

**IBM PowerHA SystemMirror for AIX  
Standard Edition**  
バージョン 7.2.2

**PowerHA SystemMirror の  
トラブルシューティング**

**IBM**



**IBM PowerHA SystemMirror for AIX  
Standard Edition**  
バージョン 7.2.2

**PowerHA SystemMirror の  
トラブルシューティング**

**IBM**

注記

本書および本書で紹介する製品をご使用になる前に、103 ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM PowerHA SystemMirror 7.2.2 Standard Edition for AIX および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： IBM PowerHA SystemMirror for AIX  
Standard Edition  
Version 7.2.2  
Troubleshooting PowerHA  
SystemMirror

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

© Copyright IBM Corporation 2017.

# 目次

本書について . . . . .	v
強調表示 . . . . .	v
AIX での大/小文字の区別 . . . . .	v
ISO 9000 . . . . .	v
関連情報 . . . . .	v
<b>PowerHA SystemMirror トラブルシュー ティング . . . . .</b>	<b>1</b>
PowerHA SystemMirror クラスターのトラブルシュー ティング . . . . .	1
問題の認識 . . . . .	2
問題の原因の特定 . . . . .	3
クラスター・マネージャーの停止 . . . . .	3
AIX データ収集ユーティリティの使用 . . . . .	3
PowerHA SystemMirror 診断ユーティリティの 使用 . . . . .	4
予想される動作の検証 . . . . .	5
問題判別ツール . . . . .	5
サンプル・ユーザー定義スクリプト . . . . .	10
クラスター・ログ・ファイルの使用 . . . . .	12
システム・コンポーネント . . . . .	36
システム・コンポーネントの調査 . . . . .	36
高可用性アプリケーションの確認 . . . . .	37
PowerHA SystemMirror 層の検査 . . . . .	37

論理ボリューム・マネージャーの確認 . . . . .	43
TCP/IP サブシステムの検査 . . . . .	49
AIX オペレーティング・システムの検査 . . . . .	53
物理ネットワークの検査 . . . . .	53
ディスクおよびディスク・アダプターの検査 . . . . .	53
クラスター通信デーモンの検査 . . . . .	54
システム・ハードウェアの検査 . . . . .	54
PowerHA SystemMirror のインストールの問題	55
共通問題の解決 . . . . .	56
PowerHA SystemMirror の始動の問題 . . . . .	56
ディスクとファイルシステムの問題 . . . . .	62
ネットワークとスイッチの問題 . . . . .	72
クラスター通信の問題 . . . . .	78
PowerHA SystemMirror のテークオーバーの問 題 . . . . .	79
クライアントの問題 . . . . .	84
その他の問題 . . . . .	86

<b>特記事項 . . . . .</b>	<b>103</b>
プライバシー・ポリシーに関する考慮事項 . . . . .	105
商標 . . . . .	105

<b>索引 . . . . .</b>	<b>107</b>
---------------------	------------



---

## 本書について

本書では、PowerHA® SystemMirror® for AIX® ソフトウェアのトラブルシューティングについて紹介します。この情報は、オペレーティング・システム付属の文書 CD にも収録されています。

---

## 強調表示

本書では、以下の強調表示規則を使用します。

太字	システムによって名前が事前に定義されているコマンド、サブルーチン、キーワード、ファイル、構造、ディレクトリー、およびその他の項目を示します。また、ユーザーが選択するボタン、ラベル、アイコンなどのグラフィカル・オブジェクトも示します。
イタリック	実際の名前または値をユーザーが指定する必要があるパラメーターを示します。
モノスペース	特定のデータ値の例、画面に表示されるものと同様のテキスト例、プログラマーが作成するものと同様のプログラム・コード部分の例、システムからのメッセージ、実際に入力する必要がある情報などを示します。

---

## AIX での大/小文字の区別

AIX オペレーティング・システムは、すべてケース・センシティブとなっています。これは、英大文字と小文字が区別されるということです。例えば、**ls** コマンドを使用するとファイルをリスト表示できます。LS と入力した場合、そのようなコマンドはないという応答がシステムから返ってきます。同様に、**FILEA**、**FiLea**、および **filea** は、同じディレクトリーにある場合でも、3 つの異なるファイル名です。予期しない処理が実行されないように、常に正しい大/小文字を使用するようにしてください。

---

## ISO 9000

当製品の開発および製造には、ISO 9000 登録品質システムが使用されました。

---

## 関連情報

- PowerHA SystemMirror バージョン 7.2.2 for AIX PDF 資料は、『PowerHA SystemMirror 7.2.2 の PDF』のトピックで入手可能です。
- PowerHA SystemMirror バージョン 7.2.2 for AIX リリース・ノートは、『PowerHA SystemMirror 7.2.2 リリース・ノート』のトピックで入手可能です。



---

## PowerHA SystemMirror トラブルシューティング

本書は PowerHA SystemMirror ソフトウェア (AIX オペレーティング・システム用) のトラブルシューティングに使用します。

関連情報:

Administering PowerHA SystemMirror

PowerHA SystemMirror プランニング・ガイド

PowerHA SystemMirror インストール・ガイド

---

## PowerHA SystemMirror クラスターのトラブルシューティング

以下のセクションでは、PowerHA SystemMirror クラスターのトラブルシューティングについて推奨される戦略を説明します。また、PowerHA SystemMirror メイン SMIT メニューから使用できる問題判別ツールを示します。さらに、クラスターのパフォーマンスを最適にするための調整についても説明します。これにより、共通問題のいくつかを回避することができます。

通常、機能している PowerHA SystemMirror クラスターに介入する必要はほとんどありません。それでも問題が発生した場合は、診断とリカバリーのスキルが重要になります。そのため、トラブルシューティングでは、問題を素早く特定し、PowerHA SystemMirror ソフトウェアに関する知識を応用してクラスターをフル稼働まで復元させる必要があります。

一般に、PowerHA SystemMirror クラスターのトラブルシューティングには以下のステップが含まれます。

- 問題の存在の認識
- 問題の原因の特定
- 問題の修正

注: 以下のトピックでは、ログ・ファイルのデフォルト・ロケーションが示されます。ログをリダイレクトしている場合は、該当するロケーションを確認してください。

関連概念:

12 ページの『クラスター・ログ・ファイルの使用』

以下のトピックでは、PowerHA SystemMirror クラスター・ログ・ファイルを使用したクラスターのトラブルシューティング方法について説明します。一部のログについては、パラメーターの管理に関するセクションもいくつか含まれています。

56 ページの『共通問題の解決』

このセクションでは、よくある問題と、推奨事項について説明します。

関連資料:

36 ページの『システム・コンポーネント』

以下のトピックでは、システム・コンポーネントを調査する手順を示し、PowerHA SystemMirror の使用時に発生する可能性がある問題を特定し、考えられる解決策を提示します。

## 問題の認識

PowerHA SystemMirror クラスター内で問題が発生した場合、イベント通知アラートにより、あるいは `errpt` ファイルまたは `hacmp.out` ファイルをモニターすることにより、問題発生を知ることができます。

また、クラスターに関する問題は、メール通知、ページャー通知またはテキスト・メッセージングから認識される場合も多くあります。

- メール通知。 PowerHA SystemMirror 標準コンポーネントでは、問題が発生した場合にシステム管理者にメールは送信されませんが、イベント・スクリプトの実行前または実行後に実行するイベント前処理スクリプトまたはイベント後処理スクリプトとして、メール通知メソッドを作成することができます。 PowerHA SystemMirror クラスター環境では、メール通知が効果的な方法として推奨されます。
- リモート通知。 SMIT インターフェースを使って、クラスター・イベントへの応答をカスタマイズして発行する通知メソッド (携帯電話を含む任意のアドレスへの数値ページ、英数字ページ、またはテキスト・メッセージ通知) を定義することもできます。
  - ページャー通知。 指定のイベントが発生したときに、ページャー番号にメッセージを送信できます。 テキスト表示 (英数字ページ) をサポートするページャーにはテキスト情報を、数値のみを表示するページャーには数値メッセージを送信できます。
  - テキスト・メッセージング。 標準 Telocator Alphanumeric Protocol (TAP) によって、標準データ・モデムと固定電話を使用してテキスト・メッセージを携帯電話に送信できます。 プロバイダーがこのサービスをサポートしている必要があります。

また、Falcom-compatible GSM モデムを使用して、SMS (Short Message Service) テキスト・メッセージ通知を無線で送信することもできます。 SMS メッセージには SMS サービス・プロバイダーのアカウントが必要です。 GSM モデムは、RS232 回線あるいは USB 回線を介して TAP モデム・プロトコルを入力として受信し、プロバイダーの携帯電話局にメッセージを無線で送信します。 プロバイダーはメッセージを宛先の携帯電話に転送します。 各プロバイダーはショート・メッセージ・サービス・センター (SMSC) を持っています。

個々のユーザーについて、すべてのイベントとノードを含むリモート通知メソッドを定義できるので、応答者の変更されても 1 ユニットとして通知メソッドを切り替えることができます。

注: 手動で各メッセージ・ファイルを各ノードに配信してください。 PowerHA SystemMirror は、同期中にファイル・コレクション・ユーティリティーが特別に設定されていない限り、自動的にファイルを他のノードには配信しません。

## システム・コンソールに表示されるメッセージ

PowerHA SystemMirror システムによって (クラスター・イベントに応じて) 実行されたスクリプトが開始、停止、またはエラー条件を検出すると、記述メッセージが生成されます。 また、PowerHA SystemMirror クラスターを構成するデーモンにより、開始された時点、停止した時点、エラー状態を検出した時点、または状態を変更した時点でも、デーモンによりメッセージが生成されます。 PowerHA SystemMirror システムは、これらのメッセージをシステム・コンソールおよび 1 つ以上のクラスター・ログ・ファイルに書き込みます。 `errpt` ファイルなどの関連システム・ファイルにもエラーが記録される場合があります。

関連概念:

12 ページの『クラスター・ログ・ファイルの使用』

以下のトピックでは、PowerHA SystemMirror クラスター・ログ・ファイルを使用したクラスターのトラブルシューティング方法について説明します。一部のログについては、パラメーターの管理に関するセクションもいくつか含まれています。

関連情報:

PowerHA SystemMirror プランニング・ガイド

PowerHA SystemMirror クラスターの検査および同期化

## 問題の原因の特定

問題があることが判明したら、問題の原因を見つける必要があります。

PowerHA SystemMirror について問題が検出された場合は、最初に問題を分析するために、以下のアクションを実行してください。

1. **snap -e** コマンドで PowerHA SystemMirror スナップショットを収集します。これは、できるだけ問題が検出された直後に実行してください。収集されるログ・ファイルにはエラーの 1 期間を保持しているためです。
2. `/usr/es/sbin/cluster/clstat` コマンドおよび `/usr/es/sbin/cluster/utilities/clRGinfo` コマンドを使用して、クラスターとリソース・グループの状態を明確にします。
3. イベント・エラーが発生した場合は、`/var/hacmp/log/hacmp.out` ファイルを検査してエラーを検索します。AIX コマンドが失敗した場合は、**snap** コマンドを使用して、対応する AIX コンポーネントのより詳細なデバッグ・データを積極的に収集してください。PowerHA SystemMirror のより詳細な問題の判別のために最もよく要求されるフラグは、**snap -egGtL** です。
4. 最新のクラスター検証の結果を、`/var/hacmp/log/clverify.log` ファイルと `/var/hacmp/log/autoverify.log` ファイルで調べます。クラスター検証を実行します。
5. C-SPOC コマンドが失敗した場合は、`/var/hacmp/log/cspoc.log.long` ファイルを調べます。
6. ノード間のネットワーク接続を検証します。
7. エラーが障害の時間ウィンドウのログに記録されている場合は、エラー・ログ (`errprt -a`) を検査して設定します。

## クラスター・マネージャーの停止

クラスターの問題を修正するには、障害のあるノードでクラスター・マネージャーを停止し、共用リソースを残りのノードにテークオーバーさせる必要があります。

「`unmanage resource groups` (リソース・グループの管理解除)」オプションでクラスター・サービスを停止した後で、クラスター・マネージャー処理を停止することもできます。このオプションはリソースをアクティブのままにしますが、ノード上ではモニターできません。停止させたら、トラブルシューティングの手順を開始できます。

他の手順がすべて失敗した場合は、すべてのクラスター・ノード上で PowerHA SystemMirror クラスター・サービスを停止させます。次に、PowerHA SystemMirror クラスター・イベント・スクリプトが開始しようとしていたアプリケーションを手動で開始し、PowerHA SystemMirror ソフトウェアなしでアプリケーションを実行します。このとき、ボリューム・グループの検証、ファイル・システムのマウント、IP アドレスを使用可能にすることが必要になる場合があります。全クラスター・ノード上で PowerHA SystemMirror クラスター・サービスを停止させた状態で、最初の問題の原因となった条件を修正します。

## AIX データ収集ユーティリティーの使用

AIX **snap** コマンドを使用して、データを PowerHA SystemMirror クラスターから収集します。

**-e** フラグは、PowerHA SystemMirror および他のコンポーネントと PowerHA SystemMirror による他のコンポーネントとの対話についての問題をトラブルシューティングする際に、IBM® サポートに役立つデ

ータを収集します。特に、**-e** フラグは、PowerHA SystemMirror ユーティリティーのすべてのログ・ファイル、PowerHA SystemMirror によって保守される ODM、いくつかの AIX ODM、最もよく必要とされる AIX 構成データ (LVM、TCP/IP、installp 情報など) を収集します。 **snap -e** コマンドは /usr/sbin/rsct/bin/ctsnap を実行し、ctsnap はグループ・サービスのデータを収集します。

PowerHA SystemMirror スナップショットは、できる限り PowerHA SystemMirror で問題が発生した直後に収集して、エラーの時間ウィンドウに関するデータが確実にログ・ファイルに組み込まれるようにしてください。

**snap -e** コマンドは、クラスター通信デーモン・サブシステム (clcomd) に依存して、データを収集します。このサブシステムがエラーによって影響を受けると、**snap -e** コマンドは失敗する場合があります。この場合は、すべてのクラスター・ノードの以下のデータを収集してください。

- ディレクトリー /var/hacmp の tar アーカイブ
- ディレクトリー /etc/es/objrepos および /usr/es/sbin/cluster/etc/objrepos/active の tar アーカイブ
- **snap -cfgGLt**

## PowerHA SystemMirror 診断ユーティリティーの使用

PowerHA SystemMirror および AIX は多くの診断ツールを提供します。

(クラスター・ログとメッセージに加えて) PowerHA SystemMirror 診断ツールには次が含まれます。

- **clRGinfo** は、リソース・グループについての情報およびトラブルシューティングのための情報を提供します。
- **clstat** は、キーになるクラスター・コンポーネント (クラスター自体、クラスターのノード、ノードに接続されたネットワーク・インターフェース、サービス・ラベル、および各ノード上のリソース・グループ) の状況を報告します。
- **cldisp** ユーティリティーは、リソース・グループとその開始、フォールオーバー、フォールバックの各ポリシーを表示します。
- SMIT の問題判別ツール。詳しくは、セクション『問題判別ツール』を参照してください。

### クラスター・スナップショット・ユーティリティーによるクラスター構成の確認

ログを収集する場合は、SMIT でログを収集するよう指定することもできます。ログの収集をスキップすると、スナップショットのサイズを縮小し、スナップショット・ユーティリティーの実行時間を減少できます。

### SMIT 問題判別ツールの操作

SMIT 「**Problem Determination Tools** (問題判別ツール)」メニューには、クラスター・スナップショット・ユーティリティーによって提供されるオプションが含まれ、問題の診断および解決にも役立ちます。

関連概念:

5 ページの『問題判別ツール』

SMIT インターフェースを使用すると、PowerHA SystemMirror の問題のトラブルシューティングに役立つことがあります。

関連情報:

PowerHA SystemMirror クラスターのモニター  
クラスター構成の保管および復元

## 予想される動作の検証

可用性の高いアプリケーションが稼働している場合は、ユーザーがそれらのアプリケーションにアクセスできることを確認します。

アプリケーションが稼働していない場合は、他の部分を調べて、クラスターに影響を与えている問題を特定する必要があります。本書では、潜在的な問題を見つけることができる方法を説明します。

## 問題判別ツール

SMIT インターフェースを使用すると、PowerHA SystemMirror の問題のトラブルシューティングに役立つことがあります。

以下のツールを PowerHA SystemMirror のトラブルシューティングに使用できます。これらのツールにアクセスするには、コマンド行に `smit sysmirror` と入力し、「問題判別ツール (Problem Determination Tools)」を選択します。

### PowerHA SystemMirror Verification (検証)

このツールを使用すると、すべてのノード上の構成が同期化されていることの確認、ユーザー定義検証メソッドのセットアップ、または自動クラスター検証のセットアップを行うことができます。

### View Current State (現在の状態の表示)

このツールを使用すると、ノード、通信インターフェース、リソース・グループの状態、および最終の 5 つのイベントのローカル・イベント要約を表示できます。

### PowerHA SystemMirror Log Viewing and Management (ログの表示および管理)

このツールを使用すると、ログ・ファイルに関連したユーティリティーのリストを表示できます。

### Recover from PowerHA SystemMirror Script Failure (スクリプト障害からの回復)

このツールを使用すると、スクリプト障害から回復できます。

### Restore PowerHA SystemMirror Configuration Database from Active Configuration (アクティブ構成からの構成データベースの復元)

このツールを使用すると、構成データベースに加えたどの変更でも、スナップショットとしてパス `/usr/es/sbin/cluster/snapshots/UserModifiedDB` に自動的に保存できます。それらの変更は、クラスター・マネージャーで実際に使用される値で構成データベースを復元する前に、保存する必要があります。

### Release Locks Set By Dynamic Reconfiguration (動的再構成によって設定されたロックの解除)

このツールを使用すると、動的再構成中に使用されたロックを解除できます。アクティブ・クラスターで構成変更を行うと、それらの変更をアクティブ構成にコミットする前に、すべてのノードに変更を配布するマルチステップの処理が行われます。この処理の各段階で、更新処理の同期化のためにソフトウェアの「ロック」が行われます。この更新のいずれかの時点で障害が発生した場合、ロックがそのまま解除されない可能性があります。そのような場合、さらに変更を行うにはまずロックを解除することが必要になります。

### Cluster Test Tool (クラスター・テスト・ツール)

このツールを使用すると、新しいクラスターを実稼働環境に入れる前にそのクラスターの回復手順をテストできます。また、既存のクラスターがサービス中でない場合は、その既存クラスターへの変更をこのツールを使用してテストすることもできます。

### PowerHA SystemMirror Trace Facility (トレース機能)

このツールを使用すると、PowerHA SystemMirror デーモンをトレースできます。

### PowerHA SystemMirror Error Notification (エラー通知)

このツールを使用すると、エラー通知を作成できます。

### AIX Tracing for Cluster Resources (クラスター・リソースの AIX トレース)

このツールを使用すると、イベント・スクリプトの実行時にクラスター・リソースの AIX トレース・データを収集できます。

### Compare Active and Default Configurations (アクティブ構成とデフォルト構成の比較)

このツールを使用すると、変更をアクティブ構成に取り込む前に、デフォルト構成内で変更を比較して識別できます。

### Replace the Primary Repository Disk (1 次リポジトリ・ディスクの交換)

このツールを使用すると、クラスター・リポジトリに使用するディスクを交換できます。

### Open a SMIT Session on a Node (ノード上で SMIT セッションを開く)

このツールを使用すると、SMIT 内からリモート・ノード上の SMIT セッションを開くことができます。

関連情報:

動的再構成の問題および同期化

PowerHA SystemMirror クラスターの検査および同期化

エラー通知のタイプ

## PowerHA SystemMirror の検証

「**Problem Determination Tools** (問題判別ツール)」メニューからこのオプションを選択すると、すべてのノード上の構成が同期化されていることの確認、ユーザー定義検証メソッドのセットアップ、または自動クラスター検証のセットアップができます。

表 1. 問題判別ツールのフィールド

フィールド	説明
Verify PowerHA SystemMirror Configuration (構成の検証)	このオプションを選択してクラスター・トポロジー・リソースを検証します。
Configure Custom Verification Method (ユーザー定義検証メソッドの構成)	このオプションを使用して、ユーザー定義検証メソッドの追加、表示、除去を行います。
Automatic Cluster Configuration Monitoring (自動クラスター構成モニター)	このオプションを選択すると 24 時間ごとにクラスターを自動的に検証でき、クラスター全体の結果が報告されます。

### PowerHA SystemMirror 構成の検証:

クラスター・トポロジー・リソースおよびカスタム定義の検証メソッドを検証できます。

PowerHA SystemMirror 構成を検証するには、以下のステップを実行します。

1. コマンド行から `smit sysmirror` と入力します。
2. SMIT で、「**Problem Determination Tools (問題判別ツール)**」 > 「**PowerHA SystemMirror Verification (検証)**」 > 「**Verifying PowerHA SystemMirror Configuration (構成の検証)**」を選択し、Enter キーを押します。
3. 以下のフィールド値を入力します。

表 2. PowerHA SystemMirror 構成フィールド

フィールド	値
PowerHA SystemMirror の検証メソッド	デフォルトでは、事前インストール・プログラムが、PowerHA SystemMirrorに付属のすべての検証メソッドを実行します。このフィールドを選択してすべての事前インストール・プログラムを実行するか、「none (なし)」を選択して先に定義したユーザー定義検証メソッドを指定することもできます。
Custom Defined Verification Method (ユーザー定義検査メソッド)	ユーザー定義検証メソッドの名前を入力します。 F4 を押して、以前に定義した検証メソッドのリストを表示することもできます。 どのメソッドも選択せず、「Base PowerHA SystemMirror Verification Method (基本 PowerHA SystemMirror 検証メソッド)」フィールドで「none (なし)」を選択すると、デフォルトでは「検証と同期」は基本検証メソッドを確認せずにエラー・メッセージを表示します。 メソッドの実行順序は、検証メソッドがリストされている順序で決まります。この実行順序は、後続の検証でも同じであり、別のメソッドが選択されるまで変わりません。「All (すべて)」を選択した場合、カスタム定義のメソッドがすべて検証されます。
Error Count (エラー・カウント)	デフォルトでは、PowerHA SystemMirror 構成の検証はエラーの全リストを発行するためにエラー発生後も実行し続けます。 エラーの数在一定数に達した時点でプログラムをキャンセルする場合は、このフィールドにエラー数を入力します。
Log File to store output (出力を保管するためのログ・ファイル)	検証の出力を保管する出力ファイルの名前を入力します。 デフォルトでは、検証出力も <code>/var/hacmp/clverify/clverify.log</code> ファイルに保管されます。
Verify Changes Only? (変更のみを検証する)	現在のクラスター構成に適用するすべての検証を実行するには、「no (いいえ)」を選択します。 変更された PowerHA SystemMirror 構成部分に関連する検査だけを実行するには、「yes (はい)」を選択します。「yes (はい)」を選択した場合、非アクティブ・クラスターは対象になりません。 注: 「yes (はい)」オプションが関係するのは、クラスター構成データベースのみです。 クラスター・ノードの AIX 構成を変更した場合には、「no (いいえ)」を選択してください。 AIX 構成を変更していない場合のみ、「yes (はい)」を選択してください。
Logging (ロギング)	「on (オン)」を選択すると、通常 <code>/var/hacmp/clverify/clverify.log</code> に送られるすべての出力がコンソールに表示されます。デフォルトは「off (オフ)」です。

#### クラスター構成の自動モニターおよび検証:

クラスターの検証ユーティリティーは、ユーザーが選択可能な 1 つの PowerHA SystemMirror クラスター・ノード上で 24 時間ごとに実行されます。

デフォルトでは、アルファベット順で最初のノードによって、真夜中に検証が実行されます。 検証中、今後どこかで問題を発生させる可能性があるエラーがあれば、すべて表示されます。 ノードおよび構成に適した時間を選択して、デフォルトを変更できます。

選択されたノードが使用できない (電源が切られている) 場合、検証は自動モニターを実行しません。 選択されたクラスター・ノードでクラスター検証が完了すると、このノードによって他のクラスター・ノードに次の検証情報が通知されます。

- 検証が実行されたノード名
- 最新検証の日付と時間
- 検証結果

この情報は PowerHA SystemMirror ログ・ファイル `/var/hacmp/log/clutils.log` に使用可能なクラスター・ノード別に格納されます。 選択されたノードが使用できない場合や、クラスター検証が完了できなかったことは、`/var/hacmp/log/clutils.log` ファイルにレポートが無いことによって検出できます。

クラスター検証が完了し、構成エラーが検出された場合は、次のような潜在的な問題について通知されます。

- クラスターの検証の終了状況が、クラスター検証プロセスの完了の情報とともにクラスター全体に連絡されます。
- ブロードキャスト・メッセージがクラスター全体に送信され、**stdout** に表示されます。これらのメッセージは、ユーザーに、検出された構成エラーについて通知します。
- クラスター上で **cluster\_notify** イベントが実行され、(クラスター・サービスが実行されている場合) **hacmp.out** ログに記録されます。

さらに詳細な情報は、**/var/hacmp/clverify/clverify.log** でクラスター検証を完了したノードで入手できます。処理中に障害が発生した場合、エラー・メッセージおよび警告により、**verification** で障害が発生したノードとその理由が明確に示されます。

自動検証の構成およびクラスター構成のモニター:

ノードを構成し、クラスターの検証が自動的に実行される時刻を指定できます。

ノード上の **/var** ファイルシステムに **/var/hacmp/log/clutils.log** ファイル用の十分なスペースがあることを確認します。

ノードを構成し、クラスターの検証が自動的に実行される時間を指定するには、次の手順を実行します。

1. コマンド行から **smit sysmirror** と入力します。
2. SMIT インターフェースから「**Problem Determination Tools (問題判別ツール)**」 > 「**PowerHA SystemMirror Verification (検証)**」 > 「**Automatic Cluster Configuration Monitoring (自動クラスター構成モニター)**」を選択し、**Enter** キーを押します。
3. 以下のフィールド値を入力します。

表 3. 自動クラスター構成モニター・フィールド

フィールド	値
* Automatic cluster configuration verification (自動クラスター構成検証)	デフォルトは「 <b>Enabled (使用可能)</b> 」です。
Node name (ノード名)	リストからいずれかのクラスター・ノードを選択します。デフォルトでは、アルファベット順の最初のノードによって、クラスター構成が検証されます。このノードは、自動検証が実行されると動的に決定されます。
*HOUR (00 - 23) (時 (00 - 23))	デフォルトでは深夜 (00) になっています。検証は、24 時間ごとに選択された時間に自動的に実行されます。

4. すべてのフィールドが正しいことを確認し、**Enter** キーを押します。
5. 変更は、クラスターが同期化されると有効になります。

関連情報:

PowerHA SystemMirror クラスターのモニター

## PowerHA SystemMirror ログの表示および管理

「**Problem Determination Tools (問題判別ツール)**」メニューからこのオプションを選択して、ログ・ファイルに関連するユーティリティーの一覧を表示します。

ここから、次の内容を実行できます。

- イベント要約の表示、保存、または削除
- 詳細 PowerHA SystemMirror ログ・ファイルの表示

## 8 PowerHA SystemMirror のトラブルシューティング

- PowerHA SystemMirror ログ・ファイル・パラメーターの変更または表示
- クラスタ・マネージャーのログ・ファイル・パラメーターの変更または表示
- クラスタ・ログ・ファイルのディレクトリーの変更または表示
- すべての Cluster Logs ディレクトリーの変更
- 問題報告のためのクラスタ・ログ・ファイルの収集

関連概念:

12 ページの『クラスタ・ログ・ファイルの使用』

以下のトピックでは、PowerHA SystemMirror クラスタ・ログ・ファイルを使用したクラスタのトラブルシューティング方法について説明します。一部のログについては、パラメーターの管理に関するセクションもいくつか含まれています。

関連情報:

PowerHA SystemMirror クラスタのテスト

## PowerHA SystemMirror スクリプト障害からの回復

PowerHA SystemMirror スクリプト障害から回復するには、「**Problem Determination Tools (問題判別ツール)**」メニューからこのオプションを選択します。

例えば、ファイルシステム・マウントの失敗のためにスクリプト障害が起こった場合、問題を修正し、ファイルシステムを手動でマウントしてから、このオプションを使用して残りのクラスタ・イベント処理を完了することができます。

「**Recover From PowerHA SystemMirror Script Failure (スクリプト障害からの回復)**」メニュー・オプションにより、指定されたノードのクラスタ・マネージャー・デーモン (**clstrmgrES**) に信号が送信され、その結果、デーモンはクラスタ・イベントの次のステップに進みます。イベント障害が引き続いて起こった場合、問題修正の処理を繰り返してから、「**Recover From PowerHA SystemMirror Script Failure (スクリプト障害からの回復)**」オプションを使用して次のステップに進む必要があります。クラスタの状態が「安定」するまで、この処理を続行する必要があります。

スクリプト障害の原因となった問題が修正されたことを確認します。 イベント・スクリプトの障害後の残りのステップを手動で完了する必要があります (**/var/hacmp/log/hacmp.out** を参照してください)。 次に、クラスタリングを再開するには、次のステップを完了して、PowerHA SystemMirror イベント・スクリプトの状態を **EVENT COMPLETED** にします。

1. **smit hacmp** と入力します。
2. **SMIT** で、「**Problem Determination Tools (問題判別ツール)**」 > 「**Recover From PowerHA SystemMirror Script Failure (スクリプト障害からの回復)**」を選択します。
3. **clruncmd** コマンドを実行するノードの IP ラベル/アドレスを選択して、**Enter** を押します。回復の試行を確認するプロンプトが表示されます。 IP ラベルが **/etc/hosts** ファイルにリストされます。このラベルは、障害が発生したノードのサービス IP アドレスに割り当てられた名前です。
4. 続行するには、**Enter** キーを押してください。スクリプトの回復の成功を確認する別の **SMIT** パネルが表示されます。

## アクティブ構成からの PowerHA SystemMirror 構成データベースの復元

クラスタ・サービスが開始され、構成に変更が加えられると、これらの変更に伴って、**DCD (default configuration directory)** も変更されます。これらの変更の影響をよく検討していないことが分かり、元に戻す場合もあります。 **ACD (active configuration directory)** は修正されていないため、**DCD** の変更を元に戻すために必要な手順は、**ACD** から **DCD** を復元することだけです。

「**Problem Determination Tools (問題判別ツール)**」メニューからこのオプションを選択して、実際にクラスター・マネージャーで使用される値で構成データベースを復元する前に、`/usr/es/sbin/cluster/snapshots/UserModifiedDB` のパスで変更をスナップショットとして構成データベースに自動的に保存します。

1. `smit hacmp` と入力します。
2. SMIT で、「**Problem Determination Tools (問題判別ツール)**」>「**Restore PowerHA SystemMirror Configuration Database from Active Configuration (アクティブ構成からの PowerHA SystemMirror 構成データベースの復元)**」を選択します。

SMIT から次の質問が表示されます。

Are you Sure?

3. `Enter` を押します。

スナップショットが保管され、アクティブ構成が DCD にコピーされます。これで、構成を表示して、さらに必要な変更を行うことができます。

関連情報:

クラスター構成の保管および復元

## サンプル・ユーザー定義スクリプト

このトピックには、カスタム・スクリプトの実行が役立つシナリオがいくつか用意されており、サンプル・スクリプトが含まれています。

### cron ジョブの高可用性

PowerHA SystemMirror 環境の管理を容易にするには、特定の **cron** ジョブを現在リソースを保持しているクラスター・ノード上でのみ実行させる必要があります。

**cron** ジョブがリソースまたはアプリケーションと合わせて実行される場合、この **cron** エントリーをリソースとともにフォールオーバーさせると有効です。ノードによって関連するリソースまたはアプリケーションが処理されなくなる場合、**cron** テーブルから **cron** エントリーを除去する必要がある場合もあります。

次の例は、カスタマイズしたスクリプトを使用してこれを行う 1 つの方法を示しています。

例に挙げるクラスターは、2 ノードのホット・スタンバイ・クラスターであり、`node1` が 1 次ノードで、`node2` がバックアップ・ノードです。`node1` は通常、共用リソース・グループおよびアプリケーションを所有しています。アプリケーションは、**cron** ジョブの実行頻度が 1 日 1 回で、ただし、現在リソースを所有しているノード上でのみジョブが実行されることを要求します。

共用リソース・グループおよびアプリケーションが `node2` へフォールオーバーした場合でも、**cron** ジョブが確実に実行されるようにするには、次のようにして、2 つのファイルを作成します。

1. `root` ユーザーが **cron** ジョブを実行中である場合、**root.resource** というファイルと **root.noresource** という別のファイルを `node1` の非共用ファイル・システムのディレクトリーに作成します。これらのファイルを、ディレクトリー `/var/spool/crontabs` にある **cron** テーブルのように変更します。

**root.resource** テーブルには、通常実行されるすべてのシステム・エントリーと、共用リソースまたはアプリケーションに関連するすべてのエントリーが含まれています。

**root.noresource** テーブルには、通常実行されるすべてのシステム・エントリーが含まれますが、共用リソースまたはアプリケーションに関連するエントリーは含まれません。

2. 両方のノードに 2 つのファイルのコピーが含まれるようにファイルを他のノードにコピーします。
3. システム起動時に双方のシステムで次のコマンドを実行します。

```
crontab root.noresource
```

これにより、システム起動時に root の **cron** テーブルに入っているのは「no resource」エントリーのみとなります。

4. 2 つの方法のいずれかを使用して、*root.resource* **cron** テーブルを活動化できます。2 つの方法のうちでは、最初の方が簡単です。
  - アプリケーション開始スクリプトの最後の行として **crontab root.resource** を実行します。アプリケーションの停止スクリプトでは、最初の行は **crontab root.noresource** となります。アプリケーションの始動スクリプトと停止スクリプトでこれらのコマンドを実行することにより、これらのコマンドが正しい時期に正しいノード上でアクティブおよび非アクティブになることが保証されます。
  - *post\_event* から *node\_up\_complete* および *node\_down\_complete* までの行として **crontab** コマンドを実行します。
    - 1 次ノードの *node\_up\_complete* で、**crontab root.resources** を実行します。
    - *node\_down\_complete* で **crontab root.noresources** を実行します。
  - また、テークオーバー・ノードによって必ずイベント・ハンドラーが使用され、適切な **cron** テーブルが実行されます。テークオーバーが起こったかどうかを特定し、**crontab root.resources** コマンドを実行するために、ロジックが *node\_down\_complete* イベントに書き込まれる必要があります。x 再統合時、必ず *node\_up* に対するイベント前処理によって、1 次ノードがクラスターに戻っているかが特定され、その後、**crontab root.noresource** コマンドが実行されます。

## 印刷キューの高可用化

フォールオーバーが起きた場合、その時点でキューに入っている印刷ジョブを保管し、残った方のノードへ移動できます。

印刷スプーリング・システムは、2 つのディレクトリー (**/var/spool/qdaemon** および **/var/spool/lpd/qdir**) で構成されています。一方のディレクトリーには、各ジョブのデータ (コンテンツ) が入ったファイルが入ります。もう一方のディレクトリーには、印刷ジョブ自体に関連した情報から成るファイルが入ります。ジョブがキューに入っていると、2 つのディレクトリーのそれぞれに、ファイルが存在します。フォールオーバーが起きると、これらのディレクトリーは通常、フォールオーバーを行わないため印刷ジョブは失われます。

この問題の解決策は、共用ボリューム・グループ上に 2 つのファイル・システムを定義することです。これらのファイルシステムを **/prtjobs** および **/prtdata** と呼びます。PowerHA SystemMirror が開始されると、これらのファイルシステムは **/var/spool/lpd/qdir** および **/var/spool/qdaemon** に対してマウントされます。

この操作を、*node\_up* のイベント後処理として実行するスクリプトを作成してください。そのスクリプトは、次のことを行います。

1. 印刷キューを停止します
2. 印刷キュー・デーモンを停止します
3. **/prtjobs** を **/var/spool/lpd/qdir** に対してマウントします
4. **/prtdata** を **/var/spool/qdaemon** に対してマウントします
5. 印刷キュー・デーモンを再始動します

6. 印刷キューを再始動します  
フォールオーバーが起きた場合、残ったノードは、次のことを行います。
7. 印刷キューを停止します
8. 印刷キュー・デーモンを停止します
9. `/prtjobs` の内容を `/var/spool/lpd/qdir` に移動します
10. `/prtdata` の内容を `/var/spool/qdaemon` に移動します
11. 印刷キュー・デーモンを再始動します
12. 印刷キューを再始動します
13. 以上の内容を実行するには、イベント後処理と呼ばれるスクリプトをテークオーバー上の `node_down_complete` に書き込みます。このスクリプトは、`node_down` が 1 次ノードからのものであるかどうかを特定する内容である必要があります。

## クラスター・ログ・ファイルの使用

以下のトピックでは、PowerHA SystemMirror クラスター・ログ・ファイルを使用したクラスターのトラブルシューティング方法について説明します。一部のログについては、パラメーターの管理に関するセクションもいくつか含まれています。

### PowerHA SystemMirror クラスター・ログ・ファイルの表示

クラスターに影響する問題を診断する場合、まず、PowerHA SystemMirror サブシステムにより出力されるメッセージが記録されているクラスター・ログ・ファイルを調べる必要があります。これらのメッセージには、クラスターの現在の状態を理解するための重要な情報が含まれています。次のセクションでは、PowerHA SystemMirror ソフトウェアにより出力されるメッセージのタイプと、これらのメッセージが記録されるログ・ファイルについて説明します。

ほとんどのトラブルシューティングにおいて、`/var/hacmp/log/hacmp.out` ファイルが最も役に立つログ・ファイルになります。最近のリリースでのリソース・グループ処理の強化に伴い、`hacmp.out` ファイルが拡張され、クラスター・イベント後のリソース・グループのアクティビティとロケーションの情報を従来よりも取得できるようになりました。例えば、`hacmp.out` ファイルは他のログ (クラスター・ヒストリー・ログなど) が報告できないリソース・グループ並列処理の詳細を収集します。このログに含まれるイベントの要約を使用すると、クラスターで最新発生したイベントを容易に素早く見ることができます。

クラスター・メッセージ・ログ・ファイルの検討:

PowerHA SystemMirror ソフトウェアは、生成するメッセージをシステム・コンソールおよびいくつかのログ・ファイルに書き込みます。各ログ・ファイルには、PowerHA SystemMirror ソフトウェアにより生成されたメッセージの別々のサブセットが含まれます。ログ・ファイルをグループとして表示すると、すべてのクラスター・アクティビティの詳細情報が提供されます。

次のリストに、PowerHA SystemMirror ソフトウェアがメッセージを書き込むログ・ファイルと、そこに含まれるクラスター・メッセージのタイプを示します。また、このリストには、それぞれのログ・ファイルについての推奨される使用方法も示されています。ここで示されているのはデフォルトのログ・ディレクトリです。選択したディレクトリにログ・ファイルをリダイレクトするオプションがあります。ログ・ファイルの格納先をリダイレクトしている場合は、適切な場所を確認してください。

表 4. クラスター・メッセージ・ログ・ファイル

ログ・ファイル名	説明
システム・エラー・ログ	<p>スクリプトおよびデーモンを含むすべての AIX サブシステムにより生成された、タイム・スタンプが付いた定様式のメッセージが格納されます。このログ・ファイルの表示とメッセージの解釈については、セクション『システム・エラー・ログの理解』を参照してください。</p> <p>推奨: システム・エラー・ログには、他の多くのシステム・コンポーネントからのタイム・スタンプ付きメッセージが記録されているため、システム・エラー・ログは、クラスター・イベントとシステム・イベントを相互に関連付ける場所として適切です。</p>
/tmp/clconvert.log	<p>最新の PowerHA SystemMirror リリースにアップグレードする際の変換処理の記録が格納されます。インストール・プロセスで <b>cl_convert</b> ユーティリティーが実行され、<b>/tmp/clconvert.log</b> ファイルが作成されます。</p> <p>推奨: コマンド行から <b>cl_convert</b> を実行するときは、<b>clconvert.log</b> を表示して、変換が正常に行われたかどうかを判断してください。</p>
/var/ha/log/grpglsm	<p>ASCII フォーマットのタイム・スタンプ付きメッセージが格納されます。これらは RSCD グループ・サービス・グローバルライズド・スイッチ・メンバーシップ・デーモンの内部アクティビティーの実行を追跡します。この情報は、IBM サポート担当員がトラブルシューティングのために使用します。このファイルは定期的に切り詰められます。そのため、このファイルが必要になった場合は、すぐに別ファイルとして保存するようにしてください。</p>
/var/hacmp/adm/cluster.log	<p>PowerHA SystemMirror のスクリプトおよびデーモンにより生成される、タイム・スタンプの付いた定様式のメッセージが格納されます。</p> <p>推奨: このログ・ファイルには、現在のクラスター状況の概要が表示されるため、クラスターの問題を診断するときは最初にこのファイルを確認してください。</p>
/var/hacmp/adm/history/cluster.mmddyyyy	<p>PowerHA SystemMirror のスクリプトにより生成される、タイム・スタンプの付いた定様式のメッセージが格納されます。クラスター・ヒストリー・ファイルは毎日作成され、ファイル名の拡張子により各ファイルが識別されます。<i>mm</i> は月、<i>dd</i> は日、<i>yyyy</i> は年をそれぞれ示します。このログ・ファイルの表示と、そのメッセージの解釈については、セクション『クラスター・ヒストリー・ログ・ファイルの理解』を参照してください。</p> <p>推奨: 時間経過によるクラスターの動作を拡張表示するには、クラスター・ヒストリー・ログ・ファイルを使用してください。</p> <p>このログは、並列処理されるリソース・グループの追跡ツールとしては適しません。並列処理では、以前には個別のイベントとして実行された特定のステップが別の方法で処理されるため、それらのステップがクラスター・ヒストリー・ログでは明確になりません。並列処理アクティビティーを追跡するには、<b>hacmp.out</b> ファイルを使用してください。</p>
/var/log/clcomd/clcomddiag.log	<p><b>clcomd</b> により生成される、タイム・スタンプ付きの定様式診断メッセージが格納されます。</p> <p>推奨: このファイルに含まれる情報は、IBM サポート担当者用の情報です。</p>
/var/hacmp/log/autoverify.log	<p>自動クラスター検証実行時に発生した警告またはエラーが含まれます。</p>
/var/hacmp/log/clavan.log	<p>PowerHA SystemMirror が管理するアプリケーションの状態遷移が含まれます。例えば、PowerHA SystemMirror が管理する各アプリケーションが起動または停止した場合や、アプリケーションを実行しているノードが停止した場合です。</p> <p>それぞれのノードには、固有のファイルのインスタンスがあります。<b>clavan.log</b> ファイル内の各レコードは、単一の行で構成されます。それぞれの行には固定部分と変数部分があります。</p> <p>推奨: ユーティリティー・プログラムは、<b>clavan.log</b> ファイルのレコードをクラスター内の各ノードから収集することで、アプリケーションの可用性時間を示す他の統計を計算するだけでなく、各アプリケーションが実行されていた期間を判別できます。</p>

表 4. クラスター・メッセージ・ログ・ファイル (続き)

ログ・ファイル名	説明
/var/hacmp/log/ clinfo.log	<b>clinfo.log</b> ファイルには、実行時のイベント・スクリプトにより生成された出力が記録されます。この情報は、 <b>/var/hacmp/log/hacmp.out</b> ファイルにある情報を補足および拡張します。
/var/hacmp/log/ clinfo.log.n, n=1,...,7	クライアント・システムとサーバー・システムの両方にクラスター情報 (Clinfo) サービスをインストールできます。クライアント・システム (cluster.es.client) には HACMP ODM (HACMPlogs など) やユーティリティー (clcycle など) はありません。したがって、Clinfo ロギングではサイクル形式やリダイレクトの利点を活用できません。  デフォルトのデバッグ・レベルは 0 または「オフ」です。コマンド行フラグを使用してロギングを使用可能にできます。 <b>clinfo -l</b> フラグを使用してログ・ファイル名を変更します。
/var/hacmp/log/ clstrmgr.debug	<b>clstrmgrES</b> サブシステムによって生成される、タイム・スタンプ付きの定様式メッセージが格納されます。デフォルトの詳細なメッセージは、通常ほとんどの問題のトラブルシューティングに対応しますが、IBM のサポートによってさらに他のデバッグ方法が提示される場合もあります。
/var/hacmp/log/ clstrmgr.debug.n, n=1,...,7	推奨: このファイルに含まれる情報は、IBM サポート担当者用の情報です。
/var/hacmp/log/ clstrmgr.debug.long	クラスター・マネージャーのアクティビティ、特に、PowerHA SystemMirror のその他のコンポーネントや RSCT との対話の概要レベルのロギング、現在実行されているイベント、およびリソース・グループに関する情報 (例えば、イベント中の獲得や解放など、その状態と実行されるアクション) が格納されます。
/var/hacmp/log/ clstrmgr.debug.long.n, n=1,...,7	推奨: このファイルに含まれる情報は、IBM サポート担当者用の情報です。
/var/hacmp/log/ cspoc.log	PowerHA SystemMirror の C-SPOC コマンドにより生成される、タイム・スタンプの付いた定様式のメッセージが格納されます。 <b>cspoc.log</b> ファイルは、C-SPOC コマンドを呼び出すノードにあります。
	推奨: クラスター・ノードで C-SPOC コマンドが実行されたときの状況を追跡する場合は、C-SPOC ログ・ファイルを使用してください。
/var/hacmp/log/ cspoc.log.long	C-SPOC ユーティリティーの概要レベルのロギング、つまり、指定されたノードで C-SPOC によって呼び出されたコマンドおよびユーティリティーと、それらの戻り状況が格納されます。
/var/hacmp/log/ cspoc.log.remote	ksh オプション <b>xtrace</b> が有効な ( <b>set -x</b> ) リモート・ノードでの C-SPOC コマンド実行のロギングが格納されます。
/var/hacmp/log/ hacmp.out	現在日に PowerHA SystemMirror スクリプトにより生成され、タイム・スタンプの付いた定様式のメッセージが格納されます。
/var/hacmp/log/ hacmp.out.n n=1,...,7	冗長モード (推奨モード) の場合このログ・ファイルには、スクリプトによって実行されたすべてのコマンドのレコードが行ごとに、すべての引数の値と共に詳細に記録されます。各上位イベントのイベント要約は、イベント詳細の最後にそれぞれ含まれます。このログ・ファイルの表示とメッセージの解釈については、セクション『hacmp.out ログ・ファイルの理解』を参照してください。  推奨: このログ・ファイル内の情報は、 <b>/var/hacmp/adm/cluster.log</b> ファイル内の情報に基づいて補足されたり拡張されたりするため、問題を調査するときの主要な情報源になります。
/var/hacmp/log/	Smart Assist を使用しているときに発生する Oracle 固有のエラーに関する情報が含まれ、Oracle Smart Assist によって使用されます。
/var/hacmp/log/ sa.log	Smart Assist を使用しているときに発生する一般的なエラーに関する情報が含まれ、Smart Assist インフラストラクチャーによって使用されます。
/var/log/clcomd/ clcomddiag.log	クラスター通信デーモン ( <b>clcomd</b> アクティビティによって生成されるタイム・スタンプ付きの定様式メッセージ) が格納されます。成功したのもも失敗したのもも含め、着信接続と発信接続に関する情報がログに示されます。また、 <b>/usr/es/sbin/cluster/etc/rhosts</b> に対するファイル権限が正しく設定されていない場合、警告が表示され、このシステムのユーザーはファイルに書き込むことができません。  推奨: PowerHA SystemMirror ユーティリティーの通信問題をトラブルシューティングするには、このファイルを使用してください。
/var/hacmp/log/ clconfigassist.log	2 ノード・クラスター構成アシストのデバッグ情報が格納されます。このアシストには、トラブルシューティング・アクティビティを支援する番号付きのログ・ファイルが最大 10 個格納されます。

表 4. クラスター・メッセージ・ログ・ファイル (続き)

ログ・ファイル名	説明
/var/hacmp/ clverify/clverify.log	<b>clverify.log</b> ファイルには、クラスターの検証ユーティリティによって出力される詳細メッセージが格納されます。このメッセージには、検証エラーが発生したノード、デバイス、コマンドなどが示されます。
/var/hacmp/log/ clutils.log	クラスターの自動構成検証を実行した日付、時刻、結果、およびノードの情報が格納されます。  また、ファイル・コレクション・ユーティリティ、2 ノード・クラスターの構成アシスト、およびクラスター・テスト・ツールに関する情報も格納されます。
/var/hacmp/log/ cl_testtool.log	<b>hacmp.out</b> ファイルからの抜粋が含まれます。クラスター・テスト・ツールは、最大 3 個のログ・ファイルを保存し、別のクラスター・テストの結果と比較できるようにそれぞれに番号を付けます。また、ツールが一番古いファイルを上書きしてファイルを置き換えます。

関連資料:

『クラスター・ログ・ファイルの理解』

**/var/hacmp/adm/cluster.log** ファイルは標準のテキスト・ファイルです。このファイルを調べるときは、初めに、問題に関連する最新のエラー・メッセージを見つけてください。そこから、ログ・ファイルをさかのぼって読み、その問題に関連する最初のメッセージを見つけます。通常、問題のソースを示す最初のエラーから、多くのメッセージが次々と生成されます。

25 ページの『クラスター・ヒストリー・ログ・ファイルの理解』

クラスター・ヒストリー・ログ・ファイルは、標準のテキスト・ファイルです。システムで割り当てられる名前は **/usr/es/sbin/cluster/history/cluster.mmddyyyy** です (*mm* は月、*dd* は日、*yyyy* は年を示します)。

17 ページの『hacmp.out ログ・ファイルの理解』

**/var/hacmp/log/hacmp.out** ファイルは標準のテキスト・ファイルです。システムの **hacmp.out** ログ・ファイルのサイクルは 7 回です。各コピーは、ファイル名に付加された数字で識別されます。最新のログ・ファイルは **/var/hacmp/log/hacmp.out**、最も古いファイル・バージョンは **/var/hacmp/log/hacmp.out.7** になります。

関連情報:

PowerHA SystemMirror クラスターのアップグレード

PowerHA SystemMirror クラスターの検査および同期化

クラスター・ログ・ファイルの理解:

**/var/hacmp/adm/cluster.log** ファイルは標準のテキスト・ファイルです。このファイルを調べるときは、初めに、問題に関連する最新のエラー・メッセージを見つけてください。そこから、ログ・ファイルをさかのぼって読み、その問題に関連する最初のメッセージを見つけます。通常、問題のソースを示す最初のエラーから、多くのメッセージが次々と生成されます。

**cluster.log** ファイルのメッセージ・フォーマット

**/var/hacmp/adm/cluster.log** ファイル内のエントリーには、次のフォーマットが使用されています。

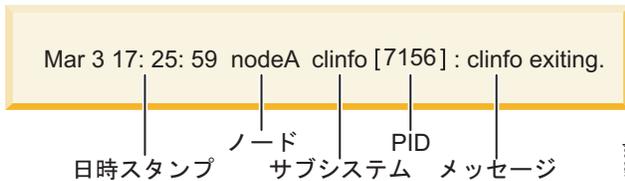


図 1. エントリーのフォーマット

各エントリーには次の情報が含まれます。

表 5. cluster.log ファイル

エントリー	説明
日時スタンプ	イベントが発生した日付と時刻。
ノード	イベントが発生したノード
サブシステム	イベントを生成した PowerHA SystemMirror サブシステム。サブシステムは、次の省略形によって識別されます。 <ul style="list-style-type: none"> <li>• clstrmgrES - クラスター・マネージャー・デーモン</li> <li>• clinfoES - クラスター情報プログラム・デーモン</li> </ul>
PID	メッセージを生成するデーモンのプロセス ID (スクリプトが出力するメッセージ用のは含まない)
メッセージ	メッセージ・テキスト

前例のエントリーは、クラスター情報プログラム (clinfoES) の実行が、3 月 3 日午後 5 時 25 分に nodeA というノード上で停止されたことを示します。

/var/hacmp/adm/cluster.log ファイルは標準の ASCII テキスト・ファイルなので、more または tail コマンドなどの標準の AIX ファイル・コマンドを使用して表示できます。ただし、SMIT インターフェースを使用することもできます。次のセクションで、これらのオプションについて説明します。

### SMIT による cluster.log ファイルの表示

SMIT を使用して /var/hacmp/adm/cluster.log ファイルを表示するには、以下の手順を実行します。

1. smit hacmp と入力します。
2. SMIT で、「**Problem Determination Tools (問題判別ツール)**」 > 「**PowerHA SystemMirror Log Viewing and Management (ログの表示および管理)**」 > 「**View Detailed PowerHA SystemMirror Log Files (詳細ログ・ファイルの表示)**」を選択して、Enter を押します。
3. 「**Scan the PowerHA SystemMirror for AIX System Log (PowerHA SystemMirror for AIX システム・ログのスキャン)**」を選択し、Enter を押します。このオプションでは、/var/hacmp/adm/cluster.log ファイルが参照されます。

注: cluster.log ファイルの内容をスキャンするか、新規イベントが追加される際に、アクティブなログ・ファイルをリアルタイムでモニターするか、いずれかを選択できます。一般に、すでに発生した問題を見つける場合にファイルをスキャンし、次に、結果を判別するために問題に対する解決策をテストする場合にファイルをモニターします。

**hacmp.out** ログ・ファイルの理解:

**/var/hacmp/log/hacmp.out** ファイルは標準のテキスト・ファイルです。システムの **hacmp.out** ログ・ファイルのサイクルは 7 回です。各コピーは、ファイル名に付加された数字で識別されます。最新のログ・ファイルは **/var/hacmp/log/hacmp.out**、最も古いファイル・バージョンは **/var/hacmp/log/hacmp.out.7** になります。

フォールオーバー環境でのリソース・グループの処理方法および優先順位付けの方法が最近変更され、**hacmp.out** ファイルにイベント要約が含まれることになりました。イベント要約はリソース・グループのアクティビティおよび場所の追跡に役立ちます。

警告メッセージが表示されるまでの待機期間をカスタマイズできます。この待機期間は **config\_too\_long** メッセージがログに書き込まれる回数に反映されるため、**config\_too\_long** コンソール・メッセージは、問題が発生しているすべての場合に表示されるわけではありません。クラスター・イベントの実行に予想した以上の時間がかかると、**hacmp.out** に警告メッセージが追加されます。これは、イベント・スクリプトに障害が発生した場合、システム・コマンドがハングした場合、あるいは単にコマンドの実行が遅い場合に起こります。

**hacmp.out** ファイルを調べるときは、EVENT FAILED メッセージを探します。このメッセージは、障害が発生したことを示します。次に、その障害のメッセージから、ログ・ファイルを逆方向に読み、実際に何が問題だったのかを正確に判別します。**hacmp.out** ログ・ファイルには、問題を調査する際に、最も重要な情報源が表示されます。

イベント・プリアンブル:

依存関係または複製リソースのあるリソース・グループがクラスター・イベントで処理されると、**hacmp.out** ファイルにイベント・プリアンブルが入れられます。

このプリアンブルは、クラスター・マネージャーが正しいノードおよびサイト上のリソース・グループのオンライン化の試行を計画するイベントのシーケンスを示します。さらに、個々のグループの依存関係およびサイト構成も考慮されます。

注: このプリアンブルは、イベントの計画段階においてクラスター・マネージャーがエンキューするイベントのシーケンスを表します。個々のイベントに障害が起こった場合、または何らかの理由でクラスター・マネージャーが計画の再見積りを行った場合は、新しいプリアンブルが生成されます。元のプリアンブルのすべてのイベントがすべて実行されるとは限りません。

例

PowerHA SystemMirror Event Preamble

```
-----  
Node Down Completion Event has been enqueued.  
-----  
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
PowerHA SystemMirror Event Preamble  
Action: Resource:  
-----  
Enqueued rg_move acquire event for resource group rg3.  
  
Enqueued rg_move release event for resource group rg3.  
  
Enqueued rg_move secondary acquire event for resource group 'rg1'.  
Node Up Completion Event has been enqueued.  
-----
```

イベント要約:

各イベントの詳細の最後に表示されるイベント要約により、**hacmp.out** ファイルのエラーの有無をより簡単にチェックできます。イベント要約には対応するイベントを指すポインターが含まれています。これによりイベントに関する出力を簡単に見つけることができます。

出力例については、セクション『**hacmp.out** ログ・ファイルの非詳細出力と詳細出力』を参照してください。

また、現在および過去の **hacmp.out** ファイルから取得したイベント要約が編集されたセクションのみも表示できます。この表示用のオプションは、「**Problem Determination Tools (問題判別ツール)**」 > **PowerHA SystemMirror 「Log Viewing and Management (ログ表示と管理)」** > **「View/Save/Remove Event Summaries (イベント要約の表示/保存/除去)」** > **「View Event Summaries (イベント要約の表示)」** SMIT パネルにあります。詳しくは、セクション『**収集された hacmp.out イベント要約の表示**』を参照してください。

関連資料:

23 ページの『**収集された hacmp.out イベント要約の表示**』  
クラスター・マネージャーがこれらのイベントを起動した後、**hacmp.out** ファイルにイベント要約が表示されます。例えば、**node\_up** と **node\_up\_complete**、および関連サブイベント (**node\_up\_local** や **node\_up\_remote\_complete** など) が表示されます。

20 ページの『**hacmp.out ログ・ファイルの非詳細出力と詳細出力**』  
詳細出力または非詳細出力のいずれかを選択できます。

**HTML** フォーマットの **hacmp.out**:

形式オプションを「**Problem Determination Tools (問題判別ツール)**」 > **「PowerHA SystemMirror Log Viewing and Management (ログ表示と管理)」** > **「Change/Show PowerHA SystemMirror Log File Parameters (PowerHA SystemMirror ログ・ファイル・パラメーターの変更/表示)」** の「SMIT 画面」に設定すると、**hacmp.out** ログ・ファイルを **HTML** フォーマットで表示できます。

詳しくは、セクション『**hacmp.out** ファイルに記録される情報のレベルとフォーマットの設定』を参照してください。

関連タスク:

22 ページの『**hacmp.out** ファイルに記録される情報のレベルとフォーマットの設定』  
**/var/hacmp/log/hacmp.out** ファイルに記録される情報のレベルを設定できます。

**hacmp.out** でのリソース・グループ獲得失敗とボリューム・グループ障害:

報告されたリソース・グループ獲得失敗 (コマンドが返した非ゼロの終了コードにより示される失敗) は、**hacmp.out** で追跡されます。

この情報には、以下が含まれます。

- イベントの起動および終了回数
- イベントの結果、影響を受けた (獲得または解放された) リソース・グループ
- イベントが失敗した場合、失敗したリソース・アクション

クラスター・マネージャーがリソースを使用可能な状態に保とうとする経過を追跡できます。

さらに、ボリューム・グループ障害の場合に実行される自動構成の AIX エラー通知メソッドによって、**hacmp.out** ログ・ファイルに以下の情報が書き込まれます。

- メソッドを起動した AIX エラー・ラベルと ID
- 影響を受けたリソース・グループ名
- エラーが発生したノード名

**node\_up** に基づくリソース・グループ・リカバリーのメッセージ:

**hacmp.out** ファイル、イベント要約、および **clstat** には、結合ノードまたは起動しているノードにオンライン接続しようとしたエラー状態のリソース・グループの情報やメッセージが含まれます。

同様に、そのようなリソース・グループの獲得失敗のケースを追跡できます。PowerHA SystemMirror は、**rg\_move** イベントを起動して、リソース・グループをノード・リストの他のノードに移動します。ノード間の連続した **rg\_move** イベントの結果として、非コンカレントのリソース・グループを取得できない場合、PowerHA SystemMirror は **hacmp.out** ファイルにメッセージを追加します。

ネットワークについて報告されるインターフェース・イベント:

ネットワークにネットワーク・インターフェースを追加する場合、この場合に実行される実際のイベントを **join\_interface** といいます。これは **hacmp.out** ファイルに反映されます。

同様に、ネットワーク・インターフェース障害が発生した場合、実行される実際のイベントは **fail\_interface** と呼ばれます。これも **hacmp.out** ファイルに反映されます。この場合に実行されるイベントは、指定したネットワークのインターフェースに障害が発生したことのみを示します。

**hacmp.out** ファイルのリソース・グループ処理メッセージ:

**hacmp.out** ファイルを使用すると、PowerHA SystemMirror のリソース・グループの処理方法を完全に追跡できます。

このトピックでは、簡単な説明にとどめます。詳細、およびジョブ・タイプ のイベント要約の例については、セクション『**hacmp.out** ファイルでのリソース・グループ並列処理/シリアル処理の追跡』を参照してください。

PowerHA SystemMirror により処理された各リソース・グループの場合は、以下の情報が **hacmp.out** ファイルに送信されます。

- リソース・グループ名
- スクリプト名
- 実行されているコマンド名

以下に一般的な出力形式を示します。

```
resource_group_name:script_name [line number] command line
```

イベント・スクリプトが特定のリソース・グループを処理しない場合、例えば、**node\_up** イベントの最初では、リソース・グループ名は取得できません。このような場合は、タグのリソース・グループ名の部分がブランクになります。

例えば、**hacmp.out** ファイルには以下のいずれかの行が含まれます。

```
cas2:node_up_local[199] set_resource_status ACQUIRING
:node_up[233] cl_ssa_fence up stan
```

さらに、**hacmp.out** ファイル内のイベント要約の個々のリソースの参照に、関連するリソース・グループに対する参照タグが含まれます。例:

関連資料:

27 ページの『**hacmp.out** ファイルでのリソース・グループ処理の追跡』

**hacmp.out** ファイルへの出力により、特定のリソース・グループとそのリソースに関連する詳細を切り分けることができます。 **hacmp.out** イベント要約の内容に基づいて、リソース・グループが期待どおりの順序で処理されているかどうかを判断できます。

**hacmp.out** ファイルでの **Config\_too\_long** メッセージ:

指定したイベント期間内に完了しないクラスター・イベントごとに、**config\_too\_long** メッセージが **hacmp.out** ファイルにログ記録されます。

次に、メッセージが以下のパターンに従ってコンソールに送信されます。

- 最初の 5 つの **config\_too\_long** メッセージは 30 秒間隔で **hacmp.out** ファイルに表示されます。
- 次の 5 つのメッセージは、間隔が 1 時間経過するまでそれまでの 2 倍の間隔で表示されます。
- これらのメッセージは、イベントが完了するまで、あるいはイベントがそのノードで終了するまで毎時間ログに記録されます。

**config\_too\_long** メッセージを送信する前に待機期間をカスタマイズできます。

関連情報:

クラスター・イベントの計画

**hacmp.out** ログ・ファイルの非詳細出力と詳細出力:

詳細出力または非詳細出力のいずれかを選択できます。

非詳細出力

非詳細出力モードでは、**hacmp.out** ログには、すべての PowerHA SystemMirror スクリプトが出力する起動、完了、およびエラー通知メッセージが含まれます。 各エントリーには次の情報が含まれます。

表 6. **hacmp.out** ログ・ファイル

エントリー	説明
日時スタンプ	イベントが発生した日付と時刻。
メッセージ	クラスター・アクティビティを記述するテキスト
戻り状況	障害を報告するメッセージには、スクリプトから戻された状況も含まれます。 正常に完了したスクリプトの場合、この情報は含まれません。
イベントの説明	ノード、ファイル・システム、またはボリューム・グループで試行または完了された特定のアクション

詳細出力

詳細モードでは、**hacmp.out** ファイルにはスクリプトやコマンドに渡される引数値とフラグ設定も含まれます。

イベント要約を含む詳細出力の例

一部のイベント (クラスター・マネージャーによって開始されたイベント) の後にはイベント要約が表示されます。その抜粋を以下に示します。

....  
Mar 25 15:20:30 EVENT COMPLETED: network\_up alcuin tmssanet\_alcuin\_bede

PowerHA SystemMirror Event Summary  
Event: network\_up alcuin tmssanet\_alcuin\_bede  
Start time: Tue Mar 25 15:20:30 2003

End time: Tue Mar 25 15:20:30 2003

Action: Resource:Script Name:

-----  
No resources changed as a result of this event  
-----

#### 整定時間のイベント要約

**CustomRG** では整定時間が構成されます。優先順位の低いノードはクラスターに結合します。

Mar 25 15:20:30 EVENT COMPLETED: node\_up alcuin

PowerHA SystemMirror Event Summary  
Event: node\_up alcuin  
Start time: Tue Mar 25 15:20:30 2003

End time: Tue Mar 25 15:20:30 2003

Action: Resource: Script Name:

-----  
No action taken on resource group 'CustomRG'.  
The Resource Group 'CustomRG' has been configured  
to use 20 Seconds Settling Time. This group will be  
processed when the timer expires.  
-----

#### フォールバック・タイマーのイベント要約

**CustomRG** には毎日フォールバック・タイマーが設定され、22 時間 10 分フォールバックします。リソース・グループは、優先順位の低いノード (bede) 上にあります。したがって、タイマーが作動していません。優先順位の高いノード (alcuin) はクラスターに結合します。

#### The message on bede

...

Mar 25 15:20:30 EVENT COMPLETED: node\_up alcuin

PowerHA SystemMirror Event Summary  
Event: node\_up alcuin  
Start time: Tue Mar 25 15:20:30 2003

End time: Tue Mar 25 15:20:30 2003

Action: Resource: Script Name:

-----  
No action taken on resource group 'CustomRG'.  
The Resource Group 'CustomRG' has been configured  
to fallback on Mon Mar 25 22:10:00 2003  
-----

#### The message on alcuin ...

Mar 25 15:20:30 EVENT COMPLETED: node\_up alcuin

PowerHA SystemMirror Event Summary  
Event: node\_up alcuin  
Start time: Tue Mar 25 15:20:30 2003

End time: Tue Mar 25 15:20:30 2003

Action: Resource: Script Name:

-----  
The Resource Group 'CustomRG' has been configured  
to fallback using daily1 Timer Policy  
-----

**SMIT** による *hacmp.out* ファイルの表示:

SMIT を使用すれば、*/var/hacmp/log/hacmp.out* ファイルを表示できます。

SMIT を使用して */var/hacmp/log/hacmp.out* ファイルを表示するには、以下の手順を実行します。

1. `smit hacmp` と入力します。
2. SMIT で、「**Problem Determination Tools (問題判別ツール)**」>「**PowerHA SystemMirror Log Viewing and Management (ログの表示および管理)**」>「**View Detailed PowerHA SystemMirror Log Files (詳細ログ・ファイルの表示)**」を選択して、Enter を押します。
3. 「**View Detailed PowerHA SystemMirror Log Files (詳細ログ・ファイルの表示)**」メニューで、*/var/hacmp/log/hacmp.out* ファイルのコンテンツをスキャンするか、新規イベントのログ・ファイルへの追加をモニターするかのいずれかを選択できます。一般に、すでに発生した問題を見つけるためにはファイルをスキャンし、次に、問題に対する解決策をテストするためにファイルをモニターします。このメニューでは、*/var/hacmp/log/hacmp.out* ファイルは PowerHA SystemMirror スクリプト・ログ・ファイルとして参照されます。
4. 「**Scan the PowerHA SystemMirror Script Log File (PowerHA SystemMirror スクリプト・ログ・ファイルのスキャン)**」を選択し、Enter を押します。
5. スクリプト・ログ・ファイルを選択し、Enter を押します。

*hacmp.out* ファイルに記録される情報のレベルとフォーマットの設定:

*/var/hacmp/log/hacmp.out* ファイルに記録される情報のレベルを設定できます。

注: これらの設定値は、設定するとただちに有効になります。

*/var/hacmp/log/hacmp.out* ファイルに記録された情報レベルを設定するには、次のようにします。

1. `smit hacmp` と入力します。
2. SMIT で、「**問題判別ツール**」>「**PowerHA SystemMirror ログの表示および管理**」>「**PowerHA SystemMirror ログ・ファイル・パラメーターの変更/表示**」を選択します。

変更するクラスター・ノード名を指定するプロンプトが表示されます。ランタイム (実行時) パラメーターは、ノードごとに構成されます。

3. ノード名を入力し、Enter を押します。

SMIT が「**PowerHA SystemMirror Log File Parameters (PowerHA SystemMirror ログ・ファイル・パラメーター)**」パネルを表示します。

4. 詳細出力を取得するには、「**Debug Level (デバッグ・レベル)**」フィールドの値を「**high (高)**」に設定します。
5. *hacmp.out* の表示フォーマットを変更するには、「**Formatting options for hacmp.out (hacmp.out の形式オプション)**」を選択します。ノードを選択して、フォーマットを「**HTML (Low) (HTML (低))**」、「**HTML (High) (HTML (高))**」、「**Default (None) (デフォルト (なし))**」、または「**Standard (標準)**」に設定します。

注: **hacmp.out** の形式オプションを「**Default (None)** (デフォルト (なし))」に設定すると、イベント要約は生成されません。 イベント要約について詳しくは、セクション『収集された **hacmp.out** イベント要約の表示』を参照してください。

6. デバッグ情報のレベルを変更するには、「**Cluster Manager debug level** (クラスター・マネージャー・デバッグ・レベル)」フィールドの値を「**standard** (標準)」または「**high** (高)」のいずれかに設定します。

関連資料:

『収集された **hacmp.out** イベント要約の表示』

クラスター・マネージャーがこれらのイベントを起動した後、**hacmp.out** ファイルにイベント要約が表示されます。 例えば、**node\_up** と **node\_up\_complete**、および関連サブイベント (**node\_up\_local** や **node\_up\_remote\_complete** など) が表示されます。

収集された **hacmp.out** イベント要約の表示:

クラスター・マネージャーがこれらのイベントを起動した後、**hacmp.out** ファイルにイベント要約が表示されます。 例えば、**node\_up** と **node\_up\_complete**、および関連サブイベント (**node\_up\_local** や **node\_up\_remote\_complete** など) が表示されます。

イベント要約が表示されないイベントもあります。 例えば、SMIT を使用してリソース・グループを移動する場合などは表示されません。

「**View Event Summaries** (イベント要約の表示)」オプションには、ノードの **hacmp.out** ファイルに書き込まれるすべてのイベント要約がまとめられて表示されます。 **hacmp.out** ファイルを新規の場所にリダイレクトしても、このユーティリティーはこうした情報を収集して表示します。 また、イベント要約を SMIT で表示する代わりに、選択したファイルに保管することもできます。

注: **hacmp.out** ファイルから取得されたイベント要約は、**/usr/es/sbin/cluster/cl\_event\_summary.txt** ファイルに保管されます。 このファイルは **hacmp.out** サイクルに従って累積され続け、自動的に切り捨てられたり置換されたりすることはありません。 その結果、このファイルが大きくなりすぎて **/usr** ディレクトリーがいっぱいになります。 SMIT の「**Remove Event Summary History** (イベント要約ヒストリーの除去)」オプションを使用して定期的にイベント要約をクリアする必要があります。

この機能は、ノード固有のもので、したがって、クラスター内の 1 つのノードのイベント要約情報に他のノードからはアクセスできません。 イベント要約を収集し表示する各ノード上で、「**View Event Summaries** (イベント要約の表示)」オプションを実行します。

イベント要約を表示すると、クラスターで最近発生したイベントの概要を素早く確認できます。 イベント要約により問題のイベントが示されたら、ソースの **hacmp.out** ファイルを調べると発生した問題の詳細を完全に確認できます。

注: **hacmp.out** の形式オプションを「**Default (None)** (デフォルト (なし))」に設定すると、イベント要約は生成されません。 「**View Event Summaries** (イベント要約の表示)」コマンドからは、何の結果も得られません。

イベント要約表示情報の収集方法:

「**Problem Determination Tools** (問題判別ツール)」 > 「**PowerHA SystemMirror Log Viewing and Management** (ログの表示と管理)」 -> 「**View/Save/Remove PowerHA SystemMirror Event Summaries** (イベント要約の表示/保存/除去)」 -> 「**View Event Summaries** (イベント要約の表示)」 というオプションでは、**hacmp.out** ログ・ファイルから情報が収集されます。 実行中の PowerHA SystemMirror から直接

収集されるものではありません。その結果、PowerHA SystemMirror が実行されていないときでもイベント要約情報にアクセスできます。要約表示は、現在の日付のイベント要約によって 1 日 1 回更新されます。

また、表示情報の下部にはリソース・グループのロケーションと状態情報が示されます。この情報には、**clRGinfo** コマンドの出力が反映されます。

**clRGinfo** では、クラスター実行時にさらに素早くリソース・グループ情報が表示されます。クラスターが実行中でない場合は、リソース・グループ情報が表示されるまでに数分かかります。

イベント要約の表示:

SMIT を使用すれば、ノードに関するイベント要約を収集リストで表示できます。

ノードに関するイベント要約を収集リストで表示するには、以下の手順を実行します。

1. `smit hacmp` と入力します。
2. SMIT で「**View Event Summaries (イベント要約の表示)**」を選択し、Enter を押します。SMIT がノードで生成されたイベント要約のリストを表示します。イベント要約が検出されなかった場合は、SMIT によって通知されます。

指定したファイルへのイベント要約の保存:

SMIT を使用すれば、ノードのイベント要約の収集リストをファイルに保管できます。

ノードのイベント要約の収集リストをファイルに保管するには、以下の手順を実行します。

1. `smit hacmp` と入力します。
2. SMIT で「**View/Save/Remove PowerHA SystemMirror Event Summaries (PowerHA SystemMirror イベント要約の表示/保存/除去)**」を選択します。
3. 「**Save Event Summaries to a file (イベント要約をファイルに保存)**」を選択します。
4. イベント要約を保管するパスやファイル名を入力します。

選択したフォーマット (.txt や .html など) に応じて、ファイルを移動してテキスト・エディターやブラウザで表示できます。

システム・エラー・ログの理解:

PowerHA SystemMirror ソフトウェアは、デーモンにより状況メッセージが生成されるたびに、メッセージをシステム・エラー・ログに記録します。

システム・エラー・ログの PowerHA SystemMirror メッセージのフォーマットは、他の AIX サブシステムにより使用されるフォーマットと同じです。システム・エラー・ログ内のメッセージは、長形式または短形式で表示できます。

短形式 (要約形式ともいう) では、システム・エラー・ログ内の各メッセージが 1 行に表示されます。以下に短形式のシステム・エラー・ログのフィールドについて説明します。

表 7. システム・エラー・ログ

フィールド	説明
Error_ID	固有のエラー ID。
タイム・スタンプ	イベントが発生した日付と時刻。
T	エラー・タイプ: 永久 (P)、未解決 (U)、一時的 (T)。
CL	エラー・クラス: ハードウェア (H)、ソフトウェア (S)、通知 (O)。
Resource_name	メッセージを生成した AIX リソースまたはサブシステムを識別するテキスト・ストリング。 PowerHA SystemMirror メッセージは、そのデーモンの名前によって識別されます。
Error_description	エラーを簡単に記述したテキスト・ストリング。

長形式では、定様式情報が各エラーについて 1 ページずつ表示されます。

PowerHA SystemMirror ログ・ファイルとは異なり、システム・エラー・ログはテキスト・ファイルではありません。

AIX **errpt** コマンドは、システム・エラー・ログのエントリーからエラー・レポートを生成します。このコマンドの使用方法に関する情報については、**errpt** マニュアル・ページを参照してください。

AIX システム・エラー・ログを表示するためには、AIX SMIT を使用する必要があります。

1. **smit** と入力します。
2. SMIT で、「**Problem Determination Tools (問題判別ツール)**」>「**PowerHA SystemMirror Log Viewing and Management (PowerHA SystemMirror ログの表示および管理)**」>「**View Detailed PowerHA SystemMirror Log Files (詳細 PowerHA SystemMirror ログ・ファイルの表示)**」>「**Scan the PowerHA SystemMirror for AIX System Log (PowerHA SystemMirror for AIX システム・ログのスキャン)**」を選択し、Enter を押します。

エラー・ログが表示されます。

クラスター・ヒストリー・ログ・ファイルの理解:

クラスター・ヒストリー・ログ・ファイルは、標準のテキスト・ファイルです。システムで割り当てられる名前は `/usr/es/sbin/cluster/history/cluster.mmddyyyy` です (*mm* は月、*dd* は日、*yyyy* は年を示します)。

これらのログ・ファイルのコピーをいくつまで保持するかを決定し、その数を超える余分なコピーは、ディスクのストレージ・スペースを確保するために定期的に除去するようにします。また、クラスター・ヒストリー・ログ・ファイルを定期的なシステム・バックアップ手順に含めることもできます。

クラスター・ヒストリー・ログ・ファイルのメッセージのフィールドについて説明します。

表 8. クラスター・ヒストリー・ログ・ファイル

フィールド	説明
日時スタンプ	イベントが発生した日時
メッセージ	メッセージのテキスト
説明	イベント・スクリプトの名前

注: このログは特定のイベントをレポートします。リソース・グループが並列処理されている場合は、以前には個別のイベントとして実行された特定のステップが別の方法で処理されるため、それらのステップはクラスター・ヒストリー・ログ・ファイルにイベントとして示されません。 **hacmp.out** ファイルを使用す

する必要があります。このファイルには、リソース・グループ・アクティビティーとロケーションが非常に詳細に記載されているので、並列処理アクティビティーを追跡できます。

クラスター・ヒストリー・ログ・ファイルは標準のテキスト・ファイルなので、**cat**、**more**、および **tail** などの標準の AIX ファイル・コマンドを使用してコンテンツを表示できます。このログ・ファイルは、SMIT では表示できません。

問題を報告するためにクラスター・ログ・ファイルを収集するには、以下の手順を実行します：

PowerHA SystemMirror に問題が発生して、IBM サポートに報告する場合は、その問題に関連するログ・ファイルを収集しておいてください。それには、PowerHA SystemMirror の「問題報告用に **PowerHA SystemMirror** ログ・ファイルを収集」SMIT パネルが役に立ちます。

#### 注意:

このパネルは、**IBM** サポート担当員からの依頼があった場合にのみ使用してください。**IBM** サポートからの指示なしでこのユーティリティーを使用する場合は、アクションおよび生じる結果を十分に理解してから行ってください。

問題報告用にクラスター・ログ・ファイルを収集するには、以下の手順を実行します。

1. `smit hacmp` と入力します。
2. SMIT で、「**Problem Determination Tools (問題判別ツール)**」> **PowerHA SystemMirror 「Log Viewing and Management (ログの表示および管理)**」> **Collect Log Files for Problem Reporting (問題報告のためのログ・ファイルの収集)**」を選択します。
3. 「**entry (エントリー)**」フィールドに以下の値を入力するか、選択します。

表 9. 問題報告フィールド用のログ・ファイルの収集

フィールド	値
Log Destination Directory (ログ宛先ディレクトリ)	クラスター・ログが収集されるディレクトリー名を入力します。デフォルトは <b>/tmp</b> です。
Collection Pass Number (収集パス番号)	このフィールドで値を選択します。デフォルトは <b>2</b> (収集する) です。必要なスペースを計算するには、 <b>1</b> を選択します。実際のデータを収集するには、 <b>2</b> を選択します。
Nodes to Collect Data from (データの収集元ノード)	データの収集元のノードを入力するか、選択します。ノード名はコンマで区切りません。デフォルトは <b>All</b> ノードです。
Debug (デバッグ)	デフォルトは「 <b>No (いいえ)</b> 」です。IBM サポートがデバッグをオンにするように要求した場合は、このオプションを使用してください。
Collect RSCT Log Files (RSCT ログ・ファイルの収集)	デフォルトは「 <b>Yes (はい)</b> 」です。RSCT データ収集をスキップします。

クラスター・ログ・ファイルの管理:

PowerHA SystemMirror は、クラスター・ログ・ファイルの管理を自動的に行います。個々のログは最大サイズに制限されており、一定期間が経過すると削除されるか、新しいバージョンで上書きされます。

一般的に、PowerHA SystemMirror ではすべてのログ・ファイルについて、デフォルトで以下の規則に従っています。

表 10. ログ・ファイルの一般規則

項目	規則
最大サイズ	サイズが 1 MB を超えるログ・ファイルは循環されます。
期限切れログの最大数	以前のバージョンのファイルが、最大で 7 個保存されます。
最大経過日数	経過時間が 1 日を超えるログ・ファイルは循環されます。

これらの一般規則で指定されている値をカスタマイズする場合は、各クラスター・ノードの `/etc/environment` ファイルで異なる値を指定してオーバーライドすることができます。

デフォルト値をオーバーライドするには、以下の項目を追加します。

表 11. オーバーライド値

項目	説明
<code>CLCYCLE_MAX_SIZE=&lt;サイズ (バイト単位)&gt;</code>	保存されるすべてのログ・ファイルの最大サイズを制限するには、この項目を追加します。
<code>CLCYCLE_MAX_LOGS=&lt;保存する古いファイルの数&gt;</code>	<code>clcycle</code> コマンドで保存する古いログ・ファイルの数を変更するには、この項目を追加します。
<code>CLCYCLE_MAX_DAYS=&lt;ログ・ファイルが循環される経過日数&gt;</code>	ログ・ファイルを循環させる経過日数を変更するには、この項目を追加します。
<code>CLCYCLE_CLUSTER_LOG= &lt;FALSE TRUE&gt;</code>	<code>cluster.log</code> ファイルが PowerHA SystemMirror によってデフォルトで管理されないようにするには、この項目を追加します。代わりに、PowerHA SystemMirror は <code>syslog.conf</code> ファイルに項目を追加します。この結果、 <code>syslog</code> サブシステム が <code>cluster.log</code> ファイルのサイズ、経過日数、およびバックアップ・コピーを管理します。 注: PowerHA SystemMirror で <code>cluster.log</code> ファイルを管理したい場合は、 <code>/etc/environment</code> ファイルで <code>CLCYCLE_CLUSTER_LOG=TRUE</code> を指定します。

## hacmp.out ファイルでのリソース・グループ処理の追跡

`hacmp.out` ファイルへの出力により、特定のリソース・グループとそのリソースに関連する詳細を切り分けることができます。 `hacmp.out` イベント要約の内容に基づいて、リソース・グループが期待どおりの順序で処理されているかどうかを判断できます。

イベント要約で反映される並列処理順序:

`hacmp.out` ファイルとイベント要約には、並列リソース・グループ処理の流れをたどるために役立ついくつかの機能がリストされています。

- `hacmp.out` ファイルのフローの各行には、適用されるリソース・グループの名前が入ります。
- イベント要約情報には、すべてのリソース・タイプの詳細が示されます。
- イベント要約内の各行には、関連するリソース・グループが示されます。

以下の例は、並列処理される `casrg1` および `casrg2` という名前のリソース・グループのイベント要約を示します。

PowerHA SystemMirror Event Summary

Event: node\_up electron

Start time: Wed May 8 11: 06: 30 2002

End time: Wed May 8 11: 07: 49 2002

Action: Resource: Script Name: -----

```
Acquiring resource group: cascrg1 process_resources
Search on: Wed. May. 8. 11: 06: 33. EDT. 2002. process_resources. cascrg1. ref
Acquiring resource group: cascrg2 process_resources
Search on: Wed. May. 8. 11: 06: 34. EDT. 2002. process_resources. cascrg2. ref
Acquiring resource: 192. 168. 41. 30 cl_swap_IP_address
Search on: Wed. May. 8. 11: 06: 36. EDT. 2002. cl_swap_IP_address. 192. 168. 41. 30
Acquiring resource: hdisk1 cl_disk_available
Search on: Wed. May. 8. 11: 06: 40. EDT. 2002. cl_disk_available.
hdisk1. cascrg1
Acquiring resource: hdisk2 cl_disk_available
Search on: Wed. May. 8. 11: 06: 40. EDT. 2002. cl_disk_available.
hdisk2. cascrg2 Resource online: hdisk1 cl_disk_available
Search on: Wed. May. 8. 11: 06: 42. EDT. 2002. cl_disk_available.
hdisk1. cascrg1 Resource online: hdisk2 cl_disk_available
Search on: Wed. May. 8. 11: 06: 43. EDT. 2002. cl_disk_available.
hdisk2. cascrg2
```

ここに示されているように、処理されるリソース・グループがすべて最初に示され、その次に処理中の個々のリソースが示されます。

ジョブ・タイプ: 並列リソース・グループ処理:

クラスターにリソース・グループ依存関係またはサイトを構成する場合、クラスター・マネージャーのアクション計画がリストされているイベント・プリアンブルを確認してください。この計画には、事前に規定されたイベントについてリソース・グループが行う処理が記載されています。

個別のイベントの実行は **hacmp.out** ファイルにトレースされます。あるイベントに問題がある場合、または予期された結果が戻されない場合、**hacmp.out** ファイルに一定のパターンとキーワードが表示されます。このファイルを使用して問題の原因の識別を試みてください。

以下の情報は、クラスター・イベント処理に関する詳細を必要とするユーザー用です。問題判別の最初の参照用ではありません。

PowerHA SystemMirror Enterprise Edition に問題がある場合、基本的な対応として、ソフトウェア障害報告手続きに従ってください。

クラスター・マネージャーは、クラスター・イベントを計画するために、「並列処理」と言われる手法を使用します。並列処理は、複数の異なるリカバリー・ステップを単一イベントに結合して、イベント処理の効率と速度を最大化します。並列処理では、リソース・タイプに基づいて各種のリソースを処理するために **process\_resources** イベント・スクリプトがメイン・イベントとして使用されます。**process\_resources** イベントは現在処理中のリソースを識別するためにキーワード「JOB\_TYPE」を使用します。

ジョブ・タイプは **hacmp.out** ログ・ファイルにリストされます。このリストは、各種のタイプのリソースの獲得または解放の際に発生するイベントの順序の識別に役立ちます。クラスター・リソース・グループの構成によっては、リソース・グループの並列処理の際に他の特定のジョブ・タイプが発生します。

- リソース・タイプごとに 1 つずつジョブ・タイプが存在します。ジョブ・タイプには以下のものがありますが、これに限られるわけではありません。DISKS、FILESYSTEMS、TAKEOVER\_LABELS、TAPE\_RESOURCES、AIX\_FAST\_CONNECTIONS、APPLICATIONS、COMMUNICATION\_LINKS、USERDEF\_RESOURCES、CONCURRENT\_VOLUME\_GROUPS、EXPORT\_FILESYSTEMS、MOUNT\_FILESYSTEMS および REMOUNT\_FILESYSTEMS。
- また、次のように、並列処理の利点を生かすために使用されるジョブ・タイプが多数あります。SETPRKEY、TELINIT、SYNC\_VGS、LOGREDO、NFS\_STOP、および UPDATESTATD。今後、関連

した操作は、リソース・グループごとにはなくイベントごとに一度のみ実行されるようになります。この変更は、特にクラスターが小さい場合、リソース・グループの並列処理の利点が活かされる主な領域の 1 つです。

#### **JOB\_TYPE=ONLINE:**

獲得イベントが完了する段階では、すべてのリソース・グループのリソースがすべて正常に獲得されたら、**ONLINE** ジョブ・タイプが実行されます。このジョブは、正常に獲得されたすべてのリソース・グループが確実にオンライン状態に設定されるようにします。**RESOURCE\_GROUPS** 変数には、獲得されたすべてのグループのリストが含まれます。

```
:process_resources[1476] clRGPA
:clRGPA[48] [[ high = high ]]
:clRGPA[48] version= 1. 16
:clRGPA[50] usingVer= clrgpa
:clRGPA[55] clrgpa
:clRGPA[56] exit 0
:process_resources[1476] eval JOB_TYPE= ONLINE RESOURCE_GROUPS="
casrg1 casrg2 conc_ rg1"

:process_resources[1476] JOB_TYPE= ONLINE RESOURCE_GROUPS=
casrg1 casrg2 conc_ rg1 :process_resources[1478] RC= 0
:process_resources[1479] set +a
:process_resources[1481] [ 0 -ne 0 ]
:process_resources[1700] set_resource_group_state UP
```

#### **JOB\_TYPE= OFFLINE:**

解放イベントが完了する段階では、すべてのリソース・グループのリソースがすべて正常に解放されたら、**OFFLINE** ジョブ・タイプが実行されます。このジョブは、正常に解放されたすべてのリソース・グループが確実にオフライン状態に設定されるようにします。**RESOURCE\_GROUPS** 変数には、解放されたすべてのグループのリストが含まれます。

```
conc_ rg1 :process_resources[1476] clRGPA
conc_ rg1 :clRGPA[48] [[ high = high ]]
conc_ rg1 :clRGPA[48] version= 1. 16
conc_ rg1 :clRGPA[50] usingVer= clrgpa
conc_ rg1 :clRGPA[55] clrgpa
conc_ rg1 :clRGPA[56] exit 0
conc_ rg1 :process_resources[1476] eval JOB_TYPE= OFFLINE RESOURCE_GROUPS=" casrg2 conc_ rg1"

conc_ rg1:process_resources[1476] JOB_TYPE= OFFLINE RESOURCE_GROUPS= casrg2 conc_ rg1

conc_ rg1 :process_resources[1478] RC= 0
conc_ rg1 :process_resources[1479] set +a
conc_ rg1 :process_resources[1481] [ 0 -ne 0 ]
conc_ rg1 :process_resources[1704] set_resource_group_state DOWN
```

#### **JOB\_TYPE=ERROR:**

エラーがリソースの獲得時または解放時に発生すると、**ERROR** ジョブ・タイプが実行されます。変数**RESOURCE\_GROUPS** には、現在のイベント時に獲得または解放が失敗したすべてのグループのリストが含まれます。これらのリソース・グループは、エラー状態に移動されます。獲得イベント時にこのジョブが実行されると、PowerHA SystemMirror はリソース・グループ獲得失敗回復機能を使用し、エラー状態の各リソース・グループごとに **rg\_move** イベントを起動します。

```
conc_ rg1: process_resources[1476] clRGPA
conc_ rg1: clRGPA[50] usingVer= clrgpa
conc_ rg1: clRGPA[55] clrgpa
conc_ rg1: clRGPA[56] exit 0
conc_ rg1: process_resources[1476] eval JOB_ TYPE= ERROR RESOURCE_GROUPS=" casrg1"
```

```
conc_rg1: process_resources[1476] JOB_TYPE= ERROR RESOURCE_GROUPS= cascrg1
conc_rg1: process_resources[1478] RC= 0
conc_rg1: process_resources[1479] set +a
conc_rg1: process_resources[1481] [ 0 -ne 0 ]
conc_rg1: process_resources[1712] set_resource_group_state ERROR
```

関連情報:

クラスター・イベントでのリソース・グループの動作

#### **JOB\_TYPE=NONE:**

現在の **process\_resources** スクリプトに対するすべての処理が完了すると、最後のジョブ・タイプ **NONE** を使用して、処理が完了しスクリプトの戻りが可能になったことを示します。 **process\_resources** スクリプトは、このジョブを受け取った後に終了すると、成功を表す **0** を必ず返します。

```
conc_rg1: process_resources[1476] c1RGPA
conc_rg1: c1RGPA[48] [[ high = high ]]
conc_rg1: c1RGPA[48] version= 1.16
conc_rg1: c1RGPA[50] usingVer= clrgpa
conc_rg1: c1RGPA[55] clrgpa
conc_rg1: c1RGPA[56] exit 0
conc_rg1: process_resources[1476] eval JOB_TYPE= NONE
conc_rg1: process_resources[1476] JOB_TYPE= NONE
conc_rg1: process_resources[1478] RC= 0
conc_rg1: process_resources[1479] set +a
conc_rg1: process_resources[1481] [ 0 -ne 0 ]
conc_rg1: process_resources[1721] break
conc_rg1: process_resources[1731] exit 0
```

#### **JOB\_TYPE=ACQUIRE:**

**ACQUIRE** ジョブ・タイプはリソース・グループ獲得イベントの始めに発生します。 **JOB\_TYPE=ACQUIRE** で **hacmp.out** を検索し、**RESOURCE\_GROUPS** 変数の値を表示して、イベント時に並列処理で獲得されるリソース・グループのリストを確認します。

```
:process_resources[1476] c1RGPA
:c1RGPA[48] [[ high = high ]]
:c1RGPA[48] version= 1.16
:c1RGPA[50] usingVer= clrgpa
:c1RGPA[55] clrgpa
:c1RGPA[56] exit 0
:process_resources[1476] eval JOB_TYPE= ACQUIRE RESOURCE_GROUPS=" cascrg1 cascrg2"
:process_resources[1476] JOB_TYPE= ACQUIRE RESOURCE_GROUPS= cascrg1 cascrg2
:process_resources[1478] RC= 0
:process_resources[1479] set +a
:process_resources[1481] [ 0 -ne 0 ]
:process_resources[1687] set_resource_group_state ACQUIRING
```

#### **JOB\_TYPE=RELEASE:**

**RELEASE** ジョブ・タイプはリソース・グループ解放イベントの始めに発生します。 **JOB\_TYPE=RELEASE** で **hacmp.out** を検索し、**RESOURCE\_GROUPS** 変数の値を表示して、イベント時に並列処理で解放されるリソース・グループのリストを確認します。

```
:process_resources[1476] c1RGPA
:c1RGPA[48] [[ high = high ]]
:c1RGPA[48] version= 1.16
:c1RGPA[50] usingVer= clrgpa
:c1RGPA[55] clrgpa
:c1RGPA[56] exit 0
:process_resources[1476] eval JOB_TYPE= RELEASE RESOURCE_GROUPS=" cascrg1 cascrg2"
:process_resources[1476] JOB_TYPE= RELEASE RESOURCE_GROUPS= cascrg1 cascrg2
```

```
:process_resources[1478] RC= 0
:process_resources[1479] set +a
:process_resources[1481] [ 0 -ne 0 ]
:process_resources[1691] set_resource_group_state RELEASING
```

#### **JOB\_TYPE= SSA\_FENCE:**

**SSA\_FENCE** ジョブ・タイプを使用して、SSA ディスクのフェンシングおよびアンフェンシングを処理します。変数 **ACTION** は、**HDISKS** 変数にリストされたディスクに実行する必要がある処理を示します。すべてのリソース・グループ (並列およびシリアル) は、ディスク・フェンシングにこのメソッドを使用します。

```
:process_resources[1476] c1RGPA FENCE
:c1RGPA[48] [[ high = high ]]
:c1RGPA[55] clrgpa FENCE
:c1RGPA[56] exit 0
:process_resources[1476] eval JOB_TYPE= SSA_FENCE ACTION= ACQUIRE
HDISKS=" hdisk6" RESOURCE_GROUPS=" conc_rgl " HOSTS=" electron"
:process_resources[1476] JOB_TYPE= SSA_FENCE ACTION= ACQUIRE
HDISKS= hdisk6 RESOURCE_GROUPS= conc_rgl HOSTS=electron
:process_resources[1478] RC= 0
:process_resources[1479] set +a
:process_resources[1481] [ 0 -ne 0 ]
:process_resources[1675] export GROUPNAME= conc_rgl conc_rgl
:process_resources[1676] process_ssa_fence ACQUIRE
```

注: ディスク・フェンシングは **process\_resources** スクリプトを使用します。したがって、ディスク・フェンシングが発生すると、誤ってリソース処理が行われているとみなす場合がありますが、実際にはディスク・フェンシングのみが行われています。ディスク・フェンシングが有効な場合、**hacmp.out** ファイルでは、ディスク・フェンシング処理がどのリソース・グループ処理よりも前に 行われていることがわかります。SSA ディスク・フェンシングは **process\_resources** スクリプトにより処理されますが、リソース・グループはシリアルに処理されます。**cl\_ssa\_fence** は、ディスク・フェンシングが必要なリソース・グループごとに一度呼び出されます。**hacmp.out** のコンテンツは、処理されるリソース・グループを示します。

```
conc_rgl: process_resources[8] export GROUPNAME
conc_rgl: process_resources[10] get_list_head hdisk6
conc_rgl: process_resources[10] read LIST_OF_HDISKS_FOR_RG
conc_rgl: process_resources[11] read HDISKS
conc_rgl: process_resources[11] get_list_tail hdisk6
conc_rgl: process_resources[13] get_list_head electron
conc_rgl: process_resources[13] read HOST_FOR_RG
conc_rgl: process_resources[14] get_list_tail electron
conc_rgl: process_resources[14] read HOSTS
conc_rgl: process_resources[18] cl_ssa_fence ACQUIRE electron hdisk6
conc_rgl: cl_ssa_fence[43] version= 1. 9. 1. 2
conc_rgl: cl_ssa_fence[44]
conc_rgl: cl_ssa_fence[44]
conc_rgl: cl_ssa_fence[46] STATUS= 0
conc_rgl: cl_ssa_fence[48] (( 3 < 3
conc_rgl: cl_ssa_fence[56] OPERATION= ACQUIRE
```

#### **JOB\_TYPE=SERVICE\_LABELS:**

**SERVICE\_LABELS** ジョブ・タイプは、サービス・ラベルの獲得または解放を処理します。変数 **ACTION** は、**IP\_LABELS** 変数にリストされたサービス IP ラベルに実行する必要がある処理を示します。

```
conc_rgl: process_resources[ 1476] c1RGPA
conc_rgl: c1RGPA[ 55] clrgpa
conc_rgl: c1RGPA[ 56] exit 0
conc_rgl: process_resources[ 1476] eval JOB_TYPE= SERVICE_LABELS
```

```

ACTION= ACQUIRE IP_LABELS=" elect_svc0: shared_svc1, shared_svc2"
RESOURCE_GROUPS=" cascrgr1 rotrgr1" COMMUNICATION_LINKS=": commlink1"
conc_rgl: process_resources[1476] JOB_TYPE= SERVICE_LABELS
ACTION= ACQUIRE IP_LABELS= elect_svc0: shared_svc1, shared_svc2
RESOURCE_GROUPS= cascrgr1 rotrgr1 COMMUNICATION_LINKS=: commlink1
conc_rgl: process_resources[1478] RC= 0
conc_rgl: process_resources[1479] set +a
conc_rgl: process_resources[1481] [ 0 -ne 0 ]
conc_rgl: process_resources[ 1492] export GROUPNAME= cascrgr1

```

このジョブ・タイプの場合、**acquire\_service\_addr** イベントが起動されます。イベント内で、個々のサービス・ラベルが獲得されます。**hacmp.out** ファイルのコンテンツは、処理されるリソース・グループを示します。各リソース・グループ内では、イベント・フローはシリアル処理の場合と同じです。

```

cascrgr1: acquire_service_addr[ 251] export GROUPNAME
cascrgr1: acquire_service_addr[251] [[ true = true ]]
cascrgr1: acquire_service_addr[254] read SERVICELABELS
cascrgr1: acquire_service_addr[254] get_list_head electron_svc0
cascrgr1: acquire_service_addr[255] get_list_tail electron_svc0
cascrgr1: acquire_service_addr[255] read IP_LABELS
cascrgr1: acquire_service_addr[257] get_list_head
cascrgr1: acquire_service_addr[257] read SNA_CONNECTIONS
cascrgr1: acquire_service_addr[258] export SNA_CONNECTIONS
cascrgr1: acquire_service_addr[259] get_list_tail
cascrgr1: acquire_service_addr[259] read _SNA_CONNECTIONS
cascrgr1: acquire_service_addr[270] clgetif -a electron_svc0

```

#### **JOB\_TYPE=VGS:**

**VGS** ジョブ・タイプはボリューム・グループの獲得または解放を処理します。変数 **ACTION** は処理されるボリューム・グループに実行する必要がある処理を示します。ボリューム・グループ名は **VOLUME\_GROUPS** および **CONCURRENT\_VOLUME\_GROUPS** 変数にリストされています。

```

conc_rgl :process_resources[1476] clRGPA
conc_rgl :clRGPA[55] clrgpa
conc_rgl :clRGPA[56] exit 0

```

```

conc_rgl :process_resources[1476] eval JOB_TYPE= VGS ACTION= ACQUIRE
CONCURRENT_VOLUME_GROUP=" con_vg6" VOLUME_GROUPS=""
casc_vg1: casc_vg2" RESOURCE_GROUPS=" cascrgr1 cascrgr2 "
EXPORT_FILESYSTEM=""

```

```

conc_rgl :process_resources[1476] JOB_TYPE= VGS
ACTION= ACQUIRE CONCURRENT_VOLUME_GROUP= con_vg6 VOLUME_GROUPS= casc_vg1: casc_vg2
RESOURCE_GROUPS= cascrgr1 cascrgr2 EXPORT_FILESYSTEM=""

```

```

conc_rgl :process_resources[1478] RC= 0
conc_rgl :process_resources[1481] [ 0 -ne 0 ]
conc_rgl :process_resources[1529]
export GROUPNAME= cascrgr1 cascrgr2

```

このジョブ・タイプは、個別のボリューム・グループを獲得する **cl\_activate\_vgs** イベント・ユーティリティ・スクリプトを実行します。**hacmp.out** ファイルのコンテンツは、処理されるリソース・グループを示します。各リソース・グループ内では、スクリプト・フローはシリアル処理の場合と同じです。

```

cascrgr1 cascrgr2 :cl_activate_vgs[256] 1> /usr/ es/ sbin/ cluster/ etc/ lsvg. out. 21266 2> /tmp/ lsvg. err

```

```

cascrgr1: cl_activate_vgs[260] export GROUPNAME
cascrgr1: cl_activate_vgs[262] get_list_head
casc_vg1: casc_vg2
cascrgr1: cl_activate_vgs[ 62] read_LIST_OF_VOLUME_GROUPS_FOR_RG
cascrgr1: cl_activate_vgs[263] get_list_tail casc_vg1: casc_vg2
cascrgr1: cl_activate_vgs[263] read VOLUME_GROUPS
cascrgr1: cl_activate_vgs[265] LIST_OF_VOLUME_GROUPS_ FOR_ RG=

```

```

cascrg1: cl_activate_vgs[ 270] fgrep -s -x casc_ vg1 /usr/ es/ sbin/
cluster/ etc/ lsvg. out. 21266
cascrg1: cl_activate_vgs[275] LIST_OF_VOLUME_GROUPS_FOR_RG= casc_vg1
cascrg1: cl_activate_vgs[275] [[ casc_ vg1 = ]]

```

ディスク・フェンシング:

ディスク・フェンシングでは、JOB\_TYPE=SSA\_FENCE を指定した **process\_resources** スクリプトを使用します。

従属リソース・グループまたはサイトを有するクラスター内の処理:

従属グループまたはサイトを設定したクラスターのリソース・グループは、動的イベント変換で処理されま

す。これらのイベントは 1 つ以上のリソース・グループを同時に処理します。複数の非コンカレント・リソース・グループは、1 つの **rg\_move** イベント内で処理できます。

関連情報:

アプリケーションおよび PowerHA SystemMirror

**Siblings** を持つイベントの **hacmp.out** への出力のサンプル:

このトピックでは、兄弟イベントを持つイベントのサンプル出力 (**hacmp.out** への出力) を示します。

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Mar 28 09:40:42 EVENT START: rg_move a2 1ACQUIRE
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
:process_resources[1952] eval JOB_TYPE=ACQUIRE RESOURCE_GROUPS="rg3"
SIBLING_GROUPS="rg1 rg3" SIBLING_NODES_BY_GROUP="b2 : b2"
SIBLING_ACQUIRING_GROUPS="" SIBLING_ACQUIRING_NODES_BY_GROUP=""
PRINCIPAL_ACTION="ACQUIRE" AUXILLIARY_ACTION="NONE"
:process_resources[1952] JOB_TYPE=ACQUIRE RESOURCE_GROUPS=rg3
SIBLING_GROUPS=rg1 rg3 SIBLING_NODES_BY_GROUP=b2 : b2
SIBLING_ACQUIRING_GROUPS= SIBLING_ACQUIRING_NODES_BY_GROUP=
PRINCIPAL_ACTION=ACQUIRE AUXILLIARY_ACTION=NONE
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
:rg_move_complete[157] eval FORCEDOWN_GROUPS="" RESOURCE_GROUPS=""
HOMELESS_GROUPS="" ERRSTATE_GROUPS="" PRINCIPAL_ACTIONS=""
ASSOCIATE_ACTIONS="" AUXILLIARY_ACTIONS="" SIBLING_GROUPS="rg1 rg3"
SIBLING_NODES_BY_GROUP="b2 : b2" SIBLING_ACQUIRING_GROUPS="" SIBLING
_ACQUIRING_NODES_BY_GROUP="" SIBLING_RELEASING_GROUPS=""
SIBLING_RELEASING_NODES_BY_GROUP=""
:rg_move_complete[157] FORCEDOWN_GROUPS= RESOURCE_GROUPS=
HOMELESS_GROUPS= ERRSTATE_GROUPS= PRINCIPAL_ACTIONS= ASSOCIATE_ACTIONS=
AUXILLIARY_ACTIONS= SIBLING_GROUPS=rg1 rg3 SIBLING_NODES_BY_GROUP=b2 :
b2 SIBLING_ACQUIRING_GROUPS= SIBLING_ACQUIRING_NODES_BY_GROUP
= SIBLING_RELEASING_GROUPS= SIBLING_RELEASING_NODES_BY_GROUP=
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
:process_resources[1952] eval JOB_TYPE=SYNC_VGS ACTION=ACQUIRE
VOLUME_GROUPS="vg3,vg3sm" RESOURCE_GROUPS="rg3 "
:process_resources[1952] JOB_TYPE=SYNC_VGS
ACTION=ACQUIRE_VOLUME_GROUPS=vg3,vg3sm RESOURCE_GROUPS=rg3
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
rg3:process_resources[1952] eval JOB_TYPE=ONLINE RESOURCE_GROUPS="rg3"
rg3:process_resources[1952] JOB_TYPE=ONLINE RESOURCE_GROUPS=rg3
rg3:process_resources[1954] RC=0
rg3:process_resources[1955] set +a
rg3:process_resources[1957] [ 0 -ne 0 ]
rg3:process_resources[2207] set_resource_group_state UP
rg3:process_resources[3] STAT=0
rg3:process_resources[6] export GROUPNAME
rg3:process_resources[7] [ UP != DOWN ]

```

```

rg3:process_resources[9] [ REAL = EMUL ]
rg3:process_resources[14] clchdaemons -d clstrmgr_scripts -t resource_locator -n a1 -o rg3 -v UP
rg3:process_resources[15] [ 0 -ne 0 ]
rg3:process_resources[26] [ UP = ACQUIRING ]
rg3:process_resources[31] [ UP = RELEASING ]
rg3:process_resources[36] [ UP = UP ]
rg3:process_resources[38] cl_RMupdate rg_up rg3 process_resources
Reference string: Sun.Mar.27.18:02:09.EST.2005.process_resources.rg3.ref
rg3:process_resources[39] continue
rg3:process_resources[80] return 0
rg3:process_resources[1947] true
rg3:process_resources[1949] set -a
rg3:process_resources[1952] clRGPA
rg3:clRGPA[33] [[ high = high ]]
rg3:clRGPA[33] version=1.16
rg3:clRGPA[35] usingVer=clrgpa
rg3:clRGPA[40] clrgpa
rg3:clRGPA[41] exit 0
rg3:process_resources[1952] eval JOB_TYPE=NONE
rg3:process_resources[1952] JOB_TYPE=NONE
rg3:process_resources[1954] RC=0
rg3:process_resources[1955] set +a
rg3:process_resources[1957] [ 0 -ne 0 ]
rg3:process_resources[2256] break
rg3:process_resources[2267] [[ FALSE = TRUE ]]
rg3:process_resources[2273] exit 0
:rg_move_complete[346] STATUS=0
:rg_move_complete[348] exit 0
Mar 27 18:02:10 EVENT COMPLETED: rg_move_complete a1 2 0

```

## ノードの **PowerHA SystemMirror** ログ・ファイル・パラメーターの管理

各クラスター・ノードは 2 つのログ・ファイル・パラメーターをサポートします。

これらのパラメーターによって、以下のことを実行できます。

- **PowerHA SystemMirror** スクリプトによるデバッグ情報出力のレベルの設定。デフォルトでは、**PowerHA SystemMirror** によってデバッグ情報パラメーターが「high (高)」に設定され、詳細なスクリプトの実行結果出力が作成されます。
- **hacmp.out** ログ・ファイルの出力フォーマットを設定します。

ノードのログ・ファイル・パラメーターを変更するには、以下の手順を実行します。

1. `smit hacmp` と入力します。
2. SMIT で、「**Problem Determination Tools (問題判別ツール)**」 > **PowerHA SystemMirror 「Log Viewing and Management (ログの表示および管理)」** > **Change/Show PowerHA SystemMirror Log File Parameters (PowerHA SystemMirror ログ・ファイル・パラメーターの変更/表示)** を選択して、`Enter` を押します。
3. リストからノードを選択します。
4. 以下のフィールド値を入力します。

表 12. PowerHA SystemMirror ログ・ファイル・パラメーターの変更/表示のフィールド

フィールド	値
Debug Level (デバッグ・レベル)	クラスター・イベント・スクリプトには、2 つのログ・レベルがあります。低レベルは、スクリプト実行中に発生したイベントおよびエラーのみ記録します。高 (デフォルト) レベルは、スクリプトが実行したすべてのコマンドを記録します。強くお勧めするレベルです。高レベルは多くのクラスターの問題を解決するためのスクリプト追跡レベルを提供します。
Formatting options for hacmp.out (hacmp.out の形式オプション)	「Default (None) (デフォルト (なし))」(特定の形式なし)、 「Standard (標準)」(検索文字列を含む)、 「HTML (Low) (HTML (低))」(HTML 制限形式)、および 「HTML (High) (HTML (高))」(HTML 完全形式) からいずれか 1 つを選択します。

5. Enter を押して、AIX 構成データベースの PowerHA SystemMirror に値を入力します。
6. PowerHA SystemMirror のメインメニューに戻ります。「Extended Configuration (拡張構成)」> 「Extended Verification and Synchronization (拡張検証および同期化)」を選択します。

任意のクラスター・ノードでクラスター・サービスが実行されているかどうかを検査されます。実行されている場合、検証をスキップするオプションはありません。

7. 検証に使用するオプションを選択して Enter を押し、クラスター全体でクラスター構成とノード環境を同期します。

関連情報:

PowerHA SystemMirror クラスターの検査および同期化

## clcomd のログ

clcomd デーモンのログを **clcomd.log** および **clcomddiag.log** に記録する処理はデフォルトでオンになっています。

**clcomd.log** 内の情報には、ディスクバリー実行時に確立された最初の接続情報を含むデーモンとのすべての接続情報が含まれています。**clcomddiag.log** には、デーモンの診断情報が記載されているため、トラブルシューティングではこのファイルは通常使用しません。

以下の例は、**clcomd.log** ファイルで生成される出力タイプを示します。2 番目と 3 番目のエントリは、ディスクバリー・プロセス中に生成されます。

```
Wed May 7 12:43:13 2003: Daemon was successfully started
Wed May 7 12:44:10 2003: Trying to establish connection to node
temporarynode0000001439363040
Wed May 7 12:44:10 2003: Trying to establish connection to node
temporarynode0000002020023310
Wed May 7 12:44:10 2003: Connection to node temporarynode0000002020023310, success, 192.0.24.4->
Wed May 7 12:44:10 2003: CONNECTION: ACCEPTED: test2: 192.0.24.4->192.0.24.4
Wed May 7 12:44:10 2003: WARNING: /usr/es/sbin/cluster/etc/rhosts permissions
must be -rw-----
Wed May 7 12:44:10 2003: Connection to node temporarynode0000001439363040: closed
Wed May 7 12:44:10 2003: Connection to node temporarynode0000002020023310: closed
Wed May 7 12:44:10 2003: CONNECTION: CLOSED: test2: 192.0.24.4->192.0.24.4
Wed May 7 12:44:11 2003: Trying to establish connection to node test1
Wed May 7 12:44:11 2003: Connection to node test1, success, 192.0.24.4->192.0.24.5
Wed May 7 12:44:11 2003: Trying to establish connection to node test3.
```

AIX の **vi** コマンドまたは **more** コマンドを使用すれば、**clcomd.log** ファイルまたは **clcomddiag.log** ファイルの内容を表示できます。

AIX の `tracesoff` コマンドを使用して、`clcomddiag.log` へのログ記録を一時的にオフにできます (次の再起動またはこのコンポーネントへのログ記録を再度有効にするまで)。 `clcomddiag.log` へのログ記録を永続的に停止するには、以下のコマンドを使用して、`-d` フラグなしで SRC からデーモンを起動します。

```
chssys -s clcomd -a ""
```

## PowerHA SystemMirror クラスター・ログ・ファイルのリダイレクト

PowerHA SystemMirror は通常の操作中に、システムのモニターやデバッグに使用できる各種の出力ログ・ファイルを作成します。必要な場合には、クラスター・ログをデフォルト・ディレクトリー以外のロケーションに保管することもできます。 そのように保管する場合は、ほとんどのクラスター・ログの最小ディスク・スペースは 2MB であることに留意してください。 `hacmp.out` には 14MB が推奨されます。

注: ログはローカル・ファイル・システムにリダイレクトする必要があります。共有したり NFS ファイル・システムにリダイレクトすることはできません。 それらのファイル・システムにログを置くと、フォールオーバー・イベント時に、そのファイル・システムをアンマウントする必要がある場合に、問題が発生するおそれがあります。 また、ログを NFS ファイル・システムにリダイレクトした場合、ノードの再統合時にクラスター・サービスが開始できなくなることがあります。

このログ・ファイルのリダイレクト機能は、以下のように実行されます。

- ターゲット・ディレクトリーの場所を検査し、そのディレクトリーがローカルとリモートのどちらのファイル・システムに属しているかを判別します。
- ターゲット・ディレクトリーが PowerHA SystemMirror によって管理されているかどうかを判別するための検査が実施されます。管理されている場合、ログ・ファイルをリダイレクトする試みは、すべて失敗します。
- ターゲット・ディレクトリーが、相対パス (「`mylogdir`」など) ではなく、絶対パス (「`/mylogdir`」など) により指定されていることを確認します。

これらの検査によって、選択されたファイル・システムが予期せず使用できなくなる可能性が低くなります。

注: ターゲット・ディレクトリーに読み取り書き込みアクセスを持つ必要があります。

---

## システム・コンポーネント

以下のトピックでは、システム・コンポーネントを調査する手順を示し、PowerHA SystemMirror の使用時に発生する可能性がある問題を特定し、考えられる解決策を提示します。

コンソールにエラー・メッセージが表示されず、ログ・ファイルを調査しても有用な情報が得られない場合は、PowerHA SystemMirror 環境の各コンポーネントを調査し、問題の原因となるコンポーネントを除去します。

## システム・コンポーネントの調査

PowerHA SystemMirror と AIX のいずれにも、PowerHA SystemMirror クラスターとそのクラスター内のリソースの状態を判別するために使用できるユーティリティーが用意されています。これらのコマンドを使用すると、ボリューム・グループやネットワークに関する情報を収集できます。

この場合、PowerHA SystemMirror システムに関する知識が必要です。あらかじめ通常のクラスターの特性を理解し、クラスター・コンポーネントを調査するときに通常の状態からの逸脱を見つける必要があります。多くの場合、残っているクラスター・ノードが、システム・パラメーターやその他のクラスター構成情報の正しい設定の例になります。

確認可能な PowerHA SystemMirror クラスター・コンポーネントを検討し、有用なユーティリティーを挙げておいてください。クラスター・ログ・ファイルを検査しても問題の原因が判別されない場合は、各層をたどるトップダウン方法で各システム・コンポーネントを検査する必要があります。コンポーネントは、必ず以下の順序で調査してください。

1. アプリケーション層
2. PowerHA SystemMirror 層
3. 論理ボリューム・マネージャー層
4. TCP/IP 層
5. AIX 層
6. 物理ネットワーク層
7. 物理ディスク層
8. システム・ハードウェア層

各層を調べるときに期待すべき内容、および層の調査に使用すべきツールについても知っておく必要があります。

## 高可用性アプリケーションの確認

クラスターに影響を及ぼす問題を検出するための最初のステップとして、クラスターで実行されている各高可用性アプリケーションを検査します。アプリケーション固有のログ・ファイルをすべて調査し、各アプリケーションの資料で推奨されているトラブルシューティング手順をすべて実行します。

さらに、以下の点についても検査します。

- 例えば、記録の追加や削除を試行するデータベース・アプリケーション用の簡単なテストを行います。
- **ps** コマンドを使用して必要なプロセスが実行されているかを検査したり、そのプロセスが正常に停止されたかどうかを検証します。
- アプリケーションが必要とするリソースが存在するかどうかを検査し、それらのリソースが使用可能であることを確認します (例: ファイル・システムやボリューム・グループ)。

## PowerHA SystemMirror 層の検査

アプリケーション層を検査しても問題の原因が判明しない場合は、PowerHA SystemMirror 層を検査します。

調査対象は以下の 2 つの領域です。

- PowerHA SystemMirror コンポーネントと必要なファイル
- クラスター・トポロジーと構成

注: これらのステップでは、ログ・ファイルをすでに検査しており、ログ・ファイルに問題が示されていないことを前提としています。

## PowerHA SystemMirror コンポーネントの検査

PowerHA SystemMirror クラスターは、いくつかの必須のファイルおよびデーモンから構成されます。以下のセクションでは、PowerHA SystemMirror 層で検査する対象について説明します。

## PowerHA SystemMirror で必要なファイルの確認

クラスターに必要な PowerHA SystemMirror ファイルが適切な場所に存在し、適切なアクセス権 (読み取り可能および実行可能) が設定されており、長さがゼロでないことを確認します。 PowerHA SystemMirror ソフトウェアによって変更される PowerHA SystemMirror ファイルおよび AIX ファイルは、製品に付属している README ファイルにリストされています。

### クラスター・サービスおよびプロセスの確認

以下の PowerHA SystemMirror デーモンの状況を検査します。

- クラスター・マネージャー (**clstrmgrES**) デーモン
- クラスター通信 (**clcomdES**) デーモン
- クラスター情報プログラム (**clinfoES**) デーモン

これらのコンポーネントが通常通りに応答しない場合、デーモンがクラスター・ノードでアクティブかどうか判別します。 SMIT の「**System Management (C-SPOC) (システム管理 (C-SPOC))**」「**Cluster Services (クラスター・サービス)**」「**Show Cluster Services (クラスター・サービスの表示)**」パネルのオプション、または **lssrc** コマンドのいずれかを使用します。

例えば、SRC で制御されるすべてのデーモンの状況を調べるには、以下のように入力します。

```
lssrc -a | grep active
syslogdras 290990active
sendmail mail270484active
portmapportmap286868active
inetd tcpip 295106active
snmpd tcpip 303260active
dpid2 tcpip 299162active
hostmibd tcpip 282812active
aixmibdtcpip 278670active
biodnfs 192646active
rpc.statd nfs 254122 active
rpc.lockd nfs 274584active
qdaemonspooler196720active
writesrv spooler250020active
ctrmc rsct98392 active
clcomdES clcomdES 204920active
IBM.CSMAgentRMrsc_rm90268 active
IBM.ServiceRM rsct_rm229510active
IBM.ERRM rsct_rm188602active
IBM.AuditRMrsc_rm151722active
topsvcstopsvcs602292active
grpsvcsgprsvcs569376active
emsvcs emsvcs 561188active
emaixosemsvcs 557102active
clstrmgrEScluster544802active
gsclvmd565356active
IBM.HostRMrsc_rm442380active
```

SRC で制御されるすべてのクラスター・デーモンの状況を調べるには、次のように入力します。 **lssrc -g cluster**

注: **lssrc** コマンドで **-g** フラグを使用すると、サブシステムが非アクティブの場合、状況情報にはサブシステムの状況が記載されません。 この情報が必要な場合は、代わりに **-a** フラグを使用してください。

**lssrc** コマンドの詳細については、マニュアル・ページを参照してください。

デーモンの状況の追加情報を表示させるため `clcheck_server` コマンドを実行します。 `clcheck_server` コマンドは、`lssrc` コマンドよりも詳細に繰り返し検査を試みます。詳しくは、`clcheck_server` マニュアル・ページを参照してください。

クラスター・マネージャーが実行されているかどうか、あるいは、クラスター・マネージャーによって起動されたプロセスが現在ノード上で実行されているかどうかを判別するには、`ps` コマンドを使用します。

例えば、`clstrmgrES` デーモンが実行されているかどうかを判別するには、以下のように入力します。

```
ps -ef | grep clstrmgrES
root 18363 3346 3 11:02:05 - 10:20 /usr/es/sbin/cluster/clstrmgrES
root 19028 19559 2 16:20:04 pts/10 0:00 grep clstrmgrES
```

このコマンドの使用について詳しくは、`ps` マニュアル・ページを参照してください。

## クラスター構成問題の検査

PowerHA SystemMirror クラスターが正常に機能するためには、クラスター・トポロジー、ネットワーク構成、および PowerHA SystemMirror リソースの所有権とテークオーバーが、クラスター内のすべてのノードで一致している必要があります。この情報は、各クラスター・ノードの構成データベースに格納されます。

構成の問題を調べる前に、システムを破損させるような変更を最近行ったかどうかを考えてみてください。コンポーネントの追加や削除を行いませんでしたか。新しいソフトウェアをマシンにロードしませんでしたか。新しい PTF やアプリケーションの更新を実行しませんでしたか。システムのバックアップを復元しませんでしたか。次に、検証を実行して AIX ソフトウェアに対して PowerHA SystemMirror 固有の変更が適切に実施され、クラスター構成が有効であることを確認します。

クラスターの検証ユーティリティーによって、多くのクラスター構成が多面的に検査され、不整合があれば報告されます。このユーティリティーを使用すると、次の診断タスクを実行できます。

- すべてのクラスター・ノードに、同じクラスター・トポロジーの情報が含まれていることを検証します。
- すべてのネットワーク・インターフェース・カードが適切に構成されており、所有するすべてのノードから共用ディスクにアクセスできるかどうかを検査します。
- 定義済みリソース (ファイルシステム、ログ・ファイル、ボリューム・グループ、ディスク、アプリケーション・コントローラーなど) の所有権が、すべてのノード間で一致しているかどうかを調べます。
- クラスター名、ノード名、ネットワーク名、ネットワーク・インターフェース名、リソース・グループ名に無効な文字がないかどうかを検査します。
- テークオーバー情報を検証します。

検証ユーティリティーは、以下に関する診断情報も出力します。

- ユーザー定義スナップショット・メソッド
- カスタム検証メソッド
- ユーザー定義のイベント前処理または後処理
- クラスター・ログ・ファイルのリダイレクト

メイン PowerHA SystemMirror SMIT パネルで、「**Problem Determination Tools (問題判定ツール)**」 > 「**PowerHA SystemMirror Verification (検証)**」 > 「**Verify PowerHA SystemMirror Configuration (構成の検証)**」を選択します。構成に関する問題が見つかったら、その問題を訂正してから、クラスターを再度同期化します。

注: エラーによっては、各クラスター・ノードで変更が必要な場合もあります。例えば、アプリケーションの始動スクリプトが欠落している場合やボリューム・グループが `autovaryon=TRUE` の場合は、影響を受ける各ノードで訂正を行う必要があります。PowerHA SystemMirror ファイル・コレクションを使用して、処理できる問題もあります。

クラスター・トポロジーの完全なリストを参照するには、`/usr/es/sbin/cluster/utilities/cltopinfo` コマンドを実行します。PowerHA SystemMirror 検証プロセスを実行するほか、ノード構成ファイルに対する最近の変更についても検査してください。

コマンド `ls -lt /etc` は、`/etc` ディレクトリー内の全てのファイルをリストし、AIX の構成に重要な最後に変更された、例えば以下のようなファイルを表示します。

- `etc/inetd.conf`
- `/etc/hosts`
- `etc/services`

必ず、検証プロセスで検出されない可能性があるエラーがリソース・グループの構成に含まれていないかどうかも検査してください。例えば、アプリケーション・コントローラーが必要とするファイルシステムがアプリケーションと共にリソース・グループに含まれているようにする必要があります。

各リソース・グループのノードが所定のノードであり、適切な順序でリストされているかどうかを検査します。メイン PowerHA SystemMirror SMIT パネルからクラスター・リソース構成の情報を表示するには、「**Extended Configuration (拡張構成)**」>「**Extended Resource Configuration (拡張リソース構成)**」>「**PowerHA SystemMirror Extended Resource Group Configuration (拡張リソース・グループ構成)**」>「**Show All Resources by Node or Resource Group (ノードまたはリソース・グループ別によりソースをすべて表示)**」を選択します。

また、`/usr/es/sbin/cluster/utilities/clRGinfo` コマンドを実行して、リソース・グループ情報を表示できます。

注: クラスターの検証ユーティリティーの実行後に、クラスター構成の問題が発生した場合、この環境では C-SPOC コマンドを実行しないでください。このコマンドはクラスター・ノードで実行できない場合があります。

関連情報:

PowerHA SystemMirror クラスターの検査および同期化

## クラスター・スナップショット・ファイルの検査

PowerHA SystemMirror クラスター・スナップショット機能 (`/usr/es/sbin/cluster/utilities/clsnapshots`) を使用すると、特定のクラスター構成を定義するすべてのデータのレコードをファイルに保存できます。また、独自のユーザー定義スナップショット・メソッドを作成し、構成に対して重要な追加情報を保管することもできます。このスナップショットは、クラスターに関する問題のトラブルシューティングに使用できます。

スナップショットを保管または検索するためのデフォルトのディレクトリー・パスは、`/usr/es/sbin/cluster/snapshots` です。

クラスター・スナップショット機能は、異なるバージョンの PowerHA SystemMirror が並行して稼働しているクラスター内では使用できません。

関連情報:

クラスター構成の保管および復元

クラスター・スナップショットに保管される情報:

クラスター・スナップショットに保管される主要な情報は、PowerHA SystemMirror 構成データベースのクラス (HACMPcluster、HACMPnode、および HACMPnetwork など) に保管された情報です。この情報は、クラスター・スナップショットを適用する場合、クラスター構成を再作成するために使用されます。

クラスター・スナップショットには、ユーザーがカスタマイズしたスクリプト、アプリケーション、および PowerHA SystemMirror 用でないその他の構成パラメーターは保存されません。例えば、アプリケーション・コントローラーの名前やその始動スクリプトおよび停止スクリプトのロケーションは、PowerHA SystemMirror サーバー構成データベースのオブジェクト・クラスに保管されます。ただし、スクリプトそのものと、スクリプトが呼び出す可能性のあるアプリケーションは、保管されません。

また、クラスター・スナップショットでは、PowerHA SystemMirror の適用範囲外のデバイス・データや構成固有のデータは保管されません。例えば、共用ファイル・システムおよびボリューム・グループの名前は保管されますが、NFS オプションや LVM ミラーリング構成などの他の詳細情報は保管されません。

リソース・グループ管理ユーティリティー **clRGmove** を使用してリソース・グループを移動した場合に、スナップショットを適用すると、リソース・グループの動作はデフォルト・ノード・リストで指定されたものに戻ります。スナップショットの適用後にクラスターを調査するには、**clRGinfo** を実行してリソース・グループのロケーションおよび状況を表示します。

この ODM データのほか、クラスター・スナップショットには、さまざまな PowerHA SystemMirror や標準 AIX のコマンドやユーティリティーによって生成される出力も組み込まれます。このデータには、各クラスター・ノードによって表示されるクラスター、ノード、ネットワーク、およびネットワーク・インターフェースの現在の状態と、実行中の PowerHA SystemMirror デーモンの状態も含まれます。

クラスター・スナップショットには、以下のコマンドからの出力が含まれます。

- **cllscf**
- **df**
- **lsfs**
- **netstat**
- **cllsnw**
- **exportfs**
- **lspp**
- **no**
- **cllsif**
- **ifconfig**
- **lslv**
- **clchsyncd**
- **clshowres**
- **ls**
- **lsvg**
- **cltopinfo**

ログの収集をスキップすると、スナップショットのサイズを縮小し、スナップショット・ユーティリティーの実行をスピード・アップできます。

SMIT を使用してクラスター・ログ・ファイルを集集し、問題を報告できます。このオプションは、SMIT メニューの「**Problem Determination Tools (問題判別ツール)**」>「**PowerHA SystemMirror Log Viewing and Management (PowerHA SystemMirror ログの表示および管理)**」>「**Collect Cluster log files for Problem Reporting (問題を報告するためのクラスター・ログ・ファイルの収集)**」で使用できます。IBM サポート担当者より依頼がある場合にのみ、このオプションを使用することをお勧めします。

また、AIX `snap -e` コマンドを使用して、`hacmp.out` や `clstrmgr.debug` ログ・ファイルなどの PowerHA SystemMirror クラスター・データを収集できます。

関連情報:

クラスター構成の保管および復元

クラスター・スナップショット・ファイル:

クラスター・スナップショット機能は、保管するデータを構成データベース・データ・ファイルとクラスター状態の情報ファイルの 2 つのファイルに格納します。それぞれのファイルは 3 つのセクションで情報を示します。

構成データベース・データ・ファイル (`.odm`):

このファイルには、そのクラスターの PowerHA SystemMirror ODM オブジェクト・クラスに格納されたすべてのデータが入ります。

このファイルには、ファイル拡張子が `.odm` のユーザー定義のベース名が与えられます。構成データベース情報はすべてのクラスター・ノード上でほとんど同じであるはずなので、クラスター・スナップショットは、1 つのノードの値のみを保管します。クラスター・スナップショットの構成データベース・データ・ファイルは、ASCII テキスト・ファイルで、次の 3 つのセクションがあります。

表 13. データベース・データ・ファイル (`.odm`) のセクション

セクション	説明
バージョン・セクション	このセクションは、クラスター・スナップショットのバージョンを示します。<VER という文字でこのセクションの始まりを示し、</VER という文字でこのセクションの終わりを示します。クラスター・スナップショット・ソフトウェアはバージョン番号を設定します。
説明セクション	このセクションには、クラスター・スナップショットを説明するユーザー定義のテキストが入ります。最大 255 文字までの説明文を指定することができます。<DSC という文字でこのセクションの始まりを示し、</DSC という文字でこのセクションの終わりを示します。
ODM データ・セクション	このセクションには、汎用 AIX ODM スタンザ形式の PowerHA SystemMirror ODM オブジェクト・クラスが含まれます。<ODM という文字でこのセクションの始まりを示し、</ODM という文字でこのセクションの終わりを示します。

以下の例は、サンプル・クラスター・スナップショット構成データベース・データ・ファイルからの抜粋で、保管されている ODM スタンザの一部を示しています。

```
<VER
1.0
</VER
```

```
<DSC
My Cluster Snapshot
</DSC
```

```
<ODM
```

```
PowerHA SystemMirror cluster:
  id = 1106245917
  name = "HA52_TestCluster"
```

```

nodename = "mynode"
sec_level = "Standard"
sec_level_msg = ""
sec_encryption = ""
sec_persistent = ""
last_node_ids = ""
highest_node_id = 0
last_network_ids = ""
highest_network_id = 0
last_site_ids = ""
highest_site_id = 0
handle = 1
cluster_version = 7
reserved1 = 0
reserved2 = 0
wlm_subdir = ""
settling_time = 0
rg_distribution_policy = "node"
noautoverification = 0
clvernodename = ""
clverhour = 0

```

```

PowerHA SystemMirror node:
name = "mynode"
object = "VERBOSE_LOGGING"
value = "high"
.
.
</ODM

```

#### クラスター状態情報ファイル (.info):

このファイルには、AIX と PowerHA SystemMirror の標準システム管理コマンドの出力が格納されます。

このファイルには、ファイル拡張子が **.info** のユーザー定義のベース名が与えられます。ユーザー定義スナップショット・メソッドを定義した場合は、その出力がこのファイルに追加されます。クラスター状態の情報ファイルには、以下の 3 つのセクションがあります。

表 14. クラスター状態情報ファイル (.info)

セクション	説明
バージョン・セクション	このセクションは、クラスター・スナップショットのバージョンを示します。 <VER という文字でこのセクションの始まりを示し、</VER という文字でこのセクションの終わりを示します。クラスター・スナップショット・ソフトウェアはこのセクションを設定します。
説明セクション	このセクションには、クラスター・スナップショットを説明するユーザー定義のテキストが入ります。最大 255 文字までの説明文を指定することができます。<DSC という文字でこのセクションの始まりを示し、</DSC という文字でこのセクションの終わりを示します。
コマンド出力セクション	このセクションには、AIX および PowerHA SystemMirror ODM コマンドによって生成された出力が入ります。このセクションには、実行されたコマンドおよび関連した出力がリストされます。このセクションは、区切り文字などで区切られません。

## 論理ボリューム・マネージャーの確認

PowerHA SystemMirror クラスターのトラブルシューティングを行う場合、ボリューム・グループ、物理ボリューム、論理ボリューム、およびファイルシステムの LVM エンティティーを調べる必要があります。

## ボリューム・グループ定義の検査

クラスター内のすべての共有ボリューム・グループが適切なノードでアクティブであるかどうかを検査します。ボリューム・グループがアクティブでない場合は、構成に応じて適切なコマンドを使用してオンに変更します。

SMIT パネルで、「**Initialization and Standard Configuration (初期化および標準構成)**」>「**Configure PowerHA SystemMirror Resource Groups (PowerHA SystemMirror リソース・グループの構成)**」>「**Change/Show Resources for a Resource Group (standard) (リソース・グループのリソースの変更/表示 (標準))**」を選択し、リソース・グループの「**Volume Groups (ボリューム・グループ)**」フィールドにリストされているすべてのボリューム・グループを、そのリソース・グループがオンラインのノード上でオンに変更する必要があります。

## ボリューム・グループを検査する **lsvg** コマンドの使用

クラスター・ノードのボリューム・グループ定義間の不整合を検査するには、**lsvg** コマンドを使用してクラスターの各ノードに定義されたボリューム・グループの情報を表示します。

```
lsvg
```

システムは、以下のようなボリューム・グループ情報を戻します。

```
rootvg  
datavg
```

システムのアクティブな (オンに変更された) ボリューム・グループのみを表示するには、**lsvg -o** コマンドを以下のように使用します。

```
lsvg -o
```

システムは、以下のようなボリューム・グループ情報を戻します。

```
rootvg
```

ボリューム・グループのすべての論理ボリュームをリストし、ボリューム・グループ状態および属性を検査するには、**lsvg -l** コマンドを使用して、以下の例で示すようにボリューム・グループ名を指定します。

```
lsvg -l rootvg
```

注: **lsvg-l** コマンドが使用できるように、ボリューム・グループをオンに変更する必要があります。

また、PowerHA SystemMirror SMIT を使用して矛盾点を検査することもできます。「**System Management (C-SPOC) (システム管理 (C-SPOC))**」>「**PowerHA SystemMirror Logical Volume Management (PowerHA SystemMirror 論理ボリューム管理)**」>「**Shared Volume Groups (共有ボリューム・グループ)**」オプションを選択してクラスターの共有ボリューム・グループの情報を表示します。

## ボリューム・グループの **varyon** 状態の確認

**lsvg < vgname >** コマンドを発行して、ボリューム・グループの状態を検査できます。

構成に応じて、**lsvg** コマンドは以下のオプションを戻します。

**vg state** は **active** (アクティブな varyon の場合)、または **passive only** (パッシブな varyon の場合) になります。

**vg mode** は **concurrent** または **enhanced concurrent** になります。

以下に **lsvg** 出力の例を示します。

```
# lsvg myvg

VOLUME GROUP: Volume_Group_01 VG IDENTIFIER: 0002231b00004c00000000f2801b1cc3
VG STATE: active PP SIZE: 16 megabyte(s)
VG PERMISSION:read/write TOTAL PPs: 1084 (17344 megabytes)
MAX LVs:256FREE PPs:977 (15632 megabytes)
LVs:4USED PPs:107 (1712 megabytes)
OPEN LVs: 0QUORUM:2
TOTAL PVs:2VG DESCRIPTORS:3
STALE PVs:0 STALE PPs 0
ACTIVE PVs: 2AUTO ON:no
MAX PPs per PV1016 MAX PVs: 32
LTG size: 128 kilobyte (s) AUTO SYNC:no
HOT SPARE:no
```

## C-SPOC ユーティリティーによる共用ボリューム・グループの確認

2 ノード C-SPOC 環境でクラスター・ノード上のボリューム・グループ定義間に不整合があるかどうかを確認するには、以下の手順を実行します。

1. smitty hacmp と入力します。
2. SMIT で、「**System Management (C-SPOC)** (システム管理 (C-SPOC))」>「**PowerHA SystemMirror Logical Volume Management (PowerHA SystemMirror 論理ボリューム管理)**」>「**Shared Volume Groups (共用ボリューム・グループ)**」>「**List All Shared Volume Groups (共用ボリューム・グループをすべてリスト)**」を選択し、Enter を押してデフォルト (「no (いいえ)」) を受け入れます。

C-SPOC 環境のすべての共用ボリューム・グループのリストが表示されます。このリストには、非コンカレント・リソース・グループのリソースとして含まれる拡張コンカレント・ボリューム・グループも含まれます。

また、コマンド行から C-SPOC `cl_lsvg` コマンドを使用して、この情報を表示できます。

## 物理ボリュームの検査

各ノードに定義された物理ボリュームの不整合を検査するには、システムが認識しているすべての物理ボリュームのリストを取得し、このリストを「**Command Status** (コマンドの状態)」画面の「**Disks** (ディスク)」フィールドに指定したディスクのリストと比較します。SMIT の「**Extended Configuration** (拡張構成)」>「**Extended Resource Configuration** (拡張リソース構成)」>「**PowerHA SystemMirror Extended Resource Group Configuration** (拡張リソース・グループ構成)」>「**Show All Resources by Node or Resource Group** (ノードまたはリソース・グループ別にリソースをすべて表示)」画面を介して「**Command Status** (コマンドの状態)」画面にアクセスします。

ノードが認識しているすべての物理ボリュームのリストを取得し、属するボリューム・グループを見つけるには、`lspv` コマンドを使用します。ボリューム・グループ名を引数として指定しない場合、`lspv` コマンドはシステムに既知の物理ボリュームをすべて表示します。その例を次に示します。

```
lspv
hdisk00000914312e971arootvg
hdisk100000132a78e213rootvg
hdisk200000902a78e21adatavg
hdisk300000321358e354datavg
```

カラム表示の 1 列目には、ディスクの論理名が表示されます。2 列目には、ディスクの物理ボリューム識別子がリストされます。3 列目には、属するボリューム・グループ (ある場合) がリストされます。

各クラスター・ノードで、AIX は同じ物理ボリュームに異なる名前 (hdisk 番号) を割り当てます。特定の物理ボリュームに対応する名前を調べるには、各ノードにリストされた物理ボリューム識別子を比較します。

物理ボリュームの論理デバイス名 (hdisk *x*) を **lspv** コマンドの引数として指定すると、アクティブ (オンに変更された) かどうかなどの物理ボリュームの情報が表示されます。その例を次に示します。

```
lspv hdisk2
PHYSICAL VOLUME:hdisk2 VOLUME GROUP:abalonevg
PV IDENTIFIER: 0000301919439ba5 VG IDENTIFIER: 00003019460f63c7
PV STATE:active VG STATE:active/complete
STALE PARTITIONS: 0 ALLOCATABLE:yes
PP SIZE: 4 megabyte(s)LOGICAL VOLUMES:2
TOTAL PPs: 203 (812 megabytes) VG DESCRIPTORS: 2
FREE PPs:192 (768 megabytes)
USED PPs:11 (44 megabytes)
FREE DISTRIBUTION: 41..30..40..40..41
USED DISTRIBUTION:00..11..00..00..00
```

物理ボリュームが非アクティブの場合 (**PV STATE** フィールドで疑問符で示されているように、オンに変更されていない場合)、構成用の適切なコマンドを使用して、物理ボリュームを含むボリューム・グループをオンに変更します。ただし、これを行う前に、システム・エラー・レポートを検査して、ディスクに関する問題が存在するかどうかを判別してください。システム・エラー・レポートを検査するには、以下のコマンドを入力します。

```
errpt -a|more
```

また、**lsdev** コマンドを使用して、システムに認識されたすべての物理ボリュームの可用性、つまり状態を検査することもできます。

## 論理ボリュームの検査

物理ボリュームに定義された論理ボリュームの状態を検査するには、**lspv -l** コマンドを使用して、検査するディスクの論理名を指定します。

以下の例に示すように、このコマンドを使用して物理ボリュームで定義されている論理ボリュームの名前を判別できます。

```
lspv -l hdisk2
LV NAMELPs PPs DISTRIBUTIONMOUNT POINT
lv02 50 50 25..00..00..00..25/usr
lv04 44 44 06..00..00..32..06/clusterfs
```

**lslv logicalvolume** コマンドを使用して、**LV STATE** フィールドで示されるように特定の論理ボリュームの状態 (オープンまたはクローズ) に関する情報を表示します。その例を次に示します。

```
lslv nodeA1v

LOGICAL VOLUME: nodeA1v VOLUME GROUP:nodeAvg
LV IDENTIFIER: 00003019460f63c7.1PERMISSION: read/write
VG STATE:active/complete LV STATE:opened/syncd
TYPE: jfs WRITE VERIFY:off
MAX LPs: 128 PP SIZE: 4 megabyte(s)
COPIES: 1 SCHED POLICY:parallel
LPs: 10PPs: 10
STALE PPs:0 BB POLICY:relocatable
INTER-POLICY:minimum RELOCATABLE: yes
INTRA-POLICY:middleUPPER BOUND: 32
MOUNT POINT: /nodeAfsLABEL:/nodeAfs
MIRROR WRITE CONSISTENCY: on
EACH LP COPY ON A SEPARATE PV ?: yes
```

論理ボリュームが非アクティブの場合 (つまり、「LV STATE」フィールドで示されているように、クローズしている場合)、構成用の適切なコマンドを使用して、論理ボリュームを含むボリューム・グループをオンに変更します。

## C-SPOC ユーティリティーによる共用論理ボリュームの確認

クラスター・ノード上の共用論理ボリュームの状態を検査するには、次の手順を実行します。

SMIT で「**System Management (C-SPOC) (システム管理 (C-SPOC))**」>「**PowerHA SystemMirror Logical Volume Management (PowerHA SystemMirror 論理ボリューム管理)**」>「**Shared Logical Volumes (共用論理ボリューム)**」>「**List All Shared Logical Volumes by Volume Group (ボリューム・グループ別に共用論理ボリュームをすべてリスト)**」を選択します。すべての共用論理ボリュームのリストが表示されます。

また、コマンド行から C-SPOC `cl_lslv` コマンドを使用して、この情報を表示できます。

## ファイル・システムの検査

必要なファイル・システムがマウントされているかどうか、およびマウントされている場所を確認します。この情報を PowerHA SystemMirror 定義と比較し、差があるかどうかを調べます。ファイル・システムのアクセス権およびファイル・システムで使用可能なスペースの量を検査します。

以下のコマンドを使用して、ファイル・システムに関するこの情報を取得します。

- `mount` コマンド。
- `df` コマンド。
- `lsfs` コマンド。

`cl_lsfs` コマンドを使用して、C-SPOC ユーティリティーの実行時にファイル・システムの情報をリストします。

ファイル・システムのリストの取得:

`mount` コマンドを使用して、システムに現在マウントされているすべてのファイル・システム (JFS および NFS の両方) とそのマウント・ポイントをリストします。

その例を次に示します。

```
mount
```

```
node mountedmounted over vfs date options
```

```
-----  
/dev/hd4 / jfs Oct 06 09:48 rw,log=/dev/hd8  
/dev/hd2 /usr jfs Oct 06 09:48 rw,log=/dev/hd8  
/dev/hd9var /var jfs Oct 06 09:48 rw,log=/dev/hd8  
/dev/hd3 /tmp jfs Oct 06 09:49 rw,log=/dev/hd8  
/dev/hd1 /home jfs Oct 06 09:50 rw,log=/dev/hd8  
pearl /home/home nfs Oct 07 09:59 rw,soft,bg,intr  
jade /usr/local /usr/localnfs Oct 07 09:59 rw,soft,bg,intr
```

ファイル・システムがマウントされているかどうか、およびファイル・システムがマウントされている場所を判別し、この情報を PowerHA SystemMirror 定義と比較し、差があるかどうかを調べます。

使用可能なファイルシステム・スペースの確認:

ファイル・システムで使用可能なスペースを確認するには、**df** コマンドを使用します。

その例を次に示します。

```
df
```

```
File System Total KB free %usediused %iused Mounted on
/dev/hd4 12288 530856% 896 21%/
/dev/hd2 4136962676893%19179 18%/usr
/dev/hd9var8192 373654% 115 5%/var
/dev/hd38192 7576 7%72 3%/tmp
/dev/hd14096 3932 4%17 1%/home
/dev/crab1lv8192 7904 3%17 0%/crab1fs
/dev/crab3lv 1228811744 4%16 0%/crab3fs
/dev/crab4lv 1638415156 7%17 0%/crab4fs
/dev/crablv4096 325220%17 1%/crabfs
```

使用可能なスペースの 90% 以上を超えて使用しているファイル・システムの **%used** 列を検査します。次に、**free** 列を検査して、残りの正確な空きスペースを判別します。

### マウント・ポイント、アクセス権、ファイルシステム情報の確認

**lsfs** コマンドを使用して、マウント・ポイント、アクセス権、ファイル・システムのサイズなどの情報を表示します。

その例を次に示します。

```
lsfs
```

```
Name Nodename Mount PtVFS Size Options Auto
/dev/hd4 --/jfs 24576 -- yes
/dev/hd1 --/homejfs 8192 -- yes
/dev/hd2 --/usrjfs 827392 -- yes
/dev/hd9var--/varjfs 16384-- yes
/dev/hd3 -- /tmp jfs 16384-- yes
/dev/hd7 --/mntjfs -- -- no
/dev/hd5 --/blvjfs -- -- no
/dev/crab1lv --/crab1fsjfs 16384rw no
/dev/crab3lv --/crab3fsjfs 24576rw no
/dev/crab4lv --/crab4fsjfs 32768rw no
/dev/crablv-- /crabfs jfs 8192 rw no
```

重要: ファイル・システムを NFS エクスポートする場合は、これらのファイル・システムの論理ボリューム名がクラスター全体で整合していることを確認します。

### C-SPOC ユーティリティーによる共用ファイルシステムの確認:

必要な共用ファイルシステムがマウントされているかどうか、およびその共用ファイルシステムがマウントされている 2 ノード C-SPOC 環境内のクラスター・ノード上の場所を確認します。

SMIT で「**System Management (C-SPOC) (システム管理 (C-SPOC))**」>「**PowerHA SystemMirror Logical Volume Management (PowerHA SystemMirror 論理ボリューム管理)**」>「**Shared Filesystems (共用ファイルシステム)**」を選択します。「**Journaled Filesystems (Journaled ファイルシステム)**」>「**List All Shared Filesystems (共用ファイルシステムをすべてリスト)**」または「**Enhanced Journaled Filesystems (拡張 Journaled ファイルシステム)**」>「**List All Shared Filesystems (共用ファイルシステムをすべてリスト)**」を選択して、共用ファイルシステムのリストを表示します。

また、コマンド行から C-SPOC **cl\_lsfs** コマンドを使用して、この情報を表示できます。

ファイルシステムの自動マウント属性の検査:

ブート時に、AIX では **fsck** コマンドを実行して **check=true** 属性により **/etc/filesystems** にリストされたすべてのファイル・システムを検査します。

AIX がファイル・システムを検査できない場合は、以下のエラーがレポートされます。

```
Filesystem helper: 0506-519 Device open failed
```

PowerHA SystemMirror で制御されるファイル・システムの場合、通常、このエラー・メッセージは問題を示していません。ファイル・システムの検査が失敗するのは、ファイル・システムが定義されているボリューム・グループがブート時にはオンに変更されていないためです。

このメッセージが生成されないようにするには、**/etc/filesystems** ファイルを編集して、共有ファイル・システムのスタンザに **check=true** 属性が含まれないようにします。

## TCP/IP サブシステムの検査

AIX コマンドで TCP/IP サブシステムを調査できます。

以下のようなコマンドがあります。

- **netstat** コマンドを使用して、必ずネットワーク・インターフェースが初期化され、通信パスがローカル・ノードとターゲット・ノード間に存在することを確認します。
- **ping** コマンドを使用して、ノード間の Point-to-Point 接続を確認します。
- すべてのネットワーク・インターフェースで **ifconfig** コマンドを使用して、無効な IP アドレス、不正なサブネット・マスク、および不適切なブロードキャスト・アドレスを検出します。
- **/var/hacmp/log/hacmp.out** ファイルを調べて、**/etc/rc.net** スクリプトが正常に実行されたことを確認します。ゼロの終了状況を探してください。
- IP アドレス・テークオーバーが有効な場合、**/etc/rc.net** スクリプトが実行され、サービス・インターフェースが基本 (ブート) アドレスではなく、サービス・アドレス上に存在していることを確認します。
- **lssrc -g tcpip** コマンドを使用して、**inetd** デーモンが実行されていることを確認します。
- **lssrc -g portmap** コマンドを使用して、**portmapper** デーモンが実行されていることを確認します。
- **arp** コマンドを使用して、クラスター・ノードが同一の IP アドレスまたはハードウェア・アドレスを使用していないことを確認します。
- **netstat** コマンドを使用して、以下を行います。
  - ノードに定義されたネットワーク・インターフェースの状況を表示します。
  - ローカル・ノードからターゲット・ノードの経路を定義するかどうかを判別します。

**netstat -in** コマンドはノードの初期化済みインターフェースすべてのリスト、およびそのインターフェースの接続先のネットワークと IP アドレスを表示します。このコマンドを使用すると、サービス・インターフェースとブート・インターフェースが別々のサブネットに存在するかどうかを判別できます。サブネットは、「**Network** (ネットワーク)」列に表示されます。

```
netstat -in
```

```
Name Mtu NetworkAddress IpktsIerrs OpktsOerrsColl
lo0 1536 <Link> 18406 0 18406 00
lo0 1536 127.0.0.118406 0 18406 00
en1 1500 <Link> 11116260 58643 00
en1 1500 100.100.86.100.100.86.136 11116260 58643 00
```

```
en0 1500 <Link> 943656 0 52208 00
en0 1500 100.100.83.100.100.83.136 943656 0 52208 00
tr1 1492 <Link> 18790 165600
tr1 1492 100.100.84.100.100.84.136 18790 165600
```

出力の 1 列目、3 列目、4 列目を確認します。「**Name (名前)**」列は、このノードで定義され使用できるすべてのインターフェースをリストします。名前の前のアスタリスクは、インターフェースがダウンしている (使用できない) ことを示します。「**Network (ネットワーク)**」列は、インターフェースの接続先 (そのサブネット) を示します。「**Address (アドレス)**」列は、ノードに割り当てられている IP アドレスを識別します。

**netstat -rn** コマンドは、ターゲット・ノードの経路が定義されているかどうかを示します。定義済みの経路をすべて表示するには、以下のように入力します。

```
netstat -rn
```

以下の例のような情報が表示されます。

```
Routing tables
DestinationGateway  Flags  Refcnt  UseInterface
Netmasks:
(root node)
(0)0
(0)0 ff00 0
(0)0 ffff 0
(0)0 ffff ff80 0
(0)0 70 204 1 0
(root node)Route Tree for Protocol Family 2:
(root node)
127 127.0.0.1U 3 1436 lo0
127.0.0.1 127.0.0.1UH0456 lo0
100.100.83.128100.100.83.136 U 6 18243 en0
100.100.84.128100.100.84.136 U 1 1718 tr1
100.100.85.128100.100.85.136 U 2 1721 tr0
100.100.86.128100.100.86.136 U 8 21648 en1
100.100.100.128 100.100.100.136 U 039 en0
(root node)Route Tree for Protocol Family 6:
(root node)
(root node)
```

ネットワークへの特定の経路 (例えば、100.100.83) をテストするには、以下のように入力します。

```
netstat -nr | grep '100¥.100¥.83'
```

```
100.100.83.128100.100.83.136 U 6 18243 en0
```

同じテストを、経路指定テーブルにこの経路が存在しないシステムで実行しても、応答は戻されません。サービス・インターフェースとブート・インターフェースがブリッジ、ルーター、またはハブによって分離され、ネットワーク・デバイスとの通信に問題が発生する場合は、デバイスが 2 つのネットワーク・セグメントを 1 つの物理ネットワークとして処理するように設定されていない可能性があります。構成から独立したデバイスをテストするか、システム管理者に支援を求めてください。

ネットワークにアクティブなインターフェースが 1 つしかない場合は、クラスター・マネージャーはそのインターフェースの障害イベントを生成しません。

このコマンドの使用について詳しくは、**netstat** マニュアル・ページを参照してください。

関連情報:

ネットワーク・インターフェース・イベント

## Point-to-Point 接続の確認

**ping** コマンドは、クラスター内における 2 つのノード間の point-to-point 接続をテストします。 **ping** コマンドを使用して、ターゲット・ノードがネットワークに接続されているかどうか、ノード間のネットワーク接続が信頼できるかどうかを判別します。

ノード (サービスおよびブート) 上で構成されているすべての TCP/IP インターフェースを必ずテストしてください。

例えば、ローカル・ノードから *nodeA* という名前のリモート・ノードへの接続をテストするには、以下のように入力します。

```
/etc/ping nodeA
```

```
PING testcluster.nodeA.com: (100.100.81.141): 56 data bytes
64 bytes from 100.100.81.141: icmp_seq=0 ttl=255 time=2 ms
64 bytes from 100.100.81.141: icmp_seq=1 ttl=255 time=1 ms
64 bytes from 100.100.81.141: icmp_seq=2 ttl=255 time=2 ms
64 bytes from 100.100.81.141: icmp_seq=3 ttl=255 time=2 ms
```

Control-C と入力して、パケットの表示を終了します。以下の統計情報が表示されます。

```
----testcluster.nodeA.com PING Statistics----
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1/1/2 ms
```

**ping** コマンドはパケットを特定のノードに送信して、応答を要求します。正しい応答が返されると、**ping** はパケットが失われていないことを示す前述の出力と同様のメッセージを出力します。これにより、ノード間の接続が有効であることが示されます。

**ping** コマンドがハングする場合は、**ping** コマンドを発行しているノードと到達しようとしているノード間のパスが無効であることを示します。必要な TCP/IP デモンが実行されていないことを示している場合もあります。2 つのノードの間の物理接続を検査してください。構成を確認するには、**ifconfig** コマンドおよび **netstat** コマンドを使用します。「bad value (値が無効)」であることを示すメッセージは、IP アドレスまたはサブネット定義に関する問題を示しています。

「DUP!」が **ping** 応答の最後に表示された場合は、**ping** コマンドが同一アドレスで複数の応答を受信したことを示します。通常、この応答が発生するのは、ネットワーク・インターフェースの構成が誤っているか、IP アドレス・テークオーバー中にクラスター・イベントが失敗したケースです。サブネット上のすべてのインターフェースの構成を検査し、1 つのアドレスに 1 つのインターフェースしか存在しないことを検証してください。詳しくは、**ping** マニュアル・ページを参照してください。

さらに、永続ノード IP ラベルをノードのクラスター・ネットワークに割り当てることができます。使用しているサービス IP ラベルがそのノードにあるいずれのリソース・グループに属しているかどうかを気にせず、**ping** または **telnet** コマンドを使用して、管理目的でクラスターの特定のノードに到達する場合は、そのノードに定義された永続ノード IP ラベルを使用すると便利です。

関連情報:

PowerHA SystemMirror プランニング・ガイド

PowerHA SystemMirror クラスター・トポロジーとリソースの構成 (拡張)

## IP アドレスとネットマスクの検査

**ifconfig** コマンドを使用して、IP アドレスおよびネットマスクが正しいことを確認します。調査対象のネットワーク・インターフェースの名前を指定して **ifconfig** を呼び出します。

例えば、最初のイーサネット・インターフェースを検査するには、以下のように入力します。

```
ifconfig en0

en0: flags=2000063<UP,BROADCAST,NOTRAILERS,RUNNING,NOECHO>
    inet 100.100.83.136 netmask 0xfffff00 broadcast 100.100.83.255
    inet6 fe80::214:5eff:fe4d:6045/64
    tcp_sendspace 131072 tcp_recvspace 65536 rfc1323 0
```

指定のインターフェースが存在しない場合、**ifconfig** が次のように応答します。

```
No such device
```

**ifconfig** コマンドは複数の行の出力を表示します。1 行目には、インターフェースの名前および特性が表示されます。以下の特性を検査してください。

表 15. **ifconfig** コマンドの出力

フィールド	値
UP	インターフェースが使用可能です。インターフェースが停止している場合は、 <b>ifconfig</b> コマンドを使用してそのインターフェースを初期化してください。次に例を示します。  <code>ifconfig en0 up</code>  インターフェースが起動しない場合は、インターフェース・ケーブルを取り替えて再試行してください。それでも起動しない場合は、 <b>diag</b> コマンドを使用してデバイスを検査してください。
RUNNING	インターフェースが稼働しています。インターフェースが実行されていない場合は、このインターフェースのドライバーが正常にインストールされていないか、インターフェースが正常に構成されていない可能性があります。このインターフェースのインストールに必要なすべてのステップを確認し、誤りや飛ばしたステップがないかどうかを確認してください。 注: <b>ifconfig</b> コマンドはアダプターの構成のみを表示し、機能状態は表示しません。構成された状態が「稼働中」であっても、それ以外の何らかの問題で通信できないことがあります。

**ifconfig** コマンドの残りの出力には、インターフェースで構成された各アドレスに関する情報が含まれます。これらのフィールドを検査し、ネットワーク・インターフェースが正常に構成されていることを確認します。

詳細については、**ifconfig** マニュアル・ページを参照してください。

## arp コマンドを使用する方法

**arp** コマンドを使用して、ホストの **arp** キャッシュにリストされているノードに関連付けられた現在の IP アドレスを表示します。その例を次に示します。

```
arp -a

flounder (100.50.81.133) at 8:0:4c:0:12:34 [ethernet]
cod (100.50.81.195) at 8:0:5a:7a:2c:85 [ethernet]
pollock (100.50.81.147) at 10:0:5a:5c:36:b9 [ethernet]
```

この出力は、ホスト・ノードが現在認識している **flounder** ノード、**cod** ノード、**seahorse** ノード、および **pollock** ノードの IP アドレスと MAC アドレスを示しています。(ハードウェア・アドレス・テークオーバーなしで IP アドレス・テークオーバーが実行されると、ホストの **arp** キャッシュ内の IP アドレスに関連付けられている MAC アドレスが古くなってしまふことがあります。これを訂正するには、ホストの **arp** キャッシュを更新します。)

詳細については、**arp** マニュアル・ページを参照してください。

## AIX オペレーティング・システムの検査

クラスターに影響を及ぼす可能性のあるハードウェアやソフトウェアのエラーを表示する場合、**errpt** コマンドを使用してください。

実際の障害を示すディスクおよびネットワークに関するエラー・メッセージ (特に永久的なエラー) を探します。詳細については、**errpt** マニュアル・ページを参照してください。

## 物理ネットワークの検査

物理ネットワークおよび接続を確認します。

物理接続を調査する際のチェックポイントを以下に示します。

- ノードの各ペアの間のシリアル回線を検査します。
- イーサネットを使用している場合は、以下のように実行します。
  - **diag** コマンドを使用して、ネットワーク・インターフェース・カードおよびケーブルが適正であることを確認します。
  - IBM System p 用のイーサネット・アダプターは、カード上のトランシーバーと外部トランシーバーのいずれとも使用できます。使用している側を指定するジャンパーが NIC にあります。ジャンパーが正常に設定されているかどうかを検証します。
  - 接続されているケーブルのハブのライトがオンになっていることを確認します。

関連情報:

クラスター・ネットワーク接続の計画

## ディスクおよびディスク・アダプターの検査

**diag** コマンドを使用して、アダプター・カードが正常に機能していることを確認します。問題が発生している場合は、SCSI バスに沿ってジャンパー、ケーブル、およびターミネーターを検査します。

IBM SCSI ディスクやアレイなどの SCSI ディスクの場合は、SCSI バス上の各アレイ・コントローラー、アダプター、および物理ディスクに一意的 SCSI ID が付いていることを確認してください。一部の SCSI アダプターには、設定できる SCSI ID に制限がありますが、バスの各 SCSI ID は、0 から 15 までの整数値でなければなりません。デバイス固有の制限については、デバイスに関する資料を参照してください。一般的な構成では、ノード上のアダプターの SCSI ID には、共用デバイスの SCSI ID より大きい値を設定します。値の大きい ID を持つデバイスは、SCSI バスの競合で優先されます。

例えば、標準の SCSI アダプターが 5 および 6 の ID を使用する場合は、0 から 4 までの値をバス上の他のデバイスに割り当てます。アダプターの SCSI ID を 5 および 6 に設定すると、デフォルトでは、7 の ID が常に使用されるために、他のブート・デバイスの **mksysb** テープからサービス・モードでいずれかのシステムをブートする際に、考えられる競合を防止できます。

SCSI アダプターが 14 および 15 の ID を使用する場合は、バス上の他のデバイスに 3 から 13 の値を割り当てます。

**lsattr** または **lsdev** コマンドを使用して、アダプターとディスクの SCSI ID を検査できます。例えば、アダプター *scsi1* (SCSI-3) の SCSI ID を判別するには、以下の **lsattr** コマンドを使用してアダプターの論理名を引数として指定します。

```
lsattr -E -l scsi1 | grep id
```

デバイス名を指定するコマンド行では、ワイルドカード文字または絶対パス名は使用しないでください。

重要: クラスタ構成のバックアップを既存のシステムに復元する場合、SCSI ID を再検査するかリセットして、共用バス上の考えられる SCSI ID 競合を防ぎます。システムのバックアップを復元すると、アダプター SCSI ID はデフォルトの 7 の SCSI ID にリセットされます。

SCSI ID 競合に関連するディスクとディスク・アダプターへの SCSI ID の設定については、「プランニング・ガイド」を参照してください。

ディスクの SCSI ID を判別するには、以下のように入力します。

```
lsdev -Cc disk -H
```

関連情報:

PowerHA SystemMirror プランニング・ガイド

## PCI ホット・プラグ NIC 障害からの回復

回復不能エラーが原因で PCI ホット・リプレース・プロセスが失敗した場合、NIC が未構成または保守モードのままの状態である場合があります。新しいカードやそのカードが挿入されている PCI スロットがこの時点で損傷している可能性があります。このノードを完全に稼働した状態に回復するには、ユーザーの介入が必要です。

詳細については、ハードウェア・マニュアルを参照するかまたは IBM の Web サイトのデバイス情報を検索してください。

## クラスタ通信デーモンの検査

クラスタが同期化された後で AIX アダプター構成の IP アドレスを変更または除去すると、クラスタ通信デーモンが `/etc/cluster/rhosts` ファイルまたは PowerHA SystemMirror 構成データベース・エントリーに対してそれらのアドレスを検証できない場合があります。この問題が発生すると、構成の処理中または検証と同期化の実行中に PowerHA SystemMirror から 1 つ以上のエラーが出されることがあります。

あるいは、クラスタの同期化中にエラーを受け取る場合があります。

その場合、`/etc/cluster/rhosts` ファイルに保管されているすべてのクラスタ・ノードに関する情報を更新し、`clcomd` コマンドを更新して、変更を認識させる必要があります。クラスタの同期化と検証を再び行うと、`clcomd` コマンドは PowerHA SystemMirror 構成データベースに追加された IP アドレスの使用を開始します。

クラスタ通信デーモンを更新するには、次のコマンドを使用します。

```
refresh -s clcomd
```

さらに、現在 PowerHA SystemMirror がノード間通信に使用しているすべてのアドレスが含まれるように `/etc/cluster/rhosts` ファイルを構成してから、このファイルをすべてのクラスタ・ノードにコピーします。`/etc/cluster/rhosts` ファイルには IPv4 アドレスと IPv6 アドレスを含めることができます。

関連資料:

78 ページの『クラスタ通信の問題』

以下のトピックでは、潜在的なクラスタ通信の問題について説明します。

## システム・ハードウェアの検査

電源装置および LED 表示を検査し、エラー・コードが表示されているかどうかを確認します。AIX の `diag` コマンドを実行して、システム装置をテストします。

**diag** は、引数がない場合、メニュー方式プログラムとして実行されます。 **diag** は、特定ハードウェア部分に対しても実行できます。その例を次に示します。

```
diag -d hdisk0 -c
```

```
Starting diagnostics.  
Ending diagnostics.
```

この出力は、hdisk0 が正常であることを示します。

## PowerHA SystemMirror のインストールの問題

以下のトピックでは、潜在的なインストールの問題についていくつか説明します。

### ブート時にファイルシステムが見つからない

このトピックでは、AIX がブート時にファイルシステムを検出できないときの問題と解決策について説明します。

#### 問題

ブート時に、AIX では **fsck** コマンドを実行して **check=true** 属性により **/etc/filesystems** にリストされたすべてのファイル・システムを検査します。ファイル・システムを検査できない場合、AIX からエラーが報告されます。システムから次の出力が表示されます。

```
+-----+  
Filesystem Helper: 0506-519 Device open failed  
+-----+
```

#### 解決方法

PowerHA SystemMirror で制御されるファイル・システムの場合、通常、このエラーは問題を示していません。ファイル・システムの検査が失敗するのは、ファイル・システムが定義されているボリューム・グループがブート時にはオンに変更されていないためです。このメッセージが生成されないようにするには、**/etc/filesystems** ファイルを編集して、共用ファイル・システムのスタンザに **check=true** 属性が含まれないようにします。

### インストールが失敗したために **cl\_convert** が実行されない

このトピックでは、インストールが失敗したために **cl\_convert** が実行されないときの問題と解決策について説明します。

#### 問題

PowerHA SystemMirror をインストールすると、**cl\_convert** が自動的に実行されます。ソフトウェアは、既存の PowerHA SystemMirror 構成があるかどうかを検査し、その構成を、インストール中のソフトウェアのバージョンで使用する形式に変換しようとします。ただし、インストールが失敗したら、**cl\_convert** は結果として実行されません。したがって、以前の PowerHA SystemMirror バージョンの構成データベースから現行バージョンの構成データベースへの変換も失敗します。

#### 解決方法

コマンド行から **cl\_convert** を実行します。変換が成功したかどうかを判別するには、変換の進捗状況をログ記録した **clconvert.log** ファイルを参照してください。

**cl\_convert** を実行するには、ルート・ユーザーのアクセス権が必要です。

注意:

変換を行う前に、**ODMDIR** 環境変数が **/etc/es/objrepos** に設定されていることを確認してください。

**cl\_convert** フラグについて詳しくは、**cl\_convert** マニュアル・ページを参照してください。

## インストール時に構成ファイルをマージできなかった

このトピックでは、インストール時の構成ファイルの問題について説明します。

### 問題

PowerHA SystemMirror クライアント・ソフトウェアのインストール中に、以下のメッセージが表示されます。

```
+-----+
Post-installation Processing...
+-----+
Some configuration files could not be automatically merged into
the system during the installation. The previous versions of these files
have been saved in a configuration directory as listed below. Compare
the saved files and the newly installed files to determine if you need
to recover configuration data. Consult product documentation
to determine how to merge the data.
Configuration files, which were saved in /usr/lpp/save.config:
/usr/es/sbin/cluster/utilities/clexit.rc
```

### 解決方法

PowerHA SystemMirror インストール・プロセスの一部として、サイト固有の変更が含まれる可能性がある PowerHA SystemMirror ファイルのコピーは、上書きされる前に **/usr/lpp/save.config** ディレクトリに保管されます。メッセージに示されるように、ユーザーがサイト固有の構成情報を新しくインストールされたファイルにマージする必要があります。

## PowerHA SystemMirror および Tivoli System Automation for Multiplatform のトラブルシューティング

Tivoli® System Automation for Multiplatform を、PowerHA SystemMirror バージョン 7.1.0 以降が既にインストールされている同じノードにインストールすることはできません。

PowerHA SystemMirror 7.1.0 以降は、Cluster Aware AIX (CAA) 機能を基礎としています。Tivoli System Automation for Multiplatform は、高信頼性スケーラブル・クラスター・テクノロジー (RSCT) (Reliable Scalable Cluster Technology (RSCT)) 機能を基礎としています。つまり、PowerHA SystemMirror と Tivoli System Automation for Multiplatform は異なるクラスタリング機能を基礎としているため、これらを同じノード上で使用することはできません。

---

## 共通問題の解決

このセクションでは、よくある問題と、推奨事項について説明します。

### PowerHA SystemMirror の始動の問題

以下のトピックでは、潜在的な PowerHA SystemMirror 始動の問題について説明します。

#### ODMPATH 環境変数が正しく設定されない

このトピックでは、照会したオブジェクトが見つからない場合の考えられる原因について説明します。

## 問題

照会したオブジェクトが見つかりません。

## 解決方法

PowerHA SystemMirror は、構成データを格納する特定の ODM リポジトリのロケーションに依存します。ODMPATH 環境変数により、ODM コマンドおよびサブルーチンは、照会したオブジェクトがデフォルトのロケーションに存在しない場合でもデフォルト以外のロケーションを照会できます。この変数は設定できますが、デフォルトのロケーション、`/etc/objrepos` を含める必要があります。そうしないと、構成情報の整合性が失われる場合があります。

## clinfo デーモンが始動後に終了する

このトピックでは、**clinfoES** デーモンを **-a** オプションとともに開始した後に発生する「smux-connect」エラーについて説明します。

## 問題

**clinfoES** デーモンを **-a** オプションとともに開始した後に「smux-connect」エラーが発生します。別のプロセスがポート 162 を使用してトラップを受信しています。

## 解決方法

NetView® for AIX の **trapgend smux** サブエージェントまたは AIX **sysmond** デーモンのシステム・モニターなどの別のプロセスがポート 162 を使用しているかどうかを検査します。使用している場合は、**-a** オプションなしに **clinfoES** を再始動し、NetView for AIX を構成して、SNMP トラップを受信します。**clinfoES** が通常の方法で **startsrc** コマンドを使用して開始されている場合、このエラーは発生しません。

## ノードの電源が遮断されている、またはクラスター・マネージャーが開始されない

このトピックでは、ノードの電源自体が遮断されているか、またはクラスター・マネージャーの開始後にハングしたようになっているときの問題と解決策について説明します。

## 問題

クラスター・サービスの開始後にノードの電源が自動的に遮断されるかハング状態になります。errpt レポートには、システムに **halt -q** コマンドを発行する **clexit.rc** スクリプトによってログに記録されたオペレーター・メッセージが示されます。

## 解決方法

クラスターの検証ユーティリティを使用して、すべてのクラスター・ノード上のクラスター構成情報の非整合を明らかにします。

クラスターの検証ユーティリティによって明らかにされた構成エラーを修正します。PowerHA SystemMirror 構成 SMIT パネルを使用して、必要な変更を加えます。問題を修正した後、「**Verify and Synchronize PowerHA SystemMirror Configuration (構成の検証および同期化)**」オプションを使用して、すべてのノード間でクラスター構成を同期化します。次に、SMIT パネルで「**System Management (C-SPOC) (システム管理 (C-SPOC))**」>「**Manage PowerHA SystemMirror Services (サービスの管理)**」から「**Start Cluster Services (クラスター・サービスの始動)**」オプションを選択して、クラスター・マネージャーを始動します。

この問題は、構成が検証に合格しクラスター全体で同期化された後は起こらないはずです。問題が解消されない場合は、ソフトウェア障害報告手続きに従って、IBM サポートに問題を報告してください。

**snap -e** コマンドの詳細については、『AIX データ収集ユーティリティーの使用』セクションを参照してください。

関連資料:

3 ページの『AIX データ収集ユーティリティーの使用』

AIX **snap** コマンドを使用して、データを PowerHA SystemMirror クラスターから収集します。

関連情報:

クラスター・マネージャー・デーモンの異常終了

## configchk コマンドが不明ホスト・メッセージを返す

このトピックでは、**configchk** コマンドが不明ホスト・メッセージを返す特定の状態について説明します。

### 問題

各クラスター・ノードの **/etc/hosts** ファイルには、クラスター内の別のノードの IP ラベルは含まれていません。例えば、4 ノード・クラスター、ノード A、ノード B、およびノード C の **/etc/hosts** ファイルには、別のクラスター・ノードの IP ラベルは含まれていません。

このような状況の場合、**configchk** コマンドは以下のメッセージをコンソールに戻します。

```
"your hostname not known," "Cannot access node x."
```

このメッセージは、ノード x 上の **/etc/hosts** ファイルにあなたのノードのエントリーが含まれていないことを示しています。

### 解決方法

PowerHA SystemMirror ソフトウェアを始動する前に、各ノードの **/etc/hosts** ファイルに各クラスター・ノードのサービスおよびブート IP ラベルが含まれていることを確認します。

## 再構成時にクラスター・マネージャーがハングする

このトピックでは、再構成時にクラスター・マネージャーがハングする状態について説明します。

### 問題

再構成中にクラスター・マネージャーがハングし、以下のようなメッセージが生成されます。

```
The cluster has been in reconfiguration too long;Something may be wrong.
```

すべてのイベント・スクリプトが失敗しました。

### 解決方法

**/var/hacmp/log/hacmp.out** ファイルを調べてゼロ以外の状況で終了したプロセスを確認し、スクリプトが失敗した原因を判別します。 **/var/hacmp/adm/cluster.log** ファイルのエラー・メッセージも有用です。 ログ・ファイルで示された問題を修正します。 コマンド行または、SMIT 「**Problem Determination Tools** (問題判別ツール)」 > 「**Recover From PowerHA SystemMirror Script Failure** (スクリプト障害からの回復)」 パネルを使用して、**clrunccmd** コマンドを実行します。 **clrunccmd** コマンドは、クラスター・マネージャーにシグナルを送信してクラスター処理を再開させます。

新たにインストールした **AIX** ノード上で **clcomd** および **clstrmgrES** が起動に失敗  
この問題では、新たにインストールした **AIX** ノード上で **clcomd** および **clstrmgrES** が起動に失敗する  
状態について説明します。

## 問題

新たにインストールされた **AIX** ノード上で、**clcomd** および **clstrmgrES** が開始されません。

## 解決方法

**AIX** のインストールが完了したことを (**AIX** インストール・アシストの) システム・コンソールに手動で  
指示します。

通常新しくインストールされた **AIX** ノードでこの問題は発生します。すなわち、最初のブートで **AIX** は  
**/etc/inittab** からインストール・アシストを実行し、このファイルの他のエントリーの実行を続けません。  
**AIX** インストール・アシストはシステム・コンソールへの入力を待っています。インストールが完了した  
ことを指示するまで、以降の起動時に毎回 **AIX** はインストール・アシストを実行します。完了を指示する  
と、システムはクラスター通信デーモン (**clcomd**) とクラスター・マネジャー・デーモン (**clstrmgr**) の始  
動に移ります。

## アップグレード後にノードにイベント前処理または後処理が存在しない

このトピックでは、アップグレード後にノードにイベント前処理や後処理がない問題について説明します。

## 問題

クラスターの検証ユーティリティーは、新規の **PowerHA SystemMirror** ソフトウェアのバージョンにア  
ップグレードされた後、イベント前処理またはイベント処理後がノードに存在しないことを示します。

## 解決方法

定義済みの名前前のスクリプトが存在し、すべてのクラスター・ノードで実行可能であることを確認します。

各ノードは、定義済みのイベント前処理または後処理に関連したスクリプトを含む必要があります。各ノ  
ードでスクリプトの内容が同じである必要はありませんが、スクリプトの名前はクラスター全体にわたって  
整合している必要があります。特定のノードでアクションが必要でない場合は、同じイベント・スクリプ  
ト名を持つ **no-op** スクリプトが、処理が行われないノードに存在する必要があります。

## 構成時にノード障害が発生し、**869 LED** が表示される

このトピックでは、システムがハングして **869** が表示される状態について説明します。

## 問題

システムがハングしているように見えます。システム **LED** に **869** が継続的に表示されます。

## 解決方法

この表示は、さまざまな状態で発生します。SCSI バスに接続された各デバイスの SCSI ID が固有であ  
り、SCSI ID が競合しないことを確認します。特に、SCSI バスに接続されている各クラスター・ノード  
のアダプターおよびデバイスに、異なる SCSI ID が付けられているかどうかを検査します。デフォルトで  
は、アダプターの構成時に **AIX** は 7 の ID を SCSI アダプターに割り当てます。SCSI ID の確認およ  
び設定について詳しくは、「プランニング・ガイド」を参照してください。

関連情報:

## ノードを動的に除去した後にクラスターに再結合できない

このトピックでは、クラスターから動的に除去されて再結合できなくなったノードについて説明します。

### 解決方法

クラスターからノードを除去する場合、クラスター定義はノードの構成データベースにそのまま存在します。除去したノード上でクラスター・サービスを始動すると、ノードは、このクラスター構成データを読み取り、除去元のクラスターと再結合しようとしています。他のノードはこのノードをクラスターのメンバーとして認識しないため、このノードの結合を拒否します。クラスターへの結合を要求しているノードのクラスター名が既存のクラスターと同じであるため、このクラスターが不安定になったり、既存のノードをクラッシュさせたりする場合があります。

古くなった構成データベース情報で除去したノードが再始動されないようにするには、次の手順を実行してノードからクラスター定義を除去します。

1. 以下のコマンドを使用して、除去されるノード上のクラスター・サービスを停止します。

```
clstop -R
```

重要: ノードのクラスター・サービスを停止してから、ノードをクラスターから除去してください。

**-R** フラグは、`/etc/inittab` ファイルの `PowerHA SystemMirror` エントリーを削除し、ノードのリブート時にクラスター・サービスが自動的に始動するのを防ぎます。

2. 以下のコマンドを使用して、`rc.net` ファイルから `PowerHA SystemMirror` エントリーを削除します。

```
clchipat false
```

3. 以下のコマンドを使用して、ノードの構成データベースからクラスター定義を削除します。

```
clrmclstr
```

このタスクは、SMIT パネルから「**Extended Configuration (拡張構成)**」>「**Extended Topology Configuration (拡張トポロジー構成)**」>「**Configure a PowerHA SystemMirror Cluster (クラスターの構成)**」>「**Remove a PowerHA SystemMirror Cluster (クラスターの除去)**」を選択しても実行できません。

## リソース・グループの移行がクラスターの始動後に永続的でなくなる

このトピックでは、リソース・グループの移行がクラスターの始動後に永続的でなくなる状態について説明します。

### 問題

リソース・グループ移行ユーティリティーを使用して、リソース・グループ移行の操作を指定しました。ここで、フラグを `true` に設定して (または `clRGmove` コマンドを発行して)、この特定の移行「**Persists across Cluster Reboot (クラスターのリブート後も持続)**」を要求しました。しかし、クラスター・サービスを停止して再始動した後、このポリシーがクラスター内のいずれかのノードで適用されません。

### 解決方法

この問題が発生するのは、永続リソース・グループ移行を指定したとき、ノードがダウンしてアクセス不能であった場合です。この場合、ノードは永続リソース・グループ移行に関する情報は取得していません。クラスター・サービスの再始動後、このノードが最初にクラスターに結合する場合は、「**Persist across Cluster Reboot (クラスターのリブート後も持続)**」の設定を認識しません。このため、リソース・グルー

プ移行が永続的でなくなります。永続的な移行の設定を復元するには、「**Extended Resource Configuration (拡張リソース構成)**」>「**PowerHA SystemMirror Resource Group Configuration (リソース・グループ構成)**」SMIT メニューで、SMIT に再指定する必要があります。

## PowerHA SystemMirror へのアップグレード後にクラスターが始動しない

このトピックでは、PowerHA SystemMirror へのアップグレード後にクラスターが始動しない状態について説明します。

### 問題

グループ「**hacmp**」の ODM エントリが SP ノードで除去されています。この問題は、クラスターが起動しない場合や、**clcomd** エラーの場合に発生します。

### 解決方法

PowerHA SystemMirror 構成データベース (ODM) を以下のように強化して、セキュリティーをさらに向上させてください。

- 所有権。ユーザー **root** およびグループ **hacmp** が、すべての PowerHA SystemMirror ODM ファイルを所有しています。さらに、**root** 以外のユーザーが使用するすべての PowerHA SystemMirror バイナリー・ファイルは、ユーザー **root** およびグループ **hacmp** も所有しています。
- アクセス権。600 のアクセス権を持つ **hacmpdiskssystem** ファイルを除くすべての PowerHA SystemMirror ODM ファイルには、640 のアクセス権 (ユーザー **root** およびグループ **hacmp** が読み取り可能で、ユーザー **root** が書き込み可能) が設定されています。**root** 以外のユーザーが使用するすべての PowerHA SystemMirror バイナリー・ファイルには、2555 のアクセス権が設定されます (すべてのユーザーが読み取りおよび実行可能で、**setgid** ビットがオンになりプログラムがグループ **hacmp** として実行される)。

インストール時に PowerHA SystemMirror は、グループ「**hacmp**」がまだ存在していない場合、すべてのノードに作成します。デフォルトで、グループ **hacmp** には PowerHA SystemMirror ODM を読み取るアクセス権があるが、その他には特別な権限はありません。セキュリティー上の理由から、グループ **hacmp** の権限を拡張しないことをお勧めします。

PowerHA SystemMirror ODM に直接アクセスするプログラムを使用する場合、**root** 以外のユーザーが実行する際にはそのプログラムを再度書き込む必要があります。

- **root** 以外のユーザーによる ODM データへのすべてのアクセスは、所定の PowerHA SystemMirror ユーティリティーを介して処理される必要があります。
- さらに、PSSP ファイル・コレクション機能を使用して、**/etc/group** の整合性を保持する場合、個々のクラスター・ノードでのインストール時に作成される新規のグループ「**hacmp**」は、次のファイルが同期化されると失われる場合があります。

### さまざまなファイルセット・レベルがノードにある場合の検証問題

クラスターにさまざまなファイルセット・レベル (**cluster.es.server.diag** など) のノードがあると、**clverify** プログラムがハングしたり、コアをダンプしたりすることがあります。

一般に、PowerHA SystemMirror ノードには同じファイルセット・レベルがありますが、ノード単位のローリング PTF アップグレードの実行中には、このような状況になる傾向があります。このようなタイプのエラーは、正常なクラスターの始動を妨げます。

この状況でクラスターを始動するときは、検証エラーを無視してください。 そのためには、**smit sysmirror**>「**System Management (C-SPOC)** (システム管理 (C-SPOC))」> **Manage PowerHA SystemMirror Services** (サービスの管理)」>「**Start Cluster Services** (クラスター・サービスの始動)」という SMIT パスを入力 (選択) します。

このパネルで、「Ignore verification errors? (検証エラーを無視する) (デフォルトは「false (いいえ)」) を「true (はい)」に変更します。

その後、クラスターを起動して、問題の **clverify** プログラムを回避できます。

注: この手順を実行しなくてもすむようにするには、できるだけ早めに、ノードが同じファイルセット・レベルにあることを確認してください。 検証エラーを無視しなくてもすむはずですが。

## ディスクとファイルシステムの問題

以下のトピックでは、潜在的なディスクとファイルシステムの問題について説明します。

### AIX ボリューム・グループ・コマンドが原因でシステム・エラー・レポートが生成される

このトピックでは、AIX ボリューム・グループ・コマンドが原因で生成されるシステム・エラー・レポートについて説明します。

#### 問題

**redefinevg**、**varyonvg**、**lqueryvg**、および **syncvg** コマンドは失敗し、システム再始動時に共有ボリューム・グループのエラーが報告されます。 これらのコマンドは、共有ボリューム・グループを自動的にオンに変更するときにコンソールにメッセージを送信します。 共有ディスク用にボリューム・グループを構成するとき、「**autovaryon at boot**」が無効にされませんでした。 起動しているノードが共有ドライブを所有し、他のノードが共有ボリューム・グループをオンに変更しようとする、さまざまな **varyon** エラー・メッセージが表示されます。

#### 解決方法

共有ボリューム・グループを設定する場合、「システム再起動時にボリューム・グループを自動的に活動化する」フィールドを SMIT 画面の「システム管理 (C-SPOC)」>「**PowerHA SystemMirror 論理ボリューム管理**」>「共有ボリューム・グループ」>「共有ボリューム・グループの作成」画面で「いいえ」に設定します。別のクラスター・ノードの共有ボリューム・グループをインポートした後、以下のコマンドを使用して、各ノードのボリューム・グループが「**autovaryon at boot**」に設定されていないことを確認します。

```
chvg -an vgname
```

### **varyonvg** コマンドがボリューム・グループで失敗する

このトピックでは、ボリューム・グループで失敗する **varyonvg** コマンドによって示されるさまざまな問題について説明します。

#### 問題

PowerHA SystemMirror ソフトウェア (**/var/hacmp/log/hacmp.out** ファイル) は、**varyonvg** コマンドがボリューム・グループをオンに変更しようとして失敗したことを示します。

## 解決方法

ボリューム・グループが、いずれのノード上でも「**autovaryon**」に設定されていないくて、ボリューム・グループ (コンカレント・アクセス・モードでない場合) が別のノードによりオンにまだ変更されていないことを確認します。

**lsvg -o** コマンドを使用して、共有ボリューム・グループがアクティブであるかどうかを判別します。 ボリューム・グループが活動化されているノードで `lsvg volume_group_name` と入力し、「**AUTO ON** (自動オン)」フィールドを調べて、ボリューム・グループが自動的にオンに設定されているかどうかを判別します。「**AUTO ON (自動 ON)**」が「**yes (はい)**」に設定されている場合は、`chvg -an volume_group_name` と入力して、これを修正します。

## 問題 2

ディスク上のボリューム・グループ情報がデバイス構成データベースの情報と異なっています。

## 解決方法 2

以下のように、情報が誤っているノードのデバイス構成データベースを訂正します。

1. **smit exportvg** 高速パスを使用して、ボリューム・グループ情報をエクスポートします。このステップにより、デバイス構成データベースからボリューム・グループ情報が除去されます。
2. **smit importvg** 高速パスを使用して、ボリューム・グループをインポートします。このステップにより、ディスク上の情報から新規デバイス構成データベース・エントリが作成されます。インポート後にボリューム・グループを次のシステム・ブート時に **autovaryon** に変更しないようにしてください。
3. SMIT の「**Problem Determination Tools (問題判別ツール)**」>「**Recover From PowerHA SystemMirror Script Failure (スクリプト障害からの回復)**」パネルを使用して、**clruncmd** コマンドを発行し、クラスター・マネージャーにシグナルを送信してクラスター処理を再開します。

## 問題 3

PowerHA SystemMirror ソフトウェアは、ボリューム・グループが見つからなかったために、**varyonvg** コマンドが失敗したことを示します。

## 解決方法 3

ボリューム・グループがシステムに定義されていません。ボリューム・グループが新規に作成されエクスポートされるか、**mksysb** システム・バックアップが復元された場合は、ボリューム・グループをインポートする必要があります。『問題 2』のステップに従って、正しいボリューム・グループ名が参照されているかどうかを検証します。

## 問題 4

PowerHA SystemMirror ソフトウェアは、次の論理ボリュームが不完全であるために **varyonvg** コマンドが失敗したことを表しています。

<name>

これが不完全です。

## 解決方法 4

これは、SMIT でボリューム・グループに対して強制 varyon 属性が構成されており、強制 varyon 操作の試行時に PowerHA SystemMirror がこのボリューム・グループに指定された論理ボリュームの単一の完全なコピーを検出しなかったことを示します。

また、強制 varyon 操作を要求しましたが、ミラーリングされた論理ボリュームに対して **super strict** 割り当てポリシーを指定しなかった可能性があります。この場合、**varyon** コマンドが成功する保証はありません。

関連情報:

HACMP リソース・グループの構成 (拡張)

共用 LVM コンポーネントの計画

### cl\_nfskill コマンドが失敗する

このトピックでは、**cl\_nfskill** コマンドが失敗する状態について説明します。

#### 問題

**/var/hacmp/log/hacmp.out** ファイルは、**cl\_nfskill** コマンドが NFS マウントされたファイル・システムを強制的にアンマウントしようとして失敗したことを示します。NFS には、**cl\_nfskill** コマンドにより強制的なアンマウントの影響を受けないファイル・システムの特定期限のロックがあります。

#### 解決方法

**cl\_nfskill** コマンドを実行する前に、別個のディレクトリーに **/etc/locks** ファイルをコピーします。その後で、元の **/etc/locks** ファイルを削除し、**cl\_nfskill** コマンドを実行します。コマンドが成功したら節約済みのコピーを使用して、**/etc/locks** ファイルを再作成します。

### cl\_scdiskreset コマンドが失敗する

このトピックでは、**cl\_scdiskreset** コマンドが失敗したときに発生するエラー・メッセージについて説明します。

#### 問題

**cl\_scdiskreset** コマンドは、エラー・メッセージを **/var/hacmp/log/hacmp.out** ファイルに記録します。SCSI デバイスの 1 つのシステムが保持した予約を削除するために、PowerHA SystemMirror ディスク・ユーティリティーは **cl\_scdiskreset** コマンドを発行します。バックレベルのハードウェアが SCSI バス (アダプター、ケーブルまたはデバイス) に存在するか、SCSI ID 競合がバスに存在する場合、**cl\_scdiskreset** コマンドは失敗することがあります。

#### 解決方法

SCSI アダプター、ケーブル、およびデバイスを検査するには、『クラスター・ログ・ファイルの使用』の該当するセクションを参照してください。最新のアダプターおよびケーブルを使用していることを確認します。各 SCSI デバイスの SCSI ID は、異なっている必要があります。

関連概念:

12 ページの『クラスター・ログ・ファイルの使用』

以下のトピックでは、PowerHA SystemMirror クラスター・ログ・ファイルを使用したクラスターのトラブルシューティング方法について説明します。一部のログについては、パラメーターの管理に関するセクションもいくつか含まれています。

## ブート時に **fsck** コマンドが失敗する

このトピックでは、ブート時に **fsck** コマンドが失敗する場合について説明します。

### 問題

ブート時に、AIX は **fsck** コマンドを実行して **check=true** 属性により **/etc/filesystems** にリストされたすべてのファイル・システムを検査します。AIX がファイル・システムを検査できない場合は、以下のエラーがレポートされます。

```
Filesystem Helper: 0506-519 Device open failed
```

### 解決方法

PowerHA SystemMirror で制御されるファイル・システムの場合、通常、このメッセージは問題を示していません。ファイル・システムの検査が失敗するのは、ファイル・システムを定義しているボリューム・グループがオンに変更されていないためです。ブート・プロシージャは、PowerHA SystemMirror 制御のボリューム・グループを自動的に **varyon** しません。

このメッセージが表示されないようにするには、PowerHA SystemMirror 制御下のすべてのファイル・システムの **/etc/filesystems** スタンザに **check=true** 属性を設定しないようにします。この属性を削除したり **check=false** に変更したりするには、**/etc/filesystems** ファイルを編集します。

## 指定のファイルシステムをシステムでマウントできない

このトピックでは、指定のファイルシステムをシステムでマウントできない状態について説明します。

### 問題

**/etc/filesystems** ファイルは論理ボリュームのログ名への変更を反映するように更新されていません。その論理ボリューム用のファイル・システムの作成後に、論理ボリューム名を変更すると、そのログの **/etc/filesystems** エントリは更新されません。このため、ファイル・システムをマウントするときに、PowerHA SystemMirror ソフトウェアが、論理ボリューム名に関する必要な情報を古いログ名から取得しようとしています。この情報は更新されていないため、ファイル・システムをマウントできません。

### 解決方法

論理ボリューム名を変更した後は、必ず **/etc/filesystems** ファイルを更新してください。

## クラスター・ディスク交換プロセスが失敗する

このトピックでは、クラスター・ディスク交換プロセスが失敗する状態についていくつか説明します。

### 問題

**node\_down** イベントが原因でディスク交換プロセスが完了しませんでした。

### 解決方法

ノードがオンラインに戻ったら、このノードで PowerHA SystemMirror を開始する前に、ボリューム・グループをエクスポートしてから再びインポートします。

## 問題 2

ディスク交換プロセスが、**replacepv** コマンドの実行中に失敗しました。

## 解決方法 2

`/tmp/replacepv` ディレクトリーを削除し、交換プロセスをやり直します。

別のディスクでこのプロセスを実行してみることもできます。

## 問題 3

VPATH デバイスが交換に使用可能ですが、「no free disks (空きディスクがありません)」というメッセージが表示されて、ディスク交換プロセスが失敗しました。

## 解決方法 3

必ず VPATH デバイスのボリューム・グループを `hdisks` に変換して、交換プロセスを再試行します。ディスクを交換したら、`hdisks` を VPATH デバイスに戻します。

関連情報:

共用 LVM コンポーネントの管理

## ファイルシステム変更がレイジー更新で認識されない

このトピックでは、レイジー更新で認識されないファイルシステム変更について説明します。

## 問題

ファイル・システム名を変更したり、ファイル・システムを除去してレイジー更新を実行する場合、レイジー更新は `imfs` コマンドを実行してから `imfs -lx` コマンドを実行します。これによって、フォールオーバー時に失敗が起きたり、PowerHA SystemMirror クラスタ・サービスの再起動に失敗することがあります。

## 解決方法

C-SPOC ユーティリティーを使用して、ファイル・システムを変更または除去します。これによって、`imfs -lx` が実行された後 `imfs` が実行され、クラスタのすべてのノードで変更が更新されます。

エラー報告機能は、クラスタ全体におけるボリューム・グループ状態の矛盾に関して、詳細な情報を提供します。エラー報告機能が動作した場合、手動で修正措置を取る必要があります。ファイル・システムの変更がすべてのノード上で更新されない場合、この情報を使用して手動でノードを更新します。

## clam\_nfsv4 アプリケーション・モニター障害

このトピックでは、`clam_nfsv4` アプリケーション・モニターに障害が起こって反応しなくなった場合に問題を修正する方法について説明します。

## 問題

`clam_nfsv4` アプリケーション・モニターが完了するまでの時間が 60 秒を超えました。モニターは応答せず、停止しました。そのため、ネットワーク・ファイルシステム (NFS) ノードでフォールオーバーが起きました。このフォールオーバーは、通常、アプリケーション・モニターをホストするシステムのパフォーマンス・ワークロードが高いときに発生します。

## 解決方法

この問題を修正するには、システムのワークロードを減らす必要があります。また、システムに APAR IV08873 を適用して、clam\_nfsv4 アプリケーション・モニター・スクリプトの実行にかかる時間を短縮することもできます。

関連情報:

clam\_nfsv4 アプリケーション・モニターの概念

PowerHA SystemMirror での NFS の使用

PowerHA SystemMirror での NFS クロスマウント

 [APAR IV08873: NFSV4 モニター・スクリプト実行時間の改善](#)

## LVM 分割サイト・ミラーリングのトラブルシューティング

PowerHA SystemMirror および LVM には、ミラー・プールの定義時に指定された情報を除いて、ディスクの物理的なロケーションに関する情報はありません。

以下の情報を検討して、LVM 分割サイト・ミラーリングの問題に関する可能な解決策を確認します。

- コマンド行から `smitty cl_mirrorpool_mgt` と入力し、「**Show Mirror Pools for a Volume Group** (ボリューム・グループのミラー・プールの表示)」を選択して、ミラー・プールへのディスクの割り当てを検証します。
- コマンド行から `smitty cl_lv` と入力し、「**Show Characteristics of a Logical Volume** (論理ボリューム特性の表示)」を選択して、個々のファイルシステムおよび論理ボリュームのミラーリングが正しいことを検証します。
- コマンド行から `smitty cl_vgsc` と入力し、「**Change/Show characteristics of a Volume Group** (ボリューム・グループの特性を変更/表示)」を選択して、ボリューム・グループが非常に厳密であることを検証します。
- 再同期が失敗した場合は、AIX エラー・ログで、ボリューム・グループのディスクに関連した問題を調べます。ボリューム・グループを手動で再同期するには、コマンド行から `smitty cl_syncvg` と入力し、「**Synchronize LVM Mirrors by Volume Group** (ボリューム・グループによる LVM ミラーリングの同期化)」を選択します。

## リポジトリー・ディスクのトラブルシューティング

クラスター内のいずれかのノードがリポジトリー・ディスクのエラーまたはディスク・アクセス中の障害を検出した場合、クラスターは限定された、すなわち、制限された操作モードになります。この操作モードでは、トポロジー関連の操作の大部分は実行できず、再始動したノードはいずれもクラスターに再結合できません。

リポジトリー・ディスクに障害が起こると、ディスク障害通知が出されます。PowerHA SystemMirror は、リポジトリー・ディスク障害通知を、解決されるまで、出し続けます。

リポジトリー・ディスクにどのような問題が起こっているかを判別するには、以下のログ・ファイルを調べてください。

- `hacmp.out`
- AIX エラー・ログ (`errpt` コマンドを使用)

## 例: hacmp.out ログ

次に示すものは、リポジトリー・ディスクに障害が起こった場合に hacmp.out ログ・ファイルに書き込まれるエラー・メッセージの一例です。

```
ERROR: rep_disk_notify : Tue Jan 10 13:38:22 CST 2012 : Node "r6r4m32"  
(0x54628FEA1D0611E183EE001A64B90DF0) on Cluster r6r4m31_32_33_34 has lost access to repository  
disk hdisk75.
```

## 例: AIX エラー・ログ

ノードがリポジトリー・ディスクにアクセスできなくなると、問題のあるノードごとに AIX エラー・ログに項目が立てられます。

次に示すものは、リポジトリー・ディスクに障害が起こった場合にエラー・ログ・ファイルに書き込まれるエラー・メッセージの一例です。

注: AIX エラー・ログを表示するには、**errpt** コマンドを使用してください。

```
LABEL:          OPMSG  
IDENTIFIER:     AA8AB241
```

```
Date/Time:      Tue Jan 10 13:38:22 CST 2012  
Sequence Number: 21581  
Machine Id:     00CDB2C14C00  
Node Id:        r6r4m32  
Class:          0  
Type:           TEMP  
WPAR:           Global  
Resource Name:  clevmgrd
```

```
Description  
OPERATOR NOTIFICATION
```

```
User Causes  
ERRLOGGER COMMAND
```

```
Recommended Actions  
REVIEW DETAILED DATA
```

```
Detail Data  
MESSAGE FROM ERRLOGGER COMMAND  
Error: Node 0x54628FEA1D0611E183EE001A64B90DF0 has lost access to repository disk hdisk75.
```

## 障害または破損のあるリポジトリー・ディスクの交換

リポジトリー・ディスクに障害が発生した場合、すべてのクラスターの動作を回復するために、そのリポジトリー・ディスクを異なるディスク上で復旧する必要があります。ご使用のクラスター環境およびリポジトリー・ディスクの障害のタイプによって、リポジトリー・ディスクの復旧に使用可能な方法は異なります。

以下に、リポジトリー・ディスクに障害が発生した場合の 2 つのシナリオ、およびそのリポジトリー・ディスクを新しいストレージ・ディスク上で復旧する方法について示します。

リポジトリー・ディスクに障害が発生したが、クラスターは作動可能である

このシナリオでは、クラスター内の 1 つ以上のノードでリポジトリー・ディスクのアクセスが失われているとします。この障害が発生すると、Cluster Aware AIX (CAA) は、メモリーにキャッシュされているリポジトリー・ディスクの情報を使用して、制限されたモードで動作を続行しま

す。CAA がクラスター内の 1 つのノードでアクティブなままである場合、古いリポジトリ・ディスクからの情報を使用して、新しいリポジトリ・ディスクを再ビルドすることができます。障害の発生後にリポジトリ・ディスクを再ビルドするには、CAA がアクティブになっているノードから以下の手順を実行します。

1. **lscluster -c** コマンド、次に **lscluster -m** コマンドを使用して、CAA がノード上でアクティブであることを確認します。
2. SMIT を使用したりポジトリ・ディスクの交換のトピックにある手順を実行して、リポジトリ・ディスクを交換します。PowerHA SystemMirror が問題を認識し、新しいストレージ・ディスク上にリポジトリ・ディスクを再ビルドするために CAA と相互に作用します。

注: この手順では、PowerHA SystemMirror の構成データに保管されたりポジトリ情報が更新されます。

3. SMIT インターフェースから「クラスター・ノードおよびネットワーク」 > 「クラスター構成の検証と同期化」を選択し、PowerHA SystemMirror クラスター構成情報を同期化します。

リポジトリ・ディスクに障害が発生し、クラスター内のノードがリブートした

まれに、以下のシナリオのようにクリティカルな障害が連続して発生することにより、リポジトリ・ディスクへのアクセスが失われ、クラスター内のすべてのノードがリブートされるという最悪のケースとなる場合があります。これにより、障害発生時にクラスター内のすべてのノードがオンラインではなくなるため、AIX オペレーティング・システムのメモリーからリポジトリ・ディスクを再ビルドすることはできません。ノードが再びオンラインに戻っても、クラスター内にリポジトリ・ディスクが存在しないために、ノードは CAA を始動することができません。この問題を修正するために理想的なのは、リポジトリ・ディスクを復旧し、クラスターを自己回復させることです。この方法が不可能な場合、新しいストレージ・ディスク上にリポジトリ・ディスクを再ビルドし、そのリポジトリ・ディスクを使用して CAA クラスターを始動する必要があります。

リポジトリ・ディスクを再ビルドしてクラスター・サービスを開始するには、以下の手順を実行します。

1. SMIT を使用したりポジトリ・ディスクの交換のトピックにある手順を実行して、クラスター内の 1 つのノードでリポジトリを再ビルドします。PowerHA SystemMirror が問題を認識し、新しいストレージ・ディスク上にリポジトリ・ディスクを再ビルドするために CAA と相互に作用します。

注: この手順では、PowerHA SystemMirror の構成データに保管されたりポジトリ情報が更新され、CAA クラスター・キャッシュ・ファイルからリポジトリ・ディスクが再ビルドされます。

2. クラスター・サービスの開始のトピックにある手順を実行して、リポジトリ・ディスクをホストするノード上でクラスター・サービスを開始します。
3. クラスター内の他のすべてのノードは、元のリポジトリ・ディスクへのアクセス試行を続行します。新しいリポジトリ・ディスクを使用するようにこれらのノードを構成して、CAA クラスター・サービスを開始する必要があります。**lscluster -m** コマンドを使用して、これらのいずれのノードでも CAA クラスターがアクティブになっていないことを確認します。CAA クラスターが非アクティブであるか、またはローカル・ノードがダウン状態である場合は、次のコマンドを入力して、古いリポジトリ・ディスクの情報を削除します。

```
export CAA_FORCE_ENABLED=true
clusterconf -fu
```

4. 他のノードを CAA クラスターに参加させるには、新規作成したりポジトリ・ディスクについて、アクティブなノードで次のコマンドを使用します。

```
clusterconf -p
```

- 最初に **lscluster -c** コマンド、次に **lscluster -m** コマンドを使用して、CAA がアクティブであることを確認します。
- SMIT インターフェースから「クラスター・ノードおよびネットワーク」 > 「クラスター構成の検証と同期化」を選択し、新規作成したリポジトリ・ディスクに関する PowerHA SystemMirror クラスター構成情報を、他のすべてのノードに同期化します。
- SMIT インターフェースから「システム管理 (C-SPOC)」 > 「PowerHA SystemMirror サービス」 > 「クラスター・サービスの開始」を選択して、(リポジトリ・ディスクが作成された最初のノードに加えて) すべてのノードで PowerHA SystemMirror クラスター・サービスを開始します。

## スナップショットのマイグレーションおよびリポジトリ・ディスク

オンライン・クラスターでのスナップショットのマイグレーション・プロセスを行うには、スナップショットのクラスター情報がオンラインのクラスター情報と一致している必要があります。この要件は、リポジトリ・ディスクにも適用されます。リポジトリ・ディスクの構成を変更した場合、これらの変更を反映するためにスナップショットを更新してから、スナップショットのマイグレーション・プロセスを実行する必要があります。

関連情報:

リポジトリ・ディスクのプランニング

リポジトリ・ディスクの障害

クラスター構成のスナップショットの作成

スナップショットを使用した PowerHA SystemMirror のアップグレード

## ディスク・フェンシングのトラブルシューティング

ディスク・フェンシングは、PowerHA SystemMirror での検疫ポリシーにのみ使用できます。

### 問題

ディスク・フェンシングが、ご使用の環境に不要になりました。ディスク・フェンシングを無効にして、ディスクやボリューム・グループの予約を解除できます。

### 解決方法

ディスク・フェンシングを無効にして、ディスクやボリューム・グループの予約を解除するには、以下の手順を実行します。

- コマンド行から以下のコマンドを実行して、ディスクまたはボリューム・グループから予約を解除します。

```
clgmr modify physical_volume <disk> scsibr_clear={yes}  
clgmr modify volume_group <vg> scsibr_clear={yes}
```

*disk* はディスクの名前です。 *vg* はボリューム・グループの名前です。

- コマンド行から **smit sysmirror** と入力します。
- SMIT インターフェースから「ユーザー定義クラスター構成」 > 「クラスター・ノードおよびネットワーク」 > 「初期クラスター・セットアップ (ユーザー定義)」 > 「クラスター分割およびマージ・ポリシーの構成」 > 「検疫ポリシー」 > 「ディスク・フェンシング」を選択し、Enter を押します。

4. 「ディスク・フェンシング」フィールドに対して「いいえ」を指定し、「クリティカル・リソース・グループ」フィールドでクリティカル・リソース・グループを指定します。Enter を押して変更を保存します。
5. 「検疫ポリシー」パネルで「アクティブ・ノード停止ポリシー」 > 「アクティブ・ノード停止ポリシーの構成」を選択し、Enter を押します。
6. 「アクティブ・ノード停止ポリシー」フィールドに対して「いいえ」を指定し、「クリティカル・リソース・グループ」フィールドでクリティカル・リソース・グループを指定します。Enter を押して変更を保存します。

注: 指定するクリティカル・リソース・グループは、ステップ 4 で指定したクリティカル・リソース・グループと同じでなければなりません。

## 問題

アクティブ・クラスターにおいてリソース・グループがエラー状態になります。そのリソース・グループがエラー状態になるのは、ノードにおいてリソース・グループ内の単一ボリューム・グループを登録して予約することができないためです。

## 解決方法

リソース・グループに関するこの問題を修正するには、以下のいずれかのオプションを使用します。

- **cl\_scsiopr\_revoover\_rg** スクリプトを実行します。 **cl\_scsiopr\_revoover\_rg** スクリプトにより、エラー状態にあるリソース・グループのボリューム・グループが登録されて予約されます。
- SMIT インターフェースを使用してこの問題を修正するには、以下の手順を実行します。
  1. コマンド行から **smit sysmirror** と入力します。
  2. SMIT インターフェースから「問題判別ツール」 > 「SCSI 永続予約エラーからリソース・グループを回復 (**Recover Resource Group from SCSI Persistent Reserve Error**)」を選択し、Enter を押します。
  3. エラー状態にあるリソースを選択し、Enter を押します。
  4. SMIT インターフェースから「システム管理 (**C-SPOC**)」 > 「リソース・グループおよびアプリケーション」 > 「リソース・グループのオンライン」を選択し、Enter を押します。
  5. 再びオンラインにするリソース・グループを選択し、Enter を押します。

コマンドが実行されたり特定のイベントが発生したりしたときのディスク・フェンシングに関する各種シナリオを次の表に示します。これらのシナリオの構成では、サイトに NodeA (クリティカル・リソース・グループが含まれる) および NodeB (クリティカル・リソース・グループは含まれない) が含まれています。また、この構成では、リソース・グループに属するすべてのディスク上で NodeA と NodeB が登録されます。

表 16. ディスク・フェンシング・シナリオ

シナリオ	NodeA の観察	NodeB の観察
hmc シャットダウン	NodeA はディスク上に登録されます。	NodeB はディスク上に登録されません。
hmc リブート	NodeA はディスク上に登録されます。	NodeB はディスク上に登録されません。
reboot	リブート後であっても、リソース・グループが取得されるより前にリブートが行われたため、ディスクは依然としてそのままです。	NodeB はディスク上に登録されません。

表 16. ディスク・フェンシング・シナリオ (続き)

シナリオ	NodeA の観察	NodeB の観察
reboot -q	リブート後であっても、リソース・グループが取得されるより前にリブートが行われたため、ディスクは依然としてそのままです。	NodeB はディスク上に登録されません。
shutdown -Fr	NodeA はディスク上に登録されません。	NodeB はディスク上に登録されません。
shutdown	NodeA はディスク上に登録されます。	NodeB はディスク上に登録されません。
halt -q	NodeA はディスク上に登録されます。	NodeB はディスク上に登録されません。
halt	NodeA はディスク上に登録されます。	NodeB はディスク上に登録されません。
ノード・クラッシュ	NodeA はディスク上に登録されます。	NodeB はディスク上に登録されません。
clstop (リソース・グループのオフライン)	NodeA はディスク上に登録されます。	NodeB はディスク上に登録されます。
clstop (リソース・グループの移動)	NodeA はディスク上に登録されます。	NodeB はディスク上に登録されます。
clstop (リソース・グループの管理解除)	NodeA はディスク上に登録されません。	NodeB はディスク上に登録されません。

関連情報:

ディスク・フェンシングの計画  
 検疫ポリシーの構成

## ネットワークとスイッチの問題

以下のトピックでは、潜在的なネットワークおよびスイッチの問題について説明します。

### スイッチ・ネットワークでの予期しないネットワーク・インターフェース障害

このトピックでは、スイッチ・ネットワークを使用した PowerHA SystemMirror 構成での予期しないネットワーク・インターフェース障害について説明します。

#### 問題

スイッチ・ネットワークを使用した PowerHA SystemMirror 構成では、ネットワークとスイッチが正しく定義および構成されていないと、予期しないネットワーク・インターフェース障害が発生する可能性があります。

#### 解決方法

スイッチとネットワークを正しく構成してください。

関連情報:

スイッチ・ネットワークの PowerHA SystemMirror 構成

### ネットワーク検証におけるマルチキャスト

デフォルトでは、PowerHA SystemMirror はハートビートにユニキャスト通信を使用します。クラスター通信では、使用するネットワークがマルチキャスト通信をサポートするように構成されている場合、オプションとしてマルチキャスト・アドレスを構成することを選択できるか、CAA に自動的にマルチキャスト・アドレスを選択させることができます。マルチキャスト通信の使用を選択した場合は、クラスターに含まれるすべてのノード間でマルチキャスト・パケットを正常に送信できることを確認するまでは、クラスターの作成を試みないでください。

ネットワーク上にクラスターを作成するために使用するすべてのノードについてエンドツーエンド・マルチキャスト通信をテストするには、**mping** コマンドを実行してノード間でパケットの送受信を行います。

PowerHA SystemMirror 7.1.1 以降を使用する場合、**mping** コマンドが失敗するとクラスターを作成できません。**mping** コマンドが失敗した場合、ネットワークはマルチキャスト通信に正しくセットアップされていません。そのような場合は、スイッチおよびルーターの資料を検討してマルチキャスト通信を使用可能にしてください。

特定のマルチキャスト・アドレスを指定して **mping** コマンドを実行できます。指定しない場合はデフォルトのマルチキャスト・アドレスが使用されます。**mping** コマンドの入力に使用するマルチキャスト・アドレスは、クラスターの作成に使用するものと同じでなければなりません。

注: **mping** コマンドではデフォルトの経路を持つインターフェースが使用されます。デフォルトの経路を持たない別のインターフェースでマルチキャスト通信をテストするために **mping** コマンドを使用する場合、マルチキャスト IP アドレスへの必要なインターフェースがある静的ルートを一時的に追加する必要があります。

以下の例では、**mping** コマンドの成功例と失敗例を示します。ノード A は受信側、ノード B は送信側です。

成功例:

#### Receiver

```
root@nodeA:/# mping -r -R -c 5
mping version 1.1
Listening on 227.1.1.1/4098:

Replying to mping from 9.3.207.195 (nodeB.aus.stglabs.ibm.com) bytes=32 seqno=0 ttl=1
Replying to mping from 9.3.207.195 (nodeB.aus.stglabs.ibm.com) bytes=32 seqno=1 ttl=1
Replying to mping from 9.3.207.195 (nodeB.aus.stglabs.ibm.com) bytes=32 seqno=2 ttl=1
Replying to mping from 9.3.207.195 (nodeB.aus.stglabs.ibm.com) bytes=32 seqno=3 ttl=1
Replying to mping from 9.3.207.195 (nodeB.aus.stglabs.ibm.com) bytes=32 seqno=4 ttl=1
```

#### Sender

```
root@nodeB:/# mping -R -s -c 5
mping version 1.1
mpinging 227.1.1.1/4098 with ttl=1:

32 bytes from 9.3.207.190 (nodeA.aus.stglabs.ibm.com) seqno=0 ttl=1 time=0.985 ms
32 bytes from 9.3.207.190 (nodeA.aus.stglabs.ibm.com) seqno=1 ttl=1 time=0.958 ms
32 bytes from 9.3.207.190 (nodeA.aus.stglabs.ibm.com) seqno=2 ttl=1 time=0.998 ms
32 bytes from 9.3.207.190 (nodeA.aus.stglabs.ibm.com) seqno=3 ttl=1 time=0.863 ms
32 bytes from 9.3.207.190 (nodeA.aus.stglabs.ibm.com) seqno=4 ttl=1 time=0.903 ms

--- 227.1.1.1 mping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.863/0.941/0.998 ms
```

失敗例:

#### Receiver

```
root@nodeA:/# mping -r -R -c 5 -6
mping version 1.1
Listening on ff05::7F01:0101/4098:

Replying to mping from fe80::18ae:19ff:fe72:1a15 bytes=48 seqno=0 ttl=1
Replying to mping from fe80::18ae:19ff:fe72:1a15 bytes=48 seqno=1 ttl=1
Replying to mping from fe80::18ae:19ff:fe72:1a15 bytes=48 seqno=2 ttl=1
Replying to mping from fe80::18ae:19ff:fe72:1a15 bytes=48 seqno=3 ttl=1
Replying to mping from fe80::18ae:19ff:fe72:1a15 bytes=48 seqno=4 ttl=1
```

#### Sender

```
root@nodeB:/# mping -R -s -c 5 -6
mping version 1.1
mpinging ff05::7F01:0101/4098 with ttl=1:

--- ff05::7F01:0101 mping statistics ---
5 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.000/0.000/0.000 ms
```

注: 結果を検証するには、**mping** コマンドの送信側のみをチェックする必要があります。また、パケット・ロスのパーセンテージもメモしておいてください。ネットワーク上でマルチキャストが機能しているかどうかを確認するには、両方のノードを送信側と受信側の両方としてテストするように、**mping** テストを実行する必要があります。一般的に、非詳細出力でも必要な情報を得ることができます。ただし、**mping** コマンドで **-v** フラグの使用を選択した場合、プログラム内部に関して十分な知識が必要です。そうでない場合、詳細出力の理解を誤ってしまう可能性があります。また、**mping** コマンドの送信側の戻りコードもチェックすることができます。エラーが発生すると、送信側は **255** を返します。成功時には **0** を返します。

クラスターの作成時にマルチキャスト・アドレスを指定しなかった場合、Cluster Aware AIX (CAA) はデフォルトのマルチキャスト・アドレスを選択します。このデフォルトのマルチキャスト・アドレスは、値 (228.0.0.0) と、ノード IP アドレスの下位 24 ビットの論理和として作成されます。例えば、IP アドレスが 9.3.199.45 であれば、デフォルトのマルチキャスト・アドレスは 228.3.199.45 です。

PowerHA SystemMirror 7.1.2 以降ではインターネット・プロトコル・バージョン 6 (IPv6) アドレスがサポートされます。クラスターで IPv6 アドレスが構成されると、Cluster Aware AIX (CAA) は IPv6 マルチキャスト・アドレスを使用して IPv6 アドレスのハートビートを活動化します。環境内の IPv6 接続でマルチキャスト・アドレスとの通信が可能であることを確認する必要があります。

環境内で IPv6 マルチキャスト通信が正しく構成されているかどうかを検証するには、**-6** オプションを指定した **mping** コマンドを実行します。この **mping** コマンドを実行すると、デフォルトの IPv6 マルチキャスト・アドレスとの IPv6 マルチキャスト通信の検証が行われます。特定の IPv6 マルチキャスト・アドレスを指定するには、**-a** オプションと IPv6 マルチキャスト・アドレスを指定して **mping** コマンドを実行します。**-a** オプションを使用すれば、**-6** オプションの指定は必要ありません。**mping** コマンドは、**-a** オプションで渡されたアドレスのファミリーを自動的に判別します。

関連情報:

 Cisco マルチキャスト・スイッチのトラブルシューティング

 Cisco スwitchのマルチキャスト・サポート

## マルチキャストのトラブルシューティング

**mping** コマンドを使用して、ノードでマルチキャスト・パケットの送受信ができるかどうかをテストします。**mping** コマンドが失敗した場合、ネットワーク環境内の問題を特定する必要があります。

ネットワーク内のマルチキャスト問題のトラブルシューティングを行うには、以下のガイドラインを検討してください。

- マルチキャスト通信に使用しているスイッチの資料を検討します。
- マルチキャスト通信に使用しているスイッチ上の Internet Group Management Protocol (IGMP) スヌープ機能を使用不可にします。

注: ネットワーク・インフラストラクチャーのため IGMP スヌープ機能を永続的に使用不可にできない場合は、スイッチ上のスヌープ機能を一時的に使用不可にしてから、追加のネットワーク・コンポーネントを一度に 1 つずつ追加して、問題のトラブルシューティングを行うことができます。

- クラスターのノード間のカスケード・スイッチをすべて除去します。つまり、クラスターのノード間には単一のスイッチのみがあるようにします。

関連情報:

 Cisco マルチキャスト・スイッチのトラブルシューティング

 Cisco スwitchのマルチキャスト・サポート

## ユニキャストのトラブルシューティング

デフォルトでは、PowerHA SystemMirror はクラスター内のノード間で、ユニキャスト・ソケット・ベースの通信を使用します。

ユニキャスト通信に問題がある場合は、汎用のネットワーク・トラブルシューティング手順に従います。その例を次に示します。

- **ifconfig** コマンドと **netstat** コマンドを使用して、IP アドレスの構成とルーティングを確認します。
- **ping** コマンドと **tracert** コマンドを使用して、ノードとアダプターが通信できることを確認します。
- 上記の手順で問題を識別できない場合は、**iptrace** コマンドを使用して、低レベルのパケット・アクティビティをトレースします。

## システム・リブート時の IPv6 アドレスの持続化

インターネット・プロトコル・バージョン 6 (IPv6) は、AIX オペレーティング・システムと同様に動的構成用に設計されています。IPv6 アドレスはシステム・リブート操作時に持続しません。

リブート後に IPv6 アドレスを構成するには、**autoconf6** コマンドを手動で実行します。あるいは、クラスター・サービスを開始する前に PowerHA SystemMirror に **autoconf6** コマンドを自動的に実行させることもできます。

AIX オペレーティング・システム用に自動的に実行されるように **autoconf6** コマンドを構成するには、以下のステップを実行して `/etc/rc.tcpip` ファイルを変更します。

1. **autoconf6** コマンドが実行されるように以下の行のコメントを外します。

```
# Start up autoconf6 process
start /usr/sbin/autoconf6
```

注: **-i** フラグを入力して個別のインターフェースを指定することができます。例:

```
# Start up autoconf6 process
start /usr/sbin/autoconf6 "" "-i en1"
```

2. **ndpd** デーモンが開始されるように以下の行のコメントを外します。

```
# Start up ndpd-host daemon
start /usr/sbin/ndpd-host "$src_running"

# Start up the ndpd-router daemon
start /usr/sbin/ndpd-router "$src_running"
```

関連情報:

**autoconf6** コマンド

**ndpd-host** デーモン

ndpd-router デーモン

## VLAN のトラブルシューティング

このトピックでは、仮想ローカル・エリア・ネットワークのインターフェース障害のトラブルシューティングについて説明します。

### 問題

仮想 LAN ネットワーク (これ以降は VLAN (仮想ローカル・エリア・ネットワーク) と表記) でのインターフェース障害

### 解決方法

PowerHA SystemMirror に定義されている VLAN インターフェースをトラブルシューティングして、インターフェース障害を検出するには、これらのインターフェースを単一アダプター・ネットワーク上に定義されたインターフェースと見なします。

具体的には、VLAN に属するネットワーク・インターフェースを `/usr/es/sbin/cluster/etc/clinfo.rc` スクリプトの `ping_client_list` 変数にリストして、`clinfo` を実行します。このようにすることで、クラスター・イベントが発生すると必ず、`clinfo` は、リストされているネットワーク・インターフェースをモニターして障害を検出します。仮想ローカル・エリア・ネットワークの性質上、ネットワーク・インターフェースの障害を検出するこれ以外のメカニズムは無効です。

### クラスター・ノードが通信できない

このトピックでは、分割されたクラスターがある場合の問題と解決策について説明します。

### 問題

構成において複数のノードを単一のネットワークで接続している場合は、クラスターが分割される場合があります。クラスターの分割が発生するのは、クラスター・ノードが通信できない場合です。通常的环境下、ノードのサービス・ネットワーク・インターフェースの障害により、クラスター・マネージャーは `swap_adapter` イベントを認識し、処理します。つまり、サービス IP ラベル/アドレスが別の IP ラベル/アドレスに交換されます。ハートビートは共有ディスクを介して交換されます。しかし、ノードがクラスターから分離される可能性があります。他のノードのクラスター・マネージャーが実行された `swap_adapter` イベントを認識しても、通信パスが存在しないため現在分離された (分割された) ノードに通信できません。

### 解決方法

単一障害点がなくなるようにネットワークを構成します。

### 分散 SMIT で予測不能な結果が発生する

このトピックでは、PowerHA SystemMirror クラスター・サービスの開始/停止以外の操作で分散 SMIT を使用したときの問題と解決策について説明します。

### 問題

PowerHA SystemMirror クラスター・サービスの始動および停止以外の操作に AIX ユーティリティー DSMIT を使用すると、予測不能な結果が発生する場合があります。

## 解決方法

DSMIT はネットワーク化された IBM System p プロセッサの操作を管理します。これには、全ネットワーク・ノードにおける AIX コマンドの実行を制御するために必要なロジックが組み込まれています。PowerHA SystemMirror 機能と競合する可能性があるため、DSMIT の使用は PowerHA SystemMirror クラスター・サービスの始動および停止に限定してください。

## PCI ホット・プラグ NIC 障害からの回復

このトピックでは、PCI ホット・プラグ NIC 障害からの回復について説明します。

### 問題

回復不能エラーが原因で PCI ホット・リプレース・プロセスが失敗した場合、NIC が未構成のままの状態になり、ノードが保守モードのままの状態になる場合があります。新しい NIC やその NIC が挿入されている PCI スロットがこの時点で損傷している可能性があります。

## 解決方法

このノードを完全に稼働した状態に回復するには、ユーザーの介入が必要です。

関連情報:

オペレーティング・システムおよびデバイスの管理

## PowerHA SystemMirror の IP ラベルが AIX インターフェースから切断される

このトピックでは、PowerHA SystemMirror の IP ラベルが AIX インターフェースから切断される状態について説明します。

### 問題

PowerHA SystemMirror の IP ラベルを入力または選択してクラスター構成にネットワーク・インターフェースを定義すると、PowerHA SystemMirror は関連する AIX ネットワーク・インターフェース名をディスクバーします。PowerHA SystemMirror ではこの関係が未変更であると仮定します。クラスターを構成および同期化したあとに AIX ネットワーク・インターフェースの名前を変更した場合は、PowerHA SystemMirror は正しく機能しません。

## 解決方法

この問題が発生した場合、SMIT PowerHA SystemMirror 「**System Management (C-SPOC) (システム管理 (C-SPOC))**」パネルからネットワーク・インターフェース名をリセットできます。

関連情報:

クラスター・リソースの管理

## データ送信時にパケットが損失する

このトピックでは、データが遷移時に断続的に失われる場合の問題と解決策について説明します。

### 問題

送信中に途中でデータを損失した場合、各ノード上で最大送信単位 (MTU) の設定が異なっている可能性があります。例えば、ノード A が 1.5 K のパケットを受信できるノード B に 8 K のパケットを送った場合、ノード B はメッセージが完了したと認識しますが、データは失われていることがあります。

## 解決方法

クラスターの検証ユーティリティを実行して、同じネットワーク時にすべてのクラスター・ノード上のすべてのネットワーク・インターフェース・カードが、MTU サイズに関して同等に設定されていることを確認します。MTU サイズがネットワーク上で整合していない場合、エラーが表示されるので、これによって調整するノードを判別できます。

注: MTU サイズは、以下のコマンドを使用して変更できます。

```
chev -l en0 -a mtu=<new_value_from_1_to_8>
```

## クラスター通信の問題

以下のトピックでは、潜在的なクラスター通信の問題について説明します。

### メッセージの暗号化が失敗する

このトピックでは、メッセージの暗号化が失敗したときの問題と解決策について説明します。

#### 問題

暗号化または暗号化解除が、セキュリティーの有効化後に失敗し、**clcomd** デーモンのコミュニケーションが複数のノードで失敗する。暗号化または暗号化解除が失敗する可能性があるか検査するには、**clcomddiag.log** ファイルを表示します。

#### 解決方法

マスター・ノードまたは任意のノードから **SMIT** を使用してセキュリティーを無効にしてから、すべてのノードの **PowerHA SystemMirror** コミュニケーション・デーモンを停止して始動します。

セキュリティーの有効化前にインストールされた以下のファイル・セットをクラスター・ノードが持っているかどうか確認します。

- DES メッセージ認証を使用したデータ暗号化の場合: **rsct.crypt.des**
- 標準のトリプル DES メッセージ認証を使用したデータ暗号化の場合: **rsct.crypt.3des**
- Advanced Encryption Standard (AES) メッセージ認証を使用したデータ暗号化の場合: **rsct.crypt.aes256**。 **clic** バージョン 4.7 のファイル・セットをインストールしたに違いありません。

必要に応じて、AIX 拡張パック CD-ROM から、これらのファイルセットをインストールします。

**PowerHA SystemMirror** の実行後にファイルセットをインストールする場合は、**PowerHA SystemMirror** クラスター通信デーモンを始動して停止し、**PowerHA SystemMirror** がこれらのファイルセットを使用できるようにします。クラスター通信デーモンを再始動するには、次を実行します。

```
stopscr -s clcomd  
startsrc -s clcomd
```

ファイルセットが存在し、暗号化エラーが表示される場合、**PowerHA SystemMirror** の実行後に暗号化ファイルセットがインストールされたか、再インストールされた可能性があります。この場合は、上述したようにクラスター通信デーモンを再始動します。

### クラスター・ノードが相互に通信しない

このトピックでは、相互に通信しないクラスター・ノードについて説明します。

## 問題

クラスター・ノードは相互に通信できないため、以下のいずれかを構成しました。

- メッセージ認証、またはメッセージ認証および暗号化を使用可能にしました。
- VPN トンネルの永続 IP ラベルの使用

## 解決方法

ネットワークが操作可能であることを確認します。『ネットワークとスイッチの問題』を参照してください。

クラスターに永続 IP ラベルが設定されているかどうかを検査します。設定されている場合は、正しい設定がされているかどうか、IP ラベルを ping できるかどうかを確認します。

メッセージ認証、またはメッセージ認証および暗号化を使用している場合は、以下のように実行してください。

- メッセージ認証モードに対する各クラスター・ノードの設定が同じであるかどうかを確認します。モードが異なる場合は、各ノードでメッセージ認証を「None (なし)」に設定し、メッセージ認証を再構成します。
- 各ノードで `/usr/es/sbin/cluster/etc` ディレクトリーに同じタイプの暗号鍵が設定されているかどうかを確認します。暗号鍵は、他のディレクトリーに存在できません。

VPN に永続 IP ラベルを構成した場合は、以下のように実行します。

1. 「**User Persistent Labels (ユーザー永続ラベル)**」を「**No (いいえ)**」に変更します。
2. Synchronize Cluster Configuration (クラスター構成を同期化する)
3. 「**User Persistent Labels (ユーザー永続ラベル)**」を「**Yes (はい)**」に変更します。

関連概念:

72 ページの『ネットワークとスイッチの問題』

以下のトピックでは、潜在的なネットワークおよびスイッチの問題について説明します。

## PowerHA SystemMirror のテークオーバーの問題

以下のトピックでは、潜在的なテークオーバーの問題について説明します。

PowerHA SystemMirror のリソース・グループの移動を調査し、**rg\_move** イベントが発生した理由を知るには、必ず `/var/hacmp/log/hacmp.out` ファイルを調べてください。特に PowerHA SystemMirror において、リソース・グループの処理方法やフォールオーバー環境の優先順位付けが最近変更されたので **hacmp.out** ファイルおよびイベント要約が、リソース・グループのアクティビティと場所を追跡する際にさらに重要になりました。さらに、リソース・グループの並列処理の場合、**hacmp.out** ファイルはクラスター・ヒストリー・ログまたは **clstrmgr.debug** ログ・ファイルに示されない詳細を報告します。テークオーバー・アクティビティの後にリソース・グループの移動を調査する場合は、必ず早い段階で **hacmp.out** ログを検査してください。

### テークオーバー時に **varyonvg** コマンドが失敗する

このトピックでは、PowerHA SystemMirror ソフトウェアが共用ボリューム・グループを varyon できなかった理由について説明します。

## 問題

PowerHA SystemMirror ソフトウェアが共有ボリューム・グループを varyon でできませんでした。  
PowerHA SystemMirror 構成データベース・オブジェクト・クラスにボリューム・グループ名がないか、  
ボリューム・グループ名が誤っています。

## 解決方法

- `/var/hacmp/log/hacmp.out` ファイルを検査し、`varyonvg` の失敗に関連するエラーがないかどうかを検査してください。
- `lsvg` コマンドを使用して、システムが認識しているすべてのボリューム・グループをリストして、PowerHA SystemMirror リソース構成データベース・オブジェクト・クラスで使用されているボリューム・グループ名が正しいかどうかを検査します。構成データベース内のボリューム・グループ名を変更するには、メインの PowerHA SystemMirror SMIT パネルで、「**Initialization and Standard Configuration (初期化および標準構成)**」>「**Configure PowerHA SystemMirror Resource Groups (PowerHA SystemMirror リソース・グループの構成)**」>「**Change/Show Resource Groups (リソース・グループの変更/表示)**」を選択し、ボリューム・グループを含めるリソース・グループを選択します。「**Change/Show Resources and Attributes for a Resource Group (リソース・グループのリソースおよび属性の変更/表示)**」パネルの「**Volume Groups (ボリューム・グループ)**」または「**Concurrent Volume Groups (コンカレント・ボリューム・グループ)**」フィールドを使用して、ボリューム・グループ名を設定します。問題を訂正してから、SMIT の「**Problem Determination Tools (問題判別ツール)**」>「**Recover From PowerHA SystemMirror Script Failure (スクリプト障害からの回復)**」パネルを使用して、`clruncmd` コマンドを発行し、クラスター・マネージャーにシグナルを送信してクラスター処理を再開します。
- クラスター検証ユーティリティを実行して、クラスター・リソースを検証します。

## 高可用性アプリケーションが失敗する

このトピックでは、高可用性アプリケーションが失敗する状態について説明します。

## 問題

リソース・グループが UNMANAGED 状態にあったクラスター・サービスの停止後にユーザーが手動で停止したアプリケーションは、ノードの再統合により再始動しません。

## 解決方法

`/usr/es/sbin/cluster/server.status` ファイル内の関連アプリケーション・エントリーが、ノードの再統合前に削除されたかどうかを検査します。

`/usr/es/sbin/cluster/server.status` ファイルのアプリケーション・エントリーは、ノードですでに実行されているすべてのアプリケーションをリストするため、PowerHA SystemMirror は `server.status` ファイル内のエントリーのアプリケーションを再始動しません。

再統合前に、関連のアプリケーション `server.status` エントリーを削除すると、高可用性アプリケーションが実行していないこと、およびノードで再始動する必要があることを、PowerHA SystemMirror が認識できます。

## AIX でのボリューム・グループ・クォーラム損失エラーによって PowerHA SystemMirror 選択フォールオーバーがトリガーされない

このトピックでは、PowerHA SystemMirror 選択フォールオーバーについて説明します。

## 問題

PowerHA SystemMirror は、ボリューム・グループのクォーラムの損失が発生したときに、影響を受けるリソース・グループを選択して別のクラスター・ノードに移動することができません。

## 解決方法

クラスター・ノード上のリソース・グループに属するボリューム・グループのクォーラムを損失すると、システムは、そのノードの AIX エラー・ログ・ファイルに LVM\_SA\_QUORCLOSE エラーが表示されたかどうかを検査し、影響を受けるリソース・グループを選択して移動するようにクラスター・マネージャーに通知します。PowerHA SystemMirror がこのエラー通知メソッドを使用するのは、ミラーリングされたボリューム・グループでクォーラムが使用可能になっている場合のみです。

フォールオーバーが発生しない場合は、AIX エラー・ログに LVM\_SA\_QUORCLOSE エラーが表示されたかどうかを検査します。AIX エラー・ログ・バッファがいっぱいになると、新規エントリはバッファ・スペースが使用可能になるまで破棄され、エラー・ログ・エントリによってこの問題が通知されません。この問題を解決するには、デバイス・ドライバーの AIX エラー・ログ内部バッファのサイズを増やしてください。

## グループ・サービスが **GS\_DOM\_MERGE\_ER** メッセージを送信する

このトピックでは、グループ・サービスのマージ・メッセージについて説明します。

## 問題

グループ・サービスのマージ・メッセージが表示され、このメッセージを受信したノードが自動的にシャットダウンします。GS\_DOM\_MERGE\_ER エラー・ログ・エントリのほか、グループ・サービス・デーモンのログ・ファイルにも以下のメッセージが出力されます。

```
"A better domain XXX has been discovered, or domain master requested to dissolve the domain."
```

ノードでクラスターとの通信が失われ、通信を再確立するときには、グループ・サービスのマージ・メッセージが送信されます。

## 解決方法

欠落しているノードおよびそのリソースの状態を判別する (そして、クラスター再結合時のデータの相違を回避する) のは困難なため、ノードをシャットダウンし、そのリソースのテークオーバーを正常に完了します。

例えば、クラスター・ノードが他のノードと通信できなくなった場合、そのノードが引き続きプロセス・テーブルを処理していても、他のノードが「欠落している」ノードからキープアライブ・メッセージを受信しなくなるため、他のノードは「欠落している」ノードで障害が発生したと判断します。残りのノードは、必要なイベントを処理し、ディスク、IP アドレスなどのリソースを「欠落している」ノードから獲得します。このリソースのテークオーバーにより、複数接続されたディスクがリセットを受信し、「欠落している」ノードから解放し、IP アドレス・テークオーバー・スクリプトを始動します。

ディスクがテークオーバー・ノードによって獲得されると (またはディスクが獲得され、アプリケーションが実行された後)、「欠落している」ノードはプロセス・テーブルを完了し (またはアプリケーションの問題をクリアし)、キープアライブ・メッセージを再送してクラスターを再結合しようとします。ディスクおよび IP アドレスがテークオーバーされているため、ネットワークで IP アドレスが重複し、ディスクのデータ・バスで余分なトラフィックが発生する場合があります。

ノードが「欠落した」理由が判別していないため、問題が今後も繰り返し発生する可能性がある判断し、ノードだけでなくクラスターやアプリケーションのダウン時間が長くなる場合があります。このため、最高のクラスター可用性を確保するために、GS のマージ・メッセージがすべての「欠落している」クラスター・ノードに送信され、ノードの分離を識別し、リソースを正常にテークオーバーできるようにし、テークオーバー・ノードと再結合する「欠落している」ノードの両方がディスクに書き込もうとした場合でもデータが破壊されないようにします。また、IP アドレスが同じノードがネットワークに 2 つ存在する場合は、トランザクションが失われ、アプリケーションがハングする場合があります。

クラスターが区画化されている場合は、各側の区画のノードがこれを検出し、反対側の区画のノードの **node\_down** を実行します。この実行中または通信が回復された後で、パーティションの両側がノードがクラスターのメンバーとして残すパーティションについて一致しない場合、作動したままにする区画が決定され、他の区画は、他の区画のノードからの GA マージ、または GS マージをノード自体に送信するノードによってシャットダウンされます。

3 つ以上のノードから構成されるクラスターでは、最も多くのノードが残っている区画が判断され、その区画が残されます。各区画のノード数が等しい場合は (2 ノード・クラスターの場合は常にこの状態になります)、起動しておくノードはノード番号によって決定されます (クラスターで最も低いノード番号のものが残ります。一般にアルファベット順で先頭となるノードです)。

グループ・サービス・ドメインのマージ・メッセージには、ノード分離の問題が処理され、リソースの可用性ができるだけ高くなるように保持されたことが示され、後でこの問題およびその原因を調査できます。ドメイン・マージが発生すると、グループ・サービスおよびクラスター・マネージャーは終了します。

**clstrmgr.debug** ファイルに、以下のエラーが記録されます。

```
"announcementCb: GRPSVCS announcement code=n; exiting"  
"CHECK FOR FAILURE OF RSCT SUBSYSTEMS (topsvcs or grpsvcs)"
```

## **cfgmgr** コマンドを実行するとクラスター内で不要な動作が行われる

このトピックでは、**cfgmgr** コマンドと、このコマンドの実行時にクラスター内で不要な動作が行われる状態について説明します。

### 問題

「**Configure Devices Added After IPL (IPL 後に追加したデバイスの構成)**」のような SMIT コマンドは、**cfgmgr** コマンドを使用します。場合によっては、このコマンドによってクラスターで不要な動作が発生する場合があります。例えば、ネットワーク・インターフェース・スワップが発生すると、**cfgmgr** コマンドはネットワーク・インターフェースを再度スワップするため、クラスター・マネージャーに障害が発生します。

### 解決方法

**rc.net** の変更について「インストール・ガイド」を参照し、この問題を回避してください。この手法は IP アドレス・テークオーバーの場合に限らず常に使用できますが、テークオーバー全体の時間が追加されるため、お勧めしません。

### 関連情報:

PowerHA SystemMirror インストール・ガイド

## **rmdev** 「*device busy* (デバイスが使用中)」エラーのために、ネットワーク・インターフェース・スワップが失敗する

このトピックでは、**rmdev** 「*device busy* (デバイスが使用中)」エラーが原因でネットワーク・インターフェース・スワップが失敗したときの問題と解決策について説明します。

## 問題

**rmdev** 「*device busy* (デバイスが使用中)」エラーのために、ネットワーク・インターフェース・スワップが失敗する。例えば、**/var/hacmp/log/hacmp.out** には以下のようなメッセージが示されます。

```
Method error (/etc/methods/ucfgdevice):  
0514-062 Cannot perform the requested function because the specified device is busy.
```

## 解決方法

システムで以下のアプリケーションが実行されているかどうかを検査します。これらのアプリケーションはデバイスを使用中として保持する場合があります。

### • SNA

以下のコマンドを使用して SNA が実行されているかどうかを調べます。

```
lssrc -g sna
```

以下のコマンドを使用して、SNA を停止します。

```
stopsrc -g sna
```

停止しない場合は、以下のコマンドを使用します。

```
stopsrc -f -s sna
```

停止しない場合は、以下のコマンドを使用します。

```
/usr/bin/sna -stop sna -t forced
```

停止しない場合は、以下のコマンドを使用します。

```
/usr/bin/sna -stop sna -t cancel
```

### • Netview / Netmon

**sysmond** デーモンが **-H** フラグで開始されたことを確認します。この場合、SM/6000 が状態を読み取るたびに、ネットワーク・インターフェースがオープンおよびクローズします。さらに、ハードウェア・アドレスのスイッチング前の **ifconfig** のデタッチ後に **rmdev** コマンドを実行すると **cl\_swap\_HW\_address** スクリプトが成功します。

以下のコマンドを使用して、すべての Netview デーモンを停止します。

```
/usr/0V/bin/nv6000_smit stopdaemons
```

### • IPX

以下のコマンドを使用して IPX が実行されているかどうかを調べます。

```
ps -ef |grep npsd  
ps -ef |grep sapd
```

以下のコマンドを使用して、IPX を停止します。

```
/usr/lpp/netware/bin/stopnps
```

### • NetBIOS

以下のコマンドを使用して NetBIOS が実行されているかどうかを調べます。

```
ps -ef |grep netbios
```

以下のコマンドを使用し、NetBIOS を停止して NetBIOS ストリームをアンロードします。

```
mcsadm stop; mcs0 unload
```

- 適用可能な場合は (ファイルが存在する場合は)、以下のように各種のストリームをアンロードします。

```
cd /etc
strload -uf /etc/dlpi.conf
strload -uf /etc/pse.conf
strload -uf /etc/netware.conf
strload -uf /etc/xtiso.conf
```

- デバイスを使用中として保持するカスタマー・アプリケーションもあります。 共用アプリケーションが正常に停止していることを確認します。

## クライアントの問題

このセクションでは、PowerHA SystemMirror クライアントの潜在的な問題について説明します。

### ネットワーク・インターフェース・スワップが原因でクライアント接続問題が発生する

このトピックでは、ネットワーク・インターフェース・スワップが原因でクライアント接続問題が発生する状態について説明します。

#### 問題

クライアントがクラスターに接続できません。 クライアント・ノードの ARP キャッシュには、フォールオーバー・ノードではなく、障害が発生したノードのアドレスが格納されたままになっています。

#### 解決方法

クラスター・ノードからクライアントに **ping** コマンドを発行して、クライアントの ARP キャッシュを更新します。 このコマンドの引数には、必ずクライアント名を指定してください。 **ping** コマンドは、クライアントが **clinfoES** を実行していない場合でも、クライアントの ARP キャッシュを更新します。アプリケーションのイベント前処理スクリプトまたはイベント後処理スクリプトに **ping** コマンドの呼び出しを追加して、特定のクライアント上でこの更新を自動化します。

### クライアントがアプリケーションにアクセスできない

このトピックでは、クライアントがアプリケーションにアクセスできない状態について説明します。

#### 問題

**SNMP** 処理は失敗しました。

#### 解決方法

**SNMP** がクラスター・ノードの IP ラベルまたはアドレスを含むことを確認できなかったノードの **/etc/hosts** ファイルを検査します。 『クライアントがクラスターを見つけられない』も参照してください。

#### 関連資料:

85 ページの『クライアントがクラスターを見つけられない』

このトピックでは、クライアント上で実行されている **clstat** ユーティリティーがクラスターを検出できない状態について説明します。

## クライアントがクラスターを見つけられない

このトピックでは、クライアント上で実行されている **clstat** ユーティリティーがクラスターを検出できない状態について説明します。

### 問題

クライアント上で実行されている **clstat** ユーティリティーが、クラスターを検出できません。 **clinfoES** デーモンは、そのデーモンが通信できる SNMP 処理を見つけられなかったため、クライアント (**clstat** など) に作成したデータ構成を適切に管理していません。これは、**clinfoES** は、クラスター状態情報を SNMP から獲得するため、SNMP がこのデーモンと通信できなければ PowerHA SystemMirror MIB にデータを設定できないからです。その結果として、さまざまな断続的な問題が SNMP と **clinfoES** 間で発生する場合があります。

### 解決方法

自動訂正アクションを有効にして検証を実行することにより、更新されたクライアント・ベースの **clhosts** ファイルを作成します。これにより、サーバー・ノードに **clhosts.client** ファイルが作成されます。このファイルをクライアントの **/usr/es/sbin/cluster/etc/** ディレクトリーにコピーして、名前を **clhosts** に変更します。**clinfoES** デーモンはこのファイル内のアドレスを使用して、PowerHA SystemMirror サーバー上で実行される SNMP 処理との通信を試みます。

また、SNMP 処理が実行されているノードや、**clstat** や他の **clinfo** API プログラムに問題があるノード上の **/etc/hosts** ファイルを検査します。

## Clinfo が実行中であるように見えない

このトピックでは、**clinfo** が実行されていないようになっている状態について説明します。

### 問題

**clinfoES** を始動したクラスター・ノードのサービスおよびブート・アドレスが、クライアント・ベースの **clhosts** ファイルに存在しません。

### 解決方法

自動訂正アクションを有効にして検証を実行することにより、更新されたクライアント・ベースの **clhosts** ファイルを作成します。これにより、サーバー・ノードに **clhosts.client** ファイルが作成されます。このファイルをクライアントの **/usr/es/sbin/cluster/etc/** ディレクトリーにコピーして、名前を **clhosts** に変更します。その後で、**clstat** コマンドを実行します。

## ノードがダウンしていることを clinfo が報告しない

このトピックでは、ノードがダウンしていても、ノードが稼働中であると SNMP デーモンおよび **clinfoES** が報告する状態について説明します。

### 問題

ノードがダウンしていても、SNMP デーモンおよび **clinfoES** はノードが稼働中であると報告します。すべてのノードのインターフェースがダウンしているとリストされます。

## 解決方法

1 つ以上のノードがアクティブで、他のノードがクラスターに結合しようとする場合、現在のクラスター・ノードは結合するノードが稼働中であるという情報を SNMP デーモンに送信します。何らかの理由で、ノードがクラスターとの結合に失敗した場合、**clinfoES** は、ノードがダウンしているということを報告する別のメッセージを SNMP デーモンに送信しません。

クラスターの状態情報を修正するには、「PowerHA SystemMirror クラスター・サービス」SMIT パネルのオプションを使用して、SNMP デーモンを再始動します。

## その他の問題

以下のトピックでは、特定のカテゴリに分類されない潜在的な PowerHA SystemMirror の問題について説明します。

PowerHA SystemMirror のリソース・グループの移動を調査し、**rg\_move** イベントが発生した理由を知るには、必ず **/var/hacmp/log/hacmp.out** ファイルを調べてください。特に PowerHA SystemMirror において、リソース・グループの処理方法やフォールオーバー環境の優先順位付けが最近変更されたので **hacmp.out** ファイルおよびイベント要約が、リソース・グループのアクティビティと場所を追跡する際にさらに重要になりました。さらに、リソース・グループの並列処理の場合、**hacmp.out** ファイルはクラスター・ヒストリー・ログまたは **clstrmgr.debug** ファイルに示されない詳細を報告します。テークオーバー・アクティビティの後にリソース・グループの移動を調査する場合は、必ず早い段階でこのログを検査してください。

## **/var/hacmp/log/hacmp.out** に対して **tail -f** コマンドを実行するとき出力が限定される場合

このトピックでは、**/var/hacmp/log/hacmp.out** ファイルで出力が限定される状態について説明します。

### 問題

スクリプト始動メッセージのみが **/var/hacmp/log/hacmp.out** ファイルに表示されます。メッセージに指定されたスクリプトは実行不可能であるか、DEBUG レベルが「**low (低)**」に設定されます。

## 解決方法

**chmod** コマンドを使用して、実行可能なアクセス権をスクリプトに追加し、DEBUG レベルが「**high (高)**」に設定されているかどうかを確認します。

## クラスターの検証で、不必要なメッセージが出力される

このトピックでは、自動エラー通知を構成したかどうかに関係なく、クラスターの検証でメッセージが返される状態について説明します。

### 問題

自動エラー通知を構成したかどうかにかかわらず、以下のメッセージが出力されます。

```
"Remember to redo automatic error notification if configuration has changed."
```

## 解決方法

自動エラー通知を構成していない場合は、このメッセージを無視します。

## config\_too\_long メッセージが表示される

このトピックでは、config\_too\_long メッセージが表示されるシナリオについて説明します。

このメッセージは、指定されたタイムアウト期間内にクラスター・イベントが完了しないたびに出されます。

4.5 より前のバージョンでは、タイムアウト期間はすべてのクラスター・イベントについて固定され、デフォルトでは 360 秒に設定されていました。 **node\_up** または **node\_down** イベントなどのクラスター・イベントが、360 秒より長く続いた場合、30 秒ごとに PowerHA SystemMirror は、**hacmp.out** ファイルにログ記録された **config\_too\_long** 警告メッセージを発行しました。

PowerHA SystemMirror では、クラスター・イベントに許容される時間枠をカスタマイズして、PowerHA SystemMirror がシステム警告を出す前にクラスター・イベントを完了させることができます。

**hacmp.out** の「Event Start (イベントの開始)」にこのメッセージが出力される場合は、以下のように出力されます。

```
config_too_long $sec $event_name $argument<
```

- \$event\_name は、失敗した再構成イベントです。
- \$argument は、イベントが使用するパラメーターです。
- \$sec はメッセージの送信までの秒数です。

PowerHA SystemMirror 4.5 より前のバージョンでは、アクションが実行されるまで、30 秒ごとに **hacmp.out** ファイルに **config\_too\_long** メッセージが付加され続けます。

バージョン 4.5 で始動し、指定したイベント期間内に完了しない各クラスター・イベントの場合は、**config\_too\_long** メッセージが **hacmp.out** ファイルにログ記録され、以下の形式にしたがってコンソールに送信されます。

- 最初の 5 つの **config\_too\_long** メッセージは 30 秒間隔で **hacmp.out** ファイルに表示されます。
- 次の 5 つのメッセージは、間隔が 1 時間に達するまで直前の 2 倍の間隔で表示されます。
- これらのメッセージは、イベントが完了するまで、あるいはイベントがそのノードで終了するまで毎時間ログに記録されます。

このメッセージは、以下の問題に対する応答として出力されます。

### 問題

スクリプトが実行しているアクティビティーに、完了までに指定の時間より長い時間がかかります (この状態は、多くのディスクや複雑なスクリプトが関係するイベントなどで発生します)。

### 解決方法

- 実行に長い時間がかかっているプロセスを判別し、可能な場合はそのプロセスを訂正するか簡素化します。
- **config\_too\_long** を呼び出すまでの待機時間を増やします。

SMIT の「Change/Show Time Until Warning (警告までの時間の変更/表示)」パネルを使用すれば、「Event Duration Time (イベント期間)」をカスタマイズできます。このパネルにアクセスするには、「Extended Configuration (拡張構成)」>「Extended Event Configuration (拡張イベント構成)」SMIT パネルを選択します。

## 問題

再開を実行するまでコマンドがハングし、イベント・スクリプトが待機します。 その場合、AIX プロセス・テーブル (**ps -ef**) でコマンドを表示できます。 可能性が非常に高いのは、**config\_too\_long** スクリプト出力の前の **/var/hacmp/log/hacmp.out** ファイルの最後のコマンドです。

## 解決方法

**hung** コマンドを終了する必要がある場合があります。

## 問題

アプリケーション・コントローラーの始動スクリプト用にフォアグラウンド始動プロセスが指定されましたが、そのスクリプトが終了しません。

注: この問題は、PowerHA SystemMirror 7.1.1 以降を使用する場合にのみ発生します。

## 解決方法

始動スクリプトが正しく機能しているかどうかを調べます。スクリプトのハングの可能性がある場合は、フォアグラウンド始動ではなく、バックグラウンド始動オプションを始動モニターと組み合わせて使用することを検討してください。

### 関連資料:

93 ページの『動的再構成でロックが設定される』

このトピックでは、動的再構成の試行時にエラー・メッセージが生成される状態について説明します。

### 関連情報:

警告が出されるまでのイベント期間の調整

## コンソールに **SNMP** メッセージが表示される

このトピックでは、**/etc/syslogd** ファイルが出力を誤ったロケーションに送信する状態について説明します。

## 問題

**/etc/syslogd** ファイルが、**daemon.notice** 出力を **/dev/console** に送信するよう変更されました。

## 解決方法

**daemon.notice** 出力を **/usr/tmp/snmpd.log** にリダイレクトするように **/etc/syslogd** ファイルを編集します。 **snmpd.log** ファイルは、メッセージのロギング用のデフォルト・ロケーションです。

## 計画外のシステム・リブートによりフォールオーバーの試行が失敗する

このトピックでは、計画外のシステム・リブートが行われたときにフォールオーバーしようとしてもフォールオーバーできない可能性がある状況について説明します。

## 問題

システムをリブートした後にクラスター・ノードがフォールオーバーしません。

## 解決方法

計画外のシステム・リポートによるクラスター環境でのフォールオーバーの障害を回避するには、クラスターのすべてのノードで、「Change/Show Characteristics of Operating System (オペレーティング・システムの特性の変更/表示)」SMIT パネルの「**Automatically REBOOT a system after a crash** (クラッシュ後にシステムを自動リブート)」フィールドで、「**false**」に設定するか、通常の操作時に、IBM System p キーをセキュア・モードにしておく必要があります。

いずれの方法も **shutdown** コマンドが不用意に発行された場合に、システムがリブートするのを回避します。これらの方法が 1 つでも適切でない場合、計画外のリポートが発生すると、リブート中ノードのディスクに対するアクティビティにより、他のノードが正常にディスクを獲得できなくなる場合があります。

**削除されたオブジェクトまたは関係のないオブジェクトが NetView マップに表示される**  
このトピックでは、NetView マップについて、および削除されたオブジェクトや、関係のないオブジェクトが表示された場合の対処方法について説明します。

## 問題

NetView マップに、すでに削除したオブジェクト・シンボルまたは余分なオブジェクト・シンボルが出力されます。

## 解決方法

### NetView データベースの構築

NetView データベースを再構築するために、NetView サーバーで以下の手順を実行します。

1. すべての NetView デーモンを停止します。  
`/usr/OV/bin/ovstop -a`
2. NetView サーバーからデータベースを削除します。  
`rm -rf /usr/OV/database/*`
3. 以下のように、NetView オブジェクト・データベースを始動します。  
`/usr/OV/bin/ovstart ovwdb`
4. 「NetView/HAView」フィールドを復元します。  
`/usr/OV/bin/ovw -fields`
5. すべての NetView デーモンを始動します。  
`/usr/OV/bin/ovstart -a`

## F1 キーを押しても SMIT パネルにヘルプが表示されない

このトピックでは、F1 キーを押しても SMIT パネルにヘルプが表示されないシナリオについて説明します。

## 問題

SMIT パネルで F1 キーを押してもヘルプは表示されません。

## 解決方法

ヘルプが表示されるのは、PowerHA SystemMirror でサポートされるいずれかの言語に LANG 変数を設定し、関連する PowerHA SystemMirror メッセージ・カタログをインストールしている場合のみです。PowerHA SystemMirror でサポートされる言語は以下のとおりです。

- en\_US
- ja\_JP

インストール済みロケール (bsl LPP) をリストするには、以下のように入力します。

```
locale -a
```

アクティブなロケールをリストするには、以下のように入力します。

```
locale
```

LANG 環境変数がアクティブ・ロケールを決定するため、LANG=en\_US の場合のロケールは en\_US となります。

## イベント要約表示が大きくなりすぎる

このトピックでは、/usr/es/sbin/cluster/cl\_event\_summary.txt ファイル (イベント要約表示) が大きくなりすぎる状態について説明します。

## 問題

PowerHA SystemMirror では、イベント要約は、**hacmp.out** ファイルから呼び出され、**cl\_event\_summary.txt** ファイルに保管されます。このファイルは hacmp.out サイクルに従って累積され続け、自動的に切り捨てられたり置換されたりすることはありません。その結果、このファイルが大きくなりすぎて /usr ディレクトリーがいっぱいになります。

## 解決方法

SMIT の「**Problem Determination Tools (問題診断ツール)**」>「**PowerHA SystemMirror Log Viewing and Management (PowerHA SystemMirror ログの表示および管理)**」>「**View/Save/Remove PowerHA SystemMirror Event Summaries (PowerHA SystemMirror イベント要約の表示/保存/除去)**」>「**Remove Event Summary History (イベント要約ヒストリーの除去)**」オプションを使用して、定期的にイベント要約をクリアしてください。

「**View event summaries (イベント要約の表示)**」でリソース・グループ情報が予想どおりには表示されない

このトピックでは、「View event summaries (イベント要約の表示)」でリソース・グループ情報が予想どおりには表示されない状況について説明します。

## 問題

PowerHA SystemMirror では、イベント要約は **hacmp.out** ファイルから呼び出され、SMIT の「**Problem Determination Tools (問題診断ツール)**」>「**PowerHA SystemMirror Log Viewing and Management (PowerHA SystemMirror ログの表示および管理)**」>「**View/Save/Delete Event Summaries (イベント要約の表示/保管/除去)**」>「**View Event Summaries (イベント要約の表示)**」オプションを使用して表示できます。この表示の末尾には、リソース・グループの状況およびロケーション情報が含まれます。リソース・グループ情報は、**clRGinfo** により収集され、「**View Event Summaries (イ**

ベント要約の表示)」 オプションの実行時にクラスターが実行されていない場合には、余分な時間がかかることがあります。

## 解決方法

**clRGinfo** は、クラスター実行時にはより迅速にリソース・グループ情報を表示します。

クラスターが実行中でない場合は、リソース・グループ情報が表示されるまでに数分かかります。

## アプリケーション・モニターの問題

アプリケーション・モニターを実行している場合に、モニターの状態や構成を調べる必要が生じることがあります。発生する可能性のあるいくつかの問題と、その診断方法および処置を以下に示します。

### 問題

アプリケーション・モニターの状態の検査。状況によっては、アプリケーション・モニターが現在実行中かどうかははっきりしないことがあります。アプリケーション・モニターの状態をチェックするには、次のコマンドを実行します。

```
ps -ef | grep <application controller name> | grep clappmond
```

アプリケーションがモニターされている場合、このコマンドにより、長い行の冗長な出力が生成されます。

出力がない場合、アプリケーションはモニターされていません。

## 解決方法

アプリケーション・モニターが稼働していない場合、以下のような多くの理由が考えられます。

- アプリケーション・コントローラー用にモニターが構成されていません。
- 安定化間隔が完了していないため、モニターがまだ開始していません。
- モニターが中断状態になっています。
- モニターが正しく構成されていません。
- エラーが発生しました。

何か異常があると結論づける前に、モニターが構成されているか、安定化間隔が終了しているか、モニターが停止状態になっていないかをチェックしてください。

明らかに障害が発生している場合は、**SMIT** でモニターの元の構成を再度検査して、必要に応じて再構成します。

## 問題 2

アプリケーション・モニターが、指定した障害処置を実行しません。アプリケーションに明らかに障害が発生しているにもかかわらず、指定された障害処置が行われません。

## 解決方法 2

再始動間隔を検査します。設定が短すぎる場合、再始動カウンターがすぐにゼロにリセットされてしまうため、再始動の試行が限りなく続けられ、他の処置が実行できない可能性があります。

### 問題 3

アプリケーション・モニターで、アプリケーションが正しく機能していることが示されないことがあります。

#### 解決方法 3

- モニターがあらゆる場合に正しい終了コードを返すように作成されていることを確認します。戻り値は、アプリケーションが良好に機能している場合はゼロになり、アプリケーションに障害が発生している場合はゼロ以外の値にならなければなりません。
- 終了コードがアプリケーションの状態と整合することを確認するために、エラー・パスも含め、コードを通過する可能性があるすべてのパスを検査します。

### 問題 4

どのような場合にモニターが稼働するのか判別できません。

#### 解決方法 4

モニターによって作成されたログ・ファイルを検査します。モニターは、メッセージを標準出力 **stdout** ファイルに出力することによって、メッセージをログに記録することができます。長期モニターの場合、出力は `/var/hacmp/log/clappmond.application monitor name.resource group name.monitor.log` ファイルに保管されます。始動モニターの場合、この出力は `/var/hacmp/log/clappmond.application server name.resource group name.monitor.log` ファイルに保管されます。モニター・ログ・ファイルは、アプリケーション・モニターが実行されるたびに上書きされます。

### クラスター・ディスク交換プロセスが失敗する

このトピックでは、クラスター・ディスク交換プロセスが失敗したときの対処方法について説明します。

#### 問題

ディスク交換プロセスが、**replacepv** コマンドの実行中に失敗しました。

#### 解決方法

`/tmp/replacepv` ディレクトリーを削除し、交換プロセスを再度試みます。

別のディスクでこのプロセスを実行してみることもできます。

### rg\_move イベントが一度にいくつかのリソース・グループを処理する

このトピックでは、**rg\_move** イベントが一度に複数のリソース・グループを処理する状態について説明します。

#### 問題

**hacmp.out** では、**rg\_move** イベントが複数の非コンカレント・リソース・グループを一度の操作で処理します。

#### 解決方法

これは予想された動作です。依存関係を持つクラスターでは、PowerHA SystemMirror はすべてのリソース・グループを **node\_up** イベント時に **rg\_move** イベントを介して処理します。単一の **rg\_move** イベント

ト時には、PowerHA SystemMirror は 1 つのイベント内で複数の非コンカレント・リソース・グループを処理します。

#### 関連資料:

33 ページの『従属リソース・グループまたはサイトを有するクラスター内の処理』  
従属グループまたはサイトを設定したクラスターのリソース・グループは、動的イベント変換で処理されません。

## ファイルシステムがアンマウントされない

このトピックでは、ファイルシステムがアンマウントされないシナリオについて説明します。

### 問題

リソース・グループをオフラインにするオプションを指定してクラスター・サービスを停止するなどのイベント時に、ファイル・システムは正常にアンマウントされません。

### 解決方法

リソース・グループをオフラインにするオプションを指定してクラスター・サービスを停止するとファイル・システムがアンマウントに失敗する一般的な理由の 1 つに、ファイル・システムが使用中である場合があります。ファイル・システムを正常にアンマウントするには、アンマウント時に、プロセスまたはユーザーが、そのファイル・システムにアクセスしてはなりません。ユーザーまたはプロセスが保持していると、ファイル・システムは「busy (使用中)」のまま、アンマウントされません。

削除されたファイルが開いたままになっている場合、同じ問題が発生する可能性があります。

アプリケーション停止スクリプトには、共用ファイル・システムが使用中でなく、削除されておらず、オープンされた状態でもないことを確認する検査も含める必要があります。この確認には、**fuser** コマンドを使用します。スクリプトで **fuser** コマンドを使用し、問題のファイル・システムにアクセスしているプロセスまたはユーザーを確認します。これらのプロセスの PID を取得して **kill** することができます。ファイル・システムが解放されれば、アンマウントすることができます。

このコマンドの詳細については、AIX のマニュアル・ページを参照してください。

## 動的再構成でロックが設定される

このトピックでは、動的再構成の試行時にエラー・メッセージが生成される状態について説明します。

### 問題

他の動的再構成 (DARE) 操作が処理中の場合、または以前の DARE 操作が適切に完了しなかった場合には、DARE 操作の試行時に、DARE ロックに関するエラー・メッセージが生成される場合があります。

エラー・メッセージは、DARE 操作が処理中でない場合に、ロックをクリアする処置を取る必要があることを示します。この場合の「In process (処理中)」は、発行された可能性がある別の DARE 操作を示しますが、正しく完了しなかった以前の DARE 操作も示します。

### 解決方法

最初のステップは、クラスター・ノードの **/var/hacmp/log/hacmp.out** ログを調べて、以前の DARE が失敗した原因を判別します。イベント・スクリプトの操作が長くなり完了できない場合に、**config\_too\_long**

エントリーが **hacmp.out** に表示されることがよくあります。 **hacmp.ou** が、一部のエラーが原因でスクリプトが完了されなかったことを示す場合は、この問題を修正し、イベントの完了に必要な残りの手順を手動で完了します。

PowerHA SystemMirror SMIT の「**Problem Determination Tools (問題判別ツール)**」 > 「**Recover from PowerHA SystemMirror Script Failure (スクリプト障害からの回復)**」 オプションを実行します。これによって、クラスターのノードが次の完全イベント状態になります。

PowerHA SystemMirror SMIT の「**Recover from PowerHA SystemMirror Script Failure (PowerHA SystemMirror スクリプト障害の回復)**」でクリアできなかった場合は、PowerHA SystemMirror SMIT オプションの「**Problem Determination Tools (問題判別ツール)**」 > 「**Release Locks Set by Dynamic Configuration (動的構成によって設定されたロックの解除)**」を選択すると DARE ロックをクリアできます。

## WPAR 対応のリソース・グループに関する問題

このトピックでは、WPAR 対応のリソース・グループに関して発生する可能性のある問題について説明します。

### 問題

リソース・グループが、特定のノード上の WPAR でオンラインになりません。

### 解決方法

1. 問題のノードが WPAR 対応であることを確認します。 WPAR 対応の AIX ノードには **bos.wpars** ファイルセットがインストールされていなければなりません。 ノードが WPAR 対応でない場合、リソース・グループは WPAR で実行されません。 このファイルセットがインストールされているかどうかをチェックするには、次のコマンドを発行します。

```
lsllp -L "bos.wpars"
```

2. 指定のノードで、WPAR 対応のリソース・グループと同じ名前を持つ WPAR があることを確認します。 この確認には、**lswpar <resource group name>** コマンドを使用します。 指定の名前を持つ WPAR がない場合は、**mkwpar** コマンドで作成します。 WPAR を作成した後、WPAR 対応のリソース・グループに関連付けられたすべてのユーザー定義スクリプトが WPAR 内にアクセスできることを確認してください。
3. ノード上のファイルシステムがいっぱいになっていないことを確認します。 いっぱいになっている場合は、いくつかのファイルを外部ストレージに移動して、ディスク・スペースを一部空けてください。
4. 対応する WPAR で **rsh** サービスが使用可能になっていることを確認します。 これは次のようにして実行できます。

- WPAR で次のコマンドを発行して、**inetd** サービスが WPAR で実行されていることを確認します。

```
lssrc -s inetd
```

**inetd** サービスがアクティブではない場合は、**startsrc** コマンドで **inetd** サービスを開始します。

- WPAR 内の **/etc/inetd.conf** ファイルに **rsh** が、認識されているサービスとしてリストされていることを確認します。

## SNMP ベースの状況コマンドのトラブルシューティング

このセクションでは、SNMP ベースの状況コマンド (**clstat**、**cldump**、および **cldisp**) の失敗の原因になる可能性がある問題について説明します。

Simple Network Management Protocol (SNMP) は、管理情報ベース (MIB) と呼ばれる、状況変数と構成変数のデータベースへのアクセスを提供します。基本 AIX で提供される SNMP サブシステムは、MIB 全体のサブセットを提供するだけでなく、MIB のその他の部分へのアクセスを提供するピア・デーモンとも連携することができます。SystemMirror クラスタ・マネージャー・デーモンは、そのようなピアとして動作し、MIB 内の SystemMirror 固有の変数へのアクセスを提供します。

SNMP やそれに依存するユーティリティーで問題が起きたときは、まず基本 SNMP 構成が機能していることを確認し、次に SystemMirror 固有機能の検査に進みます。

**snmpinfo** コマンドを使用して、SNMP の基本機能を検査できます。MIB のデフォルト部分を表示するには、**snmpinfo -m dump** コマンドを使用します。このコマンドで出力が生成されない場合は、SNMP の基本セットアップと **snmpd** サブシステム自体に問題があります。**snmpd** サブシステムが稼働していることを確認し、以下のセクションの手順に従って、基本的な **snmpinfo** コマンドが機能していることを確認してください。

基本機能が稼働していることを確認した後、次のコマンドを使用して、MIB の SystemMirror 固有の部分に照会を行うことができます。

```
snmpinfo -m dump -v -o /usr/sbin/cluster/hacmp.defs risc6000clsmuxpd
```

上記のコマンドで出力が生成されない (しかも、**snmpinfo -m dump** では生成される) 場合は、MIB の SystemMirror 部分に固有な問題です。下記の手順に従って、SystemMirror 固有のコンポーネントの状況と構成を確認してください。

## 問題

AIX オペレーティング・システムに同梱されている **snmpdv3.conf** ファイルには、2 つの一般的な問題があります。それらは、以下のとおりです。

- SNMP 管理情報ベース (MIB) の **internet** 部分へのアクセスがコメント化されています。
- PowerHA SystemMirror 7.1.2 では、IPv6 ループバック・アドレスの **COMMUNITY** エントリーがありません。

これらの問題を解決するには、『一般的な SNMP の問題のトラブルシューティング』の手順を実行します。ただし、最初の 2 つの問題を修正した後でも、他の問題によって、依然として SNMP 状況コマンドの正しい機能が阻害される場合があります。これらの問題を解決するには、97 ページの『SNMP の状況コマンドのトラブルシューティング』の手順を実行します。それでも状況コマンドが失敗する場合は、98 ページの『**snmpdv3.conf** ファイルのトラブルシューティング』セクションの手順を実行して、残りの問題を解決します。

## 解決方法

一般的な **SNMP** の問題のトラブルシューティング:

このトピックは、一般的な 2 つの問題の解決に役立ちます。通常、これらの問題の修正により、問題は解決し、他のセクションを読む必要がなくなる可能性があります。

1. SNMP 構成ファイルで、SNMP 管理情報ベース (MIB) の PowerHA 部分に対するアクセス許可があるかどうかを検査します。**/etc/snmpdv3.conf** ファイル内で **defaultView** エントリーを見つけます。

```
# grep defaultView /etc/snmpdv3.conf
#VACM_VIEW defaultView      internet                - included -
VACM_VIEW defaultView      1.3.6.1.4.1.2.2.1.1.0  - included -
VACM_VIEW defaultView      1.3.6.1.4.1.2.6.191.1.6 - included -
VACM_VIEW defaultView      snmpModules           - excluded -
VACM_VIEW defaultView      1.3.6.1.6.3.1.1.4     - included -
```

```
VACM_VIEW defaultView      1.3.6.1.6.3.1.1.5      - included -
VACM_VIEW defaultView      1.3.6.1.4.1.2.6.191     - excluded -
VACM_ACCESS group1 - - noAuthNoPriv SNMPv1 defaultView - defaultView -
```

AIX 7.1 以降、セキュリティー上の予防措置として、**snmpdv3.conf** ファイルは **internet** アクセスがコメント化されて出荷されています。上記の例は、未修正の構成ファイルを示しています。**internet** ディスクリプターがコメント化されており、これは、PowerHA 情報を含む MIB の大部分に対するアクセス権限がないことを意味します。(その他の **included** エントリーは、これ以外の MIB の限られた部分に対するアクセスを提供します。)AIX 7.1 以降のデフォルトでは、**snmpdv3.conf** ファイルを編集しない限り、PowerHA の SNMP ベースのコマンドは機能しません。PowerHA MIB へのアクセスを提供するには、次の 2 つの方法があります。

- **snmpdv3.conf** ファイルで、次の **internet** 行をアンコメントします。

```
VACM_VIEW defaultView internet - included -
```

これにより、MIB 全体にアクセスできるようになります。

- MIB 全体へのアクセス権を提供したくない場合は、**snmpdv3.conf** ファイルに次の行を追加します。これにより、PowerHA MIB だけにアクセスできるようになります。

```
VACM_VIEW defaultView risc6000clsmuxpd - included -
```

注: SNMP 構成ファイルを編集した後、以下のコマンドを使用して、**snmpd** をいったん停止して再始動してから、クラスター・マネージャーをリフレッシュする必要があります。

```
stopsrc -s snmpd
startsrc -s snmpd
refresh -s clstrmgrES
```

SNMP ベースの状況コマンドを再試行します。コマンドが機能する場合、このセクションの残りの部分を読む必要はありません。

2. PowerHA SystemMirror 7.1.2 またはそれ以降を使用している場合は、**clinfoES** および **snmpd** の構成ファイルに正しい IPv6 エントリーが存在するかどうかを検査します。PowerHA 7.1.2 では、IPv6 をサポートするために、**/usr/es/sbin/cluster/etc/clhosts** ファイルに 1 つのエントリーが追加されています。しかし、これに対応する必須エントリーが **/etc/snmpdv3.conf** ファイルに追加されていません。このため、**clstat** コマンドで再現性の低い問題が発生します。この問題に対処するには、次の 2 つの方法があります。

- IPv6 を使用する計画でない場合は、**/usr/es/sbin/cluster/etc/clhosts** ファイル内の該当する行をコメント化し、以下のコマンドを使用して **clinfoES** を再始動します。

```
# ::1      # PowerHA SystemMirror
stopsrc -s clinfoES
startsrc -s clinfoES
```

SNMP ベースの状況コマンドを再試行します。コマンドが機能する場合、このセクションの残りの部分を読む必要はありません。

- 将来 IPv6 を使用する計画の場合は、**/snmpdv3.conf** ファイルに次の行を追加します。

```
COMMUNITY public public noAuthNoPriv :: 0 -
```

別の (**public** 以外の) コミュニティーを使用している場合は、**public** という語をそのコミュニティの名前に置き換えます。

注: SNMP 構成ファイルを編集した後、以下のコマンドを使用して、**snmpd** をいったん停止して再始動してから、クラスター・マネージャーをリフレッシュする必要があります。

```
stopsrc -s snmpd
startsrc -s snmpd
refresh -s clstrmgrES
```

SNMP ベースの状況コマンドを再実行します。コマンドが機能する場合、次のセクションを読む必要はありません。

### SNMP の状況コマンドのトラブルシューティング:

このトピックは、一般的な問題を修正した後でも依然として SNMP ベースの状況コマンドの機能を阻害する可能性がある、その他の問題の解決に役立ちます。

1. 次のコマンドを実行して、**snmpd** が稼働しているかどうかを検査します。

```
lssrc -s snmpd
```

稼働していない場合は、次のコマンドを実行して、**snmpd** を始動します。

```
startsrc -s snmpd
```

2. 次のコマンドを実行して、クラスター・サービスが稼働しているかどうかを検査します。

```
lssrc -ls clstrmgrES | grep state (looking for a state of ST_STABLE)
```

稼働していない場合は、クラスター・サービスを開始します。クラスター・サービスが稼働していない場合は、どの SNMP 状況コマンドも機能しません。

3. **clstat** コマンドを使用している場合は、**/usr/es/sbin/cluster/etc/clhosts** ファイルが正しいかどうかを検査します。**clhosts** ファイルには、**clinfoES** デーモンが通信できる PowerHA ノードの IP アドレスのリストが入っている必要があります。(好ましいのは永続アドレスです。クラスター・ノードに属さないアドレスがファイルに入っている場合、さらなる問題の原因になる可能性があります。) あるシステムのファイルを編集したら、そのシステムで **clinfoES** を再始動する必要があります。

- クラスター・ノードでの操作

- デフォルトでは、**clhosts** ファイルに事前にローカル・ホスト・アドレスが入力されています。クラスター内のすべてのノードのエントリを追加することができ、これにより、**clstat** コマンドはクラスター・サービスがノード上で稼働している間、機能します。
- PowerHA SystemMirror 7.1.2 以降、デフォルトの **clhosts** ファイルに IPv6 ループバック・アドレスのエントリが追加されています。95 ページの『一般的な SNMP の問題のトラブルシューティング』セクションで説明されているように、この行をコメント化するか、または SNMP 構成ファイルに IPv6 ループバック・アドレスの行を追加することができます。

- クライアント・システムでの操作

- デフォルトでは、**clhosts** ファイルは空になっています。ユーザーは、クラスター・ノードのアドレスを追加する必要があります。

4. **clstat** コマンドを使用している場合は、次のコマンドを実行して、**clinfoES** が稼働しているかどうかを検査します。

```
lssrc -s clinfoES
```

稼働していない場合は、次のコマンドを実行して始動します。

```
startsrc -s clinfoES
```

ヒント: この問題を回避するために、クラスター・サービスを開始するときは、常に **clinfoES** を始動してください。

5. **snmpd** が **smux** ポートで **listen** を行っているかどうか、およびクラスター・マネージャーが接続されているかどうかを検査します。次の **netstat** コマンドを実行して、**smux** ポートを使用しているアクティブなソケットをリストします。

```
# netstat -Aa | grep smux
f1000e0002988bb8 tcp 0 *.smux *.* LISTEN
f1000e00029d8bb8 tcp4 0 0 loopback.smux loopback.32776 ESTABLISHED
f1000e00029d4bb8 tcp4 0 0 loopback.32776 loopback.smux ESTABLISHED
f1000e000323fbb8 tcp4 0 0 loopback.smux loopback.34266 ESTABLISHED
f1000e0001b86bb8 tcp4 0 0 loopback.34266 loopback.smux ESTABLISHED
```

**LISTEN** 状態にあるソケットが表示されない場合は、次のコマンドを使用して、**snmpd** をいったん停止してから始動します。

```
stopsrc -s snmpd; startsrc -s snmpd
```

6. **LISTEN** 状態の **smux** ソケットができれば、クラスター・マネージャーが所有するソケットの 1 つを持つ **ESTABLISHED** 状態のソケット・ペアを探します。**rmsock** コマンドを使用して、どのプロセスがソケットを所有しているかを知ることができます。**snmpd** を再始動したばかりである場合は、**smux** ポートに **LISTEN** ソケットが存在することを確認します。**ESTABLISHED** 状態の **smux** ソケットが 1 つも表示されない場合は、クラスター・マネージャーをリフレッシュするか (**refresh -s clstrmgrES**)、数分間待ちます。その後、**netstat -Aa** コマンドを再実行します。クラスター・マネージャーはサービスが開始されたとき、およびサービス開始後の数分ごとに、**snmpd** への接続を試みます。リフレッシュ・コマンドを実行すると、クラスター・マネージャーは即時に **snmpd** への接続を試みます。クラスター・マネージャーに対しては、**stopsrc** および **startsrc** を使用しないでください。
7. **rmsock** を使用して、**ESTABLISHED** 状態の **smux** ソケットの所有者を見つけます。**netstat** 出力の最初のフィールド (これはソケットのメモリー・アドレスです) を、**rmsock** への引数として使用します。次に例を示します。

```
# rmsock f1000e00029d4bb8 tcpcb
The socket 0xf1000e00029d4808 is being held by proccess 4063356 (muxatmd).
# rmsock f1000e0001b86bb8 tcpcb
The socket 0xf1000e0001b86808 is being held by proccess 18546850 (clstrmgr).
```

この例では、2 つの **ESTABLISHED** ソケット・ペアが存在します。**snmpd** と **muxatmd** の間の 1 つと、**snmpd** とクラスター・マネージャーの間の 1 つです。

8. **SNMP** ベースの状況コマンドを再実行します。コマンドが機能する場合、次のセクションを読む必要はありません。

#### **snmpdv3.conf** ファイルのトラブルシューティング:

このトピックは、**SNMP** 構成ファイルに関連した問題の解決に役立ちます。

1. 次のコマンドを使用して、稼働している **snmpd** のバージョンを判別します。

```
# ls -l /usr/sbin/snmpd
lrwxrwxrwx 1 root system 9 May 14 22:19 /usr/sbin/snmpd -> snmpdv3ne
```

**snmpdv1** は **/etc/snmpd.conf** ファイルを使用し、**snmpdv3** は **/etc/snmpdv3.conf** ファイルを使用します。

注: この手順の残りの部分では、デフォルト・バージョンである **snmpdv3** デーモンが稼働していることを想定しています。

2. **snmpdv3.conf** ファイルの認証とアクセス制御 (許可) の設定を検査します。**clinfoES**、**cldump**、および **cldisp** は、コミュニティ・ベースの認証を使用します。それらは、構成ファイルにリストされている最初のコミュニティを使用します。まれにですが、**clinfoES** に対してコミュニティを指定することもできます。この設定を検査するには、次のコマンドを使用します。

```
odmget SRCsubsys | grep -p clinfo
```

**cmdargs** フィールドの値を探します。

- このフィールドが空の場合、**clinfoES** は構成ファイル内の最初の **COMMUNITY** エントリーを使用します。
- このフィールドが **-c community\_name** に設定されている場合、**clinfoES** は **community\_name** を使用します。

注: **clinfoES** によって使用されるコミュニティを変更したい場合は、**chssys** コマンドを使用します。**clinfoES** によって使用されるコミュニティを変更した後、**clinfoES** を再始動する必要があります。

### 3. **snmpdv3.conf** ファイル内で最初の SNMP コミュニティーを見つけます。

```
# grep -i comm /etc/snmpdv3.conf | grep -v ^#
COMMUNITY powerha powerha noAuthNoPriv 0.0.0.0 0.0.0.0 -
COMMUNITY test test noAuthNoPriv 0.0.0.0 0.0.0.0 -
```

この例では、最初のコミュニティは **powerha** です。

- アンコメントされたコミュニティ・エントリーがない場合は、**snmpdv3.conf** ファイルにエントリーを追加する必要があります。これらのエントリーをテンプレートとして使用できます。任意のテキスト・ストリングをコミュニティ名として使用します (ただし、**public** は、ありふれているので、良い選択肢とは考えられません)。コミュニティ名は、行内の 2 番目と 3 番目のフィールドにする必要があります。
  - 変更を有効にするためには、ファイルを編集した後に **snmpd** を再始動する必要があります。ただし、再始動の前に、その他の変更が必要かどうか、ファイルの残りの部分を確認してください。
- ### 4. **snmpdv3** デーモンは、アクセス制御に View-based Access Control Model (VACM) を使用します。使用しているコミュニティに関連付けられている **VACM\_GROUP**、**VACM\_ACCESS**、**VACM\_VIEW** の各エントリーを見つけます。

- a. 最初のコミュニティに関連付けられているグループを見つけます。構成ファイルの中で、コミュニティ名を検索してください。その例を次に示します。

```
# grep powerha /etc/snmpdv3.conf
VACM_GROUP group1 SNMPv1 powerha -
TARGET_PARAMETERS trapparms1 SNMPv1 SNMPv1 powerha noAuthNoPriv -
COMMUNITY powerha powerha noAuthNoPriv 0.0.0.0 0.0.0.0 -
```

この例では、**VACM\_GROUP** が **group1** です。

- b. 識別されたグループを検索して、このグループに関連付けられているビューを見つけます。そのビューは **VACM\_ACCESS** エントリーにリストされています。

```
# grep group1 /etc/snmpdv3.conf
VACM_GROUP group1 SNMPv1 powerha -
VACM_ACCESS group1 - - noAuthNoPriv SNMPv1 defaultView - defaultView -
```

**VACM\_ACCESS** エントリーの構文は、以下のとおりです。

```
VACM_ACCESS groupName contextPrefix contextMatch securityLevel
securityModel readView writeView notifyView storageType
```

**readView** アクセスのビューの名前を探します。この例では、グループ **group1** の **readView** および **notifyView** アクセスに **defaultView** が使用されています。**writeView** と **storageType** のアクセスは提供されていません。

- c. 識別したビューを検索することにより、このコミュニティに関連付けられている **VACM\_VIEW** エントリーを見つけます。

```
# grep defaultView /etc/snmpdv3.conf
#VACM_VIEW defaultView      internet                - included -
VACM_VIEW defaultView      1.3.6.1.4.1.2.2.1.1.0  - included -
VACM_VIEW defaultView      1.3.6.1.4.1.2.6.191.1.6 - included -
VACM_VIEW defaultView      snmpModules            - excluded -
VACM_VIEW defaultView      1.3.6.1.6.3.1.1.4      - included -
VACM_VIEW defaultView      1.3.6.1.6.3.1.1.5      - included -
VACM_VIEW defaultView      1.3.6.1.4.1.2.6.191    - excluded -
VACM_ACCESS group1 - - noAuthNoPriv SNMPv1 defaultView - defaultView -
```

1) PowerHA MIB へのアクセス権限を与える **VACM\_VIEW** エントリーを探します。MIB 内のロケーションは、数値のストリング (オブジェクト ID (OID)) または名前 (オブジェクト・ディスクリプター) のいずれかによって識別されます。この例では、最初のエントリーはオブジェクト・ディスクリプター **internet** を使用しています。それは OID **1.3.6.1** に対応します。この行がアンコメントされている場合は、MIB 全体、つまり、1.3.6.1 と 1.3.6.1 で始まるすべてのもの (これは事実上 SNMP MIB 全体です) へのアクセスが許可されます。

2) ただし、この例では、**internet** ディスクリプターがコメント化されており、これは、そのレベルのアクセス権限がないことを意味します。AIX 7.1 以降、セキュリティ上の予防措置として、**snmpdv3.conf** ファイルは **internet** アクセスがコメント化されて出荷されています。これは、AIX 7.1 以降では、**snmpdv3.conf** ファイルを編集しない限り、デフォルトで PowerHA SNMP ベースの状況コマンドが機能しないことを意味します。また、関連する **VACM\_VIEW** エントリーの最後から 2 番目のフィールドに、**excluded** でなく **included** という語があることを確認してください。

3) 95 ページの『一般的な SNMP の問題のトラブルシューティング』のセクションで説明されているように、PowerHA MIB へのアクセスを提供するには、2 つの方法があります。

- **snmpdv3.conf** 内の **internet** 行をアンコメントします。これにより、MIB 全体にアクセスできるようになります。
- PowerHA MIB のみへのアクセスを提供する行を追加します。PowerHA MIB は、オブジェクト・ディスクリプターまたは OID によって識別できます。

5. **snmpdv3.conf** ファイルを編集して、最初のコミュニティが PowerHA MIB にアクセスできるようにします。必ず、ファイル内の最初の **COMMUNITY** エントリーが **VACM\_GROUP** エントリーにマップされ、そのエントリーが **VACM\_ACCESS** エントリーにマップされ、そのエントリーが、PowerHA MIB を含んでいる **VACM\_VIEW** にマップされるようにする必要があります。この例で必要となる変更は、**risc6000clsmuxpd** オブジェクト・ディスクリプターの **VACM\_VIEW** エントリーを追加することだけです。

```
VACM_VIEW defaultView      risc6000clsmuxpd      - included -
```

6. **snmpdv3.conf** ファイルを編集した場合は、**snmpd** を再始動します。

注: **snmpd** に対して、**refresh** コマンドでなく、**stopsrc** コマンドと **startsrc** コマンドを使用する必要があります。

```
stopsrc -s snmpd; startsrc -s snmpd
```

7. 97 ページの『SNMP の状況コマンドのトラブルシューティング』のセクションで説明されているように、ステップ 5、6、7 を繰り返して、クラスター・マネージャーが確実に **snmpd** に接続されるようにします。

8. SNMP ベースの状況コマンドを再実行します。

## ノードとリポジトリ・ディスクで同時に障害が発生する

このトピックでは、ノードとリポジトリ・ディスクに同時に障害が発生したときの対処方法について説明します。

## 問題

データ・センター障害などのイベント時に、ノードとリポジトリ・ディスクで同時に障害が発生する。

## 解決方法

データ・センターの障害時などにおけるノードとリポジトリ・ディスクの同時障害では、すべてのノードを再始動する前に、リポジトリ・ディスクの交換が必要になる場合があります。

1. リポジトリ・ディスクを交換するには、次の SMIT (System Management Interface Tool) パスを使用します。

```
$ smitty sysmirror
```

```
>Problem Determination Tools > Replace the Primary Repository Disk
```

注: リポジトリ・ディスクの交換中にダウン状態にあるノードは、リブートを行った後も、元の (交換前の) リポジトリ・ディスクへのアクセスを続行します。元の (交換前の) リポジトリ・ディスクが再び使用可能になると、Cluster Aware AIX (CAA) クラスタ・サービスは、そのディスクの使用を開始します。ノードはダウン状態のままです。

2. ノードの状況を確認するには、次のコマンドを入力します。

```
lscluster -m
```

このコマンドにより、次のような出力が作成されます。

```
Calling node query for all nodes...
Node query number of nodes examined: 2
Node name: ha1c1A
Cluster shorthand id for node: 1
UUID for node: 1ab63438-d7ed-11e2-91ce-46fc4000a002
State of node: DOWN  NODE_LOCAL
...
-----
Node name: ha2c1A
Cluster shorthand id for node: 2
UUID for node: 1ac309e2-d7ed-11e2-91ce-46fc4000a002
State of node: UP
...
Points of contact for node: 2
-----
Interface      State Protocol  Status
-----
en0             UP        IPv4       none
en1             UP        IPv4       none
```

3. 以前に障害が発生したノードが新しいリポジトリ・ディスクを強制的に使用するようにするには、影響を受けるノードで次のコマンドを入力します。

a. **\$ export CAA\_FORCE\_ENABLED=true**

b. **\$ clusterconf -fu**

4. CAA クラスタ・サービスがアクティブであるかどうかを確認するには、次のコマンドを入力します。

```
lscluster -c
```

注: 新しいリポジトリ・ディスクを使用して、ノードが CAA クラスタに参加するには、最大で 10 分を要する場合があります。

5. CAA クラスタ・サービスが正常に再始動したことを確認するには、次のコマンドを入力します。

a. **lscluster -c**

b. **lscluster -m**

6. 影響を受けるノードで PowerHA を再始動する前に、PowerHA 構成を同期化する必要があります。この同期化は、リポジトリ・ディスクの交換時に稼働中の状態であったノードで開始する必要があります。ノードで確認プロセスおよび同期化プロセスを開始するには、以下の SMIT パスを使用します。

```
$ smitty sysmirror
```

```
>Cluster Nodes and Networks > Verify and Synchronize Cluster Configuration
```

注: 使用可能なノードが複数あり、そのすべてのノードで PowerHA が稼働していない場合は、同期化を開始するためにアクティブなノードを選択する必要があります。

ステップ 6 の確認および同期化が正常に終了したら、次の SMIT パスを使用して、以前に障害が発生したノードで PowerHA を再始動することができます。

```
$ smitty sysmirror
```

```
>System Management (C-SPOC) > PowerHA SystemMirror Services > Start Cluster Services
```

---

## 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510

東京都中央区日本橋箱崎町19番21号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

*IBM Director of Licensing*

*IBM Corporation*

*North Castle Drive, MD-NC119*

*Armonk, NY 10504-1785*

*US*

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

記載されている性能データとお客様事例は、例として示す目的でのみ提供されています。実際の結果は特定の構成や稼働条件によって異なります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名前はすべて架空のものであり、名前や住所が類似する個人や企業が実在しているとしても、それは偶然にすぎません。

#### 著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物には、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年).

このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。

© Copyright IBM Corp. \_年を入れる\_.

---

## プライバシー・ポリシーに関する考慮事項

サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品（「ソフトウェア・オファリング」）では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オファリングにより個人情報が収集されることはありません。IBM の「ソフトウェア・オファリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オファリング」が、これらのCookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的事項を確認ください。

この「ソフトウェア・オファリング」は、Cookie もしくはその他のテクノロジーを使用して個人情報を収集することはありません。

この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人を特定できる情報を収集する機能を提供する場合、お客様は、このような情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意の要求も含まれますがそれらには限られません。

このような目的での Cookie などの各種テクノロジーの使用については、『IBM オンラインでのプライバシー・ステートメントのハイライト』(<http://www.ibm.com/privacy/jp/ja/>)、『IBM オンラインでのプライバシー・ステートメント』(<http://www.ibm.com/privacy/details/jp/ja/>) の『クッキー、ウェブ・ビーコン、その他のテクノロジー』というタイトルのセクション、および『IBM Software Products and Software-as-a-Service Privacy Statement』(<http://www.ibm.com/software/info/product-privacy>) を参照してください。

---

## 商標

IBM、IBM ロゴおよび [ibm.com](http://www.ibm.com) は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。



# 索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

## [ア行]

アクセス権

検査 48

アプリケーション

検査 37

イベント要約 18

収集された hacmp.out の表示 23

並列処理順序 27

保存 24

イベント・ブリアンブル 17

印刷キュー

高可用性 11

オペレーティング・システム

検査 53

## [カ行]

管理

クラスター・ログ 26

ログ

パラメーターの管理 34

ログ・ファイル・パラメーター

node 34

node

ログ・ファイル・パラメーターの管理 34

クライアントの問題 84

クラスター

構成の確認 39

スナップショットの確認 40

通信デーモンの検査 54

通信の問題 78

停止 3

メッセージ・ログ・ファイルの検討 12

リソース・グループの追跡 27

ログ・ファイルの収集 26

ログ・ファイルの表示 12

ログ・ファイルの理解 15, 17

クラスター状態情報ファイル 43

クラスター・ヒストリー・ログ

理解 25

クラスター・ログ

管理 26

検査

アクセス権 48

アプリケーション 37

検査 (続き)

クラスター構成 39

クラスター通信デーモン 54

クラスター・スナップショット 40

システム・ハードウェア 55

ディスク 53

ディスク・アダプター 53

ネットマスク 52

ファイルシステム 47

ファイルシステム情報 48

物理ネットワーク 53

物理ボリューム 45

ボリューム・グループ

定義 44

varyon 状態 44

マウント・ポイント 48

論理ボリューム 46

論理ボリューム・マネージャー 44

AIX オペレーティング・システム 53

IP アドレス 52

Point-to-Point 接続 51

PowerHA SystemMirror コンポーネント 38

TCP/IP サブシステム 49

検討

メッセージ・ログ・ファイル 12

構成データベースの復元

問題判別ツール 10

構成データベース・データ・ファイル 42

## [サ行]

システム・エラー・ログ 12

理解 24

システム・コンポーネント

調査 36

収集

クラスター・ログ・ファイル 26

使用

診断ユーティリティ 4

問題判別ツール 5

ジョブ・タイプ

並列リソース・グループ処理 28

診断ユーティリティ

使用 4

スイッチの問題 72

スクリプト障害

リカバリー

問題判別ツール 9

その他の問題 86

## [タ行]

### 調査

システム・コンポーネント 36

### 停止

クラスター・マネージャー 3

### ディスク

検査 53

ディスクの問題 62

ディスク・アダプター

検査 53

ディスク・フェンシング 70

トラッキング

リソース・グループ処理 27

## [ナ行]

ネットマスク

検査 52

ネットワークの問題 72

## [ハ行]

ハードウェア

検査 55

### 表示

クラスター・ログ・ファイル 12

収集された hacmp.out イベント要約 23

SMIT での hacmp.out 22

ファイルシステム

検査 47

ファイルシステム情報

検査 48

ファイルシステムの問題 62

物理ネットワーク

検査 53

物理ボリューム

検査 45

### 変更

印刷キューを高可用性に 11

cron ジョブを高可用性に 10

### 保存

イベント要約 24

ボリューム・グループ

定義の確認 44

varyon 状態の検査 44

## [マ行]

マウント・ポイント

検査 48

マルチキャスト

テスト 72

トラブルシューティング 74

メッセージ・ログ

検討 12

### 問題

クライアント 84

クラスター通信 78

検索 2, 3

スイッチ 72

その他 86

ディスク 62

ネットワーク 72

ファイルシステム 62

PowerHA SystemMirror 始動 56

PowerHA SystemMirror テークオーバー 79

### 問題判別ツール

構成データベースの復元 10

使用 5

スクリプト障害からの回復 9

PowerHA SystemMirror ログの表示および管理 8

## [ヤ行]

ユニキャスト

トラブルシューティング 75

## [ラ行]

### 理解

クラスター・ログ・ファイル 15, 17

システム・エラー・ログ 24, 25

リソース・グループ

hacmp.out での追跡 27

リポジトリ・ディスク 67

### ログ

クラスター・メッセージの検討 12

システム・エラー・ログの理解 24, 25

収集 26

情報のレベルの設定 22

表示 12

問題判別ツール

管理 8

表示 8

理解 15, 17

ログ・ファイルの管理 26

SMIT を使用した表示 22

論理ボリューム

検査 46

論理ボリューム・マネージャー

検査 44

## A

AIX オペレーティング・システム

検査 53

## C

clam\_nfsv4 アプリケーション 66  
cluster.log 15  
cron ジョブ  
    高可用性 10

## H

hacmp.out 12, 17, 22  
    イベント要約 18  
    イベント要約の保存 24  
    イベント・プリアンブル 17  
    記録される 情報のレベルの設定 22  
    収集されたイベント要約の表示 23  
    ジョブ・タイプ  
        並列リソース・グループ処理 28  
    並列処理順序 27  
    リソース・グループの追跡 27

## I

IP アドレス  
    検査 52  
IPv6 アドレス 75

## J

JOB\_TYPE  
    ACQUIRE 30  
    ERROR 29  
    NONE 30  
    OFFLINE 29  
    ONLINE 29  
    RELEASE 30  
    SERVICE\_LABELS 31  
    SSA\_FENCE 31  
    VGS 32

## L

LVM 分割サイト・ミラーリング 67

## P

Point-to-Point 接続  
    検査 51  
PowerHA SystemMirror コンポーネント  
    検査 38  
PowerHA SystemMirror 始動の問題 56  
PowerHA SystemMirror のテークオーバーの問題 79

## S

SMIT  
    hacmp.out の表示 22

## T

TCP/IP  
    サブシステムの確認 49  
tmp/clconvert.log 12

## [特殊文字]

.info 43  
.odm 42  
/usr/es/sbin/cluster/history/cluster.mmddyyyy  
    理解 25  
/usr/es/sbin/cluster/snapshots/ clsnapshot.log 12  
/usr/es/sbin/cluster/wsm/logs/ wsm\_smit.log 12  
/var/hacmp/adm/cluster.log 12, 15  
/var/hacmp/adm/history/cluster.mmddyyyy 12  
/var/hacmp/clverify/clverify.log 12  
/var/hacmp/log/ 12  
/var/hacmp/log/ clconfigassist.log 12  
/var/hacmp/log/ clstrmgr.debug.long 12  
/var/hacmp/log/ cl\_testtool.log 12  
/var/hacmp/log/autoverify.log 12  
/var/hacmp/log/clavan.log 12  
/var/hacmp/log/clinfo.log 12  
/var/hacmp/log/clstrmgr.debug 12  
/var/hacmp/log/clutils.log 12  
/var/hacmp/log/cspoc.log 12  
/var/hacmp/log/cspoc.log.long/var/hacmp/log/  
    cspoc.log.remote 12  
/var/hacmp/log/hacmp.out 12, 17, 22  
    イベント要約 18  
    イベント・プリアンブル 17  
    記録される 情報のレベルの設定 22  
    収集されたイベント要約の表示 23  
/var/hacmp/log/migration.log 12  
/var/hacmp/log/sa.log 12  
/var/ha/log/grpqlsm 12  
/var/ha/log/grpsvcs 12







Printed in Japan