

瞭解有關Debug命令的重要資訊

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[慣例](#)

[背景資訊](#)

[警告](#)

[偵錯之前](#)

[獲取調試輸出](#)

[控制檯埠](#)

[Aux埠](#)

[VTY埠](#)

[將消息記錄到內部緩衝區](#)

[將消息記錄到UNIX Syslog伺服器](#)

[其他調試前任務](#)

[停止除錯](#)

[使用debug ip packet命令](#)

[警告](#)

[有條件觸發的調試](#)

[相關資訊](#)

簡介

本文檔介紹使用 debug 命令(包括 debug ip packet 在Cisco IOS®平台上可用命令)的一般指南。

必要條件

需求

思科建議您瞭解以下主題：

-

- [使用控制檯、aux和vty埠連線到路由器](#)

-

- [一般Cisco IOS配置問題](#)

•

解釋Cisco IOS調試輸出

採用元件

本文件所述內容不限於特定軟體和硬體版本。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

慣例

如需文件慣例的詳細資訊，請參閱思科技術提示慣例。

背景資訊

本頁提供有關使用Cisco IOS平台上可用調試功能的一些一般準則，以及正確使用 **debug ip packet** 命令和條件調試的示例。

注意：本文檔不介紹如何使用和解釋特定調試命令和輸出。有關特定debug命令的資訊，請參閱相應的Cisco Debug命令參考文檔。

特權EXEC命令 **debug** 的輸出可提供診斷資訊，其中包括與協定狀態和一般網路活動相關的各種互聯事件。

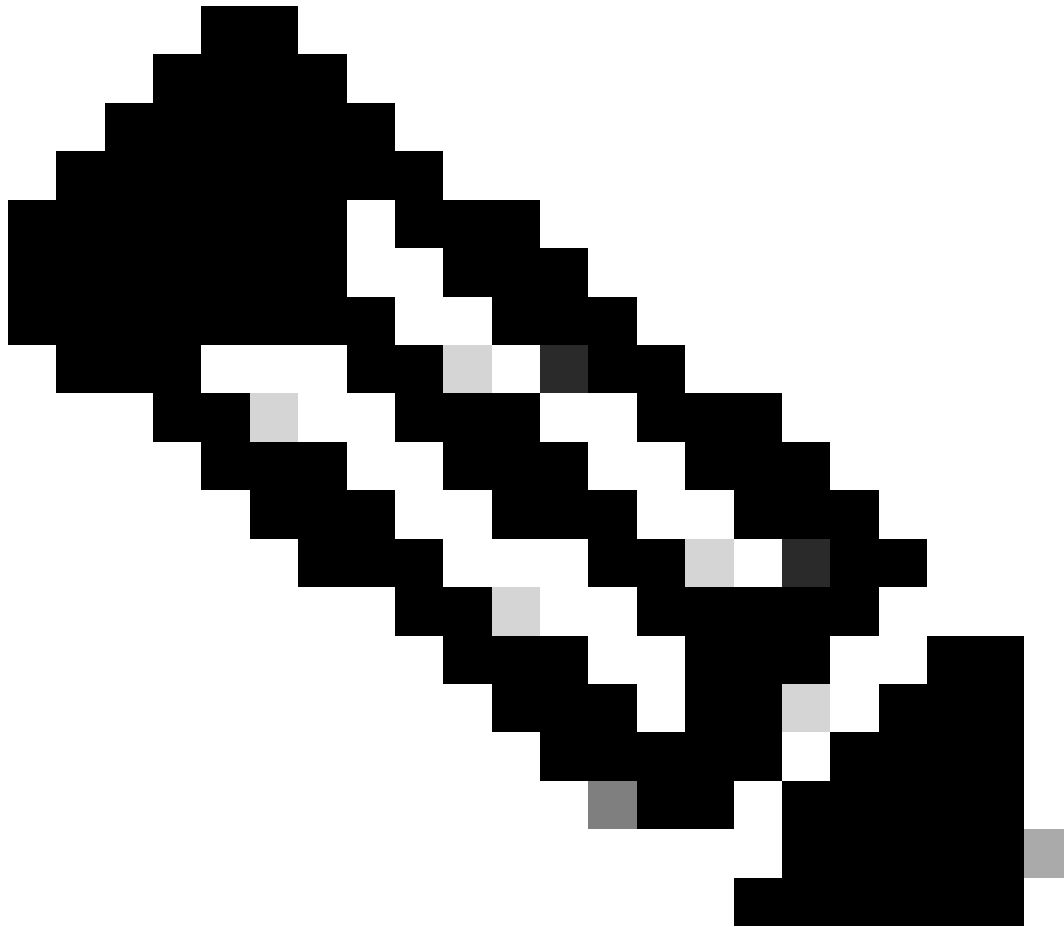
警告

debug 請謹慎使用命令。一般而言，在排除特定故障時，建議僅在路由器技術支援代表的指導下使用這些命令。

當網際網路遇到高負載情況時，啟用調試可能會中斷路由器的運行。因此，如果啟用了日誌記錄，當控制檯埠因日誌消息過載時，訪問伺服器可能會間歇性地凍結。

在啟動 `debug` 命令之前，請始終考慮此命令可生成的輸出以及它可花費的時間。例如，如果您的路由器具有一個基本速率介面 (BRI)，則 `debug isdn q931` 可能不會對系統產生危害。但是，對具有完整E1配置的AS5800執行相同的調試可能會生成大量輸入，以致於它會掛起並停止響應。

在調試之前，請使用 `show processes cpu` 命令檢視您的CPU負載。在開始調試之前，請確認您有充足的CPU可用。有關如何處理高CPU負載的詳細資訊，請參閱[對Cisco路由器上的高CPU利用率進行故障排除](#)。例如，如果您的Cisco 7200路由器的ATM介面正在進行橋接，則根據配置的子介面數量，重新啟動路由器會佔用大量CPU。原因是，對於每個虛電路 (VC)，都需要生成網橋協定資料單元 (BPDU) 資料包。在如此關鍵的時間啟動調試會導致CPU利用率大幅上升，並導致掛起或網路連線丟失。



注意：當調試運行時，通常看不到路由器提示，尤其當調試密集時。但是，在大多數情況下，您可以使用 `no debug all` 或 `undebug all` 命令來停止調試。有關安全使用調試的詳細資訊，請參閱獲取調試輸出部分。

偵錯之前

除了上述幾點之外，請務必了解調試對平台穩定性的影響。您還必須考慮必須連線到路由器上的哪個介面。本節有一些指導原則。

獲取調試輸出

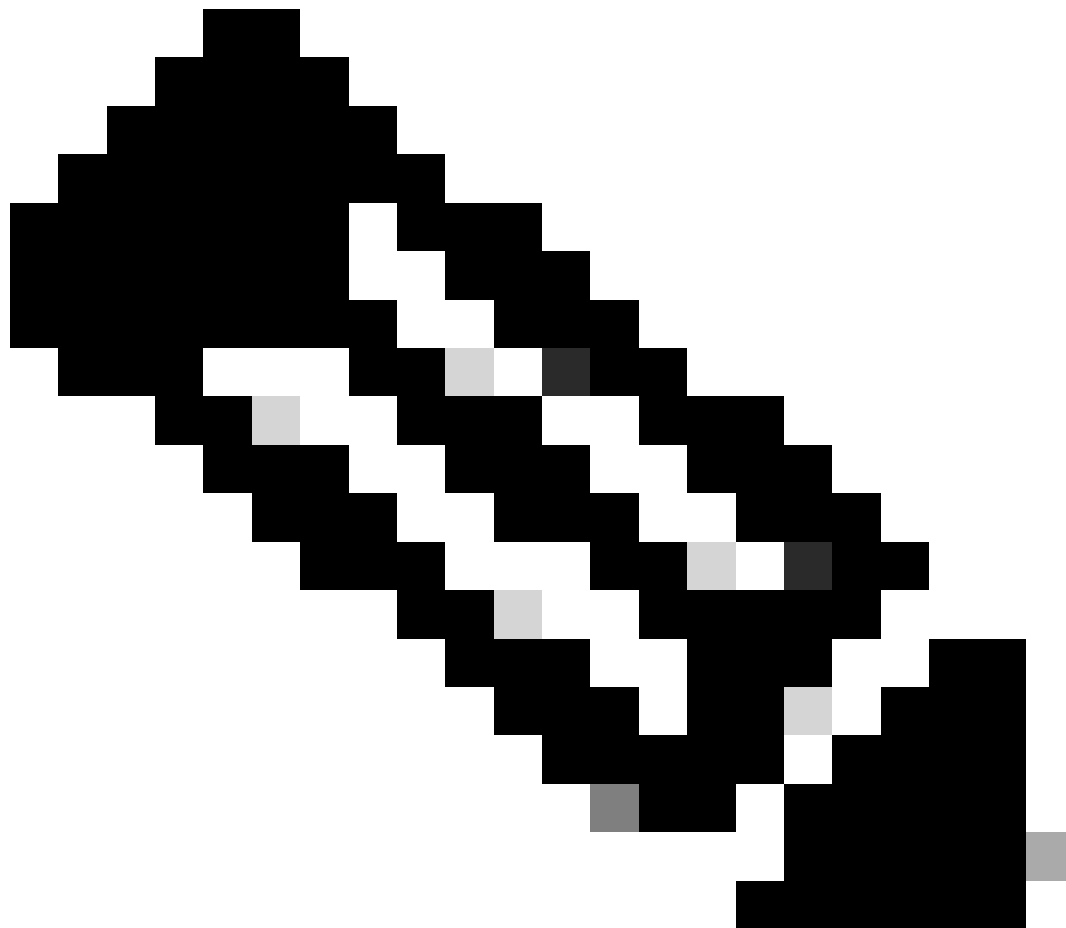
路由器可以顯示到各種介面的調試輸出，包括控制檯、aux和vty埠。路由器還可以將消息記錄到外部unix syslog伺服器的內部緩衝區。下面將討論每種方法的說明和警告：

控制檯埠

如果您已在控制檯上連線，則在正常配置下，不需要執行任何額外工作。必須自動顯示調試輸出。但是，請確保 **logging console level** 已根據需要進行設定，並且未使用 **no logging console** 命令停用日誌記錄。



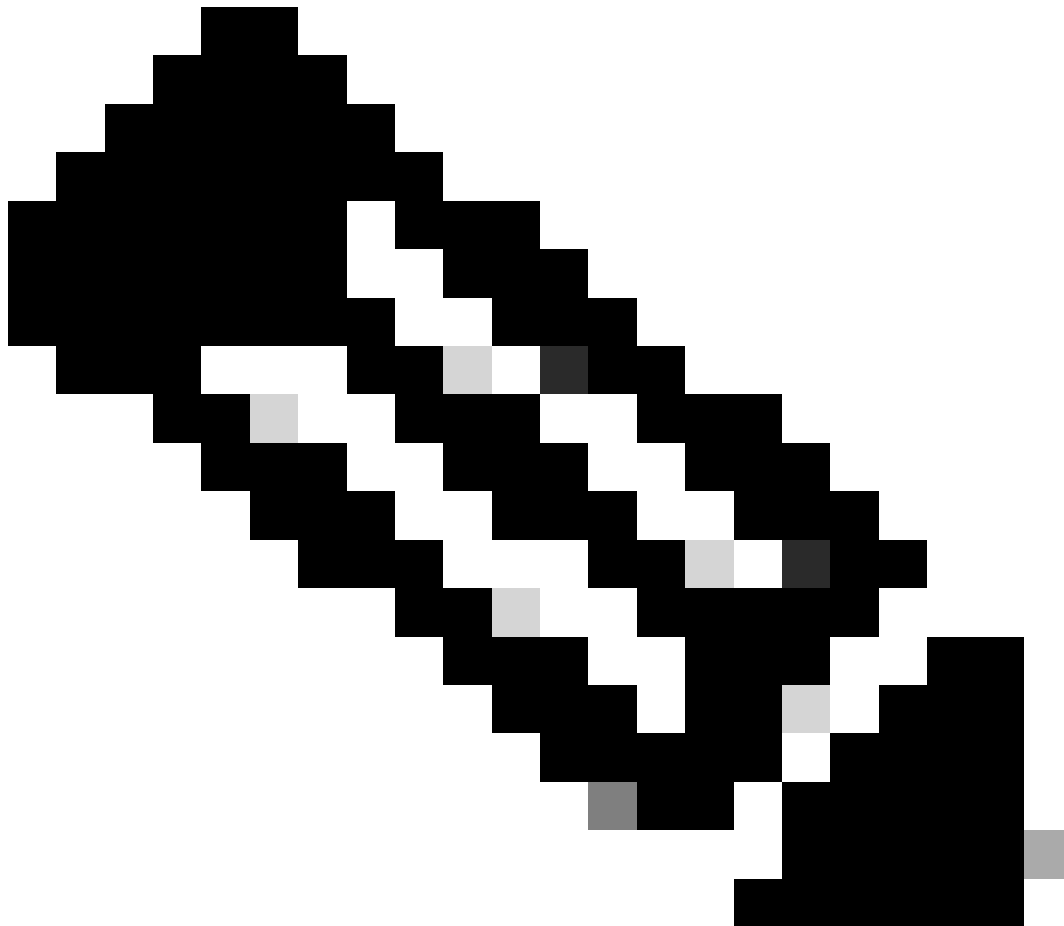
警告：對路由器的控制檯埠進行過度調試會導致其掛起。這是因為路由器會自動將主控台輸出的優先順序排在其他路由器功能之前。因此，如果路由器正在處理到控制檯埠的大量調試輸出，它可能會掛起。因此，如果調試輸出過多，請使用 vty (telnet)埠或日誌緩衝區來獲取調試。下面提供了更多資訊。



注意：預設情況下，在控制檯埠上啟用日誌記錄。因此，即使實際上您正在使用其他某種埠或方法（如Aux、vty或緩衝區）來捕獲輸出，控制檯埠也始終會處理debug輸出。因此，Cisco建議您在正常操作條件下，始終啟用no logging console命令，並使用其他方法捕獲調試。在需要使用控制檯的情況下，暫時重新打開日誌記錄控制檯。

埠

如果透過輔助埠連線，請鍵入 `terminal monitor` 命令。此外，請驗證路由器上是否尚未啟用 `no logging on` 命令。



註：如果使用Aux埠監控路由器，請記住，當路由器重新啟動時，Aux埠不顯示引導序列輸出。連線到控制檯埠以檢視引導順序。

VTY埠

如果透過輔助埠或telnet進行連線，請鍵入 `terminal monitor` 命令。另請確認 `no logging on` 命令未被使用。

將消息記錄到內部緩衝區

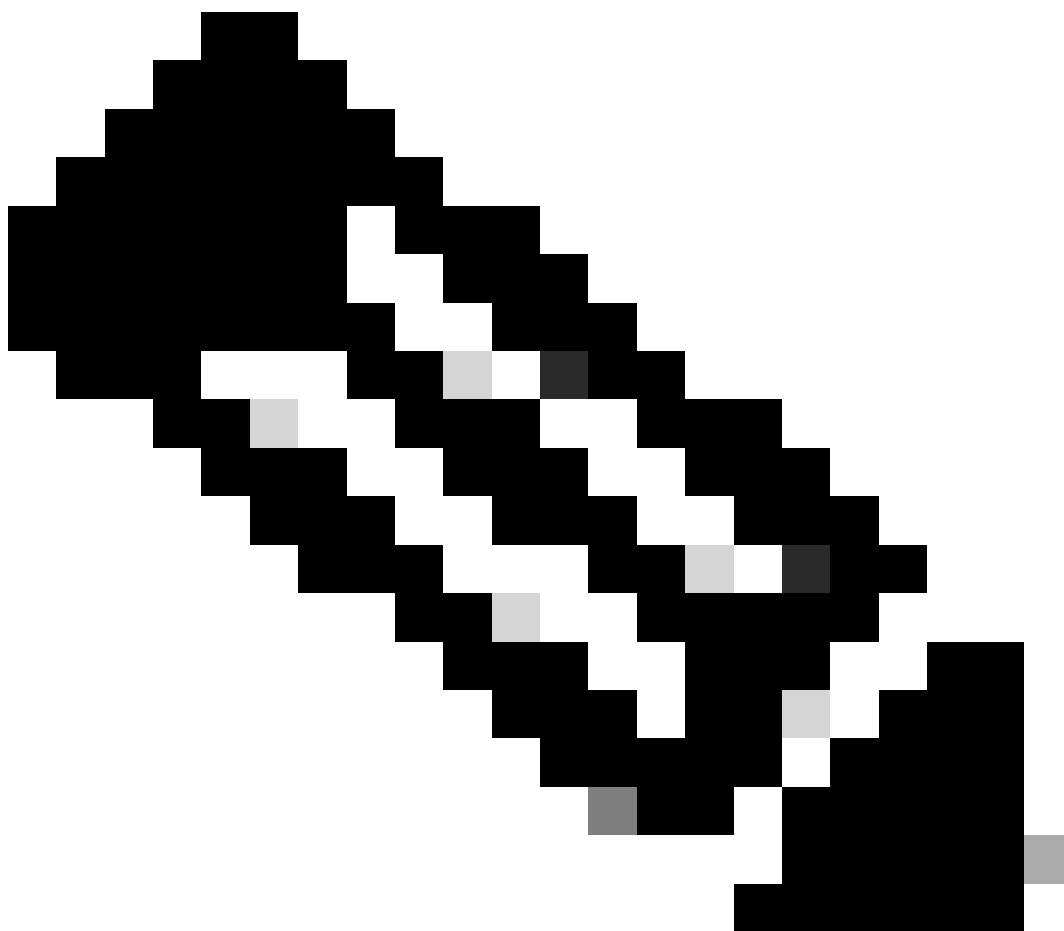
預設日誌記錄裝置是控制檯；除非另有指定，否則所有消息都會顯示在控制檯上。

要將消息記錄到內部緩衝區，請使用 `logging buffered` 路由器配置命令。以下是此命令的完整語法：

```
<#root>
```

```
logging buffered no logging buffered
```

`logging buffered` 命令將日誌消息複製到內部緩衝區，而不是寫入控制檯。緩衝區本質上是循環的，因此較新的消息會覆蓋較舊的消息。要顯示緩衝區中所記錄的消息，請使用特權EXEC命令 `show logging`。第一個顯示的訊息是緩衝區中最舊的訊息。您可以指定緩衝區的大小以及要記錄的消息的嚴重性級別。



註：在輸入緩衝區大小之前，請確保框中具有足夠的可用記憶體。使用Cisco IOS show proc mem 命令檢視可用的記憶體。

no logging buffered 此命令會取消緩衝區的使用，並將訊息寫入主控台（預設值）。

將消息記錄到UNIX Syslog伺服器

要將消息記錄到系統日誌伺服器主機，請使用logging router configuration命令。此命令的完整語法如下：

<#root>

```
logging <ip-address> no logging <ip-address>
```

logging該命令標識要接收日誌記錄消息的系統日誌伺服器主機。<ip-address>引數是主機的IP地址。透過多次發出此命令，您可以構建接收日誌記錄消息的系統日誌伺服器清單。

no logging 該命令從系統日誌清單中刪除具有指定地址的系統日誌伺服器。

其他調試前任務

-

設定終端模擬器軟體（例如HyperTerminal），以便其能夠將調試輸出捕獲到檔案中。例如，在HyperTerminal中，按一下 **Transfer**，然後按一下 **Capture Text**，然後選擇適當的選項。有關詳細資訊，請參閱[捕獲超級終端中的文本輸出](#)。有關其他終端模擬器軟體，請參閱軟體文檔。

-

使用 **service timestamps** 命令啟用毫秒(msec)時間戳：

<#root>

```
router(config)#
```

```
service timestamps debug datetime msec
```

```
router(config)#
```

```
service timestamps log datetime msec
```

這些命令以MMM DD HH : MM : SS格式將時間戳增加到調試中，根據系統時鐘指示日期和時間。如果未設定系統時鐘，日期和時間前面會加上星號(*)，表示日期和時間可能不正確。

通常建議配置毫秒時間戳，因為這樣在檢視調試輸出時具有高度的清晰度。毫秒時間戳可以更好地指示各種調試事件的相對計時。但是，請注意，當控制檯埠輸出大量消息時，它們無法與事件的實際時間相關聯。例如，如果在包含200條VC的框上啟 **debug x25** 用 **all**，並且輸出記錄到緩衝區(使用 **no logging console logging buffered** theandcommand)，則調試輸出中顯示的時間戳（在緩衝區內）不能是資料包透過介面的確切時間。因此，不要使用msec時間戳來證明效能問題，而是要獲取有關事件發生時間的相關資訊。

停止除錯

若要停止調試，請使用**no debug all**undebg all命令。使用命令**show debug**，驗證是否已經關閉調試。

請記住，**no logging console terminal no monitor** 命令只能分別防止輸出在控制檯、Aux或vty上輸出。它不會停止調試，因此會耗盡路由器資源。

使用debug ip packet命令

debug ip packet 該命令可生成有關資料包的資訊，這些資料包未經路由器進行快速交換。但是，由於它為每個資料包生成一個輸出，因此輸出可能非常大，從而導致路由器掛起。因此，如本節所述，只 **debug ip packet** 會在最嚴格的控制下使用。

限制輸出的最佳方式 **debug ip packet** 是建立與偵錯連結的存取清單。只有符合訪問清單標準的資料包才能使用 **debug ip packet**。此訪問清單不需要應用於任何介面，而是應用於調試操作。

在使用 **debugging ip packet** 之前，請注意路由器在預設情況下執行快速交換，或者如果配置為執行CEF交換，則路由器可以執行CEF交換。這意味著，一旦這些技術到位，資料包就不會提供給處理器，因此調試不會顯示任何內容。若要使用此功能，您需要使用 **no ip route-cache**（針對單點傳播封包）或 **no ip mroute-cache**（針對多點傳播封包）停用路由器上的快速交換功能。這必須應用到流量應該流動的介面上。使用 **show ip route** 命令進行檢驗。

警告

.

在處理大量資料包的路由器上停用快速交換可能會導致CPU利用率激增，從而導致該盒掛起或失去與其對等體的連線。

-

請勿在運行多協定標籤交換(MPLS)的路由器上停用快速交換。MPLS與CEF一起使用。因此，在介面上停用快速交換可能會造成災難性後果。

請考慮以下示例情景：



在router_122上配置的訪問清單為：

```
<#root>
```

```
access-list 105 permit icmp host 10.10.10.2 host 10.1.1.1 access-list 105 permit icmp host 10.1.1.1 host
```

此存取清單允許從主機router_121 (IP位址為10.10.10.2) 到主機router_123 (IP位址為10.1.1.1) 的任何網際網路控制訊息通訊協定(ICMP)封包，以及允許向另一個方向傳達。請務必允許兩個方向的資料包，否則路由器可能會丟棄返回的ICMP資料包。

刪除router_122上僅一個介面上的快速交換。這意味著，您只能看到發往該介面的資料包的調試，從攔截資料包的Cisco IOS的角度即可看到。從調試中，此類資料包以「d=」顯示。因為您尚未關閉另一個介面上的快速交換，所以返回的資料包不受 **debug ip packet**。此輸出顯示如何停用快速交換：

```
<#root>
```

```
router_122(config)#  
interface virtual-template 1  
  router_122(config-if)#  
no ip route-cache  
  router_122(config-if)#  
end
```

現在必須使用以前定義的訪問清單(access-list 105)啟 **debug ip packet** 動。

```
<#root>
```

```
router_122#
```

```
debug ip packet
```

```
detail 105 IP packet debugging is on (detailed) for access list 105 router_122# 00:10:01: IP: s=10.1.1
```

現在刪除另一個介面 (router_122上) 上的快速交換。這表示經過這兩個介面的所有封包現在都已進行封包交換(這是 **debug ip packet**

```
<#root>
```

```
router_122(config)#
```

```
interface serial 3/0
```

```
router_122(config-if)#
```

```
no ip route-cache
```

```
router_122(config-if)#
```

```
end
```

```
router_122# 00:11:57: IP:
```

```
s=10.10.10.2
```

```
(Virtual-Access1),
```

```
d=10.1.1.1
```

```
(Serial3/0), g=172.16.1.6, len 100, forward 00:11:57:
```

```
ICMP type=8
```

```
, code=0 ! -- ICMP packet (echo) from 10.10.10.2 to 10.1.1.1 00:11:57: IP:
```

```
s=10.1.1.1
```

```
(Serial3/0),
```

```
d=10.10.10.2
```

```
(Virtual-Access1), g=10.10.10.2, len 100, forward 00:11:57:
```

```
ICMP type=0
```

```
, code=0 ! -- ICMP return packet (echo-reply) from 10.1.1.1 to 10.10.10.2 00:11:57: IP: s=10.10.10.2
```

請注意，debug ip packet輸出不會顯示任何與訪問清單條件不匹配的資料包。有關此過程的其他資訊，請參閱[瞭解Ping和Traceroute命令](#)。

有關如何構建訪問清單的詳細資訊，請參閱[標準IP訪問清單日誌記錄](#)。

有條件觸發的調試

啟用條件觸發調試功能時，路由器為進入或離開指定介面上的路由器的資料包生成調試消息；路由器不為透過不同介面進入或離開路由器的資料包生成調試輸出。

請看條件式偵錯的簡易實作。請考慮以下場景：下面顯示的路由器(trabol)有兩個介面 (serial 0和serial 3)，兩個介面均運行HDLC封裝。

您可以使用 **debug serial interface** 正常命令來觀察所有介面上所接收的HDLC keepalive。您可以觀察兩個介面上的keepalive。

```
<#root>
```

```
traxbol#
```

```
debug serial interface
```

```
Serial network interface debugging is on traxbol# *Mar 8 09:42:34.851:
```

```
Serial0: HDLC
```

```
myseq 28, mineseen 28*, yourseen 41, line up ! -- HDLC keepalive on interface Serial 0 *Mar 8 09:42:
```

```
Serial3: HDLC
```

```
myseq 26, mineseen 26*, yourseen 27, line up ! -- HDLC keepalive on interface Serial 3 *Mar 8 09:42:
```

啟用介面serial 3的條件調試。這意味著只顯示介面serial 3的調試。使用 **debug interface <interface_type interface_number >**命令。

```
<#root>
```

```
traxbol#
```

```
debug interface serial 3
```

```
Condition 1 set
```

使用 **show debug condition** 命令驗證條件調試是活動的。請注意，介面serial 3的條件處於活動狀態。

```
<#root>
```

```
traxbol#
```

```
show debug condition
```

```
Condition 1: interface Se3 (1 flags triggered) Flags: Se3 traxbol#
```

請注意，現在僅顯示介面serial 3的調試

```
<#root>
```

```
*Mar 8 09:43:04.855:
```

```
Serial3: HDLC
```

```
myseq 29, mineseen 29*, yourseen 30, line up *Mar 8 09:43:14.855:
```

```
Serial3: HDLC
```

```
myseq 30, mineseen 30*, yourseen 31, line up
```

請使用 `undebug interface <interface_type interface_number>` 命令刪除條件調試。建議您先關閉調試（例如，使用 `undebug all`），然後再刪除條件觸發器。這是為了避免移除條件時產生大量偵錯輸出。

```
<#root>
```

```
traxbol#
```

```
undebug interface serial 3
```

```
This condition is the last interface condition set. Removing all conditions can cause a flood of debug  
y
```

```
Condition 1 has been removed traxbol
```

現在您可以看到同時顯示介面serial 0和serial 3的調試。

```
<#root>
```

```
*Mar 8 09:43:34.927:
```

```
Serial3: HDLC
```

```
myseq 32, mineseen 32*, yourseen 33, line up *Mar 8 09:43:44.923:
```

```
Serial0: HDLC
```

```
myseq 35, mineseen 35*, yourseen 48, line up
```



警告：某些除錯作業本身是有條件的。例如atm調試。使用ATM調試時，您必須明確指定必須為其啟用調試的介面，而不是在所有atm介面上啟用調試並指定條件。

本節介紹將ATM資料包調試限制在一個子介面的正確方法：

```
<#root>
```

```
arielle-nrp2#
```

```
debug atm packet interface atm 0/0/0.1
```

```
!--- Note that you explicitly specify the sub-interface to be used for debugging ATM packets debuggi
```

```
Displaying packets on interface ATM0/0/0.1 only
```

```
arielle-nrp2# *Dec 21 10:16:51.891: ATM0/0/0.1(O): VCD:0x1 VPI:0x1 VCI:0x21 DM:0x100 SAP:AAAA CTL:03 O
```

如果嘗試啟 **atm debugging** 用所有介面（在應用條件的情況下），則當路由器具大量ATM子介面時，路由器可能會掛起。下面是 atm 調試錯誤方法的示例。

在這種情況下，您可以看到應用了某個條件，但您還會看到此操作不起作用。您仍然可以看到來自另一個介面的資料包。在本實驗場景中，您只有兩個介面，流量非常少。如果介面的數量很高，則所有介面的調試輸出都非常高，這可能會導致路由器掛起。

```
<#root>
```

```
arielle-nrp2#
```

```
show debugging condition
```

```
Condition 1: interface ATM0/0/0.1
```

```
(1 flags triggered) Flags: ATM0/0/0.1 ! -- A condition for a specific interface. arielle-nrp2#  
debug atm packet
```

```
ATM packets debugging is on Displaying all ATM packets arielle-nrp2# *Dec 21 10:22:06.727:
```

```
ATM0/0/0.2
```

```
(O): ! -- You see debugs from interface ATM0/0/0.2, even though the condition ! -- specified ONLY ATM0/0/0.1
```

```
ATM0/0/0.1
```

```
(O): !--- You also see debugs for interface ATM0/0/0.1 as you wanted. VCD:0x1 VPI:0x1 VCI:0x21 DM:0x100
```

相關資訊

- [撥號和接入技術支援](#)
- [思科技術支援與下載](#)

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。