Adabas for Linux, UNIX and Windows

XA サポート

バージョン 6.6

2017 年 10 月

ADABAS & NATURAL

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。

ドキュメント IDは: ADAOS-XA-66-20200619JA

目次

XA サポート

このドキュメントには、XA サポートに関する情報が含まれます。

次のトピックについて説明します。

■ *XA サポート*

1

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	*folder.subfolder:service* 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。

もしご質問があれは、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 クレデンシャ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 クレデンシャルがない場合は、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 XA サポート

一般的な知識

Adabas は、X/OPEN CAE Specification on Distributed Transaction Processing（XA 仕様、X/OPEN ドキュメント番号 XO/CAE/91/300）に準拠し、グローバルトランザクションをサポートします。グローバルトランザクションは、一般的にはトランザクションモニタの制御下で実行されるか（この場合、アプリケーションはトランザクションモニタに対する X/OPEN TX 仕様インターフェイスを使用できます）、またはアプリケーションが XA インターフェイスを直接使用できます。グローバルトランザクションは通常、複数のリソースマネージャ間でやり取りされます。リソースマネージャとしては、例えば Adabas ニュークリアスがありますが、Adabas ニュークリアスが複数あるということは、Adabas データベースも複数あることを意味しています。

XA 仕様の要件に従って、このようなグローバルトランザクションは 2 フェーズコミットプロトコル（2PC）を使ってコミットする必要があります。Adabas が 2PC の第 1 フェーズにトランザクションブランチのコミットを認めた場合、2PC はそれ自体の判断でトランザクションを終了（コミットまたはロールバック）してはいけません。

第 1 フェーズのトランザクションを "保留トランザクション" と呼びます。しかし、トランザクションに関わる別のコンポーネントが失敗すると、そのトランザクションは長い間保留状態のままになる可能性があります。保留トランザクションは、データベースが異常終了したり、シャットダウンされたりした場合でも残っていなければなりません。残っていれば、保留トランザクションはデータベースの自動再スタートで復元されます。

データベースの内容を変更するユーティリティは、当然、保留しているトランザクションがある間は実行できません。しかし、保留状態のままのトランザクションが実用的でない場合があります。例えば、データベースがダンプされているとき（ADABCK DUMP=*）、または新しいプロテクションログファイルがオープンされようとしているとき（ADAOPR FEOF=PLOG）、つまりすべてのトランザクションが終了している必要があるときです。この場合、Adabas は XA 仕様に従って、いわゆるトランザクションの"ヒューリスティックな終了"を実行します。これは、トランザクションマネージャと連携せずに、Adabas が保留トランザクションを終了することを意味します（詳細については、『*メッセージおよびコード*』セクションで HEURB と HEURC のメッセージを参照してください）。

Adabas はヒューリスティックにトランザクションを終了させる場合でも、トランザクションマネージャが明示的に終了を許可しない限り、そのようなトランザクションを記録しておく必要があります。保留トランザクションが存在しない間には、データベースのデータを変更するユーティリティは実行できますが、Adabas がヒューリスティックに終了したトランザクションが存在する場合には、（ADADBM の NEW_DBID 機能などの）データベースの ID を変更するユーティリティは実行できません。

保留トランザクションのログ情報を長い期間、保持しなければならないことがあります。しかし、WORK1 のリングバッファに常時置いておくことはできないため、WORK1 の不変的な領域、つまり XA 領域（詳細については、ADAPLP の WXA パラメータを参照）にコピーしなければならないことがあります。保留状態のグローバルトランザクションもこの領域がオーバーフローすると、ヒューリスティックに終了されます。

ユーザーキューの取り扱い

XA セッション（xa_open コールで開始する）1 つに対して、ユーザーキューエレメントが 1 つ使用されます。このユーザーキューエレメントをXA マスタユーザーキューエレメントと呼びます。さらに、グローバルトランザクションのブランチ（xa_start コールで開始する）1 つに対して、別のユーザーキューエレメントが 1 つ使用されます。このユーザーキューエレメントを XA スレーブユーザーキューエレメントと呼びます。XA スレーブユーザーキューエレメントは、擬似的なユーザーとしてトランザクションブランチの間、つまり、グローバルトランザクションが最終的に（xa_commit を通じて）コミットされる、（xa_rollback を通じて）ロールバックされる、または（xa_forget を通じて）棄却されるまで動作します。

グローバルトランザクションは、XA 仕様に従って、トランザクション ID によって識別された 1 つ以上のトランザクションブランチから構成されます。つまり、実際には存在しない Adabas ユーザーのトランザクションブランチからトランザクションを構成することが可能です。これは、通常の Adabas トランザクションが、実際に存在する 1 つのユーザーに属するのとは対照的です。このため、スレーブユーザーキューエレメントが1つのグローバルトランザクションそのものか、または一部に相当する場合は、マスタユーザーキューエレメントも実際に存在するユーザーに相当するものとして見なすことができます。

トランザクションブランチをいったんサスペンドし（TM_SUSPEND フラグを設定した状態で xa_end をコール）、再開する（TM_RESUME または TM_MIGRATE フラグを設定した状態で xa_start をコール）と、別のトランザクション ID を指定して xa_start を何回でもコールできるようになります。したがって、1つのマスタユーザーキューエレメントに対して、同時に存在してもよいスレーブユーザーキューエレメントの数は複数になります。その一方で、グローバルトランザクションは、同じトランザクション ID を xa_start に対する複数のコール内で使用する場合、2つ以上のユーザーキューエレメントから構成できます。この場合、このトランザクション ID を持つ xa_prepare コールはすべてのユーザーキューエレメントを保留状態にし、同じトランザクション ID の xa_commit コールや xa_rollback コールによって、グローバルトランザクションを形成するすべてのユーザーキューエレメントを終了します。この場合、複数のスレーブユーザーキューエレメントを同時に消去できます。

XA マスタユーザーキューエレメントはユーザー ID 'xamaster' で識別し、XA スレーブユーザーキューエレメントはユーザー ID 'xaslave' で識別します。さらに、XA スレーブユーザーキューエレメントは、'.X'（'.' は NULL 文字）で始まるログイン ID で識別します。ログイン ID の残り 6 バイトは、XA ライブラリで維持する内部カウンタの役をします。内部カウンタは xa_start にコールするごとに 1 つずつ増えていきます（詳細については、ADAOPR DISPLAY=UQ 機能の出力を参照）。

ともに属するすべてのマスタユーザーキューエレメントとスレーブユーザーキューエレメントは環境固有の ID（UNIX および PC プラットフォームではプロセス ID）が同じになります。マスタユーザーキューエレメントは、グローバルトランザクションが終了（xa_commitやxa_rollback）しても、コマンド ID が消えないようにするために必要です。グローバルトランザクションが終了すると、xa_start コールで生成されたユーザーキューエレメントは除去されます。

スレーブユーザーキューエレメントは、保留状態でない限り、タイムアウト制限と ADAOPR STOP の制約を受けます。一度保留状態になると、前述どおり、ヒューリスティックに終了されない場合は永久に存在し続けます。マスタユーザーキューエレメントは、タイムアウト制限の制約を受けません。xa_close コールを受け取るか、ADAOPR STOP コマンドを使用して消去される（マスタユーザーキューエレメントに属するスレーブが存在しない場合 ADAOPR STOP コマンドがマスタユーザーキューエレメントを消去する）場合にだけ、消失します。

マスタユーザーキューエレメントが停止した場合、関連するコマンド ID すべてが失われます。スレーブユーザーキューエレメントを新規に作成しようとする、次の xa_start コールは、このセッションのマスタユーザーキューエレメントは使用できなくなったことを示す XA_RBTRANSIENT のリターンコードを受け取ります。

マルチプロセストランザクションモニタの使用

マルチプロセスで構成されるトランザクションモニタに XA ライブラリが使用される場合、わずかに異なる操作が必要になります。その場合、実際に存在するユーザーからのコールをトランザクションモニタの異なるワーク処理を介して Adabas に送ることができます。実際に存在するユーザーが最初に提供したコールを Adabas がユニークに識別できることは不可欠なことです。正常な処理で、呼び出し側の派生元のノード名（ノード ID）、ログインユーザー名（ログイン ID）および環境固有の ID（プロセス ID）がユーザーを識別するために使用されます。異なるトランザクションモニタ処理を経由してコールがサブミットされた場合、この識別は行えません（使用される実際に存在するユーザーとは異なるプロセス ID を含んでいます）。

問題を解決するために XA ユーザー出口を使用しなければなりません。ユーザー出口の作成と定義については「*ユーザー出口とハイパー出口*」を参照してください。

XA ユーザー出口は環境変数 XAUEX_0 で識別され、次のインターフェイスを使用します。

```
#include <adabas.h>
int xauex_0 (char *ubuf)
```

ubuf が 8 バイトの文字列を受け取るバッファエリアへのポインタである場合、それは実際に存在するユーザーをユニークに識別します。XAUEX_OK のリターンコードは正常終了を示し、XAUEX_IGNORE のリターンコードはユーザー出口の結果が無視されることを示します。どちらのリターンコードも、XA リターンコード XAER_RMFAIL にマッピングされます。xauex_0 は XA ユーザー出口で使用されるデフォルトの関数名です。セクション「ユーザー出口とハイパー出口」に従って、異なるエントリ関数名を定義することもできます。

XA ユーザー出口は、トランザクションブランチが生成されるごとに xa_start コールの一部として毎回コールされます。ユーザー出口によって返される文字列はユーザーキューエレメントのログイン ID をオーバーレイします。したがって、xa_start への各コールは、XAUEX_0 ユーザー出口を使用して、ログイン ID として返された文字列を示すスレーブユーザーキューエレメントを生成します。カウンタは通常、ログイン ID の一部として維持されますが、その場合、環境固有の ID として維持されます。xa_start コールは、ログイン ID として XA ユーザー出口によって返された文字列を使用し、環境固有の ID のゼロを使用して、このユーザーに対するマスタユー

ザーキューエレメントも暗黙的に生成します。したがって、この環境では、トランザクションモニタの各ワークプロセスおよび実際に存在する各ユーザーに対するマスタユーザーキューエレメントがあります。マスタおよびスレーブユーザーキューエレメントは、ユーザー ID の対応する xamaster または xaslave の文字列によって再度識別されます。このように追加のマスタユーザーキューエレメントは削除されません。xa_close コールは、このようなエレメントには受け入れられないからです（xa_close は、xa_open が始めでコールされたときのユーザーキューエレメントに対してだけコールされます）。

マスタユーザーキューエレメント数が増加してユーザーキューがいっぱいになることを回避するために、Adabas ユーザーインターフェイスに追加ルーチン（remove_xa_context）が実装されています。これは、ユーザーアプリケーションやトランザクションモニタとリンクされます。このルーチンに対するただ 1 つのパラメータはログイン ID を構成する文字列へのポインタです。使用されなくなったマスタユーザーキューエレメントを削除するために、XA ユーザー出口や Adabas アプリケーションなどから、このルーチンをコールできます（ルーチンは、ユーザーキューエレメントを正常に削除した場合は 0 を返し、エラーの場合は -1 を返します）。さらに、マスタユーザーキューエレメントに属するスレーブがない場合は、ADAOPR の STOP コマンドを使用して、マスタユーザーキューエレメントを削除できます。

ニュークリアスパラメータ

XA 機能を使用する場合、ニュークリアスは XA オプション設定で起動する必要があります。このオプションが設定されていないときに XA コールを経由してニュークリアスにアクセスしようとすると Adabas レスポンスコード 230 が返されます。Adabas XA ライブラリは、このリターンコードを XA 仕様に定義されている XAER_PROTO リターンコードにマッピングします。

XA をオンにしてデータベースニュークリアスを実行する場合、前回のニュークリアスセッションから保留しているトランザクションや、ヒューリスティックに完了しているトランザクションが存在しなければ、XA をオフにして後続のセッションを開始できません。

XA 領域は、XA オンのとき WORK1 に割り当てられます。この領域の大きさは、ADANUC の LPXA パラメータで指定します。

XA をコールする、または XA コールを提供するトランザクションモニタを使用する実際に存在する各ユーザーは、マスタユーザーキューエレメントを占有します。マルチプロセストランザクションモニタの場合、1 つのマスタユーザーキューエレメントが各トランザクションモニタ処理に使用され、1 つのマスタユーザーキューエレメントが XA ユーザー出口に返されたユニークな各 ID に使用されます（前述のように remove_xa_context によってまだ削除されていない場合）。加えてすべての場合で、コミットやロールバックされていない各トランザクションブランチは、スレーブユーザーキューエレメントを占有します。保留していたり、ヒューリスティックに終了されたりした各トランザクションは、最低 1 つのキューエレメントを占有します。つまり、XA をオンにしてニュークリアスを実行するときはニュークリアスパラメータ NU（ユーザーキューエレメント数）を増やす必要があります。

XA 機能を使用するアプリケーション

XA 仕様に従って、Adabas は次の情報を公開する必要があります（XA 仕様の詳細については、セクション 7.2 を参照）。

- XA 仕様は、エントリポイントと他の情報をトランザクションマネージャに与える、構造体 xa_switch_t を定義します。該当する Adabas 構造体の名前は adaxasw です。
- リソースマネージャの名前を指定する、リソースマネージャスイッチ内のテキスト文字列は "ADABAS" です。
- xa_open コールは、「dbid=<dbid>」 形式の情報を文字列で受け取ります。<dbid> は使用する Adabas データベースのデータベース ID です。xa_close コールは空の文字列でも構いません。
- DTP（分散トランザクション処理）でない操作と次の違いがあります。
  - ET データは使用できません。これは、xa_commit コールを経由して Adabas に渡す方法がないことが理由です。
  - ET/BT に対するマルチフェッチオプションはサポートしません。

制御のスレッドとみなされるオペレーティングシステムのエンティティは、オペレーティングシステムのプロセスです。Adabas XA インターフェイスは基礎的なオペレーティングシステムのマルチスレッド機能のいかなる使用もサポートしていません。

XA 仕様では、リソースマネージャの実装者に実装の選択の余地を与えています。Adabas に対して次の選択ができます。

- 読み取り専用の最適化が実装されています。つまり、データベース更新コマンドを発行しないトランザクションに xa_prepare が呼び出された場合（スレーブユーザーキューエレメントは ET 状態のまま）、トランザクションは直ちに実行され、レスポンス XA_RDONLY が返されます（xa_commit や xa_rollback に対する個別のコールはなし）。
- 関連付けの移行はサポートしません。これは、中断された関連付け（TMSUSPEND および TMMIGRATE フラグを設定して xa_end で作成した）は、中断が生じたスレッド以外の制御スレッドで再開することができることを意味します（この場合、TMRESUME フラグを設定して xa_start をコールしなければなりません）。
- Adabas はトランザクションマネージャを使用してダイナミックに登録しないので、トランザクションマネージャは各グローバルトランザクションに xa_start をコールしなければなりません。
- 非同期コールモードはサポートしません。すべての XA コールはブロックされています。
- 前述どおり、Adabas による自己判断で終了が行われます。
- Adabas は xa_start_2 コールをサポートしません。

XA 機能は Adabas XA インターフェイスを通じてのみ利用できます。これは、グローバルトランザクションを開始または終了するための Adabas ダイレクトコールは存在しないことを意味します。しかし、XA インターフェイスは既存の Adabas ダイレクトコールと連携します。Adabas トランザクションロジック（BT、CL、ET、OP）に影響を与えるダイレクトコールは、グローバルトランザクションがアクティブ（レスポンスコード 230）の場合は認められません。