



ユーザーズ・ガイド

**Replication Server[®] Data
Assurance オプション 15.7.1**

ドキュメント ID：DC01895-01-1571-01

改訂：2012年4月

Copyright © 2012 by Sybase, Inc. All rights reserved.

このマニュアルは Sybase ソフトウェアの付属マニュアルであり、新しいマニュアルまたはテクニカル・ノートで特に示されないかぎり、後続のリリースにも付属します。このマニュアルの内容は予告なしに変更されることがあります。このマニュアルに記載されているソフトウェアはライセンス契約に基づいて提供されるものであり、無断で使用することはできません。

アップグレードは、ソフトウェア・リリースの所定の日時に定期的に提供されます。このマニュアルの内容を弊社の書面による事前許可を得ずに、電子的、機械的、手作業、光学的、またはその他のいかなる手段によっても、複製、転載、翻訳することを禁じます。

Sybase の商標は、Sybase の商標リスト (<http://www.sybase.com/detail?id=1011207>) で確認できます。Sybase およびこのリストに掲載されている商標は、米国法人 Sybase, Inc. の商標です。® は、米国における登録商標であることを示します。

このマニュアルに記載されている SAP、その他の SAP 製品、サービス、および関連するロゴは、ドイツおよびその他の国における SAP AG の商標または登録商標です。

Java および Java 関連のすべての商標は、米国またはその他の国での Oracle およびその関連会社の商標または登録商標です。

Unicode と Unicode のロゴは、Unicode, Inc. の登録商標です。

このマニュアルに記載されている上記以外の社名および製品名は、当該各社の商標または登録商標の場合があります。

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

目次

表記規則	1
Replication Server Data Assurance オプション	3
システム・アーキテクチャ	3
Data Assurance エージェント	6
接続プロファイル	7
comparesets	7
ジョブと比較	8
比較と調整の方式	8
主要オプション	10
ジョブの履歴	13
Data Assurance System Database	13
Replication Server との統合	14
導入ガイド	15
シングルサーバ配備	15
分散配備	22
データベース・テーブルの調整	24
管理タスク	31
ジョブの作成	31
スキーマ・ジョブの作成	31
Replication Server からのジョブのインポート	32
サーバの設定パラメータの設定	33
DASD のバックアップとリストア	33
データとログ・ファイルの削除	33
Data Assurance サーバのコマンド・リファレンス	35
エージェントのコマンド	35
alter agent	35
create agent	36
depend agent	37
drop agent	37

show agent	38
show agent connection	38
show agent dts	39
show agent jvm	39
show agent system	40
show agent task	41
test agent	41
接続プロファイルのコマンド	42
alter connection	42
create connection	43
depend connection	46
drop connection	47
show connection	47
test connection	48
比較セットのコマンド	48
alter compareset	48
create compareset	51
depend compareset	54
drop compareset	55
show compareset	55
ロー比較ジョブのコマンド	57
alter job	57
create job	63
drop job	69
show job	69
スキーマ比較ジョブのコマンド	70
alter schema job	70
create schema job	73
drop schema job	75
show schema job	75
ジョブ管理のコマンド	76
abort job	76
disable job	76
drop history	77
enable job	77

monitor job	78
run job	79
show history	80
show reconcile	81
show report	82
truncate history	83
ジョブのインポートのコマンド	83
import job	83
Data Assurance System Database (DASD) コマンド ..	88
create backup	88
drop backup	89
restore backup	89
show backup	89
truncate backup	90
その他のコマンド	90
config	91
password	101
role	101
show jvm	102
show system	103
sslconfig	103
trace	106
version	107
Data Assurance サーバの予約語	107
リモート Data Assurance エージェントのコマンド・リ	
ファレンス	109
config	109
password	110
role	111
show connection	112
show dts	112
show jvm	113
show system	113
show task	114
sslconfig	114

trace	117
version	118
Data Assurance エージェントの予約語	118
セキュリティとアクセス制御	121
Kerberos セキュリティ	121
Kerberos を使用するための DA エージェントの 設定	121
LDAP 認証	122
DA 管理者の役割	123
LDAP 認証のための DA の設定	123
SSL セキュリティ	124
SSL の概要	125
SSL の有効化	125
パスワード管理	129
パスワード・ポリシー	129
パスワードを紛失した場合または忘れた場合 のリセット	129
パスワードの暗号化	131
パフォーマンスとチューニング	133
一般設定	133
ロー比較の最適化	134
トラブルシューティング	137
概数値データ型の比較	137
DA サーバ・メモ不足エラー	137
外部ソート・オプションの設定	138
比較が失敗し、矛盾が記録される	139
ジョブ比較が応答しなくなる	139
スタック領域エラーで比較に失敗する	140
圧縮テーブルに対する比較の失敗	140
追加の説明や情報の入手	143
サポート・センタ	143

Sybase EBF と Maintenance レポートのダウンロード	143
Sybase 製品およびコンポーネントの動作確認	144
MySybase プロファイルの作成	144
アクセシビリティ機能	145
用語解説	147
索引	149

目次

表記規則

ここでは、Sybase® マニュアルで使用しているスタイルおよび構文の表記規則について説明します。

表記規則

構文要素	定義
等幅 (固定幅)	<ul style="list-style-type: none"> SQL およびプログラム・コード 表示されたとおりに入力する必要があるコマンド ファイル名 ディレクトリ名
斜体等幅	SQL またはプログラム・コードのスニペット内では、ユーザ指定の値のプレースホルダ (以下の例を参照)
斜体	<ul style="list-style-type: none"> ファイルおよび変数の名前 他のトピックまたはマニュアルとの相互参照 本文中では、ユーザ指定の値のプレースホルダ (以下の例を参照) 用語解説に含まれているテキスト内の用語
bold sans serif	<ul style="list-style-type: none"> コマンド、関数、ストアド・プロシージャ、ユーティリティ、クラス、メソッドの名前 用語解説のエントリ (用語解説内) メニュー・オプションのパス 番号付きの作業または手順内では、クリックの対象となるボタン、チェック・ボックス、アイコンなどのユーザ・インタフェース (UI) 要素

必要に応じて、プレースホルダ (システムまたは設定固有の値) の説明が本文中に追加されます。例：

次のコマンドを実行します。

```
installation directory\start.bat
```

installation directory は、アプリケーションのインストール・ロケーションです。

構文の表記規則とコマンド規則

構文要素	定義
{ }	中カッコで囲まれたオプションの中から必ず1つ以上を選択する。コマンドには中カッコは入力しない。
[]	角カッコは、オプションを選択しても省略してもよいことを意味する。コマンドには角カッコは入力しない。
()	このカッコはコマンドの一部として入力する。
	縦線はオプションのうち1つのみを選択できることを意味する。
,	カンマは、表示されているオプションを必要な数だけ選択でき、選択したものをコマンドの一部として入力するときにカンマで区切ることを意味する。
...	省略記号(...)は、直前の要素を必要な回数だけ繰り返し指定できることを意味する。省略記号はコマンドには入力しない。

- 有効な入力値の文字は、`_`、`a-z`、`A-Z`、`0-9`、`-`、`:` です。その他の文字はすべて引用符で囲む必要があります。空白スペースを含む入力文字列の値はすべて、引用符 (一重または二重) で囲む必要があります。
- キーワード **go** はコマンド・ターミネータとして扱われます。
- コマンド・バッファをクリアするには、**reset** を使用します。
- スクリプトで1行をコメントにするには、“`--`” または “`#`” を使用します。

大文字と小文字の区別

- すべてのコマンド構文およびコマンドの例は、小文字で表記しています。ただし、Replication Server® Data Assurance オプションのコマンド名は大文字と小文字が区別されません。たとえば、**CONFIG**、**Config**、**config** はすべて同じです。
- 設定パラメータの名前は、大文字と小文字が区別されません。たとえば、**MAX_CONCURRENT_COMPARISONS** は **max_concurrent_comparisons** と同じです。
- ユーザ接続プロパティは、大文字と小文字が区別されます。例：


```
alter connection myconn
with properties
set you=sybase and set YOU=sybase
```
- Adaptive Server® のデータベース・オブジェクトは、大文字と小文字が区別されます。DA 設定でデータベース・オブジェクトを指定するときは、テーブル名の大文字と小文字を正しく使い分けてください。

Replication Server Data Assurance オプション

Replication Server Data Assurance オプションは、2 つ以上の Adaptive Server® Enterprise データベースのロー・データおよびスキーマを比較して、不一致をレポートします。

Replication Server Data Assurance オプションは、スケーラブルで大容量の設定可能なデータ比較製品です。ダウン・タイムを排除するための“wait and retry”手法を使用することで、複写中でも比較ジョブを実行できます。

各比較ジョブでは、どのデータをどのように比較するかを指定した多数の設定を使用してデータの不一致を確認できます。Replication Server Data Assurance オプションには、あらゆる比較とレポート作成のジョブを実行するためのコマンド・ライン・ツール (CLT) が搭載されています。ユーザはジョブのモニタおよびアポートを実行したり、詳細な比較レポートを作成したりできます。Replication Server Data Assurance オプションは、SySAM ライセンス・マネージャを通じてライセンスされ、複数のプラットフォームで使用できます。

SySAM の詳細については、使用しているプラットフォームの『インストール・ガイド』、または SySAM Web サイト (<http://www.sybase.com/sysam>) を参照してください。

システム・アーキテクチャ

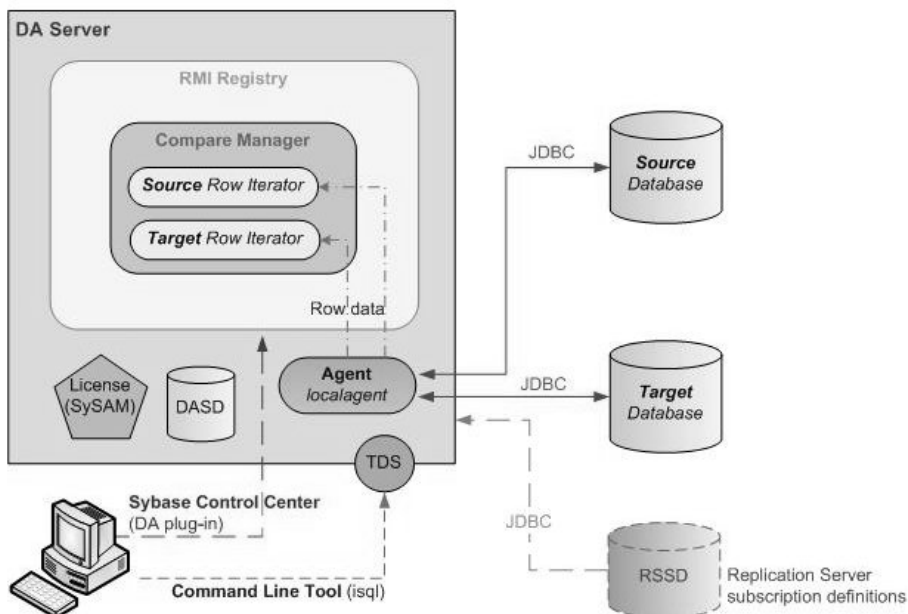
Replication Server Data Assurance オプション・システムには、サテライト DA エージェントを 0 個以上搭載した主要な Data Assurance (DA) サーバ・コンポーネントが用意されています。サーバとエージェントの正確な数は、使用している配備タイプがシングルサーバであるか分散であるかで異なります。

シングルサーバ配備

ここでは、以下で構成されるシングルサーバ配備のサンプル・アーキテクチャを示します。

- DA サーバ 1 台 (組み込みエージェントあり)
- プライマリ (送信元) データベース 1 台
- レプリケート (ターゲット) データベース 1 台
- Data Assurance System Database (DASD)
- コマンド・ライン・ツール (CLT)
- Replication Server システム・データベース (RSSD)
- コンポーネント間で使用されるプロトコル

図 1 : シングルサーバ配備

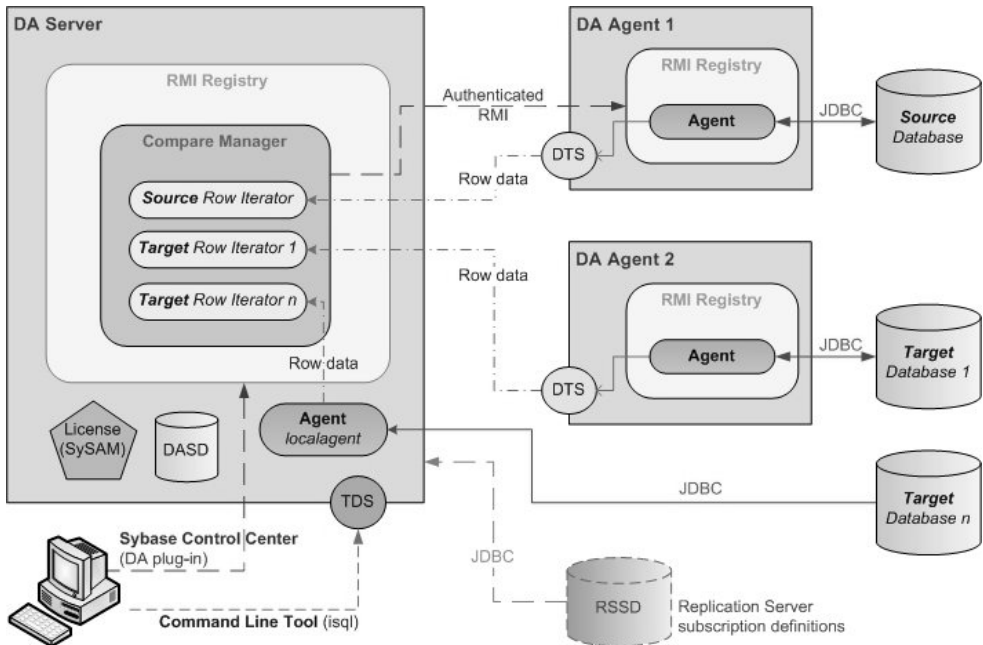


分散配備

ここでは、以下で構成される分散配備のサンプル・アーキテクチャを示します。

- DA サーバ 1 台 (組み込みエージェントあり)
- リモート DA エージェント 2 台
- プライマリ (送信元) データベース 1 台
- レプリケート (ターゲット) データベース 2 台
- Data Assurance System Database (DASD)
- コマンド・ライン・ツール (CLT)
- Replication Server システム・データベース (RSSD)
- コンポーネント間で使用されるプロトコル

図 2 : 分散配備



DA サーバは、DASD およびローカル・エージェントと呼ばれる組み込み DA エージェントを搭載したコア・コンポーネントです。スタンド・アロンのリモート DA エージェントは、DA サーバによって比較ジョブのパフォーマンスを向上するために使用されます。使用しているデータベースとの高速イーサネット・コネクションを共有するマシン上にリモート DA エージェントをインストールすることをおすすめします。

DA サーバ：

- ローの比較
- スキーマの比較
- ジョブの作成
- ジョブのスケジュールの作成
- 調整用のスクリプトの作成
- 自動調整用のデータ操作言語 (DML) コマンドの作成
- ジョブ・ステータスのチェック
- レポートの生成

使用しているデータベースまたは複製環境の要件に基づいてインストールする DA エージェントの数を決定してください。たとえば、プライマリ (送信元) データベース用にリモート DA エージェント 1 台とレプリケート (ターゲット) データベース用にリモート DA エージェント 1 台をインストールすることを選択するとしま

す。異なるデータベースに接続する異なるジョブがある場合、複数のリモート DA エージェントをインストールすることもできます。単一のプライマリ・データベースと 10 台のレプリケート・データベースがある場合でも、ローカル・エージェントを使用して比較タスク全体を実行することもできます。

DA サーバまたは DA エージェントにコマンドを送信するには、次のいずれかを使用します。

- 対話型 SQL (**isql**) ツール。 **isql** パーサを使用すると、 **isql** オプション・ファイル内の複数のコマンドを順番に実行できます。 Adaptive Server Enterprise の『ユーティリティ・ガイド』の「コマンド・ラインからの Interactive **isql** の使用」を参照してください。
- Sybase Control Center Data Assurance プラグイン。『Sybase Control Center 3.2.6 for Data Assurance Documentation』を参照してください。

DA サーバは、Replication Server と統合され、RSSD 内の情報に基づいて DA サーバ・ジョブを自動的に生成します。Replication Server は、テーブル複写定義からのみジョブをインポートできます。データベース複写定義からジョブをインポートすることはできません。

DA サーバは 1 つのログイン、`da_admin` のみをサポートします。これにはすべての管理者権限が割り当てられています。

参照：

- Data Assurance エージェント (6 ページ)
- 接続プロファイル (7 ページ)
- Data Assurance System Database (13 ページ)
- Replication Server との統合 (14 ページ)
- `import job` (83 ページ)

Data Assurance エージェント

Data Assurance (DA) エージェントは、データをデータベースからフェッチし、DA サーバに圧縮します。

エージェントには、2 つのタイプがあります。DA サーバ組み込みのローカル・エージェントと、DA サーバが使用してジョブのパフォーマンスを向上させることができるスタンド・アロン (リモート) エージェントです。

DA エージェントは、1 つ以上の JDBC 接続を開き、比較に必要なロー・データを読み込みます。スタンドアロンの DA エージェントがない場合は、組み込みのローカル・エージェントを使用する必要があります。

DA サーバの要求に基づいて、DA エージェントは次の処理を実行します。

- 設定に応じて事前比較用にローを圧縮する
- 比較用にローをフェッチする
- 設定に応じて事前比較用にローをハッシュ処理する
- 設定に応じて事前比較用にローをソートする
- 設定に応じてデータベースで **insert/delete/update** ステートメントを実行する

参照：

- create agent (36 ページ)

接続プロファイル

接続プロファイルには、データベース・コネクションを確立するために必要な情報が格納されています。

DA サーバでは、送信元データベースとターゲット・データベースの間のデータを比較するための接続プロファイルが作成されます。ログイン・クレデンシャルが含まれる接続プロファイルは、Data Assurance システム・データベース (DASD) に格納されます。各データベース・コネクションは、単一の DA エージェントが所有しており、比較セットとスキーマ・ジョブによって何回でも使用できます。

注意： 比較セットおよびスキーマ・ジョブを作成するには、事前に接続プロファイルを設定する必要があります。

参照：

- create connection (43 ページ)

comparesets

comparesets は、テーブルとカラムのセットで構成され、特定のジョブで比較されるデータを定義します。

compareset には、次の内容が含まれます。

- 比較するテーブル
- ローをユニークに定義するキー・カラム
- 比較するカラム
- 比較するローを定義する **WHERE** 句

compareset では、ソースとターゲットのテーブルとカラムが同じ名前である必要はありません。各 compareset には、1つのソースと1つ以上のターゲットが必要です。任意の数のジョブで使用できます。

参照：

- create compareset (51 ページ)

ジョブと比較

ジョブは、1つ以上のローまたはスキーマの比較タスクのコレクションです。ジョブは、手動で作成することも、Replication Server システム・データベース (RSSD) 内の情報に基づいて自動的に生成することもできます。ジョブの実行は手動で行うことも、特定の時刻または間隔で実行するようにスケジュールすることもできます。

ロー比較では、対象のデータベース・テーブルすべてのローをプライマリ・キーまたはユニーク・キーでソートし、1つずつ比較します。比較の概要は、Data Assurance システム・データベース (DASD) に保存されます。ローの相違を示す詳細なテキストまたは XML レポートは、データ・ディレクトリに保存されます。

スキーマ比較では、1つのプライマリ・データベースのスキーマを1つまたは複数のレプリケート・データベースと比較できます。自動テーブル名マッピングを使用してデータベース・スキーマ全体を比較したり、テーブル名マッピングを使用して特定のテーブルを比較したりすることができます。

参照：

- create job (63 ページ)
- create schema job (73 ページ)
- ジョブの作成 (31 ページ)
- スキーマ・ジョブの作成 (31 ページ)

比較と調整の方式

DA サーバの比較方式と調整は、ローとスキーマの比較ジョブの計画と管理に役立ちます。

DA サーバの比較と調整には以下の段階があります。

- 最初の比較
- インフライト・データ・オプション
- 差分の検証
- 調整

最初の比較

すべてのジョブで必要な最初の比較中、DA エージェントはクエリを使用して送信元とターゲットのデータベースからローをフェッチします。次のいずれかのオプションを指定して、DA サーバでローの比較を指定できます。

- カラム・ハッシュ (column_hash) – 各カラムの値が独自のハッシュ値になります。
- ロー・ハッシュ (row_hash) – 複数のカラムの値が1つのハッシュ値になります。
- リテラル比較 – 完全なカラム・データを比較します (値と値の比較)。
- 混合比較モード – 一部のカラムをハッシュごとに比較し、その他のカラムをリテラルごとに比較します。

注意： column_hash と row_hash など、一部はロー比較ジョブにのみ適用されます。

インフライト・データ・オプション

複製中のインフライト・データが原因で、比較中にローの差異が生じる可能性があります。DA サーバでは、ターゲット・データベースからのみロー・データを選択して、ローの差異を再チェックできます。テーブルのフル・チェックを実行する必要はありません。

ローの差異は次の3種類に分類されます。

- 消失 – プライマリ・テーブルのローがレプリケート・テーブルにない。
- 孤立 – レプリケート・テーブルのローがプライマリ・テーブルにない。
- 一貫性がない – 両方のテーブルにローがあるが、ローのデータが異なる。

DA サーバが最初の比較でローの差異を識別した場合は、インフライト・データ比較がそれらのローを再チェックして、差異が調整されたかどうかを確認します。これは、ターゲット・データベースの更新時に遅延がある複製環境で特に重要です。

オプションのインフライト・データ比較は、最初の比較中にデータの不整合を示したローすべてに“wait and retry”手法を適用します。たとえば、8:00 p.m. の最初の比較で同期がずれているローが1つ検出され、待機時間が120秒の場合、複製がそのローにインフライトの変更を適用できるようになる8:02 p.m. まで再比較は開始されません。

注意： インフライトの比較は送信元データベースには影響しません。差異が検出された送信元ローはすべて、ターゲット・データベースから再選択されたローに対して再度比較するためにキャッシュされます。

差分の検証

DA サーバは、送信元データベースとターゲット・データベースで異なるすべてのローのリテラル・データをフェッチして、それをカラム・ログに書き込みます。ジョブを作成するときに、**CREATE_COL_LOG** を true に設定して、このオプションを有効にします。カラム・ログには、消失ロー、孤立したロー、一貫性のないローの値 (キーとカラム) がすべて表示されます。

調整

ジョブ設定に従って、データの差異を自動的に調整するか、調整スクリプトを生成して調整します。DA サーバによって差異が検証され、ターゲット・テーブルが送信元テーブルと同じ状態になるように SQL 文を生成します。ローの差異の種類に応じて、DA サーバは以下を実行します。

- 消失ローのターゲット・テーブルの **insert** 文
- 孤立したローのターゲット・テーブルの **delete** 文
- 一貫性のないローのターゲット・テーブルの **update** 文

参照：

- ロー比較の最適化 (134 ページ)

主要オプション

DA サーバには比較とジョブのさまざまなオプションが備わっているので、これらを使用してローとスキーマの比較クエリを最適化できます。

Compressed Data Transfer (CDT)

CDT 中、リモート DA エージェント (ローカル DA エージェントを除く) と DA サーバ間のロー・データは圧縮されるので、ネットワーク遅延が大きい分散環境で全体的な比較時間が短縮されます。次のようなデータが圧縮されます。

- ローまたはキーの最初の比較中に送信されるロー・データすべて
- ローまたはキーの比較のリトライすべて
- ローまたはキーの比較の検証済み差分

CDT はロー比較のオプションで、スキーマ比較ではサポートされていません。ハッシュは圧縮率が低く、カラムで "column_hash" または "row_hash" オプションを使用している場合、最初の比較とインフライト・データの比較で CDT を使用する利点はほとんどありません。一方、リテラル・データの圧縮率は高く、カラムで "literal" カラム・オプションを使用している場合、最初の比較とインフライト・データの比較で CDT を使用する利点があります。差分の検証段階では常に、CDT を使用する利点があります。

以下の場合に CDT を検討してください。

- ローカル・エリア・ネットワーク (LAN) または広域ネットワーク (WAN) がボトルネックである。
- リテラル比較を実行している。
- 数百または数千以上の差異が予想される。
- プライマリ・キー・カラムが大きい (キー・カラムはハッシュされることがないため)。

以下の場合には CDT を選択しないでください。

- ローカル DA エージェントを使用している。
- LAN または WAN のパフォーマンスに問題がない。
- 常にハッシュを使用し、「差分を検証」オプションを使用しない。または、使用しても差異が少ないかほとんどない。

このオプションを使用するには、**compress_data_transfer** を true に設定します。

外部ソート・オプション

order by 句は、キー・カラムの値を基準にローがソートされた結果セットを **select** 文が返すように指定します。DA サーバでは、送信元とターゲットのテーブル・ローを比較前にソートする必要があります。大容量のテーブルでは、このソートが Adaptive Server の一時データベース領域に悪影響を及ぼします。**order by** 句の処理がデータベースに及ぼす影響を減らすには、次のような外部ソート・オプションを使用します。

- **order by** 句を省略し、データベースからソートされていないローを受け取る。
- システムでフラット・ファイルに書き込むときにローをソートする。
- すべてのフラット・ファイルから同時に読み取り、比較に対してソートされたローを返す。

関連する設定パラメータを結果が最適化されるように調整して、外部ソート・オプションの制御と設定を実行できます。

このオプションを使用するには、**external_sort** を true に設定します。

データベース・ハッシュ比較

注意： Adaptive Server は、バージョン 15.0.2 以降で **hashbytes** 関数をサポートしています。送信元またはターゲットのデータベースが 15.0.2 より前の場合は、**database_hash** オプションを使用できません。

database_hash カラム比較オプションは、text、image、unitext のようなラージ・オブジェクト (LOB) をサポートしていません。これは、Adaptive Server の **hashbytes** 関数が LOB 型をパラメータとして受け入れないからです。LOB 型はすべて “first N bytes” パラメータを使用します。この N は設定可能なパラメータです。LOB カラム内のバイト数が “N” より少ない場合は、カラム値全体が使用されます。

Adaptive Server Hashbytes の Null 処理

Adaptive Server Transact-SQL® の **hashbytes** 関数は Null 値を無視します。

たとえば、送信元テーブルに `column_a=34` と `column_b=NULL` があり、ターゲット・テーブルに `column_a=NULL` と `column_b=34` がある場合、等価性テストは次のようになります。

```
hashbytes(34, NULL) = hashbytes(NULL, 34),
```

これは次のように計算されます。

```
hashbytes(34) = hashbytes(34)、(「誤検出」一致)
```

Adaptive Server の **hashbytes** の制限事項に対処するため、DA サーバは設定パラメータ **db_hash_ase_ignore_null** を提供して、「誤検出」によるロー一致の確率を減らしています。**db_hash_ase_ignore_null** を `false` に設定すると、カラムの“is null”状態を示す余分な値を追加して、この問題を解消できます。上の例は次のようになります。

```
hashbytes(0, 34, 1, NULL) = hashbytes(1, NULL, 0, 34),
```

これは次のように計算されます。

```
hashbytes(0, 34, 1) = hashbytes(1, 0, 34).
```

データ調整オプション

DA サーバは送信元データベースとターゲット・データベース間のデータの差異を修正できます。新しいジョブを作成するとき、DA サーバは次の2つの比較オプションを提供します。

- **create_recon_script** – このオプションを `true` に設定すると、**insert**、**update**、**delete** 文を含むスクリプトを生成します。
- **auto_reconcile** – このオプションを `true` に設定すると、調整が必要なデータベースに **insert**、**update**、**delete** 文を生成して実行します。

注意： 調整オプションを機能させるには、**create_col_log** を `true` に設定してください。

スケジュール・オプション

比較ジョブを作成するとき、その比較ジョブに特定のスケジュールを割り当てることができます。ジョブは、日、週、月に基づいてスケジュールできます。また、指定した日時にコマンドを実行する UNIX クロック・デーモン `cron` を使用して設定することもできます。

注意： スケジュールのフォーマットは `cron` に従いますが、UNIX の `cron` コマンドは使用しません。DA サーバがスケジュールを管理します。

参照：

- create job (63 ページ)
- ジョブの作成 (31 ページ)

ジョブの履歴

終了済みのジョブは、それぞれ、送信元データベースとターゲット・データベースを比較するために使用されたパラメータおよび比較の結果を含むジョブ・レポートを生成します。

レポートは、テキストおよび XML のファイル形式で、Replication Server Data Assurance オプションのインストール・ディレクトリにある data サブディレクトリに保存されます。data ディレクトリは、各ジョブのジョブ名とタイムスタンプによってさらに分類されています。例：

```
C:\¥Sybase¥DA-15_5¥server¥instance¥data
¥job2¥2010-10-13¥14.38.11.762¥report.txt
```

構文の説明は次のとおりです。

- *job2* – ジョブ名を指定します。
- *2010-10-13* – ジョブが送信された日付を指定します。
- *14.38.11.762* – ジョブが送信された時刻を、時間、分、秒、ミリ秒で指定します。

各レポート・ファイルには、次の情報を含む詳細な情報が記載されています。

- 使用された比較オプション
- 比較されたローの数

レポート・ファイルは、次のコマンドを実行すると生成されます。

```
show history jobname historyid
```

create_col_log が true に設定されている場合、XML およびテキストのレポート・ファイルには、各差異の詳細も記載されます。

参照：

- show history (80 ページ)

Data Assurance System Database

Data Assurance System Database (DASD) は、比較情報をすべて格納しています。

DASD は以下のデータを格納しています。

Replication Server Data Assurance オプション

- システムと構成の設定
- エージェント・コネクション、データベース・コネクション、比較セット、ジョブ (比較とスケジュールを含む) の構成設定
- レポート用のタスク実行履歴

DASD は \$SYBASE/DA-15_5/DA/server/instance/dasd/dasd.db にあります。

参照：

- create backup (88 ページ)
- DASD のバックアップとリストア (33 ページ)

Replication Server との統合

Replication Server Data Assurance オプションは、Replication Server と統合されており、Replication Server システム・データベース (RSSD) 内の情報に基づいて DA サーバ・ジョブが自動的にインポートされます。DA サーバは、RSSD に直接接続し、テーブル複写定義とサブスクリプションに関する情報を取得して、プライマリ・データベースとレプリケート・データベースの間の複写を定義するテーブルとカラムを決定します。

注意： Replication Server Data Assurance オプションは、データベース複写定義をサポートしていません。

Replication Server からインポートされるジョブには、次のようなジョブがありません。

- 比較タスク名
- プライマリ (送信元) およびレプリケート (ターゲット) データベース
- テーブルおよびカラム - 比較 SQL 文用
- スケジュール

注意： DA サーバは、インポートされたジョブのスケジュールを自動的に作成しません。

参照：

- import job (83 ページ)
- Replication Server からのジョブのインポート (32 ページ)

導入ガイド

Replication Server Data Assurance オプション用にシングルサーバまたは分散配備を設定します。

これらの例では、pubs2 Adaptive Server Enterprise サンプル・データベースを使用します。pubs2 は、両方の Adaptive Server Enterprise サーバにインストールする必要があります。『Adaptive Server Enterprise インストール・ガイド Windows 版』の「インストール後の作業」で、「サンプル・データベースのインストール」を参照してください。

シングルサーバ配備

DA サーバとデータベース・サーバ間のネットワーク遅延が少なく、同時比較があまり必要でない場合には、シングルサーバ配備をおすすめします。シングルサーバ配備は、分散配備に比べて配備と管理も簡単です。

作業を始める前に

この例では、ローカルな埋め込みエージェントのある Data Assurance (DA) サーバ 1 台を使用しています。リモート DA エージェントは使用されていません。

表 1：配備の概要

コンポーネント名	マシン名	ポート番号
DA サーバ	mars	<ul style="list-style-type: none"> • 4500 – RMI • 4501 – TDS • 4503 – DASD
Adaptive Server Enterprise	venus	5000 – サーバ
Adaptive Server Enterprise	pluto	5000 – サーバ

1. DA サーバのインスタンスを起動します。

```
$SYBASE/DA-15_5/server/instance/RUN_instance_64.sh
```

\$SYBASE は、Data Assurance オプションをインストールしたディレクトリ、*instance* は DA サーバのインスタンス名、*RUN_instance_64.sh* は起動スクリプトです。

注意： Windows では、起動スクリプト・ファイルの名前は

RUN_instance_32.bat または *RUN_instance_64.bat* で、*instance* は DA

サーバのインスタンス名です。UNIX または Linux プラットフォームでは、ファイル名は `RUN_instance_64.sh` です。

2. `isql` から管理者として、次のように DA サーバにログインします。

```

$SYBASE/OCS-15_0/bin/isql -S mars:4501 -U da_admin -P password -w 250

```

注意：4501 は DA サーバのデフォルトの TDS ポート番号です。コマンド・ライン・ツールが `isql` を使用して DA サーバに接続している場合は、TDS ポートが必要です。

3. ローカル DA エージェントのデータベース・コネクションを作成します。

```

create connection conn_venus
    set agent=localagent
    and set host=venus
    and set port=5000
    and set database=pubs2
    and set user=sa
    and set password=''
go

create connection conn_pluto
    set agent=localagent
    and set host=pluto
    and set port=5000
    and set database=pubs2
    and set user=sa
    and set password=''
go

```

注意：この例では、組み込み DA エージェント `localagent` は、`venus` および `pluto` 上にインストールされた Adaptive Server データベースに接続します。Adaptive Server Enterprise では、ユーザ `sa` と null パスワードがデフォルトです。null パスワードでは、パスワード・パラメータを省略することもできます。

4. DA で定義されているデータベース・コネクションを次のように表示します。

```

show connection
go

```

NAME	TYPE	AGENT	HOST	PORT	DATABASE	USER	DESCRIPTION
conn_venus	ASE	localagent	venus	5000	pubs2	sa	
conn_pluto	ASE	localagent	pluto	5000	pubs2	sa	

(0 rows affected)

5. データベース・コネクションを次のようにテストします。

```

test connection conn_venus
go

```

RESULT

Succeeded
(0 rows affected)


```
test connection conn_pluto
go
RESULT
-----
Succeeded
(0 rows affected)
```

6. `au_id`、`au_lname`、および `au_fname` カラムのみを比較する、`authors` テーブル用のサンプルの比較セットを次のように作成します。

```
create compareset authors_demo1
with
source conn_venus dbo authors s1
target conn_pluto dbo authors t1
map
s1.au_id = t1.au_id set key=true
and s1.au_lname = t1.au_lname
and s1.au_fname = t1.au_fname
go
```

7. `authors` テーブル内のすべてのカラムを比較する、さらに複雑な比較セットを **where** 句を使用して作成します。state カラムの値が CA であるすべてのローを比較の対象から次のように除外します。

```
create compareset authors_demo2
with
source conn_venus dbo authors s1
where "state != 'CA'"
target conn_pluto dbo authors t2
where "state != 'CA'"
map
s1.au_id = t2.au_id set key=true
and s1.au_lname = t2.au_lname
and s1.au_fname = t2.au_fname
and s1.phone = t2.phone
and s1.address = t2.address
and s1.city = t2.city
and s1.state = t2.state
and s1.country = t2.country
and s1.postalcode = t2.postalcode
go
```

8. 比較セットを表示するには、次のように入力します。

```
show compareset authors_demo1
go
```

TYPE	CONNECTION	OWNER	TABLE	WHERE CONSTRAINT
S	conn_venus	dbo	authors	
T	conn_pluto	dbo	authors	

(0 rows affected)

```
show compareset authors_demo2
go
```

TYPE	CONNECTION	OWNER	TABLE	WHERE CONSTRAINT
S	conn_venus	dbo	authors	state != 'CA'

```
T      conn_pluto  dbo  authors state != 'CA'
(0 rows affected)
```

注意： 比較セットのカラムのマッピングを表示するには、**columns** オプションを **show compareset** コマンドで使用します。例：

```
show compareset authors_demo1 columns
```

9. authors_demo1 比較セットを使用してロー比較ジョブを作成するには、次のように入力します。

```
create job authors_job1
  add comparison cmp_authors1
  set compareset=authors_demo1
go
```

10. authors_demo2 比較セットを使用して別のジョブを作成し、比較オプションを明示的に設定するには、次のように入力します。

```
create job authors_job2
  set max_concurrent_comparisons = 10
  add comparison cmp_authors2
  set compareset=authors_demo2
  and set abort_diff_max to 1000
  and set abort_diff_row_count to true
  and set auto_reconcile to false
  and set compare_mode to row_compare
  and set compress_data_transfer to false
  and set create_col_log to false
  and set create_recon_script to false
  and set enable_row_count to true
  and set external_sort to false
  and set hash_type to database_hash
  and set priority to normal
  and set retry_delay_sec to 10
  and set retry_diff to wait_and_retry
  and set retry_max to 3
  with column option
    set city = literal
    and set postalcode to column_hash
go
```

注意： ジョブまたは比較オプションを変更するには、**alter job** を使用します。

11. 新しく作成されたジョブ authors_job2 を表示します。

```
show job authors_job2
go
```

```
OPTION                                VALUE
-----                                -
MAX_CONCURRENT_COMPARISONS 10
(0 rows affected)

COMPARISON  ACTIVE COMPARESET  PRIORITY COMPARE MODE RETRY
-----
```

```

cmp_authors2 true authors_demo2 NORMAL ROW_COMPARE WAIT_AND_RETRY
DESCRIPTION
-----
(0 rows affected)

SCHEDULE ACTIVE TYPE EVERY START DATE TIME KEEP KEEP UNIT CRON
-----
DESCRIPTION
-----
(0 rows affected)

```

12. 新しく作成されたジョブ authors_job1 の比較 cmp_authors1 を表示します。

```

show job authors_job1 cmp_authors1
go

OPTION                                VALUE
-----                                -
ABORT_DIFF_MAX                        1000
ABORT_DIFF_ROW_COUNT                  true
AUTO_RECONCILE                        false
COMPARE_MODE                          ROW_COMPARE
COMPRESS_DATA_TRANSFER                false
CREATE_COL_LOG                        false
CREATE_RECON_SCRIPT                   false
ENABLE_ROW_COUNT                      true
EXTERNAL_SORT                         false
HASH_TYPE                             DATABASE_HASH
RETRY_DELAY_SEC                       10
RETRY_DIFF                             NEVER
RETRY_MAX                              3
(0 rows affected)

COLUMN    COMPARE MODE
-----
au_fname ROW_HASH
au_id    LITERAL
au_lname ROW_HASH
(0 rows affected)

```

13. これらのジョブを次のように実行します。

```

run job authors_job1
go
(1 row affected)

```

```

run job authors_job2
go
(1 row affected)

```

14. ジョブの進行状態をモニタするには、次のように入力します。

```

monitor job authors_job1
go

COMPARISON    STATUS    SUBMIT TIME                END TIME                RUN PROGRESS
-----

```

導入ガイド

```
cmp_authors1 FINISHED 2011-11-15 21:26:20 2011-11-15 21:26:26 1 100%
```

```
NEXT RETRY ERROR
```

```
monitor job authors_job2  
go
```

```
COMPARISON STATUS SUBMIT TIME END TIME RUN PROGRESS
```

```
-----  
cmp_authors1 FINISHED 2011-11-15 21:26:35 2011-11-15 21:26:36 1 100%
```

```
NEXT RETRY ERROR
```

15. 各ジョブ内の個々の比較をモニタするには、次のように入力します。

```
monitor job authors_job1 cmp_authors1  
go
```

```
COMPARISON SUBMIT TIME END TIME  
-----  
cmp_authors1 2011-11-15 21:33:28 2011-11-15 21:33:29
```

```
(0 rows affected)
```

```
RUN PHASE TYPE SUMMARY START TIME  
END TIME COUNT READ M O I R PROGRESS ESTIMATE END  
ERROR  
-----  
1 COMPARE_ALL S conn_venus/dbo.authors 2011-11-15 21:33:28  
2011-11-15 21:33:29 23 23 100%  
T conn_pluto/dbo.authors 2011-11-15 21:33:28  
2011-11-15 21:33:29 23 23 0 0 0 100%
```

```
(0 rows affected)
```

```
monitor job authors_job2 cmp_authors2  
go
```

```
COMPARISON SUBMIT TIME END TIME  
-----  
cmp_authors2 2011-11-15 21:35:46 2011-11-15 21:35:50
```

```
(0 rows affected)
```

```
RUN PHASE TYPE SUMMARY START TIME  
END TIME COUNT READ M O I R PROGRESS ESTIMATE  
END ERROR  
-----  
1 COMPARE_ALL S conn_venus/dbo.authors 2011-11-15 21:35:46  
2011-11-15 21:35:46 8 8 100%  
T conn_pluto/dbo.authors 2011-11-15 21:35:46
```

```
2011-11-15 21:35:47 8      8      0 0 0      100%
(0 rows affected)
```

16. ジョブ履歴リストを表示するには、次のように入力します。

```
show history authors_job1
go
HISTORY ID SUBMIT TIME          FINISH TIME
-----
3          2011-11-15 21:33:28 2011-11-15 21:33:29
1          2011-11-15 21:26:19 2011-11-15 21:26:23
(0 rows affected)
```

17. 個々のジョブ履歴を表示するには、ジョブの HISTORY_ID 番号を次のように指定します。

```
show history authors_job1 3
go
COMPARISON  RUN  PHASE          TYPE  SUMMARY
-----
START TIME          END TIME          COUNT  READ M O I R
-----
ERROR
-----
cmp_authors1 1  COMPARE_ALL  S      venus:5000/pubs2.dbo.authors
2011-11-15 21:33:28 2011-11-15 21:33:29      23      23
                                T      pluto:5000/pubs2.dbo.authors
2011-11-15 21:33:28 2011-11-15 21:33:29      23      23 0 0 0
(0 rows affected)

FILE          SERVER PATH
-----
Text report  /Sybase/DA-15_5/server/instance/data/authors_job
/2011-11-15/21.33.28.795/report.txt
XML report   /Sybase/DA-15_5/server/instance/data/authors_job
/2011-11-15/21.33.28.795/report.xml
(0 rows affected)
```

出力の末尾に指定されているレポート・ファイルへのパスをここに入力します。

以下にテキスト・レポート・ファイルからの抜粋を示します。

```
source venus:5000/pubs2.dbo.authors
starttime 2011-11-15 21:33:00
endtime 2011-11-15 21:33:00

target pluto:5000/pubs2.dbo.authors
starttime 2011-11-15 21:33:00
endtime 2011-11-15 21:33:00
missing 0 orphaned 0 inconsistent 0
```

注意： サーバの設定パラメータ数が、ジョブのパフォーマンスに影響を及ぼすことがあります。**config** を使用して、これらの設定パラメータのデフォルト値を変更します。

分散配備

DA サーバとデータベース・サーバの間に大幅なネットワーク遅延があるときや、多数の同時比較が必要なとき、または配備とメンテナンスのしやすさよりもパフォーマンス要件が重要なときには、分散配備をお勧めします。

作業を始める前に

この例では、Data Assurance (DA) サーバ 1 台とリモート DA エージェント 2 台を使用しています。ローカル DA エージェントは使用していません。

表 2：配備の概要

コンポーネント名	マシン名	ポート番号
DA サーバ	mars	<ul style="list-style-type: none"> • 4500 – RMI • 4501 – TDS • 4503 – DASD
DA エージェント	venus	<ul style="list-style-type: none"> • 4510 – RMI • 4511 – TDS • 4512 – DTS
DA エージェント	pluto	<ul style="list-style-type: none"> • 4510 – RMI • 4511 – TDS • 4512 – DTS
Adaptive Server Enterprise	venus	5000 – サーバ
Adaptive Server Enterprise	pluto	5000 – サーバ

1. DA サーバのインスタンスを起動します。

```
$SYBASE/DA-15_5/server/instance/RUN_instance_64.sh
```

`$SYBASE` は、Data Assurance オプションをインストールしたディレクトリ、`instance` は DA サーバのインスタンス名、`RUN_instance_64.sh` は起動スクリプトです。

注意： Windows では、起動スクリプト・ファイルの名前は

`RUN_instance_32.bat` または `RUN_instance_64.bat` で、`instance` は DA

サーバのインスタンス名です。UNIX または Linux プラットフォームでは、ファイル名は `RUN_instance_64.sh` です。

2. `venus` および `pluto` という名前のマシンで DA エージェントのインスタンスを起動します。

```
$$SYBASE/DA-15_5/agent/instance/RUN_instance_64.sh
```

`$$SYBASE` は Data Assurance エージェントをインストールしたディレクトリ、`instance` は DA エージェントのインスタンス名、`RUN_instance_64.sh` は起動スクリプトです。

注意： Windows では、起動スクリプト・ファイルの名前は

`RUN_instance_32.bat` または `RUN_instance_64.bat` で、`instance` は DA サーバのインスタンス名です。UNIX または Linux プラットフォームでは、ファイル名は `RUN_instance_64.sh` です。

3. `isql` から、DA サーバに管理者としてログインします。

```
$$SYBASE/OCS-15_0/bin/isql -S mars:4501 -U da_admin -P password -w 250
```

注意： 4501 は DA サーバのデフォルトの TDS ポート番号です。コマンド・ライン・ツールが `isql` を使用して DA サーバに接続している場合は、TDS ポートが必要です。

DA エージェントのインスタンスにも同様にログインできます。例：

```
$$SYBASE/OCS-15_0/bin/isql -S venus:4511 -U da_admin -P password -w 250
```

4. DA サーバに接続するための DA エージェント・プロファイルを作成します。

```
create agent agent_venus
  set host=venus
  and set port=4510
  and set user=da_admin
  and set password=password
go
```

```
create agent agent_pluto
  set host=pluto
  and set port=4510
  and set user=da_admin
  and set password=password
go
```

5. 新しく作成された DA エージェントを表示します。

```
show agent
go
```

6. DA エージェントの接続設定をテストします。

```
test agent agent_venus
go
```

```
test agent agent_pluto
go
```

7. 新しい DA エージェントのデータベース・コネクションを作成します。

```
create connection conn_venus
  set agent=agent_venus
  and set host=venus
  and set port=5000
  and set database=pubs2
  and set user=sa
  and set password=' '
go
```

```
create connection conn_pluto
  set agent=agent_pluto
  and set host=pluto
  and set port=5000
  and set database=pubs2
  and set user=sa
  and set password=' '
go
```

注意： この例では、*venus* にインストールした Adaptive Server データベースに *agent_venus* が接続し、*pluto* にインストールした Adaptive Server に *agent_pluto* が接続します。

8. 新しく作成されたデータベース・コネクションを表示します。

```
show connection
go
```

9. シングルサーバ配備の場合は、前の例の手順 5 から続けます。

データベース・テーブルの調整

DA では、送信元データベースとターゲット・データベースの間の差異が自動的に調整されるか、データベース管理者がターゲット・データベースを手動で調整できるようにするための SQL スクリプトが作成されます。DA サーバは、この両方のタスクを同時に実行できるように設定できます。

この例では、ローカルな埋め込みエージェントのある単一の DA サーバを使用し、送信元テーブルとの差異として、消失ロー、一貫性のないロー、孤立したローが 1 つずつあるターゲット・テーブルを調整するスクリプトを生成する方法を説明します。

表 3 : 配備の概要

コンポーネント名	マシン名	ポート番号
DA サーバ	mars	<ul style="list-style-type: none"> • 4500 – RMI • 4501 – TDS • 4503 – DASD

コンポーネント名	マシン名	ポート番号
Adaptive Server Enterprise	venus	5000 – サーバ
Adaptive Server Enterprise	pluto	5000 – サーバ

1. シングルサーバ配備の例にある手順1～手順5に従い、DA サーバ・インスタンスの起動とデータベースへの接続を行います。
2. 新しい比較セットを作成して、送信元テーブル全体を次のようにマップします。

```
create compareset authors_demo3
with
  source conn_venus dbo authors s
  target conn_pluto dbo authors t
map
  s.au_id = t.au_id set key=true
  and s.au_lname = t.au_lname
  and s.au_fname = t.au_fname
  and s.phone = t.phone
  and s.address = t.address
  and s.city = t.city
  and s.state = t.state
  and s.country = t.country
  and s.postalcode = t.postalcode
go
```

警告！ DA サーバは、比較セット内でマップされているカラムのみを調整します。テーブルを部分的にマップする比較セットを調整用に使用すると、自動調整でのエラーおよび不完全な調整スクリプトが生成される場合があります。

3. 新しいジョブを次のように作成します。

```
create job authors_job3
  add comparison cmp_authors3
  set compareset = authors_demo3
  and set create_col_log = true
  and set create_recon_script = true
go
```

注意： ジョブ比較の `auto_reconcile` オプションを `true` に設定すると、データの差異が自動的に調整されます。

4. 新しいジョブを次のように実行します。

```
run job authors_job3
go
(1 row affected)
```

5. 次のようにジョブをモニタします。

```
monitor job authors_job3 cmp_authors3
go
```

導入ガイド

```
COMPARISON      SUBMIT TIME      END TIME
-----
cmp_authors3    2012-03-30 10:31:36  2012-03-30 10:31:42

(0 rows affected)

RUN PHASE          TYPE SUMMARY          START TIME          END
TIME              COUNT READ M O I R
-----
PROGRESS ESTIMATE END
ERROR
1 COMPARE_ALL      S   conn_venus/dbo.authors 2012-03-30 10:31:39
2012-03-30 10:31:39 23   23
100%
2012-03-30 10:31:39 23   T   conn_pluto/dbo.authors 2012-03-30 10:31:39
100%
2012-03-30 10:31:39 23   23 1 1 1
2 VERIFY_DIFFERENCES S           2012-03-30 10:31:40
2012-03-30 10:31:41 2
100%
2012-03-30 10:31:41 2           T           2012-03-30 10:31:40 2012-03-30
10:31:41 2           1 1 1
100%
```

6. ジョブ履歴 ID を次のように取得します。

```
show history authors_job3
go
```

```
HISTORY ID      SUBMIT TIME      FINISH TIME
-----
1               2012-03-30 10:31:36  2012-03-30 10:31:42
```

(0 rows affected)

7. 履歴 ID を次のように使用してジョブの履歴を表示します。

```
show history authors_job3 1
go
```

```
COMPARISON  RUN PHASE          TYPE SUMMARY
START TIME  END TIME
-----
COUNT READ M O I R ERROR
-----
cmp_authors3 1 COMPARE_ALL      S   venus:5000/pubs2.dbo.authors
2012-03-30 10:31:39 2012-03-30 10:31:39
23 23
2012-03-30 10:31:39 2012-03-30 10:31:39
23 23 1 1 1
T   pluto:5000/pubs2.dbo.authors
2 VERIFY_DIFFERENCES S
```

```

2012-03-30 10:31:40 2012-03-30 10:31:41
      2
                                     T
2012-03-30 10:31:40 2012-03-30 10:31:41
      2  1 1 1
      3 CREATE RECONCILIATION SCRIPT T
2012-03-30 10:31:41 2012-03-30 10:31:42
      3

(0 rows affected)

FILE                SERVER PATH
-----
Recon script C:\Sybase\DA-15_5\server\instance\data\authors_job3\
2012-03-30\10.31.36.099\cmp_authors3_T_recon.sql
Text report  C:\Sybase\DA-15_5\server\instance\data\authors_job3\
2012-03-30\10.31.36.099\report.txt
XML report   C:\Sybase\DA-15_5\server\instance\data\authors_job3\
2012-03-30\10.31.36.099\report.xml

(0 rows affected)

```

以下にテキスト・レポート・ファイルからの抜粋を示します。

```

source venus:5000/pubs2.dbo.authors
starttime 2012-03-30 10:31:39
endtime 2012-03-30 10:31:39

target pluto:5000/pubs2.dbo.authors
starttime 2012-03-30 10:31:40
endtime 2012-03-30 10:31:41
missing 1 orphaned 1 inconsistent 1

```

```

diff  |au_id      |au_lname  |au_fname  |phone      |
address  |city
-----|-----
| state  |country  |postalcode
-----|-----
      |172-32-1176  |Roberts   |Alex      |408 496-7223  |10932 Bigge
Rd.     |Menlo Park
|CA     |USA      |94025
      |172-32-1176  |White    |Johnson   |408 496-7223  |10932
Bigge Rd. |Menlo Park
|CA     |USA      |94025
|       |         |^^^^^^^^  |^^^^^^^^  |
|
|
O       |213-46-8915  |Green    |Marjorie  |415 986-7020  |309 63rd
St. #411 |Oakland
|CA     |USA      |94618

```

導入ガイド

```
M      |321-78-9087      |Jones      |Steve      |412 555-6434      |48 Barnaby
Close  |Walnut Creek
|CA      |USA      |94592

reconciliation script
  starttime 2012-03-30 10:31:41
  endtime 2012-03-30 10:31:42
  reconciled 3
(0 rows affected)
```

8. 調整スクリプトをターゲット・データベースに対して次のように実行します。

```
C:¥>isql -S pluto:5000 -U sa -i "C:¥Sybase¥DA-15_5¥server¥myserver¥data¥
authors_job3¥2012-03-30¥10.31.36.099¥cmp_authors3_T_recon.sql"

Password:
(1 row affected)
(1 row affected)
(1 row affected)
```

以下に調整スクリプトの例を示します。

```
--
-- Replication Server Data Assurance Option/15.7.1/DA Server/P/generic/
generic/dal57x/121/VM: Sun Microsystems Inc. 1.6.0_24/OPT/Tue 24 Apr
2012 09:24:31 GMT
-- Reconciliation Script (Auto-generated); fixes 3 difference(s).
-- Missing: 1 (insert)
-- Inconsistent: 1 (update)
-- Orphaned: 1 (delete)
--
-- Date Created: 2012-03-30 10:31:42
-- File encoding: UTF-8
--
-- Source: dbo.authors on venus:5000/pubs2
-- Target: dbo.authors on pluto:5000/pubs2
--
use pubs2
go
--
-- Missing: 1 rows
--
begin tran
insert into dbo.authors
au_id,au_lname,au_fname,phone,address,city,state,country,postalcode)
values ('321-78-9087','Jones','Steve','412 555-6434','48 Barnaby
Close','Walnut Creek','CA','USA','94592      ')
commit tran
go
--
-- Inconsistent: 1 rows
--
begin tran
update dbo.authors set au_lname = 'Roberts', au_fname = 'Alex' where
au_id = '172-32-1176'
```

```
commit tran
go
--
-- Orphaned: 1 rows
--
begin tran
delete from dbo.authors where au_id = '213-46-8915'
commit tran
go
```


管理タスク

比較ジョブの作成、Replication Server からのジョブのインポート、Data Assurance System Database のバックアップ、サーバ・パラメータの設定。

ジョブの作成

DA サーバでローの比較を実行します。

1. 次のいずれかを行います。
 - ローカル・エージェントを使用する。
 - インストール過程で作成したりリモート DA エージェントを選択する。
2. ソースおよびターゲットのデータベース接続プロファイルを作成します。
3. 比較セットを作成します。
4. ジョブを作成します。
5. ジョブを実行します。
6. 実行中のジョブをモニタします。
7. (オプション) ジョブ履歴を表示します。

参照：

- create agent (36 ページ)
- create connection (43 ページ)
- create compareset (51 ページ)
- create job (63 ページ)
- run job (79 ページ)
- monitor job (78 ページ)
- show history (80 ページ)

スキーマ・ジョブの作成

DA サーバでデータベース・オブジェクトのスキーマを比較します。

1. 次のいずれかを行います。
 - ローカル・エージェントを使用します。

管理タスク

- インストール過程で作成したリモート DA エージェントを選択します。
2. ソースおよびターゲットのデータベース接続プロファイルを作成します。
 3. スキーマ・ジョブを作成します。
 4. ジョブを実行します。
 5. (オプション) ジョブ履歴を表示します。

参照：

- create agent (36 ページ)
- create connection (43 ページ)
- create schema job (73 ページ)
- run job (79 ページ)
- show history (80 ページ)

Replication Server からのジョブのインポート

事前に定義されたテーブル複写定義とサブスクリプションに基づいて、ジョブを Replication Server システム・データベースから DA サーバにインポートします。

1. 次のいずれかを行います。
 - ローカル・エージェントを使用します。
 - インストール過程で作成したリモート DA エージェントを選択します。
2. ソースおよびターゲットのデータベース接続プロファイルを作成します。
3. RSSD データベース接続プロファイルを作成します。
4. Replication Server からジョブをインポートします。
5. (オプション) インポートされたジョブの比較オプションを変更します。
6. (オプション) インポートされたジョブのスケジュールを変更するか、インポートされたジョブにスケジュールを追加します。
7. ジョブを実行します。
8. 実行中のジョブをモニタします。

参照：

- create agent (36 ページ)
- create connection (43 ページ)
- import job (83 ページ)
- alter job (57 ページ)
- run job (79 ページ)

- monitor job (78 ページ)

サーバの設定パラメータの設定

DA サーバのジョブ実行方法を定義するサーバ設定パラメータを調整して、システム・パフォーマンスを向上させます。

1. デフォルトのサーバ設定パラメータを表示します。
2. 該当するサーバ設定パラメータのデフォルト値を変更します。

参照：

- config (91 ページ)

DASD のバックアップとリストア

現在の DASD のバックアップを作成し、バックアップ・コピーから DASD をリストアします。

1. 現在の DASD のバックアップを作成します。
2. バックアップ・コピーを表示します。
3. バックアップ・コピーから DASD をリストアします。
DASD のリストア後、DA サーバは自動的に停止します。
4. DA サーバを再起動します。

『Replication Server Data Assurance オプション・インストール・ガイド』の「インストール後の使用の開始」を参照してください。

参照：

- create backup (88 ページ)
- show backup (89 ページ)
- restore backup (89 ページ)

データとログ・ファイルの削除

ジョブ履歴と DASD バックアップを削除します。

- ジョブ履歴を削除するには、次のいずれかを使用します。
 - **drop history**

管理タスク

- **truncate history**

注意： **drop history** は履歴 ID によって履歴レコードを削除します。

- バックアップ (DASD コピー) を削除するには、次のいずれかを使用します。
 - **drop backup**
 - **truncate backup**

参照：

- drop history (77 ページ)
- truncate history (83 ページ)
- drop backup (89 ページ)
- truncate backup (90 ページ)

Data Assurance サーバのコマンド・リファレンス

DA サーバのコマンドは **isql** または Sybase Control Center Data Assurance プラグインを使用して実行できます。

注意： DA サーバのすべてのコマンドを実行するには、“da_admin” パーミッションが必要です。

エージェントのコマンド

DA サーバのエージェントを作成して管理するためのコマンドです。

alter agent

既存のエージェントの属性を変更します。1 つまたは複数のエージェントの属性を変更できます。

構文

```
alter agent agent_name
[set host [{{to|=}} host]
[and set port [{{to|=}} port]
[and set user [{{to|=}} user]
[and set password [{{to|=}} password]
[and set desc [{{to|=}} description]
```

パラメータ

- **agent_name** – エージェントの名前。
- **host** – エージェントがインストールされているマシンのホスト名。
- **port** – エージェントがインストールされているマシンのポート番号。
- **user** – 管理者のログイン名。
- **password** – ログイン名に関連付けられているパスワード。
- **description** – エージェントの説明。予約語または空白スペースを使用する場合は二重引用符を使用します。

例

- **例 1** – myagent のユーザ名およびパスワードを変更するには、次のコマンドを実行します。

```
alter agent myagent
and set user=youruser
and set password=yourpwd
go
```

create agent

エージェント・プロファイルを作成します。

構文

```
create agent agent_name
set host [{{to|=}}] host
and set port [{{to|=}}] port
and set user [{{to|=}}] user
[and set password [{{to|=}}] password]
[and set desc [{{to|=}}] description]
```

パラメータ

- **agent_name** – 作成するエージェントの名前。
- **host** – エージェントがインストールされているマシンのホスト名。
- **port** – エージェントがインストールされているマシンのポート番号。
- **user** – エージェントにアクセスするための Replication Server Data Assurance オプションの管理者のログイン名。
- **password** – (オプション) エージェントおよびデータベースに接続するための管理ユーザ・ログイン名用のパスワード。
- **description** – (オプション) エージェントの説明。予約語または空白スペースを使用する場合は二重引用符を使用します。

例

- **例 1** – 新しいエージェントを作成するには、次のコマンドを実行します。

```
create agent myagent
set host=myhost
and set port=1111
and set user=myuser
and set password=mypwd
go
```

参照：

- create connection (43 ページ)
- alter connection (42 ページ)
- test agent (41 ページ)

depend agent

指定したエージェントに依存する接続名のリストを表示します。

構文

```
depend agent agent_name
```

パラメータ

- **agent_name** – エージェントの名前。

例

- **例 1** – “myagent” の接続依存性を表示します。

```
depend agent myagent  
go
```

結果は次のようになります。

```
CONNECTION  
-----  
connection1  
connection2
```

参照：

- show connection (47 ページ)

drop agent

既存のエージェントを削除します。

構文

```
drop agent agent_name
```

パラメータ

- **agent_name** – 削除するエージェントの名前。

例

- **例 1** – システムから “myagent” を削除します。

```
drop agent myagent  
go
```

show agent

1つまたはすべてのエージェントの詳細を表示します。

構文

```
show agent [agent_name]
```

パラメータ

- **agent_name** – (オプション) 詳細を表示する対象となるエージェントの名前。このパラメータが指定されていない場合、すべてのエージェントの詳細が表示されます。

例

- **例 1** – すべてのエージェントについての情報を表示するには、次のコマンドを実行します。

```
show agent
go
```

結果は次のようになります。

NAME	HOST	PORT	USER	DESCRIPTION
localagent	localhost	0	localuser	
ragent	myuser	4510	da_admin	remote agent 1

show agent connection

エージェントのデータベース・コネクションを表示します。

構文

```
show agent connection agent_name
```

パラメータ

- **agent_name** – エージェントの名前。

例

- **例 1** – “localagent” のデータベース・コネクションを表示します。

```
show agent connection localagent
go
```

結果は次のようになります。

NAME	TYPE	CONNECTED
------	------	-----------

```
-----
conn_soka3    ASE    0
conn_soka2    ASE    0
```

show agent dts

指定したエージェントで実行している DTS (Data Transfer Stream) の情報を表示します。

構文

```
show agent dts agent_name
```

パラメータ

- **agent_name** – エージェントの名前。

例

- **例 1** – “agent1” のすべての DTS 情報を表示します。

```
show agent dts agent1
go
```

結果は次のようになります。

SERVER	TASK ID	CONNECTION	OBJECT	STAGE	OBJ PROCESSED
PREDICATE SQL PROCESSED					
myserver:4500@soka2	4	conn1	dbo.da1_10m	0	
0	1				
myserver:4500@soka2	2	conn2	dbo.da1_10m	0	
0	1				
myserver:4500@soka2	6	conn3	dbo.da1_10m	0	
0	1				

(0 rows affected)

TASK ID	ESTIMATE	COUNT	QUEUE	TAKEN	ESTIMATE	SECONDS	LEFT
4	10000000	0	10000000	0			
2	10000000	0	10000000	0			
6	10000000	0	10000000	0			

show agent jvm

リモート DA エージェントの Java Virtual Machine (JVM) の重要な詳細をいくつか表示します。

構文

```
show agent jvm agent_name
```

パラメータ

- **agent_name** – リモート・エージェントの名前。

例

- **例 1** – “myagent” の JVM の詳細を表示します。

```
show agent jvm myagent
go
```

結果は次のようになります。

```
JVM NAME                JVM INFO    JVM VENDOR
      JVM VERSION
-----
Java HotSpot(TM) Server VM mixed mode Sun Microsystems Inc.
      Java 1.6.0_24, VM 19.1-b02

(0 rows affected)

JVM TOTAL MEM JVM FREE MEM JVM MAX MEM
-----
31.8 MB        27.1 MB    455.1 MB
```

show agent system

DA エージェントの重要なシステム・プロパティをいくつか表示します。

構文

```
show agent system agent_name
```

パラメータ

- **agent_name** – エージェントの名前。

例

- **例 1** – “myagent” のシステムの詳細を表示します。

```
show agent system myagent
go
```

結果は次のようになります。

```
NAME      HOST          LOCALE    TIME_ZONE          DATE          TIME
-----
myagent   10.65.0.111  en_GB    Greenwich Mean Time 2011-06-10 16:05:44
```


OS NAME	OS VERSION	OS ARCH	OS LOAD AVG
Windows XP	5.1	x86	14.897%

show agent task

エージェントのタスク情報を表示します。

構文

```
show agent task agent_name
```

パラメータ

- **agent_name** – エージェントの名前。

例

- **例 1** – “localagent” のすべてのタスクを表示します。

```
show agent task localagent
go
```

結果は次のようになります。

TASK ID	CONNECTION	OBJECT	PREDICATE	SQL	PROCESSED
3	conn_soka2	dbo.dal_10m			

test agent

既存のエージェントが使用可能かどうかを確定化します。使用可能であれば、エージェントとの接続を確立して認証します。

構文

```
test agent agent_name
```

パラメータ

- **agent_name** – テストするエージェントの名前。

例

- **例 1** – エージェント “MyAgent” をテストします。

```
test agent MyAgent
go
```

結果は次のようになります。

```
RESULT
```

```
-----
Succeeded
```

接続プロファイルのコマンド

送信元とターゲットのデータベース接続プロファイルを作成して管理するためのコマンドです。

alter connection

既存の接続プロファイルの属性を変更します。

構文

```
alter connection connection_name
[set agent [{to|=}] agent_name]
[and set host [{to|=}] host]
[and set port [{to|=}] port]
[and set database [{to|=}] database_name]
[and set user [{to|=}] username]
[and set password [{to|=}] password]
[and set desc [{to|=}] description]
```

クラスタ・コネクション用のノード情報を変更するには、次のコマンドを実行します。

```
alter connection connection_name
with node
set host [{to|=}] host and set port [{to|=}] port
[and node set host [{to|=}] host and set port [{to|=}] port] ...
```

情報プロパティを変更するには、次のコマンドを実行します。

```
alter connection connection_name
with properties
set property_name [{to|=}] property_value
[and set property_name [{to|=}] property_value ...]
```

プロパティに関する情報を削除するには、次のコマンドを実行します。

```
alter connection connection_name
with properties drop property_name | ALL
```

パラメータ

- **connection_name** – 変更するコネクションの名前。
- **agent_name** – データベースへのコネクションを確立するエージェントの名前。
- **database_name** – ターゲット・データベースの名前。
- **host** – ターゲット・データベースがインストールされているマシンのホスト名。

- **port** – ターゲット・データベースがインストールされているマシンのポート番号。
- **user** – データベースのログイン名。ユーザには `select` パーミッションが必要です。一貫性のないロー、消失ロー、孤立したローをターゲット・データベース内で自動的に調整するには、ユーザに、`update`、`insert`、および `delete` の各パーミッションが付与されている必要があります。
- **password** – (オプション) ユーザのログイン名のパスワード。

注意： 使用している Adaptive Server のインスタンスにあるシステム管理者 (sa) ログインが null パスワードを使用している場合、空白のパスワードを設定するか、パスワードを設定しないことで、DA サーバでも null パスワードを使用する必要があります。

- **description** – エージェントの説明。予約語または空白スペースを使用する場合は二重引用符を使用します。
- **property_name** – データベースへのコネクションを確立する Sybase JDBC ドライバである jConnect™ for JDBC™ 用の接続プロパティ。

jConnect for JDBC 7.0 の『jConnect for JDBC 7.0 プログラマーズ・リファレンス』で、「プログラミング情報」の「接続の確立」の「接続プロパティ」を参照してください。

- **property_value** – 設定するプロパティの値を指定します。

例

- **例 1** – コネクション “MyConnPDB” のユーザ名とパスワードをリセットするには、次のコマンドを実行します。

```
alter connection MyConnPDB
set user=myuser2
and set password=mypwd2
go
```

結果は次のようになります。

```
Connection "MyConnPDB" was altered successfully
```

create connection

エージェントおよび JDBC のパラメータを含むデータベース接続プロファイルを作成します。

構文

```
create connection connection_name
set type [{to|=}] ASE | ASE_CLUSTER | RSSD
and set agent [{to|=}] agent_name
and set host [{to|=}] host
and set port [{to|=}] port
```

```
and set database [{{to|=}}] database_name
and set user [{{to|=}}] user
[and set password [{{to|=}}] password]
[and set desc [{{to|=}}] description]
[with node set host [{{to|=}}] host and set port [{{to|=}}] port
[and node set host [{{to|=}}] host
and set port [{{to|=}}] port]...]
[with properties set property_name [{{to|=}}] property_value
[and set property_name [{{to|=}}] property_value ]...]
```

パラメータ

- **connection_name** – 作成するコネクションの名前。
- **type** – (オプション) Replication Server Data Assurance オプションによってサポートされているコネクション・タイプは、Adaptive Server Enterprise (ASE)、ASE_CLUSTER、および Replication Server システム・データベース (RSSD) です。タイプが指定されていない場合でもノード・パラメータが指定されている場合、デフォルト値は ASE_CLUSTER になります。それ以外の場合、デフォルトは ASE になります。

ASE_CLUSTER コネクションに対してホスト名およびポート番号を設定することはできません。これらの値は、ノード定義内で設定する必要があります。コネクション・タイプが Replication Server システム・データベース (RSSD) である場合、**agent_name** は無効です。RSSD コネクションは、エージェントを定義できません。

- **agent_name** – データベースへのコネクションを確立するエージェントの名前。
- **database_name** – ターゲット・データベースの名前。
- **host** – ターゲット・データベースがインストールされているマシンのホスト名。
- **port** – ターゲット・データベースがインストールされているマシンのポート番号。
- **user** – データベースのログイン名。ユーザには select パーミッションが必要です。一貫性のないロー、消失ロー、孤立したローをターゲット・データベース内で自動的に調整するには、ユーザに、update、insert、および delete の各パーミッションが付与されている必要があります。
- **password** – (オプション) ユーザのログイン名のパスワード。

注意： 使用している Adaptive Server のインスタンスにあるシステム管理者 (sa) ログインが null パスワードを使用している場合、空白のパスワードを設定するか、パスワードを設定しないことで、DA サーバでも null パスワードを使用する必要があります。

- **description** – (オプション) エージェントの説明。予約語または空白スペースを使用する場合は二重引用符を使用します。
- **property_name** – データベースへのコネクションを確立する Sybase JDBC ドライバである jConnect for JDBC 用の接続プロパティ。

jConnect for JDBC 7.0 の『jConnect for JDBC 7.0 プログラマーズ・リファレンス』で、「プログラミング情報」の「接続の確立」の「接続プロパティ」を参照してください。

- **property_value** – 設定するプロパティの値。

例

- **例 1** – “MyConnPDB” という名前の Adaptive Server コネクションを作成するには、次のコマンドを実行します。

```
create connection MyConnPDB
set type=ASE
and set agent=pdbAgt
and set database=mypdb
and set host=myhost
and set port=1111
and set user=myuser
and set password=mypwd
go
```

結果は次のようになります。

```
Connection "MyConnPDB" was created successfully
```

- **例 2** – RSSD コネクションを作成するには、次のコマンドを実行します。

```
create connection MyRSSDConn
set type=RSSD
and set host=rshost
and set database=RSSD_database_name
and set port=2222
and set user=rsuser
and set password=rspwd
go
```

結果は次のようになります。

```
Connection "MyRSSDConn" was created successfully.
```

- **例 3** – Kerberos 認証サーバを使用してユーザを認証する Adaptive Server コネクションを作成するには、次のコマンドを実行します。

```
create connection MyConnPDB
set type=ASE
and set agent=pdbAgt
and set database=mypdb
and set host=myhost
and set port=1111
and set user=myuser
and set password=mypwd
with properties
set REQUEST_KERBEROS_SESSION=true
and set SERVICE_PRINCIPAL_NAME=myserver
go
```

結果は次のようになります。

```
Connection "MyConnPDB" was created successfully.
```

- **例 4** – ログイン・パスワードを暗号化する Adaptive Server コネクションを作成するには、次のコマンドを実行します。

```
create connection MyEncryptedPasswordConn
set type=ASE
and set agent=pdbAgt
and set database=mypdb
and set host=myhost
and set port=4901
and set user=myuser
and set password=mypwd
with properties
set ENCRYPT_PASSWORD=true
and set JCE_PROVIDER_CLASS="com.certicom.ecc.jcae.Certicom"
go
```

参照：

- create compareset (51 ページ)
- create schema job (73 ページ)
- show agent connection (38 ページ)
- test connection (48 ページ)

depend connection

指定したコネクションに依存する比較セット名のリストとスキーマ・ジョブ比較のリストを表示します。また、リストは、指定したコネクションを使用しているソース・データベースかターゲット・データベースかどうかを示します。

構文

```
depend connection connection_name
```

パラメータ

- **connection_name** – コネクションの名前。

例

- **例 1** – “MyConnPDB” の比較セットの依存性を表示します。

```
depend connection MyConnPDB
go
```

結果は次のようになります。

```
COMPARESET      TYPE
-----
compareset1     S
```

```

compareset2      T
SCHEMA JOB      COMPARISON      TYPE
-----
schema_job2     comparison2    S
schema_job3     comparison2    T

```

drop connection

既存の接続プロファイルを削除します。

構文

```
drop connection connection_name
```

パラメータ

- **connection_name** – 削除する接続の名前。

例

- **例 1** – 接続 “MyConnPDB” を削除するには、次のコマンドを実行します。

```
drop connection MyConnPDB
go
```

結果は次のようになります。

```
Connection "MyConnPDB" was dropped successfully
```

show connection

エージェントおよび JDBC のパラメータを含んでいる、既存の 0 以上の接続プロファイルを表示します。

構文

```
show connection [connection_name]
```

パラメータ

- **connection_name** – (オプション) 表示する接続の名前。このパラメータを指定しない場合、すべての接続が表示されます。

例

- **例 1** – 既存のすべてのデータベース接続プロファイルを表示するには、次のコマンドを実行します。

```
show connection
go
```

結果は次のようになります。

```
NAME  TYPE  AGENT      HOST  PORT  DATABASE  USER  DESCRIPTION
-----
pdb   ASE   localagent users  5010  pdb       sa    primary database
rdb   ASE   localagent users  5010  rdb       sa    replicate database

(0 rows affected)
```

test connection

データベースへのコネクションを確立および認証し、簡単なクエリを実行します。

構文

```
test connection connection_name
```

パラメータ

- **connection_name** – テストするコネクションの名前。

例

- **例1** – コネクション“MyConnPDB”をテストするには、次のコマンドを実行します。

```
test connection MyConnPDB
go
```

結果は次のようになります。

```
RESULT
-----
Succeeded
(0 rows affected)
```

比較セットのコマンド

比較セットを作成して管理するためのコマンドです。

alter compareset

既存の比較セットの属性を変更します。変更中の比較セットが実行中のジョブで使用されている場合、**alter compareset** は、実行に失敗します。

構文

説明を変更するには、次のコマンドを実行します。


```
alter compareset compareset_name [force]
set desc [{to|=}] description
```

ターゲット・テーブルを削除するには、次のコマンドを実行します。

```
alter compareset compareset_name [force]
drop target target_connection_name owner_name
target_table_name
[and target_connection_name owner_name
target_table_name ...]
```

ターゲット・コネクションを追加するには、次のコマンドを実行します。

```
alter compareset compareset_name [force]
  add target target_connection_name owner_name
  target_table_name target_alias [where "constraint"]
[and target target_connection_name owner_name
target_table_name target_alias2 where "constraint"...]
map
  source.column_name = target_alias.column_name
[ = target_alias2.column_name ...],
[and source.column_name = target_alias2.column_name
[ = target_alias2.column_name ...]...]
```

カラムのマッピングを削除するには、次のコマンドを実行します。

```
alter compareset compareset_name [force]
drop map source_column_name [and source_column_name...]
```

カラムのマッピングを追加するには、次のコマンドを実行します。

```
alter compareset compareset_name [force]
  with target target_connection_name owner_name
  target_table_name target_alias
[and target target_connection_name owner_name
target_table_name target_alias2]]
add map
  source.column_name = {target1}target_alias.column_name
[ = {target2}target_alias2.column_name] [set key=true],
[and source.column_name = {target1}target_alias2.column_name
[ = {target2}target_alias2.column_name] [set key=true]...]
```

where constraint を変更するには、次のコマンドを実行します。

```
alter compareset compareset_name [force]
[alter source where constraint]
[alter target target_connection_name owner_name
target_table_name where constraint]
```

注意：送信元の **where constraint** とターゲットの **where constraint** を同時に変更することはできません。

where constraint を削除するには、次のコマンドを実行します。

```
alter compareset compareset_name [force]
[alter source where ""]
[alter target target_connection_name owner_name
target_table_name where ""]
```

キー・カラムを変更するには、次のコマンドを実行します。

```
alter compareset compareset_name [force]
alter map
source.column_name set key {false | true}
[source.column_name set key {false | true}...]
```

パラメータ

- **compareset_name** – 比較セットの名前。
- **description** – エージェントの説明。予約語または空白スペースを使用する場合は二重引用符を使用します。
- **source_connection_name** – 送信元の接続の名前。
- **target_connection_name** – ターゲット・接続の名前。
- **owner_name** – 送信元テーブルまたはターゲット・テーブルの所有者の名前。
- **source_table_name** – 送信元テーブルの名前。
- **target_table_name** – ターゲット・テーブルの名前。
- **source_alias** – 送信元の接続、所有者、およびテーブルを参照するエイリアス。
- **target_alias** – ターゲット・接続、所有者、およびテーブルを参照するエイリアス。
- **column_name** – カラムの名前。
- **force** – 1つ以上の比較を指す比較セットを変更します。このような比較を変更するには、**force** を使用する必要があります。

例

- **例 1** – ターゲット・接続“conn_bak1”を既存の比較セット“cust_orders”から削除するには、次のコマンドを実行します。

```
alter compareset cust_orders
drop target conn_bak1 cust_owner cust_orders
go
```

結果は次のようになります。

```
Compareset "cust_orders" was altered successfully.
```

- **例 2** – 一部のカラム・マッピングを既存の比較セット“cust_orders”から削除するには、次のコマンドを実行します。

```
alter compareset cust_orders
drop map
id and cust_id
go
```

結果は次のようになります。

```
Compareset "cust_orders" was altered successfully.
```

使用法

- ターゲット・エイリアスを再定義しない場合、キーワード“target1”および“target2”を使用してターゲット・テーブルを参照できます。この場合、シーケンスは、比較セットを作成したときに使用した定義と一致している必要があります。
- **add map** 句を **alter compareset** で使用するときは、すべてのターゲット・コネクションを含める必要があります。

create compareset

データベース接続プロファイル、比較する送信元テーブルとターゲット・テーブルを含む比較セットを作成します。

構文

```
create compareset compareset_name
[set desc [{to|=}] description]
with source source_connection_name owner_name
source_table_name source_alias [where "constraint"]
target target_connection_name owner_name
target_table_name target_alias [where "constraint"]
[and target target_connection_name owner_name
target_table_name target_alias [where "constraint"...]]
map {all [set strict_column=true | false]
[set strict_type=true | false]
[set keep_computed=true | false]
[set keep_encrypted=true | false]}
source_table_alias.column_name = target_table_alias.column_name
[ = target_table_alias.column_name ...]
[set key=true ],
[and source_table_alias.column_name =
target_table_alias.column_name [ =
target_table_alias.column_name ...]
[set key=true]... ]
```

パラメータ

- **compareset_name** – 作成する比較セットの名前。
- **description** – (オプション) エージェントの説明。予約語または空白スペースを使用する場合は二重引用符を使用します。
- **source_connection_name** – 送信元のコネクションの名前。
- **target_connection_name** – ターゲット・コネクションの名前。
- **owner_name** – 送信元テーブルまたはターゲット・テーブルの所有者の名前。
- **source_table_name** – 送信元テーブルの名前。

- **target_table_name** – ターゲット・テーブルの名前。
- **source_alias** – 送信元の接続、所有者、およびテーブルを参照するエイリアス。
- **target_alias** – ターゲット・接続、所有者、およびテーブルを参照するエイリアス。
- **column_name** – カラムの名前。
- **key** – カラムがプライマリ・キーまたは一意のキーであるかどうかを指定します。

注意： 比較セット内には、少なくとも1つのキー・カラムを定義する必要があります。**map all** 文の使用時には、マッピングとキー・プロパティの両方がデータベースから派生します。

- **strict_column** – このパラメータが **true** に設定されている場合、1つ以上の送信元カラムがターゲット・テーブル内に存在しないと例外がスローされます。デフォルトでは、**strict_column** は **false** に指定されています。
- **strict_type** – このパラメータが **true** に設定されている場合、タイプ名、スケール、および精度がチェックされます。デフォルトでは、**strict_type** は **false** に指定されています。

注意： **strict_type**、**strict_column**、**keep_computed**、および **keep_encrypted** は、送信元データベースとターゲット・データベースをオンラインであることを要求する **map all** 文を使用する場合にのみ使用されます。明示的マッピングの場合は、データベースはオンラインである必要はありません。

例

- **例1** – “cust_orders” という名前の新しい比較セットを作成するには、次のコマンドを実行します。

```
create compareset cust_orders
with source conn_prod cust_owner cust_orders t1
where "id>100"
target conn_bak1 cust_owner cust_orders t2
where "id>100"
and target conn_bak2 cust_owner cust_orders t3
where "id>100"
map
t1.id = t2.id = t3.id set key=true
and t1.cust_id = t2.cust_id = t3.cust_id
and t1.sku = t2.sku = t3.sku
and t1.date = t2.date = t3.date
go
```

結果は次のようになります。

```
Compareset "cust_orders" was created successfully
```

- **例 2**–“cust_orders” という名前の新しい比較セットを作成し、すべてのターゲット・カラムと送信元カラムを自動的にマップするには、次のコマンドを実行します。

```
create compareset cust_orders
with source conn_prod cust_owner cust_orders t1
where "id>100"
target conn_bak1 cust_owner cust_orders t2
where "id>100"
and target conn_bak2 cust_owner cust_orders t3
where "id>100"
map all
go
```

結果は次のようになります。

```
Compareset "cust_orders" was created successfully
```

使用法

- すべてのターゲットには、同じ数のマップ・カラムが存在する必要があります。
- 少なくとも 1 つのキー・カラムが必要です。複数のキー・カラムを使用することもできます。

参照：

- create connection (43 ページ)
- create job (63 ページ)

制限事項

create compareset の **map all** 句には複数の制限があります。

- **map all** 文を **create compareset** で使用する場合は、送信元データベース・テーブルとターゲット・データベース・テーブルにプライマリ・キー・カラムまたは identity カラムが少なくとも 1 つ定義されている必要があります。たとえば、プライマリ・キーまたは identity カラムのあるテーブルを作成すると、**map all** 文が機能します。

```
CREATE TABLE orders (
  order_num INTEGER NOT NULL PRIMARY KEY,
  date_ordered DATE,
  name CHAR(80)
)
```

```
CREATE TABLE orders (
  order_num INTEGER IDENTITY,
  date_ordered DATE,
  name CHAR(80)
)
```

- テーブルにプライマリ・キーまたは identity カラムがないと、**sp_primarykey** ストアド・プロシージャを使用してプライマリ・キーを強制的に適用しても、**map all** は機能しません。

たとえば、次のように定義されたテーブルは **map all** でマップできません。

```
CREATE TABLE orders (  
    order_num INTEGER NOT NULL,  
    date_ordered DATE,  
    name CHAR(80)  
)  
sp_primarykey orders, order_num  
go  
create unique clustered index ordernumidx on orders(order_num)  
go
```

この場合、**sp_primarykey** システム・プロシージャを使用して定義されたプライマリ・キーがあるテーブルには、まだプライマリ・キーの整合性制約が欠けているため、**map all** 句を使用できません。このような場合は、プライマリ・キーの整合性制約が実行されるように、テーブルを変更できます。例：

```
drop index orders.order_num  
go  
alter table orders  
add constraint order_num_pk  
primary key (order_num)  
go
```

sp_helpconstraint システム・プロシージャを使用して、テーブルにプライマリキー制約があることを確認できます。『Adaptive Server Enterprise Transact-SQL ユーザーズ・ガイド』の「データベースおよびテーブルの作成」の「テーブルの整合性制約の定義」で「一意性制約およびプライマリ・キー制約の指定」を参照してください。

- **map all** 文を **create compareset** で使用する場合は、送信元とターゲットのテーブル・カラムが同じ名前である必要があります。

depend compareset

指定した比較セットに依存する (スキーマ以外の) ジョブ比較のリストを表示します。

構文

```
depend compareset compareset_name
```

パラメータ

- **compareset_name** – 比較セットの名前。

例

- **例 1** – “cust_orders” のジョブ比較依存性を表示します。

```
depend compareset cust_orders
go
```

結果は次のようになります。

```
JOB COMPARISON
-----
job4/comparison3
job5/comparison1
```

参照：

- show job (69 ページ)

drop compareset

既存の比較セットを削除します。この比較セットが既存のジョブで使用されている場合、**drop compareset** は、実行に失敗します。

構文

```
drop compareset compareset_name
```

パラメータ

- **compareset_name** – 削除する比較セットの名前。

例

- **例 1** – 比較セット “cust_orders” を削除するには、次のコマンドを実行します。

```
drop compareset cust_orders
go
```

結果は次のようになります。

```
Compareset "cust_orders" was dropped successfully
```

show compareset

データベース接続プロファイル、比較対象のテーブル、およびカラム・マッピングを含む 0 個以上の比較セットを表示します。

構文

```
show compareset [compareset_name[columns]]
```

パラメータ

- **compareset_name** – (オプション) 表示する比較セットの名前。

例

- **例 1** – 比較セット “cust_orders” を表示するには、次のコマンドを実行します。

```
show compareset cust_orders
go
```

結果は次のようになります。

TYPE	CONNECTION	OWNER	TABLE	WHERE	CONSTRAINT
S	conn_prod	dbo	cust_orders		
T	conn_bak1	dbo	cust_orders		
T	conn_bak2	dbo	cust_orders		

(0 rows affected)

- **例 2** – すべての比較セットを表示するには、次のコマンドを実行します。

```
show compareset
go
```

結果は次のようになります。

NAME	DESCRIPTION
cust	The customer tables.
cust_orders	The customer orders tables.

- **例 3** – 比較セット “cust_orders” 内のカラム・マッピングを表示するには、次のコマンドを実行します。

```
show compareset cust_orders columns
go
```

結果は次のようになります。

TYPE	CONN	TABLE	MAP_ID	COL_NAME	COL_TYPE	P_KEY
S	conn_prod	cust_orders	1	id	numeric(11)	Y
T	conn_bak1	cust_orders	1	id	numeric(11)	Y
T	conn_bak2	cust_orders	1	id	numeric(11)	Y
S	conn_prod	cust_orders	2	cust_id	numeric(9)	N
T	conn_bak1	cust_orders	2	cust_id	numeric(9)	N
T	conn_bak2	cust_orders	2	cust_id	numeric(9)	N
S	conn_prod	cust_orders	3	sku	varchar(50)	N
T	conn_bak1	cust_orders	3	sku	varchar(50)	N
T	conn_bak2	cust_orders	3	sku	varchar(50)	N
S	conn_prod	cust_orders	4	date	datetime	N

T	conn_bak1	cust_orders	4	date	datetime	N
T	conn_bak2	cust_orders	4	date	datetime	N

ロー比較ジョブのコマンド

ロー比較ジョブを作成して管理するためのコマンドです。

alter job

既存のジョブの属性を変更します。

構文

ジョブ比較を削除するには、次のコマンドを実行します。

```
alter job job_name
  drop comparison comparison_name [and comparison_name2 [...]]
```

ジョブ比較を追加するには、次のコマンドを実行します。

```
alter job job_name
  add comparison comparison_name
  set COMPARESET =compareset_name
  [and set ABORT_DIFF_MAX [{to|=}] number_of_differences
  [and set ABORT_DIFF_ROW_COUNT [{to|=}] {true|false}
  [and set AUTO_RECONCILE [{to|=}] {true|false}
  [and set COMPARE_MODE [{to|=}] {row_compare | key_compare |
row_checksum | table_checksum | row_count}
  [and set COMPRESS_DATA_TRANSFER [{to|=}] {true|false}
  [and set CREATE_COL_LOG [{to|=}] {true|false}
  [and set CREATE_RECON_SCRIPT [{to|=}] {true|false}
  [and set DESC [{to|=}] description
  [and set ENABLE_ROW_COUNT [{to|=}] {true|false}
  [and set EXTERNAL_SORT [{to|=}] {true|false}
  [and set HASH_TYPE [{to|=}] {database_hash | agent_hash}
  [and set PRIORITY [{to|=}] {highest | high | normal | low}
  [and set RETRY_DELAY_SEC [{to|=}] number_delay_second
  [and set RETRY_DIFF [{to|=}] {never | wait_and_retry }
  [and set RETRY_MAX [{to|=}] number_of_retries
  ]]]]]]]]]]]]]]]]
  [with column option
  set column_name [{to|=}] {literal | column_hash | row_hash}
  [set column_name [{to|=}] {literal | column_hash | row_hash}}
  [...]]
  [and comparison comparison_name2
  ...]
```

ジョブ比較を変更するには、次のコマンドを実行します。

```
alter job job_name
  alter comparison comparison_name
  [and set ABORT_DIFF_MAX [{to|=}] number_of_differences
  [and set ABORT_DIFF_ROW_COUNT [{to|=}] {true|false}
  [and set AUTO_RECONCILE [{to|=}] {true|false}
```



```
[and set KEEP_UNIT [{to|=}] {day | week | month | forever}
[and set CRON [{to|=}] cron_value
[and set DESC [{to|=}] description
]]]]]]]
    [and schedule schedule_name2
[set TYPE [{to|=}] {once | cron | every_day | every_week |
every_month}
[and set EVERY [{to|=}] n
[and set DATE [{to|=}] date_value
[and set TIME [{to|=}] time_value
[and set KEEP [{to|=}] keep_value
[and set KEEP_UNIT [{to|=}] {day | week | month | forever}
[and set CRON [{to|=}] cron_value
[and set DESC [{to|=}] description
]]]]]]].....}]
```

スケジュール・オプションを削除するには、次のコマンドを実行します。

```
alter job job_name
drop schedule schedule_name[and schedule_name2[.....]]
```

パラメータ

- **job_name** – ジョブの名前。
- **comparison_name** – ジョブに追加する比較の名前。
- **compareset_name** – 比較内に追加する比較セットの名前。
- **schedule_name** – 追加するスケジュールの名前。
- **max_concurrent_comparisons** – このジョブを使用して同時実行できる比較の数。
- **description** – エージェントの説明。予約語または空白スペースを使用する場合は二重引用符を使用します。

表 4: 比較オプション

パラメータ	値
ABORT_DIFF_MAX	差異のカウントが指定値を超えた場合、ロー比較をアポートする。 有効な値：1 ~ 9223372036854775807 デフォルト値：1000
ABORT_DIFF_ROW_COUNT	テーブルのロー・カウントが一致しない場合にロー比較を中止するかどうかを決定する。 有効な値：true または false デフォルト：false

パラメータ	値
AUTO_RECONCILE	調整スクリプトを自動的に適用するかどうかを示す。 有効な値：true または false デフォルト：false 注意： AUTO_RECONCILE を有効にするには、 CREATE_COL_LOG を true に設定します。
COMPARE_MODE	ロー比較モードを指定する。 <ul style="list-style-type: none"> • row_compare – すべてのテーブル・ローを比較する。 • key_compare – プライマリ・キー・カラムを比較する。 • row_count – ロー・カウントを比較する。 デフォルト：row_compare
COMPRESS_DATA_TRANSFER	ロー・データをエージェントとサーバの間で比較する。 有効な値：true または false デフォルト：false
CREATE_COL_LOG	すべての消失ローの値、孤立したローの値、一貫性のないローの値 (キーおよびカラム) をリストしたカラムの差異についてのログを生成する。以下を実行する場合は、カラム・ログを作成する。 <ul style="list-style-type: none"> • 調整スクリプトを生成する • 自動調整を実行する • 詳細なレポートを生成する 有効な値：true または false デフォルト：false
CREATE_RECON_SCRIPT	調整スクリプトを生成する。このパラメータを使用するには、 CREATE_COL_LOG も true に設定する必要がある。 有効な値：true または false デフォルト：false

パラメータ	値
ENABLE_ROW_COUNT	<p>送信元テーブルおよびターゲット・テーブルのローを比較前にカウントするかどうかを指定する。DA サーバは、比較の進行状況と終了時刻を見積もるためにロー・カウントを使用する。</p> <p>注意： COMPARE_MODE が <code>row_count</code> の場合に、DA サーバはローをカウントします。ENABLE_ROW_COUNT は、COMPARE_MODE の値が <code>row_count</code> 以外の場合にのみ使用してください。</p> <p>有効な値：true または false デフォルト：true</p>
EXTERNAL_SORT	<p>エージェント上のローをソートすることで、データベースで ORDER BY 句の処理の影響を低減する。</p> <p>有効な値：true または false デフォルト：false</p>
HASH_TYPE	<p>比較のハッシュ・タイプを指定する。</p> <ul style="list-style-type: none"> • <code>database_hash</code> – データベースによって指定されたハッシュ関数を使用する。 • <code>agent_hash</code> – Replication Server Data Assurance オプションによって指定されたハッシュ関数を使用する。 <p>デフォルト：<code>database_hash</code></p>
PRIORITY	<p>比較キュー内のジョブ比較の順序を指定する。</p> <p>正しい値は、次のとおり。</p> <ul style="list-style-type: none"> • <code>highest</code> • <code>high</code> • <code>normal</code> • <code>low</code> <p>デフォルト：<code>normal</code></p>

パラメータ	値
RETRY_DIFF	リトライ・オプションを指定する。 <ul style="list-style-type: none"> • never – 再比較は実行されない。 • wait_and_retry – RETRY_MAX の設定および RETRY_DELAY_SEC の設定に基づいて再比較が実行される。 デフォルト：never
RETRY_DELAY_SEC	再比較と再比較の遅延の秒数を指定する。 有効な値：0 ~ 86400 デフォルト：10
RETRY_MAX	前の比較と異なる結果があるローの再比較を行う合計回数を指定する。 有効な値：0 ~ 100 デフォルト：3

表 5：カラム比較オプション

カラム・オプション	値
COMPARE_MODE	各カラムの比較方法を指定する。 <ul style="list-style-type: none"> • column_hash – カラムのハッシュ値を使用して比較を実行する。 • row_hash – 合計ハッシュ値を使用して、このオプションが指定されたすべてのカラムを一緒に比較する。 • literal – カラムのリテラル値を使用して比較を実行する。

表 6：スケジュール・オプション

パラメータ	値
date_value	スケジュールで日付を指定する。
time_value	スケジュールで時刻を指定する。
keep_value	スケジュールをアクティブに保つ期間の単位の数を指定する。
cron_value	スケジュールで cron オプションの値を指定する。

例

- **例1** – “myjob_1” を変更してその比較を削除するには、次のコマンドを実行します。

```
alter job myjob_1
drop comparison mycomparison_1
go
```

結果は次のようになります。

```
Job "myjob_1" was altered successfully.
```

- **例2** – “myjob_1” を変更して比較を追加するには、次のコマンドを実行します。

```
alter job myjob_1
add comparison mycomparison_2
set compareset=mycompareset_2
and set priority = high
go
```

結果は次のようになります。

```
Job "myjob_1" was altered successfully.
```

create job

1つ以上の比較セット、スケジュール、および比較オプションから新しいジョブを作成します。

構文

```
create job job_name
[set MAX_CONCURRENT_COMPARISONS [{{to|=}}
number_of_max_concurrent_comparisons]
[and set DESC [{{to|=}}] description]
add comparison comparison_name
set COMPARESET=compareset_name
[and set ABORT_DIFF_MAX [{{to|=}}] number_of_differences
[and set ABORT_DIFF_ROW_COUNT [{{to|=}}] {true|false}
[and set AUTO_RECONCILE [{{to|=}}] {true|false}
[and set COMPARE_MODE [{{to|=}}] {row_compare | key_compare |
row_checksum | table_checksum | row_count}
[and set COMPRESS_DATA_TRANSFER [{{to|=}}] {true|false}
[and set CREATE_COL_LOG [{{to|=}}] {true|false}
[and set CREATE_RECON_SCRIPT [{{to|=}}] {true|false}
[and set DESC [{{to|=}}] description
[and set ENABLE_ROW_COUNT [{{to|=}}] {true|false}
[and set EXTERNAL_SORT [{{to|=}}] {true|false}
[and set HASH_TYPE [{{to|=}}] {database_hash | agent_hash}
[and set PRIORITY [{{to|=}}] {highest | high | normal | low}
[and set RETRY_DELAY_SEC [{{to|=}}] number_delay_second
[and set RETRY_DIFF [{{to|=}}] {never | wait_and_retry }
[and set RETRY_MAX [{{to|=}}] number_of_retries
]]]]]]]]]]]]]]]]
[with column option
```

```

set column_name [{to|=}] {literal | column_hash | row_hash}
[set column_name [{to|=}] {literal | column_hash | row_hash}]
[...]]

[add schedule schedule_name
[set TYPE [{to|=}] {once | cron | every_day | every_week |
every_month}
[and set EVERY [{to|=}] n
[and set DATE [{to|=}] date_value
[and set TIME [{to|=}] time_value
[and set KEEP [{to|=}] keep_value
[and set KEEP_UNIT [{to|=}] {day | week | month | forever}
[and set CRON [{to|=}] cron_value
[and set DESC [{to|=}] description
]]]]]]]]]]

```

既存のジョブを複製するには、次のコマンドを実行します。

```
create job job_name with exist_job_name
```

注意：スケジュールのあるジョブを複製すると、新しいジョブには複製されたスケジュールも含まれますが、それらのスケジュールは常にアクティブでない状態になっています。

パラメータ

- **job_name** – 作成するジョブの名前。
- **exist_job_name** – 複製する既存のジョブの名前。
- **comparison_name** – ジョブに追加する比較の名前。
- **compareset_name** – 比較内に追加する比較セットの名前。
- **schedule_name** – 追加するスケジュールの名前。
- **max_concurrent_comparisons** – (オプション) このジョブを使用して同時実行できる比較の数。デフォルト値は5です。
- **description** – (オプション) エージェントの説明。予約語または空白スペースを使用する場合は二重引用符を使用します。

表 7：比較オプション

パラメータ	値
ABORT_DIFF_MAX	差異のカウントが指定値を超えた場合、ロー比較をアポートする。 有効な値：1 ~ 9223372036854775807 デフォルト値：1000

パラメータ	値
ABORT_DIFF_ROW_COUNT	<p>テーブルのロー・カウントが一致しない場合にロー比較を中止するかどうかを決定する。</p> <p>有効な値：true または false</p> <p>デフォルト：false</p>
AUTO_RECONCILE	<p>調整スクリプトを自動的に適用するかどうかを示す。</p> <p>有効な値：true または false</p> <p>デフォルト：false</p> <hr/> <p>注意： AUTO_RECONCILE を有効にするには、CREATE_COL_LOG を true に設定します。</p>
COMPARE_MODE	<p>ロー比較モードを指定する。</p> <ul style="list-style-type: none"> • row_compare – すべてのテーブル・ローを比較する。 • key_compare – プライマリ・キー・カラムを比較する。 • row_count – ロー・カウントを比較する。 <p>デフォルト：row_compare</p>
COMPRESS_DATA_TRANSFER	<p>ロー・データをエージェントとサーバの間で比較する。</p> <p>有効な値：true または false</p> <p>デフォルト：false</p>
CREATE_COL_LOG	<p>すべての消失ローの値、孤立したローの値、一貫性のないローの値(キーおよびカラム)をリストしたカラムの差異についてのログを生成する。以下を実行する場合は、カラム・ログを作成する。</p> <ul style="list-style-type: none"> • 調整スクリプトを生成する • 自動調整を実行する • 詳細なレポートを生成する <p>有効な値：true または false</p> <p>デフォルト：false</p>
CREATE_RECON_SCRIPT	<p>調整スクリプトを生成する。このパラメータを使用するには、CREATE_COL_LOG も true に設定する必要がある。</p> <p>有効な値：true または false</p> <p>デフォルト：false</p>

パラメータ	値
ENABLE_ROW_COUNT	<p>送信元テーブルおよびターゲット・テーブルのローを比較前にカウントするかどうかを指定する。DA サーバは、比較の進行状況と終了時刻を見積もるためにロー・カウントを使用する。</p> <p>注意： COMPARE_MODE が <code>row_count</code> の場合に、DA サーバはローをカウントします。 ENABLE_ROW_COUNT は、 COMPARE_MODE の値が <code>row_count</code> 以外の場合にのみ使用してください。</p> <p>有効な値：true または false デフォルト：true</p>
EXTERNAL_SORT	<p>エージェント上のローをソートすることで、データベースで ORDER BY 句の処理の影響を低減する。</p> <p>有効な値：true または false デフォルト：false</p>
HASH_TYPE	<p>比較のハッシュ・タイプを指定する。</p> <ul style="list-style-type: none"> • <code>database_hash</code> – データベースによって指定されたハッシュ関数を使用する。 • <code>agent_hash</code> – Replication Server Data Assurance オプションによって指定されたハッシュ関数を使用する。 <p>デフォルト：<code>database_hash</code></p>
PRIORITY	<p>比較キュー内のジョブ比較の順序を指定する。</p> <p>正しい値は、次のとおり。</p> <ul style="list-style-type: none"> • <code>highest</code> • <code>high</code> • <code>normal</code> • <code>low</code> <p>デフォルト：<code>normal</code></p>

パラメータ	値
RETRY_DIFF	リトライ・オプションを指定する。 <ul style="list-style-type: none"> • never – 再比較は実行されない。 • wait_and_retry – RETRY_MAX の設定および RETRY_DELAY_SEC の設定に基づいて再比較が実行される。 デフォルト：never
RETRY_DELAY_SEC	再比較と再比較の遅延の秒数を指定する。 有効な値：0 ~ 86400 デフォルト：10
RETRY_MAX	前の比較と異なる結果があるローの再比較を行う合計回数を指定する。 有効な値：0 ~ 100 デフォルト：3

表 8：カラム比較オプション

カラム・オプション	値
COMPARE_MODE	各カラムの比較方法を指定する。 <ul style="list-style-type: none"> • column_hash – カラムのハッシュ値を使用して比較を実行する。 • row_hash – 合計ハッシュ値を使用して、このオプションが指定されたすべてのカラムを一緒に比較する。 • literal – カラムのリテラル値を使用して比較を実行する。

表 9：スケジュール・オプション

パラメータ	値
date_value	スケジュールで日付を指定する。
time_value	スケジュールで時刻を指定する。
keep_value	スケジュールをアクティブに保つ期間の単位の数を指定する。
cron_value	スケジュールで cron オプションの値を指定する。

例

- **例1**–“myjob_1”という名前の新しいジョブを作成するには、次のコマンドを実行します。

```
create job myjob_1
set max_concurrent_comparisons = 3
add comparison mycomparison_1
set compareset=mycompareset_1
and set priority = high
with column option
and set a = literal
set b = hash
and comparison mycomparison_2
set compareset=mycompareset_2
and set priority = normal
with schedule myschedule_1
set type=every_day
and set every=2
and set time=10:00
and set keep=1
and set keep_unit=month
and set date=2011-05-05
go
```

結果は次のようになります。

```
Job "myjob_1" was created successfully.
```

- **例2**–“myjob_1”を新しいジョブ“myjob_2”に複製するには、次のコマンドを実行します。

```
create job myjob_2 with myjob_1
go
```

結果は次のようになります。

```
Job "myjob_2" was created successfully.
```

使用法

- 複製されたジョブ内の比較の名前は、自動的に生成されます (明示的に指定されなかった場合)。命名規則は *job_name_cloned_comparison_sequence_number* で、*sequence_number* は1から始まります。
- ジョブの複製時には、スケジューリングのオプションのみを再定義できます。比較オプションは、既存のジョブから自動的にインポートされます。

drop job

既存のジョブを削除します。

構文

```
drop job job_name
```

パラメータ

- **job_name** – 削除するジョブの名前。

例

- **例 1** – “myjob_1” を削除するには、次のコマンドを実行します。

```
drop job myjob_1
go
```

結果は次のようになります。

```
Job 'myjob_1' was dropped successfully.
```

show job

それぞれが 1 つまたは複数の比較を含むゼロ以上の既存のジョブを表示します。

構文

```
show job [job_name [,comparison_name]]
```

パラメータ

- **job_name** – 表示するジョブの名前。
- **comparison_name** – 表示する比較の名前。

例

- **例 1** – 既存のジョブをステータスと共に表示します。

```
show job
go
```

結果は次のようになります。

Name	Active	Description
-----	-----	-----
job_1	true	my_job1
job_2	true	my_job2
job_3	true	my_job3

スキーマ比較ジョブのコマンド

スキーマ比較ジョブを作成して管理するためのコマンドです。

alter schema job

既存のスキーマ・ジョブの属性を変更します。

構文

スキーマ・ジョブに比較を追加するには、次のコマンドを実行します。

```
alter schema job sc_job_name
add comparison comparison_name
    With source source_connection_name source_connection_name_alias
    target target_connection_name target_connection_name_alias
        [and target target2_connection_name
target2_connection_name_alias]...
    [include all tables]
    [map tables

source_connection_name_alias.source_object_schema.source_object_name=
target_connection_name_alias.target_object_schema.target_object_name[=
target2_connection_name_alias.target2_object_schema.target2_object_name]
...]
    [and
source_connection_name_alias.source_object2_schema.source_object2_name=
target_connection_name_alias.target_object2_schema.target_object2_name[=
target2_connection_name_alias.target2_object2_schema.target2_object2_name]...]
    ]
    [exclude tables
source_object_schema.source_object_name
[and source_object2_schema.source_object2_name]...]
```

スキーマ・ジョブから比較を削除するには、次のコマンドを実行します。

```
alter schema job sc_job_name
drop comparison comparison_name
```

スキーマ比較のターゲット・コネクションを変更するスキーマ比較の変更を行うには、次のコマンドを実行します。

```
alter schema job sc_job_name
alter comparison comparison_name
drop target target_connection_name
    [and target2_connection_name]...
```

ターゲット・コネクションを追加するスキーマ比較の変更を行うには、次のコマンドを実行します。

```
alter schema job sc_job_name
alter comparison comparison_name
    add target new_target_connection_name
    new_target_connection_name_alias
        [and target new_target2_connection_name
new_target2_connection_name_alias]...
[map tables
source.source_object_schema.source_object_name=
new_target_connection_name_alias.new_target_object_schema.new_target_object_name[=
new_target2_connection_name_alias.new_target2_object_schema.new_target2_object_name]...
[and source.source_object2_schema.source_object2_name=
new_target_connection_name_alias.new_target_object2_schema.new_target_object2_name[=
new_target2_connection_name_alias.new_target2_object2_schema.new_target2_object2_name]...]]
```

テーブル・マッピングを削除するスキーマ比較の変更を行うには、次のコマンドを実行します。

```
alter schema job sc_job_name
alter comparison comparison_name
drop map tables
    source.source_object_schema.source_object_name
        [and
source_connection_name.source_object2_schema.source_object2_name]...
```

テーブル・マッピングを追加するスキーマ比較の変更を行うには、次のコマンドを実行します。

```
alter schema job sc_job_name
alter comparison comparison_name
add map tables
source_connection_name.source_object_schema.source_object_name=[
new_target_connection_name.new_target_object_schema.new_target_object_name[=
new_target_connection_name2.new_target_object_schema2.new_target_object_name2]...]]...
```

注意：新たに追加されたマップ・エントリのキーが既存のマップに含まれている場合、追加されたマップ・エントリは既存のマップ・エントリよりも優先されません。

テーブル除外を追加するスキーマ比較の変更を行うには、次のコマンドを実行します。

```
alter schema job sc_job_name
alter comparison comparison_name
add map tables
source_connection_name.source_object_schema.source_object_name=[
new_target_connection_name.new_target_object_schema.new_target_object_name[=
```

```
new_target_connection_name2.new_target_object_schema2.new_target_object_name2]...]...
```

スキーマ比較に **all tables** 句を追加するには、次のコマンドを実行します。

```
alter schema job sc_job_name
alter comparison comparison_name
add include all tables
```

include all tables 句を削除するスキーマ比較の変更を行うには、次のコマンドを実行します。

```
alter schema job sc_job_name
alter comparison comparison_name
drop include all tables
```

スキーマ比較のジョブ・オプションを変更するには、次のコマンドを実行します。

```
alter schema job sc_job_name
set max_concurrent_comparisons [{to|=}]
number_of_max_concurrent_comparisons
[and set desc [{to|=}] description]
```

パラメータ

- **sc_job_name** – スキーマ比較ジョブの名前。
- **comparison_name** – スキーマ比較ジョブの名前。
- **source_connection_name** – 送信元の接続の名前。
- **source_connection_name_alias** – 送信元の接続のエイリアス名。
- **target_connection_name** – ターゲット・接続の名前。
- **target_connection_name_alias** – ターゲット・接続のエイリアス名。
- **source_object_schema** – ソース・オブジェクトのスキーマ名。
- **source_object_name** – ソース・オブジェクトの名前。
- **new_source_connection_name** – 送信元の接続の新しい名前。
- **new_source_object_name** – ソース・オブジェクトの新しい名前。
- **new_source_object_schema** – ソース・オブジェクトの新しいスキーマ名。
- **target_object_schema** – ターゲット・オブジェクトのスキーマ名。
- **target_object_name** – ターゲット・オブジェクトの名前。
- **new_target_connection_name** – ターゲット・接続の新しい名前。
- **new_target_object_name** – ターゲット・オブジェクトの新しい名前。
- **new_target_object_schema** – ターゲット・オブジェクトの新しいスキーマ名。
- **max_concurrent_comparisons** – 同時比較の最大数。
- **description** – エージェントの説明。予約語または空白スペースを使用する場合は二重引用符を使用します。

例

- **例 1** – “schema_job” 内の **include all tables** 句を無効にするには、次のコマンドを実行します。

```
alter schema job schema_job
alter comparison sj_cmp
drop include all tables
go
```

結果は次のようになります。

```
Job "schema_job" was altered successfully.
```

- **例 2** – “schema_job” のジョブの説明を変更するには、次のコマンドを実行します。

```
alter schema job schema_job
set desc="my schema job"
go
```

結果は次のようになります。

```
Job "schema_job" was altered successfully.
```

create schema job

データベース・オブジェクトのスキーマを比較するために、新しいスキーマ・ジョブを作成します。

構文

```
create schema job sc_job_name
set max_concurrent_comparisons = 100
[and set desc [{to|=}] description]
  add comparison comparison_name
    With source source_connection_name source_alias
    target target_connection_name target_alias
      [and target target2_connection_name target2_connection_name_alias]
...
[include all tables]
  [map tables
    source_connection_name_alias.source_schema.source_object_name=
    target_connection_name_alias.target_schema.target_object_name[=
    target2_connection_name_alias.target2_schema.target2_object_name]...
  [and
source_connection_name_alias.source_object2_schema.source_object2_name=
target_connection_name_alias.target_schema.target_object2_name[=
target2_connection_name_alias.target2_schema.target2_object2_name]...
  ]
    [exclude tables
source_schema.source_object_name
[and source_schema.source_object2_name]...]
```

パラメータ

- **sc_job_name** – スキーマ比較ジョブの名前。
- **comparison_name** – スキーマ比較の名前。
- **max_concurrent_comparisons** – (オプション) 同時比較の最大数。デフォルト値は5です。
- **description** – (オプション) エージェントの説明。予約語または空白スペースを使用する場合は二重引用符を使用します。
- **source_connection_name** – 送信元の接続の名前。
- **source_alias** – 送信元の接続のエイリアス名。
- **target_connection_name** – ターゲット・接続の名前。
- **target_alias** – ターゲット・接続のエイリアス名。
- **source_schema** – ソース・オブジェクトのスキーマ名。
- **source_object_name** – ソース・オブジェクトの名前。
- **target_schema** – ターゲット・オブジェクトのスキーマ名。
- **target_object_name** – ターゲット・オブジェクトの名前。

例

- **例1** – スキーマ比較ジョブ“sc_job_test”を作成するには、次のコマンドを実行します。

```
create schema job sc_job_test
  set max_concurrent_comparisons = 100
  add comparison cmp1
    with source s1_con s1
      target t1_con t1
      and target t2_con t2
  include all tables
  map tables
    s1.tab_a = t1.tab_b
    and s1.tab_b = t2.tab_c
  exclude tables
    s1.tab_a2
```

結果は次のようになります。

```
Schema job "sc_job_test" was created successfully.
```

使用法

- **include all tables** 句は、送信元データベースにあるすべてのテーブルがスキーマ比較オブジェクトのリストにあることを指定し、テーブルのスキーマを比較するために送信元データベースとターゲット・データベースの間で自動の名前マッピングを使用します。**exclude table** 句は、スキーマ・ジョブ用に **include all**

tables を設定した後で、送信元データベースで除外するテーブルを指定します。

- **map tables** 句は、オブジェクト・マッピングを指定します。ソース・オブジェクトは、**map tables** と **exclude tables** に同時に存在することはできません。オブジェクト・マッピングは、**map tables** 句よりも優先されます。現在のリリースのオブジェクト・マッピングは、テーブルに限定されています。

drop schema job

既存のスキーマ・ジョブを削除します。

構文

```
drop schema job sc_job_name
```

パラメータ

- **sc_job_name** – 削除するスキーマ比較ジョブの名前。

例

- **例 1** – スキーマ・ジョブ “sc_job_test” を削除するには、次のコマンドを実行します。

```
drop schema job sc_job_test  
go
```

結果は次のようになります。

```
Schema job "sc_job_test" was dropped successfully.
```

show schema job

それぞれが 1 つまたは複数の比較で構成されたゼロ以上の既存のスキーマ比較ジョブを表示します。

構文

```
show schema job [schema_job_name [schema_job_comparison_name]]
```

パラメータ

- **schema_job_name** – 表示するスキーマ比較ジョブの名前。
- **schema_job_comparison_name** – 表示するスキーマ比較の名前。

例

- **例 1** – 既存のスキーマ・ジョブをステータスと共に表示します。

```
show schema job
go
```

結果は次のようになります。

Name	Active	Description
scjob_1	true	my_schemajob1
scjob_2	true	my_schemajob2
scjob_3	true	my_schemajob3

ジョブ管理のコマンド

ジョブの実行と履歴を作成するためのコマンドです。

abort job

実行中のジョブをアボートします。

構文

```
abort job job_name
```

パラメータ

- **job_name** – アボートするジョブの名前。

例

- **例 1** – “myjob_1” をアボートするには、次のコマンドを実行します。

```
abort job myjob_1
go
```

結果は次のようになります。

```
Job 'myjob_1' was aborted successfully.
```

disable job

指定されたジョブを無効にします。無効にされたジョブは、スケジュールから除外されます。また、無効にされたジョブを実行することはできません。

構文

```
disable job job_name
```

パラメータ

- **job_name** – 無効にするジョブの名前。

例

- **例 1** – “myjob_1” を無効にするには、次のコマンドを実行します。

```
disable job myjob_1  
go
```

結果は次のようになります。

```
Job 'myjob_1' was disabled successfully.
```

drop history

レポート・ファイルおよび調整スクリプト・ファイルを含む、既存のジョブの履歴を削除します。

構文

```
drop history job_name n
```

パラメータ

- **job_name** – 履歴を削除する対象となるジョブの名前。
- **n** – 削除する履歴のジョブ履歴シーケンス ID。

例

- **例 1** – ジョブ履歴 ID 1 の “myjob_1” の履歴を削除するには、次のコマンドを実行します。

```
drop history myjob_1 1  
go
```

enable job

指定されたジョブを有効にします。

構文

```
enable job job_name
```

パラメータ

- **job_name** – 有効にするジョブの名前。

例

- **例 1** – “myjob_1” を有効にするには、次のコマンドを実行します。

```
enable job myjob_1
go
```

結果は次のようになります。

```
Job `myjob_1` was enabled successfully.
```

monitor job

実行中のジョブ、または終了したばかりのジョブについてのランタイム・ステータス情報を表示します。

構文

```
monitor job [job_name [comparison_name]]
```

パラメータ

- **job_name** – 表示するジョブの名前。
- **comparison_name** – 表示する比較の名前。

例

- **例 1** – すべてのジョブのランタイム情報を表示するには、次のコマンドを実行します。

```
monitor job
go
```

結果は次のようになります。

NAME	TYPE	STATUS	SUBMIT TIME	FINISH TIME	ERROR
job2	ROW_COMPARE_JOB	RUNNING	2010-10-18	09:14:53.358	
job6	ROW_COMPARE_JOB	RUNNING	2010-10-18	09:14:57.093	

- **例 2** – ジョブ “j1” の比較情報を表示するには、次のコマンドを実行します。

```
monitor job j1 c1
go
```

結果は次のようになります。

COMPARISON	SUBMIT TIME	END TIME
c1	2011-03-01 16:49:34	2011-03-01 16:49:47

(0 rows affected)

RUN PHASE	TYPE SUMMARY	START TIME	END
-----------	--------------	------------	-----

```

TIME
-----
COUNT READ  M    O I R  PROGRESS ESTIMATE END ERROR
-----
1  COMPARE_ALL  S  c1/dbo.person  2011-03-01 16:49:34
2011-03-01 16:49:34
3072 3072 100%
T  c1/dbo.person2 2011-03-01 16:49:34
2011-03-01 16:49:34
1109 1109 1963 0 0 100%
T2 c1/dbo.person3 2011-03-01 16:49:34
2011-03-01 16:49:34
1109 1109 1963 0 0 100%
2  RECHECK_DIFFERENCES T 2011-03-01 16:49:45
2011-03-01 16:49:45
0 1963 0 0 100%
T2 2011-03-01 16:49:45
2011-03-01 16:49:45
0 1963 0 0 100%
3  VERIFY_DIFFERENCES S 2011-03-01 16:49:45
2011-03-01 16:49:46
1963 100%
T 2011-03-01 16:49:45
2011-03-01 16:49:46
0 1963 0 0 100%
T2 2011-03-01 16:49:45
2011-03-01 16:49:46
0 1963 0 0 100%
4  APPLY_RECONCILIATION T 2011-03-01 16:49:46
2011-03-01 16:49:47
1963 100%
T2 2011-03-01 16:49:46
2011-03-01 16:49:47
1963 100%

```

run job

指定されたジョブを起動します。

構文

```
run job job_name
```

run job では、既存のジョブ・スケジュールに関係なくすぐにジョブを実行します。

パラメータ

- **job_name** – 起動するジョブの名前。

例

- **例 1** – “myjob_1” を実行するには、次のコマンドを実行します。

```
run job myjob_1
go
```

結果は次のようになります。

```
Job 'myjob_1' was started successfully.
```

show history

レポート・ファイルおよび調整ファイル・パスを含む、1つのジョブの履歴を表示します。

構文

```
show history job_name[historyid]
```

パラメータ

- **job_name** – 履歴を表示する対象となるジョブの名前。
- **historyid** – ジョブ履歴のシーケンス ID。

例

- **例 1** – “job2” の履歴を表示するには、次のコマンドを実行します。

```
show history job2
go
```

結果は次のようになります。

HISTORY ID	SUBMIT TIME	FINISH TIME
12	2010-10-13 14:38:11.783	2010-10-13 14:38:19.41

- **例 2** – 履歴 ID 12 の “job2” の履歴を表示するには、次のコマンドを実行します。

```
show history job2 12
go
```

結果は次のようになります。

COMPARISON	RUN	PHASE	TYPE	SUMMARY
START TIME				
2011-02-22 16:09:54	1	COMPARE_ALL	S	MACHINEXP1:5000/test.dbo.mycash
2011-02-22 16:09:54	3		3	
2011-02-22 16:09:54			T	MACHINEXP1:5000/test.dbo.mycash2
2011-02-22 16:09:54	3		3	
2011-02-22 16:09:54	3		0 0 3	


```

2 RECHECK_DIFFERENCES T
2011-02-22 16:09:59
2011-02-22 16:09:59 3 0 0 3
3 VERIFY_DIFFERENCES S
2011-02-22 16:10:00
2011-02-22 16:10:00 3
T
2011-02-22 16:10:00
2011-02-22 16:10:00 3 0 0 3
4 CREATE_RECONCILIATION_SCRIPT T
2011-02-22 16:10:00
2011-02-22 16:10:00 0
5 APPLY_RECONCILIATION T
2011-02-22 16:10:00
2011-02-22 16:10:00 0
(0 rows affected)

FILE SERVER PATH
-----
Recon script C:\$Sybase\DA-15_5\server\instance\data\job2\
2011-02-22\14.38.11\c1_T_recon.sql
Text report C:\$Sybase\DA-15_5\server\instance\data\job2\
2011-02-22\14.38.11\report.txt

```

注意： `show history` を実行すると、出力が表示される前に、レポートがまだ作成されていない場合にはテキストおよび XML のレポートが生成されます。差異が多いほど、`show history` によるレポートの作成により長い時間がかかり、結果の表示の遅延の原因になります。

show reconcile

指定された履歴 ID を持つジョブの調整スクリプト・パスを表示します。

構文

```
show reconcile job_name historyid
```

`show reconcile` は、ロー比較ジョブにのみ適用できます。このコマンドは、スキーマ比較ジョブには使用できません。

パラメータ

- **job_name** – 調整スクリプトを表示する対象となるジョブの名前。
- **historyid** – 表示する調整スクリプトのジョブ履歴シーケンス ID。

例

- **例 1** – 履歴 ID が 29 の “job6” 調整スクリプトを表示します。

```
show reconcile job6 29
go
```

結果は次のようになります。

```

COMPARISON TARGET                                RUN  SUBMIT TIME
-----
FINISH TIME                                RECONCILE SCRIPT
-----
cmp6      myhost:5000/dadb.dbo.da1_10m 1      2010-10-15 14:04:04.573
2010-10-15 14:04:04.573  N/A
          2010-10-15 14:04:28.73  C:¥Sybase¥DA-15_5¥server¥instance¥data
¥job6¥2012-05-10¥14.04.03¥cmp6_0_recon.sql
(0 rows affected)

```

show report

指定された履歴 ID を持つジョブのレポート・ファイル・パスを生成および表示します。

構文

```
show report job_name historyid
```

パラメータ

- **job_name** – レポートを表示する対象となるジョブの名前。
- **historyid** – 表示するレポートのジョブ履歴シーケンス ID。

例

- **例 1** – 履歴 ID が 29 の “job6” レポート・ファイル・パスを表示するには、次のコマンドを実行します。

```
show report job6 29
go
```

結果は次のようになります。

```

REPORT TYPE  SERVER PATH
-----
TEXT         C:¥Sybase¥DA-15_5¥server¥instance¥data¥job6¥
2010-10-15¥14.04.03¥report.txt
XML          C:¥Sybase¥DA-15_5¥server¥instance¥data¥job6¥
2010-10-15¥14.04.03¥report.xml
(0 row affected)

```

truncate history

レポートおよび調整スクリプトを含む、既存のジョブ履歴のレコードまたは1つのジョブに属する履歴レコードを削除します。

構文

```
truncate history all |(job_name all|historyid)
```

パラメータ

- **all** - すべてのジョブ履歴レコードをトランケートします。
- **job_name** - 履歴をトランケートする対象となるジョブの名前。
- **historyid** - ジョブ履歴のシーケンス ID。

例

- **例 1** - すべてのジョブに対する履歴レコードを削除するには、次のコマンドを実行します。

```
truncate history all
go
```

- **例 2** - 特定のジョブ履歴 ID までの履歴レコードや、特定のジョブ履歴 ID を含む履歴レコードを削除するには、次のコマンドを実行します。

```
truncate history job_name 1
go
```

ジョブのインポートのコマンド

import job は、RSSD でテーブル複写定義とサブスクリプションに基づいてジョブの作成と設定を行う場合に使用します。

import job

Replication Server 内で事前に定義されたテーブル複写定義とサブスクリプションからジョブを作成します。

構文

```
import job rs_job_name
with connection rssid_connection_name
with map da_connection_name repdefs_ds repdef_db
[and da_connection_name repdefs_ds repdefs_db[...]]
[set MAX_CONCURRENT_COMPARISONS [{to|=}]
```

```

number_of_max_concurrent_comparisons]
[and set DESC [{to|=}] description]
[with comparison option
set ABORT_DIFF_MAX [{to|=}] number_of_differences
[and set ABORT_DIFF_ROW_COUNT [{to|=}] {true|false}
[and set AUTO_RECONCILE [{to|=}] {true|false}
[and set COMPARE_MODE [{to|=}] {row_compare | key_compare |
row_checksum | table_checksum | row_count}
[and set COMPRESS_DATA_TRANSFER [{to|=}] {true|false}
[and set CREATE_COL_LOG [{to|=}] {true|false}
[and set CREATE_RECON_SCRIPT [{to|=}] {true|false}
[and set DESC [{to|=}] description
[and set ENABLE_ROW_COUNT [{to|=}] {true|false}
[and set EXTERNAL_SORT [{to|=}] {true|false}
[and set HASH_TYPE [{to|=}] {database_hash | agent_hash}
[and set PRIORITY [{to|=}] {highest | high | normal | low}
[and set RETRY_DELAY_SEC [{to|=}] number_delay_second
[and set RETRY_DIFF [{to|=}] {never | wait_and_retry }
[and set RETRY_MAX [{to|=}] number_of_retries
]]]]]]]]]]]]]]]
[add schedule schedule_name
[set TYPE [{to|=}] {once | cron | every_day | every_week |
every_month}
[and set EVERY [{to|=}] n
[and set DATE [{to|=}] date_value
[and set TIME [{to|=}] time_value
[and set KEEP [{to|=}] keep_value
[and set KEEP_UNIT [{to|=}] {day | week | month | forever}
[and set CRON [{to|=}] cron_value
[and set DESC [{to|=}] description
]]]]]]]]]]]]]]]

```

パラメータ

- **rs_job_name** – 作成する Replication Server ジョブの名前。
- **rssd_connection_name** – 既存の RSSD コネクションの名前。
- **da_connection_name** – Data Assurance (DA) サーバ・コネクションの名前。
- **repdef_ds** – 複写定義に定義されているデータ・ソースの名前。
- **repdef_db** – 複写定義に定義されているデータベースの名前。
- **schedule_name** – 追加するスケジュールの名前。
- **max_concurrent_comparisons** – (オプション) このジョブを使用して同時実行できる比較の数。デフォルト値は 5 です。
- **description** – (オプション) エージェントの説明。予約語または空白スペースを使用する場合は二重引用符を使用します。

表 10 : 比較オプション

パラメータ	値
ABORT_DIFF_MAX	<p>差異のカウントが指定値を超えた場合、ロー比較をアボートする。</p> <p>有効な値：1 ~ 9223372036854775807</p> <p>デフォルト値：1000</p>
ABORT_DIFF_ROW_COUNT	<p>テーブルのロー・カウントが一致しない場合にロー比較を中止するかどうかを決定する。</p> <p>有効な値：true または false</p> <p>デフォルト：false</p>
AUTO_RECONCILE	<p>調整スクリプトを自動的に適用するかどうかを示す。</p> <p>有効な値：true または false</p> <p>デフォルト：false</p> <hr/> <p>注意： AUTO_RECONCILE を有効にするには、CREATE_COL_LOG を true に設定します。</p>
COMPARE_MODE	<p>ロー比較モードを指定する。</p> <ul style="list-style-type: none"> • row_compare - すべてのテーブル・ローを比較する。 • key_compare - プライマリ・キー・カラムを比較する。 • row_count - ロー・カウントを比較する。 <p>デフォルト：row_compare</p>
COMPRESS_DATA_TRANSFER	<p>ロー・データをエージェントとサーバの間で比較する。</p> <p>有効な値：true または false</p> <p>デフォルト：false</p>
CREATE_COL_LOG	<p>すべての消失ローの値、孤立したローの値、一貫性のないローの値(キーおよびカラム)をリストしたカラムの差異についてのログを生成する。以下を実行する場合は、カラム・ログを作成する。</p> <ul style="list-style-type: none"> • 調整スクリプトを生成する • 自動調整を実行する • 詳細なレポートを生成する <p>有効な値：true または false</p> <p>デフォルト：false</p>

パラメータ	値
CREATE_RECON_SCRIPT	調整スクリプトを生成する。このパラメータを使用するには、 CREATE_COL_LOG も true に設定する必要がある。 有効な値：true または false デフォルト：false
ENABLE_ROW_COUNT	送信元テーブルおよびターゲット・テーブルのローを比較前にカウントするかどうかを指定する。DA サーバは、比較の進行状況と終了時刻を見積もるためにロー・カウントを使用する。 注意 ： COMPARE_MODE が row_count の場合に、DA サーバはローをカウントします。 ENABLE_ROW_COUNT は、 COMPARE_MODE の値が row_count 以外の場合にのみ使用してください。 有効な値：true または false デフォルト：true
EXTERNAL_SORT	エージェント上のローをソートすることで、データベースで ORDER BY 句の処理の影響を低減する。 有効な値：true または false デフォルト：false
HASH_TYPE	比較のハッシュ・タイプを指定する。 <ul style="list-style-type: none"> • database_hash – データベースによって指定されたハッシュ関数を使用する。 • agent_hash – Replication Server Data Assurance オプションによって指定されたハッシュ関数を使用する。 デフォルト：database_hash
PRIORITY	比較キュー内のジョブ比較の順序を指定する。 正しい値は、次のとおり。 <ul style="list-style-type: none"> • highest • high • normal • low デフォルト：normal

パラメータ	値
RETRY_DIFF	リトライ・オプションを指定する。 <ul style="list-style-type: none"> • never – 再比較は実行されない。 • wait_and_retry – RETRY_MAX の設定および RETRY_DELAY_SEC の設定に基づいて再比較が実行される。 デフォルト：never
RETRY_DELAY_SEC	再比較と再比較の遅延の秒数を指定する。 有効な値：0 ~ 86400 デフォルト：10
RETRY_MAX	前の比較と異なる結果があるローの再比較を行う合計回数を指定する。 有効な値：0 ~ 100 デフォルト：3

表 11：カラム比較オプション

カラム・オプション	値
COMPARE_MODE	各カラムの比較方法を指定する。 <ul style="list-style-type: none"> • column_hash – カラムのハッシュ値を使用して比較を実行する。 • row_hash – 合計ハッシュ値を使用して、このオプションが指定されたすべてのカラムを一緒に比較する。 • literal – カラムのリテラル値を使用して比較を実行する。

表 12：スケジュール・オプション

パラメータ	値
date_value	スケジュールで日付を指定する。
time_value	スケジュールで時刻を指定する。
keep_value	スケジュールをアクティブに保つ期間の単位の数を指定する。
cron_value	スケジュールで cron オプションの値を指定する。

例

- **例 1** – “myrsjob_1” という名前の新しいジョブを作成するには、次のコマンドを実行します。

```
import job myrsjob_1
with rssid connection MyRSSDConn
with map MyConnPDB1 repdef_ds repdef_db
with map MyConnRDB1 repdef_ds2 repdef_db2
set max_concurrent_comparisons = 3
with comparison options
set COMPARE MODE= row_compare
and set ABORT_DIFF_MAX = 20
and set ABORT_DIFF_ROW_COUNT = true
and set RETRY_DIFF = wait_and_retry
and set RETRY_MAX= 2
and set RETRY_DELAY_SEC = 10
and set HASH_TYPE = database_hash
with schedule myschedule_1
set type=every_day
and set every=2
and set time=10:00
and set keep=1
and set keep_unit=month
go
```

結果は次のようになります。

```
Job "myrsjob_1" was created successfully.
```

Data Assurance System Database (DASD) コマンド

DASD を管理するためのコマンドです。

create backup

現在の Data Assurance System Database (DASD) データベースのバックアップを作成します。バックアップ・ファイルは `da¥server¥instance¥dasd¥backup ¥unique_backup_id` に保存されます。

構文

```
create backup
```

例

- **例 1** – DASD のバックアップを作成します。

```
create backup
go
```


drop backup

`backup_index` で指定した特定のバックアップを削除します。

構文

```
drop backup backup_index
```

パラメータ

- **backup_index** – バックアップのインデックス・エントリを指定します。

例

- **例 1** – インデックス・エントリ 3 のバックアップを削除します。

```
drop backup 3  
go
```

restore backup

Data Assurance System Database (DASD) データベースをバックアップ・コピーからリストアします。

構文

```
restore backup backup_index
```

例

- **例 1** – DASD をリストアします。

```
restore backup 3  
go
```

使用法

- **restore backup** に成功するとサーバは自動的に停止するので、手動で再起動する必要があります。

show backup

Data Assurance System Database (DASD) がバックアップされている場所を表示します。

構文

```
show backup
```

例

- **例 1** – DASD バックアップのパスを表示します。

```
show backup
go
```

結果は次のようになります。

INDEX	DATE	PATH
1	2011-1-12 13:29:58	C:\¥Sybase¥DA-15_5¥server¥myserver¥dasd¥backup¥1297407

(0 row affected)

truncate backup

既存のバックアップまたは特定のバックアップをすべて削除します。

構文

```
truncate backup all | backup_index
```

パラメータ

- **all** – すべてのバックアップを切り詰めます。
- **backup_index** – バックアップのインデックス・エントリです。

例

- **例 1** – すべてのバックアップを削除します。

```
truncate backup all
go
```

- **例 2** – バックアップのインデックス・エントリ 3 を削除します。

```
truncate backup 3
go
```

注意： この例では、*backup_index* で指定されるものも含め、**truncate backup** によって以前のバックアップ (1 と 2) がすべて削除されます。

その他のコマンド

DA サーバの設定とトラブルシューティングのコマンドです。

config

DA サーバの設定パラメータを設定して表示します。

構文

```
config [parameter_name [parameter_value]]
```

パラメータ

- **parameter_name** – 設定する DA サーバのパラメータ。
- **parameter_value** – DA サーバのパラメータの値。

グローバル設定パラメータの現在の値はすべて、DASD (Data Assurance System Database) に格納されています。

表 13 : グローバル設定パラメータ

parameter_name	parameter_value
agent_client_ctx_timeout_secs	DA サーバと DA エージェント間の接続タイムアウトを秒単位で指定します。 デフォルト値は 5 最小：1 最大：2147483647 このパラメータは、DA サーバを再起動しなくても有効になります。
agent_access_timeout_mins	DA サーバと DA エージェントの間にアクティビティがなくても接続が開いたままになる時間を指定します。 デフォルト値は 60 最小：1 最大：2147483647 このパラメータは、DA サーバを再起動しなくても有効になります。

parameter_name	parameter_value
agent_max_queue	<p>エージェントが出力キューにバッファするローの最大数を指定します。DA サーバはキューからローを読み取ります。キューが満杯になると、エージェントはデータベース・テーブルからのローの読み取りを一時的に停止します。</p> <p>デフォルト値は 1000</p> <p>最小：1</p> <p>最大：2147483647</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
agent_max_request_queue	<p>サーバの比較と調整のリトライ要求の最大キュー・サイズを指定します。</p> <p>デフォルト値は 100</p> <p>最小：1</p> <p>最大：2147483647</p> <p>このパラメータを有効にするには、DA サーバの再起動が必要になります。</p>
auto_recon_stmt_batch_size	<p>1回の呼び出しで DA サーバが DA エージェントに送信する SQL 文の最大数を指定します。</p> <p>デフォルト値は 100</p> <p>最小：1</p> <p>最大：10000</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>

parameter_name	parameter_value
clt_password_encryption_reqd	<p>サーバが必要とするパスワード暗号化のレベルを決定します。 デフォルト値は 0</p> <p>有効な値</p> <ul style="list-style-type: none"> 0 - クライアントがネットワーク・ログイン・パスワードに使用する暗号化アルゴリズムを選択できます。暗号化しないオプションもあります。 1 - クライアントがネットワーク・ログイン・パスワードの暗号化に RSA 暗号化アルゴリズムのみを使用するように制限します。これにより、パスワードの強力な RSA 暗号化が実現します。RSA 暗号化を使わずに接続しようとするクライアントは、接続に失敗します。 <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
comparer_max_concurrent_threads	<p>同時比較に使用される比較スレッドの最大数を指定します。 デフォルト値は 5</p> <p>最小：1</p> <p>最大：2147483647</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
comparer_retry_delay_threshold_secs	<p>DA サーバの比較がリトライ前に比較スレッドを維持できるしきい値を秒単位で指定します。</p> <p>値が retry_delay_sec より大きい場合、DA サーバの比較はリトライの待機中、現在の比較スレッドを保持します。このため、キュー内にある別の比較スレッドの開始が遅れる可能性があります。</p> <p>値が retry_delay_sec 以下の場合、DA サーバの比較は現在の比較スレッドを解放し、キュー内の次の比較を処理し始めます。</p> <p>デフォルト値は 20</p> <p>最小：1</p> <p>最大：2147483647</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>

parameter_name	parameter_value
comparer_recently_finished_ttl_secs	<p>ジョブ情報がモニタ・ジョブのビューに残る最大時間を秒単位で指定します。</p> <p>デフォルト値は 300</p> <p>最小：1</p> <p>最大：2147483647</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
comparer_retry_max_keys_per_clause	<p>WHERE 句の単一キーの最大数。</p> <p>デフォルト値は 10</p> <p>最小：1</p> <p>最大：100</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
comparer_retry_min_keys_in_range	<p>キーの選択条件を、個別ではなく範囲として計算するとき使用するキーの最小数を指定します。</p> <p>デフォルト値は 5</p> <p>最小：2</p> <p>最大：100</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>

parameter_name	parameter_value
comparer_retry_min_fill_percent	<p>「単一」キーを範囲に結合するときに必要な最小フィル・パーセンテージを指定します。隣接または近くのロー・キーの結果セットを選択するとき、通常は、文で各キーを別々に指定するよりも範囲でキーを選択する方が時間短縮になります。</p> <p>たとえば、1 から 100 までのローを 1 つ置きに選択するには、以下を使用します。</p> <pre>"select ... where id in(1,3,5,7..97,99)"</pre> <p>または、次のように範囲内のローをすべてフェッチすることもできます。</p> <pre>"select ... where id >=1 and id <= 100"</pre> <p>通常、範囲内のすべてのローをフェッチすると、1 つまたは複数の in (...) 文を実行するよりも時間を短縮できます。上の例では、選択したローの半分だけが必要であるため、フィル・パーセンテージは 50 になります。DA サーバは余分なローをすべてスキップします。</p> <p>デフォルト値は 10</p> <p>最小：1</p> <p>最大：100</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
comparer_retry_min_fill_percent_literal	<p>リテラル比較のために「単一」キーを範囲に結合するときに必要な最小フィル・パーセンテージを指定します。</p> <p>デフォルト値は 90</p> <p>最小：1</p> <p>最大：100</p> <hr/> <p>注意： 余分なリテラル・ロー・データを伝送するコストは、すぐに範囲選択のパフォーマンス上の利得を上回るため、通常は、comparer_retry_min_fill_percent_literal を comparer_retry_min_fill_percent よりも大きい値に設定してください。</p> <hr/> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>

parameter_name	parameter_value
db_connection_retry_times	<p>コネクション・マネージャが最初のデータベース接続に失敗した場合に再試行する最大回数を指定します。</p> <p>デフォルト値は 2</p> <p>最小：1</p> <p>最大：2147483647</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
db_connection_retry_interval	<p>コネクション・マネージャがデータベース接続を連続試行するときの最大待ち時間を指定します。</p> <p>デフォルト値は 3</p> <p>最小：1</p> <p>最大：2147483647</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
default_column_compare_mode	<p>カラムのデフォルトの比較モードを指定します。</p> <p>デフォルト値は column_hash</p> <p>有効な値 literal、column_hash、row_hash</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
db_hash_ase_algorithm	<p>Adaptive Server データベースのハッシュ・アルゴリズムを指定します。</p> <p>デフォルト値は MD5</p> <p>有効な値：MD5 または SHA</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
db_hash_ase_ignore_null	<p>Adaptive Server データベースのマルチハッシュの値を計算するときに、Adaptive Server のハッシュバイト制限の問題を無視するかどうかを指定します。</p> <p>デフォルト値は true</p> <p>有効な値 true または false</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>

parameter_name	parameter_value
db_hash_ase_using_option	<p>Adaptive Server データベースのハッシュ・アルゴリズムのバイト順序オプションを指定します。</p> <p>デフォルト値は UNICODE_LSB</p> <p>有効な値 LSB、MSB、UNICODE、UNICODE_LSB、UNICODE_MSB</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
enable_report_generator	<p>ジョブ・レポートを生成するかどうかを指定します。</p> <p>enable_report_generator を false に設定すると、ジョブ履歴項目が表示されてレポートがまだ生成されていないときに、レポート・ジェネレータが XML およびテキスト・レポートを作成できないようにします。これは、カラム・ログが大容量で、レポートの生成に時間がかかる場合に便利です。</p> <p>デフォルト値は true</p> <p>有効な値 true または false</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
external_sort_max_thread	<p>外部ソートに使用される最大スレッド数を指定します。</p> <p>デフォルト値は 5</p> <p>最小：3</p> <p>最大：10</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
external_sort_max_size	<p>メモリでソートできるローの最大数を指定します。</p> <p>デフォルト値は 1000000</p> <p>最小：2</p> <p>最大：2147483647</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>

parameter_name	parameter_value
external_sort_max_file	<p>外部ソートに使用される最大ファイル数を指定します。</p> <p>デフォルト値は 60</p> <p>最小：10</p> <p>最大：2147483647</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
external_sort_compress_file	<p>データ・ファイルを圧縮するかどうかを指定します。</p> <p>デフォルト値は false</p> <p>有効な値 true または false</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
external_sort_activate_size	<p>外部ソートをアクティブにするためにデータベース・テーブルに必要なローの最小数を指定します。</p> <p>デフォルト値は 1000000</p> <p>最小：2</p> <p>最大：2147483647</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
file_output_encoding	<p>調整スクリプトとレポート・ファイルすべてのファイル出力コードを指定します。</p> <p>有効な値 DA サーバの JRE (Java Runtime Environment) でサポートされている文字セット・コード</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>
lob_fetch_size	<p>読み取って比較するラージ・オブジェクト (LOB) の最大バイト数を指定します。</p> <p>デフォルト値は 1024</p> <p>最小：1</p> <p>最大：1048576</p> <p>このパラメータは、DA サーバを再起動しなくても有効になります。</p>

parameter_name	parameter_value
recon_tran_max_stmts	調整トランザクションの文の最大数を指定します。必要な文の数がこの数よりも多い場合は、複数のトランザクションが必要になります。ゼロの値はトランザクションの文の数に制限がないという意味です。 デフォルト値は 0 最小：0 最大：2147483647 このパラメータは、DA サーバを再起動しなくても有効になります。
text_report_max_column_width	テキスト・レポートの最大カラム幅を指定します。 デフォルト値は 30 最小：10 最大：80 このパラメータは、DA サーバを再起動しなくても有効になります。
text_report_max_line_length	テキスト・レポートの行の最大長を指定します。 デフォルト値は 200 最小：100 最大：1000 このパラメータは、DA サーバを再起動しなくても有効になります。

例

- **例 1** – すべての設定パラメータを表示します。

```
config
go
```

結果は次のようになります。

NAME	VALUE	PENDING	REQUIRE RESTART
agent_access_timeout_mins	60		false
agent_client_ctx_timeout	5		false
agent_max_queue	1000		false
agent_max_request_queue	100		true
auto_recon_stmt_batch_size	100		false
clt_password_encryption_reqd	1		false
comparer_max_concurrent_threads	5		false

Data Assurance サーバのコマンド・リファレンス

comparer_recently_finished_ttl_secs	300	false
comparer_retry_delay_threshold_secs	20	false
comparer_retry_max_keys_per_clause	10	false
comparer_retry_min_fill_percent	10	false
comparer_retry_min_fill_percent_literal	90	false
comparer_retry_min_keys_in_range	5	false
db_connection_retry_interval	3	false
db_connection_retry_times	2	false
db_hash_ase_algorithm	MD5	false
db_hash_ase_ignore_null	false	false
db_hash_ase_using_option	UNICODE_LSB	false
default_column_compare_mode	ROW_HASH	false
enable_report_generator	true	false
external_sort_activate_size	1000000	false
external_sort_compress_file	false	false
external_sort_max_file	64	false
external_sort_max_size	100000	false
external_sort_max_thread	5	false
file_output_encoding	cp936	false
lob_fetch_size	1024	false
recon_tran_max_stmts	0	false
text_report_max_column_width	30	false
text_report_max_line_length	200	false

- **例 2** – `db_hash_ase_algorithm` のデフォルト値 (MD5) を SHA に変更します。

```
config db_hash_ase_algorithm SHA
go
```

結果は次のようになります。

NAME	VALUE	PENDING
db_hash_ase_algorithm	SHA	
(1 rows affected)		
DEFAULT	VALID	EXPLANATION
MD5	MD5,SHA	The database hash algorithm for ASE

- **例 3** – 必要な暗号化レベルを “encryption level 1” に変更します。この設定パラメータをゼロ以外の値に設定すると、警告メッセージが表示されます。

```
config clt_password_encryption_reqd 1
go
```

結果は次のようになります。

```
[#90] Warning: you have set the password encryption level to 1;
please ensure your client tool supports this level of encryption,
otherwise you will not be able to login again without upgrading
your client tool.
```

```
(1 row affected)
```

password

DA サーバのログイン・パスワードを変更します。

password は結果セットを返しません。現在のパスワードが間違っていたり、新しいパスワードが無効であったりすると、エラー・メッセージが表示されます。

構文

```
password current_password new_password
```

パラメータ

- **current_password** – 管理ユーザ・ログイン名の既存のパスワード。
- **new_password** – 管理ユーザ・ログイン名の新しいパスワード。デフォルトのパスワード長は最小が 6 で最大が 30 です。パスワード長は *instance.cfg* で設定できます。入力値の有効な文字は、a-z、A-Z、0-9、-、_ です。

例

- **例 1** – da_admin パスワードを “sybase” から “onesybase” に変更します。

```
password sybase onesybase
go
```

参照：

- パスワード・ポリシー (129 ページ)

role

LDAP ユーザを DA 管理者の役割にマップします。

構文

```
role [rolename [add|drop user username]]
```

パラメータ

- **rolename** – 大文字と小文字が区別される役割の名前。
- **username** – 大文字と小文字が区別されるユーザ名。

例

- **例 1** – すべての役割とユーザを表示するには、次のように入力します。

```
role
go
```

結果は次のようになります。

ROLE	USER	LOCAL USER
DA_Admin	da_admin	true
DA_Admin	srjones	false

- **例2**–DA サーバ管理者の役割を持つすべてのユーザを表示するには、次のように入力します。

```
role DA_Admin
go
```

結果は次のようになります。

ROLE	USER	LOCAL USER
DA_Admin	da_admin	true
DA_Admin	srjones	false

- **例3**–"tabraham" を DA サーバ管理者の役割に追加するには、次のように入力します。

```
role DA_Admin add user tabraham
go
```

- **例4**–DA サーバ管理者の役割から "tabraham" を削除するには、次のように入力します。

```
role DA_Admin drop user tabraham
go
```

show jvm

Java Virtual Machine (JVM) の重要な詳細をいくつか表示します。

このコマンドには引数は必要ありません。

構文

```
show jvm
```

例

- **例1**–JVM の詳細を表示します。

```
show jvm
go
```

結果は次のようになります。

JVM NAME	JVM INFO	JVM VERSION	JVM VENDOR

```

-----
Java HotSpot(TM) Server VM mixed mode Sun Microsystems Inc.
  Java 1.6.0_24, VM 19.1-b02
(0 rows affected)

JVM TOTAL MEM JVM FREE MEM JVM MAX MEM
-----
31.8 MB      27.1 MB      455.1 MB

```

show system

重要なシステム・プロパティをいくつか表示します。

このコマンドには引数は必要ありません。

構文

```
show system
```

例

- **例 1** – システムの詳細を表示します。

```
show system
go
```

結果は次のようになります。

NAME	HOST	LOCALE	TIME ZONE	DATE	TIME
myserver	10.65.0.111	en_GB	Greenwich Mean Time	2011-06-10	16:05:44
OS NAME	OS VERSION	OS ARCH	OS LOAD AVG		
Windows XP	5.1	x86	14.897%		

sslconfig

すべての SSL (Secure Sockets Layer) 設定パラメータを設定および表示します。

構文

```
sslconfig [parameter_name [parameter_value]]
```

パラメータ

- **parameter_name** – 設定する SSL パラメータ。
- **parameter_value** – SSL パラメータの値。

parameter_name	parameter_value
dts_client_ssl_required	<p>カンマまたはセミコロンで区切られた DA エージェントのホスト名のリスト。</p> <p>DA サーバが DA エージェントの Data Transfer Socket に接続すると、そのサーバはこのリストを使用して DA エージェントのホスト名をチェックします。その DA エージェントのホスト名が検出されると、DA サーバは SSL ソケットを使用して接続を行います。</p> <p>このパラメータを有効にするには、DA サーバを再起動する必要があります。</p>
keypair_passwd	<p>キーストア内にあるパブリック・キー/プライベート・キーのペアへのアクセスを許可するためのパスワード。</p> <p>この値は、ディスクでは暗号化されています。この値がコマンド・ライン・ツール (CLT) に表示される場合は、***** プレースホルダによって空白でない値が設定されていることを示します。</p> <p>このパラメータを有効にするには、DA サーバを再起動する必要があります。</p>
keystore	<p>使用するパブリック・キー/プライベート・キーのペアを格納するキーストア (フラット・ファイル) への絶対パス。</p> <p>このパラメータを有効にするには、DA サーバを再起動する必要があります。</p>
keystore_passwd	<p>キーストアへのアクセスを許可するためのパスワード。</p> <p>この値は、ディスクでは暗号化されています。この値が CLT に表示される場合は、***** プレースホルダによって空白でない値が設定されていることを示します。</p> <p>このパラメータを有効にするには、DA サーバを再起動する必要があります。</p>
rmi_client_ssl_required	<p>カンマまたはセミコロンで区切られた DA エージェントのホスト名のリスト。</p> <p>DA サーバが DA エージェントへの RMI コネクションを作成すると、そのサーバはこのリストを使用して DA エージェントのホスト名をチェックします。その DA エージェントのホスト名が検出されると、DA サーバは SSL ソケットを使用して接続を行います。</p> <p>このパラメータを有効にするには、DA サーバを再起動する必要があります。</p>

parameter_name	parameter_value
rmi_server_ssl_enabled	すべての RMI クライアントへの接続に SSL を使用するかどうかを指定する。この値が true に設定されている場合、DA サーバによってすべての RMI クライアントが SSL を使用した接続を行うように要求されます。 デフォルト値は false このパラメータを有効にするには、DA サーバの再起動が必要になります。
truststore	信頼された証明書を格納するトラストストア (フラット・ファイル) への絶対パス。 このパラメータを有効にするには、DA サーバを再起動する必要があります。
truststore_passwd	トラストストアへのアクセスを許可するためのパスワード。 この値は、ディスクでは暗号化されています。この値が CLT に表示される場合は、***** プレースホルダによって空白でない値が設定されていることを示します。 このパラメータを有効にするには、DA サーバを再起動する必要があります。

例

- **例 1** – すべての SSL 設定パラメータを次のように表示します。

```
sslconfig
go
```

結果は次のようになります。

```
NAME                               VALUE                               PENDING  REQUIRE RESTART
-----                               -----                               -
dts_client_ssl_required            venus,pluto                          true
keypair_passwd                     *****                              true
keystore                           C:/SSL/mars_trust.ks                 true
keystore_passwd                    *****                              true
rmi_client_ssl_required            venus,pluto                          true
rmi_server_ssl_enabled             true                                  true
truststore                         C:/SSL/mars_trust.ks                 true
truststore_passwd                  *****                              true
(0 rows affected)
```

- **例 2** – SSL パラメータの現在の値とその記述を次のように表示します。

Data Assurance サーバのコマンド・リファレンス

```
sslconfig keystore
go
```

結果は次のようになります。

NAME	VALUE	PENDING	REQUIRE RESTART
keystore	C:/SSL/mars_trust.ks		true

(0 rows affected)

DEFAULT	MINIMUM	MAXIMUM	EXPLANATION
			The absolute path to a keystore (server-side configuration).

(0 rows affected)

- **例 3** – SSL パラメータのデフォルト値を次のように変更します。

```
sslconfig rmi_server_ssl_enabled false
go
```

trace

システム・トレースのレベルを設定し、トレース・フラグの設定を返します。

構文

```
trace [flag| all [level]]
```

パラメータ

- **flag** – トレース・フラグの名前。サーバ・コンテナで使用可能なフラグ名は、agent、audit、clt、compare、container、dasd、license、server、sql、std です。
- **all** – システム内のすべてのトレース・フラグを指定します。
- **level** – トレース・レベルを指定します。使用可能なレベルは、off、severe、warning、info、config、fine、finer、finest、all です。

例

- **例 1** – トレース・レベルを表示します。

```
trace
go
```

結果は次のようになります。

TRACE	LEVEL
agent	INFO
audit	ALL

```

clt                INFO
compare            INFO
container          INFO
dasd                SEVERE
license            INFO
server             INFO
sql                INFO
std                ALL

```

version

Replication Server Data Assurance オプションの現在のバージョンを表示します。

構文

```
version
```

例

- **例 1** – バージョンを表示します。

```
version
go
```

結果は次のようになります。

```
VERSION
```

```
-----
Replication Server Data Assurance Option/15.7.1/DA Server/P/generic/
generic/da157x/121/VM: Sun Microsystems Inc. 1.6.0_24/OPT/Tue 24 Apr
2012 09:24:31 GMT
```

Data Assurance サーバの予約語

予約語は、DA サーバでコマンドの一部として使用されると特別な意味を持ちます。DA サーバでは、単語を二重引用符で囲んで設定しない限り、コマンド構文の一部として使用できません。

表 14 : DA サーバの予約語

	予約語
A	abort、add、agent、all、alter、
B	backup
C	compareset、comparison、config、connection、create
D	dasd、depend、disable、drop、dts

	予約語
E	enable、exclude
F	force
H	history
I	immediately、import、include
J	job、jvm
L	license
M	map、monitor
N	node
O	option
P	password、properties
R	reconcile、report、restore、role、rsjob、run
S	schedule、schema、set、show、shutdown、source、sslconfig、system
T	tables、target、task、test、to、trace、truncate
U	user
V	version
W	where、with

リモート Data Assurance エージェントのコマンド・リファレンス

リモート DA エージェントのコマンドは、**isql** または Sybase Control Center Data Assurance プラグインを使用して実行できます。

注意： DA エージェントのすべてのコマンドを実行するには、“da_admin” パラメーションが必要です。

config

DA エージェントの設定パラメータを設定および表示します。

構文

```
config [parameter_name [parameter_value]]
```

パラメータ

- **parameter_name** – 設定する DA エージェントのパラメータ。
- **parameter_value** – DA エージェントのパラメータの値。

現在の設定パラメータの値は、設定ファイルに格納されています。

parameter_name	parameter_value
clt_password_encryption_reqd	<p>エージェントが必要とするパスワード暗号化のレベルを決定する。</p> <p>デフォルト値は 0</p> <p>有効な値</p> <ul style="list-style-type: none"> • 0 – クライアントがネットワーク・ログイン・パスワードに使用する暗号化アルゴリズムを選択できる。暗号化しないオプションもある。 • 1 – クライアントがネットワーク・ログイン・パスワードの暗号化に RSA 暗号化アルゴリズムのみを使用するように制限する。これにより、パスワードの強力な RSA 暗号化が実現する。RSA 暗号化を使わずに接続しようとするクライアントは、接続に失敗する。 <p>このパラメータは、DA エージェントを再起動しなくても有効になる。</p>

例

- **例 1** – 必要な暗号化レベルを “encryption level 1” に変更します。この設定パラメータをゼロ以外の値に設定すると、警告メッセージが表示されます。

```
config clt_password_encryption_reqd 1
go
```

結果は次のようになります。

```
[#90] Warning: you have set the password encryption level to 1;
please ensure your client tool supports this level of encryption,
otherwise you will not be able to login again without upgrading
your client tool.
```

```
(1 row affected)
```

password

DA エージェントのログイン・パスワードを変更します。

password は結果セットを返しません。現在のパスワードが間違っていたり、新しいパスワードが無効であったりすると、エラー・メッセージが表示されます。

構文

```
password current_password new_password
```

パラメータ

- **current_password** – 管理ユーザ・ログイン名の既存のパスワード。
- **new_password** – 管理ユーザ・ログイン名の新しいパスワード。デフォルトのパスワード長は最小が 6 で最大が 30 です。パスワード長は *instance.cfg* で設定できます。入力値の有効な文字は、a-z、A-Z、0-9、-、_ です。

例

- **例 1** – da_admin パスワードを “sybase” から “onesybase” に変更します。

```
password sybase onesybase
go
```

使用法

DA エージェントのパスワードを変更する場合は、その DA エージェントに接続している DA サーバで設定されたエージェント・パスワードも変更する必要があります。変更しないと、DA サーバが DA エージェントを認証できなくなります。

参照：

- alter agent (35 ページ)
- パスワード・ポリシー (129 ページ)

role

LDAP ユーザを DA 管理者の役割にマップします。

構文

```
role [rolename [add|drop user username]]
```

パラメータ

- **rolename** – 大文字と小文字が区別される役割の名前。
- **username** – 大文字と小文字が区別されるユーザ名。

例

- **例 1** – すべての役割とユーザを表示するには、次のように入力します。

```
role
go
```

結果は次のようになります。

ROLE	USER	LOCAL USER
-----	-----	-----
DA_Admin	da_admin	true
DA_Admin	srjones	false

- **例 2** – DA 管理者の役割を持つすべてのユーザを表示するには、次のように入力します。

```
role DA_Admin
go
```

結果は次のようになります。

ROLE	USER	LOCAL USER
-----	-----	-----
DA_Admin	da_admin	true
DA_Admin	srjones	false

- **例 3** – "tabraham" を DA 管理者の役割に追加するには、次のように入力します。

```
role DA_Admin add user tabraham
go
```

- **例 4** – DA 管理者の役割から "tabraham" を削除するには、次のように入力します。

```
role DA_Admin drop user tabraham
go
```

show connection

リモート・エージェントのデータベース・コネクションを表示します。

構文

```
show connection
```

例

- **例 1** – リモート DA エージェントのコネクションを表示します。

```
show connection
go
```

結果は次のようになります。

SERVER	NAME	TYPE	CONNECTED
myserver:4500@soka.sybase.com	conn1_23mw	ASE	3
myserver:4500@soka.sybase.com	soka2_ra	ASE	2
myserver:4500@etlaix61.sybase.com	conn1_h33	ASE	2

show dts

リモート・エージェントで実行している DTS (Data Transfer Stream) の情報を表示します。

構文

```
show dts
```

例

- **例 1** – リモート・エージェントの DTS 情報をすべて表示します。

```
show dts
go
```

結果は次のようになります。

```
TASK ID ESTIMATE COUNT FETCHING QUEUE TAKEN ESTIMATE SECONDS LEFT
```



```
-----
3          1000                                0    0
-----
```

show jvm

Java Virtual Machine (JVM) の重要な詳細をいくつか表示します。

このコマンドには引数は必要ありません。

構文

```
show jvm
```

例

- **例 1** – JVM の詳細を表示します。

```
show jvm
go
```

結果は次のようになります。

```
JVM NAME                JVM INFO    JVM VENDOR
      JVM VERSION
-----
Java HotSpot(TM) Server VM mixed mode Sun Microsystems Inc.
      Java 1.6.0_24, VM 19.1-b02
(0 rows affected)

JVM TOTAL MEM JVM FREE MEM JVM MAX MEM
-----
31.8 MB        27.1 MB    455.1 MB
```

show system

重要なシステム・プロパティをいくつか表示します。

このコマンドには引数は必要ありません。

構文

```
show system
```

例

- **例 1** – システムの詳細を表示します。

リモート Data Assurance エージェントのコマンド・リファレンス

```
show system
go
```

結果は次のようになります。

NAME	HOST	LOCALE	TIME_ZONE	DATE	TIME
myagent	10.65.0.111	en_GB	Greenwich Mean Time	2011-06-10	16:05:44
OS NAME	OS VERSION	OS ARCH	OS LOAD AVG		
Windows XP	5.1	x86	14.897%		

show task

リモート・エージェントのタスク情報を表示します。

構文

```
show task
```

例

- **例 1** – リモート DA エージェントのすべてのタスクを表示します。

```
show task
go
```

結果は次のようになります。

SERVER	TASK ID	CONNECTION	OBJECT	STAGE	OBJ
PROCESSED	PREDICATE	SQL	PROCESSED		
myserver:4500@soka.sybase.com	35	conn1_1t8p	dbo.da1_10m	0	
myserver:4500@soka.sybase.com	37	conn1_23mw	dbo.da1_10m	0	

TASK ID	ESTIMATE	COUNT	QUEUE	TAKEN	ESTIMATE	SECONDS	LEFT
35	10000	0					
37	10000	0					

sslconfig

すべての SSL (Secure Sockets Layer) 設定パラメータを設定および表示します。

構文

```
sslconfig [parameter_name [parameter_value]]
```

パラメータ

- **parameter_name** – 設定する SSL パラメータ。
- **parameter_value** – SSL パラメータの値。

parameter_name	parameter_value
dts_server_ssl_enabled	<p>DA エージェントが SSL を使用した Data Transfer Socket を作成するかどうかを指定する。</p> <p>デフォルト値は false</p> <p>このパラメータを有効にするには、DA エージェントを再起動する必要があります。</p>
keypair_passwd	<p>キーストア内にあるパブリック・キー/プライベート・キーのペアへのアクセスを許可するためのパスワード。</p> <p>この値は、ディスクでは暗号化されています。この値がコマンド・ライン・ツール (CLT) に表示される場合は、***** プレースホルダによって空白でない値が設定されていることを示します。</p> <p>このパラメータを有効にするには、DA エージェントを再起動する必要があります。</p>
keystore	<p>使用するパブリック・キー/プライベート・キーのペアを格納するキーストア (フラット・ファイル) への絶対パス。</p> <p>このパラメータを有効にするには、DA エージェントを再起動する必要があります。</p>
keystore_passwd	<p>キーストアへのアクセスを許可するためのパスワード。</p> <p>この値は、ディスクでは暗号化されています。この値が CLT に表示される場合は、***** プレースホルダによって空白でない値が設定されていることを示します。</p> <p>このパラメータを有効にするには、DA エージェントを再起動する必要があります。</p>
rmi_server_ssl_enabled	<p>DA サーバがすべての RMI クライアントに対して SSL を使用した接続を要求するかどうかを指定する。</p> <p>デフォルト値は false</p> <p>このパラメータを有効にするには、DA エージェントを再起動する必要があります。</p>

parameter_name	parameter_value
truststore	信頼された証明書を格納するトラストストア (フラット・ファイル) への絶対パス。 このパラメータを有効にするには、DA エージェントを再起動する必要があります。
truststore_passwd	トラストストアへのアクセスを許可するためのパスワード。 この値は、ディスクでは暗号化されています。この値が CLT に表示される場合は、***** プレースホルダによって空白でない値が設定されていることを示します。 このパラメータを有効にするには、DA エージェントを再起動する必要があります。

例

- 例 1 – すべての SSL 設定パラメータを次のように表示します。

```
sslconfig
go
```

結果は次のようになります。

NAME	VALUE	PENDING	REQUIRE	RESTART
dts_server_ssl_enabled	false			true
keypair_passwd				true
keystore				true
keystore_passwd				true
rmi_server_ssl_enabled	false			true
truststore				true
truststore_passwd				true

(0 rows affected)

- 例 2 – SSL パラメータの現在の値とその記述を次のように表示します。

```
sslconfig dts_server_ssl_enabled
go
```

結果は次のようになります。

NAME	VALUE	PENDING	REQUIRE	RESTART
dts_server_ssl_enabled	false			true

(0 rows affected)

DEFAULT MINIMUM MAXIMUM EXPLANATION

```
-----
Whether this DA agent should use
SSL when streaming data via its DTS.
```

(0 rows affected)

- **例 3** – SSL パラメータのデフォルト値を次のように変更します。

```
sslconfig rmi_server_ssl_enabled true
go
```

trace

リモート DA エージェントのシステム・トレースのレベルを設定し、トレース・フラグの設定を返します。

構文

```
trace [flag| all [level]]
```

パラメータ

- **flag** – トレース・フラグの名前。エージェント・コンテナで使用可能なフラグ名は、agent、audit、clt、container、sql、std です。
- **all** – システム内のすべてのトレース・フラグを指定します。
- **level** – トレース・レベルを指定します。使用可能なレベルは、off、severe、warning、info、config、fine、finer、finest、all です。

例

- **例 1** – トレース・レベルを表示します。

```
trace
go
```

結果は次のようになります。

TRACE	LEVEL
agent	INFO
audit	ALL
clt	INFO
container	INFO
sql	INFO
std	ALL

version

Replication Server Data Assurance オプションの現在のバージョンを表示します。

構文

```
version
```

例

- **例 1** – バージョンを表示します。

```
version
go
```

結果は次のようになります。

```
VERSION
-----
Replication Server Data Assurance Option/15.7.1/DA Agent/P/generic/
generic/dal57x/121/VM: Sun Microsystems Inc. 1.6.0_24/OPT/Tue 24 Apr
2012 09:24:31 GMT
```

Data Assurance エージェントの予約語

予約語は、DA エージェントでコマンドの一部として使用されると特別な意味を持ちます。DA エージェントでは、単語を二重引用符で囲んで設定しない限り、コマンド構文の一部として使用できません。

表 15 : DA エージェントの予約語

	予約語
C	config、connection
D	dts
J	jvm
P	password
R	role
S	show、shutdown、sslconfig、system
T	task

	予約語
U	user
V	version

セキュリティとアクセス制御

Data Assurance での管理者のセキュリティとアクセス制御

Kerberos セキュリティ

Kerberos は、クライアント／サーバ通信のネットワークベースの認証プロトコルです。

Kerberos では、Kerberos インフラストラクチャを使用するエンタープライズ環境において、集中化された安全な認証メカニズムを提供します。KDC (Key Distribution Center) と呼ばれる信頼されたサード・パーティのサーバを使用して認証が行われ、クライアントとサーバの両方が検証されます。

Kerberos を使用するための DA エージェントの設定

Java Database Connectivity (JDBC) を使用してデータベースに接続するときに、分散配備で DA エージェントが Kerberos の設定を受け入れるように設定します。

この例では、リモート DA エージェントが “omnivore” というサーバにインストールされています。

注意： スタンドアロンの DA サーバ配備では、(DA サーバに組み込まれている) ローカル・エージェントが Kerberos に対応するように、これと同じ手順で設定します。

1. `$$SYBASE/DA-15_5/agent/instance/instance.cfg` に移動します。
2. 問題のトラブルシューティングを実行する場合は、`instance.cfg` ファイルを編集して、`sun.security.krb5.debug` を `true` に設定します。

```
#
# Kerberos
#
javax.security.auth.useSubjectCredsOnly=false
java.security.auth.login.config=${da.instance.dir}/security/
kerberos.conf
sun.security.krb5.debug=true
```

3. `$$SYBASE/DA-15_5/agent/instance/security/` に移動します。
4. `kerberos.conf` ファイルを編集して、`principal` 名と `keytab` ファイル・ロケーションを含めます。

```
com.sun.security.jgss.initiate {
```

セキュリティとアクセス制御

```
com.sun.security.auth.module.Krb5LoginModule required
doNotPrompt=true
debug=true principal=USERNAME
useKeyTab=true keyTab="C:\¥¥ASE1503_krb¥¥SERVERNAME_key"
storeKey=true;
```

5. DA エージェントを再起動します。
6. DA エージェントがサーバ “omnivore” にインストールされていることを確認します。

```
show agent a1
go
```

NAME	HOST	PORT	USER	DESCRIPTION
a1	omnivore	4510	da_admin	

(0 rows affected)

7. DA エージェント “a1” のデータベース・コネクションをダミー・ユーザ名 “my_user” で作成します。

```
create connection c2
set agent a1
and set host omnivore
and set port 5000
and set database dadb
and set user my_user
with properties
    set request_kerberos_session true
    and set service_principal_name "OMNIVORE@ASE"
go
```

8. データベース・コネクションの設定をテストします。

```
test connection c2
go
```

```
RESULT
-----
Succeeded
(0 rows affected)
```

LDAP 認証

LDAP (Lightweight Directory Access Protocol) は、ディレクトリ・サービスにアクセスするための業界標準のクライアント／サーバ・プロトコルです。LDAP サーバは、ユーザ・レポジトリおよび中央認証サービスとして使用される場合があります。

DA では、LDAP ユーザを DA 管理者としてバインドする機能および LDAP ユーザ認証を外部 LDAP 認証サーバに委任する機能がサポートされています。

DA 管理者の役割

事前に定義された関数のセットとその関数を呼び出すことを許可されたユーザのセットで構成された役割です。

DA サーバと DA エージェントによって、DA_Admin と呼ばれる単一の DA 管理者の役割が定義されます。

DA_Admin の役割のメンバには、次が含まれます。

- da_admin – DA サーバおよび DA エージェントに組み込まれた管理者アカウント。
- LDAP ユーザ – `role` コマンドを使用すると、LDAP ユーザを DA 管理者の役割にバインドできます。

参照：

- [role \(111 ページ\)](#)
- [role \(101 ページ\)](#)

LDAP 認証のための DA の設定

DA サーバと DA エージェントを LDAP 認証のために設定するには、`csi.xml` ファイルを変更します。

1. 任意のテキスト・エディタを使用して `csi.xml` ファイルを編集します。

- DA サーバ：
\$SYBASE/DA_15-5/server/instance/security/csi.xml
- DA エージェント：
\$SYBASE/DA_15-5/agent/instance/security/csi.xml

2. `authenticationProvider` パラメータを設定して LDAP サーバを使用します。

```
<?xml version="1.0" encoding="UTF-8"?>
  <configuration xmlns:config="http://www.sybase.com/csi/2.5/
  config"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <authenticationProvider
      name="com.sybase.security.ldap.LDAPLoginModule">
      <options name="ServerType" value="sunone5" />
      <options name="ProviderURL" value="ldap://
  ldap.myserver.com:389" />
      <options name="DefaultSearchBase"
  value="dc=sybase,dc=com" />
      <options name="AuthenticationScope" value="subtree"
    />
    </authenticationProvider>
  </configuration>
```

構文の説明は次のとおりです。

表 16 : LDAP 設定オプション

オプション	説明
ServerType	<p>(オプション) 接続先の LDAP サーバのタイプを指定する。この値によって、いくつかの一般的な設定プロパティのデフォルト値が決定されます。</p> <p>有効な値</p> <ul style="list-style-type: none"> • sunone5 – SunOne 5.x または iPlanet 5.x • msad2k – Microsoft ActiveDirectory、Windows 2000 • nsds4 – Netscape Directory Server 4.x • openldap – OpenLDAP Directory Server 2.x
ProviderURL	<p>LDAP サーバへの接続に使用する URL を指定する。</p> <p>デフォルトは <code>ldap://localhost:389</code> です。</p> <p>このデフォルト値は、LDAP サーバが CSI 対応の製品と同じマシン上にあり、LDAP サーバがデフォルトのポート (389) にインストールされている場合に有効になります。</p>
DefaultSearch-Base	<p>別の検索ベースが認証、役割、属性、および自己登録用に指定されていない場合に使用する LDAP 検索ベースを指定する。</p> <ul style="list-style-type: none"> • <code>dc=<domainname>,dc=<tld></code> – たとえば、<code>sybase.com</code> ドメイン内のマシンには <code>dc=sybase,dc=com</code> という検索ベースがあります。 • <code>o=<company name>,c=<country code></code> – たとえば、Sybase の組織内のマシンでは <code>o=Sybase,c=us</code> となる場合があります。
Authentication-Scope	<p>異なる認証スコープ用のクレデンシャルを定義する。</p> <p>デフォルト値は <code>onelevel</code></p> <p>有効な値</p> <ul style="list-style-type: none"> • <code>onelevel</code> • <code>subtree</code>

3. DA サーバと DA エージェントを再起動します。

SSL セキュリティ

Replication Server Data Assurance オプションでは、SSL (Secure Sockets Layer) がサポートされています。SSL は、クレジットカード番号や株取引などの機密情報をインターネットまたはその他の TCP/IP ネットワーク経由で転送する際のセキュリティを保護するための標準です。

SSL の概要

SSL プロトコルは、TCP/IP の上や、Remote Method Invocation (RMI) または Tabular Data Stream™ (TDS) などのアプリケーション・プロトコルの下で実行されます。

SSL 接続が確立される前に、サーバとクライアントは一連のメッセージを交換し、安全な暗号化セッションをネゴシエートして承認します。

SSL は、認証局 (CA) によって発行された証明書を使用して、身元の確認と検証を行います。証明書は電子的なパスポートのようなもので、認証されたエンティティのパブリック・キーや発行元 CA の署名など、エンティティを識別するために必要な情報がすべて含まれています。

サード・パーティの SSL セキュリティ・メカニズムを使用する方法の詳細については、そのソフトウェアのマニュアルを参照してください。追加情報については、IETF (Internet Engineering Task Force) の Web サイトも参照してください。

SSL のインストールには、次が必要です。

- キーストア –サーバ側 Java KeyStore (JKS)。このキーストアには、DA サーバまたは DA エージェントのプライベート・キーが格納されています。
- トラストストア –クライアント側 Java KeyStore (JKS)。このトラストストアには、クライアントが信頼する DA サーバまたは DA エージェントの証明書が格納されています。

注意： キーストアおよびトラストストアの作成については、このマニュアルでは説明していません。

SSL の有効化

`sslconfig` コマンドを使用して、トランスポート・レイヤ・セキュリティを Remote Method Invocation (RMI) と Data Transfer Socket (DTS) 通信に追加します。

SSL を使用した暗号化は次のとおりです。

- Sybase Control Center Data Assurance プラグインと DA サーバとの間の RMI 通信
- DA サーバとリモート DA エージェントとの間の RMI 通信
- DA サーバとリモート DA エージェントとの間の DTS 通信

SCC Data Assurance プラグインから DA Server RMI への通信のための SSL の有効化

DA サーバと Sybase Control Center (SCC) Data Assurance プラグインを設定すると、SSL を使用してすべての RMI ネットワーク通信を暗号化できます。

1. DA サーバを設定します。

- a) DA サーバへの RMI クライアント・コネクションを設定して、コマンドを発行します。次に例を示します。

```
sslconfig rmi_server_ssl_enabled true
go
sslconfig keystore location_of_keystore_file
go
sslconfig keystore_passwd password
go
sslconfig keypair_passwd password
go
```

- b) DA サーバを再起動します。

2. SCC Data Assurance プラグインを設定します。

- a) SCC agent-plugin.xml スクリプトを編集用に開きます。

```
%SYBASE%\SCC-3_2\plugins\DAMAP\agent-plugin.xml
```

- b) DA RMI および Java トラストストア SSL のプロパティを次のように設定します。

DA サーバ証明書は、独自のトラストストアに追加することも、
\$SCC_HOME/services/EmbeddedWebContainer/cacerts にある SCC
内の既存のトラストストアに追加することもできます。デフォルトのパスワードは changeit です。

例：

```
<properties>
  <set-property property="da.rmi.client.ssl.required" value="myserver" />
  <set-property property="da.rmi.client.debug" value="false" />
  <set-property property="javax.net.ssl.trustStore" value="C:\Sybase
  \SCC-3_2\services\EmbeddedWebContainer\cacerts"/>
  <set-property property="javax.net.ssl.trustStorePassword"
  value="changeit" />
</properties>
```

構文の説明は次のとおりです。

- **da.rmi.client.ssl.required** – すべての RMI 通信が SSL を使用して暗号化されていることを要求する DA サーバのホスト名です。

注意： 複数の DA サーバ用に、カンマで区切られたホスト名のリストを追加できます。

- **da.rmi.client.debug** – デバッグ・モードを有効にします。デフォルトは `false` です。
 - **javax.net.ssl.trustStore** – トラストストア・ファイルのロケーションです。
 - **javax.net.ssl.trustStorePassword** – トラストストア・パスワードです。
- c) Sybase Control Center Data Assurance プラグインを再起動します。

DA サーバから DA エージェントへの RMI 通信のための SSL の有効化

DA サーバとリモート DA エージェントを設定すると、SSL を使用してすべての RMI ネットワーク通信を暗号化できます。

1. DA エージェントを設定します。

- a) DA エージェントへのコマンド・ライン・ツール (CLT) のセッションを確立します。
- b) キーストアが設定されていることを確認します。DA サーバから DA エージェントへの DTS 通信用に SSL が既に有効になっている場合、この手順を省略できます。

キーストアを設定するには、次のコマンドを発行します。

```
sslconfig keystore location_of_keystore_file
go
sslconfig keystore_passwd password
go
sslconfig keypair_passwd password
go
```

- c) **rmi_server_ssl_enabled** オプションを `true` に設定します。

```
sslconfig rmi_server_ssl_enabled true
go
```

- d) DA エージェントを再起動します。

2. DA サーバを設定します。

- a) トラストストアが設定されていることを確認します。DA サーバから DA エージェントへの DTS 通信用に SSL が既に有効になっている場合、この手順を省略できます。

トラストストアを設定するには、次のコマンドを発行します。

```
sslconfig truststore truststore_file_location
go
sslconfig truststore_passwd password
go
```

- b) DA エージェントのホスト名を **rmi_client_ssl_required** ホスト・リストで設定するには、次のように入力します。

```
sslconfig rmi_client_ssl_required host_list
go
```

host list パラメータは、SSL 対応の DTS を必要とする DA エージェント・ホストのカンマ区切りのリストです。

- c) DA サーバを再起動します。

DA サーバから DA エージェントへの DTS 通信のための SSL の有効化

DA サーバとリモート DA エージェントを設定すると、SSL を使用してすべての DTS ネットワーク通信を暗号化できます。

- 1. DA エージェントを設定します。

- a) DA エージェントへのコマンド・ライン・ツール (CLT) のセッションを確立します。
- b) キーストアが設定されていることを確認します。DA サーバから DA エージェントへの RMI 通信用に SSL が既に有効になっている場合、この手順を省略できます。

キーストアを設定するには、次のコマンドを発行します。

```
sslconfig keystore location_of_keystore_file
go
sslconfig keystore_passwd password
go
sslconfig keypair_passwd password
go
```

- c) **dts_client_ssl_required** オプションを **true** に設定します。

```
sslconfig dts_client_ssl_required true
go
```

- d) DA エージェントを再起動します。

- 2. DA サーバを設定します。

- a) トラストストアが設定されていることを確認します。DA サーバから DA エージェントへの RMI 通信用に SSL が既に有効になっている場合、この手順を省略できます。

トラストストアを設定するには、次のコマンドを発行します。

```
sslconfig truststore truststore_file_location
go
sslconfig truststore_passwd password
go
```

- b) DA エージェントのホスト名を **dts_client_ssl_required** ホスト・リストで設定するには、次のように入力します。

```
sslconfig dts_client_ssl_required host_list
go
```

host list パラメータは、SSL 対応の DTS を必要とする DA エージェント・ホストのカンマ区切りのリストです。

- c) DA サーバを再起動します。

パスワード管理

パスワード・ポリシーを設定し、パスワードの暗号化を有効にして、紛失したパスワードまたは忘れたパスワードをリセットします。

パスワード・ポリシー

パスワード・ポリシーによって、DA 管理者のパスワードの十分な安全性を確保します。

パスワード・ポリシーに適用されるルールは、次のとおりです。

- デフォルトの最小のパスワード長は 6 文字。
- デフォルトの最大のパスワード長は 30 文字。
- パスワードで有効な文字は次のとおりです。
 - 0-9
 - A~Z, a~z
 - ハイフン (-) およびアンダースコア (_)

`da.sec.passwdMinLength` プロパティと `da.sec.passwdMaxLength` プロパティを `instance.cfg` に追加すると、パスワード長の値を上書きできます。

- DA サーバ - `SYBASE/DA-15_5/server/instance/instance.cfg`
- DA エージェント - `SYBASE/DA-15_5/agent/instance/instance.cfg`

たとえば、パスワード長の最小値および最大値をそれぞれ 8 および 20 に変更するには、次を追加します。

```
da.sec.passwdMinLength=8
da.sec.passwdMaxLength=20
```

パスワードを紛失した場合または忘れた場合のリセット

パスワードを紛失した場合または忘れた場合にそのパスワードをリセットします。

-P パスワード・リカバリの起動パラメータを使用すると、`da_admin` ユーザのパスワードをリセットできます。このパラメータを使用して、他のアカウントのパスワードをリセットすることはできません。

1. DA サーバまたは DA エージェントが実行されている場合は、次のように停止します。

- 別の DA 管理者が LDAP ログインを使用して認証されている場合、DA 管理者はサーバを停止できます。
 - それ以外の場合は、DA サーバまたは DA エージェント・プロセスを終了します。このプロセスは、プラットフォームによって異なります。
2. 起動スクリプトを次のように実行します。

- DA サーバでは、次のようになります。

- Windows 32 ビット版の場合：

```
%SYBASE%\DA-15_5\server\instance\RUN_instance_32.bat -P
```

- Windows 64 ビット版の場合：

```
%SYBASE%\DA-15_5\server\instance\RUN_instance_64.bat -P
```

- Unix 64 ビット版の場合：

```
$SYBASE/DA-15_5/server/instance/RUN_instance_64.sh -P
```

- DA エージェントでは、次のようになります。

- Windows 32 ビット版の場合：

```
%SYBASE%\DA-15_5\agent\instance\RUN_instance_32.bat -P
```

- Windows 64 ビット版の場合：

```
%SYBASE%\DA-15_5\agent\instance\RUN_instance_64.bat -P
```

- UNIX 64 ビット版の場合：

```
$SYBASE/DA-15_5/agent/instance/RUN_instance_64.sh -P
```

構文の説明は次のとおりです。

- `$SYBASE` (UNIX の場合) または `%SYBASE%` (Windows の場合) は、Data Assurance オプションをインストールしたディレクトリです。
- `instance` は、DA サーバ・インスタンスまたは DA エージェント・インスタンスの名前です。
- `RUN_instance_32.bat` または `RUN_instance_64.bat` は、Windows 上の起動スクリプト・ファイルです。
- `RUN_instance_64.sh` は、UNIX 上の起動スクリプト・ファイルです。
- `-P` は、パスワード・リカバリの起動パラメータです。

DA サーバまたは DA エージェントにより、起動時に新しい `da_admin` パスワードが生成され、ログ・ファイルに書き込まれます。

3. 新しいパスワードを次のように取得します。

- DA サーバ：

```
$SYBASE/DA_15-5/server/instnace/log/da_0.log
```

- DA エージェント：

```
$SYBASE/DA_15-5/agent/instnace/log/da_0.log
```

例：

```
S 2012-04-03 11:59:27.027 CONTAINER
FileLoginModule.changePassword@1
#260 Generated a new password for user "da_admin": "l3Fcza7I"
```

この例での新しいパスワードは、l3Fcza7I です。

4. 新しいパスワードを使用して DA にログインします。

ここで、da_admin パスワードを自分が選択するパスワードに変更することをおすすめします。

パスワードの暗号化

isql -X オプションを使用すると、DA サーバおよび DA エージェントにログインするときにパスワードを暗号化できます。

パスワードの暗号化のレベルは、**clt_password_encryption_reqd** 設定パラメータを使用すると設定できます。

参照：

- config (91 ページ)
- config (109 ページ)

パフォーマンスとチューニング

サーバ設定パラメータのデフォルト値の変更、適切な比較オプションの使用、配備の変更を行うことで、DA サーバのパフォーマンスを調整できます。

配備設定

配備は、パフォーマンスに重大な影響を与える可能性があります。最適なパフォーマンスを得るには、配備を設定するときに次のガイドラインに従ってください。

- データベースからエージェントへの JDBC ネットワーク・トラフィックを最小限に抑えるには、使用しているデータベースとの高速イーサネット・コネクションを共有するマシン上にインストールされている DA エージェントを使用した、分散環境を使用します。
- DA サーバを別のマシンで実行します。

ネットワーク遅延

ネットワーク全体のパフォーマンス、またはネットワーク遅延は、システム・パフォーマンスにおける主要な要素です。DA サーバと DA エージェント間のネットワーク・パフォーマンスを最大化してください。たとえば、WAN ではなく LAN を使用することでネットワークのスループットを高く保ちます。

参照：

- config (91 ページ)

一般設定

ジョブの実行時に全体的なシステム・パフォーマンスを改善する場合に役立つガイドラインです。

- 要件に応じた比較のレベルを選びます。たとえば、ロー・カウントの簡易チェックを毎日実行し、徹底的なロー・データ・チェックを1週間に1度実行するようにスケジュールします。
- 可能な限り、複写の完了後に比較が実行されるようにスケジュールします。
- データベースが **select** 文と **order by** 文に対して最適化されるように設定します。
- できれば、比較する各テーブルにカラム・プライマリ・キーを1個だけ含めます。

- `literal` オプションではなく、`database_hash` オプションを使用して比較を実行します。
- ジョブのカラム・ログではなく、サマリ・レポートを生成します。カラム・ログを選択すると、カラム値のデータベース・ルックアップが余分に加わります。
- 差異が多すぎる場合にジョブがアボートされるように設定します。

参照：

- ロー比較の最適化 (134 ページ)

ロー比較の最適化

ハッシュ・タイプ、カラム比較タイプ、ロー・カウントなど、さまざまな要素を微調整して、ローの比較を最適化します。

DA サーバのパフォーマンスが最大化されるように設定するとき、データベース・サーバそのものがボトルネックになる場合がよくあります。サーバがデータを読み取ってソートして返す速度には限界があります。以下は、ロー比較のパフォーマンスを最適化するためのガイドラインです。

- キー・カラム以外は `row_hash` オプションを使用して比較されるように設定します。
- オプションを次の順序で使用してロー比較を設定します。
 1. `database_hash`
 2. `agent_hash`。そして、データベースと高速イーサネット接続でつながっているマシンに DA エージェントをインストールします。

`database_hash` オプションはロー比較の最速の選択肢ですが、DA サーバと同じマシンに DA エージェントをインストールしている場合は、`agent_hash` よりも `database_hash` を先に使用するメリットは少なくなります。理想的なパフォーマンスの鍵は、DA エージェントを使用してデータベースから DA サーバに送信するデータ量を最小限にすることです。

表 17 : ロー比較の考慮事項

要素	説明
ハッシュ・タイプ	<p>agent_hash よりも database_hash を選択することをおすすめします。</p> <p>データベースの各ローのハッシュは事実上、圧縮の一形態です。ハッシュされたデータは小さいので、送信するデータが少なく、比較するデータが少なくなります。</p> <p>database_hash オプションは送信元データベースでデータを圧縮し、リモート DA エージェントとローカル DA エージェントを使用しているときにパフォーマンスを最大化します。</p> <p>agent_hash を選択すると、DA エージェントはハッシュを実行する前に、まず各ローを完全な形で受け取らなければならないため、通常はこちらの方が時間がかかります。</p>
カラム比較オプション	<p>一般的には、column_hash や literal よりも row_hash オプションを選択します。</p> <p>row_hash は、ローのすべてのカラムに 1 つのハッシュ値を作成します。これは DA サーバが送信して比較できる最小量のデータですが、2 つ以上のロー間の差異も確実に識別できます。</p> <p>個々のカラムの差異を見るには、literal よりも column_hash を選択してください。column_hash で設定されたカラムにはそれぞれ独自のハッシュ値が割り当てられます。ハッシュ・オプションを使用する場合は、データ型が大きいほど利点が大きくなります。</p>
ロー・カウント	<p>インデックスのない大きいテーブルがある場合や、比較セットが複雑な where 制約を定義している場合は、最初の select count (1) クエリの実行に時間がかかる可能性があります。このような場合に、比較を加速するには、enable_row_count を false に設定します。</p>

トラブルシューティング

問題の原因を突き止め、推奨されている解決法を適用してください。

DA サーバまたは DA エージェントのエラーが発生すると、エラー・ログにメッセージが記録されます。比較ジョブの実行中に DA サーバで発生したエラーの診断情報のログを確認してください。

概数値データ型の比較

問題：概数値データ型を使用しているカラムに対し、比較エラーが生成される。

考えられる原因：概数値データ型には、float、double precision、real があります。概数値データ型の正確な値はプラットフォームによって異なる場合があります、次のような比較エラーを引き起こす可能性があります。

- キー・カラムに概数値データ型が含まれている場合、DA サーバがソース・カラムとターゲット・カラムを正しく照合するとは限りません。失敗するたびに、次の2つの誤差が生じます。1つはソース・データベースの「消失」ローで、もう1つはターゲット・データベースの「孤立した」ローです。
- 比較のカラム・セットに概数値データ型が含まれている場合は、ソースとターゲットの一致する2つのローが必ずしも同等であるとは限りません。失敗するたびに、「一貫性のない」誤差が生じます。

解決法：なし。

カラム・マッピングの作成時に概数値データ型をスキップすると、誤差を回避できる可能性があります。ただし、これによって概数値データ型の2つのカラム間の本当の違いを見逃すリスクが生じます。

注意：調整で誤差を修正することはできません。

DA サーバ・メモ不足エラー

問題:DA サーバでメモリ領域が不足し、パフォーマンスの問題が発生します。

解決法 1：比較の最大同時スレッド数を減らす

比較中に、DA サーバは、さまざまな速度で DA エージェントからロー・データを受信するため、サーバは常に各ソースとターゲットのローを何十または何百もバッファリングする可能性があります。個別のローが大きく (ユーザ・データベ

トラブルシューティング

ス・テーブルに依存)、比較の数が多い(ユーザが設定)と、このバッファにより、DA サーバでメモリ不足が発生します。

この問題を解決するには、**comparer_max_concurrent_threads** を現在よりも低い値に設定します。

解決法 2 : LOB フェッチ・サイズを減らす

設定パラメータ **lob_fetch_size** が高い値に設定されている場合があります。

この問題を解決するには、**lob_fetch_size** を現在よりも低い値に設定します。

解決法 3 : 外部ソートの最大サイズを減らす

外部ソート・オプションは、大量のメモリを使用します。デフォルトでは、外部ソートは、メモリに何千ものローを保持します。通常、このことは問題になりませんが、各ローのサイズと、同じ Java 仮想マシン (JVM) 内で同時に発生するアクティビティによって問題が発生する可能性があります。たとえば、DA エージェントを使用して同時に 5 つの比較を行うと、必要なメモリは 5 倍に増えます。または "ローカル・エージェント" が使用されると、ソースおよびターゲットのエージェントと比較関数は同じ JVM メモリ割り当てを共有します。 .

この問題を解決するには、**external_sort_max_size** 設定パラメータの値を変更して、DA エージェントがメモリに保存するローの数を減らします。

解決法 4 : DA サーバが使用できるメモリを増やす

グローバルな解決法:多くのメモリで起動するように DA サーバを設定すると、上記のメモリ不足の問題のすべての原因に対応できます。デフォルトでは、DA サーバを実行する JVM は 512 MB を使用します。DA サーバの **RUN_<instance>.bat** ファイルを編集して、値を増やします (プラットフォームと使用可能なシステム・メモリの量によって異なります)。

外部ソート・オプションの設定

問題:外部ソートを設定するように DA エージェントを設定しましたが、データベースが引き続きソート処理を実行しています。

テーブル内のローの数が **external_sort_activate_size** の値 (デフォルトで 1000 万) よりも少ないので、有効なソート・オプションがアクティブになりません。

解決法:**external_sort_activate_size** の値を減らし、ソースおよびターゲットのテーブル内のローの数よりも少なくします。

比較が失敗し、矛盾が記録される

問題:ジョブ比較の結果では、データベース内のハッシュ・カラムの比較の矛盾は記録されません。

解決法 1: LOB フェッチ・サイズを増やす

image などのラージ・オブジェクト (LOB) カラムに矛盾が存在する可能性があります。デフォルトでは、DA サーバは LOB カラムの最初の 1024 バイトのみを比較するので、LOB 値の一部は比較されません。

この問題を解決するには、`lob_fetch_size` の値を増やします。

解決法 2: LOB フェッチ・サイズを増やす

ソースおよびターゲットのカラム値で、同じ MD5/SHA/CRC32 ハッシュ値が生成される場合があります。

この問題を解決するには、`[literal]` オプションを使用して、ローを再度比較します。

ジョブ比較が応答しなくなる

問題:ジョブの実行に成功したが、その比較 (送信元またはターゲット) の 1 つまたは複数に応答しなくなり、カウントが -1、進行状況が 0 パーセントとなっても、エラー・メッセージが表示されない。

考えられる原因: DA サーバがロー・カウントの完了を待機しています。データベース・テーブルが `select count` クエリ用に最適化されていないと、データベース・サーバがロー・カウントを実行するのに時間がかかる場合があります。DA サーバの待機中は、コマンド・プロンプトにマイナスのカウント、0 パーセントの進行状況が表示され、ジョブのエラー・メッセージは表示されません。

次のいずれかの解決方法を試してください。

- `select count` クエリの実行が加速されるように、キー・カラムに新しいインデックスを作成して、データベース・テーブルを最適化します。
- ジョブ比較を変更して、`enable_row_count` を `false` に設定します。

スタック領域エラーで比較に失敗する

問題：ジョブ比較が完了せず、次のように表示される。大量のスタック領域を使用したため、トランザクションがアボートされました。

考えられる原因：compareset テーブルには数百のカラムが含まれているため、DA エージェントは大容量の select クエリ文字列を作成します。このクエリ文字列が大きすぎて、データベース・サーバにクエリを処理する内部スタック領域が十分でない可能性があります。

次のいずれかの解決方法を使用してください。

- Adaptive Server ストアド・プロシージャ **sp_configure** を使用して、データベース・サーバのスタック領域を増やす。
- DA サーバの設定パラメータ **db_hash_ase_ignore_null** が **false** に設定されている場合は、**true** に設定する。これにより、select クエリ文字列のサイズが減少します。
- それぞれがデータベース・テーブルの半分だけを比較する新しい比較セットを2つ作成してから、現在の比較を置き換える新しい比較を2つ作成し、データベース・テーブルが2段階で2つの比較セットを使用して完全に比較されるようにします。

圧縮テーブルに対する比較の失敗

問題：比較セットが、Adaptive Server 15.7 で作成された1つまたは複数の圧縮テーブルを指している場合、比較に繰り返し失敗する。

考えられる原因：Adaptive Server 15.7 比較メモリ・プールにある欠陥によって、Adaptive Server が、再起動を行わないとリカバリできないエラー状態になります。この欠陥は、Adaptive Server が比較メモリの割り当てに失敗すると発生します。Adaptive Server ログでエラーを確認してください。

次のいずれかを行います。

- **comparer_max_concurrent_threads** および **comparer_max_concurrent_threads** を使用して、Adaptive Server 15.7 から同時に選択する DA スレッドの数を減らす。
 - 次の設定パラメータを編集して Adaptive Server の比較メモリのサイズを増やす。
 - **compression memory size**
 - **compression info pool size**
- これらのパラメータをどの程度増加させるかは、場合によって異なります。

設定パラメータのアルファベット順のリストについては、『Adaptive Server Enterprise システム管理ガイド：第 1 巻』を参照してください。

参照：

- ロー比較ジョブのコマンド (57 ページ)
- config (91 ページ)

追加の説明や情報の入手

Sybase Getting Started CD、製品マニュアル Web サイト、オンライン・ヘルプを利用すると、この製品リリースについて詳しく知ることができます。

- Getting Started CD (またはダウンロード) – PDF フォーマットのリリース・ノートとインストール・ガイド、その他のマニュアルや更新情報が収録されています。
- Sybase 製品マニュアル Web サイト (<http://sybooks.sybase.com/>) にある製品マニュアルは、Sybase マニュアルのオンライン版であり、標準の Web ブラウザを使用してアクセスできます。マニュアルはオンラインで参照することも PDF としてダウンロードすることもできます。この Web サイトには、製品マニュアルの他に、EBFs/Maintenance、Technical Documents、Case Management、Solved Cases、Community Forums/Newsgroups、その他のリソースへのリンクも用意されています。
- 製品のオンライン・ヘルプ (利用可能な場合)

PDF 形式のドキュメントを表示または印刷するには、Adobe の Web サイトから無償でダウンロードできる Adobe Acrobat Reader が必要です。

注意： 製品リリース後に追加された製品またはマニュアルについての重要な情報を記載したさらに新しいリリース・ノートを製品マニュアル Web サイトから入手できることがあります。

サポート・センタ

Sybase 製品に関するサポートを得ることができます。

組織でこの製品の保守契約を購入している場合は、サポート・センタとの連絡担当者が指定されています。マニュアルだけでは解決できない問題があった場合には、担当の方を通して Sybase 製品のサポート・センタまでご連絡ください。

Sybase EBF と Maintenance レポートのダウンロード

EBF と Maintenance レポートは、Sybase Web サイトからダウンロードしてください。

1. Web ブラウザで <http://www.sybase.com/support> を指定します。

2. メニュー・バーまたはスライド式メニューの [Support (サポート)] で [EBFs/Maintenance (EBF/メンテナンス)] を選択します。
3. ユーザ名とパスワードの入力が求められたら、MySybase のユーザ名とパスワードを入力します。
4. (オプション) [Display (表示)] ドロップダウン・リストからフィルタを指定し、期間を指定して、[Go (実行)] をクリックします。
5. 製品を選択します。

鍵のアイコンは、「Authorized Support Contact」として登録されていないため、一部の EBF/Maintenance リリースをダウンロードする権限がないことを示しています。未登録ではあるが、Sybase 担当者またはサポート・センタから有効な情報を得ている場合は、[My Account (マイ・アカウント)] をクリックして、「Technical Support Contact」役割を MySybase プロファイルに追加します。

6. EBF/Maintenance レポートを表示するには [Info] アイコンをクリックします。ソフトウェアをダウンロードするには製品の説明をクリックします。

Sybase 製品およびコンポーネントの動作確認

動作確認レポートは、特定のプラットフォームでの Sybase 製品のパフォーマンスを検証します。

動作確認に関する最新情報は次のページにあります。

- パートナー製品の動作確認については、http://www.sybase.com/detail_list?id=9784 にアクセスします。
- プラットフォームの動作確認については、<http://certification.sybase.com/ucr/search.do> にアクセスします。

MySybase プロファイルの作成

MySybase は無料サービスです。このサービスを使用すると、Sybase Web ページの表示方法を自分専用カスタマイズできます。

1. <http://www.sybase.com/mysybase> を開きます。
2. [Register Now (今すぐ登録)] をクリックします。

アクセシビリティ機能

アクセシビリティ機能を使用すると、身体障害者を含むすべてのユーザーが電子情報に確実にアクセスできます。

Sybase 製品のマニュアルには、アクセシビリティを重視した HTML 版もあります。

オンライン・マニュアルは、スクリーン・リーダーで読み上げる、または画面を拡大表示するなどの方法により、視覚障害を持つユーザがその内容を理解できるよう配慮されています。

Sybase の HTML マニュアルは、米国のリハビリテーション法第 508 条のアクセシビリティ規定に準拠していることがテストにより確認されています。第 508 条に準拠しているマニュアルは通常、World Wide Web Consortium (W3C) の Web サイト用ガイドラインなど、米国以外のアクセシビリティ・ガイドラインにも準拠しています。

注意：アクセシビリティ・ツールを効率的に使用するには、設定が必要な場合もあります。一部のスクリーン・リーダーは、テキストの大文字と小文字を区別して発音します。たとえば、すべて大文字のテキスト (ALL UPPERCASE TEXT など) はイニシャルで発音し、大文字と小文字の混在したテキスト (Mixed Case Text など) は単語として発音します。構文規則を発音するようにツールを設定すると便利かもしれませんが、詳細については、ツールのマニュアルを参照してください。

Sybase のアクセシビリティに対する取り組みについては、Sybase Accessibility サイト (<http://www.sybase.com/products/accessibility>) を参照してください。このサイトには、第 508 条と W3C 標準に関する情報へのリンクもあります。

製品マニュアルには、アクセシビリティ機能に関する追加情報も記載されています。

追加の説明や情報の入手

用語解説

Replication Server Data Assurance オプションに関連する用語の定義です。

Adaptive Server – Sybase バージョン 11.5 およびそれ以降のリレーショナル・データベース・サーバです。Replication Server の設定中に RSSD オプションを選択すると、Adaptive Server は RSSD データベースの Replication Server システム・テーブルを管理します。

DA エージェント – データベースから DA サーバにデータをフェッチして圧縮します。

比較セット – 特定のジョブで比較対象となるものを定義するテーブルとカラムのセットです。

接続プロファイル – データベース・コネクションを確立するために必要な情報です。

データベース – 相互に関連するデータ・テーブルとその他のオブジェクトのセットが、特定の目的に合わせて編成および表現されたものです。

Data Assurance System Database (DASD) – システムや構成の設定を格納する DA サーバのデータベースです。

Data Transfer Stream (DTS) – 比較中に DA エージェントがデータをストリームするために使用するアプリケーション・プロトコルです。

Hypertext Transfer Protocol (HTTP) – インターネットまたはローカル・エリア・ネットワーク上で Web サーバの接続に使用される通信プロトコルです。

Java Database Connectivity (JDBC) – Java アプリケーションで SQL (Structured Query Language) を使用して複数のデータベース管理システムにアクセスするための API (アプリケーション・プログラム・インタフェース) の仕様です。

jConnect – Sybase の高性能 JDBC ドライバです。

ジョブ – 1 つまたは複数の比較タスクのコレクションです。

一貫性のないロー – プライマリとレプリケートの両方のデータベースに存在するテーブル・ローですが、比較される 1 つまたは複数のカラムの値が異なります。

ローカル・エリア・ネットワーク (LAN) – コンピュータとプリンタや端末などのデバイスを、データやデバイスの共有のためにケーブルで接続したシステムです。

消失ロー – プライマリ・データベースにあって、レプリケート・データベースにはないテーブル・ローです。

孤立したロー – レプリケート・データベースにあって、プライマリ・データベースにはないテーブル・ローです。

パラメータ – プロシージャの実行時に提供される値を表す識別子です。ファンクション文字列で使用するパラメータ名は @ 記号で始まります。

プライマリ・キー – 各ローをユニークに識別するテーブル・カラムのセットです。

引用符付き識別子 – スペースや英数字以外の特殊文字が含まれるオブジェクト名、アルファベット以外の文字で始まるオブジェクト名、予約語に相当するオブジェクト名は、正しく解析されるように引用符 (一重または二重) で囲む必要があります。

調整 – ターゲットのデータベース・テーブルがソースのデータベース・テーブルと一致するように更新するプロセスです。

レプリケーション – 1つのデータベースのデータの変更 (レコードの作成、更新、削除など) が別のデータベースの対応するレコードにも適用されるプロセスです。

Remote Method Invocation (RMI) – DA サーバと DA エージェント間の通信に使用されるリモート・プロシージャ・コールです。

Replication Server – Sybase のサーバ・プログラムです。通常、LAN 上で複製データを管理し、同じ LAN または WAN 上にある別の Replication Server から受け取ったデータのトランザクションを処理します。

Replication Server Data Assurance オプション – 複数の Adaptive Server データベース間でロー・データやスキーマを比較して、不一致をレポートする Sybase サーバ・プログラムです。

Replication Server システム・データベース (RSSD) – Replication Server のシステム・テーブルを格納する Adaptive Server データベースです。Replication Server システム・テーブルを、RSSD と SQL Anywhere® ERSSD のどちらに格納するかを選択できます。

ロー比較ジョブ – ローの比較に使用されるジョブです。

スキーマ – データベースの構造体です。

スキーマ比較ジョブ – データベース・オブジェクトのスキーマの比較に使用されるジョブです。

Tabular Data Stream™ (TDS) – Open Client™ アプリケーションと Open Server™ アプリケーションの情報交換に使用されるアプリケーション・プロトコルです。

広域ネットワーク (WAN) – ローカル・エリア・ネットワーク (LAN) をデータ通信回線で接続しているシステムです。

索引

A

abort job コマンド 76
 agent_access_timeout_mins 91
 agent_client_ctx_timeout_secs 91
 agent_max_queue 92
 agent_max_request_queue 92
 alter agent コマンド 35
 alter compareset コマンド 38
 alter connection コマンド 42
 alter job コマンド 57
 alter schema job コマンド 70
 auto_recon_stmt_batch_size 92

C

clt_password_encryption_reqd 93
 comparer_max_concurrent_threads 93
 comparer_recently_finished_ttl_secs 94
 comparer_retry_delay_threshold_secs 93
 comparer_retry_max_keys_per_clause 94
 comparer_retry_min_fill_percent 95
 comparer_retry_min_fill_percent_literal 95
 comparer_retry_min_keys_in_range 94
 comparesets 7
 config コマンド 91, 109
 create agent コマンド 36
 create backup コマンド 88
 create compareset コマンド 51
 create connection コマンド 43
 create job コマンド 63
 create schema job コマンド 73

D

DA サーバのコマンド 35
 DASD 13
 コマンド 88
 Data Assurance エージェント
 スタンドアロン・エージェント 6
 リモート・エージェント 6
 ローカル・エージェント 6
 db_connection_retry_interval 96

db_connection_retry_times 96
 db_hash_ase_algorithm 96
 db_hash_ase_ignore_null 96
 db_hash_ase_using_option 97
 default_column_compare_mode 96
 depend agent コマンド 37
 depend compareset コマンド 54
 depend connection コマンド 46
 disable job コマンド 76
 drop agent コマンド 37
 drop backup コマンド 89
 drop compareset コマンド 55
 drop connection コマンド 47
 drop history コマンド 77
 drop job コマンド 69
 drop schema job コマンド 75

E

enable job コマンド 77
 enable_report_generator 97
 external_sort_activate_size 98
 external_sort_compress_file 98
 external_sort_max_file 98
 external_sort_max_size 97
 external_sort_max_thread 97

F

file_output_encoding 98

I

import job コマンド 83

K

Kerberos
 エージェントの設定 121
 セキュリティ 121
 セキュリティ・メカニズム 121
 ネットワークベース・セキュリティ 121

索引

L

LDAP 認証 122, 123
ユーザ 123
LDAP 認証サーバ
authenticationProvider 123
csi.xml 123
lob_fetch_size 98

M

map all の使用 53
map all の制限事項 53
monitor job コマンド 78

R

recon_tran_max_stmts 99
Replication Server Data Assurance オプション
システム・アーキテクチャ 3
概要 3
統合 14
restore backup コマンド 89
role コマンド 101, 111
run job コマンド 79

S

Secure Sockets Layer 124
show agent connection コマンド 38
show agent dts コマンド 39
show agent system コマンド 40
show agent task コマンド 41
show agent コマンド 38
show backup コマンド 89
show compareset コマンド 55
show connection コマンド 47, 112
show dts コマンド 112
show history コマンド 80
show job コマンド 69
show jvm コマンド 102
show reconcile コマンド 81
show report コマンド 82
show schema job コマンド 75

show system コマンド 103, 113
show task コマンド 114
SSL
DA サーバと Sybase Control Center for Data
Assurance 126
DA サーバとリモート DA エージェント 127,
128
Data Transfer Socket (DTS) 125
DTS プロトコル 128
Remote Method Invocation (RMI) 125
RMI プロトコル 126, 127
Tabular Data Stream (TDS) 125
概要 125
SSL (Secure Sockets Layer) 124
sslconfig コマンド 103, 114
Sybase 共通セキュリティ・インフラストラク
チャ (CSI: Common Security
Infrastructure) 123

T

test agent コマンド 41
test connection コマンド 48
text_report_max_column_width 99
text_report_max_line_length 99
trace コマンド 106, 117
truncate backup コマンド 90
truncate history コマンド 83

V

version コマンド 107, 118

え

エージェントのコマンド 35

か

外部ソートの問題 138
カラムの比較の問題、データベース・ハッシ
ュ・ 139

こ

コマンド

abort job 76
 alter agent 35
 alter compareset 48
 alter connection 42
 alter job 57
 alter schema job 70
 config 91, 109
 create agent 36
 create backup 88
 create compareset 51
 create connection 43
 create job 63
 create schema job 73
 depend agent 37
 depend connection 46
 disable job 76
 drop agent 37
 drop backup 89
 drop compareset 55
 drop connection 47
 drop history 77
 drop job 69
 drop schema job 75
 enable job 77
 import job 83
 monitor job 78
 restore backup 89
 role 101, 111
 run job 79
 show agent 38
 show agent connection 38
 show agent dts 39
 show agent system 40
 show agent task 41
 show backup 89
 show compareset 55
 show connection 47, 112
 show dts 112
 show history 80
 show job 69
 show jvm 102
 show reconcile 81
 show report 82
 show schema job 75
 show system 103, 113
 show task 114
 sslconfig 103, 114

test agent 41
 test connection 48
 trace 106, 117
 truncate backup 90
 truncate history 83
 version 107, 118
 ユーザ名 101, 111

コマンド、
 depend compareset 54
 コマンド・ライン・ツール 35

さ

作業の開始 15
 削除
 ジョブ履歴 33
 バックアップ 33

し

システム・データベース 13
 主要オプション 10
 ジョブ
 スキーマ比較 8
 ロー比較 8
 ジョブ・オプション 10
 ジョブの管理 76
 ジョブの履歴 13
 ジョブ比較の計画 8
 ジョブ・レポート 13

す

スキーマ比較ジョブのコマンド 70
 スタンドアロン・エージェント 6

せ

制限事項 53
 セキュリティ
 LDAP 122
 パスワード・ポリシー 129
 パスワードのリセット 129
 パスワード管理 129
 接続プロファイルのコマンド 42

索引

設定パラメータ

- agent_access_timeout_mins 91
- agent_client_ctx_timeout_secs 91
- agent_max_queue 91
- auto_recon_stmt_batch_size 91
- clt_password_encryption_reqd 91, 109
- comparer_max_concurrent_threads 91
- comparer_recently_finished_ttl_secs 91
- comparer_retry_delay_threshold_secs 91
- comparer_retry_max_keys_per_clause 91
- comparer_retry_min_fill_percent 91
- comparer_retry_min_fill_percent_literal 91
- comparer_retry_min_keys_in_range 91
- db_connection_retry_interval 91
- db_connection_retry_times 91
- db_hash_ase_algorithm 91
- db_hash_ase_ignore_null 91
- db_hash_ase_using_option 91
- default_column_compare_mode 91
- enable_report_generator 91
- external_sort_activate_size 91
- external_sort_compress_file 91
- external_sort_max_file 91
- external_sort_max_size 91
- external_sort_max_thread 91
- lob_fetch_size 91
- recon_tran_max_stmts 91
- text_report_max_column_width 91
- text_report_max_line_length 91

そ

その他のコマンド 90

た

- タスク・フロー
 - サーバ設定パラメータ 33
 - ジョブ 31
 - ジョブのインポート 32
 - スキーマ・ジョブ 31
 - データ・ファイルの削除 33
 - バックアップ 33
 - リストア 33
 - ログ・ファイルの削除 33

ち

- 調整 8
 - スクリプト 24

- データベース・テーブル 24
- 自動 24

て

- データベース接続 7, 42
- データベース・ハッシュ・カラムの比較の問題 139

と

- トラブルシューティング 137
 - 外部ソートの問題 138
 - ジョブ比較が応答していない 139
 - スタック領域エラー 140
 - データベース・ハッシュ・カラムの比較の問題 139
 - メモリ不足エラー 137
 - 圧縮テーブル 140
 - 概数値データ型 137
 - 比較メモリの割り当て 140

は

- バージョン
 - DA サーバ 107
 - リモート DA エージェント 118
- パスワード
 - パスワード長 129
 - ポリシー 129
 - リセット 129
- パスワードの暗号化 131
- パフォーマンスとチューニング 133
 - ロー比較の最適化 134
 - 一般設定 133

ひ

- 比較オプション 10
- 比較セット
 - コマンド 48
- 比較タスク 8
- 比較方式 8
- 表記規則
 - スタイル 1

構文 1

め

メモリ不足エラー、DA サーバ 137

ゆ

ユーザ・タスク・フローの例 15, 22

よ

予約語

DA サーバ 107

リモート DA エージェント 118

り

リモート DA エージェントのコマンド 109

show connection 112

show dts 112

show jvm 113

show task 114

sslconfig 114

trace 117

れ

レポート 13

ろ

ローカル・エージェント 6

ロー比較ジョブのコマンド 57

