



# 条件付きデバッグとラジオアクティブトレース

- [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 \(2 ページ\)](#)
- [トレースファイルの場所 \(2 ページ\)](#)
- [条件付きデバッグの設定 \(GUI\) \(3 ページ\)](#)
- [条件付きデバッグの設定 \(4 ページ\)](#)
- [トレースファイルの推奨ワークフロー \(5 ページ\)](#)
- [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 \(7 ページ\)](#)
- [例：SISF のラジオアクティブトレースログの確認 \(7 ページ\)](#)

## 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。
  - 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。

## 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。



- (注)
- プラットフォームに適用されているデバッグ条件を削除するには、**clear platform condition all** コマンドを使用します。

## 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\_namechassis 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その 1 MB 制限またはブート時に設定されたサ

イズに達した場合、ローテーションから外れ、**tracelogs** ディレクトリの **/crashinfo** パーティションの下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。ファイルがそのファイルサイズの制限に達すると、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。ファイルサイズはプロセスに依存し、一部のプロセスではより大きなファイルサイズ (最大 10MB) が使用されます。同様に、**tracelogs** ディレクトリ内のファイル数もプロセスによって決定されます。たとえば、WNCNプロセスでは、プラットフォームに応じて、インスタンスごとに 400 ファイルの制限が使用されます。

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 (GUI)

### 手順

- 
- ステップ 1 [Troubleshooting] > [Radioactive Trace] を選択します。
  - ステップ 2 [Add] をクリックします。
  - ステップ 3 [MAC/IP Address] を入力します。MAC アドレスは、xx:xx:xx:xx:xx:xx、xx-xx-xx-xx-xx-xx、または xxxx.xxxx.xxxx のいずれかの形式で指定できます。
  - ステップ 4 [Apply to Device] をクリックします。
  - ステップ 5 条件付きデバッグを開始する場合は [Start]、停止する場合は [Stop] をクリックします。
  - ステップ 6 [Generate] をクリックして、放射線トレースログを作成します。
  - ステップ 7 オプションボタンをクリックして、時間間隔を設定します。
  - ステップ 8 トレースファイル名の横に表示される [Download Logs] アイコンをクリックして、ログをローカルフォルダにダウンロードします。
  - ステップ 9 トレースファイル名の横に表示される [View Logs] アイコンをクリックして、GUI ページでログファイルを表示します。[Load More] をクリックして、ログファイルの他の行を表示します。
  - ステップ 10 [Apply to Device] をクリックします。
-

## 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 | <b>debug platform condition feature wireless mac {mac-address}</b><br>例：<br>デバイス# <b>debug platform condition feature wireless mac b838.61a1.5433</b> | 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。<br>(注) これは、AP またはクライアント MAC/IP でサポートされ、CMX IP アドレスとモビリティピア IP でもサポートされます。          |
| ステップ 2 | <b>debug platform condition start</b><br>例：<br>デバイス# <b>debug platform condition start</b>  | 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）。<br>(注) これは、AP またはクライアント MAC/IP でサポートされ、CMX IP アドレスとモビリティピア IP でもサポートされます。 |
| ステップ 3 | <b>show platform condition</b> または <b>show debug</b><br>例：<br>デバイス# <b>show platform condition</b><br>デバイス# <b>show debug</b>                         | 現在設定されている条件を表示します。   |
| ステップ 4 | <b>debug platform condition stop</b><br>例：<br>デバイス# <b>debug platform condition stop</b>  | 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）。<br>(注) これは、AP またはクライアント MAC/IP でサポートされ、CMX IP アドレスとモビリティピア IP でもサポートされます。                 |
| ステップ 5 | <b>show logging profile wireless [counter   [last]{x days/hours}   filter mac{&lt;mac address&gt;} [to-file]{&lt;destination&gt;}</b>                 | 最新のワイヤレスプロファイルからのログを表示します。   |

|        | コマンドまたはアクション  | 目的   |
|--------|---|--|
|        | 例：<br>デバイス# <code>show logging profile wireless start last 20 minutes to-file bootflash:logs.txt</code>                       | (注) ログを収集するには、 <code>show logging profile wireless</code> コマンドまたは <code>show logging process</code> コマンドを使用できます。 |
| ステップ 6 | <b>show logging process &lt;process name&gt;</b><br>例：<br>デバイス# <code>show logging process wncd to-file flash:wncd.txt</code> | プロセスに固有のログコレクションを表示します。  |
| ステップ 7 | <b>clear platform condition all</b><br>例：<br>デバイス# <code>clear platform condition all</code>                                  | すべての条件をクリアします。   |

#### 次のタスク



(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..>` は最も重要なファイルで、デバッグする MAC 用のメッセージが含まれます。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。

## トレース ファイルの推奨ワークフロー

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。  
デバイス# `show logging process wncd to-file flash:wncd.txt`
2. ロケーション (/flash:) にトレースログのテキストファイルが生成されます。
3. 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. ロケーション (/flash:) からトレースログファイル (.txt) を削除します。これにより、他の操作に十分な領域が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。

```
デバイス# dir flash:/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。

```
デバイス# copy flash:/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 サーバーにコピーするための一般的な構文は次のとおりです。

```
デバイス# copy source: tftp:
デバイス# 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。

```
デバイス# show platform condition
Conditional Debug Global State: Stop
Conditions Direction
```

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

```
-----|-----
デバイス#
```

次に、`show debug` コマンドの出力例を示します。

```
デバイス# 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
```

```
-----|-----
デバイス#
```

## 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                               |
|--|----------------------------------|
| <code>show platform condition</code>                       | 現在設定されている条件を表示します。               |
| <code>show debug</code>                                    | 現在設定されているデバッグ条件を表示します。           |
| <code>show platform software trace filter-binary</code>    | 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。    |
| <code>request platform software trace filter-binary</code> | 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 sisf` コマンドの出力例を示します。

## 例 : SISF のラジオアクティブトレース ログの確認

```
デバイス# 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): Gil/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): Gil/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): Gil/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): Gil/0/5 vlan 10 aaaa.bbbb.cccc Updated Mac Timer : 299998
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sif]: [5437]: UUID: 4800000000060, ra: 7
(debug): Gil/0/5 vlan 10 aaaa.bbbb.cccc Before Timer : 350000
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sif]: [5437]: UUID: 4800000000060, ra: 7
(debug): Gil/0/5 vlan 10 aaaa.bbbb.cccc Timer 0, default value is 350000
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sif]: [5437]: UUID: 4800000000060, ra: 7
(debug): Allocating timer wheel for 0
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sif]: [5437]: UUID: 4800000000060, ra: 7
(debug): Gil/0/5 vlan 10 aaaa.bbbb.cccc No timer running
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sif]: [5437]: UUID: 4800000000060, ra: 7
(debug): Granularity for timer MAC_T1 is 1000
2017/10/26 13:46:10.667 {IOSRP_R0-0}{1}: [sif]: [5437]: UUID: 4800000000060, ra: 7
(debug): Gil/0/5 vlan 10 aaaa.bbbb.cccc Current State :MAC-STALE, Req Timer : MAC_T1
Current Timer MAC_T1
```