



条件付きデバッグ、ラジオアクティブトレース、およびパケットトレース

- [条件付きデバッグの概要 \(1 ページ\)](#)
- [ラジオアクティブトレースの概要 \(2 ページ\)](#)
- [条件付きデバッグおよび放射線トレース \(3 ページ\)](#)
- [トレースファイルの場所 \(3 ページ\)](#)
- [条件付きデバッグの設定 \(3 ページ\)](#)
- [L2 マルチキャストの放射線トレース \(6 ページ\)](#)
- [トレースファイルの推奨ワークフロー \(6 ページ\)](#)
- [ボックス外へのトレースファイルのコピー \(6 ページ\)](#)
- [条件付きデバッグの設定例 \(7 ページ\)](#)
- [条件付きデバッグの確認 \(8 ページ\)](#)
- [例：SISF のラジオアクティブトレースログの確認 \(8 ページ\)](#)
- [パケットトレースについて \(9 ページ\)](#)
- [条件付きデバッグパケットトレースの設定 \(10 ページ\)](#)
- [AP ごとの条件付きデバッグパケットトレースの設定 \(11 ページ\)](#)
- [クライアントごとの条件付きデバッグパケットトレースの設定 \(12 ページ\)](#)
- [条件付きデバッグパケットトレースの設定の確認 \(12 ページ\)](#)

条件付きデバッグの概要

条件付きデバッグ機能によって、定義した条件に基づき、特定の機能のデバッグおよびロギングを選択して有効にすることができます。この機能は、多くの機能がサポートされているシステムで有用です。

条件付きデバッグでは、多数の機能が導入されていて大規模に稼働しているネットワークにおけるきめ細かなデバッグが可能です。これにより、システム内の細かなインスタンスに対しても、詳細なデバッグを実行できます。これは、何千ものセッションのうち特定のセッションのみをデバッグするような場合に、非常に有用です。条件は複数指定することもできます。

条件とは、機能またはアイデンティティをいいます。アイデンティティは、インターフェイス、IP アドレス、MAC アドレスなどです。

これは、処理する機能オブジェクトを区別せずに出力を生成する、一般的なデバッグコマンドとは対照的です。一般的なデバッグ コマンドは、多数のシステム リソースを消費し、システム パフォーマンスに影響します。

ラジオアクティブトレースの概要

ラジオアクティブトレース (RA) により、冗長性のレベルを高めた状態で、システムの全体にわたって目的とする動作を連鎖的に実行できます。また、複数のスレッド、プロセス、および関数呼び出しにわたって、デバッグ情報を条件に基づいて (DEBUG レベルまで、または指定のレベルまで) 出力する方法を提供します。



- (注)
- ラジオアクティブトレースではファーストホップセキュリティ (FHS) がサポートされています。
ファーストホップセキュリティ機能の詳細については、[System Management] > [Wireless Multicast] > [Information About Wireless Multicast] > [Information About IPv6 Snooping] を参照してください。
 - 証明書が有効でない場合、ラジオアクティブトレース フィルタは機能しません。
 - メッシュ機能の問題を効果的にデバッグできるようにするため、ログの収集時に、イーサネットアドレスと無線 MAC アドレスの両方を RA トレースの条件付き MAC として追加してください。
 - ワイヤレス IP のデバッグを有効にするには、**debug platform condition feature wireless ip ip-address** コマンドを使用します。

表 1: 無線アクティブトレースをサポートするコンポーネント

コンポーネント	詳細
SISF または FHS	ファーストホップセキュリティ機能。Pv6 アドレス収集と IPv6 デバイストラッキングを含みます。詳細については、「IPv6 スヌーピングに関する情報」を参照してください。
LISP	Locator/ID Separation Protocol。

条件付きデバッグおよび放射線トレース

条件付きデバッグと組み合わせた放射線トレースによって、条件に関連するすべての実行コンテキストをデバッグする単一のデバッグ CLI を取得できます。これは、ボックス内の機能のさまざまな制御フロー プロセスを認識していなくても行うことができ、これらのプロセスでデバッグを個別に発行する必要もありません。

トレースファイルの場所

デフォルトでは、トレースファイルログは各プロセスで生成され、`/tmp/rp/trace` または `/tmp/fp/trace` ディレクトリに保存されます。この一時ディレクトリで、トレース ログがファイルに書き込まれます。各ファイルは 1 MB サイズです。これらのログ（プロセス単位）は `show platform software trace message process_name chassis active R0` コマンドを使用して確認できます。このディレクトリでは、特定のプロセスのこうしたファイルを、最大 25 件保持できます。`/tmp` ディレクトリのトレースファイルが自身の 1 MB 制限またはブート時に設定されたサイズに達すると、ローテーションから外れ、`tracelogs` ディレクトリ配下の `/crashinfo` パーティションにあるアーカイブの場所に移動します。

`/tmp` ディレクトリが 1 つのプロセスで保持するトレースファイルは 1 つのみです。ファイルが自身のファイルサイズ制限に達すると、ローテーションから外れ、`/crashinfo/tracelogs` に移動します。アーカイブ ディレクトリに蓄積されるファイルは最大 25 ファイルであり、その後は最も古いものから順に、`/tmp` から新たにローテーションされたファイルに置換されます。

`crashinfo` ディレクトリ内のトレースファイルは次の形式で配置されます。

1. `Process-name_Process-ID_running-counter.timestamp.gz`
例 : `IOSRP_R0-0.bin_0.14239.20151101234827.gz`
2. `Process-name_pmanlog_Process-ID_running-counter.timestamp.bin.gz`
例 : `wncmgrd_R0-0.27958_1.20180902081532.bin.gz`

条件付きデバッグの設定

条件付きデバッグを設定するには、次の手順に従います。

手順

	コマンドまたはアクション	目的
ステップ 1	<code>enable</code> 例 :	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
	Device> enable	
ステップ 2	debug platform condition mac <i>{mac-address}</i> 例： Device# debug platform condition mac bc16.6509.3314	ワイヤレス フローで指定された MAC アドレスの条件付きデバッグを設定します。
ステップ 3	debug platform condition feature wireless mac <i>{mac-address}</i> 例： Device# debug platform condition feature wireless mac b838.61a1.5433	指定された MAC アドレスを使用する機能の条件付きデバッグを設定します。
ステップ 4	debug platform condition start 例： Device# debug platform condition start	条件付きデバッグを開始します（上記のいずれかの条件に一致すると放射線トレースを開始します）。
ステップ 5	show platform condition または show debug 例： Device# show platform condition Device# show debug	現在設定されている条件を表示します。
ステップ 6	debug platform condition stop 例： Device# debug platform condition stop	条件付きデバッグを停止します（放射線トレースを停止します）。
ステップ 7	request platform software trace archive [last <i>{number}</i> days] [target <i>{crashinfo: flashinfo:}</i>] 例： Device# request platform software trace archive last 2 days	（任意）システムのマージされたトレースファイルの履歴ログを表示します。日数またはロケーションの組み合わせのフィルタ。
ステップ 8	request platform software trace filter-binary <i>{wire wireless}</i> [context <i>{mac-address}</i>] [level module] 例： Device# request platform software trace filter-binary wireless context bc16.6509.3314	（任意）指定された MAC アドレスのコンテキストと情報（ネットワークまたはワイヤレス）を照合するには、モジュールをフィルタリングします。これらのログはオフラインで確認できます。

	コマンドまたはアクション	目的
ステップ 9	show platform software trace filter-binary wireless 例： <pre>Device# show platform software trace filter-binary wireless</pre>	(任意) すべての機能によって生成された統合ログを表示します。
ステップ 10	show platform software trace [counter filter-binary level message] 例： <pre>Device# show platform software trace message</pre>	(任意) 最新のトレースファイルからマージされたログを表示します。アプリケーションの状態、トレースモジュール名およびトレースレベルをさまざまな組み合わせでフィルタリングします。 (注) このコマンドは、関係のあるすべてのプロセスから生成されるすべてのログファイルを順に並べて1つのログ出力にします。
ステップ 11	clear platform condition all 例： <pre>Device# clear platform condition all</pre>	すべての条件をクリアします。

次のタスク



(注) **request platform software trace filter-binary** コマンドと **show platform software trace filter-binary** コマンドは同様の動作をします。唯一の違いは次のとおりです。

- **request platform software trace filter-binary** : データ ソースとして履歴ログを使用します。
- **show platform software trace filter-binary** : データ ソースとしてフラッシュの一時ディレクトリを使用します。



(注) コマンド **request platform software trace filter-binary wireless {mac-address}** は次の3つのフラッシュファイルを生成します。

- *collated_log <.date.>*
- *mac_log <..date.>*
- *mac_database ..file*

その中でも、`mac_log <..date.>` は、デバッグする MAC 用のメッセージを伝えるため、最も重要なファイルです。コマンド `show platform software trace filter-binary` も同じフラッシュ ファイルを生成し、また、画面に `mac_log` を出力します。

L2 マルチキャストの放射線トレース

特定のマルチキャスト受信者を特定するには、参加者または受信側クライアントの MAC アドレス、グループのマルチキャスト IP アドレスおよびスヌーピング VLAN を指定します。また、デバッグのトレース レベルを有効にします。デバッグ レベルでは、詳細なトレースとシステムへの高い可視性が提供されます。

```
debug platform condition feature multicast controlplane mac client MAC address ip Group
IP address vlan id level debug level
```

トレース ファイルの推奨ワークフロー

トレース ファイルの推奨ワークフローの概要は次のとおりです。

1. 特定の時間帯のトレースログを要求する場合。
たとえば 1 日。
使用するコマンドは、次のとおりです。
`Device#request platform software trace archive last 1 day`
2. システムは、`/flash:` ロケーション内のトレースログの tar ball (.gz ファイル) を生成します。
3. スイッチデバイスの外にファイルをコピーします。ファイルをコピーすることによって、オフラインでトレースログが使用できます。ファイルのコピーについての詳細は、次のセクションを参照してください。
4. `/flash: location` からトレースログファイル (.gz) ファイルを削除します。これにより、他の操作に十分な領域がスイッチデバイスに確保されます。

ボックス外へのトレースファイルのコピー

トレース ファイルの例を以下に示します。

```
Device# dir crashinfo:/tracelogs
Directory of crashinfo:/tracelogs/

50664 -rwx 760 Sep 22 2015 11:12:21 +00:00 plogd_F0-0.bin_0.gz
50603 -rwx 991 Sep 22 2015 11:12:08 +00:00 fed_pmanlog_F0-0.bin_0.9558.20150922111208.gz
50610 -rw- 11 Nov 2 2015 00:15:59 +00:00 timestamp
50611 -rwx 1443 Sep 22 2015 11:11:31 +00:00
```

```

auto_upgrade_client_sh_pmanlog_R0-.bin_0.3817.20150922111130.gz
50669 -rwx 589 Sep 30 2015 03:59:04 +00:00 cfgwr-8021_R0-0.bin_0.gz
50612 -rwx 1136 Sep 22 2015 11:11:46 +00:00 reflector_803_R0-0.bin_0.1312.20150922111116.gz
50794 -rwx 4239 Nov 2 2015 00:04:32 +00:00 IOSRP_R0-0.bin_0.14239.20151101234827.gz
50615 -rwx 131072 Nov 2 2015 00:19:59 +00:00 linux_iosd_image_pmanlog_R0-0.bin_0

```

トレース ファイルは、次に示すさまざまなオプションのいずれかを使用して、コピーできます。

```

Device# copy crashinfo:/tracelogs ?
crashinfo: Copy to crashinfo: file system
flash: Copy to flash: file system
ftp: Copy to ftp: file system
http: Copy to http: file system
https: Copy to https: file system
null: Copy to null: file system
nvram: Copy to nvram: file system
rcp: Copy to rcp: file system
running-config Update (merge with) current system configuration
scp: Copy to scp: file system
startup-config Copy to startup configuration
syslog: Copy to syslog: file system
system: Copy to system: file system
tftp: Copy to tftp: file system
tmpsys: Copy to tmpsys: file system

```

TFTP サーバにコピーするための一般的な構文は次のとおりです。

```

Device# copy source: tftp:
Device# copy crashinfo:/tracelogs/IOSRP_R0-0.bin_0.14239.20151101234827.gz tftp:
Address or name of remote host []? 2.2.2.2
Destination filename [IOSRP_R0-0.bin_0.14239.20151101234827.gz]?

```



(注) `tracelog` および他の目的に使用可能な空き容量があることを確認するために、生成されたレポート/アーカイブ ファイルをスイッチからクリアすることが重要です。

条件付きデバッグの設定例

次に、`show platform condition` コマンドの出力例を示します。

```

Device# show platform condition
Conditional Debug Global State: Stop
Conditions Direction
-----|-----
MAC Address 0024.D7C7.0054 N/A
Feature Condition Type Value
-----|-----
Device#

```

次に、`show debug` コマンドの出力例を示します。

```

Device# show debug
IOSXE Conditional Debug Configs:

```

```
Conditional Debug Global State: Start
Conditions Direction
```

```
-----|-----
MAC Address 0024.D7C7.0054 N/A
Feature Condition Type Value
```

```
-----|-----
Packet Infra debugs:
Ip Address Port
```

```
-----|-----
Device#
```

条件付きデバッグの確認

次の表に、条件付きデバッグの確認に使用できる各種コマンドを示します。

コマンド	目的
show platform condition	現在設定されている条件を表示します。
show debug	現在設定されているデバッグ条件を表示します。
show platform software trace filter-binary	最新のトレース ファイルからマージされたログを表示します。
request platform software trace filter-binary	システムにマージされたトレース ファイルの履歴ログを表示します。

例：SISF のラジオアクティブトレース ログの確認

次に、`show platform software trace message ios chassis active R0 | inc sif` コマンドの出力例を示します。

```
Device# show platform software trace message ios chassis active R0 | inc sif

2017/10/26 13:46:22.104 {IOSRP_R0-0}{1}: [parser]: [5437]: UUID: 0, ra: 0 (note): CMD:
'show platform software trace message ios switch active R0 | inc sif' 13:46:22 UTC Thu
Oct 26 2017
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sif]: [5437]: UUID: 4800000000060, ra: 7
(debug): FF8E802918 semaphore system unlocked
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sif]: [5437]: UUID: 4800000000060, ra: 7
(debug): Unlocking, count is now 0
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sif]: [5437]: UUID: 4800000000060, ra: 7
(debug): FF8E802918 semaphore system unlocked
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sif]: [5437]: UUID: 4800000000060, ra: 7
(debug): Unlocking, count is now 1
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sif]: [5437]: UUID: 4800000000060, ra: 7
(debug): Gi1/0/5 vlan 10 aaaa.bbbb.cccc Setting State to 2
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sif]: [5437]: UUID: 4800000000060, ra: 7
(debug): Gi1/0/5 vlan 10 aaaa.bbbb.cccc Start timer 0
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sif]: [5437]: UUID: 4800000000060, ra: 7
(debug): Gi1/0/5 vlan 10 aaaa.bbbb.cccc Timer value/granularity for 0 :299998/1000
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sif]: [5437]: UUID: 4800000000060, ra: 7
(debug): Gi1/0/5 vlan 10 aaaa.bbbb.cccc Updated Mac Timer : 299998
```

```
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sisf]: [5437]: UUID: 4800000000060, ra: 7
(debug): Gi1/0/5 vlan 10 aaaa.bbbb.cccc Before Timer : 350000
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sisf]: [5437]: UUID: 4800000000060, ra: 7
(debug): Gi1/0/5 vlan 10 aaaa.bbbb.cccc Timer 0, default value is 350000
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sisf]: [5437]: UUID: 4800000000060, ra: 7
(debug): Allocating timer wheel for 0
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sisf]: [5437]: UUID: 4800000000060, ra: 7
(debug): Gi1/0/5 vlan 10 aaaa.bbbb.cccc No timer running
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sisf]: [5437]: UUID: 4800000000060, ra: 7
(debug): Granularity for timer MAC_T1 is 1000
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sisf]: [5437]: UUID: 4800000000060, ra: 7
(debug): Gi1/0/5 vlan 10 aaaa.bbbb.cccc Current State :MAC-STALE, Req Timer : MAC_T1
Current Timer MAC_T1
```

パケットトレースについて

パケットトレース機能は、クラウドソフトウェアの Cisco Catalyst 9800 シリーズ ワイヤレスコントローラのデータプレーンパケットトレースを実行する方法について、その詳細を網羅するものです。

この機能では、次の問題が特定されます。

- 設定ミス
- キャパシティの過負荷
- トラブルシューティング中のソフトウェアのバグ

この機能により、システム内のパケットに何が生じているかが特定されます。条件付きデバッグパケットトレース機能は、ユーザ定義の条件に関するパケット単位の処理の詳細をアカウントリングおよびキャプチャするために使用されます。

次の手順を使用して、コントローラ上のパケットをトレースできます。

1. コントローラ上でトレースする必要がある、選択したパケットまたはトラフィックについて、条件付きデバッグを有効にします。
2. パケットトレースを有効にします (AP 単位またはクライアント単位)。



- (注) AP とコントローラが同じ VLAN 内にある場合は、MAC アドレスをフィルタとして使用する AP 単位の条件付きデバッグを使用する必要があります。これらが同じ VLAN 内でない場合、MAC アドレスを使用する AP 単位のパケットトレースでは、MAC アドレスが変化するため、パケットがキャプチャされません。

条件付きデバッグパケットトレースの制限事項

MAC または IP フィルタは外部イーサネットまたは IP ヘッダーにのみ適用されるため、パケットが CAPWAP でカプセル化されている場合、MAC または IP は内部 802.11 MAC または IP には適用されません。

条件付きデバッグパケットトレースの設定

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	debug platform packet-trace packet packet-count circular fia-trace data-size data-size 例： Device# debug platform packet-trace packet 8192 circular fia-trace data-size 2048	最後のパケットセットをキャプチャするようにパケットトレースを設定します。 ここで、各変数は次のように定義されません。 packet-count ：有効な範囲は 16 ～ 8192 です。 data-size ：有効な範囲は 2048 ～ 16384 バイトです。
ステップ 3	debug platform packet-trace copy packet both size packet-size 例： Device# debug platform packet-trace copy packet both size 2048	パケットデータのコピーのパケットトレースを設定します。 ここで、各変数は次のように定義されません。 packet-size ：有効な範囲は 16 ～ 2048 バイトです。
ステップ 4	debug platform condition interface {intf-name cpp} {mac ipv4 match} {both ingress egress} 例： TenGigabitEthernet 0/0/0 の条件付きデバッグを有効にし、送信元と宛先の MAC が 0001.0001.0001 であるパケットを照合します。 Device# debug platform condition interface TenGigabitEthernet 0/0/0 mac 0001.0001.0001 both	インターフェイス、MAC、または IP フィルタの条件付きデバッグを有効にします。 ここでインターフェイスとは、任意の物理ポート、ポートチャネル、内部 vlan、SVI、またはワイヤレスクライアントを指します。
ステップ 5	debug platform condition start 例： Device# debug platform condition start	条件付きデバッグパケットトレースを開始します。

	コマンドまたはアクション	目的
ステップ 6	debug platform condition stop 例： Device# debug platform condition stop	条件付きデバッグパケットトレースを停止します。
ステップ 7	show platform hardware chassis active qfp feature packet-trace packet all redirect bootflash:packet_trace.txt 例： Device# show platform hardware chassis active qfp feature packet-trace packet all redirect bootflash:packet_trace.txt	トレースされたすべてのパケットをブートフラッシュにリダイレクトします。 packet_trace.txt を pcap に変換し、pcap ファイルをダウンロードします。この操作には次のリンクを使用できます。 http://www.win-dharon-dev.cisco.com/pactrac2pcap.html

AP ごとの条件付きデバッグパケットトレースの設定

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	debug platform condition interface {intf-name cpp} {mac [mac-address access-list acl-name] ipv4 match} {both ingress egress} 例： Device# debug platform condition interface TenGigabitEthernet 0/0/0 mac 0001.0001.0001 both Device# debug platform condition interface TenGigabitEthernet 0/0/0 mac access-list mac-acl-name both	MAC フィルタによる条件付きデバッグを有効にします。 ここでは、CLI は、送信元または宛先 MAC アドレスが 0001.0001.0001 であるパケットを照合します。
ステップ 3	debug platform condition interface TenGigabitEthernet intf-number match mac {H.H.H any host} {both ingress egress} 例： Device# debug platform condition interface TenGigabitEthernet 0/0/0 match mac 0001.0001.0001 both	インライン MAC MAC による条件付きデバッグを有効にします。

	コマンドまたはアクション	目的
ステップ 4	debug platform condition interface TenGigabitEthernet <i>intf-number</i> ipv4 {<i>A.B.C.D/nn</i> access-list <i>acl-name</i> both egress ingress} {both egress ingress} 例 : <pre>Device# debug platform condition interface TenGigabitEthernet 0/0/0 ipv4 192.168.1.2/32 both Device# debug platform condition interface TenGigabitEthernet 0/0/0 ipv4 access-list ip-acl-name both Device# debug platform condition interface TenGigabitEthernet 0/0/0 match ipv4 192.168.1.2/32 both</pre>	IP フィルタによる条件付きデバッグを有効にします。 ここで、各変数は次のように定義されます。 <i>intf-number</i> : GigabitEthernet インターフェイス番号。有効な範囲は 1 ~ 32 です。

クライアントごとの条件付きデバッグパケットトレースの設定

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例 : <pre>Device> enable</pre>	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	debug platform condition interface {<i>intf-name</i> cpp <i>cpp-handle-index</i>} {mac ipv4 match [<i>ipv4</i> <i>ipv6</i> <i>mac</i>]} {both ingress egress} 例 : <pre>Device# debug platform condition interface cpp 0xa0000001 match ipv4 protocol icmp host 192.168.1.100 host 192.168.1.1 both</pre>	ワイヤレス クライアント インターフェイスの条件付きデバッグを有効にします。 ここで、各変数は次のように定義されます。 <i>cpp-handle-index</i> : 有効な範囲は 1 ~ 4294967295 です。

条件付きデバッグパケットトレースの設定の確認

トレースされたパケットのサマリーを表示するには、次のコマンドを使用します。

```
Device# show platform packet-trace summary
```

特定のトレースされたパケットを表示するには、次のコマンドを使用します。

```
Device# show platform packet-trace packet packet-number
```

ワイヤレスクライアントインターフェイスハンドルを表示するには、次のコマンドを使用します。

```
Device# show platform hardware chassis active qfp feature wireless wlclient cpp-client
mac-address client-mac details
Device# show platform hardware chassis active qfp feature wireless wlclient cpp-client
mac-address 8825.93b0.b51f details
Client Details for client cpp_if_handle: 0x34
Name : WLCLIENT-IF-0x00a0000001
Mac Addr : 8825.93b0.b51f
pal_if_handle : 0xa0000001
Mobility State : LOCAL
Multicast Action : FORWARD
Auth State : RUN
```

