Release Notes for

FreeBSD package version QAT1.7.B.3.5.0-00011.tar.gz (December 2019)

The documentation for this production release is provided in this note. It can be read in conjunction with these documents:

- Intel® Communications Chipset 8925 to 8955 Series Software - Programmer's Guide
- Intel® Communications Chipset 89xx Series Software for Linux – Getting Started Guide
- Intel® QuickAssist Technology Software for Linux* - Programmer's Guide - HW version 1.7
- Intel® QuickAssist Technology Software for Linux* - Getting Started Guide - HW version 1.7

Contents

Revision History 2
Related Documentation 2
Release Overview 3
The release delivers the following features: 3
Environmental Assumptions: 3
Limitations with this production release: 3
MD5 Checksum Information 4
Licensing for FBSD* Acceleration Software 4
QuickAssist Driver Package Installation on FreeBSD Environment 4
Compiling the Driver 4
Compiling and execute performance sample code 5
Uninstalling the driver 6
EC Point Multiplication functional sample code compilation and execution 6
API's in this release: 6
EC Point Multiplication Functions: 7
EC Point Multiplication Parameters: 7
EC Point Multiplication use case 8
Perform EC Point Multiplication operation 10
HKDF API operational data update: 11
Known Issues 11
QATE-52976 – AlgChain and HKDF threads cannot use the same cy instance 11

QATE-31888 – Possible performance degradation 11

QATE-30931 - Process exit with orphan rings when spawning multiple processes 12

QATE-30360 - Full device pass-through not available on KVM guests 12

QATE-5092 - AES-XTS does not support buffers sizes that are not a multiple of 16B 12

QATE-7325 - AES-GCM operation with zero length plain text results in an incorrect tag result 12

QATE-33751 - Library and driver do not support devices enumerated in a PCI domain different than 0 13

QATE-39335 - Compression instances do not work on Virtual Machine with Linux Host QAT driver without CnVnR support 13

QATE-40359 - Multiprocess 32 with LimitDevAccess = 0 fails with OpenSSL Speed tests 13

QATE-39216 - Kasumi test duration issue 13

QATE-41846 - GEN - QAT API submissions with bad addresses that trigger DMA to invalid or unmapped addresses can cause a platform hang 14

QATE-41486 - Misleading message observed in dmesg on LBG device with LimitDevAccess = 1 set in configuration file. 14

Revision History

Date	Revision	Description
October 2019	001	Initial 3.5.0 Product release

Related Documentation

Document Name	Reference Number
Intel® QuickAssist Technology API Programmer's Guide	330684
Intel® QuickAssist Technology Cryptographic API Reference Manual	330685
Intel® QuickAssist Technology Data Compression API Reference Manual	330686
Intel® QuickAssist Technology Performance Optimization Guide	330687
Using Intel® Virtualization Technology (Intel® VT) with Intel® QuickAssist Technology Application Note	330689

Release Overview

The QAT R3.5.0 FreeBSD package is provided as production quality release and is therefore intended to be used in a production environment.
This software release is intended for platforms that contain:

- Intel® C62x Chipset
- Intel Atom® C3000 processor product family
- Intel® QuickAssist Adapter 8960/Intel® QuickAssist Adapter 8970 (formerly known as “Lewis Hill”)
- Intel® Communications Chipset 8925 to 8955 Series

The release delivers the following features:

- Montgomery and Edwards 25519 and 448 Elliptic Curve Point Multiplication support

Environmental Assumptions:

The following assumptions are made with regard to the deployment environment

- The driver object/executable file on disk should be protected using the normal file protection mechanisms so that it is writable only by trusted users, for example, a privileged user or an administrator.
- The public key firmware image on disk should be protected using normal file protection mechanisms so that it is writable only by trusted users, for example, a privileged user or an administrator.
- The QAT device should not be exposed (via SR-IOV) to untrusted guests.
- The QAT device should not be exposed (via the "user space direct" deployment model) to untrusted users.
- DRAM is considered to be inside the trust boundary. The normal memory protection schemes provided by the Intel® architecture processor and memory controller, and by the operating system, prevent unauthorized access to these memory regions.
- Persistent keys were not considered, but the storage media are also considered inside the cryptographic boundary. For the details please refer to: Intel® QuickAssist Technology (Intel® QAT) Software for Linux.

Limitations with this production release:

- FreeBSD as a host environment with QAT is not supported
- Any version of FreeBSD other than 11.2 is not supported
- Symmetric session update feature is not supported
- NRBG is not supported
- The HKDF operational data has to be allocated with USDM to be pinned in physical memory

There are known issues with this release of the driver as described in

.

MD5 Checksum Information

The table below gives MD5 checksum information.

	Package	Checksum
QAT Package	QAT1.7.B.3.5.0-00011.tar.gz	5b3aba095a9ff8355910f6a7dd668550

Licensing for FBSD* Acceleration Software

The acceleration software is provided under the following license as listed in the table below. When using or redistributing dual-licensed components, you may do so under either license.

Component	Licence	Directories
User Space Library	BSD	./quickassist/build_system ./quickassist/include ./quickassist/lookaside ./quickassist/utilities/osal
Kernel space driver	BSD	./quickassist/qat/drivers ./quickassist/utilities/adf_ctl
User Space DMA-able Memory Driver	BSD	./quickassist/utilities/libusdm
Libcrypto	OpenSSL	./quickassist/utilities/osal/src/linux/user_space/openssl
CPM Firmware	Redistribution	./quickassist/qat/fw

QuickAssist Driver Package Installation on FreeBSD Environment

User must have root privileges to perform the following.

Compiling the Driver

Step 1: Copy package onto the system.

Step 2: Extract package.

```
# cd /root/
# mkdir QAT
# cd QAT
# tar -xzomf <path_to>/ QAT1.7.B.3.5.0-00011.tar.gz
```

Step 3: Set network proxy (if required)

```
# setenv http_proxy http://<proxy_server>:<proxy_port>
```

Step 4: Install dependencies

1. gmake:
   # cd /usr/ports/devel/gmake
   # make config-recursive
   # make install
2. Boost
   # pkg install boost-all
3. Automake & autoconf
   # pkg install automake
   # pkg install autoconf
4. Bash
   # pkg install bash
5. pkg-config
   # cd /usr/ports/devel/pkgconf/
   # make
   # make install

Step 5: Setup the environment to build driver.

```
# cd /root/QAT/
# ./configure
```

Step 6: Build and install driver

```
# gmake install
```

Compiling and execute performance sample code

Step 1: Build application

```
# cd /root/QAT/
# gmake samples-install
```

Step 2: Run application

```
# cd ./build
# ./cpa_sample_code signOfLife=1  <- sign of life tests
# ./cpa_sample_code               <- full application run
```

Uninstalling the driver

Step 1: Bring down the driver

```
# cd /root/QAT/build
# ./adf_ctl down
```

Step 2: Uninstall driver

```
# cd /root/QAT/
# gmake uninstall
```

EC Point Multiplication functional sample code compilation and execution

EC Point Multiplication functional sample code could be built in following ways:

1. Standalone EC Point Multiplication functional sample built:

```
# cd /root/QAT
# setenv ICP_ROOT `pwd`
# setenv ICP_OS freebsd
# setenv WITH_CMDRV 1
# cd ./quickassist/lookaside/access_layer/src/sample_code/functional/asym/
ec_montedwds_sample/
# gmake
# ./ec_montedwds_sample
```

2. Build all functional samples:

```
# cd /root/QAT
# setenv ICP_ROOT `pwd`
# setenv ICP_OS freebsd
# setenv WITH_CMDRV 1
# cd ./quickassist/lookaside/access_layer/src/sample_code/
# gmake func
# cd ./functional/build
# ./ec_montedwds_sample
```

The sample is located:

```
# ./asym/ec_montedwds_sample/ec_montedwds_sample
```

API's in this release:

This is a list of the new Montgomery and Edwards 25519 and 448 Elliptic Curve Point Multiplication API's provided in this release.

EC Point Multiplication Functions:

Provides Elliptic Curves Point Multiplication interface for Montgomery and Edwards curves.

CpaStatus
cpaCyEcMontEdwdsPointMultiply(const CpaInstanceHandle instanceHandle,
const CpaCyEcPointMultiplyCbFunc pCb,
*void *pCallbackTag,*
*const CpaCyEcMontEdwdsPointMultiplyOpData *pOpData,*
*CpaBoolean *pMultiplyStatus,*
*CpaFlatBuffer *pXk,*
*CpaFlatBuffer *pYk);*

EC Point Multiplication Parameters:

CpaInstanceHandle instanceHandle – handle to the crypto instance.

const *CpaCyEcPointMultiplyCbFunc pCb* – pointer to completion callback function.

void *pCallbackTag – opaque user data that will be passed unchanged in the callback.

```
typedef enum _CpaCyEcMontEdwdsCurveType
{
   CPA_CY_EC_MONTEDWDS_CURVE25519_TYPE = 1,
   /**< Montgomery 25519 curve */
   CPA_CY_EC_MONTEDWDS_ED25519_TYPE,
   /**< Twisted Edwards 25519 curve */
   CPA_CY_EC_MONTEDWDS_CURVE448_TYPE,
   /**< Montgomery 448 curve */
   CPA_CY_EC_MONTEDWDS_ED448_TYPE,
   /**< Twisted Edwards 448 curve */
} CpaCyEcMontEdwdsCurveType;

typedef struct _CpaCyEcMontEdwdsPointMultiplyOpData {
   CpaCyEcMontEdwdsCurveType curveType;
   /**< field type for the operation */
   CpaBoolean  generator;
   /**< True if the operation is a generator multiplication (kG)
    *   False if it is a variable point multiplcation (kP). */
   CpaFlatBuffer  k;
   /**< k or generator for the operation */
   CpaFlatBuffer  x;
   /**< x value.  Used in scalar varable point multiplication operations.
    * Not required if the generator is True. Must be NULL if not required.
    * The size of the buffer MUST be 32B for 25519 curves and 64B for 448
```

```
    * curves */
   CpaFlatBuffer  y;
   /**< y value.  Used in variable point multiplication of operations.
    * Not required for curves defined only on scalar operations.
    * Not required if the generator is True.
    * Must be NULL if not required.
    * The size of the buffer MUST be 32B for 25519 curves and 64B for 448
    * curves */
} CpaCyEcMontEdwdsPointMultiplyOpData;
```

const CpaCyEcMontEdwdsPointMultiplyOpData * *pOpData* – input structure with data needed to perform TLS key generation operation. This structure has to be allocated with USDM to be pinned in physical memory.

*CpaBoolean * pMultiplyStatus* – handle to multiply status in synchronous mode.

*CpaFlatBuffer * pXk* - handle to output buffer with generated data. Caller MUST allocate sufficient buffer to hold the key generation output.

*CpaFlatBuffer * pYk* - handle to output buffer with generated data. Caller MUST allocate sufficient buffer to hold the key generation output.

For details on any changes to the Intel® QuickAssist Technology APIs, refer to the Revision History pages in the following API reference manuals:
• Intel® QuickAssist Technology Cryptographic API Reference Manual
• Intel® QuickAssist Technology Data Compression API Reference Manual

EC Point Multiplication use case

This is sample code that demonstrates usage of the asymmetric API, and specifically using this API to perform an EC Point Multiplication based operations. It performs: variable point multiplication on Twisted Edwards 448 curve, generator multiplication on Montgomery 448 curve, variable point multiplication on Montgomery 25519 curve.

Note this example is simplified to demonstrate the basics of how to use the API and how to build the structures required. This example does not demonstrate the optimal way to use the API to get maximum performance for a particular implementation.

This sample is located in:
./quickassist/lookaside/access_layer/src/sample_code/functional/asym/ec_montedwds_sample

Instance configuration and memory allocation

- Cryptographic service instances are discovered and started in the same way and using the same API as the traditional symmetric use cases.
- If the instance is polled start the polling thread. Note that how the polling is done is implementation-dependent.
- Allocate memory for EC Point Multiplication operation data and x, y, k points buffers:
  OS_MALLOC(&opData, sizeof(CpaCyEcMontEdwdsPointMultiplyOpData));
  PHYS_CONTIG_ALLOC_ALIGNED(
  &opData->x.pData, dataLenInBytes, BYTE_ALIGNMENT_64);
  PHYS_CONTIG_ALLOC_ALIGNED(
  &opData->y.pData, dataLenInBytes, BYTE_ALIGNMENT_64);
  PHYS_CONTIG_ALLOC_ALIGNED(
  &opData->k.pData, dataLenInBytes, BYTE_ALIGNMENT_64);

- Allocate memory for EC Point Multiplication *pXk* and *pYk* output data. Output data is CpaFlatBuffer type:
  OS_MALLOC(&pXk, sizeof(CpaFlatBuffer));
  OS_MALLOC(&pYk, sizeof(CpaFlatBuffer));
  PHYS_CONTIG_ALLOC_ALIGNED(&pXk->pData, dataLenInBytes, BYTE_ALIGNMENT_64);
  PHYS_CONTIG_ALLOC_ALIGNED(&pYk->pData, dataLenInBytes, BYTE_ALIGNMENT_64);

Point multiplication on Twisted Edwards 448 curve

- To perform point multiplication on Twisted Edwards 448 curve, in *CpaCyEcMontEdwdsPointMultiplyOpData* structure *generator* have to be set to *FALSE* and *curveType* have to be set to *CPA_CY_EC_MONTEDWDS_ED448_TYPE*. Length of *x, y, k* have to be provided and all data copied into *x, y, k*.

  opData->generator = FALSE;
  opData->curveType = CPA_CY_EC_MONTEDWDS_ED448_TYPE

  opData->x.dataLenInBytes = dataLenInBytes;
  memcpy(opData->x.pData, pointX, dataLenInBytes);

  opData->y.dataLenInBytes = dataLenInBytes;
  memcpy(opData->y.pData, pointY, dataLenInBytes);

  opData->k.dataLenInBytes = dataLenInBytes;
  memcpy(opData->k.pData, pointK, dataLenInBytes);

Generator multiplication on Montgomery 448 curve

- To perform generator multiplication on Montgomery 448 curve, in *CpaCyEcMontEdwdsPointMultiplyOpData* structure generator have to be set to TRUE and curveType have to be set to CPA_CY_EC_MONTEDWDS_CURVE448_TYPE. Length of k have

to be provided and all data copied into k.

opData->generator = TRUE;
opData->curveType = CPA_CY_EC_MONTEDWDS_ED448_TYPE

opData->k.dataLenInBytes = dataLenInBytes;
memcpy(opData->k.pData, pointK, dataLenInBytes);

Point multiplication on Montgomery 25519 curve

- To perform point multiplication on Montgomery 25519 curve, in *CpaCyEcMontEdwdsPointMultiplyOpData* structure *generator* have to be set to *FALSE* and *curveType* have to be set to *CPA_CY_EC_MONTEDWDS_CURVE25519_TYPE*. Length of *x, k* have to be provided and all data copied into *x, k*.

  opData->generator = FALSE;
  opData->curveType = CPA_CY_EC_MONTEDWDS_CURVE25519_TYPE

  opData->x.dataLenInBytes = dataLenInBytes;
  memcpy(opData->x.pData, pointX, dataLenInBytes);

  opData->k.dataLenInBytes = dataLenInBytes;
  memcpy(opData->k.pData, pointK, dataLenInBytes);

Perform EC Point Multiplication operation

- To execute the EC Point Multiplication operation, the crypto instance must be specified in the *instanceHandle*. When the operation is performed asynchronously, the callback function and callback tag should be set in *CpaCyEcPointMultiplyCbFunc* and *pCallbackTag* arguments. Operational data is provided in CpaCyEcMontEdwdsPointMultiplyOpData. In synchronous mode multiply status has to be provided in *pMultiplyStatus*. The output will be passed to *pXk* and *pYk CpaFlatBuffer*. Depending on what operations are performed the ouput buffers length should be adjusted.

```
cpaCyEcMontEdwdsPointMultiply(cyInstHandle, /* Instance handle */
                              ecMontEdwdsPointMultiplyPerformCallback,
                              /* Callback function */
                              (void *)&complete, /* Callback tag */
                              opData, /* operational data */
                              &multiplyStatus,
                              /* multiply status in synchronous mode */
                              pXk, /* Xk output buffer */
                              pYk); /* Yk output buffer */
```

HKDF API operational data update:

1. The *secret* and *seed* field swap for HKDF extract like operations
   Since 3.5.0 release the *secret* field will held input key material and *seed* field will held salt data for the EXTRACT like operations.

2. The *secret* field size extended to *CPA_CY_HKDF_KEY_MAX_SECRET_SZ* (64 bytes) for X448 elliptic curve input key material support.

```
typedef struct _CpaCyKeyGenHKDFOpData
{
   CpaCyKeyHKDFOp hkdfKeyOp;
   /**< Keying operation to be performed. */
   Cpa8U secretLen;
   /**< Length of secret field */
   Cpa16U seedLen;
   /**< Length of seed field */
   Cpa16U infoLen;
   /**< Length of info field */
   Cpa16U numLabels;
   /**< Number of filled CpaCyKeyGenHKDFExpandLabel elements */
   Cpa8U secret[CPA_CY_HKDF_KEY_MAX_SECRET_SZ];
   /**< Input Key Material or PRK */
   Cpa8U seed[CPA_CY_HKDF_KEY_MAX_HMAC_SZ];
   /**< Input salt */
   Cpa8U info[CPA_CY_HKDF_KEY_MAX_INFO_SZ];
   /**< info field */
   CpaCyKeyGenHKDFExpandLabel label[CPA_CY_HKDF_KEY_MAX_LABEL_COUNT];
   /**< array of Expand Label structures */
} CpaCyKeyGenHKDFOpData;
```

Known Issues

QATE-52976 – AlgChain and HKDF threads cannot use the same cy instance

Title	**AlgChain and HKDF threads cannot use the same cy instance**
Reference #	QATE-52976
Description	Possible bus error when symmetric and HKDF operation share the same instance due to the request being overwritten.
Implication	It is impossible to share the same instance for symmetric and HKDF operations.
Resolution	It is suggested to use separate instances for symmetric and HKDF operations.
Affected OS	FBSD11.2
Driver/Module	CPM IA - Common

QATE-31888 – Possible performance degradation

Title	**Possible performance degradation**
Reference #	QATE-31888
Description	The integrated configuration for FreeBSD kernel is not optimized for all relevant QAT driver scenarios (issue with threading and scheduling).

Implication	Degradation of QAT data throughput can be observed in the deployment with FreeBSD. The use cases: - sharing the same core for the threads using request ring (submission / working thread) and response ring (polling thread) - sharing the same core for among more working threads - extensive number of threads waiting on mutex queue for responses
Resolution	- Try to balance the thread workload onto several cores - Design the application so the synchronization locks are not shared among many threads
Affected OS	FBSD11.2
Driver/Module	CPM IA - Common

QATE-30931 - Process exit with orphan rings when spawning multiple processes

Title	**Process exit with orphan rings when spawning multiple processes**
Reference #	QATE-30931
Description	If multiple processes start a user space service access layer (icp_sal_userStart) and they all exit together, the syslog may show a message "Process <PID> <NAME> exit with orphan rings.
Implication	A kernel panic might happen at reboot if an application is using QAT.
Resolution	The suggested workaround is to fork the process only after the previous child process runs icp_sal_userStartMultiProcess successfully.
Affected OS	FBSD11.2
Driver/Module	CPM IA - Common

QATE-30360 - Full device pass-through not available on KVM guests

Title	**Full device pass-through not available on KVM guests**
Reference #	QATE-30360
Description	The new firmware authentication feature requires PF devices to be reset via function level reset (FLR) before firmware download. In KVM guests, all pass-through devices attached to a VM are reset at boot time. Any further device reset is trapped by the hypervisor and not issued. This causes firmware authentication to fail after the first firmware download. Full device pass-through might work in some conditions when using vfio and if the host kernel and the platform support it.
Implication	Direct mode feature not available on KVM guests for devices on full pass-through mode.
Resolution	Refer to appendix A of *Using Intel® Virtualization Technology (Intel® VT) with Intel® QuickAssist Technology* (document number 330689-007) for instructions on how to pass through a QAT PF to a VM. Talk to your Intel® representative for more information.
Affected OS	FBSD 11.2
Driver/Module	CPM IA - Common

QATE-5092 - AES-XTS does not support buffers sizes that are not a multiple of 16B

Title	**AES-XTS does not support buffers sizes that are not a multiple of 16B**
Reference #	QATE-5092
Description	A single request with a data size that is not a multiple of 16B for AESXTS will fail with an invalid param check.
Implication	The user cannot submit AES-XTS Crypto requests with buffers that are not multiples of 16B
Resolution	The suggestion is to add padding to AES-XTS to align with 16B multiplied value.
Affected OS	FBSD11.2
Driver/Module	CPM IA – Crypto

QATE-7325 - AES-GCM operation with zero length plain text results in an incorrect tag result

Title	**AES-GCM operation with zero length plain text results in an incorrect tag result**
Reference #	QATE-7325

Description	Sending an AES-GCM operation with zero length plain text may generate an incorrect tag result
Implication	Potentially bad record errors and failing connections
Resolution	GMAC should be used instead of AES-GCM in case of zero length plain text operations.
Affected OS	FreeBSD 11.2
Driver/Module	CPM IA - Crypto

QATE-33751 - Library and driver do not support devices enumerated in a PCI domain different than 0

Title	**Library and driver do not support devices enumerated in a PCI domain different than 0**
Reference #	QATE-33751
Description	The user space driver and the QAT library cannot handle devices enumerated in a domain different than 0.
Implication	It is not possible to use the software in systems where the device is enumerated with a PCI domain different than 0.
Resolution	Use system where device is enumerated with PCI domain 0.
Affected OS	FreeBSD 11.2
Driver/Module	CPM IA - Common

QATE-39335 - Compression instances do not work on Virtual Machine with Linux Host QAT driver without CnVnR support

Title	**Compression instances do not work on Virtual Machine with Linux Host QAT driver without CnVnR support**
Reference #	QATE-39335
Description	FreeBSD QAT VF driver does not get host capabilities - the CnVnR support is enabled by default.
Implication	The driver may fail to start compression instances on Virtual Machine with VF driver if no CnVnR support on Host QAT driver firmware.
Resolution	Use Linux QAT driver CnVnR support (4.3.0 or above) on Linux Host system.
Affected OS	FreeBSD 11.2
Driver/Module	CPM IA - Compression

QATE-40359 - Multiprocess 32 with LimitDevAccess = 0 fails with OpenSSL Speed tests

Title	**Multiprocess failure with NumProcesses > 16 for LBG/DNV and NumProcesses > 32 for CLC and LimitDevAccess = 0**
Reference #	QATE-40359
Description	Multiprocess application that uses more than 16 processes for LBG/DNV and 32 processes for CLC fails during bundle allocation.
Implication	It is impossible to successfully run multiprocess application with more processes than 16 for LBG/DNV and 32 for CLC.
Resolution	There is limitation to use up to 16 processes for LBG/DNV and up to 32 for CLC per device.
Affected OS	FreeBSD 11.2
Driver/Module	CPM IA - Multiprocess

QATE-39216 - Kasumi test duration issue

Title	**Kasumi test duration issue**
Reference #	QATE-39216
Description	Sample code benchmark tests included in the software package
Implication	The performance degradation when running the sample code can be observed in case the system runs excessive number of threads.
Resolution	Avoid calling the cpaCyInstanceGetInfo2 function if possible (i.e. by caching the info data) and try to tune the FreeBSD scheduler.
Affected OS	FreeBSD 11.2

Driver/Module	CPM IA - Crypto

QATE-41846 - GEN - QAT API submissions with bad addresses that trigger DMA to invalid or unmapped addresses can cause a platform hang

Title	**GEN - QAT API submissions with bad addresses that trigger DMA to invalid or unmapped addresses can cause a platform hang**
Reference #	QATE-41846
Description	This version of the QAT hardware does not perform request checking. It follows that a malicious application can submit requests that can bring down an entire QAT endpoint, which can impact other QAT jobs associated with the hardware. Furthermore, if any QAT API submission have bad addresses that would trigger DMA to invalid or unmapped addresses, these can induce a platform hang. This presents a risk to be managed by the host and guest operating systems and other system policies. The exposure can extend to other guest operating systems or applications outside of the typical access boundary of the malicious guest or application.
Implication	All guest operating systems or other applications using QAT must be trusted, and/or other steps must be taken to ensure that an untrusted application or guest cannot submit incorrectly formatted requests.
Resolution	There is no workaround available. However, system policies (including limiting certain operating system permissions) can help to mitigate this issue.
Affected OS	FreeBSD 11.2
Driver/Module	CPM IA - Crypto

QATE-41486 - Misleading message observed in dmesg on LBG device with LimitDevAccess = 1 set in configuration file.

Title	**Misleading message observed in dmesg on LBG device with LimitDevAccess = 1 set in configuration file**
Reference #	QATE-41486
Description	When using LimitDevAccess = 1 with more than one device in up state, the "qatX: failed to get NumberCyInstaces value from config!" message could be observed in dmesg for other devices than configured one. This message indicates only that for the other devices the configuration was not found, what is expected.
Implication	This is internal message only, and should not be threat as an error.
Resolution	Ignore error message when use LimitDevAccess parameter.
Affected OS	FreeBSD 11.2
Driver/Module	CPM