

アサーション・エラーの対処方法

このマニュアルでは、アサーション・エラー・メッセージが表示されたときにデータベースをリカバリする手順について説明します。また、アサーション・エラーについてテクニカル・サポートに問い合わせるときに必要な情報のチェックリストも含まれています。

このマニュアルは、Adaptive Server Anywhere バージョン 5.5 ~ 11.0.1 を対象にしています。参照先のマニュアルは、すべて SQL Anywhere Studio 11.0.1 のマニュアルです。アサーション・エラーに関してさらに支援が必要な場合は、テクニカル・サポートにお問い合わせ頂くか、SQL Anywhere 掲示板に質問を投稿してください。SQL Anywhere 掲示板については、<http://www.ianywhere.jp/developers/newsgroups.html> を参照してください。

- ・ [アサーション・エラーとは](#)
- ・ [アサーション・エラー・メッセージが表示されたときに行うこと](#)
- ・ [アサーション・エラーからデータベースを保護するには](#)
- ・ [データベースとトランザクション・ログの有効なバックアップからリカバリする方法](#)
- ・ [有効なバックアップはないが、データベース作成時から使用してきた有効なトランザクション・ログがある場合のデータベースのリカバリ方法](#)
- ・ [データベースをレプリケーションまたは同期する場合の注意点](#)
- ・ [有効なバックアップが存在せず、データベースは起動するが、アサーション・エラーが頻繁に発生する場合のデータベースのサルベージ](#)
- ・ [有効なバックアップが存在せず、データベースを起動するとすぐにアサーション・エラーが発生する場合のデータベースのサルベージ](#)
- ・ [破損したデータベースに関するチェックリスト](#)

アサーション・エラーとは

SQL Anywhere には、できるかぎり早期にデータベースの破損を検出するための多くの内部チェックがあります。アサーション・エラーが発生すると、ただちにデータベース・サーバが現在の要求をすべてキャンセルし、データベース・サーバがシャットダウンするまで、それ以降のすべての要求に対してエラーをレポートします。データベース・サーバは、アサーション・エラー発生後にデータベース処理の続行を拒否することで、破損がデータベース内に広がる危険性を最小限に抑えます。

アサーションが発生すると、データベース・サーバが処理を中断し、[Server Messages] ウィンドウまたは出力ログに以下の内容が出力されます。

```
*** ERROR *** Assertion failed: 123456 (x.0.x.xxxx) (** エラー ** アサーションに失敗  
しました: 123456 (x.0.x.xxxx))
```

エラー・メッセージの後にアサーションの原因に関する情報が出力されます。

注: アサーション・エラーは、SQL Anywhere Studio のバージョンによって異なる場合があります。アサーション・エラーの発生時に表示されるエラー・メッセージは、SQL Anywhere Studio のリリースごとに若干異なります。アサーション・エラー・メッセージの内容をすべてメモしておくことをおすすめします。

アサーション・エラーの発生後、アサーション番号とアサーションの原因となったエラーの詳細が表示されます。アサーション・エラーは、データベース・サーバが予期しない状態に遭遇したことを示します。予期しない状態が発生すると、データベースの破損の進行を防

止するために、データベース・サーバは現在のすべての接続を終了し、新しい接続を拒否して、データベースを停止します。

予期しない状態が発生すると、ハードまたはソフトのいずれかのアサーション・エラーが発生することがあります。また、ソフトウェアのバグが原因の場合もあります。ハード・アサーションは、通常は外部プロセスやハードウェアの障害が原因で、物理的なデータベース・ファイルまたはトランザクション・ログの整合性に問題が発生したことを示します。ソフト・アサーションは、誤ったクエリ結果、バックグラウンド・プロセスの不正動作、または複数の接続に影響する可能性があるその他の予期しない障害の原因となる状態にデータベース・サーバが遭遇したときに発生することがあります。データベースのアサーション・エラーは、物理データベース・ファイルの問題が根本的な原因である可能性があります。

アサーション・エラー・メッセージが表示されたときに行うこと

1. データベース・サーバのメッセージ・ウィンドウまたはログ・ファイルに出力されたアサーション番号とメッセージを記録します。アサーション番号とメッセージは、アサーションの原因を特定する際に重要な情報です。テクニカル・サポートへお問い合わせいただくときに、この情報が必要になります。
2. データベース・サーバがまだ実行されている場合はシャットダウンします。これは、複数のデータベースを 1 つのデータベース・サーバで実行している場合に非常に重要です。これにより、同一データベース・サーバで実行されている他のデータベースの破損を防止します。

3. データベース・ファイルとトランザクション・ログのバックアップ・コピーを作成します。
また、まだ削除していない以前のトランザクション・ログのバックアップ・コピーも作成します。これらのファイルのバックアップ・コピーは、単純なファイル・コピーで作成する必要があります (dbbackup または他のバックアップ・プロシージャは使用しないでください)。以前のデータベース・ファイルとトランザクション・ログのバックアップ・コピーは上書きしないでください。リカバリを行うには、データベース・ファイルとトランザクション・ログをアサーション・エラーの発生直後の状態に維持することが重要です。
4. データベース・サーバを再起動し、データベース・ファイルをロードします。
5. データベースが正常に起動した場合、検証ユーティリティ (dbvalid または sa_validate プロシージャを使用) を使用してデータベースを検証します。
 - dbvalid でエラーがレポートされた場合、または別のアサーション・エラーが発生した場合は、データベース・ファイルが破損している可能性があります。この場合は、検証済みのバックアップとリカバリ方法を実行することをおすすめします。検証済みのバックアップとリカバリ方法がない場合は、テクニカル・サポートにお問い合わせください。
 - dbvalid が正常に終了し、エラーがレポートされなかった場合は、そのデータベースを運用状態に戻すことができます。それ以降にアサーション・エラーが発生した場合は、検証済みのバックアップとリカバリ方法を再度実行してください。検証済みのバックアップとリカバリ方法を実行した後も問題が発生する場合は、テクニカル・サポートにお問い合わせください。

データベース検証後に dbvalid でエラーがレポートされない場合でも、わずかな破損が発生している可能性があります。データベース・ファイルの破損を解消するには、unload/reload プロセスを使用してデータベースを再構築する必要があります。データベースを再起動するだけでは、データベース・ファイルに破損がないことを保証できません。

検証ユーティリティとその構文の詳細については、オンラインマニュアルの『SQL Anywhere サーバ - データベース管理』の「データベースの管理」の「データベース管理ユーティリティ」の「検証ユーティリティ (dbvalid)」をご参照下さい。

sa_validate システム・プロシージャの詳細については、オンラインマニュアルの『SQL Anywhere サーバ - SQL リファレンス』の「システム・オブジェクト」の「システム・プロシージャ」の「システム・プロシージャのアルファベット順リスト」の「sa_validate システム・プロシージャ」をご参照下さい。

6. データベース・サーバが正常に起動した場合、データベースのリカバリ・プロセスが試行されます。リカバリ・プロセスが成功したときに、検証して破損の修復を完了する前にデータベースを運用状態に戻す必要がある場合は、新しいトランザクション・ログを使用してデータベースを再起動します。新しいトランザクション・ログを使用してデータベースを起動するには、すでに実行中のデータベースをシャットダウンし、以前のトランザクション・ログの名前を変更してから、データベースを再起動します。

手順 5 でデータベースが正常に起動しない場合は、データベースをトランザクション・ログなしで強制起動することができます。ただし、データベースがレプリケーションや同期 (Mobile Link、SQL Remote、または Replication Server) に関係していない場合にかぎり、これには、以下のコマンドを使用します。

```
dbengX -f dbfile.db
```

- X には 50, 6, 7, 8, 9, 10, 11 のいずれかの数字を指定します。
- *dbfile.db* にはデータベース・ファイルのパスを指定します。

通常は、データベース・サーバをリカバリするときに、データベース・ファイルが前のチェックポイントまでロールバックされ、そのチェックポイント以降のトランザクション・ログ内のすべてのトランザクションが適用されます。dbengX -f を実行すると、トランザクション・ログ内のトランザクションを適用せずに、最後のチェックポイントまでリカバリできます。これは、データベース・ファイルが想定しているトランザクション・ログの場所にトランザクション・ログがない場合にかぎり、トランザクション・ログがある場合は、データベース・サーバのコマンド・ラインで -f を使用しているかどうかに関係なく、データベースはログ内で最後のチェックポイント以降のトランザクションを適用しようとします。トランザクション・ログの破損が疑われる場合は、元のトランザクション・ログが使用されないように、トランザクション・ログの名前を変更してから、dbengX -f を実行する必要があります。

アサーション・エラーからデータベースを保護するには

アサーション・エラーに対する最善の予防策は、オペレーティング・システムのクラッシュ、ディスク障害、ファイル破損、マシンの全体的な故障を考慮した、検証済みのバックアップとリカバリ方法です。検証済みのバックアップとリカバリ方法により、アサーション・エラー発生時のダウンタイムを最小限に抑えることができます。

バックアップとリカバリの詳細については、オンラインマニュアルの『SQL Anywhere サーバ - データベース管理 》データベースの保守 》バックアップとデータ・リカバリ』をご参照下さい。

データベースとトランザクション・ログの有効なバックアップからリカバリする方法

データベースのアサーションまたは破損からリカバリする最善策は、データベース・ファイルとトランザクション・ログの有効なバックアップからデータベースをリカバリすることです。最後の有効なバックアップからアサーション・エラーの発生時までの間のシーケンスで失われたトランザクション・ログがなければ、アサーション・エラーによるデータ損失はありません。

以下の手順は、バックアップ方法の一環としてすでに実行されているはずです。データベースのバックアップのコピーを事前に検証し、バックアップの整合性を確認しておくことが理想的です。

注: データベースが有効かどうかをテストする場合は、必ずバックアップのコピーを使用するか、または読み取り専用モードでデータベース・サーバを実行してください。データベ

スを起動することにより、トランザクション・ログへのチェックポイントの書き込みなどの動作が発生し、トランザクション・ログの終了ログ・オフセットが変わってしまうため、これらのテストは、バックアップのコピーに対して実行するか、読み取り専用モードでデータベース・サーバを実行して行う必要があります。したがって、後でこのバックアップに対して別のトランザクションを適用した場合、最初のトランザクション・ログ・ファイルの終了トランザクション・ログ・オフセットが、次のトランザクション・ログ・ファイルの開始ログ・オフセットと一致しなくなります。

バックアップの整合性をまだ確認していない場合は、バックアップのコピーに対して検証ユーティリティ (dbvalid) を使用し、そのデータベースのバックアップ・コピーが有効かどうかを確認してください。これにより、ほとんどの無効なデータベースが検出されますが、それでもデータベースに存在するわずかな破損が検出されない場合があります。バックアップ・データベースが破損していないことを確実にするには、dbunload を実行し、バックアップ・コピーから新しいデータベースを作成してください。リロード時にエラーが発生せずにデータベースが再構築されれば、バックアップ・コピーは有効です。

バージョン 10.0.0 以前のデータベースの再構築の詳細については、オンラインマニュアルの『SQL Anywhere 11 - 変更点とアップグレード』の「SQL Anywhere 11 へのアップグレード」の「SQL Anywhere のアップグレード」の「バージョン 9 以前のデータベースをバージョン 11 用に再構築」をご参照下さい。

バージョン 10.0.1 以降のデータベースの再構築の詳細については、オンラインマニュアルの『SQL Anywhere 11 - 変更点とアップグレード』の「SQL Anywhere 11 へのアップグレ

ード » SQL Anywhere のアップグレード » バージョン 10 以降のデータベースの再構築」をご参照下さい。

これで、データベースのバックアップの検証が完了したので、バックアップのコピーをリストアします。ここでリストアしたデータベースのコピーに対して、トランザクション・ログを適用します。トランザクション・ログを適用すると、アサーション・エラーの発生時までにはデータベースに対して実行されたすべてのトランザクションが実行されます。最後のバックアップ以降にトランザクション・ログがトランケートされていない場合は、適用する必要があるトランザクション・ログは 1 つだけです。これは、以下のように実行します。

```
dbengX dbfile.db -a dbfile.log
```

- ・ X には 50, 6, 7, 8, 9, 10, 11 のいずれかの数字を指定します。
- ・ *dbfile.db* にはデータベース・ファイルのパスを指定します。
- ・ *dbfile.log* にはデータベース・トランザクション・ログのパスを指定します。

トランザクション・ログの名前が変更されてトランケートされている場合は、このコマンドを使用して、各トランザクション・ログを順に適用します。最初に、データベース・ファイルとともにバックアップされたトランザクション・ログを適用します。名前の矛盾を避けるために、適用するすべてのトランザクション・ログは、データベース・ファイルとは別のディレクトリに保存してください。すべてのトランザクション・ログを適用したら、データベースを再起動します。データベースの新しいトランザクション・ログが作成され、現行のトランザクション・ログになります。

バージョン 10 以降の場合、-a オプションの代わりに -ad オプションを使用できます。このオプションは、データベースに適用するトランザクション・ログ・ファイルを格納するディレクトリを指定します。また、トランザクション・ログの適用順序も指定します。

```
dbengX dbfile.db -ad c:¥¥backup
```

このプロセスでは同期やレプリケーションは中断されませんが、これらのトランザクション・ログのいずれかに、同期またはレプリケーションに必要なオフセットがある場合に備えて、適用したトランザクション・ログをすべて保存しておく必要があります。これが同期またはレプリケーション用の環境でない場合は、適用したトランザクション・ログを保存しておく必要はありません。

有効なバックアップはないが、データベース作成時から使用してきた有効なトランザクション・ログがある場合のデータベースのリカバリ方法

SQL Anywhere のデータベース・アーキテクチャは、トランザクション・ログとデータベース・ファイルを使用します。トランザクション・ログには、データベースに対して実行されたすべての文が記録されます。したがって、すべての文が記録された単一のトランザクション・ログを使用して、データベース全体を作成し直すことができます。このプロセスは、同期およびレプリケーションを中断します。

LOAD TABLE 操作が実行され、元のトランザクション・ログ・ファイルにログが記録されていた場合、データベース・サーバはこの操作を再び実行しようとします。これらのコマンド

が成功するには、データ・ファイルが元の場所にまだ存在し、同じデータが格納されていない
ければなりません。

このプロセスを使ったデータベースのリカバリ方法

1. 初期化ユーティリティを使用して、既存のデータベースと同一の初期化パラメータ
で新しいデータベースを作成します。ほとんどの初期化パラメータは、情報ユーテ
ィリティ (dbinfo) を使用して取得できます。場合によっては、破損したデータベース
に対して dbinfo を実行できないことがあります。

注:情報ユーティリティは、SQL Anywhere 8 または 9 のデータベース内の Java
についての情報だけを返します。Adaptive Server Anywhere 7 以前のデータベ
ースで Java を使用している可能性がある場合は [SAIL の TechDoc\(その他技術資
料\)](#) から「包括的な単一のログ・ファイルを使用したリカバリ」で説明されているプロ
セスをご参照下さい。

初期化ユーティリティの詳細については、オンラインマニュアルの『SQL
Anywhere サーバ - データベース管理 》 データベースの管理 》 データベース
管理ユーティリティ 》 初期化ユーティリティ (dbinit)』をご参照下さい。

情報ユーティリティの詳細については、オンラインマニュアルの『SQL Anywhere
サーバ - データベース管理 》 データベースの管理 》 データベース管理ユー
ティリティ 》 情報ユーティリティ (dbinfo) 』をご参照下さい。

2. トランザクション・ログ変換の試行時に問題が発生した場合に備えて、トランザクション・ログのバックアップ・コピーを作成します。
3. ログ変換ユーティリティを使用して、トランザクション・ログを変換します。

これで、データベースに対して過去に実行されたすべての文を含む SQL スクリプト・ファイルが作成されます (デフォルトのファイル名は *transaction-log-file.sql*)。

```
dbtran dbfile.log
```

- *dbfile.log* にはトランザクション・ログ・ファイルのパスを指定します。

ログ変換ユーティリティの詳細については、オンラインマニュアルの『SQL Anywhere サーバ - データベース管理 》 データベースの管理 》 データベース管理ユーティリティ 》 ログ変換ユーティリティ (dbtran)』をご参照下さい。

データベース・サーバを起動します。

```
dbengX -n reload dbfile.db
```

- X には 50, 6, 7, 8, 9, 10, 11 のいずれかの数字を指定します。

4. *dbfile.db* にはデータベース・ファイルのパスを指定します。

- ・ Interactive SQL を使用して、ログ変換ユーティリティで作成された SQL スクリプト・ファイルを適用します。

```
dbisql -c "UID=DBA;PWD=sql;ENG=reload"  
c:¥transaction-log-file.sql
```

- ・ c:¥*transaction-log-file*.sql には dbtran で作成された SQL スクリプト・ファイルのパスを指定します。

Interactive SQL の詳細については、オンラインマニュアルの『SQL Anywhere サーバ

- データベース管理 » データベースの管理 » データベース管理ユーティリティ »

Interactive SQL ユーティリティ (dbisql)』をご参照下さい。

- ・ Interactive SQL を使用して、ログ変換ユーティリティで作成された SQL スクリプト・ファイルを適用します。

```
dbisql -c "UID=DBA;PWD=sql;ENG=reload"  
c:¥transaction-log-file.sql
```

- ・ c:¥*transaction-log-file*.sql には dbtran で作成された SQL スクリプト・ファイルのパスを指定します。

Interactive SQL の詳細については、オンラインマニュアルの『SQL Anywhere サーバ

- データベース管理 » データベースの管理 » データベース管理ユーティリティ »

Interactive SQL ユーティリティ (dbisql)』をご参照下さい。

データベースをレプリケーションまたは同期する場合の注意点

レプリケーションまたは同期用の環境では、トランザクション・ログとトランザクション・ログ・オフセットについて注意する必要があります。データベースを再構築すると、トランザクション・ログ・オフセットが元のデータベースとは同一でないため、これらの環境では問題が発生します。このため、トランザクション・ログなしでデータベースを強制起動しないでください。データベースの再構築が必要な場合は、以下のように実行してください。

```
dbunload -ar path-for-old-dbfiles -c  
"UID=dba;PWD=sql;ENG=dbserver"
```

もしくは、オンラインマニュアルの『SQL Anywhere サーバ - SQL の使用法』の「リモート・データとバルク・オペレーション」の「データのインポートとエクスポート」の「データベースの再構築」の「同期やレプリケーションに関連するデータベースの再構築」の手順に従って、手動でデータベースを再構築することもできます。

注: Mobile Link 統合データベースの場合、トランザクション・ログ・オフセットに依存しないため、この制限は適用されません。ただし、この場合もトランザクション・ログについては注意が必要です。Mobile Link のリモート・データベースについては、この制限が適用されません。

レプリケーションまたは同期に関係する環境でアサーション・エラーからリカバリする最善策は、検証済みのバックアップとリカバリ方法を使用することです。

バックアップとリカバリ方法の作成の詳細については、オンラインマニュアルの『SQL Anywhere サーバ - データベース管理』の「データベースの保守」の「バックアップとデータ・リカバリ」の「バックアップとリカバリのプランの設計」をご参照下さい。

有効なバックアップが存在せず、データベースは起動するが、アサーション・エラーが頻りに発生する場合のデータベースのサルベージ

データベースの起動には成功し、データベースに対して特定の処理を実行したときにだけアサーション・エラーが発生する場合は、インデックスまたはテーブルの破損がアサーション・エラーの原因である可能性があります。このような破損は、多くの場合、破損したインデックスまたはテーブルに対処するためにデータベースを再構築することで解決できます。

破損したインデックスがあるデータベースでは、以下のようなアサーション・メッセージが表示される可能性があります。

```
*** ERROR *** Assertion failed: 100305 (12.0.0.2483)
Invalid index page encountered during an index scan @1
(table id 12, page 0x1a8)

(*** エラー *** アサーションに失敗しました: 100305 (12.0.0.2483)

インデックス・スキャン @1 の途中で無効なインデックス・ページが見つかりました

(テーブル ID 12、ページ 0x1a8))
```

このメッセージは、インデックスの破損によって発生する可能性のあるエラーのうちの 1 つです。インデックスの破損が原因となるアサーション・エラー・メッセージは多数あります。

アサーション・エラー・メッセージの情報だけで、破損しているインデックスを特定できることもあります。そのような場合は、以下の手順を実行します。

1. データベース・ファイルとトランザクション・ログのバックアップ・コピーを作成します。
2. 特定された破損インデックスを削除し、再作成します。
3. データベースを検証します。
4. データベースの検証でエラーが発生しなければ、インデックスは修正されていません。

破損しているインデックスを特定できない場合、または上記の手順によりデータベースで別のアサーション・エラーが発生した場合は、データベースを無順序で再構築します。すでに上記の手順を実行した場合は、データベース・ファイルとトランザクション・ログのバックアップ・コピーに戻してから、無順序の再構築を実行します。無順序の再構築を実行するには、以下のコマンドを使用してデータベースをアンロードします。 *dbfile.db* にはデータベース・ファイルのパスを指定します。

```
dbunload -u -ar path-for-old-dbfiles -c
"UID=DBA;PWD=sql;DBF=dbfile.db"
```

もしくは、以下のコマンドでもデータベースをアンロードできます。

```
dbunload -u -c "UID=dba;PWD=sql;DBF=dbfile.db" c:¥unload
dbinit newdbfile.db
dbengX -n new newdbfile.db
```

reload.sql ファイルを適用するために以下のコマンドを実行します。

- `dbisql -c "UID=DBA;PWD=sql;ENG=new" reload.sql`
 - Replace *reload.sql* with the path to the SQL script file created by `dbunload`

破損したテーブルがあるデータベースでは、以下のようなアサーション・メッセージが表示されます。

```
*** ERROR *** Assertion failed: 201501 (11.0.0.2483)
Page for requested record not a table page or record not
present on page
```

(*** エラー *** アサーションに失敗しました: 201501 (11.0.0.2483))

要求されたレコードのページがテーブル・ページではないかまたはレコードがページに存在しません)

テーブル破損によって引き起こされるアサーション・エラーから回復するための最初の手順は、破損している 1 つまたは複数のテーブルを特定することです。これには、検証ユーティリティ (dbvalid) を使用してデータベースを検証します。

検証ユーティリティの詳細については、オンラインマニュアルの『SQL Anywhere サーバ - データベース管理 》 データベースの管理 》 データベース管理ユーティリティ 》 検証ユーティリティ (dbvalid) 』をご参照下さい。

破損しているテーブルを特定できたら、[SAIL の TechDoc\(その他技術資料\)](#) から「データベースに不正なページがあるときにデータをサルベージする」の手順を参照して実行します。

有効なバックアップが存在せず、データベースを起動するとすぐにアサーション・エラーが発生する場合のデータベースのサルベージ

上記のいずれの方法でもデータベース・アサーションを解決できない場合は、テクニカル・サポートにお問い合わせいただくことをおすすめします。

iAnywhere Solutions は、破損したデータベースからデータをリカバリするデータ・サルベージ・サービスを提供しています。これは有料サービスで、テクニカル・サポート・ケースを開くことによりご利用いただけます。データ・サルベージ料金は、テクニカル・サポートの契約料金とは別料金です。データ・サルベージ・サービスでは、テクニカル・サポート・ケースを開く企業と iAnywhere Solutions の間での署名入りの契約書が必要になります。契約書には、サルベージ可能なデータがある場合のデータ量、その整合性、またはサービス時のサルベージに要する時間について、iAnywhere Solutions が一切の保証を行わないことが明示されています。このサービスは、他のどの方法でもデータをリカバリできなかった場合の最終手段として利用してください。データベース内のデータを保護する最善策は、検証済みのバックアップとリカバリ方法を実行することです。破損したデータベースからデータをリカバリするデータ・サルベージ・サービスは、検証済みのバックアップとリカバリ方法の代替方法ではありません。

データ・サルベージのためにデータベースを送信する必要がある場合、テクニカル・サポートは以下の質問に対してできるだけ多くの回答が必要です。この情報は、データベースをできるかぎり効率的にリカバリし、破損の原因を特定するのに役立ちます。

破損したデータベースに関するチェックリスト

1. DBA (または DBA 権限) のユーザ名 ID とパスワード

データの所有権を特定するために必要な情報です。お客様がデータの正当な所有者であることを確認します。

2. アサーション・エラー・メッセージの正確な番号と内容

アサーション・エラー・メッセージの正確な番号と内容は、アサーションの原因を特定する際に重要な情報です。アサーションの原因を特定することで、アサーションに対処するための手順を決定できます。メッセージには、そのアサーションに固有の情報、つまり、アサーションごとに異なる値が含まれています。

3. このデータベースの作成に使用したソフトウェアのバージョン

アサーションの原因が、データベース作成に使用したソフトウェアのバージョンに関連するソフトウェア内のバグであった場合、問題がすでに解決されている可能性があります。この情報があれば、アサーションの原因を特定してケースを解決するまでの時間を大幅に短縮できます。

4. データベース・サーバとして使用している SQL Anywhere のバージョンとビルド

SQL Anywhere のバージョンとビルドは、このアサーションがソフトウェアのバグに關係するかどうかを判断するために重要な情報です。データベース・サーバが最新の EBF でない場合、最新の EBF の README ファイルを読み、データベー

ス・アサーションの原因となっている可能性のあるソフトウェア・バグがないかどうかを確認することをおすすめします。テクニカル・サポートの担当者は、データベース・アサーションの原因を診断するために、オペレーティング・システムおよび SQL Anywhere のバージョンとビルドを確認する必要があります。

5. (可能な場合、必要に応じてバックアップからの) dbinfo 出力

SQL Anywhere バージョン 6 以降を使用している場合は、以下のコマンドを使用して dbinfo 出力を取得します。お使いのデータベースのユーザ ID、パスワード、およびデータベース・ファイル名を指定してください。

```
dbinfo -o c:¥info.txt -c "UID=dba;PWD=sql;DBF=dbfile.db"
```

SQL Anywhere Studio バージョン 5 を使用している場合は、以下のコマンドを使用して dbinfo 出力を取得します。

```
dbinfo -o c:¥info.txt dbfile.db
```

SQL Anywhere バージョン 8 以降を使用している場合は、以下のコマンドを使用して dbtsinfo からの出力を提供してください。

```
dbtsinfo -o c:¥dbtsinfo.txt -c  
"UID=DBA;PWD=sql;DBF=dbfile.db"
```

6. データベース・ファイルに対して dbupgrad を実行したことがあるかどうか

アップグレードの場合、再構築と違って、新しいデータベース・ファイル形式に依存しないシステム・スキーマのみが変更されます。そのため、特定のバージョンにアップグレードされたデータベースは、そのバージョンで初期化されたデータベース

と必ずしも同一にはなりません。クラスタード・インデックスのサポートの導入など、ファイル形式の変更が必要な例は多数あります。アップグレードされたデータベースには、クラスタード・インデックス機能がありません。

7. データベース・サーバを実行しているオペレーティング・システムとそのバージョンおよびビルド

特定のプラットフォームだけに存在するバグがある可能性があります。オペレーティング・システムの情報がわかれば、当社のナレッジ・ベースを検索し、合理的な検索を行うことができます。

8. サーバのファイル・システム

特定のファイル・システムだけに存在するバグがある可能性があります。ファイル・システムの情報がわかれば、当社のナレッジ・ベースを検索し、合理的な検索を行うことができます。

9. サーバのファイル・システム

特定のファイル・システムだけに存在するバグがある可能性があります。ファイル・システムの情報がわかれば、当社のナレッジ・ベースを検索し、合理的な検索を行うことができます。

10. 使用しているハードディスク・コントローラの種類

この情報を確認するには、[コントロール パネル] から [管理ツール] [コンピュータの管理] [デバイス マネージャ] を選択し、[ディスク ドライブ] 項目を展開します。型名が表示されます。ページのさらに下の方に、コントローラの種類が表示されます (IDE/SATA または SCSI など)。バージョン情報がわかる場合はそれも提供してください。

注: 上記の手順は Windows 専用です。この情報を調べるには、お使いのオペレーティング・システムに付属しているマニュアルを参照してください。

11. ハードウェア・キャッシングがあるかどうか

Windows の場合、[コントロール パネル] から [管理ツール] [コンピュータの管理] [デバイス マネージャ] を選択し、[ディスク ドライブ] 項目を展開します。お使いのデータベースが格納されているドライブを右クリックします。[プロパティ] を選択し、[ポリシー] タブを表示します。[ディスクの書き込みキャッシュを有効にする] が選択されているかどうかを確認します。

注: 上記の手順は Windows 専用です。この情報を調べるには、お使いのオペレーティング・システムに付属しているマニュアルを参照してください。

12. msinfo32.exe の情報

この情報を取得するには、[スタート] [ファイル名を指定して実行] から msinfo32.exe を実行します。ファイルを保存し、保存した .NFO ファイルをテクニカル・サポートに送信してください。

注:上記の手順は Windows 専用です。この情報を調べるには、お使いのオペレーティング・システムに付属しているマニュアルを参照してください。

13. サーバ・マシンに搭載されている CPU の個数

マシン上の CPU の個数に問題が関係している場合があります。

14. データベース・サーバを起動するコマンド・ライン

データベース・サーバに対して実行したオプションについての情報はとても重要です。アサーションのデバッグに役立ちます。

15. サーバ・マシンで実行しているウイルス・スキャナがあるかどうか。ある場合は、そのバージョンとブランド

ウイルス・スキャナがデータベースまたはトランザクション・ログ・ファイルをロックする可能性があります。

16. データベースのバックアップ方法

SQL Anywhere に付属のツール (dbbackup) を使用しているか、またはサード・パーティの製品を利用しているか。バックアップ方法に関する情報はリカバリ方法の特定に役立ちます。

17. データベースの最新のバックアップがあるかどうか

バックアップを使用することで、データ・サルベージのコストを節約できます。バックアップがデータベースのトランザクション・ログの履歴にすぎない場合でも、バックアップはリカバリで使用できる唯一のメカニズムです。

18. データベースのページ・サイズ

この情報を調べるには、dbinfo コマンドを実行します (上記 5 を参照)。

19. アサーション・エラー発生時に何が起きていたか (わかる場合)

データベースのロード中であったか、異常終了からデータベースをリカバリ中であったか、停電が起きたかなど。アサーションは一般的なものもあります。データベースで何が起きていたかがわかれば、特にデータベースの破損が原因でない場合、アサーションの原因の特定に役立ちます。

20. データベースと同じディレクトリ内に ASSERT.DMP ファイルがあるか

ある場合、そのファイルを送信してください。バージョン 9.0.2 の場合、sa_dump.dmp という名前のファイルがあります。バージョン 10.0.0 以降の場合、.mini_core という拡張子が付いたファイルを探してください。さらに何らかのクラッシュ情報があれば、クラッシュの根本的な原因の特定に役立ちます。assert.dmp ファイルは、検出された破損ページのダンプです。生成されるダンプとは異なります。

21. データベースとトランザクション・ログ・ファイルの両方の送信

データベースとトランザクション・ログの両方があれば、クラッシュ原因の特定に役立ちます。正常時のファイルよりも、アサーション・エラー発生時のデータベース・ファイルの方が理想的です。問題発生直後のデータベース・ファイルのコピーを作成することをおすすめします。それにより、データベース起動時に消去される可能性がある情報を保持できます。

22. このデータベースがレプリケーションに関係するかどうか (SQL Remote, Replication Server、または Mobile Link)

テクニカル・サポートがデータベースを起動するためにトランザクション・ログを処分してよいかどうかを判断するときにこの情報が役立つため、これは重要な情報です。データベースを起動できる場合、Interactive SQL (dbisql) で SELECT DB_PROPERTY('LTMTTrunc'), DB_PROPERTY('RemoteTrunc'), DB_PROPERTY('SyncTrunc') を実行します。3 つの戻り値のうちのいずれかが 2⁶⁴ -1 (つまり、18446755073709551615) でない場合、データベースがレプリケーションまたは同期に関係しています。