

Multiple Dcardmgrs support per SF card in VPC-DI

- Multi dcardmgr instances, on page 1
- Configure multiple dcardmgr instances , on page 1

Multi dcardmgr instances

A multi dcardmgr instance is a system configuration that

- allows multiple Dcard manager instances to operate on a single Service Function card,
- increases the number of dcardmgr instances per card, and
- improves overall call setup time for high-subscriber environments

Supports up to eight dcardmgr instances per Service Function (SF) card on VPC-DI platforms. When multiple dcardmgr instances are configured for each SF, ipsecmgrs interact with the mapped dcardmgr to perform flow operations.

Configuration requirements include:

- **Default**: One dcardmgr is created per card.
- **Scaling**: More than one dcardmgr is required if the number of subscribers per card exceeds 4 million.
- **Limit**: A maximum of eight instances per SF card can be configured.
- **Platform**: This configuration applies only to VPC-DI platforms.

Configure multiple dcardmgr instances

Set the required number of dcardmgr instances per session card to improve call processing capacity on VPC-DI platforms.

By default, the system uses one dcardmgr instance per session card . If sessions are more than 4 million, its recommended to configure more than dcardmgr. You can configure up to eight instances per SF card to scale performance and handle higher call establishment rates as needed. This configuration applies only to VPC-DI platforms.

Note For releases starting from 2025.03.0 and prior to 2026.03.0, if you configure two Dcardmgr instances, refer to the Configuring the Number of Dcardmgrs Per Session Card section in the ePDG Administration Guide for functional behavior details.

Follow these steps to configure the number of dcardmgrs per session card.

Before you begin

Ensure you are operating on a VPC-DI platform.

Procedure

Step 1 Enter Global Configuration mode.

Example:

```
[pdif]ePDG# configure
```

Step 2 Configure the number of dcardmgrs.

Note
Changing the number of dcardmgr instances requires a system reload.

To set the desired number of dcardmgr instances per session card, use the **task facility dcardmgr per-sesscard-count** *integer value* command. Supported integer values: 1 - 8.

Example:

```
[pdif]ePDG# configure

[pdif]ePDG(config)# task facility dcardmgr per-sesscard-count

<integer value (1 .. 8)>

[pdif]ePDG(config)# task facility dcardmgr per-sesscard-count 7

Warning: Save config and reload
```

Step 3 Save the configuration and reload the system to apply the changes.

To reset the number of dcardmgr instances per session card to its default value, use the **default task facility dcardmgr per-sesscard-count** command or the **task facility dcardmgr per-sesscard-count 1** command..

Example:

```
[local]EPDGCHASSIS(config)#default task facility dcardmgr per-sesscard-count

<cr> - newline

[local]EPDGCHASSIS(config)# default task facility dcardmgr per-sesscard-count

[local]EPDGCHASSIS(config)
```

Step 4 Verify the configured number of dcardmgrs using the **show configuration** command and **show configuration verbose** command to view default dcardmgrs configuration.

Example:

```
[local]ePDG(config)# task facility dcardmgr per-sesscard-count 7
Warning: Save config and reload
[local]EPDGCHASSIS#
[local]EPDGCHASSIS# show configuration | grep "task facility dcardmgr"
task facility dcardmgr per-sesscard-count 7
[local]EPDGCHASSIS#

[local]EPDGCHASSIS#
[local]EPDGCHASSIS# configure
[local]EPDGCHASSIS(config)# default task facility dcardmgr per-sesscard-count
[local]EPDGCHASSIS(config)# end
[local]EPDGCHASSIS#
[local]EPDGCHASSIS# show configuration verbose | grep "task facility dcardmgr"
task facility dcardmgr per-sesscard-count 1
[local]EPDGCHASSIS#
```

Step 5 Reload the system to activate the new configuration.

Step 6 View session details for each configured dcardmgr instances by using the **show daughtercard sessions verbose** command.

Example:

```
Card 4 CPU 0 Instance 41:
  96 sessions created for slot 4 cpu 0 instance 41:
 --- SessionId(hex)   ----  SPI  ----  SessType -- Instance - In HW - seqno
       0000bff8         0x40001d01       IPsec       17          Y      0
       0000bff9         0x04004304       IPsec       17          Y      1


Card 4 CPU 0 Instance 42:
  132 sessions created for slot 4 cpu 0 instance 42:
 --- SessionId(hex)   ----  SPI  ----  SessType -- Instance - In HW - seqno
       0000ffe9         0x40019101       IPsec       18          Y      0
       0000ffea         0x04008304       IPsec       18          Y      1
```

Step 7 Verify the operational counters for crypto sub-systems per card and instance using the **show daughtercard counters** command.

Example:

```
[local]EPDGCHASSIS# show daughtercard counters
========================== ============ + ========================== ============
Card: 3, CPU: 0, Crypto Sub-system: UCAD-iftask LCORES:16
Counter                          Value |  Counter                          Value
-------------------------- ------------ + -------------------------- ------------
2:encr_sa                            0 | 2:encr_sa_del                         0
:
2:sw_q_full                          0 | 2:sw_q_failure                       0
2:op_req                            71 | 2:p1_req                            0
:
3:sw_q_full                          0 | 3:sw_q_failure                       0
3:op_req                           121 | 3:p1_req                            0
:
6:ecn_drop_pkt                       0 | 6:ecn_cu_pkt                         0
========================== ============ + ========================== ============
Card: 3, CPU: 0, Crypto Sub-system: UCAD-iftask LCORES:16
Counter                          Value |  Counter                          Value
-------------------------- ------------ + -------------------------- ------------
2:sw_q_full                          0 | 2:sw_q_failure                       0
2:op_req                            71 | 2:p1_req                            0
```

```
::
::
6:ecn_drop_pkt                          0 | 6:ecn_cu_pkt                          0
========================= ============ + ========================= ============
Card: 6, CPU: 0, Crypto Sub-system: UCAD-iftask LCORES:16
Counter                           Value |  Counter                          Value
------------------------- ------------ + ------------------------- ------------
2:encr_sa                           256 | 2:encr_sa_del                         0
:
2:encr_ah_failure                     0 | 2:encr_ipv4_dscp_marked               0
2:encr_ipv6_dscp_marked               0 | 2:decr_sa                           256
2:decr_sa_del                         0 | 2:decr_rx                             0
:
2:op_req                            773 | 2:p1_req                              0
2:p2_req                              0 | 2:sa_req                            512
:
3:encr_ah_failure                     0 | 3:encr_ipv4_dscp_marked               0
3:encr_ipv6_dscp_marked               0 | 3:decr_sa                           583
3:decr_sa_del                         0 | 3:decr_rx                             0
3:op_req                           2105 | 3:p1_req                              0
3:p2_req                              0 | 3:sa_req                           1166
3:sa_del                              0 | 3:sa_get                              0
[local]EPDGCHASSIS#
```

Step 8 View utilization and performance data for crypto threads and processes by using the **show crypto process**

Example:

```
[local]EPDGCHASSIS# show crypto process
Crypto Process Data for Card: 3 CPU: 0

 Process Data
       AES-NI: Enabled
    Dedicated: 0 Threads
       Shared: 2 Threads

 Performance Data
Iftask Utilization Per Thread
    5-Sec Avg: lcore01| lcore02| lcore03| lcore04| lcore05|
         Idle:    100%|     98%|    100%|    100%|    100%|
       PortTX:      0%|      1%|      0%|      0%|      0%|
       Cipher:      0%|      1%|      0%|      0%|      0%|

Packets Per Second Per Thread
       Thread:   lcore02|   lcore03|   lcore04|   lcore05|   lcore06|
    1-Min Avg:         0|         0|         0|         0|         0|
    5-Min Avg:         0|         0|         0|         0|         0|
   15-Min Avg:         0|         0|         0|         0|         0|

Crypto Process Data for Card: 3 CPU: 0

 Process Data
       AES-NI: Enabled
    Dedicated: 0 Threads
       Shared: 2 Threads

 Performance Data
Iftask Utilization Per Thread
    5-Sec Avg: lcore01| lcore02| lcore03| lcore04| lcore05|
         Idle:    100%|     98%|    100%|    100%|    100%|
       PortTX:      0%|      1%|      0%|      0%|      0%|
       Cipher:      0%|      1%|      0%|      0%|      0%|
```

```
Packets Per Second Per Thread
       Thread:   lcore02|   lcore03|   lcore04|   lcore05|   lcore06|
    1-Min Avg:         0|         0|         0|         0|         0|
    5-Min Avg:         0|         0|         0|         0|         0|
   15-Min Avg:         0|         0|         0|         0|         0|

Crypto Process Data for Card: 3 CPU: 0
::
```

The system is updated to run the desired number of dcardmgr instances per session card, improving call processing capacity as required. Verification commands confirm successful configuration and operational status.