

キューでのトレース データの設定と袖手

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[トレースの概要](#)

[Advanced Integration Module \(AIM \) とネットワーク モジュール \(NM \)](#)

[トレースの設定](#)

[トレース データの収集](#)

[トレース バッファの操作](#)

[保存されたトレース ログ ファイル](#)

[FTP サーバへのトレース](#)

[JTAPI トレース](#)

[トレースの電源オフ](#)

[デフォルト トレースの最有効化](#)

[関連情報](#)

概要

このドキュメントでは、Cisco Unity Express (CUE) のトレース機能の概要を示しています。トレースは Cisco Unity Express のデバッグ機能で、さまざまな問題をトラブルシューティングするために使用されます。トレース機能を有効にすると、システム パフォーマンスに悪影響を与えることがあります。このため、特定の問題に関する情報を収集するようにシスコのテクニカル サポートが依頼した場合にのみトレースを有効にすることを推奨します。ラボまたはメンテナンス時間帯のシステムでは、Cisco Unity Express の動作をトラブルシューティングし、理解するために、トレース機能を使用できます。

前提条件

要件

コマンドライン インターフェイス (CLI) によって Cisco Unity Express Administration を管理し使用する上での基本的な知識があることが推奨されます。

使用するコンポーネント

この機能には Cisco Unity Express ソフトウェア バージョン 1.0 以降が必要です。統合方法 (Cisco CallManager または Cisco CallManager Express) は重要ではありません。このドキュメ

ントのサンプル設定と画面出力はすべて、Cisco Unity Express バージョン 1.1.1 のものです。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。対象のネットワークが実稼働中である場合には、どのようなコマンドについても、その潜在的な影響について確実に理解しておく必要があります。

表記法

ドキュメント表記の詳細は、『[シスコテクニカルティップスの表記法](#)』を参照してください。

トレースの概要

Cisco IOS® ソフトウェアに慣れているユーザは、CLI や強力なデバッグ機能についてあまり馴染みがないことがよくあります。Cisco Unity Express には機能が似たツールが存在しますが、いくつか重要な違いがあります。Cisco Unity Express には、**debug コマンドがありません**。代わりに、**trace コマンドがあります**。トレース機能は、メモリのカーネル バッファ内にメッセージを書き込む診断機能です。最大で 10 MB 程度になるこのメモリ領域は、ローカルトレース ファイル (atrace.log) と外部 FTP サーバのファイルのどちらか一方、またはその両方に定期的書き込むように設定できます。

注： atrace.log ファイルと、FTP サーバに記録されたトレースデータは、プレーンテキストではありません。データを診断するには、シスコテクニカルサポートに送信する必要があります。

Cisco Unity Express モジュールに書き込まれた各ファイル (atrace.log と messages.log その他) は、外部 FTP サーバに手動でコピーできます。

Cisco Unity Express は、Cisco IOS ソフトウェアの syslog に似たログ機能もサポートします。これらのメッセージは、オペレーティングシステムや他のアプリケーションソースからのメッセージで、異なるレベルに分類されています。これらのメッセージには「Info」、「Warning」、「Error」、「Fatal」のレベルがあり、Cisco Unity Express の別のファイル (messages.log) に書き込まれます。さらに、Cisco Unity Express のコンソールだけでなく、外部の syslog サーバにも書き込むことができます。

CUE モジュールで INFO メッセージを外部の syslog サーバに記録するには、CUE モジュールで次のコマンドを発行します。

```
CUE> config t
      CUE(config)>log server
```

デフォルトでは、「Fatal」メッセージのみが AIM に記録されます。ほとんどの一般的な問題で、messages.log ファイルと、その障害に関するトレース情報が必要になります。

シスコテクニカルサポートから特定のトレースを収集するように指示を受けた場合、有効にする必要のある特定のトレース、およびキャプチャ方式についての合意が必要になります。たとえば、リアルタイムトレースの使用、トレースメモリバッファの表示、FTP サーバのトレースデー

タのキャプチャなどを行う場合があります。

Advanced Integration Module (AIM) とネットワーク モジュール (NM)

Cisco Unity Express には、AIM と NM という 2 つのハードウェア モデルがあります。トレース機能の点で、両者にはいくつかの重要な違いがあります。

AIM	NM
atrace.log ファイルが、デフォルトで無効。開始する場合は <code>log trace local enable</code> コマンドを発行し、停止する場合は <code>log trace local disable</code> コマンドを発行します。	atrace.log ファイルが、デフォルトで有効。外部サーバへのトレースもサポートされています。
atrace.log の最大サイズは 10 MB。	atrace.log の最大サイズは 100 MB。
ファイルが一杯の場合、atrace.log ファイルはラップされません。	ファイルが一杯の場合、atrace.log ファイルはラップされます。

トレース データを有効にし表示する方法については、このドキュメントの後の部分で詳しく説明します。AIM は、デフォルトで、トレース情報をフラッシュ自体には保存しません。また、トレース データ用に内部ストレージ容量を有効にしている場合、容量はさらに限られたものになります。これは、AIM の内蔵コンパクト フラッシュ カードの寿命が、作成される書き込みの回数の影響を受けるためです。常にトレースを書き込むと、寿命が著しく短くなります。

注：最大サイズに達したAIMでatrace.logファイルを再起動するには、`log trace local disable` コマンドを発行し、次に `log trace local enable` コマンドを設定モードで発行します。このようにすると、元の atrace.log ファイルが削除され、新しいファイルが作成されます。

ログ機能の面でも、重要な違いがあります。

AIM	NM
「Fatal」メッセージのみ、messages.log ファイルにデフォルトで記録される。すべてのメッセージを確認するには、コンフィギュレーション モードで <code>log console info</code> コマンドを発行します。	すべてのカテゴリのメッセージが messages.log ファイルに記録される。

トレースの設定

注意：Cisco Unity Expressでのトレースの設定は、特に有効になっているローカルログファイルに書き込む場合にシステムパフォーマンスに悪影響を及ぼす可能性があります。これには、プロンプトやデュアルトーン多重周波数 (DTMF) トーン応答時間の遅延、録音音声や再生音声の品質に関する問題などが含まれます。トレースの設定は注意して行ってください。

トレースの設定により、トレースメモリバッファに書き込まれるメッセージのタイプが決まります。この 10 MB のバッファは、最新のトレース情報が保持されるよう、常にラップされます。システムにはさまざまなレベルのアクティビティが存在するので、このトレースバッファがカバーしている時間を知ることは不可能です。ただし、バッファはローカルの `atrace.log` か FTP サーバに書き込まれるように設定できます。

トレースに関しては、Cisco Unity Express の CLI からのみ設定できます。有効になっている現在のトレースを表示するには、`show trace` コマンドを発行します。

以下に、いくつかの例を示します。

```
vnt-3745-44a#service-module service-Engine 4/0 session
Trying 172.18.106.66, 2129 ... Open
vnt-nm-cue#
vnt-nm-cue#show trace
MODULE          ENTITY          SETTING
ccn             Engine          00000001
ccn             LibLdap         00000001
ccn             SubsystemAppl  00000001
ccn             ManagerAppl    00000001
ccn             ManagerChannel 00000001
ccn             SubsystemJtapi 00000001
ccn             SubsystemSip   00000001
ccn             StackSip       00000001
ccn             SubsystemHttp  00000001
ccn             VbrowserCore   00000001
ccn             SubsystemCmt   00000001
ccn             LibMedia       00000001
ccn             ManagerContact 00000001
ccn             StepCall       00000001
ccn             StepMedia      00000001
config-ccn     sip-subsystem  00000001
config-ccn     jtapi-subsystem 00000001
config-ccn     sip-trigger    00000001
config-ccn     jtapi-trigger  00000001
config-ccn     http-trigger   00000001
config-ccn     group         00000001
config-ccn     application    00000001
config-ccn     script        00000001
config-ccn     prompt        00000001
config-ccn     miscellaneous  00000001
voicemail     database       0000008f
voicemail     mailbox        0000003f
voicemail     message       0000002f
webInterface  initwizard     00000001
vnt-nm-cue#
```

これらは、NM と AIM 双方のデフォルトのトレース設定です。AIM は、デフォルトで、これらのトレース出力をどこにも保存しません。ほとんどの一般的なトラブルシューティングでは、このトレースレベルで十分です。最近発生した問題であれば、トレースメモリバッファに履歴が残っている場合があります。

トレースを有効にするには、`trace module entity activity` コマンドを発行します。以下に、いくつかの例を示します。

```
vnt-nm-cue#trace ccn subsystemsip debug
```

次の例は CUE 1.1.1 用モジュールです。

```
vnt-nm-cue#trace ?
```

```
BackupRestore Module
all                Every module, entity and activity
ccn                Module
config-ccn        Module
dns                Module
superthread       Module
udppacer           Module
voicemail          Module
webInterface      Module
```

各モジュールに多くのエンティティがあります。アクティビティレベルはいくらか異なります（通常はモジュールごと）。一般的に、各エンティティには少なくとも *debug* (*DEBUG*とも表記) と *all* のアクティビティレベルがあります。通常は、*debug* アクティビティレベルで十分です。

trace module entity activity コマンドを複数回発行することで、必要なモジュールとエンティティすべてのトレースを有効にできます。

どのトレースが設定されるかは関係ありません。リロードすると、システムはデフォルトのトレースレベルに戻ります。リブート後もトレースレベルを維持するように、これらのデフォルト設定を変更するには、***log traceboot boot*** コマンドを発行する必要があります。

トレースデータの収集

すべてのトレースを設定すると、データがメモリバッファに書き込まれます。すると、メッセージ受信時にリアルタイムでデータを表示するか、イベントまたはテストの実行後にバッファを表示できるようになります。

トレースバッファの操作

メモリベースのトレースバッファは、最初にトレースを確認すべき場所の1つです。これはリアルタイムで表示でき、そのように設定するとトレースメッセージの受信時に表示されます。代わりに、メモリバッファの全体または一部を表示して確認することもできます。

リアルタイムトレース

リアルタイムトレースは、コントロールされたシステム（同時コールが多くないか、システムが未稼働の場合）の問題をトラブルシューティングする際に特に役立ちます。トレースの出力行は大抵ラップされていて、ほとんどの場合その情報は読むことが不可能な速さでスクロールするので、リアルタイムトレースをオンにする前に、コンソール出力をテキストファイルに記録するようにします。これにより、情報をテキストエディタで後から参照できます。たとえば、Microsoft HyperTerminal では、**[Transfer] > [Capture Text]** の順に選択し、キャプチャファイルを指定できます。

リアルタイムトレース機能を使用すると、システムのパフォーマンスに非常に大きな影響を与えます。注意して使用してください。

トレース情報をリアルタイムで表示するには、***show trace buffer tail*** コマンドを発行します。以下に、いくつかの例を示します。

```
vnt-nm-cue>show trace buffer tail
```

```

Press <CTRL-C> to exit...
295 06/22 10:39:55.428 TRAC TIMZ 1 EST EDT 18000
2019 06/22 11:20:15.164 ACCN SIPL 0 receive 1098 from 172.18.106.66:54948
2020 06/22 11:20:15.164 ACCN SIPL 0 not found header for Date
2020 06/22 11:20:15.164 ACCN SIPL 0 not found header for Supported
2020 06/22 11:20:15.164 ACCN SIPL 0 not found header for Min-SE
2020 06/22 11:20:15.165 ACCN SIPL 0 not found header for Cisco-Guid
2020 06/22 11:20:15.165 ACCN SIPL 0 not found header for Remote-Party-ID
2020 06/22 11:20:15.165 ACCN SIPL 0 not found header for Timestamp
2020 06/22 11:20:15.165 ACCN SIPL 0 not found header for Call-Info
2020 06/22 11:20:15.165 ACCN SIPL 0 not found header for Allow-Events
2020 06/22 11:20:15.166 ACCN SIPL 0 -----
INVITE sip:18999@172.18.106.88:5060 SIP/2.0
Via: SIP/2.0/UDP 172.18.106.66:5060;branch=z9hG4bK1678
From: "Caller1" <sip:201@172.18.106.66>;tag=23F5B364-22C9
To: <sip:18999@172.18.106.88>
Date: Tue, 22 Jun 2004 15:20:14 GMT
Call-ID: 7E86EC94-C39611D8-AF50DA50-D3EDBBC9@172.18.106.66
Supported: 100rel,timer
Min-SE: 1800
Cisco-Guid: 2092538615-3281392088-2941114960-3555572681
...

```

この情報は、Cisco IOS ソフトウェアの debug 出力と同様にスクロールします。1 つの違いは、Ctrl + C キーと一緒に押すだけでこれを停止できる点です。

[トレース メモリ バッファの表示](#)

メモリのトレース バッファは最大で 10 MB です。把握しておく必要があるコマンドライン パラメータがいくつかあります。

```

vnt-nm-cue>show trace buffer ?
<cr>
containing    Only display events matching a regex pattern
long          Show long format
short         Show short format
tail          Wait for events and print them as they occur
|             Pipe output to another command

```

ほとんどの場合、使用する必要があるオプションは **show trace buffer long** のみです。paged というキーワードを末尾に追加して、一度に 1 ページずつ表示されるようにすることができます。long 形式には多くのエラーやリターン コードに関する補足テキストが含まれるのに対して、short 形式には 16 進数のコードのみが含まれます。通常は、ターミナル プログラムのキャプチャ機能を使用してすべてを把握し、テキスト エディタの検索機能を使用して特定の情報を見つける方法が簡単ですが、特定のエラー状態のみを検索する必要がある場合は、*containing* キーワードが便利です。この方法では、正規表現パラメータを使用して、ディスプレイへの出力内容をフィルタ処理できます。

注： containing キーワードのみで特定のコールまたはポート番号を検索することはできません。

```

vnt-nm-cue>show trace buffer long paged
2029 06/24 17:48:40.479 ACCN SIPL 0 receive 1096 from 172.18.106.66:49255
2030 06/24 17:48:40.480 ACCN SIPL 0 not found header for Date
2030 06/24 17:48:40.480 ACCN SIPL 0 not found header for Supported
2030 06/24 17:48:40.480 ACCN SIPL 0 not found header for Min-SE
2030 06/24 17:48:40.480 ACCN SIPL 0 not found header for Cisco-Guid
2030 06/24 17:48:40.480 ACCN SIPL 0 not found header for Remote-Party-ID
2030 06/24 17:48:40.480 ACCN SIPL 0 not found header for Timestamp
2030 06/24 17:48:40.480 ACCN SIPL 0 not found header for Call-Info

```

```
2030 06/24 17:48:40.480 ACCN SIPL 0 not found header for Allow-Events
2030 06/24 17:48:40.481 ACCN SIPL 0 -----
INVITE sip:18900@172.18.106.88:5060 SIP/2.0
Via: SIP/2.0/UDP 172.18.106.66:5060;branch=z9hG4bK1128
From: "Caller1" <sip:201@172.18.106.66>;tag=2FA6AE58-20E5
To: <sip:18900@172.18.106.88>
Date: Thu, 24 Jun 2004 21:48:40 GMT
Call-ID: 16EEB21C-C55F11D8-BF05DA50-D3EDBBC9@172.18.106.66
Supported: 100rel,timer
Min-SE: 1800
Cisco-Guid: 384701940-3311342040-3204635216-3555572681
User-Agent: Cisco-SIPGateway/IOS-12.x
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE,
NOTIFY, INFO, UPDATE, REGISTER
CSeq: 101 INVITE
Max-Forwards: 6
```

トレースメモリバッファをクリアするには、**clear trace** コマンドを発行します。ほとんどのトラブルシューティング状況で、収集するトレースを設定し、**clear trace** コマンドを発行してバッファをクリアし、トレースの収集対象となるアクションを再作成すると、**show trace buffer long** コマンドの出力をキャプチャできます。この方法は、繰り返し発生する問題のトレースを収集するのに最も効果的です。

保存されたトレースログファイル

NM および AIM では (イネーブルの場合)、トレースは `atrace.log` ファイルに書き込まれます。**show logs** コマンドを実行すると、使用可能なログファイルすべてが表示されます。

```
vnt-nm-cue>show logs
dmesg
syslog.log
atrace.log
atrace.log.prev
klog.log
messages.log
messages.log.prev
root_javacore828.1087272313.txt
tomcat_javacore1094.1087272313.txt
workflow_javacore1096.1087272313.txt
```

重要なファイルは、`atrace.log` ファイルと `messages.log` ファイルです。`messages.log` ファイルには、システムメッセージがすべて格納されています (AIM の場合、「Fatal」および「Error」メッセージのみ)。特に AIM の場合、`messages.log` ファイルが、履歴情報を含む唯一のログファイルである場合があります。`_javacore` ファイルはシステムの再起動時に書き込まれますが、通常は他のファイル (`dmesg`、`syslog.log`、`klog.log`) ほど重要ではありません。`atrace.log.prev` ファイルと `messages.log.prev` ファイルも重要です (存在する場合)。これらは `atrace.log` ファイルと `messages.log` ファイルの古いバージョンです。たとえば、`atrace.log` ファイルがいっぱいになると、そのファイルが `atrace.log.prev` にコピーされ、新しい `atrace.log` ファイルが作成されます。元の `atrace.log.prev` は置換され、その情報は失われます。

各ファイルは、FTP サーバに個別にコピーされる必要があります。

`atrace.log` ファイルは大きなファイルになる可能性があるため (NM で最大 100 MB、AIM で最大 10 MB)、通常は FTP サーバにコピーします。このような場合、**copy log** コマンドを使用します。この例では、FTP のユーザ名 (`jdoe`) とパスワード (`mypass`) が URL の一部になっています。


```
vnt-nm-cue>copy log atrace.log url ftp://jdoe:mypass@172.18.106.10/cue/atrace.log
```

% Total	% Received	% Xferd	Average Speed	Time	Curr.
			Dload Upload Total	Current	Left
100	1387k	0	0	100	1387k
			0	4476k	0:00:00
			0:00:00	0:00:00	0:00:00
					6104k

注： atrace.logファイルはプレーンテキストファイルではありません。診断するには、シスコテクニカル サポートに送信する必要があります。

Cisco Unity Express モジュール自体から、保存されたログ ファイルを表示することもできます。ただし、atrace.log ファイルについてはファイルがプレーン テキストに正しく変換されないため、これは推奨されません。messages.log ファイルを使用する例を示します。

```
cue-3660-41a#show log name messages.log
```

```
#!/bin/cat
```

```
19:46:08 logmgr: BEGIN FILE
```

```
19:46:08 logmgr: START
```

```
<45>Feb 26 19:46:08 localhost syslog-ng[134]: syslog-ng version 1.6.0rc1 starting
```

```
<197>Feb 26 19:46:08 localhost syslog-ng: INFO startup.sync syslog-ng arrived phase online
```

```
<197>Feb 26 19:46:10 localhost err_handler: INFO Recovery Recovery startup :CUE Recovery Script started.
```

```
<197>Feb 26 19:46:10 localhost err_handler: INFO Recovery Recovery LDAPVerify Verifying LDAP integrity
```

```
...
```

注： show log nameコマンドを使用してログファイルを表示すると、Control-Cキーの組み合わせを押して、コマンド出力を中断します。その後プロンプトに戻るまでに数秒かかることに注意してください。

Cisco Unity Express に保存されている atrace.log ファイルに関しては、show trace store コマンドを発行します (atrace.log.prev ファイルの場合は show trace store-prev コマンド)。

```
vnt-nm-cue>show trace store ?
```

```
<cr>
```

```
containing Only display events matching a regex pattern
```

```
long Show long format
```

```
short Show short format
```

```
tail Wait for events and print them as they occur
```

```
| Pipe output to another command
```

```
vnt-nm-cue>show trace store long paged
```

```
236 02/26 14:46:24.029 TRAC TIMZ 0 UTC UTC 0
```

```
236 02/26 14:46:24.031 TRAC TIMZ 0 UTC UTC 0
```

```
885 06/04 13:14:40.811 WFSP MISC 0 WFSysdbLimits::WFSysdbLimits hwModuleType=NM
```

```
885 06/04 13:14:40.812 WFSP MISC 0 WFSysdbProp::getProp
```

```
885 06/04 13:14:40.812 WFSP MISC 0 keyName = limitsDir
```

```
str = /sw/apps/wf/ccnapps/limits
```

```
885 06/04 13:14:40.819 WFSP MISC 0 WFSysdbProp::getNodeXml
```

```
885 06/04 13:14:40.819 WFSP MISC 0 WFSysdbProp::getProp
```

```
885 06/04 13:14:40.820 WFSP MISC 0 keyName = limits
```

```
str =
```

```
885 06/04 13:14:40.822 WFSP MISC 0 WFSysdbProp::getNodeXml(str, str)
```

```
885 06/04 13:14:40.822 WFSP MISC 0 WFSysdbProp::getProp
```

```
885 06/04 13:14:40.822 WFSP MISC 0 keyName = app
```

```
str =
```

メモリ内のトレース バッファを表示する場合は、long 形式が重要です。show trace store long コマンドを発行します。これは、atrace.log ファイルの最初からの情報で、NM の場合は最大 100 MB、AIM の場合は最大 10 MB になります。この状況では、特定のイベントを検索する必要がある場合、containing キーワードを使用すると便利なことがあります。

注：AIMのatrace.logファイルが最大サイズに達すると、ログファイルへのトレースの記録が停止します。トレースのロギングを再び開始するには、次のコマンドを実行します。

```
VNT-AIM-CUE1>configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
VNT-AIM-CUE1(config)>log trace local disable
VNT-AIM-CUE1(config)>log trace local enable
```

注：これらのコマンドは古いatrace.logファイルを削除し、新しいファイルを開始します。

FTP サーバへのトレース

大量のデータをトレースするのに最適なのは、特に AIM の場合、情報を直接 FTP サーバに記録する方法です。オフライントレースも、パフォーマンスへの影響を最小限に抑えることができます。コンフィギュレーションモードからこれを行えます。

注：Cisco Unity ExpressシステムがAIMの場合、次のコマンドが必要です（ネットワークモジュールはデフォルトで情報レベル以上を記録します）。

```
vnt-nm-cue(config)>log console info
```

注：このコマンドは、スペースの制約により2行目に表示されています。

```
vnt-nm-cue(config)>log trace server url
ftp//172.18.106.10/path/ username jdoe password mypass
```

注：Cisco Unity Expressバージョン7.xを使用している場合は、前のコマンドをlog trace server url "ftp//172.18.106.10/path/" username jdoe password mypassとして使用します。

注：ログをFTPサーバに送信する場合は、ログトレースサーバを有効にするも設定する必要があります。

```
vnt-nm-cue(config)>log trace server enable
```

注：システムは、FTPサーバの指定パスにファイルを生成します。指定したディレクトリにファイルを作成し変更する権限が必要です。また、そのディレクトリが存在している必要があります。パーサーは、設定ファイル自体に暗号化されているユーザ名とパスワードを抽出します（show running-config）。

注：FTPサーバに記録されたトレースファイルはプレーンテキストファイルではありません。診断するには、シスコテクニカルサポートに送信する必要があります。

JTAPI トレース

JTAPI トレースは、Cisco Unity Express の他のトレース機能とは別個のものです。これらは、Cisco CallManager 環境にのみ適用されます。現在有効になっている JTAPI トレースを表示するには、show ccn trace jtapi コマンドを発行します。

注：デフォルトでは、すべてのJTAPIトレースが無効になっています。

```
VNT-AIM-CUE1>show ccn trace jtapi
Warning: 0
Informational: 0
Jtapi Debugging: 0
Jtapi Implementation: 0
CTI Debugging: 0
CTI Implementation: 0
Protocol Debugging: 0
Misc Debugging: 0
```

すべてのトレースを有効にするには、次のコマンドを発行します。

```
VNT-AIM-CUE1>ccn trace jtapi debug all
You will have to reload the system for your changes to take effect
VNT-AIM-CUE1>ccn trace jtapi informational all
You will have to reload the system for your changes to take effect
VNT-AIM-CUE1>ccn trace jtapi warning all
You will have to reload the system for your changes to take effect
VNT-AIM-CUE1>show ccn trace jtapi
Warning: 1
Informational: 1
Jtapi Debugging: 1
Jtapi Implementation: 1
CTI Debugging: 1
CTI Implementation: 1
Protocol Debugging: 1
Misc Debugging: 1
```

システムをリロードします。後でこれを無効にする場合も、ここに示したのと同じ **ccn trace** コマンドを発行します。ただし、それぞれのコマンドの前に *no* というキーワードを追加します。たとえば、**no ccn trace jtapi debug all** のように入力します。AIM では特にこれが重要になります。この手順を実行しないと、パフォーマンスの低下や、AIM のコンパクト フラッシュ カードの寿命が短くなる可能性があります。

リロード後、CiscoJtapi1.log ファイルの書き込みが開始され、最初のファイルが最大サイズに達すると CiscoJtapi2.log ファイルの書き込みが開始されます。

Cisco Unity Express でこれらのログを表示するには、**show log name CiscoJtapi1.log** コマンドを発行します。ログ ファイルを FTP サーバにコピーしてオフラインで情報を表示する場合は、**copy log CiscoJtapi1.log url ftp://user:passwd@ftpservipaddr/** コマンドを発行します。

トレースの電源オフ

no trace module entity activity CLI コマンドを使用すると、トレースの電源をオフにできます。自信がなければ、**no trace all** 発行してすべての電源をオフにすることもできます。

コンフィギュレーション モードで **no log trace local enable** コマンドを使用すると、トレースの設定はそのまま、トレース ファイルへの書き込みのみ無効にできます。AIM の場合、過剰に書き込みを実行すると内蔵フラッシュ カードの寿命が短くなるため、この方法をお勧めします。以下が一例です。

```
vnt-nm-cue>configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

```
vnt-nm-cue(config)>no log trace local enable
vnt-nm-cue(config)>
```

次のコマンドを発行して、FTP サーバへのトレースを無効にします。

```
vnt-nm-cue>configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
vnt-nm-cue(config)>log trace server disable
vnt-nm-cue(config)>
```

デフォルトトレースの最有効化

特定の問題をトラブルシューティングする場合、通常、特定のトレースのみを有効にするのが適切です。トラブルシューティングが完了したら、デフォルトのトレース設定を再度有効にすることが通常は推奨されます。これを行うには、**no trace all** コマンドですべてのトレースを無効に**しません**。次に、以下のコマンドを Cisco Unity Express の CLI (コンフィギュレーション モード以外) に貼り付けて、デフォルトのトレースを有効にします。

```
trace ccn engine debug
trace ccn libldap debug
trace ccn subsystemappl debug
trace ccn managerappl debug
trace ccn managerchannel debug
trace ccn subsystemjtapi debug
trace ccn subsystemsip debug
trace ccn stacksip debug
trace ccn subsystemhttp debug
trace ccn vbrowsercore debug
trace ccn subsystemcmt debug
trace ccn libmedia debug
trace ccn managercontact debug
trace ccn stepcall debug
trace ccn stepmedia debug
trace config-ccn sip-subsystem debug
trace config-ccn jtapi-subsystem debug
trace config-ccn sip-trigger debug
trace config-ccn jtapi-trigger debug
trace config-ccn http-trigger debug
trace config-ccn group debug
trace config-ccn application debug
trace config-ccn script debug
trace config-ccn prompt debug
trace config-ccn miscellaneous debug
trace voicemail database query
trace voicemail database results
trace voicemail database transaction
trace voicemail database connection
trace voicemail database execute
trace voicemail mailbox login
trace voicemail mailbox logout
trace voicemail mailbox send
trace voicemail mailbox save
trace voicemail mailbox receive
trace voicemail mailbox delete
trace voicemail message create
trace voicemail message dec
trace voicemail message delete
trace voicemail message get
trace voicemail message inc
```

```
trace webinterface initwizard init
```

関連情報

- [音声に関する技術サポート](#)
- [音声とユニファイド コミュニケーションに関する製品サポート](#)
- [Cisco IP Telephony のトラブルシューティング](#)
- [テクニカル サポートとドキュメント - Cisco Systems](#)