Cryptography and Encryption

Table of contents

This page contains commands for configuring, generating and modifying x.509 certificates used in the system. Certificates are used for creating a trusted SSL connection to the system.

Crypto commands also cover IPSec configuration commands used for establishing a secure connection between hosts over IP layer which is useful for transferring sensitive information.

System File Encryption

This feature encrypts all sensitive data on NVIDIA systems including logs certificates, keys, etc.

To activate encryption on the switch:

1. Enable encryption and configure key location as USB (if you are using a USB device).

```
switch (config)# crypto encrypt-data key-location usb key
mypassword

Warning! All sensitive files are about to be encrypted
- System will perform reset factory, configuration files will
be preserved
- System will be rebooted
- Active configuration will be preserved
- Do not power-off, wait for the system to boot

Type 'YES' to confirm this action: YES
```

> **Note**
>
> ***IMPORTANT***
>
> Encryption and decryption perform “reset factory keep-config” on the switch system once configured. This means that sysdumps, logs, and images are deleted.

Note

The key may be saved locally as well by using the parameter “local” instead of “usb” but that configuration is less secure.

2. After the system reboots, verify configuration.

```
switch (config)# show crypto encrypt-data
Sensitive files encryption:
   Status:          enabled
   Key location:    usb
   Cipher:          aes256
```

Note

Once encryption is enabled, reverting back to an older version while encrypted is not possible. The command “no crypto encrypt-data” must be run before attempting to downgrade to an older OS version.

Note

If encryption is enabled, upgrading to a new OS version maintains the encryption configuration.

Changing a Certificate in the System

To change the default certificate for the system, to the following:

1. Import the certificate to be used (e.g., a certificate created by openssl outside the switch).

```
switch (config) # crypto certificate name <cert_name> public-cert
pem "-----BEGIN CERTIFICATE-----
>
MIIDYzCCAksCCQC9EPbMuxjNBzANBgkqhkiG9w0BAQsFADBeMQswCQYDVQQGEw
...
>
fEt2ui9taB1dl9480xDsGUxwUDX4YOs/bQDjp99z+cKXUe2eYzeEwnTdrCzPZu
> -----END CERTIFICATE-----"
Successfully installed certificate with name '<cert_name>'
```

Or use a new self-signed certificate via switch CLI and export it as a CSR (certificate signing request) and send said CSR to the root CA for signing:

```
switch (config) # crypto certificate name <cert_name> generate
self-signed
Successfully generated certificate with name '<cert_name>'

switch (config) # show crypto certificate name <cert_name> csr-
pem

-----BEGIN CERTIFICATE REQUEST-----
MIICuDCCAaACAQAwczELMAkGA1UEBhMCSVMxDDAKBgNVBAgMA1RCRDEMMAoGA1
BwwDVEJEMQwwCgYDVQQKDANUQkQxDDAKBgNVBAsMA1RCRDEYMBYGA1UEAwwPYn
bGRvZy1xcDEtMTMzMRIwEAYJKoZIhvcNAQkBFgNUQkQwggEiMA0GCSqGSIb3DQ
AQUAA4IBDwAwggEKAoIBAQC34xRVh9BaBUPIilV6kiSOAVAnOFgreWtEYoWeGp
XGZQBwewFx4TGptYo5fZ4KcnYcQxrcW7gYycQB9Y+9vUVvvPi3b4aYc2FkoNtn
0BRTxEcIiwXY7LQxIA23Zuv/OlhjTkpe0+OYtpJSFeIDKMIX4Uy2BfevG06YLC
tuju2FLQVkexayNK/HFLa5POpVt+16JLB1eV0bcC38Mq9JNIgPspJ7JIjo+Bjz
43iEY41hlRzoalu78nBBd0HbAddxCF1Uc+8PLuPLCIjGbV9ehPJNWSsA/T9jUE
```

```
90KaI0/k05JqCXWnpvKz3opQraHsVAbsxG312pnmbTFNAgMBAAGgADANBgkqhk
9w0BAQsFAAOCAQEAhpgZRNW/jleyhUbtGEr0CzdNbJ70V8w2lGr6bDhZgrQ/I4
1K1D1hvfrVWYRB0SSPFmCmVmFmC7BQne8xrbL2It3ZdSKd82Ts36/Uxjtb63hy
GBzCas7qypsbCVW42UHuD+259Yu5xpi9haspzD8Wg2ZKU5e6SjcH+JIchkM9mh
BQJo4shybTgPfT+mFUCCygWmf5aLyQ9TrZpaUQ7cOk6BZB1RRkOVvA6uCfrwlB
X72LleceL4fP9dtML4VMzMMAf+wOUNxWP9+lqkKMaDhroDP5qlo/lr5BLSlRVe
z7zb3xSaPrhnefoGr88WFO74d9RxLPPdHcfMFw==
-----END CERTIFICATE REQUEST-----
```

2. Import key of certificate.

```
switch (config) # crypto certificate name <cert_name> private-key
pem "-----BEGIN RSA PRIVATE KEY-----
>
MIIEpAIBAAKCAQEAypJnZkwbhmt71Kf/MO6cy7QmWWHhCozzWRwuWGKse+MxSm
...
> QAuPOVR1lSyIEnYU+X0rMHc/9tgUh/8C7mBKwj7dccMmnRWz2djsjg==
> -----END RSA PRIVATE KEY-----"
```

3. Designate <cert_name> as the global default certificate for authentication of this system to clients.

```
switch (config) # crypto certificate default-cert name
<cert_name>
```

4. (Optional) Import the Certificate Authority (CA) certificate which signed for the controller.

```
switch (config) # # crypto certificate name rootCA public-cert
pem "-----BEGIN CERTIFICATE-----
```

```
>
MIIDjzCCAnegAwIBAgIJALVou4mcQtxlMA0GCSqGSIb3DQEBCwUAMF4xCzAJBgl
...
>
+ZfQIOCFS8gY4BDq73W4ugr38mqIA8UXXAMPwgjCbk4NyOh0rJ1P6WT8fYzvun
> -----END CERTIFICATE-----"
Successfully installed certificate with name 'rootCA'
```

5. Adds the “rootCA” to the default CA certificate list.

```
switch (config) # crypto certificate ca-list default-ca-list name
rootCA
```

6. Save configuration.

```
switch (config) # configuration write
```

7. Verify configuration.

```
switch (config) # show crypto certificate
Certificate with name 'system-self-signed'
    Comment:                        system-generated self-
signed certificate
    Private Key:                    present
    Serial Number:                  0x543e2efc3a5ecdbe18b5b5e744598424
    SHA-1 Fingerprint:
14e1d36035c7a5fea9f7f0f423572c9954cb9fac

    Validity:
        Starts:                     2022/09/12 12:44:10
        Expires:                    2023/09/12 12:44:10
```

```
 Subject:
        Common Name:                switch
        Country:                    IS
        State or Province:          TBD
        Locality:                   TBD
        Organization:               TBD
        Organizational Unit:        TBD
        E-mail Address:             TBD

    Issuer:
        Common Name:                switch
        Country:                    IS
        State or Province:          TBD
        Locality:                   TBD
        Organization:               TBD
        Organizational Unit:        TBD
        E-mail Address:             TBD

Certificate with name '<cert_name>' (default-cert)
    Private Key:                    present
    Serial Number:                  0xbd10f6ccbb18cd07
    SHA-1 Fingerprint:
1e0e3302182ab56f2cbd3ca21722dec55299d670

    Validity:
        Starts:                     2021/09/12 15:16:48
        Expires:                    2023/01/25 14:16:48

    Subject:
        Common Name:                switch
        Country:                    *
        State or Province:          Some-State
        Locality:                   *
        Organization:               NVIDIA
        Organizational Unit:        e2e
        E-mail Address:             none@nowhere.com
```

```
    Issuer:
        Common Name:            ca
        Country:                *
        State or Province:      Some-State
        Locality:               *
        Organization:           NVIDIA
        Organizational Unit:    e2e
    Certificate with name 'rootCA'
    Private Key:                not present
    Serial Number:              0xb568bb899c42dc65
    SHA-1 Fingerprint:
9855536f6ee0177356ffbdc54ffe803bc83fb4c6
    Validity:
        Starts:                 2020/09/08 10:34:23
        Expires:                2023/06/29 10:34:23

    Subject:
        Common Name:            ca
        Country:                *
        State or Province:      Some-State
        Locality:               *
        Organization:           NVIDIA
        Organizational Unit:    e2e

    Issuer:
        Common Name:            ca
        Country:                *
        State or Province:      Some-State
        Locality:               *
        Organization:           NVIDIA
        Organizational Unit:    e2e
```

Cryptographic and Encryption Commands

crypto encrypt-data

	crypto encrypt-data key-location <local | usb> key <password> no crypto encrypt-data Enables and configures system file encryption. The no form of the command decrypts sensitive information on the system.	
Syntax Description	key-location	Configures where to store the encryption key: • local—stores the key locally • usb—stores the key on a USB device
	key	Configures a key
Default	N/A	
Configuration Mode	config	
History	3.6.1002	
Example		
Related Commands	show crypto certificate	
Notes	• It is recommended to store the encryption password on a USB device rather than locally • Enabling encryption may slightly slow system performance • If the key is stored on the USB, it must be plugged into the switch in order for the switch to boot. After the switch has booted, the USB key is no longer required and, for security purposes, it is recommended to remove it after running “usb eject”. The USB key may be needed again if the switch is rebooted or if the switch needs to be decrypted.	

crypto ipsec ike

	crypto ipsec ike {clear sa [peer {any | <IPv4 or IPv6 address>} local <IPv4 or IPv6 address>] | restart} Manages the IKE (ISAKMP) process or database state.

Syntax Description	clear	Clears IKE (ISAKMP) peering state
	sa	Clears IKE generated ISAKMP and IPSec security associations (remote peers are affected)
	peer	Clears security associations for the specified IKE peer (remote peers are affected) • all—clears security associations for all IKE peerings with a specific local address (remote peers are affected) • IPv4 or IPv6 address—clears security associations for specific IKE peering with a specific local address (remote peers are affected)
	IPv4 or IPv6 address	Clears security associations for the specified IKE peering (remote peer is affected)
	local	Clear security associations for the specified/all IKE peering (remote peer is affected)
	restart	Restarts the IKE (ISAKMP) daemon (clears all IKE state, peers may be affected)
Default	N/A	
Configuration Mode	config	
History	3.2.3000	
Example	switch (config)# crypto ipsec ike restart	
Related Commands	show crypto certificate	
Notes		

crypto ipsec peer local

	crypto ipsec peer local {enable | keying {ike negotiation {ikev1 | ikev2} | [auth { hmac-sha1 | hmac-sha256 | hmac-sha512 | aes-xcbc} | dh-group | disable | encrypt { 3des-cbc| aes-cbc | aes-gcm} | exchange-mode | lifetime | local | mode | peer-identity | pfs-group | preshared-key | prompt-preshared-key | transform-set] | manual [auth | disable | encrypt | local-spi | mode | remote-spi]}}

	Configures IPSec in the system.	
Syntax Description	enable	Enables IPSec peering.
	ike	Configures IPSec peering using IKE ISAKMP to manage SA keys. The following optional parameters are available: • auth—configures the authentication algorithm for IPSec peering • dh-group—configures the phase1 Diffie-Hellman group proposed for secure IKE key exchange • disable—configures this IPSec peering administratively disabled • encrypt—configures the encryption algorithm for IPSec peering • exchange-mode—configures the IKE key exchange mode to propose for peering • lifetime—configures the SA lifetime to propose for this IPSec peering • local-identity—configures the ISAKMP payload identification value to send as local endpoint's identity • mode—configures the peering mode for this IPSec peering • peer-identity—configures the identification value to match against the peer's ISAKMP payload identification • pfs-group—configures the phase2 PFS (Perfect Forwarding Secrecy) group to propose for Diffie-Hellman exchange for this IPSec peering • preshared-key—configures the IKE pre-shared key for the IPSec peering • prompt-preshared-key—prompts for the pre-shared key, rather than entering it on the command line • transform-set—configures transform proposal parameters
	keying	Configures key management for this IPSec peering. • auth—configures the authentication algorithm for this IPSec peering

		• disable—configures this IPSec peering administratively disabled • encrypt—configures the encryption algorithm for this IPSec peering • local-spi—configures the local SPI for this manual IPSec peering • mode—configures the peering mode for this IPSec peering • remote-spi—configures the remote SPI for this manual IPSec peering
	manual	Configures IPSec peering using manual keys.
Default	N/A	
Configuration Mode	config	
History	3.2.3000 3.9.3100: Added support for IKEv2 and new ciphers	
Example	switch (config)# crypto ipsec peer 10.10.10.10 local 10.7.34.139 enable	
Related Commands	show crypto certificate	
Notes	As of version 3.9.3100, NULL will not be supported as an authentication or encryption algorithm for IPsec peering. New ciphers are supported (hmac-sha512 and aes-xcbc for authentication and aes-gcm for encryption. 1, 2, 5, 22, 23, 24 pfs/dh-groups will not be supported, while 19, 20, 21 will be supported only with IKEv2. The transform-set options ah-and-esp-ah are no longer supported. Libreswan is used instead of openswan.	

crypto certificate ca-list

	crypto certificate ca-list [default-ca-list name {<cert-name> | system-self-signed}] no crypto certificate ca-list [default-ca-list name {<cert-name> | system-self-signed}] Adds the specified CA certificate to the default CA certificate list. The no form of the command removes the certificate from the default CA certificate list.

Syntax Description	cert-name	The name of the certificate
Default	N/A	
Configuration Mode	config	
History	3.2.3000	
Example	switch (config) # crypto certificate default-cert name test	
Related Commands	show crypto certificate	
Notes	• Two certificates with the same subject and issuer fields cannot both be placed onto the CA list • The no form of the command does not delete the certificate from the certificate database • Unless specified otherwise, applications that use CA certificates will still consult the well-known certificate bundle before looking at the default-ca-list	

crypto certificate default-cert

	crypto certificate default-cert name {<cert-name> | system-self-signed} no crypto certificate default-cert name {<cert-name> | system-self-signed} Designates the named certificate as the global default certificate role for authentication of this system to clients. The no form of the command reverts the default-cert name to “system-self-signed” (the “cert-name” value is optional and ignored).	
Syntax Description	cert-name	The name of the certificate
Default	N/A	
Configuration Mode	config	
History	3.2.3000	
Example	switch (config) # crypto certificate default-cert name test	

Related Commands	show crypto certificate
Notes	• A certificate must already be defined before it can be configured in the default-cert role • If the named default-cert is deleted from the database, the default-cert automatically becomes reconfigured to the factory default, the “system-self-signed” certificate

crypto certificate generation

	crypto certificate generation default {country-code | days-valid > | ca-valid <true/false> | email-addr | hash-algorithm {sha1 | sha256} | key-size-bits | locality | org-unit | organization | state-or-prov} Configures default values for certificate generation.	
Syntax Description	country-code	Configures the default certificate value for country code with a two-alphanumeric-character code or -- for none.
	days-valid	Configures the default certificate valid days Default value: 365 days
	email-addr	Configures the default certificate value for email address
	hash-algorithm {sha1 | sha256}	Configures the default certificate hashing algorithm
	key-size-bits	Configures the default certificate value for private key size (private key length in bits—at least 1024, but 2048 is strongly recommended)
	locality	Configures the default certificate value for locality
	org-unit	Configures the default certificate value for organizational unit
	organization	Configures the default certificate value for the organization name
	state-or-prov	Configures the default certificate value for state or province

	ca-valid {true | false}	Configures the default certificate CA Basic Constraints flag set to TRUE/FALSE
Default	hash-algorithm – sha1	
Configuration Mode	config	
History	3.2.1000 3.3.4350: Added “hash-algorithm” parameter 3.6.4000: Added “days-valid” parameter 3.8.2100: Added "ca-valid" parameter	
Example	switch (config) # crypto certificate generation default hash-algorithm sha256	
Related Commands	show crypto certificate	
Notes		

crypto certificate name

	crypto certificate name {<cert-name> | system-self-signed} {comment <new comment> | generate selfsigned [comment <cert-comment> | common-name <domain> | country-code <code> | days-valid <days> | ca-valid <true/false> | email-addr <address> | hash-algorithm {sha1 | sha256} | key-size-bits <bits> | locality <name> | org-unit <name> | organization <name> | serial-num <number> | state-or-prov <name>]} | private-key pem <PEM string> | prompt-private-key | public-cert [comment <comment string> | pem <PEM string>] | regenerate days-valid <days> | ca-valid <true/false> | rename <new name>} no crypto certificate name <cert-name> Configures default values for certificate generation. The no form of the command clears/deletes certain certificate settings.	
Syntax Description	cert-name	Unique name by which the certificate is identified.
	comment	Specifies a certificate comment.
	generate self-signed	Generates certificates. This option has the following parameters which may be entered sequentially in any order: • comment—specifies a certificate comment (free string)

		• common-name—specifies the common name of the issuer and subject (e.g. a domain name) • country-code—specifies the country codwo-alphanumeric-character country code, or “--” for none) • days-valid—specifies the number of days the certificate is valid • email-addr— s pecifies the email address • hash-algorithm—specifies the hashing function used for signature algorithm. Default value is SHA256. • key-size-bits—specifies the size of the private key in bits (private key length in bits - at least 1024 but 2048 is strongly recommended) • locality—specifies the locality name • org-unit—specifies the organizational unit name • organization—specifies the organization name • serial-num—specifies the serial number for the certificate (a lower-case hexadecimal serial number prefixed with “0x”) • state-or-prov—specifies the state or province name • ca-valid—Specifies certificate CA Basic Constraints flag set to TRUE/FALSE
	private-key pem	Specifies certificate contents in PEM format
	prompt-private-key	Prompts for certificate private key with secure echo
	public-cert	Installs a certificate
	regenerate	Regenerates the named certificate using configured certificate generation default values for the specified validity period
	rename	Renames the certificate
Default	N/A	
Configuration Mode	config	
History	3.2.3000 3.3.4402: Added “hash-algorithm” parameter 3.6.4000: Added “days-valid” parameter 3.8.2100: Added "ca-valid" parameter	

Example	switch (config) # crypto certificate name system-self-signed generate self-signed hash-algorithm sha256
Related Commands	show crypto certificate
Notes	

crypto certificate system-self-signed

	crypto certificate system-self-signed regenerate {[days-valid <days>] | ca-valid <true/false>} Configures default values for certificate generation.	
Syntax Description	days-valid	Specifies the number of days the certificate is valid
	ca-valid	Specifies certificate CA Basic Constraints flag set to TRUE/FALSE
Default	N/A	
Configuration Mode	config	
History	3.2.1000 3.8.2100: Added the ca-valid option	
Example	switch (config) # crypto certificate system-self-signed regenerate days-valid 3 switch (config) # crypto certificate system-self-signed regenerate ca-valid false	
Related Commands	show crypto certificate	
Notes		

show crypto certificate

	show crypto certificate [detail | public-pem | default-cert [detail | public-pem] | [name <cert-name> [detail | public-pem] | ca-list [default-ca-list]] Displays information about all certificates in the certificate database.

Syntax Description	ca-list	Displays the list of supplemental certificates configured for the global default system CA certificate role
	default-ca-list	Displays information about the currently configured default certificates of the CA list
	default-cert	Displays information about the currently configured default certificate
	detail	Displays all attributes related to the certificate
	name	Displays information about the certificate specified
	public-pem	Displays the uninterpreted public certificate as a PEM formatted data string
Default	N/A	
Configuration Mode	config	
History	3.2.1000 3.8.2100: Updated output	

Example

```
switch (config) # show crypto certificate

Certificate with name 'system-self-signed' (default-cert)
   Comment:                system-generated self-signed certificate
   Private Key:            present
   Serial Number:          0x546c935511bcafc21ac0e8249fbe0844
   SHA-1 Fingerprint:      fe6df38dd26801971cb2d44f62dbe492b6063c5f
   Validity:
      Starts:              2012/12/02 13:45:05
      Expires:             2013/12/02 13:45:05
   Subject:
      Common Name:         IBM-DEV-Bay4
      Country:             IS
      State or Province:
      Locality:
      Organization:
      Organizational Unit:
      E-mail Address:
   Issuer:
      Common Name:         IBM-DEV-Bay4
      Country:             IS
      State or Province:
      Locality:
      Organization:
      Organizational Unit:
```

E-mail Address: X509 Extensions: Basic Constraints: CA: TRUE	
Related Commands	
Notes	

show crypto encrypt-data

	show encrypt-data Displays sensitive data encryption information.
Syntax Description	N/A
Default	N/A
Configuration Mode	config
History	3.6.1002
Example	switch (config)# show crypto encrypt-data Sensitive files encryption: Status: enabled Key location: usb Cipher: aes256
Related Commands	
Notes	

show crypto ipsec

	show crypto ipsec [brief | configured | ike | policy | sa] Displays information ipsec configuration.
Syntax Description	N/A

Default	N/A
Configuration Mode	config
History	3.2.1000

Example	switch (config)# show crypto ipsec
	IPSec Summary
	---------- 	
	Crypto IKE is using pluto (Openswan) daemon.
	Daemon process state is stopped.
	  No IPSec peers configured.
	IPSec IKE Peering State
	---------- 	
	Crypto IKE is using pluto (Openswan) daemon.
	Daemon process state is stopped.
	  No active IPSec IKE peers.
	IPSec Policy State
	---------- 	
	  No active IPSec policies.
	IPSec Security Association State
	---------- 	
	  No active IPSec security associations.

Related Commands	
Notes	

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation (“NVIDIA”) makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer (“Terms of Sale”). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer’s own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, “MATERIALS”) ARE BEING PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF

ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

NVIDIA and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.