



## Debug Commands

---

The commands in this section are for troubleshooting the GGSN. For information about other debug commands, see the *Cisco IOS Debug Command Reference*.

This chapter contains the following command:

- [debug gprs dfp, page 304](#)
- [debug gprs dhcp, page 305](#)
- [debug gprs gtp, page 307](#)
- [debug gprs gtp-director, page 309](#)
- [debug gprs gtp parsing, page 315](#)
- [debug gprs gtp ppp, page 316](#)
- [debug gprs gtp ppp-regeneration, page 318](#)
- [debug gprs radius, page 322](#)

# debug gprs dfp

To display debug messages for GPRS DFP weight calculation, use the **debug gprs dfp** privileged EXEC command. To disable debugging output, use the **no** form of this command.

**debug gprs dfp**

**no debug gprs dfp**

---

**Syntax Description** This command has no arguments or keywords.

---

**Defaults** No default behavior or values.

---

Command History	Release	Modification
	12.1(9)E	This command was introduced.
	12.2(4)MX	This command was incorporated in Cisco IOS Release 12.2(4)MX.
	12.2(8)YD	This command was incorporated in Cisco IOS Release 12.2(8)YD.
	12.2(8)YW	This command was incorporated in Cisco IOS Release 12.2(8)YW.

---

**Usage Guidelines** See the following caution before using **debug** commands:



**Caution**

---

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to use **debug** commands during periods of lower network flows and fewer users. Debugging during these periods reduces the effect these commands have on other users on the system.

---

This command displays debug messages for GPRS DFP weight calculation. To display debug messages for the DFP agent subsystem, use the **debug ip dfp agent** command.

---

**Examples** The following example configures a debug session to check all GPRS DFP weight calculation:

```
Router# debug gprs dfp
GPRS DFP debugging is on
Router#
```

The following example stops all debugging:

```
Router# no debug all
All possible debugging has been turned off
Router#
```

# debug gprs dhcp

To display information about Dynamic Host Configuration Protocol (DHCP) processing on the GGSN, use the **debug gprs dhcp** privileged EXEC command. To disable debugging output, use the **no** form of this command.

**debug gprs dhcp**

**no debug gprs dhcp**

## Syntax Description

This command has no arguments or keywords.

## Defaults

No default behavior or values.

## Command History

Release	Modification
12.2(4)MX	This command was introduced.
12.2(8)YD	This command was incorporated in Cisco IOS Release 12.2(8)YD.
12.2(8)YW	This command was incorporated in Cisco IOS Release 12.2(8)YW.

## Usage Guidelines

This command is useful for system operators and development engineers if problems are encountered with DHCP processing on the GGSN. To display standard debug messages between the DHCP client on the router and a DHCP server, you can also use the **debug dhcp** or **debug dhcp detail** commands with the **debug gprs dhcp** command.



### Caution

Because the **debug gprs dhcp** command generates a significant amount of output, use it only when traffic on the GPRS network is low, so other activity on the system is not adversely affected.

## Examples

The following example shows sample output for DHCP processing on the GGSN:

```
Router# debug gprs dhcp
2d13h: GPRS:DHCP req:TID 1111111100000099, Req 1
2d13h: GPRS:Requesting IP address for pdp 1111111100000099 from server 172.16.0.8 tableid
0
2d13h: GPRS:DHCP ip allocation pass (10.88.17.43) for pdp 1111111100000099
2d13h: GPRS:Using DHCP ip address 10.88.17.43 for pdp 1111111100000099
```

The following example shows sample output for standard debug messaging for DHCP processing on the router between the DHCP client and a DHCP server:

```
2d13h: DHCP: proxy allocate request
2d13h: DHCP: new entry. add to queue
2d13h: DHCP: SDiscover attempt # 1 for entry:
2d13h: DHCP: SDiscover: sending 283 byte length DHCP packet
2d13h: DHCP: SDiscover with directed serv 172.16.0.8, 283 bytes
2d13h: DHCP: XID MATCH in dhcpc_for_us()
2d13h: DHCP: Received a BOOTREP pkt
2d13h: DHCP: offer received from 172.16.0.8
```

■ **debug gprs dhcp**

```
2d13h: DHCP: SRequest attempt # 1 for entry:
2d13h: DHCP: SRequest- Server ID option: 172.16.0.8
2d13h: DHCP: SRequest- Requested IP addr option: 10.88.17.43
2d13h: DHCP: SRequest placed lease len option: 604800
2d13h: DHCP: SRequest: 301 bytes
2d13h: DHCP: SRequest: 301 bytes
2d13h: DHCP: XID MATCH in dhcpc_for_us()
2d13h: DHCP: Received a BOOTREP pkt
2d13h: DHCP Proxy Client Pooling: ***Allocated IP address: 10.88.17.43
```

---

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>debug dhcp</b>	Displays debug messages between the DHCP client on the router and a DHCP server.

---

# debug gprs gtp

To display information about the GPRS Tunneling Protocol (GTP), use the **debug gprs gtp** privileged EXEC command. To disable debugging output, use the **no** form of this command.

```
debug gprs gtp { events | messages | packets | ppp { details | events } }
```

```
no debug gprs gtp { events | messages | packets | ppp { details | events } }
```

## Syntax Description

<b>events</b>	Displays events related to GTP processing on the GGSN.
<b>messages</b>	Displays GTP signaling messages that are sent between the SGSN and GGSN.
<b>packets</b>	Displays GTP packets that are sent between the SGSN and GGSN.
<b>ppp { details   events }</b>	Displays GTP PPP packets that are sent between the SGSN and GGSN. The <b>details</b> keyword generates more extensive debug output. The <b>events</b> keyword generates output specific to certain conditions that are occurring, which helps qualify the output being received using the <b>details</b> option.

## Defaults

No default behavior or values.

## Command History

Release	Modification
12.1(1)GA	This command was introduced.
12.1(5)T	This command was integrated in Cisco IOS Release 12.1(5)T.
12.2(4)MX	This command was incorporated in Cisco IOS Release 12.2(4)MX, and the <b>ppp { details   events }</b> option was added.
12.2(8)YD	This command was incorporated in Cisco IOS Release 12.2(8)YD.
12.2(8)YW	This command was incorporated in Cisco IOS Release 12.2(8)YW.

## Usage Guidelines

This command is useful for system operators and development engineers if problems are encountered with communication between the GGSN and the SGSN using GTP.



### Caution

Because the **debug gprs gtp** command generates a significant amount of output, use it only when traffic on the GPRS network is low, so other activity on the system is not adversely affected.

## Examples

The following example enables the display of events related to GTP processing on the GGSN:

```
Router# debug gprs gtp events
```

The following example enables the display of GTP signaling messages:

```
Router# debug gprs gtp messages
```

The following example enables the display of GTP packets sent between the SGSN and GGSN:

```
Router# debug gprs gtp packets
```

The following example enables the display of GTP PPP events between the SGSN and GGSN:

```
Router# debug gprs gtp ppp events
```

The following example enables the display of detailed GTP PPP debug output along with GTP PPP events between the SGSN and GGSN:

```
Router# debug gprs gtp ppp details
```

```
Router# debug gprs gtp ppp events
```

# debug gprs gtp-director

To display information about the GTP Director Module (GDM), use the **debug gprs gtp-director** privileged EXEC command. To disable debugging output, use the **no** form of this command.

**debug gprs gtp-director {events | packets}**

**no debug gprs gtp-director {events | packets}**

## Syntax Description

<b>events</b>	Displays events related to GDM processing.
<b>packets</b>	Displays packets that are sent between GDM and a GGSN.

## Defaults

No default behavior or values.

## Command History

Release	Modification
12.2(4)MX	This command was introduced.
12.2(8)YD	This command was incorporated in Cisco IOS Release 12.2(8)YD.
12.2(8)YW	This command was incorporated in Cisco IOS Release 12.2(8)YW.

## Usage Guidelines

This command is useful for system operators and development engineers if problems are encountered with communication between GDM and an SGSN, or between GDM and a GGSN.



### Caution

Because the **debug gprs gtp-director** command generates a significant amount of output, use it only when traffic on the GPRS network is low, so other activity on the system is not adversely affected.

## Examples

The following debug examples provide sample output for a create PDP context request, delete PDP context request, and clear PDP context using PPP regeneration on the GGSN. The first three examples show output related to debug events messaging only. The last three examples show output while both debug events and details are enabled on the GGSN.

### Example 1

The following example displays events related to PPP regeneration processing for a create PDP context requested received by the GGSN:

```
Router# debug gprs gtp-director events
*Mar 1 00:02:42.787: GPRS:1111110000000000:Authen: PAP username: user@pdn.com
*Mar 1 00:02:42.787: GPRS:1111110000000000:Processing Initiate PPP regen from reqQ
*Mar 1 00:02:42.787: GPRS:1111110000000000:got event [REQUEST PPP REGEN] in state [IDLE]
*Mar 1 00:02:42.787: GPRS:1111110000000000:state [IDLE->AUTHORIZING] on event [REQUEST
PPP REGEN]
*Mar 1 00:02:42.787: GPRS:1111110000000000:Got VPN authorization info
*Mar 1 00:02:42.787: GPRS:1111110000000000:got event [AUTHOR SUCCESS] in state
[AUTHORIZING]
*Mar 1 00:02:42.787: GPRS:1111110000000000:state [AUTHORIZING->VPDN CONNECTING] on event
[AUTHOR SUCCESS]
```

```

*Mar 1 00:02:42.787: GPRS:1111110000000000:Author succeeded, establishing the tunnel
*Mar 1 00:02:42.787: GPRS:1111110000000000:Create/Clone vaccess to negotiate PPP
*Mar 1 00:02:42.791: GPRS:1111110000000000:MS no static IP addr. Get one via IPCP
*Mar 1 00:02:42.827: GPRS:1111110000000000:VPDN to inform PPP regen: CONNECTED
*Mar 1 00:02:42.827: GPRS:1111110000000000:got event [VPDN CONNECTED] in state [VPDN
CONNECTING]
*Mar 1 00:02:42.827: GPRS:1111110000000000:state [VPDN CONNECTING->PPP NEGOTIATING] on
event [VPDN CONNECTED]
*Mar 1 00:02:42.827: GPRS:1111110000000000:Start PPP negotiations on vaccess
*Mar 1 00:02:42.831: %LINK-3-UPDOWN: Interface Virtual-Access3, changed state to up
*Mar 1 00:02:42.835: GPRS:1111110000000000:IPCP is up
*Mar 1 00:02:42.835: GPRS:1111110000000000:IP addr 10.10.1.187 is negotiated for MS
*Mar 1 00:02:42.835: GPRS:1111110000000000:DNS - Primary: 10.3.0.1 Secondary: 0.0.0.0
NetBios - Primary: 0.0.0.0, Secondary: 0.0.0.0
*Mar 1 00:02:42.835: GPRS:1111110000000000:PPP connected
*Mar 1 00:02:42.835: GPRS:1111110000000000:got event [PPP NEGOTIATED] in state [PPP
NEGOTIATING]
*Mar 1 00:02:42.835: GPRS:1111110000000000:state [PPP NEGOTIATING->PPP CONNECTED] on
event [PPP NEGOTIATED]
*Mar 1 00:02:42.835: GPRS:1111110000000000:PPP succeeded negotiation, session established
*Mar 1 00:02:43.835: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access3,
changed state to up

```

### Example 2

The following example displays events related to PPP regeneration processing for a delete PDP context requested received by the GGSN:

```

Router# debug gprs gtp-director events
*Mar 1 00:03:18.331: GPRS:1111110000000000:GTP disconnecting the PPP regen session
*Mar 1 00:03:18.331: GPRS:1111110000000000:Processing Disconnect PPP regen from reqQ
*Mar 1 00:03:18.331: GPRS:1111110000000000:got event [CANCEL REGEN'ED PPP] in state [PPP
CONNECTED]
*Mar 1 00:03:18.331: GPRS:1111110000000000:state [PPP CONNECTED->PPP TERMINATING] on
event [CANCEL REGEN'ED PPP]
*Mar 1 00:03:18.331: GPRS:1111110000000000:Cancel request after VPND tunnel is up
*Mar 1 00:03:18.335: GPRS:1111110000000000:PPP down
*Mar 1 00:03:18.335: GPRS:1111110000000000:got event [PPP FAILED] in state [PPP
TERMINATING]
*Mar 1 00:03:18.339: GPRS:1111110000000000:state [PPP TERMINATING->IDLE] on event [PPP
FAILED]
*Mar 1 00:03:18.339: GPRS:1111110000000000:PPP failed negotiation
*Mar 1 00:03:18.339: GPRS:1111110000000000:got event [CLEANUP CONTEXT] in state [IDLE]
*Mar 1 00:03:18.339: GPRS:1111110000000000:VPDN to inform PPP regen: DISCONNECTED
*Mar 1 00:03:18.339: GPRS:1111110000000000:got event [VPDN DISCONNECTED] in state [IDLE]
*Mar 1 00:03:18.339: GPRS:1111110000000000:state [IDLE->IDLE] on event [CLEANUP CONTEXT]
*Mar 1 00:03:18.339: GPRS:1111110000000000:Freeing context structure
*Mar 1 00:03:18.339: %LINK-3-UPDOWN: Interface Virtual-Access3, changed state to down
*Mar 1 00:03:19.331: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access3,
changed state to down

```

### Example 3

The following example displays events related to PPP regeneration processing as the GGSN clears a PDP context request:

```

Router# debug gprs gtp-director events
*Mar 1 00:04:50.083: GPRS:1111110000000000:GTP disconnecting the PPP regen session
*Mar 1 00:04:50.083: GPRS:1111110000000000:Processing Disconnect PPP regen from reqQ
*Mar 1 00:04:50.083: GPRS:1111110000000000:got event [CANCEL REGEN'ED PPP] in state [PPP
CONNECTED]
*Mar 1 00:04:50.083: GPRS:1111110000000000:state [PPP CONNECTED->PPP TERMINATING] on
event [CANCEL REGEN'ED PPP]
*Mar 1 00:04:50.083: GPRS:1111110000000000:Cancel request after VPND tunnel is up

```



```

*Mar 1 00:04:50.087: GPRS:11111000000000:PPP down
*Mar 1 00:04:50.087: GPRS:11111000000000:got event [PPP FAILED] in state [PPP
TERMINATING]
*Mar 1 00:04:50.091: GPRS:11111000000000:state [PPP TERMINATING->IDLE] on event [PPP
FAILED]
*Mar 1 00:04:50.091: GPRS:11111000000000:PPP failed negotiation
*Mar 1 00:04:50.091: GPRS:11111000000000:got event [CLEANUP CONTEXT] in state [IDLE]
*Mar 1 00:04:50.091: GPRS:11111000000000:VPDN to inform PPP regen: DISCONNECTED
*Mar 1 00:04:50.091: GPRS:11111000000000:got event [VPDN DISCONNECTED] in state [IDLE]
*Mar 1 00:04:50.091: GPRS:11111000000000:state [IDLE->IDLE] on event [CLEANUP CONTEXT]
*Mar 1 00:04:50.091: GPRS:11111000000000:Freeing context structure
*Mar 1 00:04:50.091: %LINK-3-UPDOWN: Interface Virtual-Access4, changed state to down
*Mar 1 00:04:51.083: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access4,
changed state to down

```

#### Example 4

The following example displays both debug events and details related to PPP regeneration processing for a create PDP context requested received by the GGSN:

```

Router# debug gprs gtp-director events
Router# debug gprs gtp-director details
*Mar 1 00:05:21.083: PPP-REGEN state counters: pending counter is 0
*Mar 1 00:05:21.083: State[IDLE] counter is 0
*Mar 1 00:05:21.083: State[AUTHORIZING] counter is 0
*Mar 1 00:05:21.083: State[VPDN CONNECTING] counter is 0
*Mar 1 00:05:21.083: State[PPP NEGOTIATING] counter is 0
*Mar 1 00:05:21.083: State[PPP CONNECTED] counter is 0
*Mar 1 00:05:21.083: State[PPP TERMINATING] counter is 0
*Mar 1 00:05:21.087: PPP-REGEN state counters: pending counter is 1
*Mar 1 00:05:21.087: State[IDLE] counter is 1
*Mar 1 00:05:21.087: State[AUTHORIZING] counter is 0
*Mar 1 00:05:21.087: State[VPDN CONNECTING] counter is 0
*Mar 1 00:05:21.087: State[PPP NEGOTIATING] counter is 0
*Mar 1 00:05:21.087: State[PPP CONNECTED] counter is 0
*Mar 1 00:05:21.087: State[PPP TERMINATING] counter is 0
*Mar 1 00:05:21.087: GPRS:11111000000000:Authen: PAP username: user@pdn.com
*Mar 1 00:05:21.087: GPRS:11111000000000:Session timer started
*Mar 1 00:05:21.087: GPRS:11111000000000:Processing Initiate PPP regen from reqQ
*Mar 1 00:05:21.087: GPRS:11111000000000:got event [REQUEST PPP REGEN] in state [IDLE]
*Mar 1 00:05:21.087: PPP-REGEN state counters: pending counter is 1
*Mar 1 00:05:21.087: State[IDLE] counter is 0
*Mar 1 00:05:21.087: State[AUTHORIZING] counter is 1
*Mar 1 00:05:21.087: State[VPDN CONNECTING] counter is 0
*Mar 1 00:05:21.087: State[PPP NEGOTIATING] counter is 0
*Mar 1 00:05:21.087: State[PPP CONNECTED] counter is 0
*Mar 1 00:05:21.087: State[PPP TERMINATING] counter is 0
*Mar 1 00:05:21.087: GPRS:11111000000000:state [IDLE->AUTHORIZING] on event [REQUEST
PPP REGEN]
*Mar 1 00:05:21.087: GPRS:11111000000000:Got VPN authorization info
*Mar 1 00:05:21.087: GPRS:11111000000000:got event [AUTHOR SUCCESS] in state
[AUTHORIZING]
*Mar 1 00:05:21.087: PPP-REGEN state counters: pending counter is 1
*Mar 1 00:05:21.087: State[IDLE] counter is 0
*Mar 1 00:05:21.087: State[AUTHORIZING] counter is 0
*Mar 1 00:05:21.087: State[VPDN CONNECTING] counter is 1
*Mar 1 00:05:21.087: State[PPP NEGOTIATING] counter is 0
*Mar 1 00:05:21.087: State[PPP CONNECTED] counter is 0
*Mar 1 00:05:21.087: State[PPP TERMINATING] counter is 0
*Mar 1 00:05:21.087: GPRS:11111000000000:state [AUTHORIZING->VPDN CONNECTING] on event
[AUTHOR SUCCESS]
*Mar 1 00:05:21.087: GPRS:11111000000000:Author succeeded, establishing the tunnel
*Mar 1 00:05:21.087: GPRS:11111000000000:Create/Clone vaccess to negotiate PPP
*Mar 1 00:05:21.091: GPRS:11111000000000:MS no static IP addr. Get one via IPCP

```

```

*Mar 1 00:05:21.127: GPRS:11111000000000:VPDN to inform PPP regen: CONNECTED
*Mar 1 00:05:21.127: GPRS:11111000000000:got event [VPDN CONNECTED] in state [VPDN
CONNECTING]
*Mar 1 00:05:21.127: PPP-REGEN state counters: pending counter is 1
*Mar 1 00:05:21.127: State[IDLE] counter is 0
*Mar 1 00:05:21.127: State[AUTHORIZING] counter is 0
*Mar 1 00:05:21.127: State[VPDN CONNECTING] counter is 0
*Mar 1 00:05:21.127: State[PPP NEGOTIATING] counter is 1
*Mar 1 00:05:21.127: State[PPP CONNECTED] counter is 0
*Mar 1 00:05:21.127: State[PPP TERMINATING] counter is 0
*Mar 1 00:05:21.127: GPRS:11111000000000:state [VPDN CONNECTING->PPP NEGOTIATING] on
event [VPDN CONNECTED]
*Mar 1 00:05:21.127: GPRS:11111000000000:Start PPP negotiations on vaccess
*Mar 1 00:05:21.131: %LINK-3-UPDOWN: Interface Virtual-Access5, changed state to up
*Mar 1 00:05:22.135: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access5,
changed state to up
*Mar 1 00:05:23.143: GPRS:11111000000000:IPCP is up
*Mar 1 00:05:23.143: GPRS:11111000000000:LNS allocates 10.10.1.187 for MS
*Mar 1 00:05:23.143: GPRS:11111000000000:IP addr 10.10.1.187 is negotiated for MS
*Mar 1 00:05:23.143: GPRS:11111000000000:DNS - Primary: 10.3.0.1 Secondary: 0.0.0.0
NetBios - Primary: 0.0.0.0, Secondary: 0.0.0.0
*Mar 1 00:05:23.143: GPRS:11111000000000:PPP connected
*Mar 1 00:05:23.143: GPRS:11111000000000:got event [PPP NEGOTIATED] in state [PPP
NEGOTIATING]
*Mar 1 00:05:23.143: PPP-REGEN state counters: pending counter is 0
*Mar 1 00:05:23.143: State[IDLE] counter is 0
*Mar 1 00:05:23.143: State[AUTHORIZING] counter is 0
*Mar 1 00:05:23.143: State[VPDN CONNECTING] counter is 0
*Mar 1 00:05:23.143: State[PPP NEGOTIATING] counter is 0
*Mar 1 00:05:23.143: State[PPP CONNECTED] counter is 1
*Mar 1 00:05:23.143: State[PPP TERMINATING] counter is 0
*Mar 1 00:05:23.143: GPRS:11111000000000:state [PPP NEGOTIATING->PPP CONNECTED] on
event [PPP NEGOTIATED]
*Mar 1 00:05:23.143: GPRS:11111000000000:PPP succeeded negotiation, session established
*Mar 1 00:05:23.143: GPRS:11111000000000:Session timer stopped

```

### Example 5

The following example displays both debug events and details related to PPP regeneration processing for a delete PDP context requested received by the GGSN:

```

Router# debug gprs gtp-director events
Router# debug gprs gtp-director details
*Mar 1 00:05:52.399: PPP-REGEN state counters: pending counter is 0
*Mar 1 00:05:52.399: State[IDLE] counter is 0
*Mar 1 00:05:52.399: State[AUTHORIZING] counter is 0
*Mar 1 00:05:52.399: State[VPDN CONNECTING] counter is 0
*Mar 1 00:05:52.399: State[PPP NEGOTIATING] counter is 0
*Mar 1 00:05:52.399: State[PPP CONNECTED] counter is 1
*Mar 1 00:05:52.399: State[PPP TERMINATING] counter is 0
*Mar 1 00:05:52.399: GPRS:11111000000000:PPP regen current state PPP CONNECTED
*Mar 1 00:05:52.399: GPRS:11111000000000:GTP disconnecting the PPP regen session
*Mar 1 00:05:52.399: GPRS:11111000000000:Processing Disconnect PPP regen from reqQ
*Mar 1 00:05:52.399: GPRS:11111000000000:got event [CANCEL REGEN'ED PPP] in state [PPP
CONNECTED]
*Mar 1 00:05:52.399: PPP-REGEN state counters: pending counter is 1
*Mar 1 00:05:52.399: State[IDLE] counter is 0
*Mar 1 00:05:52.399: State[AUTHORIZING] counter is 0
*Mar 1 00:05:52.399: State[VPDN CONNECTING] counter is 0
*Mar 1 00:05:52.399: State[PPP NEGOTIATING] counter is 0
*Mar 1 00:05:52.399: State[PPP CONNECTED] counter is 0
*Mar 1 00:05:52.399: State[PPP TERMINATING] counter is 1
*Mar 1 00:05:52.399: GPRS:11111000000000:state [PPP CONNECTED->PPP TERMINATING] on
event [CANCEL REGEN'ED PPP]

```

```

*Mar 1 00:05:52.399: GPRS:11111000000000:Cancel request after VPND tunnel is up
*Mar 1 00:05:52.403: GPRS:11111000000000:PPP down
*Mar 1 00:05:52.403: GPRS:11111000000000:got event [PPP FAILED] in state [PPP
TERMINATING]
*Mar 1 00:05:52.407: PPP-REGEN state counters: pending counter is 1
*Mar 1 00:05:52.407: State[IDLE] counter is 1
*Mar 1 00:05:52.407: State[AUTHORIZING] counter is 0
*Mar 1 00:05:52.407: State[VPDN CONNECTING] counter is 0
*Mar 1 00:05:52.407: State[PPP NEGOTIATING] counter is 0
*Mar 1 00:05:52.407: State[PPP CONNECTED] counter is 0
*Mar 1 00:05:52.407: State[PPP TERMINATING] counter is 0
*Mar 1 00:05:52.407: GPRS:11111000000000:state [PPP TERMINATING->IDLE] on event [PPP
FAILED]
*Mar 1 00:05:52.407: GPRS:11111000000000:PPP failed negotiation
*Mar 1 00:05:52.407: GPRS:11111000000000:got event [CLEANUP CONTEXT] in state [IDLE]
*Mar 1 00:05:52.407: GPRS:11111000000000:VPDN to inform PPP regen: DISCONNECTED
*Mar 1 00:05:52.407: GPRS:11111000000000:got event [VPDN DISCONNECTED] in state [IDLE]
*Mar 1 00:05:52.407: GPRS:11111000000000:state [IDLE->IDLE] on event [CLEANUP CONTEXT]
*Mar 1 00:05:52.407: GPRS:11111000000000:Freeing context structure
*Mar 1 00:05:52.407: GPRS:11111000000000:Session timer stopped
*Mar 1 00:05:52.407: PPP-REGEN state counters: pending counter is 0
*Mar 1 00:05:52.407: State[IDLE] counter is 0
*Mar 1 00:05:52.407: State[AUTHORIZING] counter is 0
*Mar 1 00:05:52.407: State[VPDN CONNECTING] counter is 0
*Mar 1 00:05:52.407: State[PPP NEGOTIATING] counter is 0
*Mar 1 00:05:52.407: State[PPP CONNECTED] counter is 0
*Mar 1 00:05:52.407: State[PPP TERMINATING] counter is 0
*Mar 1 00:05:52.407: GPRS:11111000000000:PPP regen context 0x6219F4BC released
*Mar 1 00:05:52.407: GPRS:GTP-PPP-REGEN context magic(0x619D4FBC) invalid
*Mar 1 00:05:52.407: %LINK-3-UPDOWN: Interface Virtual-Access5, changed state to down
*Mar 1 00:05:53.399: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access5,
changed state to down

```

### Example 6

The following example displays both debug events and details related to PPP regeneration processing as the GGSN clears a PDP context request:

```

Router# debug gprs gtp-director events
Router# debug gprs gtp-director details
*Mar 1 00:06:34.907: PPP-REGEN state counters: pending counter is 0
*Mar 1 00:06:34.907: State[IDLE] counter is 0
*Mar 1 00:06:34.907: State[AUTHORIZING] counter is 0
*Mar 1 00:06:34.907: State[VPDN CONNECTING] counter is 0
*Mar 1 00:06:34.907: State[PPP NEGOTIATING] counter is 0
*Mar 1 00:06:34.907: State[PPP CONNECTED] counter is 1
*Mar 1 00:06:34.907: State[PPP TERMINATING] counter is 0
*Mar 1 00:06:34.907: GPRS:11111000000000:PPP regen current state PPP CONNECTED
*Mar 1 00:06:34.907: GPRS:11111000000000:GTP disconnecting the PPP regen session
*Mar 1 00:06:34.907: GPRS:11111000000000:Processing Disconnect PPP regen from reqQ
*Mar 1 00:06:34.907: GPRS:11111000000000:got event [CANCEL REGEN'ED PPP] in state [PPP
CONNECTED]
*Mar 1 00:06:34.907: PPP-REGEN state counters: pending counter is 1
*Mar 1 00:06:34.907: State[IDLE] counter is 0
*Mar 1 00:06:34.907: State[AUTHORIZING] counter is 0
*Mar 1 00:06:34.907: State[VPDN CONNECTING] counter is 0
*Mar 1 00:06:34.907: State[PPP NEGOTIATING] counter is 0
*Mar 1 00:06:34.907: State[PPP CONNECTED] counter is 0
*Mar 1 00:06:34.907: State[PPP TERMINATING] counter is 1
*Mar 1 00:06:34.907: GPRS:11111000000000:state [PPP CONNECTED->PPP TERMINATING] on
event [CANCEL REGEN'ED PPP]
*Mar 1 00:06:34.907: GPRS:11111000000000:Cancel request after VPND tunnel is up
*Mar 1 00:06:34.911: GPRS:11111000000000:PPP down

```

```

*Mar 1 00:06:34.911: GPRS:1111110000000000:got event [PPP FAILED] in state [PPP
TERMINATING]
*Mar 1 00:06:34.915: PPP-REGEN state counters: pending counter is 1
*Mar 1 00:06:34.915: State[IDLE] counter is 1
*Mar 1 00:06:34.915: State[AUTHORIZING] counter is 0
*Mar 1 00:06:34.915: State[VPDN CONNECTING] counter is 0
*Mar 1 00:06:34.915: State[PPP NEGOTIATING] counter is 0
*Mar 1 00:06:34.915: State[PPP CONNECTED] counter is 0
*Mar 1 00:06:34.915: State[PPP TERMINATING] counter is 0
*Mar 1 00:06:34.915: GPRS:1111110000000000:state [PPP TERMINATING->IDLE] on event [PPP
FAILED]
*Mar 1 00:06:34.915: GPRS:1111110000000000:PPP failed negotiation
*Mar 1 00:06:34.915: GPRS:1111110000000000:got event [CLEANUP CONTEXT] in state [IDLE]
*Mar 1 00:06:34.915: GPRS:1111110000000000:VPDN to inform PPP regen: DISCONNECTED
*Mar 1 00:06:34.915: GPRS:1111110000000000:got event [VPDN DISCONNECTED] in state [IDLE]
*Mar 1 00:06:34.915: GPRS:1111110000000000:state [IDLE->IDLE] on event [CLEANUP CONTEXT]
*Mar 1 00:06:34.915: GPRS:1111110000000000:Freeing context structure
*Mar 1 00:06:34.915: GPRS:1111110000000000:Session timer stopped
*Mar 1 00:06:34.915: PPP-REGEN state counters: pending counter is 0
*Mar 1 00:06:34.915: State[IDLE] counter is 0
*Mar 1 00:06:34.915: State[AUTHORIZING] counter is 0
*Mar 1 00:06:34.915: State[VPDN CONNECTING] counter is 0
*Mar 1 00:06:34.915: State[PPP NEGOTIATING] counter is 0
*Mar 1 00:06:34.915: State[PPP CONNECTED] counter is 0
*Mar 1 00:06:34.915: State[PPP TERMINATING] counter is 0
*Mar 1 00:06:34.915: GPRS:1111110000000000:PPP regen context 0x62196E10 released
*Mar 1 00:06:34.915: GPRS:GTP-PPP-REGEN context magic(0x619D4FBC) invalid
*Mar 1 00:06:34.915: %LINK-3-UPDOWN: Interface Virtual-Access3, changed state to down
*Mar 1 00:06:35.907: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access3,
changed state to down

```

# debug gprs gtp parsing

To display information about the parsing of GPRS Tunneling Protocol (GTP) information elements (IEs) in signaling requests, use the **debug gprs gtp parsing** privileged EXEC command. To disable debugging output, use the **no** form of this command.

**debug gprs gtp parsing**

**no debug gprs gtp parsing**

## Syntax Description

This command has no arguments or keywords.

## Defaults

No default behavior or values.

## Command History

Release	Modification
12.2(4)MX	This command was introduced.
12.2(8)YD	This command was incorporated in Cisco IOS Release 12.2(8)YD.
12.2(8)YW	This command was incorporated in Cisco IOS Release 12.2(8)YW.

## Usage Guidelines

This command is useful for system operators and development engineers to verify parsing of GTP IEs in signaling requests that are received by GDM or by the GGSN. If the packet is parsed successfully, you will receive a message along with the TID for the packet as shown in the following example:

```
GPRS:TID:7300000000000000:Packet Parsed successfully
```

The **debug gprs gtp parsing** command can be used to verify GDM or GGSN processing of IEs.



### Caution

Because the **debug gprs gtp parsing** command generates a significant amount of output, use it only when traffic on the GPRS network is low, so other activity on the system is not adversely affected.

## Examples

The following example enables the display of debug messages that occur while GDM or the GGSN parses GTP IEs:

```
Router# debug gprs gtp parsing
```

# debug gprs gtp ppp

To display information about PPP PDP type processing on the GGSN, use the **debug gprs gtp ppp** privileged EXEC command. To disable debugging output, use the **no** form of this command.

**debug gprs gtp ppp {events | details}**

**no debug gprs gtp ppp {events | details}**

## Syntax Description

<b>events</b>	Displays messages specific to certain conditions that are occurring during PPP PDP type processing.
<b>details</b>	Displays more extensive and lower-level messages related to PPP PDP type processing.

## Defaults

No default behavior or values.

## Command History

Release	Modification
12.2(4)MX	This command was introduced.
12.2(8)YD	This command was incorporated in Cisco IOS Release 12.2(8)YD.
12.2(8)YW	This command was incorporated in Cisco IOS Release 12.2(8)YW.

## Usage Guidelines

This command is useful for system operators and development engineers if problems are encountered with PPP PDP type processing on the GGSN.

You can enable both forms of the **debug gprs gtp ppp** command at the same time, as separate command line entries. The **events** keyword generates output specific to certain conditions that are occurring, which helps qualify the output being received using the **details** option.



### Caution

Because the **debug gprs gtp ppp** command generates a significant amount of output, use it only when traffic on the GPRS network is low, so other activity on the system is not adversely affected.

## Examples

The following debug examples provide sample output for a create PDP context request and clear PDP context using PPP PDP type on the GGSN. The examples show output while both debug events and details are enabled on the GGSN.

### Example 1

The following example displays details and events output related to PPP PDP context processing for a create PDP context requested received by the GGSN:

```
Router# debug gprs gtp ppp events
GTP PPP events display debugging is on
Router# debug gprs gtp ppp details
GTP PPP details display debugging is on
tb9-7200b#
```

```

3d23h: GPRS:
3d23h: GTP-PPP Fal/0: Create new gtp_ppp_info
3d23h: GPRS:
3d23h: GTP-PPP: domain gprs.cisco.com not in any VPDN group
3d23h: GPRS:
3d23h: GTP-PPP: aaa-group accounting not configured under APN gprs.cisco.com
3d23h: GPRS:GTP-PPP: Don't cache internally generated pak's header
3d23h: %LINK-3-UPDOWN: Interface Virtual-Access2, changed state to up
3d23h: GPRS:
3d23h: GTP-PPP Vi2: gtp_ppp_cstate_react changing states
3d23h: GPRS:GTP-PPP: pdp_entry 0x62F442A4, recv ppp data pak
3d23h: GPRS:GTP-PPP Vi2: proc_udp_input pak's linktype = 30
3d23h: GPRS:GTP-PPP: pdp_entry 0x62F442A4, recv ppp data pak
3d23h: GPRS:GTP-PPP Vi2: proc_udp_input pak's linktype = 30
3d23h: GPRS:GTP-PPP: pdp_entry 0x62F442A4, recv ppp data pak
3d23h: GPRS:GTP-PPP Vi2: proc_udp_input pak's linktype = 30
3d23h: GPRS:
3d23h: GTP-PPP: Vi2: Concat names user00 & gprs.cisco.com
3d23h: GPRS:
3d23h: GTP-PPP: New username after concat: user00@gprs.cisco.com
3d23h: GPRS:
3d23h: GTP-PPP: Vi2: Concat names user00@gprs.cisco.com & gprs.cisco.com
3d23h: GPRS:
3d23h: GTP-PPP: New username after concat: user00@gprs.cisco.com
3d23h: GPRS:GTP-PPP: pdp_entry 0x62F442A4, recv ppp data pak
3d23h: GPRS:GTP-PPP Vi2: proc_udp_input pak's linktype = 30
3d23h: GPRS:GTP-PPP: pdp_entry 0x62F442A4, recv ppp data pak
3d23h: GPRS:GTP-PPP Vi2: proc_udp_input pak's linktype = 30
3d23h: GPRS:GTP-PPP: pdp_entry 0x62F442A4, recv ppp data pak
3d23h: GPRS:GTP-PPP Vi2: proc_udp_input pak's linktype = 30
3d23h: GPRS:GTP-PPP: pdp_entry 0x62F442A4, recv ppp data pak
3d23h: GPRS:GTP-PPP Vi2: proc_udp_input pak's linktype = 30
3d23h: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access2, changed state to up
3d23h: GPRS:GTP-PPP: pdp_entry 0x62F442A4, recv ppp data pak
3d23h: GPRS:GTP-PPP Vi2: proc_udp_input pak's linktype = 30
3d23h: GPRS:GTP-PPP: pdp_entry 0x62F442A4, recv ppp data pak
3d23h: GPRS:GTP-PPP Vi2: proc_udp_input pak's linktype = 30
3d23h: GPRS:
3d23h: GTP-PPP Vi2: gtp_ppp_protocol_up is notified about intf UP
3d23h: GPRS:
3d23h: GTP-PPP Vi2: PDP w/ MS addr 98.102.0.1 inserted into IP radix tree

```

## Example 2

The following example displays both details and events related to PPP PDP type processing after clearing PDP contexts on the GGSN:

```

Router# clear gprs gtp pdp-context all
3d23h: GPRS:GTP-PPP: pdp_entry 0x62F442A4, recv ppp data pak
3d23h: GPRS:GTP-PPP Vi2: proc_udp_input pak's linktype = 30
3d23h: GPRS:GTP-PPP: pdp_entry 0x62F442A4, recv ppp data pak
3d23h: GPRS:GTP-PPP Vi2: proc_udp_input pak's linktype = 30
3d23h: GPRS:
3d23h: GTP-PPP Vi2: gtp_ppp_pdp_terminate shutting down the vaccess
3d23h: GPRS:
3d23h: GTP-PPP Vi2: gtp_ppp_pdp_shut_va shutting down intf
3d23h: %LINK-3-UPDOWN: Interface Virtual-Access2, changed state to down
3d23h: GPRS:
3d23h: GTP-PPP Vi2: gtp_ppp_cstate_react changing states
3d23h: GPRS:
3d23h: GTP-PPP Vi2: gtp_ppp_free_va resetting intf vectors
3d23h: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access2, changed state to down

```

# debug gprs gtp ppp-regeneration

To display information about PPP regeneration processing on the GGSN, use the **debug gprs gtp ppp-regeneration** privileged EXEC command. To disable debugging output, use the **no** form of this command.

```
debug gprs gtp ppp-regeneration {events | details}
```

```
no debug gprs gtp ppp-regeneration {events | details}
```

## Syntax Description

<b>events</b>	Displays messages specific to certain conditions that are occurring during PPP regeneration processing.
<b>details</b>	Displays more extensive and lower-level messages related to PPP regeneration processing.

## Defaults

No default behavior or values.

## Command History

Release	Modification
12.2(4)MX	This command was introduced.
12.2(8)YD	This command was incorporated in Cisco IOS Release 12.2(8)YD.
12.2(8)YW	This command was incorporated in Cisco IOS Release 12.2(8)YW.

## Usage Guidelines

This command is useful for system operators and development engineers if problems are encountered with communication between GDM and a GGSN.

You can enable both forms of the **debug gprs gtp ppp-regeneration** command at the same time, as separate command line entries. The **events** keyword generates output specific to certain conditions that are occurring, which helps qualify the output being received using the **details** option.



### Caution

Because the **debug gprs gtp ppp-regeneration** command generates a significant amount of output, use it only when traffic on the GPRS network is low, so other activity on the system is not adversely affected.

## Examples

The following debug examples provide sample output for a create PDP context request and clear PDP context using PPP regeneration on the GGSN. The examples show output while both debug events and details are enabled on the GGSN.

### Example 1

The following example displays details and events output related to PPP regeneration processing for a create PDP context requested received by the GGSN:

```
Router# debug gprs gtp ppp-regeneration details
GTP PPP regeneration details display debugging is on
Router# debug gprs gtp ppp-regeneration events
GTP PPP regeneration events display debugging is on
```



```

06:24:02: PPP-REGEN state counters: pending counter is 0
06:24:02:      State[IDLE] counter is 0
06:24:02:      State[AUTHORIZING] counter is 0
06:24:02:      State[VPDN CONNECTING] counter is 0
06:24:02:      State[PPP NEGOTIATING] counter is 0
06:24:02:      State[PPP CONNECTED] counter is 0
06:24:02:      State[PPP TERMINATING] counter is 0
06:24:02: PPP-REGEN state counters: pending counter is 1
06:24:02:      State[IDLE] counter is 1
06:24:02:      State[AUTHORIZING] counter is 0
06:24:02:      State[VPDN CONNECTING] counter is 0
06:24:02:      State[PPP NEGOTIATING] counter is 0
06:24:02:      State[PPP CONNECTED] counter is 0
06:24:02:      State[PPP TERMINATING] counter is 0
06:24:02: GPRS:101111111500001:Authen: PAP username: tomy1@corporate_1.com
06:24:02: GPRS:101111111500001:Session timer started
06:24:02: GPRS:Processing PPP regen reqQ
06:24:02: GPRS:101111111500001:Processing Initiate PPP  regen from reqQ
06:24:02: GPRS:101111111500001:got event [REQUEST PPP REGEN] in state [IDLE]
06:24:02: PPP-REGEN state counters: pending counter is 1
06:24:02:      State[IDLE] counter is 0
06:24:02:      State[AUTHORIZING] counter is 1
06:24:02:      State[VPDN CONNECTING] counter is 0
06:24:02:      State[PPP NEGOTIATING] counter is 0
06:24:02:      State[PPP CONNECTED] counter is 0
06:24:02:      State[PPP TERMINATING] counter is 0
06:24:02: GPRS:101111111500001:state [IDLE->AUTHORIZING] on event [REQUEST PPP REGEN]
06:24:02: GPRS:101111111500001:Got VPN authorization info
06:24:02: GPRS:101111111500001:got event [AUTHOR SUCCESS] in state [AUTHORIZING]
06:24:02: PPP-REGEN state counters: pending counter is 1
06:24:02:      State[IDLE] counter is 0
06:24:02:      State[AUTHORIZING] counter is 0
06:24:02:      State[VPDN CONNECTING] counter is 1
06:24:02:      State[PPP NEGOTIATING] counter is 0
06:24:02:      State[PPP CONNECTED] counter is 0
06:24:02:      State[PPP TERMINATING] counter is 0
06:24:02: GPRS:101111111500001:state [AUTHORIZING->VPDN CONNECTING] on event [AUTHOR
SUCCESS]
06:24:02: GPRS:101111111500001:Author succeeded, establishing the tunnel
06:24:02: GPRS:101111111500001:Create/Clone vaccess to negotiate PPP
06:24:02: GPRS:101111111500001:no need to set NS ppp_config
06:24:02: GPRS:101111111500001:MS no static IP addr. Get one via IPCP
06:24:02: GPRS:101111111500001:VPDN to inform PPP regen: CONNECTED
06:24:02: GPRS:101111111500001:got event [VPDN CONNECTED] in state [VPDN CONNECTING]
06:24:02: PPP-REGEN state counters: pending counter is 1
06:24:02:      State[IDLE] counter is 0
06:24:02:      State[AUTHORIZING] counter is 0
06:24:02:      State[VPDN CONNECTING] counter is 0
06:24:02:      State[PPP NEGOTIATING] counter is 1
06:24:02:      State[PPP CONNECTED] counter is 0
06:24:02:      State[PPP TERMINATING] counter is 0
06:24:02: GPRS:101111111500001:state [VPDN CONNECTING->PPP NEGOTIATING] on event [VPDN
CONNECTED]
06:24:02: GPRS:101111111500001:Start PPP negotiations on vaccess
06:24:02: %LINK-3-UPDOWN: Interface Virtual-Access2, changed state to up
06:24:02: GPRS:101111111500001:IPCP is up
06:24:02: GPRS:101111111500001:LNS allocates 10.100.1.1 for MS
06:24:02: GPRS:101111111500001:IP addr 10.100.1.1 is negotiated for MS
06:24:02: GPRS:101111111500001:PPP connected
06:24:02: GPRS:101111111500001:got event [PPP NEGOTIATED] in state [PPP NEGOTIATING]
06:24:02: PPP-REGEN state counters: pending counter is 0
06:24:02:      State[IDLE] counter is 0
06:24:02:      State[AUTHORIZING] counter is 0
06:24:02:      State[VPDN CONNECTING] counter is 0

```

```

06:24:02:          State[PPP NEGOTIATING] counter is 0
06:24:02:          State[PPP CONNECTED] counter is 1
06:24:02:          State[PPP TERMINATING] counter is 0
06:24:02: GPRS:1011111111500001:state [PPP NEGOTIATING->PPP CONNECTED] on event [PPP
NEGOTIATED]
06:24:02: GPRS:1011111111500001:PPP succeeded negotiation, session established
06:24:02: GPRS:1011111111500001:Session timer stopped
06:24:03: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access2, changed state
to up

```

## Example 2

The following example displays both details and events related to PPP regeneration processing after clearing PDP contexts on the GGSN:

```

Router# clear gprs gtp pdp-context all
06:28:05: PPP-REGEN state counters: pending counter is 0
06:28:05:          State[IDLE] counter is 0
06:28:05:          State[AUTHORIZING] counter is 0
06:28:05:          State[VPDN CONNECTING] counter is 0
06:28:05:          State[PPP NEGOTIATING] counter is 0
06:28:05:          State[PPP CONNECTED] counter is 1
06:28:05:          State[PPP TERMINATING] counter is 0
06:28:05: GPRS:1011111111500001:PPP regen current state PPP CONNECTED
06:28:05: GPRS:1011111111500001:GTP disconnecting the PPP regen session
06:28:05: GPRS:Processing PPP regen reqQ
06:28:05: GPRS:1011111111500001:Processing Disconnect PPP regen from reqQ
06:28:05: GPRS:1011111111500001:got event [CANCEL REGEN'ED PPP] in state [PPP CONNECTED]
06:28:05: PPP-REGEN state counters: pending counter is 1
06:28:05:          State[IDLE] counter is 0
06:28:05:          State[AUTHORIZING] counter is 0
06:28:05:          State[VPDN CONNECTING] counter is 0
06:28:05:          State[PPP NEGOTIATING] counter is 0
06:28:05:          State[PPP CONNECTED] counter is 0
06:28:05:          State[PPP TERMINATING] counter is 1
06:28:05: GPRS:1011111111500001:state [PPP CONNECTED->PPP TERMINATING] on event [CANCEL
REGEN'ED PPP]
06:28:05: GPRS:1011111111500001:Cancel request after VPND tunnel is up
06:28:05: PPP-REGEN state counters: pending counter is 1
06:28:05:          State[IDLE] counter is 0
06:28:05:          State[AUTHORIZING] counter is 0
06:28:05:          State[VPDN CONNECTING] counter is 0
06:28:05:          State[PPP NEGOTIATING] counter is 0
06:28:05:          State[PPP CONNECTED] counter is 0
06:28:05:          State[PPP TERMINATING] counter is 1
06:28:05: GPRS:1011111111500001:PPP down
06:28:05: GPRS:1011111111500001:got event [PPP FAILED] in state [PPP TERMINATING]
06:28:05: PPP-REGEN state counters: pending counter is 1
06:28:05:          State[IDLE] counter is 1
06:28:05:          State[AUTHORIZING] counter is 0
06:28:05:          State[VPDN CONNECTING] counter is 0
06:28:05:          State[PPP NEGOTIATING] counter is 0
06:28:05:          State[PPP CONNECTED] counter is 0
06:28:05:          State[PPP TERMINATING] counter is 0
06:28:05: GPRS:1011111111500001:state [PPP TERMINATING->IDLE] on event [PPP FAILED]
06:28:05: GPRS:1011111111500001:LCP went down
06:28:05: GPRS:1011111111500001:VPDN disconnect
06:28:05: GPRS:1011111111500001:got event [CLEANUP CONTEXT] in state [IDLE]
06:28:05: GPRS:1011111111500001:state [IDLE->IDLE] on event [CLEANUP CONTEXT]
06:28:05: GPRS:1011111111500001:Freeing context structure
06:28:05: GPRS:1011111111500001:VPDN handle invalid, no need to free it
06:28:05: GPRS:1011111111500001:remove PPP regen context from Vi2
06:28:05: GPRS:1011111111500001:Session timer stopped

```

```
06:28:05: PPP-REGEN state counters: pending counter is 0
06:28:05:           State[IDLE] counter is 0
06:28:05:           State[AUTHORIZING] counter is 0
06:28:05:           State[VPDN CONNECTING] counter is 0
06:28:05:           State[PPP NEGOTIATING] counter is 0
06:28:05:           State[PPP CONNECTED] counter is 0
06:28:05:           State[PPP TERMINATING] counter is 0
06:28:05: GPRS:101111111500001:PPP regen context 0x633F196C released
06:28:05: %LINK-3-UPDOWN: Interface Virtual-Access2, changed state to down
06:28:06: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access2, changed state
to down
```

# debug gprs radius

To display information about Remote Access Dial-In User Service (RADIUS) processing on the GGSN, use the **debug gprs radius** privileged EXEC command. To disable debugging output, use the **no** form of this command.

**debug gprs radius**

**no debug gprs radius**

## Syntax Description

This command has no arguments or keywords.

## Defaults

No default behavior or values.

## Command History

Release	Modification
12.2(4)MX	This command was introduced.
12.2(8)YD	This command was incorporated in Cisco IOS Release 12.2(8)YD.
12.2(8)YW	This command was incorporated in Cisco IOS Release 12.2(8)YW.

## Usage Guidelines

This command is useful for system operators and development engineers if problems are encountered with communication between a RADIUS server and the GGSN.



### Caution

Because the **debug gprs radius** command generates a significant amount of output, use it only when traffic on the GPRS network is low, so other activity on the system is not adversely affected.

## Examples

The following example enables the display of debug messages related to RADIUS processing on the GGSN:

```
Router# debug gprs radius
```