

“アサーション・エラー – 50213 または 200601 を dbunload で修正する手続き”

* この技術資料は、Sybase Inc.にて作成された英文資料を SybaseK.K にて日本語翻訳したものです。

(原文: "Assertion Error – 50213/200601 dbunload Fix Procedure" <http://my.sybase.com/detail?id=1010806>)

本書では、アサーション・エラー50213 または 200601 のメッセージを受信した場合にとる手順について説明します。

テクニカル・サポートに報告される、ある種類のデータベース破壊は、通常はアサーション・エラー50213「ページのページ番号が、要求されたページと一致しません」に関連しています。ASA6 または 7 で実行している場合は、おそらくアサーションは 200601「要求されたレコードのページがテーブル・ページでないか、レコードがページ上に存在しません」になります。ただし、最も多いのは、データベースがバージョン 5.x で作成されていて、ソフトウェアの更新前にそのデータベースが壊れてしまうケースです。最初に失敗した操作によっては、別のアサーションが報告されることもあります。

当社の製品品質部門は、データベース破壊の原因と頻度を調査するために、現在開発中であるバージョンのソフトウェア(本書の作成時点では、SSA 5.5.05、ASA 6.0.4、ASA 7.0.1)上で壊れてしまうデータベースを社内ですべて引き取りたい意向を示しています。ただし、場合によっては、下記の手続きの方が役立つこともあります。また、更新されなくなったソフトウェア(通常は、現在販売されている保守レベルとその1つ前のレベル)上で実行している場合は、このような情報の有用性は限定されます。

データベースが壊れた場合は、有効なバックアップを見つけて、そのバックアップ以後のログ・ファイルをすべて適用するのが望ましい解決法です。

ただし、特定のデータベース・ページをアクセスしたときにしか表面化しないため、このタイプの破壊には特に注意が必要です。そのページのデータをあまり使用しない場合は、お客様が管理しているバックアップの多く、または全部で破損が生じている可能性が十分にあります。

場合によっては、次の手続きでデータベースをリカバリすることができます。ただし、データの一部が失われる可能性があります。リカバリされたデータベースのデータの正当性と完全性を確認するのはお客様の責任となります。

dbunload コマンドを使用することになるので、以下の dbunload スイッチを確認しておいてください。

(コマンド・プロンプトで、dbunload /?と入力する)

用法: dbunload[switches] <directory> [table-name-list]

スイッチ(次のように、指定された小文字を使用してください):

- c "keyword=value; ..."
データベースの接続パラメータを指定する
- d データのみアンロードする
- e リストされているテーブルのデータを出力しない
- ii 内部アンロード、内部再ロード(デフォルト)
- ix 内部アンロード、外部再ロード
- j <count> ビュー作成文の繰り返し回数
- n データなし - スキーマ定義のみ
- o <file> 出力メッセージをファイルに記録する
- p <char> エスケープ文字(デフォルトは"¥")
- q クワイエット: メッセージを出力しない、またはウィンドウを表示しない
- r <file> 生成された再ロード ISQL コマンド・ファイルの名前を指定する(デフォルトは"reload.sql")
- u データの順序付けはしない
- v 冗長メッセージ
- xi 外部アンロード、内部再ロード
- xx 外部アンロード、外部再ロード
- y 確認メッセージを表示せずにコマンド・ファイルを上書きする

注意: <directory>は、外部アンロードを使用する場合を除き、エンジン(またはサーバ)に対して意味のあるパスとして指定する必要があります。

Adaptive Server Anywhere 6 および 7 にはスイッチが追加されています。その中でも注目に値するのは-an と-ac です。これらのスイッチを使用すると、リカバリ処理全体をまとめて 1 つのコマンド・ラインにすることができます。ただし、これらのスイッチの使用時に再ロードに失敗した場合は、リカバリ処理全体

を最初からやり直す必要があります。そのため、ここでは、部分的に障害が生じた場合でもデータとコマンド・ファイルを残す 3 段階の手動プロセスを使用します。

作業ディレクトリ(たとえば、c:¥working)とアンロード・ディレクトリ(たとえば、c:¥unload)を作成します。これらのディレクトリをルートから離して配置すると、作業が楽になります。データベースとログ・ファイルを作業ディレクトリにコピーします。作業ディレクトリでコマンド・プロンプトを開きます。次のコマンド・ラインを実行します(この例では、ASA6 に付属の asademo.db を使用しています)。

```
dbunload -c "uid=dba;pwd=sql;dbf=c:¥working¥asademo.db" -u c:¥unload
```

出力は次のようになります。

```
Unloading "DBA"."sales_order" into c:¥unload¥192.dat (relative to server)
Unloading "DBA"."sales_order_items" into c:¥unload¥193.dat (relative to server)
Unloading "DBA"."contact" into c:¥unload¥194.dat (relative to server)
Unloading "DBA"."customer" into c:¥unload¥195.dat (relative to server)
Unloading "DBA"."fin_code" into c:¥unload¥196.dat (relative to server)
Unloading "DBA"."fin_data" into c:¥unload¥197.dat (relative to server)
Unloading "DBA"."product" into c:¥unload¥198.dat (relative to server)
Unloading "DBA"."department" into c:¥unload¥199.dat (relative to server)
Unloading "DBA"."employee" into c:¥unload¥200.dat (relative to server)
```

当社テクニカル・サポートの場合は、この時点で上記例のアサーション・エラー・メッセージを確認します。テーブル"customer"でアサーション・エラーが発生したと仮定します。

次のように入力します。

```
dbunload -c "uid=dba;pwd=sql;dbf=c:¥working¥asademo.db" -u c:¥unload -e dba.customer
```

出力は次のようになります。

```
Unloading "DBA"."sales_order" into c:¥unload¥192.dat (relative to server)
Unloading "DBA"."sales_order_items" into c:¥unload¥193.dat (relative to server)
Unloading "DBA"."contact" into c:¥unload¥194.dat (relative to server)
Unloading "DBA"."fin_code" into c:¥unload¥196.dat (relative to server)
Unloading "DBA"."fin_data" into c:¥unload¥197.dat (relative to server)
Unloading "DBA"."product" into c:¥unload¥198.dat (relative to server)
Unloading "DBA"."department" into c:¥unload¥199.dat (relative to server)
Unloading "DBA"."employee" into c:¥unload¥200.dat (relative to server)
```

この処理は、正常に完了するか、または別の破損テーブルを発生させます。アンロードが正常に完了するまで、除外するテーブルのコンマ区切りリストを使用してプロセスを再実行します。

これで、破損テーブルを除き、必要なデータを備えた一連の*.DAT ファイルと、データベースのスキーマ

を備えた reload.sql というファイルが用意されたこととなります。次に、データベースの完全なスキーマが必要となります。RELOAD1.SQL を作成するため、次のように入力します。

```
dbunload" -c "uid=dba;pwd=sql;dbf=c:¥working¥asademo.db" -n -r ELOAD1.SQL
```

好みのテキスト・エディタ(ノートパッドなど)を使用して RELOAD.SQL と RELOAD1.SQL を開き、RELOAD DATA を検索します。このセクションの全内容を reload.sql から RELOAD1.SQL の同じセクションにコピーします。

このセクションには、次に示す LOAD TABLE パラグラフが含まれます。

```
LOAD TABLE "DBA"."CONTACT"  
FROM 'c:¥¥unload¥¥430.dat'  
FORMAT 'ASCII'  
QUOTES ON ESCAPES ON STRIP OFF  
CHECK CONSTRAINTS OFF  
go
```

アンロードを実行するエンジンのバージョンによって、細かい部分が多少異なります。これらのパラグラフのいずれかを、壊れている各テーブルごとに1回ずつ複製します。テーブル名("DBA"."CONTACT")を壊れたテーブルの名前("DBA"."CUSTOMER")に置き換えます。DAT ファイルを、同じディレクトリ内の一意のファイル名('c:¥¥unload¥¥customer.dat')に置き換えます。

```
LOAD TABLE "DBA"."CUSTOMER"  
FROM 'c:¥¥unload¥¥customer.dat'  
FORMAT 'ASCII'  
QUOTES ON ESCAPES ON STRIP OFF  
CHECK CONSTRAINTS OFF  
go
```

RELOAD1.SQL ファイルを保存します。

次に、壊れたテーブルから正常なデータを取り出す必要があります。スタンドアロン・エンジン上でデータベースを起動し、Interactive SQL に接続します。次の構文を使用して、テーブルからデータを選択して出力します。

```
SELECT * FROM customer ORDER BY cust_id ASC >># c:¥unload¥customer.dat
```

これにより、便宜上、プライマリ・キー順にエンジンがアサートするまで、そのテーブルから customer.dat に可能な限り多くのデータが出力されます。

エンジンを再起動し、Interactive SQL に再接続して、ソート順を反転します。

```
SELECT * FROM customer ORDER BY cust_id DESC >># c:¥unload¥customer.dat
```

二重の>記号を入力すると、新しい出力が最初のセットに結合されます。エンジンは再度アサートします。書き込まれたローの数と期待数を比較するか、customer.dat を開いて第 1 パスの最終ローのプライマリ・キー値を第 2 パスの最終ローのプライマリ・キー値と比較して、失われたデータの量を確認します。ある程度のデータ損失は避けられませんが、かなりの数のローがなくなっている場合は、次の構文を使用してプロセスを続行する必要があるかもしれません。

```
SELECT * FROM customer where cust_id between 1600 and 20000 ORDER BY cust_id ASC  
>># c:¥unload¥customer.dat
```

複数のローが壊れていて、それらのローがかなり離れている場合は、この構文が冗長になる可能性があります。幸いなことに、これらのローはプライマリ・キーで頻繁にグループ化されます。グループ化されない場合は、インデックスを持つ別のカラムでソートするか、データを取り出す前にこのテーブルのインデックスをすべて削除することができます。

できるかぎり多くのデータを取り出したら、Sybase Central または dbinit コマンドを使用して新しいデータベースを作成する必要があります。照合順、ブランク埋め込み設定、大文字と小文字の区別、およびページ・サイズは、必ず元のデータベースと一致させてください。

大量のキャッシュを使用してスタンドアロン・エンジン/パーソナル・サーバで新しいデータベースを起動し、Interactive SQL でそのデータベースに接続して、次のように入力します。

```
READ C:¥WORKING¥RELOAD1.SQL
```

これにより、元のデータベースから取り出すことができたスキーマとデータがすべて読み取られます。もう 1 つの問題は、外部キー関係を適用するときに発生する可能性があります。親レコードはないが子レコードが存在するという場合は、外部キー関係は確立されません。この外部キー関係を機能させるには、子レコードの削除、親レコードの置換、または親レコードのダミーの作成を行う必要があります。

元のデータベースとは異なる名前で作成された場合は、DOS/Explorer を使用してファイルの名前を変更し、dblog を実行してログ・ファイルの名前を変更します。

```
DBLOG -t asademo.log asademo.db
```

データが完全で正確であることを確認するのはお客様の責任となります。

このタイプの破壊は旧バージョンのソフトウェアでよく起こるため、バージョン 5.5.05 以降のソフトウェアにアップグレードすることを強くお勧めします。