

Oracle TimesTen In-Memory Database

追加情報

リリース 7.0.4

E06116-01



Oracle TimesTen In-Memory Database 追加情報, リリース 7.0.4

部品番号: E06116-01

原本名: Oracle TimesTen In-Memory Database Documentation Addendum Release 7.0.4

Copyright ©1996, 2008, Oracle. All rights reserved.

制限付権利の説明

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権と工業所有権に関する法律により保護されています。

独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバース・エンジニアリング、逆アセンブル、逆コンパイル等は禁止されています。

このドキュメントの情報は、予告なしに変更される場合があります。オラクル社およびその関連会社は、このドキュメントに誤りが無いことの保証は致し兼ねます。これらのプログラムのライセンス契約で許諾されている場合を除き、プログラムを形式、手段（電子的または機械的）、目的に関係なく、複製または転用することはできません。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の注意が適用されます。

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションへの用途を目的としておりません。このプログラムをかかるとして使用する際、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。万一かかるプログラムの使用に起因して損害が発生いたしましても、オラクル社およびその関連会社は一切責任を負いかねます。

Oracle, JD Edwards, PeopleSoft, Siebel, TimesTen, TimesTen アイコン、

MicroLogging, Direct Data Access は米国 Oracle Corporation およびその子会社、関連会社の登録商標です。その他の名称は、他社の商標の可能性があり得ます。

このプログラムは、第三者の Web サイトへリンクし、第三者のコンテンツ、製品、サービスへアクセスすることがあります。オラクル社およびその関連会社は第三者の Web サイトで提供されるコンテンツについては、一切の責任を負いかねます。当該コンテンツの利用は、お客様の責任になります。第三者の製品またはサービスを購入する場合は、第三者と直接の取引となります。オラクル社およびその関連会社は、第三者の製品およびサービスの品質、契約の履行（製品またはサービスの提供、保証義務を含む）に関しては責任を負いかねます。また、第三者との取引により損失や損害が発生いたしましても、オラクル社およびその関連会社は一切の責任を負いかねます。

2008年3月

Printed in the United States of America

目次

このマニュアルについて

このマニュアルで使用されている表記規則.....	2
テクニカル・サポート.....	3

1 プラットフォーム・サポートとインストールの前提条件

プラットフォーム・サポート.....	6
JDK サポート.....	6
Cache Connect to Oracle.....	8
HP-UX プラットフォームでのインストールの前提条件.....	8
Veritas ファイル・システム.....	8
共有メモリーのデーモン・オプション.....	9

2 AWT キャッシュ・グループを使用したアクティブ・スタンバイ・ペア・レプリケーションによる障害時リカバリ

概要.....	12
アクティブ・スタンバイ・ペアでの障害時リカバリ・サブスクリバ使用時の要件.....	13
障害時リカバリ・サブスクリバの作成.....	13
障害時リカバリ・サイトへのスイッチオーバー.....	15
障害時リカバリ・サイトにスイッチオーバーした後の新しいアクティブ・スタンバイ・ペアの作成.....	15
単一のデータ・ストアへのスイッチオーバー.....	16
プライマリ・サイトの元の構成に戻す.....	17

3 問合せのしきい値に関する警告

概要.....	20
SNMP トラップ.....	21
警告メッセージ.....	21
問合せのしきい値の設定.....	21
一般的な QueryThreshold 接続属性.....	22
ttIsql を使用した問合せのしきい値の設定.....	22
ODBC オプションを使用した問合せのしきい値の設定.....	23
JDBC での問合せのしきい値の設定.....	23
TTClasses を使用した問合せのしきい値の設定.....	24
問合せのしきい値の取得.....	24
ttIsql を使用した QueryThreshold 属性の取得.....	24
TT_QUERY_THRESHOLD ODBC オプションの取得.....	24

JDBC を使用した問合せのしきい値の取得	24
TTCclasses を使用した問合せのしきい値の取得	24
レプリケーションのための問合せのしきい値の設定	25
ttRepQueryThresholdSet	25
ttAdmin -repQueryThresholdSet	26
レプリケーションのための問合せのしきい値の取得	26
ttRepQueryThresholdGet	26
ttAdmin -repQueryThresholdGet	27

4 AWT キャッシュ・グループのパフォーマンスの監視

概要	30
AWT 監視の有効化	30
ttCacheAwtMonitorConfig	30
AWT 監視結果の表示	31
ttRepAdmin -showstatus -awtmoninfo	31

5 Oracle での自動リフレッシュ・オブジェクトのクリーンアップ

概要	34
自動リフレッシュのクリーンアップ・スクリプトの使用	34

6 アクティブ・スタンバイ・ペアのデュアル・アクティブ・マスターの検出

7 トランザクション・ログ API (XLA)

8 新しいエラー・メッセージおよび SNMP トラップ

エラー・メッセージ	42
SNMP トラップ	46

索引

このマニュアルについて

このマニュアルでは、Oracle TimesTen In-Memory Database リリース 7.0.4 で追加された機能について説明します。Oracle TimesTen In-Memory Database の詳細については、リリース 7.0.3 のドキュメント・セットを参照してください。

このマニュアルで使用されている表記規則

TimesTen は複数のプラットフォームをサポートしています。このマニュアルに記述されている内容は、特に説明がないかぎり、サポートされるすべてのプラットフォームに適用されます。Windows は Windows 2000、Windows XP および Windows Server 2003 を表します。UNIX は Solaris、Linux、HP-UX、Tru64 および AIX を表します。

このマニュアルの表記規則は、次のとおりです。

表記	意味
固定幅フォント	コード例、ファイル名、パス名などです。 例: <code>.odbc.ini</code> または <code>ttconnect.ini</code> ファイル
固定幅フォントのイタリック体	コード例において置き換える必要がある変数です。 例: <code>Driver=install_dir/lib/libtten.sl</code> <code>install_dir</code> をご使用の TimesTen インストール・ディレクトリに置き換えます。

このマニュアルでは、コマンドラインの例および説明に次の変数を使用します。

表記	意味
固定幅フォントのイタリック体	置き換える必要がある変数です。たとえば、組込みプロシージャのパラメータ値などは、値を一重引用符 (' ') で囲む必要がある場合もあります。
[]	大カッコは、カッコ内の項目を任意に選択することを表します。大カッコは、入力しないでください。
{ }	中カッコは、カッコ内の項目のうち、1 つが必須であることを表します。中カッコは、入力しないでください。
	縦線は、大カッコまたは中カッコ内の複数の選択項目の区切りに使用します。項目のうち 1 つを入力します。縦線は、入力しないでください。
...	引数の後の水平の省略記号は、1 行のコマンドラインで複数の引数を使用できることを表します。
%	% は、UNIX のシェル・プロンプトを表します。
#	# は、UNIX のルート・プロンプトを表します。

このマニュアルでは、次の変数を使用して、パス、ファイルおよびユーザー名を表します。

表記	意味
<code>install_dir</code>	TimesTen の現在のリリースがインストールされているディレクトリのパスです。
<code>TTinstance</code>	TimesTen のインストールで指定するインスタンス名です。インストール時に、英数字の一意のインスタンス名で TimesTen の各インストールを指定する必要があります。この名前はインストール・パスに表示されます。このマニュアルでは、例としてインスタンス名「giraffe」を使用しています。
<code>bits</code> または <code>bb</code>	32 または 64 の 2 桁の数字は、32-bit または 64-bit のオペレーティング・システムを表します。
<code>release</code> または <code>rr</code>	冒頭の 2 桁は、TimesTen の現在のリリース番号を示します。ドットが付く場合と付かない場合があります。例: 70 または 7.0 は、TimesTen リリース 7.0 を表します。
<code>jdk_version</code>	2 桁の数字は、JDK のメジャー・リリースのバージョン番号を表します。たとえば、JDK 1.4 は 14 と表します。JDK 5 は 5 と表します。
<code>timesten</code>	TimesTen のインスタンス管理者を示す名前の例です。TimesTen の管理者名に、適切な名前を任意に使用することができます。Windows の場合、TimesTen のインスタンス管理者は Administrators グループのメンバーである必要があります。すべての TimesTen インスタンスに、一意のインスタンス管理名があります。
<code>DSN</code>	データソースの名前です。

テクニカル・サポート

TimesTen 製品のテクニカル・サポートに関するお問合せ先は次のとおりです。

<http://www.oracle.co.jp/support/>

プラットフォーム・サポートとインストールの前提条件

この章では、Oracle TimesTen In-Memory Database リリース 7.0.4.0.0 に必要なプラットフォーム・サポートとインストールの前提条件に対する、変更点と追加情報を示します。まだ TimesTen をインストールしていない場合、またはご使用のプラットフォームへのインストールまたは構成に関して不明な点がある場合は、まず『Oracle TimesTen In-Memory Database インストレーション・ガイド』を読むことをお勧めします。ここでは、次の項目に対する変更点および追加情報を示します。

- [JDK サポート](#)
- [Cache Connect to Oracle](#)
- [HP-UX プラットフォームでのインストールの前提条件](#)

プラットフォーム・サポート

JDK サポート

7.0.4 の Oracle TimesTen In-Memory Database では、新しく JDK 6 がサポートされています。ここでは、JDK サポートの全体をわかりやすく表に示します。

注意：この表は、『Oracle TimesTen In-Memory Database インストレーション・ガイド』に示す表よりも優先です。

TimesTen では、示されたプラットフォームで次の JDK がサポートされます。

環境	JDK 1.4	JDK 5.0	JDK 6.0
Asianux 2.0 Intel IA-32	可 (Sun および BEA JRockit)	可 (Sun および BEA JRockit)	可 (Sun および BEA JRockit)
Asianux 2.0 EM64T および AMD Opteron CPU (32-bit および 64-bit)		可 (Sun および BEA JRockit)	可 (Sun および BEA JRockit)
HP-UX 11i, 11i v2 および 11iv3 PA-RISC (32-bit および 64-bit)	可 (HP 1.4.2)	可 (HP)	
HP-UX 11i v2 および 11iv3 Itanium2 (32-bit および 64-bit)	可 (HP 1.4.2)	可 (HP)	
HP-UX Tru64 UNIX 5.1B AlphaChip EV68 CPU	可 (HP)		
IBM AIX 5L 5.2 および 5.3 POWER CPU (32-bit および 64-bit)	可 (IBM 1.4.2 以上)	可 (IBM)	
Microsoft Windows 2000、Windows XP および Windows Server 2003 Intel IA-32、EM64T および AMD Opteron CPU (32-bit および 64-bit)	可 (Sun 32-bit)	可 (Sun および BEA JRockit)	可 (Sun および BEA JRockit)
MontaVista Linux Carrier Grade Edition リリース 4.0 Intel IA-32 および EM64T CPU (32-bit および 64-bit)	可 (Sun 32-bit)	可 (Sun および BEA JRockit)	可 (Sun および BEA JRockit)

環境	JDK 1.4	JDK 5.0	JDK 6.0
Oracle Enterprise Linux 4 および 5 Intel IA-32 CPU	可 (Sun)	可 (Sun および BEA JRockit)	可 (Sun および BEA JRockit)
Oracle Enterprise Linux 4 および 5 EM64T および AMD Opteron CPU (32-bit および 64-bit)		可 (Sun および BEA JRockit)	可 (Sun および BEA JRockit)
Red Hat Enterprise Linux 3、4 および 5 Intel IA-32	可 (Sun)	可 (Sun および BEA JRockit)	可 (Sun および BEA JRockit)
Red Hat Enterprise Linux 3、4 および 5 Intel Itanium2 CPU	可 (Sun)	可 (Sun および BEA JRockit)	
Red Hat Enterprise Linux 3、4 および 5 EM64T および AMD Opteron CPU (32-bit および 64-bit)		可 (Sun および BEA JRockit)	可 (Sun および BEA JRockit)
Solaris 8、9 および 10 UltraSparc CPU (32-bit および 64-bit)	可 (Sun)	可 (Sun および BEA JRockit 64-bit)	可 (Sun)
Solaris 10 AMD Opteron CPU (32-bit および 64-bit)		可 (Sun)	可 (Sun)
SuSE LINUX Enterprise Server 9 および 10 Intel IA-32、EM64T および AMD Opteron CPU	可 (Sun 32-bit)	可 (Sun および BEA JRockit)	可 (Sun および BEA JRockit)
SuSE LINUX Enterprise Server 10 Itanium2 CPU	可 (Sun)	可 (BEA JRockit)	

Cache Connect to Oracle

TimesTen Cache Connect to Oracle を使用すると、Oracle Database のデータを TimesTen にキャッシュできます。次のリリースの Oracle クライアントおよびサーバーが TimesTen 7.0.4 のこのオプションでサポートされています。

- Oracle Database 9i リリース 2 (Oracle 9.2.0.8.0 以上)
- Oracle Database 10g リリース 1 (Oracle 10.1.0.5.0 以上)
- Oracle Database 10g リリース 2 (Oracle 10.2.0.1.0 以上)
- 次のプラットフォームの Oracle Database 11g リリース 1:
 - Microsoft Windows (32-bit)
 - Linux (32-bit および 64-bit)

MontaVista CGE 4.0 で TimesTen Cache Connect to Oracle を使用するには、Linux 版の Oracle Instant Client 10.2.0.3 または 11.1.0.6.0 をインストールしてください。

HP-UX プラットフォームでのインストールの前提条件

Veritas ファイル・システム

TimesTen をインストールする前に、該当する要件がご使用のオペレーティング・システムで満たされていることを確認します。

ファイル・システム・オプション

Veritas ファイル・システムでは、オプション `discovered_direct_iosz` および `max_direct_iosz` を 3MB に設定します。

ダイレクト I/O がこのように設定されていないと、TimesTen 操作におけるファイル・システムのパフォーマンスが低下する可能性があります。

これらのオプションを設定するには、`root` としてログインし、次のコマンドを使用します。

```
# /usr/sbin/vxtunefs -o discovered_direct_iosz=3145728
# /usr/sbin/vxtunefs -o max_direct_iosz=3145728
```

注意: `vxtunefs` オンライン・オプションを使用するには、Advanced VxFS が必要です。

Veritas ファイル・システムで TimesTen を実行する場合の詳細は、『Oracle TimesTen In-Memory Database インストレーション・ガイド』のインストールの前提条件に関する項を参照してください。

共有メモリーのデーモン・オプション

今回のリリースでは、HP-UX ccNUMA システムで TimesTen データ・ストアが格納される共有メモリー・セグメントの場所を指定できるデーモン・オプションが新しく追加されています。ccNUMA システムには、データの場所によって不規則なメモリー待機時間があります。リモートのセルにあるデータへのアクセスは、ローカルのセルにあるデータへのアクセスよりも時間がかかります。

TimesTen の操作結果が最適になるようにするには、IPC_MEM_LOCAL を設定し、TimesTen の処理をローカルのセルに限定します。

共有メモリー・セグメントの場所に関するヒントを設定するには、ttendaemon.options ファイルで次の行を個別の行で追加します。

```
-shmLocalityHint locality_hint
```

locality_hint に有効な値は次のとおりです。

- IPC_MEM_LOCAL
- IPC_MEM_INTERLEAVED
- IPC_MEM_FIRST_TOUCH
- IPC_MEM_STRIPED

指定できる値の文字列は 1 つのみです。場所に関するヒントを指定すると、TimesTen では、そのヒントを使用して、インスタンスのすべてのデータ・ストアの共有メモリー・セグメントを作成しようとします。

注意：このオプションは、インスタンス管理者がメモリー・リソースへのアクセス権を持つ場合にのみ有効になります。

ヒントのセマンティクスについては、shmget () の man ページを参照してください。デフォルトでは、ヒントを使用せずにセグメントが作成されます。デーモン・オプションが指定されていない場合、または正しく指定されていない場合は、デフォルトの動作になります。セグメントがヒントを使用して作成されているかどうかを確認するには、HP-UX の pstat () の機能を使用します。pstat () については、HP-UX の man ページを参照してください。

TimesTen デーモン・オプションの概要については、『Oracle TimesTen In-Memory Database オペレーション・ガイド』の TimesTen デーモン・オプションの管理に関する項を参照してください。

AWT キャッシュ・グループを使用したアクティブ・スタンバイ・ペア・レプリケーションによる障害時リカバリ

TimesTen 7.0.4.0.0 では、アクティブ・スタンバイ・ペアのレプリケーション・スキームで使用する特別なサブスライバがサポートされるようになりました。このサブスライバを使用すると、非同期の WRITETHROUGH (AWT) キャッシュ・グループを使用して Oracle Database に更新を伝播できます。この章では、リモートの障害時リカバリ・サイトでこのサブスライバをデプロイおよび使用する方法を説明します。内容は次のとおりです。

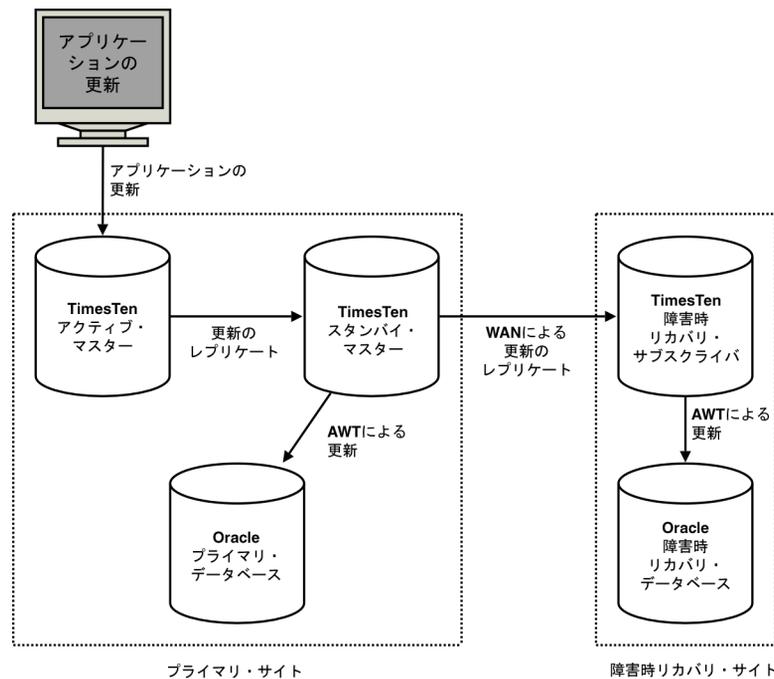
- [概要](#)
- [障害時リカバリ・サブスライバの作成](#)
- [障害時リカバリ・サイトへのスイッチオーバー](#)
- [プライマリ・サイトの元の構成に戻す](#)

概要

TimesTen のアクティブ・スタンバイ・ペア・レプリケーションを使用すると、データ・センター内でデータ・ストアを高速に切り替えることが可能になるため、高可用性を得ることができます。また、AWT キャッシュ・グループを使用して Oracle Database に変更を伝播するデータ・ストアを自動的に変更することも可能です（詳細は、『Oracle TimesTen Replication - TimesTen to TimesTen 開発者および管理者ガイド』のアクティブ・スタンバイ・ペアにおける

ASYNCHRONOUS WRITETHROUGH キャッシュ・グループに関する項を参照）。ただし、データ・センターをまたいで更なる高可用性を得るには、サイト全体の障害からのリカバリが可能である必要があります。このような障害には、アクティブ・スタンバイ・ペアの TimesTen マスター・データ・ストアとキャッシュ・グループで使用する Oracle Database の両方の障害が含まれます。

図 2.1 障害時リカバリ・サブスライバを使用したアクティブ・スタンバイ・ペア



アクティブ・スタンバイ・ペア・レプリケーション構成の一部として読取り専用の特別な障害時リカバリ・サブスライバを作成すると、障害全体をリカバリできます。この特別なサブスライバはリモートの障害時リカバリ・サイトに存在し、更新をもう 1 つの Oracle Database に伝播できます。この Oracle Database も障害時リカバリ・サイトに存在します。障害時リカバリ・サブスライバは、プライマリ・サイト全体に障害が発生した場合に、障害時リカバリ・サイトで新しいアクティブ・スタンバイ・ペアのアクティブ・マスターとして処理を引き継ぎます。いずれのアプリケーションも、その後は障害時リカバリ・サイトに接続し、サービスの中断を最小限に抑えて操作を続行します。

アクティブ・スタンバイ・ペアでの障害時リカバリ・サブスクリバ使用時の要件

障害時リカバリ・サブスクリバを使用するには、次のことが必要です。

- プライマリ・サイトでは、AWT キャッシュ・グループを含むアクティブ・スタンバイ・ペア構成を使用する必要があります。『Oracle TimesTen Replication - TimesTen to TimesTen 開発者および管理者ガイド』のアクティブ・スタンバイ・ペアの設定に関する項を参照してください。
- プライマリ・サイトから障害時リカバリ・サイトへの継続的な WAN 接続が必要です。この接続には、標準的なボリュームのトランザクションが適切な速度で障害時リカバリ・サブスクリバにレプリケートされるように、必要最小限の帯域幅が必要です。
- プライマリ・サイトのデータベースと同じスキーマを持つ表を含めるために、障害時リカバリ・サイトに Oracle Database が構成されている必要があります。このデータベースは、プライマリ・サイトからレプリケートされた更新を獲得することのみを目的としており、障害時リカバリ・サブスクリバが作成されたときにキャッシュ・グループによって書き込まれたデータが表に存在する場合、このデータは削除されることに注意してください。
- プライマリ・サイトと障害時リカバリ・サイトの両方で、キャッシュ・グループ管理者の同じユーザー ID とパスワードを使用する必要があります。

必須ではありませんが、障害時リカバリ・サイトに、TimesTen データ・ストアをもう 1 つ構成することをお勧めします。このデータ・ストアは、プライマリ・サイトで障害が発生した後、障害時リカバリ・サブスクリバがアクティブ・マスター・データ・ストアになったときのスタンバイ・マスター・データ・ストアとして動作します。

障害時リカバリ・サブスクリバの作成

障害時リカバリ・サブスクリバを作成するには、次の手順を実行します。

1. プライマリ・サイトで、AWT キャッシュ・グループを含むアクティブ・スタンバイ・ペアを作成します。『Oracle TimesTen Replication - TimesTen to TimesTen 開発者および管理者ガイド』のアクティブ・スタンバイ・ペアの設定に関する項を参照してください。
2. **ttRepAdmin** コマンドに `-duplicate` および `-cacheInitDR` オプションを指定して、障害時リカバリ・サイトで障害時リカバリ・サブスクリバを作成します。`-cacheUid` および `-cachePwd` オプションを使用して、障害時リカバリ・サイトでも Oracle Database のキャッシュ・グループ管理者とパスワードを指定する必要があります。

データ・ストアに複数のキャッシュ・グループが含まれる場合は、キャッシュ・グループを並行してフラッシュするために起動されるスレッド数を指定する `-nThreads` オプションを使用して、複製操作の効率を上げることもできます。各スレッドは、キャッシュ・グループ全体を Oracle にフラッシュした後、フラッシュするキャッシュ・グループがまだある場合は、次のキャッシュ・グループに移動します。

-nThreads に値を指定しないと、1つのフラッシュ・スレッドのみが起動されま
す。

たとえば、ホスト名が **primary**、キャッシュ・ユーザー ID が **system**、パスワードが **manager** のシステムで、スタンバイ・マスター・データ・ストア **mast2** を
障害時リカバリ・サブスライバ **drsub** に複製するために、キャッシュ・グ
ループのフラッシュ・スレッドを2つを使用するには、次のコマンドを使用しま
す。

```
ttRepAdmin -duplicate -from mast2 -host primary -cacheInitDR  
-nThreads 2 -cacheUid system -cachePwd manager drsub
```

C で **ttRepDuplicateEx** 関数を使用する場合は、**ttRepDuplicateExArg.flags** で
TT_REPDUPE_INITCACHEDR フラグを設定する必要があります。また、オプ
ションで **ttRepDuplicateExArg.nThreads4InitDR** の値を指定できます。

```
int rc;  
ttUtilHandle utilHandle;  
ttRepDuplicateExArg arg;  
memset( &arg, 0, sizeof( arg ) );  
arg.size = sizeof( ttRepDuplicateExArg );  
arg.flags = TT_REPDUPE_INITCACHEDR;  
arg.nThreads4InitDR = 2;  
arg.cacheuid = "system";  
arg.cachepwd = "manager";  
arg.localHost = "disaster";  
rc = ttRepDuplicateEx( utilHandle, "DSN=drsub",  
"mast2", "primary", &arg );
```

サブスライバを複製した後、**TimesTen** では自動的に非同期の
WRITETHROUGH レプリケーション・スキームが構成されます。このスキーム
では、キャッシュ・グループでの更新が **Oracle Database** に伝播され、**TimesTen**
のキャッシュ・グループに対応する **Oracle Database** の表が切り捨てられた後、
キャッシュ・グループのすべてのデータが **Oracle Database** にフラッシュされま
す。

3. **ttRepStart** プロシージャ、またはオプション **-repstart** を指定した **ttAdmin** コ
マンドを使用して、障害時リカバリ・サブスライバのレプリケーション・エー
ジェントを起動します。たとえば、次のようになります。

```
ttAdmin -repstart drsub
```

これで、更新はスタンバイ・マスター・データ・ストアから障害時リカバリ・サ
ブスライバにレプリケートされた後、障害時リカバリ・サイトの **Oracle**
Database に伝播されます。

障害時リカバリ・サイトへのスイッチオーバー

プライマリ・サイトで障害が発生した場合は、次のいずれかの方法で障害時リカバリ・サイトにスイッチオーバーできます。障害時リカバリ・サイトでのデータ損失の危険性を最小限に抑えることが目的の場合は、障害時リカバリ・サブスクライバをアクティブ・マスター・データ・ストアとして使用して、新しいアクティブ・スタンバイ・ペアを作成します。後で障害時リカバリ・データ・ストアで障害が発生した場合のデータ損失の危険性はあっても、アプリケーションの停止時間を最小限に抑えることが目的である場合は、障害時リカバリ・サブスクライバからレプリケーション・スキームを削除し、それを単一の非レプリケーション・データ・ストアとして使用する方法もあります。後で、障害時リカバリ・サイトでアクティブ・スタンバイ・ペアをデプロイできます。

障害時リカバリ・サイトにスイッチオーバーした後の新しいアクティブ・スタンバイ・ペアの作成

1. 読取り専用のアプリケーションであれば、すぐに障害時リカバリ・サブスクライバにリダイレクトできます。データ・ストアを更新するアプリケーションのリダイレクトは手順7で行います。

2. キャッシュ・グループに最近行ったすべての更新が、**ttRepSubscriberWait** プロシージャ、または `-wait` オプションを指定した **ttRepAdmin** コマンドを使用して Oracle Database に伝播されていることを確認します。

```
ttRepSubscriberWait( null, null, '_ORACLE', null, 600 );
```

ttRepSubscriberWait が 0x01 を戻す (タイムアウトを示す) 場合は、手順3に進む前に、なぜキャッシュ・グループが伝播を完了しなかったかを調査する必要があります。

3. **ttRepStop** プロシージャ、または `-repstop` オプションを指定した **ttAdmin** コマンドを使用して、障害時リカバリ・サブスクライバのレプリケーション・エージェントを停止します。たとえば、サブスクライバ `drsub` のレプリケーション・エージェントを停止するには、次のコマンドを使用します。

```
call ttRepStop;
```

4. **DROP ACTIVE STANDBY PAIR** 文を使用して、サブスクライバのアクティブ・スタンバイ・ペアのレプリケーション・スキームを削除します。たとえば、次のようになります。

```
DROP ACTIVE STANDBY PAIR;
```

5. **CREATE ACTIVE STANDBY PAIR** 文を使用して、新しいアクティブ・スタンバイ・ペアのレプリケーション・スキームを作成します。このとき、障害時リカバリ・サブスクライバをアクティブ・マスター・データ・ストアとして指定します。たとえば、前述のサブスクライバ `drsub` をアクティブ・マスター、新しいデータ・ストア `drstandby` をスタンバイ・マスターとして指定し、**RETURN TWOSAFE RETURN** サービスを使用して新しいアクティブ・スタンバイ・ペアを作成するには、次のコマンドを使用します。

```
CREATE ACTIVE STANDBY PAIR drsub, drstandby RETURN TWOSAFE;
```

6. **ttRepStateSet** プロシージャを使用して、新しいアクティブ・スタンバイ・マスター・データ・ストアの状態を **ACTIVE** に設定します。たとえば、この例のデータ・ストア **drsub** で次のように実行します。

```
call ttRepStateSet( 'ACTIVE' );
```
7. **TimesTen** データ・ストアに書込みを行う必要があるアプリケーションは、新しいアクティブ・マスター・データ・ストアにリダイレクトされるようになります。
8. 『Oracle TimesTen Replication - TimesTen to TimesTen 開発者および管理者ガイド』のアクティブ・スタンバイ・ペアの設定に関する項に従って、手順 6 以降のアクティブ・スタンバイ・ペアの作成を続行します。

単一のデータ・ストアへのスイッチオーバー

1. 読取り専用のアプリケーションであれば、すぐに障害時リカバリ・サブスクリバにリダイレクトできます。データ・ストアを更新するアプリケーションのリダイレクトは手順 5 で行います。
2. **ttRepStop** プロシージャ、または **-repstop** オプションを指定した **ttAdmin** コマンドを使用して、障害時リカバリ・サブスクリバのレプリケーション・エージェントを停止します。たとえば、サブスクリバ **drsub** のレプリケーション・エージェントを停止するには、次のコマンドを使用します。

```
call ttRepStop;
```
3. **DROP ACTIVE STANDBY PAIR** 文を使用して、サブスクリバのアクティブ・スタンバイ・ペアのレプリケーション・スキームを削除します。たとえば、次のようになります。

```
DROP ACTIVE STANDBY PAIR;
```
4. 構成されているアクティブ・スタンバイ・ペアはありませんが、AWT キャッシュ・グループではレプリケーション・エージェントを起動する必要があります。**ttRepStart** プロシージャ、またはオプション **-repstart** を指定した **ttAdmin** コマンドを使用して、データ・ストアのレプリケーション・エージェントを起動します。たとえば、データ・ストア **drsub** のレプリケーション・エージェントを起動するには、次のコマンドを使用します。

```
call ttRepStart;
```
5. **TimesTen** データ・ストアに書込みを行う必要があるアプリケーションは、このデータ・ストアにリダイレクトされるようになります。

注意: 後で、障害時リカバリ・サイトでアクティブ・スタンバイ・ペアを作成することもできます。これには、[15 ページの「障害時リカバリ・サイトにスイッチオーバーした後の新しいアクティブ・スタンバイ・ペアの作成」](#)の手順 2 以降 (手順 4 を除く) を実行します。

プライマリ・サイトの元の構成に戻す

再度プライマリ・サイトが使用可能になったときに、動作中のアクティブ・スタンバイ・ペアを障害時リカバリ・サイトからプライマリ・サイトに戻す場合があります。サービスの中断を最小限に抑えてこれを行うには、元の障害時リカバリ・サイトの作成とこのサイトへのスイッチオーバーに使用したプロセスを逆に実行します。次の手順を実行します。

1. 必要に応じて、**ttDestroy** コマンドを使用して、プライマリ・サイトで元のアクティブ・マスター・データ・ストアを破棄します。たとえば、**mast1** というデータ・ストアを破棄するには、次のコマンドを使用します。

```
ttDestroy mast1
```
2. 「[障害時リカバリ・サブスクリバの作成](#)」の手順に従って、プライマリ・サイトで障害時リカバリ・サブスクリバを作成します。新しい障害時リカバリ・サブスクリバに元のアクティブ・マスター・データ・ストアを使用します。
3. 「[障害時リカバリ・サイトへのスイッチオーバー](#)」に従って、プライマリ・サイトで新しい障害時リカバリ・サブスクリバにスイッチオーバーします。スタンバイ・マスター・データ・ストアも作成します。
4. 「[障害時リカバリ・サブスクリバの作成](#)」に従って、障害時リカバリ・サイトで新しい障害時リカバリ・サブスクリバを作成します。

3

問合せのしきい値に関する警告

この章の内容は次のとおりです。

- [概要](#)
- [問合せのしきい値の設定](#)
- [問合せのしきい値の取得](#)
- [レプリケーションのための問合せのしきい値の設定](#)
- [レプリケーションのための問合せのしきい値の取得](#)

概要

TimesTen では、SQL 文の実行時間が指定した期間を超えたときにサポート・ログに警告を書き込み、SNMP トラップをスローするように構成できます。この場合も実行は継続され、問合せのしきい値による影響は受けません。

問合せのしきい値は、次の ODBC および JDBC コールに適用されます。

- `SQLExecute`、`SQLExecDirect` および `SQLFetch` ODBC 関数
- JDBC Statement インタフェースの `execute()`、`executeBatch()`、`executeQuery()` および `executeUpdate()` メソッド
- JDBC ResultSet インタフェースの `next()` メソッド

つまり、問合せのしきい値は、プロシージャ・コールおよび SQL 文に適用されます。

次のいずれかの方法で、問合せのしきい値を設定します。

- ODBC.INI ファイルまたは `ttIsql` ユーティリティを使用して、一般的な `QueryThreshold` 接続属性を設定します。
- `SQLSetConnectOption` および `SQLSetStmtOption` ODBC 関数を使用して、`TT_QUERY_THRESHOLD` オプションを設定します。
- JDBC 接続 URL で `QueryThreshold` 属性を設定します。
- `TimesTenStatement` インタフェースの `setQueryTimeThreshold` メソッドを使用します。
- `TTCmd::setQueryThreshold` `TTClasses` メソッドを使用します。

次のいずれかの方法で、問合せのしきい値を取得します。

- `ttIsql` ユーティリティ
- `SQLGetConnectOption` または `SQLGetStmtOption` ODBC 関数を使用します。
- `TTCmd::getQueryThreshold` `TTClasses` メソッドを使用します。

レプリケーション・エージェントでは、ODBC.INI ファイルの `QueryThreshold` 設定を継承しません。`ttRepQueryThresholdSet` 組込みプロシージャを使用して、レプリケーション・エージェントの問合せのしきい値を設定します。または、`ttAdmin` ユーティリティの `-repQueryThresholdSet` オプションを指定することもできます。レプリケーション・エージェントの問合せのしきい値は、マテリアライズド・ビューのディテール表での SQL 実行、ON DELETE CASCADE 操作、および SQL 文を実行する一部の内部操作に適用されます。

レプリケーションの問合せのしきい値は、`ttRepQueryThresholdGet` 組込みプロシージャ、または `ttAdmin` ユーティリティの `-repQueryThresholdGet` オプションを使用して取得できます。

キャッシュ・エージェントが実行する SQL 文の問合せのしきい値は設定できません。

SNMP トラップ

SNMP トラップの名前は `ttQueryThresholdWarnTrap` です。SNMP トラップの構成の詳細は、『Oracle TimesTen In-Memory Database エラー・メッセージおよび SNMP トラップ』を参照してください。

警告メッセージ

TimesTen では、問合せのしきい値を超えると、サポート・ログに警告メッセージが書き込まれます。

例 3.1 サポート・ログの警告メッセージの例を次に示します。

```
2007-12-03 15:55:33.67 Warn:      : 9692: 7373/0x95c5038: QT rep1 7373
DSN=rep1 Transaction Id=2 Transaction Counter=34 Statement Id=1 SQL
statement execution exceeded the QueryThreshold value of 10 seconds.
Statement=select count(*) from a,t;
```

警告メッセージの内容は次のとおりです。

- タイムスタンプ : 2007-12-03 15:55:33.67
- デーモン・プロセス ID: 9692
- SQL 文を発行したプロセスの ID: 7373
- 接続名 : rep1 (「QT」の後)。(レプリケーション・エージェントの接続名は RECEIVER です。)
- DSN: rep1
- トランザクション ID、トランザクション・カウンタおよび文 ID:
Transaction Id=2 Transaction Counter=34 Statement Id=1
- 問合せのしきい値 (秒) : 10
- SQL 文 : `select count(*) from a,t;`
- SQL 文のいずれかの入力パラメータの値

SQL 文が警告メッセージの長さの上限を超えると、文は後続のメッセージで継続されます。警告メッセージの長さをユーザーが設定することはできません。

問合せのしきい値の設定

TimesTen がサポート・ログに警告を書き込み、SNMP トラップをスローするまで、SQL 文が実行される秒数を指定できます。この場合も実行は継続され、問合せのしきい値による影響は受けません。

次のいずれかの方法で、問合せのしきい値を設定します。

- 一般的な `QueryThreshold` 接続属性
- `ttlsql` を使用した問合せのしきい値の設定
- ODBC オプションを使用した問合せのしきい値の設定
- JDBC での問合せのしきい値の設定
- `TTClasses` を使用した問合せのしきい値の設定

レプリケーション・エージェントが実行する問合せについては、[25 ページの「レプリケーションのための問合せのしきい値の設定」](#)を参照してください。

キャッシュ・エージェントが実行する問合せのしきい値は設定できません。

一般的な QueryThreshold 接続属性

この属性は、SQL 文の実行時間が指定した期間を超えたときにサポート・ログに警告を書き込み、SNMP トラップをスローするために使用します。レプリケーション・エージェントが実行する問合せについては、[25 ページの「レプリケーションのための問合せのしきい値の設定」](#)を参照してください。キャッシュ・エージェントが実行する SQL 文の問合せのしきい値は設定できません。QueryThreshold の値はすべての接続に適用されます。また、レプリケーション・エージェントまたはキャッシュ・エージェントが実行した SQL 文を除くすべての SQL 文にも適用されます。

この属性には、0 以上の整数を指定できます。デフォルト値は 0 です。値が 0 の場合は、警告が発行されないことを示します。単位は秒です。

設定 QueryThreshold は次のように設定します。

属性の設定場所	属性の表現方法	設定
C または Java プログラム、 あるいは UNIX ODBC.INI ファイル	QueryThreshold	負でない整数。デフォルトは 0 で、警告が発行されないことを示します。
Windows ODBC Data Source Administrator	問合せのしきい値 (秒) フィールド	負でない整数。デフォルトは 0 で、警告が書き込まれないことを示します。

ttIsql を使用した問合せのしきい値の設定

デフォルトでは、ttIsql ユーティリティが ODBC.INI ファイルから問合せのしきい値の設定を取得します。次の接続文字列で問合せのしきい値の値を指定すると、この値を変更できます。

```
% ttisql -connstr "DSN=mydsn;QueryThreshold=10"
```

この場合、この接続で実行されるすべての SQL 文に問合せのしきい値 10 が適用されます。

ttIsql set コマンドを使用すると、問合せのしきい値を変更できます。たとえば、次のようになります。

```
Command> set querythreshold 20;
```

この場合、この接続で実行される後続のすべての SQL 文に問合せのしきい値 20 が適用されます。

ODBC オプションを使用した問合せのしきい値の設定

SQLSetConnectOption および SQLSetStmtOption ODBC 関数の TT_QUERY_THRESHOLD オプションを使用すると、問合せのしきい値を設定できます。

デフォルトでは、アプリケーションが ODBC.INI ファイルから問合せのしきい値の設定を取得します。SQLSetConnectOption ODBC 関数で TT_QUERY_THRESHOLD オプションを設定すると、現行の接続の ODBC.INI ファイルの設定を変更できます。

SQLSetStmtOption ODBC 関数で TT_QUERY_THRESHOLD オプションを設定すると、ODBC.INI 設定および SQLSetConnectOption 設定の両方を変更できます。この変更は、この ODBC 文で実行される SQL 文に適用されます。

SQLSetConnectOption を使用して、問合せのしきい値を設定します。

```
RETCODE SQLSetConnectOption(hdbc,  
                             TT_QUERY_THRESHOLD,  
                             seconds)
```

SQLSetStmtOption を使用して、問合せのしきい値を設定します。

```
RETCODE SQLSetStmtOption(hdbc,  
                          TT_QUERY_THRESHOLD,  
                          seconds)
```

JDBC での問合せのしきい値の設定

デフォルトでは、アプリケーションが ODBC.INI ファイルから問合せのしきい値の設定を取得します。JDBC 接続の問合せのしきい値は、データ・ソースへの接続 URL に QueryThreshold 属性を含めることによって変更できます。たとえば、myDSN データ・ストアの QueryThreshold を 5 秒に設定するには、次のように指定します。

```
jdbc:timesten:direct:dsn=myDSN;QueryThreshold=5
```

問合せのしきい値の設定には、TimesTenStatement インタフェースの setQueryTimeThreshold メソッドを使用することもできます。これによって、ODBC.INI の設定とその Connection 設定が変更されます。この変更は、このメソッドで実行される SQL 文に適用されます。setQueryTimeThreshold メソッドの使用の詳細は、『Oracle TimesTen In-Memory Database JDBC API Extensions』を参照してください。このドキュメントは `install_dir/doc/ttjava.zip` にあります。

TTClasses を使用した問合せのしきい値の設定

TTCmd クラスの `setQueryThreshold` メソッドを使用すると、問合せのしきい値を設定できます。

```
void TTMd::setQueryThreshold(const int seconds, TTStatus &stat)
```

問合せのしきい値の取得

問合せのしきい値が設定された方法に応じて、`ttIsql`、`SQLGetConnectOption` または `SQLGetStmtOption` ODBC 関数、あるいは `TTClasses` を使用して、その問合せのしきい値を取得できます。

ttIsql を使用した QueryThreshold 属性の取得

`ttIsql` を使用すると、`QueryThreshold` 属性を取得できます。

```
Command> show querythreshold;  
Query threshold = 0 seconds (no threshold)
```

TT_QUERY_THRESHOLD ODBC オプションの取得

`SQLGetConnectOption` または `SQLGetStmtOption` ODBC 関数を使用すると、`TT_QUERY_THRESHOLD` ODBC オプションの現行の値を取得できます。

```
RETCODE SQLGetConnectOption(hdbc,  
                             TT_QUERY_THRESHOLD,  
                             int seconds)  
  
RETCODE SQLGetStmtOption(hdbc,  
                          TT_QUERY_THRESHOLD,  
                          int seconds)
```

JDBC を使用した問合せのしきい値の取得

`TimesTenStatement` インタフェースの `getQueryTimeThreshold()` メソッドを使用すると、現行の問合せのしきい値を取得できます。『Oracle TimesTen In-Memory Database JDBC API Extensions』を参照してください。このドキュメントは `install_dir/doc/ttjava.zip` にあります。

TTClasses を使用した問合せのしきい値の取得

TTCmd クラスの `getQueryThreshold` メソッドを使用すると、現行の問合せのしきい値を取得できます。

```
int TTMd::getQueryThreshold(TTStatus &stat)
```

レプリケーションのための問合せのしきい値の設定

レプリケーション・エージェントでは、ODBC.INI ファイルの QueryThreshold 設定を継承しません。ttRepQueryThresholdSet 組込みプロシージャ、または ttAdmin ユーティリティの -repQueryThresholdSet オプションを使用して、レプリケーション・エージェントの問合せのしきい値を設定します。レプリケーション・エージェントの問合せのしきい値は、マテリアライズド・ビューの詳細表での SQL 実行、ON DELETE CASCADE 操作、および SQL 文を実行する一部の内部操作に適用されます。

TimesTen では、問合せのしきい値を超えると、サポート・ログに警告が書き込まれ、SNMP トラップがスローされます。この場合も実行は継続され、問合せのしきい値による影響は受けません。

ttRepQueryThresholdSet

説明 この組込みプロシージャでは、TimesTen がサポート・ログに警告を書き込み、SNMP トラップをスローするまで、レプリケーション・エージェントによって問合せが実行される秒数を指定できます。指定された値は、次にレプリケーション・エージェントが起動されたときに有効になります。レプリケーション・エージェントの問合せのしきい値は、マテリアライズド・ビューの詳細表での SQL 実行、ON DELETE CASCADE 操作、および SQL 文を実行する一部の内部操作に適用されます。

アクセス制御 TimesTen インタフェースでアクセス制御が有効になっている場合、この組込みプロシージャには ADMIN 権限が必要です。

構文 ttRepQueryThresholdSet (*seconds*)

パラメータ ttRepQueryThresholdSet はパラメータをとります。

パラメータ	型	説明
<i>seconds</i>	TT_INTEGER NOT NULL	TimesTen がサポート・ログに警告を書き込み、SNMP トラップをスローするまで、レプリケーション・エージェントによって SQL 文が実行される秒数です。0 以上の値を指定する必要があります。デフォルトは 0 で、警告が書き込まれないことを示します。

結果セット この組込みプロシージャで結果は戻されません。

例 Command> call ttRepQueryThresholdSet (4);

ttAdmin -repQueryThresholdSet

説明 このオプションでは、TimesTen がサポート・ログに警告を書き込み、SNMP トラップをスローするまで、レプリケーション・エージェントによって問合せが実行される秒数を指定できます。指定された値は、次にレプリケーション・エージェントが起動されたときに有効になります。レプリケーション・エージェントの問合せのしきい値は、マテリアライズド・ビューのディテール表での SQL 実行、ON DELETE CASCADE 操作および一部の内部操作に適用されます。

0 以上の整数を指定する必要があります。デフォルトは 0 で、警告が送信されないことを示します。

構文 ttAdmin [repQueryThresholdSet seconds]
[-connstr connectionString|DSN]

結果セット -repQueryThresholdSet オプションでは、その他の ttAdmin 出力は戻されません。

例 \$ ttAdmin -repQueryThresholdSet 4 myDSN
RAM Residence Policy : inUse
Replication Agent Policy : manual
Replication Manually Started : True
Cache Agent Policy : manual
Cache Agent Manually Started : False

参照 『Oracle TimesTen In-Memory Database API リファレンス・ガイド』の ttAdmin ユーティリティに関する項

レプリケーションのための問合せのしきい値の取得

ttRepQueryThresholdGet 組込みプロシージャ、または ttAdmin ユーティリティの -repQueryThresholdGet オプションを使用すると、指定したレプリケーションの問合せの最新のしきい値を取得できます。

ttRepQueryThresholdGet

説明 この組込みプロシージャでは、レプリケーション・エージェントの問合せのしきい値として指定した最新の秒数が戻されます。戻される秒数は、実際には問合せのしきい値とは異なる場合があります。問合せのしきい値に指定された新しい値は、次にレプリケーション・エージェントが起動されたときに有効になります。

アクセス制御 TimesTen インタフェースでアクセス制御が有効になっている場合、この組込みプロシージャには ADMIN 権限が必要です。

構文 ttRepQueryThresholdGet

パラメータ この組込みプロシージャはパラメータをとりません。

結果セット この組み込みプロシージャでは、レプリケーション・エージェントの間合せのしきい値として指定した最新の秒数が戻されます。数値は、TT_INTEGER データ型として戻されます。

例 Command> call ttRepQueryThresholdGet;
< 4 >
1 row found.

ttAdmin -repQueryThresholdGet

説明 このオプションでは、レプリケーション・エージェントの間合せのしきい値として指定した最新の秒数が戻されます。戻される秒数は、実際には間合せのしきい値とは異なる場合があります。間合せのしきい値に指定された新しい値は、次にレプリケーション・エージェントが起動されたときに有効になります。

構文 ttAdmin [-repQueryThresholdGet] [-connstr *connectionString*|*DSN*]

結果セット ttAdmin ユーティリティのこのオプションでは、間合せのしきい値（秒）が戻されます。

例 \$ ttAdmin -repQueryThresholdGet myDSN
QueryThreshold in seconds : 4
RAM Residence Policy : inUse
Replication Agent Policy : manual
Replication Manually Started : True
Cache Agent Policy : manual
Cache Agent Manually Started : False

参照 『Oracle TimesTen In-Memory Database API リファレンス・ガイド』の ttAdmin ユーティリティに関する項

AWT キャッシュ・グループの パフォーマンスの監視

この章の内容は次のとおりです。

- [概要](#)
- [AWT 監視の有効化](#)
- [AWT 監視結果の表示](#)

概要

ASYNCHRONOUS WRITETHROUGH (AWT) キャッシュ・グループのパフォーマンスを監視すると、AWT ワークフローでのタスクの実行にかかる時間を判断できます。ttCacheAwtMonitorConfig プロシージャを使用して監視を有効にします。監視結果を表示するには、-awtmoninfo および -showstatus コマンドを使用して ttRepAdmin ユーティリティを実行します。

AWT 監視の有効化

ttCacheAwtMonitorConfig プロシージャをコールして AWT 監視を有効にします。

ttCacheAwtMonitorConfig

説明 このプロシージャでは、AWT キャッシュ・グループのワークフローに含まれる各要素でかかった時間を判断するための監視が有効になります。監視結果を表示するには、-awtmoninfo および -showstatus コマンドを使用して ttRepAdmin ユーティリティを実行します。

レプリケーション・エージェントが再起動されると、監視は無効になります。

監視の状態を OFF に設定すると、監視ツールの内部カウンタがリセットされます。

アクセス制御 TimesTen インタフェースでアクセス制御が有効になっている場合、このプロシージャには ADMIN 権限が必要です。

構文 ttCacheAwtMonitorConfig('state', samplingFactor)

パラメータ ttCacheAwtMonitorConfig はパラメータをとります。

パラメータ	型	説明
state	CHAR (10)	AWT 監視の有効化および無効化を行います。値は ON または OFF です。デフォルトは OFF です。
samplingFactor	TT_INTEGER	AWT ワークフローがサンプリングされる頻度を指定する正の整数です。samplingFactor を 1 に設定すると、すべての AWT 操作が監視されます。値を大きくすると、サンプリングの頻度は低くなります。正確さとパフォーマンスを考慮した推奨値は 16 です。state を ON に設定した場合の samplingFactor のデフォルトは 16 です。state を OFF に設定した場合の samplingFactor のデフォルトは 0 です。

結果セット ttCacheAwtMonitorConfig では、現行の状態と `samplingFactor` が戻されます。レプリケーション・エージェントが実行されていない場合、または AWT キャッシュ・グループが作成されていない場合は、エラーが戻されます。

例

例 4.1 監視が無効な場合に現行の状態と `samplingFactor` を取得します。

```
Command> CALL ttCacheAwtMonitorConfig;
< OFF, 0 >
1 row found.
```

例 4.2 監視を有効にし、サンプリング頻度を 16 に設定します。

```
Command> CALL ttCacheAwtMonitorConfig('ON',16);
< ON, 16 >
1 row found.
```

例 4.3 監視を無効にします。

```
Command> CALL ttCacheAwtMonitorConfig('OFF');
< OFF, 0 >
1 row found.
```

AWT 監視結果の表示

`-awtmoninfo` および `-showstatus` コマンドを使用して `ttRepAdmin` ユーティリティを実行すると、監視を有効にした AWT キャッシュ・グループの監視情報を表示できます。

ttRepAdmin -showstatus -awtmoninfo

説明 `ttCacheAwtMonitorConfig` プロシージャをコールして AWT キャッシュ・グループの監視を有効にしている場合は、`-awtmoninfo` オプションを指定して `ttRepAdmin -showstatus` コマンドを実行することで、監視結果を表示できます。

構文 `ttRepAdmin -showstatus -awtmoninfo {DSN | -connStr connectionString}`

結果セット AWT 監視が有効な場合は、このユーティリティによって、他の `ttRepAdmin -showstatus` 出力に加えて、次の情報が表示されます。

- TimesTen の処理時間: 監視が有効になった後、AWT トランザクション・データの処理にかかった合計時間 (ミリ秒)。
- Oracle のブックマーク管理時間: 監視が有効になった後、Oracle での AWT メタデータの管理にかかった合計時間 (ミリ秒)。
- Oracle の実行時間: 監視が有効になった後、AWT SQL 操作のための OCI 準備、バインドおよび実行にかかった合計時間 (ミリ秒)。この統計には、TimesTen と Oracle 間のネットワーク待機時間が含まれます。

- Oracle のコミット時間 : 監視が有効になった後、Oracle での AWT 更新のコミットにかかった合計時間 (ミリ秒)。この統計には、TimesTen と Oracle 間のネットワーク待機時間が含まれます。
- 監視が開始されてからの時間
- TimesTen の行操作の合計数 : 監視が有効になった後、AWT キャッシュ・グループで更新された行の合計数。
- TimesTen トランザクションの合計数 : 監視が有効になった後、AWT キャッシュ・グループに含まれるトランザクションの合計数。
- Oracle へのフラッシュの合計数 : TimesTen データが Oracle に送信された回数

出力には、TimesTen の処理、Oracle のブックマーク管理、Oracle の実行および Oracle のコミットにかかった時間の割合も含まれます。

例

```
ttRepAdmin -showstatus -awtmoninfo myDSN
[other -showstatus output]
...
AWT Monitoring statistics
-----
TimesTen processing time : 0.689000 millisecs (0.164307 %)
Oracle bookmark management time : 3.229000 millisecs (0.770027%)
Oracle execute time : 342.908000 millisecs (81.774043 %)
Oracle commit time : 72.450000 millisecs (17.277315 %)
Time since monitoring was started: 8528.641000 millisecs
Cache-connect Operational Stats :
    Total Number of TimesTen row operations : 2
    Total Number of TimesTen transactions : 2
    Total Number of flushes to Oracle : 2
```

参照

『Oracle TimesTen In-Memory Database API リファレンス・ガイド』の
ttRepAdmin ユーティリティに関する項

5

Oracle での自動リフレッシュ・オブジェクトのクリーンアップ

この章の内容は次のとおりです。

- [概要](#)
- [自動リフレッシュのクリーンアップ・スクリプトの使用](#)

概要

TimesTen では、READONLY キャッシュ・グループ、またはキャッシュ・グループの作成時に AUTOREFRESH INCREMENTAL 属性を設定したキャッシュ・グループで指定された Oracle 実表のトリガーおよびログ表が自動的に作成されます。キャッシュ・グループを削除すると、オブジェクトは削除されます。これらの Oracle オブジェクトの詳細は、『Oracle TimesTen Cache Connect to Oracle 開発者および管理者ガイド』の READONLY、AUTOREFRESH および AWT キャッシュ・グループでの Oracle オブジェクトの管理に関する項を参照してください。

キャッシュ・グループを削除せずに TimesTen データ・ストアが使用不可になった場合、自動リフレッシュ・オブジェクトは Oracle に残ります。たとえば、ホスト・マシンが完全にオフラインになった場合やデータ・ストアが破棄された場合は、TimesTen データ・ストアが使用不可になります。この後 TimesTen データ・ストアを使用しない場合でも、キャッシュ・グループが削除されていなければ、自動リフレッシュ・オブジェクトは残ります。変更ログ・レコードは変更ログ表に累積されます。これは、アクティブ・データ・ストアの自動リフレッシュのパフォーマンスに影響します。したがって、使用できないデータ・ストアまたは破棄されたデータ・ストアに関連付けられた Oracle オブジェクトはクリーンアップすることをお勧めします。

TimesTen では、Oracle Database の自動リフレッシュ・オブジェクトをクリーンアップできるスクリプトが用意されています。スクリプトの場所は次のとおりです。

```
install_dir/bin/autorefreshCleanUp.sql
```

スクリプトの実行時に指定したデータ・ストアおよびホストに関連する自動リフレッシュ・オブジェクトが削除されます。他のデータ・ストアの自動リフレッシュ・オブジェクトには影響しません。

自動リフレッシュのクリーンアップ・スクリプトの使用

SQL*Plus を使用して、Oracle Database で自動リフレッシュのクリーンアップ・スクリプトを実行します。スクリプトを実行するには、Oracle Database にキャッシュ管理ユーザーとして接続する必要があります。

自動リフレッシュのクリーンアップ・スクリプトには、TimesTen が実行されているホスト名およびデータ・ストアのパス名という 2 つの入力パラメータが必要です。ホスト名およびデータ・ストアのパス名の文字列は、Oracle に格納されている文字列と同じである必要があります。正しいホスト名およびデータ・ストアのパス名は、autorefreshChangeLogInfo.sql スクリプト (`install_dir/bin` にあります) を実行すると確認できます。Windows の場合、ホスト名およびデータ・ストアのパス名は小文字で入力します。

ホスト名およびデータ・ストアのパス名は、コマンドラインまたは SQL*Plus セッションからスクリプトに渡します。

- コマンドラインからの場合：

```
$ sqlplus cache_admin_uid@oracle_id @install_dir/bin/autorefreshCleanUp.sql  
"host_name" "data_store_path_name"
```

- SQL*Plus セッションからの場合：

```
SQL> @install_dir/bin/autorefreshCleanUp.sql "host_name" "data_store_path_name"
```

パラメータを指定しなかった場合は、スクリプトによって指定が求められます。パラメータ 1 がホスト名、パラメータ 2 がデータ・ストアのパス名です。

スクリプトを実行すると、自動リフレッシュ・オブジェクトをクリーンアップするために Oracle で実行する SQL が表示されます。

- 例 5.1** 次の出力例では、トリガーおよびログ表によって参照される Oracle 表は、1 つ以上の増分自動リフレッシュ・キャッシュ・グループによって引き続きキャッシュされます。指定したホストおよびデータ・ストアのエントリは削除されますが、トリガーおよびログ表は削除されていないことに注意してください。

```
Performing cleanup for object_id: 91771 which belongs to table :  
TTUSER.MYTABLE  
Executing: delete from tt_03_agent_status where host = my-pc and  
datastore = c:¥data¥rep2 and object_id = 91771  
Executing: update tt_03_user_count set usercount =  
:usecount,usercount = 1
```

- 例 5.2** 次の出力例では、トリガーおよびログ表によって参照される Oracle 表は、いずれの増分自動リフレッシュ・キャッシュ・グループでもキャッシュされません。ログ表およびトリガーが削除されていることに注意してください。

```
Performing cleanup for object_id: 83560 which belongs to table :  
TTUSER.MYTABLE  
Executing: delete from tt_03_agent_status where host = my-pc and  
datastore = c:¥data¥tt60 and object_id = 83560  
Executing: drop table tt_03_83560_L  
Executing: drop trigger tt_03_83560_T  
Executing: delete from tt_03_user_count where object_id = object_id1
```


アクティブ・スタンバイ・ペア のデュアル・アクティブ・マスタ ターの検出

通常、アクティブ・スタンバイ・ペアのレプリケーション・スキームでのアクティブ・マスターおよびスタンバイ・マスターのデータ・ストアの設定は、ユーザーが明示的に制御します（『Oracle TimesTen Replication - TimesTen to TimesTen 開発者および管理者ガイド』のアクティブ・スタンバイ・ペアの管理に関する項を参照）。ただし、スタンバイ・マスター・データ・ストアの役割をアクティブに変更するときに、アクティブ・マスターおよびスタンバイ・マスターのデータ・ストアの両方を変更できない場合もあります。

たとえば、アクティブ・マスター・データ・ストアのサイトへのネットワーク通信が中断され、別のサイトのスタンバイ・マスター・データ・ストアにアクティブとしての役割を引き継がせる必要があるとします。この場合に、現行のアクティブ・マスターでのレプリケーションを停止したり、役割を手動で変更することができない場合があります。最初にアクティブ・マスターでのレプリケーションを停止せずにスタンバイ・マスター・データ・ストアをアクティブに変更すると、両方のマスターが **ACTIVE** な状態になり、トランザクションを受け入れることとなります。このような状況では、アクティブとスタンバイの両方のストア間のネットワーク通信が回復したときに、**TimesTen** が自動的にマスター・データ・ストアのアクティブ / スタンバイの役割を決定します。

データ・ストア間の初回のハンドシェイクで、アクティブ・スタンバイ・ペアのレプリケーション・スキームにある両方のマスター・データ・ストアの状態が ACTIVE であると判断された場合、TimesTen では次の操作が自動的に実行されます。

- 最後に状態が ACTIVE に設定されたデータ・ストアの状態は ACTIVE のままで、アプリケーションへの接続は続行され、更新されます。
- 最初に状態が ACTIVE に設定されたデータ・ストアは無効になります。すべてのアプリケーションの接続は切断されます。
- 無効だったデータ・ストアが再度有効になったときに、他のマスター・データ・ストアにまだレプリケートされていないストアでトランザクションが発生していないかどうかを確認されます。このようなトランザクションが発生している場合、それらはトラップされ、データ・ストアの状態は IDLE のままになります。データ・ストアは、スタンバイ・マスターになるために、アクティブ・マスターから複製される必要があります。トラップされたトランザクションがない場合は、データ・ストアをスタンバイ・マスター・データ・ストアとして使用でき、その状態は自動的に STANDBY に設定されます。

トランザクション・ログ API (XLA)

今回のリリースでは、XLA 更新レコードの `ttXlaUpdateDesc_t` ヘッダーに、カスケード削除およびエージング削除を示すフラグが 1 行ずつ追加されています。

フラグ `TT_CASCADEDEL` は、XLA 更新がカスケード削除操作の一部として生成されたことを示し、`TT_AGING` は、エージング操作の一部として生成されたことを示します。

ヘッダー・ファイルには、次のフラグが含まれます。

```
/*
 * Values for ttXlaUpdateDesc_t.flags bits
 */
#define TT_UPDCOMMIT 1 /* Implied commit on this record */
#define TT_UPDFIRST 2 /* First record in transaction */
#define TT_UPDREPL 4 /* Record generated by replication */
#define TT_UPDCOLS 8 /* List of updated columns present */
#define TT_UPDINT1 16 /* Internal Use */
#define TT_UPDINT2 32 /* Internal Use */
#define TT_UPDDEFAULT 64 /* Default column values present */
#define TT_UPDINT3 128 /* Internal Use */
#define TT_CASCADEDEL 256 /* Cascade delete */
#define TT_AGING 512 /* Aging delete */
```


新しいエラー・メッセージ および SNMP トラップ

この章では、今回のリリースで追加されたエラー・メッセージについて説明します。その他のエラー・メッセージについては、『Oracle TimesTen In-Memory Database エラー・メッセージおよび SNMP トラップ』を参照してください。

エラー・メッセージ

エラー番号	エラーまたは警告メッセージ	詳細
2819	Invalid value 'value' for argument number for procedure name. The valid values are <values>.	組込みプロシージャをコールするときに、示されているパラメータに無効な値を渡しました。無効なパラメータの位置、無効な値および有効な値のリストが表示されます。パラメータに有効な値を渡す組込みプロシージャをコールしてください。
12252	Could not check Oracle DB	このメッセージとともに、キャッシュ・レイヤーからの追加のエラー・メッセージが表示されます。そのメッセージを確認し、問題を特定して修正してください。
12253	The Active Standby Oracle Database is the same as the DR Oracle Database	ttRepAdmin -duplicate -initCacheDR オプションを使用した場合は、DR Oracle Database をアクティブ / スタンバイ Oracle サブスクライブ・データベースと同じデータベースにすることはできません。いずれかのアクティブ・スタンバイ・マスター・ストアから DR Oracle Database への tt_03_reppeers エントリが確認されています。この競合が解消されるように、データ・リカバリを再構成してください。
16268	Failed to retrieve aging configuration for table <owner>.tablename>	これはレプリケーションの一部として表示されます。トランスミッタがはじめて特定の表にトランザクションを送信する場合、TimesTen ではその表定義が送信されます。このメッセージは、エージング情報の取得中に問題が発生したことを示しています。特定の障害を説明する追加のエラー・メッセージが表示されます。そのメッセージを確認し、問題を特定して修正してください。

エラー番号	エラーまたは警告メッセージ	詳細
16269	In an Active/Standby scenario, AWT Monitoring should be enabled on the Standby node	AWT キャッシュ・グループを使用するアクティブ・スタンバイ・ペアでは、監視によってスタンバイ・ノードにのみ関連する結果が戻されます。このスタンバイ・ノードが、Oracle Database にワークロードを伝播します。アクティブ・ノードで AWT 監視を有効にしても、監視情報は記録されません。スタンバイ・ノードで障害が発生し、アクティブ・ノードが処理を引き継ぐ場合は、 ttCacheAwtMonitorConfig 組込みプロシージャをコールすることによって、自動的に AWT 統計の監視が有効になります。
16270	AWT Monitoring should not be enabled on the Subscriber node in an Active/Standby scenario	サブスクライバおよび AWT キャッシュ・グループを使用するアクティブ・スタンバイ・ペアでは、サブスクライバを監視しても、有効な情報は得られません。 ttCacheAwtMonitorConfig 組込みプロシージャをコールして、AWT 統計の監視を無効にしてください。
16271	Monitoring could not be configured because replication agent does not have an active AWT thread	このエラーは、レプリケーション・エージェントが無効か、初期化フェーズにあるときにスローされます。予期しない状況になったために AWT の受信側が強制終了した場合、レプリケーション・エージェントは無効な状態になります。また、AWT の受信側がレプリケーション・エージェントによって起動されていない場合があります。この場合は初期化フェーズにあります。このエラーを解決するには、状態パラメータを ON に設定して、 ttCacheAwtMonitorConfig 組込みプロシージャを再度コールします。
17041	Failed to create Oracle DR scheme	これは一般的なエラーで、実際の障害に関連する詳細なエラー・メッセージも同時に表示されます。そのメッセージを確認し、問題を特定して修正してください。

エラー番号	エラーまたは警告メッセージ	詳細
17042	A Cache DR scheme can only be created on an ACTIVE STANDBY SUBSCRIBER	ttRepAdmin -duplicate -initCacheDR オプションは、アクティブ・スタンバイ・サブスクライバを複製するときのみ使用できます。アクティブ/スタンバイ・レプリケーションに含まれないデータ・ストアで、障害時リカバリ・スキーマを作成しようとした。これはサポートされていません。
17043	Failed to initialize Oracle DR tables	Oracle Disaster Recovery 表の初期化に失敗しました。ユーザー・エラー・ログで、エラーが発生した操作で戻されたエラー・メッセージの詳細を確認してください。宛先データ・ストアを破棄し、障害時リカバリ・プロセスを再初期化してください。
17044	Exited early because could not get cache group information	障害時リカバリ・プロセスが、必要な情報を取得できなかったために終了しました。ユーザー・エラー・ログで、エラーが発生した操作で戻されたエラー・メッセージの詳細を確認してください。宛先データ・ストアを破棄し、障害時リカバリ・プロセスを再初期化してください。
17045	Could not truncate cached tables on Oracle DR site	障害時リカバリ中に、レプリケーション・エージェントがキャッシュ表を切り捨てるできませんでした。ユーザー・エラー・ログで、エラーが発生した操作で戻されたエラー・メッセージの詳細を確認してください。宛先データ・ストアを破棄し、障害時リカバリ・プロセスを再初期化してください。
17046	Could not flush data to Oracle DR site	障害時リカバリ中に、レプリケーション・エージェントがデータを Oracle にフラッシュできませんでした。ユーザー・エラー・ログで、エラーが発生した操作で戻されたエラー・メッセージの詳細を確認してください。宛先データ・ストアを破棄し、障害時リカバリ・プロセスを再初期化してください。

エラー番号	エラーまたは警告メッセージ	詳細
17047	Could not commit data to the Oracle DR site	障害時リカバリ中に、レプリケーション・エージェントがデータを Oracle にコミットできませんでした。ユーザー・エラー・ログで、エラーが発生した操作で戻されたエラー・メッセージの詳細を確認してください。宛先データ・ストアを破棄し、障害時リカバリ・プロセスを再初期化してください。
17048	Could not create thread to initialize the Oracle DR site	障害時リカバリ中に、レプリケーション・エージェントが Oracle を初期化するためのスレッドを作成できませんでした。ユーザー・エラー・ログで、エラーが発生した操作で戻されたエラー・メッセージの詳細を確認してください。宛先データ・ストアを破棄し、障害時リカバリ・プロセスを再初期化してください。
17049	Error invalidating store due to scheme inconsistency	この問題を修正するには、この操作に含まれるスキーマが同じであることを確認してください。
17050	Conflict Replication Reporting Starting	競合レポートの一時停止 / 再開がシステムで構成され、競合率が再開の値を下回りました。レポートが送信されています。これは情報メッセージです。
17051	Conflict Replication Reporting Stopping	競合レポートの一時停止 / 再開がシステムで構成され、競合率が再開の値を超えました。レポートが一時停止されます。これは情報メッセージです。
47136	The cache agent hasn't fully initialized yet. The cache group <cache_group_name> will be refreshed after initialization is complete.	これは情報メッセージです。キャッシュ・エージェントを初期化する前に、CREATE または ALTER CACHE GROUP 文が発行されました。CREATE または ALTER 操作は、初期化が完了した後で有効になります。処置は必要ありません。

SNMP トラップ

今回のリリースでは、次の新しい SNMP トラップが追加されています。

エラーまたは警告メッセージ	エラー、警告または情報	詳細
ttQueryThresholdWarnTrap	警告	SQL 問合せがユーザーが指定したしきい値を超えました。問合せのテキストは、ユーザー・ログ・メッセージにあります。問合せのトランザクション ID およびステートメント ID は、いずれもトラップおよびユーザー・ログ・メッセージにあります。トラップを発行した後も、問合せは引き続き実行されます。

索引

A

AIX 6

C

Cache Connect to Oracle サポート 8

ccNUMA 9

H

HP-UX 6

J

JDBC

問合せのしきい値の設定 23

JDK サポート 6

L

Linux、MontaVista Carrier Grade 6

Linux、SUSE 7

M

Microsoft Windows 6

MontaVista Carrier Grade Linux 6

S

shmget 9

shmLocalityHint デーモン・オプション 9

Solaris 7

SQLSetConnectOption ODBC 関数

問合せのしきい値の設定 23

SQLSetStmtOption ODBC 関数

問合せのしきい値の設定 23

SUSE LINUX 7

T

Tru64 6

TT_QUERY_THRESHOLD ODBC オプション

23

ttAdmin ユーティリティ

-repQueryThresholdGet オプション 20

-repQueryThresholdSet オプション 20

TTClasses

問合せのしきい値の設定 24

ttQueryThresholdWarnTrap SNMP トラップ
21

ttRepQueryThresholdGet プロシージャ 20

ttRepQueryThresholdSet 組込みプロシージャ
20

ttXlaUpdateDesc_t 39

V

Veritas ファイル・システム・オプション 8

い

一般的な QueryThreshold 接続属性 20, 22

き

共有メモリーのデーモン・オプション 9

し

自動リフレッシュのクリーンアップ・スクリプト 34

て

デーモン・オプション

shmLocalityHint 9

と

問合せのしきい値

JDBC の使用 23

ttIsql での設定 22

問合せのしきい値の設定

TTClasses 24

