



debug コマンド

- [debug arp](#) (2 ページ)
- [debug ble](#) (2 ページ)
- [debug capwap client](#) (3 ページ)
- [debug capwap client avc](#) (4 ページ)
- [debug cdp](#) (5 ページ)
- [debug cleanair](#) (6 ページ)
- [debug dhcp](#) (7 ページ)
- [debug dot11 driver level](#) (7 ページ)
- [debug dot11 client data-path](#) (8 ページ)
- [debug dot11 client management](#) (9 ページ)
- [debug dot11 client probe](#) (9 ページ)
- [debug dot11 driver slot](#) (10 ページ)
- [debug dot11 firmware](#) (11 ページ)
- [debug dot11 sensor](#) (12 ページ)
- [debug dtls client](#) (13 ページ)
- [debug ethernet](#) (13 ページ)
- [debug flexconnect](#) (14 ページ)
- [debug lldp](#) (15 ページ)
- [debug memory](#) (16 ページ)
- [debug memory pool](#) (16 ページ)
- [debug memory pool alloc](#) (17 ページ)
- [debug memory pool free](#) (17 ページ)
- [debug mesh](#) (18 ページ)
- [debug mesh adjacency](#) (19 ページ)
- [debug mesh path-control](#) (20 ページ)
- [debug rrm neighbor](#) (20 ページ)
- [debug rrm reports](#) (21 ページ)
- [debug sip](#) (22 ページ)
- [debug wips](#) (22 ページ)

- [debug process memory](#) (23 ページ)
- [debug traffic](#) (23 ページ)
- [debug tunnel](#) (24 ページ)
- [debug client trace](#) (25 ページ)
- [no](#) (26 ページ)
- [traceroute](#) (26 ページ)
- [undebug](#) (27 ページ)

debug arp

ARP のデバッグを有効にするには、**debug arp** コマンドを使用します。

debug arp {**errors** | **events** | **packets**}

構文の説明

errors ARP エラーのデバッグを有効にする

events ARP イベントのデバッグを有効にする

packets ARP Tx および Rx パケットのデバッグを有効にする

コマンドモード

Privileged EXEC (#)

コマンド履歴

リリー 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、ARP エラーのデバッグを有効にする例を示します。

```
cisco-ap# debug arp errors
```

debug ble

Bluetooth Low Energy (BLE) のデバッグを有効にするには、**debug ble** コマンドを使用します。

debug ble {**critical** | **error** | **events** | **fastpath** {**rss** | **scan** | **sync**} | **receive** | **transmit**}

構文の説明

critical BLE クリティカル イベントのデバッグを有効にする

error BLE エラー イベントのデバッグを有効にする

events	BLE イベントのデバッグを有効にする
fastpath {rssi scan sync}	CMXにエクスポートされたデータを表示する。次のオプションを使用できます。 <ul style="list-style-type: none"> • RSSI データ • スキャン データ • 同期データ
receive	BLE 無線から受信した BLE パケットのデバッグを有効にする
transmit	BLE 無線に送信された BLE パケットのデバッグを有効にする

コマンドモード Privileged EXEC (#)

コマンド履歴	リリース 変更内容
	8.7 このコマンドが導入されました。

例

次に、BLE クリティカル イベントのデバッグを有効にする例を示します。

```
cisco-ap# debug ble critical
```

debug capwap client

CAPWAP クライアントのデバッグを有効にするには、**debug capwap client** コマンドを使用します。

debug capwap client {ble | detail | efficient-upgrade | error | events | flexconnect | info | keepalive | payload | pmtu | qos | reassembly | security}

構文の説明	ble	CAPWAP BLE 詳細のデバッグを有効する
	detail	CAPWAP 詳細のデバッグを有効する
	efficient-upgrade	イメージのプレダウロードのデバッグを有効にする
	error	CAPWAP エラーのデバッグを有効にする
	events	CAPWAP のイベントのデバッグを有効にする
	flexconnect	CAPWAP FlexConnect モードのイベントのデバッグを有効にする

info	CAPWAP の情報のデバッグを有効にする
keepalive	CAPWAP のキープアライブのデバッグを有効にする
payload	CAPWAP ペイロードのデバッグを有効にする
pmtu	CAPWAP パス MTU のデバッグを有効にする
qos	CAPWAP QoS のデバッグを有効にする
reassemble	CAPWAP リアセンブルのデバッグを有効にする
security	CAPWAP セキュリティのデバッグを有効にする

コマンドモード

Privileged EXEC (#)

コマンド履歴

リリー 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、CAPWAP クライアントの詳細のデバッグを有効にする例を示します。

```
cisco-ap# debug capwap client detail
```

debug capwap client avc

CAPWAP クライアント AVC のデバッグを有効にするには、**debug capwap client avc** コマンドを使用します。

```
debug capwap client avc {all | detail | error | event | info | netflow {all | detail | error | event | packet} | numflows}
```

構文の説明

all	すべての CAPWAP クライアントの AVC のデバッグを有効にする
detail	CAPWAP AVC 詳細のデバッグを有効にする
error	CAPWAP AVC エラーのデバッグを有効にする
event	CAPWAP AVC のイベントのデバッグを有効にする
info	CAPWAP AVC の情報のデバッグを有効にする
netflow	CAPWAP クライアントの AVC NetFlow のデバッグを有効にする

netflow all	すべての CAPWAP クライアントの AVC NetFlow のデバッグを有効にする
netflow detail	CAPWAP クライアントの AVC NetFlow 詳細のデバッグを有効にする
netflow error	CAPWAP クライアントの AVC NetFlow エラーのデバッグを有効にする
netflow event	CAPWAP クライアントの AVC NetFlow イベントのデバッグを有効にする
netflow packet	CAPWAP クライアントの AVC NetFlow パケットのデバッグを有効にする
numflows	CAPWAP クライアント AVC numflows のデバッグを有効にする

コマンドモード Privileged EXEC (#)

コマンド履歴 リリー 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、すべての CAPWAP クライアント AVC のデバッグを有効にする例を示します。

```
cisco-ap# debug capwap client avc all
```

debug cdp

Controller Discovery Protocol (CDP) のデバッグを有効にするには、**debug cdp** コマンドを使用します。

debug cdp {adjacency | events | ilp | packets}

構文の説明	adjacency CDP ネイバーのデバッグを有効にする
	events CDP イベントのデバッグを有効にする
	ilp インライン パワーのデバッグを有効にする
	packets CDP パケットのデバッグを有効にする

コマンドモード Privileged EXEC (#)

コマンド履歴	リリー 変更内容 ス
	8.1.111.0 このコマンドが導入されました。

例

次に、CDP イベントのデバッグを有効にする例を示します。

```
cisco-ap# debug cdp events
```

debug cleanair

CleanAir のデバッグを設定するには、**debug cleanair** コマンドを使用します。

```
debug cleanair {bringup | event | logdebuglow | major | nsi | offchan {0 | 1}}
```

構文の説明	bringup CleanAir ポートまたは起動のデバッグを有効にする
	events 通常の CleanAir イベントのデバッグを有効にする
	logdebug CleanAir のデバッグ出力をログファイルに記録する
	low 一部のメッセージの 16 進ダンプのデバッグを有効にする
	major 主要な CleanAir イベントのデバッグを有効にする
	nsi NSI メッセージのデバッグを有効にする
	offchan 0 1 CleanAir MSMT 要求のデバッグを有効にする。無線スロットを 0 または 1 で指定する必要がある

コマンドモード Privileged EXEC (#)

コマンド履歴	リリー 変更内容 ス
	8.1.111.0 このコマンドが導入されました。

例

次に、主要な CleanAir イベントのデバッグを有効にする例を示します。

```
cisco-ap# debug cleanair major
```

debug dhcp

DHCP のデバッグを設定するには、**debug dhcp** コマンドを使用します。

debug dhcp { **errors** | **events** | **packets** }

構文の説明	errors DHCP エラーのデバッグを有効にする
	events DHCP イベントのデバッグを有効にする
	packets DHCP パケットのデバッグを有効にする

コマンドモード Privileged EXEC (#)

コマンド履歴 リリー 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、DHCP エラーのデバッグを有効にする例を示します。

```
cisco-ap# debug dhcp errors
```

debug dot11 driver level

802.11 のデバッグを有効にするには、**debug dot11** コマンドを使用します。

debug dot11 driver level { **critical** | **errors** | **events** | **info** }

構文の説明	critical 802.11 のクリティカル レベルのデバッグを有効にする
	errors 802.11 のエラー レベルのデバッグを有効にする
	events 802.11 のイベント レベルのデバッグを有効にする
	info 802.11 の情報レベルのデバッグを有効にする

コマンドモード Privileged EXEC (#)

コマンド履歴

 リリー 変更内容
 ス

 8.1.111.0 このコマンドが導入されました。

例

次に、802.11 のエラー レベルのデバッグを有効にする例を示します。

```
cisco-ap# debug dot11 errors
```

debug dot11 client data-path

802.11 クライアントデータパスのデバッグを有効にするには、**debug dot11 client data-path** コマンドを使用します。

```
debug dot11 client data-path { { all-types | arp | dhcp | eapol | ipv6-ra | opendns | dns-acl } { addr { mac-addr1 | mac-addr2 | mac-addr3 | mac-addr4 } }
```

構文の説明

arp	クライアント データパスの ARP デバッグを有効にする
dhcp	クライアント データパスの DHCP デバッグを有効にする
eapol	クライアント データパスの EAPOL デバッグを有効にする
dns-acl	クライアント データパスの DNS-ACL デバッグを有効にする
ipv6-ra	クライアント データパスの IPv6 RA-MC2UC デバッグを有効にする
opendns	クライアント データパスの openDNS デバッグを有効にする
{addr all-types}	特定のクライアントまたはすべてのクライアントの MAC アドレスを指定するオプション
<i>{mac-addr1 mac-addr2 mac-addr3 mac-addr4}</i>	入力する必要があるクライアントの MAC アドレス

コマンドモード

Privileged EXEC (#)

コマンド履歴

 リリー 変更内容
 ス

 8.1.111.0 このコマンドが導入されました。

例

次に、クライアントデータパスの ARP のデバッグを有効にする例を示します。

```
cisco-ap# debug dot11 client data-path arp
```

debug dot11 client management

802.11 クライアントデバッグレベルを有効にするには、**debug dot11 client management** コマンドを使用します。

```
debug dot11 client management { critical | errors | events | info } { addr { mac-addr1 | mac-addr2 | mac-addr3 | mac-addr4 } }
```

構文の説明

critical	クライアントのクリティカル レベルのデバッグを有効にする
errors	クライアントのエラー レベルのデバッグを有効にする
events	クライアントのイベント レベルのデバッグを有効にする
info	クライアントの情報レベルのデバッグを有効にする
{ <i>mac-addr1</i> <i>mac-addr2</i> <i>mac-addr3</i> <i>mac-addr4</i> }	入力する必要があるクライアントの MAC アドレス

コマンドモード

Privileged EXEC (#)

コマンド履歴

リリー 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、イベントレベルでクライアントのデバッグを有効にする例を示します。

```
cisco-ap# debug dot11 client management events e1:90:6f:7e:e6:29
```

debug dot11 client probe

802.11 クライアントのデバッグプローブを有効にするには、**debug dot11 client probe** コマンドを使用します。

```
debug dot11 client probe {{ address mac-addr1 | mac-addr2 | mac-addr3 | mac-addr4 } | all
}
```

構文の説明

address MAC アドレスを使用して特定のクライアントをプローブする。

mac-addr クライアントの MAC アドレス。最大 4 つの MAC アドレスを入力できる。

all AP に関連付けられているすべてのクライアントをプローブする。

コマンドモード

Privileged EXEC (#)

コマンド履歴

リリー 変更内容
ス

8.10 このコマンドが導入されました。

例

次に、すべてのクライアントのデバッグを有効にする例を示します。

```
cisco-wave2-ap# debug dot11 client probe all
```

debug dot11 driver slot

802.11 ドライバのデバッグを有効にするには、**debug dot11 driver slot** コマンドを使用します。

```
debug dot11 driver slot {0 | 1} { all-types | { cac { info | metrics } } | chd |
save-accounting-data | save-on-failure [ extended ] | stop-on-failure | metrics traffic
| metrics video | type { all | association | authentication | dhcp | eap | icmp
| probe } mac-addr1 | mac-addr2 | mac-addr3 | mac-addr4
```

構文の説明

slot {0 | 1} 無線ごとの 802.11 ドライバのデバッグを有効にする

all-types すべての 802.11 ドライバのデバッグを有効にする

cac 802.11 CAC デバッグを有効にする

cac info 802.11 CAC の情報レベルのデバッグを有効にする

cac metrics 802.11 CAC メトリックのデバッグを有効にする

chd 802.11 CHD デバッグを有効にする

save-accounting-data 無線アカウンティング データを保存する

save-on-failure 無線の障害発生時に無線クラッシュ情報を保存する

save-on-failure extended 無線障害に関する詳細情報を保存する

stop-on-failure	無線障害時に AP の再起動を停止する
metrics traffic	802.11 トラフィック ストリーム メトリックのデバッグを有効にする
metrics video	802.11 ビデオ メトリックのデバッグを有効にする
type	デバッグタイプを有効にする。
all	all タイプのデバッグを有効にする。
association	関連付けのデバッグを有効にする。
authentication	認証のデバッグを有効にする。
dhcp	dhcp のデバッグを有効にする。
eap	eap のデバッグを有効にする。
icmp	icmp のデバッグを有効にする。
probe	プローブのデバッグを有効にする。
mac-addr	クライアントの MAC アドレス。最大 4 つの MAC アドレスを入力できる。

コマンドモード

Privileged EXEC (#)

コマンド履歴

リリース	変更内容
8.1.111.0	このコマンドが導入されました。
8.5.140.0 および 8.8	type パラメータを追加することで、このコマンドが拡張されました。

例

次に、情報レベルで CAC のデバッグを有効にする例を示します。

```
cisco-ap# debug dot11 driver slot cac info
```

debug dot11 firmware

802.11 ファームウェアをデバッグするには、**debug dot11 firmware** コマンドを使用します。

```
debug dot11 firmware slot slot_ID level { all-level | critical | emergency | error | info } address { mac-addr1 | mac-addr2 | mac-addr3 | mac-addr4 }
```

構文の説明

slot_ID 無線ごとの 802.11 ドライバのデバッグを有効にする

all-level	すべてのデバッグレベルを有効にする。
critical	クリティカルレベルのデバッグを有効にする。
emergency	緊急レベルのデバッグを有効にする。
error	エラーレベルのデバッグを有効にする。
info	情報レベルのデバッグを有効にする。
address	ドライバ/ファームウェアのデバッグ用のクライアントアドレスを追加する。
mac-addr	クライアントの MAC アドレス。最大 4 つの MAC アドレスを入力できる。

コマンドモード

特権 EXEC (#)

コマンド履歴

リリース	変更内容
8.5.140.0 および 8.8	このコマンドが導入されました。

例

次に、802.11 の緊急レベルのデバッグを有効にする例を示します。

```
cisco-wave2-ap# debug dot11 firmware slot 1 emergency address 92:FB:D6:B3:7A:6C
```

debug dot11 sensor

802.11 センサーのデバッグを有効にするには、**debug dot11 sensor** コマンドを使用します。

```
debug dot11 sensor {dns | file-transfer | mail-server | ping | radius | ssh | telnet | web-server}
```

構文の説明

dns	802.11 センサーの DNS のデバッグを有効にする
file-transfer	802.11 センサーのファイル転送のデバッグを有効にする
mail-server	802.11 センサーのメール サーバのデバッグを有効にする
ping	802.11 センサーの ping のデバッグを有効にする
radius	802.11 センサーの radius のデバッグを有効にする
ssh	802.11 センサーの SSH のデバッグを有効にする
telnet	802.11 センサーの Telnet のデバッグを有効にする

web-server 802.11 センサーの Web サーバのデバッグを有効にする

コマンドモード Privileged EXEC (#)

コマンド履歴 リリース 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、802.11 センサーのファイル転送のデバッグを有効にする例を示します。

```
cisco-ap# debug dot11 sensor file-transfer
```

debug dtls client

DTLS クライアントエラーとイベントのデバッグを設定するには、**debug dtls client** コマンドを使用します。

debug dtls client {**error** | **event** [**detail**]}

構文の説明	error DTLS クライアントエラーのデバッグを設定する
	event [detail] DTLS クライアントイベントのデバッグを設定する

コマンドモード Privileged EXEC (#)

コマンド履歴 リリース 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、DTLS クライアントイベントのデバッグを有効にする例を示します。

```
cisco-ap# debug dtls client event
```

debug ethernet

イーサネットのデバッグを設定するには、**debug ethernet** コマンドを使用します。

debug ethernet *interface-number* {**both** | **rcv** | **xmt**}

構文の説明	<i>interface-number</i> インターフェイス番号で 0 または 1 を入力する必要がある
both	送信と受信の両方のデバッグを有効にする
rcv	受信のデバッグを有効にする
xmt	送信のデバッグを有効にする

コマンドモード Privileged EXEC (#)

コマンド履歴 リリリー 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、インターフェイス 0 の送信のデバッグを有効にする例を示します。

```
cisco-ap# debug ethernet 0 xmt
```

debug flexconnect

FlexConnect 機能をデバッグするには、**debug flexconnect** コマンドを使用します。

debug flexconnect {**acl** | **cckm** | **dot11r** | **event** | **multicast** {**igmp** | **traffic**} | **pmk** | **proxy-arp** | **vsa** | **wlan-vlan** | **wsastats**}

構文の説明	acl FlexConnect ACL のデバッグを設定する
cckm	CCKM のデバッグを設定する
dot11r	802.11r のデバッグを設定する
event	Wireless Control Protocol (WCP) イベントのデバッグを設定する
multicast igmp	マルチキャスト IGMP のデバッグを設定する
multicast traffic	マルチキャスト トラフィックのデバッグを設定する
pmk	Opportunistic Key Caching (OKC) またはペアワイズ マスター キー キャッシングのデバッグを設定する
vsa	AAA ベンダー固有属性 (VSA) のデバッグを設定する

wlan-vlan	WLAN-VLAN マッピングのデバッグを設定する
wsastats	RADIUS または DHCP ワイヤレス サービス保証の統計情報のデバッグを設定する

コマンドモード

Privileged EXEC (#)

コマンド履歴

リリー 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、FlexConnect ACL のデバッグを有効にする例を示します。

cisco-ap# **debug flexconnect acl**

debug lldp

LLDP をデバッグするには、**debug lldp** コマンドを使用します。**debug lldp {errors | events | packet}**

構文の説明

errors LLDP エラーをデバッグする**events** LLDP イベントをデバッグする**packet** LLDP パケットをデバッグする

コマンドモード

Privileged EXEC (#)

コマンド履歴

リリー 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、LLDP エラーのデバッグを有効にする例を示します。

cisco-ap# **debug lldp errors**

debug memory

メモリをデバッグするには、**debug memory** コマンドを使用します。

debug memory {clear | save}

構文の説明

clear ブートアップ時にメモリのデバッグを削除する

save 現在のデバッグ レベルを保存し、次のブート時に適用する

コマンドモード

Privileged EXEC (#)

コマンド履歴

リリー 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、ブートアップ時にメモリ デバッグを削除する例を示します。

```
cisco-ap# debug memory clear
```

debug memory pool

メモリプールをデバッグするには、**debug memory pool** コマンドを使用します。

debug memory pool {diff | realtime interval *1-1000000-seconds* | start}

構文の説明

diff メモリ プールのデバッグの差分を詳細に示す

realtime interval *1-1000000-seconds* メモリプールのリアルタイムの間隔を設定する。

start メモリ プールのデバッグを開始する

コマンドモード

Privileged EXEC (#)

コマンド履歴

リリー 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、メモリ プールにリアルタイムの間隔として 180 秒を設定する例を示します。

```
cisco-ap# debug memory pool realtime interval 180
```

debug memory pool alloc

メモリプールの割り当てコールをデバッグするには、**debug memory pool alloc** コマンドを使用します。

```
debug memory pool alloc {all | name pool-name} {diff | realtime interval 1-1000000-seconds | start}
```

構文の説明

all	すべてのメモリ プールの割り当てコールにデバッグを設定する
name pool-name	特定のメモリ プールの割り当てコールにデバッグを設定する
diff	メモリ プールのデバッグの割り当てコールの差分を詳細に示す
realtime interval 1-1000000-seconds	メモリ プールの割り当てコールにリアルタイムの間隔を設定する
start	メモリ プールの割り当てコールのデバッグを開始する

コマンドモード

Privileged EXEC (#)

コマンド履歴

リリー 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、すべてのメモリ プールの割り当てコールにデバッグの開始を設定する例を示します。

```
cisco-ap# debug memory pool alloc all start
```

debug memory pool free

メモリプールの解放コールをデバッグするには、**debug memory pool free** コマンドを使用します。

```
debug memory pool free {all | name pool-name} {diff | realtime interval 1-1000000-seconds
| start}
```

構文の説明	all	すべてのメモリ プールの解放コールにデバッグを設定する
	name pool-name	特定のメモリ プールの解放コールにデバッグを設定する
	diff	メモリ プールのデバッグの解放コールの差分を詳細に示す
	realtime interval 1-1000000-seconds	メモリ プールの解放コールにリアルタイムの間隔を設定する
	start	メモリ プールの解放コールのデバッグを開始する

コマンドモード Privileged EXEC (#)

コマンド履歴

リリー 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、すべてのメモリ プールの解放コールにデバッグの開始を設定する例を示します。

```
cisco-ap# debug memory pool free all start
```

debug mesh

メッシュネットワークのデバッグを設定するには、**debug mesh** コマンドを使用します。

```
debug mesh {channel | clear | convergence | events | forward-mcast | forward-packet |
forward-table | linktest | path-control | port-control | security | trace}
```

構文の説明	channel	メッシュ チャネルのデバッグを設定する
	clear	すべてのメッシュ デバッグをリセットする
	convergence	メッシュ コンバージェンスのデバッグを設定する
	events	メッシュ イベントのデバッグを設定する
	forward-mcast	メッシュ転送マルチキャストのデバッグを設定する
	forward-packet	メッシュ転送パケットのデバッグを設定する

forward-table	メッシュ転送テーブルのデバッグを設定する
linktest	メッシュリンクテストのデバッグを設定する
port-control	メッシュポート制御のデバッグを設定する
security	メッシュセキュリティのデバッグを設定する
trace	メッシュトレースのデバッグを設定する

コマンドモード Privileged EXEC (#)

コマンド履歴 リリー 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、メッシュチャネルのデバッグを有効にする例を示します。

```
cisco-ap# debug mesh channel
```

debug mesh adjacency

メッシュ隣接関係をデバッグするには、**debug mesh adjacency** コマンドを使用します。

debug mesh adjacency {child | clear | dfs | message | packet | parent }

構文の説明

adjacency	メッシュ隣接関係をデバッグする
child	メッシュ隣接関係の子をデバッグする
clear	メッシュ隣接関係のクリアをデバッグする
dfs	メッシュ DFS をデバッグする
message	メッシュ隣接関係のメッセージをデバッグする
packet	メッシュ隣接関係のパケットをデバッグする
parent	メッシュ隣接関係の親をデバッグする

コマンドモード Privileged EXEC (#)

コマンド履歴	リリー 変更内容 ス
	8.1.111.0 このコマンドが導入されました。

例

次に、メッシュ隣接関係の親のデバッグを有効にする例を示します。

```
cisco-ap# debug mesh adjacency parent
```

debug mesh path-control

メッシュパス制御のデバッグを設定するには、**debug mesh path-control** コマンドを使用します。

debug mesh path-control {error | events | packets }

構文の説明	error メッシュパス制御エラーのデバッグを設定する
	events メッシュパス制御イベントのデバッグを設定する
	packets メッシュパス制御パケットのデバッグを設定する

コマンドモード Privileged EXEC (#)

コマンド履歴	リリー 変更内容 ス
	8.1.111.0 このコマンドが導入されました。

例

次に、メッシュパス制御エラーのデバッグを有効にする例を示します。

```
cisco-ap# debug mesh path-control error
```

debug rrm neighbor

RRM ネイバーのデバッグを有効にするには、**debug rrm neighbor** コマンドを使用します。

debug rrm neighbor {tx | rx | detail }

構文の説明	tx	RRM ネイバー Tx のデバッグを有効にする
	rx	RRM ネイバー Rx のデバッグを有効にする
	detail	RRM ネイバーの詳細のデバッグを有効にする

コマンドモード Privileged EXEC (#)

コマンド履歴	リリース 変更内容 ス
	8.1.111.0 このコマンドが導入されました。

例

次に、RRM ネイバーの送信のデバッグを有効にする例を示します。

```
cisco-ap# debug rrm neighbor tx
```

debug rrm reports

RRM レポートのデバッグを有効にするには、**debug rrm reports** コマンドを使用します。

debug rrm reports

構文の説明	reports	RRM レポートのデバッグを有効にする
-------	----------------	---------------------

コマンドモード Privileged EXEC (#)

コマンド履歴	リリース 変更内容 ス
	8.1.111.0 このコマンドが導入されました。

例

次に、RRM レポートのデバッグを有効にする例を示します。

```
cisco-ap# debug rrm reports
```

debug sip

Session Initiation Protocol (SIP) のデバッグを有効にするには、**debug sip** コマンドを使用します。

debug sip {**all** | **tx** | **rx**}

構文の説明	all SIP の送信と受信のデバッグを有効にする
	tx SIP の送信のデバッグを有効にする
	rx SIP の受信のデバッグを有効にする
コマンドモード	Privileged EXEC (#)
コマンド履歴	リリー 変更内容 ス 8.1.111.0 このコマンドが導入されました。

例

次に、SIP の送信と受信のデバッグを有効にする例を示します。

```
cisco-ap# debug sip all
```

debug wips

wIPS デバッグを有効にするには、**debug wips** コマンドを使用します。

debug wips {**errors** | **events** | **critical**}

構文の説明	errors wIPS エラー レベルのデバッグを有効にする
	events wIPS イベント レベルのデバッグを有効にする
	critical wIPS クリティカル レベルのデバッグを有効にする
コマンドモード	Privileged EXEC (#)

コマンド履歴	リリース 変更内容 ス
	8.1.111.0 このコマンドが導入されました。

例

次に、wIPS エラー レベルのデバッグ を有効にする例を示します。

```
cisco-ap# debug wips errors
```

debug process memory

プロセスメモリをデバッグするには、**debug process memory** コマンドを使用します。

```
debug process memory {diff | realtime [interval interval-in-seconds] | start}
```

構文の説明	diff プロセス メモリのデバッグで差分を表示する
	realtime プロセス メモリをリアルタイムにデバッグする
	interval 更新間隔。有効な範囲は 1 ~ 1000000 秒
	start プロセス メモリのデバッグを開始する

コマンドモード	Privileged EXEC (#)
---------	---------------------

コマンド履歴	リリース 変更内容 ス
	8.1.111.0 このコマンドが導入されました。

例

次に、プロセス メモリのデバッグの開始を有効にする例を示します。

```
cisco-ap# debug process memory start
```

debug traffic

トラフィックのデバッグを有効にするには、**debug traffic** コマンドを使用します。

```
debug traffic {host {icmpv6 | ip | ipv6 | tcp | udp { verbose}} | wired {ip | tcp | udp { verbose}}}
```

構文の説明	host	ホスト トラフィックのデバッグを有効にする
	wired	有線トラフィックのデバッグを有効にする
	verbose	詳細な出力を表示する
	icmpv6	ホスト ICMPv6 トラフィック ダンプを有効にする
	ip	ホスト IP トラフィック ダンプを有効にする
	ipv6	ホスト IPv6 トラフィック ダンプを有効にする
	tcp	TCP トラフィック ダンプを有効にする
	udp	UDP トラフィック ダンプを有効にする

コマンドモード Privileged EXEC (#)

コマンド履歴	リリース	変更内容
	8.1.111.0	このコマンドが導入されました。

例

次に、ホスト IP トラフィック ダンプのデバッグを有効にする例を示します。

```
cisco-ap# debug traffic host ip
```

debug tunnel

トンネルのデバッグを設定するには、**debug tunnel** コマンドを使用します。

debug tunnel eogre

構文の説明	eogre	EoGRE トンネルのデバッグを設定する
-------	--------------	----------------------

コマンドモード Privileged EXEC (#)

コマンド履歴	リリース	変更内容
	8.1.111.0	このコマンドが導入されました。

例

次に、EoGRE トンネルのデバッグを有効にする例を示します。

```
cisco-ap# debug tunnel eogre
```

debug client trace

クライアントトレースのデバッグを有効にするには、**debug client trace** コマンドを使用します。

```
debug client trace {all | address mac-address | enable | filter {assoc | auth | dhcp | eap | icmp | mgmt | probe | proto}}
```

構文の説明	all	すべてのクライアントトレースを設定する
	address	トレースするアドレスを設定する
	<i>mac-address</i>	トレースする MAC アドレス
	enable	トレースを有効にする
	filter	トレース フィルタを設定する
	assoc	関連付けパケットをトレースする
	auth	認証パケットをトレースする
	dhcp	DHCP パケットをトレースする
	eap	EAP パケットをトレースする
	icmp	ICMP パケットをトレースする
	mgmt	probe、assoc、auth、EAP パケットをトレースする
	probe	プローブ パケットをトレースする
	proto	DHCP、ICMP パケットをトレースする
コマンドモード	Privileged EXEC (#)	
コマンド履歴	リリー	変更内容
	8.1.111.0	このコマンドが導入されました。

例

次に、すべてのクライアントのトレースを有効にする例を示します。

```
cisco-ap# debug client trace all
```

no

コマンドを無効にするか、デフォルトに設定するには、**no** コマンドを使用します。

no**コマンドモード**

Privileged EXEC (#)

コマンド履歴リリー 変更内容
ス

8.1.111.0 このコマンドが導入されました。

コマンドを無効にするか、デフォルトに設定するには、次のコマンドを使用します。

```
cisco-ap# no debug
```

traceroute

ネットワーク内を移動するパケットのルートを表示するには、**traceroute** コマンドを使用します。

traceroute *destination-address*

構文の説明

destination-address パケットの宛先の IP アドレス

コマンドモード

Privileged EXEC (#)

コマンド履歴リリー 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、ネットワーク内を移動する、指定した宛先 IP アドレスを持つパケットのルートを表示する例を示します。

```
cisco-ap# traceroute 209.165.200.224
```

undebug

アクセスポイントでデバッグを無効にするには、**undebug** コマンドを使用します。

undebug [all]

構文の説明

a すべてのデバッグ メッセージを無効にする

コマンドモード

Privileged EXEC (#)

コマンド履歴

リリー 変更内容
ス

8.1.111.0 このコマンドが導入されました。

例

次に、すべてのデバッグ メッセージを無効にする例を示します。

```
cisco-ap# undebug all
```


翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。