

Adabas for Linux, UNIX and Windows

データベースのモニタリングとチューニング

バージョン 6.6

2017 年 10 月

このマニュアルは Adabas for Linux, UNIX and Windows バージョン 6.6 およびそれ以降のすべてのリリースに適用されます。

このマニュアルに記載される仕様は変更される可能性があります。変更は以降のリリースノートまたは新しいマニュアルに記述されます。

Copyright © 1987-2017 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their licensors.

The name Software AG, webMethods and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Software AG およびその子会社が所有する登録商標および特許の詳細については、<http://documentation.softwareag.com/legal/> を確認してください。

本ソフトウェアの一部にはサードパーティ製製品が含まれています。サードパーティの著作権表示およびライセンス規約については『License Texts, Copyright Notices and Disclaimers of Third-Party Products』を参照してください。このドキュメントは製品ドキュメントセットの一部であり、<http://documentation.softwareag.com/legal/>上、またはライセンス製品のルートインストールディレクトリ内にあります。

本ソフトウェアの利用は、Software AGのライセンス規約に則って行われるものとします。ライセンス規約は製品ドキュメントセット内、<http://documentation.softwareag.com/legal/>上、またはライセンス製品のルートインストールディレクトリ内にあります。

ドキュメント IDは: **ADAOS-DBMON-66-20200619JA**

目次

データベースのモニタリングとチューニング	v
1	1
表記規則	2
オンライン情報	2
データ保護	3
2 データベースの監視とチューニング	5
バッファプールマネージャ	6
リソース使用のモニタリング	10
データベース使用のレポート	11
データベース制御のモニタリング	11
パフォーマンス管理、統計、チューニング	12
コマンドロギング	13
showcls (UNIX のみ)	13
showipc (UNIX のみ)	14
Adabas サポートへの診断情報の提供	21

データベースのモニタリングとチューニング

このドキュメントでは、データベースパフォーマンスのモニタリングとチューニングについて説明します。

次のトピックについて説明します。

- [データベースのモニタリングとチューニング](#)

1

■ 表記規則	2
■ オンライン情報	2
■ データ保護	3

表記規則

規則	説明
太字	画面上の要素を表します。
モノスペースフォント	<code>folder.subfolder:service</code> という規則を使用して webMethods Integration Server 上のサービスの保存場所を表します。
大文字	キーボードのキーを表します。同時に押す必要があるキーは、プラス記号 (+) で結んで表記されます。
斜体	独自の状況または環境に固有の値を指定する必要がある変数を表します。本文で最初に出現する新しい用語を表します。
モノスペースフォント	入力する必要があるテキストまたはシステムから表示されるメッセージを表します。Program code.
{ }	選択肢のセットを表します。ここから1つ選択する必要があります。中カッコの内側にある情報のみを入力します。{} 記号は入力しません。
	構文行で相互排他的な2つの選択肢を区切ります。いずれかの選択肢を入力します。 記号は入力しません。
[]	1つ以上のオプションを表します。大カッコの内側にある情報のみを入力します。[] 記号は入力しません。
...	同じ種類の情報を複数回入力できることを示します。情報だけを入力してください。実際のコードに繰り返し記号 (...) を入力しないでください。

オンライン情報

Software AG マニュアルの Web サイト

マニュアルは、Software AG マニュアルの Web サイト (<http://documentation.softwareag.com>) で入手できます。このサイトでは Empower クレデンシャルが必要です。Empower クレデンシャルがない場合は、TECHcommunity Web サイトを使用する必要があります。

Software AG Empower 製品のサポート Web サイト

もしまだ Empower のアカウントをお持ちでないのなら、こちらへ empower@softwareag.com 電子メールにてあなたのお名前、会社名、会社の電子メールアドレスをお書きの上、アカウントを請求してください。

いったんアカウントをお持ちになれば、Empower <https://empower.softwareag.com/> の eService セクションにてサポートインシデントをオンラインで開くことができます。

製品情報は、Software AG Empower 製品のサポート Web サイト (<https://empower.softwareag.com>) で入手できます。

機能および拡張機能に関するリクエストの送信、製品の可用性に関する情報の取得、製品のダウンロードを実行するには、Products に移動します。

修正に関する情報を取得し、早期警告、技術論文、Knowledge Base の記事を読むには、[Knowledge Center](#) に移動します。

もしご質問があれば、こちらのhttps://empower.softwareag.com/public_directory.asp グローバルサポート連絡一覧の、あなたの国の電話番号を選んで、わたくし共へご連絡ください。

Software AG TECHcommunity

マニュアルおよびその他の技術情報は、Software AG TECHcommunity Web サイト (<http://techcommunity.softwareag.com>) で入手できます。以下の操作を実行できます。

- TECHcommunity クレデンシアルを持っている場合は、製品マニュアルにアクセスできます。TECHcommunity クレデンシアルがない場合は、登録し、関心事の領域として [マニュアル] を指定する必要があります。
- 記事、コードサンプル、デモ、チュートリアルにアクセスする。
- Software AG の専門家によって承認されたオンライン掲示板フォーラムを使用して、質問したり、ベストプラクティスを話し合ったり、他の顧客が Software AG のテクノロジーをどのように使用しているかを学んだりすることが可能です。
- オープンスタンダードや Web テクノロジーを取り扱う外部 Web サイトにリンクできます。

データ保護

Software AG 製品は、EU 一般データ保護規則 (GDPR) を尊重した個人データの処理機能を提供します。該当する場合、適切な手順がそれぞれの管理ドキュメントに記載されています。

2 データベースの監視とチューニング

■ バッファプールマネージャ	6
■ リソース使用のモニタリング	10
■ データベース使用のレポート	11
■ データベース制御のモニタリング	11
■ パフォーマンス管理、統計、チューニング	12
■ コマンドロギング	13
■ showcls (UNIX のみ)	13
■ showipc (UNIX のみ)	14
■ Adabas サポートへの診断情報の提供	21

バッファプールマネージャ

バッファプールの概要

バッファプールマネージャは、パフォーマンスを向上させるために導入されたデータベースシステムの重要な機能です。

データベースブロックが必要な場合、ディスクから直接読み取られず、バッファプールマネージャから要求されます。バッファプールマネージャは、データベースブロックをメモリ（バッファプール）に保持します。データベースブロックが要求された場合、バッファプールマネージャは、要求されたブロックがすでにバッファプールにあるかどうかを確認します。要求されたブロックがすでにバッファプールにある場合、ディスクから再度読み取る必要はなく、呼び出し元に直接返すことができます。ブロックがまだバッファプールにない場合にのみ、ディスクから読み込む必要があります。通常、要求されるほとんどのデータベースブロックは比較的頻繁に使用されているので、この方法で、物理ディスク I/O 数を大幅に削減できます。（1-物理 I/O / 論理 I/O）と定義されるバッファプール効率が、99% 以上になることがよくあります。

Adabas では、一時ブロック（NUCTMP、NUCSRT）の保存にもバッファプールを使用します。これらのブロックは、現在のニュークリアスセッションおよび内部作業領域の割り当てにのみ関係します。

バッファフラッシュ

バッファプールによって、読み込み I/O 数だけでなく、書き込み I/O 数も削減されます。データベースブロックが論理的にバッファプールに書き込まれた場合、それらはディスクに直接書き込まれるのではなく、特定の数のブロックが変更済みとしてマークされたときにのみ、「バッファフラッシュ」が実行されます。同じデータベースブロックが、高速で連続して数回更新されることがよくあるので、この方法で書き込み I/O 数が減少します。ただし、電源障害などでニュークリアスがクラッシュした場合でも、トランザクションが失われないことを保証する必要があります。これはログを使用して実行されます。このログは、トランザクションをロールバックするために必要です。Adabas では、この目的のために WORK コンテナが使用されます。

通常、バッファフラッシュは、個別のバッファフラッシュスレッドによって、Adabas ニュークリアスの通常の処理と並行して実行されます。したがって、通常、バッファフラッシュの影響はそれほど大きくありませんが、以下のような性能低下が起こる可能性があります。

- バッファフラッシュによって生じる I/O 負荷が高いため、ニュークリアスで実行される他の I/O が遅くなる場合があります。その結果、I/O を必要とするコマンドが遅くなる場合があります。
- I/O により、CPU 負荷が増大することもあります。例えば、I/O バッファへのデータコピーや、データが暗号化されてディスクに書き込まれる場合の暗号化などで負荷が増大します。

バッファフラッシュによるパフォーマンス低下の程度は、システムによって大きく異なります。ある環境では影響がないに等しく、他の環境では顕著になったり、支障が生じたりする場合があります。

通常、ディスクにまだ書き込まれていない変更されたデータベースブロックを小さく維持する必要があります。これは、WORK のログから再生成するために必要な時間を短くするためです。対照的に、一時ブロックは現在の Adabas ニュークリアスセッションにのみ関係します。そのため、ニュークリアスのクラッシュ時に再生成する必要はありません。バッファオーバーフローを回避するために、ディスクにのみ書き込む必要があります。

これらの理由から、2つの個別のフラッシュリスト（ディスクに書き込まれるブロックのリスト）を定義しました。1つはデータベースブロック用、もう1つは一時ブロック用です。また、これらのフラッシュリストでは、データベースブロックと一時ブロックに別々のバッファフラッシュが生成されます。使用可能なメモリが十分にある場合は、一時ブロックをディスクにまったく書き込む必要がないサイズのバッファプールを使用することをお勧めします。

バッファフラッシュを実行する理由は次のとおりです。

- 明示的なバッファフラッシュ：状況によっては、Adabas がバッファのフラッシュを明示的に開始します。例えば、ニュークリアスの起動中やシャットダウン処理中にチェックポイントエントリが作成された後で、ニュークリアスが明示的なバッファフラッシュを実行します。一部のユーティリティは、Adabas ファイルをディスクから直接読み取るので、明示的なバッファフラッシュを開始します。したがって、ユーティリティが処理を開始する前に、バッファプール内にある当該ファイルのすべての変更済みデータベースブロックをディスクに書き込む必要があります。
- 書き込み制限の超過：データベースブロックと一時ブロックには、異なる書き込み制限があります。データベースブロックの書き込み制限は、ニュークリアスのパラメータ WRITE_LIMIT で定義できます。一時ブロックの書き込み制限は 50% です。
- WORK 制限バッファフラッシュ：まだディスクに書き込まれていないブロックは、WORK に保存されているログレコードを使用して再生成する必要があります。これらのブロックの再生成に必要な WORK ブロックが多すぎる場合、WORK オーバーフローを防ぐためにバッファフラッシュが開始されます。
- スペース待機バッファフラッシュ：別のブロックを割り当てるためのバッファプールにフリースペースがない状態です。この場合、コマンド処理はバッファフラッシュが終了するまで待機します。これにより、コマンドの実行時間が長くなる可能性があるため、十分なメモリがあれば、書き込み制限を減らすか、バッファプールを増やして、この状態を回避する必要があります。
- 緊急バッファフラッシュ：WORK が実際にオーバーフローすると、バッファフラッシュが終了するまでコマンド処理が停止します。コマンドの実行時間が長くなる可能性があるため、この状態を回避する必要があります。書き込み制限を減らすか、WORK を増やすと役立つことがあります。
- 無視されたブロックのバッファフラッシュ：バッファフラッシュがブロックをディスクに書き込もうとしたときに、変更されたブロックが再び更新されると、バッファフラッシュはこのブロックを無視し、次のブロックの書き込みを続行します。ただし、緊急バッファフラッシュを

避けるために、このバッファフラッシュが終了した直後に、別のバッファフラッシュが開始され、無視されたブロックもディスクに書き込まれます。

自動再起動処理

バッファプール管理に関連する重要な問題として、自動再起動処理があります。電源障害などでニュークリアスが異常終了した後に、ニュークリアスの起動中に自動再起動が実行されます。自動再起動中には、ニュークリアスの異常終了時にバッファプールにあるものの、まだディスクに書き込まれていないブロックが再生成され、オープンしたすべてのトランザクションがロールバックされます。本番データベースの場合、自動再起動にかかる時間は、通常、本番運用のダウンタイムを意味するので、自動再起動時間は最小限に抑える必要があります。自動再起動時間には、次が関連します。

- データベースブロックの書き込み制限（ニュークリアスパラメータ `WRITE_LIMIT`）。書き込み制限が大きいほど、バッファフラッシュ中に再生成する必要があるバッファプール内に、より多くの変更されたブロックが存在する可能性があります。これにより、自動再起動の時間が長くなります。
- 一部のアプリケーションでは、`WORK` コンテナのサイズが重要です。`WORK` は、自動再起動処理のデータベース変更コマンドを記録するために使用されます。同じデータベースブロックが頻繁に更新される場合は、更新操作を記録するために多数の `WORK` ブロックが書き込まれることがあります。バッファフラッシュが発生する前に `WORK` ブロックが多く書き込まれるほど、より多くの更新操作を再生成する必要があります。これにより、自動再起動の時間が長くなります。このような場合は、書き込み制限でバッファのフラッシュがトリガされる前に、`Work` の自動再起動が実行されるので、小さい `WORK` を使用すれば、自動再起動時間を短縮できます。

バッファプール構成

バッファプールの構成が、データベース全体のパフォーマンスに大きな影響を与えることは、容易に想像できます。この目的のために、以下の `Adabas` ニュークリアスパラメータがあります。

- `BFIO_PARALLEL_LIMIT` - バッファフラッシュ I/O 並列制限。このパラメータは、並列に開始される I/O 数を制限します。この値が高すぎると、バッファフラッシュと並列に実行される読み取り I/O が非常に低速になることがあります。これは、読み取り操作の前に、最初に書き込み I/O が実行されることが理由です。最適な値は、ストレージシステムとオペレーションシステムで行われる I/O 最適化に大きく依存しますが、ほとんどの場合、デフォルト値の 50 が良好なパフォーマンスをもたらします。このパラメータは、ニュークリアスの実行中にユーティリティ `ADAOPR` を使用して動的に変更できます。
- `LBP` - バッファプール長：バッファプールの効率が 99% 以上になるように、バッファプールを十分な大きさに定義する必要があります。また、一時ブロックのバッファフラッシュが実行されないサイズのバッファプールが推奨されます。一方、最適なバッファプールサイズは使用可能なメモリによって制限されます。`LBP` は、オペレーティングシステムがページングを開始するほど大きくしないでください。

- READ_PARALLEL_LIMITSを使用すると、順次処理のためにデータベースブロックを先読みすることができます。このパラメータは、ニュークリアスの実行中にユーティリティ ADAOPR を使用して動的に変更できます。ただし、これでパフォーマンスが向上するのは、大部分の Adabas コマンドが、バッチプログラムなどでデータベースを順次処理する場合だけです。それ以外の場合は、使用されていないブロックを読み取ります。したがって、先読みを実際に活用するプログラムを実行する場合のみ、このパラメータを明示的にオンにすることをお勧めします。
- WRITE_LIMIT。このパラメータは、バッファフラッシュの開始トリガとなる、変更されたデータベースブロックの数を指定します。
 - 大きな値を指定すると、通常、書き込み I/O 数が減少します。ブロックが2回更新されたと仮定します。WRITE_LIMIT が小さい場合は、最初のアップデートの後、2 番目のアップデートの前にブロックがすでに書き込まれていることがあります。WRITE_LIMIT が大きい場合は、時間がかかります。
 - 一方、WRITE_LIMIT の値を大きくすると、電源障害などのニュークリアスのクラッシュが発生した場合、バッファプール内に、まだディスクに書き込まれていない、より多くの変更されたブロックが存在する可能性があります。これらのブロックは、データベースが再起動されたときに実行される「自動再起動」で再作成する必要があります。つまり、障害発生後にデータベースが再び作動するまでの時間が長くなるということです。
 - WRITE_LIMIT の値を小さくすると、スペース待機バッファフラッシュおよび緊急バッファフラッシュの回避に役立ちます。その後、バッファフラッシュが早期にトリガされ、変更されたブロックがディスクに書き込まれてから、スペース待機バッファフラッシュまたは緊急バッファフラッシュが発生する可能性があります。

WORK サイズの考慮事項

WORK コンテナはデータベースの変更を記録するために使用されます。自動再起動にも必要とされるので、バッファプール管理にも WORK サイズが重要となります。データベースの WORK サイズを定義する場合は、次の点に注意してください。

- WORK は、オープンしたすべてのトランザクションを記録するのに十分な大きさである必要があります。小さすぎると、トランザクションがロールバックされ、Adabas レスポンスコード 9、サブコード 3、「LP」が返されます。これが頻繁に発生し、長時間オープンになっているトランザクションだけに発生しているのではない場合は、WORK を増やす必要があります。
- 特に次の機能を使用する場合は、WORK に多くのログスペースが必要です。
 - ラージオブジェクト (LOB) 値。LOB 値の挿入または削除で必要となる追加の WORK スペースは、LOB 値に一部の管理データを加えたサイズです。LOB 値の更新には、2 倍のスペースが必要です。
 - Adabas-to-Adabas レプリケーション。すべてのデータベース変更は Adabas ファイル、レプリケーションシステムファイルに記録され、レプリケーションシステムファイルの変更も WORK に記録されます。レプリケーションを使用した WORK 要件は、レプリケーションを使用しない WORK 要件と比較して最大で 4 倍になる場合があります。

- 同じデータベースブロックが頻繁に更新される場合は、WORK オーバーフローを回避するために、書き込み制限を超過する前に、バッファフラッシュをトリガする必要があることがあります。これが大きな WORK コンテナで発生した場合、WORK サイズが自動再スタート時間の重要な要因となる可能性があります。WORK サイズを縮小すると役立つ場合があります。ただし、トランザクションのロールバックが必要になるまでは、WORK サイズを縮小しないでください。
- WORK バッファのフラッシュがトリガされ、データベースの更新負荷が高い場合、データベースブロックがディスクに十分な速度で書き込まれないので、緊急バッファフラッシュが必要になることがあります。これは、Adabas のパフォーマンスを大幅に低下させる可能性があるため、回避する必要があります。WORK を大きくするか、書き込みの制限を小さくすると役立つことがあります。

バッファプールフラグメント

データベースコンテナを定義する際に、ブロックサイズは 1 KB から 32 KB までの任意の倍数にすることができます。NUCTMP ブロックのブロックサイズは通常 4 KB です。

ただし、互いに倍数のブロックサイズのみを使用することをお勧めします。特に、2 の累乗のブロックサイズを使用することをお勧めします。そうしないと、バッファプールがフラグメント化する可能性があります。また、バッファプール内の完全なスペースを使用できない可能性があります。

バッファプール動作の監視

バッファプールの動作（および一般的にはデータベースの動作）を監視するために、次を使用できます。

- ADAOPR DISPLAY コマンド。特に DISPLAY = BP_STATISTICS および DISPLAY = BF_STATISTICS。
- ADAMON ユーティリティ。

詳細については、『ユーティリティマニュアル』を参照してください。

リソース使用のモニタリング

データベースの完全性が維持され、かつ効果的なレベルのサービスが提供されていることを保証するため、継続してデータベース環境のモニタリングを行うのは DBA の役目です。DBA の役割としては、さまざまな活動や手順が含まれます。このセクションでは、その中のいくつかを取り上げています。

DBA には、ある一定のレベルにデータベースパフォーマンスを保つ役目がありますが、これは重要で、しかも容易なことではありません。サービスの低下を招く原因は数多くあり、多くの場

合、それらを追求することは容易ではありません。原因が判明した場合でも、その調整には複雑なプロセスを要する場合があります。

DBA は、サービスの低下が起こる前にこれを予測したり、データベースのオペレーションや設計を定期的に制御された方法で調整する一連の手順を設定したりしなければなりません。これらの手順には以下の活動が含まれます。

- サービスの低下を引き起す可能性のある原因の識別
- データベースパフォーマンスをモニタするツールの設定
- 調整作業
- 新しい要求に応えるためのデータベースの実装または再設計

データベース使用のレポート

DBA は、データ処理部門とユーザー部門のマネジメントに対して、データベースの使用とパフォーマンスに関してレポートを作成し、定期的に報告する義務があります。このレポートに含まれる情報は、できるだけ事実に沿ったものでなくてはならず、同時にデータベース環境に対するチューニングの勧告も含まれていなくてはなりません。チューニングは組織全体には利益をもたらしますが、一部のユーザーにはサービスの低下になることもあります。したがって、チューニングに関する決定は、影響を受けるユーザーのことを考慮しなければなりません。

データベース制御のモニタリング

DBA は、データベースの完全性を保証するために適切な制御方法を設定し、これをモニタしなければなりません。

このコンピュータが作成する管理情報（コントロールトータル）がコンピュータ処理実行時間や生成レポートについてチェックするとき役に立ちます。バッチレスポンス（または問い合わせ）では、実行時間、サーチパラメータ、最終データ更新時刻およびプライマリパラメータコントロールのような情報を含みます。これは、信頼レベルを高め、データベースの完全性を保証するうえで役に立ちます。

コントロールトータルの問題は、インストールによってその形態もさまざまです。そのため、これについて厳しい固定的な規則を作成するのは不可能といえますが、一般的な指針であれば設定することはできます。

データベースを使用する各アプリケーションシステムの設計時に、DBA は次のような考慮がされているかを確認する必要があります。

- 各バッチ更新処理の実行時、どのようなチェックを行えるか。例えば、レコード件数、追加、削除、更新を行うときなど。

- これらをチェックするのにどんな制御がファイル全体にわたって必要か。例えば、値フィールドのハッシュトータルなど。
- ある期間の終わりに、コントロールトータルを調べ誤りであることがわかったとき、データを回復するためにどんな入力トランザクション、Adabas ログ等を保持しておかなければならないか。
- 地域別のコントロールトータル（支店別、製品グループ別）から、ファイルコントロールトータルエラーの影響を受ける領域を特定できるか。

パフォーマンス管理、統計、チューニング

次の表に、使用する統計のモニタリングのいくつかとデータベース環境に対する調整（またはチューニング）を示します。

変化するもの	必要なチューニング				
	データベースの構造	使用するアクセスメソッド	ハードウェアまたはソフトウェア構成	処理優先度	ディスクストレージ割り当て
端末およびライントラフィック		X	X	X	X
レスポンスタイム（アプリケーションのパフォーマンス）	X	X	X	X	X
ユーザーおよびディスクリプタによる総アクセス	X	X			X
データベースサイズ	X	X	X		X
データベースの成長度	X	X	X		X

本番データベースに何らかの変化があったときは、信頼性や完全性を高度に保つよう十分な注意が必要です。何が変化しても、DBA は正しい判断を下しそれが正確に行われるよう保証しなければなりません。チューニング処理に対してDBA は絶対的なコントロールを持ち、正式な受入れ手順に沿っているかを確認しなければなりません。

上の表中の項目の変化について過度に対応することがないようDBA は注意しなければなりません。ライントラフィックやレスポンスタイムなどが突然変化した場合、一時的な現象かもしれないので、通常のオペレーティング状態でしばらく様子を見て、恒久的な傾向なのか一時的な現象なのかを判断することをお勧めします。もう1つの表の見方として、新規プロジェクトの導入により、端末およびライントラフィック、レスポンスタイムなどに大幅な変化がみられるとき、チューニングが必要になる可能性があるのかどうかを確認することもできます。DBA は、新規アプリケーションシステムを実装する前に、これらの影響を最小限にとどめるように対応できます。

コマンドロギング

Adabas ニュークリアスの Adabas コマンドログオプションを使って、Adabas に対してユーザーが出した全コマンドに関する情報を生成できます。

得られる情報には、以下のものが含まれます。

- ユーザー ID
- 日付、時間
- 使用されたコマンド
- アクセスされたファイル
- アクセスされたレコード
- 受け取った Adabas レスポンスコード
- コマンドの実行にかかった時間

この情報を参照することで、データベースのアクティビティについて詳しく把握することができます。この情報は、ユーティリティ ADACLIP を使用することにより表示できます。

showcls (UNIX のみ)

showcls は、データベースのクライアントキューセマフォを分析するツールです。これは、次の構文で呼び出されます。

```
showcls <dbid>
```

showcls は、データベースの各クライアントキュー要素のステータスを表示します。

ステータス	説明
Free	クライアントキューエントリは、新しいクライアント接続に使用できます。
廃止	クライアントキューエントリは、次のクライアントキューのクリーンアップ後に再利用できません。
In use	クライアントキューエントリは現在使用中です。

showipc (UNIXのみ)

showipc は、DBA が UNIX カーネルの IPC 構成を見たり、Adabas/Entire Net-Work の IPC 構成を見たり、それらを削除するのに使用するツールです。各データベースに対して、そのデータベースに関連する共有メモリ、メッセージキュー、セマフォを showipc は表示します。

次のオプションを利用できます。

```
showipc [-adfhiksv] [-e<string>] [-g<g_name>] [-r<sec>] [-u<u_name>] [<dbid>...]
```

1 つ以上の <dbid> 引数を使用できます。<dbid> が省略された場合、showipc はすべてのアクティブなデータベースに対する IPC 構成を処理します。有効な DBID を表す数値に加えて、文字列 CSCI、ACS、U も使用できます。CSCI (Client Server Communication Interface) を指定すると、CSCI ユーザーの全 IPC エLEMENT の情報が返されます。ACS が指定されると、NETACS (NET-WORK アクセスサーバー) が使用している IPC ELEMENT の情報が返されます。U は、データベースを関連付けることができないすべての要素を示します (データベース不明)。

また showipc は、複数の論理 NET-WORK ノードをサポートしています。NET-WORK の IPC 構成は、<dbid> ゼロと NET-WORK ID によって識別されます。-a および -e オプション (下記説明を参照) が使用されない場合は、デフォルトの NET-WORK ID に関する情報が表示されます。環境変数 NET_WORK_ID が設定されていると、一致する NET-WORK ID も処理されます。



注意: 上記の NET-WORK に関する情報は、Adabas バージョン 6 ではサポートされなくなった NET-WORK バージョン 2 にのみ関係します。ただし、Adabas バージョン 6 の showipc は、Adabas バージョン 5 データベースもサポートしており、NET-WORK バージョン 2 でも引き続き使用できます。

次の表は提供されているさまざまなオプションの意味について説明しています。

オプション	説明
-a	同時に稼動している論理 NET-WORK の全ノードの構造を表示します。
-d	このオプションは、Adabas バージョン 3.1 以前で使用されていました。互換性の理由から、このオプションは現在のバージョンの Adabas でも指定できますが、効果はありません。
-e<string>	情報を表示させる NET-WORK ID を指定します。NET-WORK ID の最初の文字はユニークでなければいけないので、<string> のそれぞれの文字が 1 つの NET-WORK ID として認識されます。
-f	-k オプションとともに使用します。これは NETACS と CSCI (その他のものは削除されない) のELEMENT を強制的に削除します。
-g<g_name>	デフォルトユーザー ('sag', 'adabas', 'natural', 'esq', \$SIPGROUP) の代わりに、グループ <g_name> が所有している IPC ELEMENT を探します。
-h	showipc の使用方法について説明するヘルプ画面を表示します。

オプション	説明
-i	-k オプションとともに使用します。これは IPC エlement を削除する前に、削除の確認をします。
-k	使用されていない IPC Element を削除します。削除する前に、データベースの GCB 共有メモリが使用中かどうかを、showipc はチェックします。まだ使用中なら、削除をやめます。Software AG では、-i オプションと -k オプションを併用することを強く推奨しています。showipc は root の有効なユーザー ID で実行されます。そうしないと、別のユーザーが起動した Adabas ニュークリアスプロセスによって作成された IPC リソースを削除できません。
-r<sec>	<sec> 秒ごとに指定されたコマンドを繰り返します。Ctrl キーと C キーを同時に押すと、コマンドの実行が終了します。
-s	実行中の UNIX カーネルの IPC 構成値を表示します。これは UNIX カーネルが正しく構成されているかのチェックにも役立ちます。
-u<u_name>	デフォルトユーザー ('sag', 'adabas', 'natural', 'esq', \$SIPUSER) の代わりに、ユーザー <u_name> が所有している IPC Element を探します。
-v	データベースの詳細な情報を表示します。

showipc の出力

デフォルト出力フォーマットの最初の列には、NET-WORK の場合には「NET」（デフォルトの NET-WORK ID でなければ NET-WORK ID が続きます）、NETACS の場合は「NACS」、CSCI の場合は「CSCI」、対象データベースの DBID の数値のいずれかが表示されます。この後に、データベースに対する IPC Element のリストが続きます。各 Element のエントリは、そのタイプを示す略語、区切りを表すコロン、その Element の IPC ID から構成されます。この ID は「ipcrm」コマンドで使用されます（ただし showipc-k <dbid> の方がはるかに便利）。以下の略語が使用されます。

共有メモリ

ATB アタッチバッファ共有メモリ

CSA 共通共有エリア

CSM CSCI 共有メモリ

DRV Adabas IPC ドライバ共有メモリ

GCB ジェネラルコントロールブロック

GDT グローバルデータベース (Net-Work)

OPR ADAOPR (CSA= オプション使用時) によって作成された共有メモリ

PHx ID x を持つプロトコルハンドラ

ESi Adabas SQL サーバー共有メモリ ID の i

メッセージキュー

CLM コミュニケーションクライアントメッセージキュー
CSQ CSCI メッセージキュー
RSQ ユーザーレスポンスキュー
SRV コミュニケーションサーバーメッセージキュー
SVQ ニュークリアススレッドキュー
USQ ユーザー要求キュー

セマフォ

CLS コミュニケーションクライアントセマフォ
CSS CSCI セマフォ
PSE プライベートセマフォ
SEM 他のセマフォ
ESQ Adabas SQL サーバーセマフォ

詳細出力形式 (-v オプション) を指定すると、各エレメントに対する付加情報が表示されます。データベースごとに3つの表（共有メモリ、メッセージキュー、セマフォごとに1つずつの表）が表示されます。これらの表の列は次のような意味を持っています。

共有メモリ

列	説明
TYPE	IPC 要素のタイプ（省略形を参照）
NI	NET-WORK ID（デフォルト以外の場合）。1文字に切り捨てられることがあります。
ID	IPC ID
SIZE	共有メモリのサイズ（バイト単位）

列	説明
LOPPID	最後に shmop() コールを処理したプロセスのプロセス ID
CRPID	共有メモリを作成したプロセスのプロセス ID
ATTPROC	アタッチされたプロセスの数（オペレーティングシステムによって常にサポートされているわけではありません）

メッセージキュー

列	説明
TYPE	IPC 要素のタイプ (省略形を参照)
NI	NET-WORK ID (デフォルト以外の場合)。1 文字に切り捨てられることがあります。
ID	IPC ID
NBYTES	メッセージキューのバイト数
NMSGGS	メッセージキューのメッセージ数
LSND	最後にメッセージを送信したプロセスのプロセス ID
LRCV	最後にメッセージを受信したプロセスのプロセス ID。

セマフォ

列	説明
TYPE	IPC 要素のタイプ (省略形を参照)
NI	NET-WORK ID (デフォルト以外の場合)。1 文字に切り捨てられることがあります。
ID	IPC ID

NETACS エレメントの場合には、TYPE 列は、NETACS ドメインを表す DOMN に置き換わります。



注意: showipc はカーネルのメモリ構造を読み取るので、root 権限を持たせて showipc を実行しなければなりません。

実行カーネルにデフォルトの名前 (例えば HP-UX での「/hp-ux」など) がない場合、showipc は該当のエラーメッセージを発行して終了します。この現象は環境変数 SIP_KERNAM に正しい名前を設定することで回避できます。例えば、csh の場合は、「setenv SIP_KERNAM 'mykernel'」というように設定してください。

環境変数

次の環境変数を showipc と組み合わせて使用できます。

環境変数	説明
NET_WORK_ID	ここにはデフォルトの NET-WORK ID に加えて処理する NET-WORK ID を設定します (詳細については、showipc の -e オプションと -a オプションを参照)。
SIP_KERNAM	ここには UNIX カーネルの名前を設定します (詳細については、前述の注意を参照)。
SIPGROUP	環境変数 SIPGROUP を設定した場合、その内容がデフォルトグループのリストに付加されます (オプション -g を参照)。
SIPUSER	環境変数 SIPUSER を設定すると、その内容がデフォルトユーザーのリストに付加されます (オプション -u を参照)。例えば、SIPUSER に「harry」を設定すると、デフォルトユーザー (現時点では「sag」、「adabas」、「esq」、「natural」など) と「harry」に属するすべての IPC エレメントが表示されます。

メッセージ

これより、showipc 実行時に受け取るメッセージについて説明します。

Cleared up driver resources for DB xx

- 説明 showipc は、-k オプションを指定した場合、Adabas バージョン 2.1 以上の IPC 構造を削除するだけでなく、対応するデータベースの IPC ドライバのリソースも解放します。
- 対処 対処は必要ありません。

Could not attach to global data area of database xxx

- 説明 データベース xxx の CSA が不整合を起こしているか、showipc に適したフォーマットになっていません。
- 対処 この問題がデータベース稼動時に起きたら、サポートセンターにご連絡ください。

Database yy of Adabas Vz.z might be incorrectly displayed by showipc V x.x

- 説明 データベースのバージョンが showipc のバージョンよりも新しいために、このデータベースの IPC 構造の出力が正しいかどうか不明です。
- 対処 より新しいバージョンの適切な showipc を使用します。

Db xx is of version ww - please use version ww of showipc to remove

- 説明 データベース xx のニュークリアスは、呼び出された showipc よりも新しいバージョンです。showipc は、このデータベースの ipc 構造を削除しません。
- 対処 より新しいバージョンの適切な showipc を使用します。

Found inconsistent data structures for DB xxx

- 説明 データベース xxx の内部データ構造が showipc と矛盾しています。
- 対処 データベースのバージョンをチェックします。使用している showipc のバージョンよりデータベースのバージョンが新しい場合には、適切なバージョンの showipc を使用します。それでも問題が発生する場合には、サポートセンターにご連絡ください。

GCB of DB xx still in use, I do not remove anything

- 説明 まだアクティブなデータベース（ニュークリアスが稼働中か、Adabas ユーティリティがアクセスしている）に対して showipc -k がコールされました。IPC 構造は削除されません。
- 対処 どうしてもそのデータベースの IPC 構造を削除したい場合には、そのデータベースにアクセスしているプロセスを終了して（showipc -v <xx> を使用してどのプロセスかを確認してください）、再度実行してみてください。

GDT is still in use by DB xx; not removed

- 説明 少なくとも1つのGDTを使用しているデータベースがあります。これは、showipc
では削除できません。
- 対処 対処は必要ありません。

I do not remove private semaphores

- 説明 showipcは、指定したデータベースに関連しないAdabasセマフォを削除しません。
- 対処 対処は必要ありません。Adabasユーティリティの失敗／異常終了で残ったセマフォ
は ipcrm を使用して削除できます。

NET-WORK xx still running, I do not remove anything

- 説明 指定されたIDを使用したNET-WORKはまだアクティブです。showipcは、この
NET-WORK IDに関連付けられているipc構造を削除しません。
- 対処 対処は必要ありません。

Sorry, but this program has to be run with supervisor privileges

- 説明 showipc 実行可能ファイルに正しい権限が設定がされていません。ユーザーIDの
権限が設定され、オーナーはrootでなければなりません。
- 対処 権限をチェックし、必要であれば、権限を正しく設定してください。

Sorry, I will remove CSCI elements only with -f option

- 説明 -f オプションを使用すると、CSCI エlementだけが削除されます。
- 対処 -k オプションに -f オプションを追加します。

Sorry, I will remove CSCI elements only with -f option

- 説明 -f オプションを使用すると、NET-ACS エlementだけが削除されます。
- 対処 -k オプションに -f オプションを追加します。

Structure mismatch for DB xx - Cannot identify version

- 説明 問題のデータベースの内部データ構造を正しく解釈することができません。
- 対処 このメッセージは、データベースの起動中に発生することがあります。この場合は
無視してかまいません。

Warning: could not find adanod for clean up of Adabas driver

- 説明 Adabas IPC ドライバリソースを削除するときに、ユーティリティ adanod が必要になります。
- 対処 環境変数 ADANOD に adanod をフルパス名で設定してください。

Warning: Found <structure> with invalid database ID xx contained

- 説明 関連ユーザーの 1 人に属する構造が間違っただけを持っています。その構造とは、共有メモリ、メッセージキュー、セマフォのどれかです。
- 対処 該当する構造を作成したプロセスをチェックします (showipc は作成したプロセスのプロセス ID を表示します)。そのプロセスが Software AG 製品でない場合は、Adabas に干渉しないようにその製品を修正してください。そのプロセスが Software AG 製品の場合は、サポートセンターにご連絡ください。

Warning: Some NET-WORK IDs may be displayed truncated to one character

- 説明 showipc の -e または -a オプション使用時に、showipc はいくつかの NET-WORK ID の中から 1 文字だけ取り込みました。NET-WORK ID は最初の 1 文字でユニークかどうかを判断します。
- 対処 環境変数 NET_WORK_ID を適宜、設定することによって、他の NET-WORK ID も対象にできます。

例

```
showipc
```

すべての Adabas データベースおよびデフォルトの NET-WORK ID の IPC 構造を表示します。

```
showipc -av
```

すべての Adabas データベースおよびすべての論理 NET-WORK ID の IPC 構造を詳細表示形式で表示します。

```
showipc -ki 127
```

データベースニュークリアスが実行されていない場合、データベース 127 のすべての ipc 構造が削除されます。showipc で ipc 構造が削除される前に確認を求められます。

Adabas サポートへの診断情報の提供

全般的な情報

eServiceまたは電子メールでインシデントをサポートするために、通常、添付ファイルのフォーマットで診断情報を送信する必要があります。5MB以上の添付ファイルは、FTP、SFTPまたはFTPESで送信する必要があります（お住まいの地域での以下の指示に従ってください）。Software AGからの要請があった場合のみ、DVD（またはその他の承認済みの物理メディア）を送付してください。

すべての種類の診断データについて、FTP、SFTP、またはFTPESを使用してソフトウェア AG にアップロードする際には、次のファイル命名規則を使用してください。ファイルの名前は、2文字の「SR」で始め、その後7桁のサポートインシデント番号を続ける必要があります。このプレフィックスに続けて、意味のあるわかりやすいファイル名とファイル拡張子を使用できます。有効な例は次のとおりです。

- SR1234567Diagnostics1.zip
- SR1234567Diagnostics2.zip
- SR1234567_more.1000.Z
- SR1234567_configuration_information.txt
- SR1234567ACMETestCorp.acme
- SR1234567_no_file_extension

大きなファイルをアップロードする際に問題が発生した場合は、WinZip や 7-Zip などのファイル圧縮ツールで提供されている Split to Volumes 機能を使用する必要があります。

送信した文書の完全性を確認するために、ファイルのMD5ハッシュ合計を入力することを検討してください。

FTP 経由でファイルを送信する

➤ **手順 2.1. 「プレーン」 FTP 経由でファイルをアップロードするには、次の操作を行います。**

- 1 最も近いサーバーを選択します。
 - ftp.softwareag.com（ドイツの Software AG 本社にあります）
 - ftp.softwareagusa.com（バージニア州レストンの Software AG 米国支社にあります）
- 2 次の資格情報でログインします。
 - ユーザー ID：customer

- パスワード：customer



注意: 「匿名」を使用した場合、ファイルはサポートインシデントに関連付けられません。

ディレクトリは空であるように見え、DIR コマンドや LS コマンドは実行できません。そのため、アップロード後はファイルを表示できません。ただし、約 45 分間の処理時間が経過すれば、Empower の eService セクションがファイルを受信したことを確認できます。サポートインシデントには、タイプが「FTP - Inbound」のステップが表示されます。

FTP クライアントの「BIN」オプションを使用してください。

障害情報の準備

カスタマサイトで障害が発生した場合、Software AG では、迅速な解決のためにできる限りの努力をします。障害の分析に必要な時間を最小にして、解決策を提供するため、担当者は障害についての詳細な情報を必要とします。

標準的な障害情報には次の項目が含まれます。

- 障害について
 - 使用した Adabas コンポーネントまたは機能
 - エラーメッセージ（メッセージ ID）
 - 障害の簡単な説明
- 環境について
 - Adabas バージョン
 - 施した修正
 - オペレーティングシステムのバージョン
 - ハードウェア環境（CPU タイプ、ディスクデバイスタイプなど）
- 障害の再現について
 - 再現するための手順
 - 再現に必要なデータ

障害が再現しない場合、障害情報には次の項目も必要です。

- 障害が発生した場所
 - すべてのデータベースに発生するのか、または単一データベースだけに発生するのか
 - 特定環境のデータベースだけに発生するのか
 - 障害が発生しない環境
- 障害発生タイミング

- 常に発生するのか、または散発的に発生するのか
- 他のイベントと並行して発生するのか
- 障害が発生しない期間
- 問題が最初に発生したのはいつか
- その頃に行われた環境の変更
- それ以前で最後に行われた環境の変更
- データ破壊の障害があった場合：
- データベースチェックが正常に実行された最終日付
- ファイルが破壊された状態で使用したユーティリティ（チェックポイントリスト）
- 発生したディスク障害

