



## 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*.



### Caution

---

Because debugging output is assigned high priority in the CPU process, it can diminish the performance of the router or even render it 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 traffic and fewer users. Debugging during these periods decreases the likelihood that increased **debug** command processing overhead will affect system use.

---

This chapter contains the following section and commands:

- [TID/IMSI/MSISDN-Based Conditionally Triggered Debugging, page 538](#)
- [debug condition calling, page 540](#)
- [debug data-store, page 542](#)
- [debug data-store detail, page 543](#)
- [debug diameter, page 544](#)
- [debug ggsn quota-server, page 545](#)
- [debug gprs category fsm event, page 546](#)
- [debug gprs dcca, page 547](#)
- [debug gprs dfp, page 548](#)
- [debug gprs dhcp, page 550](#)
- [debug gprs gtp, page 552](#)
- [debug gprs gtp parsing, page 554](#)
- [debug gprs gtp ppp, page 555](#)
- [debug gprs gtp ppp-regeneration, page 558](#)
- [debug gprs radius, page 562](#)
- [debug gprs redundancy, page 563](#)

# TID/IMSI/MSISDN-Based Conditionally Triggered Debugging

When the TID/IMSI/MSISDN-based conditionally triggered debugging feature is enabled, the GGSN generates debugging messages for PDP contexts that match a particular tunnel ID (TID), International Mobile Subscriber Identity (IMSI) value, or Mobile Station ISDN number (MSISDN) entering or leaving the GGSN. The GGSN will not generate debugging output for PDP contexts containing a different TID, IMSI, or MSISDN value.

Normally, the GGSN will generate debugging messages for every PDP context, resulting in a large number of messages that consume system resources and can make it difficult to find the specific information you need. By limiting the number of debugging messages, you can receive messages related to only to PDP contexts you want to troubleshoot.

## Usage Guidelines for TID/IMSI/MSISDN-Based Conditional Debugging

Use the following guidelines when configuring TID/IMSI/MSISDN-based conditional debugging on a GGSN.

1. Before enabling a **debug gprs** command, first enable TID/IMSI/MSISDN-based debugging using the **debug condition calling** command. Ensure that the TID/IMSI or MSISDN string match the ones from the Create Request.

For examples:

For a create request with TID 12345678090000B0, you would enter:

```
GGSN# debug condition calling 12345678090000B0
Condition 1 set
GGSN#
```

For a create request with IMSI 21436579000000, you would enter:

```
GGSN# debug condition calling 21436579000000
Condition 2 set
GGSN#
```

For a create request with MSISDN 1112223344, you would enter:

```
GGSN# debug condition calling msisdn-1112223344
Condition 3 set
GGSN#
```

To verify the set conditions, enter:

```
GGSN# show debug condition all
Condition 1: calling 12345678090000B0 (0 flags triggered)
Condition 2: calling 21436579000000 (0 flags triggered)
Condition 3: calling 1112223344 (0 flags triggered)
GGSN#
```

2. After turning on TID, IMSI, or MSISDN-based debugging, turn on GPRS debugging by entering the **debug gprs gtp** and/or **debug gprs charging** commands.

Once this step is completed, when PDP Context Create Requests are received, the GGSN will display debug messages for those create requests with either a matching TID, IMSI, or MSISDN.

3. Because the **no debug all** command does not disable conditional debug flags, to ensure that you do not receive a flood of debugging messages when disabling debugging, turn off GPRS debug flags first using the **no debug all** command as follows:

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

```
GGSN# show debug condition all
Condition 1: calling 12345678090000B0 (1 flags triggered)
Condition 2: calling 21436579000000 (1 flags triggered)
Condition 3: calling 1112223344 (1 flags triggered)
```

```
GGSN#
```

4. Disable the conditional debug flags using the **no debug condition all** command:

```
GGSN# no debug condition all
Removing all conditions may cause a flood of debugging messages to result, unless
specified debugging flags are first removed.
```

```
Proceed with the removal of all conditions [yes/no] y
2 conditions have been removed
```

5. Verify that the conditional debug flags have been removed using the **show debug condition all** command:

```
GGSN# show debug condition all
% No conditions found
```

# debug condition calling

To limit output for some debug commands based on specified conditions, use the **debug condition** command in privileged EXEC mode. To remove the specified condition, use the **no** form of this command.

```
debug condition { username username | called dial-string | caller dial-string | vcid vc-id |
ip ip-address | calling [tid | imsi | msisdn-msisdn] }
```

```
no debug condition { condition-id | all }
```

## Syntax Description

<b>username</b> <i>username</i>	Generates debugging messages for interfaces with the specified username.
<b>called</b> <i>dial-string</i>	Generates debugging messages for interfaces with the called party number.
<b>caller</b> <i>dial-string</i>	Generates debugging messages for interfaces with the calling party number.
<b>vcid</b> <i>vc-id</i>	Generates debugging messages for the VC ID specified.
<b>ip</b> <i>ip-address</i>	Generates debugging messages for the IP address specified.
<b>calling</b> [ <i>tid</i>   <i>imsi</i> string   <b>msisdn</b> - <i>msisdn</i> ]	Displays events related to GTP processing on the GGSN based on tunnel identifier (TID), international mobile system identifier (IMSI), or Mobile Station ISDN number (MSISDN) in a PDP Context Create Request message.
<i>condition-id</i>	Removes the condition indicated.
<b>all</b>	Removes all conditional debugging conditions.

## Defaults

No default behavior or values.

## Command History

Release	Modification
12.3(2)XB	This command was introduced on the GGSN.
12.3(8)XU	This command was integrated into Cisco IOS Release 12.3(8)XU.
12.3(11)YJ	This command was integrated into Cisco IOS Release 12.3(11)YJ.
12.3(14)YQ	This command was integrated into Cisco IOS Release 12.3(14)YQ.
12.3(14)YU	This command was integrated into the Cisco IOS Release 12.3(14)YU and the <b>msisdn</b> keyword option was added.
12.4(2)XB	This command was integrated into Cisco IOS Release 12.4(2)XB.

## Usage Guidelines

Ensure that you enable TID/IMSI/MSISDN-based conditional debugging using the **debug condition calling** command before configuring the **debug gprs gtp** and **debug gprs charging**. In addition, ensure that you disable the **debug gprs gtp** and **debug gprs charging** commands using the **no debug all** command before disabling conditional debugging using the **no debug condition** command. This will prevent a flood of debug messages when you disable conditional debugging.

For more information on using the GGSN TID/IMSI/MSISDN-based conditional debugging, see [“TID/IMSI/MSISDN-Based Conditionally Triggered Debugging”](#) section on page 538.

---

**Examples****Example 1**

The following examples configure a conditional debug session based on a TID 12345678090000B0, IMSI 21436579000000, and MSISDN 408525823010:

```
GGSN# debug condition calling 12345678090000B0  
Condition 1 set  
GGSN#
```

```
GGSN# debug condition calling 21436579000000  
Condition 2 set  
GGSN#
```

```
GGSN# debug condition calling msisdn 408525823010  
Condition 3 set  
GGSN#
```

**Example 2**

The following example stops all conditional debugging:

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

# debug data-store

To display PSD related debugging messages for the gateway GPRS support node (GGSN), use the **debug data-store** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

**debug data-store**

**no debug data-store**

---

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

---

**Defaults** No default behavior or values

---

**Command Modes** Privileged EXEC

---

Command History	Release	Modification
	12.3(14)YU	This command was introduced.
	12.4(2)XB	This command was integrated into Cisco IOS Release 12.4(2)XB.

---



---

**Usage Guidelines** This command displays PSD-related debugging messages for the GGSN.



**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.

---



---

**Examples** The following example configures a debugging session to check PSD-related parameters:

```
Router# debug data-store
```

# debug data-store detail

To display extended details for PSD related debugging information, use the **debug data-store detail** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

**debug data-store detail**

**no debug data-store detail**

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

**Defaults** No default behavior or values

**Command Modes** Privileged EXEC

Command History	Release	Modification
	12.3(14)YU	This command was introduced.
	12.4(2)XB	This command was integrated into Cisco IOS Release 12.4(2)XB.

**Usage Guidelines** This command displays PSD-related debugging messages for the GGSN.



**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.

**Examples** The following example configures a detailed PSD-related debugging session:

```
Router# debug data-store details
```

# debug diameter

To display information about Diameter processing on the gateway GPRS support node (GGSN), use the **debug diameter** command in privilege EXEC mode.

**debug diameter {dcca | connection | error | packet | event | fsm | failover | all}**

## Syntax Description

<b>dcca</b>	Displays Diameter Credit Control Application-related information.
<b>connection</b>	Displays Diameter peer connection information.
<b>error</b>	Displays errors related to Diameter processing.
<b>packet</b>	Displays Diameter packets.
<b>event</b>	Displays Diameter-related events.
<b>fsm</b>	Displays Diameter-related fault state machine messages.
<b>failover</b>	Displays information about DCCA server failovers.
<b>all</b>	Displays all Diameter-related information.

## Defaults

No default behavior or values.

## Command Modes

Privilege EXEC

## Command History

Release	Modification
12.3(14)YQ	This command was introduced.
12.4(2)XB	This command was integrated into Cisco IOS Release 12.4(2)XB.

## Usage Guidelines

This command is useful for system operators and development engineers if problems are encountered with Diameter processing.

## Examples

The following configuration example displays Diameter-related events:

```
debug diameter event
```

# debug ggsn quota-server

To display debug information related to quota server processing on the gateway GPRS support node (GGSN), use the **debug ggsn quota-server** command in privilege EXEC mode.

**debug ggsn quota-server [details | packets [dump] | events | parsing | errors]**

## Syntax Description

<b>details</b>	Displays extended details about quota server operations on the GGSN.
<b>packets</b>	Displays packets sent between the quota server process on the GGSN and the CSG. Optionally, displays output in hexadecimal notation.
<b>events</b>	Displays events related to quota server processing on the GGSN.
<b>parsing</b>	Displays details about GTP TLV parsing between the quota server and the Content Services Gateway.
<b>errors</b>	Displays errors related to quota server processing on the GGSN

## Defaults

No default behavior or values.

## Command Modes

Privilege EXEC

## Command History

Release	Modification
12.3(14)YQ	This command was introduced.
12.3(14)YU	This command was integrated into Cisco IOS Release 12.3(14)YU.
12.4(2)XB	This command was integrated into Cisco IOS Release 12.4(2)XB.

## Usage Guidelines

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

## Examples

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

```
Router# debug ggsn quota-server events
```

The following example enables the display of packets sent between the quota server process on the GGSN and the CSG:

```
Router# debug ggsn quota-server packets
```

The following example enables the display of detailed quota server processing debug output:

```
Router# debug ggsn quota-server details
```

# debug gprs category fsm event

To display debug information related to service-aware gateway GPRS support node (GGSN) category events, and state transactions, use the **debug gprs category fsm event** command in privilege EXEC mode.

## debug gprs category fsm event

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

**Defaults** No default behavior or values.

**Command Modes** Privilege EXEC

Command History	Release	Modification
	12.3(14)YQ	This command was introduced.
	12.3(14)YU	This command was integrated into Cisco IOS Release 12.3(14)YU.
	12.4(2)XB	This command was integrated into Cisco IOS Release 12.4(2)XB.

**Usage Guidelines** This command is useful for system operators and development engineers if problems are encountered with eGGSN processing.

**Examples** The following example enables the display of eGGSN events and state transactions:

```
Router# debug ggsn eggsn category fsm event
```

# debug gprs dcca

To display troubleshooting information about DCCA processing on the gateway GPRS support node (GGSN), use the **debug gprs dcca** command in privilege EXEC mode.

## **debug gprs dcca**

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

**Defaults** No default behavior or values.

**Command Modes** Privilege EXEC

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	12.3(14)YQ	This command was introduced.
	12.3(14)YU	This command was integrated into Cisco IOS Release 12.3(14)YU.
	12.4(2)XB	This command was integrated into Cisco IOS Release 12.4(2)XB.

**Usage Guidelines** This command is useful for system operators and development engineers if Diameter protocol problems are encountered on the GGSN.

**Examples** The following configuration example displays information specific to DCCA processing:

```
debug gprs dcca
```

# debug gprs dfp

To display debug messages for GPRS DFP weight calculation, use the **debug gprs dfp** command in privileged EXEC mode. 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 integrated into Cisco IOS Release 12.2(4)MX.
12.2(8)YD	This command was integrated into Cisco IOS Release 12.2(8)YD.
12.2(8)YW	This command was integrated into Cisco IOS Release 12.2(8)YW.
12.3(2)XB	This command was integrated into Cisco IOS Release 12.3(2)XB.
12.3(8)XU	This command was integrated into Cisco IOS Release 12.3(8)XU.
12.3(11)YJ	This command was integrated into Cisco IOS Release 12.3(11)YJ.
12.3(14)YQ	This command was integrated into Cisco IOS Release 12.3(14)YQ.
12.3(14)YU	This command was integrated into Cisco IOS Release 12.3(14)YU.
12.4(2)XB	This command was integrated into Cisco IOS Release 12.4(2)XB.

## 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 gateway GPRS support node (GGSN), use the **debug gprs dhcp** command in privileged EXEC mode. 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 integrated into Cisco IOS Release 12.2(8)YD.
12.2(8)YW	This command was integrated into Cisco IOS Release 12.2(8)YW.
12.3(2)XB	This command was integrated into Cisco IOS Release 12.3(2)XB.
12.3(8)XU	This command was integrated into Cisco IOS Release 12.3(8)XU.
12.3(11)YJ	This command was integrated into Cisco IOS Release 12.3(11)YJ.
12.3(14)YQ	This command was integrated into Cisco IOS Release 12.3(14)YQ.
12.3(14)YU	This command was integrated into Cisco IOS Release 12.3(14)YU.
12.4(2)XB	This command was integrated into Cisco IOS Release 12.4(2)XB.

## 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
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

Command	Description
<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** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

**debug gprs gtp** {events | messages | packets}

**no debug gprs gtp** {events | messages | packets}

## 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.

## 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 into Cisco IOS Release 12.1(5)T.
12.2(4)MX	This command was integrated into Cisco IOS Release 12.2(4)MX, and the <b>ppp</b> { <b>details</b>   <b>events</b> } option was added.
12.2(8)YD	This command was integrated into Cisco IOS Release 12.2(8)YD.
12.2(8)YW	This command was integrated into Cisco IOS Release 12.2(8)YW.
12.3(2)XB	This command was integrated into Cisco IOS Release 12.3(2)XB.
12.3(8)XU	This command was integrated into Cisco IOS Release 12.3(8)XU.
12.3(11)YJ	This command was integrated into Cisco IOS Release 12.3(11)YJ.
12.3(14)YQ	This command was integrated into Cisco IOS Release 12.3(14)YQ.
12.3(14)YU	This command was integrated into Cisco IOS Release 12.3(14)YU.
12.4(2)XB	This command was integrated into Cisco IOS Release 12.4(2)XB.

## 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 parsing

To display information about the parsing of GPRS Tunneling Protocol (GTP) information elements (IEs) in signaling requests, use the **debug gprs gtp parsing** command in privileged EXEC mode. 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 integrated into Cisco IOS Release 12.2(8)YD.
12.2(8)YW	This command was integrated into Cisco IOS Release 12.2(8)YW.
12.3(2)XB	This command was integrated into Cisco IOS Release 12.3(2)XB.
12.3(8)XU	This command was integrated into Cisco IOS Release 12.3(8)XU.
12.3(11)YJ	This command was integrated into Cisco IOS Release 12.3(11)YJ.
12.3(14)YQ	This command was integrated into Cisco IOS Release 12.3(14)YQ.
12.3(14)YU	This command was integrated into Cisco IOS Release 12.3(14)YU.
12.4(2)XB	This command was integrated into Cisco IOS Release 12.4(2)XB.

## 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 gateway GPRS support node (GGSN), use the **debug gprs gtp ppp** command in privileged EXEC mode. 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 integrated into Cisco IOS Release 12.2(8)YD.
12.2(8)YW	This command was integrated into Cisco IOS Release 12.2(8)YW.
12.3(2)XB	This command was integrated into Cisco IOS Release 12.3(2)XB.
12.3(8)XU	This command was integrated into Cisco IOS Release 12.3(8)XU.
12.3(11)YJ	This command was integrated into Cisco IOS Release 12.3(11)YJ.
12.3(14)YQ	This command was integrated into Cisco IOS Release 12.3(14)YQ.
12.3(14)YU	This command was integrated into Cisco IOS Release 12.3(14)YU.
12.4(2)XB	This command was integrated into Cisco IOS Release 12.4(2)XB.

## 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
7200b#
3d23h: GPRS:
3d23h: GTP-PPP Fa1/0: 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: 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 integrated into Cisco IOS Release 12.2(8)YD.
12.2(8)YW	This command was integrated into Cisco IOS Release 12.2(8)YW.
12.3(2)XB	This command was integrated into Cisco IOS Release 12.3(2)XB.
12.3(8)XU	This command was integrated into Cisco IOS Release 12.3(8)XU.
12.3(11)YJ	This command was integrated into Cisco IOS Release 12.3(11)YJ.
12.3(14)YQ	This command was integrated into Cisco IOS Release 12.3(14)YQ.
12.3(14)YU	This command was integrated into Cisco IOS Release 12.3(14)YU.
12.4(2)XB	This command was integrated into Cisco IOS Release 12.4(2)XB.

## 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: tomyl@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:1011111111500001:IPCP is up
06:24:02: GPRS:1011111111500001:LNS allocates 10.100.1.1 for MS
06:24:02: GPRS:1011111111500001:IP addr 10.100.1.1 is negotiated for MS
06:24:02: GPRS:1011111111500001:PPP connected
06:24:02: GPRS:1011111111500001: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:101111111500001:state [PPP TERMINATING->IDLE] on event [PPP FAILED]
06:28:05: GPRS:101111111500001:LCP went down
06:28:05: GPRS:101111111500001:VPDN disconnect
06:28:05: GPRS:101111111500001:got event [CLEANUP CONTEXT] in state [IDLE]
06:28:05: GPRS:101111111500001:state [IDLE->IDLE] on event [CLEANUP CONTEXT]
06:28:05: GPRS:101111111500001:Freeing context structure
06:28:05: GPRS:101111111500001:VPDN handle invalid, no need to free it
06:28:05: GPRS:101111111500001:remove PPP regen context from Vi2
06:28:05: GPRS:101111111500001: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 gateway GPRS support node (GGSN), use the **debug gprs radius** command in privileged EXEC mode. 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 integrated into Cisco IOS Release 12.2(8)YD.
	12.2(8)YW	This command was integrated into Cisco IOS Release 12.2(8)YW.
	12.3(2)XB	This command was integrated into Cisco IOS Release 12.3(2)XB.
	12.3(8)XU	This command was integrated into Cisco IOS Release 12.3(8)XU.
	12.3(11)YJ	This command was integrated into Cisco IOS Release 12.3(11)YJ.
	12.3(14)YQ	This command was integrated into Cisco IOS Release 12.3(14)YQ.
	12.3(14)YU	This command was integrated into Cisco IOS Release 12.3(14)YU.
	12.4(2)XB	This command was integrated into Cisco IOS Release 12.4(2)XB.

**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
```

# debug gprs redundancy

To display debug messages, errors, events, or packets related to GTP session redundancy (GTP-SR), use the **debug gprs redundancy** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

**debug gprs redundancy** [**debug** | **errors** | **events** | **packets**]

**no debug gprs redundancy** [**debug** | **errors** | **events** | **packets**]

## Syntax Description

<b>debug</b>	Displays debug messages related to GTP-SR.
<b>errors</b>	Displays errors related to GTP-SR.
<b>events</b>	Displays events related to GTP-SR.
<b>packets</b>	Displays packets related to GTP-SR packets.

## Defaults

Disabled.

## Command Modes

Global configuration

## Command History

Release	Modification
12.3(11)YJ	This command was introduced.
12.3(14)YQ	This command was integrated into Cisco IOS Release 12.3(14)YQ.
12.3(14)YU	This command was integrated into Cisco IOS Release 12.3(14)YU.
12.4(2)XB	This command was integrated into Cisco IOS Release 12.4(2)XB.

## Usage Guidelines

This command displays debug level messages, errors, events, or packets for GTP-SR. It is useful for system operators and development engineers if problems are encountered with communication between the two GGSNs configured as an redundant pair and on which GTP-SR is enabled.

## Examples

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

```
Router# debug gprs redundancy
```

## Related Commands

Command	Description
<b>clear gprs redundancy statistics</b>	Clears statistics related to GTP-SR.
<b>gprs redundancy</b>	Enables GTP-SR on a GGSN.
<b>gprs redundancy charging sync-window cdr rec-seqnum</b>	Configures the window size used to determine when the CDR record sequence number needs to be synchronized to the Standby GGSN.

Command	Description
<b>gprs redundancy charging sync-window gtp seqnum</b>	Configures the window size used to determine when the GTP' sequence number needs to be synchronized to the Standby GGSN.
<b>show gprs redundancy</b>	Displays statistics related to GTP-SR.