO
- struct TPM_BOUND_DATA
- struct TPM_STORED_DATA
- struct TCS_AUTH
- struct pack_buf_t
- struct pack_constbuf_t

Macros

- #define TPM_DIGEST_SIZE 20
- #define NULL_PACK_BUF {0,0}
- #define TRUE 0x01
- #define FALSE 0x00
- #define TCPA_MAX_BUFFER_LENGTH 0x2000
- #define TPM_PROTECTED_ORDINAL 0x00000000UL
- #define TPM_UNPROTECTED_ORDINAL 0x80000000UL
- #define TPM_CONNECTION_ORDINAL 0x40000000UL
- #define TPM_VENDOR_ORDINAL 0x20000000UL
- #define TPM_ORD_OIAP (10UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_OSAP (11UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_ChangeAuth (12UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_TakeOwnership (13UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_ChangeAuthAsymStart (14UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_ChangeAuthAsymFinish (15UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_ChangeAuthOwner (16UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Extend (20UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_PcrRead (21UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Quote (22UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Seal (23UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Unseal (24UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_DirWriteAuth (25UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_DirRead (26UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_UnBind (30UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_CreateWrapKey (31UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_LoadKey (32UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_GetPubKey (33UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_EvictKey (34UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_CreateMigrationBlob (40UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_ReWrapKey (41UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_ConvertMigrationBlob (42UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_AuthorizeMigrationKey (43UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_CreateMaintenanceArchive (44UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_LoadMaintenanceArchive (45UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_KillMaintenanceFeature (46UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_LoadManuMaintPub (47UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_ReadManuMaintPub (48UL + TPM_PROTECTED_ORDINAL)

- #define TPM_ORD_CertifyKey (50UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Sign (60UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_GetRandom (70UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_StirRandom (71UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_SelfTestFull (80UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_SelfTestStartup (81UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_CertifySelfTest (82UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_ContinueSelfTest (83UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_GetTestResult (84UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Reset (90UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_OwnerClear (91UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_DisableOwnerClear (92UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_ForceClear (93UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_DisableForceClear (94UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_GetCapabilitySigned (100UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_GetCapability (101UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_GetCapabilityOwner (102UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_OwnerSetDisable (110UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_PhysicalEnable (111UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_PhysicalDisable (112UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_SetOwnerInstall (113UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_PhysicalSetDeactivated (114UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_SetTempDeactivated (115UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_CreateEndorsementKeyPair (120UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_MakeIdentity (121UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_ActivateIdentity (122UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_ReadPubek (124UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_OwnerReadPubek (125UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_DisablePubekRead (126UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_GetAuditEvent (130UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_GetAuditEventSigned (131UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_GetOrdinalAuditStatus (140UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_SetOrdinalAuditStatus (141UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Terminate_Handle (150UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Init (151UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_SaveState (152UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Startup (153UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_SetRedirection (154UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_SHA1Start (160UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_SHA1Update (161UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_SHA1Complete (162UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_SHA1CompleteExtend (163UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_FieldUpgrade (170UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_SaveKeyContext (180UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_LoadKeyContext (181UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_SaveAuthContext (182UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_LoadAuthContext (183UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_SaveContext (184UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_LoadContext (185UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_FlushSpecific (186UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_PCR_Reset (200UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_NV_DefineSpace (204UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_NV_WriteValue (205UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_NV_WriteValueAuth (206UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_NV_ReadValue (207UL + TPM_PROTECTED_ORDINAL)

- #define TPM_ORD_NV_ReadValueAuth (208UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Delegate_UpdateVerification (209UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Delegate_Manage (210UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Delegate_CreateKeyDelegation (212UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Delegate_CreateOwnerDelegation (213UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Delegate_VerifyDelegation (214UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Delegate_LoadOwnerDelegation (216UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Delegate_ReadAuth (217UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_Delegate_ReadTable (219UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_CreateCounter (220UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_IncrementCounter (221UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_ReadCounter (222UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_ReleaseCounter (223UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_ReleaseCounterOwner (224UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_EstablishTransport (230UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_ExecuteTransport (231UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_ReleaseTransportSigned (232UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_GetTicks (241UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_TickStampBlob (242UL + TPM_PROTECTED_ORDINAL)
- #define TPM_ORD_MAX (256UL + TPM_PROTECTED_ORDINAL)
- #define TSC_ORD_PhysicalPresence (10UL + TPM_CONNECTION_ORDINAL)
- #define TPM_BASE 0x0
- #define TPM_VENDOR_ERROR 0x00000400
- #define TPM_NON_FATAL 0x00000800
- #define TPM_SUCCESS TPM_BASE
- #define TPM_AUTHFAIL TPM_BASE + 1
- #define TPM_BADINDEX TPM_BASE + 2
- #define TPM_BAD_PARAMETER TPM_BASE + 3
- #define TPM_AUDITFAILURE TPM_BASE + 4
- #define TPM_CLEAR_DISABLED TPM_BASE + 5
- #define TPM_DEACTIVATED TPM_BASE + 6
- #define TPM_DISABLED TPM_BASE + 7
- #define TPM_DISABLED_CMD TPM_BASE + 8
- #define TPM_FAIL TPM_BASE + 9
- #define TPM_BAD_ORDINAL TPM_BASE + 10
- #define TPM_INSTALL_DISABLED TPM_BASE + 11
- #define TPM_INVALID_KEYHANDLE TPM_BASE + 12
- #define TPM_KEYNOTFOUND TPM_BASE + 13
- #define TPM_INAPPROPRIATE_ENC TPM_BASE + 14
- #define TPM_MIGRATEFAIL TPM_BASE + 15
- #define TPM_INVALID_PCR_INFO TPM_BASE + 16
- #define TPM_NOSPACE TPM_BASE + 17
- #define TPM_NOSRK TPM_BASE + 18
- #define TPM_NOTSEALED_BLOB TPM_BASE + 19
- #define TPM_OWNER_SET TPM_BASE + 20
- #define TPM_RESOURCES TPM_BASE + 21
- #define TPM_SHORTRANDOM TPM_BASE + 22
- #define TPM_SIZE TPM_BASE + 23
- #define TPM_WRONGPCRVAL TPM_BASE + 24
- #define TPM_BAD_PARAM_SIZE TPM_BASE + 25
- #define TPM_SHA_THREAD TPM_BASE + 26
- #define TPM_SHA_ERROR TPM_BASE + 27
- #define TPM_FAILEDSELFTEST TPM_BASE + 28
- #define TPM_AUTH2FAIL TPM_BASE + 29
- #define TPM_BADTAG TPM_BASE + 30

- #define TPM_IOERROR TPM_BASE + 31
- #define TPM_ENCRYPT_ERROR TPM_BASE + 32
- #define TPM_DECRYPT_ERROR TPM_BASE + 33
- #define TPM_INVALID_AUTHHANDLE TPM_BASE + 34
- #define TPM_NO_ENDORSEMENT TPM_BASE + 35
- #define TPM_INVALID_KEYUSAGE TPM_BASE + 36
- #define TPM_WRONG_ENTITYTYPE TPM_BASE + 37
- #define TPM_INVALID_POSTINIT TPM_BASE + 38
- #define TPM_INAPPROPRIATE_SIG TPM_BASE + 39
- #define TPM_BAD_KEY_PROPERTY TPM_BASE + 40
- #define TPM_BAD_MIGRATION TPM_BASE + 41
- #define TPM_BAD_SCHEME TPM_BASE + 42
- #define TPM_BAD_DATASIZE TPM_BASE + 43
- #define TPM_BAD_MODE TPM_BASE + 44
- #define TPM_BAD_PRESENCE TPM_BASE + 45
- #define TPM_BAD_VERSION TPM_BASE + 46
- #define TPM_NO_WRAP_TRANSPORT TPM_BASE + 47
- #define TPM_AUDITFAIL_UNSUCCESSFUL TPM_BASE + 48
- #define TPM_AUDITFAIL_SUCCESSFUL TPM_BASE + 49
- #define TPM_NOTRESETABLE TPM_BASE + 50
- #define TPM_NOTLOCAL TPM_BASE + 51
- #define TPM_BAD_TYPE TPM_BASE + 52
- #define TPM_INVALID_RESOURCE TPM_BASE + 53
- #define TPM_NOTFIPS TPM_BASE + 54
- #define TPM_INVALID_FAMILY TPM_BASE + 55
- #define TPM_NO_NV_PERMISSION TPM_BASE + 56
- #define TPM_REQUIRES_SIGN TPM_BASE + 57
- #define TPM_KEY_NOTSUPPORTED TPM_BASE + 58
- #define TPM_AUTH_CONFLICT TPM_BASE + 59
- #define TPM_AREA_LOCKED TPM_BASE + 60
- #define TPM_BAD_LOCALITY TPM_BASE + 61
- #define TPM_READ_ONLY TPM_BASE + 62
- #define TPM_PER_NOWRITE TPM_BASE + 63
- #define TPM_FAMILYCOUNT TPM_BASE + 64
- #define TPM_WRITE_LOCKED TPM_BASE + 65
- #define TPM_BAD_ATTRIBUTES TPM_BASE + 66
- #define TPM_INVALID_STRUCTURE TPM_BASE + 67
- #define TPM_KEY_OWNER_CONTROL TPM_BASE + 68
- #define TPM_BAD_COUNTER TPM_BASE + 69
- #define TPM_NOT_FULLWRITE TPM_BASE + 70
- #define TPM_CONTEXT_GAP TPM_BASE + 71
- #define TPM_MAXNVWRITES TPM_BASE + 72
- #define TPM_NOOPERATOR TPM_BASE + 73
- #define TPM_RESOURCEMISSING TPM_BASE + 74
- #define TPM_DELEGATE_LOCK TPM_BASE + 75
- #define TPM_DELEGATE_FAMILY TPM_BASE + 76
- #define TPM_DELEGATE_ADMIN TPM_BASE + 77
- #define TPM_TRANSPORT_EXCLUSIVE TPM_BASE + 78
- #define TPM_ST_CLEAR 0x0001
- #define TPM_ST_STATE 0x0002
- #define TPM_ST_DEACTIVATED 0x003
- #define TPM_TAG_RQU_COMMAND 0x00c1
- #define TPM_TAG_RQU_AUTH1_COMMAND 0x00c2
- #define TPM_TAG_RQU_AUTH2_COMMAND 0x00c3
- #define TPM_TAG_RSP_COMMAND 0x00c4

- #define TPM_TAG_RSP_AUTH1_COMMAND 0x00c5
- #define TPM_TAG_RSP_AUTH2_COMMAND 0x00c6
- #define TPM_PT_ASYM 0x01
- #define TPM_PT_BIND 0x02
- #define TPM_PT_MIGRATE 0x03
- #define TPM_PT_MAINT 0x04
- #define TPM_PT_SEAL 0x05
- #define TPM_ET_KEYHANDLE 0x0001
- #define TPM_ET_OWNER 0x0002
- #define TPM_ET_DATA 0x0003
- #define TPM_ET_SRK 0x0004
- #define TPM_ET_KEY 0x0005
- #define TPM_RT_KEY 0x00000001

  TPM_ResourceTypes.
- #define TPM_RT_AUTH 0x00000002
- #define TPM_RT_HASH 0x00000003
- #define TPM_RT_TRANS 0x00000004
- #define TPM_RT_CONTEXT 0x00000005
- #define TPM_RT_COUNTER 0x00000006
- #define TPM_RT_DELEGATE 0x00000007
- #define TPM_RT_DAA_TPM 0x00000008
- #define TPM_RT_DAA_V0 0x00000009
- #define TPM_RT_DAA_V1 0x0000000A
- #define TPM_PID_OIAP 0x0001
- #define TPM_PID_OSAP 0x0002
- #define TPM_PID_ADIP 0x0003
- #define TPM_PID_ADCP 0x0004
- #define TPM_PID_OWNER 0x0005
- #define TPM_ALG_RSA 0x00000001
- #define TPM_ALG_DES 0x00000002
- #define TPM_ALG_3DES 0X00000003
- #define TPM_ALG_SHA 0x00000004
- #define TPM_ALG_HMAC 0x00000005
- #define TCPA_ALG_AES 0x00000006
- #define TPM_ES_NONE 0x0001
- #define TPM_ES_RSAESPKCSv15 0x0002
- #define TPM_ES_RSAESOAEP_SHA1_MGF1 0x0003
- #define TPM_SS_NONE 0x0001
- #define TPM_SS_RSASSAPKCS1v15_SHA1 0x0002
- #define TPM_SS_RSASSAPKCS1v15_DER 0x0003
- #define TPM_CAP_ORD 0x00000001
- #define TPM_CAP_ALG 0x00000002
- #define TPM_CAP_PID 0x00000003
- #define TPM_CAP_FLAG 0x00000004
- #define TPM_CAP_PROPERTY 0x00000005
- #define TPM_CAP_VERSION 0x00000006
- #define TPM_CAP_KEY_HANDLE 0x00000007
- #define TPM_CAP_CHECK_LOADED 0x00000008
- #define TPM_CAP_SYM_MODE 0x00000009
- #define TPM_CAP_KEY_STATUS 0x0000000C
- #define TPM_CAP_NV_LIST 0x0000000D
- #define TPM_CAP_MFR 0x00000010
- #define TPM_CAP_NV_INDEX 0x00000011
- #define TPM_CAP_TRANS_ALG 0x00000012

- #define TPM_CAP_HANDLE 0x00000014
- #define TPM_CAP_TRANS_ES 0x00000015
- #define TPM_CAP_AUTH_ENCRYPT 0x00000017
- #define TPM_CAP_SELECT_SIZE 0x00000018
- #define TPM_CAP_DA_LOGIC 0x00000019
- #define TPM_CAP_VERSION_VAL 0x0000001A
- #define TPM_CAP_PROP_PCR 0x00000101
- #define TPM_CAP_PROP_DIR 0x00000102
- #define TPM_CAP_PROP_MANUFACTURER 0x00000103
- #define TPM_CAP_PROP_KEYS 0x00000104
- #define TPM_CAP_PROP_MIN_COUNTER 0x00000107
- #define TPM_CAP_FLAG_PERMANENT 0x00000108
- #define TPM_CAP_FLAG_VOLATILE 0x00000109
- #define TPM_CAP_PROP_AUTHSESS 0x0000010A
- #define TPM_CAP_PROP_TRANSESS 0x0000010B
- #define TPM_CAP_PROP_COUNTERS 0x0000010C
- #define TPM_CAP_PROP_MAX_AUTHSESS 0x0000010D
- #define TPM_CAP_PROP_MAX_TRANSESS 0x0000010E
- #define TPM_CAP_PROP_MAX_COUNTERS 0x0000010F
- #define TPM_CAP_PROP_MAX_KEYS 0x00000110
- #define TPM_CAP_PROP_OWNER 0x00000111
- #define TPM_CAP_PROP_CONTEXT 0x00000112
- #define TPM_CAP_PROP_MAX_CONTEXT 0x00000113
- #define TPM_CAP_PROP_FAMILYROWS 0x00000114
- #define TPM_CAP_PROP_TIS_TIMEOUT 0x00000115
- #define TPM_CAP_PROP_STARTUP_EFFECT 0x00000116
- #define TPM_CAP_PROP_DELEGATE_ROW 0x00000117
- #define TPM_CAP_PROP_MAX_DAASESS 0x00000119
- #define TPM_CAP_PROP_DAASESS 0x0000011A
- #define TPM_CAP_PROP_CONTEXT_DIST 0x0000011B
- #define TPM_CAP_PROP_DAA_INTERRUPT 0x0000011C
- #define TPM_CAP_PROP_SESSIONS 0x0000011D
- #define TPM_CAP_PROP_MAX_SESSIONS 0x0000011E
- #define TPM_CAP_PROP_CMK_RESTRICTION 0x0000011F
- #define TPM_CAP_PROP_DURATION 0x00000120
- #define TPM_CAP_PROP_ACTIVE_COUNTER 0x00000122
- #define TPM_CAP_PROP_MAX_NV_AVAILABLE 0x00000123
- #define TPM_CAP_PROP_INPUT_BUFFER 0x00000124
- #define TPM_KEY_EK 0x0000
- #define TPM_KEY_SIGNING 0x0010
- #define TPM_KEY_STORAGE 0x0011
- #define TPM_KEY_IDENTITY 0x0012
- #define TPM_KEY_AUTHCHANGE 0X0013
- #define TPM_KEY_BIND 0x0014
- #define TPM_KEY_LEGACY 0x0015
- #define TPM_AUTH_NEVER 0x00
- #define TPM_AUTH_ALWAYS 0x01
- #define TPM_OWNER_KEYHANDLE 0x40000001
- #define TPM_SRK_KEYHANDLE 0x40000000
- #define ERRORDIE(s)
- #define TPMTRY(s, c)
- #define TPMTRYRETURN(c)

Typedefs

- typedef unsigned char BYTE
- typedef unsigned char BOOL
- typedef unsigned short UINT16
- typedef unsigned int UINT32
- typedef unsigned long long UINT64
- typedef UINT32 TPM_RESULT
- typedef UINT32 TPM_PCRINDEX
- typedef UINT32 TPM_DIRINDEX
- typedef UINT32 TPM_HANDLE
- typedef TPM_HANDLE TPM_AUTHHANDLE
- typedef TPM_HANDLE TCPA_HASHHANDLE
- typedef TPM_HANDLE TCPA_HMACHANDLE
- typedef TPM_HANDLE TCPA_ENCHANDLE
- typedef TPM_HANDLE TPM_KEY_HANDLE
- typedef TPM_HANDLE TCPA_ENTITYHANDLE
- typedef UINT32 TPM_RESOURCE_TYPE
- typedef UINT32 TPM_COMMAND_CODE
- typedef UINT16 TPM_PROTOCOL_ID
- typedef BYTE TPM_AUTH_DATA_USAGE
- typedef UINT16 TPM_ENTITY_TYPE
- typedef UINT32 TPM_ALGORITHM_ID
- typedef UINT16 TPM_KEY_USAGE
- typedef UINT16 TPM_STARTUP_TYPE
- typedef UINT32 TPM_CAPABILITY_AREA
- typedef UINT16 TPM_ENC_SCHEME
- typedef UINT16 TPM_SIG_SCHEME
- typedef UINT16 TPM_MIGRATE_SCHEME
- typedef UINT16 TPM_PHYSICAL_PRESENCE
- typedef UINT32 TPM_KEY_FLAGS
- typedef BYTE TPM_AUTHDATA [TPM_DIGEST_SIZE]
- typedef TPM_AUTHDATA TPM_SECRET
- typedef TPM_AUTHDATA TPM_ENCAUTH
- typedef BYTE TPM_PAYLOAD_TYPE
- typedef UINT16 TPM_TAG
- typedef UINT32 TCS_AUTHHANDLE
- typedef UINT32 TCS_CONTEXT_HANDLE
- typedef UINT32 TCS_KEY_HANDLE
- typedef struct TPM_VERSION TPM_VERSION
- typedef struct TPM_DIGEST TPM_DIGEST
- typedef TPM_DIGEST TPM_PCRVALUE
- typedef TPM_DIGEST TPM_COMPOSITE_HASH
- typedef TPM_DIGEST TPM_DIRVALUE
- typedef TPM_DIGEST TPM_HMAC
- typedef TPM_DIGEST TPM_CHOSENID_HASH
- typedef struct TPM_NONCE TPM_NONCE
- typedef struct TPM_KEY_PARMS TPM_KEY_PARMS
- typedef struct TPM_RSA_KEY_PARMS TPM_RSA_KEY_PARMS
- typedef struct TPM_STORE_PUBKEY TPM_STORE_PUBKEY
- typedef struct TPM_PUBKEY TPM_PUBKEY
- typedef struct TPM_KEY TPM_KEY
- typedef struct TPM_PCR_SELECTION TPM_PCR_SELECTION
- typedef struct TPM_PCR_COMPOSITE TPM_PCR_COMPOSITE
- typedef struct TPM_PCR_INFO TPM_PCR_INFO

- typedef struct TPM_BOUND_DATA TPM_BOUND_DATA
- typedef struct TPM_STORED_DATA TPM_STORED_DATA
- typedef struct TCS_AUTH TCS_AUTH
- typedef struct pack_buf_t pack_buf_t
- typedef struct pack_constbuf_t pack_constbuf_t

4.169.1 Macro Definition Documentation

4.169.1.1 #define ERRORDIE(*s*)

Value:

```
do { status = s; \
                        fprintf (stderr, "*** ERRORDIE in %s at %s: %i\n", __func__, __FILE__, __LINE__);
      \
                        goto abort_egress; } \
                    while (0)
```

4.169.1.2 #define FALSE 0x00

4.169.1.3 #define NULL_PACK_BUF {0,0}

4.169.1.4 #define TCPA_ALG_AES 0x00000006

4.169.1.5 #define TCPA_MAX_BUFFER_LENGTH 0x2000

4.169.1.6 #define TPM_ALG_3DES 0X00000003

4.169.1.7 #define TPM_ALG_DES 0x00000002

4.169.1.8 #define TPM_ALG_HMAC 0x00000005

4.169.1.9 #define TPM_ALG_RSA 0x00000001

4.169.1.10 #define TPM_ALG_SHA 0x00000004

4.169.1.11 #define TPM_AREA_LOCKED TPM_BASE + 60

4.169.1.12 #define TPM_AUDITFAIL_SUCCESSFUL TPM_BASE + 49

4.169.1.13 #define TPM_AUDITFAIL_UNSUCCESSFUL TPM_BASE + 48

4.169.1.14 #define TPM_AUDITFAILURE TPM_BASE + 4

4.169.1.15 #define TPM_AUTH2FAIL TPM_BASE + 29

4.169.1.16 #define TPM_AUTH_ALWAYS 0x01

4.169.1.17 #define TPM_AUTH_CONFLICT TPM_BASE + 59

4.169.1.18 #define TPM_AUTH_NEVER 0x00

4.169.1.19 #define TPM_AUTHFAIL TPM_BASE + 1

4.169.1.20 #define TPM_BAD_ATTRIBUTES TPM_BASE + 66

4.169.1.21 #define TPM_BAD_COUNTER TPM_BASE + 69

4.169.1.22 #define TPM_BAD_DATASIZE TPM_BASE + 43

4.169.1.23 #define TPM_BAD_KEY_PROPERTY TPM_BASE + 40

4.169.1.24 #define TPM_BAD_LOCALITY TPM_BASE + 61

4.169.1.25 #define TPM_BAD_MIGRATION TPM_BASE + 41

4.169.1.26 #define TPM_BAD_MODE TPM_BASE + 44

4.169.1.27 #define TPM_BAD_ORDINAL TPM_BASE + 10

4.169.1.28 #define TPM_BAD_PARAM_SIZE TPM_BASE + 25

4.169.1.29 #define TPM_BAD_PARAMETER TPM_BASE + 3

4.169.1.30 #define TPM_BAD_PRESENCE TPM_BASE + 45

4.169.1.31 #define TPM_BAD_SCHEME TPM_BASE + 42

4.169.1.32 #define TPM_BAD_TYPE TPM_BASE + 52

4.169.1.33 #define TPM_BAD_VERSION TPM_BASE + 46

4.169.1.34 #define TPM_BADINDEX TPM_BASE + 2

4.169.1.35 #define TPM_BADTAG TPM_BASE + 30

4.169.1.36 #define TPM_BASE 0x0

4.169.1.37 #define TPM_CAP_ALG 0x00000002

4.169.1.38 #define TPM_CAP_AUTH_ENCRYPT 0x00000017

4.169.1.39 #define TPM_CAP_CHECK_LOADED 0x00000008

4.169.1.40 #define TPM_CAP_DA_LOGIC 0x00000019

4.169.1.41 #define TPM_CAP_FLAG 0x00000004

4.169.1.42 #define TPM_CAP_FLAG_PERMANENT 0x00000108

4.169.1.43 #define TPM_CAP_FLAG_VOLATILE 0x00000109

4.169.1.44 #define TPM_CAP_HANDLE 0x00000014

4.169.1.45 #define TPM_CAP_KEY_HANDLE 0x00000007

4.169.1.46 #define TPM_CAP_KEY_STATUS 0x0000000C

4.169.1.47 #define TPM_CAP_MFR 0x00000010

4.169.1.48 #define TPM_CAP_NV_INDEX 0x00000011

4.169.1.49 #define TPM_CAP_NV_LIST 0x0000000D

4.169.1.50 #define TPM_CAP_ORD 0x00000001

4.169.1.51 #define TPM_CAP_PID 0x00000003

4.169.1.52 #define TPM_CAP_PROP_ACTIVE_COUNTER 0x00000122

4.169.1.53 #define TPM_CAP_PROP_AUTHSESS 0x0000010A

4.169.1.54 #define TPM_CAP_PROP_CMK_RESTRICTION 0x0000011F

4.169.1.55 #define TPM_CAP_PROP_CONTEXT 0x00000112

4.169.1.56 #define TPM_CAP_PROP_CONTEXT_DIST 0x0000011B

4.169.1.57 #define TPM_CAP_PROP_COUNTERS 0x0000010C

4.169.1.58 #define TPM_CAP_PROP_DAA_INTERRUPT 0x0000011C

4.169.1.59 #define TPM_CAP_PROP_DAASESS 0x0000011A

4.169.1.60 #define TPM_CAP_PROP_DELEGATE_ROW 0x00000117

4.169.1.61 #define TPM_CAP_PROP_DIR 0x00000102

4.169.1.62 #define TPM_CAP_PROP_DURATION 0x00000120

4.169.1.63 #define TPM_CAP_PROP_FAMILYROWS 0x00000114

4.169.1.64 #define TPM_CAP_PROP_INPUT_BUFFER 0x00000124

4.169.1.65 #define TPM_CAP_PROP_KEYS 0x00000104

4.169.1.66 #define TPM_CAP_PROP_MANUFACTURER 0x00000103

4.169.1.67 #define TPM_CAP_PROP_MAX_AUTHSESS 0x0000010D

4.169.1.68 #define TPM_CAP_PROP_MAX_CONTEXT 0x00000113

4.169.1.69 #define TPM_CAP_PROP_MAX_COUNTERS 0x0000010F

4.169.1.70 #define TPM_CAP_PROP_MAX_DAASESS 0x00000119

4.169.1.71 #define TPM_CAP_PROP_MAX_KEYS 0x00000110

4.169.1.72 #define TPM_CAP_PROP_MAX_NV_AVAILABLE 0x00000123

4.169.1.73 #define TPM_CAP_PROP_MAX_SESSIONS 0x0000011E

4.169.1.74 #define TPM_CAP_PROP_MAX_TRANSESS 0x0000010E

4.169.1.75 #define TPM_CAP_PROP_MIN_COUNTER 0x00000107

4.169.1.76 #define TPM_CAP_PROP_OWNER 0x00000111

4.169.1.77 #define TPM_CAP_PROP_PCR 0x00000101

4.169.1.78 #define TPM_CAP_PROP_SESSIONS 0x0000011D

4.169.1.79 #define TPM_CAP_PROP_STARTUP_EFFECT 0x00000116

4.169.1.80 #define TPM_CAP_PROP_TIS_TIMEOUT 0x00000115

4.169.1.81 #define TPM_CAP_PROP_TRANSESS 0x0000010B

4.169.1.82 #define TPM_CAP_PROPERTY 0x00000005

4.169.1.83 #define TPM_CAP_SELECT_SIZE 0x00000018

4.169.1.84 #define TPM_CAP_SYM_MODE 0x00000009

4.169.1.85 #define TPM_CAP_TRANS_ALG 0x00000012

4.169.1.86 #define TPM_CAP_TRANS_ES 0x00000015

4.169.1.87 #define TPM_CAP_VERSION 0x00000006

4.169.1.88 #define TPM_CAP_VERSION_VAL 0x0000001A

4.169.1.89 #define TPM_CLEAR_DISABLED TPM_BASE + 5

4.169.1.90 #define TPM_CONNECTION_ORDINAL 0x40000000UL

4.169.1.91 #define TPM_CONTEXT_GAP TPM_BASE + 71

4.169.1.92 #define TPM_DEACTIVATED TPM_BASE + 6

4.169.1.93 #define TPM_DECRYPT_ERROR TPM_BASE + 33

4.169.1.94 #define TPM_DELEGATE_ADMIN TPM_BASE + 77

4.169.1.95 #define TPM_DELEGATE_FAMILY TPM_BASE + 76

4.169.1.96 #define TPM_DELEGATE_LOCK TPM_BASE + 75

4.169.1.97 #define TPM_DIGEST_SIZE 20

4.169.1.98 #define TPM_DISABLED TPM_BASE + 7

4.169.1.99 #define TPM_DISABLED_CMD TPM_BASE + 8

4.169.1.100 #define TPM_ENCRYPT_ERROR TPM_BASE + 32

4.169.1.101 #define TPM_ES_NONE 0x0001

4.169.1.102 #define TPM_ES_RSAESOAEP_SHA1_MGF1 0x0003

4.169.1.103 #define TPM_ES_RSAESPKCSv15 0x0002

4.169.1.104 #define TPM_ET_DATA 0x0003

4.169.1.105 #define TPM_ET_KEY 0x0005

4.169.1.106 #define TPM_ET_KEYHANDLE 0x0001

4.169.1.107 #define TPM_ET_OWNER 0x0002

4.169.1.108 #define TPM_ET_SRK 0x0004

4.169.1.109 #define TPM_FAIL TPM_BASE + 9

4.169.1.110 #define TPM_FAILEDSELFTEST TPM_BASE + 28

4.169.1.111 #define TPM_FAMILYCOUNT TPM_BASE + 64

4.169.1.112 #define TPM_INAPPROPRIATE_ENC TPM_BASE + 14

4.169.1.113 #define TPM_INAPPROPRIATE_SIG TPM_BASE + 39

4.169.1.114 #define TPM_INSTALL_DISABLED TPM_BASE + 11

4.169.1.115 #define TPM_INVALID_AUTHHANDLE TPM_BASE + 34

4.169.1.116 #define TPM_INVALID_FAMILY TPM_BASE + 55

4.169.1.117 #define TPM_INVALID_KEYHANDLE TPM_BASE + 12

4.169.1.118 #define TPM_INVALID_KEYUSAGE TPM_BASE + 36

4.169.1.119 #define TPM_INVALID_PCR_INFO TPM_BASE + 16

4.169.1.120 #define TPM_INVALID_POSTINIT TPM_BASE + 38

4.169.1.121 #define TPM_INVALID_RESOURCE TPM_BASE + 53

4.169.1.122 #define TPM_INVALID_STRUCTURE TPM_BASE + 67

4.169.1.123 #define TPM_IOERROR TPM_BASE + 31

4.169.1.124 #define TPM_KEY_AUTHCHANGE 0X0013

4.169.1.125 #define TPM_KEY_BIND 0x0014

4.169.1.126 #define TPM_KEY_EK 0x0000

4.169.1.127 #define TPM_KEY_IDENTITY 0x0012

4.169.1.128 #define TPM_KEY_LEGACY 0x0015

4.169.1.129 #define TPM_KEY_NOTSUPPORTED TPM_BASE + 58

4.169.1.130 #define TPM_KEY_OWNER_CONTROL TPM_BASE + 68

4.169.1.131 #define TPM_KEY_SIGNING 0x0010

4.169.1.132 #define TPM_KEY_STORAGE 0x0011

4.169.1.133 #define TPM_KEYNOTFOUND TPM_BASE + 13

4.169.1.134 #define TPM_MAXNVWRITES TPM_BASE + 72

4.169.1.135 #define TPM_MIGRATEFAIL TPM_BASE + 15

4.169.1.136 #define TPM_NO_ENDORSEMENT TPM_BASE + 35

4.169.1.137 #define TPM_NO_NV_PERMISSION TPM_BASE + 56

4.169.1.138 #define TPM_NO_WRAP_TRANSPORT TPM_BASE + 47

4.169.1.139 #define TPM_NON_FATAL 0x00000800

4.169.1.140 #define TPM_NOOPERATOR TPM_BASE + 73

4.169.1.141 #define TPM_NOSPACE TPM_BASE + 17

4.169.1.142 #define TPM_NOSRK TPM_BASE + 18

4.169.1.143 #define TPM_NOT_FULLWRITE TPM_BASE + 70

4.169.1.144 #define TPM_NOTFIPS TPM_BASE + 54

4.169.1.145 #define TPM_NOTLOCAL TPM_BASE + 51

4.169.1.146 #define TPM_NOTRESETABLE TPM_BASE + 50

4.169.1.147 #define TPM_NOTSEALED_BLOB TPM_BASE + 19

4.169.1.148 #define TPM_ORD_ActivateIdentity (122UL + TPM_PROTECTED_ORDINAL)

4.169.1.149 #define TPM_ORD_AuthorizeMigrationKey (43UL + TPM_PROTECTED_ORDINAL)

4.169.1.150 #define TPM_ORD_CertifyKey (50UL + TPM_PROTECTED_ORDINAL)

4.169.1.151 #define TPM_ORD_CertifySelfTest (82UL + TPM_PROTECTED_ORDINAL)

4.169.1.152 #define TPM_ORD_ChangeAuth (12UL + TPM_PROTECTED_ORDINAL)

4.169.1.153 #define TPM_ORD_ChangeAuthAsymFinish (15UL + TPM_PROTECTED_ORDINAL)

4.169.1.154 #define TPM_ORD_ChangeAuthAsymStart (14UL + TPM_PROTECTED_ORDINAL)

4.169.1.155 #define TPM_ORD_ChangeAuthOwner (16UL + TPM_PROTECTED_ORDINAL)

4.169.1.156 #define TPM_ORD_ContinueSelfTest (83UL + TPM_PROTECTED_ORDINAL)

4.169.1.157 #define TPM_ORD_ConvertMigrationBlob (42UL + TPM_PROTECTED_ORDINAL)

4.169.1.158 #define TPM_ORD_CreateCounter (220UL + TPM_PROTECTED_ORDINAL)

4.169.1.159 #define TPM_ORD_CreateEndorsementKeyPair (120UL + TPM_PROTECTED_ORDINAL)

4.169.1.160 #define TPM_ORD_CreateMaintenanceArchive (44UL + TPM_PROTECTED_ORDINAL)

4.169.1.161 #define TPM_ORD_CreateMigrationBlob (40UL + TPM_PROTECTED_ORDINAL)

4.169.1.162 #define TPM_ORD_CreateWrapKey (31UL + TPM_PROTECTED_ORDINAL)

4.169.1.163 #define TPM_ORD_Delegate_CreateKeyDelegation (212UL + TPM_PROTECTED_ORDINAL)

4.169.1.164 #define TPM_ORD_Delegate_CreateOwnerDelegation (213UL + TPM_PROTECTED_ORDINAL)

4.169.1.165 #define TPM_ORD_Delegate_LoadOwnerDelegation (216UL + TPM_PROTECTED_ORDINAL)

4.169.1.166 #define TPM_ORD_Delegate_Manage (210UL + TPM_PROTECTED_ORDINAL)

4.169.1.167 #define TPM_ORD_Delegate_ReadAuth (217UL + TPM_PROTECTED_ORDINAL)

4.169.1.168 #define TPM_ORD_Delegate_ReadTable (219UL + TPM_PROTECTED_ORDINAL)

4.169.1.169 #define TPM_ORD_Delegate_UpdateVerification (209UL + TPM_PROTECTED_ORDINAL)

4.169.1.170 #define TPM_ORD_Delegate_VerifyDelegation (214UL + TPM_PROTECTED_ORDINAL)

4.169.1.171 #define TPM_ORD_DirRead (26UL + TPM_PROTECTED_ORDINAL)

4.169.1.172 #define TPM_ORD_DirWriteAuth (25UL + TPM_PROTECTED_ORDINAL)

4.169.1.173 #define TPM_ORD_DisableForceClear (94UL + TPM_PROTECTED_ORDINAL)

4.169.1.174 #define TPM_ORD_DisableOwnerClear (92UL + TPM_PROTECTED_ORDINAL)

4.169.1.175 #define TPM_ORD_DisablePubekRead (126UL + TPM_PROTECTED_ORDINAL)

4.169.1.176 #define TPM_ORD_EstablishTransport (230UL + TPM_PROTECTED_ORDINAL)

4.169.1.177 #define TPM_ORD_EvictKey (34UL + TPM_PROTECTED_ORDINAL)

4.169.1.178 #define TPM_ORD_ExecuteTransport (231UL + TPM_PROTECTED_ORDINAL)

4.169.1.179 #define TPM_ORD_Extend (20UL + TPM_PROTECTED_ORDINAL)

4.169.1.180 #define TPM_ORD_FieldUpgrade (170UL + TPM_PROTECTED_ORDINAL)

4.169.1.181 #define TPM_ORD_FlushSpecific (186UL + TPM_PROTECTED_ORDINAL)

4.169.1.182 #define TPM_ORD_ForceClear (93UL + TPM_PROTECTED_ORDINAL)

4.169.1.183 #define TPM_ORD_GetAuditEvent (130UL + TPM_PROTECTED_ORDINAL)

4.169.1.184 #define TPM_ORD_GetAuditEventSigned (131UL + TPM_PROTECTED_ORDINAL)

4.169.1.185 #define TPM_ORD_GetCapability (101UL + TPM_PROTECTED_ORDINAL)

4.169.1.186 #define TPM_ORD_GetCapabilityOwner (102UL + TPM_PROTECTED_ORDINAL)

4.169.1.187 #define TPM_ORD_GetCapabilitySigned (100UL + TPM_PROTECTED_ORDINAL)

4.169.1.188 #define TPM_ORD_GetOrdinalAuditStatus (140UL + TPM_PROTECTED_ORDINAL)

4.169.1.189 #define TPM_ORD_GetPubKey (33UL + TPM_PROTECTED_ORDINAL)

4.169.1.190 #define TPM_ORD_GetRandom (70UL + TPM_PROTECTED_ORDINAL)

4.169.1.191 #define TPM_ORD_GetTestResult (84UL + TPM_PROTECTED_ORDINAL)

4.169.1.192 #define TPM_ORD_GetTicks (241UL + TPM_PROTECTED_ORDINAL)

4.169.1.193 #define TPM_ORD_IncrementCounter (221UL + TPM_PROTECTED_ORDINAL)

4.169.1.194 #define TPM_ORD_Init (151UL + TPM_PROTECTED_ORDINAL)

4.169.1.195 #define TPM_ORD_KillMaintenanceFeature (46UL + TPM_PROTECTED_ORDINAL)

4.169.1.196 #define TPM_ORD_LoadAuthContext (183UL + TPM_PROTECTED_ORDINAL)

4.169.1.197 #define TPM_ORD_LoadContext (185UL + TPM_PROTECTED_ORDINAL)

4.169.1.198 #define TPM_ORD_LoadKey (32UL + TPM_PROTECTED_ORDINAL)

4.169.1.199 #define TPM_ORD_LoadKeyContext (181UL + TPM_PROTECTED_ORDINAL)

4.169.1.200 #define TPM_ORD_LoadMaintenanceArchive (45UL + TPM_PROTECTED_ORDINAL)

4.169.1.201 #define TPM_ORD_LoadManuMaintPub (47UL + TPM_PROTECTED_ORDINAL)

4.169.1.202 #define TPM_ORD_MakeIdentity (121UL + TPM_PROTECTED_ORDINAL)

4.169.1.203 #define TPM_ORD_MAX (256UL + TPM_PROTECTED_ORDINAL)

4.169.1.204 #define TPM_ORD_NV_DefineSpace (204UL + TPM_PROTECTED_ORDINAL)

4.169.1.205 #define TPM_ORD_NV_ReadValue (207UL + TPM_PROTECTED_ORDINAL)

4.169.1.206 #define TPM_ORD_NV_ReadValueAuth (208UL + TPM_PROTECTED_ORDINAL)

4.169.1.207 #define TPM_ORD_NV_WriteValue (205UL + TPM_PROTECTED_ORDINAL)

4.169.1.208 #define TPM_ORD_NV_WriteValueAuth (206UL + TPM_PROTECTED_ORDINAL)

4.169.1.209 #define TPM_ORD_OIAP (10UL + TPM_PROTECTED_ORDINAL)

4.169.1.210 #define TPM_ORD_OSAP (11UL + TPM_PROTECTED_ORDINAL)

4.169.1.211 #define TPM_ORD_OwnerClear (91UL + TPM_PROTECTED_ORDINAL)

4.169.1.212 #define TPM_ORD_OwnerReadPubek (125UL + TPM_PROTECTED_ORDINAL)

4.169.1.213 #define TPM_ORD_OwnerSetDisable (110UL + TPM_PROTECTED_ORDINAL)

4.169.1.214 #define TPM_ORD_PCR_Reset (200UL + TPM_PROTECTED_ORDINAL)

4.169.1.215 #define TPM_ORD_PcrRead (21UL + TPM_PROTECTED_ORDINAL)

4.169.1.216 #define TPM_ORD_PhysicalDisable (112UL + TPM_PROTECTED_ORDINAL)

4.169.1.217 #define TPM_ORD_PhysicalEnable (111UL + TPM_PROTECTED_ORDINAL)

4.169.1.218 #define TPM_ORD_PhysicalSetDeactivated (114UL + TPM_PROTECTED_ORDINAL)

4.169.1.219 #define TPM_ORD_Quote (22UL + TPM_PROTECTED_ORDINAL)

4.169.1.220 #define TPM_ORD_ReadCounter (222UL + TPM_PROTECTED_ORDINAL)

4.169.1.221 #define TPM_ORD_ReadManuMaintPub (48UL + TPM_PROTECTED_ORDINAL)

4.169.1.222 #define TPM_ORD_ReadPubek (124UL + TPM_PROTECTED_ORDINAL)

4.169.1.223 #define TPM_ORD_ReleaseCounter (223UL + TPM_PROTECTED_ORDINAL)

4.169.1.224 #define TPM_ORD_ReleaseCounterOwner (224UL + TPM_PROTECTED_ORDINAL)

4.169.1.225 #define TPM_ORD_ReleaseTransportSigned (232UL + TPM_PROTECTED_ORDINAL)

4.169.1.226 #define TPM_ORD_Reset (90UL + TPM_PROTECTED_ORDINAL)

4.169.1.227 #define TPM_ORD_ReWrapKey (41UL + TPM_PROTECTED_ORDINAL)

4.169.1.228 #define TPM_ORD_SaveAuthContext (182UL + TPM_PROTECTED_ORDINAL)

4.169.1.229 #define TPM_ORD_SaveContext (184UL + TPM_PROTECTED_ORDINAL)

4.169.1.230 #define TPM_ORD_SaveKeyContext (180UL + TPM_PROTECTED_ORDINAL)

4.169.1.231 #define TPM_ORD_SaveState (152UL + TPM_PROTECTED_ORDINAL)

4.169.1.232 #define TPM_ORD_Seal (23UL + TPM_PROTECTED_ORDINAL)

4.169.1.233 #define TPM_ORD_SelfTestFull (80UL + TPM_PROTECTED_ORDINAL)

4.169.1.234 #define TPM_ORD_SelfTestStartup (81UL + TPM_PROTECTED_ORDINAL)

4.169.1.235 #define TPM_ORD_SetOrdinalAuditStatus (141UL + TPM_PROTECTED_ORDINAL)

4.169.1.236 #define TPM_ORD_SetOwnerInstall (113UL + TPM_PROTECTED_ORDINAL)

4.169.1.237 #define TPM_ORD_SetRedirection (154UL + TPM_PROTECTED_ORDINAL)

4.169.1.238 #define TPM_ORD_SetTempDeactivated (115UL + TPM_PROTECTED_ORDINAL)

4.169.1.239 #define TPM_ORD_SHA1Complete (162UL + TPM_PROTECTED_ORDINAL)

4.169.1.240 #define TPM_ORD_SHA1CompleteExtend (163UL + TPM_PROTECTED_ORDINAL)

4.169.1.241 #define TPM_ORD_SHA1Start (160UL + TPM_PROTECTED_ORDINAL)

4.169.1.242 #define TPM_ORD_SHA1Update (161UL + TPM_PROTECTED_ORDINAL)

4.169.1.243 #define TPM_ORD_Sign (60UL + TPM_PROTECTED_ORDINAL)

4.169.1.244 #define TPM_ORD_Startup (153UL + TPM_PROTECTED_ORDINAL)

4.169.1.245 #define TPM_ORD_StirRandom (71UL + TPM_PROTECTED_ORDINAL)

4.169.1.246 #define TPM_ORD_TakeOwnership (13UL + TPM_PROTECTED_ORDINAL)

4.169.1.247 #define TPM_ORD_Terminate_Handle (150UL + TPM_PROTECTED_ORDINAL)

4.169.1.248 #define TPM_ORD_TickStampBlob (242UL + TPM_PROTECTED_ORDINAL)

4.169.1.249 #define TPM_ORD_UnBind (30UL + TPM_PROTECTED_ORDINAL)

4.169.1.250 #define TPM_ORD_Unseal (24UL + TPM_PROTECTED_ORDINAL)

4.169.1.251 #define TPM_OWNER_KEYHANDLE 0x40000001

4.169.1.252 #define TPM_OWNER_SET TPM_BASE + 20

4.169.1.253 #define TPM_PER_NOWRITE TPM_BASE + 63

4.169.1.254 #define TPM_PID_ADCP 0x0004

4.169.1.255 #define TPM_PID_ADIP 0x0003

4.169.1.256 #define TPM_PID_OIAP 0x0001

4.169.1.257 #define TPM_PID_OSAP 0x0002

4.169.1.258 #define TPM_PID_OWNER 0x0005

4.169.1.259 #define TPM_PROTECTED_ORDINAL 0x00000000UL

4.169.1.260 #define TPM_PT_ASYM 0x01

4.169.1.261 #define TPM_PT_BIND 0x02

4.169.1.262 #define TPM_PT_MAINT 0x04

4.169.1.263 #define TPM_PT_MIGRATE 0x03

4.169.1.264 #define TPM_PT_SEAL 0x05

4.169.1.265 #define TPM_READ_ONLY TPM_BASE + 62

4.169.1.266 #define TPM_REQUIRES_SIGN TPM_BASE + 57

4.169.1.267 #define TPM_RESOURCEMISSING TPM_BASE + 74

4.169.1.268 #define TPM_RESOURCES TPM_BASE + 21

4.169.1.269 #define TPM_RT_AUTH 0x00000002

4.169.1.270 #define TPM_RT_CONTEXT 0x00000005

4.169.1.271 #define TPM_RT_COUNTER 0x00000006

4.169.1.272 #define TPM_RT_DAA_TPM 0x00000008

4.169.1.273 #define TPM_RT_DAA_V0 0x00000009

4.169.1.274 #define TPM_RT_DAA_V1 0x0000000A

4.169.1.275 #define TPM_RT_DELEGATE 0x00000007

4.169.1.276 #define TPM_RT_HASH 0x00000003

4.169.1.277 #define TPM_RT_KEY 0x00000001

TPM_ResourceTypes.

4.169.1.278 #define TPM_RT_TRANS 0x00000004

4.169.1.279 #define TPM_SHA_ERROR TPM_BASE + 27

4.169.1.280 #define TPM_SHA_THREAD TPM_BASE + 26

4.169.1.281 #define TPM_SHORTRANDOM TPM_BASE + 22

4.169.1.282 #define TPM_SIZE TPM_BASE + 23

4.169.1.283 #define TPM_SRK_KEYHANDLE 0x40000000

4.169.1.284 #define TPM_SS_NONE 0x0001

4.169.1.285 #define TPM_SS_RSASSAPKCS1v15_DER 0x0003

4.169.1.286 #define TPM_SS_RSASSAPKCS1v15_SHA1 0x0002

4.169.1.287 #define TPM_ST_CLEAR 0x0001

4.169.1.288 #define TPM_ST_DEACTIVATED 0x003

4.169.1.289 #define TPM_ST_STATE 0x0002

4.169.1.290 #define TPM_SUCCESS TPM_BASE

4.169.1.291 #define TPM_TAG_RQU_AUTH1_COMMAND 0x00c2

4.169.1.292 #define TPM_TAG_RQU_AUTH2_COMMAND 0x00c3

4.169.1.293 #define TPM_TAG_RQU_COMMAND 0x00c1

4.169.1.294 #define TPM_TAG_RSP_AUTH1_COMMAND 0x00c5

4.169.1.295 #define TPM_TAG_RSP_AUTH2_COMMAND 0x00c6

4.169.1.296 #define TPM_TAG_RSP_COMMAND 0x00c4

4.169.1.297 #define TPM_TRANSPORT_EXCLUSIVE TPM_BASE + 78

4.169.1.298 #define TPM_UNPROTECTED_ORDINAL 0x80000000UL

4.169.1.299 #define TPM_VENDOR_ERROR 0x00000400

4.169.1.300 #define TPM_VENDOR_ORDINAL 0x20000000UL

4.169.1.301 #define TPM_WRITE_LOCKED TPM_BASE + 65

4.169.1.302 #define TPM_WRONG_ENTITYTYPE TPM_BASE + 37

4.169.1.303 #define TPM_WRONGPCRVAL TPM_BASE + 24

4.169.1.304 #define TPMTRY(*s*, c)

Value:

```
if (c != TPM_SUCCESS) { \
                        status = s; \
                        printf("ERROR in %s at %s:%i code: %s.\n", __func__, __FILE__, __LINE__,
      tpm_get_error_name(status)); \
                        goto abort_egress; \
                     } else {\
                        status = c; \
                     }
```

4.169.1.305 #define TPMTRYRETURN(c)

Value:

```
do { status = c; \
                              if (status != TPM_SUCCESS) { \
                                fprintf(stderr, "ERROR in %s at %s:%i code: %s.\n", __func__, __FILE__,
      __LINE__, tpm_get_error_name(status)); \
                                goto abort_egress; \
                              } \
                        } while(0)
```

4.169.1.306 #define TRUE 0x01

4.169.1.307 #define TSC_ORD_PhysicalPresence (10UL + TPM_CONNECTION_ORDINAL)

4.169.2 Typedef Documentation

4.169.2.1 typedef unsigned char BOOL

4.169.2.2 typedef unsigned char BYTE

4.169.2.3 typedef struct pack_buf_t pack_buf_t

4.169.2.4 typedef struct pack_constbuf_t pack_constbuf_t

4.169.2.5 typedef TPM_HANDLE TCPA_ENCHANDLE

4.169.2.6 typedef TPM_HANDLE TCPA_ENTITYHANDLE

4.169.2.7 typedef TPM_HANDLE TCPA_HASHHANDLE

4.169.2.8 typedef TPM_HANDLE TCPA_HMACHANDLE

4.169.2.9 typedef struct TCS_AUTH TCS_AUTH

4.169.2.10 typedef UINT32 TCS_AUTHHANDLE

4.169.2.11 typedef UINT32 TCS_CONTEXT_HANDLE

4.169.2.12 typedef UINT32 TCS_KEY_HANDLE

4.169.2.13 typedef UINT32 TPM_ALGORITHM_ID

4.169.2.14 typedef BYTE TPM_AUTH_DATA_USAGE

4.169.2.15 typedef BYTE TPM_AUTHDATA[TPM_DIGEST_SIZE]

4.169.2.16 typedef TPM_HANDLE TPM_AUTHHANDLE

4.169.2.17 typedef struct TPM_BOUND_DATA TPM_BOUND_DATA

4.169.2.18 typedef UINT32 TPM_CAPABILITY_AREA

4.169.2.19 typedef TPM_DIGEST TPM_CHOSENID_HASH

4.169.2.20 typedef UINT32 TPM_COMMAND_CODE

4.169.2.21 typedef TPM_DIGEST TPM_COMPOSITE_HASH

4.169.2.22 typedef struct TPM_DIGEST TPM_DIGEST

4.169.2.23 typedef UINT32 TPM_DIRINDEX

4.169.2.24 typedef TPM_DIGEST TPM_DIRVALUE

4.169.2.25 typedef UINT16 TPM_ENC_SCHEME

4.169.2.26 typedef TPM_AUTHDATA TPM_ENCAUTH

4.169.2.27 typedef UINT16 TPM_ENTITY_TYPE

4.169.2.28 typedef UINT32 TPM_HANDLE

4.169.2.29 typedef TPM_DIGEST TPM_HMAC

4.169.2.30 typedef struct TPM_KEY TPM_KEY

4.169.2.31 typedef UINT32 TPM_KEY_FLAGS

4.169.2.32 typedef TPM_HANDLE TPM_KEY_HANDLE

4.169.2.33 typedef struct TPM_KEY_PARMS TPM_KEY_PARMS

4.169.2.34 typedef UINT16 TPM_KEY_USAGE

4.169.2.35 typedef UINT16 TPM_MIGRATE_SCHEME

4.169.2.36 typedef struct TPM_NONCE TPM_NONCE

4.169.2.37 typedef BYTE TPM_PAYLOAD_TYPE

4.169.2.38 typedef struct TPM_PCR_COMPOSITE TPM_PCR_COMPOSITE

4.169.2.39 typedef struct TPM_PCR_INFO TPM_PCR_INFO

4.169.2.40 typedef struct TPM_PCR_SELECTION TPM_PCR_SELECTION

4.169.2.41 typedef UINT32 TPM_PCRINDEX

4.169.2.42 typedef TPM_DIGEST TPM_PCRVALUE

4.169.2.43 typedef UINT16 TPM_PHYSICAL_PRESENCE

4.169.2.44 typedef UINT16 TPM_PROTOCOL_ID

4.169.2.45 typedef struct TPM_PUBKEY TPM_PUBKEY

4.169.2.46 typedef UINT32 TPM_RESOURCE_TYPE

4.169.2.47 typedef UINT32 TPM_RESULT

4.169.2.48 typedef struct TPM_RSA_KEY_PARMS TPM_RSA_KEY_PARMS

4.169.2.49 typedef TPM_AUTHDATA TPM_SECRET

4.169.2.50 typedef UINT16 TPM_SIG_SCHEME

4.169.2.51 typedef UINT16 TPM_STARTUP_TYPE

4.169.2.52 typedef struct TPM_STORE_PUBKEY TPM_STORE_PUBKEY

4.169.2.53 typedef struct TPM_STORED_DATA TPM_STORED_DATA

4.169.2.54 typedef UINT16 TPM_TAG

4.169.2.55 typedef struct TPM_VERSION TPM_VERSION

4.169.2.56 typedef unsigned short UINT16

4.169.2.57 typedef unsigned int UINT32

4.169.2.58 typedef unsigned long long UINT64

4.170 xen/tools/vtpm_manager/vtpmmgrtalk/vtpmmgrtalk.c File Reference

```
#include <unistd.h>
#include <fcntl.h>
#include <sys/file.h>
#include <stdio.h>
#include <errno.h>
#include <string.h>
#include <stdint.h>
#include "../manager/vtpm_manager.h"
```

Functions

- int main (int argc, char **argv)

4.170.1 Function Documentation

4.170.1.1 int main (int *argc*, char ** *argv*)

4.171 xen/tools/xenstore/xenstored_core.c File Reference

```
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/socket.h>
#include <sys/select.h>
#include <sys/un.h>
#include <sys/time.h>
#include <time.h>
#include <unistd.h>
#include <fcntl.h>
#include <stdbool.h>
#include <stdio.h>
#include <stdarg.h>
#include <stdlib.h>
#include <syslog.h>
#include <string.h>
#include <errno.h>
#include <dirent.h>
#include <getopt.h>
#include <signal.h>
#include <assert.h>
#include <setjmp.h>
#include "utils.h"
#include "list.h"
#include "talloc.h"
#include "xs_lib.h"
#include "xenstored_core.h"
#include "xenstored_watch.h"
#include "xenstored_transaction.h"
#include "xenstored_domain.h"
#include "xenctrl.h"
#include "tdb.h"
#include "hashtable.h"
```

Macros

- #define log(...)
- #define TDB_FLAGS 0

Functions

- LIST_HEAD (connections)
- TDB_CONTEXT * tdb_context (struct connection *conn)
- bool replace_tdb (const char *newname, TDB_CONTEXT *newtdb)
- void trace (const char *fmt,...)
- void trace_create (const void *data, const char *type)
- void trace_destroy (const void *data, const char *type)
- bool is_child (const char *child, const char *parent)
- struct node * get_node (struct connection *conn, const char *name, enum xs_perm_type perm)

- unsigned int get_strings (struct buffered_data ∗data, char ∗vec[], unsigned int num)
- void send_reply (struct connection ∗conn, enum xsd_sockmsg_type type, const void ∗data, unsigned int len)
- void send_ack (struct connection ∗conn, enum xsd_sockmsg_type type)
- void send_error (struct connection ∗conn, int error)
- bool is_valid_nodename (const char ∗node)
- char ∗ canonicalize (struct connection ∗conn, const char ∗node)
- bool check_event_node (const char ∗node)
- struct connection ∗ new_connection (connwritefn_t ∗write, connreadfn_t ∗read)
- void dump_conn (struct connection ∗conn)
- int main (int argc, char ∗argv[])

Variables

- xc_evtchn ∗ xce_handle
- int quota_nb_entry_per_domain = 1000
- int quota_nb_watch_per_domain = 128
- int quota_max_entry_size = 2048
- int quota_max_transaction = 10

4.171.1 Macro Definition Documentation

4.171.1.1 #define log(...)

Value:

```
do {                                    \
        char *s = talloc_asprintf(NULL, __VA_ARGS__);        \
        trace("%s\n", s);                    \
        syslog(LOG_ERR, "%s",  s);               \
        talloc_free(s);                      \
    } while (0)
```

4.171.1.2 #define TDB_FLAGS 0

4.171.2 Function Documentation

4.171.2.1 char∗ canonicalize (struct connection ∗ *conn,* const char ∗ *node*)

4.171.2.2 bool check_event_node (const char ∗ *node*)

4.171.2.3 void dump_conn (struct connection ∗ *conn*)

4.171.2.4 struct node∗ get_node (struct connection ∗ *conn,* const char ∗ *name,* enum xs_perm_type *perm*)

If it fails, returns NULL and sets errno.

4.171.2.5 unsigned int get_strings (struct buffered_data ∗ *data,* char ∗ *vec[],* unsigned int *num*)

Break input into vectors, return the number, fill in up to num of them. Always returns the actual number of nuls in the input. Stores the positions of the starts of the nul-terminated strings in vec. Callers who use this and then rely only on vec[] will ignore any data after the final nul.

4.171.2.6 bool is_child (const char ∗ *child,* const char ∗ *parent*)

Is child a subnode of parent, or equal?

4.171.2.7 bool is_valid_nodename (const char * *node*)

4.171.2.8 LIST_HEAD (connections)

4.171.2.9 int main (int *argc,* char * *argv[]*)

4.171.2.10 struct connection* new_connection (connwritefn_t * *write,* connreadfn_t * *read*)

4.171.2.11 bool replace_tdb (const char * *newname,* TDB_CONTEXT * *newtdb*)

4.171.2.12 void send_ack (struct connection * *conn,* enum xsd_sockmsg_type *type*)

Some routines (write, mkdir, etc) just need a non-error return

4.171.2.13 void send_error (struct connection * *conn,* int *error*)

4.171.2.14 void send_reply (struct connection * *conn,* enum xsd_sockmsg_type *type,* const void * *data,* unsigned int *len*)

4.171.2.15 TDB_CONTEXT* tdb_context (struct connection * *conn*)

4.171.2.16 void trace (const char * *fmt,* ...)

4.171.2.17 void trace_create (const void * *data,* const char * *type*)

4.171.2.18 void trace_destroy (const void * *data,* const char * *type*)

4.171.3 Variable Documentation

4.171.3.1 int quota_max_entry_size = 2048

4.171.3.2 int quota_max_transaction = 10

4.171.3.3 int quota_nb_entry_per_domain = 1000

4.171.3.4 int quota_nb_watch_per_domain = 128

4.171.3.5 xc_evtchn* xce_handle

4.172 xen/tools/xenstore/xenstored_domain.c File Reference

```
#include <stdio.h>
#include <sys/mman.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdarg.h>
#include <xenctrl.h>
#include "utils.h"
#include "talloc.h"
#include "xenstored_core.h"
#include "xenstored_domain.h"
#include "xenstored_transaction.h"
#include "xenstored_watch.h"
```

Data Structures

- struct domain

Macros

- #define DOMID_DOMT 1
- #define DOMID_DOMT 1

Functions

- void handle_event (void)
- bool domain_can_read (struct connection ∗conn)
- bool domain_is_unprivileged (struct connection ∗conn)
- bool domain_can_write (struct connection ∗conn)
- void do_introduce (struct connection ∗conn, struct buffered_data ∗in)
- void do_set_target (struct connection ∗conn, struct buffered_data ∗in)
- void do_release (struct connection ∗conn, const char ∗domid_str)
- void do_resume (struct connection ∗conn, const char ∗domid_str)
- void do_get_domain_path (struct connection ∗conn, const char ∗domid_str)
- void do_is_domain_introduced (struct connection ∗conn, const char ∗domid_str)
- const char ∗ get_implicit_path (const struct connection ∗conn)
- void restore_existing_connections (void)
- void domain_init (void)
- void domain_entry_inc (struct connection ∗conn, struct node ∗node)
- void domain_entry_dec (struct connection ∗conn, struct node ∗node)
- void domain_entry_fix (unsigned int domid, int num)
- int domain_entry (struct connection ∗conn)
- void domain_watch_inc (struct connection ∗conn)
- void domain_watch_dec (struct connection ∗conn)
- int domain_watch (struct connection ∗conn)

Variables

- xc_evtchn ∗ xce_handle = NULL

4.172.1 Macro Definition Documentation

4.172.1.1 #define DOMID_DOMT 1

4.172.1.2 #define DOMID_DOMT 1

4.172.2 Function Documentation

4.172.2.1 void do_get_domain_path (struct connection ∗ *conn,* const char ∗ *domid_str*)

4.172.2.2 void do_introduce (struct connection ∗ *conn,* struct buffered_data ∗ *in*)

domid, mfn, evtchn, path

4.172.2.3 void do_is_domain_introduced (struct connection * *conn,* const char * *domid_str*)

4.172.2.4 void do_release (struct connection * *conn,* const char * *domid_str*)

domid

4.172.2.5 void do_resume (struct connection * *conn,* const char * *domid_str*)

4.172.2.6 void do_set_target (struct connection * *conn,* struct buffered_data * *in*)

4.172.2.7 bool domain_can_read (struct connection * *conn*)

4.172.2.8 bool domain_can_write (struct connection * *conn*)

4.172.2.9 int domain_entry (struct connection * *conn*)

4.172.2.10 void domain_entry_dec (struct connection * *conn,* struct node * *node*)

4.172.2.11 void domain_entry_fix (unsigned int *domid,* int *num*)

4.172.2.12 void domain_entry_inc (struct connection * *conn,* struct node * *node*)

4.172.2.13 void domain_init (void)

4.172.2.14 bool domain_is_unprivileged (struct connection * *conn*)

4.172.2.15 int domain_watch (struct connection * *conn*)

4.172.2.16 void domain_watch_dec (struct connection * *conn*)

4.172.2.17 void domain_watch_inc (struct connection * *conn*)

4.172.2.18 const char* get_implicit_path (const struct connection * *conn*)

Returns the implicit path of a connection (only domains have this)

4.172.2.19 void handle_event (void)

4.172.2.20 void restore_existing_connections (void)

Restore existing connections.

4.172.3 Variable Documentation

4.172.3.1 xc_evtchn* xce_handle = NULL

4.173 xen/tools/xenstore/xenstored_transaction.c File Reference

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/wait.h>
#include <sys/time.h>
#include <time.h>
#include <assert.h>
#include <stdarg.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
#include "talloc.h"
#include "list.h"
#include "xenstored_transaction.h"
#include "xenstored_watch.h"
#include "xenstored_domain.h"
#include "xs_lib.h"
#include "utils.h"
```

Data Structures

- struct changed_node
- struct changed_domain
- struct transaction

Functions

- TDB_CONTEXT * tdb_transaction_context (struct transaction *trans)
- void add_change_node (struct transaction *trans, const char *node, bool recurse)
- struct transaction * transaction_lookup (struct connection *conn, uint32_t id)
- void do_transaction_start (struct connection *conn, struct buffered_data *in)
- void do_transaction_end (struct connection *conn, const char *arg)
- void transaction_entry_inc (struct transaction *trans, unsigned int domid)
- void transaction_entry_dec (struct transaction *trans, unsigned int domid)
- void conn_delete_all_transactions (struct connection *conn)

Variables

- int quota_max_transaction

4.173.1 Function Documentation

4.173.1.1 void add_change_node (struct transaction * *trans*, const char * *node*, bool *recurse*)

Callers get a change node (which can fail) and only commit after they've finished. This way they don't have to unwind eg. a write.

4.173.1.2 void conn_delete_all_transactions (struct connection * *conn*)

4.173.1.3 void do_transaction_end (struct connection * *conn,* const char * *arg*)

4.173.1.4 void do_transaction_start (struct connection * *conn,* struct buffered_data * *in*)

4.173.1.5 TDB_CONTEXT* tdb_transaction_context (struct **transaction** * *trans*)

Return tdb context to use for this connection.

4.173.1.6 void transaction_entry_dec (struct **transaction** * *trans,* unsigned int *domid*)

4.173.1.7 void transaction_entry_inc (struct **transaction** * *trans,* unsigned int *domid*)

4.173.1.8 struct **transaction*** transaction_lookup (struct connection * *conn,* uint32_t *id*)

4.173.2 Variable Documentation

4.173.2.1 int quota_max_transaction

4.174 xen/tools/xenvfsd/xenvfsd.c File Reference

```
#include <err.h>
#include <errno.h>
#include <error.h>
#include <fcntl.h>
#include <getopt.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <termios.h>
#include <unistd.h>
#include <sys/stat.h>
#include <sys/ioctl.h>
#include <sys/mman.h>
#include <assert.h>
#include <sys/types.h>
#include <libxlutil.h>
#include <xenctrl.h>
#include <xs.h>
#include <vfs_intf.h>
```

Macros

- #define _GNU_SOURCE
- #define DOMID_Z 0
- #define DOMID_T 1
- #define SET_OTHEREND(a)

Functions

- int read_arg (struct xs_handle *xsh, char **target, int num)

- int read_arg_int (struct xs_handle ∗xsh, int ∗itarget, int num)
- int xs_write_int (struct xs_handle ∗h, xs_transaction_t t, const char ∗path, int data)
- int read_to_gntref (xc_gnttab ∗xcg, int fd, uint32_t gref, size_t nbyte)
- int write_from_gntref (xc_gnttab ∗xcg, int fd, uint32_t gref, size_t nbyte)
- int stat_to_gntref (xc_gnttab ∗xcg, char ∗path, uint32_t gref)
- void ∗ map_gref_map (xc_gnttab ∗xcg, domid_t otherend, uint32_t gref, uint32_t ∗num)
- int chk_map_gref_map (uint8_t ∗∗p, size_t nbyte, xc_gnttab ∗xcg, domid_t otherend, uint32_t gref, uint32_t ∗num)
- domid_t read_dyn_domid (void)
- int main (int argc, char ∗∗argv)

Variables

- const char ∗ path_req = VFS_PATH "/req"
- const char ∗ path_rsp = VFS_PATH "/rsp"
- const char ∗ path_arg1 = VFS_PATH "/arg1"
- const char ∗ path_arg2 = VFS_PATH "/arg2"
- const char ∗ path_arg3 = VFS_PATH "/arg3"
- const char ∗ path_arg4 = VFS_PATH "/arg4"
- int dynamic = 0
- char ∗ dyn_file = NULL

4.174.1 Macro Definition Documentation

4.174.1.1 #define _GNU_SOURCE

4.174.1.2 #define DOMID_T 1

4.174.1.3 #define DOMID_Z 0

4.174.1.4 #define SET_OTHEREND(a)

Value:

```
do {                                                                    \
        if (!dynamic)                                                       \
            a = DOMID_T;                                                    \
        else                                                                \
            a = read_dyn_domid();                                           \
    } while(0)
```

4.174.2 Function Documentation

4.174.2.1 int chk_map_gref_map (uint8_t ∗∗ *p*, size_t *nbyte*, xc_gnttab ∗ *xcg*, domid_t *otherend*, uint32_t *gref*, uint32_t ∗ *num*)

4.174.2.2 int main (int *argc*, char ∗∗ *argv*)

4.174.2.3 void ∗ map_gref_map (xc_gnttab ∗ *xcg*, domid_t *otherend*, uint32_t *gref*, uint32_t ∗ *num*)

4.174.2.4 int read_arg (struct xs_handle ∗ *xsh*, char ∗∗ *target*, int *num*)

4.174.2.5 int read_arg_int (struct xs_handle ∗ *xsh*, int ∗ *itarget*, int *num*)

4.174.2.6 domid_t read_dyn_domid (void)

4.174.2.7 int read_to_gntref (xc_gnttab * *xcg*, int *fd*, uint32_t *gref*, size_t *nbyte*)

4.174.2.8 int stat_to_gntref (xc_gnttab * *xcg*, char * *path*, uint32_t *gref*)

4.174.2.9 int write_from_gntref (xc_gnttab * *xcg*, int *fd*, uint32_t *gref*, size_t *nbyte*)

4.174.2.10 int xs_write_int (struct xs_handle * *h*, xs_transaction_t *t*, const char * *path*, int *data*)

4.174.3 Variable Documentation

4.174.3.1 char* dyn_file = NULL

4.174.3.2 int dynamic = 0

4.174.3.3 const char * path_arg1 = VFS_PATH "/arg1"

4.174.3.4 const char * path_arg2 = VFS_PATH "/arg2"

4.174.3.5 const char * path_arg3 = VFS_PATH "/arg3"

4.174.3.6 const char * path_arg4 = VFS_PATH "/arg4"

4.174.3.7 const char* path_req = VFS_PATH "/req"

4.174.3.8 const char * path_rsp = VFS_PATH "/rsp"

4.175 xen/xen/arch/x86/domain_build.c File Reference

```
#include <xen/config.h>
#include <xen/init.h>
#include <xen/lib.h>
#include <xen/ctype.h>
#include <xen/sched.h>
#include <xen/sched-if.h>
#include <xen/smp.h>
#include <xen/delay.h>
#include <xen/event.h>
#include <xen/console.h>
#include <xen/kernel.h>
#include <xen/domain.h>
#include <xen/version.h>
#include <xen/iocap.h>
#include <xen/bitops.h>
#include <xen/compat.h>
#include <xen/libelf.h>
#include <asm/regs.h>
#include <asm/system.h>
#include <asm/io.h>
#include <asm/processor.h>
#include <asm/desc.h>
#include <asm/i387.h>
#include <asm/paging.h>
#include <asm/p2m.h>
#include <asm/e820.h>
#include <asm/acpi.h>
#include <asm/setup.h>
#include <asm/bzimage.h>
#include <asm/io_apic.h>
#include <public/version.h>
```

Macros

- #define CAAS_DEVELOP
- #define round_pgup(_p) (((_p)+(PAGE_SIZE-1))&PAGE_MASK)
- #define round_pgdown(_p) ((_p)&PAGE_MASK)
- #define REVERSE_START ((v_end - v_start) >> PAGE_SHIFT)
- #define pfn (nr_pages - 1 - (pfn - (alloc_epfn - alloc_spfn)))
- #define TPM_LOC0 0xFED40000 >> PAGE_SHIFT
- #define TPM_LOCALL 0xFED44000 >> PAGE_SHIFT

Functions

- integer_param ("caas_enforcing", caas_enforcing)
- custom_param ("dom0_mem", parse_dom0_mem)
- integer_param ("dom0_max_vcpus", opt_dom0_max_vcpus)
- integer_param ("domt_max_vcpus", opt_domt_max_vcpus)
- struct vcpu *__init alloc_dom0_vcpu0 (void)
- struct vcpu *__init alloc_domt_vcpu0 (void)
- boolean_param ("dom0_shadow", opt_dom0_shadow)
- string_param ("dom0_ioports_disable", opt_dom0_ioports_disable)

- int __init construct_dom (enum domtype t, struct domain *d, const module_t *image, unsigned long image_headroom, module_t *initrd, void *(*bootstrap_map)(const module_t *), char *cmdline)

4.175.1 Macro Definition Documentation

4.175.1.1 #define CAAS_DEVELOP

4.175.1.2 #define pfn (nr_pages - 1 - (pfn - (alloc_epfn - alloc_spfn)))

4.175.1.3 #define REVERSE_START ((v_end - v_start) >> PAGE_SHIFT)

4.175.1.4 #define round_pgdown(_p) ((_p)&PAGE_MASK)

4.175.1.5 #define round_pgup(_p) (((_p)+(PAGE_SIZE-1))&PAGE_MASK)

4.175.1.6 #define TPM_LOC0 0xFED40000 >> PAGE_SHIFT

4.175.1.7 #define TPM_LOCALL 0xFED44000 >> PAGE_SHIFT

4.175.2 Function Documentation

4.175.2.1 struct vcpu* __init alloc_dom0_vcpu0 (void)

4.175.2.2 struct vcpu* __init alloc_domt_vcpu0 (void)

4.175.2.3 boolean_param ("dom0_shadow" , opt_dom0_shadow)

4.175.2.4 int __init construct_dom (enum domtype *t*, struct domain * *d*, const module_t * *image*, unsigned long *image_headroom*, module_t * *initrd*, void *(*)(const module_t *) *bootstrap_map*, char * *cmdline*)

4.175.2.5 custom_param ("dom0_mem" , parse_dom0_mem)

4.175.2.6 integer_param ("caas_enforcing" , caas_enforcing)

4.175.2.7 integer_param ("dom0_max_vcpus" , opt_dom0_max_vcpus)

4.175.2.8 integer_param ("domt_max_vcpus" , opt_domt_max_vcpus)

4.175.2.9 string_param ("dom0_ioports_disable" , opt_dom0_ioports_disable)

4.176 xen/xen/arch/x86/setup.c File Reference

```
#include <xen/config.h>
#include <xen/init.h>
#include <xen/lib.h>
#include <xen/sched.h>
#include <xen/sched-if.h>
#include <xen/domain.h>
#include <xen/serial.h>
#include <xen/softirq.h>
#include <xen/acpi.h>
#include <xen/console.h>
#include <xen/trace.h>
#include <xen/multiboot.h>
#include <xen/domain_page.h>
#include <xen/version.h>
#include <xen/gdbstub.h>
#include <xen/percpu.h>
#include <xen/hypercall.h>
#include <xen/keyhandler.h>
#include <xen/numa.h>
#include <xen/rcupdate.h>
#include <xen/vga.h>
#include <xen/dmi.h>
#include <xen/nodemask.h>
#include <public/version.h>
#include <asm/bitops.h>
#include <asm/smp.h>
#include <asm/processor.h>
#include <asm/mpspec.h>
#include <asm/apic.h>
#include <asm/desc.h>
#include <asm/paging.h>
#include <asm/e820.h>
#include <xsm/acm/acm_hooks.h>
#include <xen/kexec.h>
#include <asm/edd.h>
#include <xsm/xsm.h>
#include <asm/tboot.h>
#include <asm/bzimage.h>
#include <asm/mach-generic/mach_apic.h>
#include <asm/setup.h>
#include <xen/cpu.h>
```

Data Structures

- struct boot_video_info

Macros

- #define EARLY_FAIL(f, a...)
- #define BOOTSTRAP_MAP_BASE (16UL << 20)
- #define BOOTSTRAP_MAP_LIMIT (1UL << L3_PAGETABLE_SHIFT)

Functions

- boolean_param ("nosmp", opt_nosmp)
- integer_param ("maxcpus", max_cpus)
- boolean_param ("watchdog", opt_watchdog)
- invbool_param ("smep", disable_smep)
- custom_param ("acpi", parse_acpi_param)
- boolean_param ("acpi_skip_timer_override", acpi_skip_timer_override)
- boolean_param ("noapic", skip_ioapic_setup)
- boolean_param ("cpuidle", xen_cpuidle)
- DEFINE_PER_CPU (struct tss_struct, init_tss)
- char __attribute__ ((__section__(".bss.stack_aligned")))
- unsigned long __init initial_images_nrpages (void)
- void __init discard_initial_images (void)
- void __devinit srat_detect_node (int cpu)
- void set_pdx_range (unsigned long smfn, unsigned long emfn)
- void init_done (void)
- void __init __start_xen (unsigned long mbi_p)
- void arch_get_xen_caps (xen_capabilities_info_t *info)
- int xen_in_range (unsigned long mfn)

Variables

- u16 boot_edid_caps
- u8 boot_edid_info [128]
- struct boot_video_info boot_vid_info
- s8 __read_mostly xen_cpuidle = -1
- bool_t __read_mostly early_boot = 1
- cpumask_t __read_mostly cpu_present_map
- unsigned long __read_mostly xen_phys_start
- unsigned long __read_mostly mmu_cr4_features = X86_CR4_PSE | X86_CR4_PGE | X86_CR4_PAE
- bool_t __initdata acpi_disabled
- bool_t __initdata acpi_force
- char __init_begin []
- char __init_end []
- char __bss_start []

4.176.1 Macro Definition Documentation

4.176.1.1 #define BOOTSTRAP_MAP_BASE (16UL << 20)

4.176.1.2 #define BOOTSTRAP_MAP_LIMIT (1UL << L3_PAGETABLE_SHIFT)

4.176.1.3 #define EARLY_FAIL(*f, a...*)

Value:

```
do {                    \
    printk( f , ## a );                         \
    for ( ; ; ) halt();                         \
} while (0)
```

4.176.2 Function Documentation

4.176.2.1 char __attribute__ ((__section__(".bss.stack_aligned")))

4.176.2.2 void __init __start_xen (unsigned long *mbi_p*)

4.176.2.3 void arch_get_xen_caps (xen_capabilities_info_t * *info*)

4.176.2.4 boolean_param ("nosmp" , opt_nosmp)

4.176.2.5 boolean_param ("watchdog" , opt_watchdog)

4.176.2.6 boolean_param ("acpi_skip_timer_override" , acpi_skip_timer_override)

4.176.2.7 boolean_param ("noapic" , skip_ioapic_setup)

4.176.2.8 boolean_param ("cpuidle" , xen_cpuidle)

4.176.2.9 custom_param ("acpi" , parse_acpi_param)

4.176.2.10 DEFINE_PER_CPU (struct tss_struct , init_tss)

4.176.2.11 void __init discard_initial_images (void)

4.176.2.12 void init_done (void)

4.176.2.13 unsigned long __init initial_images_nrpages (void)

4.176.2.14 integer_param ("maxcpus" , max_cpus)

4.176.2.15 invbool_param ("smep" , disable_smep)

4.176.2.16 void set_pdx_range (unsigned long *smfn,* unsigned long *emfn*)

4.176.2.17 void __devinit srat_detect_node (int *cpu*)

4.176.2.18 int xen_in_range (unsigned long *mfn*)

4.176.3 Variable Documentation

4.176.3.1 char __bss_start[ ]

4.176.3.2 char __init_begin[ ]

4.176.3.3 char __init_end[ ]

4.176.3.4 bool_t __initdata acpi_disabled

4.176.3.5 bool_t __initdata acpi_force

4.176.3.6 u16 boot_edid_caps

4.176.3.7 u8 boot_edid_info[128]

4.176.3.8 struct boot_video_info boot_vid_info

4.176.3.9 cpumask_t __read_mostly cpu_present_map

4.176.3.10 bool_t __read_mostly early_boot = 1

4.176.3.11 unsigned long __read_mostly mmu_cr4_features = X86_CR4_PSE | X86_CR4_PGE | X86_CR4_PAE

4.176.3.12 s8 __read_mostly xen_cpuidle = -1

4.176.3.13 unsigned long __read_mostly xen_phys_start

4.177 xen/xen/common/domain.c File Reference

```
#include <xen/config.h>
#include <xen/compat.h>
#include <xen/init.h>
#include <xen/lib.h>
#include <xen/ctype.h>
#include <xen/errno.h>
#include <xen/sched.h>
#include <xen/domain.h>
#include <xen/mm.h>
#include <xen/event.h>
#include <xen/time.h>
#include <xen/console.h>
#include <xen/softirq.h>
#include <xen/tasklet.h>
#include <xen/domain_page.h>
#include <xen/rangeset.h>
#include <xen/guest_access.h>
#include <xen/hypercall.h>
#include <xen/delay.h>
#include <xen/shutdown.h>
#include <xen/percpu.h>
#include <xen/multicall.h>
#include <xen/rcupdate.h>
#include <xen/wait.h>
#include <acpi/cpufreq/cpufreq.h>
#include <asm/debugger.h>
#include <public/sched.h>
#include <public/sysctl.h>
#include <public/vcpu.h>
#include <xsm/xsm.h>
#include <xen/trace.h>
#include <xen/tmem.h>
```

Data Structures

- struct migrate_info

Macros

- #define DOMAIN_HASH_SIZE 256
- #define DOMAIN_HASH(_id) ((int)(_id)&(DOMAIN_HASH_SIZE-1))

Functions

- boolean_param ("dom0_vcpus_pin", opt_dom0_vcpus_pin)
- custom_param ("cpufreq", setup_cpufreq_option)
- DEFINE_SPINLOCK (domlist_update_lock)
- DEFINE_RCU_READ_LOCK (domlist_read_lock)
- int current_domain_id (void)
- struct vcpu * alloc_vcpu (struct domain *d, unsigned int vcpu_id, unsigned int cpu_id)
- custom_param ("extra_guest_irqs", parse_extra_guest_irqs)
- struct domain * domain_create (domid_t domid, unsigned int domcr_flags, ssidref_t ssidref)
- void domain_update_node_affinity (struct domain *d)
- struct domain * get_domain_by_id (domid_t dom)
- struct domain * rcu_lock_domain_by_id (domid_t dom)
- int rcu_lock_target_domain_by_id (domid_t dom, struct domain **d)
- int rcu_lock_remote_target_domain_by_id (domid_t dom, struct domain **d)
- int domain_kill (struct domain *d)
- void __domain_crash (struct domain *d)
- void __domain_crash_synchronous (void)
- void domain_shutdown (struct domain *d, u8 reason)
- void domain_resume (struct domain *d)
- int vcpu_start_shutdown_deferral (struct vcpu *v)
- void vcpu_end_shutdown_deferral (struct vcpu *v)
- void domain_pause_for_debugger (void)
- void domain_destroy (struct domain *d)
- void vcpu_pause (struct vcpu *v)
- void vcpu_pause_nosync (struct vcpu *v)
- void vcpu_unpause (struct vcpu *v)
- void domain_pause (struct domain *d)
- void domain_unpause (struct domain *d)
- void domain_pause_by_systemcontroller (struct domain *d)
- void domain_unpause_by_systemcontroller (struct domain *d)
- int boot_vcpu (struct domain *d, int vcpuid, vcpu_guest_context_u ctxt)
- void vcpu_reset (struct vcpu *v)
- long do_vcpu_op (int cmd, int vcpuid, XEN_GUEST_HANDLE(void) arg)
- long vm_assist (struct domain *p, unsigned int cmd, unsigned int type)
- int continue_hypercall_on_cpu (unsigned int cpu, long(*func)(void *data), void *data)

Variables

- unsigned int xen_processor_pmbits = XEN_PROCESSOR_PM_PX
- enum cpufreq_controller cpufreq_controller = FREQCTL_xen
- struct domain * domain_list
- struct domain * dom0
- struct domain * domt
- struct vcpu *idle_vcpu[NR_CPUS] __read_mostly
- vcpu_info_t dummy_vcpu_info

4.177.1 Macro Definition Documentation

4.177.1.1 #define DOMAIN_HASH(_id) ((int)(_id)&(DOMAIN_HASH_SIZE-1))

4.177.1.2 #define DOMAIN_HASH_SIZE 256

4.177.2 Function Documentation

4.177.2.1 void __domain_crash (struct domain * d)

Mark specified domain as crashed. This function always returns, even if the caller is the specified domain. The domain is not synchronously descheduled from any processor.

4.177.2.2 void __domain_crash_synchronous (void)

Mark current domain as crashed and synchronously deschedule from the local processor. This function never returns.

4.177.2.3 struct vcpu* alloc_vcpu (struct domain * d, unsigned int vcpu_id, unsigned int cpu_id)

4.177.2.4 boolean_param ("dom0_vcpus_pin" , opt_dom0_vcpus_pin)

4.177.2.5 int boot_vcpu (struct domain * d, int vcpuid, vcpu_guest_context_u ctxt)

4.177.2.6 int continue_hypercall_on_cpu (unsigned int cpu, long(*)(void *data) func, void * data)

Continue the current hypercall via func(data) on specified cpu. If this function returns 0 then the function is guaranteed to run at some point in the future. If this function returns an error code then the function has not been and will not be executed.

4.177.2.7 int current_domain_id (void)

4.177.2.8 custom_param ("cpufreq" , setup_cpufreq_option)

4.177.2.9 custom_param ("extra_guest_irqs" , parse_extra_guest_irqs)

4.177.2.10 DEFINE_RCU_READ_LOCK (domlist_read_lock)

4.177.2.11 DEFINE_SPINLOCK (domlist_update_lock)

4.177.2.12 long do_vcpu_op (int cmd, int vcpuid, XEN_GUEST_HANDLE(void) arg)

4.177.2.13 struct domain* domain_create (domid_t domid, unsigned int domcr_flags, ssidref_t ssidref)

4.177.2.14 void domain_destroy (struct domain * d)

Release resources belonging to task .

4.177.2.15 int domain_kill (struct domain * d)

4.177.2.16 void domain_pause (struct domain * d)

4.177.2.17 void domain_pause_by_systemcontroller (struct domain * d)

4.177.2.18 void domain_pause_for_debugger (void)

4.177.2.19 void domain_resume (struct domain * d)

4.177.2.20 void domain_shutdown (struct domain * d, u8 reason)

4.177.2.21 void domain_unpause (struct domain * d)

4.177.2.22 void domain_unpause_by_systemcontroller (struct domain * d)

4.177.2.23 void domain_update_node_affinity (struct domain * d)

4.177.2.24 struct domain* get_domain_by_id (domid_t dom)

4.177.2.25 struct domain* rcu_lock_domain_by_id (domid_t dom)

rcu_lock_domain_by_id() is more efficient than get_domain_by_id(). This is the preferred function if the returned domain reference is short lived, but it cannot be used if the domain reference needs to be kept beyond the current scope (e.g., across a softirq). The returned domain reference must be discarded using rcu_unlock_domain().

4.177.2.26 int rcu_lock_remote_target_domain_by_id (domid_t dom, struct domain ** d)

As rcu_lock_target_domain_by_id(), but will fail EPERM rather than resolve to local domain. Successful return always resolves to a remote domain that the local domain is privileged to control.

4.177.2.27 int rcu_lock_target_domain_by_id (domid_t dom, struct domain ** d)

As above function, but accounts for current domain context:

- Translates target DOMID_SELF into caller's domain id; and
- Checks that caller has permission to act on the target domain.

4.177.2.28 void vcpu_end_shutdown_deferral (struct vcpu * v)

4.177.2.29 void vcpu_pause (struct vcpu * v)

4.177.2.30 void vcpu_pause_nosync (struct vcpu * v)

4.177.2.31 void vcpu_reset (struct vcpu * v)

4.177.2.32 int vcpu_start_shutdown_deferral (struct vcpu * v)

4.177.2.33 void vcpu_unpause (struct vcpu * v)

4.177.2.34 long vm_assist (struct domain * p, unsigned int cmd, unsigned int type)

4.177.3 Variable Documentation

4.177.3.1 struct vcpu* idle_vcpu [NR_CPUS] __read_mostly

4.177.3.2 enum cpufreq_controller cpufreq_controller = FREQCTL_xen

4.177.3.3 struct domain* dom0

4.177.3.4 struct domain* domain_list

4.177.3.5 struct domain* domt

4.177.3.6 vcpu_info_t dummy_vcpu_info

4.177.3.7 unsigned int xen_processor_pmbits = XEN_PROCESSOR_PM_PX

4.178 xen/xen/common/domctl.c File Reference

```
#include <xen/config.h>
#include <xen/types.h>
#include <xen/lib.h>
#include <xen/mm.h>
#include <xen/sched.h>
#include <xen/sched-if.h>
#include <xen/domain.h>
#include <xen/event.h>
#include <xen/domain_page.h>
#include <xen/trace.h>
#include <xen/console.h>
#include <xen/iocap.h>
#include <xen/rcupdate.h>
#include <xen/guest_access.h>
#include <xen/bitmap.h>
#include <xen/paging.h>
#include <asm/current.h>
#include <public/domctl.h>
#include <xsm/xsm.h>
```

Functions

- long arch_do_domctl (struct xen_domctl *op, XEN_GUEST_HANDLE(xen_domctl_t) u_domctl)
- int cpumask_to_xenctl_cpumap (struct xenctl_cpumap *xenctl_cpumap, cpumask_t *cpumask)
- int xenctl_cpumap_to_cpumask (cpumask_t *cpumask, struct xenctl_cpumap *xenctl_cpumap)
- void getdomaininfo (struct domain *d, struct xen_domctl_getdomaininfo *info)
- bool_t domctl_lock_acquire (void)
- void domctl_lock_release (void)
- long do_domctl (XEN_GUEST_HANDLE(xen_domctl_t) u_domctl)

4.178.1 Function Documentation

4.178.1.1 long arch_do_domctl (struct xen_domctl * *op*, XEN_GUEST_HANDLE(xen_domctl_t) *u_domctl*)

4.178.1.2 int cpumask_to_xenctl_cpumap (struct xenctl_cpumap * *xenctl_cpumap*, cpumask_t * *cpumask*)

4.178.1.3 long do_domctl (XEN_GUEST_HANDLE(xen_domctl_t) *u_domctl*)

4.178.1.4 bool_t domctl_lock_acquire (void)

4.178.1.5 void domctl_lock_release (void)

4.178.1.6 void getdomaininfo (struct domain * *d*, struct xen_domctl_getdomaininfo * *info*)

4.178.1.7 int xenctl_cpumap_to_cpumask (cpumask_t * *cpumask,* struct xenctl_cpumap * *xenctl_cpumap*)

4.179 xen/xen/common/memory.c File Reference

```
#include <xen/config.h>
#include <xen/types.h>
#include <xen/lib.h>
#include <xen/mm.h>
#include <xen/perfc.h>
#include <xen/sched.h>
#include <xen/event.h>
#include <xen/paging.h>
#include <xen/iocap.h>
#include <xen/guest_access.h>
#include <xen/hypercall.h>
#include <xen/errno.h>
#include <xen/tmem.h>
#include <xen/tmem_xen.h>
#include <asm/current.h>
#include <asm/hardirq.h>
#include <xen/numa.h>
#include <public/memory.h>
#include <xsm/xsm.h>
#include <xen/trace.h>
```

Data Structures

- struct memop_args

Functions

- int guest_remove_page (struct domain *d, unsigned long gmfn)
- long do_memory_op (unsigned long cmd, XEN_GUEST_HANDLE(void) arg)

4.179.1 Function Documentation

4.179.1.1 long do_memory_op (unsigned long *cmd,* XEN_GUEST_HANDLE(void) *arg*)

4.179.1.2 int guest_remove_page (struct domain * *d,* unsigned long *gmfn*)

4.180 xen/xen/include/asm-x86/setup.h File Reference

```
#include <xen/multiboot.h>
```

Enumerations

- enum domtype { DOM0, DOMT }

Functions

- void init_done (void)
- void early_cpu_init (void)
- void early_time_init (void)
- void early_page_fault (void)
- int intel_cpu_init (void)
- int amd_init_cpu (void)
- int cyrix_init_cpu (void)
- int nsc_init_cpu (void)
- int centaur_init_cpu (void)
- int transmeta_init_cpu (void)
- void numa_initmem_init (unsigned long start_pfn, unsigned long end_pfn)
- void arch_init_memory (void)
- void subarch_init_memory (void)
- void init_IRQ (void)
- void vesa_init (void)
- void vesa_mtrr_init (void)
- int construct_dom (enum domtype t, struct domain *d, const module_t *kernel, unsigned long kernel_headroom, module_t *initrd, void *(*bootstrap_map)(const module_t *), char *cmdline)
- unsigned long initial_images_nrpages (void)
- void discard_initial_images (void)

Variables

- bool_t early_boot
- unsigned long xenheap_initial_phys_start

4.180.1 Enumeration Type Documentation

4.180.1.1 enum domtype

Enumerator

DOM0

DOMT

4.180.2 Function Documentation

4.180.2.1 int amd_init_cpu (void)

4.180.2.2 void arch_init_memory (void)

4.180.2.3 int centaur_init_cpu (void)

4.180.2.4 int construct_dom (enum domtype *t*, struct domain * *d*, const module_t * *kernel*, unsigned long *kernel_headroom*, module_t * *initrd*, void *(*)(const module_t *) *bootstrap_map*, char * *cmdline*)

4.180.2.5 int cyrix_init_cpu (void)

4.180.2.6 void discard_initial_images (void)

4.180.2.7 void early_cpu_init (void)

4.180.2.8 void early_page_fault (void)

4.180.2.9 void early_time_init (void)

4.180.2.10 void init_done (void)

4.180.2.11 void init_IRQ (void)

4.180.2.12 unsigned long initial_images_nrpages (void)

4.180.2.13 int intel_cpu_init (void)

4.180.2.14 int nsc_init_cpu (void)

4.180.2.15 void numa_initmem_init (unsigned long *start_pfn,* unsigned long *end_pfn*)

4.180.2.16 void subarch_init_memory (void)

4.180.2.17 int transmeta_init_cpu (void)

4.180.2.18 void vesa_init (void)

4.180.2.19 void vesa_mtrr_init (void)

4.180.3 Variable Documentation

4.180.3.1 bool_t early_boot

4.180.3.2 unsigned long xenheap_initial_phys_start

4.181 xen/xen/include/public/domctl.h File Reference

```
#include "xen.h"
#include "grant_table.h"
```

Data Structures

- struct xen_domctl_createdomain
- struct xen_domctl_getdomaininfo
- struct xen_domctl_getmemlist
- struct xen_domctl_getpageframeinfo
- struct xen_domctl_getpageframeinfo2
- struct xen_domctl_getpageframeinfo3
- struct xen_domctl_shadow_op_stats
- struct xen_domctl_shadow_op
- struct xen_domctl_max_mem
- struct xen_domctl_vcpucontext
- struct xen_domctl_getvcpuinfo
- struct xen_domctl_vcpuaffinity
- struct xen_domctl_max_vcpus
- struct xen_domctl_scheduler_op
- struct xen_domctl_setdomainhandle
- struct xen_domctl_setdebugging

- struct xen_domctl_irq_permission
- struct xen_domctl_iomem_permission
- struct xen_domctl_ioport_permission
- struct xen_domctl_hypercall_init
- struct xen_domctl_arch_setup
- struct xen_domctl_settimeoffset
- struct xen_domctl_hvmcontext
- struct xen_domctl_address_size
- struct xen_domctl_real_mode_area
- struct xen_domctl_sendtrigger
- struct xen_domctl_assign_device
- struct xen_domctl_get_device_group
- struct xen_domctl_bind_pt_irq
- struct xen_domctl_memory_mapping
- struct xen_domctl_ioport_mapping
- struct xen_domctl_pin_mem_cacheattr
- struct xen_domctl_ext_vcpucontext
- struct xen_domctl_set_opt_feature
- struct xen_domctl_set_target
- struct xen_domctl_subscribe
- struct xen_domctl_debug_op
- struct xen_domctl_hvmcontext_partial
- struct xen_domctl_disable_migrate
- struct xen_guest_tsc_info
- struct xen_domctl_tsc_info
- struct xen_domctl_gdbsx_memio
- struct xen_domctl_gdbsx_pauseunp_vcpu
- struct xen_domctl_gdbsx_domstatus
- struct xen_domctl_mem_event_op
- struct xen_domctl_mem_sharing_op
- struct xen_domctl_set_access_required
- struct xen_domctl

Macros

- #define XEN_DOMCTL_INTERFACE_VERSION 0x00000007
- #define _XEN_DOMCTL_CDF_hvm_guest 0
- #define XEN_DOMCTL_CDF_hvm_guest (1U<<_XEN_DOMCTL_CDF_hvm_guest)
- #define _XEN_DOMCTL_CDF_hap 1
- #define XEN_DOMCTL_CDF_hap (1U<<_XEN_DOMCTL_CDF_hap)
- #define _XEN_DOMCTL_CDF_s3_integrity 2
- #define XEN_DOMCTL_CDF_s3_integrity (1U<<_XEN_DOMCTL_CDF_s3_integrity)
- #define _XEN_DOMCTL_CDF_oos_off 3
- #define XEN_DOMCTL_CDF_oos_off (1U<<_XEN_DOMCTL_CDF_oos_off)
- #define _XEN_DOMCTL_CDF_priv 5
- #define XEN_DOMCTL_CDF_priv (1U<<_XEN_DOMCTL_CDF_priv)
- #define _XEN_DOMINF_dying 0
- #define XEN_DOMINF_dying (1U<<_XEN_DOMINF_dying)
- #define _XEN_DOMINF_hvm_guest 1
- #define XEN_DOMINF_hvm_guest (1U<<_XEN_DOMINF_hvm_guest)
- #define _XEN_DOMINF_shutdown 2
- #define XEN_DOMINF_shutdown (1U<<_XEN_DOMINF_shutdown)
- #define _XEN_DOMINF_paused 3
- #define XEN_DOMINF_paused (1U<<_XEN_DOMINF_paused)

- #define _XEN_DOMINF_blocked 4
- #define XEN_DOMINF_blocked (1U<<_XEN_DOMINF_blocked)
- #define _XEN_DOMINF_running 5
- #define XEN_DOMINF_running (1U<<_XEN_DOMINF_running)
- #define _XEN_DOMINF_debugged 6
- #define XEN_DOMINF_debugged (1U<<_XEN_DOMINF_debugged)
- #define XEN_DOMINF_shutdownmask 255
- #define XEN_DOMINF_shutdownshift 16
- #define XEN_DOMCTL_PFINFO_LTAB_SHIFT 28
- #define XEN_DOMCTL_PFINFO_NOTAB (0x0U<<28)
- #define XEN_DOMCTL_PFINFO_L1TAB (0x1U<<28)
- #define XEN_DOMCTL_PFINFO_L2TAB (0x2U<<28)
- #define XEN_DOMCTL_PFINFO_L3TAB (0x3U<<28)
- #define XEN_DOMCTL_PFINFO_L4TAB (0x4U<<28)
- #define XEN_DOMCTL_PFINFO_LTABTYPE_MASK (0x7U<<28)
- #define XEN_DOMCTL_PFINFO_LPINTAB (0x1U<<31)
- #define XEN_DOMCTL_PFINFO_XTAB (0xfU<<28) /* invalid page */
- #define XEN_DOMCTL_PFINFO_PAGEDTAB (0x8U<<28)
- #define XEN_DOMCTL_PFINFO_LTAB_MASK (0xfU<<28)
- #define XEN_DOMCTL_SHADOW_OP_OFF 0
- #define XEN_DOMCTL_SHADOW_OP_ENABLE 32
- #define XEN_DOMCTL_SHADOW_OP_CLEAN 11
- #define XEN_DOMCTL_SHADOW_OP_PEEK 12
- #define XEN_DOMCTL_SHADOW_OP_GET_ALLOCATION 30
- #define XEN_DOMCTL_SHADOW_OP_SET_ALLOCATION 31
- #define XEN_DOMCTL_SHADOW_OP_ENABLE_TEST 1
- #define XEN_DOMCTL_SHADOW_OP_ENABLE_LOGDIRTY 2
- #define XEN_DOMCTL_SHADOW_OP_ENABLE_TRANSLATE 3
- #define XEN_DOMCTL_SHADOW_ENABLE_REFCOUNT (1 << 1)
- #define XEN_DOMCTL_SHADOW_ENABLE_LOG_DIRTY (1 << 2)
- #define XEN_DOMCTL_SHADOW_ENABLE_TRANSLATE (1 << 3)
- #define XEN_DOMCTL_SHADOW_ENABLE_EXTERNAL (1 << 4)
- #define XEN_SCHEDULER_SEDF 4
- #define XEN_SCHEDULER_CREDIT 5
- #define XEN_SCHEDULER_CREDIT2 6
- #define XEN_SCHEDULER_ARINC653 7
- #define XEN_DOMCTL_SCHEDOP_putinfo 0
- #define XEN_DOMCTL_SCHEDOP_getinfo 1
- #define _XEN_DOMAINSETUP_hvm_guest 0
- #define XEN_DOMAINSETUP_hvm_guest (1UL<<_XEN_DOMAINSETUP_hvm_guest)
- #define _XEN_DOMAINSETUP_query 1 /* Get parameters (for save) */
- #define XEN_DOMAINSETUP_query (1UL<<_XEN_DOMAINSETUP_query)
- #define _XEN_DOMAINSETUP_sioemu_guest 2
- #define XEN_DOMAINSETUP_sioemu_guest (1UL<<_XEN_DOMAINSETUP_sioemu_guest)
- #define XEN_DOMCTL_SENDTRIGGER_NMI 0
- #define XEN_DOMCTL_SENDTRIGGER_RESET 1
- #define XEN_DOMCTL_SENDTRIGGER_INIT 2
- #define XEN_DOMCTL_SENDTRIGGER_POWER 3
- #define XEN_DOMCTL_SENDTRIGGER_SLEEP 4
- #define DPCI_ADD_MAPPING 1
- #define DPCI_REMOVE_MAPPING 0
- #define XEN_DOMCTL_MEM_CACHEATTR_UC 0
- #define XEN_DOMCTL_MEM_CACHEATTR_WC 1
- #define XEN_DOMCTL_MEM_CACHEATTR_WT 4
- #define XEN_DOMCTL_MEM_CACHEATTR_WP 5

- #define XEN_DOMCTL_MEM_CACHEATTR_WB 6
- #define XEN_DOMCTL_MEM_CACHEATTR_UCM 7
- #define XEN_DOMCTL_DEBUG_OP_SINGLE_STEP_OFF 0
- #define XEN_DOMCTL_DEBUG_OP_SINGLE_STEP_ON 1
- #define XEN_DOMCTL_MEM_EVENT_OP_ENABLE 0
- #define XEN_DOMCTL_MEM_EVENT_OP_DISABLE 1
- #define XEN_DOMCTL_MEM_EVENT_OP_PAGING 1
- #define XEN_DOMCTL_MEM_EVENT_OP_PAGING_NOMINATE 0
- #define XEN_DOMCTL_MEM_EVENT_OP_PAGING_EVICT 1
- #define XEN_DOMCTL_MEM_EVENT_OP_PAGING_PREP 2
- #define XEN_DOMCTL_MEM_EVENT_OP_PAGING_RESUME 3
- #define XEN_DOMCTL_MEM_EVENT_OP_ACCESS 2
- #define XEN_DOMCTL_MEM_EVENT_OP_ACCESS_RESUME 0
- #define XEN_DOMCTL_MEM_SHARING_OP_CONTROL 0
- #define XEN_DOMCTL_MEM_SHARING_OP_NOMINATE_GFN 1
- #define XEN_DOMCTL_MEM_SHARING_OP_NOMINATE_GREF 2
- #define XEN_DOMCTL_MEM_SHARING_OP_SHARE 3
- #define XEN_DOMCTL_MEM_SHARING_OP_RESUME 4
- #define XEN_DOMCTL_MEM_SHARING_OP_DEBUG_GFN 5
- #define XEN_DOMCTL_MEM_SHARING_OP_DEBUG_MFN 6
- #define XEN_DOMCTL_MEM_SHARING_OP_DEBUG_GREF 7
- #define XEN_DOMCTL_MEM_SHARING_S_HANDLE_INVALID (-10)
- #define XEN_DOMCTL_MEM_SHARING_C_HANDLE_INVALID (-9)
- #define XEN_DOMCTL_createdomain 1
- #define XEN_DOMCTL_destroydomain 2
- #define XEN_DOMCTL_pausedomain 3
- #define XEN_DOMCTL_unpausedomain 4
- #define XEN_DOMCTL_getdomaininfo 5
- #define XEN_DOMCTL_getmemlist 6
- #define XEN_DOMCTL_getpageframeinfo 7
- #define XEN_DOMCTL_getpageframeinfo2 8
- #define XEN_DOMCTL_setvcpuaffinity 9
- #define XEN_DOMCTL_shadow_op 10
- #define XEN_DOMCTL_max_mem 11
- #define XEN_DOMCTL_setvcpucontext 12
- #define XEN_DOMCTL_getvcpucontext 13
- #define XEN_DOMCTL_getvcpuinfo 14
- #define XEN_DOMCTL_max_vcpus 15
- #define XEN_DOMCTL_scheduler_op 16
- #define XEN_DOMCTL_setdomainhandle 17
- #define XEN_DOMCTL_setdebugging 18
- #define XEN_DOMCTL_irq_permission 19
- #define XEN_DOMCTL_iomem_permission 20
- #define XEN_DOMCTL_ioport_permission 21
- #define XEN_DOMCTL_hypercall_init 22
- #define XEN_DOMCTL_arch_setup 23
- #define XEN_DOMCTL_settimeoffset 24
- #define XEN_DOMCTL_getvcpuaffinity 25
- #define XEN_DOMCTL_real_mode_area 26
- #define XEN_DOMCTL_resumedomain 27
- #define XEN_DOMCTL_sendtrigger 28
- #define XEN_DOMCTL_subscribe 29
- #define XEN_DOMCTL_gethvmcontext 33
- #define XEN_DOMCTL_sethvmcontext 34
- #define XEN_DOMCTL_set_address_size 35

- #define XEN_DOMCTL_get_address_size 36
- #define XEN_DOMCTL_assign_device 37
- #define XEN_DOMCTL_bind_pt_irq 38
- #define XEN_DOMCTL_memory_mapping 39
- #define XEN_DOMCTL_ioport_mapping 40
- #define XEN_DOMCTL_pin_mem_cacheattr 41
- #define XEN_DOMCTL_set_ext_vcpucontext 42
- #define XEN_DOMCTL_get_ext_vcpucontext 43
- #define XEN_DOMCTL_set_opt_feature 44
- #define XEN_DOMCTL_test_assign_device 45
- #define XEN_DOMCTL_set_target 46
- #define XEN_DOMCTL_deassign_device 47
- #define XEN_DOMCTL_unbind_pt_irq 48
- #define XEN_DOMCTL_set_cpuid 49
- #define XEN_DOMCTL_get_device_group 50
- #define XEN_DOMCTL_set_machine_address_size 51
- #define XEN_DOMCTL_get_machine_address_size 52
- #define XEN_DOMCTL_suppress_spurious_page_faults 53
- #define XEN_DOMCTL_debug_op 54
- #define XEN_DOMCTL_gethvmcontext_partial 55
- #define XEN_DOMCTL_mem_event_op 56
- #define XEN_DOMCTL_mem_sharing_op 57
- #define XEN_DOMCTL_disable_migrate 58
- #define XEN_DOMCTL_gettscinfo 59
- #define XEN_DOMCTL_settscinfo 60
- #define XEN_DOMCTL_getpageframeinfo3 61
- #define XEN_DOMCTL_setvcpuextstate 62
- #define XEN_DOMCTL_getvcpuextstate 63
- #define XEN_DOMCTL_set_access_required 64
- #define XEN_DOMCTL_gdbsx_guestmemio 1000
- #define XEN_DOMCTL_gdbsx_pausevcpu 1001
- #define XEN_DOMCTL_gdbsx_unpausevcpu 1002
- #define XEN_DOMCTL_gdbsx_domstatus 1003

Typedefs

- typedef struct xen_domctl_createdomain xen_domctl_createdomain_t
- typedef struct xen_domctl_getdomaininfo xen_domctl_getdomaininfo_t
- typedef struct xen_domctl_getmemlist xen_domctl_getmemlist_t
- typedef struct xen_domctl_getpageframeinfo xen_domctl_getpageframeinfo_t
- typedef struct xen_domctl_getpageframeinfo2 xen_domctl_getpageframeinfo2_t
- typedef struct xen_domctl_shadow_op_stats xen_domctl_shadow_op_stats_t
- typedef struct xen_domctl_shadow_op xen_domctl_shadow_op_t
- typedef struct xen_domctl_max_mem xen_domctl_max_mem_t
- typedef struct xen_domctl_vcpucontext xen_domctl_vcpucontext_t
- typedef struct xen_domctl_getvcpuinfo xen_domctl_getvcpuinfo_t

- typedef struct xen_domctl_vcpuaffinity xen_domctl_vcpuaffinity_t
- typedef struct xen_domctl_max_vcpus xen_domctl_max_vcpus_t
- typedef struct xen_domctl_scheduler_op xen_domctl_scheduler_op_t
- typedef struct xen_domctl_setdomainhandle xen_domctl_setdomainhandle_t
- typedef struct xen_domctl_setdebugging xen_domctl_setdebugging_t
- typedef struct xen_domctl_irq_permission xen_domctl_irq_permission_t
- typedef struct xen_domctl_iomem_permission xen_domctl_iomem_permission_t
- typedef struct xen_domctl_ioport_permission xen_domctl_ioport_permission_t
- typedef struct xen_domctl_hypercall_init xen_domctl_hypercall_init_t
- typedef struct xen_domctl_arch_setup xen_domctl_arch_setup_t
- typedef struct xen_domctl_settimeoffset xen_domctl_settimeoffset_t
- typedef struct xen_domctl_hvmcontext xen_domctl_hvmcontext_t
- typedef struct xen_domctl_address_size xen_domctl_address_size_t
- typedef struct xen_domctl_real_mode_area xen_domctl_real_mode_area_t
- typedef struct xen_domctl_sendtrigger xen_domctl_sendtrigger_t
- typedef struct xen_domctl_assign_device xen_domctl_assign_device_t
- typedef struct xen_domctl_get_device_group xen_domctl_get_device_group_t
- typedef enum pt_irq_type_e pt_irq_type_t
- typedef struct xen_domctl_bind_pt_irq xen_domctl_bind_pt_irq_t
- typedef struct xen_domctl_memory_mapping xen_domctl_memory_mapping_t
- typedef struct xen_domctl_ioport_mapping xen_domctl_ioport_mapping_t
- typedef struct xen_domctl_pin_mem_cacheattr xen_domctl_pin_mem_cacheattr_t
- typedef struct xen_domctl_ext_vcpucontext xen_domctl_ext_vcpucontext_t
- typedef struct xen_domctl_set_opt_feature xen_domctl_set_opt_feature_t
- typedef struct xen_domctl_set_target xen_domctl_set_target_t
- typedef struct xen_domctl_subscribe xen_domctl_subscribe_t
- typedef struct xen_domctl_debug_op xen_domctl_debug_op_t
- typedef struct xen_domctl_hvmcontext_partial xen_domctl_hvmcontext_partial_t
- typedef struct xen_domctl_disable_migrate xen_domctl_disable_migrate_t
- typedef struct xen_guest_tsc_info xen_guest_tsc_info_t
- typedef struct xen_domctl_tsc_info xen_domctl_tsc_info_t

- typedef struct xen_domctl_mem_event_op xen_domctl_mem_event_op_t
- typedef struct xen_domctl_mem_sharing_op xen_domctl_mem_sharing_op_t
- typedef struct xen_domctl_set_access_required xen_domctl_set_access_required_t
- typedef struct xen_domctl xen_domctl_t

Enumerations

- enum pt_irq_type_e { PT_IRQ_TYPE_PCI, PT_IRQ_TYPE_ISA, PT_IRQ_TYPE_MSI, PT_IRQ_TYPE_MSI_TRANSLATE }

Functions

- DEFINE_XEN_GUEST_HANDLE (xen_domctl_createdomain_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_getdomaininfo_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_getmemlist_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_getpageframeinfo_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_getpageframeinfo2_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_shadow_op_stats_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_shadow_op_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_max_mem_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_vcpucontext_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_getvcpuinfo_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_vcpuaffinity_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_max_vcpus_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_scheduler_op_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_setdomainhandle_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_setdebugging_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_irq_permission_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_iomem_permission_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_ioport_permission_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_hypercall_init_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_arch_setup_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_settimeoffset_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_hvmcontext_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_address_size_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_real_mode_area_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_sendtrigger_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_assign_device_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_get_device_group_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_bind_pt_irq_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_memory_mapping_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_ioport_mapping_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_pin_mem_cacheattr_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_ext_vcpucontext_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_set_opt_feature_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_set_target_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_subscribe_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_debug_op_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_hvmcontext_partial_t)
- DEFINE_XEN_GUEST_HANDLE (xen_guest_tsc_info_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_mem_event_op_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_mem_sharing_op_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_set_access_required_t)
- DEFINE_XEN_GUEST_HANDLE (xen_domctl_t)

4.181.1 Macro Definition Documentation

4.181.1.1 #define _XEN_DOMAINSETUP_hvm_guest 0

4.181.1.2 #define _XEN_DOMAINSETUP_query 1 /∗ Get parameters (for save) ∗/

4.181.1.3 #define _XEN_DOMAINSETUP_sioemu_guest 2

4.181.1.4 #define _XEN_DOMCTL_CDF_hap 1

4.181.1.5 #define _XEN_DOMCTL_CDF_hvm_guest 0

4.181.1.6 #define _XEN_DOMCTL_CDF_oos_off 3

4.181.1.7 #define _XEN_DOMCTL_CDF_priv 5

4.181.1.8 #define _XEN_DOMCTL_CDF_s3_integrity 2

4.181.1.9 #define _XEN_DOMINF_blocked 4

4.181.1.10 #define _XEN_DOMINF_debugged 6

4.181.1.11 #define _XEN_DOMINF_dying 0

4.181.1.12 #define _XEN_DOMINF_hvm_guest 1

4.181.1.13 #define _XEN_DOMINF_paused 3

4.181.1.14 #define _XEN_DOMINF_running 5

4.181.1.15 #define _XEN_DOMINF_shutdown 2

4.181.1.16 #define DPCI_ADD_MAPPING 1

4.181.1.17 #define DPCI_REMOVE_MAPPING 0

4.181.1.18 #define XEN_DOMAINSETUP_hvm_guest (1UL<<_XEN_DOMAINSETUP_hvm_guest)

4.181.1.19 #define XEN_DOMAINSETUP_query (1UL<<_XEN_DOMAINSETUP_query)

4.181.1.20 #define XEN_DOMAINSETUP_sioemu_guest (1UL<<_XEN_DOMAINSETUP_sioemu_guest)

4.181.1.21 #define XEN_DOMCTL_arch_setup 23

4.181.1.22 #define XEN_DOMCTL_assign_device 37

4.181.1.23 #define XEN_DOMCTL_bind_pt_irq 38

4.181.1.24 #define XEN_DOMCTL_CDF_hap (1U<<_XEN_DOMCTL_CDF_hap)

4.181.1.25 #define XEN_DOMCTL_CDF_hvm_guest (1U<<_XEN_DOMCTL_CDF_hvm_guest)

4.181.1.26 #define XEN_DOMCTL_CDF_oos_off (1U<<_XEN_DOMCTL_CDF_oos_off)

4.181.1.27 #define XEN_DOMCTL_CDF_priv (1U<<_XEN_DOMCTL_CDF_priv)

4.181.1.28 #define XEN_DOMCTL_CDF_s3_integrity (1U<<_XEN_DOMCTL_CDF_s3_integrity)

4.181.1.29 #define XEN_DOMCTL_createdomain 1

4.181.1.30 #define XEN_DOMCTL_deassign_device 47

4.181.1.31 #define XEN_DOMCTL_debug_op 54

4.181.1.32 #define XEN_DOMCTL_DEBUG_OP_SINGLE_STEP_OFF 0

4.181.1.33 #define XEN_DOMCTL_DEBUG_OP_SINGLE_STEP_ON 1

4.181.1.34 #define XEN_DOMCTL_destroydomain 2

4.181.1.35 #define XEN_DOMCTL_disable_migrate 58

4.181.1.36 #define XEN_DOMCTL_gdbsx_domstatus 1003

4.181.1.37 #define XEN_DOMCTL_gdbsx_guestmemio 1000

4.181.1.38 #define XEN_DOMCTL_gdbsx_pausevcpu 1001

4.181.1.39 #define XEN_DOMCTL_gdbsx_unpausevcpu 1002

4.181.1.40 #define XEN_DOMCTL_get_address_size 36

4.181.1.41 #define XEN_DOMCTL_get_device_group 50

4.181.1.42 #define XEN_DOMCTL_get_ext_vcpucontext 43

4.181.1.43 #define XEN_DOMCTL_get_machine_address_size 52

4.181.1.44 #define XEN_DOMCTL_getdomaininfo 5

4.181.1.45 #define XEN_DOMCTL_gethvmcontext 33

4.181.1.46 #define XEN_DOMCTL_gethvmcontext_partial 55

4.181.1.47 #define XEN_DOMCTL_getmemlist 6

4.181.1.48 #define XEN_DOMCTL_getpageframeinfo 7

4.181.1.49 #define XEN_DOMCTL_getpageframeinfo2 8

4.181.1.50 #define XEN_DOMCTL_getpageframeinfo3 61

4.181.1.51 #define XEN_DOMCTL_gettscinfo 59

4.181.1.52 #define XEN_DOMCTL_getvcpuaffinity 25

4.181.1.53 #define XEN_DOMCTL_getvcpucontext 13

4.181.1.54 #define XEN_DOMCTL_getvcpuextstate 63

4.181.1.55 #define XEN_DOMCTL_getvcpuinfo 14

4.181.1.56 #define XEN_DOMCTL_hypercall_init 22

4.181.1.57 #define XEN_DOMCTL_INTERFACE_VERSION 0x00000007

4.181.1.58 #define XEN_DOMCTL_iomem_permission 20

4.181.1.59 #define XEN_DOMCTL_ioport_mapping 40

4.181.1.60 #define XEN_DOMCTL_ioport_permission 21

4.181.1.61 #define XEN_DOMCTL_irq_permission 19

4.181.1.62 #define XEN_DOMCTL_max_mem 11

4.181.1.63 #define XEN_DOMCTL_max_vcpus 15

4.181.1.64 #define XEN_DOMCTL_MEM_CACHEATTR_UC 0

4.181.1.65 #define XEN_DOMCTL_MEM_CACHEATTR_UCM 7

4.181.1.66 #define XEN_DOMCTL_MEM_CACHEATTR_WB 6

4.181.1.67 #define XEN_DOMCTL_MEM_CACHEATTR_WC 1

4.181.1.68 #define XEN_DOMCTL_MEM_CACHEATTR_WP 5

4.181.1.69 #define XEN_DOMCTL_MEM_CACHEATTR_WT 4

4.181.1.70 #define XEN_DOMCTL_mem_event_op 56

4.181.1.71 #define XEN_DOMCTL_MEM_EVENT_OP_ACCESS 2

4.181.1.72 #define XEN_DOMCTL_MEM_EVENT_OP_ACCESS_RESUME 0

4.181.1.73 #define XEN_DOMCTL_MEM_EVENT_OP_DISABLE 1

4.181.1.74 #define XEN_DOMCTL_MEM_EVENT_OP_ENABLE 0

4.181.1.75 #define XEN_DOMCTL_MEM_EVENT_OP_PAGING 1

4.181.1.76 #define XEN_DOMCTL_MEM_EVENT_OP_PAGING_EVICT 1

4.181.1.77 #define XEN_DOMCTL_MEM_EVENT_OP_PAGING_NOMINATE 0

4.181.1.78 #define XEN_DOMCTL_MEM_EVENT_OP_PAGING_PREP 2

4.181.1.79 #define XEN_DOMCTL_MEM_EVENT_OP_PAGING_RESUME 3

4.181.1.80 #define XEN_DOMCTL_MEM_SHARING_C_HANDLE_INVALID (-9)

4.181.1.81 #define XEN_DOMCTL_mem_sharing_op 57

4.181.1.82 #define XEN_DOMCTL_MEM_SHARING_OP_CONTROL 0

4.181.1.83 #define XEN_DOMCTL_MEM_SHARING_OP_DEBUG_GFN 5

4.181.1.84 #define XEN_DOMCTL_MEM_SHARING_OP_DEBUG_GREF 7

4.181.1.85 #define XEN_DOMCTL_MEM_SHARING_OP_DEBUG_MFN 6

4.181.1.86 #define XEN_DOMCTL_MEM_SHARING_OP_NOMINATE_GFN 1

4.181.1.87 #define XEN_DOMCTL_MEM_SHARING_OP_NOMINATE_GREF 2

4.181.1.88 #define XEN_DOMCTL_MEM_SHARING_OP_RESUME 4

4.181.1.89 #define XEN_DOMCTL_MEM_SHARING_OP_SHARE 3

4.181.1.90 #define XEN_DOMCTL_MEM_SHARING_S_HANDLE_INVALID (-10)

4.181.1.91 #define XEN_DOMCTL_memory_mapping 39

4.181.1.92 #define XEN_DOMCTL_pausedomain 3

4.181.1.93 #define XEN_DOMCTL_PFINFO_L1TAB (0x1U<<28)

4.181.1.94 #define XEN_DOMCTL_PFINFO_L2TAB (0x2U<<28)

4.181.1.95 #define XEN_DOMCTL_PFINFO_L3TAB (0x3U<<28)

4.181.1.96 #define XEN_DOMCTL_PFINFO_L4TAB (0x4U<<28)

4.181.1.97 #define XEN_DOMCTL_PFINFO_LPINTAB (0x1U<<31)

4.181.1.98 #define XEN_DOMCTL_PFINFO_LTAB_MASK (0xfU<<28)

4.181.1.99 #define XEN_DOMCTL_PFINFO_LTAB_SHIFT 28

4.181.1.100 #define XEN_DOMCTL_PFINFO_LTABTYPE_MASK (0x7U<<28)

4.181.1.101 #define XEN_DOMCTL_PFINFO_NOTAB (0x0U<<28)

4.181.1.102 #define XEN_DOMCTL_PFINFO_PAGEDTAB (0x8U<<28)

4.181.1.103 #define XEN_DOMCTL_PFINFO_XTAB (0xfU<<28) /* invalid page */

4.181.1.104 #define XEN_DOMCTL_pin_mem_cacheattr 41

4.181.1.105 #define XEN_DOMCTL_real_mode_area 26

4.181.1.106 #define XEN_DOMCTL_resumedomain 27

4.181.1.107 #define XEN_DOMCTL_SCHEDOP_getinfo 1

4.181.1.108 #define XEN_DOMCTL_SCHEDOP_putinfo 0

4.181.1.109 #define XEN_DOMCTL_scheduler_op 16

4.181.1.110 #define XEN_DOMCTL_sendtrigger 28

4.181.1.111 #define XEN_DOMCTL_SENDTRIGGER_INIT 2

4.181.1.112 #define XEN_DOMCTL_SENDTRIGGER_NMI 0

4.181.1.113 #define XEN_DOMCTL_SENDTRIGGER_POWER 3

4.181.1.114 #define XEN_DOMCTL_SENDTRIGGER_RESET 1

4.181.1.115 #define XEN_DOMCTL_SENDTRIGGER_SLEEP 4

4.181.1.116 #define XEN_DOMCTL_set_access_required 64

4.181.1.117 #define XEN_DOMCTL_set_address_size 35

4.181.1.118 #define XEN_DOMCTL_set_cpuid 49

4.181.1.119 #define XEN_DOMCTL_set_ext_vcpucontext 42

4.181.1.120 #define XEN_DOMCTL_set_machine_address_size 51

4.181.1.121 #define XEN_DOMCTL_set_opt_feature 44

4.181.1.122 #define XEN_DOMCTL_set_target 46

4.181.1.123 #define XEN_DOMCTL_setdebugging 18

4.181.1.124 #define XEN_DOMCTL_setdomainhandle 17

4.181.1.125 #define XEN_DOMCTL_sethvmcontext 34

4.181.1.126 #define XEN_DOMCTL_settimeoffset 24

4.181.1.127 #define XEN_DOMCTL_settscinfo 60

4.181.1.128 #define XEN_DOMCTL_setvcpuaffinity 9

4.181.1.129 #define XEN_DOMCTL_setvcpucontext 12

4.181.1.130 #define XEN_DOMCTL_setvcpuextstate 62

4.181.1.131 #define XEN_DOMCTL_SHADOW_ENABLE_EXTERNAL (1 << 4)

4.181.1.132 #define XEN_DOMCTL_SHADOW_ENABLE_LOG_DIRTY (1 << 2)

4.181.1.133 #define XEN_DOMCTL_SHADOW_ENABLE_REFCOUNT (1 << 1)

4.181.1.134 #define XEN_DOMCTL_SHADOW_ENABLE_TRANSLATE (1 << 3)

4.181.1.135 #define XEN_DOMCTL_shadow_op 10

4.181.1.136 #define XEN_DOMCTL_SHADOW_OP_CLEAN 11

4.181.1.137 #define XEN_DOMCTL_SHADOW_OP_ENABLE 32

4.181.1.138 #define XEN_DOMCTL_SHADOW_OP_ENABLE_LOGDIRTY 2

4.181.1.139 #define XEN_DOMCTL_SHADOW_OP_ENABLE_TEST 1

4.181.1.140 #define XEN_DOMCTL_SHADOW_OP_ENABLE_TRANSLATE 3

4.181.1.141 #define XEN_DOMCTL_SHADOW_OP_GET_ALLOCATION 30

4.181.1.142 #define XEN_DOMCTL_SHADOW_OP_OFF 0

4.181.1.143 #define XEN_DOMCTL_SHADOW_OP_PEEK 12

4.181.1.144 #define XEN_DOMCTL_SHADOW_OP_SET_ALLOCATION 31

4.181.1.145 #define XEN_DOMCTL_subscribe 29

4.181.1.146 #define XEN_DOMCTL_suppress_spurious_page_faults 53

4.181.1.147 #define XEN_DOMCTL_test_assign_device 45

4.181.1.148 #define XEN_DOMCTL_unbind_pt_irq 48

4.181.1.149 #define XEN_DOMCTL_unpausedomain 4

4.181.1.150 #define XEN_DOMINF_blocked (1U<<_XEN_DOMINF_blocked)

4.181.1.151 #define XEN_DOMINF_debugged (1U<<_XEN_DOMINF_debugged)

4.181.1.152 #define XEN_DOMINF_dying (1U<<_XEN_DOMINF_dying)

4.181.1.153 #define XEN_DOMINF_hvm_guest (1U<<_XEN_DOMINF_hvm_guest)

4.181.1.154 #define XEN_DOMINF_paused (1U<<_XEN_DOMINF_paused)

4.181.1.155 #define XEN_DOMINF_running (1U<<_XEN_DOMINF_running)

4.181.1.156 #define XEN_DOMINF_shutdown (1U<<_XEN_DOMINF_shutdown)

4.181.1.157 #define XEN_DOMINF_shutdownmask 255

4.181.1.158 #define XEN_DOMINF_shutdownshift 16

4.181.1.159 #define XEN_SCHEDULER_ARINC653 7

4.181.1.160 #define XEN_SCHEDULER_CREDIT 5

4.181.1.161 #define XEN_SCHEDULER_CREDIT2 6

4.181.1.162 #define XEN_SCHEDULER_SEDF 4

4.181.2 Typedef Documentation

4.181.2.1 typedef enum **pt_irq_type_e pt_irq_type_t**

4.181.2.2 typedef struct **xen_domctl_address_size xen_domctl_address_size_t**

4.181.2.3 typedef struct **xen_domctl_arch_setup xen_domctl_arch_setup_t**

4.181.2.4 typedef struct **xen_domctl_assign_device xen_domctl_assign_device_t**

4.181.2.5 typedef struct xen_domctl_bind_pt_irq xen_domctl_bind_pt_irq_t

4.181.2.6 typedef struct xen_domctl_createdomain xen_domctl_createdomain_t

4.181.2.7 typedef struct xen_domctl_debug_op xen_domctl_debug_op_t

4.181.2.8 typedef struct xen_domctl_disable_migrate xen_domctl_disable_migrate_t

4.181.2.9 typedef struct xen_domctl_ext_vcpucontext xen_domctl_ext_vcpucontext_t

4.181.2.10 typedef struct xen_domctl_get_device_group xen_domctl_get_device_group_t

4.181.2.11 typedef struct xen_domctl_getdomaininfo xen_domctl_getdomaininfo_t

4.181.2.12 typedef struct xen_domctl_getmemlist xen_domctl_getmemlist_t

4.181.2.13 typedef struct xen_domctl_getpageframeinfo2 xen_domctl_getpageframeinfo2_t

4.181.2.14 typedef struct xen_domctl_getpageframeinfo xen_domctl_getpageframeinfo_t

4.181.2.15 typedef struct xen_domctl_getvcpuinfo xen_domctl_getvcpuinfo_t

4.181.2.16 typedef struct xen_domctl_hvmcontext_partial xen_domctl_hvmcontext_partial_t

4.181.2.17 typedef struct xen_domctl_hvmcontext xen_domctl_hvmcontext_t

4.181.2.18 typedef struct xen_domctl_hypercall_init xen_domctl_hypercall_init_t

4.181.2.19 typedef struct xen_domctl_iomem_permission xen_domctl_iomem_permission_t

4.181.2.20 typedef struct xen_domctl_ioport_mapping xen_domctl_ioport_mapping_t

4.181.2.21 typedef struct xen_domctl_ioport_permission xen_domctl_ioport_permission_t

4.181.2.22 typedef struct xen_domctl_irq_permission xen_domctl_irq_permission_t

4.181.2.23 typedef struct xen_domctl_max_mem xen_domctl_max_mem_t

4.181.2.24 typedef struct xen_domctl_max_vcpus xen_domctl_max_vcpus_t

4.181.2.25 typedef struct xen_domctl_mem_event_op xen_domctl_mem_event_op_t

4.181.2.26 typedef struct xen_domctl_mem_sharing_op xen_domctl_mem_sharing_op_t

4.181.2.27 typedef struct xen_domctl_memory_mapping xen_domctl_memory_mapping_t

4.181.2.28 typedef struct xen_domctl_pin_mem_cacheattr xen_domctl_pin_mem_cacheattr_t

4.181.2.29 typedef struct xen_domctl_real_mode_area xen_domctl_real_mode_area_t

4.181.2.30 typedef struct xen_domctl_scheduler_op xen_domctl_scheduler_op_t

4.181.2.31 typedef struct xen_domctl_sendtrigger xen_domctl_sendtrigger_t

4.181.2.32 typedef struct xen_domctl_set_access_required xen_domctl_set_access_required_t

4.181.2.33 typedef struct xen_domctl_set_opt_feature xen_domctl_set_opt_feature_t

4.181.2.34 typedef struct xen_domctl_set_target xen_domctl_set_target_t

4.181.2.35 typedef struct xen_domctl_setdebugging xen_domctl_setdebugging_t

4.181.2.36 typedef struct xen_domctl_setdomainhandle xen_domctl_setdomainhandle_t

4.181.2.37 typedef struct xen_domctl_settimeoffset xen_domctl_settimeoffset_t

4.181.2.38 typedef struct xen_domctl_shadow_op_stats xen_domctl_shadow_op_stats_t

4.181.2.39 typedef struct xen_domctl_shadow_op xen_domctl_shadow_op_t

4.181.2.40 typedef struct xen_domctl_subscribe xen_domctl_subscribe_t

4.181.2.41 typedef struct xen_domctl xen_domctl_t

4.181.2.42 typedef struct xen_domctl_tsc_info xen_domctl_tsc_info_t

4.181.2.43 typedef struct xen_domctl_vcpuaffinity xen_domctl_vcpuaffinity_t

4.181.2.44 typedef struct xen_domctl_vcpucontext xen_domctl_vcpucontext_t

4.181.2.45 typedef struct xen_guest_tsc_info xen_guest_tsc_info_t

4.181.3 Enumeration Type Documentation

4.181.3.1 enum pt_irq_type_e

Enumerator

PT_IRQ_TYPE_PCI

PT_IRQ_TYPE_ISA

PT_IRQ_TYPE_MSI

PT_IRQ_TYPE_MSI_TRANSLATE

4.181.4 Function Documentation

4.181.4.1 DEFINE_XEN_GUEST_HANDLE (xen_domctl_createdomain_t)

4.181.4.2 DEFINE_XEN_GUEST_HANDLE (xen_domctl_getdomaininfo_t)

4.181.4.3 DEFINE_XEN_GUEST_HANDLE (xen_domctl_getmemlist_t)

4.181.4.4 DEFINE_XEN_GUEST_HANDLE (xen_domctl_getpageframeinfo_t)

4.181.4.5 DEFINE_XEN_GUEST_HANDLE (xen_domctl_getpageframeinfo2_t)

4.181.4.6 DEFINE_XEN_GUEST_HANDLE (xen_domctl_shadow_op_stats_t)

4.181.4.7 DEFINE_XEN_GUEST_HANDLE (xen_domctl_shadow_op_t)

4.181.4.8 DEFINE_XEN_GUEST_HANDLE (xen_domctl_max_mem_t)

4.181.4.9 DEFINE_XEN_GUEST_HANDLE (xen_domctl_vcpucontext_t)

4.181.4.10 DEFINE_XEN_GUEST_HANDLE (xen_domctl_getvcpuinfo_t)

4.181.4.11 DEFINE_XEN_GUEST_HANDLE (xen_domctl_vcpuaffinity_t)

4.181.4.12 DEFINE_XEN_GUEST_HANDLE (xen_domctl_max_vcpus_t)

4.181.4.13 DEFINE_XEN_GUEST_HANDLE (xen_domctl_scheduler_op_t)

4.181.4.14 DEFINE_XEN_GUEST_HANDLE (xen_domctl_setdomainhandle_t)

4.181.4.15 DEFINE_XEN_GUEST_HANDLE (xen_domctl_setdebugging_t)

4.181.4.16 DEFINE_XEN_GUEST_HANDLE (xen_domctl_irq_permission_t)

4.181.4.17 DEFINE_XEN_GUEST_HANDLE (xen_domctl_iomem_permission_t)

4.181.4.18 DEFINE_XEN_GUEST_HANDLE (xen_domctl_ioport_permission_t)

4.181.4.19 DEFINE_XEN_GUEST_HANDLE (xen_domctl_hypercall_init_t)

4.181.4.20 DEFINE_XEN_GUEST_HANDLE (xen_domctl_arch_setup_t)

4.181.4.21 DEFINE_XEN_GUEST_HANDLE (xen_domctl_settimeoffset_t)

4.181.4.22 DEFINE_XEN_GUEST_HANDLE (xen_domctl_hvmcontext_t)

4.181.4.23 DEFINE_XEN_GUEST_HANDLE (xen_domctl_address_size_t)

4.181.4.24 DEFINE_XEN_GUEST_HANDLE (xen_domctl_real_mode_area_t)

4.181.4.25 DEFINE_XEN_GUEST_HANDLE (xen_domctl_sendtrigger_t)

4.181.4.26 DEFINE_XEN_GUEST_HANDLE (xen_domctl_assign_device_t)

4.181.4.27 DEFINE_XEN_GUEST_HANDLE (xen_domctl_get_device_group_t)

4.181.4.28 DEFINE_XEN_GUEST_HANDLE (xen_domctl_bind_pt_irq_t)

4.181.4.29 DEFINE_XEN_GUEST_HANDLE (xen_domctl_memory_mapping_t)

4.181.4.30 DEFINE_XEN_GUEST_HANDLE (xen_domctl_ioport_mapping_t)

4.181.4.31 DEFINE_XEN_GUEST_HANDLE (xen_domctl_pin_mem_cacheattr_t)

4.181.4.32 DEFINE_XEN_GUEST_HANDLE (xen_domctl_ext_vcpucontext_t)

4.181.4.33 DEFINE_XEN_GUEST_HANDLE (xen_domctl_set_opt_feature_t)

4.181.4.34 DEFINE_XEN_GUEST_HANDLE (xen_domctl_set_target_t)

4.181.4.35 DEFINE_XEN_GUEST_HANDLE (xen_domctl_subscribe_t)

4.181.4.36 DEFINE_XEN_GUEST_HANDLE (xen_domctl_debug_op_t)

4.181.4.37 DEFINE_XEN_GUEST_HANDLE (xen_domctl_hvmcontext_partial_t)

4.181.4.38 DEFINE_XEN_GUEST_HANDLE (xen_guest_tsc_info_t)

4.181.4.39 DEFINE_XEN_GUEST_HANDLE (xen_domctl_mem_event_op_t)

4.181.4.40 DEFINE_XEN_GUEST_HANDLE (xen_domctl_mem_sharing_op_t)

4.181.4.41 DEFINE_XEN_GUEST_HANDLE (xen_domctl_set_access_required_t)

4.181.4.42 DEFINE_XEN_GUEST_HANDLE (xen_domctl_t)

4.182 xen/xen/include/public/xen.h File Reference

```
#include "xen-compat.h"
```

Data Structures

- struct mmuext_op
- struct mmu_update
- struct multicall_entry
- struct vcpu_time_info
- struct vcpu_info
- struct shared_info
- struct start_info
- struct xen_multiboot_mod_list
- struct dom0_vga_console_info

Macros

- #define __HYPERVISOR_set_trap_table 0
- #define __HYPERVISOR_mmu_update 1
- #define __HYPERVISOR_set_gdt 2
- #define __HYPERVISOR_stack_switch 3
- #define __HYPERVISOR_set_callbacks 4
- #define __HYPERVISOR_fpu_taskswitch 5
- #define __HYPERVISOR_sched_op_compat 6 /∗ compat since 0x00030101 ∗/
- #define __HYPERVISOR_platform_op 7
- #define __HYPERVISOR_set_debugreg 8
- #define __HYPERVISOR_get_debugreg 9
- #define __HYPERVISOR_update_descriptor 10
- #define __HYPERVISOR_memory_op 12
- #define __HYPERVISOR_multicall 13
- #define __HYPERVISOR_update_va_mapping 14
- #define __HYPERVISOR_set_timer_op 15
- #define __HYPERVISOR_event_channel_op_compat 16 /∗ compat since 0x00030202 ∗/
- #define __HYPERVISOR_xen_version 17
- #define __HYPERVISOR_console_io 18
- #define __HYPERVISOR_physdev_op_compat 19 /∗ compat since 0x00030202 ∗/
- #define __HYPERVISOR_grant_table_op 20
- #define __HYPERVISOR_vm_assist 21
- #define __HYPERVISOR_update_va_mapping_otherdomain 22

- #define __HYPERVISOR_iret 23 /* x86 only */
- #define __HYPERVISOR_vcpu_op 24
- #define __HYPERVISOR_set_segment_base 25 /* x86/64 only */
- #define __HYPERVISOR_mmuext_op 26
- #define __HYPERVISOR_xsm_op 27
- #define __HYPERVISOR_nmi_op 28
- #define __HYPERVISOR_sched_op 29
- #define __HYPERVISOR_callback_op 30
- #define __HYPERVISOR_xenoprof_op 31
- #define __HYPERVISOR_event_channel_op 32
- #define __HYPERVISOR_physdev_op 33
- #define __HYPERVISOR_hvm_op 34
- #define __HYPERVISOR_sysctl 35
- #define __HYPERVISOR_domctl 36
- #define __HYPERVISOR_kexec_op 37
- #define __HYPERVISOR_tmem_op 38
- #define __HYPERVISOR_arch_0 48
- #define __HYPERVISOR_arch_1 49
- #define __HYPERVISOR_arch_2 50
- #define __HYPERVISOR_arch_3 51
- #define __HYPERVISOR_arch_4 52
- #define __HYPERVISOR_arch_5 53
- #define __HYPERVISOR_arch_6 54
- #define __HYPERVISOR_arch_7 55
- #define __HYPERVISOR_sched_op __HYPERVISOR_sched_op_compat
- #define __HYPERVISOR_event_channel_op __HYPERVISOR_event_channel_op_compat
- #define __HYPERVISOR_physdev_op __HYPERVISOR_physdev_op_compat
- #define __HYPERVISOR_dom0_op __HYPERVISOR_platform_op
- #define VIRQ_TIMER 0 /* V. Timebase update, and/or requested timeout. */
- #define VIRQ_DEBUG 1 /* V. Request guest to dump debug info. */
- #define VIRQ_CONSOLE 2 /* G. (DOM0) Bytes received on emergency console. */
- #define VIRQ_DOM_EXC 3 /* G. (DOM0) Exceptional event for some domain. */
- #define VIRQ_TBUF 4 /* G. (DOM0) Trace buffer has records available. */
- #define VIRQ_DEBUGGER 6 /* G. (DOM0) A domain has paused for debugging. */
- #define VIRQ_XENOPROF 7 /* V. XenOprofile interrupt: new sample available */
- #define VIRQ_CON_RING 8 /* G. (DOM0) Bytes received on console */
- #define VIRQ_PCPU_STATE 9 /* G. (DOM0) PCPU state changed */
- #define VIRQ_MEM_EVENT 10 /* G. (DOM0) A memory event has occured */
- #define VIRQ_ARCH_0 16
- #define VIRQ_ARCH_1 17
- #define VIRQ_ARCH_2 18
- #define VIRQ_ARCH_3 19
- #define VIRQ_ARCH_4 20
- #define VIRQ_ARCH_5 21
- #define VIRQ_ARCH_6 22
- #define VIRQ_ARCH_7 23
- #define NR_VIRQS 24
- #define MMU_NORMAL_PT_UPDATE 0 /* checked '*ptr = val'. ptr is MA. */
- #define MMU_MACHPHYS_UPDATE 1 /* ptr = MA of frame to modify entry for */
- #define MMU_PT_UPDATE_PRESERVE_AD 2 /* atomically: *ptr = val | (*ptr&(A|D)) */
- #define MMUEXT_PIN_L1_TABLE 0
- #define MMUEXT_PIN_L2_TABLE 1
- #define MMUEXT_PIN_L3_TABLE 2
- #define MMUEXT_PIN_L4_TABLE 3
- #define MMUEXT_UNPIN_TABLE 4

- #define MMUEXT_NEW_BASEPTR 5
- #define MMUEXT_TLB_FLUSH_LOCAL 6
- #define MMUEXT_INVLPG_LOCAL 7
- #define MMUEXT_TLB_FLUSH_MULTI 8
- #define MMUEXT_INVLPG_MULTI 9
- #define MMUEXT_TLB_FLUSH_ALL 10
- #define MMUEXT_INVLPG_ALL 11
- #define MMUEXT_FLUSH_CACHE 12
- #define MMUEXT_SET_LDT 13
- #define MMUEXT_NEW_USER_BASEPTR 15
- #define MMUEXT_CLEAR_PAGE 16
- #define MMUEXT_COPY_PAGE 17
- #define MMUEXT_FLUSH_CACHE_GLOBAL 18
- #define MMUEXT_MARK_SUPER 19
- #define MMUEXT_UNMARK_SUPER 20
- #define UVMF_NONE (0UL<<0) /* No flushing at all. */
- #define UVMF_TLB_FLUSH (1UL<<0) /* Flush entire TLB(s). */
- #define UVMF_INVLPG (2UL<<0) /* Flush only one entry. */
- #define UVMF_FLUSHTYPE_MASK (3UL<<0)
- #define UVMF_MULTI (0UL<<2) /* Flush subset of TLBs. */
- #define UVMF_LOCAL (0UL<<2) /* Flush local TLB. */
- #define UVMF_ALL (1UL<<2) /* Flush all TLBs. */
- #define CONSOLEIO_write 0
- #define CONSOLEIO_read 1
- #define VMASST_CMD_enable 0
- #define VMASST_CMD_disable 1
- #define VMASST_TYPE_4gb_segments 0
- #define VMASST_TYPE_4gb_segments_notify 1
- #define VMASST_TYPE_writable_pagetables 2
- #define VMASST_TYPE_pae_extended_cr3 3
- #define MAX_VMASST_TYPE 3
- #define DOMID_FIRST_RESERVED (0x7FF0U)
- #define DOMID_SELF (0x7FF0U)
- #define DOMID_IO (0x7FF1U)
- #define DOMID_XEN (0x7FF2U)
- #define DOMID_COW (0x7FF3U)
- #define DOMID_INVALID (0x7FF4U)
- #define DOMID_IDLE (0x7FFFU)
- #define NR_EVENT_CHANNELS (sizeof(unsigned long) * sizeof(unsigned long) * 64)
- #define MAX_GUEST_CMDLINE 1024
- #define console_mfn console.domU.mfn
- #define console_evtchn console.domU.evtchn
- #define SIF_PRIVILEGED (1<<0) /* Is the domain privileged? */
- #define SIF_INITDOMAIN (1<<1) /* Is this the initial control domain? */
- #define SIF_MULTIBOOT_MOD (1<<2) /* Is mod_start a multiboot module? */
- #define SIF_MOD_START_PFN (1<<3) /* Is mod_start a PFN? */
- #define SIF_PM_MASK (0xFF<<8) /* reserve 1 byte for xen-pm options */
- #define XEN_VGATYPE_TEXT_MODE_3 0x03
- #define XEN_VGATYPE_VESA_LFB 0x23
- #define xen_vga_console_info dom0_vga_console_info
- #define xen_vga_console_info_t dom0_vga_console_info_t
- #define __mk_unsigned_long(x) x ## UL
- #define mk_unsigned_long(x) __mk_unsigned_long(x)

Typedefs

- typedef struct mmuext_op mmuext_op_t
- typedef uint16_t domid_t
- typedef struct mmu_update mmu_update_t
- typedef struct multicall_entry multicall_entry_t
- typedef struct vcpu_time_info vcpu_time_info_t
- typedef struct vcpu_info vcpu_info_t
- typedef struct shared_info shared_info_t
- typedef struct start_info start_info_t
- typedef struct dom0_vga_console_info dom0_vga_console_info_t
- typedef uint8_t xen_domain_handle_t [16]

Functions

- DEFINE_XEN_GUEST_HANDLE (char)
- __DEFINE_XEN_GUEST_HANDLE (uchar, unsigned char)
- DEFINE_XEN_GUEST_HANDLE (int)
- __DEFINE_XEN_GUEST_HANDLE (uint, unsigned int)
- DEFINE_XEN_GUEST_HANDLE (long)
- __DEFINE_XEN_GUEST_HANDLE (ulong, unsigned long)
- DEFINE_XEN_GUEST_HANDLE (void)
- DEFINE_XEN_GUEST_HANDLE (uint64_t)
- DEFINE_XEN_GUEST_HANDLE (xen_pfn_t)
- DEFINE_XEN_GUEST_HANDLE (mmuext_op_t)
- DEFINE_XEN_GUEST_HANDLE (mmu_update_t)
- DEFINE_XEN_GUEST_HANDLE (multicall_entry_t)
- __DEFINE_XEN_GUEST_HANDLE (uint8, uint8_t)
- __DEFINE_XEN_GUEST_HANDLE (uint16, uint16_t)
- __DEFINE_XEN_GUEST_HANDLE (uint32, uint32_t)
- __DEFINE_XEN_GUEST_HANDLE (uint64, uint64_t)

4.182.1 Macro Definition Documentation

4.182.1.1 #define __HYPERVISOR_arch_0 48

4.182.1.2 #define __HYPERVISOR_arch_1 49

4.182.1.3 #define __HYPERVISOR_arch_2 50

4.182.1.4 #define __HYPERVISOR_arch_3 51

4.182.1.5 #define __HYPERVISOR_arch_4 52

4.182.1.6 #define __HYPERVISOR_arch_5 53

4.182.1.7 #define __HYPERVISOR_arch_6 54

4.182.1.8 #define __HYPERVISOR_arch_7 55

4.182.1.9 #define __HYPERVISOR_callback_op 30

4.182.1.10 #define __HYPERVISOR_console_io 18

4.182.1.11 #define __HYPERVISOR_dom0_op __HYPERVISOR_platform_op

4.182.1.12 #define __HYPERVISOR_domctl 36

4.182.1.13 #define __HYPERVISOR_event_channel_op 32

4.182.1.14 #define __HYPERVISOR_event_channel_op __HYPERVISOR_event_channel_op_compat

4.182.1.15 #define __HYPERVISOR_event_channel_op_compat 16 /* compat since 0x00030202 */

4.182.1.16 #define __HYPERVISOR_fpu_taskswitch 5

4.182.1.17 #define __HYPERVISOR_get_debugreg 9

4.182.1.18 #define __HYPERVISOR_grant_table_op 20

4.182.1.19 #define __HYPERVISOR_hvm_op 34

4.182.1.20 #define __HYPERVISOR_iret 23 /* x86 only */

4.182.1.21 #define __HYPERVISOR_kexec_op 37

4.182.1.22 #define __HYPERVISOR_memory_op 12

4.182.1.23 #define __HYPERVISOR_mmu_update 1

4.182.1.24 #define __HYPERVISOR_mmuext_op 26

4.182.1.25 #define __HYPERVISOR_multicall 13

4.182.1.26 #define __HYPERVISOR_nmi_op 28

4.182.1.27 #define __HYPERVISOR_physdev_op 33

4.182.1.28 #define __HYPERVISOR_physdev_op __HYPERVISOR_physdev_op_compat

4.182.1.29 #define __HYPERVISOR_physdev_op_compat 19 /* compat since 0x00030202 */

4.182.1.30 #define __HYPERVISOR_platform_op 7

4.182.1.31 #define __HYPERVISOR_sched_op 29

4.182.1.32 #define __HYPERVISOR_sched_op __HYPERVISOR_sched_op_compat

4.182.1.33 #define __HYPERVISOR_sched_op_compat 6 /* compat since 0x00030101 */

4.182.1.34 #define __HYPERVISOR_set_callbacks 4

4.182.1.35 #define __HYPERVISOR_set_debugreg 8

4.182.1.36 #define __HYPERVISOR_set_gdt 2

4.182.1.37 #define __HYPERVISOR_set_segment_base 25 /* x86/64 only */

4.182.1.38 #define __HYPERVISOR_set_timer_op 15

4.182.1.39 #define __HYPERVISOR_set_trap_table 0

4.182.1.40 #define __HYPERVISOR_stack_switch 3

4.182.1.41 #define __HYPERVISOR_sysctl 35

4.182.1.42 #define __HYPERVISOR_tmem_op 38

4.182.1.43 #define __HYPERVISOR_update_descriptor 10

4.182.1.44 #define __HYPERVISOR_update_va_mapping 14

4.182.1.45 #define __HYPERVISOR_update_va_mapping_otherdomain 22

4.182.1.46 #define __HYPERVISOR_vcpu_op 24

4.182.1.47 #define __HYPERVISOR_vm_assist 21

4.182.1.48 #define __HYPERVISOR_xen_version 17

4.182.1.49 #define __HYPERVISOR_xenoprof_op 31

4.182.1.50 #define __HYPERVISOR_xsm_op 27

4.182.1.51 #define __mk_unsigned_long(*x*) x ## UL

4.182.1.52 #define console_evtchn console.domU.evtchn

4.182.1.53 #define console_mfn console.domU.mfn

4.182.1.54 #define CONSOLEIO_read 1

4.182.1.55 #define CONSOLEIO_write 0

4.182.1.56 #define DOMID_COW (0x7FF3U)

4.182.1.57 #define DOMID_FIRST_RESERVED (0x7FF0U)

4.182.1.58 #define DOMID_IDLE (0x7FFFU)

4.182.1.59 #define DOMID_INVALID (0x7FF4U)

4.182.1.60 #define DOMID_IO (0x7FF1U)

4.182.1.61 #define DOMID_SELF (0x7FF0U)

4.182.1.62 #define DOMID_XEN (0x7FF2U)

4.182.1.63 #define MAX_GUEST_CMDLINE 1024

4.182.1.64 #define MAX_VMASST_TYPE 3

4.182.1.65 #define mk_unsigned_long(*x*) __mk_unsigned_long(x)

4.182.1.66 #define MMU_MACHPHYS_UPDATE 1 /∗ ptr = MA of frame to modify entry for ∗/

4.182.1.67 #define MMU_NORMAL_PT_UPDATE 0 /∗ checked '∗ptr = val'. ptr is MA. ∗/

4.182.1.68 #define MMU_PT_UPDATE_PRESERVE_AD 2 /∗ atomically: ∗ptr = val | (∗ptr&(A|D)) ∗/

4.182.1.69 #define MMUEXT_CLEAR_PAGE 16

4.182.1.70 #define MMUEXT_COPY_PAGE 17

4.182.1.71 #define MMUEXT_FLUSH_CACHE 12

4.182.1.72 #define MMUEXT_FLUSH_CACHE_GLOBAL 18

4.182.1.73 #define MMUEXT_INVLPG_ALL 11

4.182.1.74 #define MMUEXT_INVLPG_LOCAL 7

4.182.1.75 #define MMUEXT_INVLPG_MULTI 9

4.182.1.76 #define MMUEXT_MARK_SUPER 19

4.182.1.77 #define MMUEXT_NEW_BASEPTR 5

4.182.1.78 #define MMUEXT_NEW_USER_BASEPTR 15

4.182.1.79 #define MMUEXT_PIN_L1_TABLE 0

4.182.1.80 #define MMUEXT_PIN_L2_TABLE 1

4.182.1.81 #define MMUEXT_PIN_L3_TABLE 2

4.182.1.82 #define MMUEXT_PIN_L4_TABLE 3

4.182.1.83 #define MMUEXT_SET_LDT 13

4.182.1.84 #define MMUEXT_TLB_FLUSH_ALL 10

4.182.1.85 #define MMUEXT_TLB_FLUSH_LOCAL 6

4.182.1.86 #define MMUEXT_TLB_FLUSH_MULTI 8

4.182.1.87 #define MMUEXT_UNMARK_SUPER 20

4.182.1.88 #define MMUEXT_UNPIN_TABLE 4

4.182.1.89 #define NR_EVENT_CHANNELS (sizeof(unsigned long) ∗ sizeof(unsigned long) ∗ 64)

4.182.1.90 #define NR_VIRQS 24

4.182.1.91 #define SIF_INITDOMAIN (1<<1) /∗ Is this the initial control domain? ∗/

4.182.1.92 #define SIF_MOD_START_PFN (1<<3) /∗ Is mod_start a PFN? ∗/

4.182.1.93 #define SIF_MULTIBOOT_MOD (1<<2) /∗ Is mod_start a multiboot module? ∗/

4.182.1.94 #define SIF_PM_MASK (0xFF<<8) /∗ reserve 1 byte for xen-pm options ∗/

4.182.1.95 #define SIF_PRIVILEGED (1<<0) /* Is the domain privileged? */

4.182.1.96 #define UVMF_ALL (1UL<<2) /* Flush all TLBs. */

4.182.1.97 #define UVMF_FLUSHTYPE_MASK (3UL<<0)

4.182.1.98 #define UVMF_INVLPG (2UL<<0) /* Flush only one entry. */

4.182.1.99 #define UVMF_LOCAL (0UL<<2) /* Flush local TLB. */

4.182.1.100 #define UVMF_MULTI (0UL<<2) /* Flush subset of TLBs. */

4.182.1.101 #define UVMF_NONE (0UL<<0) /* No flushing at all. */

4.182.1.102 #define UVMF_TLB_FLUSH (1UL<<0) /* Flush entire TLB(s). */

4.182.1.103 #define VIRQ_ARCH_0 16

4.182.1.104 #define VIRQ_ARCH_1 17

4.182.1.105 #define VIRQ_ARCH_2 18

4.182.1.106 #define VIRQ_ARCH_3 19

4.182.1.107 #define VIRQ_ARCH_4 20

4.182.1.108 #define VIRQ_ARCH_5 21

4.182.1.109 #define VIRQ_ARCH_6 22

4.182.1.110 #define VIRQ_ARCH_7 23

4.182.1.111 #define VIRQ_CON_RING 8 /* G. (DOM0) Bytes received on console */

4.182.1.112 #define VIRQ_CONSOLE 2 /* G. (DOM0) Bytes received on emergency console. */

4.182.1.113 #define VIRQ_DEBUG 1 /* V. Request guest to dump debug info. */

4.182.1.114 #define VIRQ_DEBUGGER 6 /* G. (DOM0) A domain has paused for debugging. */

4.182.1.115 #define VIRQ_DOM_EXC 3 /* G. (DOM0) Exceptional event for some domain. */

4.182.1.116 #define VIRQ_MEM_EVENT 10 /* G. (DOM0) A memory event has occured */

4.182.1.117 #define VIRQ_PCPU_STATE 9 /* G. (DOM0) PCPU state changed */

4.182.1.118 #define VIRQ_TBUF 4 /* G. (DOM0) Trace buffer has records available. */

4.182.1.119 #define VIRQ_TIMER 0 /* V. Timebase update, and/or requested timeout. */

4.182.1.120 #define VIRQ_XENOPROF 7 /* V. XenOprofile interrupt: new sample available */

4.182.1.121 #define VMASST_CMD_disable 1

4.182.1.122 #define VMASST_CMD_enable 0

4.182.1.123 #define VMASST_TYPE_4gb_segments 0

4.182.1.124 #define VMASST_TYPE_4gb_segments_notify 1

4.182.1.125 #define VMASST_TYPE_pae_extended_cr3 3

4.182.1.126 #define VMASST_TYPE_writable_pagetables 2

4.182.1.127 #define xen_vga_console_info **dom0_vga_console_info**

4.182.1.128 #define xen_vga_console_info_t **dom0_vga_console_info_t**

4.182.1.129 #define XEN_VGATYPE_TEXT_MODE_3 0x03

4.182.1.130 #define XEN_VGATYPE_VESA_LFB 0x23

4.182.2 Typedef Documentation

4.182.2.1 typedef struct **dom0_vga_console_info dom0_vga_console_info_t**

4.182.2.2 typedef uint16_t **domid_t**

4.182.2.3 typedef struct **mmu_update mmu_update_t**

4.182.2.4 typedef struct **mmuext_op mmuext_op_t**

4.182.2.5 typedef struct **multicall_entry multicall_entry_t**

4.182.2.6 typedef struct **shared_info shared_info_t**

4.182.2.7 typedef struct **start_info start_info_t**

4.182.2.8 typedef struct **vcpu_info vcpu_info_t**

4.182.2.9 typedef struct **vcpu_time_info vcpu_time_info_t**

4.182.2.10 typedef uint8_t xen_domain_handle_t[16]

4.182.3 Function Documentation

4.182.3.1 __DEFINE_XEN_GUEST_HANDLE (uchar , unsigned *char*)

4.182.3.2 __DEFINE_XEN_GUEST_HANDLE (uint , unsigned *int*)

4.182.3.3 __DEFINE_XEN_GUEST_HANDLE (ulong , unsigned *long*)

4.182.3.4 __DEFINE_XEN_GUEST_HANDLE (uint8 , uint8_t)

4.182.3.5 __DEFINE_XEN_GUEST_HANDLE (uint16 , uint16_t)

4.182.3.6 __DEFINE_XEN_GUEST_HANDLE (uint32 , uint32_t)

4.182.3.7 __DEFINE_XEN_GUEST_HANDLE (uint64 , uint64_t)

4.182.3.8 DEFINE_XEN_GUEST_HANDLE (char)

4.182.3.9 DEFINE_XEN_GUEST_HANDLE (int)

4.182.3.10 DEFINE_XEN_GUEST_HANDLE (long)

4.182.3.11 DEFINE_XEN_GUEST_HANDLE (void)

4.182.3.12 DEFINE_XEN_GUEST_HANDLE (uint64_t)

4.182.3.13 DEFINE_XEN_GUEST_HANDLE (xen_pfn_t)

4.182.3.14 DEFINE_XEN_GUEST_HANDLE (mmuext_op_t)

4.182.3.15 DEFINE_XEN_GUEST_HANDLE (mmu_update_t)

4.182.3.16 DEFINE_XEN_GUEST_HANDLE (multicall_entry_t)

4.183 xen/xen/include/public/xsm/caas_op.h File Reference

Data Structures

- struct caas_op

Macros

- #define CAAS_EXEMPT 1
- #define CAAS_LAST CAAS_EXEMPT

Typedefs

- typedef struct caas_op caas_op_t

Functions

- DEFINE_XEN_GUEST_HANDLE (caas_op_t)

4.183.1 Macro Definition Documentation

4.183.1.1 #define CAAS_EXEMPT 1

4.183.1.2 #define CAAS_LAST CAAS_EXEMPT

4.183.2 Typedef Documentation

4.183.2.1 typedef struct caas_op caas_op_t

4.183.3 Function Documentation

4.183.3.1 DEFINE_XEN_GUEST_HANDLE (caas_op_t)

4.184 xen/xen/include/xen/domain.h File Reference

```
#include <public/xen.h>
#include <asm/domain.h>
```

Data Structures

- union vcpu_guest_context_u

Functions

- struct vcpu * alloc_vcpu (struct domain *d, unsigned int vcpu_id, unsigned int cpu_id)
- int boot_vcpu (struct domain *d, int vcpuid, vcpu_guest_context_u ctxt)
- struct vcpu * alloc_dom0_vcpu0 (void)
- struct vcpu * alloc_domt_vcpu0 (void)
- void vcpu_reset (struct vcpu *v)
- void getdomaininfo (struct domain *d, struct xen_domctl_getdomaininfo *info)
- struct domain * alloc_domain_struct (void)
- void free_domain_struct (struct domain *d)
- struct vcpu * alloc_vcpu_struct (void)
- void free_vcpu_struct (struct vcpu *v)
- int vcpu_initialise (struct vcpu *v)
- void vcpu_destroy (struct vcpu *v)
- int arch_domain_create (struct domain *d, unsigned int domcr_flags)
- void arch_domain_destroy (struct domain *d)
- int arch_set_info_guest (struct vcpu *, vcpu_guest_context_u)
- void arch_get_info_guest (struct vcpu *, vcpu_guest_context_u)
- int domain_relinquish_resources (struct domain *d)
- void dump_pageframe_info (struct domain *d)
- void arch_dump_vcpu_info (struct vcpu *v)
- void arch_dump_domain_info (struct domain *d)
- void arch_vcpu_reset (struct vcpu *v)
- bool_t domctl_lock_acquire (void)
- void domctl_lock_release (void)
- int continue_hypercall_on_cpu (unsigned int cpu, long(*func)(void *data), void *data)

Variables

- unsigned int xen_processor_pmbits

4.184.1 Function Documentation

4.184.1.1 struct vcpu* alloc_dom0_vcpu0 (void)

4.184.1.2 struct domain* alloc_domain_struct (void)

4.184.1.3 struct vcpu* alloc_domt_vcpu0 (void)

4.184.1.4 struct vcpu* alloc_vcpu (struct domain * *d*, unsigned int *vcpu_id*, unsigned int *cpu_id*)

4.184.1.5 struct vcpu* alloc_vcpu_struct (void)

4.184.1.6 int arch_domain_create (struct domain * d, unsigned int domcr_flags)

4.184.1.7 void arch_domain_destroy (struct domain * d)

4.184.1.8 void arch_dump_domain_info (struct domain * d)

4.184.1.9 void arch_dump_vcpu_info (struct vcpu * v)

4.184.1.10 void arch_get_info_guest (struct vcpu *, vcpu_guest_context_u)

4.184.1.11 int arch_set_info_guest (struct vcpu *, vcpu_guest_context_u)

4.184.1.12 void arch_vcpu_reset (struct vcpu * v)

4.184.1.13 int boot_vcpu (struct domain * d, int vcpuid, vcpu_guest_context_u ctxt)

4.184.1.14 int continue_hypercall_on_cpu (unsigned int cpu, long(*)(void *data) func, void * data)

Continue the current hypercall via func(data) on specified cpu. If this function returns 0 then the function is guaranteed to run at some point in the future. If this function returns an error code then the function has not been and will not be executed.

4.184.1.15 int domain_relinquish_resources (struct domain * d)

4.184.1.16 bool_t domctl_lock_acquire (void)

4.184.1.17 void domctl_lock_release (void)

4.184.1.18 void dump_pageframe_info (struct domain * d)

4.184.1.19 void free_domain_struct (struct domain * d)

4.184.1.20 void free_vcpu_struct (struct vcpu * v)

4.184.1.21 void getdomaininfo (struct domain * d, struct xen_domctl_getdomaininfo * info)

4.184.1.22 void vcpu_destroy (struct vcpu * v)

4.184.1.23 int vcpu_initialise (struct vcpu * v)

Initialise/destroy arch-specific details of a VCPU.

- vcpu_initialise() is called after the basic generic fields of the VCPU structure are initialised. Many operations can be applied to the VCPU at this point (e.g., vcpu_pause()).
- vcpu_destroy() is called only if vcpu_initialise() previously succeeded.

4.184.1.24 void vcpu_reset (struct vcpu * v)

4.184.2 Variable Documentation

4.184.2.1 unsigned int xen_processor_pmbits

4.185 xen/xen/include/xen/sched.h File Reference

```
#include <xen/config.h>
#include <xen/types.h>
#include <xen/spinlock.h>
#include <xen/smp.h>
#include <xen/shared.h>
#include <public/xen.h>
#include <public/domctl.h>
#include <public/sysctl.h>
#include <public/vcpu.h>
#include <public/xsm/acm.h>
#include <xen/time.h>
#include <xen/timer.h>
#include <xen/grant_table.h>
#include <xen/rangeset.h>
#include <xen/domain.h>
#include <xen/xenoprof.h>
#include <xen/rcupdate.h>
#include <xen/irq.h>
#include <xen/mm.h>
#include <xen/tasklet.h>
#include <public/mem_event.h>
#include <xen/cpumask.h>
#include <xen/nodemask.h>
#include <xen/multicall.h>
```

Data Structures

- struct evtchn
- struct vcpu
- struct mem_event_domain
- struct domain
- struct domain_setup_info

Macros

- #define BITS_PER_EVTCHN_WORD(d) BITS_PER_LONG
- #define MAX_EVTCHNS(d) (BITS_PER_EVTCHN_WORD(d) ∗ BITS_PER_EVTCHN_WORD(d))
- #define EVTCHNS_PER_BUCKET 128
- #define NR_EVTCHN_BUCKETS (NR_EVENT_CHANNELS / EVTCHNS_PER_BUCKET)
- #define ECS_FREE 0 /∗ Channel is available for use. ∗/
- #define ECS_RESERVED 1 /∗ Channel is reserved. ∗/
- #define ECS_UNBOUND 2 /∗ Channel is waiting to bind to a remote domain. ∗/
- #define ECS_INTERDOMAIN 3 /∗ Channel is bound to another domain. ∗/
- #define ECS_PIRQ 4 /∗ Channel is bound to a physical IRQ line. ∗/
- #define ECS_VIRQ 5 /∗ Channel is bound to a virtual IRQ line. ∗/
- #define ECS_IPI 6 /∗ Channel is bound to a virtual IPI line. ∗/
- #define runstate_guest(v) ((v)->runstate_guest)
- #define domain_lock(d) spin_lock_recursive(&(d)->domain_lock)
- #define domain_unlock(d) spin_unlock_recursive(&(d)->domain_lock)
- #define domain_is_locked(d) spin_is_locked(&(d)->domain_lock)
- #define NR_DOMAIN_WATCHDOG_TIMERS 2
- #define PAEKERN_no 0

- #define PAEKERN_yes 1
- #define PAEKERN_extended_cr3 2
- #define PAEKERN_bimodal 3
- #define is_idle_domain(d) ((d)->domain_id == DOMID_IDLE)
- #define is_idle_vcpu(v) (is_idle_domain((v)->domain))
- #define DOMAIN_DESTROYED (1<<31) /* assumes atomic_t is >= 32 bits */
- #define put_domain(_d) if (atomic_dec_and_test(&(_d)->refcnt)) domain_destroy(_d)
- #define _DOMCRF_hvm 0
- #define DOMCRF_hvm (1U<<_DOMCRF_hvm)
- #define _DOMCRF_hap 1
- #define DOMCRF_hap (1U<<_DOMCRF_hap)
- #define _DOMCRF_s3_integrity 2
- #define DOMCRF_s3_integrity (1U<<_DOMCRF_s3_integrity)
- #define _DOMCRF_dummy 3
- #define DOMCRF_dummy (1U<<_DOMCRF_dummy)
- #define _DOMCRF_oos_off 4
- #define DOMCRF_oos_off (1U<<_DOMCRF_oos_off)
- #define _DOMCRF_priv 5
- #define DOMCRF_priv (1U<<_DOMCRF_priv)
- #define domain_crash(d)
- #define domain_crash_synchronous()
- #define set_current_state(_s) do { current->state = (_s); } while (0)
- #define hypercall_preempt_check()
- #define for_each_domain(_d)
- #define for_each_vcpu(_d, _v)
- #define _VPF_blocked 0
- #define VPF_blocked (1UL<<_VPF_blocked)
- #define _VPF_down 1
- #define VPF_down (1UL<<_VPF_down)
- #define _VPF_blocked_in_xen 2
- #define VPF_blocked_in_xen (1UL<<_VPF_blocked_in_xen)
- #define _VPF_migrating 3
- #define VPF_migrating (1UL<<_VPF_migrating)
- #define _VPF_mem_event 4
- #define VPF_mem_event (1UL<<_VPF_mem_event)
- #define cpu_is_haltable(cpu)
- #define IS_PRIV(_d) ((_d)->is_privileged)
- #define IS_PRIV_FOR(_d, _t) (IS_PRIV(_d) || ((_d)->target && (_d)->target == (_t)))
- #define VM_ASSIST(_d, _t) (test_bit((_t), &(_d)->vm_assist))
- #define is_hvm_domain(d) ((d)->is_hvm)
- #define is_hvm_vcpu(v) (is_hvm_domain(v->domain))
- #define is_pinned_vcpu(v)
- #define need_iommu(d) ((d)->need_iommu)
- #define CPUPOOLID_NONE -1
- #define num_cpupool_cpus(c) (cpus_weight((c)->cpu_valid))

Enumerations

- enum cpufreq_controller { FREQCTL_none, FREQCTL_dom0_kernel, FREQCTL_xen }

Functions

- int evtchn_init (struct domain *d)
- void evtchn_destroy (struct domain *d)
- void evtchn_destroy_final (struct domain *d)
- void domain_update_node_affinity (struct domain *d)
- struct domain * domain_create (domid_t domid, unsigned int domcr_flags, ssidref_t ssidref)
- struct domain * rcu_lock_domain_by_id (domid_t dom)
- int rcu_lock_target_domain_by_id (domid_t dom, struct domain **d)
- int rcu_lock_remote_target_domain_by_id (domid_t dom, struct domain **d)
- struct domain * get_domain_by_id (domid_t dom)
- void domain_destroy (struct domain *d)
- int domain_kill (struct domain *d)
- void domain_shutdown (struct domain *d, u8 reason)
- void domain_resume (struct domain *d)
- void domain_pause_for_debugger (void)
- int vcpu_start_shutdown_deferral (struct vcpu *v)
- void vcpu_end_shutdown_deferral (struct vcpu *v)
- void __domain_crash (struct domain *d)
- void __domain_crash_synchronous (void) __attribute__((noreturn))
- void scheduler_init (void)
- int sched_init_vcpu (struct vcpu *v, unsigned int processor)
- void sched_destroy_vcpu (struct vcpu *v)
- int sched_init_domain (struct domain *d)
- void sched_destroy_domain (struct domain *d)
- int sched_move_domain (struct domain *d, struct cpupool *c)
- long sched_adjust (struct domain *, struct xen_domctl_scheduler_op *)
- long sched_adjust_global (struct xen_sysctl_scheduler_op *)
- int sched_id (void)
- void sched_tick_suspend (void)
- void sched_tick_resume (void)
- void vcpu_wake (struct vcpu *d)
- void vcpu_sleep_nosync (struct vcpu *d)
- void vcpu_sleep_sync (struct vcpu *d)
- void sync_vcpu_execstate (struct vcpu *v)
- void sync_local_execstate (void)
- void context_switch (struct vcpu *prev, struct vcpu *next)
- void context_saved (struct vcpu *prev)
- void continue_running (struct vcpu *same)
- void startup_cpu_idle_loop (void)
- unsigned long hypercall_create_continuation (unsigned int op, const char *format,...)
- void vcpu_unblock (struct vcpu *v)
- void vcpu_pause (struct vcpu *v)
- void vcpu_pause_nosync (struct vcpu *v)
- void domain_pause (struct domain *d)
- void vcpu_unpause (struct vcpu *v)
- void domain_unpause (struct domain *d)
- void domain_pause_by_systemcontroller (struct domain *d)
- void domain_unpause_by_systemcontroller (struct domain *d)
- void cpu_init (void)
- struct scheduler * scheduler_get_default (void)
- struct scheduler * scheduler_alloc (unsigned int sched_id, int *perr)
- void scheduler_free (struct scheduler *sched)
- int schedule_cpu_switch (unsigned int cpu, struct cpupool *c)
- void vcpu_force_reschedule (struct vcpu *v)

- int cpu_disable_scheduler (unsigned int cpu)
- int vcpu_set_affinity (struct vcpu *v, cpumask_t *affinity)
- void vcpu_runstate_get (struct vcpu *v, struct vcpu_runstate_info *runstate)
- uint64_t get_cpu_idle_time (unsigned int cpu)
- void watchdog_domain_init (struct domain *d)
- void watchdog_domain_destroy (struct domain *d)
- void set_vcpu_migration_delay (unsigned int delay)
- unsigned int get_vcpu_migration_delay (void)
- struct cpupool * cpupool_get_by_id (int poolid)
- void cpupool_put (struct cpupool *pool)
- int cpupool_add_domain (struct domain *d, int poolid)
- void cpupool_rm_domain (struct domain *d)
- int cpupool_do_sysctl (struct xen_sysctl_cpupool_op *op)

Variables

- struct domain * dom0
- struct domain * domt
- spinlock_t domlist_update_lock
- rcu_read_lock_t domlist_read_lock
- struct vcpu * idle_vcpu [NR_CPUS]
- struct domain * domain_list
- bool_t sched_smt_power_savings
- enum cpufreq_controller cpufreq_controller

4.185.1 Macro Definition Documentation

4.185.1.1 #define _DOMCRF_dummy 3

4.185.1.2 #define _DOMCRF_hap 1

4.185.1.3 #define _DOMCRF_hvm 0

4.185.1.4 #define _DOMCRF_oos_off 4

4.185.1.5 #define _DOMCRF_priv 5

4.185.1.6 #define _DOMCRF_s3_integrity 2

4.185.1.7 #define _VPF_blocked 0

4.185.1.8 #define _VPF_blocked_in_xen 2

4.185.1.9 #define _VPF_down 1

4.185.1.10 #define _VPF_mem_event 4

4.185.1.11 #define _VPF_migrating 3

4.185.1.12 #define BITS_PER_EVTCHN_WORD(d) BITS_PER_LONG

4.185.1.13 #define cpu_is_haltable(*cpu*)

Value:

```
(!softirq_pending(cpu) &&                    \
     cpu_online(cpu) &&                          \
     !per_cpu(tasklet_work_to_do, cpu))
```

4.185.1.14 #define CPUPOOLID_NONE -1

4.185.1.15 #define domain_crash(d)

Value:

```
do {                                                \
    printk("domain_crash called from %s:%d\n", __FILE__, __LINE__);        \
    __domain_crash(d);                                                      \
} while (0)
```

4.185.1.16 #define domain_crash_synchronous( )

Value:

```
do {                                      \
    printk("domain_crash_sync called from %s:%d\n", __FILE__, __LINE__);  \
    __domain_crash_synchronous();                                          \
} while (0)
```

4.185.1.17 #define DOMAIN_DESTROYED (1<<31) /* assumes atomic_t is >= 32 bits */

4.185.1.18 #define domain_is_locked(d) spin_is_locked(&(d)->domain_lock)

4.185.1.19 #define domain_lock(d) spin_lock_recursive(&(d)->domain_lock)

4.185.1.20 #define domain_unlock(d) spin_unlock_recursive(&(d)->domain_lock)

4.185.1.21 #define DOMCRF_dummy (1U<<_DOMCRF_dummy)

4.185.1.22 #define DOMCRF_hap (1U<<_DOMCRF_hap)

4.185.1.23 #define DOMCRF_hvm (1U<<_DOMCRF_hvm)

4.185.1.24 #define DOMCRF_oos_off (1U<<_DOMCRF_oos_off)

4.185.1.25 #define DOMCRF_priv (1U<<_DOMCRF_priv)

4.185.1.26 #define DOMCRF_s3_integrity (1U<<_DOMCRF_s3_integrity)

4.185.1.27 #define ECS_FREE 0 /* Channel is available for use. */

4.185.1.28 #define ECS_INTERDOMAIN 3 /* Channel is bound to another domain. */

4.185.1.29 #define ECS_IPI 6 /* Channel is bound to a virtual IPI line. */

4.185.1.30 #define ECS_PIRQ 4 /* Channel is bound to a physical IRQ line. */

4.185.1.31 #define ECS_RESERVED 1 /* Channel is reserved. */

4.185.1.32 #define ECS_UNBOUND 2 /* Channel is waiting to bind to a remote domain. */

4.185.1.33 #define ECS_VIRQ 5 /* Channel is bound to a virtual IRQ line. */

4.185.1.34 #define EVTCHNS_PER_BUCKET 128

4.185.1.35 #define for_each_domain(_d)

Value:

```
for ( (_d) = rcu_dereference(domain_list);      \
       (_d) != NULL;                            \
       (_d) = rcu_dereference((_d)->next_in_list )) \
```

4.185.1.36 #define for_each_vcpu(_d, _v)

Value:

```
for ( (_v) = (_d)->vcpu ? (_d)->vcpu[0] : NULL; \
       (_v) != NULL;                            \
       (_v) = (_v)->next_in_list )
```

4.185.1.37 #define hypercall_preempt_check( )

Value:

```
(unlikely(    \
        softirq_pending(smp_processor_id()) |    \
        local_events_need_delivery()             \
    ))
```

4.185.1.38 #define is_hvm_domain(d) ((d)->is_hvm)

4.185.1.39 #define is_hvm_vcpu(v) (is_hvm_domain(v->domain))

4.185.1.40 #define is_idle_domain(d) ((d)->domain_id == DOMID_IDLE)

4.185.1.41 #define is_idle_vcpu(v) (is_idle_domain((v)->domain))

4.185.1.42 #define is_pinned_vcpu(v)

Value:

```
((v)->domain->is_pinned || \
                           cpus_weight((v)->cpu_affinity) == 1)
```

4.185.1.43 #define IS_PRIV(_d) ((_d)->is_privileged)

4.185.1.44 #define IS_PRIV_FOR(_d, _t) (IS_PRIV(_d) || ((_d)->target && (_d)->target == (_t)))

4.185.1.45 #define MAX_EVTCHNS(d) (BITS_PER_EVTCHN_WORD(d) * BITS_PER_EVTCHN_WORD(d))

4.185.1.46 #define need_iommu(d) ((d)->need_iommu)

4.185.1.47 #define NR_DOMAIN_WATCHDOG_TIMERS 2

4.185.1.48 #define NR_EVTCHN_BUCKETS (NR_EVENT_CHANNELS / EVTCHNS_PER_BUCKET)

4.185.1.49 #define num_cpupool_cpus(c) (cpus_weight((c)->cpu_valid))

4.185.1.50 #define PAEKERN_bimodal 3

4.185.1.51 #define PAEKERN_extended_cr3 2

4.185.1.52 #define PAEKERN_no 0

4.185.1.53 #define PAEKERN_yes 1

4.185.1.54 #define put_domain(_d) if (atomic_dec_and_test(&(_d)->refcnt)) domain_destroy(_d)

4.185.1.55 #define runstate_guest(v) ((v)->runstate_guest)

4.185.1.56 #define set_current_state(_s) do { current->state = (_s); } while (0)

4.185.1.57 #define VM_ASSIST(_d, _t) (test_bit((_t), &(_d)->vm_assist))

4.185.1.58 #define VPF_blocked (1UL<<_VPF_blocked)

4.185.1.59 #define VPF_blocked_in_xen (1UL<<_VPF_blocked_in_xen)

4.185.1.60 #define VPF_down (1UL<<_VPF_down)

4.185.1.61 #define VPF_mem_event (1UL<<_VPF_mem_event)

4.185.1.62 #define VPF_migrating (1UL<<_VPF_migrating)

4.185.2 Enumeration Type Documentation

4.185.2.1 enum cpufreq_controller

Enumerator

FREQCTL_none

FREQCTL_dom0_kernel

FREQCTL_xen

4.185.3 Function Documentation

4.185.3.1 void __domain_crash (struct domain * d)

Mark specified domain as crashed. This function always returns, even if the caller is the specified domain. The domain is not synchronously descheduled from any processor.

4.185.3.2 void __domain_crash_synchronous (void)

Mark current domain as crashed and synchronously deschedule from the local processor. This function never returns.

4.185.3.3 void context_saved (struct vcpu * *prev*)

As described above, context_switch() must call this function when the local CPU is no longer running in 's context, and 's context is saved to memory. Alternatively, if implementing lazy context switching, ensure that invoking sync_-vcpu_execstate() will switch and commit .

4.185.3.4 void context_switch (struct vcpu * *prev,* struct vcpu * *next*)

Called by the scheduler to switch to another VCPU. This function must call context_saved() when the local CPU is no longer running in 's context, and that context is saved to memory. Alternatively, if implementing lazy context switching, it suffices to ensure that invoking sync_vcpu_execstate() will switch and commit 's state.

4.185.3.5 void continue_running (struct vcpu * *same*)

Called by the scheduler to continue running the current VCPU.

4.185.3.6 int cpu_disable_scheduler (unsigned int *cpu*)

4.185.3.7 void cpu_init (void)

4.185.3.8 int cpupool_add_domain (struct domain * *d,* int *poolid*)

4.185.3.9 int cpupool_do_sysctl (struct xen_sysctl_cpupool_op * *op*)

4.185.3.10 struct cpupool* cpupool_get_by_id (int *poolid*)

4.185.3.11 void cpupool_put (struct cpupool * *pool*)

4.185.3.12 void cpupool_rm_domain (struct domain * *d*)

4.185.3.13 struct domain* domain_create (domid_t *domid,* unsigned int *domcr_flags,* ssidref_t *ssidref*)

4.185.3.14 void domain_destroy (struct domain * *d*)

Release resources belonging to task .

4.185.3.15 int domain_kill (struct domain * *d*)

4.185.3.16 void domain_pause (struct domain * *d*)

4.185.3.17 void domain_pause_by_systemcontroller (struct domain * *d*)

4.185.3.18 void domain_pause_for_debugger (void)

4.185.3.19 void domain_resume (struct domain * *d*)

4.185.3.20 void domain_shutdown (struct domain * *d,* u8 *reason*)

4.185.3.21 void domain_unpause (struct domain * *d*)

4.185.3.22 void domain_unpause_by_systemcontroller (struct domain * *d*)

4.185.3.23 void domain_update_node_affinity (struct domain * *d*)

4.185.3.24 void evtchn_destroy (struct domain * d)

4.185.3.25 void evtchn_destroy_final (struct domain * d)

4.185.3.26 int evtchn_init (struct domain * d)

4.185.3.27 uint64_t get_cpu_idle_time (unsigned int cpu)

4.185.3.28 struct domain* get_domain_by_id (domid_t dom)

4.185.3.29 unsigned int get_vcpu_migration_delay (void)

4.185.3.30 unsigned long hypercall_create_continuation (unsigned int op, const char * format, ...)

Creates a continuation to resume the current hypercall. The caller should return immediately, propagating the value returned from this invocation. The format string specifies the types and number of hypercall arguments. It contains one character per argument as follows: 'i' [unsigned] {char, int} 'l' [unsigned] long 'h' guest handle (XEN_GUEST_-HANDLE(foo))

4.185.3.31 struct domain* rcu_lock_domain_by_id (domid_t dom)

rcu_lock_domain_by_id() is more efficient than get_domain_by_id(). This is the preferred function if the returned domain reference is short lived, but it cannot be used if the domain reference needs to be kept beyond the current scope (e.g., across a softirq). The returned domain reference must be discarded using rcu_unlock_domain().

4.185.3.32 int rcu_lock_remote_target_domain_by_id (domid_t dom, struct domain ** d)

As rcu_lock_target_domain_by_id(), but will fail EPERM rather than resolve to local domain. Successful return always resolves to a remote domain that the local domain is privileged to control.

4.185.3.33 int rcu_lock_target_domain_by_id (domid_t dom, struct domain ** d)

As above function, but accounts for current domain context:

- Translates target DOMID_SELF into caller's domain id; and
- Checks that caller has permission to act on the target domain.

4.185.3.34 long sched_adjust (struct domain * , struct xen_domctl_scheduler_op *)

4.185.3.35 long sched_adjust_global (struct xen_sysctl_scheduler_op *)

4.185.3.36 void sched_destroy_domain (struct domain * d)

4.185.3.37 void sched_destroy_vcpu (struct vcpu * v)

4.185.3.38 int sched_id (void)

4.185.3.39 int sched_init_domain (struct domain * d)

4.185.3.40 int sched_init_vcpu (struct vcpu * v, unsigned int processor)

4.185.3.41 int sched_move_domain (struct domain * d, struct cpupool * c)

4.185.3.42 void sched_tick_resume (void)

4.185.3.43 void sched_tick_suspend (void)

4.185.3.44 int schedule_cpu_switch (unsigned int *cpu,* struct cpupool ∗ *c*)

4.185.3.45 struct scheduler∗ scheduler_alloc (unsigned int *sched_id,* int ∗ *perr*)

4.185.3.46 void scheduler_free (struct scheduler ∗ *sched*)

4.185.3.47 struct scheduler∗ scheduler_get_default (void)

4.185.3.48 void scheduler_init (void)

4.185.3.49 void set_vcpu_migration_delay (unsigned int *delay*)

4.185.3.50 void startup_cpu_idle_loop (void)

4.185.3.51 void sync_local_execstate (void)

As above, for any lazy state being held on the local CPU.

4.185.3.52 void sync_vcpu_execstate (struct **vcpu** ∗ *v*)

Force synchronisation of given VCPU's state. If it is currently descheduled, this call will ensure that all its state is committed to memory and that no CPU is using critical state (e.g., page tables) belonging to the VCPU.

4.185.3.53 void vcpu_end_shutdown_deferral (struct **vcpu** ∗ *v*)

4.185.3.54 void vcpu_force_reschedule (struct **vcpu** ∗ *v*)

4.185.3.55 void vcpu_pause (struct **vcpu** ∗ *v*)

4.185.3.56 void vcpu_pause_nosync (struct **vcpu** ∗ *v*)

4.185.3.57 void vcpu_runstate_get (struct **vcpu** ∗ *v,* struct vcpu_runstate_info ∗ *runstate*)

4.185.3.58 int vcpu_set_affinity (struct **vcpu** ∗ *v,* cpumask_t ∗ *affinity*)

4.185.3.59 void vcpu_sleep_nosync (struct **vcpu** ∗ *d*)

4.185.3.60 void vcpu_sleep_sync (struct **vcpu** ∗ *d*)

4.185.3.61 int vcpu_start_shutdown_deferral (struct **vcpu** ∗ *v*)

4.185.3.62 void vcpu_unblock (struct **vcpu** ∗ *v*)

4.185.3.63 void vcpu_unpause (struct **vcpu** ∗ *v*)

4.185.3.64 void vcpu_wake (struct **vcpu** ∗ *d*)

4.185.3.65 void watchdog_domain_destroy (struct **domain** ∗ *d*)

4.185.3.66 void watchdog_domain_init (struct **domain** ∗ *d*)

4.185.4 Variable Documentation

4.185.4.1 enum cpufreq_controller cpufreq_controller

4.185.4.2 struct domain* dom0

4.185.4.3 struct domain* domain_list

4.185.4.4 rcu_read_lock_t domlist_read_lock

4.185.4.5 spinlock_t domlist_update_lock

4.185.4.6 struct domain* domt

4.185.4.7 struct vcpu* idle_vcpu[NR_CPUS]

4.185.4.8 bool_t sched_smt_power_savings

4.186 xen/xen/include/xsm/xsm.h File Reference

```
#include <xen/sched.h>
#include <xen/multiboot.h>
```

Macros

- #define xsm_call(fn) 0
- #define XSM_MAGIC 0x00000000

Typedefs

- typedef void xsm_op_t
- typedef u32 xsm_magic_t

Functions

- DEFINE_XEN_GUEST_HANDLE (xsm_op_t)

Variables

- struct xsm_operations * xsm_ops

4.186.1 Macro Definition Documentation

4.186.1.1 #define xsm_call(*fn*) 0

4.186.1.2 #define XSM_MAGIC 0x00000000

4.186.2 Typedef Documentation

4.186.2.1 typedef u32 xsm_magic_t

4.186.2.2 typedef void xsm_op_t

4.186.3 Function Documentation

4.186.3.1 DEFINE_XEN_GUEST_HANDLE (xsm_op_t)

4.186.4 Variable Documentation

4.186.4.1 struct xsm_operations* xsm_ops

4.187 xen/xen/xsm/caas/hooks.c File Reference

```
#include <xen/init.h>
#include <xen/lib.h>
#include <xen/sched.h>
#include <xen/paging.h>
#include <xen/xmalloc.h>
#include <xsm/xsm.h>
#include <xen/spinlock.h>
#include <xen/cpumask.h>
#include <xen/errno.h>
#include <xen/guest_access.h>
#include <public/xen.h>
#include <public/physdev.h>
#include <public/platform.h>
#include <public/memory.h>
#include <public/xsm/caas_op.h>
```

Macros

- #define DOMID_DZ 0
- #define DOMID_DT 1
- #define CAAS_OK 0
- #define CAAS_COPY_IN
- #define CAAS_COPY_OUT
- #define CAAS_DEVELOP
- #define EXBUF_N 4
- #define CAAS_DENY(d)
- #define CAAS_ALLOWED_Z(d)
- #define CAAS_ALLOWED_Z_DUO(d1, d2)
- #define CAAS_ALLOWED_T(d)
- #define CAAS_ALLOWED_T_DUO(d1, d2)
- #define CAAS_ALLOWED_NONE(d) CAAS_DENY(d)
- #define CAAS_PASS(d)

Functions

- integer_param ("caas_enforcing", caas_enforcing)
- xsm_initcall (caas_init)

Variables

- uint64_t exbuf [EXBUF_N]
- int exbuf_i = 0
- struct xsm_operations * original_ops = NULL

4.187.1 Macro Definition Documentation

4.187.1.1 #define CAAS_ALLOWED_NONE(d) CAAS_DENY(d)

4.187.1.2 #define CAAS_ALLOWED_T(d)

Value:

```
do {                                                                    \
    if (d->domain_id != DOMID_DT) { CAAS_DENY(d); }                         \
    } while(0)
```

4.187.1.3 #define CAAS_ALLOWED_T_DUO(*d1, d2*)

Value:

```
do {                                                                    \
    if (d1->domain_id != d2->domain_id)                                     \
        CAAS_ALLOWED_T(d1);                                                 \
    } while(0)
```

4.187.1.4 #define CAAS_ALLOWED_Z(d)

Value:

```
do {                                                                    \
    if (d->domain_id != DOMID_DZ) { CAAS_DENY(d); }                         \
    } while(0)
```

4.187.1.5 #define CAAS_ALLOWED_Z_DUO(*d1, d2*)

Value:

```
do {                                                                    \
    if (d1->domain_id != d2->domain_id)                                     \
        CAAS_ALLOWED_Z(d1);                                                 \
    } while(0)
```

4.187.1.6 #define CAAS_COPY_IN

Value:

```
( \
        1UL<<CAAS_EXEMPT \
         )
```

4.187.1.7 #define CAAS_COPY_OUT

Value:

```
( \
        1UL<<CAAS_OK \
    )
```

4.187.1.8 #define CAAS_DENY(d)

Value:

```
do {                                                                \
    printk("CaaS: Denied access for %d in %s\n", d->domain_id, __func__);   \
    if (!caas_enforcing)                                            \
        printk("CaaS: not in enforcing mode!");                     \
    else                                                            \
        return 1;                                                   \
    } while(0)
```

4.187.1.9 #define CAAS_DEVELOP

4.187.1.10 #define CAAS_OK 0

4.187.1.11 #define CAAS_PASS(d)

Value:

```
do {                                                                \
    printk("CaaS: PASSED access for %d in %s\n", d->domain_id, __func__);   \
    return 0;                                                       \
    } while(0)
```

4.187.1.12 #define DOMID_DT 1

4.187.1.13 #define DOMID_DZ 0

4.187.1.14 #define EXBUF_N 4

4.187.2 Function Documentation

4.187.2.1 integer_param ("caas_enforcing" , caas_enforcing)

4.187.2.2 xsm_initcall (caas_init)

4.187.3 Variable Documentation

4.187.3.1 uint64_t exbuf[EXBUF_N]

4.187.3.2 int exbuf_i = 0

4.187.3.3 struct xsm_operations* original_ops = NULL

4.188 xen/xen/xsm/dummy.c File Reference

```
#include <xen/sched.h>
#include <xsm/xsm.h>
```

Macros

- #define set_to_dummy_if_null(ops, function)

Functions

- void xsm_fixup_ops (struct xsm_operations *ops)

Variables

- struct xsm_operations dummy_xsm_ops

4.188.1 Macro Definition Documentation

4.188.1.1 #define set_to_dummy_if_null(*ops, function*)

Value:

```
do {                                                            \
        if ( !ops->function )                                   \
        {                                                       \
            ops->function = dummy_##function;                   \
            dprintk(XENLOG_DEBUG, "Had to override the " #function \
                " security operation with the dummy one.\n");   \
        }                                                       \
    } while (0)
```

4.188.2 Function Documentation

4.188.2.1 void xsm_fixup_ops (struct xsm_operations * *ops*)

4.188.3 Variable Documentation

4.188.3.1 struct xsm_operations dummy_xsm_ops

4.189 xen/xen/xsm/xsm_core.c File Reference

```
#include <xen/init.h>
#include <xen/errno.h>
#include <xen/lib.h>
#include <xen/hypercall.h>
#include <xsm/xsm.h>
```

Functions

- long do_xsm_op (XEN_GUEST_HANDLE(xsm_op_t) op)

4.189.1 Function Documentation

4.189.1.1 long do_xsm_op (XEN_GUEST_HANDLE(xsm_op_t) *op*)