



# 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 \(2 ページ\)](#)
- [条件付きデバッグとラジオアクティブトレースの設定方法 \(2 ページ\)](#)
- [条件付きデバッグのモニタリング \(6 ページ\)](#)
- [条件付きデバッグの設定例 \(7 ページ\)](#)
- [条件付きデバッグとラジオアクティブトレースに関するその他の関連資料 \(7 ページ\)](#)
- [条件付きデバッグとラジオアクティブトレースの機能履歴と情報 \(8 ページ\)](#)

## 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。



(注) コントロールプレーントレースのみがサポートされています。

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。



(注) サポートされる条件は MAC アドレスであることのみです。

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) デフォルトのレベルは **DEBUG** です。ユーザは別のレベルに変更することはできません。

ラジオアクティブトレースでは、次の機能が有効になっています。

- IGMP スヌーピング
- レイヤ2 マルチキャスト

## 条件付きデバッグとラジオアクティブトレースの設定方法

### 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。

`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 : `wcm_pmanlog_R0-0.30360_0.20151028233007.bin.gz`

## 条件付きデバッグの設定

条件付デバッグを設定するには、以下の手順に従います。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>enable</b> 例 : <code>Device&gt; enable</code>	特権 EXEC モードを有効にします。  • パスワードを入力します (要求された場合)。
ステップ 2	<b>debug platform condition mac</b> <code>{mac-address}</code> 例 : <code>Device# debug platform condition mac bc16.6509.3314</code>	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	<b>debug platform condition start</b> 例 : <code>Device# debug platform condition start</code>	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 4	<b>show platform condition</b> または <b>show debug</b> 例 : <code>Device# show platform condition</code> <code>Device# show debug</code>	現在設定されている条件を表示します。
ステップ 5	<b>debug platform condition stop</b> 例 : <code>Device# debug platform condition stop</code>	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)。

	コマンドまたはアクション	目的
ステップ 6	<b>request platform software trace archive</b> <b>[last {number} days] [target {crashinfo:   flashinfo:}]</b>  例： <pre># request platform software trace archive last 2 days</pre>	(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 7	<b>show platform software trace [filter-binary   level   message]</b>  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。 <ul style="list-style-type: none"> <li>• <b>filter-binary</b> : 照合するモジュールをフィルタリングします。</li> <li>• <b>level</b> : トレース レベルを表示します。</li> <li>• <b>message</b> : トレース メッセージのリングの内容を表示します。</li> </ul> (注) デバイス上では次が可能です。 <ul style="list-style-type: none"> <li>• Linux シェルだけでなく、IOS のコンソールからも使用できます。</li> <li>• マージされたログでファイルを生成します。</li> <li>• ステージング エリアからのみマージされたログを表示します。</li> </ul>
ステップ 8	<b>clear platform condition all</b>  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。

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
--More-

```

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モニタリング

以下の表に、条件付きデバッグのモニタに使用できる各種コマンドを示します。

コマンド	目的
<b>show platform condition</b>	現在設定されている条件を表示します。
<b>show debug</b>	現在設定されているデバッグ条件を表示します。

コマンド	目的
<b>show platform software trace filter-binary</b>	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。
<b>request platform software trace filter-binary</b>	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。

## 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#
```

次に、*debug platform condition stop* コマンドの例を示します。

```
Device# debug platform condition stop
Conditional Debug Global State: Stop
```

## 条件付きデバッグとラジオアクティブトレースに関するその他の関連資料

### 関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	<i>Command Reference (Catalyst 9400 Series Switches)</i>

## 条件付きデバッグとラジオアクティブトレースの機能履歴と情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェアリリースでもサポートされます。

リリース	変更内容
Cisco IOS XE Everest 16.6.1	この機能が導入されました。