



システム管理ガイド：第 1 巻

Adaptive Server[®] Enterprise

15.7

ドキュメント ID : DC36426-01-1570-01

改訂 : 2011 年 9 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。

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 trademarks ページ** (<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 関連の商標は、米国およびその他の国における Sun Microsystems, Inc. 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

第 1 章	システム管理の概要	1
	Adaptive Server の管理作業.....	1
	システム管理作業に必要な役割.....	2
	システム管理作業のための isql の使用.....	5
	システム管理作業での Sybase Central の使用.....	6
	システム・テーブル.....	7
	システム・テーブルの問い合わせ.....	8
	システム・テーブル内のキー.....	8
	システム・テーブルの更新.....	9
	システム・プロシージャ.....	10
	システム・プロシージャの使用.....	10
	システム・プロシージャ・テーブル.....	11
	システム・プロシージャの作成.....	12
	システム拡張ストアド・プロシージャ.....	12
	システム ESP の作成.....	13
	エラー・メッセージのログ.....	13
	Adaptive Server との接続.....	14
	interfaces ファイル.....	14
	ディレクトリ・サービス.....	15
	ディレクトリ・サービスとしての LDAP.....	15
	Adaptive Server で使用できるセキュリティ機能.....	18
第 2 章	システム・データベースとオプションのデータベース	19
	システム・データベースの概要.....	19
	master データベース.....	21
	master でのオブジェクト作成の制御.....	22
	master のバックアップとシステム・テーブルのコピー.....	22
	model データベース.....	23
	sybssystemprocs データベース.....	24
	tempdb データベース.....	24
	テンポラリ・テーブルの作成.....	25
	sybsecurity データベース.....	26
	sybssystemdb データベース.....	26
	sybmgmtdb データベース.....	26

	pubs2 と pubs3 のサンプル・データベース	27
	サンプル・データベースの管理	27
	pubs2 image データ	28
	dbccdb データベース	28
	sybdiag データベース	28
	インストール・スクリプトのバージョンの確認	29
第 3 章	システム管理の基礎	31
	論理ページ・サイズ	31
	テスト・サーバの使用方法	32
	リソースの計画	32
	パフォーマンスの目標の達成	33
	Sybase 製品のインストール時の考慮事項	33
	製品の互換性のチェック	33
	Adaptive Server のインストールまたはアップグレード	33
	追加のサードパーティ・ソフトウェアのインストール	34
	クライアント接続の設定とテスト	34
	物理リソースの割り付け	34
	専用サーバと共有サーバ	35
	意思決定支援処理と OLTP アプリケーション	35
	リソースの使用計画	36
	オペレーティング・システムの設定	36
	バックアップとリカバリ	37
	master の最新のバックアップの保持	37
	バックアップ手順の自動化	38
	データベースをバックアップする前のデータの一貫性の確認	39
	ログ・サイズのモニタ	40
	継続して実行する管理作業とトラブルシューティング	40
	Adaptive Server の起動と停止	40
	エラー・ログの表示と削除	40
	記録の保管	41
	連絡先の情報	41
	設定情報	41
	管理作業のスケジュール	42
	システム情報	42
	災害時のリカバリ計画	43
	その他のリソース	43
第 4 章	Adaptive Server の管理とモニタリング	45
	Sybase Control Center for Adaptive Server	45
	Adaptive Server Sybase Central プラグイン	46
	Adaptive Server プラグインの使用	47
	Sybase Central の起動と停止	48
	Adaptive Server プラグインの登録	48
	共通の作業の実行	49
	Interactive SQL の使用	55

第 5 章	設定パラメータ	57
	概要	57
	Adaptive Server の設定ファイル	58
	設定パラメータの変更	58
	設定パラメータの変更に必要な役割	58
	sp_configure による単位の指定	60
	グローバル設定とセッション設定	60
	設定パラメータのヘルプ情報の取得	61
	sp_configure の使用	62
	構文の要素	63
	設定ファイルを指定して sp_configure を使用する方法	63
	パラメータの階層	67
	パラメータ階層のユーザ定義サブセット (表示レベル)	70
	sp_configure と sp_sysmon によるパフォーマンス・チューニング	71
	クラスタード環境における設定パラメータの使用	71
	sp_configure 出力	72
	Named Cache 設定パラメータ	74
	sysconfigures テーブルと syscurconfigs テーブル	75
	syscurconfigs と sysconfigures へのクエリ (例)	75
	設定パラメータ	75
	設定パラメータのアルファベット順リスト	76
第 6 章	ディスク・リソースについての概要	273
	デバイスの割り付けとオブジェクトの配置	273
	ディスク・リソースの管理に使用するコマンド	274
	記憶領域の管理に関する考慮事項	276
	リカバリ	276
	パフォーマンス	277
	インストール時のステータスおよびデフォルト設定	277
	記憶領域を管理するシステム・テーブル	278
	sysdevices テーブル	279
	sysusages テーブル	280
	syssegments テーブル	281
	sysindexes テーブル	281
	syspartitions テーブル	281
第 7 章	リモート・サーバの管理	283
	概要	283
	リモート・サーバの管理	285
	リモート・サーバの追加	285
	リモート・サーバ名の管理	287
	サーバ接続オプションの設定	287
	サーバ情報の取得	289
	リモート・サーバの削除	289

	リモート・ログインの追加	289
	ユーザのサーバ ID のマッピング方法	290
	リモート・ログインを特定のローカル名にマップする方法	290
	すべてのリモート・ログインを 1 つのローカル名に マップする方法	291
	ローカル・サーバのリモート・ログイン名の保持	292
	リモート・ユーザ・ログインのマッピング例	292
	リモート・ユーザのパスワードの検査	294
	untrusted モードを使用した場合の影響	294
	リモート・ログイン情報の取得	295
	リモート・ログインの設定パラメータ	295
第 8 章	データベース・デバイスの初期化.....	297
	データベース・デバイス	297
	disk init コマンドの使用	298
	disk init の構文	299
	論理デバイス名の指定	299
	物理デバイス名の指定	299
	デバイス番号の選択	299
	デバイス・サイズの指定	300
	dsync 設定の指定 (オプション)	301
	directio によるオペレーティング・システム・バッファの回避	303
	disk init のその他のオプション・パラメータ	304
	デバイス情報の表示	305
	デバイスの削除	307
	デフォルト・デバイスの指定	307
	デフォルト・デバイスと非デフォルト・デバイスの選択	308
	disk resize コマンドによるデバイスのサイズ拡大	308
	ディスク領域の不足	310
第 9 章	データベース・オプションの設定.....	311
	sp_dboption プロシージャの使用	311
	データベース・オプションの説明	312
	データベースの各オプションの表示	313
	sysoptions による現在設定されているスイッチの表示	314
第 10 章	文字セット、ソート順、言語の設定.....	317
	国際化とローカライゼーションの概要	317
	国際化されたシステムの利点	318
	サンプル国際化システム	319
	国際化システムの要素	321
	サーバの文字セットの選択	321
	Unicode	324
	サーバのデフォルト文字セットの選択	327

ソート順の選択	330
ソート順の使用	331
ソート順の種類	331
デフォルト・ソート順の選択	332
システム・メッセージ用言語の選択	338
サーバの設定：例	339
スペイン語版サーバ	340
アメリカ企業の日本法人	340
クライアントが複数の国にある日本企業	341
文字セット、ソート順、メッセージ言語の変更	341
デフォルト文字セットの変更	342
リソース・ファイルを使ったソート順の変更	343
デフォルト・ソート順の変更	343
文字セット、ソート順、メッセージ言語の再設定	344
Unicode の例	344
準備手順	346
ユーザのデフォルト言語の設定	347
再設定後のリカバリ	348
疑わしいパーティションの処理	351
サポートされていない言語の日付文字列のインストール	352
サーバとクライアントでの日付の解釈	353
国際化ファイルとローカライゼーション・ファイル	354
国際化ファイルの種類	354
文字セットのディレクトリ構造	354
ローカライゼーション・ファイルの種類	356
ソフトウェア・メッセージのディレクトリ構造	356
メッセージ言語とグローバル変数	357
第 11 章	
クライアント／サーバの文字セット変換の設定	359
文字セット変換	359
サポートする文字セット変換	360
ネイティブな文字セットでの変換	360
Unicode システムでの変換	361
Adaptive Server 直接変換	362
Unicode 変換	362
変換タイプの選択	363
非 Unicode クライアント／サーバ・システム	364
Unicode クライアント／サーバ・システム	364
サーバの設定	365
文字セット変換の有効化と無効化	366
変換できない文字	366
文字セット変換のエラー処理	367
変換とデータ長の変更	367
システムとアプリケーションの設定	368
ユーティリティ・プログラムのための文字セットの指定	369
表示およびファイル文字セットのコマンド・ライン・オプション	369

第 12 章	システムの問題の診断	371
	Adaptive Server のエラー・メッセージ	371
	エラー・ログのフォーマット	373
	エラー・メッセージおよびメッセージ番号	374
	エラー・メッセージ・テキスト内の変数	375
	Adaptive Server エラー・ロギング	375
	重大度レベル	376
	重大度レベル 10 ~ 18	377
	重大度レベル 19 ~ 26	380
	エラーのレポート	382
	Backup Server のエラー・ロギング	382
	プロセスの強制終了	384
	statusonly を指定した kill の使用	387
	sp_lock によるブロック・プロセスの調査	388
	ハウスキーピング機能	388
	ハウスキーピング・ウォッシュ	389
	ハウスキーピング・チョア	389
	ハウスキーピング・ガーベジ・コレクション	389
	enable housekeeper GC の設定	390
	サーバの停止	392
	Adaptive Server の停止	392
	Backup Server の停止	392
	既知の問題についての情報	393
索引		395

システム管理の概要

この章では、Adaptive Server[®] のシステム管理についての基本的なトピックを説明します。

トピック名	ページ
Adaptive Server の管理作業	1
システム・テーブル	7
システム・プロシージャ	10
システム拡張ストア・プロシージャ	12
エラー・メッセージのログ	13
Adaptive Server との接続	14
Adaptive Server で使用できるセキュリティ機能	18

Adaptive Server の管理作業

Adaptive Server の管理作業には、次のものがあります。

- Adaptive Server および Backup Server のインストール
- Adaptive Server ログイン・アカウントの作成と管理
- Adaptive Server ユーザに対する役割とパーミッションの付与
- 接続、メモリ、ディスク領域の使用の管理とモニタ
- データベースのバックアップとリストア
- システム上の問題の診断
- パフォーマンスを最大にするための Adaptive Server の設定

さらにシステム管理者は、整合性基準の適用など、データベース設計作業を支援する場合があります。このような役割は、アプリケーション設計者とも共通しています。

システム管理者は一般的に、Adaptive Server 上で実行されるアプリケーションとは直接関係のない作業を主に行いますが、すべてのアプリケーションを最もよく把握できる立場にあります。このためシステム管理者は、アプリケーション設計者に対して Adaptive Server 上の既存のデータに関するアドバイスや、複数のアプリケーションに関するデータ定義の標準化についての助言などができます。

しかし、アプリケーション独自の機能とそうでない機能の判別が難しい場合もあります。ユーザ・データベースの所有者はこのマニュアルの該当する箇所を参照してください。同様に、システム管理者とデータベース所有者は、『Transact-SQL ユーザーズ・ガイド』(特に、データ定義、ストアド・プロシージャ、トリガに関する章)を参照してください。システム管理者とアプリケーション設計者は、『パフォーマンス&チューニング・シリーズ』を参照してください。

システム管理作業に必要な役割

このマニュアルで説明されている多くのコマンドやプロシージャでは、システム管理者またはシステム・セキュリティ担当者の役割を必要とします。コマンドやプロシージャ以外の章は、データベース所有者に関連するものです。

さまざまなセキュリティ関連、管理、運用の作業は、ユーザの役割ごとに次のように分けられます。

- 「システム管理者」(sa) は、デフォルトでは次の役割を持ちます。
 - sa_role
 - sso_role
 - oper_role
 - sybase_ts_role

システム管理者の作業には次のものがあります。

- ディスク記憶領域の管理
- Adaptive Server の自動リカバリ・プロシージャのモニタ
- 設定可能なシステム・パラメータの変更による Adaptive Server のチューニング
- システムの問題の診断と報告
- データベースのバックアップとロード
- サーバ・ログイン・アカウントの変更と削除
- システム管理者の役割の付与と取り消し
- Adaptive Server ユーザへのパーミッションの付与
- ユーザ・データベースの作成とそのデータベースの所有権の付与
- パーミッションの付与と取り消しで使用できるグループの設定

- 「システム・セキュリティ担当者」は、次のようなセキュリティ関連作業を実行します。
 - サーバ・ログイン・アカウントの作成 (初期パスワードの割り当てを含む)
 - アカウントのパスワードの変更
 - システム・セキュリティ担当者とオペレータの役割の付与と取り消し
 - ユーザ定義の役割の作成、付与、取り消し
 - サーバ内で別のユーザになり代わる権限の付与
 - パスワードの有効期間の設定
 - ネットワーク・ベースのセキュリティ・サービスを使用するための Adaptive Server の設定
 - 監査システムの管理
- 「オペレータ」は、サーバ全体にわたってデータベースのバックアップとロードを実行します。オペレータの役割によって、1人のユーザが **dump database**、**dump transaction**、**load database**、**load transaction** コマンドを使い、各データベースの所有者にならなくても、サーバ上のすべてのデータベースのバックアップとリストアを実行できます。1つのデータベース内では、データベース所有者またはシステム管理者がこれらの操作を実行できますが、オペレータはすべてのデータベースに対してこれらの操作を実行できます。

これらの役割により、システムの操作と管理作業を実行するユーザの責任が明確になります。これらの作業は監査することができ、その責任は役割を付与されているユーザにあります。システム管理者は、任意アクセス制御 (DAC) 保護システムの外部で操作を行います。つまり、システム管理者がオブジェクトにアクセスするときは、Adaptive Server は DAC パーミッションをチェックしません。

さらにオブジェクト所有者には、所有するオブジェクトによって特別なステータスを持つ、2つのタイプがあります。次のタイプの所有者です。

- データベース所有者
- データベース・オブジェクトの所有者

データベース所有者

「データベース所有者」は、データベースを作成したユーザ、またはデータベースの所有権を譲渡されたユーザです。システム管理者は **grant** コマンドを使って、ユーザにデータベースを作成する権限を付与します。

データベース所有者は、自分に割り当てられたログイン名とパスワードを使って Adaptive Server にログインし、“dbo” アカウントを所有します。自分が作成したものでないデータベースにログインする場合は、ユーザは通常の自分のユーザ名で識別されます。

データベース所有者は次のことができます。

- システム・プロシージャ **sp_adduser** を実行して、他の Adaptive Server ユーザがデータベースにアクセスできるようにする。
- **grant** コマンドを使って、データベース内での、オブジェクト作成やコマンド実行のためのパーミッションを他のユーザに付与する。

データベースにユーザを追加する方法の詳細については、『セキュリティ管理ガイド』の「第3章 Adaptive Server のログイン・アカウントとデータベース・ユーザの管理」を参照してください。ユーザにパーミッションを付与する方法については、「第6章 ユーザ・パーミッションの管理」を参照してください。

データベース所有者は、他のユーザが所有しているオブジェクトのパーミッションを自動的には受け取りません。ただしデータベース所有者は、**setuser** コマンドを使っていつでもデータベース内の別のユーザになり、一時的にそのユーザのパーミッションを使うことができます。**setuser** と **grant** コマンドを組み合わせることで、データベース所有者はデータベース内のどのオブジェクトのパーミッションでも取得することができます。

注意 データベース所有者の役割は非常に強力であるため、システム管理者は、サーバ内のデータベースの所有者をどのユーザにするかを慎重に検討してください。また、システム・セキュリティ担当者は、すべてのデータベース所有者のデータベース・アクティビティを監査するようにしてください。

データベース・オブジェクトの所有者

「データベース・オブジェクト所有者」とは、データベース・オブジェクトを作成するユーザです。「データベース・オブジェクト」とは、テーブル、インデックス、ビュー、デフォルト、トリガ、ルール、制約、プロシージャです。ユーザがデータベース・オブジェクトを作成するには、データベース所有者がそのユーザに対して、特定タイプのオブジェクトを作成するためのパーミッションを付与する必要があります。データベース・オブジェクト所有者としての特別なログイン名やパスワードはありません。

データベース・オブジェクト所有者は、**create** 文を使ってオブジェクトを作成してから、他のユーザにパーミッションを付与します。

データベース・オブジェクト所有者には、そのオブジェクトに対するすべてのパーミッションが自動的に付与されます。システム管理者にも、そのオブジェクトに対するすべてのパーミッションが与えられます。オブジェクトの所有者は、他のユーザがそのオブジェクトにアクセスできるようにするために明示的にパーミッションを付与する必要があります。オブジェクトの所有者が適切なパーミッションを付与しないと、データベース所有者であってもそのオブジェクトを直接使用することはできません。ただし、データベース所有者はいつでも `setuser` コマンドを使ってオブジェクト所有者をはじめとするデータベース内の別のユーザになり代わることができます。

注意 データベース・オブジェクトがデータベース所有者以外のユーザによって所有されている場合、そのオブジェクトにアクセスするには、システム管理者であってもオブジェクト所有者の名前でオブジェクト名を修飾する (`ownername.objectname`) 必要があります。多数のユーザが同じオブジェクトまたはプロシージャにアクセスする必要がある場合、特にアドホック・クエリでアクセスする場合は、これらのオブジェクトの所有者を “dbo” にしておくと同様にアクセスできます。

システム管理作業のための `isql` の使用

このマニュアルで説明するシステム管理作業は、コマンド・ライン・ユーティリティ `isql` を使用して行うことを想定しています。この項では、`isql` の使用に関する基本的な事項を説明します。詳細については、『ユーティリティ・ガイド』を参照してください。

このマニュアルで説明している作業の多くは、Sybase® Central™ というグラフィック・ツールを使用して実行できます。詳細については、「[システム管理作業での Sybase Central の使用](#)」(6 ページ)を参照してください。

`isql` の起動

ほとんどのプラットフォームで `isql` を起動するには、オペレーティング・システムのプロンプトで次のコマンドを入力します。`username` は、システム管理者のユーザ名です。

```
isql -Uusername
```

パスワードの入力を要求するプロンプトが表示されます。

注意 パスワードの指定には、`isql` の `-P` オプションを使用しないでください。このオプションを使用すると他のユーザにパスワードがわかってしまいます。

コマンド・ライン・モードで `isql` を使用して、このマニュアルにある Transact-SQL 例の多くを入力できます。

文の入力

`isql` では、文を複数行に分けて入力することができます。新たな行で “go” を入力すると、`isql` による文の処理が開始します。次に例を示します。

```
1> select *
2> from sysobjects
3> where type = "TR"
4> go
```

このマニュアルの例では、文と文の間の `go` コマンドは示していません。例に従って入力する場合に、結果の出力を参照するには `go` コマンドを入力してください。

文の保存と再使用

Transact-SQL 文を使用してユーザ・データベースとデータベース・オブジェクトを作成または変更する場合は、Transact-SQL 文をその都度保存してください。そのためには、ASCII ファイル形式で文を作成またはコピーするのが最も簡単な方法です。そうすれば、データベースまたはデータベース・オブジェクトを後で作り直す場合に、そのファイルを使用して `isql` に文を入力できます。

ASCII フォーマットのファイルを指定して `isql` を実行する場合の構文は次のとおりです。`filename` は、Transact-SQL 文が入力されているファイルのフル・パスとファイル名です。

```
isql -Uusername -ifilename
```

UNIX と他のプラットフォームでファイルをリダイレクトするには、小なり記号 (<) を使用します。

ASCII ファイル内の Transact-SQL 文は有効な構文で記述する必要があります。また、`go` コマンドを使用する必要があります。

ファイルからコマンドを読み込む場合、次の作業を行う必要があります。

- コマンド・ラインで `-Ppassword` オプションを指定する。または
- 入力ファイルの先頭行に指定するユーザのパスワードを追加する。

システム管理作業での Sybase Central の使用

システム管理作業の多くは、Adaptive Server に付属している Sybase Control Center および Sybase Central というグラフィック・ツールを使用して実行できます。

- データベース・デバイスの初期化
- 設定パラメータの設定
- データベースの空きログ領域の容量表示
- データ定義言語 (DDL) の生成

- ログインの作成
- リモート・サーバの追加
- データベースの作成
- ストアド・プロシージャの作成
- 役割の定義
- データ・キャッシュの追加
- データベース・オプションの設定
- データベースのバックアップとリストア

Sybase Central の Monitor Viewer 機能を使用して、Adaptive Server Monitor™ にアクセスすることもできます。Sybase Central には、詳細なオンライン・ヘルプが用意されています。

Sybase Central の DDL 生成機能を使用して、作業を Transact-SQL スクリプトに記録できます。DDL 生成機能を利用すると、サーバ全体または特定のデータベース内で行う動作をスクリプトに保存できます。

システム・テーブル

master データベースには、Adaptive Server の情報を記録する「システム・テーブル」があります。また、各データベース (**master** データベースも含む) には、そのデータベース特有の情報を記録するシステム・テーブルがあります。

master データベース (Adaptive Server の制御データベース) 内の Adaptive Server によって作成されるすべてのテーブルは、システム・テーブルと見なされます。また、各ユーザ・データベースが作成される時、このようなシステム・テーブルのサブセットも作成されます。システム・テーブルは、「データ辞書」またはシステム・カタログと呼ぶこともあります。

master データベースとそのテーブルは、Adaptive Server のインストール時に自動的に作成されます。ユーザ・データベース内のシステム・テーブルは、**create database** コマンドを発行した時点で作成されます。システム・テーブル名は、すべて“sys”で始まります。ユーザ・データベース内に、システム・テーブルと同じ名前のテーブルを作成することはできません。システム・テーブルとそのカラムの詳細については、『リファレンス・マニュアル：テーブル』を参照してください。

システム・テーブルの問い合わせ

他のテーブルと同様の方法でシステム・テーブルを問い合わせることができます。例として、データベース内のすべてのトリガ名を返す文を次に示します。

```
select name
from sysobjects
where type = "TR"
```

さらに、Adaptive Server に付属している「ストアド・プロシージャ」(システム・プロシージャ)を利用して、システム・テーブルを簡単に問い合わせることができます。

次のリストは、システム・テーブルからの情報を返すシステム・プロシージャです。

• sp_commonkey	• sp_helpremotelogin
• sp_configure	• sp_help_resource_limit
• sp_countmedatada	• sp_helpprotect
• sp_dboption	• sp_helpsegment
• sp_estspace	• sp_helpserver
• sp_help	• sp_helpsort
• sp_helppartition	• sp_helptext
• sp_helpcache	• sp_helpthreshold
• sp_helpconfig	• sp_helpuser
• sp_helpconstraint	• sp_lock
• sp_helpdb	• sp_monitor
• sp_helpdevice	• sp_monitorconfig
• sp_helpgroup	• sp_showcontrolinfo
• sp_helpindex	• sp_showexeclass
• sp_helpjava	• sp_showplan
• sp_helpjoins	• sp_spaceused
• sp_helpkey	• sp_who
• sp_helplanguage	• sp_help_resource_limit
• sp_helplog	

システム・プロシージャの詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

システム・テーブル内のキー

システム・テーブルのプライマリ・キー(主キー)、外部キー、共通キーは、master データベースと model データベース内に定義されます。システム・プロシージャ `sp_helpkey` を実行すると、定義されたキーに関するレポートを出力できます。2つのシステム・テーブルをジョインするときを使用できる可能性のあるカラムに関するレポートを出力するには、`sp_helpjoins` を実行します。

『Adaptive Server システム・テーブル・ダイアグラム』には、システム・テーブルのカラム間の関係が記載されています。

システム・テーブルの更新

Adaptive Server のシステム・テーブルには、データベースを運用する上で重要な情報が格納されています。通常、システム・テーブルのデータを直接変更する必要はありません。

Sybase の製品の保守契約を結んでいるサポート・センタから指示された場合、または『トラブルシューティング&エラー・メッセージ・ガイド』やこのマニュアルに指示がある場合を除いて、システム・テーブルは更新しないでください。

システム・テーブルを更新する場合は、システム・テーブルの更新を可能にする `sp_configure` コマンドを発行しなければなりません。このコマンドが有効な間は、適切なパーミッションを持つユーザであれば誰でもシステム・テーブルを変更できます。システム・テーブルを直接変更する場合の条件は、次のとおりです。

- システム・テーブルの変更は、必ずトランザクション内で行ってください。`begin transaction` コマンドを発行してから、データ変更コマンドを発行します。
- 変更したいローだけがコマンドの影響を受けたことと、そのデータが正確に変更されたことを確認してください。
- コマンドが正しくない場合は、`rollback transaction` コマンドを発行します。コマンドが正しい場合は、`commit transaction` コマンドを発行します。

警告！ どのような状況でも、どのユーザも変更してはならないシステム・テーブルがあります。システム・テーブルには、システム・プロセスによって動的に構築されるものや、コード化された情報を含むもの、あるいは問い合わせを実行してもそのデータの一部しか表示されないものがあります。不用意に通常と違う方法でシステム・テーブルを更新すると、Adaptive Server の実行またはデータベース・オブジェクトに対するアクセスが不可能になる場合があります。また、オブジェクトに対するパーミッションの混乱を招いたり、ユーザ・セッションが終了したりする場合もあります。

さらに、システム・テーブルの定義はどのような形であっても変更しないでください。たとえば、制約を含むようにシステム・テーブルを変更しないでください。トリガ、デフォルト、ルールはシステム・テーブルでは許可されていません。トリガを作成しようとしたり、ルールやデフォルトをシステム・テーブルにバインドしようとしたりすると、エラー・メッセージが返されます。

システム・プロシージャ

システム・プロシージャの名前は、すべて“sp_”で始まります。システム・プロシージャは、**sybssystemprocs** データベース内にありますが、その多くはどのデータベース内でも実行できます。実行するには、そのデータベースからストアド・プロシージャを発行するか、プロシージャ名をデータベース名で修飾します。

Sybase が提供するシステム・プロシージャ (**sp_who** など) は、*installmaster* インストール・スクリプトを使用して作成されています。最後に実行された *installmaster* のバージョンを判断するには、**sp_version** を使用します。**sp_version** の詳細については、『リファレンス・マニュアル：システム・プロシージャ』を参照してください。

sybssystemprocs 以外のデータベースでシステム・プロシージャを実行した場合は、プロシージャの操作の対象はシステム・プロシージャが実行されたデータベース内のシステム・テーブルになります。たとえば、**pubs2** のデータベース所有者が **pubs2** から **sp_adduser** を実行するか、または **pubs2..sp_adduser** コマンドを発行すると、**pubs2..sysusers** に新しいユーザが追加されます。ただし、このことは、**master** データベース内のテーブルだけを更新するシステム・プロシージャには適用されません。

システム・プロシージャに対するパーミッションについては、『リファレンス・マニュアル：プロシージャ』を参照してください。

システム・プロシージャの使用

「パラメータ」は、ストアド・プロシージャやシステム・プロシージャの引数です。システム・プロシージャのパラメータ値に予約語、句読記、または埋め込みブランクがある場合は、一重引用符か二重引用符で囲んでください。パラメータがオブジェクト名で、そのオブジェクト名がデータベース名または所有者名で修飾されている場合は、その名前全体を一重引用符か二重引用符で囲んでください。

システム・プロシージャは、連鎖トランザクション・モードと非連鎖トランザクション・モードのどちらかを使用して、セッション中に呼び出すことができます。連鎖モードでは、データ検索文またはデータ修正文の前に暗黙的にトランザクションが開始されます。非連鎖モードでは、トランザクションを完了するために **commit transaction** 文や **rollback transaction** 文と対になる明示的 **begin transaction** 文が必要です。『Transact-SQL ユーザーズ・ガイド』の「第 21 章 トランザクション：データの一貫性およびリカバリ」を参照してください。

master データベースのシステム・テーブルにあるデータを変更するシステム・プロシージャは、トランザクション内では実行できません。このようにすると、データベースのリカバリで問題が発生する可能性があるためです。また、テンポラリ・ワークテーブルを作成するシステム・プロシージャも、トランザクション内では実行できません。

システム・プロシージャの実行時にアクティブなトランザクションがない場合、Adaptive Server は連鎖モードをオフにして、そのプロシージャの実行中は `transaction isolation level 1` を設定します。復帰する前に、セッションの連鎖モードと独立性レベル (隔離性水準ともいいます) は元の設定にリセットされます。『Transact-SQL ユーザーズ・ガイド』の「第21章 トランザクション: データの一貫性およびリカバリ」を参照してください。

すべてのシステム・プロシージャは、リターン・ステータスをレポートします。たとえば、次の例は、プロシージャが正しく実行されたことを表します。

```
return status = 0
```

システム・プロシージャが正常に実行されない場合、リターン・ステータスは 0 以外の数字になります。

システム・プロシージャ・テーブル

システム・プロシージャは、`master` データベースと `sybsystemdb` データベース内の「システム・プロシージャ・テーブル」を使用して、内部システム値 (たとえば、ステータス・ビット) を人間が判読できるフォーマットに変換します。このようなシステム・プロシージャ・テーブルの1つである `spt_values` は、次のようなさまざまなシステム・プロシージャによって使用されます。

- `sp_configure`
- `sp_dboption`
- `sp_depends`
- `sp_help`
- `sp_helpdb`
- `sp_helpdevice`
- `sp_helpindex`
- `sp_helpkey`
- `sp_helprotect`
- `sp_lock`

`spt_values` テーブルが更新されるのは、システムがアップグレードされるときだけです。それ以外では更新されません。`spt_values` テーブルの使用法を確認するには、`sp_helptext` を実行して、それを参照するシステム・プロシージャのいずれかのテキストを参照してください。

他のシステム・プロシージャ・テーブルには、`spt_monitor` と `spt_committab`、およびカタログ・ストア・プロシージャで必要とするテーブルがあります (`spt_committab` テーブルは、`sybsystemdb` データベースにあります)。

また、テンポラリー・テーブルを作成して削除するシステム・プロシージャもあります。たとえば、`sp_helpdb` は `#spdbdesc` を、`sp_helpdevice` は `#spdevtab` を、`sp_helpindex` は `#spindtab` を作成します。

システム・プロシージャの作成

システム・プロシージャの多くは、このマニュアルのシステム・プロシージャに関する章で説明しています。詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

システム管理者は、任意のデータベースから実行できるシステム・プロシージャを記述することができます。`sybssystemprocs` 内にストアド・プロシージャを作成し、“`sp_`” で始まる名前を付けてください。ストアド・プロシージャの `uid` は 1 (データベース所有者の `uid`) にしてください。

システム管理者が作成するシステム・プロシージャのほとんどは、システム・テーブルを問い合わせるものです。システム・テーブルを変更するストアド・プロシージャを作成することはおすすめしません。

システム・テーブルを変更するストアド・プロシージャを作成するには、まず、システム・セキュリティ担当者が `allow updates to system tables` 設定パラメータをオンにする必要があります。このパラメータがオンに設定されている間に作成されたストアド・プロシージャは、`allow updates to system tables` がオフに設定されても、常にシステム・テーブルを更新できます。システム・テーブルを更新するストアド・プロシージャの作成方法を次に示します。

- 1 `sp_configure` を使用して `allow updates to system tables` をオンに設定します。
- 2 `create procedure` コマンドを使用してストアド・プロシージャを作成します。
- 3 `sp_configure` を使用して `allow updates to system tables` をオフに設定します。

警告！ システム・テーブルを変更する場合は特に注意してください。システム・テーブルを変更するプロシージャは、運用データベースではなく、開発データベースやテスト・データベースでテストしてください。

システム拡張ストアド・プロシージャ

拡張ストアド・プロシージャ (ESP) を利用すると、Adaptive Server から外部言語機能呼び出すことができます。Adaptive Server には定義済みの ESP セットが付属していますが、ユーザが独自の ESP を作成することもできます。システム拡張ストアド・プロシージャの名前はすべて“`xp_`” で始まります。これらは、`sybssystemprocs` データベースにあります。

非常に便利なシステム ESP の 1 つに `xp_cmdshell` があります。これは、Adaptive Server を実行しているシステム上でオペレーティング・システム・コマンドを実行するものです。

システム ESP はシステム・プロシージャとまったく同じように呼び出すことができます。異なる点は、システム ESP は Transact-SQL 文ではなく、手続き型言語コードを実行することです。すべての ESP は、Adaptive Server と同じマシン上で実行される Open Server™ アプリケーションである XP Server™ によって実装されます。XP Server は最初の ESP 実行時に自動的に起動します。

Adaptive Server に付属するシステム ESP の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

システム ESP の作成

`create procedure` を使用して、`sybssystemprocs` データベースにシステム ESP を作成します。システム・プロシージャは自動的に `sybssystemprocs` データベースに組み込まれます。ESP とその手続き型言語関数には、“xp_” で始まる名前を付けてください。ストアド・プロシージャの `uid` は 1 (データベース所有者の `uid`) にしてください。

ESP 作成の一般的な情報については、『Transact-SQL ユーザーズ・ガイド』の「第 17 章 拡張ストアド・プロシージャの使用」を参照してください。

エラー・メッセージのログ

Adaptive Server は、起動されるたびにローカル・エラー・ログ・ファイルに起動情報を書き込みます。新しい Adaptive Server を設定すると、インストール・プログラムが自動的にエラー・ログのロケーションを設定します。エラー・ログのデフォルトのロケーションとファイル名については、使用するプラットフォームの『設定ガイド』を参照してください。

Adaptive Server からのエラー・メッセージの多くはユーザの端末にだけ表示されます。ただし、致命的なエラー・メッセージ (重大度レベル 19 以上)、カーネル・エラー・メッセージ、Adaptive Server からの情報メッセージはエラー・ログ・ファイルに記録されます。

Adaptive Server は、サーバ・プロセスが停止されるまではエラー・ログ・ファイルをオープンした状態に保ちます。古いメッセージを削除してエラー・ログのサイズを減らすには、その前に Adaptive Server プロセスを停止してください。

注意 Windows など一部のプラットフォームでは、Adaptive Server はオペレーティング・システムのイベント・ログにもエラー・メッセージを記録します。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』と『設定ガイド』を参照してください。

Adaptive Server との接続

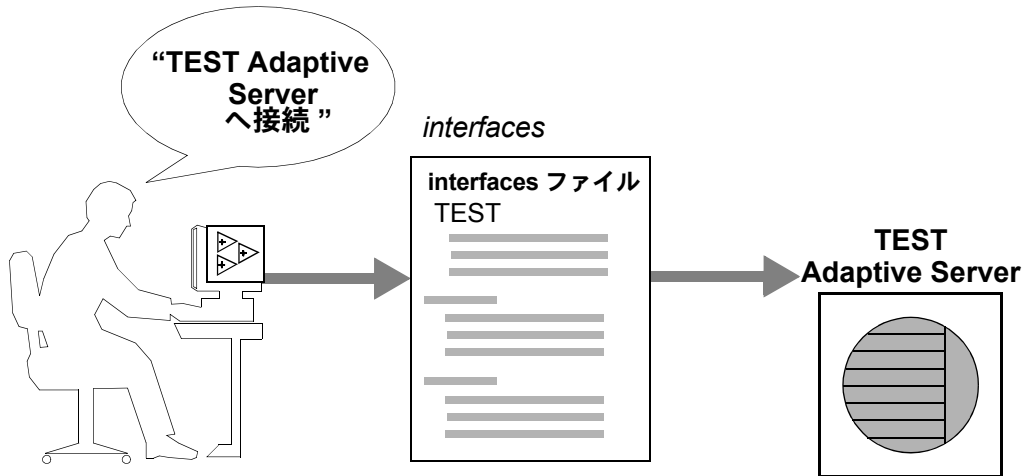
Adaptive Server は、別の Adaptive Server、Open Server アプリケーション、ネットワーク上のクライアント・ソフトウェアと通信できます。クライアントは1つ以上のサーバと通信でき、サーバはリモート・プロシージャ・コールを使用して別のサーバと通信できます。これら対話するには、それぞれがネットワーク上での相手のロケーションを知る必要があります。このネットワーク・サービス情報は *interfaces* ファイルに保管されます。

interfaces ファイル

この *interfaces* ファイルの名前は、オペレーティング・システムによって、*interfaces*、*interface*、または *sql.ini* となります。

interfaces ファイルには認識されているすべてのサーバ名とアドレスがリストされています。クライアント・プログラムを使用してサーバと接続するとき、[図 1-1](#) に示すように、プログラムは *interfaces* ファイル内でサーバ名を探し、そのアドレスを使用してサーバに接続します。

図 1-1: Adaptive Server との接続



interfaces ファイルの名前、ロケーション、内容はオペレーティング・システムによって異なります。また、*interfaces* ファイル内の Adaptive Server アドレスのフォーマットもネットワーク・プロトコルによって異なります。

Adaptive Server のインストール時に、1 つ以上のネットワーク・プロトコルによる Adaptive Server へのローカル接続用に使用できる簡単な *interfaces* ファイルが作成されます。ユーザがネットワークを通して Adaptive Server に接続できるように *interfaces* ファイルを変更してユーザに配布するのは、システム管理者です。*interfaces* ファイルの詳細については、使用しているプラットフォームの『設定ガイド』を参照してください。

interfaces ファイルとネットワーク・リスナの詳細については、『パフォーマンス&チューニング・シリーズ：基本』の「第 2 章 ネットワークとパフォーマンス」を参照してください。

ディレクトリ・サービス

ディレクトリ・サービスは、ネットワーク・サービス情報の作成、修正、取得を管理します。ディレクトリ・サービスは、プラットフォームやサードパーティのベンダによって提供されるものであり、Adaptive Server とは別に購入してインストールする必要があります。ディレクトリ・サービスの例には、レジストリと分散コンピューティング環境 (DCE) があります。

`$$SYBASE/$SYBASE_OCS/config/libtcl.cfg` ファイルは、Sybase が提供する設定ファイルで、サーバとクライアントが次の項目を決定するときに使用します。

- 使用するディレクトリ・サービス
- そのディレクトリ・サービスのドライバの場所

ディレクトリ・サービスがまったくインストールされていない場合や、*libtcl.cfg* ファイルにエントリがまったくない場合は、Adaptive Server は *interfaces* ファイルをデフォルトとして使用して、ネットワーク・サービス情報を取得します。

システム管理者は、操作環境に応じて *libtcl.cfg* ファイルを修正する必要があります。

ディレクトリ・サービスには、プラットフォーム固有のものも、いくつかの異なるプラットフォーム上で使用できるものもあります。プラットフォーム固有のディレクトリ・サービスの設定の詳細については、使用するプラットフォームの『設定ガイド』を参照してください。

ディレクトリ・サービスとしての LDAP

「LDAP」(Lightweight Directory Access Protocol) は、ディレクトリ・サービスへの業界標準のアクセス方法です。ディレクトリ・サービスを使用すると、コンポーネントは LDAP サーバから情報を DN (識別名) で検索できます。LDAP サーバは、企業またはネットワーク上で使用されるサーバ、ユーザ、ソフトウェアの情報を格納したり管理したりします。

LDAP サーバは、Adaptive Server やクライアントを実行しているプラットフォームとは別のプラットフォームに配置できます。LDAP は、クライアントとサーバが交換するメッセージの通信プロトコルと内容を定義します。メッセージとは、読み取り、書き込み、クエリのクライアント要求やサーバの応答など、メタデータ (データに関するデータ) を含むオペレータです。

LDAP サーバに格納され、取得が可能な情報は、次のとおりです。

- Adaptive Server に関する情報 (IP アドレス、ポート番号、ネットワーク・プロトコルなど)
- セキュリティ・メカニズムとフィルタ
- 高可用性コンパニオン・サーバ名
- Adaptive Server にユーザがアクセスするための認証情報

Adaptive Server にログインするユーザを認証するには、*syslogins* ディレクトリに格納されている情報を使用することも、単一のログインとパスワードを企業全体で使用可能にする集中型の LDAP サーバを使用することもできます。『セキュリティ管理ガイド』の「第 3 章 Adaptive Server のログイン・アカウントとデータベース・ユーザの管理」を参照してください。

LDAP サーバの設定時に、次のアクセス制限を指定できます。

- 匿名認証 — すべてのユーザがあらゆる情報にアクセスできます。
- ユーザ名とパスワードによる認証 — Adaptive Server は、次のファイルで指定されているデフォルトのユーザ名とパスワードを使用します。
 - UNIX、32 ビット — `$$SYBASE/$$SYBASE_OCS/config/libtcl.cfg`
 - UNIX、64 ビット — `$$SYBASE/$$SYBASE_OCS/config/libtcl64.cfg`
 - Windows — `%SYBASE%\$SYBASE_OCS%\$ini\libtcl.cfg`

ユーザ名とパスワードによる認証のプロパティによって、LDAP サーバとのセッション接続が確立され、終了します。

注意 *libtcl.cfg* に格納されている、認証目的で LDAP サーバに渡されるデフォルトのユーザ名とパスワードは、Adaptive Server へのアクセスに使用するユーザ名とパスワードとはまったく別のものです。このデフォルトのユーザ名とパスワードは、管理作業を実行するために LDAP サーバにアクセスするためのものです。

LDAP サーバを *libtcl.cfg* ファイルまたは *libtcl64.cfg* ファイル (*libtcl*.cfg* ファイルと総称) で指定する場合は、サーバ情報には LDAP サーバからのみアクセスできます。Adaptive Server は *interfaces* ファイルを無視します。

複数のディレクトリ・サービスが 1 つのサーバでサポートされる場合は、その検索の順序は *libtcl*.cfg* に指定されます。検索順は `dataserver` コマンド・ライン・オプションでは指定できません。

複数のディレクトリ・サービス

LDAP サービスは、どのようなタイプでも (実際のサーバであっても、その他の LDAP サービスへのゲートウェイであっても)、LDAP サーバと呼ばれます。

高可用性を確保するフェールオーバー保護のために、*libtcl*.cfg* ファイルに複数のディレクトリ・サービスを指定できます。リストにあるディレクトリ・サービスのすべてが LDAP サーバである必要はありません。

次の例では、*test:389* への接続が失敗した場合には、指定されたディレクトリ情報ツリー (DIT) ベースを持つ DCE ドライバへのフェールオーバーが発生します。この接続も失敗すると、*huey:11389* 上の LDAP サーバに接続しようとします。DIT ベースのフォーマットはベンダによって異なります。

[DIRECTORY]

```
ldap=libdldap.so ldap://test:389/dc=sybase,dc=com
dce=libddce.so ditbase=../subsys/sybase/dataservers
ldap=libdldap.so ldap://huey:11389/dc=sybase,dc=com
```

詳細については、『Open Client Client-Library/C プログラマーズ・ガイド』と『Open Client Client-Library/C リファレンス・マニュアル』を参照してください。

LDAP ディレクトリ・サービスと Sybase *interfaces* ファイルの違い

LDAP サーバで使用するために、LDAP ドライバでディレクトリ・サービスを実装します。LDAP インフラストラクチャの構成は、次のとおりです。

- 従来の Sybase *interfaces* ファイルに代わる、ネットワーク・ベースのしくみ
- ユーザ、ソフトウェア、リソース、ネットワーク、ファイルなどの情報を階層構造で表した単一のビュー

表 1-1 は、Sybase *interfaces* ファイルと LDAP サーバの違いをまとめたものです。

表 1-1: *interfaces* ファイルと LDAP ディレクトリ・サービスの違い

<i>interfaces</i> ファイル	ディレクトリ・サービス
プラットフォーム固有	プラットフォームに依存しない
Sybase インストール環境ごとに異なった構造	統一された階層構造
マスタ・エントリとクエリ・エントリが別々に存在する	各サーバの 1 つのエントリにクライアントとサーバの両方がアクセスできる
サーバのメタデータを保存できない	サーバのメタデータを保存できる

パフォーマンス

LDAP サーバを使用する場合は、*interfaces* ファイルを使用した場合よりもパフォーマンスが低下することがあります。これは、LDAP サーバの場合、ネットワーク接続を確立してデータを取得する必要があるため、そのために時間を要するからです。この接続は Adaptive Server を起動したときに行われるので、パフォーマンスに違いがある場合はログイン時にわかります。通常のシステム負荷では、パフォーマンスの低下を感じることはありません。特に短い間隔で接続を繰り返す場合など、接続数の増加によってシステム負荷が高まると、LDAP サーバを使用した場合と従来の *interfaces* ファイルを使用した場合とで全体的なパフォーマンスにはっきりとした違いが現れることがあります。

Adaptive Server で使用できるセキュリティ機能

Adaptive Server で使用できる主なセキュリティ機能は以下のとおりです。

- 識別と認証の制御 — 承認されたユーザだけがシステムにログインできるようにする。Adaptive Server は、パスワードベースのログイン認証の他に、Kerberos、LDAP、PAM (Pluggable Authentication Modules) による外部認証もサポートしている。
- 任意アクセス制御 (DAC) — オブジェクトの所有者がオブジェクトへのアクセスを制限できるようにするアクセス制御機能。通常は **grant** コマンドと **revoke** コマンドを使用する。この種の制御は、オブジェクトの所有者が自由に設定できる。
- 役割の分担 — 権限が付与された役割を複数の指定ユーザに割り当てて、指定ユーザだけが特定のタスクを実行できるようにする。Adaptive Server には、システム管理者やシステム・セキュリティ担当者などの「システム標準の役割」と呼ばれる、事前に定義された役割がある。また、システム・セキュリティ担当者が「ユーザ定義の役割」と呼ばれる追加の役割を定義できる。
- 責任範囲 — ログイン、ログアウト、サーバの起動操作、リモート・プロセス・コール、データベース・オブジェクトへのアクセス、特定ユーザによってまたは特定の役割をアクティブにして実行されたすべてのアクションなどのイベントを監査する機能。1つのオプションを設定するだけで、サーバ全体にわたる一連のセキュリティ関連イベントを監査することもできる。
- データの機密保持 — クライアント/サーバ間の通信に Kerberos や SSL (Secure Sockets Layer) による暗号化を使用して、データの機密性を保持する。アクティブでないデータは、パスワードで保護されたデータベース・バックアップによって機密性を保持される。

『セキュリティ管理ガイド』の「第2章 Adaptive Server のセキュリティ管理について」を参照してください。

システム・データベースとオプションのデータベース

この章では、すべての Adaptive Server システムに存在するシステム・データベースについて説明します。また、ユーザがインストール可能な Sybase が提供するオプションのデータベースや、Sybase 製品の保守契約を結んでいるサポート・センタが診断の目的でインストールする sybdiag データベースについても説明します。

トピック名	ページ
システム・データベースの概要	19
master データベース	21
model データベース	23
sybssystemprocs データベース	24
tempdb データベース	24
sybsecurity データベース	26
sybssystemdb データベース	26
sybmgmtdb データベース	26
pubs2 と pubs3 のサンプル・データベース	27
dbccdb データベース	28
sybdiag データベース	28
インストール・スクリプトのバージョンの確認	29

システム・データベースの概要

Adaptive Server をインストールすると、デフォルトで次のシステム・データベースもインストールされます。

- master データベース
- model データベース
- システム・プロシージャ・データベース sybssystemprocs
- 2 フェーズ・コミット・トランザクション・データベース sybssystemdb
- テンポラリ・データベース tempdb

オプションで次のデータベースもインストールできます。

- 監査データベース **sybsecurity**
- サンプル・データベース **pubs2** と **pubs3**
- dbcc データベース **dbccdb**
- Job Scheduler データベース **sybmgmtdb**

master、**model**、**sybssystemprocs**、**tempdb**、**sybmgmtdb** の各データベースのインストールの詳細については、使用するプラットフォームの『インストール・ガイド』を参照してください。**dbccdb** のインストールについては、『システム管理ガイド 第2巻』の「10章 データベースの一貫性の検査」を参照してください。Job Scheduler の使用方法については、『Job Scheduler ユーザーズ・ガイド』を参照してください。

master、**model**、**sybssystemdb**、テンポラリの各データベースは、インストール中に指定したマスタ・デバイス上にすべて常駐します。**master** データベースは全部がマスタ・デバイスに入っていて、他のデバイスに拡張することはありません。その他のデータベースやユーザ・オブジェクトは、すべて他のデバイス上に作成してください。

警告！ ユーザ・データベースをマスタ・デバイスに保管しないでください。保管してしまうと、システム・データベースとマスタ・デバイスに保管されたユーザ・データベースのリカバリが困難になります。

sybsecurity データベースと **sybmgmtdb** データベースは、専用のデバイスとセグメントにインストールします。使用しているプラットフォームの『インストール・ガイド』を参照してください。

sybssystemprocs データベースは、ユーザが選択したデバイスにインストールできます。**pubs2** 用と **pubs3** 用のインストール・スクリプトを変更して、**sybssystemprocs** 用に作成したデバイスを共有できます。

installjsdb スクリプト (*\$SYBASE/ASE-15_0/scripts* にあります) を使用して **sybmgmtdb** データベースをインストールします。*installjsdb* は *sybmgmtdev* という名前のデバイスを探し、このデバイスに **sybmgmtdb** データベースとそのテーブル、ストアド・プロシージャを作成します。**sybmgmtdb** データベースが既に存在する場合、*installjsdb* は既存のデータベースに Job Scheduler テーブルとストアド・プロシージャを作成します。*sybmgmtdev* デバイスも **sybmgmtdb** データベースも見つからない場合、*installjsdb* はマスタ・デバイスに **sybmgmtdb** データベースを作成します。ただし、マスタ・デバイスから **sybmgmtdb** データベースを削除することを強くおすすめします。

installpubs2 スクリプトと *installpubs3* スクリプトは、**create database** 文内ではデバイスを指定しないため、デフォルト・デバイスに作成されます。インストール時には、マスタ・デバイスがデフォルト・デバイスになります。デバイスを変更するには、スクリプトを編集するか、「[第8章 データベース・デバイスの初期化](#)」の指示に従ってください。

master データベース

master データベースは、Adaptive Server のオペレーションを制御し、ユーザ・データベースとそれに関連するデータベース・デバイスについての情報をすべて保管します。表 2-1 は、master データベースに記録される情報を示します。

表 2-1: master データベースに保管される情報

情報	システム・テーブル
ユーザ・アカウント	syslogins
リモート・ユーザ・アカウント	sysremotelogins
このサーバが対話できるリモート・サーバ	syssservers
進行中のプロセス	sysprocesses
設定可能な環境変数	sysconfigures
システム・エラー・メッセージ	sysmessages
Adaptive Server 上のデータベース	sysdatabases
各データベースに割り付けられている記憶領域	sysusages
システムにマウントされたテープとディスク	sysdevices
アクティブ状態のロック	syslocks
文字セット	syscharsets
言語	syslanguages
サーバ全体に適用される役割を持つユーザ	sysloginroles
サーバの役割	sysssrroles
オンラインの Adaptive Server エンジン	sysengines

master データベースは、ユーザ・データベースとデバイスに関する情報を保管するので、`create database`、`alter database`、`disk init`、`disk refit`、`disk reinit`、`ディスク・ミラーリング`の各コマンドは master データベース内から発行する必要があります。

master データベースの最小サイズは、サーバの論理ページ・サイズによって異なります。master データベースは 6656 以上の論理ページを持つため、論理ページごとの最小物理サイズは次のようになります。

- 2K ページ - 13MB
- 4K ページ - 26MB
- 8K ページ - 52MB
- 16K ページ - 104MB

master でのオブジェクト作成の制御

Adaptive Server をインストールした直後は、master データベースにオブジェクトを作成できるのはシステム管理者だけです。システム管理者は、暗黙のうちに、使用するデータベースの所有者“dbo”になるからです。master データベースに作成するオブジェクトは、システム管理のためだけに使用してください。一般ユーザが master にオブジェクトを作成できないように、パーミッションを設定してください。

警告！ master 内にはユーザ・オブジェクトを置かないでください。master 内にユーザ・オブジェクトを置くと、トランザクション・ログがすぐにいっぱいになってしまいます。トランザクション・ログが領域を完全に使い果たしてしまうと、dump transaction コマンドを使用して master 内の領域を解放できなくなります。

alter login を使用してユーザのデフォルト・データベース (ユーザがログイン時に接続するデータベース) を変更する方法もあります。『セキュリティ管理ガイド』の「第 3 章 Adaptive Server のログイン・アカウントとデータベース・ユーザの管理」を参照してください。

システム・プロシージャは master データベースではなく、sysystemprocs データベースに作成します。

master のバックアップとシステム・テーブルのコピー

Adaptive Server 上でのハードウェアやソフトウェアの障害に備えて、次のタスクを行います。

- master データベースとすべてのユーザ・データベースの頻繁なバックアップ。「[master の最新のバックアップの保持](#)」(37 ページ) および『システム管理ガイド 第 2 巻』の「第 13 章 システム・データベースのリストア」を参照してください。
- システム・テーブル sysusages, sysdatabases, sysdevices, sysloginroles, syslogins のコピーの保存 (なるべくオフラインで)。「[システム・テーブルのオフライン・コピーの保存](#)」(38 ページ) を参照してください。これらのスクリプトのコピーが保存されていれば、ハード・ディスクの故障などの障害によってデータベースが使用できなくなった場合でも、『システム管理ガイド 第 2 巻』の「第 13 章 システム・データベースのリストア」で説明している手順を使用してリカバリできます。スクリプトの最新コピーが保存されていない場合は、master データベースが損傷を受けた場合に Adaptive Server のリカバリが非常に難しくなります。

model データベース

Adaptive Server には、model データベースが含まれています。このデータベースは、新しいユーザ・データベース用のテンプレート(プロトタイプ)として使用されます。create database コマンドが実行されるたびに、Adaptive Server は model データベースのコピーを作成して、新しいデータベースのサイズを create database コマンドで指定されたとおりに拡張します。

注意 新しいデータベースは、少なくとも、model データベースと同等の大きさでなければなりません。

model データベースには、それぞれのユーザ・データベースに必要なシステム・テーブルがあります。model を変更することにより、新しく作成されるデータベースの構造をカスタマイズできます。model に対して行った変更は、新しいデータベースにすべて反映されます。システム管理者が通常行う model への変更は次のとおりです。

- ユーザ定義のデータ型、ルール、またはデフォルトの追加。
- Adaptive Server 上のすべてのデータベースにアクセスできるユーザの追加。
- デフォルト権限、特に“guest”アカウントのデフォルト権限の付与。
- select into/bulkcopy/pllsort などのデータベース・オプションの設定。これらの設定は、すべての新しいデータベースに反映されます。model のオプションのデフォルト設定は off です。「第 9 章 データベース・オプションの設定」を参照してください。

通常、model データベースを変更するパーミッションは、ほとんどのユーザには与えられていません。model データベースの内容はすべて Adaptive Server によって新しいユーザ・データベースにコピーされるので、読み込みパーミッションの付与も、あまり意味がありません。

model データベースは、tempdb よりも大きくすることはできません。デフォルトでは、model データベースのサイズは 6 アロケーション・ユニット(1 アロケーション・ユニットは 256 論理ページ)です。tempdb よりも大きくなるように model のサイズを拡張しようとすると、エラー・メッセージが表示されます。

注意 model データベースのバックアップ・コピーを取っておいてください。また、model を変更するたびに dump database を使って model をバックアップします。メディア障害が発生した場合は、ユーザ・データベースの場合と同じ方法で model をリストアします。

sybssystemprocs データベース

Sybase のシステム・プロシージャは、**sybssystemprocs** データベースに保管されています。データベースのユーザがシステム・ストアド・プロシージャ (**sp_** で始まる名前のプロシージャ) を実行すると、Adaptive Server は最初にユーザの現在のデータベース内からそのプロシージャを探します。現在のデータベース内にその名前のプロシージャが存在しない場合は、**sybssystemprocs** 内で探します。**sybssystemprocs** 内にもそのプロシージャがない場合は、**master** 内でそのプロシージャを探します。

プロシージャによってシステム・テーブルが変更される (たとえば、**sp_adduser** によって **sysusers** テーブルが変更される) と、プロシージャを実行したデータベース内でその変更が行われます。

システム・プロシージャのデフォルト・パーミッションを変更するには、**sybssystemprocs** でのパーミッションを変更します。

注意 **sybssystemprocs** を変更する場合は、データベースをバックアップしてください。

tempdb データベース

Adaptive Server には、**tempdb** という「テンポラリ・データベース」があります。テンポラリ・データベースは、テンポラリ・テーブルやその他の一時的な作業に使用される記憶領域です。**tempdb** の領域は、サーバ上の全データベースの全ユーザ間で共有されます。

tempdb のデフォルト・サイズは、サーバの論理ページ・サイズが 2K、4K、8K、16K のいずれであるかによって決まります。一部のアクティビティのために、**tempdb** のサイズを大きくしなければならない場合があります。

- 大規模なテンポラリ・テーブル
- テンポラリ・テーブル上での多数のアクティビティ。これによって、**tempdb** のログがいっぱいになります。
- 大規模な、または同時に行われる多数のソート。サブクエリや **group by** による集約によっても、**tempdb** 内にアクティビティが発生します。

tempdb のサイズを拡張するには、**alter database** コマンドを使います。**tempdb** は、最初はマスタ・デバイス上に作成されます。マスタ・デバイスからでも、その他のデータベース・デバイスからでも **tempdb** に領域を追加できます。

`update index statistics` を大きなテーブルに対して実行するときに、`tempdb` がコマンドを処理するのに十分な大きさでない場合、コマンドは失敗します (エラー番号 1105)。

システム・テンポラリー・データベース (つまり、`tempdb`) の他に、複数のテンポラリー・データベースを作成し、管理することができます。複数のテンポラリー・データベースを使用すると、システム・カタログやシステム `tempdb` のログに対する競合が減少します。

テンポラリー・テーブルの作成

テンポラリー・テーブルを作成するときや、テンポラリー・データベース内の記憶領域が必要となるコマンドを実行するときも、特別なパーミッションは必要ありません。

テンポラリー・テーブルを作成するには、`create table` 文中でテーブル名の前にシャープ記号 (#) を付けるか、または名前のプレフィクス “`tempdb..`” を指定します。

シャープ記号を付けて作成されたテンポラリー・テーブルにアクセスできるのは、Adaptive Server の現在のセッションだけです。その他のセッションのユーザはアクセスできません。このような共有できないテンポラリー・テーブルは、各セッションの終了時に破壊されます。テーブル名の最初の 13 バイト (シャープ記号 (#) も含む) は、ユニークでなければなりません。Adaptive Server は、このようなテーブル名に 17 バイトの数字サフィックスを割り当てます (`tempdb..sysobjects` を問い合わせれば、このサフィックスを参照できます)。

“`tempdb..`” プレフィクスを付けて作成されたテンポラリー・テーブルは、`tempdb` に保管され、Adaptive Server セッション間で共有できます。Adaptive Server は、このようにして作成されたテンポラリー・テーブルの名前を変更しません。このテンポラリー・テーブルは、Adaptive Server が再起動されるか、テーブルの所有者が `drop table` を使用してテーブルを削除するまで存在します。

システム・プロシージャはテンポラリー・テーブル上で機能します。ただし、システム・プロシージャを `tempdb` から使用した場合にかぎります。

ストアド・プロシージャが作成したテンポラリー・テーブルは、プロシージャが終了すると削除されます。セッション終了前に、明示的にテンポラリー・テーブルを削除することもできます。

警告！ 他のユーザやセッションとの間でテーブルを共有する場合以外は、ストアド・プロシージャ内から “`tempdb..`” プレフィクスを付けたテンポラリー・テーブルを作成しないでください。

Adaptive Server の再起動のたびに `model` が `tempdb` にコピーされ、これによってデータベースがクリアされます。テンポラリー・テーブルはリカバリできません。

sybsecurity データベース

Adaptive Server の監査のシステムを格納している sybsecurity データベースには次のものが含まれます。

- 監査証跡が保存されるシステム・テーブル `sysaudits_01`、`sysaudits_02`、... `sysaudits_08`
- グローバルな監査オプションを記述したローが保存されている `sysauditoptions` テーブル
- `model` から抽出された、その他すべてのデフォルト・システム・テーブル『セキュリティ管理ガイド』の「第7章 監査」を参照してください。

sybsystemdb データベース

sybsystemdb データベースは、分散トランザクションに関する情報を格納しません。Adaptive Server バージョン 12.0 以降では、リモート・プロシージャ・コール (RPC) またはコンポーネント統合システム (CIS) を使用してリモート・サーバにトランザクションを送信するためのトランザクション・コーディネーション・サービスを実行することができます。分散トランザクションに参加しているリモート・サーバに関する情報は、`syscoordinations` テーブルに格納されます。

sybsystemdb データベースには、Sybase 2 フェーズ・コミット・プロトコルを使用する SYB2PC トランザクションに関する情報も格納されています。それぞれの 2 フェーズ・コミット・トランザクションに関する情報を格納し、その完了ステータスを追跡する `spt_committab` テーブルは、sybsystemdb データベースに格納されています。

2 フェーズ・コミット・トランザクションの情報と、sybsystemdb データベースの作成方法については、使用するプラットフォームの『設定ガイド』を参照してください。

sybmgmtadb データベース

ジョブ、スケジュール、スケジュール・ジョブ情報、Job Scheduler タスクで内部処理のために必要なデータは、sybmgmtadb データベースに格納されます。また、実行したそれらのタスクの結果と出力も、sybmgmtadb データベースに格納されます。『Job Scheduler ユーザーズ・ガイド』を参照してください。

pubs2 と pubs3 のサンプル・データベース

サンプル・データベース pubs2 と pubs3 のインストールは任意です。これらのデータベースは、Adaptive Server の学習ツールとして用意されています。Adaptive Server のマニュアルに記載されている例のほとんど (pubs3 データベースを使用していることが明記されている例は除く) で、pubs2 サンプル・データベースが使用されています。

サンプル・データベースは訓練用としてのみ提供されています。Adaptive Server の運用環境にはインストールしないでください。

pubs2 と pubs3 のインストール方法については、使用するプラットフォームの『インストール・ガイド』を参照してください。サンプル・データベースの内容については、『Transact-SQL ユーザーズ・ガイド』を参照してください。

サンプル・データベースの管理

サンプル・データベースには“guest”ユーザ・ログインが登録されているので、認可された Adaptive Server のユーザであれば誰でも“guest”ユーザとしてそのデータベースにアクセスできます。pubs2 と pubs3 では、ユーザ・テーブルを選択 (select)、挿入 (insert)、更新 (update)、削除 (delete) するためのパーミッションなど、幅広い権限が“guest”ログインに与えられています。『セキュリティ管理ガイド』の「第3章 Adaptive Server のログイン・アカウントとデータベース・ユーザの管理」を参照してください。

pubs2 データベースと pubs3 データベースのサイズは、サーバの論理ページ・サイズが 2 K、4 K、8 K、16K のいずれかであるかによって決まります。可能であれば、新しいユーザには pubs2 と pubs3 の変更を加えていないコピーを提供してください。そうすることにより、新しいユーザが他のユーザの加えた変更戸惑うことがなくなります。特定のデータベース・デバイス上に pubs2 と pubs3 を置く場合は、インストール・スクリプトを編集してからデータベースをインストールしてください。

空き領域の問題がある場合は、begin transaction コマンドを発行してからサンプル・データベースを更新するように、ユーザに指示を与えてください。こうすると、サンプル・データベースの更新が終わった後で、rollback transaction コマンドを発行して変更を元に戻すことができます。

pubs2 image データ

Adaptive Server には、pubs2 データベースに image データをインストールするためのスクリプトがあります (pubs3 では image データを使用しません)。image データは 6 つのピクチャで構成され、PICT、TIF、Sun raster の各ファイル・フォーマットが 2 つずつあります。Sybase は、image データを表示するためのツールを提供していません。イメージを表示するには、データベースから image データを抽出した後で、適切なスクリーン・グラフィック・ツールを使用してください。

image データを pubs2 にインストールする方法については、使用するプラットフォームの『インストール・ガイド』を参照してください。

dbccdb データベース

dbcc checkstorage を実行すると、「ターゲット・データベース」の設定情報、オペレーション・アクティビティ、そのオペレーションの結果が dbccdb データベースに記録されます。このデータベースには、dbccdb の作成と管理を行ったり、dbcc checkstorage オペレーションの結果についてのレポートを生成したりする dbcc スタアド・プロシージャが格納されます。『システム管理ガイド 第 2 巻』の「第 10 章 データベースの一貫性の検査」を参照してください。

sybdiag データベース

Sybase 製品の保守契約を結んでいるサポート・センタは、デバッグのために、ご使用のシステム上に sybdiagdb データベースを作成することがあります。このデータベースには診断設定データが保持されていて、顧客が使用することはできません。

インストール・スクリプトのバージョンの確認

`sp_version` を使用して、Adaptive Server にインストールされているスクリプト (`installmaster`、`installdbccdb` など) の現在のバージョン、それらのスクリプトが正常に実行されたかどうか、および、実行に要した時間を確認できます。

`sp_version` の構文は次のとおりです。

```
sp_version [script_file [, "all"]]
```

各要素の意味は次のとおりです。

- `script_file` は、インストール・スクリプトの名前 (デフォルト値は NULL) です。
- `all` はスクリプトについての詳細な情報 (実行された日付や実行に要した時間など) をレポートします。

`sp_version` をパラメータなしで実行すると、スクリプトの全情報をレポートします。

次の例では、実行されたインストール・スクリプト、それらのインストール・スクリプトの実行時刻、終了時刻についてレポートします。

```
sp_version null, 'all'
Script      Version
Status
-----
installmaster 15.0/EBF XXXXX/B/Sun_svr4/OS 5.8/asemain/1/32-bit/OPT/Thu Sep 23
22:12:12 2004
Complete [Started=Sep 24 2004 3:39PM]-[Completed=Sep 24 2004 3:45PM]
```


この章では、次のことについて説明します。

- 新しいシステム管理者に対する重要なトピックの説明
- システム管理者向けの Sybase のマニュアル内の情報の参照先

この章は、経験豊富なシステム管理者にとっても、継続的な管理アクティビティの整理に役立ちます。

トピック名	ページ
論理ページ・サイズ	31
テスト・サーバの使用方法	32
Sybase 製品のインストール時の考慮事項	33
物理リソースの割り付け	34
バックアップとリカバリ	37
継続して実行する管理作業とトラブルシューティング	40
記録の保管	41
その他のリソース	43

論理ページ・サイズ

データベース・オブジェクトは、論理ページを使用して構築されます。データベースとそれに関連する任意のオブジェクトは、同じ論理ページ・サイズを使用します。つまり、複数の論理ページ・サイズを使用するサーバを作成することはできません。Adaptive Server では、マスタ・デバイスと master データベースを作成するときに論理ページ・サイズを 2K、4K、8K、または 16K とすることができます。ただし、1 台のサーバ・インストール環境で使用できるのは、この 4 種類の論理ページ・サイズのうちの 1 つだけです。

サーバ内のすべてのデータベース、および、各データベースにあるすべてのオブジェクトに、同じ論理ページ・サイズが使用されます。たとえば、サーバの論理ページ・サイズが 4K の場合は、ページによっては最初の 2K を超える部分を使用しないことがあるとしても、すべてのページが 4K でなければなりません。

`dataserver -z` を使用してマスタ・デバイスを作成するときに、ページ・サイズを選択します。

`dataserver` コマンド (マスタ・デバイスの作成に使用されるコマンド) の詳細については、『ユーティリティ・ガイド』を参照してください。論理ページ・サイズの詳細については、『システム管理ガイド 第2巻』の「第3章 メモリの設定」を参照してください。

テスト・サーバの使用方法

Sybase では、テスト用または開発用の Adaptive Server をインストールして使用し、サーバ管理の経験を積んでからそのサーバを削除し、実際の運用サーバを作成することをおすすめします。このようにしてテスト・サーバを使用すれば、さまざまな設定のプランとテストが簡単にでき、間違えた場合でもリカバリの苦勞がほとんどありません。実際の運用サーバの再起動や運用データベースの作り直しの必要がなければ、新しい機能のインストールと管理の方法も、ずっと学びやすくなります。

テスト・サーバを使用する場合は、Adaptive Server のインストールまたはアップグレードから始まるサーバ設定作業全体でテスト・サーバを使用することをおすすめします。最終的な運用システムに関する非常に重要な決定はこの手順の中で行われます。次の項では、テスト・サーバがシステム管理者にとってどのように役立つかを説明します。

リソースの計画

テスト・サーバを使用することによって、システムに必要とされる最終的なリソース要件の計画を立てることができ、予想していなかったリソースの不足を発見できます。

特にディスク・リソースは運用システムの最終的な設計に劇的な影響を及ぼすことがあります。たとえば、あるデータベースではメディア障害が発生した場合にノンストップ・リカバリが必要であると決定したとします。このような状況では、重要なデータベースをミラーリングするのに追加データベース・デバイスを1つ以上設定する必要があります。テスト・サーバでこのようなりソース要件を発見すれば、データベースの利用者に影響を与えることなく、データベースとテーブルの物理的なレイアウトを変更できます。

テスト・サーバを使用すると、異なるハードウェア設定を使用して Adaptive Server とユーザ・アプリケーションのベンチマーク・テストを行うこともできます。この場合は、Adaptive Server レベルとオペレーティング・システム・レベルの両方で物理リソースの最適な設定を決定してから、システム全体をオンラインにして通常使用を開始します。

パフォーマンスの目標の達成

パフォーマンスの目標のほとんどは、データベースの設計と設定を慎重に計画しなければ達成できません。たとえば、特定のテーブルの挿入処理と I/O パフォーマンスがボトルネックであることを発見したとします。この場合、そのテーブルを専用セグメント上に作成し直してテーブルを分割するのが最善の策と考えられます。しかし、そのような変更は運用システムにとっては混乱のもとであり、設定パラメータの変更だけであっても Adaptive Server を再起動することになります。

Sybase 製品のインストール時の考慮事項

Adaptive Server と他の Sybase 製品のインストールを行うときに、システム管理者が担当責任者になることがあります。インストールを担当する場合は、次の指標を使用して処理に役立ててください。

製品の互換性のチェック

新しい製品をインストールするときや既存の製品をアップグレードするときには、その前に、製品に添付されている『リリース・ノート』を読み、システムに影響する互換性の問題について理解してください。互換性の問題は、ハードウェアとソフトウェア間、同じソフトウェアの異なるリリース・レベル間で発生する可能性があります。前もって『リリース・ノート』を読んでおくことによって、互換性に関する既知の問題の解決に費やす時間を節約し、無用な推測を避けることができます。リリース・ノートに記載されている既知の問題には、特に注意してください。

Adaptive Server のインストールまたはアップグレード

新規インストールやアップグレードを始める前に、使用するプラットフォームの『インストール・ガイド』全体に目を通してください。Adaptive Server の稼働に必要なオペレーティング・システムの条件について検討するには、オペレーティング・システムの管理者に相談することも役立ちます。この稼働条件には、使用するプラットフォームに応じて、メモリ、ロー・デバイス、非同期 I/O、その他の機能の設定が含まれます。このタスクの多くは、インストールを開始する前に実行する必要があります。

サーバをアップグレードする場合は、始める前に必ず master データベース、ユーザ・データベース、トリガ、システム・プロシージャをはじめとするすべてのデータをオフラインでバックアップします。特に古いバージョンと新しいバージョンの間でダンプ・ファイルの互換性がない場合は、アップグレード後すぐにデータの完全なバックアップを別に作成します。

追加のサードパーティ・ソフトウェアのインストール

Adaptive Server は基本的に、各ハードウェア・プラットフォームで一般的なネットワーク・プロトコルをサポートしています。ネットワークが別のプロトコルもサポートしている場合は、必要なプロトコル・サポートをインストールしてください。

Sybase の *interfaces* ファイルの代わりに、ディレクトリ・サービスを使用してサーバのアドレスや他のネットワークの情報を入手できます。ディレクトリ・サービスは、プラットフォームやサードパーティのベンダによって提供されるものであり、Adaptive Server のインストールとは別に購入してインストールする必要があります。Adaptive Server によって現在サポートされているディレクトリ・サービスのリストについては、使用するプラットフォームの「[ディレクトリ・サービス](#)」(15 ページ)と『Adaptive Server Enterprise 設定ガイド』を参照してください。

クライアント接続の設定とテスト

クライアントが正しく接続できるかどうかは、Adaptive Server、クライアント・ソフトウェア、ネットワーク製品の組み合わせに依存します。Adaptive Server とともにインストールされるネットワーク・プロトコルを使用する場合のネットワーク接続のテスト方法については、プラットフォームの『Adaptive Server Enterprise 設定ガイド』を参照してください。他のネットワーク・プロトコルを使用する場合は、そのネットワーク製品に添付されている資料を参照してください。Adaptive Server とクライアントとの接続をテストするには、Sybase のコネクティビティ製品付属の“ping”ユーティリティを使用することもできます。クライアントが Adaptive Server に接続する方法の概要については、「[Adaptive Server との接続](#)」(14 ページ)を参照してください。*interfaces* ファイルの名前と内容の詳細については、プラットフォームの『Adaptive Server Enterprise 設定ガイド』を参照してください。

物理リソースの割り付け

物理リソースの割り付けとは、パフォーマンスとリカバリの目標を達成するために必要なメモリ、ディスク領域、ワーカー・プロセス、CPU パワーを Adaptive Server に提供することです。新しくサーバをインストールするときに、システム管理者はリソースの使用方法について決定する必要があります。プラットフォームをアップグレードする場合、またはデータベース・システムの設計を変更する場合は、後からメモリ、ディスク・コントローラ、または CPU を追加することによって Adaptive Server のリソースを再割り付けする必要もあります。Adaptive Server とユーザ・アプリケーションのベンチマーク・テストを早めに行えば、パフォーマンスのボトルネックになるハードウェア・リソースの不足を特定するのに役立ちます。

Adaptive Server が必要とするディスク リソースの種類を理解するには、『システム管理ガイド 第2巻』の「第16章 ディスク・リソースの概要」を参照してください。メモリとCPUのリソースについては、『システム管理ガイド 第2巻』の「第3章 メモリの設定」と「第5章 マルチプロセッサ・サーバの管理」を参照してください。

次の項では、物理リソース要件を調べるために役立つ指標について説明します。

専用サーバと共有サーバ

Adaptive Server のリソース計画作成の最初の手順は、同じマシン上で稼働する他のアプリケーションが必要とするリソースを確認することです。通常は、Adaptive Server 専用のマシンを用意します。専用とは、オペレーティング・システムとネットワーク・ソフトウェアが使用する分を除いたリソースを Adaptive Server が自由に使用できるということです。共有システムの場合は、Adaptive Server のクライアント・プログラムやプリント・サーバといった他のアプリケーションが、Adaptive Server と同じマシン上で稼働します。アプリケーションのタイプと使用パターンは時間の経過とともに変化する可能性があるため、共有システム上で Adaptive Server が使用できるリソースを計算するのは困難です。

Adaptive Server 用のリソースを設定するときに、オペレーティング・システム、クライアント・プログラム、ウィンドウ・システムなどで使用されるリソースについて考慮するのはシステム管理者の責任です。使用可能なリソースだけを使用するように Adaptive Server を設定してください。このようにしないと、サーバのパフォーマンスが低下することや、起動できなくなることがあります。

意思決定支援処理と OLTP アプリケーション

Adaptive Server には、OLTP や意思決定支援処理が行われる環境や負荷が一様でない環境でのパフォーマンスを最適化するための多くの機能があります。ただし、このような機能を最大限に活用するには、あらかじめシステム内のアプリケーションの稼働条件を決定します。

負荷が一様でないシステムの場合は、アプリケーションのタイプごとに最も多く使用すると予想される個々のテーブルのリストを作成しておきます。このリストは、アプリケーションにとって最高のパフォーマンスを達成するために役立ちます。

リソースの使用計画

リソースの使用方法について事前に理解し、計画を立てることは非常に大切です。たとえば、ディスク・リソースの場合、Adaptive Server 用にデバイスを初期化して割り付けた後は、Adaptive Server のデータでそのデバイスを使い切ることがないとわかっていても、そのデバイスを他の目的に使用することはできません。同様に、Adaptive Server は設定されたメモリを自動的に予約しますが、このメモリを他のアプリケーションが使用することはできません。

リソースの使用を計画する際の考慮事項

- リカバリのためには、必ず、データベースのトランザクション・ログをデータとは別の物理デバイスに保管してください。『システム管理ガイド 第2巻』の「第6章 ユーザ・データベースの作成と管理」を参照してください。
- ミッション・クリティカルなデータを保管するデバイスをミラーリングします。『システム管理ガイド 第2巻』の「第2章 データベース・デバイスのミラーリング」を参照してください。オペレーティング・システムがディスク・アレイとディスク・ミラーリングをサポートしている場合は、Adaptive Server のデータに対してこれらの機能を使用することも検討します。
- テスト用の Adaptive Server を使用している場合は、データベース・デバイスをロー・デバイスではなくオペレーティング・システム・ファイルとして初期化の方が簡単なことがあります。Adaptive Server のデバイスには、ロー・パーティションと動作確認済みのファイル・システムのどちらも使用できます。
- 設定オプションの変更は、Adaptive Server が物理リソース (特にメモリ) を消費する方法に影響する可能性があります。それぞれのパラメータが使用するメモリの量については、「[第5章 設定パラメータ](#)」を参照してください。

オペレーティング・システムの設定

Adaptive Server で使用可能なリソースと必要なリソースが確定したら、オペレーティング・システム・レベルで次の物理リソースの設定を行います。

- ロー・パーティションを使用する場合は、Adaptive Server が必要とするサイズにロー・デバイスを初期化します。Adaptive Server 用に初期化したロー・デバイスを、オペレーティング・システム・ファイルの保管などの他の目的で使用することはできません。ロー・デバイスを必要なサイズに初期化して設定するときは、オペレーティング・システム管理者に相談してください。
- ネットワーク接続数を設定します。Adaptive Server が稼働するマシンが、設定した数の接続を実際にサポートできることを確認してください。使用するオペレーティング・システム用のマニュアルを参照してください。

- 使用するオペレーティング・システムとアプリケーションの設定がさらに必要な場合があります。使用しているプラットフォームの『ASE インストール・ガイド』を参照してください。また、アプリケーションを実行するためのオペレーティング・システムの条件については、クライアント・ソフトウェアのマニュアルを参照するか、エンジニアに相談してください。

バックアップとリカバリ

データベースを定期的にバックアップするのは、データベース・システムの整合性を保つために重要なことです。Adaptive Server は、システムのクラッシュ(停電による停止など)、またはサーバの障害からは自動的にリカバリを行いますが、メディア障害によって生じるデータの消失からのリカバリができるのは「システム管理者」だけです。

『システム管理ガイド 第2巻』の以下の章では、バックアップとリカバリ計画に関する作成と実行について説明します。

- 「第11章 バックアップおよびリカバリ・プランの作成」
- 「第12章 ユーザ・データベースのバックアップとリストア」
- 「第13章 システム・データベースのリストア」
- 「第16章 スレッシュホールドによる空き領域の管理」

master の最新のバックアップの保持

master データベースのバックアップの作成は、バックアップとリカバリの計画において最も重要な要素です。**master** データベースには、データベース・システム全体の構造についての詳細な情報が格納されています。**master database** には、Adaptive Server データベース、デバイス、データベースを構成するデバイス・フラグメントの情報が保存されています。Adaptive Server のリカバリ時にこの情報が必要となるので、常に **master** データベースの最新のバックアップ・コピーを保持することがきわめて重要です。

master データベースのバックアップを常に最新の状態に保つには、ディスク、記憶領域、データベース、またはセグメントに影響するコマンドや次のような手順を実行するたびにデータベースをバックアップします。

- データベースの作成または削除
- 新しいデータベース・デバイスの初期化
- 新しいダンプ・デバイスの追加
- デバイス・ミラーリングに関するコマンドの使用

- **master** データベースに保管されているシステム・ストア・プロシージャの作成または削除
- セグメントの作成、削除、変更
- 新しい Adaptive Server ログインの追加

master をテープ・デバイスにバックアップするには、**isql** を起動して次のコマンドを入力します。

```
dump database master to "tape_device"
```

ここで、*tape_device* はテープ・デバイスの名前です (たとえば */dev/rmt0*)。

システム・テーブルのオフライン・コピーの保存

master の定期的なバックアップに加えて、**sysdatabases**、**sysdevices**、**sysusages**、**sysloginroles**、**syslogins** の各システム・テーブルのオフライン・コピーを保存してください。これは、『ASE ユーティリティ・ガイド』で説明している **bcp** ユーティリティを使用し、それぞれのシステム・テーブルの内容のハードコピーを保管することによって行います。次の出力を印刷してハードコピーを作成します。

```
select * from sysusages order by vstart
select * from sysdatabases
select * from sysdevices
select * from sysloginroles
select * from syslogins
```

これらのテーブルのコピーが保存されていれば、ハード・ディスクの故障などの障害によってデータベースが使用できなくなった場合でも、『システム管理ガイド 第2巻』の「第13章 システム・データベースのリストア」で説明している手順を使用してリカバリできます。

「[記録の保管](#)」(41 ページ) で説明しているように、ユーザ・オブジェクトのデータ定義言語 (DDL) スクリプトのコピーも保存しておいてください。

バックアップ手順の自動化

自動化したバックアップ手順を作成すると、処理を簡単かつすばやく実行できます。バックアップの自動化は、必要なバックアップ・コマンドを実行するためのオペレーティング・システムのスクリプトまたはユーティリティ (UNIX の **cron** ユーティリティなど) と同様に簡単に使用できます。スレッシュホールドを使用してさらに手順を自動化することもできます。これについては、『システム管理ガイド 第2巻』の「第16章 スレッシュホールドによる空き領域の管理」を参照してください。

❖ 自動バックアップ手順の作成

自動化スクリプトの作成に必要なコマンドは使用しているオペレーティング・システムによって異なりますが、スクリプトで実行する基本的な手順は同じです。

- 1 isql を起動して、たとえばテンポラリ・ファイルのような保管領域にトランザクション・ログをダンプします。
- 2 ダンプ・ファイル名にダンプの日付、時刻、データベース名が含まれるように名前を変更します。
- 3 履歴ファイル内に新しいバックアップに関する情報を記録します。
- 4 ダンプ中に発生したエラーを別のエラー・ファイルに記録します。
- 5 エラーが発生した場合は、システム管理者に自動的にメールを送ります。

データベースをバックアップする前のデータの一貫性の確認

データベースをバックアップする際は、データに一貫性のある正確なバックアップを作成する必要があります。これは **master** データベースについては特に必要なことです。内部的にエラーがあるデータベースをバックアップすると、それをリストアしたデータベースでも同じエラーが発生します。

バックアップを実行する前にデータベースにエラーがあるかどうかをチェックするには、**dbcc** コマンドを使用します。ダンプの前には必ず **dbcc** を使用してデータベースの整合性を検証してください。**dbcc** によってエラーが検出された場合は、エラーを修正してからデータベースをダンプします。

時間の経過とともに、**dbcc** を実行していてほとんどエラーが検出されなかった場合は、データベース破壊の危険性が少ないと考えて、**dbcc** を実行する回数を減らすことができます。データ消失によって受ける影響が大きい場合は、引き続きデータベースのバックアップのたびに **dbcc** コマンドを実行します。

注意 パフォーマンスを考慮して、通常は **dbcc** のチェックをピーク時を避けて行うか、別のサーバで行います。

『システム管理ガイド 第2巻』の「第10章 データベースの一貫性の検査」を参照してください。

ログ・サイズのモニタ

トランザクション・ログの空きがほとんどないときは、トランザクションをダンプするという標準の方法では領域を再利用できなくなる場合があります。システム管理者はログ・サイズをモニタし、定期的にトランザクション・ログのダンプ (通常のデータベース・ダンプの他に) を実行して、そのような事態を回避してください。スレッシュホールド・ストアード・プロシージャを設定して、ログが所定の容量に達した場合にシステム管理者が通知を受ける (または、ログをダンプする) ようにします。『システム管理ガイド 第2巻』の「第16章 スレッシュホールドによる空き領域の管理」を参照してください。Sybase では、データベースのダンプとロードの時間を短縮するために、データベースの完全ダンプを行う直前にトランザクション・ログのダンプを行うこともおすすめします。

`sp_helpsegment` を使用して、ログ・セグメント内の領域の使用状況を手作業でモニタできます。詳細については、『システム管理ガイド 第2巻』の「第8章 セグメントの作成と使用」を参照してください。

継続して実行する管理作業とトラブルシューティング

この項では、スケジュール化した定期的なバックアップに加えて、Adaptive Server が使用されている間にシステム管理者が実行する管理作業について説明します。

Adaptive Server の起動と停止

サーバ・マシンの起動と同時に Adaptive Server の起動が行われるように、多くのシステム管理者はこの手順を自動化しています。このように自動化するには、オペレーティング・システムの起動スクリプトを編集するか、オペレーティング・システムの他の手順を使用して行います。Adaptive Server の起動と停止の方法については、使用するプラットフォームの『Adaptive Server Enterprise 設定ガイド』を参照してください。

エラー・ログの表示と削除

エラー・ログの内容を定期的に調べて、重大なエラーが発生していないかどうかを確認してください。オペレーティング・システムのスクリプトを使用して、特定のメッセージを探すためにエラー・ログをスキャンできます。また、特定のエラーが発生したときにシステム管理者に自動的に通知できます。エラー・ログを定期的に調べると、継続的に発生する同質の問題があるか、または特定のデータベース・デバイスに障害が発生しやすくなっていないかどうかを調べることができます。エラー・メッセージとその重大度レベルについては、「第12章 システムの問題の診断」を参照してください。

Adaptive Server を起動するたびに情報メッセージとステータス・メッセージがエラー・ログ・ファイルに追加されるので、時間がたつにつれてエラー・ログ・ファイルのサイズが大きくなります。定期的にログ・ファイルを開いて古い記録を削除することによって、ログ・ファイルを「小さく」することができます。ログ・ファイルを管理しやすいサイズに保つようにすれば、ディスク領域の節約につながりエラーの場所も見つけやすくなります。

記録の保管

Adaptive Server システムについての記録を保管することは、システム管理者の重要な作業のひとつです。変更した内容と発生した問題についての正確な記録は、Sybase 製品の保守契約を結んでいるサポート・センタに連絡する場合や、データベースをリカバリする場合に貴重な参考資料になります。また、システム管理者が不在の間、Adaptive Server システムを管理する担当者のために重要な情報を提供できます。

連絡先の情報

システム・セキュリティ担当者、オペレータ、システム上のデータベース所有者だけでなく、システム管理者の連絡先情報のリストも保管してください。それぞれの役割の副担当者の連絡先も記録しておいてください。問題のレポートと対策の要求が適切な担当者へ届くように、Adaptive Server のすべてのユーザにこの情報を伝えます。

設定情報

データベースとデータベース・オブジェクトの作成、および Adaptive Server の設定にはスクリプト・ファイルを使用するようにし、このスクリプト・ファイルを安全な場所に保管するというのが理想的な方法です。スクリプト・ファイルを保管しておく、システム障害の場合にもシステム全体を作成し直すことができます。また、スクリプト・ファイルを使用すると、新しいハードウェア・プラットフォームで評価用のデータベース・システムを迅速に再作成できます。サードパーティ・ツールを使用してシステム管理を行っている場合は、管理作業を実行した後で同じスクリプトを生成してください。

次のような情報を記録してください。

- データベースとデータベース・オブジェクトを作成するために使用するコマンド (DDL スクリプト)
- Adaptive Server の新しいログインとデータベース・ユーザを追加するコマンド

- 現在の Adaptive Server の設定ファイル (「[設定ファイルを指定して sp_configure を使用する方法](#)」(63 ページ) を参照)
- データベース・デバイスとして初期化されたすべてのファイルとロー・デバイスの名前、ロケーション、サイズ

Adaptive Server の設定に対するすべての変更について、日付入りのログを保持してください。結果の要約だけでなく、変更の理由と日時を示す簡単な説明をそれぞれの変更に付記してください。

管理作業のスケジュール

スケジュール化した定期的な管理作業用のカレンダーを作成し、自分のサイトで実行する手順をそのカレンダーに記入します。たとえば、次のようなものがあります。

- dbcc を使用して行うデータベースの一貫性チェック
- ユーザ・データベースとシステム・データベースのバックアップ
- トランザクション・ログの空き領域のモニタ (自動的に行われない場合)
- トランザクション・ログのダンプ
- Adaptive Server、Backup Server、Adaptive Server Monitor のエラー・ログの内容の調査
- update statistics コマンドの実行 (『パフォーマンス&チューニング・シリーズ: 統計的分析によるパフォーマンスの向上』の「第 1 章 set statistics コマンドの使用」を参照)
- 監査情報の調査 (監査オプションをインストールしている場合)
- ストアド・プロシージャの再コンパイル
- サーバ・マシンのリソース使用状況のモニタ

システム情報

Adaptive Server を実行するハードウェアとオペレーティング・システムに関する次の情報も記録してください。

- オペレーティング・システムの設定ファイルまたは起動ファイルのコピー
- ネットワーク設定ファイル (たとえば、hosts ファイルと services ファイル) のコピー

- Adaptive Server の実行ファイルとデータベース・デバイスの名前とパーミッション
- バックアップに使用するテープ・デバイスの名前とロケーション
- 自動バックアップ、Adaptive Server の起動、その他のシステム管理アクティビティを行うためのオペレーティング・システム・スクリプトまたはプログラムのコピー

災害時のリカバリ計画

基本的なバックアップとリカバリの手順、「バックアップとリカバリ」(37 ページ)のガイドライン、データのリカバリについての自分自身の経験をまとめて、システムに合ったリカバリ手順を簡単なリスト形式で作成してください。このリストは、自分にとっても、緊急時に運用システムをリカバリする必要がある他のシステム管理者にも役立ちます。

その他のリソース

システム管理者が覚える情報は非常に多くありますが、基本的な管理作業の習得と簡素化に役立つソフトウェア・ツールがいくつか提供されています。これらのソフトウェア・ツールには、サーバのパフォーマンスやその他のアクティビティをモニタする Adaptive Server Monitor や、多くの管理作業を簡単に実行できる Sybase Central などがあります。また、システム管理者が行う日常の管理作業を支援するサードパーティ製のソフトウェア・パッケージもあります。

トピック名	ページ
Sybase Control Center for Adaptive Server	45
Adaptive Server Sybase Central プラグイン	46

この章では Adaptive Server の GUI 管理ツールとモニタリング・ツールについて説明します。ここではこれらのツールの概要を説明することを目的としています。完全な説明はオンライン・ヘルプを参照してください。

Sybase Control Center for Adaptive Server

Adaptive Server® の Sybase Control Center は Adaptive Server のステータスと可用性の管理とモニタリングに適したツールです。

Sybase Control Center はモニタリングと管理用の Web ベースのツールです。Sybase Control Center のダウンロードとインストールの詳細については、[インストール・ガイド](#) (<http://sybooks.sybase.com/nav/summary.do?prod=10680>) を参照してください。

Sybase Control Center の単一インスタンスを使用して、複数の Adaptive Server をモニタリングし管理することができます。

Sybase Control Center は、Adaptive Server Enterprise バージョン 15.0.2 以降をサポートしています。サポートされている機能のリストについては、[Sybase Control Center のマニュアル](#)を参照してください。

Sybase Control Center のクライアント/サーバのアーキテクチャにより、1 台または複数の Sybase Control Center サーバ経由で複数のクライアントが企業内のすべての Adaptive Server をモニタリングできます。Adaptive Server 用の Sybase Control Center により、モニタリング機能、履歴パフォーマンス・モニタリング機能、および管理機能を、その他の Sybase 製品の管理モジュールと統合された拡張可能な Web アプリケーションの形態で使用できるようになります。さまざまな場所にある異機種リソースの共有統合管理、リアルタイムで可用性とパフォーマンスのステータス・ベースおよびスレッショルド・ベースの通知を行う電子メール・アラート、およびパフォーマンスと使用量の傾向を特定するインテリジェント・ツールを提供します。これらのツールはすべて、シンクライアントなリッチ・インターネット・アプリケーション (RIA) を介して Web ブラウザから配布されます。

Adaptive Server の Sybase Control Center を使用して統計値を収集します。この統計値の時間的な傾向から、使用量のパターンおよびサーバ上でのデータベース、デバイス、キャッシュおよびプロセスの動作に関する強力な洞察を行うことができます。収集したデータはテーブルまたはグラフで表示できます。結果を任意の時間の尺度 (分単位から年単位まで) でプロットすることにより、特定のものの全体を表示したり一部を詳細に調べたりできます。サーバの過去の実行内容の詳細を知ることにより、Adaptive Server が将来のニーズに合致するようになります。

Sybase Control Center の使用に関する詳細は、オンライン・ヘルプ (<http://sybooks.sybase.com/nav/summary.do?prod=10680>) を参照してください。

Adaptive Server Sybase Central プラグイン

Sybase Central は、グラフィカル・ユーザ・インタフェース (GUI) 管理ツールです。Sybase Central では、特定の Sybase 製品を管理する各種「プラグイン」を使用できます。Adaptive Server プラグインを使用して Adaptive Server を管理することにより、Transact-SQL コマンドやシステム・ストアド・プロシージャの構文を覚えなくても複雑な管理作業を実行できます。Adaptive Server プラグインを使用して、次のような操作を実行できます。

- 1 台のコンソールから複数のサーバを管理する – Sybase Central メイン・ウィンドウから、すべての Adaptive Server インストールを管理できます。
- データベース定義言語 (DDL) を作成する – Adaptive Server でオブジェクトの DDL を作成できます。
- オブジェクトを視覚的に表示する – 各 Adaptive Server のデータベースとログイン、および各データベース内のオブジェクトを表示できます。ウィンドウを閉じたり開いたりして、データベースやログインの情報を表示できます。Adaptive Server プラグインでは、次のように多数の項目に関する情報が表示されます。
 - データベースとテーブル
 - ディスク・デバイス
 - アクティブなプロセスとロック
 - ログインとユーザ
 - データ・キャッシュ
 - ASE Replicator、Job Scheduler、メッセージング・サービス
 - Interactive SQL などの他のユーティリティへのアクセス (クエリの送信とクエリ結果の表示)

- 関連オブジェクト間のナビゲーション – 表示中のプロパティ・シートのオブジェクトと関連があるデータベース・オブジェクトの詳細を取得するには、表示されたオブジェクトのダイアログ・ボックスを通じて関連オブジェクトに直接ナビゲートします。
- クラスタの作成 – Adaptive Server Cluster Edition を購入した場合は、Adaptive Server プラグインを使用してクラスタを作成できます。オンライン・ヘルプと『Cluster ユーザーズ・ガイド』を参照してください。

Adaptive Server プラグインの使用

Adaptive Server Plug-in for Sybase Central では、直感的でわかりやすい操作によって Adaptive Server Enterprise を管理できます。Sybase Central では、左側のウィンドウ枠に Adaptive Server プラグインが表示されます。このウィンドウ枠に、プラグインで管理できるさまざまなオブジェクトのフォルダが階層形式でリストされます。次のような作業を実行できます。

- オブジェクトの特性の表示および変更
- 別のオブジェクトの作成
- オブジェクトを作成するための SQL テキストの作成 (Adaptive Server オブジェクトのリバース・エンジニアリング)
- オブジェクトの削除
- Adaptive Server の設定
- 以下の項目の管理：
 - データベース・デバイス
 - プロキシ・データベースとテンポラリ・データベース
 - インデックス
 - パーティション
 - セグメント
 - トリガ
 - ログインと役割
 - ビュー
 - ASE Replicator
- Job Scheduler での Adaptive Server ジョブの設定
- Adaptive Server の起動と停止
- クエリの実行
- ユーザの操作に基づいた、プラグインで作成した SQL 文のログ

Sybase Central の起動と停止

Sybase Central を起動するには、次の手順に従います。

- (UNIX の場合) `$$SYBASE/shared/sybcentral600` ディレクトリに移動し、`scjview.sh` スクリプトを実行します。
- (Windows の場合) [スタート] メニューから [プログラム] - [Sybase] - [Sybase Central v6] を選択するか、`%SYBASE%\Shared\Sybase Central 6.0.0` ディレクトリに移動し、`scjview.bat` スクリプトを実行します。

Sybase Central を終了するには、[ファイル] - [終了] を選択します。

Adaptive Server プラグインの登録

Adaptive Server プラグインは、サーバ・インストール環境の一部として Sybase Central に登録されます。ただし、Adaptive Server プラグインが適切に登録されない場合は、Adaptive Server プラグインを手動で登録できます。

- Unix の場合、`$$SYBASE/ASEP/bin/registerASEP` を実行します。
- Windows の場合、`%SYBASE%\ASEP\bin\registerASEP.bat` を実行します。
- Adaptive Server プラグインを手動で登録するには、次の手順に従います。
 - a [ツール] - [プラグイン] - [登録] を選択します。登録ウィザードが表示されます。
 - b [登録] を選択します。
 - c [プラグイン登録ファイルの指定によって、プラグインを登録します。] を選択します。
 - d [参照] をクリックします。
 - e `$$SYBASE/ASEP/bin` (Windows の場合は `%SYBASE%\ASEP\bin`) に移動し、`ASEPlugin.jpr` を選択します。Adaptive Server プラグインを使用して、次のような操作を実行できます。

共通の作業の実行

Adaptive Server プラグインでユーザがよく実行する作業を以下に示します。

各作業の詳細については、Adaptive Server プラグインのオンライン・ヘルプを参照してください。

Adaptive Server の起動と停止

Unified Agent で Adaptive Server をモニタしている場合は、サーバを右クリックし、[停止]、[起動]、または [再起動] を選択することによって、サーバを起動、停止、再起動します。

Adaptive Server を Unified Agent でモニタしていない場合は、[停止] を選択してサーバを停止します。

Adaptive Server との接続

Adaptive Server に接続するには、次のいずれかの方法を使用します。

- ツールバーの接続アイコンを選択します。
- [Adaptive Server Enterprise] を右クリックし、メニューから [接続] を選択します。
- サーバ・グループを右クリックし、メニューから [接続] を選択します。

[Adaptive Server Enterprise] フォルダまたは接続アイコンから接続を開始した場合は、接続されたサーバが「デフォルト」サーバ・グループに表示されます。サーバ・グループから接続を開始した場合は、該当するサーバ・グループに“Connected to server”と表示されます。

接続するサーバを指定するには、次のいずれかの方法を使用します。

- [接続] ダイアログ・ボックスで、サーバのホスト名とポート番号を指定します。
- [サーバ名] ドロップダウン・リストから、事前に定義された Adaptive Server を選択します。このドロップダウン・リストは、`interfaces` ファイル (UNIX)、`sql.ini` ファイル (Windows)、LDAP サーバのサーバ・リストで構成されます。
- [接続] ダイアログで [検索] をクリックして、使用可能な Adaptive Server を検出します。この方法を使用するには、Adaptive Server Enterprise のプロパティ・ページにある [Server Discovery] タブで、検出サーバをあらかじめ定義しておく必要があります。

データベースの作成

データベースを作成する前に、使用する予定のデータベース・デバイスに使用可能な領域が十分あることを確認してください。

データベースを作成するには、次の手順に従います。

- 右ウィンドウ枠で、[データベースの追加] アイコンを右クリックします。または、次の手順を実行します。
- 1 [データベース] フォルダを選択します。

- 2 [ファイル]-[新規]-[データベース]を選択するか、[データベース]フォルダで右クリックし、[新規]-[データベース]を選択します。[データベースの追加]ウィザードが開きます。[データベースの追加]ウィザードでは、以下の情報が要求されます。

表 4-1: [データベースの追加]ウィザードの入力項目

入力項目	説明
データベース名	データベースの名前を入力する。
データベース・デバイス	新しいデータベースを割り付けるデータベース・デバイスを指定する。
データベース・デバイスのサイズ	各データベース・デバイスのサイズを指定する。
データまたはログ	データベース・デバイスにデータとトランザクション・ログのどちらを保存するかを指定する。
上書き	同じデバイスにデータとログを保存する場合は、上書きを指定する。
ロード用	バックアップからリストアできるようにデータベースを作成する場合は、[ロード用]チェック・ボックスをオンにする。これは、メディア障害からリカバリする場合、またはある場所から別の場所にデータベースを移動する場合のみに適用される。
guest アカウント	データベースに guest ユーザを作成するかどうかを指定する。

サイズを入力しなかった場合は、**database size** 設定変数の値と *model* データベースのサイズのうち、大きい方が割り当てられます。

記憶領域が限られており、かつ、トランザクション・ログとデータを同じ論理デバイスに格納しなければならない場合は、[上書き]をオンにすると、データと別のデバイス・フラグメントでログを管理できます。

データベースの作成後は、データベースを削除しない限り、データベース・デバイスの削除や変更を行うことはできません。

警告! データベースを削除すると、そのデータベースのすべてのオブジェクトが削除されます。

データベースの削除

データベースを削除できるのは、そのデータベースの所有者のみです。

データベースを削除するには、次の手順に従います。

- 1 データベース・アイコンを選択します。
- 2 [編集]-[削除]を選択します。
- 3 [削除の確認]ダイアログ・ボックスで削除を確認します。

注意 ユーザ・データベースを削除した後、**master** データベースをバックアップすることをおすすめします。

ユーザの追加

データベース所有者は、自分が所有するデータベースでユーザを追加および削除できます。

ユーザを作成するには、次の手順に従います。

- 1 [データベース] フォルダを展開し ([+] アイコンを選択)、[ユーザ] フォルダを選択します。
- 2 [ファイル]-[新規]-[ユーザ] を選択します。
[ユーザの追加] ウィザードが起動し、以下の情報を要求します。

表 4-2: [ユーザの追加] ウィザードの入力項目

入力項目	説明
名前	ユーザの名前。この名前はログイン名と同じである必要はない。
ログイン名	このユーザが割り当てられるログイン。
グループ	ユーザにグループを割り当てる (任意選択)。デフォルト: public

注意 ユーザをいずれかのグループに割り当てることができます。どのグループにも割り当てられていないユーザは、デフォルトの“public”グループのメンバーになります。

または、[ユーザ] フォルダを選択します。[ユーザ] フォルダで右クリックし、[新規]-[ユーザ] を選択します。

ユーザの削除

オブジェクトを所有しているユーザは削除できません。オブジェクトの所有権を譲渡するコマンドはないので、ユーザが所有しているオブジェクトを削除してからユーザを削除してください。同様に、他のユーザにパーミッションを付与しているユーザを削除するには、最初にパーミッションをカスケード付きで取り消す必要があります。必要に応じて、他のユーザに再度パーミッションを付与します。

ログインのロックは、ユーザ削除の簡単な代替方法です。

ユーザを削除するには、次の手順に従います。

- 1 ユーザ・アイコンを選択します。
- 2 [編集]-[削除] を選択します。
- 3 [削除の確認] ダイアログ・ボックスで削除を確認します。

または、ユーザ・アイコンを右クリックして [ユーザ] フォルダを選択し、[削除] を選択することもできます。

ユーザを削除する前に、次の操作を行ってください。

- 1 ユーザのコマンドとオブジェクトのパーミッションをカスケード付きで取り消します。
- 2 必要に応じて、他のユーザに再度パーミッションを付与します。
- 3 ユーザのオブジェクトを削除します。

テーブルの作成

create table パーミッションを持つデータベース所有者またはユーザだけがテーブルを作成できます。

テーブルを作成するには、次の手順に従います。

- 1 作業しているデータベースで [ユーザ・テーブル] フォルダを選択します。
- 2 [ファイル]-[新規]-[テーブル] を選択するか、[ユーザ・テーブル] フォルダで右クリックし、[新規]-[テーブル] を選択します。

テーブル・エディタが開きます。

- 3 [名前] ボックスに名前を入力します。
- 4 [所有者] リストから所有者を選択します。デフォルトは “dbo” です。

または、[ユーザ・テーブル] フォルダを選択します。右ウィンドウ枠で、[テーブルの追加] アイコンをダブルクリックします。

テーブルの削除

テーブルの削除は、そのテーブルを参照している他のオブジェクトがないことを確認してから行ってください。参照しているオブジェクトがある場合は、それらのオブジェクトを編集してエラーを回避します。他のオブジェクトがテーブルを参照しているかどうかを確認するには、その依存性を調べます。

注意 テーブルを削除すると、そのテーブルに関連付けられているインデックスとトリガが削除され、そのカラムにバインドされているルールまたはデフォルトがバインド解除されます。

テーブルを削除できるのは、テーブルの所有者だけです。

テーブルを削除するには、次の手順に従います。

- 次の手順に従ってください。
 - テーブル・アイコンを選択します。
 - [編集]-[削除] を選択します。
 - [削除の確認] ダイアログ・ボックスで削除を確認します。
- または、テーブル・アイコンを右クリックしてテーブルを選択し、[削除] を選択することもできます。

- サーバ・グループの作成** サーバ・グループを作成するには、次の手順に従います。
- 1 [Adaptive Server Enterprise] を選択します。
 - 2 [ファイル]-[新規]-[サーバ・グループ] を選択します。
 - 3 [サーバグループの追加] ウィザードの手順に従います。
- または、右ウィンドウ枠で [新規]-[サーバ・グループ] を選択してサーバ・グループを追加することもできます。
- サーバ・ステータスの取得** Adaptive Server を Unified Agent でモニタしている場合は、次のいずれかの方法でサーバのステータスを確認します。
- サーバが所属するサーバ・グループをクリックします。サーバ・グループの [詳細] ウィンドウ枠の [ステータス] カラムを確認します。
 - [Sybase Central] の下の [Adaptive Server Enterprise] をクリックし、右ウィンドウ枠の [サーバ] タブをクリックします。サーバのステータスは [ステータス] カラムに表示されます。
 - サーバ・アイコンの右下に緑色の三角形が表示されている場合、Adaptive Server は稼動中です。赤い正方形は、Adaptive Server が停止中であることを示します。
-
- 注意** デフォルトでは、Adaptive Server プラグインのサーバ・ステータスのチェックは無効になっています。Unified Agent で Adaptive Server をモニタできるようにするには、次の手順に従います。
- [Adaptive Server Enterprise] を右クリックし、[プロパティ] を選択します。
 - [ユーザ独自の設定] を選択し、[Unified Agent (UA) 関連の機能を有効にする] を選択します。
-
- サーバ・ログの取得** Unified Agent で Adaptive Server をモニタしている場合は、サーバを選択し、右ウィンドウ枠の [Server Log] タブをクリックして、サーバ・ログを取得します。
- サーバ・ログは、サーバ・ログのフィルタ設定に基づいて取得されます。サーバ・ログのフィルタを設定するには、サーバを右クリックし、[サーバ・ログ・フィルタ] を選択します。デフォルトでは、最新の 1000 行をサーバ・ログから取得します。サーバのフィルタ設定に基づいて、次のサーバ・ログを取得できます。
- ログ・ファイル全体
 - 最新の n 行
 - 最新の n 日間のログ
 - 正規表現に一致する行

- SQL 文のロギング** Adaptive Server プラグインによって実行されたすべての SQL 文のログを取るには、次の手順に従います。
- サーバを右クリックし、[SQL 文のログを取る] を選択します。
 - SQL 文のログをウィンドウに直接送るか、ファイルに送るかを選択します。
- SQL 文の実行** Interactive SQL クエリ・ツールを使用して、Adaptive Server プラグイン内から SQL 文を実行します。Interactive SQL ツールを起動するには、次のいずれかの手順に従います。
- SQL 文を実行するサーバを右クリックし、メニューから [対話型 SQL のオープン] を選択します。または、以下の手順に従います。
 - 1 [Adaptive Server Enterprise] をクリックします。
 - 2 右ウィンドウ枠の [ユーティリティ] タブをクリックし、[Interactive SQL] を選択します。
- サーバ・グループに所属するサーバのセットに対して SQL 文を同時に実行します。次の手順に従います。
- 1 サーバ・グループを右クリックし、[SQL の実行] を選択します。
 - 2 SQL 文を実行するサーバを選択します。
 - 3 [実行] をクリックします。
- 各サーバの結果セットは、[SQL Execution] ダイアログの [結果セット] ウィンドウ枠に表示されます。
- SQL の実行プランとコスト情報の表示** Adaptive Server プラグインを使用して、各クエリの SQL 実行プランの GUI バージョン (showplan の GUI バージョン)、およびストアド・プロシージャのすべてのクエリの実行プランの GUI バージョンを表示します。この GUI 表示では、実行プランの演算子ごとにノードが含まれます。
- GUI でプランを取得するには、次の手順に従います。
- 1 Interactive SQL を起動します。
 - 2 クエリまたはストアド・プロシージャを実行します。
 - 3 [ツール]-[プランビューワを開く] を選択します。
 - 4 SQL ウィンドウ枠下のドロップダウン・リストからクエリを選択します。
 - 5 [詳細] タブをクリックして、選択したクエリの GUI プランを表示します。演算子ノードをクリックして、ノードの統計情報の詳細を表示します。
 - 6 [XML] タブをクリックして、選択したクエリの実行プランを XML 形式で表示します。
 - 7 [Text] タブをクリックして、送信したクエリの実行プランをテキスト形式で表示します。
- Interactive SQL の詳細については、「[Interactive SQL の起動](#)」(56 ページ) を参照してください。

オブジェクト・プロパティの表示と更新

[プロパティ] ダイアログを使用して、Adaptive Server プラグインで表示されるすべてのオブジェクトの設定を表示および修正します。

[プロパティ] ダイアログを開くには、次の手順に従います。

- 1 表示または修正するオブジェクトをクリックします。
- 2 オブジェクトを右クリックし、[プロパティ] を選択します。
- 3 実行する作業に合った適切なタブを選択します。
- 4 [プロパティ] ダイアログで、必要な修正を加えます。
- 5 [適用]、[OK]、または [キャンセル] をクリックします。

オブジェクトを作成するための SQL テキストの生成

オブジェクトの作成に必要な SQL テキストを生成します。これは、オブジェクトのリバース・エンジニアリングが可能であることを意味します。SQL テキストを作成するには、オブジェクトを右クリックし、[DDL の生成] を選択します。

Adaptive Server 設定パラメータの表示と更新

[サーバのプロパティ] ダイアログで、Adaptive Server 設定パラメータを表示および更新します。

- 1 サーバを右クリックし、メニューから [設定] を選択します。
- 2 [設定パラメータを表示] のドロップダウン・リストで、機能グループを選択します。
- 3 表示または更新するパラメータを見付けて選択します。
- 4 更新する必要がある場合は、[値] カラムに、新しい値を入力します。
- 5 必要に応じて、[適用]、[OK]、または [キャンセル] をクリックします。

Interactive SQL の使用

Interactive SQL を使用すると、サーバに対して SQL 文の実行、スクリプトの作成、データベース・データの表示を行うことができます。Interactive SQL は、次の目的で使用できます。

- データベース内の情報をブラウズする。
- アプリケーションで使用する予定の SQL 文をテストする。
- クエリ結果をファイルに保存する。
- 結果セットのデータを編集する。
- データをデータベースにロードし、管理作業を実行する。

Interactive SQL では、コマンド・ファイルまたはスクリプト・ファイルを実行することもできます。たとえば、データベースに実行する繰り返し可能なスクリプトを作成し、Interactive SQL を使用してそれらのスクリプトをバッチで実行できます。

Interactive SQL の起動

Sybase Central からの Interactive SQL の起動

Interactive SQL を起動するには、次のいずれかの手順を実行します。

- Sybase Central でデータベースを選択し、[ファイル]-[対話型 SQL のオープン] を選択します。Interactive SQL がデータベースに接続します。または、データベースを右クリックし、[対話型 SQL のオープン] を選択することもできます。

メニュー項目 [対話型 SQL のオープン] は、サーバへの接続をオープンします。ただし、サーバのメニュー項目を選択した場合は、そのサーバのデフォルト・データベースへの接続をオープンします。[対話型 SQL のオープン] メニューから特定のデータベースを選択すると、選択されたデータベースに対して Interactive SQL がオープンします。

- サーバに接続しないで Interactive SQL を起動するには、[ツール]-[Adaptive Server Enterprise]-[対話型 SQL] を選択します。[接続] ダイアログが表示されます。

コマンド・ラインからの Interactive SQL の起動

コマンド・ラインから Interactive SQL を起動する手順は、使用するオペレーティング・システムによって異なります。

Interactive SQL を単独で起動すると、[接続] ダイアログが表示され、Sybase Central の場合と同じ方法でデータベースに接続できます。

- UNIX の場合は、`$$SYBROOT/DBISQL/bin` ディレクトリに移動し、次のように入力します。

```
dbisql
```

Windows の場合は、`%SYBROOT%\DBISQL\bin` ディレクトリに移動し、次のように入力します。

```
dbisql.bat
```

- [接続] ダイアログで、[接続] ダイアログ・ボックスにデータベースへの接続情報を入力し、[OK] をクリックします。

新しい Interactive SQL ウィンドウを開くには、次の手順に従います。

- 1 [ウィンドウ]-[新しいウィンドウ] を選択します。[接続] ダイアログが表示されます。
- 2 [接続] ダイアログで接続オプションを入力し、[OK] をクリックして接続します。

接続情報 (データベース名、ユーザ ID、データベース・サーバなど) は、[SQL 文] ウィンドウ枠の上のタイトル・バーに表示されます。

また、SQL メニューの [接続] コマンドと [切断] コマンドを使用するか、SQL 文のウィンドウ枠で `connect` 文または `disconnect` 文を実行することにより、データベースに接続したり接続を切断したりできます。

設定パラメータ

この章では、Adaptive Server の設定パラメータについて説明します。パラメータはアルファベット順に並べてあります。

設定パラメータは、`sp_configure` を使用してユーザが設定できるパラメータです。設定パラメータは、基本的なサーバ操作から特殊なサーバ操作に至る広範囲なサービスと、パフォーマンス・チューニングに使用します。

トピック名	ページ
概要	57
sp_configure の使用	62
sp_configure 出力	72
Named Cache 設定パラメータ	74
sysconfigures テーブルと syscurconfigs テーブル	75
設定パラメータ	75

概要

設定パラメータは、Adaptive Server の動作に関するさまざまな制御を行うためにユーザが設定できるパラメータです。すべての設定パラメータにデフォルト値が用意されています。設定パラメータを使用して、インストール環境の特定のニーズを満たすように Adaptive Server をカスタマイズします。

この章を参照してサーバのパフォーマンスを最適化するためにどの設定パラメータを設定し直したらよいか、慎重に調べてください。

警告！ 設定パラメータの変更は、十分注意して行ってください。パラメータ値を不用意に変更すると、Adaptive Server のパフォーマンスやサーバ・オペレーションの他の部分に悪影響を与えることがあります。

Adaptive Server の設定ファイル

Adaptive Server の設定パラメータの値は、ASCII テキスト・ファイルである設定ファイルに保存されます。新しく Adaptive Server をインストールするときは、パラメータはデフォルト値に設定されます。デフォルトでは、この設定ファイルの名前は *server_name.cfg* で、Sybase Adaptive Server のホーム・ディレクトリ (*\$SYBASE_ASE*) に保存されます。設定パラメータを変更するたびに、それまでの設定ファイルのコピーが作成されますが、その名前は *server_name.001*、*server_name.002*、*server_name.003*、...、*server_name.999* というように付けられます。新しい値は、*server_name.cfg* ファイルまたは起動時に指定したファイルに書き込まれます。

設定パラメータの変更

設定パラメータを設定または変更するには、次の 3 つの方法があります。

- `sp_configure` に該当するパラメータと値を指定して実行する。
- 設定ファイルを編集してから、`configuration file` オプションを指定して `sp_configure` を実行する。
- 起動時に設定ファイル名を指定する。

設定パラメータには、動的パラメータと静的パラメータがあります。動的パラメータは、`sp_configure` を実行するとすぐに有効になります。静的パラメータの場合は、メモリを再割り付けする必要があるため、Adaptive Server を再起動しないと有効にはなりません。この章の各パラメータの説明には、そのパラメータが静的か動的かが記載されています。

ユーザが値を変更すると、新しい値がシステム・テーブル `sysconfigures` と設定ファイルに書き込まれます。現在の設定ファイルと `sysconfigures` は、実行値ではなく、設定値を反映しています。システム・テーブル `syscurconfigs` は、設定パラメータの現在の実行値を反映しています。

設定パラメータの変更に必要な役割

`sp_configure` を使用するために必要な役割は次のとおりです。

- どのユーザも `sp_configure` を実行して、パラメータおよびパラメータの現在の値を表示できます。
- `sp_configure` を実行して設定パラメータを変更できるのは、システム管理者とシステム・セキュリティ担当者だけです。

- `sp_configure` を実行して次の設定パラメータの値を変更できるのは、システム・セキュリティ担当者だけです。
 - `allow procedure grouping`
 - `allow remote access`
 - `allow sendmsg`
 - `allow updates to system tables`
 - `auditing`
 - `audit queue size`
 - `check password for digit`
 - `current audit table`
 - `enable ldap user auth`
 - `enable pam user auth`
 - `enable ssl`
 - `log audit logon failure`
 - `log audit logon success`
 - `maximum failed logins`
 - `minimum password length`
 - `msg confidentiality reqd`
 - `msg integrity reqd`
 - `secure default login`
 - `select on syscomments.text`
 - `SQL Perfmon Integration`
 - `syb_sendmsg port number`
 - `suspended audit when device full`
 - `systemwide password expiration`
 - `unified login required`
 - `use security services`

sp_configure による単位の指定

sp_configure では、設定パラメータの値の単位を単位指定子で指定できます。単位指定子には、ページ数を表す **p** または **P**、メガバイト数を表す **m** または **M**、ギガバイト数を表す **g** または **G** があります。メモリの量を指定するパラメータを設定するときに単位を指定しなかった場合は、論理ページ・サイズが基本単位として使用されます。

注意 メモリに関するパラメータを設定するときは、**P** (ページ・サイズ) パラメータ以外の単位指定は使用しないでください。メモリに関するパラメータの設定時に他のパラメータを使用すると、算術オーバーフローのエラー・メッセージが返されることがあります。

単位指定の構文は次のとおりです。

```
sp_configure "parameter name", 0, "p|P|k|K|m|M|g|G"
```

“0” をプレースホルダとして必ず指定します。

この単位指定はすべてのパラメータで使用できます。たとえば、**number of locks** を 1024 に設定する場合は、次のように入力します。

```
sp_configure "number of locks", 1024
```

または

```
sp_configure "number of locks", 0, "1K"
```

この機能を使用しても、**sp_configure** の出力の内容は変わりません。

グローバル設定とセッション設定

Adaptive Server のグローバル設定には、セッション・レベルの設定と類似したものもあります。たとえば、**sp_configure** と **sp_passwordpolicy** の両方に **minimum password length** パラメータが含まれています。

グローバル・パラメータとセッション・レベル・パラメータが同じ設定の場合、ログインにはセッション・レベル設定が適用されます (ログイン・レベル設定がグローバル設定を上書きします)。

たとえば、このグローバル設定パラメータを次のように設定するとします。

```
sp_configure 'minimum password length' 12
```

そして長さの不十分なパスワードで、**joe** というユーザをサーバに追加しようとします。

```
sp_adduser joe, joejoe
```

このコマンドは失敗します。

```
Msg 10317, Level 14, State 1:
Procedure 'sp_password', Line 118:
The specified password is too short. Passwords must be at least
12 character(s) long.
Msg 17720, Level 16, State 1:
Procedure 'sp_password', Line 128:
Error: Unable to set the Password.
```

しかし、**sp_passwordpolicy** を短い長さに設定すると、Adaptive Server はこのセッションでは短いパスワードを要求します。

```
sp_passwordpolicy 'set', 'minimum password length', 6
```

このコマンドは成功します。

```
sp_addlogin joe, joejoe

Password correctly set.
Account unlocked.
New login created.
(return status = 0)
```

Adaptive Server はすべての設定の制限を独立にテストします。このためログインが成功するには、適用される制限すべてに合格する必要があります。

設定パラメータのヘルプ情報の取得

特定の設定パラメータのヘルプ情報を参照するには、**sp_helpconfig** または **sp_configure** を使用します。次に例を示します。

```
sp_helpconfig "number of open"
```

Configuration option is not unique.

option_name	config_value	run_value
number of open databases	12	12
number of open indexes	500	500
number of open objects	500	500

```
sp_helpconfig "number of open indexes"
```

number of open indexes sets the maximum number of indexes that can be open at one time on SQL Server. The default value is 500.

Minimum Value	Maximum Value	Default Value	Current Value	Memory Used
100	2147483647	500	500	208

```
sp_configure "number of open indexes"
```

Parameter Name	Default	Memory Used	Config Value	Run Value
----------------	---------	-------------	--------------	-----------

```
-----
number of open indexes      500      208      500      500
-----
```

『システム管理ガイド 第2巻』の「第3章 メモリの設定」を参照してください。

sp_configure の使用

sp_configure は、設定パラメータを表示または再設定します。sp_displaylevel を使用して次のレベルのいずれかに表示レベルを設定することによって、sp_configure が表示するパラメータの数を制限できます。

- 基本
- 中間
- 包括

表示レベルの詳細については、「[パラメータ階層のユーザ定義サブセット \(表示レベル\)](#)」(70 ページ)を参照してください。sp_displaylevel の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

表 5-1 は、sp_configure の構文の説明です。「結果」の欄は、表示レベルを「包括」に設定した場合についての説明です。

表 5-1: sp_configure の構文

コマンド	結果
sp_configure	すべての設定パラメータについて、現在の値、デフォルト値、前回の設定値、設定に必要なメモリ量をグループごとに表示する。
sp_configure "parameter"	現在の値、デフォルト値、前回の設定値、指定されたパラメータが使用するメモリ量を表示する。
sp_configure "parameter", value	parameter の値を value に再設定する。
sp_configure "parameter", 0, "default"	指定されたパラメータをデフォルト値にリセットする。
sp_configure "group_name"	group_name 内のすべての設定パラメータについて、現在の値、デフォルト値、前回の設定値、設定に必要なメモリ量を表示する。
sp_configure "configuration file", 0, "sub_command", "file_name"	設定ファイルから設定パラメータを設定する。パラメータの詳細については、「 設定ファイルを指定して sp_configure を使用する方法 」(63 ページ)を参照してください。

構文の要素

表 5-1 のコマンドで使用している変数は次のとおりです。

- **parameter** は、有効な Adaptive Server 設定パラメータまたはパラメータの部分文字列です。
- **value** は、そのパラメータに有効な範囲内の整数値です。有効な範囲については、それぞれのパラメータの説明を参照してください。トグル式のパラメータに有効な値は、1 (オン) と 0 (オフ) だけです。
- **group_name** は、パラメータ階層内のグループの名前です。

パラメータの解析

`sp_configure` は、それぞれのパラメータ (およびパラメータ名の一部) を “%parameter%” として解析します。文字列によって特定されるパラメータが 1 つでない場合は、その文字列に一致するすべてのパラメータの値を返します。

次の例では、`lock shared memory`、`number of locks`、`lock promotion HWM`、`server clock tick length`、`print deadlock information`、`deadlock retries` などの “lock” を含むすべての設定パラメータの値が返されます。

```
sp_configure "lock"
```

注意 `sp_configure` でパラメータ名の一部を指定してパラメータ値を設定するときに、一致するパラメータが 2 つ以上ある場合は、そのパラメータ名部分に一致するすべてのパラメータの現在値が返され、ユニークなパラメータ名の入力を要求するプロンプトが表示されます。

設定ファイルを指定して `sp_configure` を使用する方法

Adaptive Server の設定は、上記のように `sp_configure` を使用して対話型で行うことも、編集またはリストアした設定ファイルから値を読み込む非対話型で行うこともできます。

設定ファイルから変更することで、以下のことを行えます。

- 同一の設定ファイルを使用することによって、複数のサーバに特定の設定を複製する
- 各自のサーバ上で設定値をテストするための基準として、設定ファイルを使用する
- 実際に値を設定する前に、パラメータ値を検証するために設定ファイルを使用する
- 複数の設定ファイルを作成し、リソースの変化に応じて設定ファイルを切り替える

ファイルの編集方法については、「[設定ファイルの編集](#)」(65 ページ)を参照してください。起動時に設定ファイル名を指定する方法については、「[設定ファイルを指定した Adaptive Server の起動](#)」(67 ページ)を参照してください。

設定ファイルの名前付けについてのヒント

設定ファイルの名前をデフォルトの名前から変更するとき、ファイル名の *server_name* 部分をそのまま保持する場合は、必ず拡張子に英字を 1 つ以上入れてください (例: *my_server.A001*)。または、ファイル名の *server_name* 部分を変更することもできます (例: *A_my_server.001*)。このようにすれば、パラメータを修正するとき自動的に生成されるバックアップ設定ファイルとの混同を避けることができます。

sp_configure を使用した設定ファイルの読み込みまたは書き込み

sp_configure で configuration file オプションを使用する場合の構文は、次のとおりです。

```
sp_configure "configuration file", 0, "subcommand", "file_name"
```

各要素の意味は次のとおりです。

- “configuration file” – (引用符も必要) は、このコマンドで設定ファイルを使用することを指定する。
- 0 – configuration file パラメータの後に必要 (下位互換性のため)。
- “subcommand” – 以下のいずれかを指定する。
 - **write** は、現在の設定を使用してファイル *file_name* を作成します。*file_name* が既に存在する場合は、エラー・ログにメッセージが出力され、既存ファイルの名前は *server_name.001*、*server_name.002*、... という命名規則に従って変更されます。静的パラメータを変更した後にはサーバを再起動しないで **write** を使用した場合は、そのパラメータの現在実行している値が表示されます。*file_name* のディレクトリを指定しない場合は、ファイルは Adaptive Server が起動されたディレクトリに書き込まれます。
 - **read** は、*file_name* 内に記述されている値を検証し、検証をパスした値をサーバに読み込みます。*file_name* に記述されていないパラメータには、そのパラメータの現在の値が使用されます。
file_name 内の静的パラメータの値が現在の実行値と異なる場合、**read** は失敗し、メッセージが表示されます。ただし、この場合も *file_name* 内の値の検証は実行されます。
 - **verify** は、*file_name* 内の値の検証を実行します。このサブコマンドは、不正な値を指定してサーバを設定するのを防ぐので、設定ファイルを編集した場合に役立ちます。

- **restore** は、最新の設定値を使用して *file_name* を作成します。静的パラメータの新しい値を設定した後でこのサブコマンドを使用すると、現在の実行値ではなく、設定した値がファイルに書き込まれます。このコマンドは、設定ファイルのコピーがすべて消失してしまった場合に新しいコピーを生成するときに便利です。*file_name* のディレクトリを指定しない場合は、ファイルは Adaptive Server が起動されたディレクトリに書き込まれます。
- *file_name* には、*subcommand* とともに使用する設定ファイルを指定します。ファイル名の一部としてディレクトリを指定しない場合は、Adaptive Server の起動ディレクトリが使用されます。

例

例 1. *srv.config* ファイル内の値の検証を実行し、検証をパスしたパラメータをサーバに読み込みます。検証をパスしない値に対しては現在の実行値が使用されます。

```
sp_configure "configuration file", 0, "read", "srv.config"
```

例 2. *my_server.config* ファイルを作成し、サーバが使用している現在の設定値をこのファイルに書き込みます。

```
sp_configure "configuration file", 0, "write", "my_server.config"
```

設定ファイルの編集

設定ファイルは ASCII ファイルであるので、テキスト・エディタを使用して編集し、ASCII フォーマットで保存することができます。それぞれのパラメータの構文は、次のとおりです。

```
parameter_name={value | DEFAULT}
```

構文の説明は、次のとおりです。

- *parameter_name* は、指定するパラメータの名前です。
- *value* は、指定した *parameter_name* を設定する数値です。
- “DEFAULT” は、*parameter_name* にデフォルト値を使用する場合に指定します。

例

例 1. 次の例では、インデックスの **page split** または縮小中にデッドロックが発生した場合に、トランザクションがロックの取得を 1 回だけリトライするように指定します。

```
deadlock retries = 1
```

例 2. 次の例では、パラメータ **cpu accounting flush interval** にはデフォルト値を使用することを指定します。

```
cpu accounting flush interval=DEFAULT
```

設定ファイルを編集する場合は、**verify** オプションを使用してファイルを検証するか、**read** オプションを使用してファイルを読み込むか、またはその設定ファイルを指定して Adaptive Server を再起動するまでは、編集内容の検証は行われません。

すべての設定ファイルが消失するか破損した場合は、**restore** サブコマンドを使用することによって、稼働中のサーバから設定ファイルを作り直して、新しいファイル名を指定できます。新しいファイル内のパラメータは、サーバが現在実行している値に設定されます。

設定ファイルに対するパーミッション

設定ファイルは暗号化されていない ASCII テキスト・ファイルです。デフォルトでは、ファイルの所有者に対しては読み取りと書き込みのパーミッションが、他のすべてのユーザに対しては読み取りパーミッションが設定されて作成されます。オペレーティング・システム・レベルで設定ファイルを作成する場合は、作成者がファイルの所有者になります。**write** または **restore** パラメータを使用して Adaptive Server から設定ファイルを作成する場合は、ファイルの所有者は Adaptive Server を起動したユーザになります。通常、これは“sybase”というユーザです。設定ファイルへのアクセスを制限するには、オペレーティング・システムのファイル・パーミッション・コマンドを使用して、読み込み、書き込み、実行の許可を設定してください。

注意 パーミッションは、作成した設定ファイルごとに設定する必要があります。

設定ファイルのバックアップ

master データベースのバックアップを実行しても、設定ファイルは自動的にバックアップされません。設定ファイルはオペレーティング・システム・ファイルであるため、他のオペレーティング・システム・ファイルをバックアップするのと同じ方法でバックアップしてください。

現在使用されている設定ファイル名の確認

sp_configure の出力では、表示領域の制限により設定ファイル名はトランケートされます。設定ファイル名を完全に表示するには、次の構文を使用します。

```
select s1.value2
from syscurconfigs s1, sysconfigures s2
where s1.config = s2.config
and s2.name = "configuration file"
```

設定ファイルを指定した Adaptive Server の起動

デフォルトでは、Adaptive Server の起動時に、起動ディレクトリ内にある設定ファイル `server_name.cfg` が読み込まれます。このファイルが存在しない場合は、新しいファイルが作成され、すべての値にデフォルト値が使用されます。

Adaptive Server の起動時に設定ファイルを指定することができます。詳細については、『ASE ユーティリティ・ガイド』を参照してください。

指定した設定ファイルが存在しない場合、Adaptive Server はエラー・メッセージを表示し、起動しません。

コマンドが正常に処理されると、ファイル `server_name.bak` が作成されます。このファイルには、指定した設定ファイルから読み込んだ値で `sysconfigures` が更新される前に、`sysconfigures` 内に保管されていた設定値が含まれます。このファイルは以降の起動のたびに上書きされます。

設定ファイルのエラー

設定ファイルにエラーがある場合は、Adaptive Server が起動しないか、デフォルト値が使用されるかのいずれかとなります。

次の場合はデフォルト値が使用されます。

- 無効な値がある場合。たとえば、数値を必要としているパラメータに対して、設定ファイルでは文字列が指定されている場合は、デフォルト値が使用されます。
- 値が最小許容値より小さい場合

パラメータの階層

設定パラメータは、対象となる Adaptive Server の動作領域に従ってグループ分けされています。このグループ分けによって、Adaptive Server の特定の領域のパフォーマンスを改善するために調整を必要とする、すべてのパラメータを簡単に識別できます。

パラメータはそれぞれ 1 つのプライマリ・グループに属していますが、多くのパラメータはセカンダリ・グループにも属しています。たとえば、`number of remote connections` は、ネットワーク通信グループにプライマリとして属するとともに、メモリ使用グループにもセカンダリとして属します。これは、パラメータの中には Adaptive Server の複数の動作領域に関連するものがあることを表しています。`sp_configure` を実行すると、パラメータは所属するすべてのグループに表示されます。

表 5-2 は、設定パラメータのグループを示します。

表 5-2: 設定グループ

パラメータ・グループ	Adaptive Server の設定対象
Backup/Recovery	データのバックアップとリカバリ
Cache manager	データ・キャッシュとプロシージャ・キャッシュ
Component Integration Services administration	コンポーネント統合サービス
DTM administration	DTM (分散トランザクション管理) 機能
Diagnostics	診断の原則
Disk I/O	ディスク I/O
Error log	Windows イベント・ログに記録される Adaptive Server のエラー・メッセージとイベント・ログ
Extended stored procedures	ESP (拡張ストアド・プロシージャ) の動作
General information	システムの基本的な管理
Java services	Adaptive Server 内の Java のメモリ データベースにおける Java の詳細については、『Adaptive Server Enterprise における Java』を参照してください。 JDBC へのメソッド呼び出しを使用する場合には、ユーザが使用できる実行スタックのサイズを大きくしなければならない場合があります。「 stack size 」(251 ページ)を参照してください。
Languages	言語、ソート順、文字セット
Lock manager	ロック
Memory use	メモリの消費
Metadata caches	頻繁に使用されるシステム・カタログ情報用のメタデータ・キャッシュ・サイズの設定。「メタデータ・キャッシュ」は、データベース、インデックス、オブジェクトに関する情報の追跡に使用する、予約済みのメモリ領域です。オープンしているデータベース、インデックス、オブジェクトの数が多いほど、メタデータ・キャッシュ・サイズは大きくなります。メモリ使用に関して使用されるメタデータ・キャッシュの詳細については、『システム管理ガイド 第 2 巻』の「第 3 章 メモリの設定」を参照してください。
Monitoring	モニタリング情報の収集。デフォルトでは、Adaptive Server はモニタリング情報を収集しません。 『パフォーマンス&チューニング・シリーズ: モニタリングと分析』の「第 2 章 モニタリング・テーブル」を参照してください。
Network communication	Adaptive Server とリモート・サーバ間、Adaptive Server とクライアント・プログラム間の通信
O/S resources	オペレーティング・システム・リソースの使用
Physical memory	マシンの物理メモリ・リソース
Processors	SMP 環境のプロセッサ
Query Tuning	クエリの最適化
RepAgent thread administration	Replication Server を介した複写
Shared Disk Cluster	クラスタ・エディション。
SQL Server administration	Adaptive Server の全般的な管理

パラメータ・グループ	Adaptive Server の設定対象
Security related	セキュリティ関連機能
Unicode	Unicode 関連機能
User environment	ユーザ環境

すべてのグループとそのグループに属するパラメータ、およびパラメータの現在の値を表示するための構文は次のとおりです。

```
sp_configure
```

注意 `sp_configure` が返すパラメータの数は、設定した表示レベルの値によって変わります。「[パラメータ階層のユーザ定義サブセット \(表示レベル\)](#)」(70 ページ)を参照してください。

特定のグループとそのグループに属するパラメータを表示するための構文は次のとおりです。

```
sp_configure "group_name"
```

たとえば、ディスク I/O (Disk I/O) グループを表示するには次のように入力します。

```
sp_configure "Disk I/O"

Group: Disk I/O
Parameter Name      Default Memory Used Config Value Run Value
unit                type
-----
-----
allow sql server async i/o      1          0          1          1
switch                static
diabile disk mirroring          1          0          1          1
switch                static
disk i/o structures            256         0          256         256
number                dynamic
number of devices             10          0          10          10
number                dynamic
number of large I/O buffers     6          12352         6          6
number                dynamic
page utilization percent       95          0          95          95
percent                dynamic
```

注意 サーバで大文字と小文字を区別しないソート順が使用されている場合に、パラメータを指定しないで `sp_configure` を実行すると、グループ分けは表示されず、すべての設定パラメータとグループのアルファベット順リストが返されます。

パラメータ階層のユーザ定義サブセット (表示レベル)

Adaptive Server の使用形態によっては、一部のパラメータを他のパラメータよりも頻繁に調整することが必要です。パラメータのサブセットを操作する方が簡単な場合があります。

デフォルトの表示レベルは **comprehensive** です。設定した表示レベルは、以降のセッションにも適用されます。ただし、いつでも再設定できます。

- **basic** (基本) – 最も基本的な設定パラメータだけを表示し、一般的なサーバ・チューニングに適しています。
- **intermediate** (中間) – 「基本レベル」で表示されるパラメータに加えて、やや複雑なパラメータも表示されます。
- **comprehensive** (包括) – 最も複雑なものまでを含むすべてのパラメータが表示されます。このレベルは、かなり詳細なサーバ調整を行う場合に適しています。

現在の表示レベルを表示するための構文は、次のとおりです。

```
sp_displaylevel
```

表示レベルを設定するには以下を使用します。ここで **user_name** は、実行するユーザの Adaptive Server ログイン名です。

```
sp_displaylevel user_name[, basic | intermediate | comprehensive]
```

sp_configure 出力への表示レベルの影響

表示レベルが「基本レベル」か「中間レベル」に設定されている場合に **sp_configure** を実行すると、返されるパラメータは、「包括レベル」の場合に返されるパラメータのサブセットだけです。たとえば、表示レベルが「中間レベル」に設定されているときに、言語グループ内のパラメータを表示するには、次のように入力します。

```
sp_configure "Languages"
```

出力は次のようになります。

```
sp_configure
Group: Languages
```

Parameter Name	Default	Memory	Used	Config	Value	Run	Value	Unit	Type
default character set	1	0		1	1	id	static		
default language id	0	0		0	0	id	dyna		
...									

ここで表示されるのは、言語グループ内のパラメータのサブセットだけです。一部の言語パラメータは、表示レベルが「包括レベル」のときだけ表示されます。

sp_configure と sp_sysmon によるパフォーマンス・チューニング

sp_sysmon は、Adaptive Server のパフォーマンスをモニタし、Adaptive Server システムの動作を表す統計情報を生成します。『パフォーマンス&チューニング・シリーズ: sp_sysmon による Adaptive Server の監視』を参照してください。

sp_configure を使用する前と使用後に sp_sysmon を実行して、設定パラメータを調整できます。出力からパフォーマンス・チューニングの基礎情報が得られ、設定変更の結果を監視できます。

クラスタード環境における設定パラメータの使用

Cluster Edition では、クラスタ全体とインスタンス固有の設定の両方がサポートされています。クラスタ全体設定パラメータは、クラスタ内のインスタンスすべてに適用されます。ローカル設定パラメータは、指定されたインスタンスだけに適用されます。

- ローカル設定は、クラスタ全体設定を無効にする。
- インスタンス固有の設定が適用されていない場合、クラスタ全体設定が適用される。
- 一部のパラメータ (default character set id など) は、特定のインスタンスに適用できない。それらのパラメータはクラスタ全体にだけ適用できる。

クラスタ設定ファイルには、インスタンス固有の設定ブロックがあります。インスタンス固有ブロックのパラメータ設定は、クラスタ全体設定を無効にします。次に例を示します。

```
max online engines = DEFAULT

[Instance:ase1]
max online engines = 5
[Instance:ase2]
max online engines = 3
```

『Cluster ユーザーズ・ガイド』を参照してください。

sp_configure 出力

次の出力例は、表示レベルを「**包括レベル**」に設定し、パラメータを指定しないで `sp_configure` を実行した場合に出力される情報を示しています。出力される値は、プラットフォームや既に変更している値によって異なります。

```
sp_configure
Group: Configuration Options
```

```
Group: Backup/Recovery
```

Parameter Name	Default	Memory Used	Config Value	Run Value	Unit	Type
allow remote access	1	0	1	1	switch	dyn
print recovery info	0	0	0	0	switch	dyn
recovery interval in m	5	0	5	5	minutes	dyn
...						

注意 表示レベルを「**包括レベル**」に設定している場合は、すべての設定グループとパラメータが出力されます。

上記のパラメータの意味は、次のとおりです。

- “Default” カラムにはデフォルト値が表示されます。パラメータを明示的に再設定しなければ、そのパラメータにはデフォルト値が使用されます。
- “Memory Used” カラムには現在の値のパラメータが使用しているメモリの量が、キロバイト単位で表示されます。関連するパラメータどうしが、同じメモリ・プールのメモリを使用することがあります。たとえば、`stack size` と `stack guard size` が使用するメモリは、`number of user connections` が使用するメモリの一部として含まれています。これらの各パラメータが使用するメモリを別々に加算すると、総計は実際に使用されているメモリ量よりも多くなります。他のパラメータとメモリを共有しているパラメータには、シャープ記号 (#) が付きます。
- “Config Value” カラムには、設定パラメータの最新の値が表示されます。`sp_configure` を実行して動的パラメータを変更すると、次のようになります。
 - 設定値と実行値が更新されます。
 - 設定ファイルが更新されます。
 - 変更はすぐに有効になる。

静的パラメータを変更すると、次のようになります。

- 設定値が更新されます。
- 設定ファイルが更新されます。
- 変更内容は、Adaptive Server を再起動しないと有効にならない。
- “Run Value” カラムには Adaptive Server が現在使用している値が表示されます。この値は、動的パラメータの場合は値を変更したときに変更されます。静的パラメータの場合は、Adaptive Server を再起動したときに変更されます。
- “Unit” カラムには設定パラメータの値の単位が表示されます。Adaptive Server の情報の表示に使用される単位は次のとおりです。

単位名	説明
number	項目数。
clock ticks	クロック・チック数。
microseconds	マイクロ秒数。
milliseconds	ミリ秒数。
seconds	秒数。
minutes	分数。
hours	時数。
bytes	バイト数。
days	日数。
kilobytes	キロバイト数。
megabytes	メガバイト数。
memory pages (2K)	2K のメモリ・ページ数。
virtual pages (2K)	2K の仮想ページ数。
logical pages	論理ページ数。この値は、サーバが使用している論理ページ・サイズ (2K、4K、8K、または 16K) によって異なる。
percent	パーセントとして設定されたパラメータの値。
ratio	比率として設定されたパラメータの値。
switch	TRUE (オン) または FALSE (オフ) として設定されているパラメータの値。
id	調査している設定パラメータの ID。
name	パラメータの実行値または設定値に割り当てられた文字列名。たとえば、sp_configure "lock scheme" の出力の “Run Value” カラムまたは “Config Value” カラムに “binary” という文字列が表示される。
row	ロー数。

- “Type” カラムには、設定オプションが静的と動的のどちらであるかが表示されます。静的パラメータへの変更を有効にするには、Adaptive Server を再起動する必要があります。動的パラメータへの変更はただちに反映されるため、Adaptive Server を再起動する必要はありません。

Named Cache 設定パラメータ

Named Cache 設定パラメータ・グループは、名前付きキャッシュの詳細を提示します。

- **cache size** – キャッシュのサイズ。デフォルトでは、Adaptive Server は 8MB キャッシュを作成します。sp_cacheconfig を使用してこのパラメータを動的に変更するか、サーバ設定ファイルの値を変更すると、サーバの再起動後に変更が反映されます。
- **cache status** – キャッシュのステータス。取り得る値は、default data cache、log only、mixed、in-memory storage のいずれかです。キャッシュのデフォルトのステータスは、default data cache でなければならず、これは変更できません。名前付きキャッシュの cache status は、log only か mixed であり、インメモリ・データベースでは in-memory storage (インメモリ・データベースの cache status は変更できません) です。

他のインスタンスが別の cache status を使用している間は、ローカル・キャッシュ上でクラスタード環境における cache status を log only から動的に変更できません。

- **cache replacement** – キャッシュ置換方式を記述する。名前付きキャッシュとデフォルトのデータ・キャッシュの場合、置換方式は strict LRU (ストリクト LRU) か relaxed LRU (リラククス LRU) です。sp_cacheconfig を使用してこのパラメータを動的に変更するか、サーバ設定ファイルの値を変更すると、サーバの再起動後に変更が反映されます。インメモリ・データベースの場合、キャッシュ置換方式は none (なし) でなければなりません。インメモリ・データベースでは、バッファもページ置換も使用しないからです。
- **local cache partition number** – キャッシュ・パーティション数。1つの名前付きキャッシュを複数のキャッシュ・パーティションに分割できます。使用できる値は 0、2、4、8、16、32、64、128 です。キャッシュ・パーティション数は動的に変更できません。変更を反映するには、Adaptive Server を再起動する必要があります。

sysconfigures テーブルと syscurconfigs テーブル

sp_configure によって表示されるレポートは、主に master..sysconfigures システム・テーブルと master..syscurconfigs システム・テーブルから生成され、sysattributes、sysdevices などのシステム・テーブルの追加情報が含まれます。

sysconfigures テーブルの value カラムには、sp_configure または設定ファイルによって設定した最終値が記録されます。syscurconfigs の value カラムには、現在使用されている値が格納されます。動的パラメータの場合、2つの値は一致します。静的パラメータは、サーバを再起動するまで有効にならないため、Adaptive Server の起動後に値を変更した場合は、2つの値は異なります。デフォルト値を使用している場合も値が異なることがあります。この場合、sysconfigures には 0 が格納され、syscurconfigs には Adaptive Server が計算して使用している値が格納されます。

sp_configure は、sp_configure によってレポートされた値を表示するために、sysconfigures と syscurconfigs でジョインを実行します。

syscurconfigs と sysconfigures へのクエリ (例)

sysconfigures と syscurconfigs に対してクエリを実行し、独自に編成した情報を取得できます。たとえば、引数を指定しないで sp_configure を実行すると、設定パラメータに使用されているメモリはリストされますが、最小値と最大値はリストされません。次のクエリを使用して、最大値、最小値、デフォルト値以外に、現在のメモリ使用量もすべて記載した一覧を取得します。

```
select b.name, memory_used, minimum_value, maximum_value,
       defvalue
from master.dbo.sysconfigures b,
     master.dbo.syscurconfigs c
where b.config *= c.config and parent != 19
and b.config > 100
```

設定パラメータ

多くの場合、設定パラメータの最大許容値は、sp_configure の制限値ではなく使用できるメモリによって制限されます。

注意 プラットフォームおよび Adaptive Server のバージョンに応じた設定可能な最大値については、使用しているプラットフォームの『インストール・ガイド』の「Adaptive Server の仕様」を参照してください。

設定パラメータのアルファベット順リスト

それぞれの設定パラメータの要約と詳細について、以降の項で説明します。

abstract plan cache

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

abstract plan cache パラメータは、抽象プラン・ハッシュ・キーのキャッシングを有効にします。『パフォーマンス&チューニング・シリーズ：クエリ処理と抽象プラン』の「第 12 章 抽象プランの作成と使用」を参照してください。プランのキャッシングを有効にするには、**abstract plan load** を有効にする必要があります。

abstract plan dump

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

abstract plan dump パラメータは、**ap_stdout** 抽象プラン・グループへの抽象プランの保存を有効にします。詳細については、『パフォーマンス&チューニング・シリーズ：クエリ処理と抽象プラン』の「第 12 章 抽象プランの作成と使用」を参照してください。

abstract plan load

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

abstract plan load パラメータは、**ap_stdin** 抽象プラン・グループの抽象プランとクエリとの関連付けを有効にします。詳細については、『パフォーマンス&チューニング・シリーズ：クエリ処理と抽象プラン』の「第 12 章 抽象プランの作成と使用」を参照してください。

abstract plan replace

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

abstract plan replace パラメータは、**ap_stdout** 抽象プラン・グループの抽象プランのプラン置換を有効にします。『パフォーマンス&チューニング・シリーズ：クエリ処理と抽象プラン』の「第 12 章 抽象プランの作成と使用」を参照してください。**replace** モードを有効にするには、**abstract plan load** を有効にする必要があります。

additional network memory

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、ネットワーク通信、物理メモリ

additional network memory は、ネットワーク・パケットのサイズがデフォルトのパケット・サイズより大きい場合に使用される追加メモリの最大サイズを設定します。入力した値は、2K の倍数になるように切り捨てられます。デフォルト値は、大きなパケット用の追加領域は割り付けられていないことを意味します。

ログイン時にデフォルトよりも大きなパケット・サイズの要求があった場合、Adaptive Server は、要求に応えることが可能なメモリを使用できるかどうかを調べます。メモリが不足している場合は、使用可能なメモリの中から最大サイズのブロックが検出され、最大メモリ・ブロックを下回る適切なサイズ (**default network packet size** の倍数) のメモリを使用できるかどうかを検証されます。使用できない場合は、要求から **default network packet size** 分のメモリが差し引かれ、その値に相当するメモリが使用できるかどうかを確認されます。メモリが使用可能になるか、引き算の結果が **default network packet size** と同じになるまで、この引き算処理が最大 10 回繰り返されます。引き算処理が 10 回に達すると、**default network packet size** の値がパケット・サイズとして使用されます。

max network packet size の値を増やす場合、**additional network memory** の値も増やす必要があります。割り付けられたネットワーク・メモリはすべて、デフォルト・サイズでユーザ用に予約されているからです。デフォルトのパケット・サイズであれば、すべてのユーザ接続が Adaptive Server に確実にログインできます。

max network packet size を増やしたが、**additional network memory** を増やさなかった場合、デフォルト・サイズより大きいネットワーク・パケット・サイズを要求するクライアントは、必ずしも要求したパケット・サイズでログインできるとは限りません。

additional network memory を増やすと、大量のデータを転送するアプリケーションの場合はパフォーマンスを改善できることがあります。アプリケーションが大きなパケット・サイズを使用する場合に **additional network memory** の値を決めるには、次の手順に従います。

- 1 大きなパケット・サイズを要求する同時ユーザの数と、そのアプリケーションが要求するサイズを見積もります。
- 2 それぞれの接続は 3 つのバッファを必要とするので、この合計を 3 倍します。
- 3 32 ビット・サーバでは 2 パーセント、64 ビット・サーバでは 4 パーセントの値をオーバーヘッドとして加算します。
- 4 値が 2048 の倍数になるように切り上げます。

次の例は、複数のアプリケーションが大きいパケット・サイズを必要とする場合の見積もりを示します。

アプリケーション	パケット・サイズ	オーバーヘッド
bcp	8192	
Client-Library	8192	
Client-Library	4096	
Client-Library	4096	
合計	24576	
3 パッファ／ユーザを乗算	* 3=73728	
2% のオーバーヘッドを計算		* .02=1474
オーバーヘッドを加算	+ 1474	
追加のネットワーク・メモリ	75202	
2,048 の倍数になるように切り上げる	75776	

additional network memory を 75,776 バイトに設定します。

allocate max shared memory

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、物理メモリ

allocate max shared memory は、Adaptive Server の起動時に **max memory** で指定されたメモリをすべて割り付けるか、設定パラメータに必要な量のメモリだけを割り付けるかを指定します。

allocate max shared memory を 0 に設定すると、Adaptive Server が使用する共有メモリの量は現在の設定に必要な量だけとなり、起動時に割り付けられるのは設定パラメータが必要とする量だけ、つまり **max memory** の値よりは少なくなります。

allocate max shared memory を 1 に設定すると、**max memory** に指定された量のメモリがすべて Adaptive Server の起動時に割り付けられます。**allocate max shared memory** を 1 に設定し、**max memory** を増やすと、Adaptive Server はメモリをすぐに割り付けようとします。メモリの割り付けに失敗した場合、メッセージがエラー・ログに書き込まれます。エラー・ログをチェックして、エラーが何も発生していないことを確認してください。

メモリ割り付けに成功した場合、Adaptive Server はメモリ設定がどのように変更されても必要なメモリを常に確保します。また、サーバがメモリ追加の調整を行っている間にパフォーマンスが低下することはありません。ただし、メモリの増加量を正確に予測できない場合、**max memory** を大きな値に設定すると、物理メモリを浪費するおそれがあります。

allow backward scans

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	クエリ・チューニング

allow backward scans パラメータは、**order by...desc** コマンドが含まれる **select** クエリをオプティマイザが実行する方法を制御します。

- この値が 1 に設定されているときは、オプティマイザはページ・チェーンをインデックスの降順に検索してインデックスまたはテーブル・ローにアクセスできます。
- この値が 0 に設定されているときは、オプティマイザはインデックス・ページ・ポインタを昇順にたどりながらローを選択してワークテーブルに入れ、このワークテーブルを降順でソートします。

最初の方法、つまり後方スキャンを使用すれば、結果をカラム値の降順で並べる必要がある場合にテーブルへのアクセスを高速化できます。ただし、アプリケーションによっては、後方スキャンによるデッドロックが発生することがあります。特に、同じインデックスを使用して前方スキャンを行う **delete**、または **update** クエリがある場合は、デッドロックが増加しているかどうかを調べてください。また、インデックス内のページ分割によるデッドロックが発生することもあります。

print deadlock information パラメータを使用して、デッドロックについてのメッセージをエラー・ログに送信します。「[print deadlock information](#)」(225 ページ)を参照してください。または、システム・プロシージャ **sp_sysmon** を使用してデッドロックがあるかどうかを確認します。詳細については、『パフォーマンス&チューニング・シリーズ:ロックと同時実行制御』を参照してください。

allow nested triggers

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

allow nested triggers パラメータは、ネストされたトリガの使用を制御します。値を 1 に設定すると、トリガによるデータ変更で別のトリガを起動することができます。ネストされたトリガを使用できないようにするには、**allow nested triggers** を 0 に設定します。**set** の **self_recursion** オプションは、トリガによる変更で、そのトリガを再び起動できるようにするかどうかを制御します。

allow procedure grouping

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

allow procedure grouping パラメータは、同じ名前のストアド・プロシージャを 1 つの **drop procedure** 文で削除できるようにグループ化する機能を制御します。

allow remote access

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	バックアップとリカバリ、ネットワーク通信

allow remote access は、リモートの Adaptive Server からのログインを制御します。デフォルト値は 1 で、このとき Adaptive Server は Backup Server と通信できます。

注意 この値を 0 に設定すると、サーバ間のリモート・プロシージャ・コール (RPC) は使用できなくなります。Adaptive Server は RPC を使用して Backup Server と通信するので、このパラメータを 0 に設定すると、データベースのバックアップが実行できなくなります。

Backup Server 以外のリモート・サーバが RPC を実行できるようにするには、この他にもシステム管理作業が必要であるので、このオプションの設定を 1 のままにしておいてもセキュリティ上の危険はありません。

allow resource limits

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、SQL Server 管理

allow resource limits パラメータは、リソース制限の使用を制御します。値を 1 に設定すると、サーバは時間範囲、リソース制限、内部サーバ・アラーム用に内部メモリを割り付けます。また、サーバは、ユーザ・セッションに対して適用可能な範囲と制限を内部的に割り当てます。**showplan** と **statistics io** の出力には、オブティマイザによるクエリの見積もりコストが表示されます。リソース制限をすべて無効にするには、**allow resource limits** を 0 に設定してください。

allow sendmsg

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム・セキュリティ担当者
設定グループ	ネットワーク通信

`allow sendmsg` は、Adaptive Server から UDP (User Datagram Protocol) ポートへのメッセージ送信を有効または無効にします。`allow sendmsg` を 1 に設定すると、すべてのユーザが `sp_sendmsg` または `syb_sendmsg` を使用してメッセージを送信できます。Adaptive Server が使用するポート番号の設定については、「`syb_sendmsg port number`」(258 ページ)を参照してください。

注意 UDP ポートへのメッセージ送信は Windows ではサポートされていません。

allow sql server async i/o

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ディスク I/O

`allow sql server async i/o` は、Adaptive Server が非同期ディスク I/O で実行できるようにします。非同期ディスク I/O を使用するには、Adaptive Server とオペレーティング・システムの両方で設定を有効にする必要があります。オペレーティング・システム・レベルで非同期 I/O を使用可能にするための情報については、それぞれのオペレーティング・システムのマニュアルを参照してください。

ディスク I/O は同期式よりも非同期式の方が必ず高速に実行されます。これは、Adaptive Server が非同期 I/O を発行するときに、応答を待たずに次の I/O を発行できるためです。

allow updates to system tables

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

allow updates to system tables パラメータは、システム管理者の役割を持つユーザがシステム・テーブルを変更したり、システム・テーブルを変更できるストアド・プロシージャを作成したりできるようにします。**allow updates to system tables** が有効になっている場合、データベース管理者は自分が所有するすべてのテーブル内のシステム・テーブルを更新できます。

システム・テーブルには次のものがあります。

- **master** データベース内の Sybase が提供するすべてのテーブル
- 名前が “sys” で始まり、**sysobjects** テーブルでの ID 値が 100 以下である、ユーザ・データベース内のすべてのテーブル

警告！ システム・テーブルに対する変更が正しくない場合は、データベースが破壊されてデータが消失することがあります。システム・テーブルを変更する場合は、データベースを破壊させる可能性があるエラーから保護するために、常に **begin transaction** を使用してください。変更を終了したら、すぐに **allow updates to system tables** を無効にしてください。

allow updates to system tables パラメータがオンに設定されている間に作成されたストアド・プロシージャとトリガは、このパラメータをオフにした後でも、システム・テーブルを更新できます。**allow updates to system tables** をオンに設定すると、その間はユーザがシステム・テーブルを変更したり、後でシステム・テーブルの変更に使用できるストアド・プロシージャを作成したりできるので、「脆弱な時間帯」を作り出すことになります。

システム・テーブルは非常に重要なので、十分に制御された状況下以外ではこのパラメータをオンにしないことをおすすめします。システム・テーブルの直接更新が可能である間は Adaptive Server に他のユーザが一切アクセスできないようにするために、Adaptive Server をシングルユーザ・モードで再起動します。詳細については、『ユーティリティ・ガイド』の **startserver** と **dataserver** の説明を参照してください。

注意 サーバ全体の設定オプション **allow updates to system tables** は、**allow updates to system tables** のストアド・プロシージャ設定よりも優先されます。サーバ・レベルで **allow updates to system tables** を有効にしていない場合、システム・カタログを変更できるかどうかは、個々のストアド・プロシージャ設定によって決まります。

average cap size

要約	
デフォルト値	200
値の範囲	100 ~ 10000
ステータス	静的
表示レベル	
必要な役割	
設定グループ	診断

今後のために予約済み。

audit queue size

要約	
デフォルト値	100
値の範囲	1 ~ 65535
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	メモリ使用、セキュリティ関連

メモリ内の監査キューは、ユーザ処理によって生成された監査レコードの処理と監査証跡への書き込みが可能になるまで、そのレコードを保持します。システム・セキュリティ担当者は、**audit queue size** を使用して監査キューのサイズを変更できます。キューのサイズを設定するときは、パフォーマンスとリスクの間のトレードオフがあります。キューが大きすぎると、レコードが長い間キューの中にとどまることがあります。レコードがキューの中にある間は、システムに障害が発生した場合に消失する危険があります。しかし、キューが小さすぎるとすぐに空きがなくなり、システム全体のパフォーマンスが低下します。監査キューの空きがなくなると、監査レコードを生成するユーザの処理はスリープします。

監査キューの大きさを決定するためのガイドラインには、次のようなものがあります。また、実行する監査の量も考慮する必要があります。

- 1つの監査レコードに必要なメモリは424バイトですが、データ・ページに書き込まれるときのレコードは22バイトまで小さくできます。
- システム障害で消失する監査レコードの最大数は、監査キューのサイズ(レコード単位)に20を加えた値です。監査キューから取り除かれたレコードは、ディスク上の現在の監査テーブルに書き込まれるまでバッファ・ページに残ります。ページは20レコードごとにディスクにフラッシュされます(監査プロセスがビジー状態になることが少なければ、20レコード未満でもフラッシュされます)。

- システム監査テーブル内の **extrainfo** フィールドと名前用のフィールドは可変長であるため、名前情報をすべて記録する監査レコードは一般に大きくなります。

1 ページに収まる監査レコードの数は、4 から 80 程度までの間で変化します。デフォルトの監査キュー・サイズ 100 に対するメモリ必要量は約 42K です。

auditing

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

auditing パラメータは、Adaptive Server の監査を有効または無効にします。

automatic cluster takeover

要約	
デフォルト値	1
有効な値	1 (有効)、0 (無効)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	共有ディスク・クラスタ

automatic cluster takeover を 1 に設定すると、起動中のインスタンスが突然の全体的クラスタ障害から自動的にリカバリできます。**automatic cluster takeover** を 0 に設定すると、**--cluster_takeover** パラメータを含めないかぎり、クラスタは突然のクラスタ・フェールオーバーからリカバリできないことがあります。

Cluster Edition は、クォーラム・ハートビートとクラスタ・テイクオーバー・アルゴリズムを使用して、クラスタ・テイクオーバーをいつ実行すべきかを判断します。このアルゴリズムでは、起動中のインスタンスがクラスタにジョインできないのは、クラスタがクラッシュしたため (この場合にはテイクオーバーが適切) なのか、それともネットワーク接続がないため (この場合にはテイクオーバーは不適切) なのかを区別することができます。

automatic cluster takeover が無効にされる (0 に設定) と、Cluster Edition はアルゴリズムの結果をアドバイス・メッセージとしてエラー・ログに書き込んでから、終了します。

automatic cluster takeover が有効にされる (1 に設定) と、Cluster Edition はクラスタ・コーディネータとして起動し、データベースをリカバリします。これは、I/O フェンシングが有効に設定された環境で安全な操作であることが保証されています。

I/O フェンシングがない環境では、アルゴリズムに不具合があるとデータ破損が発生する可能性があるため、この設定パラメータを 0 に設定してこのアルゴリズムを無効に設定することをおすすめします。ただし、I/O フェンシングがない環境はデータ破損の危険性があるので、自動クラスタ・テイクオーバー機能を無効にしても、この危険性がすべて低減されるわけではありません。

builtin date strings

要約	
デフォルト値	0
値の範囲	0 ~ 1
ステータス	動的
表示レベル	
必要な役割	
設定グループ	クエリ・チューニング

日付順の値ではなく文字列が引数として指定された場合、サーバは示された精度にかかわらず、その文字列を **datetime** 値として解釈します。デフォルトの動作は、設定パラメータ **builtin date strings** または設定オプション **builtin_date_strings** を設定することで変更できます。これらのオプションを設定すると、サーバは日付順の組み込みに提示された文字列を **bigdatetime** として解釈します。

caps per ccb

要約	
デフォルト値	50
値の範囲	5 ~ 50
ステータス	静的
表示レベル	
必要な役割	
設定グループ	診断

今後のために予約済み。

capture compression statistics**要約**

デフォルト値	0 (オフ)
値の範囲	1 (オン)、0 (オフ)
ステータス	動的
表示レベル	10
必要な役割	システム管理者
設定グループ	SQL Server 管理

monTableCompression モニタリング・テーブルが圧縮統計の取得を開始できるように設定します。

check password for digit**要約**

デフォルト値	0 (オフ)
値の範囲	1 (オン)、0 (オフ)
ステータス	動的
表示レベル	10
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

システム・セキュリティ担当者は、サーバワイドの設定パラメータ **check password for digit** を使用して、パスワードに数字が 1 字以上含まれていることをチェックするようにサーバに指示することができます。このパラメータを設定しても、既存のパスワードに影響を与えることはありません。

CIPC large message pool size**要約**

デフォルト値	512
有効な値	512 ~ 2147483647
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	共有ディスク・クラスタ

CIPC large message pool size は、起動時に CIPC によって割り付けられる大きいメッセージのバッファの数を指定します。

CIPC regular message pool size

要約	
デフォルト値	8192
有効な値	2048 ~ 2147483647
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	共有ディスク・クラスタ

CIPC regular message pool size は、起動時に CIPC によって割り付けられる通常サイズのメッセージのバッファの数を指定します。

cis bulk insert array size

要約	
デフォルト値	50
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	コンポーネント統合サービス

Adaptive Server から別の Adaptive Server へのデータのバルク転送を実行するとき、CIS はローを内部的にバッファし、Open Client バルク・ライブラリに対してバッファ内のローを 1 つのブロックとして転送するように要求します。配列のサイズは `cis bulk insert array size` で設定します。

cis bulk insert batch size

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	コンポーネント統合サービス

`cis bulk insert batch size` は、`select into` を使用してソース・テーブルからターゲット・テーブルに 1 つのバッチとしてバルク・コピーされるローの数を決定します。

`cis bulk insert batch size` を 0 のままにしておくと、すべてのローが 1 つのバッチとしてコピーされます。0 以外の場合、このパラメータに指定した数のローがターゲット・テーブルにコピーされた後、サーバがターゲット・サーバにバルク・コミットを発行することにより、バッチがコミットされます。

クライアントが生成する通常バルク・コピー操作 (`bcp` ユーティリティで生成されるような)を受け取ったときは、クライアントがバルク・バッチのサイズを制御するものと見なされ、サーバはこの設定パラメータの値を無視します。

cis connect timeout

要約	
デフォルト値	0
値の範囲	0 ~ 32767
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	コンポーネント統合サービス

`cis connect timeout` は、Client-Library 接続が正常に完了するまでの待ち時間を秒単位で定義します。

cis cursor rows

要約	
デフォルト値	50
値の範囲	1 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	コンポーネント統合サービス

`cis cursor rows` は、`cursor open` 操作と `cursor fetch` 操作でのカーソル・ロー・カウントを指定します。この値を増やすと、1 つの操作でより多くのローがフェッチされるようになります。これによって処理速度は速くなりますが、メモリがより多く必要になります。

cis idle connection timeout

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	コンポーネント統合サービス

cis idle connection timeout は、指定された秒数よりも長い時間未使用になっているリモート・サーバへの CIS 接続がないか確認するように Adaptive Server を設定します。Adaptive Server は未使用の接続を削除し、それらのリソースを再割り付けします。

指定する数値は秒単位ですが、ハウスキーピング・タスクは 1 分間に多くても 1 回しかウェイクアップしないので、アイドル接続は設定された値よりもかなり長くアイドル状態になる場合があります。トランザクションが接続でアクティブな場合、アイドル接続は削除されず、ユーザが接続にアクセスするためにコマンドを実行すると自動的に接続が再確立されます。

cis packet size

要約	
デフォルト値	512
値の範囲	512 ~ 32768
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	コンポーネント統合サービス

cis packet size は、サーバとリモート・サーバの間で接続開始時に交換される TDS (Tabular Data Stream™) パケットのサイズを指定します。

ほとんどのシステムではデフォルトのパケット・サイズは 512 バイトであり、これはほとんどのアプリケーションに十分な値です。ただし、特に **text**、**unitext**、**image** データまたはバルク・データが関係する場合は、パケット・サイズをこれより大きくするとクエリのパフォーマンスが大幅に向上することがあります。

デフォルトよりも大きいパケット・サイズを指定する場合は、ターゲット・サーバで可変長のパケット・サイズを処理できるように設定します。次のとおりにします。

- additional netmem
- maximum network packet size

cis rpc handling

要約	
デフォルト値	0 (オフ)、Cluster Edition では 1
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	コンポーネント統合サービス

cis rpc handling は、リモート・プロシージャ・コール (RPC) のデフォルトの処理方法を指定します。**cis rpc handling** を 0 に設定すると、Adaptive Server のサイト・ハンドラがデフォルトの RPC 処理メカニズムとして設定されます。このパラメータを 1 に設定すると、RPC 処理にはコンポーネント統合サービスのアクセス・メソッドが使用されます。『コンポーネント統合サービス・ユーザーズ・ガイド』の **cis rpc handling** の設定の説明を参照してください。

cluster heartbeat interval

要約	
デフォルト値	10
有効な値	1 ~ 127
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	共有ディスク・クラスタ

cluster heartbeat interval は、クラスタ・インスタンスがハートビート・ステータスの送信とチェックに使用する間隔を制御します。

cluster heartbeat interval に低い値を指定すると、障害検出時間が短縮されますが、一時的な問題 (CPU のオーバーロードなど) が原因で誤検出の危険性が高まります。**cluster heartbeat interval** を高い値に調整すると、誤検出の危険性は低下しますが、障害の検出に要する時間が長くなります。

cluster heartbeat retries

要約	
デフォルト値	1
有効な値	1 ~ 127
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	共有ディスク・クラスタ

cluster heartbeat retries は、インスタンスが障害モードに入るまでに、失敗したクラスタ・ハートビートを再試行する回数を制御します。

cluster heartbeat retries を低い値に調整すると、障害検出時間が短縮されますが、一時的な問題 (CPU のオーバーロードなど) が原因で誤検出の危険性が高まります。cluster heartbeat retries を高い値に調整すると、誤検出の危険性は低下しますが、障害の検出に要する時間が長くなります。

cluster redundancy level

要約	
デフォルト値	1
有効な値	1 ~ n - 1。ここで n は cluster.cfg またはクォーラム・ファイルで指定されるインスタンスの最大数。
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	共有ディスク・クラスタ

cluster redundancy level (CRL) は、共有ディスク・クラスタ内のエラーが発生したインスタンスのリカバリ可能な数を制御します。これは、同時に障害が発生しても許容されるインスタンスの最大数であり、この数値内であれば、リカバリして他のアクティビティを同時に続行できます。エラーが発生したインスタンスの数が CRL によって指定されている最大数を超えると、クラスタはシャット・ダウンします。

クラスタが起動するためには、cluster redundancy level の値が cluster.cfg またはクォーラム・ファイルで指定されているインスタンスの最大数より少なくとも 1 少ない値である必要があります。そのため、次のいずれかに設定した場合は、クラスタが起動できません。

- cluster redundancy level の値と等しいかそれより小さい値のインスタンスの最大数。
- インスタンスの最大数に等しいかまたはそれより大きい cluster redundancy level の値。

`cluster redundancy level` 値が増加するとクラスタ内に複数のロックのコピーが存在することになり、メッセージング・トラフィックが増え、この冗長レベルを維持するために必要なオーバーヘッドも増加します。

`cluster redundancy level` が 1 より大きい場合は、`number of locks` や `cache size` などの他の設定パラメータにもより多くのリソースが必要になります。これはつまり、`number of locks` の値が同じでも、非クラスタ環境よりも `max memory` を増やす必要があるということです。

複数のインスタンス・エラーが頻発しない限りは、`cluster redundancy level` を 1 に設定することをおすすめします。

cluster vote timeout

要約	
デフォルト値	60
有効な値	1 ~ 127
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	共有ディスク・クラスタ

`cluster vote timeout` は、あるインスタンスが投票期間中に他のインスタンスが投票するのを待つ最長時間を制御します。インスタンスが待つのは、稼働中だと思われるインスタンスだけです。

`cluster vote timeout` を低い値に調整すると、フェールオーバー時間が短縮されますが、稼働中のインスタンスが新しいクラスタ・ビューから除外されてしまう危険性が高まります。`cluster vote timeout` を高い値に調整すると、稼働中のインスタンスが新しいクラスタ・ビューから除外されてしまう危険性は低下しますが、フェールオーバー時間が長くなる可能性があります。

column default cache size

要約	
デフォルト値	1024 ページ (2MB)
値の範囲	128 ~ 8192 (ページ)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ユーザ環境

`column default cache size` は、マテリアライズされていないカラム用のデフォルトを提供するために Adaptive Server がメモリ内に確保する必要のあるキャッシュ・サイズを決定します。

compression info pool size

要約	
デフォルト値	4096
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	10
必要な役割	システム管理者
設定グループ	SQL Server 管理

圧縮に使用するメモリ・プールのサイズを決定します。

compression memory size

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	物理メモリ

圧縮ダンプをアーカイブ・データベースにロードする間に使用されます。**compression memory size** は、Adaptive Server が圧縮ダンプを圧縮解除するために使用するメモリ・プールのサイズ (2KB ページ単位) を決定します。**compression memory size** が 0 に設定された場合、プールは作成されず、圧縮ダンプをロードできません。

『システム管理ガイド 第 2 巻』の「第 14 章 アーカイブ・データベースへのアクセス」の「圧縮メモリ・プールの作成」を参照してください。

configuration file

要約	
デフォルト値	0 (オフ)
値の範囲	0、verify、read、write、restore のいずれか
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	一般情報

configuration file は、現在使用中の設定ファイルのロケーションを指定します。構成ファイルの詳細については、「[設定ファイルを指定して sp_configure を使用する方法](#)」(63 ページ)を参照してください。

sp_configure で出力される “Run Value” カラムには 10 文字しか表示されません。そのため、設定ファイルのパスと名前を完全な形では表示できない場合があります。

cost of a logical io

要約	
デフォルト値	2
値の範囲	0 ~ 254
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

cost of a logical io は、1 つの論理 I/O のコストを指定します。

cost of a physical io

要約	
デフォルト値	25
値の範囲	0 ~ 254
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

cost of a physical io は、1 つの物理 I/O のコストを指定します。

cost of a cpu unit

要約	
デフォルト値	1000
値の範囲	1 ~ 65534
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

`cost of a cpu unit` は、1 つの CPU 処理のコストを指定します。

オプティマイザの逐次プランのコストを求める公式は、次のとおりです。

$$\text{Cost} = \text{PIO} \times \text{estimated_pio} + \text{LIO} \times \text{estimated_lio} + 100 \times \text{estimated_cpu} / \text{CPU}$$

デフォルト値は次のとおりです。

- `estimated_pio` = 25
- `estimated_lio` = 2
- `estimated_cpu` = 1000

Adaptive Server にメモリが十分にある場合、テーブルはすべてメモリ内に存在します。`cost of a physical io` の値は 0 が適切です。

CPU が十分に高速であるため `cost of a cpu unit` の値が問題にならない場合には、この公式を使用して CPU のコストを求めます。これは、2 LIO と 25 PIO (いずれもデフォルト値) を組み合わせたものです。

$$\text{CPU} \times 100 / \text{configuration_value}$$

`configuration_value` のデフォルト値は 1000 です。

`cost of a cpu unit` の値を増やすにつれて、CPU がコストに及ぼす影響は小さくなっていきます。

cpu accounting flush interval

要約	
デフォルト値	200
値の範囲	1 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

`cpu accounting flush interval` パラメータは、Adaptive Server が各ユーザの CPU 使用統計を `sysprocesses` から `syslogins` にフラッシュするまでの待機時間を、マシンのクロック・チック (Adaptive Server のクロック・チックではない) 単位で指定します。これは、チャージバック アカウンティングで使用される手順です。

ユーザが Adaptive Server にログインすると、その時点以降のそのユーザ・プロセスによる CPU 使用量が `sysprocesses` に蓄積されます。ユーザが Adaptive Server からログオフするか、`cpu accounting flush interval` の時間が過ぎると、蓄積された CPU 使用統計情報は `sysprocesses` から `syslogins` にフラッシュされます。この統計情報は、合計がクリアされるまでは、`syslogins` に引き続き蓄積されます。`syslogins` から現在の合計を表示するには、`sp_reportstats` を使用します。

`cpu accounting flush interval` に設定する値は、目的とするレポートのタイプによって異なります。月単位でレポートを作成する場合は、`cpu accounting flush interval` には比較的大きな値を設定します。レポート処理の頻度が低ければ、`syslogins` 内のデータを頻繁に更新することはそれほど重要ではありません。

ただし、プロセスによる CPU の使用量を調べるために、アドホック・クエリを使用して `syslogins` の `totcpu` カラムからの選択を定期的に行う場合は、`cpu accounting flush interval` の設定値を小さくします。このようにすれば、選択を実行するときに `syslogins` 内のデータが最新のものである可能性が高くなります。

`cpu accounting flush interval` の設定値を小さくすると、プロセスがデッドロックのピクティムの候補であるとロック・マネージャが誤って判断することがあります。ロック・マネージャは、デッドロックを検出すると、競合するプロセスのそれぞれによって蓄積された CPU 使用時間をチェックします。この値が小さい方のプロセスがデッドロック・ピクティムとして選択され、ロック・マネージャによって終了させられます。また、`cpu accounting flush interval` の設定値が小さいと、プロセスの CPU 使用情報を保管するタスク・ハンドラが初期化される頻度が上がるので、プロセスが実際に使用した CPU 時間よりも蓄積された時間が少ないように認識されることがあります。このため、実際にはプロセスが蓄積した CPU 使用時間が競合するプロセスよりも多いにもかかわらず、ロック・マネージャがそのプロセスをデッドロック・ピクティムとして選択することがあります。

CPU 使用時間のレポートがまったく必要ない場合は、`cpu accounting flush interval` を最大値に設定してください。これにより、`syslogins` が更新される回数と、そのページをディスクに書き込まなければならない回数が減ります。

cpu grace time

要約	
デフォルト値	500
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

`cpu grace time` パラメータは、`time slice` パラメータとともに使用します。この時間が経過するまではユーザ・プロセスは CPU を解放することなく実行できますが、この値を超えると、Adaptive Server はそのプロセスの制御を横取りしてタイムスライス・エラーで終了させます。`cpu grace time` の単位は、`sql server clock tick length` で定義されているタイム・チックです。「[sql server clock tick length](#)」(247 ページ)を参照してください。

プロセスの実行時間が **cpu grace time** を超過すると、Adaptive Server は内部キューからプロセスを取り除くことによってそのプロセスに「影響を及ぼし」ます。プロセスは強制終了されますが、Adaptive Server は影響を受けません。こうして、プロセスの暴走によって CPU が独占されるのを防ぎます。ユーザ・プロセスの中断が発生した場合は、**cpu grace time** の値を増やすことによって一時的にこの問題を避けることができます。ただし、問題が本当に暴走しているプロセスによるものではなく、現在の **cpu grace time** の範囲内で完了できないプロセスのためであることを確認する必要があります。

cpu grace time を一時的に増やすことは対処方法ではありますが、他の問題を引き起こす可能性があるため、永久的な解決策ではありません。これについては、「[time slice](#)」(262 ページ) を参照してください。また、タスク・スケジューリングの詳細については、『パフォーマンス&チューニング・シリーズ：基本』の「第 4 章 エンジンと CPU の使用方法」を参照してください。

current audit table

要約	
デフォルト値	1
値の範囲	0 ~ 8
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

current audit table パラメータは、Adaptive Server が監査ローを書き込むテーブルを設定します。システム・セキュリティ担当者は、次の構文を使用して現在の監査テーブルを変更できます。

```
sp_configure "current audit table", n
[, "with truncate"]
```

ここで **n** は、次に示すように現在の新しい監査テーブルを決定する整数値です。

- 1 は **sysaudits_01**、2 は **sysaudits_02** を意味し、最大 8 まで設定できます。
- 0 は、次のテーブルを現在の監査テーブルとするように Adaptive Server に指示します。たとえば、インストール環境に 3 つの監査テーブル **sysaudits_01**、**sysaudits_02**、**sysaudits_03** がある場合、現在の監査テーブルは次のように設定されます。
 - 現在の監査テーブルが **sysaudits_01** の場合は 2
 - 現在の監査テーブルが **sysaudits_02** の場合は 3
 - 現在の監査テーブルが **sysaudits_03** の場合は 1

"with truncate" は、新しいテーブルが空でない場合に、そのテーブルをトランケートすることを指定します。このオプションが指定されていないときに、テーブルが空でなければ、`sp_configure` コマンドは失敗します。

注意 Adaptive Server が現在の監査テーブルをトランケートしたときに、データがアーカイブ済みでなければ、そのテーブルの監査レコードは失われます。監査データがアーカイブされていることを確認してから、`with truncate` オプションを使用してください。

`sp_configure` を実行して現在の監査テーブルを変更するには、`sso_role` をアクティブにしてください。スレッシュホールド・プロシージャを作成して、現在の監査テーブルを自動的に変更することもできます。

deadlock checking period

要約	
デフォルト値	500
値の範囲	0 ~ 2147483
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ロック・マネージャ

`deadlock checking period` パラメータは、ロックの解放を待っているプロセスに対して Adaptive Server がデッドロックのチェックを開始するまでの最小時間を、ミリ秒単位で指定します。このデッドロックのチェックは、デッドロックがまったく発生しないか、少しだけ発生するアプリケーションにとっては時間のかかるオーバーヘッドであり、ロックを待つ必要があるロック要求の割合が増えるに従って、オーバーヘッドが増加します。

`deadlock checking period` を 0 以外の値 (n) に設定すると、プロセスの待ち時間が n ミリ秒以上となったときにデッドロックのチェックが開始します。たとえば、次のように入力することで、プロセスがロックを待つ時間が 700 ミリ秒に達してからデッドロックがチェックされるように設定できます。

```
sp_configure "deadlock checking period", 700
```

`deadlock checking period` パラメータを 0 に設定すると、それぞれのプロセスがロック待ち状態となると同時にデッドロックのチェックが開始します。クロック・チックのミリ秒数より小さい値はすべて 0 と見なされます。「[sql server clock tick length](#)」(247 ページ)を参照してください。

`deadlock checking period` を大きな値に設定すると、デッドロックが検出されるまでの時間が長くなります。ただし、設定された時間が経過する前にほとんどのロック要求が受け入れられるので、それらのロック要求に対するデッドロックのチェックのためのオーバーヘッドは回避されます。アプリケーションでのデッドロックの頻度が低い場合は、`deadlock checking period` を高い値に設定します。それ以外の場合は、デフォルト値の 500 ミリ秒で十分です。

使用しているシステムにおけるデッドロックの頻度と `deadlock checking period` パラメータの最適な設定を判断するには、`sp_sysmon` を使用してください。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

deadlock pipe active

要約	
デフォルト値	0
値の範囲	0 ~ 1
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、モニタリング

`deadlock pipe active` は、Adaptive Server でデッドロック・メッセージを収集するかどうかを制御します。`deadlock pipe active` と `deadlock pipe max messages` を両方とも有効にすると、Adaptive Server は各デッドロックのテキストを収集します。収集されたデッドロック・メッセージは、`monDeadLock` を使用して取得できます。

deadlock pipe max messages

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	モニタリング

`deadlock pipe max messages` は、Adaptive Server が格納するデッドロック・メッセージ数をエンジンごとに決定します。`monSQLText` テーブル内のメッセージ数の合計は、`sql text pipe max messages` に実行中のエンジン数を掛け合わせた値になります。

deadlock retries

要約	
デフォルト値	5
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ロック・マネージャ、SQL Server 管理

deadlock retries パラメータは、インデックスのページ分割または縮小中にデッドロックが発生した場合にトランザクションがロックの取得を試行できる回数を指定します。

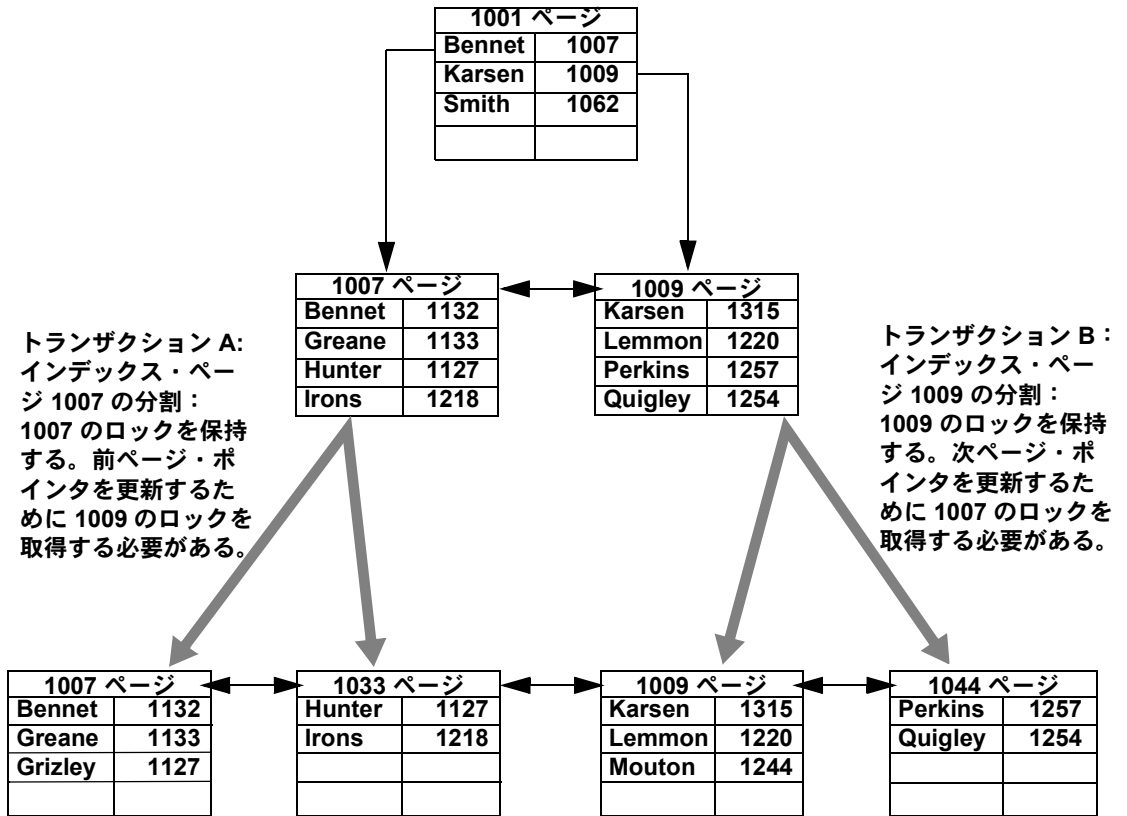
たとえば、[図 5-1](#) は次の状況を示しています。

- トランザクション A はページ 1007 をロックします。また、ページ分割用のページ・ポインタを更新するために、ページ 1009 のロックを取得する必要があります。
- トランザクション B もインデックス・ローを挿入するのでページ分割が発生し、ページ 1009 をロックします。また、ページ 1007 のロックを取得する必要があります。

この状況では、Adaptive Server はデッドロックの犠牲者となるプロセスをすぐに選択するのではなく、いずれかのトランザクションのインデックス・ロックを解放させます。ほとんどの場合、これによって他方のトランザクションは完了し、ロックを解放できます。

ロックの試行を放棄するトランザクションでは、インデックスはルート・ページから再スキャンされ、ページ分割オペレーションは **deadlock retries** で指定されている回数だけ再試行されます。

図 5-1: クラスタード・インデックス内でページ分割中のデッドロック



sp_sysmon は、デッドロックとデッドロック・リトライをレポートします。詳細については、『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』を参照してください。

default character set id

要約	
デフォルト値	1
値の範囲	0 ~ 255
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	言語

default character set id パラメータは、サーバによって使用されるデフォルトの文字セットの番号を指定します。デフォルトはインストール時に設定されますが、Sybase インストール・ユーティリティを使用して後で変更できます。「[第 10 章 文字セット、ソート順、言語の設定](#)」を参照してください。

default database size

要約	
デフォルト値	3MB
値の範囲	2 ^a ~ 10000 a. 最小値はサーバの論理ページ・サイズによって決まる
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

default database size パラメータは、create database 文にサイズ・パラメータの指定をせずに発行した場合に、新しいユーザ・データベースに割り付けられるデフォルト・サイズをメガバイト単位で設定します。create database 文で指定されたデータベース・サイズは、この設定パラメータによって設定する値よりも優先します。

新規データベースのほとんどが論理ページ 1 ページ分よりも多くの領域を必要とする場合は、デフォルト・データベース・サイズの値を大きくする必要があります。

注意 create database コマンドは model データベースをコピーして新しいユーザ・データベースを作成するので、model データベースを変更した場合は、default database size も増やす必要があります。

default exp_row_size percent

要約	
デフォルト値	5
値の範囲	0 ~ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

`default exp_row_size percent` パラメータは、データオンリーロック・テーブルでの拡張更新用に領域を予約します。この目的は、ローの転送を減らすことです。「拡張更新」とは、ローの長さが増えるような、データ・ローへの更新のことです。null 値を持つことができるデータ・ローや可変長カラムのあるデータ・ローへの更新は、拡張更新となる可能性があります。データオンリーロック・テーブルで拡張更新が発生したとき、データ・ローのサイズが増えてそのページに収まらなくなると、ローの転送が必要になることがあります。

デフォルト値を使用する場合は、使用可能なデータ・ページ・サイズの 5 パーセントが拡張更新用に確保されます。データオンリーロック・テーブルのページではデータの記憶領域用に 2,002 バイトが使用可能なので、拡張用には 100 バイトが残されます。この値が適用されるのは、可変長カラムのあるテーブルのページに対してだけです。

`default exp_row_size percent` を 0 に設定すると、すべてのページが最後まで使用され、拡張更新用の領域は残されません。

`exp_row_size` が `create table` に明示的に指定されていない場合や `sp_chgattribute` で設定されている場合、`default exp_row_size percent` パラメータは可変長カラムのあるデータオンリーロック・テーブルに適用されます。`create table` で指定された値は、設定パラメータの設定値よりも優先されます。詳細については、『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』を参照してください。

default fill factor percent

要約	
デフォルト値	0
値の範囲	0 ~ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

`default fill factor percent` パラメータは、既存のデータに対する新しいインデックスを作成するとき `create index` 文でフィルファクタ (fillfactor) が指定されなかった場合に、各インデックス・ページにどの程度までデータを格納するかを決定します。fillfactor の値は、インデックスを作成するときだけ使用されます。データは変更されるので、ページが特定の満杯率で維持されることはありません。

default fill factor percent パラメータは、次のものに影響を与えます。

- データが使用する記憶領域の量 – Adaptive Server は、クラスタード・インデックスを作成するときにデータを再分配します。
- パフォーマンス – ページの分割は Adaptive Server のリソースを消費します。

この値よりも create index コマンドの指定が優先するので、default fill factor percent パラメータを変更する必要はほとんどありません。『リファレンス・マニュアル：コマンド』の「インデックスの作成」を参照してください。

default language id

要約	
デフォルト値	0
値の範囲	0 ~ 32767
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	言語

default language id は、サーバで使用できる言語の中から別の言語をユーザが選択していない場合に、システム・メッセージの表示に使用される言語の番号です。us_english の ID は常に NULL です。言語を追加すると、そのときにユニークな番号がその言語に割り当てられます。

default network packet size

要約	
デフォルト値	2048
値の範囲	512 ~ 65024
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、ネットワーク通信、ユーザ環境

default network packet size は、すべての Adaptive Server ユーザに対するデフォルトのパケット・サイズを設定します。default network packet size に設定できる値は、512 バイトの倍数だけです。それ以外の値を指定した場合は、512 バイトの整数倍になるように切り捨てられます。

デフォルトのバケット・サイズでログインするすべてのユーザ用のメモリは、**total logical memory** で設定される Adaptive Server のメモリ・プールから割り付けられます。このメモリは、Adaptive Server の起動時にネットワーク・バケット用に割り付けられます。

それぞれの Adaptive Server ユーザ接続は次のバッファを使用します。

- 1つの読み込みバッファ
- 1つのメッセージ用バッファ
- 1つの書き込みバッファ

これらのバッファはそれぞれ、**default network packet size** で設定されるバイト数を必要とします。ネットワーク・バケット用に割り付けられるメモリの総量は次のとおりです。

$(\text{number of user connections} + \text{number of worker processes}) * 3 * \text{default network packet size}$

たとえば、**default network packet size** の設定値が 1024 バイトで、50 のユーザ接続と 20 のワーカー・プロセスがある場合は、必要なネットワーク・メモリの量は次のとおりです。

$(50 + 20) * 3 * 1024 = 215040$ バイト

default network packet size の値を大きくした場合は、**max network packet size** もそれと同じサイズ以上に増やす必要があります。**max network packet size** の値が **default network packet size** の値より大きい場合は、**additional network memory** の値を増やしてください。「[additional network memory](#)」(77 ページ)を参照してください。

default network packet size パラメータの変更が、ネットワーク I/O 管理とタスク切り替えにどのように影響するかを確認するには、**sp_sysmon** を使用してください。たとえば、**default network packet size** を増やしてから、**sp_sysmon** の出力をチェックすることにより、**bcp** で大きいバッチを処理するときこの設定がどのように影響するかを確認します。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

ログイン時のバケット・サイズ増加の要求

クライアントは自動的に Adaptive Server のデフォルトのバケット・サイズを使用します。Adaptive Server のクライアント・プログラムで **-A** フラグを使用すると、デフォルトよりも大きなバケット・サイズを要求できます。次に例を示します。

```
isql -A2048
```

default sortorder id

要約	
デフォルト値	50
値の範囲	0 ~ 255
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	言語

default sortorder id は、サーバにデフォルトとして現在インストールされているソート順の番号です。デフォルト・ソート順を変更するには、「[第 10 章 文字セット、ソート順、言語の設定](#)」を参照してください。

default unicode sortorder

要約	
デフォルト値	バイナリ
値の範囲	現在は使用されていない
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	Unicode

default unicode sortorder は、サーバにインストールされている Unicode のデフォルトのソート順をユニークに定義する文字列パラメータです。Unicode のデフォルト・ソート順を変更するには、「[第 10 章 文字セット、ソート順、言語の設定](#)」を参照してください。

default XML sortorder

要約	
デフォルト値	バイナリ
値の範囲	(現在は使用されていない)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	Unicode

default XML sortorder は、XML エンジンによって使用されるソート順を定義する文字列パラメータです。数値パラメータの代わりに文字列パラメータを使用することで、ID の一意性が保証されます。『Adaptive Server Enterprise における XML Services』の「第 6 章 XML における国際化のサポートを参照してください。

deferred name resolution

要約	
デフォルト値	0 (無効)
値の範囲	0 ~ 1
ステータス	動的
必要な役割	システム管理者
設定グループ	クエリ・チューニング

deferred name resolution がアクティブ (1) の場合、遅延名前解決がサーバ接続すべてにグローバルに適用されます。サーバで作成するプロシージャはすべて、遅延名前解決を使用して作成されます。

したがって、ストアド・プロシージャは、ストアド・プロシージャ内で参照されるオブジェクトを解決することなく作成され、オブジェクト解決処理は実行時まで延期されます。『Transact-SQL ユーザーズ・ガイド』の「第 17 章 ストアド・プロシージャの使用」を参照してください。

disable character set conversions

要約	
デフォルト値	0 (有効)
有効な値	0 (有効)、1 (無効)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	言語

disable character set conversions を 1 に変更すると、クライアントと Adaptive Server との間でやり取りされるデータの文字セット変換がオフになります。デフォルトでは、クライアントが使用する文字セットがサーバの文字セットとは異なる場合に、Adaptive Server との間でやり取りされるデータの変換が行われます。たとえば、あるクライアントが Latin-1 (iso_1) を使用していて、Adaptive Server がデフォルトの文字セットとして Roman-8 (roman8) を使用している場合は、クライアントのデータは Adaptive Server にロードされるときに Roman-8 に変換されます。Latin-1 を使用しているクライアントに送信されるデータは再変換されますが、Adaptive Server と同じ文字セットを使用しているクライアントの場合には変換されません。

`disable character set conversions` を設定することにより、変換を行わないことを指定できます。たとえば、すべてのクライアントが同じ文字セットを使用していて、Adaptive Server ではその文字セットですべてのデータが保存されるようにするには、`disable character set conversions` を 1 に設定すれば変換は行われません。

disable disk mirroring

要約	
デフォルト値	1
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ディスク I/O

`disable disk mirroring` は、Adaptive Server のディスク・ミラーリングを有効または無効にします。この設定パラメータはグローバル変数であるため、設定パラメータを 1 に設定して Adaptive Server を再起動した後は、ディスク・ミラーリングは一切実行されません。`disable disk mirroring` を 0 に設定すると、ディスク・ミラーリングが有効になります。

注意 Adaptive Server でフェールオーバを使用できるように設定されている場合は、ディスク・ミラーリングを無効にする必要があります。

disable varbinary truncation

要約	
デフォルト値	0 (オン)
値の範囲	0 (オン)、1 (オフ)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

`disable varbinary truncation` は、`varbinary` または `binary` の null データの最後に後続のゼロを含めるかどうかを制御します。

disk i/o structures

要約	
デフォルト値	256
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ディスク I/O、メモリ使用

disk i/o structures は、Adaptive Server が起動時に割り付けるディスク I/O 制御ブロック数の初期値を指定します。

Adaptive Server がユーザ・プロセスの I/O 要求を開始するには、そのプロセス用のディスク I/O 制御ブロックが必要です。ディスク I/O 制御ブロック用のメモリは、Adaptive Server の起動時に事前に割り付けられています。ディスク I/O 構造体が不足することがないようにするには、**disk i/o structures** をオペレーティング・システムで許容される最大の値に設定します。同時ディスク I/O については、オペレーティング・システムのマニュアルを参照してください。

ディスク I/O 構造体をさらに割り付ける必要があるかどうかを判断するには、**sp_sysmon** を使用してください。『パフォーマンス&チューニング・シリーズ: sp_sysmon による Adaptive Server の監視』を参照してください。**max async i/os per server** 設定パラメータは、**disk i/o structures** と同じ値に設定できます。[「max async i/os per server」 \(160 ページ\)](#) を参照してください。

DMA object pool size

要約	
デフォルト値	4096
有効な値	2048 ~ 2147483647
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	共有ディスク・クラスタ

DMA object pool size は、起動時に CIPC によって割り付けられるダイレクト・メモリ・アクセス (DMA: Direct Memory Access) オブジェクトの数を指定します。

dtm detach timeout period

要約	
デフォルト値	0 (分)
有効な値	0 ~ 2147483647 (分)
ステータス	動的
表示レベル	10
必要な役割	システム管理者
設定グループ	DTM 管理

dtm detach timeout period は、分散トランザクション分岐を分離した状態で保持できる時間を分単位で設定します。X/Open XA 環境では、トランザクションは制御スレッドから分離する場合があります。分離は、一般には別の制御スレッドに付加するために行います。**dtm detach timeout period** で指定された時間は、トランザクションを分離した状態に保持できます。この時間が過ぎると、Adaptive Server は分離されたトランザクションをロールバックします。

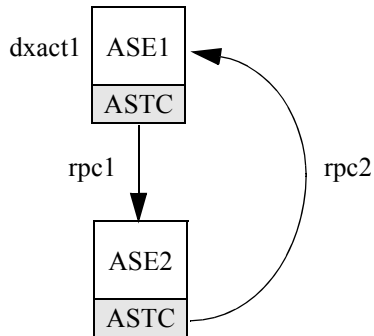
dtm lock timeout period

要約	
デフォルト値	300 (秒)
有効な値	1 ~ 2147483647 (秒)
ステータス	動的
表示レベル	10
必要な役割	システム管理者
設定グループ	DTM 管理

dtm lock timeout period は、ロック・リソースが使用可能になるまで分散トランザクション分岐が待機する最大時間を秒単位で設定します。この時間が経過すると、Adaptive Server はトランザクションがデッドロック状態にあると見なし、デッドロックを引き起こしたトランザクション分岐をロールバックします。これにより、最終的に分散トランザクション全体がロールバックされます。

リモート・サーバにトランザクションが送信された後で、このリモート・サーバから発信サーバにトランザクションが返信されると、分散トランザクション自体がデッドロックすることがあります。この状況を [図 5-2](#) に示します。分散トランザクション“dxact1”の作業が、“rpc1”を経由して Adaptive Server 2 に送信されます。このとき、Adaptive Server 2 は“rpc2”を経由してトランザクションをコーディネーティング・サーバに返信します。“rpc2”と“dxact1”は、同じ gtrid を共有しますが、分岐修飾子が異なるので、同じトランザクション・リソースを共有することはできません。“rpc2”が、“dxact1”によって保持されているロックを待機している場合は、デッドロック状態が発生します。

図 5-2: 分散トランザクション・デッドロック



Adaptive Server は、サーバ間のデッドロックを検出できません。その代わりに、`dtm lock timeout period` に依存します。図 5-2 では、`dtm lock timeout period` の期間を過ぎると、“rpc2” に対して作成されたトランザクションがアボートされます。その結果 Adaptive Server 2 は作業での失敗をレポートし、最終的に“dxact1” もアボートされます。

`dtm lock timeout period` の値は、分散トランザクションだけに適用されます。ローカル・トランザクションでは、サーバワイドの `lock wait period` パラメータで指定されるロック・タイムアウト時間を使用できます。

注意 システム・テーブル上のデッドロックの検出には `dtm lock timeout period` は使用されません。

dump on conditions

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	グループ診断

`dump on conditions` パラメータは、`maximum dump conditions` パラメータで指定された状態になったときに、ダンプ・データを共有メモリ内に生成するかどうかを決定します。

注意 `dump on conditions` パラメータは、Sybase 製品の保守契約を結んでいるサポート・センタだけが使用します。このパラメータは、Sybase 製品の保守契約を結んでいるサポート・センタから指示がないかぎり、変更しないでください。

dynamic allocation on demand

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、物理メモリ

`dynamic allocation on demand` は、動的メモリ設定パラメータの変更に応じていつメモリを割り付けるかを指定します。

`dynamic allocation on demand` を 1 に設定すると、メモリは必要になったときにのみ割り付けられます。たとえば、`number of user connections` の設定を 100 から 200 に変更した場合は、各ユーザ用のメモリはそのユーザがサーバに接続するまでは追加されません。Adaptive Server は、変更後の最大ユーザ接続数に達するまでは、メモリの追加を続けます。

`dynamic allocation on demand` を 0 に設定すると、動的設定パラメータの変更によって必要となるメモリがすべて即時に割り付けられます。したがって、ユーザ接続の最大数を 100 から 200 に変更した場合には、追加された 100 のユーザ接続に必要なメモリがただちに割り付けられます。

enable backupserver HA

要約	
デフォルト値	1
有効な値	1 (有効)、0 (無効)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	共有ディスク・クラスタ

`enable backupserver HA` を 1 に設定すると、クラスタの高可用性 Backup Server が起動します。`enable backupserver HA` を 0 に設定すると、クラスタ上の高可用性 Backup Server が無効になります。

`enable cis`

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	コンポーネント統合サービス

`enable cis` は、コンポーネント統合サービスを有効または無効にします。

`enable compression`

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

`enable cis` は、データ圧縮を有効または無効にします。『圧縮ユーザーズ・ガイド』を参照してください。

`enable console logging`

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	10
必要な役割	システム管理者
設定グループ	SQL Server 管理

`enable console logging` は Adaptive Server がメッセージをコンソールに送信するかどうかを決定します。デフォルトでは `enable console logging` は無効で、起動後、Adaptive Server はコンソールにメッセージを送信しません(エラー・ログには送信されます)。`enable console logging` を有効にすると、起動後にコンソールとエラー・ログにメッセージが常に送信されるようになります。

`enable DTM`

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	10
必要な役割	システム管理者
設定グループ	DTM 管理、SQL Server 管理

`enable DTM` は、Adaptive Server 分散トランザクション管理 (DTM: Distributed Transaction Management) 機能を有効または無効にします。DTM 機能が有効の場合は、Adaptive Server を X/Open XA システムや MSDTC システムのリソース・マネージャとして使用できます。サーバを再起動すると、このパラメータが有効になります。X/Open XA 環境での Adaptive Server の使用方法については、『XA インタフェース統合ガイド for CICS, Encina, TUXEDO』を参照してください。MSDTC 環境でのトランザクションおよび Adaptive Server ネイティブのトランザクション・コーディネーション・サービスについては、『Adaptive Server 分散トランザクション管理機能の使用』を参照してください。

`enable encrypted columns`

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	セキュリティ関連

`enable encrypted columns` は、暗号化カラムを有効にします。

ASE_ENCRYPTION ライセンスを購入してサーバへのインストールと登録を完了しないかぎり、`enable encrypted columns` を設定できません。ライセンスなしに設定しようとする、メッセージ 10834 が表示されます。

```
Configuration parameter 'enable encrypted columns'
cannot be enabled without license 'ASE_ENCRYPTION'
```

注意 暗号化カラムを使用すると、論理メモリ使用量が 8198 キロバイト増加します。

enable enterprise java beans

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	Java サービス

`enable enterprise java beans` は、Adaptive Server データベースで EJB サーバを使用できるかどうかを指定します。Adaptive Server での EJB サーバの使用を有効にしなければ、EJB サーバは使用できません。

enable file access

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	コンポーネント統合サービス

`enable file access` は、プロキシ・テーブルを介した外部ファイル・システムへのアクセスを有効にします。ASE_XFS のライセンスが必要です。

enable full-text search

要約	
デフォルト値	1
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	コンポーネント統合サービス

enable full-text search は、拡張型全文検索サービスを有効にします。ASE_EFTS のライセンスが必要です。

enable functionality group

要約	
デフォルト値	0
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

enable functionality group は、Adaptive Server バージョン 15.7 以降の以下の機能に対する変更を有効または無効にします。

- 共有可能なインライン・デフォルト — **enable functionality group** が 0 に設定されていると、Adaptive Server は共有可能なインライン・デフォルトの作成も、既存の共有可能なインライン・デフォルトの再利用もしません。(この設定パラメータを変更する前は) インライン・デフォルトを共有するカラムは、デフォルトが削除されるまでデフォルトを共有し続けます。
- 排他ロックを取得する **select for update**
- 引用符付き識別子
- Unicode 非文字
- カーソル文のモニタ
- クエリ処理遅延時間の短縮
- Job Scheduler **max task messages** の抑制

このグループのパラメータのデフォルト値は、enable functionality group に設定されている値によって変わります。enable functionality group 以外のこのグループの個々の設定パラメータの DEFAULT の値は、enable functionality group と同じ値に設定されていることを意味します。たとえば、enable functionality group を 1 に設定すると、そのグループのその他の設定パラメータの DEFAULT 値は 1 に設定されます。

enable functionality group の値を除けば、アプリケーション機能グループ内の個々の設定パラメータに対する sp_configure と sp_helpconfig からの出力の DEFAULT 値は無視できます。

enable inline default sharing

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	アプリケーション機能

enable inline default sharing は、Adaptive Server のインライン・デフォルト共有を有効にします。つまりこの設定が有効になると、Adaptive Server は同じユーザに属するデータベース内で同じ値を持つ既存の共有可能なインライン・デフォルトを検索します。既存の共有可能なデフォルトが検出された場合は、新しいデフォルトを作成する代わりに、このオブジェクトがカラムにバインドされます。しかし、既存の共有可能なインライン・デフォルトが検出されない場合には、新しいデフォルトが作成されます。

『Transact-SQL ユーザーズ・ガイド』の「データのデフォルトとルールの定義」を参照してください。

enable HA

要約	
デフォルト値	0 (オフ)
値の範囲	0 ~ 2
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

enable HA を 1 に設定すると、Adaptive Server をアクティブ/アクティブな高可用性サブシステムのコンパニオン・サーバとして設定できます。**enable HA** を 2 に設定すると、Adaptive Server をアクティブ/パッシブな高可用性サブシステムのコンパニオン・サーバとして設定できます。

Adaptive Server は、Sybase のフェールオーバー機能を使用して高可用性サブシステムと連動します。**enable HA** を 1 に設定してから、*installhasvss* スクリプト (Windows では *insthasv*) を実行してください。このスクリプトを実行すると、Sybase フェールオーバーのシステム・プロシージャがインストールされます。

注意 ライセンス情報と **enable HA** の実行値は互いに独立しています。Sybase フェールオーバーのライセンスを取得しているかどうかにかかわらず、実行値と設定値は Adaptive Server の再起動後は 1 に設定されます。ライセンスを取得しなければ、Sybase フェールオーバーを実行することはできません。有効なライセンスがインストールされていない場合は、Adaptive Server のログにエラー・メッセージが記録され、この機能はアクティブ化されません。ライセンス・キーのインストールについては、使用しているプラットフォームのインストール・ガイドを参照してください。

enable HA を 1 または 2 に設定するだけでは、Adaptive Server が高可用性システムで動作するように設定したことにはなりません。『高可用性システムにおける Sybase フェールオーバーの使用』で説明する手順を実行して、Adaptive Server が高可用性システムのコンパニオン・サーバになるように設定してください。

enable HA が 0 に設定されているときは、Sybase のフェールオーバー機能に関する設定を行うことはできません。また、*installhasvss* (Windows では *insthasv*) は実行できません。

enable housekeeper GC

要約	
デフォルト値	1 (オン)
値の範囲	0 ~ 5
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

ハウスキーピング・ガーベジ・コレクション・タスクは、データオンリーロック・テーブルの領域を再利用するための処理を実行します。ユーザ・タスクがデータオンリーロック・テーブルからローを削除すると、データ・ページとインデックス・ページにコミットされた削除があるかどうかを調べるハウスキーピングのタスクがキューイングされます。

ハウスキーピング・ガーベジ・コレクション・タスクを制御するには、**enable housekeeper GC** を使用します。『パフォーマンス&チューニング・シリーズ：基本』の「第 3 章 エンジンと CPU の使用方法」を参照してください。

enable housekeeper GC の有効値を次に示します。

- 0 – ハウスキーピング・ガーベジ・コレクション・タスクは実行しませんが、**delete** コマンドによるレイジー・ガーベジ・コレクションは実行できるようにします。**reorg reclaim_space** を使用して、空ページの割り付けを解除する必要があります。これは、パフォーマンスへの影響が最も少なく、最も低コストのオプションですが、累積した空ページの量が増えるとパフォーマンス上の問題が発生する可能性があります。この値を使用することはおすすめしません。
- 1 – ハウスキーピング・ガーベジ・コレクション・タスクと **delete** コマンドで、レイジー・ガーベジ・コレクションを実行できます。アプリケーションで許容される以上の空ページが累積する場合は、オプション 4 または 5 の使用を検討してください。**optdiag** ユーティリティを使用すると、空ページの統計情報を取得できます。
- 2 – 今後のために予約済み。
- 3 – 今後のために予約済み。
- 4 – ハウスキーピング・ガーベジ・コレクション・タスクと **delete** コマンドで、積極的ガーベジ・コレクションを実行できます。このオプションを選択すれば効果が最も高くなりますが、**delete** コマンドのコストが高くなります。このオプションは、DOL テーブルに対する一連の削除を 1 つのバッチで実行する場合に理想的です。
- 5 – ハウスキーピング・タスクではアグレッシブ・ガーベジ・コレクションを実行でき、**delete** コマンドではレイジー・ガーベジ・コレクションを実行できます。オプション 4 を選択した場合よりも、削除のコストは低くなります。このオプションは、同時トランザクションによって削除が行われる場合に適しています。

sp_sysmon は、ハウスキーピング・ガーベジ・コレクション・タスクによる領域再利用処理の実行頻度と、再利用可能となったページ数をレポートします。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

enable hp posix async i/o

要約	
デフォルト値	0
有効な値	1 (有効)、0 (無効)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ディスク I/O

enable hp posix async i/o を 1 に設定すると、HP-UX 11.31 以降のファイル・システムで作成されるデータベース・デバイスでの非同期 I/O が有効になります。まず、allow sql server async i/o 設定パラメータにより、Adaptive Server が非同期 I/O を使用できるようにする必要があります。

以下の設定パラメータの組み合わせにより、Adaptive Server がロー・パーティションとファイル・システムに対して非同期 I/O または同期 I/O を使用するかどうかを決定します。

hp posix async i/o の有効化	allow sql server async i/o = 0	allow sql server async i/o = 1
0 に設定	ファイル・システムで同期 I/O ロー・デバイスで同期 I/O	ファイル・システムで同期 I/O ロー・デバイスで /dev/async 非同期 I/O を使用
1 に設定	ファイル・システムで同期 I/O ロー・デバイスで同期 I/O	ファイル・システムで POSIX 非同期 I/O ロー・デバイスで POSIX 非同期 I/O

注意 enable hp posix async i/o はデータベース・デバイスをファイル・システム上に割り当てるときにはパフォーマンスを改善します。しかし、ロー・デバイスに割り当てられたデータベース・デバイスではパフォーマンスが低下することがあります。

enable i/o fencing

要約	
デフォルト値	0
有効な値	1 (有効)、0 (無効)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	共有ディスク・クラスタ

`enable i/o fencing` を 1 に設定すると、SCSI-3 PGR (Persistent Group Reservation) 規格をサポートする各データベース・デバイスで I/O フェンシング機能が有効になります。

enable java

要約	
デフォルト値	0 (無効)
値の範囲	0 (無効)、1 (有効)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	Java サービス

`enable java` は、Adaptive Server データベースで Java を使用できるかどうかを指定します。サーバでの Java の使用を有効にしなければ、Java クラスをインストールしたり Java の操作を実行したりすることはできません。

enable job scheduler

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

`enable job scheduler` は、Adaptive Server の起動時に Job Scheduler を起動するかどうかを指定します。

enable ldap user auth**要約**

デフォルト値	0 (オフ)
有効な値	0 (オフ) – syslogins 認証のみを許可 1 (オン) – LDAP 認証と syslogins 認証の両方を許可 2 (オン) – LDAP 認証のみを許可
ステータス	動的
表示レベル	包括
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

enable ldap user auth が 1 に設定されているときは、Adaptive Server は各ユーザを認証するために LDAP サーバを検索します。LDAP 認証に失敗したときは、syslogins を検索してユーザを認証します。レベル 1 は、Adaptive Server 認証から LDAP 認証へユーザをマイグレートしているときに使用します。

enable literal autoparam**要約**

デフォルト値	0
値の範囲	1 (有効)、0 (無効)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	クエリ・チューニング

enable literal autoparam は、サーバ全体でリテラルの自動パラメータ化を有効または無効にします。

enable logins during recovery**要約**

デフォルト値	1
値の範囲	1 (有効)、0 (無効)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	セキュリティ関連

`enable logins during recovery` は、データベース・リカバリ中にシステム管理者以外のログインを許可するかどうかを指定します。値 1 はリカバリ中にログインが許可されることを示し、値 0 はリカバリ中にログインが許可されない(システム管理者のみが Adaptive Server にログインできる)ことを示します。

enable merge join

要約	
デフォルト値	2
値の範囲	0 – サーバ・レベルのマージ・ジョインを無効にする 1 – サーバ・レベルのマージ・ジョインを有効にする 2 – サーバ・レベルのマージ・ジョインをデフォルト値に設定する
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

`enable merge join` サーバ・レベルでマージ・ジョインを有効または無効にします。

マージ・ジョインのデフォルト値は、`optimization goal` 設定パラメータの現在の値によって異なります。

最適化目標の値	マージ・ジョインのデフォルト値
<code>allrows_mix</code>	on
<code>allrows_dss</code>	on
<code>allrows_oltp</code>	off

enable metrics capture

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

`enable metrics capture` は、Adaptive Server が測定基準をサーバ・レベルで取得できるようにします。アドホック文の測定基準はシステム・カタログ内に取得され、ストアド・プロシージャに含まれる文の測定基準はプロシージャ・キャッシュに保存されます。

enable monitoring

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	モニタリング

enable monitoring は、Adaptive Server でモニタリング・テーブル・データを収集するかどうかを制御します。**enable monitoring** は、Monitoring 設定パラメータが有効になるかどうかを指定するマスタ・スイッチの働きをします。

注意 **enable monitoring** を有効にしないままでモニタリング・テーブルに移植されたデータは有効とは見なされません。

enable pam user auth

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ) – syslogins 認証のみを許可 1 (オン) – PAM 認証と syslogins 認証の両方を許可 2 (オン) – PAM 認証のみを許可
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

enable pam user auth は、PAM (Pluggable Authentication Modules) を使用してユーザを管理する能力を制御します。

enable pam user auth が 1 に設定されているときは、Adaptive Server は各ユーザを認証するために PAM プロバイダを使用します。PAM 認証に失敗したときは、**syslogins** を検索してユーザを認証します。レベル 1 は、Adaptive Server 認証から PAM 認証へユーザをマイグレートしているときに使用します。

enable pci

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)、2 (オペレーティング・システムの無効化によりオン)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ユーザ環境

enable pci は、Adaptive Server の Java PCI Bridge を有効または無効にします。

注意 Sybase 製品の保守契約を結んでいるサポート・センタの指示がないかぎり、設定 “2” (オペレーティング・システムの無効化によりオン) を使用しないでください。この設定を使用すると、PCI 機能を完全または正しくサポートしていない可能性があるオペレーティング・システムのバージョンにおいても PCI Bridge が有効になります。

enable permissive unicode

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	アプリケーション機能

enable permissive unicode により、Adaptive Server が Unicode 非文字を無視するようにできます。

詳細については、「[Unicode 非文字の許可](#)」(362 ページ) を参照してください。

enable query tuning mem limit

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	クエリ・チューニング

enable query tuning mem limit は、クエリ・チューニングのメモリ制限を有効にします。

enable query tuning time limit

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	中間
表示レベル	中間
必要な役割	システム管理者
設定グループ	クエリ・チューニング

enable query tuning time limit は、クエリ・チューニングの時間制限を有効にします。

enable real time messaging

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

enable real time messaging は、Real Time Messaging Services を有効にします。

enable rep agent threads

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、RepAgent スレッド管理

enable rep agent threads パラメータは、Adaptive Server 内で RepAgent スレッドを実行できるようにします。

複写を使用できるようにするには、他の手順も実行する必要があります。詳細については、Replication Server のマニュアルを参照してください。

enable row level access control

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

enable row level access control は、ロー・レベルのアクセス制御を有効にします。**enable row level access control** を設定するためには、事前にセキュリティ・サービス・ライセンスを有効にしておくことが必要です。

enable semantic partitioning

要約	
デフォルト値	0
値の範囲	1 (有効)、0 (無効)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

enable semantic partitioning は、Adaptive Server でラウンドロビン方式以外の分割 (リスト、ハッシュ、範囲による分割) を実行できるようにします。これらの分割スキームを使用するには、適切なライセンスを所有している必要があります。

enable sort-merge join and JTC

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

Adaptive Server が互換モードのときに使用されます。いったん有効にされると、Adaptive Server がクエリを互換モードでコンパイルすると、enable sort-merge join and JTC はクエリ・プロセッサがソート・マージ・ジョインまたはネストされたループ・ジョインを選択できるようにします。enable sort-merge join and JTC は、JTC (Join Transitive Closure) を有効にします。これにより、15.0 より前のリリースのクエリ・プロセッサは追加のジョイン句を使用できるようになります。

互換モードの詳細については、『マイグレーション技術ガイド』を参照してください。

enable sql debugger

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

Adaptive Server SQL デバッガを有効または無効にします。このデバッガを使用すると、T-SQL コードを 1 ステップずつ実行できます。

enable ssl

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	包括
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

`enable ssl` は、Secure Sockets Layer セッションベースのセキュリティを有効または無効にします。

enable stmt cache monitoring

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	モニタリング

`enable stmt cache monitoring` は、Adaptive Server がステートメント・キャッシュに関するモニタリング情報を収集する機能を有効または無効にします。いったん有効にされると、`monStatementCache` と `monCachedStatement` は有効なデータを表示します。

enable surrogate processing

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	Unicode

サロゲート処理をアクティブにし、Unicode データのサロゲート・ペアが整合性を保つようになります。`enable surrogate processing` を無効にすると、Unicode データ内のサロゲート・ペアは無視され、サロゲート・ペアの整合性を維持するコードがすべてスキップされます。これによってパフォーマンスは向上しますが、データとして表示される Unicode 文字の範囲は小さくなります。

enable unicode conversion

要約	
デフォルト値	1
値の範囲	0 – 組み込みの文字セット変換のみ使用。 1 – 組み込み変換を使用。組み込みの文字セット変換が見つからないときは、Unilib 文字変換を使用する。 2 – 適切な Unilib 変換を使用。
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	言語、Unicode

`enable unicode conversion` は、`char`、`varchar`、`text` の各データ型について、Unilib を使用した文字変換をアクティブにします。

enable unicode normalization

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	Unicode

Unilib 文字正規化をアクティブにします。正規化とは、特定の抽象文字シーケンスに対する表現がデータベース内に1つだけ存在するようにデータを修正するプロセスのことです。多くの場合、発音区別符号が後に付いた文字が、事前結合済みの形式に置き換えられます。

入力されたすべての Unicode データに対して正規化を実行する組み込みプロセスを使用する場合は、`enable unicode normalization` を 1 に設定します。このパラメータを無効にする (0 に設定する) と、正規化が省略されるため、サーバではなくクライアント・コード側で正規化を行うこととなります。正規化を無効にするとパフォーマンスが向上します。ただし、すべてのクライアントが同じ表現を使用して Unicode データをサーバに渡す場合に限られます。

注意 いったん無効にすると、再び有効にはできません。この制限により、正規化されていないデータがデータベースに格納されることが防止されます。

enable webservices

要約	
デフォルト値	0
値の範囲	1 (有効)、0 (無効)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

Web Services を有効にします。

enable xact coordination

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	10
必要な役割	システム管理者
設定グループ	DTM 管理

enable xact coordination は、Adaptive Server トランザクション・コーディネーション・サービスを有効または無効にします。このパラメータを 1 (オン) に設定するとコーディネーション・サービスが有効になり、サーバは他の Adaptive Server にトランザクションを送信できます。これが発生するのは、トランザクションがリモート・プロシージャ・コール (RPC) を実行して他のサーバのデータを更新するか、コンポーネント統合サービス (CIS) を使用して他のサーバのデータを更新する場合です。トランザクション・コーディネーション・サービスは、リモート・サーバのデータへの更新が、必ずオリジナル・トランザクションとともにコミットまたはロールバックされるようにする機能です。

このパラメータを 0 (オフ) に設定すると、Adaptive Server はリモート・サーバの作業をコーディネートしません。トランザクションで RPC を実行することや CIS を使用してデータを更新することはできますが、リモート・サーバにシステム障害が発生したときに、リモート・トランザクションがオリジナル・トランザクションとともにロールバックされることや、リモートの作業がオリジナル・トランザクションとともにコミットされることを Adaptive Server が保証することはできません。これは、バージョン 12.x より前の Adaptive Server の動作に対応しています。

enable xml

要約	
デフォルト値	0
値の範囲	1 (有効)、0 (無効)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

XML サービスを有効にします。

engine memory log size

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	
必要な役割	
設定グループ	物理メモリ

engine memory log size は、診断専用であり、運用環境には関連しません。Sybase 製品の保守契約を結んでいるサポート・センタからの指示がないかぎり、デフォルト値のままにしておいてください。

errorlog pipe active

要約	
デフォルト値	0
値の範囲	0 ~ 1
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	モニタリング

errorlog pipe active は、Adaptive Server でエラー・ログ・メッセージを収集するかどうかを制御します。errorlog pipe active と errorlog pipe max messages を両方とも有効にすると、Adaptive Server はエラー・ログに送られたすべてのメッセージを収集します。このエラー・ログ・メッセージを取得するには、monErrorLog を使用します。

errorlog pipe max messages

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、モニタリング

errorlog pipe max messages は、Adaptive Server が格納するエラー・ログのメッセージ数をエンジンごとに決定します。**monSQLText** テーブル内のメッセージ数の合計は、**sql text pipe max messages** に実行中のエンジン数を掛け合わせた値になります。

esp execution priority

要約	
デフォルト値	8
値の範囲	0 ~ 15
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	拡張ストアド・プロシージャ

esp execution priority は、ESP 実行用の XP Server スレッドの優先度を設定します。ESP は、長時間にわたって CPU を集中的に使用することがあります。また、XP Server は Adaptive Server と同じマシン上に常駐するため、Adaptive Server のパフォーマンスにも影響を与えることがあります。

Open Server スレッドのスケジューリングについては、『Open Server Server-Library/C リファレンス・マニュアル』を参照してください。

esp execution stacksize

要約	
デフォルト値	プラットフォームに依存する。 <ul style="list-style-type: none"> • 196608 – Linux AMD-64、IBM PLinux、HP IA64、Sun x86 64 • 139264 – SunSparc64 • 65536 – Sun Sparc32、Windows 32、IBM AIX 64、Linux IA 32、Windows 64 • 67584 – HP 64
値の範囲	プラットフォームのデフォルト –2147483647
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	拡張スタアド・プロシージャ

esp execution stacksize は、ESP 実行用に割り付けるスタック・サイズをバイト単位で設定します。

デフォルトの 34816 よりも大きいスタック・サイズを必要とする独自の ESP 関数を使用する場合に、このパラメータを使用します。

esp unload dll

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	拡張スタアド・プロシージャ

esp unload dll は、ESP の呼び出し完了後に、ESP をサポートする DLL を XP Server のメモリから自動的にアンロードするかどうかを指定します。

esp unload dll を 0 に設定すると、DLL は自動的にアンロードされません。1 に設定すると、自動的にアンロードされます。

esp unload dll を 0 に設定した場合でも、**sp_freedll** を使用すれば、実行時に個々の DLL を明示的にアンロードできます。

event buffers per engine

要約	
デフォルト値	100
値の範囲	1 ~ 2147483647
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、SQL Server 管理

event buffers per engine は、各エンジンが使用できるイベント・バッファの数を指定します。

イベントは、Adaptive Server のパフォーマンスを監視するために Adaptive Server Monitor が使用します。Adaptive Server Monitor を使用していない場合は、このパラメータを 1 に設定してください。

event buffers per engine に設定する値は、設定されているエンジン数、Adaptive Server のアクティビティのレベル、実行するアプリケーションの種類によって決まります。

event buffers per engine を小さな値に設定すると、イベント情報が失われることがあります。デフォルト値は、ほとんどのサイトにとって小さすぎる可能性があります。一般的なモニタには、2,000 以上の値が妥当です。ただし、サイトにとって適切な値を決めるには経験が必要です。

一般に、**event buffers per engine** の設定値を大きくすれば、Adaptive Server Monitor が原因で起きる Adaptive Server のパフォーマンス低下を緩和することができます。

それぞれのイベント・バッファは 100 バイトのメモリを使用します。**event buffers per engine** に特定の値を設定した場合に使用されるメモリの合計量を調べるには、**event buffers per engine** の値に、設定されている Adaptive Server エンジン数を掛けます。

event log computer name (Windows のみ)

要約	
デフォルト値	LocalSystem
有効な値	<ul style="list-style-type: none"> Adaptive Server のメッセージを記録するように設定されている、ネットワーク上の Windows マシンの名前 LocalSystem NULL
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	エラー・ログ

event log computer name には、Adaptive Server のメッセージを Windows のイベント・ログに記録する Windows PC マシンの名前を指定します。この機能は、Windows サーバでのみ使用できます。

LocalSystem または NULL を設定すると、デフォルトのローカル・システムが指定されます。

また、Server Config ユーティリティを使用して、**event log computer name** パラメータを設定することもできます。その場合は、[イベント・ログ] でコンピュータ名を指定します。

コマンド・ラインで **-G** オプションを指定した場合に、**sp_configure** で **event log computer name** パラメータを設定するか、[イベント・ログ] でコンピュータ名を指定すると、このオプションの結果は上書きされます。**-G** オプションを指定して Adaptive Server を起動した場合は、**event log computer name** パラメータを設定することによって送信先リモート・マシンを変更できます。

Adaptive Server メッセージをリモート・サイトでログに記録する方法の詳細については、『Adaptive Server Enterprise 設定ガイド Windows 版』を参照してください。

event logging (Windows のみ)

要約	
デフォルト値	1
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	エラー・ログ

event logging は、Windows のイベント・ログに Adaptive Server のメッセージを記録するかどうかを指定します。

デフォルト値の 1 を指定すると Adaptive Server のメッセージは Windows のイベント・ログに記録され、0 を指定するとイベント・ログには記録されません。

Server Config ユーティリティを使用して event logging パラメータを設定するには、[イベント・ログ] の [Windows イベント・ログを使用] を選択します。

コマンド・ラインで -G オプションを指定した場合に、event logging パラメータを設定するか、[Windows イベント・ログを使用] を選択すると、このオプションの結果は上書きされます。

executable codesize + overhead

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	計算された値
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用

executable codesize + overhead は、Adaptive Server の実行プログラムとオーバヘッドを合わせたサイズをキロバイト単位で表します。これは、システムによって計算される値であって、ユーザが設定できるものではありません。

extended cache size

要約	
デフォルト値	0
値の範囲	0 ~ 31457280
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	キャッシュ・マネージャ

extended cache size は、セカンダリ・キャッシュのサイズを指定します。

FIPS login password encryption

要約	
デフォルト値	0
値の範囲	0 ~ 1
ステータス	静的
表示レベル	包括
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

FIPS login password encryption を有効にするには、セキュリティ&ディレクトリサービス ライセンスが必要です。このパラメータは、FIPS 140-2 暗号化モジュールをサポートして、転送時、メモリ内、ディスク上でのパスワードを暗号化します。

Adaptive Server では、ログイン暗号化に FIPS 140-2 認定の Certicom 製セキュリティ・プロバイダが使用されます。この設定が有効でない場合、Adaptive Server は OpenSSL セキュリティ・プロバイダを使用してログイン・パスワードの暗号化を実行します。

global async prefetch limit

要約	
デフォルト値	10
値の範囲	0 ~ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	キャッシュ・マネージャ

global async prefetch limit は、非同期プリフェッチによってバッファ・プールに取り込まれたけれどもまだ読み込まれていないページを保持できる割合を指定します。このパラメータは、制限値が `sp_poolconfig` で明示的に設定されていないすべてのキャッシュ内のすべてのプールの制限値を設定します。

プールの制限値を超えた場合、読み込まれていないページの割合が制限値より小さくなるまで、非同期プリフェッチは一時的に無効になります。『パフォーマンス&チューニング・シリーズ：基本』の「第 6 章 非同期プリフェッチのチューニング」を参照してください。

global cache partition number

要約	
デフォルト値	1
値の範囲	1 ~ 64 (2 の累乗)
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	キャッシュ・マネージャ

global cache partition number は、すべてのデータ・キャッシュのキャッシュ・パーティションのデフォルト数を設定します。特定のキャッシュのパーティション数は、**sp_cacheconfig** を使用して設定できます。ローカルのパーティション数がグローバルのパーティション数よりも優先します。

キャッシュ・パーティションを使用するとキャッシュ・スピンロックの競合が減少します。一般に、キャッシュ・スピンロックの競合が 10% を超えるときは、キャッシュを分割することでパフォーマンスが向上します。パーティションの数を 2 倍にすると、スピンロックの競合がおよそ 2 分の 1 に減少します。

キャッシュ・パーティションの設定の詳細については、『システム管理ガイド 第 2 巻』の「第 4 章 データ・キャッシュの設定」を参照してください。また、『パフォーマンス&チューニング・シリーズ：基本』の「第 6 章 非同期プリフェッチのチューニング」も参照してください。

heap memory per user

要約	
デフォルト値	4K
有効な値	0 ~ 2147483647 バイト
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、物理メモリ

heap memory per user は、ユーザ当たりのヒープ・メモリ量を設定します。ヒープ・メモリ・プールは起動時に作成される内部メモリで、このプールからタスクが必要に応じて動的にメモリを割り付けます。このメモリ・プールが重要な役割を果たすのは、ワイド・カラムを使用するため大量のメモリを必要とするタスクを実行する場合です。ヒープ・メモリからテンポラリ・バッファが割り付けられることによって、ワイド・カラムを使用するタスクの実行が可能となります。タスクが使用するヒープ・メモリは、タスクが完了するとヒープ・メモリ・プールに返されます。

メモリ・プールのサイズは、ユーザ接続数によって異なります。heap memory per user は、論理ページ・サイズの 3 倍に設定することをおすすめします。

histogram tuning factor

要約	
デフォルト値	20
値の範囲	1 ~ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

histogram tuning factor は、Adaptive Server が update statistics、update index statistics、update all statistics、create index について 1 つのヒストグラムで分析するステップ数を制御します。値を 1 に設定すると、パラメータが無効になります。

注意 Adaptive Server バージョン 15.0.2 ESD #2 およびそれ以降の場合、histogram tuning factor をデフォルト値の 20 に設定し、しかもヒストグラムに多数のステップが必要とされる場合、ヒストグラムに使用される実際のステップ数は、プロシージャ・キャッシュ使用量を減らす値に制限されます。

```
min (max (400, requested_steps),
    histogram_tuning_factor X requested_steps)
```

次の例では、Adaptive Server は 30 個の値を含む 20 ステップの中間ヒストグラムを生成します。

```
sp_configure 'histogram tuning factor',20
update statistics tab using 30 values
```

Adaptive Server は、次の条件に従って、ヒストグラムを分析して結果のヒストグラムに圧縮します。

- 最初のステップは変更せずにコピーする。
- 頻度の高いステップは変更せずにコピーする。
- 連続した範囲のステップは 1 つの結果ステップにまとめる。このため、まとめられたステップの総ウェイトは値の 30 分の 1 を超えない。

`sysstatistics` 内の最終的なヒストグラムは次のようになります。

- 範囲ステップは 30 ステップの `update statistics` と同様に生成され、高頻度の範囲はヒストグラムが 600 ステップで作成された場合のように分離されます。
- 結果ヒストグラムの合計ステップ数は、30 ～ 600 の値になります。
- 均等に分散したデータの場合、値は 30 にかぎりなく近づきます。
- テーブルでの「頻度」が高い値は、ヒストグラムでのステップ数が多くなります。
- 1 つのカラムに異なる値がわずかしかない場合は、それらすべての値が高頻度セルとして表示されることがあります。

`number of histogram steps` を 600 に増やしても同じ結果が得られますが、バッファやプロシージャ・キャッシュでより多くのリソースを使用することになります。

`histogram tuning factor` を使用すると、ヒストグラムで使用されるリソースを最小限に抑えられます。リソース使用量が増えるのは、最適化のために適切な場合のみです。たとえば、カラムのデータ分布に一貫性がない場合、または重複する値がカラム内に多数存在する場合です。このような場合には、最大 600 のヒストグラム・ステップが使用されます。ただし、ほとんどの場合、`histogram tuning factor` はデフォルト値 (上記の例では 30) が使用されます。

housekeeper free write percent

要約	
デフォルト値	1
値の範囲	0 ～ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

`housekeeper free write percent` パラメータは、ハウスキーピング・ウォッシュ・タスクによるデータベースへの書き込みの最大増加率を指定します。

たとえば、データベースへの書き込み頻度が通常より 5 パーセント高くなった場合にハウスキーピング・タスクの動作を停止させるには、`housekeeper free write percent` を 5 に設定します。

Adaptive Server が処理するユーザ・タスクがなくなると、ハウスキーピング・ウォッシュ・タスクは、変更されたページをキャッシュからディスクに書き込む処理を自動的に開始します。この書き込みによって CPU 使用率が改善され、トランザクション処理中のバッファ・ウォッシングの必要性が少なくなり、チェックポイントが短くなります。

同じデータベース・ページを繰り返し更新するアプリケーションでは、ハウスキーピング・ウォッシュによるデータベースへの書き込みの中には実際には不必要なものもあります。このような書き込みはサーバのアイドル時間中にだけ発生しますが、ディスクへの負荷が大きいシステムでは許容できないことがあります。

クエリを最適化するために使用されるテーブルとインデックス統計は、クエリ処理中はメモリ構造内に保持されます。この統計が変化しても、変更内容がすぐには `sysabstats` テーブルに書き込まれません。これは、I/O 競合を削減し、パフォーマンスを向上させるためです。代わりに、ハウスキーピング・ジョブ・タスクによって統計が定期的にディスクにフラッシュされます。

デフォルト値を使用する場合は、ディスク I/O が最大 1 パーセント増加するまでハウスキーピング・ウォッシュ・タスクを実行できます。これは、ほとんどのシステムでパフォーマンスとリカバリ・スピードを改善します。

ハウスキーピング・ウォッシュ・タスクが実行されないようにするには、`housekeeper free write percent` の値を 0 に設定します。

システムでのディスク競合が多く、ハウスキーピング・ウォッシュ・タスクによる I/O の増加が許容されない場合にのみ、この値を 0 に設定します。

ハウスキーピングを無効にした場合は、統計情報を常に最新の状態に保ってください。ディスクに統計を書き込むコマンドは、次のとおりです。

- `update statistics`
- `dbcc checkdb` (データベース内のすべてのテーブルの場合) または `dbcc checktable` (1 つのテーブルの場合)
- `sp_flushstats`

統計がディスクに最後に書き込まれた後で更新されたテーブルに対して、これらのコマンドのいずれかを、次の時点で実行してください。

- データベースのダンプの前
- 正常なシャットダウンの前
- 失敗または正常なシャットダウンの後の再起動の後。このような場合、`sp_flushstats` は使用できません。`update statistics` コマンドまたは `dbcc` コマンドを使用してください。
- テーブルへの大幅な変更後。たとえば、大量のバルク・コピー・オペレーション、ロック・スキームの変更、大量のローの削除や挿入、`truncate table` コマンドの実行などがあります。

データベース書き込みの増加率に関係なくハウスキーピング・ウォッシュ・タスクが連続的に動作するようにするには、`housekeeper free write percent` を 100 に設定します。

ハウスキーピングのパフォーマンスをモニタリングするには、`sp_sysmon` を使用します。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

また、ハウスキーピング・タスクによって発生したフリー・チェックポイントの数を調べることもできます。この出力については、『パフォーマンス&チューニング・シリーズ：基本』を参照してください。

i/o accounting flush interval

要約	
デフォルト値	1000
値の範囲	1 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

`i/o accounting flush interval` は、Adaptive Server が各ユーザーについての I/O 統計情報を `sysprocesses` から `syslogins` にフラッシュするまでの待ち時間をマシンのクロック・チック単位で指定します。これはチャージバック アカウンティングに使用します。

ユーザーが Adaptive Server にログインすると、その時点以降のユーザー・プロセスの I/O 統計情報が `sysprocesses` に蓄積されます。`i/o accounting statistics interval` の時間が過ぎるか、ユーザーが Adaptive Server からログオフすると、蓄積された I/O 統計情報が `sysprocesses` から `syslogins` にフラッシュされます。この統計情報は、システム管理者が `sp_clearstats` を使用して合計をクリアするまでは、`syslogins` に引き続き蓄積されます。`syslogins` から現在の合計を表示するには、`sp_reportstats` を使用します。

`i/o accounting flush interval` に設定する値は、目的とするレポートのタイプによって異なります。月単位でレポートを作成する場合は、`i/o accounting flush interval` には比較的大きな値を設定します。レポート処理の頻度が低ければ、`syslogins` 内のデータを頻繁に更新することはそれほど重要ではありません。

プロセスによる I/O 量を調べるために、アドホック・クエリを使用して `syslogins` の `totio` カラムからの選択を定期的に行う場合は、`i/o accounting flush interval` の設定値を小さくします。このようにすれば、選択を実行するときに `syslogins` 内のデータが最新のものである可能性が高くなります。

I/O 統計情報をレポートしない場合は、`i/o accounting flush interval` を最大値に設定してください。これにより、`syslogins` が更新される回数と、そのページをディスクに書き込まなければならない回数が減ります。

i/o batch size

要約	
デフォルト値	100
値の範囲	1 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

i/o batch size は、タスクがスリープする前にバッチ内で発行される書き込みの回数を設定します。バッチが完了すると、タスクがウェイクアップして、次の書き込みバッチが発行されます。これにより、I/O サブシステムに対して大量の書き込みが同時に発生することを防止できます。*i/o batch size* に適切な値を設定することで、*checkpoint* や *dump database*、*select into* などの操作のパフォーマンスを向上させることができます。

i/o polling process count

要約	
デフォルト値	10
値の範囲	1 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

注意 *i/o polling process count* は Adaptive Server をプロセス・カーネル・モードに設定した場合だけ機能し、スレッド・カーネル・モードでは機能しません。

i/o polling process count は、Adaptive Server が実行可能なプロセスの最大数を指定します。この数に達したとき、スケジューラはディスク I/O やネットワーク I/O が完了しているかどうかを調べます。*i/o polling process count* を調整すると、Adaptive Server の応答時間とスループットの両方が影響を受けます。

Adaptive Server は、次のときにディスク I/O またはネットワーク I/O が完了しているかどうかを調べます。

- 前回 Adaptive Server が I/O の完了を調べた後に実行されたタスク数が、*i/o polling process count* の値に等しくなったとき
- Adaptive Server のクロック・チックごと

一般的には、`i/o polling process count` の値を増やすと、大量のディスク I/O とネットワーク I/O を行うアプリケーションのスループットが向上します。逆にこの値を減らすと、スループットが低下するリスクはありますが、このようなアプリケーションの処理応答時間を改善できます。

アプリケーションが I/O バウンド・タスクと CPU バウンド・タスクの両方を作成する場合は、`i/o polling process count` を 1～2 の小さな値に調整すると、I/O バウンド・タスクが確実に CPU サイクルにアクセスできるようになります。

OLTP アプリケーション (またはユーザ接続と短いトランザクションを使用する I/O バウンド・アプリケーション) の場合は、`i/o polling process count` を 20～30 の範囲の値に調整すると、スループットが増加することがありますが、応答時間が長くなる可能性もあります。

`i/o polling process count` を調整する場合は、以下のパラメータを考慮してください。

- `sql server clock tick length`。このパラメータは、Adaptive Server のクロック・チックの長さをマイクロ秒単位で指定します。[「sql server clock tick length」\(247 ページ\)](#) を参照してください。
- `time slice`。このパラメータは、Adaptive Server のスケジューラがユーザ・プロセスの実行を許容する時間をクロック・チック数として指定します。[「time slice」\(262 ページ\)](#) を参照してください。
- `cpu grace time`。このパラメータは、ユーザ・プロセスが CPU を解放せずに実行できる時間の最大長をクロック・チック単位で指定します。この時間に達すると、Adaptive Server はユーザ・プロセスの制御を横取りしてタイムスライス・エラーで終了させます。詳細については、[「cpu grace time」\(98 ページ\)](#) を参照してください。

`i/o polling process count` の変更の効果を調べるには、`sp_sysmon` を使用します。『パフォーマンス&チューニング・シリーズ: sp_sysmon による Adaptive Server の監視』を参照してください。

identity burning set factor

要約	
デフォルト値	5000
値の範囲	1 ~ 9999999
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

IDENTITY カラムは、numeric 型で位取りがゼロの、Adaptive Server によって値が生成されるカラムです。カラム値の最小値は 1 で、最大値はカラムの精度によって決まります。

Adaptive Server は、IDENTITY カラムのあるテーブルごとに、カラム値として可能な一連の値を連続した番号のブロックに分けて、メモリ内で一度に 1 ブロックずつ使用できるようにします。テーブルにローが挿入されるたびに、そのブロックから次に使用可能な値を IDENTITY カラムに割り当てます。ブロック内のすべての番号を使いきると、次のブロックが使用可能になります。

IDENTITY カラム値を選択するこの方法は、サーバのパフォーマンスを改善します。Adaptive Server は、新しいカラム値を割り当てるときに、メモリから現在の最大値を読み込んで 1 を加えます。ディスク・アクセスが必要になるのは、ブロック内の値を使いきったときだけです。サーバの障害が発生すると（または `shutdown with nowait` が実行されると）、ブロック内に残っている番号はすべて破棄されるので、この方法では IDENTITY カラム値に欠番が発生することがあります。

`identity burning set factor` は、それぞれのブロックで使用可能にすることができるカラムの値の割合を変更するために使用します。この値は適切なパフォーマンスを得るのに十分な大きさにしますが、カラム値の欠番が許容できなくなるほどには大きくしないでください。デフォルト値の 5,000 は、IDENTITY カラム値の 0.05% を一度に使用できるようにします。

`sp_configure` で値を正しく設定するには、パーセンテージを小数で表した値を 10^7 (10,000,000) 倍します。たとえば、IDENTITY カラム値の候補の 15% を一度に使用できるようにするには、 0.15×10^7 (つまり 1,500,000) を `sp_configure` で指定します。

identity grab size

要約	
デフォルト値	1
値の範囲	1 ~ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

`identity grab size` は、Adaptive Server の各プロセスが、IDENTITY カラムを持つテーブルに挿入する IDENTITY カラム値のブロックを予約できるようにするためのパラメータです。

これは、挿入処理を実行するときに、すべての挿入データの IDENTITY 番号が連続するようにする場合に便利です。たとえば支払給与データの入力中に、特定の部署に関するすべてのレコードを同じローのブロック内に置く場合は、`identity grab size` をその部署のレコード数に設定します。

`identity grab size` に設定する値は、Adaptive Server のすべてのユーザに適用されます。このため、`identity grab size` を大きな値に設定すると、IDENTITY カラムを持つテーブルに多くのユーザがデータを挿入するような場合、IDENTITY カラムに大きな欠番が生じます。

`identity grab size` は、連続するローとして挿入するレコードのグループの中で最も大きなものに対応できるよう十分に大きな値に設定することをおすすめします。

identity reservation size

要約	
デフォルト値	1
値の範囲	1 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

`identity reservation size` は、`identity` 値の数に制限を設定します。

idle migration timeout

要約	
デフォルト値	60
有効な値	0 ~ 32767
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	共有ディスク・クラスタ

`idle migration timeout` は、クライアントに送信されたマイグレーション要求を無効にせずにアイドル接続を閉じるまでの時間を指定します。これにより、アイドル状態のクライアント接続がマイグレートするのを待つことなく指定時間後にインスタンスを停止できます。

`idle migration timeout` を高い値に設定すると、適切な停止が遅くなります。これは、クライアントがマイグレーションを開始することのないマイグレーション要求を発行したアイドル接続すべてについて、インスタンスが指定時間だけ待たなければならないからです。

job scheduler interval

要約	
デフォルト値	1 (分単位)
値の範囲	1 ~ 600
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

`job scheduler interval` は、実行すべきスケジュール・ジョブはどれかを Job Scheduler がチェックする間隔を設定します。

job scheduler tasks

要約	
デフォルト値	4
値の範囲	1 ~ 640
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

`job scheduler tasks` は、Job Scheduler を介して同時に実行できるジョブの最大数を設定します。

注意 `job scheduler tasks` の値を増やす場合は、Job Scheduler が起動する前に、`number of user connections` の値も `job scheduler tasks` の値の 2 倍以上に増やす必要があります。

しかし、同じ Adaptive Server で Job Scheduler とジョブをホストする場合は、Job Scheduler が起動する前に、`number of user connections` の値を `job scheduler tasks` の値の 3 倍に増やす必要があります。

`number of user connections` を増やすと、Adaptive Server で `max memory` の値を増やす必要が生じる場合があります。

js job output width

要約	
デフォルト値	80
値の範囲	1 ~ 32768
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

js job output width は、js_output テーブルに保存されるジョブに対して出力が使用する行の長さを決定します。

kernel mode

要約	
デフォルト値	スレッド
値の範囲	threaded または process
ステータス	静的
表示レベル	基本
必要な役割	システム管理者
設定グループ	SQL Server 管理

kernel mode は、Adaptive Server のカーネルがスレッドまたはプロセスのどちらのモードを使用するかを決定します。スレッド・モードでは、Adaptive Server はオペレーティング・システム・スレッドを使用し、Adaptive Server エンジンと特殊なタスクをサポートします。このモードでは、Adaptive Server はオペレーティング・システムで実行する単一プロセスです。プロセス・モードでは、Adaptive Server は個々のプロセスを使用し、Adaptive Server エンジンをサポートします。このモードでは、各エンジンはオペレーティング・システムで実行する排他プロセスです。このプロセス・モードは、バージョン15.7 以前の Adaptive Server が使用したのと同じカーネルです。

注意 Adaptive Server は Windows プラットフォームのスレッド・カーネル・モードのみサポートします。

kernel mode の値は文字データです。そのため、数値でなければならない 2 番目の sp_configure パラメータのプレースホルダとして 0 を使用する必要があります。3 番目のパラメータに、threaded、または process のいずれかを指定します。「[sp_configure による単位の指定](#)」(60 ページ)を参照してください。

Sybase は運用サーバにスレッド・モードが使用されることを前提とし、下位互換性のためにプロセス・モードを含めています。プロセス・モードでは、Adaptive Server 15.7 以降の機能でサポートされていないものもあります。

『システム管理ガイド 第2巻』の「第5章 マルチプロセッサ・サーバの管理」を参照してください。

kernel resource memory

要約	
デフォルト値	<ul style="list-style-type: none"> • Solaris – 6184 • その他のプラットフォーム – 4096
値の範囲	75 ~ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	物理メモリ

kernel resource memory は、すべてのスレッド・プールと他のカーネル・リソースを割り当てるカーネル・リソース・メモリ・プールのサイズを 2K ページで決定します。kernel resource memory が使用できるメモリの量は、max memory の値によって変わります。kernel resource memory に割り当てる十分なメモリが max memory にない場合、増やす必要のある max memory の値を示すエラー・メッセージが表示されます。

kernel resource memory に必要な量は、主として max online engines と number of user connections の組み合わせで決定されます。エンジン数が 8 以下の設定の場合、100 以上のユーザ接続の 2 つごとに kernel resource memory の 1 ページを追加すること、エンジン数が 9 以上の設定の場合、各ユーザ接続に対して kernel resource memory の 1 ページを追加することをおすすめします。

たとえば、エンジン数が 6、ユーザ接続が 200 のサーバでは、デフォルト値に値 100 を追加する必要があります。エンジン数が 16、ユーザ接続が 10000 のサーバでは、デフォルト値に値 10000 を追加すればよいでしょう。

license information

要約	
デフォルト値	25
有効な値	0 ~ 2 ³¹
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

`license information` パラメータは、Sybase のシステム管理者が Adaptive Server で使用されているユーザ・ライセンスの数をモニタリングできるようにします。このパラメータを有効にした場合でも、発行されたライセンスの数がモニタリングされるだけで、ライセンス契約が強制されることはありません。

`license information` を 0 に設定すると、Adaptive Server はライセンスの使用をモニタリングしません。`license information` を 0 より大きな値に設定すると、ハウスキーピング・ジョブ・タスクが Adaptive Server のアイドル・サイクル中に使用されたライセンスの数をモニタリングします。`license information` は、使用しているライセンス契約で指定されたライセンスの数に設定してください。

使用されているライセンスの数が `license information` に設定されている数より大きい場合、Adaptive Server はエラー・ログに次のエラー・メッセージを書き込みます。

```
WARNING: Exceeded configured number of user licenses
```

24 時間ごとに、その時間中に使用されたライセンスの最大数が `syblicenseslog` テーブルに追加されます。Adaptive Server を再起動すると、この 24 時間の周期がリセットされます。

『セキュリティ管理ガイド』の「第 3 章 Adaptive Server のログイン・アカウントとデータベース・ユーザの管理」を参照してください。

lock address spinlock ratio

要約	
デフォルト値	100
値の範囲	1 ~ 2147483647
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ロック・マネージャ

Adaptive Server で複数のエンジンを実行する場合に、`address lock spinlock ratio` を使用して、1 つのスピロックによって保護される内部アドレス・ロックのハッシュ・テーブルのローの数を設定します。

Adaptive Server は、1031 のローを持つ内部ハッシュ・テーブル (ハッシュ・バケットと呼ばれる) を使用して、アドレス・ロックの取得と解放を管理します。このテーブルは、1 つまたは複数のスピロックを使用して、異なるエンジンで実行しているプロセス間のアクセスを直列化できます。

`address lock spinlock ratio` のデフォルト値により、アドレス・ロックのハッシュ・テーブルに対して 11 個のスピロックが定義されます。最初の 10 個のスピロックはそれぞれ 100 のローを保護し、11 番目のスピロックは残りの 31 のローを保護します。`address lock spinlock ratio` に 1031 以上の値を指定すると、テーブル全体に対してスピロックが 1 つだけ使用されます。

lock hashtable size

要約	
デフォルト値	2048
値の範囲	1 ~ 2147483647
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ロック・マネージャ、メモリ使用

lock hashtable size は、ロック・ハッシュ・テーブル内のハッシュ・バケットの数を指定します。このテーブルによって、すべてのロー、ページ、テーブルのロックとロック要求が管理されます。タスクがロックを取得するたびに、ロックはハッシュ・バケットに割り当てられ、そのロックへのロック要求は、該当するハッシュ・バケットをチェックします。この値を小さくすると、各ハッシュ・バケット内のロック数が増え、検索時間が長くなります。複数のエンジンを持つ Adaptive Server では、この設定値が小さすぎると、スピンロックの競合が増加する可能性もあります。この値は、デフォルトの 2048 よりも小さくしないでください。

lock hashtable size は、2 の累乗でなければなりません。指定した値が 2 の累乗でない場合には、**sp_configure** は次に大きい 2 の累乗に値を切り上げ、情報メッセージを表示します。

最適なハッシュ・テーブル・サイズは、同時にロックできる個別のオブジェクト (ページ、テーブル、ロー) の数と相関関係があります。ハッシュ・テーブル・サイズは、同時にロックしなければならない個別のオブジェクト数の少なくとも 20 パーセントが最適です。詳細については、『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』を参照してください。

しかし、ユーザの数が非常に多く、ロックの数が不足するのを避けるために **number of locks** パラメータを増加する必要がある場合は、ピーク時に **sp_sysmon** を使用してハッシュ・チェーンの平均長を調べる必要があります。ハッシュ・チェーンの平均長が 4 または 5 より長い場合は、**lock hashtable size** の値を現在設定されている 2 の累乗を 1 つ増やすことを検討してください。

ハッシュ・チェーンの長さは、バルク・コピー・オペレーションのように大量の挿入バッチの間は高くなります。これは、予期されている動作であり、**lock hashtable size** を再設定する必要はありません。

lock scheme

要約	
デフォルト値	allpages
値の範囲	allpages、datapages、datarows
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ロック・マネージャ

lock scheme は、create table と select into でロック・スキームが指定されていない場合に使用するデフォルトのロック・スキームを設定します。

ロック・スキームの値は文字データです。そのため、数値でなければならない 2 番目のパラメータのプレースホルダとして 0 を使用する必要があります。3 番目のパラメータに、allpages、datapages、datarows のいずれかを指定します。

```
sp_configure "lock scheme", 0, datapages
```

lock shared memory

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	物理メモリ

lock shared memory は、Adaptive Server のページをディスクにスワップさせないようにし、オペレーティング・システム・カーネルがサーバの内部ページ・ロック・コードを回避できるようにします。これにより、コストのかかるディスクの読み込みを減らせます。

すべてのプラットフォームが、共有メモリのロックをサポートしているわけではありません。プラットフォームでサポートされていても、パーミッションの設定の誤りや物理メモリの不足などが原因で、lock shared memory パラメータを設定できないことがあります。共有メモリのロックについては、使用しているプラットフォームに対応するオペレーティング・システムのガイドを参照してください。

lock spinlock ratio

要約	
デフォルト値	85
値の範囲	1 ~ 2147483647
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ロック・マネージャ、メモリ使用

Adaptive Server は、設定可能な数のハッシュ・バケットを持つ内部ハッシュ・テーブルを使用してロックの取得と解放を管理します。対称型マルチプロセッシング・システムでは、このハッシュ・テーブルは 1 つまたは複数のスピンロックを使用して、異なるエンジンで実行しているプロセス間のアクセスを直列化できます。ハッシュ・バケットの数を設定するには、[lock hashtable size](#) を使用します。

Adaptive Server が複数のエンジンを実行する場合は、[lock spinlock ratio](#) (スピンロック率) によって、1 つのスピンロックで保護されるロック・ハッシュ・バケットの数が決まります。[lock hashtable size](#) の値を大きくすると、スピンロックの数は増えますが、1 つのスピンロックで保護されるハッシュ・バケットの数は変わりません。

Adaptive Server の [lock spinlock ratio](#) のデフォルト値は 85 です。[lock hashtable size](#) がデフォルト値の 2048 に設定されている場合、デフォルトのスピンロック率から計算されるロック・ハッシュ・テーブルに対するスピンロックの数は 26 となります。『システム管理ガイド 第 2 巻』の「第 5 章 マルチプロセッサ・サーバの管理」を参照してください。

[sp_sysmon](#) は、ロック・ハッシュ・テーブルのハッシュ・チェーンの平均長についてレポートします。『パフォーマンス&チューニング・シリーズ:sp_sysmon による Adaptive Server の監視』を参照してください。

lock table spinlock ratio

要約	
デフォルト値	20
値の範囲	1 ~ 2147483647
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ロック・マネージャ

Adaptive Server で複数のエンジンを実行する場合に、**table lock spinlock ratio** を使用して、1 つのスピロックによって保護される内部テーブル・ロックのハッシュ・テーブルのロー数を設定します。

Adaptive Server は、101 のローを持つハッシュ・テーブル (ハッシュ・バケット) を使用して、テーブル・ロックの取得と解放を管理します。このテーブルは、1 つまたは複数のスピロックを使用して、異なるエンジンで実行しているプロセス間のアクセスを直列化できます。

Adaptive Server の **lock table spinlock ratio** のデフォルト値は 20 で、このときテーブル・ロック・ハッシュ・テーブルに対して 6 個のスピロックが定義されます。最初の 5 個のスピロックはそれぞれ 20 ローを保護し、6 番目のスピロックは最後のローを保護します。**lock table spinlock ratio** に 101 以上の値を指定すると、Adaptive Server はテーブル全体に対してスピロックを 1 つだけ使用します。

lock timeout pipe active

要約	
デフォルト値	1 (オフ)
値の範囲	0 (オン)、1 (オフ)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	モニタリング

lock timeout pipe active は Adaptive Server でロック・タイムアウト・メッセージを収集するかどうかを制御します。**lock timeout pipe active** と **lock timeout pipe max messages** を有効にすると、Adaptive Server は発生する各ロック・タイムアウトのデータを収集します。

monLockTimeout モニタ・テーブルからロック・タイムアウト・メッセージを取得します。

lock timeout pipe max messages

要約	
デフォルト値	0
値の範囲	0 - 2147483648
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	モニタリング

`lock timeout pipe max messages` はロック・タイムアウト・パイプの各エンジンのローの最大数を制御します。これは `monLockTimeout` モニタリング・テーブルによって返すことができるローの最大数を決定します。

lock wait period

要約	
デフォルト値	2147483647
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ロック・マネージャ

`lock wait period` は、テーブル、データ・ページ、データ・ローに対するロックを取得するまでのタスクの待機時間 (秒) を制限します。指定した時間内にロックが取得されなかった場合、Adaptive Server はエラー・メッセージ 12205 をユーザに返し、トランザクションをロールバックします。

`set` コマンドの `lock wait` オプションは、タスクがロックを待機する秒数をセッション・レベルで設定します。これを設定すると、そのセッションに対するサーバ・レベルの設定が無効になります。

`lock wait period` は、セッション・レベルの設定 `set lock wait nnn` と併用され、ユーザ定義テーブルにのみ有効です。これらの設定は、システム・テーブルには影響しません。

デフォルトでは、すべてのプロセスはロックを取得するまで無制限に待機します。デフォルト値に戻すには、値を 2147483647 にリセットするか、または次のように入力します。

```
sp_configure "lock wait period", 0, "default"
```

log audit logon failure

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	エラー・ログ

`log audit logon failure` は、イベント・ロギングが有効になっている場合に、Adaptive Server へのログインの失敗を Adaptive Server のエラー・ログ (Windows サーバの場合は Windows イベント・ログにも) に記録するかどうかを指定します。

log audit logon success

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	エラー・ログ

`log audit logon success` は、イベント・ロギングが有効になっている場合に、Adaptive Server へのログインの成功を Adaptive Server のエラー・ログ (Windows サーバの場合は Windows イベント・ログにも) に記録するかどうかを指定します。

max async i/os per engine

要約	
デフォルト値	プラットフォームに依存する
値の範囲	1 ~ プラットフォーム依存値
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	O/S リソース

`max async i/os per engine` パラメータは、同時に存在できる処理待ちの非同期ディスク I/O 要求の、エンジン当たりの最大数を指定します。

Linux プラットフォームの場合

Linux プラットフォームの場合、`max async i/os per engine` はオペレーティング・システムが予約している非同期 I/O の数を制御します。

スレッド・モードでは、Adaptive Server はエンジン・サーバ用に `max async i/os per engine` の値を予約します。プロセス・モードでは、Adaptive Server は各エンジン用に `max async i/os per engine` の値を予約します。たとえば、`max async i/os per engine` を 4096 に設定し、エンジン数が 4 の場合、スレッド・モードでは、Adaptive Server は 4096 回の I/O を予約しますが、プロセス・モードでは 16384 回の I/O を予約します。

デフォルト値より大きい数値を使用することで、使用しているシステムに利点をもたらすことができる可能性があります。

`sp_sysmon` は、`max async i/os per engine` の調整に役立ちます。`sp_sysmon` の `disk i/o section` には、サンプル期間中の各エンジンの未処理 I/O の最大数情報と、エンジンやオペレーティング・システムの制限によって処理が遅延した I/O の数に関する情報が含まれています。通常、I/O がエンジン制限で遅延した場合は、`max async i/os per engine` の値を大きくする必要があります。

Adaptive Server がデバイスに対して非同期 I/O を実行できるかどうかは、このデバイスがカーネル非同期 I/O (KAIO: kernel asynchronous I/O) をサポートしているかどうかによって決まります。Linux カーネルでは、ファイル・システム・レベルで KAIO サポートを実装する必要があります。ext3、xfs、jfs、ロー・デバイスなど、主要なほとんどのファイル・システムで KAIO がサポートされています。tmpfs ファイル・システムでは KAIO をサポートしていません。Adaptive Server は、KAIO をサポートしていないデバイスに対して非同期 I/O を実行できません。その場合、デバイスを標準の同期 I/O に戻してすべての読み書きを処理します。Adaptive Server は、デバイスが同期 I/O に切り替わったことを示す次のようなメッセージをエラー・ログに記録します。

```
00:00000:00001:2006/12/15 11:47:17.98 kernel Virtual device
'/dev/shm/tempdb.dat' does not support kernel asynchronous i/o. Synchronous i/o
will be used for this device.
```

max async i/os per server

要約	
デフォルト値	プラットフォームに依存する
値の範囲	1～プラットフォーム依存値
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	O/S リソース

`max async i/os per server` は、同時に存在できる処理待ちの非同期ディスク I/O 要求の、Adaptive Server 当たりの最大数を指定します。この制限には、Adaptive Server 当たりのオンライン・エンジンの数は影響しません。`max async i/os per engine` は、エンジン当たりの未処理 I/O の数を制限します。

ほとんどのオペレーティング・システムは、一度に処理できる非同期ディスク I/O の数を制限しています。オペレーティング・システムのプロセス当たりの数を制限するか、システム当たりの数を制限するか、あるいはその両方を制限するかは、オペレーティング・システムによって異なります。アプリケーションがこれらの制限を超えると、オペレーティング・システムはエラー・メッセージを表示します。オペレーティング・システムの呼び出しは比較的成本がかかるので、オペレーティング・システムによって拒否されるような非同期 I/O を Adaptive Server が実行しようとするのは効率的ではありません。

これを避けるために、Adaptive Server はエンジンとサーバのそれぞれについて、処理待ちの非同期 I/O の数を常に把握しています。あるエンジンが発行した非同期 I/O によって `max async i/os per engine` と `max async i/os per server` のどちらかの制限を超えることがわかった場合は、処理待ちの I/O の処理が完了して制限を下回るまで、その I/O を遅延させます。

たとえば、システム当たりの非同期 I/O 数が 200、プロセス当たりの非同期 I/O 数が 75 というオペレーティング・システムの制限があり、Adaptive Server に 3 つのオンライン・エンジンがあるとします。そして、現在の全エンジンの保留中非同期 I/O 数の合計は 200 で、その内訳は次の表のとおりであるとします。

エンジン	保留中の I/O 数	結果
0	60	エンジン 0 は、サーバ当たりの総数がオペレーティング・システムの「システム当たりの」制限値を下回るまで、以後の非同期 I/O を遅延させる。制限値を下回ると、非同期 I/O の発行を再開する。
1	75	エンジン 1 は、エンジン当たりの総数がオペレーティング・システムの「プロセス当たりの」制限値を下回るまで、以後の非同期 I/O を遅延させる。制限値を下回ると、非同期 I/O の発行を再開する。
2	65	エンジン 2 は、サーバ当たりの総数がオペレーティング・システムの「システム当たりの」制限値を下回るまで、以後の非同期 I/O を遅延させる。制限値を下回ると、非同期 I/O の発行を再開する。

非同期 I/O と同期 I/O の両方とも、すべての I/O はディスク I/O 構造体を必要とするので、処理待ちディスク I/O の総数は `disk i/o structures` の値によって制限されます。Adaptive Server の効率の点では、I/O 要求の数が `max i/os per server` を超えたことが理由で I/O を遅延させるよりも、ディスク I/O 構造体を取得できないことが理由で遅延させる方が、わずかに勝っています。`max async i/os per server` は `disk i/o structures` と同じ値に設定してください。[「disk i/o structures」\(111 ページ\)](#) を参照してください。

非同期 I/O に関するオペレーティング・システムでの制限が調整可能な場合は、Adaptive Server が動作できるように十分大きな値を設定してください。必要だけ大きい値に設定することによる不利益はありません。

サーバ当たりの制限、またはエンジン当たりの制限によって I/O の遅延が発生しているかどうかを確認するには、`sp_sysmon` を使用してください。`sp_sysmon` の結果から、処理待ち要求に関するエンジン当たりまたはサーバ当たりの制限を超えていることがわかった場合は、対応するパラメータの値を大きくします。『パフォーマンス&チューニング・シリーズ: `sp_sysmon` による Adaptive Server の監視』を参照してください。

max cis remote connections

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	コンポーネント統合サービス

max cis remote connections は、コンポーネント統合サービスによって確立できる、リモート・サーバへの Client-Library 接続の最大同時接続数を指定します。

デフォルトでは、コンポーネント統合サービスによって確立できるリモート・サーバへの同時接続数は、ユーザ当たり最大 4 つです。ユーザの最大数を 25 に設定している場合は、コンポーネント統合サービスによって最大 100 の Client-Library 接続を同時に確立できます。

この値がインストール環境のニーズを満たしていない場合は、サーバが一度に確立できる Client-Library 接続の正確な数を指定することにより、この設定を上書きできます。

max concurrently recovered db

要約	
デフォルト値	0
有効な値	1 ~ number of engines at start-up から 1 を引いた値
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	バックアップとリカバリ

max concurrently recovered db は、並列度を決定します。最小値は 1 であり、この場合は逐次リカバリが行われます。デフォルト値の 0 を指定して、セルフチューニング・アプローチを使用させることもできます。最大値は、number of engines at start-up から 1 を引いた値です。max concurrently recovered db の上限値には、number of open databases 設定パラメータの値も影響します。

max memory

要約	
デフォルト値	プラットフォーム依存
値の範囲	プラットフォーム固有の最小値 ~ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、物理メモリ

max memory は Adaptive Server が割り付ける物理メモリの総量の最大値を指定します。**max memory** は、Adaptive Server の現在の設定で使用される合計論理メモリよりも大きくなければなりません。

コンピュータで使用可能なメモリの最大量を使用するように Adaptive Server を設定しても、パフォーマンスが低下することはありません。ただし、そのシステムで Adaptive Server 以外に必要なメモリについても検討してください。このようにしないと、Adaptive Server の起動時に必要なメモリを確保できないことがあります。

『システム管理ガイド 第 2 巻』の「第 3 章 メモリの設定」を参照してください。

Adaptive Server が起動しない場合

allocate max shared memory が 1 に設定されている場合は、Adaptive Server は **max memory** で指定された量のメモリを確保しなければなりません。このメモリ量が確保できなければ、Adaptive Server は起動しません。この場合は、サーバの設定ファイルを編集して **max memory** の値を変更し、Adaptive Server に必要なメモリの量を減らしてください。**max memory** に指定したメモリ量がすべて確保できなくても起動できるように、**allocate max shared memory** の値を 0 に変更することもできます。

また、大量のメモリを必要とする他の設定パラメータの値も減らす必要がある場合もあります。その後で Adaptive Server を再起動すると、新しく指定した量のメモリが使用されます。他の設定パラメータの合計値が **max memory** の値を超えることが理由で Adaptive Server が起動できない場合は、メモリを使用する設定パラメータについて、『システム管理ガイド 第 2 巻』の「第 3 章 メモリの設定」を参照してください。

max native threads per engine

要約	
デフォルト値	50
最大値	50 ~ 1000
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ユーザ環境

注意 max native threads per engine はスレッド・モードでは無視されます。

max native threads per engine は、サーバが1つのエンジンについて生成するネイティブ・スレッドの最大数を定義します。ネイティブ・スレッドの最大数に達すると、ネイティブ・スレッドを必要とする Adaptive Server セッションは他のセッションがネイティブ・スレッドを解放するまでスリープします。

max nesting level

Adaptive Server 15.0.3 とそれ以降では、最大ネスト・レベルが 100 に引き上げられており、デフォルト値は 50 です。

要約	
デフォルト値	50
値の範囲	16 ~ 100
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ユーザ環境

maximum nesting levelは、ストアド・プロシージャとトリガの最大ネスト・レベルを設定します。ネスト・レベルを1つ上げるごとに、追加メモリが 160 バイト必要になります。たとえば、ネスト・レベルを 16 から 26 に上げると、追加で 1600 バイトのメモリが必要になります。

max network packet size

要約	
デフォルト値	2048
値の範囲	512 ~ 65024
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ネットワーク通信

max network packet size は、Adaptive Server と通信するクライアントが要求できる最大ネットワーク・パケット・サイズを指定します。

アプリケーションでネットワークを介して大量のデータを送受信する場合は、大きなパケット・サイズを使用すると、アプリケーションのパフォーマンスを大幅に改善できます。例としては、大量のバルク・コピー操作と、大きな **text** 値、**unitext** 値、**image** 値を読み書きするアプリケーションの 2 つがあります。

通常は、次のようにします。

- 短いクエリを実行するユーザについては、**default network packet size** の値を小さくする。
- 大量のデータを送受信するユーザについては、大きなパケット・サイズを要求できるように、**max network packet size** を十分に大きく設定する。

max network packet size は、**default network packet size** と同じかそれよりも大きくなければなりません。512 バイトの整数倍以外の値を指定した場合は、倍数になるように切り捨てられます。

クライアント・アプリケーションからより大きなネットワーク・パケット・サイズが明示的に要求された場合に、そのサイズのパケットをクライアントが受信できるようにするには、**additional network memory** も設定する必要があります。[「additional network memory」\(77 ページ\)](#) を参照してください。

Open Client Server は、64K を超える大きさのネットワーク・パケットは受信できません。

bcpl と **isql** プログラムで大きなパケット・サイズを使用する方法については、『ユーティリティ・ガイド』の該当するプログラムの項を参照してください。Open Client Library のマニュアルには、可変パケット・サイズの使用法の説明があります。

パケット・サイズを選択

最良のパフォーマンスを得るには、ネットワーク上の基本パケット・サイズに対して効率的なサーバ・パケット・サイズを選択してください。目標は次の 2 つです。

- ネットワークに対するサーバの読み込みと書き込みの数を減らす

- ネットワーク・パケット内の未使用領域を減らして、ネットワーク・スループットを向上させる

たとえば、ネットワーク・パケット・サイズが 1500 バイトならば、Adaptive Server のパケット・サイズを 1024 (512 * 2) に設定すると、1536 (512 * 3) に設定した場合よりもパフォーマンスが向上する可能性が高くなります。図 5-3 は、この状況での 4 とおりのパケット・サイズ設定の結果を示しています。

図 5-3: パケット・サイズを決定する要因

基本ネットワーク・パケット：1500 バイト (オーバーヘッド含まず)

パケット・サイズ 512
 使用 1024 バイト
 未使用 476 バイト
 使用率 68%
 サーバ読み込み 2 回



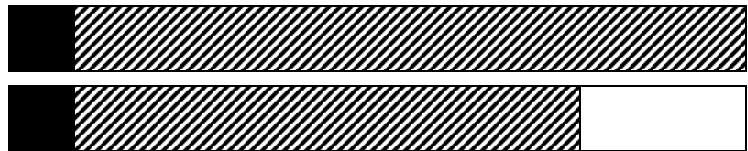
データ量に応じて 1 ネットワーク・パケットに
1 ~ 2 パケット

パケット・サイズ 1024
 使用 1024 バイト
 未使用 476 バイト
 使用率 68%
 サーバ読み込み 1 回



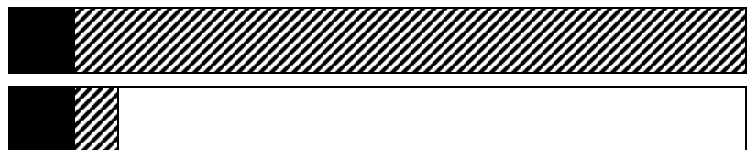
デフォルトの 512 よりパフォーマンスが良くなる

パケット・サイズ 2560
 使用 2560 バイト
 未使用 440 バイト
 使用率 85%
 サーバ読み込み 2 回



この図の中では最良の選択

パケット・サイズ 1536
 使用 1536 バイト
 未使用 1464 バイト
 使用率 51%
 サーバ読み込み 2 回



この図の中では最悪の選択

構文要素：

オーバーヘッド データ

未使用



ネットワーク上の基本パケットの使用可能なデータ領域を計算した後で、ベンチマーク・テストを行い、最適なサイズを決定します。

`max network packet size` の変更が、ネットワーク I/O 管理とタスク切り替えにどのように影響しているかを確認するには、`sp_sysmon` を使用してください。たとえば、`max network packet size` を増やした後で、`sp_sysmon` の出力をチェックすることにより、`bcp` で大きいバッチを処理するときはこの設定がどのように影響するかを確認します。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

max number network listeners

要約	
デフォルト値	5
値の範囲	0 ~ 2147483647
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、ネットワーク通信

`max number network listeners` は、Adaptive Server で同時に使用できるネットワーク・リスナの最大数を指定します。

各マスタ・ポートは、1つのネットワーク・リスナを持ちます。一般に、Adaptive Server が複数のネットワーク・タイプで通信する必要がある場合を除いて、複数のマスタ・ポートを持つ必要はありません。プラットフォームによっては、ソケットと TLI (トランスポート・レイヤ・インタフェース) の両方のネットワーク・インタフェースをサポートするものもあります。サポートされるネットワーク・タイプの詳細については、使用しているプラットフォームの『設定ガイド』を参照してください。

max online engines

要約	
デフォルト値	1
値の範囲	1 ~ 1024 (スレッド・モード) 1 ~ 128 (プロセス・モード)
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、プロセッサ

max online engines はオンラインにできるエンジン・スレッドの数の上限を設定します。起動時に使用可能な CPU の数は考慮されません。したがって、ユーザは後から CPU を増設することができます。このパラメータを各自の SMP 環境に適合するように設定する方法の詳細については、『システム管理ガイド 第 2 巻』の「第 5 章 マルチプロセッサ・サーバの管理」を参照してください。

Adaptive Server の起動時は、1 つのエンジン・スレッドで、すべてのデータベースのリカバリを含む初期化が行われます。初期化での最後のタスクは追加サーバ・エンジンを割り付けることです。それぞれのエンジンは、共有メモリ内の共通データ構造体にアクセスします。

max engines online パラメータを調整する場合は、次の点に注意してください。

- 実装されている CPU の数よりオンラインのエンジン・スレッド数を多くしないでください。
- Adaptive Server がすべてのエンジン・スレッド・プールを同時にオンラインにできるように、**max engines online** を十分大きくする必要があります。
- Adaptive Server 以外のアプリケーションを含めたシステム全体の負荷によっては、Adaptive Server 以外のプロセスの実行用に一部の CPU を残しておくことで最適なスループットを得られることがあります。
- CPU 使用率の低い多数のエンジン・スレッドを実行するより、CPU 使用率の高い少数のエンジン・スレッドを実行する方が、より良いスループットを得ることができます。
- スケーラビリティはアプリケーションによって異なります。アプリケーションについて広範囲なベンチマークを実行して、オンライン・エンジン・スレッドの最良の設定を決定してください。
- プロセス・モードでは、**sp_engine** を使用して、エンジン・スレッドのオフラインとオンラインを切り替えることができます。エンジン 0 を除くすべてのエンジン・スレッドをオフラインにできます。

『パフォーマンス&チューニング・シリーズ：基本』の「第 3 章 エンジンと CPU の使用方法」を参照してください。

max online Q engines

要約	
デフォルト値	0
値の範囲	0 ~ 127
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	プロセッサ

(プロセス・モードのみ) **max online Q engines** は、オンラインで実行できる Q エンジンの最大数を指定するもので、MQ Series に必要です (Active Messaging ユーザーズ・ガイドを参照)。

Q エンジンの最大数は **max online engines** の値によって変わります。

- **max online Q engines** の値は、**max online engines - number of engines at startup** より大きくはできません。たとえば、**max online engines** の値が 57 で、**number of engines at startup** の値が 28 の場合、**max online Q engines** の値は 29 より大きくはできません。
- **max online Q engines** を設定すると、Q エンジンに対して **max online engines** の高範囲が予約されます。**max online Q engines** を一度設定すると、Adaptive Server エンジンは、Q エンジン用に予約された範囲内のエンジンは使用しません。たとえば、**max online engines** を 10 に設定し、**max online Q engines** を 4 に設定すると、以下のようになります。
 - エンジン・グループをエンジン 6、7、8、または 9 にバインドしようとすると、Adaptive Server はエラーを返します。
 - **number of engines at startup** の値を 7、8、9 には変更できません。
 - エンジン 6、7、8、9 をエンジン・グループに追加できません。

この理由は **max online Q engines** の設定が既存の量産環境に影響するため、**max online Q engines** を増やしたのと同じ分だけ **max online engines** の値を増やすことをおすすめします。たとえば、**max online Q engines** を 4 増やす場合は、**max online engines** も 4 増やします。

max parallel degree

要約	
デフォルト値	1
値の範囲	1 ~ 255
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	クエリ・チューニング

max parallel degree は、クエリ当たりの使用可能なワーカー・プロセス数の、サーバワイドの最大値を指定します。これを「最大並列度」と呼びます。

max parallel degree が小さすぎると、クエリのパフォーマンスはそれほど向上しないこともあります。また、**max parallel degree** が大きすぎると、サーバによってコンパイルされたプランに必要なプロセス数が実行時に実際に使用できる数を超過してしまうことや、システムが飽和状態になってスループットが低下してしまうことがあります。並列パーティション・スキャンを有効にするには、クエリを行うテーブル内のパーティション数以上となるようにこのパラメータの値を設定します。

このパラメータの値は、現在の `number of worker processes` の値以下にしてください。

`max parallel degree` を 1 に設定すると、次のようになります。

- Adaptive Server はすべてのテーブルまたはインデックスを逐次スキャンする
- Adaptive Server は逐次クエリ実行を強制し、オプティマイザはこの設定が無効にされている場合よりも並列度の高いプランを選択することがある

`max parallel degree` を変更すると、プロシージャ・キャッシュ内のクエリ・プランはすべて無効になります。新しいプランは、次のストアド・プロシージャまたはトリガの実行時にコンパイルされます。

『パフォーマンス&チューニング・シリーズ：クエリ処理と抽象プラン』の「第9章 並列ソート」を参照してください。

max pci slots

要約	
デフォルト値	0
値の範囲	0 ~ 30
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ユーザ環境

Adaptive Server で許容される PCI スロットの最大数を設定します。以下の値のいずれかです。

- 0、1 – PCA を 1 つ備えたデフォルト・ブリッジ

注意 JVM サポートに必要なのは 1 つのスロットです。スロット数を増やさないでください。

- 2 ~ 30 – 今後のリリース用に割り付け

PCI スロットの詳細については、『Adaptive Server Enterprise における Java』を参照してください。

max query parallel degree

要約	
デフォルト値	1
値の範囲	1 ~ 255
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

Adaptive Server が互換モードのときに使用されます。所定のクエリに使用するワーカー・プロセスの数を定義します。このパラメータが関係するのは、並列処理をグローバルに有効にしない場合だけです。number of worker process の値は、max query parallel degree の値より小さくできません。

『パフォーマンス&チューニング・シリーズ:クエリ処理と抽象プラン』の「第 5 章 並列クエリ処理」を参照してください。

互換モードの詳細については、『マイグレーション技術ガイド』を参照してください。

max repartition degree

要約	
デフォルト値	1
値の範囲	1 ~ max parallel degree の値
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

max repartition degree は、Adaptive Server で必要な動的再分割の数を設定します。これにより、Adaptive Server は、水平並列処理を実行できます。ただし、分割の数が多すぎると、リソースを要求するワーカー・プロセスが大量に発生し、結果的にパフォーマンスが低下します。max repartition degree の値は、これらのリソースのために作成されるパーティションの最大数を決定します。すべてのテーブルとインデックスが非分割である場合は、データの再分割の結果として作成されるパーティションの数として max repartition degree の値が使用されます。

max resource granularity

要約	
デフォルト値	10
値の範囲	1 ~ 100
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

max resource granularity は、システム・リソースの何パーセントまでをクエリが使用できるかを示す上限値を示します。このパラメータは、実行時には使用されませんが、クエリ・オプティマイザにとっての参考値としてのみ使用されるため、クエリ・プロセッサでクエリを並列実行できなくなることはありません。クエリ・エンジンは、**max resource granularity** を参考値として使用することで、メモリの集中使用を回避することができます。

max scan parallel degree

要約	
デフォルト値	1
値の範囲	1 ~ 255
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	クエリ・チューニング

max scan parallel degree は、ハッシュベースのスキヤンの最大並列度をサーバワイドで指定します。これは、次のアクセス・メソッドで使用することができます。

- 分割テーブルと非分割テーブルの並列インデックス・スキヤン
- 非分割テーブルの並列テーブル・スキヤン

max scan parallel degree は、テーブルごとまたはインデックスごとに適用されます。つまり、**max scan parallel degree** が 3 に設定されており、ジョイン・クエリ内の 1 つのテーブルがハッシュ・テーブルを使用してスキヤンされ、別のテーブルがハッシュベースのインデックス・スキヤンによってアクセスされる場合は、このクエリで 9 個のワーカー・プロセスを使用できます (**max parallel degree** が 9 以上に設定されている場合)。

オプティマイザは、非パーティションベースの並列スキャン処理で使用するプロセス数を選択するときに、**max scan parallel degree** をガイドラインとして使用します。これは並列ソートには適用されません。分割によってデータが複数のデバイスに分散することはないので、複数の並列プロセスがスキャン中に同じデバイスにアクセスすることがあります。そのため、ディスクの競合とヘッドの移動が増え、その結果パフォーマンスが低下する場合があります。複数のディスク・アクセスによる問題を防ぐには、**max scan parallel degree** を使用して、テーブルに同時にアクセスできるプロセスの最大数を減らします。

この値が小さすぎると、クエリのパフォーマンスはそれほど向上しません。この値が大きすぎると、サーバによってコンパイルされたプランが使用するプロセス数が多すぎて、ディスク・アクセスの効率が低下してしまうこともあります。一般的には、このパラメータには 2～3 以下の値を設定します。これは、1 つの物理デバイスの I/O を完全に利用するには、ワーカー・プロセスは 2～3 個で十分だからです。

max scan parallel degree の値は、**max parallel degree** の現在値以下に設定してください。**max parallel degree** よりも大きな値を指定すると、Adaptive Server のエラーが返されます。

max scan parallel degree が 1 に設定されているときは、Adaptive Server はハッシュベースのスキャンは実行しません。

max scan parallel degree を変更すると、プロシージャ・キャッシュ内のクエリ・プランはすべて無効になります。新しいプランは、次のストアド・プロシージャまたはトリガの実行時にコンパイルされます。

max SQL text monitored

要約	
デフォルト値	0
値の範囲	0～2147483647
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、モニタリング

max SQL text monitored は、Adaptive Server Monitor と共有しているメモリに SQL テキストを保存するためにユーザ接続ごとに割り付けられるメモリ量を指定します。

バッチ文用に十分なメモリを割り付けないと、表示したいテキストがトランケートされてしまうことがあります。Sybase では、1 ユーザ接続当たりのメモリ初期値を 1,024 バイトにすることをおすすめします。

共有メモリから SQL テキスト用に割り付けられる合計メモリ量は、**max SQL text monitored** にユーザ接続数の現在の設定値を掛けた値です。

max transfer history

要約	
デフォルト値	10
値の範囲	1 ~ 255
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	Adaptive Server 管理

max transfer history は、Adaptive Server が各データベースの **spt_TableTransfer** テーブルに保持する転送履歴の数を制御します。追跡される各テーブルについて、**spt_TableTransfer** は次のように保持します。

$(N \text{ 個の成功したエントリ}) + (N \text{ 個の失敗したエントリ})$

ただし、 N は **max transfer history** の値です。

このパラメータの値を下げても、**spt_TableTransfer** からエントリが自動的に削除されるわけではありません。所定の転送テーブルのエントリが削除されるのは、次回そのテーブルの転送を開始するときです。転送に成功すると、テーブルの成功転送エントリがクリアされます。転送に失敗すると、失敗した転送エントリがクリアされます。

たとえば、ある 1 つのテーブルの **spt_configure** に 12 個の成功履歴エントリと 9 個の失敗履歴エントリがあり、**max transfer history** が 5 に設定された場合、そのテーブルで次回の転送に成功すると、**spt_configure** での成功エントリ数が 5 になりますが、前回の失敗エントリ数 9 はそのまま変わりません。

maximum dump conditions

要約	
デフォルト値	10
値の範囲	10 ~ 100
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	グループ診断

maximum dump conditions は、Adaptive Server が共有メモリ内にデータのダンプを生成する条件の最大数を設定します。

注意 このパラメータは、Sybase 製品の保守契約を結んでいるサポート・センタだけが使用します。このパラメータは、Sybase 製品の保守契約を結んでいるサポート・センタから指示がないかぎり、変更しないでください。

max buffers per lava operator

要約	
デフォルト値	2048
値の範囲	500 ~ 65535
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

maximum buffers per lava operator は、lava 演算子が使用するバッファの数の上限値を設定します。この演算子は、ソートまたはハッシュを実行します (これらは、処理の観点からは「高コスト」です)。lava 演算子はセッションの **tempdb** データ・キャッシュ・プールのバッファを、ローを処理するための作業領域として使用します。

lava 演算子は入力ストリームを再帰処理することがよくあります。ソートでは、残りの実行すべてをマージするのに十分なバッファが確保されるまで、それ以降のマージ・パスが必要とされます。ハッシュでは、残りのデータすべてが 1 つのメモリ内ハッシュ・テーブルにちょうど収まるまで、こぼれ落ちたセットのハッシュ・テーブルを作成するため、それ以降のパスが必要とされます。**max buffers per lava operator** の値を増やした場合、一部のクエリでは必要な I/O が減少します。これは特に、HASH DISTINCT、HASH VECTOR AGGREGATE、HASH UNION の各演算子を使用するクエリに当てはまります。

同時使用ユーザの多いサーバで **maximum buffers per lava operator** のデフォルト値を増やす場合には注意が必要です。Adaptive Server はコストの高い演算子に対してのみより多くのバッファを割り当て、ユーザのテーブルとその他のセッションのワークテーブルのキャッシュで使用できるバッファの数を減らす場合があります。**tempdb** のデータ・キャッシュの効果を分析するには、**sp_sysmon** を使用します。

maximum buffers per lava operator は、**max resource granularity** とともにバッファの使用数を制限します。制限値は、次の項目の最低値に設定されます。

- **max buffers per lava operator** の値、または
- $(\text{max resource granularity}) \times (\text{tempdb のページサイズ} \cdot \text{プールにおけるデータ} \cdot \text{バッファ数})$

ソート・バッファに割り付けられるメモリ量の設定については、「[number of sort buffers](#)」(207 ページ) を参照してください。

maximum failed logins

要約	
デフォルト値	0
値の範囲	-1 ~ 32767
ステータス	動的
表示レベル	10
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

maximum failed logins は、ログインや役割に対するログイン試行の最大回数をサーバワイドで設定するとき 사용됩니다。

値 "-1" は、認証に失敗するたびに **syslogins** カラムの **logincount** のログイン失敗回数は更新されても、アカウントはロックされないことを指定します。一方、値 0 (ゼロ) を使用すると、認証失敗のたびにカラム数が増えることはなく、認証失敗によってアカウントがロックされることもありません。

sp_modifylogin を使用して特定の 1 つの役割に対する最大失敗ログイン回数を変更する方法の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。**alter role** を使用して最大失敗ログイン回数を変更する方法の詳細については、『リファレンス・マニュアル：コマンド』を参照してください。

maximum job output

要約	
デフォルト値	32768
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

maximum job output は、1 つのジョブが生成できる出力の最大サイズをバイト単位で設定します。**maximum job output** で指定された値を上回る出力が生成された場合、超過した分のデータは破棄されます。

memory alignment boundary

要約	
デフォルト値	論理ページ・サイズ
値の範囲	2048 ^a ~ 16384 a. 最小値はサーバの論理ページ・サイズによって決まる
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	キャッシュ・マネージャ

memory alignment boundary は、データ・キャッシュを揃えるメモリ・アドレス境界を決定します。

構造体が特定のメモリ・アドレス境界に揃えられていれば、I/O のパフォーマンスが向上するマシンもあります。この整列を維持するためには、**memory alignment boundary** の値が論理ページ・サイズから 2048K までの範囲にある 2 の累乗でなければなりません。

注意 **memory alignment boundary** がサポートされるのは、特定のハードウェア・プラットフォームだけです。このパラメータは、Sybase 製品の保守契約を結んでいるサポート・センタから指示がないかぎり、変更しないでください。

memory per worker process

要約	
デフォルト値	1024
値の範囲	1024 ~ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用

memory per worker process は、ワーカー・プロセスが使用するメモリ量をバイト単位で指定します。各ワーカー・プロセスは、クエリの処理中にメッセージ用のメモリを要求します。このメモリは、共有メモリ・プールから割り付けられます。このプールのサイズは、**memory per worker process** と **number of worker processes** を乗じた値です。並列処理の使用状況に応じて 2 倍から 4 倍の値に再設定することを必要とする場合があります。**dbcc checkstorage** を使用するとき、**number of worker processes** の値が 1 に設定されている場合は、**memory per worker process** を 1,792 バイトに増やす必要があります。

『システム管理ガイド 第 2 巻』の「第 3 章 メモリの設定」を参照してください。

messaging memory

要約	
デフォルト値	400
値の範囲	60 ~ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、物理メモリ

Sybase メッセージングに使用できるメモリ量を設定します。

metrics elap max

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

metrics elap max は、QP 測定基準の最長経過時間とスレッシュホールドを設定します。

metrics exec max

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

metrics exec max は、QP 測定基準の最長実行時間とスレッシュホールドを設定します。

metrics lio max

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

metrics lio max は、QP 測定基準の最大論理 I/O とスレッシュホールドを設定します。

metrics pio max

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	

metrics pio max は、QP 測定基準の最大物理 I/O とスレッシュホールドを設定します。

min pages for parallel scan

要約	
デフォルト値	200
値の範囲	20 - 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

min pages for parallel scan は、Adaptive Server が並列にアクセスできるテーブルとインデックスの数を制御します。テーブルのページ数が設定値より低い場合、テーブルは逐次アクセスされます。min pages for parallel scan はページ・サイズを考慮しません。Adaptive Server がインデックスとテーブルにアクセスするとき、適切であればデータを再分割します。そして、適切であればスキャン数を超える並列処理を使用します。

minimum password length

要約	
デフォルト値	6
値の範囲	0 ~ 30
ステータス	動的
表示レベル	10
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

minimum password length を使用すると、パスワード値の長さを、サーバワイドで、あるいはログインごとまたは役割ごとにカスタマイズできます。ログインごとまたは役割ごとの **minimum password length** 値は、サーバワイドの設定よりも優先します。**minimum password length** の設定は、値を設定した後に作成するパスワードにのみ影響します。既存のパスワードの長さは変更されません。

minimum password length を使用して指定するサーバワイドの **minimum password length** の値は、ログインと役割の両方に適用されます。たとえば、すべてのログインと役割に対する **minimum password length** を 4 文字に設定するには、次のように入力します。

```
sp_configure "minimum password length", 4
```

ログインの作成時にそのログインの **minimum password length** を設定するには、**sp_addlogin** を使用します。たとえば、パスワードが "Djdiek3" である新しいログイン "joe" を作成し、"joe" の **minimum password length** を 4 に設定するには、次のように入力します。

```
sp_addlogin joe, "Djdiek3", minimum password length=4
```

役割の作成時にその役割の **minimum password length** を設定するには、**create role** を使用します。パスワードが "temp244" である新しい役割 "intern_role" を作成し、"intern_role" の **minimum password length** を 0 に設定するには、次のように入力します。

```
create role intern_role with passwd "temp244", minimum password length 0
```

元のパスワードは 7 文字ですが、**minimum password length** が 0 に設定されているため、変更するパスワードの長さの制限はありません。

既存のログインに対する **minimum password length** を設定または変更するには、**sp_modifylogin** を使用します。**sp_modifylogin** は、ユーザ定義の役割だけに影響し、システム標準の役割には影響しません。たとえば、ログイン "joe" に対する **minimum password length** を 8 文字に設定するには、次のように入力します。

```
sp_modifylogin "joe", @option="minimum password length", @value="8"
```

注意 *value* パラメータのデータ型は **character** です。したがって、数値には引用符が必要です。

すべてのログインに対する **minimum password length** のオーバーライドを 2 文字に変更するには、次のように入力します。

```
sp_modifylogin "all overrides", "minimum password length", @value="2"
```

すべてのログインに対する **minimum password length** のオーバーライドを削除するには、次のように入力します。

```
sp_modifylogin "all overrides", @option="minimum password length", @value="-1"
```

既存の役割に対する **minimum password length** を設定または変更するには、**alter role** を使用します。たとえば、既存の役割 “**physician_role**” に対する **minimum password length** を 5 文字に設定するには、次のように入力します。

```
alter role physician_role set min passwd length 5
```

すべての役割の **minimum password length** を無効にするには、次のように入力します。

```
alter role "all overrides" set minimum password length -1
```

mnc_full_index_filter

要約	
デフォルト値	2
値の範囲	0 ~ 2 <ul style="list-style-type: none"> • 0 – 無効化 • 1 – 有効化 • 2 – 最適化目標設定に従って設定
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

mnc_full_index_filter は、次の項目がある場合に、サーバ・レベルでの制限探索索引数を持たないカバーされていないインデックスを Adaptive Server が考慮しないように指定します。

- インデックス内のカラム
- ヒストグラムがない述語

インテリジェントなインデックス・スキャンがあるデータオンリーロック (DOL: Data-Only-Locked) テーブルに対して `mnc_full_index_filter` を使用できます。これは、インテリジェントなインデックス・スキャンが探索指数を作成する場合でも当てはまります。

`mnc_full_index_filter` の値を変更しても、Adaptive Server で使用するメモリの量は増加しません。

`mnc_full_index_filter` は、どの特定のオプション目標についても最初は有効にされていません。動作させるには、明示的に有効にする必要があります。

msg confidentiality reqd

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

`msg confidentiality reqd` は、Adaptive Server が送受信するメッセージがすべて暗号化されていなければならないことを指定します。メッセージを暗号化するには、`use security services` パラメータを 1 にします。

msg integrity reqd

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

`msg integrity reqd` は、すべてのメッセージのデータ整合性が検査されていなければならないことを指定します。メッセージの整合性検査を行うには、`use security services` を 1 に設定します。`msg integrity reqd` が 1 に設定されている場合、クライアントから Adaptive Server への接続は、クライアントが特定のセキュリティ・サービス (`message integrity`、`replay detection`、`origin checks`、または `out-of-seq checks`) を使用していない場合にきって許可されます。

net password encryption required

要約	
デフォルト値	0
値の範囲	0 ~ 2
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

`net password encryption reqd` は、ログイン認証に RSA 暗号化アルゴリズムまたは Sybase 独自の暗号化表 5-3 アルゴリズムのみを使用するように制限します。

表 5-3: net password encryption reqd の値とその説明

値	説明
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	ネットワーク・ログイン・パスワードの暗号化に、クライアントが RSA または Sybase 独自の暗号化アルゴリズムのみを使用するように制限します。これは、以前に接続があったクライアントには Sybase 独自のアルゴリズムを使用した再接続、新しいクライアントにはより強力な RSA アルゴリズムを使用した接続を可能にする、制限が段階的に強化されていく設定です。パスワードの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できません。
2	ネットワーク・ログイン・パスワードの暗号化に、クライアントが 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。これは、パスワードの強力な RSA 暗号化を実現します。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できません。

ネットワーク・パスワードの暗号化が必須であるために接続を拒否された場合、クライアントは次のようなメッセージを受け取ります。

```
Msg 1640, Level 16, State 2:
Adaptive Server requires encryption of the login
password on the network.
```

number of alarms

要約	
デフォルト値	40
値の範囲	40 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、SQL Server 管理

`number of alarms` パラメータは、Adaptive Server によって割り付けられるアラーム構造体の数を指定します。

Transact-SQL のコマンド **waitfor** は、文ブロック、ストアド・プロシージャ、またはトランザクションを実行するための、特定の時刻、時間の長さ、またはイベントを定義します。Adaptive Server は、**waitfor** コマンドを正しく実行するためにアラームを使用します。他の内部処理にもアラームが必要です。

Adaptive Server が必要とするアラーム数が、現在割り付けられている数よりも多い場合は、次のメッセージがエラー・ログに書き込まれます。

```
uasetalarm: no more alarms available
```

各アラーム構造体に必要なメモリのバイト数は小さい値です。**number of alarms value** の値を非常に大きくする場合は、**max memory** をそれに見合うように設定してください。

number of aux scan descriptors

要約	
デフォルト値	200
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、SQL Server 管理

number of aux scan descriptors パラメータは、サーバ上のすべてのユーザが共有するプールで使用できる補助スキャン記述子の数を設定します。

ユーザ接続とワーカー・プロセスにはそれぞれ 48 のスキャン記述子が排他的に割り付けられています。48 のスキャン記述子のうち、16 個がユーザ・テーブル用に予約され、12 個がワーク・テーブル用に予約され、20 個がシステム・テーブル用に予約されます (この 20 個のうち 4 個はロールバック条件用に確保されます)。クエリによって直接的または間接的に参照されるテーブルごとに 1 つの記述子が必要です。ユーザ・テーブルの場合、テーブル参照に含まれるものは次のとおりです。

- クエリの **from** 句で参照されるすべてのテーブル
- クエリで指定するビューで参照されるすべてのテーブル (ビュー自体はカウントされません)
- サブクエリで参照されるすべてのテーブル
- 参照整合性を検査する必要があるすべてのテーブル (挿入、更新、削除のみに使用されます)
- **select...into** を使用して作成されたテーブル
- クエリ用に作成されたすべてのワークテーブル

セルフジョイン、複数のビュー、または複数のサブクエリなどで同じテーブルが複数回参照される場合、そのテーブルは参照のたびにカウントされます。クエリに `union` が含まれる場合、その `union` クエリの各 `select` 文は個別のスキャンになります。クエリを並列実行する場合は、コーディネーティング・プロセスとそれぞれのワーカー・プロセスにはテーブル参照ごとにスキャン記述子が必要です。

クエリ・スキャンによって参照されるユーザ・テーブルの数が 16 を超える場合やワーク・テーブルの数が 12 を超える場合は、共有プールからのスキャン記述子が割り付けられます。インデックス・スキャンではなくテーブル・スキャンでデータオンリーロック・テーブルがアクセスされる場合も、データオンリーロック・テーブルごとに 1 つのシステム・テーブル記述子が必要です。クエリでテーブル・スキャンを使用してスキャンされるデータオンリーロック・テーブルの数が 16 を超える場合は、補助スキャン記述子がスキャンされたテーブルに割り付けられます。

スキャンで、割り付けられている数を使い果たして補助スキャン記述子が必要になったときに、共有プール内に使用できる記述子がない場合、Adaptive Server はエラー・メッセージを表示し、ユーザ・トランザクションをロール・バックします。

クエリで追加のスキャン記述子が必要となることがなくても、システムの稼働条件が増大した場合に備えて、`number of aux scan descriptors` をデフォルト値の設定のままにしておいてもかまいません。システムのユーザが実行するクエリで 16 を超えるテーブルが絶対に使用されないことと、使用するテーブルに参照整合性制約がほとんどまたはまったくないことが確かな場合のみに、この値を 0 に設定します。[「スキャン記述子の使用率のモニタリング」\(186 ページ\)](#) を参照してください。

クエリがより多くのスキャン記述子を必要とする場合は、次のいずれかの方法に従って問題を解決してください。

- クエリを書き換えるか、テンポラリ・テーブルを使用してクエリをステップに分割する。データオンリーロック・テーブルでは、テーブル・スキャンが多い場合にはインデックスを追加する。
- 使用する参照整合性制約が多い場合は、使用するスキャン記述子が少なくなるようにテーブルのスキーマを再設計する。クエリを実行する前に `set showplan, noexec on` を有効にすることで、クエリがスキャン記述子をいくつ使用するかが確認できる。
- `number of aux scan descriptors` 設定の値を増やす。

以降の項では、記述子の不足を避けるために `sp_monitorconfig` を使用して現在の使用数と最大使用数をモニタリングする方法と、必要なスキャン記述子の数を見積もる方法について説明します。

スキャン記述子の使用率のモニタリング

`sp_monitorconfig` は、未使用 (空き) のスキャン記述子の数、現在使用されている補助スキャン記述子の数、アクティブになっている割合、サーバの最後の起動以降に使用されたスキャン記述子の最大数をレポートします。スキャン記述子の使用状況をモニタリングするには、ピーク時に周期的に実行します。

次に、500 の記述子が設定されているスキャン記述子の使用状況の例を示します。

```
sp_monitorconfig "aux scan descriptors"

Usage information at date and time: Apr 22 2002  2:49PM.
Name                Num_free   Num_active  Pct_act  Max_Used
Reuse_cnt  Instance_Name
-----
number of aux                260           240    48.00       427
              0                NULL
```

補助スキャン記述子のうち 240 だけが使用され、260 が未使用のままです。ただし、Adaptive Server の最後の起動以降の特定の時点におけるスキャン記述子の最大数は 427 で、残り約 20 パーセント分は、使用が増加したり、例外的に大量に使用される期間が発生したりしても対応できます。“Re-used” はスキャン記述子には適用されません。

補助スキャン記述子の見積もりと設定

スキャン記述子の使用数を見積もるには、次の手順に従います。

- 1 `set showplan` と `set noexec` を有効にしてクエリを実行することによって、17 以上のユーザ・テーブルを参照するクエリや、多数の参照整合性制約を持つテーブルを参照するクエリでのテーブル参照の数を調べます。補助スキャン記述子が要求されると、`showplan` によって必要な数がレポートされます。

```
Auxiliary scan descriptors required: 17
```

レポートされる数には、すべてのワーカー・プロセスに必要なものも含め、クエリに必要な補助スキャン記述子のすべてが含まれています。クエリに参照整合性制約のみが関係する場合は、`sp_helpconstraint` も使用できます。これは、テーブルごとの参照整合性制約の数の総数を表示します。

- 2 補助スキャン記述子を使用するクエリごとに、そのクエリを同時に実行するユーザの数を見積もり、その数を乗算します。補助スキャン記述子が 8 個必要なクエリを実行するユーザ数を 10 と想定すると、合計 80 個が一度に必要になります。
- 3 クエリごとの結果を合計して、必要な補助スキャン記述子の数を計算します。

number of backup connections

要約	
デフォルト値	0
値の範囲	1 ~ 32768
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	ユーザ環境

number of backup connections は、Backup Server がインメモリ・データベースをダンプまたはロードするために確立するユーザ接続の最大数を設定します。**number of backup connections** の値は、アーカイブされたデータベース当たりの最大ストライプ数を制限します。これが必要なのは、**dump** または **load database** が実行されるときに Backup Server ではストライプ当たり 1 つのユーザ接続が必要とされ、しかも **dump database** コマンドの実行にもう 1 つ追加の接続が必要とされるからです。

number of backup connections は制限値にすぎないので、リソースを消費しません。**number of backup connections** が 0 に設定されると、Backup Server はユーザ接続の最大数を使用できます。

number of ccbs

要約	
デフォルト値	0
値の範囲	0 ~ 100
ステータス	静的
表示レベル	
必要な役割	
設定グループ	診断

今後のために予約済み。

number of checkpoint tasks

要約	
デフォルト値	1
有効な値	1 ~ 8
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	バックアップとリカバリ

`number of checkpoint tasks` は、並列チェックポイントを設定します。チェックポイント・タスクの値は、`number of engines at start-up` の値以下にする必要があります。最大値は、設定パラメータ `number of engines online at startup` と `number of open databases` の値によって制限され、最大値は 8 です。

デフォルト値は、逐次チェックポイントをデフォルト動作として設定します。

number of devices

要約	
デフォルト値	10
値の範囲	1 ~ 2,147,483,647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	ディスク I/O、メモリ使用

`number of devices` は、Adaptive Server が使用できるデータベース・デバイスの数を制御します。これは、データベースまたはトランザクション・ログ・ダンプ用に使用されるデバイスを含みません。

`disk init` を実行するときに、仮想デバイス番号 (`vdevno`) を割り当てることもできます。ただし、この値はオプションです。`vdevno` を割り当てなかった場合には、次に利用可能な仮想デバイス番号が割り当てられます。

仮想デバイス番号を割り当てる場合、各デバイス番号は Adaptive Server が使用するデバイス番号の中でユニークでなければなりません。番号 0 はマスタ・デバイス用に予約されています。値の有効範囲内にある未使用のデバイス番号を入力できます。

現在使われている番号を調べるには、次のように入力します。

```
select vdevno from master..sysdevices
where status & 2 = 2
```

ここで、“status 2” は物理ディスクを示します。

UNIX でのデバイス数の設定

UNIX プラットフォームで多数のデバイスを使用する場合は、設定ファイルで適切なデバイス数とユーザ接続数を設定することをおすすめします。大量のデバイスを `sp_configure` を使用して動的に設定する処理は失敗する場合があります。

`number of network connections` と `number of devices` 設定パラメータはオペレーティング・システム・ファイル記述子を使用します。これらの設定パラメータのいずれかに大きな値を使用するには、オペレーティング・システム・ファイル記述子の数を増やす必要があります。

多数のデバイスの構成要素は、プラットフォームにより異なります。Adaptive Server 用のファイル記述子の数には、以下のような上限があります。

- HPIA 64、HPPA 64、AIX 64、Windows – Adaptive Server が設定
- IBM RISC 64 ビット – 2000
- IBM PLinux – 2048
- AMD64 Linux – 10000
- IA32 ビット Linux – 10000
- IA 64 ビット Linux – 1024

注意 ファイル記述子の数の変更を有効にするには、Adaptive Server を再起動します。

ファイル記述子の数が不十分な場合、Adaptive Server は次のようなエラー・メッセージを発行します。

```
Msg 5893, Level 16, State 1:Procedure 'sp_configure', Line 1234:
The sum, ( (number of user connections) + (number of remote sites) + (max cisremote
connections) + (number of java sockets)) / (max online engines)) + (number of devices )
+ (max number network listeners),
must be no greater than '958'.
Msg 5849, Level 16, State 1:Procedure 'sp_configure', Line 1234:
Verification failed for parameter 'number of devices'.
(return status = 1)
```

次に、コマンド `ulimit -n 2048` でファイル記述子の数を 2048 に設定し、サーバを再起動する場合、**number of devices** の値を増やすと、Adaptive Server はエラー・ログに次のメッセージをレポートします。

```
kernel Using 2048 file descriptors.
```

number of disk tasks

要約	
デフォルト値	1
値の範囲	UNIX : 1 ~ 64 Windows : 1
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	ディスク I/O、プロセッサ

number of disk tasks はポーリングおよびディスク I/O の完了のための専用タスクの数を制御します。

number of dtx participants

要約	
デフォルト値	500
有効な値	100 ~ 2147483647
ステータス	動的
表示レベル	10
必要な役割	システム管理者
設定グループ	DTM 管理、メモリ使用

number of dtx participants は、Adaptive Server トランザクション・コーディネーション・サービスが同時に送信とコーディネートができるリモート・トランザクションの総数を設定します。DTX パティシパントは、コーディネーション・サービスがリモート・トランザクション分岐の管理に使用する内部メモリ構造です。トランザクションがリモート・サーバに送信されるとき、コーディネーション・サービスはその分岐を管理するために新しい DTX パティシパントを取得する必要があります。

number of dtx participants の設定値をデフォルト値より小さくすると、サーバが管理できるリモート・トランザクションの数が減少します。使用できる DTX パティシパントがない場合、新しい分散トランザクションは開始できません。新しいリモート・トランザクションを送信するために使用できる DTX パティシパントがない場合は、進行中の分散トランザクションがアボートすることがあります。

number of dtx participants の設定値をデフォルト値より大きくすると、Adaptive Server が処理できるリモート・トランザクション分岐の数が増加しますが、メモリの消費量も増加します。

使用しているシステムに対する DTX パティシパント数の最適化

ピークの時間帯に、`sp_monitorconfig` を使用して DTX パティシパントの使用状況を調査します。

```
sp_monitorconfig "number of dtx participants"

Usage information at date and time: Apr 22 2002 2:49PM.
Name                               Num_free  Num_active  Pct_act  Max_Used
Reuse_cnt  Instance_Name
-----
number of dtx                               80          20        4.00          210
0                               NULL
```

`num_free` の値がゼロまたは非常に小さい場合は、新しい分散トランザクションは DTX パティシパントの不足により、開始できない可能性があります。この場合は、**number of dtx participants** の値を増やしてください。

Max_used の値が低い場合、未使用の DTX パティシパントが他のサーバ機能によって使用できるメモリを消費している可能性があります。この場合には、number of dtx participants の値を減らしてください。

number of dump threads

要約	
デフォルト値	無効
値の範囲	1 (無効、並列なし) ~ 8 (完全に並列)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	グループ診断

number of dump threads は、Adaptive Server がメモリ・ダンプを実行するために生成するスレッド数を制御します。ダンプ・スレッド数を適切な値にすると、メモリ・ダンプ中にエンジンが停止する時間を短縮できます。

メモリのスレッド数を決めるときは、次の点を考慮してください。

- マシンのファイル・システム・キャッシュにメモリ・ダンプ全体を格納するのに十分な空きメモリがある場合は値 8 を使用します。
- マシンに十分な空きメモリがあるかどうかわからない場合は、ダンプ・スレッド数の値は、I/O システムの速度、ディスクの速度、コントローラのキャッシュ、ダンプ・ファイルがいくつかのディスクで作成された論理ボリューム・マネージャに存在するかどうかなど、多くの要因によって決まります。
- メモリ・ダンプの実行中にエンジンを停止しない場合は、次に説明するように、並列処理を無効化 (値を 1 に設定) します。

Adaptive Server がメモリ・ダンプを実行するとき、作成されるファイル数は、割り当てたメモリ・セグメントの合計数と設定したスレッド数を掛けた値になります。Adaptive Server は、別々のスレッドを使用して別々のファイルに書き込みます。ジョブが完了すると、エンジンが再起動され、これらのファイルがターゲット・ダンプ・ファイルにマージされます。このため、共有メモリを並列でダンプするときにかかる時間は逐次処理よりも長くなります。

- メモリのダンプ中にエンジンを停止する場合は、1 以外の値を使用すると、メモリ・ダンプ時のエンジンの停止時間を短縮できます。

number of engines at startup

要約	
デフォルト値	1
値の範囲	1 ~ マシンの CPU 数
ステータス	静的
表示レベル	基本
必要な役割	システム管理者
設定グループ	Java サービス、メモリ使用、プロセッサ

注意 スレッド・モード用に設定すると、Adaptive Server は **number of engines at startup** 設定パラメータを無視します。スレッド・モードでは、Adaptive Server は定義済みのスレッド・プールのサイズを使用して起動時のオンライン・エンジン数を決定します。設定ファイルにスレッド・プール設定情報が含まれていない場合 (たとえば、以前のバージョンからのアップグレード中など)、Adaptive Server は **number of engines at startup** の既存の値を使用してデフォルトのエンジン・プールのサイズを決定します。

number of engines at startup は、起動時にのみ使用されるパラメータで、オンラインにするエンジンの数を設定します。このパラメータにより大きな柔軟性が得られます。ただし、マシンに搭載された CPU の数を超える値や、**max online engines** の設定値を超える値を **number of engines at startup** で設定することはできません。起動後にエンジンをオンラインに切り替えることをしない場合は、**max online engines** と **number of engines at startup** を同じ値に設定します。**number of engines at startup** と **max online engines** の値が異なる場合は、1 つのエンジンにつき約 1.8 MB のメモリが浪費されます。

Adaptive Server では、ユーザがエンジン 0 を除くすべてのエンジンをオフラインにできます。

number of histogram steps

要約	
デフォルト値	20
値の範囲	3 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

number of histogram steps は、ヒストグラム内のステップ数を指定します。

number of index trips

要約	
デフォルト値	0
値の範囲	0 ~ 65535
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	キャッシュ・マネージャ

number of index trips は、古くなったインデックス・ページが何回 MRU/LRU (Most Recently Used/Least Recently Used) チェーンを周回してから追い出されるかを指定します。**number of index trips** パラメータの値を増やすに従って、インデックス・ページがキャッシュ内にとどまる時間が長くなります。

データ・キャッシュは MRU/LRU チェーンとして実装されます。ユーザのスレッドがデータ・ページとインデックス・ページにアクセスすると、これらのページはキャッシュの MRU/LRU チェーンの MRU 側の端に置かれます。トランザクション量の多い環境や一部のベンチマークでは、インデックス・ページがすぐに再び必要になる可能性が高いため、キャッシュ内にとどめておいても良いでしょう。**number of index trips** の設定値が大きいほどインデックス・ページはキャッシュ内に長くどまり、設定値が小さければインデックス・ページはすぐにキャッシュから追い出されます。

リラックス LRU ページに対しては、**number of index trips** を設定する必要はありません。『システム管理ガイド 第 2 巻』の「第 4 章 データ・キャッシュの設定」を参照してください。

注意 インデックスが使用するキャッシュが比較的少量で (特に他のオブジェクトと領域を共有する場合)、トランザクション量が多い場合は、**number of index trips** をあまり大きな値に設定しないでください。追い出されないページでキャッシュがいっぱいになり、プロセスがキャッシュ領域の空きを待っている間にタイムアウトになってしまう可能性があります。

number of index trips の値を 0 以外に変更する前に、すべてのインデックス、OAM、データ・ページを保管できるだけの十分なキャッシュがアプリケーションにあることを確認してください。**number of index trips** の値を変更する前に、Sybase 製品の保守契約を結んでいるサポート・センタまで連絡してください。

number of java sockets

要約	
デフォルト値	0
有効な値	0 ~ 32767
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	Java サービス、メモリ使用

number of java sockets は、Sybase がサポートする Java VM および java.net クラスを有効にします。

number of large i/o buffers

要約	
デフォルト値	6
有効な値	1 ~ 256
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ディスク I/O、メモリ使用、SQL Server 管理

number of large i/o buffers は、特定の Adaptive Server ユーティリティにおける大容量 I/O の実行用に予約される、アロケーション・ユニット・サイズのバッファの数を設定します。大容量 I/O バッファは主に **load database** コマンドで使用されます。このコマンドは、指定されたストライプ数に関係なく、1つのバッファを使用してデータベースをロードします。これらのバッファは **load transaction** では使用されません。6個を超える **load database** コマンドを同時に実行するには、**load database** コマンドごとに1つの大容量 I/O バッファを設定します。

create database と **alter database** は、データベース・ページをクリアするときに、これらの大容量 I/O バッファを使用します。**create database** および **load database** の各インスタンスは、最大 32 個の大容量 I/O バッファを使用できます。

これらのバッファは、ディスク・ミラーリングといくつかの dbcc コマンドによっても使用されます。

注意 Adaptive Server バージョン 12.5.0.3 以降では、大容量 I/O バッファのサイズは 1 エクステント (8 ページ) ではなく 1 アロケーション (256 ページ) です。このため、大容量バッファに対してこれまでより大きなメモリ割り付けが必要です。たとえば、以前のバージョンでディスク・バッファとして 8 ページのメモリが必要だった場合に、今回のバージョンでは 256 ページのメモリが必要となります。

number of locks

要約	
デフォルト値	5000
値の範囲	1000 ~ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	ロック・マネージャ、メモリ使用

number of locks は、Adaptive Server のすべてのユーザが使用できるロックの総数を設定します。

Adaptive Server が必要とするロックの総数は、同時プロセスと並列プロセスの数、およびトランザクションが実行するアクションのタイプによって異なります。ある時点でどれくらいのロックが使用されているかを調べるには、**sp_lock** を使用します。

逐次処理の場合は、初めはアクティブな同時接続 1 つ当たり 20 のロック数を割り当てることをおすすめします。

並列処理に必要なロックの数は逐次処理よりも多くなります。たとえば、クエリが平均で 5 つのワーカー・プロセスを使用することがわかった場合は、逐次処理に対して設定した **number of locks** の値を 1/3 ずつ増やしてみてください。

使用できるロックが足りなくなると、Adaptive Server は、サーバ・レベルのエラー・メッセージを表示します。ユーザがロック・エラーをレポートする場合は、**number of locks** の値を大きくする必要があるかもしれません。ただし、ロックはメモリを消費することを念頭においてください。『システム管理ガイド 第 2 巻』の「第 3 章 メモリの設定」を参照してください。

注意 データロー・ロックを使用する場合は、**number of locks** の値を変更する必要があります。詳細については、『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』を参照してください。

number of mailboxes

要約	
デフォルト値	30
値の範囲	30 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、SQL Server 管理

number of mailboxes パラメータは、Adaptive Server が割り付けるメールボックス構造体の数を指定します。メールボックスはメッセージとともに使用され、カーネル・サービス・プロセス間の通信と同期用に内部的に使用されます。ユーザ・プロセスはメールボックスを使用しません。Sybase 製品の保守契約を結んでいるサポート・センタから指示がないかぎり、このパラメータは変更しないでください。

number of messages

要約	
デフォルト値	64
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、SQL Server 管理

number of messages パラメータは、Adaptive Server が割り付けるメッセージ構造体の数を指定します。メッセージはメールボックスとともに使用され、カーネル・サービス・プロセス間の通信と同期用に内部的に使用されます。また、メッセージは、並列処理におけるプロセスのファミリー間の調整にも使用されます。Sybase 製品の保守契約を結んでいるサポート・センタから指示がないかぎり、このパラメータは変更しないでください。

number of network tasks

要約	
デフォルト値	1
値の範囲	UNIX : 1 ~ 64 Windows : 1
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	ディスク I/O、プロセッサ

number of network tasks はポーリングおよびネットワーク I/O の完了のための専用タスクの数を制御します。

number of oam trips

要約	
デフォルト値	0
値の範囲	0 ~ 65535
ステータス	動的
表示レベル	包括
必要な役割	システム管理者

number of oam trips は、古くなった **object allocation map** (OAM) ページが、何回 MRU/LRU チェーンを周回してから追い出されるかを指定します。**number of oam trips** の値を増やすに従って、古くなった OAM ページがキャッシュ内にとどまる時間が長くなります。

テーブルやインデックスごとに OAM ページがあり、それぞれの OAM ページにはテーブルやインデックスに割り当てられたページに関する情報が格納され、インデックスやテーブルに新しいページが必要になった場合にチェックされます (「[page utilization percent](#)」(218 ページ) を参照)。1 つの OAM ページは、2,000 ~ 63,750 のデータまたはインデックス・ページ用のアロケーション・マップを保持できます。

OAM ページは、オブジェクトが領域を使用するそれぞれのアロケーション・ユニットのアロケーション・ページを指します。アロケーション・ページには、アロケーション・ユニット内のエクステントとページの使用状況についての情報が記録されています。

大量のバルク・コピー操作のように非常に大きな領域の割り付けを必要とする環境やベンチマークでは、OAM ページがキャッシュ内に存在する時間が長いほどパフォーマンスが向上します。**number of oam trips** を大きな値に設定するほど、OAM ページをキャッシュ内に長くとどめておくことができます。

注意 キャッシュが比較的小さく、数多くのオブジェクトによって使用される場合は、**number of oam trips** をあまり大きな値に設定しないでください。設定値が大きすぎると、追い出されない OAM ページでキャッシュがいっぱいになり、ユーザ・スレッドのタイムアウトが発生する可能性があります。

number of oam trips の値を 0 以外に変更する前に、すべてのインデックス、OAM、データ・ページを保管できるだけの十分なキャッシュがアプリケーションにあることを確認してください。**number of oam trips** の値を変更する前に、Sybase 製品の保守契約を結んでいるサポート・センタまで連絡してください。

number of open databases

要約	
デフォルト値	12
値の範囲	6 ~ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、メタデータ・キャッシュ、SQL Server 管理

number of open databases は、Adaptive Server で同時にオープンできるデータベースの最大数を指定します。

値を計算するときは、システム・データベース **master**、**model**、**sybssystemprocs**、**tempdb** も加えてください。監査機能をインストールした場合は、**sybsecurity** データベースも加えます。また、サンプル・データベース **pubs2** と **pubs3**、構文データベース **sybsyntax**、**dbcc** データベース **dbccdb** をインストールした場合は、これらも加えてください。

別のサーバから大きなデータベースをロードするなどの大幅な変更を行おうとしている場合は、**sp_helpconfig** を使用してメタデータ・キャッシュ・サイズの概算値を計算します。**sp_helpconfig** は、特定の数のメタデータ記述子に必要なメモリ量と、特定のメモリ量で対応可能な記述子の数を表示します。データベース・メタデータ記述子は、データベースの使用中の状態、またはキャッシュされている状態を表します。

❖ **number of open databases の最適化**

Adaptive Server でオープン可能なデータベースの数を越えたことを知らせるメッセージが表示された場合は、この値を調整してください。

- 1 **sp_countmetadata** を使用して、データベース・メタデータ記述子の総数を調べます。

```
sp_countmetadata "open databases"
```

sp_countmetadata の実行に最適なのは、サーバ上にアクティビティがほとんどないときです。ピーク時に **sp_countmetadata** を実行すると、他のプロセスとの競合が発生します。

Adaptive Server から次の情報がレポートされたとします。

```
1719 K バイトのメモリを必要とする、50 データベースがあります。現在、
'open databases' 設定パラメータは 500 に設定されています。
```

- 2 **number of open databases** の値を 50 に設定します。

```
sp_configure "number of open databases", 50
```

この新しい設定値は、最終的な値ではありません。データベースの総数ではなく、アクティブなデータベースのメタデータ・キャッシュ記述子の数を基準に設定してください。

- 3 ピーク時にアクティブなメタデータ記述子の数を調べます。

```
sp_monitorconfig "open databases"
```

```
Usage information at date and time: Apr 22 2002 2:49PM.
```

Name	Num_free	Num_active	Pct_act	Max_Used
Reuse_cnt Instance_Name				
-----	-----	-----	-----	-----
number of open	50	20	.00	26
0	NULL			

この例では、20 のメタデータ・データベース記述子がアクティブになっています。サーバの最後の起動後にアクティブになった記述子の最大数は 26 です。

詳細については、『リファレンス・マニュアル：プロシージャ』の「**sp_monitorconfig**」を参照してください。

- 4 **number of open databases** を、26 に 10% (およそ 3) の追加領域を加えた値、つまり 29 に設定します。

```
sp_configure "number of open databases", 29
```

たとえば、データベースの追加や削除の操作を行っているときなどに、サーバ上に多くのアクティビティがある場合は、**sp_monitorconfig** を定期的に行います。アクティブな記述子の数の変化に応じて、キャッシュ・サイズを再設定してください。

number of open indexes

要約	
デフォルト値	500
値の範囲	100 ~ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、メタデータ・キャッシュ

number of open indexes パラメータは、Adaptive Server で同時に使用できるインデックスの最大数を設定します。

大量のインデックスを持つデータベースを別のサーバからロードするなどの大幅な変更を行おうとしている場合に、**sp_helpconfig** を使用すると、メタデータ・キャッシュ・サイズの概算値を計算できます。**sp_helpconfig** は、特定の数のメタデータ記述子に必要なメモリ量と、特定のメモリ量で対応可能な記述子の数を表示します。インデックス・メタデータ記述子は、インデックスの使用中の状態、またはキャッシュされている状態を表します。

❖ **number of open indexes の最適化**

number of open indexes のデフォルト値では不十分な場合、Adaptive Server はアクティブなインデックス記述子の再使用を試みた後でメッセージを表示します。このメッセージが表示された場合は、値を調整する必要があります。

- 1 システム・プロシージャ **sp_countmetadata** を使用して、インデックス・メタデータ記述子の総数を調べます。

```
sp_countmetadata "open indexes"
```

sp_countmetadata の実行に最適なのは、サーバ上のアクティビティがほとんどないときです。ピーク時に **sp_countmetadata** を実行すると、他のプロセスとの競合が発生します。

Adaptive Server から次の情報がレポートされたとします。

```
There are 698 user indexes in all database(s),
requiring 286.289 Kbytes of memory. The 'open
indexes' configuration parameter is currently set to
500.
```

- 2 **number of open indexes** パラメータを 698 に設定します。

```
sp_configure "number of open indexes", 698
```

この新しい設定は最終的な値ではありません。インデックスの総数ではなく、アクティブなインデックスのメタデータ・キャッシュ記述子の数を基準に設定してください。

- 3 ピーク時に、アクティブなインデックス・メタデータ記述子の数を調べます。

```
sp_monitorconfig "open indexes"

Usage information at date and time: Apr 22 2002  2:49PM.
Name                Num_free   Num_active  Pct_act   Max_Used
Reuse_cnt   Instance_Name
-----
number of open          182         516       73.92     590
                0                NULL
```

この例では、サーバの最後の起動後に使用されたインデックス記述子の最大数は 590 です。

『リファレンス・マニュアル：プロシージャ』の「sp_monitorconfig」を参照してください。

- 4 number of open indexes 設定パラメータを、590 に 10% の追加領域 (59) を加えた値、つまり 649 に設定します。

```
sp_configure "number of open indexes", 649
```

たとえば、テーブルの追加や削除などの操作を行っているときなどに、サーバ上に多くのアクティビティがある場合は、sp_monitorconfig を定期的に行います。アクティブな記述子の数の変化に応じて、キャッシュ・サイズを再設定してください。

number of open objects

要約	
デフォルト値	500
値の範囲	100 ~ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、メタデータ・キャッシュ、SQL Server 管理

number of open objects は、Adaptive Server で同時にオープンできるオブジェクトの最大数を指定します。

大量のオブジェクトが含まれるデータベースを別のサーバからロードするなどの大幅な変更を行おうとしている場合に、sp_helpconfig を使用するとメタデータ・キャッシュ・サイズの概算値を計算できます。sp_helpconfig は、特定の数のメタデータ記述子に必要なメモリ量と、特定のメモリ量で対応可能な記述子の数を表示します。オブジェクト・メタデータ記述子は、インデックスの使用中の状態、またはキャッシュされている状態を表します。

❖ **number of open objects の最適化**

number of open objects のデフォルト値では不十分な場合、Adaptive Server はアクティブなオブジェクト記述子の再使用を試みた後でメッセージを表示します。

- 1 システム・プロシージャ `sp_countmetadata` を使用して、オブジェクト・メタデータ記述子の総数を調べます。

```
sp_countmetadata "open objects"
```

`sp_countmetadata` の実行に最適なのは、サーバ上のアクティビティがほとんどないときです。ピーク時に `sp_countmetadata` を実行すると、他のプロセスとの競合が発生します。

Adaptive Server から次の情報がレポートされたとします。

```
There are 696 user objects in all database(s),
requiring 1166 Kbytes of memory. The 'open objects'
configuration parameter is currently set to 500.
```

- 2 オープンしているオブジェクトの数を占めるように `number of open objects` を設定します。

```
sp_configure "number of open objects", 1407
```

1407 は、ユーザ・オブジェクト数 1340 に、テンポラリ・テーブル用の 5% を加えた値です。

この新しい設定は最終的な値ではありません。理想的な数は、オブジェクトの総数ではなく、アクティブなオブジェクトのメタデータ・キャッシュ記述子の数を基準に設定してください。

- 3 ピーク時に、アクティブなメタデータ・キャッシュ記述子の数を調べます。

```
sp_monitorconfig "open objects"
Usage information at date and time: Aug 20 2007 1:32PM..
Name          Num_free  Num_active  Pct_act  Max_Used  Reuse_cnt
-----
Instance_Name
-----
number of open objects      560          847      71.40      1497        0
NULL
```

この例では、サーバの最後の起動後に使用されたオブジェクト記述子の最大数は 1497 です。

- 4 `number of open objects` を、1397 に 10% の追加領域 (140) を加えた値、つまり 1537 に設定します。

```
sp_configure "number of open objects", 1537
```

たとえば、テーブルの追加や削除などの操作を行っているときなどに、サーバ上に多くのアクティビティがある場合は、`sp_monitorconfig` を定期的に行います。アクティブな記述子の数の変化に応じて、キャッシュ・サイズを再設定してください。『リファレンス・マニュアル：プロシージャ』の「`sp_monitorconfig`」を参照してください。

number of open partitions

要約	
デフォルト値	500
値の範囲	100 ~ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、メタデータ・キャッシュ

Adaptive Server が一度にアクセスできるパーティションの数を指定します。

使用しているシステムに対する *number of open partitions* パラメータの最適化

`number of open partitions` のデフォルト値では不十分な場合、Adaptive Server はアクティブなパーティション記述子の再使用を試みた後でメッセージを表示します。このメッセージが表示された場合は、値を調整する必要があります。

- 1 システム・プロシージャ `sp_countmetadata` を使用して、開いているパーティションの総数を調べます。たとえば、次のように結果が表示されます。

```
sp_countmetadata "open partitions"
```

`sp_countmetadata` の実行に最適なのは、サーバ上のアクティビティがほとんどないときです。ピーク時に `sp_countmetadata` を実行すると、他のプロセスとの競合が発生します。

Adaptive Server から次の情報がレポートされたとします。

```
There are 42 user partitions in all database(s),
requiring 109 Kbytes of memory. The 'open
partitions' configuration parameter is currently set
to 110.
```

- 2 `sp_countmetadata` のレポートに従って、`number of open partitions` を 110 に設定します。

```
sp_configure "number of open partitions", 110
```

- 3 ピーク時に、アクティブなメタデータ・キャッシュ記述子の数を調べます。次に例を示します。

```
sp_monitorconfig "open partitions"
Usage information at date and time: Jun 30 2008 3:15PM.
```

Name	Num_free	Num_active	Pct_act	Max_Used
Reuse_cnt Instance_Name				
-----	-----	-----	-----	-----
number of open partitions	27	57	51.8	83
0	NULL			

この例では、サーバの最後の起動後に使用されたパーティションの最大数は 83 です。

- 4 `number of open partitions` パラメータを、83 に 10% の追加領域 (8) を加えた値、つまり 91 に設定します。

```
sp_configure "number of open partitions", 91
```

たとえば、テーブルの追加や削除などの操作を行っているときなどに、サーバ上に多くのアクティビティがある場合は、`sp_monitorconfig` を定期的に実行します。アクティブな記述子の数の変化に応じて、キャッシュ・サイズを再設定してください。『リファレンス・マニュアル：プロシージャ』の「`sp_monitorconfig`」を参照してください。

number of pre-allocated extents

要約	
デフォルト値	2
値の範囲	1 ~ 32
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

`number of pre-allocated extents` パラメータは、ページ・マネージャへの 1 回の周回で割り付けられるエクステント (8 ページ) の数を指定します。現在、このパラメータを使用するのは `bcp` だけで、その目的は大量のデータをコピーするときのパフォーマンスの改善です。デフォルトでは、`bcp` は一度に 2 つのエクステントを割り付け、そのたびに割り付けレコードをログに書き込みます。

`number of pre-allocated extents` を設定することは、`bcp` がより多くの領域を必要とするたびに指定された数のエクステントを割り付け、そのイベントについて 1 つのログ・レコードを書き込むことを意味します。

実際に必要とするよりも多くのページがオブジェクトに割り付けられることがあるため、`bcp` で処理するバッチが小さい場合は、`number of pre-allocated extents` の値を小さくしてください。 `bcp` で処理するバッチが大きい場合は、`number of pre-allocated extents` の値を大きくして、ページの割り付けに必要なオーバーヘッドの量とログ・レコード数を減らしてください。

number of pre-allocated extents への値 32 の使用

`number of pre-allocated extents` に値 32 を使用することは、設定にとって特別な意味があり、Adaptive Server が内部的に実行する領域の割り付けに影響を及ぼします。 `number of pre-allocated extents` が 32 に設定されると、Adaptive Server はエクステント数分の割り付け単位全体を `bcp-in` や `select into` などのユーティリティ操作のために予約します。これらの操作では、領域予約の大規模な割り付けスキームが使用されるからです。これにより、これらのユーティリティのパフォーマンスが大幅に向上します。特に、複数のノード上で同時に実行する場合です。したがって、値 32 を使用すると、クラスタの各ノードが独自の割り付け単位上で他のノードから干渉されることなく独立して作業できます。

以前のバージョンの Adaptive Server では、`number of pre-allocated extents` パラメータは、あらゆるサイズのテーブル向けに単一の割り付け呼び出しで予約されるエクステント数を指定していました。

このバージョンの Adaptive Server では、次のコマンドの場合にかぎり、ページが 240 行以上ある大規模なテーブルでは `number of pre-allocated extents` の値が無視されます。

- `alter table table_name add column_name ...`
- `alter table table_name modify column_name ...`
- `alter table table_name drop column_name ...`
- `alter table lock ...`
- `reorg rebuild`

これらのコマンドを 240 ページを超えるテーブルで実行するときには、Adaptive Server は割り付け単位全体 (エクステント数 32) を予約します。これにより、パフォーマンスが大幅に向上します。特に、複数のノードで同時に実行するときです。

`number of pre-allocated extents` の値は、240 ページ未満のテーブルに対する上記のコマンドだけでなく、あらゆるサイズのテーブルに対するすべてのコマンド (`select into`、`bcp`、`alter table partition` など) でも引き続き守られます。

number of Q engines at startup

要約	
デフォルト値	0
値の範囲	0 ~ 127
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	プロセッサ

number of Q engines at startup は、サーバ起動時にオンラインになる Q エンジンの数を指定し、MQ に必要です。max online Q engines の数値に対応するには、max online engines を増やす必要がある場合があります。

number of remote connections

要約	
デフォルト値	20
値の範囲	5 ~ 32767
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、ネットワーク通信

number of remote connections は、1 つの Adaptive Server との間で同時にオープンできる論理接続数を指定します。ESP 実行のための XP Server への同時接続には、それぞれ 1 つのリモート接続を使用します。「[第 7 章 リモート・サーバの管理](#)」を参照してください。

number of remote logins

要約	
デフォルト値	20
値の範囲	0 ~ 32767
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、ネットワーク通信

number of remote logins パラメータは、Adaptive Server からリモート・サーバへのアクティブなユーザ接続数を制御します。ESP 実行のための XP Server への同時接続には、それぞれ 1 つのリモート接続を使用します。このパラメータの値は、**number of remote connections** の値以下となるように設定してください。「[第 7 章 リモート・サーバの管理](#)」を参照してください。

number of remote sites

要約	
デフォルト値	10
値の範囲	0 ~ 32767
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、ネットワーク通信

number of remote sites は、Adaptive Server に同時にアクセスできるリモート・サイトの最大数を決定します。Adaptive Server と XP Server 間の接続にはそれぞれ 1 つのリモート・サイト接続を使用します。

内部的には、**number of remote sites** によって、一度にアクティブにできるサイト・ハンドラの数決定します。同じサイトからのサーバ・アクセスはすべて 1 つのサイト・ハンドラによって管理されます。たとえば、**number of remote sites** が 5 に設定されているときに、各サイトが 3 つずつリモート・プロシージャ・コールを実行すると、**sp_who** の出力には、15 個のプロセスに対する 5 つのサイト・ハンドラ・プロセスがあることが表示されます。「[第 7 章 リモート・サーバの管理](#)」を参照してください。

number of sort buffers

要約	
デフォルト値	500
値の範囲	0 ~ 32767
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

number of sort buffers は、ソート処理時に入力テーブルから読み込んだページを保持し、インデックスのマージを実行するために使用するバッファに割り付けられるメモリの量を指定します。**number of sort buffers** は並列ソート用のみ使用します。

Adaptive Server はソート・バッファをテーブルがバインドされるキャッシュから割り付けます。キャッシュがテーブルにバインドされていない場合、Adaptive Server はソート・バッファをデータベースがバインドされるキャッシュから割り付けます。キャッシュがテーブルにもデータベースにもバインドされていない場合、Adaptive Server はソート・バッファをデフォルトのデータ・キャッシュから割り付けます。

並列ソートは次の場合に使用されます。

- `updates statistics` の実行
- インデックスの作成

『パフォーマンス&チューニング・シリーズ：クエリ処理と抽象プラン』の「第10章 パフォーマンス改善のための統計値の使用」を参照してください。

`number of sort buffers` の値は、サーバのページ・サイズによって異なります。

このパラメータの値は、インデックスを並列に作成する場合を除いて、デフォルトの設定のままにしておくことをおすすめします。

設定値が大きすぎると、ソートを実行するために使用されるキャッシュ内のバッファ・プールに、ソート以外のプロセスがアクセスできなくなります。

ソート・バッファに大きな値を設定した場合、大きなテーブルのソートではプロシージャ・キャッシュがさらに必要になる可能性があります。この影響は、ローのサイズが小さいテーブルのほうが顕著になります。これは1ページあたりのローの数が多くなるためです。

次の式で、必要なプロシージャ・キャッシュの量(バイト単位)を概算できます。

$$(\text{ソート・バッファ数}) \times (I \text{ ページ当たりのロー数}) \times 100$$

ソート・バッファの数に対してプロシージャ・キャッシュが不足している場合、ソートは失敗し、エラー・メッセージ 701 が表示されることがあります。このメッセージが表示された場合は、ソート・バッファの設定数を小さくしてからソートを再実行してください。

1つの演算子が使用するバッファ数の上限値を設定する方法については、「[max buffers per lava operator](#)」(175 ページ)を参照してください。

number of user connections

要約	
デフォルト値	25
値の範囲	5 ~ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、ユーザ環境

`number of user connections` は、同時に Adaptive Server へ接続できるユーザ接続の最大数を設定します。これはプロセスの最大数を指すものではありません。プロセスの最大数はこのパラメータの値だけでなく、他のシステム・アクティビティにも依存します。

最大ユーザ接続数の上限

プロセス当たりのファイル記述子の最大許容数は、オペレーティング・システムによって異なります。プラットフォームの『Adaptive Server Enterprise 設定ガイド』を参照してください。

Adaptive Server 接続で使用可能なファイル記述子の数は、グローバル変数 `@@max_connections` に格納されています。使用しているシステムで使用可能なファイル記述子の最大数をレポートするには、次のようにします。

```
select @@max_connections
```

戻り値は、システムによってプロセスに許されるファイル記述子の最大数から、オーバヘッドを差し引いた数を表します。オーバヘッドは、エンジン数とともに増加します。Adaptive Server 接続で使用可能なファイル記述子の数に対するマルチプロセッシングの影響については、『システム管理ガイド 第 2 巻』の「第 5 章 マルチプロセッサ・サーバの管理」を参照してください。

さらに、次の項目の分の接続を確保する必要があります。これらの項目も、設定パラメータを使用して設定されます。

- データベース・デバイス (ミラー・デバイスを含む)
- サイト・ハンドラ
- ネットワーク・リスナ

`number of user connections + (number of devices * max online engines * 2) + number of remote sites + max number network listeners` が `@@max_connections` の値を超えないようにします。

予約済みの接続

設定された接続数のうち 1 つは、一時的な管理タスク用に予約されます。これは、データベース管理者が必ず Adaptive Server に接続できるようにするためです。予約済みの接続では、ログインできる時間は合計で 15 分であり、`sa_role` を持つユーザにのみ割り付けられます。15 分が経過すると接続は終了します。これは、複数のデータベース管理者がいるインストール環境で他の管理者が予約済みの接続を使用できるようにするためです。

Adaptive Server に接続するための最後のリソースがクライアントによって使用されているときも、この予約済みの接続が自動的に使用されます。

Adaptive Server が予約済みの接続を使用しているときにユーザが Adaptive Server にログインすると、次のメッセージが表示されます。

```
There are not enough user connections available; you are being connected
using a temporary administrative connection which will time out after '15'
minutes. Increase the value of th 'number of user connections' parameter
```

また、Adaptive Server への最後の接続がタイムアウトで切断されたときは、次のようなメッセージがエラー・ログに出力されます。

```
00:00000:00008:2003/03/14 11:25:31.36 server Process '16' has been
terminated as it exceeded the maximum login time allowed for such processes.
This process used a connection reserved for system administrators and has a
maximum login period of '15' minutes
```

max number of user connections の最適化

それぞれのユーザに許可する接続数を決定するための公式はありません。システムとユーザの要件に基づいてこの値を見積もってください。ユーザ数の多いシステムでは、時々または一時的にしか必要とされない接続は、一般に複数のユーザ間で共有できることも考慮に入れてください。ユーザ接続を必要とするプロセスは次のとおりです。

- `isql` を実行するユーザごとに 1 つの接続が必要となる。
- アプリケーション開発者は、編集セッションごとに 1 つの接続を使用する。
- アプリケーションを実行するユーザが必要とする接続数は、そのアプリケーションがどのようにプログラムされているかに依存する。Open Client プログラムを実行するユーザは、オープン DB-Library `dbprocess` または Client-Library™ `cs_connection` ごとに 1 つの接続を必要とする。

注意 Adaptive Server によって使用される接続の最大数を見積もり、システムに物理デバイスまたはユーザを追加するときに `number of user connections` を更新することをおすすめします。`sp_who` を使用して、Adaptive Server 上のアクティブなユーザ接続数を定期的に調べてください。

この他に、`stack size` や `default network packet size` などの設定パラメータもユーザ接続ごとのメモリ量に影響を与えます。

共有メモリへのユーザ接続 - EJB サーバ

Adaptive Server は、`number of user connections` の値を使用して、EJB サーバの共有メモリ接続の数を決定します。したがって、`number of user connections` が 30 の場合、Adaptive Server は、EJB サーバ用に 10 個の共有メモリ接続を確立します。共有メモリ接続はユーザ接続のサブセットではないので、ユーザ接続数から減算されません。

共有メモリのユーザ接続数を増やすには、次の手順に従います。

- 1 `number of user connections` を、必要な共有メモリ接続数の 3 倍に増やします。
- 2 Adaptive Server を再起動します。

`number of user connections` は動的な設定パラメータですが、共有メモリ用のユーザ接続数を変更するにはサーバを再起動する必要があります。『EJB Server ユーザーズ・ガイド』を参照してください。

Adaptive Server バージョン 12.5.3 ESD #2 では、EJB のためにソケットが自動的に予約されることはありません。ただし、トレース・フラグ 1642 を有効にすることで、以前のバージョンの機能に戻してソケットの 3 分の 1 を EJB 用に予約できます。EJB サーバをセットアップするには、トレース・フラグ 1642 を有効にします。Adaptive Server のこのバージョンでは、EJB サーバが設定されていなければ、エラー・ログに記録された "hbc_ninit: No sockets available for HBC", というメッセージを無視できます。

Adaptive Server バージョン 12.5.3 以降では、EJB サーバは有効になっているにもかかわらず、HBC ソケットが使用できない場合は "hbc_ninit: No sockets available for HBC" というメッセージがレポートされます。トレース・フラグ 1642 が有効ではない場合は、フラグを設定し、Adaptive Server を再起動します。EJB サーバが有効でない場合はメッセージはレポートされず、EJB サーバ用に予約されたソケットは自動的に無効になります。

number of worker processes

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、クエリ・チューニング

number of worker processes は、同時に実行されるすべての並列クエリ全体に対して Adaptive Server が使用できるワーカー・プロセスの最大数を指定します。

メモリ不足のため、指定された数のワーカー・プロセスを作成できない場合は、Adaptive Server の起動時に警告メッセージが表示されます。**memory per worker process** は、ワーカー・プロセスごとに割り付けられるメモリを制御します。

number of worker processes をワーカー・スレッド・プールの十分な数のスレッド数に設定していない場合、Adaptive Server はより少ない数のワーカー・スレッドを使用するように実行時にクエリ・プランを調整します。Adaptive Server が実行時にクエリを調整できない場合、クエリは逐次モードで再コンパイルされます。ただし、十分な数のワーカー・スレッドがない場合、**alter table** および **execute immediate** コマンドは中止されます。

o/s file descriptors

要約	
デフォルト値	0
値の範囲	サイト固有
ステータス	読み込み専用
表示レベル	包括
必要な役割	システム管理者
設定グループ	O/S リソース

`o/s file descriptors` は、オペレーティング・システムに設定されている、プロセス当たりのファイル記述子の最大数を表します。このパラメータは読み込み専用であり、Adaptive Server からの設定はできません。

多くのオペレーティング・システムでは、プロセスごとに使用できるファイル記述子の数を設定できます。使用するオペレーティング・システム用のマニュアルを参照してください。

Adaptive Server 接続で使用できるファイル記述子の数は、変数 `@@max_connections` に格納されています。この数は `o/s file descriptors` の値よりも小さくなります。「[最大ユーザ接続数の上限](#)」(209 ページ) を参照してください。

object lockwait timing

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	モニタリング

`object lockwait timing` は、Adaptive Server でオブジェクトに対するロック要求のタイミング統計を収集するかどうかを制御します。

open index hash spinlock ratio

要約	
デフォルト値	100
値の範囲	1 ~ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、メタデータ・キャッシュ

open index hash spinlock ratio パラメータは、1つの「スピンロック」によって保護されるインデックス・メタデータ記述子のハッシュ・テーブル数を設定します。このパラメータは、マルチプロセッシング・システムでのみ使用します。

テーブルに属しているすべてのインデックス記述子は、ハッシュ・テーブルを通じてアクセスできます。テーブルに対するクエリが実行されると、Adaptive Server はハッシュ・テーブルを使用し、必要なインデックス情報をその **sysindexes** ロー内で検索します。ハッシュ・テーブルは、情報をすばやく取り出すために Adaptive Server が使用する内部メカニズムです。

通常、このパラメータを変更する必要はありません。ただし、ごくまれに、ハッシュ・スピンロックからの競合が発生している場合に、このパラメータを再設定する必要があります。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

スピンロック率の設定の詳細については、『システム管理ガイド 第 2 巻』の「第 5 章 マルチプロセッサ・サーバの管理」を参照してください。

open index spinlock ratio

要約	
デフォルト値	100
値の範囲	1 ~ 214748364
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、メタデータ・キャッシュ

open index spinlock ratio は、1つの「スピンロック」によって保護されるインデックス・メタデータ記述子の数を設定します。

インデックス記述子の内容に複数のプロセスがアクセスすることがあるので、Adaptive Server はスピンロックを使用してインデックス記述子を保護します。**open index spinlock ratio** は、マルチプロセッシング・システムでのみ使用します。

このパラメータに指定する値は、スピンロック当たりのインデックス記述子の比率を定義します。

1つのスピンロックを共有するインデックス記述子が多すぎると、スピンロック競合が発生する場合があります。スピンロック競合のレポートを取得するには、`sp_sysmon` を使用します。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

`sp_sysmon` で出力されるインデックス記述子のスピンロック競合が3%を超えている場合は、`open index spinlock ratio` の値を小さくしてみてください。

『システム管理ガイド 第2巻』の「第5章 マルチプロセッサ・サーバの管理」を参照してください。

open object spinlock ratio

要約	
デフォルト値	100
値の範囲	1 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メタデータ・キャッシュ

`open object spinlock ratio` は、1つの「スピンロック」によって保護されるオブジェクト記述子の数を設定します。オブジェクト記述子の内容に複数のプロセスがアクセスすることがあるので、Adaptive Server はスピンロックを使用してオブジェクト記述子を保護します。`open object spinlock ratio` は、マルチプロセッシング・システムでのみ使用します。

このパラメータのデフォルト値は100です。つまり、サーバに設定されているオブジェクト記述子100個につき1個のスピンロックがあることとなります。サーバに設定されているエンジンが1つだけの場合は、オブジェクト記述子の数に関係なく、設定されるオブジェクト記述子のスピンロックは1つだけです。

1つのスピンロックを共有するオブジェクト記述子が多すぎる場合は、スピンロック競合が発生します。スピンロック競合のレポートを取得するには、`sp_sysmon` を使用します。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

`sp_sysmon` で出力されるオブジェクト記述子のスピンロック競合が3%を超えている場合は、`open object spinlock ratio` パラメータの値を小さくしてみてください。

『システム管理ガイド 第2巻』の「第5章 マルチプロセッサ・サーバの管理」を参照してください。

optimization goal

要約	
デフォルト値	allrows_mix
値の範囲	allrows_oltp、allrows_dss
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

最適化目標は、最も優れた最適化テクニックを使用してクエリ要求を満たす便利な方法であり、オプティマイザの時間とリソースの最適利用を保証します。Adaptive Server では、2 つの最適化目標を、サーバ・レベル、セッション・レベル、クエリ・レベルという 3 つの層で指定できます。

サーバ・レベルの最適化目標はセッション・レベルの最適化目標によって上書きされ、セッション・レベルの最適化目標はクエリ・レベルの最適化目標によって上書きされます。

次に示す最適化目標を設定することで、実際のクエリ環境に最も適した最適化方式を選択できます。

- **allrows_oltp** – OLTP クエリにとって最も有用な目標です。
- **allrows_dss** – 中程度から高度に複雑な DSS クエリにとってもっとも有用な目標です。

optimization timeout limit

要約	
デフォルト値	10
値の範囲	0 ~ 1000
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

optimization timeout limit は、Adaptive Server がクエリを最適化するために費やることができる時間を、クエリの推定実行時間に対する割合として指定します。

値 0 は、最適化のタイムアウトがないことを示します。

optimizer level

要約	
デフォルト値	ase_default
値の範囲	<ul style="list-style-type: none"> • ase_current – すべてのオプティマイザの変更を現在のリリース全体で有効にする • ase_default – バージョン 1503 ESD #1 以降すべてのオプティマイザの変更を無効にする • ase1503esd2 – バージョン 15.0.3 ESD #2 全体でオプティマイザの変更を有効にする • ase1503esd3 – バージョン 15.0.3 ESD #3 全体でオプティマイザの変更を有効にする
ステータス	動的
表示レベル	
必要な役割	システム管理者
設定グループ	クエリ・チューニング

optimizer level はクエリ・プロセッサが使用する最適化レベルを決定します。最適化レベルの詳細については、『パフォーマンス&チューニング・シリーズ: クエリ処理と抽象プラン』の「第7章 最適化の制御」を参照してください。

optimizer level の値は文字データです。そのため、数値でなければならない2番目のパラメータのプレースホルダとして 0 を使用する必要があります。

page lock promotion HWM

要約	
デフォルト値	200
値の範囲	2 ~ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ロック・マネージャ、SQL Server 管理

page lock promotion HWM (high-water mark) パラメータは、**page lock promotion LWM** (low-water mark) と **page lock promotion PCT** (percentage) とともに使用します。このパラメータは、ページがロックされるテーブルまたはインデックスの1回のスキャン・セッション中に許容されるページ・ロック数を指定します。この数に達すると、Adaptive Server はページ・ロックからテーブル・ロックへの拡大を試みます。

1回のスキャン・セッション中に取得したページ・ロックの数が **page lock promotion HWM** を超えると、Adaptive Server はテーブル・ロックを取得しようとします。**page lock promotion HWM** の値は、**number of locks** の値より大きくすることはできません。

スキャン・セッションの説明と、ページ・ロック・プロモーションの制限値の設定方法の詳細については、『パフォーマンス&チューニング・シリーズ:ロックと同時実行制御』の「第 2 章 ロックの設定とチューニング」を参照してください。

page lock promotion HWM のデフォルト値は、ほとんどのアプリケーションにとって適切な値です。テーブルのロックを回避するには、値を大きくします。たとえば、数千ものページがある全ページロック・テーブルまたはデータページロック・テーブルのうち 500 ページに対して定期的な更新が行われることがわかっている場合は、page lock promotion HWM を 500 に設定して、これらのテーブルの同時実行性を高めます。

また、ページロック・テーブルとビューのロック・プロモーションを、オブジェクト単位のレベルで設定することもできます。『リファレンス・マニュアル:プロシージャ』の「sp_setrowlockpromote」を参照してください。

sp_sysmon を使用して、page lock promotion HWM の変更がロック・プロモーションの数にどのように影響するかを確認します。sp_sysmon は、排他ページ・ロックから排他テーブル・ロックへのプロモーションの比率と、共有ページ・ロックから共有テーブル・ロックへのプロモーションの比率をレポートします。『パフォーマンス&チューニング・シリーズ:sp_sysmon による Adaptive Server の監視』を参照してください。

page lock promotion LWM

要約	
デフォルト値	200
値の範囲	2 ~ page lock promotion HWM の値
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ロック・マネージャ、SQL Server 管理

page lock promotion LWM (low-water mark) は、page lock promotion HWM (high-water mark) と page lock promotion PCT (percentage) とともに使用します。このパラメータは、ページがロックされるテーブルやインデックスの 1 回のスキャン・セッション中に許容されるページ・ロック数を指定します。この数に達すると、Adaptive Server はページ・ロックからテーブル・ロックへの拡大を試みます。

page lock promotion LWM で設定されたページ・ロック数に達するまでは、Adaptive Server はオブジェクトに対するテーブル・ロックを発行しません。page lock promotion LWM には、page lock promotion HWM 以下の値を設定してください。

page lock promotion LWM のデフォルト値は、ほとんどのアプリケーションにとって適切な値です。Adaptive Server がロックを使い果たす状況が繰り返し発生する場合は、**number of locks** を増やしてください。

詳細については、『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』を参照してください。

ページ・ロック・プロモーションは、オブジェクト単位のレベルで設定することもできます。『リファレンス・マニュアル：プロシージャ』の「**sp_setpglockpromote**」を参照してください。

page lock promotion PCT

要約	
デフォルト値	100
値の範囲	1 ~ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ロック・マネージャ、SQL Server 管理

オブジェクトに対して保持されているロック数が **page lock promotion LWM** (low-water mark) と **page lock promotion HWM** (high-water mark) の間の値である場合に、**page lock promotion PCT** パラメータで設定されたページ・ロックのパーセンテージ(テーブル・サイズに基づく)を超えると、Adaptive Server はテーブル・ロックの取得を試みます。

『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』の「第2章 ロックの設定とチューニング」を参照してください。

page lock promotion PCT のデフォルト値は、ほとんどのアプリケーションに適切な値です。

ページがロックされるオブジェクトに対するロック・プロモーションは、オブジェクト単位のレベルで設定することもできます。『リファレンス・マニュアル：プロシージャ』の「**sp_setpglockpromote**」を参照してください。

page utilization percent

要約	
デフォルト値	95
値の範囲	1 ~ 100
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ディスク I/O

page utilization percent はページを割り付けるときに使用され、Adaptive Server がテーブルのオブジェクト・アロケーション・マップ (OAM: Object Allocation Map) をスキャンして未使用のページを検索するか、または単純に新しいエクステントをテーブルに割り付けるかを制御します。OAM の詳細については、「[number of oam trips](#)」(197 ページ) を参照してください。**page utilization percent** パラメータは、大規模なテーブルを持つサーバのパフォーマンスを最適化し、新しいスペースを追加する時間を短縮します。

page utilization percent が 100 に設定されているときは、Adaptive Server は新しいエクステントを割り付ける前に、オブジェクトに割り付けられた未使用のページを見つけるためにすべての OAM ページをスキャンします。このパラメータの設定値が 100 より小さい場合は、**page utilization percent** の設定と、次の計算式で求めたテーブルの使用済みページ数 (used pages) と未使用ページ数 (unused pages) の比とを比較します。

$$100 * \text{used pages} / (\text{used pages} + \text{unused pages})$$

page utilization percent がこの比率より小さい場合は、未使用ページを検索せずに新しいエクステントを割り付けます。

たとえば、120 の OAM ページと未使用データ・ページを 1 つだけ持つ 10GB のテーブルにデータを挿入する場合は、次のようになります。

- **page utilization percent** が 100 ならば、未使用のデータ・ページを見つけるために 120 の OAM ページをすべてスキャンします。
- **page utilization percent** が 95 ならば、95 は使用済みページ数と未使用ページ数の合計に対する使用済みページ数の比率より小さいので、Adaptive Server は新しいエクステントをオブジェクトに割り付けます。

page utilization percent の値が小さいと未使用ページが多くなります。**page utilization percent** の値が大きいと、大規模なテーブルではページの割り付けに時間がかかります。これは、Adaptive Server は新しいエクステントを割り付ける前に未使用ページを見つけるために OAM のスキャンを実行するからです。これによって論理 I/O と物理 I/O が増加します。

ページの割り付けが (特に大量のデータ挿入の場合に) 遅いような場合は、**page utilization percent** の値を小さくできますが、データを挿入し終わったらリセットしてください。値を小さく設定するとサーバ上のすべてのテーブルに影響を与え、結果としてすべてのテーブルに未使用ページができることとなります。

高速バルク・コピーは **page utilization percent** の設定を無視して、データベース内に使用できるエクステントがなくなるまで、常に新しいエクステントを割り付けます。

partition groups

要約	
デフォルト値	1024
値の範囲	1 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、メタデータ・キャッシュ

partition groups パラメータは、Adaptive Server によって割り付け可能なパーティション・グループの最大数を指定します。パーティション・グループは、Adaptive Server がテーブルの各パーティションへのアクセスを制御するために使用する内部構造体です。パーティション・グループは、アップグレードまたは **load database** アップグレードで使用され、Adaptive Server 12.5.x 以前のパーティションの分割が解除されます。

このデフォルト値は、最大 1024 のオープン・パーティション・グループと最大 2147483647 のオープン・パーティションを可能にします。実際のパーティションの数は、パーティションのグループ化によりこれよりもやや少なくなることがあります。

partition spinlock ratio

要約	
デフォルト値	10
値の範囲	1 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、メタデータ・キャッシュ

partition spinlock ratio は、Adaptive Server の複数のエンジンが実行されている場合に、1 つの **spinlock** によって保護されるパーティション記述子内のローの数を設定します。

Adaptive Server では、テーブル・パーティションへのアクセスはパーティション記述子を使用して管理されます。各パーティション記述子は、プロセスがそのパーティションにアクセスするときに使用しなければならないパーティション情報 (たとえば、そのパーティションの最終ページ) を保管します。**number of open partitions** を使用して、パーティション記述子を設定します。

`partition spinlock ratio` のデフォルト値は、パーティション・キャッシュ 10 個ごとに 1 つのスピロックを設定します。`partition spinlock ratio` パラメータの値を減らしても、Adaptive Server のパフォーマンスにはほとんど影響はありません。このデフォルト設定は、ほとんどのサーバに適しています。

『システム管理ガイド 第 2 巻』の「第 5 章 マルチプロセッサ・サーバの管理」を参照してください。

pci memory size

要約	
デフォルト値	64MB
有効な値	0 ~ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ユーザ環境

`pci memory size` は、PCI (Pluggable Component Interface) メモリ・プールのサイズを設定します。PCI Bridge の下で稼働するすべての JVM プラグインとプラグ可能なコンポーネント アダプタ (PCA: Pluggable Component Adapter) は、1 つの専用 PCI メモリ・プールを共有します。`pci memory size` がデフォルト値より低い値に設定されても、Adaptive Server はデフォルト値を使用します。

このメモリ・プールは、完全に PCI Bridge と稼働中のプラグ可能コンポーネント専用です。その他すべてのメモリ・プールと同様に、Adaptive Server はこのメモリ・プールを制御します。ただし、他のメモリ・プールとは異なり、PCI のメモリ・プールは PCI Bridge の初期化時に割り付けられ、それ以降に大きくなることはありません。

per object statistics active

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	モニタリング

`per object statistic active` は、Adaptive Server でオブジェクトごとの統計を収集するかどうかを制御します。

percent database for history

要約	
デフォルト値	20
有効な値	0 ~ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

percent database for history は、js_history テーブルに予約されている sybmgmtdb で使用できる合計領域のパーセンテージを指定します。実行されるジョブが多い場合、または今後のクエリのために実行可能ジョブに関する履歴レコードを保存する必要がある場合は、percent database for history の値を大きくしてください。

percent database for output

要約	
デフォルト値	30
有効な値	0 ~ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

percent database for output は、ジョブ出力に予約されている sybmgmtdb で使用できる合計領域のパーセンテージを指定します。実行されるジョブが多いか、または大量の出力を生成するジョブがあり、その出力をクエリのために保存する必要がある場合は、デフォルト値を増やしてください。

percent history free

要約	
デフォルト値	30
有効な値	0 ~ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

`percent history free` は、空き領域にしておく必要がある `sybmgmtdb` 内の予約領域のパーセンテージを指定します。たとえば、デフォルト値が使用される場合、`sybmgmtdb` の 70% がいっぱいになると、Adaptive Server は最も古い履歴レコードを消去して、新しいレコードを保存するための領域を確保します。

percent output free

要約	
デフォルト値	50
有効な値	0 ~ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

`sybmgmtdb` 内の領域で、Job Scheduler の出力を格納するために空けておく予約領域をパーセンテージで指定します。たとえば、デフォルト値を使用する場合、`sybmgmtdb` の 50% がいっぱいになると、Adaptive Server は最も古い履歴レコードを消去して、新しいレコードを保存するための領域を確保します。

performance monitoring option

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	モニタリング

`performance monitoring option` は、BMC DBXray グラフィカル・パフォーマンス・モニタリングおよび診断ツールのライセンスを有効にします。

permission cache entries

要約	
デフォルト値	64
値の範囲	1 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、ユーザ環境

permission cache entries は、タスク当たりのキャッシュ・プロテクタ数を指定します。その際、各ユーザ接続とワーカー・プロセスのメモリ量が増やされます。

ユーザ・パーミッションに関する情報は、パーミッション・キャッシュに保持されます。Adaptive Server は、パーミッションをチェックするとき、最初にパーミッション・キャッシュを検索します。必要とするパーミッションが見つからない場合は **sysprotects** テーブルを検索します。このプロセスは、必要とする情報がパーミッション・キャッシュ内に見つければ、**sysprotects** を読み込む必要がないので、非常に短時間で検索が終わります。

ただし、Adaptive Server がパーミッション・キャッシュを検索するのはユーザ・パーミッションをチェックするときだけで、パーミッションを付与するときや取り消すときには検索しません。パーミッションの付与または取り消しが行われるときは、パーミッション・キャッシュ全体がフラッシュされます。これは、新しいパーミッションを付与したり、無効にしたりすることによって、既存のパーミッションのタイム・スタンプが古くなるからです。

Adaptive Server 上のユーザが、パーミッションのチェックを必要とする操作を頻繁に実行する場合は、**permission cache entries** の値を増やすことによってパフォーマンスがわずかに改善されることがあります。この効果は、広範な調整を保証できるほど大きなものではありません。

Adaptive Server 上のユーザがパーミッションの付与や取り消しを頻繁に実行する場合は、**permission cache entries** の設定値を大きくしないでください。**grant** コマンドと **revoke** コマンドを実行するたびにキャッシュがフラッシュされるので、パーミッション・キャッシュのための領域が無駄になります。

plan text pipe active

要約	
デフォルト値	0
値の範囲	0 ~ 1
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	モニタリング

plan text pipe active は、Adaptive Server でクエリ・プラン・テキストを収集するかどうかを決定します。**plan text pipe active** と **plan text pipe max messages** を両方とも有効にすると、Adaptive Server は各クエリのプラン・テキストを収集します。**monSysPlanText** を使用すると、すべてのユーザ・タスクのクエリ・プラン・テキストを取得できます。

plan text pipe max messages

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、モニタリング

plan text pipe max messages は、Adaptive Server がエンジンごとに格納するクエリ・プラン・テキストのメッセージ数を決定します。**monSQLText** テーブル内のメッセージ数の合計は、**sql text pipe max messages** に実行中のエンジン数を掛け合わせた値になります。

print deadlock information

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)、2 (オン、概要を出力)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ロック・マネージャ、SQL Server 管理

print deadlock information は、エラー・ログにデッドロック情報を出力します。

デッドロックが繰り返し発生する場合は、**print deadlock information** を 1 に設定すると、デッドロックの原因を調べるのに役立つエラー・ログの詳細情報が得られます。ただし、**print deadlock information** を 1 に設定すると、Adaptive Server のパフォーマンスが低下することがあります。このため、**print deadlock information** を設定するのは、デッドロックの原因を調べる場合だけにしてください。

使用しているアプリケーションでデッドロックが発生しているかどうかを判断するには、**sp_sysmon** を使用してください。デッドロックが発生している場合は、**print deadlock information** パラメータを 1 に設定し、その発生原因の詳細を調べてください。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

値を 2 に設定すると、デッドロック情報の概要をエラー・ログに出力できます (値を 1 に設定すると、詳細情報が出力されます)。次に例を示します。

```
Deadlock Id 34: Process (Familyid 0, Spid 70) was waiting for a 'exclusive page'
lock on page 10858346 of the 'equineline_job' table in database 18 but process
(Familyid 0, Spid 88) already held a 'exclusive page' lock on it.
Deadlock Id 34: Process (Familyid 0, Spid 88) was waiting for a 'exclusive page'
lock on page 11540986 of the 'equineline_job' table in database 18 but process
(Familyid 0, Spid 70) already held a 'update page' lock on it.
```

print recovery information

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	バックアップとリカバリ

print recovery information は、リカバリ中に Adaptive Server のコンソールに表示される情報の内容を指定します (Adaptive Server の起動時およびデータベース・ダンプをロードしたときに、各データベースのリカバリが実行されます)。デフォルト値では、データベース名と、リカバリが進行中であることを知らせるメッセージだけが表示されます。値が 1 の場合、リカバリ中に処理される各トランザクションについての情報が表示されます。これには、そのトランザクションがアポートされたかコミットされたかも含まれます。

procedure cache size

要約	
デフォルト値	7000
値の範囲	7000 ~ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、SQL Server 管理

プロシージャ・キャッシュのサイズを 2K ページ単位で指定します。Adaptive Server は、ストアド・プロシージャを実行している間はプロシージャ・キャッシュを使用します。キャッシュ内に既にプロシージャのコピーが存在していれば、そのプロシージャをディスクから読み込む必要はありません。Adaptive Server は、ストアド・プロシージャの作成時にクエリをコンパイルするときにも、プロシージャ・キャッシュ内の領域を使用します。

procedure cache size の最適値はアプリケーションごとに異なるので、設定を変更することによって Adaptive Serve のパフォーマンスが向上する可能性があります。たとえば、多数の異なるプロシージャやアドホック・クエリを実行する場合は、アプリケーションがプロシージャ・キャッシュを大量に使用するため、この値を増やす必要がある場合があります。

警告！ procedure cache size の値が小さすぎると、Adaptive Server のパフォーマンスが低下します。

アップグレードするときには、procedure cache size はアップグレード時点のプロシージャ・キャッシュ・サイズに設定されます。

procedure deferred compilation

要約	
デフォルト値	1 (有効)
値の範囲	0 ~ 1
ステータス	動的
表示レベル	
必要な役割	システム管理者
設定グループ	クエリ・チューニング

このパラメータが有効の場合、ストアド・プロシージャ内のローカル変数または一時テーブルを参照する文のコンパイルは延期されます。これは、これらの文の最適化の際に、推定値やマジック・ナンバーではなく実行時値を使用できるようにするためです。

process wait events

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、モニタリング

process wait events は、Adaptive Server ですべてのタスクの待機イベントごとの統計を収集するかどうかを制御します。**monProcessWaits** を使用すると、特定のタスクの待機情報を取得できます。

『Transact-SQL ユーザーズ・ガイド』の「第 17 章 ストアド・プロシージャの使用」を参照してください。

prod-consumer overlap factor

要約	
デフォルト値	20
値の範囲	
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	

prod-consumer overlap factor は、最適化に影響を与えるパラメータです。Adaptive Server では、**group by** アルゴリズムが変更されています。並列プランに **set statistics I/O** を使用することはできません。

quorum heartbeat interval

要約	
デフォルト値	5
有効な値	1 ~ 60
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	共有ディスク・クラスタ

quorum heartbeat interval は、クォーラム・ハートビート間の秒数を指定します。quorum heartbeat interval を低い値に設定すると、ハートビート・オーバーヘッドが増加しますが、失われたディスク・リンクの検出が速くなります。その結果、I/O フェンシングを設定したインスタンスまたは SAN リンクを失ったインスタンスがすぐに終了します。quorum heartbeat interval を高い値に設定すると、ハートビートのオーバーヘッドが減少しますが、失われたディスク・リンクの検出が遅くなります。

quorum heartbeat retries

要約	
デフォルト値	2
有効な値	0 ~ 32767
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	共有ディスク・クラスタ

quoted identifier enhancements

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	アプリケーション機能

quoted identifier enhancement は、以下に対する引用符付き識別子を Adaptive Server が使用できるようにします。

- テーブル
- ビュー
- カラム名
- インデックス名
- システム・プロシージャのパラメータ

『リファレンス・マニュアル：ビルディング・ブロック』を参照してください。

read committed with lock

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ロック・マネージャ

read committed with lock は、Adaptive Server がトランザクション独立性レベル 1 (コミット済みを読み込み) を使用している場合の **select** クエリ実行中に、データオンリーロック・テーブルのローまたはページに対して共有ロックを保持するかどうかを指定します。カーソルについては、読み込み専用として宣言されたカーソルにのみ、**read committed with lock** が適用されます。

トランザクション独立性レベルが 1 の場合は、全ページロック・テーブルに対して **select** クエリを実行すると、現在位置のページに対するロックが引き続き保持されます。データオンリーロック・テーブルの更新可能なカーソルも、現在のページまたはローに対するロックを保持します。『パフォーマンス & チューニング・シリーズ：基本』を参照してください。

recovery interval in minutes

要約	
デフォルト値	5
値の範囲	1 ~ 32767
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	バックアップとリカバリ

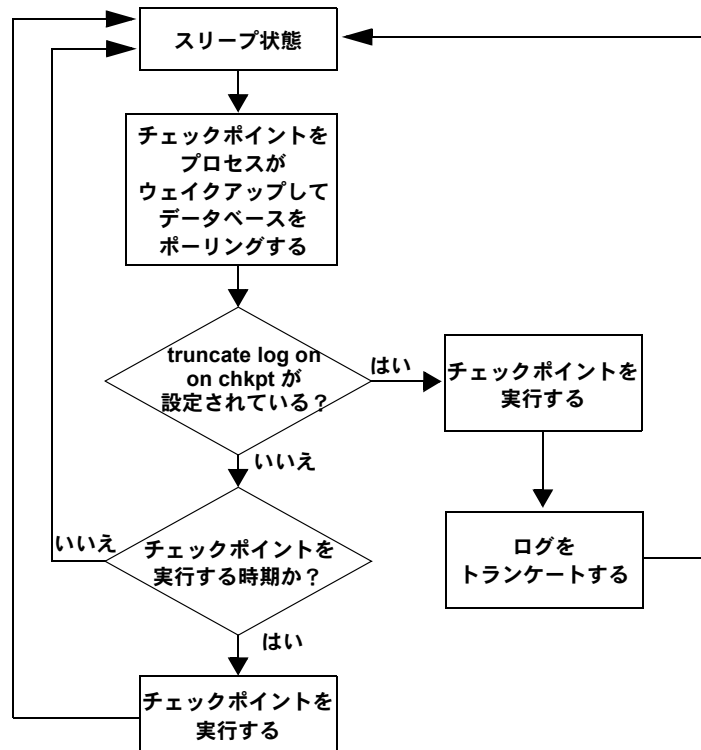
recovery interval in minutes は、システム障害が発生した場合に Adaptive Server がリカバリ・プロシージャを完了するために使用する、データベース当たりの最大時間を分単位で設定します。リカバリ・プロシージャは、チェックポイント・プロセスによって最も古いアクティブ・トランザクションとして記録されているトランザクションから始まる、トランザクションのロールバックまたはロールフォワードを行います。リカバリ・プロセスは **recovery interval in minutes** の値に応じた作業量を実行します。

Adaptive Server は、トランザクション・ログ内の 6,000 ローに対して 1 分間のリカバリ時間が必要であると見積もります。ただし、ログ・レコードのタイプによっては、リカバリにかかる時間も多少変わります。**recovery interval in minutes** を 3 に設定した場合は、最後のチェックポイント以降の **syslogs** のロー数が 18,000 を超えるまでは、変更されたページがチェックポイント・プロセスによってディスクに書き込まれることはありません。

注意 Adaptive Server の障害発生時にアクティブになっていたトランザクションのうち、実行時間が長く、ログの量が非常に少ないトランザクション (たとえば、**create index**) には、リカバリ間隔の効果はありません。このようなトランザクションの処理を元に戻すには、トランザクションを実行するのと同じぐらいの時間がかかります。あまり時間がかからないようにするには、インデックスの保守操作後に、それぞれのデータベースをダンプしてください。

Adaptive Server は、**recovery interval in minutes** の設定とそれぞれのデータベースについてのアクティビティの量を使用して、それぞれのデータベースにチェックポイントを実行する時期を決定します。Adaptive Server がデータベースのチェックポイントを実行すると、すべての「ダーティ・ページ」(キャッシュ内にある変更されたデータ・ページ) がディスクに書き込まれます。このとき、「チェックポイント・スパイク」と呼ばれる、短時間で大量の I/O が行われる状態が発生することがあります。チェックポイントの実行時には、この他に、**truncate log on chkpt** オプションが設定されているデータベースのトランザクション・ログのトランケートなどの管理タスクも実行されます。スリープしているチェックポイント・プロセスは 1 分に約 1 回の割合で「ウェイクアップ」して、**truncate log on chkpt** の設定をチェックし、リカバリ間隔をチェックしてチェックポイントが必要かどうかを調べます。図 5-4 は、このプロセスで使用されるロジックを示しています。

図 5-4: チェックポイント・プロセス



アプリケーションとその使用形態に変更がある場合は、リカバリ間隔を変更します。たとえば、Adaptive Server の更新アクティビティが増加した場合は、リカバリ間隔を短くします。リカバリ間隔を短くするとチェックポイントの頻度が上がり、チェックポイント・スパイクは小さくなりますが頻繁に発生するようになり、システムの処理速度が多少遅くなります。一方、リカバリ間隔が長すぎると、リカバリ時間が長くなりすぎる可能性があります。チェックポイントによって生じるスパイク数を減らすには、`housekeeper freewrite percent` パラメータを再設定します。「[housekeeper free write percent](#)」(143 ページ) を参照してください。パフォーマンスへの `recovery interval in minutes` の影響については、『パフォーマンス&チューニング・シリーズ：基本』の「第5章 メモリの使い方とパフォーマンス」を参照してください。

特定のリカバリ間隔がシステムにどのように影響するかを判断するには `sp_sysmon` を使用してください。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

remote server pre-read packets

要約	
デフォルト値	3
値の範囲	3 ~ 255
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、ネットワーク通信

remote server pre-read packets は、リモート・サーバとの接続中にサイト・ハンドラが「先読み」するパケットの数を決定します。

必要な接続数を少なくするために、2つのサーバ間の通信を1つのサイト・ハンドラで管理します。サイト・ハンドラは、受け取り側のプロセスがデータ・パケットを受け取る準備ができる前に、それぞれのユーザ・プロセスのためのデータ・パケットを先読みして追跡することができます。

remote server pre-read packets のデフォルト値は、ほとんどのサーバに適した値です。この値を大きくするとメモリの使用量が増え、小さくするとサーバ間のネットワーク通信が遅くなります。「第 7 章 リモート・サーバの管理」を参照してください。

restricted decrypt permission

要約	
デフォルト値	0
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	基本
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

restricted decrypt permission は、すべてのデータベースにおける制限のある decrypt パーミッションを有効または無効にします。このパラメータを設定するには、**sso_role** が必要です。

restricted decrypt permission を 0 (オフ) に設定すると、暗号化カラムに対する decrypt パーミッションの機能は 15.0.2 より前のバージョンと同じになります。

- テーブル所有者または SSO が明示的に decrypt パーミッションを付与します。ただし、decrypt パーミッションに対する **with grant option** がサポートされています。

- decrypt パーミッションは、テーブル所有者と SSO だけでなく、所有権の連鎖によってどのユーザにも暗黙的に付与されます。たとえば、ユーザ Fred が proc1 ストアド・プロシージャを所有しており、これにより、暗号化カラム fred.table1.col1 からデータが選択されるとします。Fred が proc1 の exec パーミッションを Harry に付与した場合、Harry は fred.table1.col1 に対する暗黙的な decrypt パーミッションを取得します。
- decrypt パーミッションは、alter table decrypt には必要ありません。これは、テーブル所有者が、暗号化カラムに対する暗黙的な decrypt パーミッションを取得しているためです。

restricted decrypt permission を 1 (オン) に設定した場合：

- SSO にのみ decrypt パーミッションが暗黙的に付与されます。
- SSO は、with grant option パラメータを使用して decrypt パーミッションを付与できます。これにより、SSO はシステム内のどのユーザが decrypt パーミッションを付与できるかを決定できます。たとえば、user1 が user3.user3_tab に対する decrypt permission を付与できるようにする場合、SSO は次のコマンドを発行します。

```
grant decrypt on user3.user3_tab to user1
with grant option
```

システム暗号化パスワードを使用する場合は、データのプライバシーを保護するために、decrypt パーミッションを DBO に付与しないことをおすすめします。ユーザ・パスワードを介したキーへのアクセスにより、DBO やその他のグループは、キーのパスワードを取得しない限り、データにアクセスできません。しかし、暗号化されたデータを表示できるユーザを DBO が決定するようにすると便利です。ユーザが指定したパスワードでキーとデータを保護しない場合は、SSO だけが decrypt パーミッションを付与する責任を持つようにしてください。

- テーブル所有者であっても、暗黙的な decrypt パーミッションは付与されません。つまり、暗号化カラムを含むテーブルを作成しても、明示的に付与されない限り、これらのカラムに対する decrypt パーミッションを取得しません。
- 所有権の連鎖によって、decrypt パーミッションが暗黙的に付与されることはありません。たとえば、Fred が proc1 ストアド・プロシージャを所有し、これにより、暗号化カラム fred.table1.col1 からデータが選択されるとします。Fred が proc1 の exec パーミッションを Harry に付与した場合、Harry がデータを表示するには、fred.table1.col1 に対する明示的な decrypt パーミッションも取得する必要があります。
- エイリアス・ユーザは、エイリアスが設定されているユーザのパーミッションを代用できます。同様に、いずれかのデータベース内の DBO にエイリアスが暗黙的に設定された、sa_role を持つユーザは、DBO に明示的に付与された decrypt パーミッションを継承します。

- テーブル所有者は所有するテーブルに対する暗黙的な decrypt パーミッションを持たないため、alter table decrypt 文には decrypt パーミッションが必要です。

restricted decrypt permission を 0 から 1 に変更すると、暗黙的 decrypt パーミッションを使用する、現在実行中の文が完了します。SSO が、必要なカラムに対する decrypt パーミッションをユーザに付与するまで、暗黙的 decrypt パーミッションを使用する後続の文は正常に実行されず、次のエラー・メッセージを返します。

```
Msg 10330 "DECRYPT permission denied on object object_name, database
database_name, owner owner_name."
```

restricted decrypt permission を 1 から 0 に変更すると、明示的な付与を反映するローは sysprotects システム・テーブルに保持されます。ただし、Adaptive Server は、decrypt パーミッションを暗黙的に付与できるかどうかを確認するために sysprotects をチェックしないため、これらのローは暗黙的に付与された decrypt パーミッションには影響を与えません。システムが再設定される前に明示的な decrypt パーミッションが付与されたか取り消され、暗黙的な decrypt パーミッションを現在持っているユーザにのみ、sp_helprotect は誤解を招く情報を表示します。

システムの一貫性を維持するため、ユーザに付与した明示的な decrypt パーミッションを取り消してから、restricted decrypt permission の有効または無効を切り替え、システムの一貫性を維持することをおすすめします。

decrypt パーミッションの詳細については、『暗号化カラム・ユーザーズ・ガイド』を参照してください。

row lock promotion HWM

要約	
デフォルト値	200
値の範囲	2 ~ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ロック・マネージャ、SQL Server 管理

row lock promotion HWM (high-water mark) は、row lock promotion LWM (low-water mark) と row lock promotion PCT とともに使用します。このパラメータは、テーブルまたはインデックスの 1 回のスキャン・セッション中に許容されるロー・ロックの最大数を指定します。この数に達すると、Adaptive Server はロー・ロックからテーブル・ロックへの拡大を試みます。

スキャン・セッション中に取得したロックの数が **row lock promotion HWM** を超えると、Adaptive Server はテーブル・ロックの取得を試みます。**lock promotion HWM** の値を、**number of locks** の値より大きくすることはできません。

『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』の「第2章 ロックの設定とチューニング」を参照してください。

row lock promotion HWM のデフォルト値は、ほとんどのアプリケーションに適した値です。テーブルのロックを回避するには、**row lock promotion HWM** の値を大きくします。たとえば、数千ものローがあるテーブルのうち 500 ローに対して定期的な更新が行われることがわかっている場合は、**row lock promotion HWM** を 500 前後に設定すれば、このようなテーブルの同時実行性を高めることができます。

ロー・ロック・プロモーションはオブジェクト単位のレベルでも設定できます。『リファレンス・マニュアル：プロシージャ』の「**sp_setpglockpromote**」を参照してください。

row lock promotion LWM

要約	
デフォルト値	200
値の範囲	2 ~ row lock promotion HWM の値
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ロック・マネージャ、SQL Server 管理

row lock promotion LWM (low-water mark) は、**row lock promotion HWM** (high-water mark) と **row lock promotion PCT** (percentage) とともに使用します。このパラメータは、テーブルまたはインデックスの 1 回のスキャン・セッション中に許容されるロー・ロックの数を指定します。この数に達すると、Adaptive Server はロー・ロックからテーブル・ロックへの拡大を試みます。

row lock promotion LWM で設定されたロック数に達するまでは、Adaptive Server はオブジェクトに対するテーブル・ロックの取得を試みません。**row lock promotion LWM** パラメータには、**row lock promotion HWM** 以下の値を設定してください。

row lock promotion LWM のデフォルト値は、ほとんどのアプリケーションに適した値です。Adaptive Server がロックを使い果たす状況が繰り返し発生する場合は、**number of locks** を増やしてください。

詳細については、『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』を参照してください。

ロック・プロモーションは、オブジェクト単位のレベルで設定することもできます。『リファレンス・マニュアル：プロシージャ』の「**sp_setpglockpromote**」を参照してください。

row lock promotion PCT

要約	
デフォルト値	100
値の範囲	1 ~ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ロック・マネージャ、SQL Server 管理

オブジェクトに対して保持されているロック数が row lock promotion LWM (low-water mark) と row lock promotion HWM (high-water mark) の間の値である場合に、row lock promotion PCT (percentage) パラメータで設定されたロー・ロックのパーセンテージ (テーブル内のロー数に基づく) を超えると、Adaptive Server はテーブル・ロックの取得を試みます。

row lock promotion PCT のデフォルト値は、ほとんどのアプリケーションに適した値です。

ロック・プロモーションの制限値のセットアップ方法の詳細については、『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』の「第 2 章 ロックの設定とチューニング」を参照してください。

ロー・ロック・プロモーションはオブジェクト単位のレベルでも設定できます。『リファレンス・マニュアル：プロシージャ』の「sp_sterowlockpromote」を参照してください。

rtm thread idle wait period

要約	
デフォルト値	600
値の範囲	600 ~ 4026531839
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

rtm thread idle wait period は、Adaptive Server が使用するネイティブ・スレッドが無処理のときの待ち時間の長さを秒単位で定義します。ネイティブ・スレッドに対して設定された時間に達すると、スレッドは自動的に消失します。

runnable process search count

要約	
デフォルト値	2000 (Cluster Edition ではデフォルト値は 3)
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

注意 runnable process search count は Adaptive Server をプロセス・カーネル・モードに設定した場合だけ機能し、スレッド・カーネル・モードでは機能しません。その代わりに、alter thread pool *pool_name* with idle timeout = *n* を使用します。

runnable process search count パラメータは、実行可能なタスクがあるかどうかを見つけるためにエンジンがループする回数を指定します。この数に達すると、CPU はオペレーティング・システムに解放されます。

Adaptive Server のエンジンは、タスクが完了したときや、そのエンジンでの割り当てられた時間を過ぎたときに、実行キューに実行可能なタスクがあるかどうかを調べます。場合によっては、実行キューにタスクがないこともあります。この場合、エンジンは、CPU をオペレーティング・システムに解放することも、実行するタスクを探し続けることもできます。runnable process search count を大きな値に設定すると、エンジンがより多くの回数ループするようになり、結果として長い時間 CPU を保持することになります。runnable process search count を小さな値に設定すると、エンジンは短時間で CPU を解放します。

マシンがユニプロセッサであり I/O の実行をヘルパ・スレッドに依存している場合は、runnable process search を設定してネットワーク I/O、ディスク I/O、または他のオペレーティング・システム・タスクを実行すると、パフォーマンスが良くなることがあります。バルク・コピー処理などのクライアントの処理を、ヘルパ・スレッドを使用する 1 つの CPU サーバと同じマシン上で実行している場合は、サーバとクライアントの両方を CPU にアクセスできるようにする必要があります。

注意 パフォーマンス上の問題がある場合は、runnable process search count を 3 に設定してください。

ヘルパ・スレッドを使用しないユニプロセッサのマシンで Adaptive Server を稼働させる場合や、マルチプロセッサ・マシンの場合は、デフォルト値で十分なパフォーマンスが得られます。

`runnable process search count` の値を 3 に設定すると、Cluster Edition でシステム CPU が同じマシン上で稼働している他のプロセスとさらに共有されるようになります。ただし、`runnable process search count` を 3 に設定し、しかも Adaptive Server がスタンドアロン・プロセスとして稼働している場合、サーバ応答時間が長くなる可能性があります。この場合には、`runnable process search count` を 2000 に再設定します。

`runnable process search count` パラメータが Adaptive Server による CPU サイクルの使用、エンジンからオペレーティング・システムへの CPU の解放、ブロッキング・ネットワーク・チェックにどのように影響するかを調べるには、`sp_sysmon` を使用してください。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

runnable process search count と idle timeout

`runnable process search count` と `alter thread pool ...idle timeout` は共に Adaptive Server が作業を探す方法を示します。

- `runnable process search count` は、Adaptive Server が作業を探すために行うループの回数を指定します。サーバワイドなパラメータです。
- `alter thread pool ...idle timeout` は、Adaptive Server が作業を探すために行う時間を指定するもので、個々のスレッド・プールに応じて調整されます。`idle timeout` は速度が異なるプロセッサ全体で一貫したものになります。

sampling percent

要約	
デフォルト値	0
値の範囲	0 ~ 100 パーセント
ステータス	動的
表示レベル	包括
必要な役割	システム管理者またはデータベース管理者
設定グループ	クエリ・チューニング

`sampling percent` は、サンプリング率を表す数値です。5% の場合は 5、10% の場合は 10 のように指定します。

I/O の競合とリソースを減らすには、サンプリング方式を使用して `update statistics` を実行します。これにより、メンテナンス時間が少なく、データ・セットが大きい場合の I/O と時間を削減できます。常時使用され、トランケートおよび再移植される大規模なデータ・セットまたはテーブルを更新する場合は、統計的サンプリングを行うことによって、時間と I/O サイズを削減できます。

結果が十分に正確とは限らないので、サンプリングには注意が必要です。ヒストグラム値の変化と I/O の節減のバランスをとってください。

データ・セットのサンプリングは完全に正確とは言えませんが、通常はヒストグラムも密度値も許容範囲内に収まります。

サンプリングを使用するかどうかを判断するときは、データ・セットのサイズ、作業時間の制約、生成されるヒストグラムが必要な程度に正確であることを考慮に入れてください。

サンプリングで使用するパーセンテージは要件に応じて異なります。特定のデータ・セットについての情報を最も正確に反映した結果が得られるまで、さまざまなパーセンテージをテストしてください。

統計値は、システム・テーブル `sysstabstats` と `sysstatistics` に格納されます。

secure default login

要約	
デフォルト値	0
値の範囲	0 (これに続くパラメータでデフォルト・ログイン名を指定)
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

`secure default login` パラメータは、事前に認証されているが `master.syslogins` にログインが登録されていないすべてのユーザに対するデフォルト・ログインを指定します。

セキュア・デフォルト・ログインは、次の構文を使用して設定します。

```
sp_configure "secure default login", 0, default_login_name
```

構文の説明は、次のとおりです。

- `secure default login` はパラメータの名前です。
- 0 は必須パラメータです。 `sp_configure` の 2 番目のパラメータは数値にする必要があるためです。
- `default_login_name` は、Adaptive Server のユーザとしては登録されていないが、セキュリティ・メカニズムによって既に認証されているユーザ用のデフォルト・ログイン名です。ログイン名は、`master.syslogins` に存在する有効なログインでなければなりません。

select for update

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	アプリケーション機能

select for update により、Adaptive Server が、同じトランザクション内の後続の更新、および更新可能なカーソルのためにローの排他ロックを行うことができます。これにより、同時に実行される他のタスクがこれらのローを更新したり、後続の更新をブロックしたりすることを防止します。

『Transact-SQL ユーザーズ・ガイド』の「クエリ：テーブルからのデータの選択」を参照してください。

select on syscomments.text

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

select on syscomments.text は、**syscomments** テーブルの **text** カラムに対する **select** パーミッションを制限することにより、データベース・オブジェクト・テキストの保護を有効にします。デフォルト値は、**select** パーミッションを“public”に設定します。値を 0 に設定すると、**select** パーミッションをオブジェクトの所有者とシステム管理者だけに制限します。

send doneinproc tokens

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ネットワーク通信

`send doneinproc tokens` は、Adaptive Server による `doneinproc` パケット (ストアド・プロシージャで実行される各 `select` 文の後に送信される TDS メッセージ) の送信を有効または無効にします。`send doneinproc tokens` は、`dbcc tune 'doneinproc'` およびトレース・フラグ 292 に代わるものです。

多くの場合、`send doneinproc tokens` を 1 に設定すると無難です。ただし、一部のストアド・プロシージャは CT-Lib から非同期コマンドを使用して実行され、値 0 を使用すると、一部の CT-Lib アプリケーションで状態マシン・エラーが発生することがあります。

session migration timeout

要約	
デフォルト値	600
有効な値	0 ~ 32767
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	共有ディスク・クラスタ

`session migration timeout` は、クライアントがターゲット・インスタンスに接続してマイグレーションを完了するために使用できる時間を指定します。クライアントが割り付けられた時間内にターゲット・インスタンスにマイグレートしない場合、Adaptive Server は接続できなくなります。

session tempdb log cache size

要約	
デフォルト値	論理ページ・サイズ (バイト単位)
値の範囲	2147483647 までの論理ページ・サイズ
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ユーザ環境

`session tempdb log cache size` は、各セッションの `tempdb` ログ・キャッシュのサイズを指定します。Adaptive Server は、`tempdb` オブジェクトを含むセッション用のユーザ・トランザクション・ログ・レコードのバッファとして、`tempdb` ログ・キャッシュを使用します。`session tempdb log cache` のサイズは Adaptive Server のページ・サイズによって決定されます。`session tempdb log cache size` に合理的なサイズを設定しておくこと、`tempdb` ユーザ・ログ・キャッシュをフラッシュする必要のあるセッションが原因で生じるコンテキスト切り替えを防止するのに役立ちます。

Adaptive Server にはユーザ接続ごとに 2 つのログ・キャッシュが含まれています。それらはセッション `tempdb` ログ・キャッシュとユーザ・ログ・キャッシュで、ユーザ・ログ・キャッシュは `user log cache size` パラメータで決定されます。

shared memory starting address

要約	
デフォルト値	0
値の範囲	プラットフォーム固有
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	物理メモリ

`shared memory starting address` は、Adaptive Server の共有メモリ領域の開始仮想アドレスを決定します。

`shared memory starting address` を設定し直す必要はほとんどありません。設定を検討する場合は、Sybase の保守契約を結んでいるサポート・センタに事前にご相談ください。

size of auto identity column

要約	
デフォルト値	10
値の範囲	1 ~ 38
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

`size of auto identity column` パラメータは、`sp_dboption auto identity` オプションと `unique auto_identity index` オプションで自動的に作成される IDENTITY カラムの精度を設定します。

IDENTITY カラムに挿入できる最大値は、 $10^{\text{precision} - 1}$ です。IDENTITY カラムが最大値に達すると、後続の `insert` 文はすべてエラーとなり、現在のトランザクションはアボートします。

IDENTITY カラムの最大値に達した場合に最大値を大きくするには、`alter table` コマンドを使用して修正オペレーションを実行します。実行例については、『Transact-SQL ユーザーズ・ガイド』を参照してください。

また、`create table` コマンドを使用して元のテーブルと同じテーブルを作成し、このときに `IDENTITY` カラムの精度に大きい値を指定する方法もあります。新しいテーブルを作成したら、`insert` コマンドか `bcp` を使用して古いテーブルから新しいテーブルにデータをコピーしてください。

size of global fixed heap

要約	
デフォルト値	150 ページ (32 ビット・バージョン) 300 ページ (64 ビット・バージョン)
最小値	10 ページ (32 ビット・バージョン) 20 ページ (64 ビット・バージョン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	Java サービス、メモリ使用

`size of global fixed heap` は、内部データ構造などに使用するメモリ領域を指定します。

`size of the global fixed heap` を変更する場合には、論理メモリの合計サイズも同じ量だけ変更してください。

size of process object heap

要約	
デフォルト値	1500 ページ (32 ビット・バージョン) 3000 ページ (64 ビット・バージョン)
最小値	45 ページ (32 ビット・バージョン) 90 ページ (64 ビット・バージョン)
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	Java サービス、メモリ使用

`size of process object fixed heap` は、Java VM を使用するすべてのプロセスで使用できるメモリ領域の合計サイズを指定します。

`size of process object fixed heap` を変更する場合は、`total logical memory` も同じ量だけ変更してください。

size of shared class heap

要約	
デフォルト値	1536 ページ (32 ビット・バージョン) 3072 ページ (64 ビット・バージョン)
最小値	650 ページ (32 ビット・バージョン) 1300 ページ (64 ビット・バージョン)
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	Java サービス、メモリ使用

size of shared class heap は、Java VM に呼び出されるすべての Java クラスの共有メモリ領域を指定します。Adaptive Server は、ユーザ定義の Java クラスとシステムで提供される Java クラス用の共有クラス・ヒープをサーバ全体で管理します。

size of shared class heap を変更する場合は、**total logical memory** も同じ量だけ変更してください。

size of unilib cache

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、Unicode

size of unilib cache には、最小オーバーヘッド・サイズとは別に使用されるメモリの量 (バイト単位) を 1K 単位に切り上げた値を指定します。最大の Unilib 変換テーブルと最大の Unilib ソート・テーブル全体を一度にロードするのに十分な大きさになるようにします。アジア言語を使用する環境では、Unicode ベースの変換を利用する文字セットを 1 つ追加するたびに、**size of unilib cache** を 100K 増やします。

sproc optimize timeout limit

要約	
デフォルト値	40
値の範囲	0 ~ 4000
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

sproc optimize timeout limit は、Adaptive Server が保存されたプロシージャを最適化するために費やすことができる時間を、予想実行時間に対する割合として指定します。

SQL batch capture

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	モニタリング

SQL batch capture は、Adaptive Server で SQL テキストを収集するかどうかを制御します。SQL batch capture と max SQL text monitored の両方を有効にすると、Adaptive Server は各ユーザ・タスクのバッチごとに SQL テキストを収集します。

SQL Perfmon Integration (Windows のみ)

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

SQL Perfmon Integration パラメータは、Windows パフォーマンス・モニタによる Adaptive Server の統計のモニタリングを行うかどうかを設定します。

モニタの統合をサポートするには、Adaptive Server を Windows サービスとして登録する必要があります。この登録は、Sybase for the Windows プログラム・グループの Services Manager を使用して Adaptive Server を起動すると自動的に実行されます。またこれは Sybase のインストーラまたは **syconfig** ユーティリティを使用して Adaptive Server を作成する場合のデフォルト設定です。

sybperf の 15.7 バージョンでは、Adaptive Server パフォーマンスのモニタに役立つ一連の Adaptive Server カウンタが公開されています。

sql server clock tick length

要約	
デフォルト値	プラットフォーム固有
値の範囲	プラットフォーム固有の最小値 ~ 1000000、ただしデフォルト値の倍数
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

sql server clock tick length は、サーバのクロック・チックの長さをマイクロ秒単位で指定します。デフォルト値と最小値は、どちらもプラットフォーム固有のもので、値は n の整数倍に切り上げられます (n はプラットフォーム固有のクロック・チックのデフォルト値)。 **sql server clock tick length** の現在の値は、**sp_helpconfig** または **sp_configure** を使用して確認できます。

混合使用のアプリケーションにおいて、CPU バウンド・タスクがある場合は、**sql server clock tick length** の値を小さくして、次の目的を果たします。

- I/O バウンド・タスクを実行しやすくする — これには 20,000 が適切な値です。クロック・チックの長さを短くすることは、CPU バウンド・タスクがエンジンに割り当てられた時間を超える頻度が増え、これによって他のタスクが CPU を多く使えることを意味します。
- 応答時間をわずかに長くする — Adaptive Server はサービス・タスクをクロック・チックごとに 1 回実行します。クロック・チックの長さを短くすることは、単位時間当たりのサービス・タスクの実行回数が増えることを意味します。

`sql server clock tick length` の値を増やすと、コンテキストの切り替えと切り替えの間の実行時間が長くなり、CPU バウンド・タスクに有利になります。CPU バウンドの性質を持つアプリケーションにとっては、最大値の 1,000,000 が妥当な値です。ただし、I/O バウンド・タスクにとっては結果的に不利になります。これは `cpu grace time` ([「cpu grace time」\(98 ページ\)](#) を参照) と `time slice` ([「time slice」\(262 ページ\)](#) を参照) を調整することである程度は緩和できます。

注意 `sql server clock tick length` の値を変更すると、Adaptive Server のパフォーマンスに大きな影響を与える可能性があります。この値の設定を変更する場合は、Sybase 製品の保守契約を結んでいるサポート・センタに事前にご相談ください。

`sql text pipe active`

要約	
デフォルト値	0
値の範囲	0 ~ 1
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	モニタリング

`sql text pipe active` は、Adaptive Server で SQL テキストを収集するかどうかを制御します。このオプションを有効にし、`sql text pipe max messages` を設定すると、Adaptive Server は各クエリの SQL テキストを収集します。`monSysSQLText` を使用すると、すべてのユーザ・タスクの SQL テキストを取得できます。

`sql text pipe max messages`

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、モニタリング

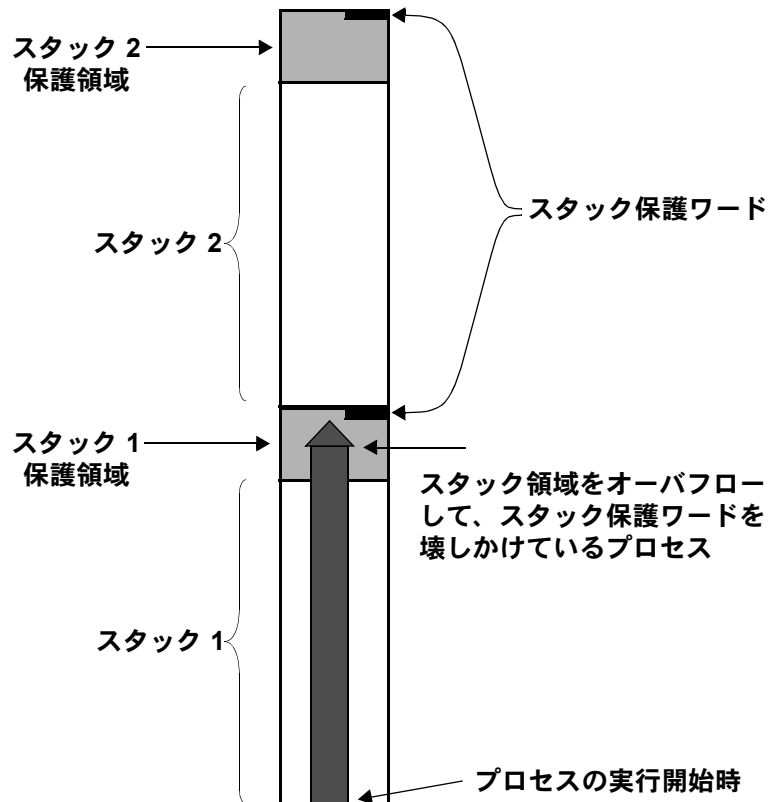
`sql text pipe max messages` は、Adaptive Server がエンジンごとに格納する SQL テキストのメッセージ数を指定します。`monSQLText` テーブル内のメッセージ数の合計は、`sql text pipe max messages` に実行中のエンジン数を掛け合わせた値になります。

stack guard size

要約	
デフォルト値	4096
値の範囲	0 ~ 2147483647
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、ユーザ環境

stack guard size は、スタック保護領域のサイズをバイト単位で設定します。「スタック保護領域」とは、各スタックの終わりにある、サイズの設定が可能なオーバーフロー・スタックです。Adaptive Server は、起動時にユーザ接続とワーカー・プロセスのそれぞれに 1 つのスタックを割り付けます。これらのスタックは、それぞれのスタックの終わりに保護領域を伴って同じメモリ領域に連続して配置されます。各スタック保護領域の終わりに、既知のパターンを持つ 4 バイトの構造体である [図 5-5](#) 「保護ワード」があります。

図 5-5: スタック保護ワードを壊しかけているプロセス



Adaptive Server は、ユーザ接続のスタック・ポインタが、そのユーザ接続に関連付けられているスタックのスタック保護領域に侵入したかどうかを定期的にチェックします。侵入している場合は、トランザクションをアボートし、そのトランザクションを生成したアプリケーションに制御を戻して、次のエラー・メッセージ 3626 を生成します。

```
The transaction was aborted because it used too much
stack space. Either use sp_configure to increase the
stack size, or break the query into smaller pieces.
spid: %d, suid: %d, hostname: %.*s, application name:
%.*s
```

また、Adaptive Server は定期的に保護ワードのパターンをチェックして、変更されていないかどうか調べます。変更されている場合は、プロセスがスタック境界をオーバーフローしたことを示します。このとき、Adaptive Server はエラー・ログに次のメッセージを出力し、停止します。

```
kernel: *** Stack overflow detected: limit: 0x%lx sp: 0x%lx
kernel: *** Stack Guardword corrupted
kernel: *** Stack corrupted, server aborting
```

最初のメッセージの“limit”はスタック保護領域の終わりのアドレスで、“sp”はスタック・ポインタの現在の値です。

さらに、Adaptive Server は定期的にスタック・ポインタをチェックして、ポインタのプロセスのスタックとスタック保護領域のどちらからも完全に外に出ているかどうかを調べます。外にある場合、Adaptive Server は保護ワードが壊されていないなくても停止します。このとき、Adaptive Server はエラー・ログに次のメッセージを出力します。

```
kernel: *** Stack overflow detected: limit: 0x%lx sp: 0x%lx
kernel: *** Stack corrupted, server aborting
```

stack guard size のデフォルト値は、ほとんどすべてのアプリケーションに適した値です。しかし、スタック保護ワードの破壊またはスタック・オーバーフローのいずれかによってサーバが停止した場合は、**stack guard size** を 2K 分増やしてください。設定されたユーザ接続とワーカー・プロセスのそれぞれにスタック保護領域が割り当てられるので、**stack guard size** を増やすと、その値に設定済みのユーザ接続とワーカー・プロセスの数を掛けた分のメモリを使用することになります。

スタックのオーバーフローの問題を避けるには、**stack guard size** を増やすのではなく、**stack size** を増やすことを検討してください (「[stack size](#)」(251 ページ)を参照してください)。スタック保護領域はオーバーフローに備えた領域であって、通常のスタックの拡張のためのものではありません。

Adaptive Server は、**stack size** パラメータと **stack guard size** パラメータの値を加算して、各タスクのスタック領域を割り付けます。**stack guard size** は、2K の倍数となるように設定してください。2K の倍数でない値を指定した場合は、2K の倍数となるように **sp_configure** 検証ルーチンによって切り上げられます。

stack size

要約	
デフォルト値	プラットフォーム固有
値の範囲	プラットフォーム固有の最小値 ~ 2147483647
ステータス	静的
表示レベル	基本
必要な役割	システム管理者
設定グループ	ユーザ環境

`stack size` パラメータは、Adaptive Server 上の各ユーザ・プロセスが使用する実行スタックのサイズをバイト単位で指定します。使用するプラットフォームの `stack size` パラメータの値を確認するには、`sp_helpconfig` または `sp_configure` を使用してください。`stack size` は、2K の倍数となるように設定してください。2K の倍数でない値を指定した場合は、2K の倍数となるように `sp_configure` 検証ルーチンによって切り上げられます。

「実行スタック」とは、ユーザ・プロセスがプロセス・コンテキストを記録し、ローカル・データを保管するための Adaptive Server メモリの領域です。

クエリによっては、スタック・オーバーフローの原因になるものがあります。たとえば、クエリの `where` 句が極端に長い場合や、`select` リストが長い場合、ストアド・プロシージャのネストが深い場合、`holdlock` で複数の `select` と `update` を実行する場合などがこれに当たります。スタック・オーバーフローが発生すると、Adaptive Server はエラー・メッセージを出力してトランザクションをロールバックします。具体的なエラー・メッセージについては、「[stack guard size](#)」(249 ページ) および『[トラブルシューティング&エラー・メッセージ・ガイド](#)』を参照してください。

スタック・オーバーフローに対処するための 2 つのオプションは、大きなクエリを小さなクエリに分割するか、`stack size` を増やすことです。`stack size` を変更すると、設定されているユーザ接続とワーカー・プロセスそれぞれに必要なメモリ量にも影響します。「[total logical memory](#)」(263 ページ) を参照してください。

クエリのサイズが実行スタックのサイズを超えるような場合は、一連の小さなクエリに書き換えます。これは、そのようなクエリの数が少ないか、たまにしか実行しないものである場合に特に当てはまります。

クエリを実際に実行せずにクエリが必要とするスタック領域の量を調べる方法はありません。各ユーザ接続とワーカー・プロセスのスタック領域は、起動時に事前に割り付けられます。

したがって、`stack size` の適切な値を決定するには、実験が必要です。`stack size` のデフォルト値を使用して、最も大きく複雑なクエリをテストしてください。エラー・メッセージを生成することなく実行できれば、おそらくデフォルト値で十分です。エラー・メッセージが生成される場合は、`stack size` を少し (2K) だけ増やしてみます。クエリを再実行し、追加した量で十分かどうかを調べます。十分でない場合は、エラー・メッセージを生成しないで実行できるようになるまで `stack size` を増やし続けます。

CIS を使用する場合、またはデータベースで Java を有効にし、JDBC を呼び出すメソッドを使用する場合は、このデフォルト値を 1.5 倍に大きくすることをおすすめします。JDBC または CIS を使用しない場合は、通常は標準のデフォルト値で十分です。

start mail session (Windows のみ)

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	拡張ストアド・プロシージャ

start mail session は、Adaptive Server 起動時に Adaptive Server のメール・セッションを自動的に開始するかどうかを指定します。

1 を指定すると、次回の Adaptive Server の起動時にメール・セッションを開始するように Adaptive Server が設定されます。0 を指定すると、次回の Adaptive Server の再起動時にメール・セッションを開始しないように Adaptive Server が設定されます。

start mail session が 0 の場合に Adaptive Server のメール・セッションを明示的に開始するには、**xp_startmail** システム ESP を使用します。

start mail session を設定する前に、Windows システムに Adaptive Server 用のメールボックスとメール・プロファイルを作成する必要があります。次に、Sybmail 用の Adaptive Server アカウントを作成します。『設定ガイド Windows 版』を参照してください。

start xp server during boot

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	
必要な役割	
設定グループ	拡張ストアド・プロシージャ

start xp server during boot は、Adaptive Server の起動時に XP Server を起動するかどうかを指定します。

1 に設定されると、Adaptive Server の起動時に XP Server も起動します。**start xp server during boot** が 0 に設定されると、**xp_cmdshell** が実行されるまでは XP Server は起動しません。

startup delay

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	クエリ・チューニング

startup delay は、サーバの起動時のどの段階で RepAgent を起動するかを制御します。デフォルトでは、RepAgent は Adaptive Server と同時に起動します。Adaptive Server は、待機時間を示すメッセージをエラー・ログに書き込みます。

statement cache size

要約	
デフォルト値	0
有効な値	キャッシュのサイズ (2K ページ単位)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、SQL Server 管理

statement cache size を使用すると、プロシージャ・キャッシュ・メモリのサーバ割り付けが増え、プロシージャ・キャッシュ・プールのうち文のキャッシュに使用されるメモリ量が制限されます。

注意 ステートメント・キャッシュを有効にする場合は、**set chained on/off** をバッチ内に設定してください。

キャッシュされた文はライトウェイト・ストアド・プロシージャに変換されるため、文のキャッシュではオープンしているオブジェクト記述子がさらに必要になります。

statement pipe active

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	モニタリング

statement pipe active は、Adaptive Server で文レベルの統計を収集するかどうかを制御します。**statement pipe active** と **statement pipe max messages** の両方を有効にすると、Adaptive Server は各クエリの文の統計を収集します。**monSysStatement** を使用すると、実行されたすべての文に関する統計を取得できます。

statement pipe max messages

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、モニタリング

statement pipe max messages は、Adaptive Server がエンジンごとに格納する文の統計のメッセージ数を決定します。**monSQLText** テーブル内のメッセージ数の合計は、**sql text pipe max messages** に実行中のエンジン数を掛け合わせた値になります。

statement statistics active

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	モニタリング

`statement statistic active` は、Adaptive Server でモニタリング・テーブルの文レベルの統計を収集するかどうかを制御します。`monProcessStatement` を使用すると、特定のタスクの文の統計を取得できます。

streamlined dynamic SQL

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	アプリケーション機能

`quoted identifier enhancement` はステートメント・キャッシュが動的 SQL 文を保存するようにします。

『パフォーマンス&チューニング・シリーズ：基本』の「メモリの使い方とパフォーマンス」を参照してください。

strict dtm enforcement

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	10
必要な役割	システム管理者
設定グループ	DTM 管理

`strict dtm enforcement` は、Adaptive Server トランザクション・コーディネーション・サービスが分散トランザクションの ACID プロパティを厳密に適用するかどうかを決定します。

Adaptive Server がトランザクションを送信し、コーディネートする相手が、トランザクション・コーディネーションをサポートする他の Adaptive Server だけである環境では、`strict dtm enforcement` をオンに設定します。トランザクション・コーディネーション・サービスをサポートしないサーバにあるデータをトランザクションが更新しようとする、Adaptive Server はそのトランザクションをアボートします。

異機種間環境では、トランザクション・コーディネーションをサポートしないサーバを使用することがあります。これには、Adaptive Server の以前のバージョンや CIS を使用して設定された Sybase 以外のデータベース・ストアも含まれます。このような状況では、**strict dtm enforcement** をオフに設定して、Adaptive Server がトランザクションを以前のバージョンの Adaptive Server や他のデータ・ストアに送信できるようにします。ただし、このように設定しても、これらのサーバのリモート作業がオリジナル・トランザクションとともにロールバックまたはコミットされることが保証されるわけではありません。

js max task message の抑制

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	10
必要な役割	システム管理者
設定グループ	アプリケーション機能

js max task message は Adaptive Server がジョブ・スケジューラの js maxtask エラー・メッセージをエラー・ログに書き込むのを防止します。

suspend audit when device full

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

suspend audit when device full は、監査デバイスの空きがまったくなくなったときの Adaptive Server の動作を決定します。

注意 複数の監査テーブルがそれぞれマスタ・デバイス以外の独立したデバイス上にあり、各監査テーブル・セグメントにスレッショルド・プロシージャが付加されていれば、監査デバイスが満杯になる状態は決して発生しません。スレッショルド・プロシージャが正常に機能していない場合だけ、「満杯」状態が発生します。

次のいずれかの値を選択します。

- 0 – 現在の監査テーブルが満杯になったときに、次の監査テーブルをトランケートし、そのテーブルを現在の監査テーブルとして使用します。**suspend audit when device full** を 0 に設定すると、監査プロセスが決して中断しないことを保証できます。ただし、古い監査レコードをアーカイブしていない場合は、それらが失われる危険性があります。
- 1 – 監査プロセスと、監査可能イベントを生成するすべてのユーザ・プロセスが中断します。通常のコマンドを再開するには、システム・セキュリティ担当者がログインして、空のテーブルを現在の監査テーブルとして設定する必要があります。この間、システム・セキュリティ担当者は、通常の監査の対象外となります。通常のコマンドであれば監査レコードが生成されるようなアクションをシステム・セキュリティ担当者が実行すると、そのイベントに関するエラー・メッセージと情報が Adaptive Server のエラー・ログに送信されます。

syb_sendmsg port number

要約	
デフォルト値	0
有効な値	0、1024 ~ 65535、またはシステム制限値
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ネットワーク通信

syb_sendmsg port number は、**sp_sendmsg** または **syb_sendmsg** でメッセージを UDP (ユーザ・データグラム・プロトコル) ポートに送信するときに Adaptive Server が使用するポート番号を指定します。

複数のエンジンが設定されている場合は、指定したポート番号以降の番号のポートが、エンジンごとに 1 つずつ使用されます。ポート番号をデフォルト値の 0 に設定すると、ポート番号は Adaptive Server によって割り当てられます。

注意 UDP ポートへのメッセージ送信は Windows ではサポートされていません。

UDP ポートにメッセージを送信できるようにするには、システム・セキュリティ担当者が **allow sendmsg** 設定パラメータを 1 に設定する必要があります。UDP のメッセージ機能を有効にするには、システム管理者が **allow sendmsg** を 1 に設定する必要があります。詳細については、「[allow sendmsg](#)」(82 ページ)を参照してください。UDP のメッセージ機能の詳細については、『リファレンス・マニュアル：プロシージャ』の「**sp_sendmsg**」を参照してください。

sysstatistics flush interval

要約	
デフォルト値	0
有効な値	0 ~ 32767
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

`sysstatistics flush interval` は、`sysstatistics` をフラッシュする間隔を分単位で指定します。

Adaptive Server は、DML 文の一部として変更されたテーブル内のローとカラムの数に関する統計を動的に保持し、`sysstatistics flush interval` の値に従ってこの統計をフラッシュします。

この統計の方が正確であるため、クエリの最適化ではこの統計が使用されます。`datachange` 関数は、前回の `update statistics` 以降に変更されたテーブル、カラム、またはパーティション・レベルのデータの量を計算し、オブジェクトの統計を更新します。

メモリ内統計は、サーバの正常なシャットダウン中に常にディスクにフラッシュされます。`sysstatistics flush interval` を使用して、ハウスキーパ・タスクがメモリ内統計を定期的にディスクにフラッシュする間隔を設定できます。このハウスキーピング・タスクを無効にするには、`sysstatistics flush interval` を 0 に設定します。

systemwide password expiration

要約	
デフォルト値	0
値の範囲	0 ~ 32767
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

`systemwide password expiration` は、変更されたパスワードの有効日数を設定します。`systemwide password expiration` を 0 に設定すると、パスワードは無期限になります。

パスワードは指定した日数が経過すると期限切れになります。たとえば、パスワードの有効期限の間隔が 30 日である新しいログオンを 2007 年 8 月 1 日の午前 10 時半に作成したとすると、2007 年 8 月 31 日の午前 10 時半にパスワードの有効期限が切れます。

アカウントのパスワードが最後に変更されたときからの期間が *number of days* を超えていると、そのアカウントのパスワードは期限切れと見なされます。

期限切れまでに残っている日数が *systemwide password expiration* の値の 25% または 7 日のいずれか大きい方よりも少なくなると、ユーザがログインするたびに期限切れまでに残っている日数を示すメッセージが表示されます。ユーザは期限が切れる前であればいつでも自分のパスワードを変更できます。

アカウントのパスワードの期限が切れてもユーザは Adaptive Server にログインできますが、*sp_password* を使用して自分のパスワードを変更するまでは、コマンドは一切実行できません。アカウントが *sp_password* 以外は実行できないモードのときに、システム・セキュリティ担当者がユーザのパスワードを変更すると、新しいパスワードが割り当てられた時点でそのアカウントは通常のモードに戻ります。

この制限が適用されるのは、パスワードの期限が切れた後に開始されるログイン・セッションだけです。パスワードの期限が切れた時点でログイン済みのユーザの場合は、次回ログインするときまでは影響がありません。

tape retention in days

要約	
デフォルト値	0
値の範囲	0 ~ 365
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	バックアップとリカバリ

tape retention in days は、データベースまたはトランザクション・ログ・ダンプに使用した後、それぞれのテープを保持する日数を指定します。このパラメータを使用すると、ダンプ・テープを誤って上書きすることを避けることができます。

たとえば、*tape retention in days* を 7 日間に設定した場合は、そのテープに最後にダンプしてから 7 日間が過ぎる前にテープを使用しようとすると、Backup Server の警告メッセージが表示されます。

ダンプ・コマンドを実行するときに *with init* オプションを使用することによって、警告を無視することができます。ただし、このようにするとテープは上書きされ、テープ上のデータはすべて消失することになります。

dump database コマンドと *dump transaction* コマンドのどちらにも、そのダンプの *tape retention in days* 値を無効にする *retaindays* オプションがあります。『システム管理ガイド 第 2 巻』の「第 12 章 ユーザ・データベースのバックアップとリストア」を参照してください。

tcp no delay

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ネットワーク通信、O/S リソース

tcp no delay は、TCP (Transmission Control Protocol) パケットのバッチ処理を制御します。デフォルト値では TCP パケットはバッチ処理されません。

TCP は、通常は物理ネットワーク・フレームをできるかぎり多くのデータで満たすために、パケットをわずかに遅延させることによって小さな論理パケットを 1 つの大きい物理パケットにまとめます。これは、ネットワークを通して送信するもののほとんどがキー入力である端末エミュレーション環境でネットワーク・スループットを改善することを目的としたものです。

しかし、小さな TDS (Tabular Data Stream) パケットを使用するアプリケーションの場合は、TCP パケットのバッチ処理を無効にすることをおすすめします。

注意 TCP パケットのバッチを無効にするということは、サイズに関係なくパケットが送られて、ネットワーク・トラフィック量が増えることを意味します。

text prefetch size

要約	
デフォルト値	16
有効な値	0 ~ 65535
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	ネットワーク通信

text prefetch size は、プリフェッチして既存のバッファ・プールに格納できる **text** データ、**unitext** データ、**image** データのページ数を制限します。Adaptive Server がプリフェッチするのは、Adaptive Server 12.x で作成されたか、または **dbcc rebuild_text** を使用してアップグレードされた **text** データ、**unitext** データ、**image** データだけです。

time slice

要約	
デフォルト値	100
値の範囲	50 ~ 1000
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

time slice は、Adaptive Server のスケジューラが1つのタスクの実行を許可する時間をミリ秒単位で設定します。**time slice** の設定値が小さすぎると、Adaptive Server がタスクの切り替えに費やす時間が過大になり、応答時間が長くなります。設定値が大きすぎると、CPU 集約タスクがエンジンを独占することがあり、この場合も応答時間が長くなります。デフォルト値を使用すると、1つのタスクが100 ミリ秒間実行されてから、CPU が別のタスクに渡されます。

『パフォーマンス&チューニング・シリーズ：基本』の「第3章 エンジンとCPUの使用方法」を参照してください。

Adaptive Server エンジンによる自発的なCPU解放への**time slice**の影響を調べるには、**sp_sysmon**を使用してください。『パフォーマンス&チューニング・シリーズ：sp_sysmonによるAdaptive Serverの監視』を参照してください。

total data cache size

要約	
デフォルト値	0
値の範囲	0 ~ 2147483647
ステータス	計算された値
表示レベル	基本
必要な役割	システム管理者
設定グループ	キャッシュ・マネージャ、メモリ使用

total data cache size は、データ、インデックス、ログ・ページ用に現在使用できるメモリの量をキロバイト単位で表します。このパラメータは、計算によって設定される値であり、ユーザは直接設定できません。

データ・キャッシュ用に使用できるメモリの量は、次のような多くの要因によって変わります。

- マシンで使用できる物理メモリの量
- 次のパラメータに設定される値
 - total logical memory
 - number of user connections
 - total procedure cache percent
 - number of open databases
 - number of open objects
 - number of open indexes
 - number of devices

この他の多数のパラメータも、その程度は大きくはありませんが、使用できるメモリの量に影響します。

Adaptive Server でのメモリ割り付け方法とデータ・キャッシュについては、「[設定パラメータ](#)」(75 ページ)を参照してください。

total logical memory

要約	
デフォルト値	該当なし
値の範囲	該当なし
ステータス	読み込み専用
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、物理メモリ

total logical memory は、Adaptive Server の現在の設定における合計論理メモリを表示します。合計論理メモリとは、Adaptive Server の現在の設定で使用されるメモリの量です。total logical memory が表すのは、確保する必要がありますが、常に使用されるとはかぎらないメモリです。特定の時点で使用されているメモリの量については、「total physical memory」を参照してください。total logical memory を使用してメモリ設定パラメータを設定することはできません。

total physical memory

要約	
デフォルト値	該当なし
値の範囲	該当なし
ステータス	読み込み専用
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用

total physical memory は読み込み専用の設定パラメータで、Adaptive Server の現在の設定における物理メモリの総量を表示します。物理メモリの総量とは、特定の時点で Adaptive Server が使用しているメモリの量をいいます。**max memory** の値が **total logical memory** より大きく、**total logical memory** の値が **total physical memory** より大きくなるように、Adaptive Server を設定してください。

transfer utility memory size

要約	
デフォルト値	4096
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

Adaptive Server は、**transfer table** コマンド用と増分転送のマークが付けられたテーブル用のメモリ・プールを維持します。このプールは、現在と過去の転送についてのステータス情報を維持するためのメモリ、および転送ファイルとの間で読み取り、書き込みを行うためのメモリを提供します。**transfer utility memory size** はこのメモリ・プールのサイズを指定します。

このプールの単位はメモリ・ページであり、2048 バイトのブロックです。デフォルト・サイズは、100 個を超える増分転送のマークが付けられたテーブルを収容するのに十分な大きさであり、すべてを同時に転送できます。

使用しているインストール環境に増分転送のマークが付けられたテーブルがなく、かつ **transfer table** コマンドが使用されない場合には、このメモリ・プールのサイズをゼロに設定してこのメモリを取り戻すこともできます。

txn to pss ratio

要約	
デフォルト値	16
有効な値	1 ~ 2147483647
ステータス	静的
表示レベル	1
必要な役割	システム管理者
設定グループ	DTM 管理、メモリ使用

Adaptive Server はトランザクションを設定可能なサーバ・リソースとして管理します。新しいトランザクションが開始するたびに、Adaptive Server は、サーバの起動時に作成されるグローバル・プールから未使用の「トランザクション記述子」を取得する必要があります。トランザクション記述子は、Adaptive Server がアクティブなトランザクションを表すために使用する内部メモリ構造です。

Adaptive Server では、以下のものを表すために、未使用のトランザクション記述子が 1 つ必要です。

- 個々のサーバ・トランザクションの外部ブロック。トランザクションの外部ブロックは、クライアントが新規に **begin transaction** コマンドを実行すると明示的に作成されます。また、クライアントがトランザクションを定義する **begin transaction** を使用せずに Transact-SQL を使用してデータを変更する場合にも、Adaptive Server によって暗黙的に作成されます。

注意 さらに **begin transaction** コマンドを使用して、ネストされたトランザクション・ブロックを作成しても、そのトランザクション・ブロック用のトランザクション記述子を取得する必要はありません。

- 「マルチデータベース・トランザクション」でアクセスされる各データベース。Adaptive Server は、新しいデータベースのデータがトランザクションによって使用または変更されるたびに、新しいトランザクション記述子を取得する必要があります。

txn to pss ratio によって、サーバで使用できるトランザクション記述子の合計数が決まります。ブート時に、この値 (比率) に PSS 構造数を乗算して、トランザクション記述子のプールが作成されます。

```
# of transaction descriptors = PSS structures * txn to pss ratio
```

デフォルト値は 16 であり、これにより 12.x より前のバージョンとの互換性が確保されます。これらのバージョンでも各ユーザ接続に 16 個のトランザクション記述子を割り付けていました。バージョン 12.x 以降では、同時に発生可能なトランザクション数は、サーバ内で使用できるトランザクション記述子の数によってのみ制限されます。

注意 1つのユーザ・トランザクションでアクセスできるデータベース数の上限は、Adaptive Server インストール環境に存在するデータベース数と同じです。たとえば、Adaptive Server のデータベース数が 25 個ならば、ユーザ・トランザクションで 25 個のデータベースを使用することができます。

使用システムに合わせた txn to pss ratio の最適化

使用のピーク時に、sp_monitorconfig を使用してトランザクション記述子の使用状況を調査します。

```
sp_monitorconfig "txn to pss ratio"

Usage information at date and time: Apr 22 2002 2:49PM.
Name                               Num_free   Num_active  Pct_act    Max_Used
Reuse_cnt   Instance_Name
-----
txn to pss ratio           784         80    10.20      523
0                          NULL
```

Num_used の値がゼロまたは非常に小さい場合は、サーバ内のトランザクション記述子が解放されるまで Adaptive Server が待機するので、トランザクションが遅延する場合があります。その場合には、txn to pss ratio の値を大きくすることを検討してください。

Max_Used の値が小さすぎるときは、他のサーバ機能が使用できるはずのメモリを、未使用のトランザクション記述子が消費している可能性があります。その場合には、txn to pss ratio の値を小さくすることを検討してください。

unified login required

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

unified login required は、Adaptive Server にログインするすべてのユーザがセキュリティ・メカニズムによって認証されることを要求します。統一化ログイン・セキュリティ・サービスを使用するには、use security services パラメータを 1 にしてください。

upgrade version

要約	
デフォルト値	15000
値の範囲	0 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	SQL Server 管理

upgrade version パラメータは、マスタ・デバイスをアップグレードしたアップグレード・ユーティリティのバージョンをレポートします。アップグレード・ユーティリティは、アップグレード中にこのパラメータをチェックし、変更します。

警告！ upgrade version を再設定しないでください。設定を変更すると、Adaptive Server の動作に重大な問題が発生することがあります。

次のように値を指定せずに upgrade version パラメータを使用すると、マスタ・デバイスのアップグレードが行われているかどうかを調べることができます。

```
sp_configure "upgrade version"
```

use security services

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

use security services は、Adaptive Server がネットワークベース・セキュリティ・サービスを使用することを指定します。このパラメータが 0 に設定されると、ネットワークベース・セキュリティ・サービスは使用できません。

user log cache size

要約	
デフォルト値	論理ページ・サイズ
値の範囲	2048 ^a ~ 2147483647 a. 最小値はサーバの論理ページ・サイズによって決まる
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、ユーザ環境

user log cache size は、それぞれのユーザのログ・キャッシュのサイズをバイト単位で指定します。このサイズは、サーバの論理ページ・サイズによって決まります。設定されているユーザ接続とワーカー・プロセスのそれぞれに1つのユーザ・ログ・キャッシュが割り当てられます。Adaptive Server は、このキャッシュをユーザ・トランザクション・ログ・レコードのバッファとして使用し、これによってトランザクション・ログの終わりでの競合を減少させます。

ユーザ・ログ・キャッシュの空きがなくなるか、トランザクションの完了などの別のイベントが発生すると、Adaptive Server はユーザ・ログ・キャッシュからデータベース・トランザクション・ログにすべてのログ・レコードを「フラッシュ」します。個々のログ・レコードをすぐにデータベースのトランザクション・ログに追加するのではなく、ユーザごとのログ・キャッシュ内にいったんまとめることによって、特に複数のエンジンが設定されている SMP システムの場合に、ログへ書き込むプロセスの競合を減少させます。

注意 データとログのセグメントが分かれていないデータベースを使用するトランザクションでは、ユーザ・ログ・キャッシュは各ログ・レコードの後でトランザクション・ログにフラッシュされます。バッファリングは行われません。データベースに専用のログ・セグメントがない場合は、**user log cache size** の値は増やさないでください。

user log cache size の設定値は、アプリケーションの1つのトランザクションで書き込まれるログ情報の最大量を超えないようにしてください。Adaptive Server はトランザクションの完了時にユーザ・ログ・キャッシュをフラッシュするので、ユーザ・ログ・キャッシュに余分に割り付けられたメモリは無駄になります。4000 バイトより大きなトランザクション・ログ・レコードを生成するトランザクションがサーバ内にないのであれば、この値を超えないように **user log cache size** を設定します。次に例を示します。

```
sp_configure "user log cache size", 4000
```

`user log cache size` を大きすぎる値に設定するとメモリを浪費します。`user log cache size` の設定値が小さすぎると、ユーザ・ログ・キャッシュが満杯になってフラッシュすることが 1 つのトランザクションについて何度も発生し、トランザクション・ログの競合が増加します。トランザクションの量が小さい場合は、トランザクション・ログの競合は大きくはありません。

このパラメータのキャッシュ動作への影響を確認するには、`sp_sysmon` を使用してください。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

user log cache spinlock ratio

要約	
デフォルト値	20
値の範囲	1 ~ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、ユーザ環境

`user log cache spinlock ratio` は、複数のエンジンを実行している Adaptive Server における、ユーザ・ログ・キャッシュの **spinlock** 当たりのユーザ・ログ・キャッシュの比率を指定します。設定されているユーザ接続ごとに 1 つのユーザ・ログ・キャッシュが存在します。

デフォルト値では、サーバに設定された 20 のユーザ接続ごとに 1 つのスピンロックがあることを意味します。

このパラメータのキャッシュ動作への影響を確認するには、`sp_sysmon` を使用してください。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

wait event timing

要約	
デフォルト値	0
値の範囲	0 ~ 1
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	メモリ使用、モニタリング

`wait event timing` は、Adaptive Server で個々の待機イベントの統計を収集するかどうかを制御します。タスクは、さまざまな理由で待機させられることがあります (たとえば、バッファの読み込みの完了を待つなど)。`monSysWaits` テーブルには、待機イベントごとの統計が含まれます。`monWaitEventInfo` には、すべての待機イベントのリストが含まれます。

workload manager cache size

要約	
デフォルト値	80
有効な値	80 ~ 2147483647
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	共有ディスク・クラスタ

`workload manager cache size` は、Workload Manager が使用できるメモリの最大量を 2K ページ単位で指定します。『Cluster ユーザーズ・ガイド』の「第 6 章 負荷の管理」を参照してください。

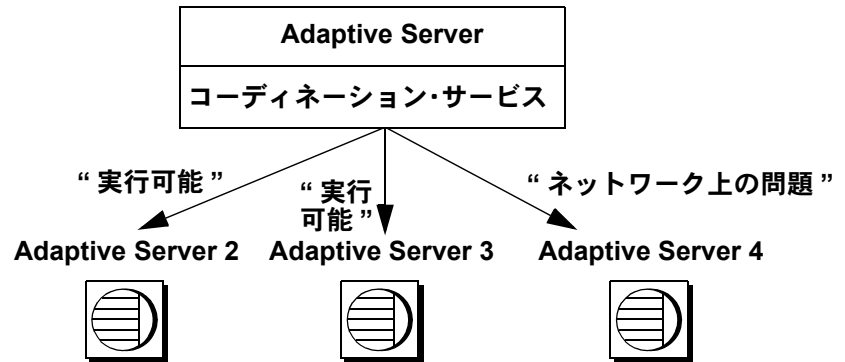
xact coordination interval

要約	
デフォルト値	60 (秒)
有効な値	1 ~ 2147483647 (秒)
ステータス	動的
表示レベル	10
必要な役割	システム管理者
設定グループ	DTM 管理

`xact coordination interval` は、リモート・サーバに送信されたトランザクション分岐の解決を試行する時間の間隔を定義します。

コーディネーティング・サーバである Adaptive Server は、分散トランザクションに参加しているリモート・サーバの処理の解決を定期的に試みます。図 5-6 に示すように、コーディネーティング・サーバは、分散トランザクションに参加している個々のリモート・サーバと順に交信します。さまざまな理由で、コーディネーティング・サービスがトランザクション分岐を解決できない場合があります。たとえば、ネットワークに問題があるためにリモート・サーバと交信できない場合には、`xact coordination level` で指定される時間が経過した後で、コーディネーティング・サーバは接続を再試行します。

図 5-6: リモート・トランザクション分岐の解決



`xact coordination interval` をデフォルト値の 60 に設定すると、Adaptive Server は 1 分おきにリモート・トランザクションの解決を試みます。この値をこれより小さくすると、分散トランザクションの処理速度が向上する可能性があります。ただし、そのトランザクション自体が 1 分以内に解決できる場合にかぎり、通常の状態では、`xact coordination interval` の値を小さくすることによってパフォーマンスが損なわれることはありません。

`xact coordination interval` の値を大きくすると、分散トランザクションの処理速度が低下し、トランザクション分岐によるリソースの保持時間が通常よりも長くなる場合があります。通常は、`xact coordination interval` の値をデフォルト値よりも大きくしないでください。

xp_cmdshell context

要約	
デフォルト値	1
有効な値	0, 1, 2
ステータス	動的
表示レベル	包括
必要な役割	システム管理者
設定グループ	拡張ストアド・プロシージャ

`xp_cmdshell context` は、`xp_cmdshell` システム ESP を使用して実行されるオペレーティング・システム・コマンドのセキュリティ・コンテキストを設定します。コンテキストの値は、コマンドを実行するアカウントを指定します。

- 0 - コマンドが XP Server のアカウントで実行される
- 1 - コマンドがユーザのアカウントで実行される
- 2 - ユーザがシステム管理者権限を持っている場合にかぎり、コマンドが XP Server のアカウントで実行される

`xp_cmdshell context` を 1 に設定すると、`xp_cmdshell` セキュリティ・コンテキストは、オペレーティング・システム・レベルのアカウントを持っているユーザに制限されます。動作はプラットフォームによって異なります。`xp_cmdshell context` が 1 に設定されている場合に `xp_cmdshell ESP` を使用するには、Adaptive Server のユーザ名と同じ名前オペレーティング・システム・ユーザ・アカウントが存在している必要があります。たとえば、Adaptive Server のユーザ名が “sa” のユーザは、“sa” というオペレーティング・システム・レベルのユーザ・アカウントを持っていなければ `xp_cmdshell` を使用することはできません。

XP Server をルートから起動すると、`xp_cmdshell` は自動的に 1 に設定され、このため安全なアクセスが自動的に有効になります。

Windows では、`xp_cmdshell context` が 1 に設定されている場合に `xp_cmdshell` が正常に実行されるのは、Adaptive Server にログインしているユーザ名が、Adaptive Server が稼働しているシステムの Windows システム管理者権限を持つ有効な Windows ユーザ名である場合だけです。

他のプラットフォームでは、`xp_cmdshell context` が 1 に設定されている場合に `xp_cmdshell` が正常に実行されるのは、Adaptive Server を起動したユーザに、オペレーティング・システム・レベルの “superuser” 権限がある場合だけです。Adaptive Server は、`xp_cmdshell` の実行要求を受け取ると、この ESP を要求したユーザ名の uid を調べ、その uid のパーミッションでオペレーティング・システムのコマンドを実行します。

`xp_cmdshell context` が 0 の場合は、`xp_cmdshell` からオペレーティング・システム・コマンドを実行するとき使用されるのは Adaptive Server が稼働しているオペレーティング・システム・アカウントのパーミッションです。これによって、ユーザ自身のオペレーティング・システム・アカウントのセキュリティ・コンテキストでは通常実行できないオペレーティング・コマンドを実行できるようになります。

トピック名	ページ
デバイスの割り付けとオブジェクトの配置	273
ディスク・リソースの管理に使用するコマンド	274
記憶領域の管理に関する考慮事項	276
インストール時のステータスおよびデフォルト設定	277
記憶領域を管理するシステム・テーブル	278

Adaptive Server の記憶領域管理のさまざまなプロパティ (データベース、テーブル、インデックスを配置する場所とそれぞれに割り付ける領域のサイズなど) のほとんどは、妥当なデフォルト値が設定されています。記憶領域の割り付けと管理は集中管理されることも多く、Adaptive Server に対するディスク・リソースの割り付けと、割り付けたディスク・リソースへのデータベース、テーブル、インデックスの物理的な配置については、通常はシステム管理者が最終的な制御権を持っています。

デバイスの割り付けとオブジェクトの配置

新しいシステムを設定するとき、システム管理者は、必要なディスク・リソースの数とサイズに直接影響するさまざまな問題を検討する必要があります。この問題とは、Adaptive Server にディスク・リソースを追加するコマンドやプロシージャをどのように実行するかということです。

表 6-1: デバイスの割り付けに関するトピック

作業	章
データベース・デバイスのデフォルト・プールの初期化と割り付け	「第 8 章 データベース・デバイスの初期化」
リカバリを目的としたデータベース・デバイスのミラーリング	『システム管理ガイド 第 2 巻』の「第 2 章 データベース・デバイスのミラーリング」

Adaptive Server の初期ディスク・リソースを割り付けた後で、システム管理者、データベース所有者、オブジェクト所有者は、データベースとデータベース・オブジェクトをどのデータベース・デバイスに配置するかを検討する必要があります。オブジェクトの配置を具体的に検討することによって、データベース・オブジェクトを、使用しているシステムのどこに常駐させるか、また、オブジェクトにデバイスを共有させるかどうかを決定します。オブジェクトを配置する作業については、表 6-2 に示す各章を含むこのマニュアル全体で説明しています。

表 6-2: オブジェクトの配置に関するトピック

作業	章
データベースを特定のデータベース・デバイスに配置する	『システム管理ガイド 第 2 巻』の「第 6 章 ユーザ・データベースの作成と管理」
テーブルとインデックスを特定のデータベース・デバイスに配置する	『システム管理ガイド 第 2 巻』の「第 6 章 ユーザ・データベースの作成と管理」

デバイスの割り付けを検討するときは、オブジェクトの配置と切り離して考えないでください。たとえば、あるテーブルを 2 つで一組の専用のデバイスに配置する場合は、最初にその 2 つのデバイスを Adaptive Server に割り付けます。この章の各項ではデバイスの割り付けとオブジェクトの配置の両方にかかわる問題に共通する概要を説明し、参照情報がある場合は参照先の章を示しています。

ディスク・リソースの管理に使用するコマンド

表 6-3 は、Adaptive Server にディスク・リソースを割り付けるためにシステム管理者が使用する主なコマンドのリストと、そのコマンドについて説明している参照先の章をまとめたものです。

表 6-3: ディスク・リソースの割り付けに使用するコマンド

コマンド	作業	参照箇所
disk init name = "dev_name" physname = "phys_name"...	特定の Adaptive Server で使用できるように物理デバイスを設定する。データベース・デバイス名 (dev_name) を割り当てる。これは、他の Adaptive Server コマンド内でこの物理デバイスを指定するときに使用される。	「第 8 章 データベース・デバイスの初期化」
sp_deviceattr logicalname, optname, optvalue	既存のデータベース・デバイス・ファイルの dsync 設定を変更する。	「第 8 章 データベース・デバイスの初期化」
sp_diskdefault "dev_name"...	dev_name をデフォルトのデータベース領域の汎用プールに追加する。	「第 8 章 データベース・デバイスの初期化」
disk resize name = "device_name", size = additional_space	データベース・デバイスのサイズを動的に拡大する。	「第 8 章 データベース・デバイスの初期化」

コマンド	作業	参照箇所
disk mirror name = "dev_name" mirror = "phys_name"...	特定の物理デバイス上のデータベース・デバイスをミラーリングする。	『システム管理ガイド 第2巻』の「第2章 データベース・デバイスのミラーリング」

表 6-4 は、オブジェクトの配置に使用するコマンドのリストです。オブジェクトの配置がパフォーマンスに及ぼす影響については、『パフォーマンス&チューニング・シリーズ：物理データベースのチューニング』の「第1章 データの物理的配置の制御」を参照してください。

表 6-4: ディスク・リソース上のオブジェクトの配置に使用するコマンド

コマンド	作業	参照箇所
create database...on dev_name または alter database...on dev_name	特定の Adaptive Server データベースで使用できるようにデータベース・デバイスを設定する。 create database の log on 句によって、このデータベースのログを特定のデータベース・デバイスに配置することを指定する。	『システム管理ガイド 第2巻』の「第6章 ユーザ・データベースの作成と管理」
create database... または alter database...	デフォルト・データベース・デバイス上の領域を割り付けるには、on dev_name 句を指定しないでこれらのコマンドを実行する。	『システム管理ガイド 第2巻』の「第6章 ユーザ・データベースの作成と管理」
sp_addsegment seg_name, dbname, devname および sp_extendsegment seg_name, dbname, devname	特定のデータベースで使用できるデバイスから、セグメント(領域の集合に名前を付けたもの)を作成する。	『システム管理ガイド 第2巻』の「第8章 セグメントの作成と使用」
create table...on seg_name または create index...on seg_name	データベース・オブジェクトを作成して、データベースに割り当てられたディスク領域の特定セグメントに配置する。	『システム管理ガイド 第2巻』の「第8章 セグメントの作成と使用」
create table... または create index...	データベースに割り付けられた領域の汎用プール(デフォルト・デバイス)にテーブルやインデックスを配置するには、on seg_name を指定しないでこれらのコマンドを実行する。	『システム管理ガイド 第2巻』の「第8章 セグメントの作成と使用」

記憶領域の管理に関する考慮事項

システム管理者は、Adaptive Server データベースへの領域の物理的な割り付けに関するさまざまな事項について決定する必要があります。この場合に考慮しなければならない重要な事項は、次のとおりです。

- リカバリ – ディスク・ミラーリングと、別の物理デバイス上でのログ保管という2つの機能によって、物理的なディスク障害が起きた場合でも完全なリカバリを実現できます。
- パフォーマンス – ディスクの読み込み／書き込み速度が非常に重要であるテーブルやデータベースについては、データベース・オブジェクトを物理デバイスに適切に配置することによってパフォーマンスが向上します。ディスク・ミラーリング機能を使用すると、ディスクの書き込み速度が低下します。

リカバリ

リカバリ機能は、複数のディスク・デバイスの使用を決定する主な要因となります。データベース・デバイスをミラーリングすると、ノンストップ・リカバリが可能になります。また、別の物理デバイス上にデータベースのログを保管することによって、完全なリカバリが保証されます。

別デバイスでのログの保管

データベース・デバイスをミラーリングしていない場合に完全なリカバリができるようにするには、データベースのトランザクション・ログを、データベース内の実際のデータ（インデックスも含む）とは別のデバイスに保管する必要があります。ハード・ディスクの障害が起きても、ログは別のデバイスに安全に保存されているので、データベースのダンプをロードしてからこのログ・レコードを適用することによって、最新のデータベースを作成できます。create database の log on 句の詳細については、『システム管理ガイド 第2巻』の「第6章 ユーザ・データベースの作成と管理」を参照してください。

ミラーリング

ハード・ディスクの障害が発生した場合に確実にノンストップ・リカバリできるようにするには、Adaptive Server の全デバイスを別の物理ディスクにミラーリングします。『システム管理ガイド 第2巻』の「第2章 データベース・デバイスのミラーリング」を参照してください。

パフォーマンス

ログとデータベース・オブジェクトを別々のデバイスに配置すると、システムのパフォーマンスを向上させることができます。

- テーブルをあるハード・ディスクに配置し、ノンクラスタード・インデックスを別のハード・ディスクに配置すると、処理が2台のディスク・ドライブに分割されるため、物理的な読み込みと書き込みが速くなります。
- サイズの大きなテーブルを2台のディスクに分割すると、特にマルチユーザ・アプリケーションのパフォーマンスが向上します。
- ログとデータが同じデバイスに配置されている場合は、トランザクション・ログ・レコードのユーザ・ログ・キャッシュ・バッファリングは行われません。
- テーブルを分割すると、ヒープ・テーブルに複数の挿入ポイントが作成され、並列クエリ処理を実行するように設定されているシステムの並列処理度が増加します。また、テーブルのI/Oを複数のデータベース・デバイス間に分散できます。

オブジェクトの配置がパフォーマンスに及ぼす影響については、『パフォーマンス&チューニング・シリーズ：物理データベースのチューニング』の「第1章データの物理的配置の制御」を参照してください。

インストール時のステータスおよびデフォルト設定

インストール・プログラムとスクリプトによって、マスタ・デバイスが初期化され、**master**、**model**、**sybssystemprocs**、**sybsecurity** の各データベースとテンポラリ・データベースが設定されます。

Adaptive Server をインストールすると、システム・データベース、システム定義のセグメント、データベース・デバイスは、次のような構成で設定されます。

- **master**、**model**、**tempdb** の各データベースは、マスタ・デバイス上にインストールされます。
- **sybssystemprocs** データベースは、指定のデバイス上にインストールされます。
- 各データベースに **system**、**default**、**logsegment** の3つのセグメントが作成されます。

- マスタ・デバイスが、ユーザが作成するすべてのデータベース用のデフォルト記憶デバイスとなります。

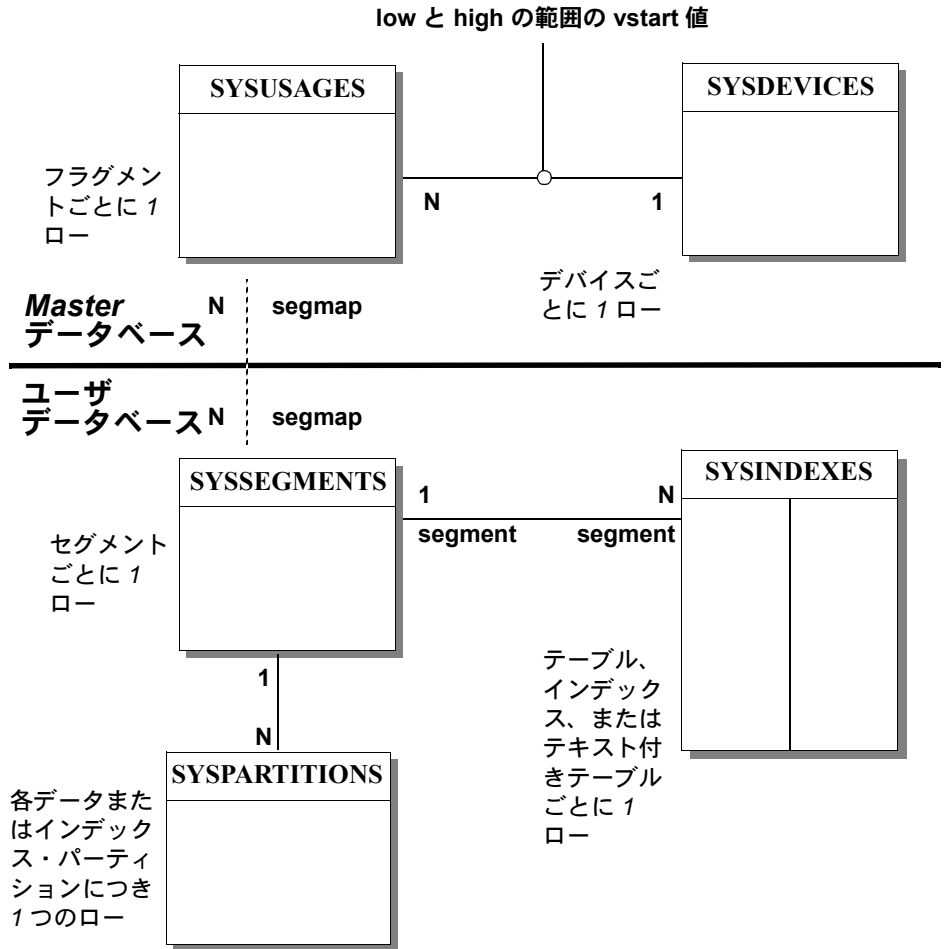
注意 デフォルトの記憶領域用の新しいデバイスを初期化した後は、`sp_diskdefault` を使用してマスタ・デバイスをデフォルトの記憶領域から除外してください。ユーザのデータベースやオブジェクトをマスタ・デバイスに格納しないでください。「[デフォルト・デバイスの指定](#)」(307 ページ)を参照してください。

- 監査データベース `sybsecurity` をインストールする場合は、このデータベースは専用のデバイス上に配置されます。

記憶領域を管理するシステム・テーブル

master データベース内の 2 つのシステム・テーブル `sysusages` および `sysdevices` と、各ユーザ・データベース内の 3 つのシステム・テーブル `syssegments`、`sysindexes`、および `syspartitions` は、データベース、テーブル (トランザクション・ログ・テーブル `syslogs` を含む)、およびインデックスの配置を記録します。これらのテーブルの関係を [図 6-1](#) に示します。

図 6-1: 記憶領域を管理するシステム・テーブル



sysdevices テーブル

master データベース内の **sysdevices** テーブルには、「データベース・デバイス」ごとにローが 1 つ格納されます。また、Adaptive Server で使用できるダンプ・デバイス (テープ、ディスク、またはオペレーティング・システム・ファイル) ごとに 1 つのローが格納されることがあります。

disk init コマンドを実行すると、データベース・デバイスに対応するエントリが master..sysdevices に追加されます。sp_addumpdevice を使用して追加されるダンプ・デバイスについては、『システム管理者ガイド第 2 巻』の「第 11 章 バックアップおよびリカバリ・プランの作成」を参照してください。

`sysdevices` には、デバイスごとに次の 2 つの名前が格納されます。

- 論理名またはデバイス名：以降のすべての記憶領域管理コマンドで使用される名前です。この名前は、`sysdevices` の `name` カラムに保管されます。通常は、ユーザにわかりやすい名前を付け、できればデバイスの用途を示すような名前にします (たとえば、“`logdev`” や “`userdbdev`”)
- 物理デバイス名：オペレーティング・システムでのこのデバイスの実際の名前です。この名前を使用するのは `disk init` コマンドだけです。それ以降の Adaptive Server のデータ記憶領域コマンドはすべて論理名を使用します。

データベースまたはトランザクション・ログを 1 つまたは複数のデバイス上に配置するには、`create database` 文または `alter database` 文内でデバイスの論理名を指定します。完全なりカバリを確実に実行できるようにするには、`create database` に `log on` 句を指定して、データベースのトランザクション・ログを別のデバイス上に配置します。また、`log on` を使用するには、`sysdevices` 内でログ・デバイスのエントリが割り当てられている必要があります。

1 つのデータベースを 1 つまたは複数のデバイス上に配置できます。また、1 つのデバイスに 1 つ以上のデータベースを保管することもできます。特定のデータベース・デバイス上にデータベースを作成する方法の詳細については、『システム管理ガイド 第 2 巻』の「第 6 章 ユーザ・データベースの作成と管理」を参照してください。

sysusages テーブル

`master` データベース内の `sysusages` テーブルには、Adaptive Server のデータベースに割り当てられているすべての領域が記録されています。

`create database` と `alter database` を実行すると、データベース・デバイスまたはデバイス・フラグメントごとに 1 つのローが `sysusages` に追加され、これによって新しい領域がデータベースに割り当てられます。`create` または `alter database` によってデバイス上の領域の一部だけが割り付けられた場合に、その部分を「フラグメント」と呼びます。

`sp_addsegment`、`sp_dropsegment`、`sp_extendsegment` を実行すると、デバイスに対応する `sysusages` テーブル内の `segmap` カラムが変更され、デバイスがセグメントにマッピングされるか、マッピングが解除されます。『システム管理ガイド 第 2 巻』の「第 8 章 セグメントの作成と使用」を参照してください。

syssegments テーブル

各データベースに1つずつ存在する **syssegments** テーブルには、データベース内のセグメントが登録されています。「セグメント」とは、特定のデータベースで使用できるデータベース・デバイスやフラグメントの集合のことです。テーブルやインデックスは、特定のセグメントに割り当てる、つまり特定の物理デバイスに割り当てることも、複数の物理デバイスの集合に割り当てることもできます。

create database コマンドは、**syssegments** テーブル内にデフォルトのエントリを作成します。**sp_addsegment** と **sp_dropsegment** は、**syssegments** テーブルのエントリの追加と削除を行います。

sysindexes テーブル

sysindexes テーブルには、各テーブルとインデックスが登録されています。また、各テーブル、クラスタド・インデックス、ノンクラスタド・インデックスが存在するセグメントや、テキスト・ページのチェーンが存在するセグメントも記録されています。さらに、そのテーブルまたはインデックスに対する **max_rows_per_page** 設定など、その他の情報も記録されています。

create table、**create index**、**alter table** の各コマンドは、**sysindexes** 内に新しいローを作成します。テーブルを分割すると、**sysindexes** 内のそのテーブルに対応するエントリの機能が変わります。

syspartitions テーブル

syspartitions テーブルには、各テーブルとインデックス・パーティションが登録されています。また、そのパーティションが存在するセグメントも記録されています。**syspartitions** は、記憶領域の管理に関する重要な情報(データ・ページ・チェーンまたはインデックス・ページ・チェーンの最初のページ、ヒープの最後のページ、インデックス・パーティションのルート・ページなど)を保持します。

create table、**create index**、**alter table** の各コマンドは、**syspartitions** 内に新しいローを作成します。

この章では、各 Adaptive Server のシステム管理者とシステム・セキュリティ担当者が、「リモート・プロシージャ・コール」(RPC)を使用するために実行する必要がある手順について説明します。

トピック名	ページ
概要	283
リモート・サーバの管理	285
リモート・ログインの追加	289
リモート・ユーザのパスワードの検査	294
リモート・ログイン情報の取得	295
リモート・ログインの設定パラメータ	295

概要

ローカル Adaptive Server 上のユーザは、リモート Adaptive Server 上のストアド・プロシージャを実行できます。RPC を実行すると、リモート・プロセスの結果が、呼び出し元プロセスに送信されます。これは通常、ユーザの画面に表示されます。

RPC を使用できるようにするには、各 Adaptive Server のシステム管理者とシステム・セキュリティ担当者が次の手順を実行する必要があります。

- ローカル・サーバ上
 - システム・セキュリティ担当者が `sp_addserver` を使用して、システム・テーブル `master..sys.servers` にローカル・サーバとリモート・サーバを登録します。
 - リモート・サーバを、ローカル・サーバの `interfaces` ファイルまたはディレクトリ・サービスに登録します。
 - ローカル・サーバを再起動します。これによって、グローバル変数 `@@servername` がローカル・サーバの名前に設定されます。この変数が正しく設定されていない場合は、ローカル・サーバからリモート・サーバ上で RPC を実行することはできません。

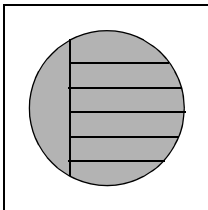
- リモート・サーバ上
 - システム・セキュリティ担当者が `sp_addserver` を使用して、システム・テーブル `master.syssservers` に RPC の発信元のサーバを登録します。
 - リモート・プロシージャを発信するユーザがこのサーバにアクセスできるようにするために、システム・セキュリティ担当者が `create login` を実行し、システム管理者が `sp_addremotelogin` を実行します。
 - リモート・ログイン名を、該当するデータベースのユーザとして追加し、プロシージャを実行するパーミッションを付与します (`execute` パーミッションが “public” に付与されている場合は、ユーザに特定のパーミッションを付与する必要はありません)。

図 7-1 は、リモート・アクセスができるようにサーバを設定する方法を示します。

図 7-1: リモート・プロシージャ・コールを可能にするサーバ設定

ROSE 上のユーザ “joe” が ZINNIA 上のストアド・プロシージャにアクセスする必要がある場合

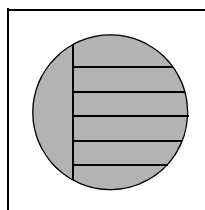
ROSE



`sp_addserver` ROSE, local
`sp_addserver` ZINNIA

Interfaces ファイルに ZINNIA の
 エントリが必要

ZINNIA



`sp_addserver` ROSE
`create login` joe
`sp_addremotelogin` ROSE, joe

`sp_adduser` joe (該当するデータベース上で)
`grant execute` on *procedure_name* to joe

リモート・サーバの処理に関するオペレーティング・システム固有の情報については、プラットフォームの『インストール・ガイド』を参照してください。

リモート・サーバの管理

表 7-1 は、リモート・サーバの管理に関連する作業と、その作業を実行するために使用するシステム・プロシージャを示します。

表 7-1: リモート・サーバの管理に関連する作業

目的	使用	参照箇所
リモート・サーバの追加	sp_addserver	「リモート・サーバの追加」(285 ページ)
リモート・サーバ名の管理	sp_addserver	「リモート・サーバ名の管理」(287 ページ)
サーバ接続オプションの変更	sp_serveroption	「サーバ接続オプションの設定」(287 ページ)
サーバに関する情報の表示	sp_helpserver	「サーバ情報の取得」(289 ページ)
サーバの削除	sp_dropserver	「リモート・サーバの削除」(289 ページ)

リモート・サーバの追加

システム・セキュリティ担当者は、sp_addserver を使用して、syssservers テーブルにエントリを追加します。RPC を発信するサーバでは、ローカル・サーバのエントリ 1 つと、呼び出すリモート・サーバごとに 1 つのエントリを追加する必要があります。

リモート・サーバのエントリを作成するときは、次のどちらかを選択できます。

- *interfaces* ファイルに登録されている名前でもリモート・サーバを参照する。
- リモート・サーバに対してローカル名を与える。たとえば、*interfaces* ファイルでの名前が“MAIN_PRODUCTION”であるリモート・サーバに、“main”という名前を付けます。

構文は次のとおりです。

```
sp_addserver lname [{, local | null}
[, pname]]
```

各パラメータの意味は次のとおりです。

- *lname* には、リモート・サーバのローカルでの「呼び名」を指定します。この名前が *interfaces* ファイルでのリモート・サーバの名前と同じでない場合は、その名前を 3 番目のパラメータ *pname* として指定する必要があります。

リモート・サーバは、ローカル・マシン上の *interfaces* ファイルに登録されていなければなりません。登録されていない場合は、リモート・サーバから *interfaces* ファイルのエントリをコピーして、既存の *interfaces* ファイルに追加します。同じポート番号を使用してください。

- `local` は、このサーバがローカル・サーバとして追加されることを示します。`local` という値が使用されるのは、起動または再起動の後のみで、ここで指定されたローカル・サーバ名が Adaptive Server の出力メッセージに表示されます。`null` は、このサーバがリモート・サーバであることを示します。

注意 ユーザがローカル・サーバから RPC を正常に実行できるようにするには、`local` オプションを指定してローカル・サーバを追加し、再起動する必要があります。再起動は、グローバル変数 `@@servername` を設定するために必要です。

- `pname` には、リモート・サーバを指定します。これは、`lname` という名前で `interfaces` ファイルに登録されているサーバです。このオプション引数を指定すると、通信する必要がある他の Adaptive Server、Open Server、Backup Server 用のローカル・エイリアスを設定できます。`pname` を指定しない場合のデフォルト値は `lname` になります。

リモート・サーバの追加の例

次の例は DOCS という名前のローカル・サーバ用のエントリを作成します。

```
sp_addserver DOCS, local
```

次の例は GATEWAY という名前のリモート・サーバ用のエントリを作成します。

```
sp_addserver GATEWAY
```

GATEWAY サーバ上で `sp_who` などのリモート・プロシージャを実行するには、次のいずれかを実行します。

```
GATEWAY.sybsystemprocs.dbo.sp_who
```

または

```
GATEWAY...sp_who
```

次の例は、MAIN_PRODUCTION という名前のリモート・サーバにローカル・エイリアス “main” を与えます。

```
sp_addserver main, null, MAIN_PRODUCTION
```

この場合は、ユーザは次のように入力できます。

```
main...sp_who
```

リモート・サーバ名の管理

master.dbo.syssservers テーブルには、次の2つのサーバ名のカラムがあります。

- `srvname` は、ユーザがリモート・プロシージャ・コールを実行するときに指定するユニークなサーバ名です。
- `srvnetname` はサーバのネットワーク名であり、`interfaces` ファイル内の名前と一致する必要があります。

サーバをネットワークに追加する場合やネットワークから削除する場合は、`sp_addserver` を使用して、`srvnetname` のサーバのネットワーク名を更新します。

たとえば、サーバMAINをネットワークから削除して、リモート・アプリケーションをTEMPという名前のサーバに移動するには、次の文を使用すると、ローカル・エイリアスを保持したままネットワーク名を変更できます。

```
sp_addserver MAIN, null, TEMP
```

`sp_addserver` は、既存のサーバ・エントリのネットワーク名を変更しようとしていることを通知するメッセージを表示します。

サーバ接続オプションの設定

`sp_serveroption` は、リモート・サーバとの接続に影響を与えるサーバ・オプション `timeouts`、`net password encryption`、および `security mechanism` を設定します。

`sp_serveroption` で指定するオプションは、Adaptive Server と Backup Server の間の通信には影響しません。

次の各項では、`timeouts`、`net password encryption` について説明します。

`timeouts` オプションの使用

システム管理者は `timeouts` オプションを使用して、ローカル・サーバが使用する通常のタイムアウト・コードを無効または有効にできます。

デフォルトでは、`timeouts` は `true` に設定され、リモート・ユーザのアクティビティが行われない状態が1分間続くと、リモート・ログインを管理するサイト・ハンドラ・プロセスはタイムアウトになります。リモート・プロシージャ・コールに関係するサーバの両方で `timeouts` が `false` に設定されている場合は、自動タイムアウトは行われません。`timeouts` を `false` に変更するには次のようになります。

```
sp_serveroption GATEWAY, "timeouts", false
```

両方のサーバで `timeouts` が `false` に設定された後で、いずれかのサーバから RPC が実行されると、各マシンのサイト・ハンドラはどちらかのサーバが停止するまで動作を続けます。サーバが再び起動されると、オプションは `false` のままで、ユーザが次に RPC を実行するときにサイト・ハンドラが再設定されます。ユーザが頻繁に RPC を実行する場合は、このオプションを `false` に設定した方が、システム・リソースの点からは効率的です。物理的接続の設定にはかなりのシステム・オーバーヘッドがかかるからです。

net password encryption オプションの使用

システム・セキュリティ担当者は `net password encryption` オプションを使用して、リモート・サーバとの接続をクライアント側パスワード暗号化ハンドシェイクによって開始するか、または通常の非暗号化パスワード・ハンドシェイク・シーケンスによって開始するかを指定できます。デフォルトは `false` です。

`net password encryption` オプションを `true` に設定すると、処理の順序は次のようになります。

- 1 ログイン開始パケットが、パスワードなしで送信されます。
- 2 クライアントは、暗号化が要求されていることをリモート・サーバに知らせます。
- 3 リモート・サーバは暗号化キーを返します。これは、クライアントがプレーン・テキストのパスワードを暗号化するために使用するキーです。
- 4 次にクライアントは自分のパスワードを暗号化し、リモート・サーバは渡されたパスワードをこのキーを使って認証します。

次の例は、`net password encryption` を `true` に設定します。

```
sp_serveroption GATEWAY, "net password encryption",  
true
```

security mechanism オプションの使用

`security mechanism` オプションを使用すると、リモート・サーバへの接続に対して Kerberos 認証を有効にできます。

注意 Adaptive Server リリース 15.7 以降では、セキュリティ・メカニズムとして Kerberos がサポートされています。リモート・メカニズムへのセキュリティ・メカニズムを有効にするために、`rpc security model A` と `rpc security model B` は不要で、`sp_helpserver` は `rpc` セキュリティ・モデルを表示しません。下位互換性のために、このリリースにアップグレードされた DCE メカニズムを使用するデータベースは以前と同様に機能します。

サーバ情報の取得

`sp_helpserver` はサーバに関する情報を表示します。引数を指定しないで `sp_helpserver` を使用する場合は、`sys.servers` に登録されているすべてのサーバについての情報が表示されます。サーバ名を指定して `sp_helpserver` を使用すると、そのサーバの情報だけが表示されます。

```
sp_helpserver [server]
```

`sp_helpserver` は、`master.sysremotelogins` テーブル内の `srvname` と `srvnetname` の両方をチェックします。

リモート・サーバの設定に関するオペレーティング・システム固有の情報については、プラットフォームの『インストール・ガイド』を参照してください。

リモート・サーバの削除

システム・セキュリティ担当者は、`sp_dropserver` システム・プロシージャを使用して、`sys.servers` からサーバを削除できます。

```
sp_dropserver server [, droplogins]
```

各パラメータの意味は次のとおりです。

- `server` は、削除するサーバの名前です。
- `droplogins` を使用すると、リモート・サーバと、そのサーバのリモート・ログイン情報すべてを削除できます。`droplogins` オプションを使用しない場合は、リモート・ログインが関連付けられているサーバを削除することはできません。

次の文は、GATEWAY サーバと、このサーバに関連付けられているすべてのリモート・ログインを削除します。

```
sp_dropserver GATEWAY, droplogins
```

ローカル・サーバの削除には `droplogins` を使用する必要はありません。ローカル・サーバに関連付けられているリモート・ログイン情報はないためです。

リモート・ログインの追加

Adaptive Server のシステム・セキュリティ担当者とシステム管理者は、そのサーバにアクセスできるリモート・ユーザとそのリモート・ユーザの ID の管理を分担します。システム管理者は `sp_addremotelogin` を使ってリモート・ログインを追加し、`sp_dropremotelogin` を使ってリモート・ログインを削除します。システム・セキュリティ担当者は、`sp_remoteoption` を使って、パスワードの検査が必要かどうかを管理します。

ユーザのサーバ ID のマッピング方法

リモート・サーバからのログインをローカル・サーバにマップするには、次の3つの方法があります。

- 特定のリモート・ログインを特定のローカル・ログイン名にマップします。たとえば、リモート・サーバ上のユーザ“joe”を、“joesmith”にマップします。
- 1つのリモート・サーバからのすべてのログインを、1つのローカル名にマップします。たとえば、MAINサーバからリモート・プロシージャ・コールを送信するすべてのユーザを“remusers”にマップします。
- 1つのリモート・サーバからのすべてのログインが各自のリモート・ログイン名を使用するようにします。

最初のオプションは他の2つのオプションと組み合わせることができ、その個別のマッピングは他の2つの全体的なマッピングに優先します。2番目と3番目のオプションは互いに排他的です。どちらか一方だけを使用することはできません。

マッピング・オプション
の変更

`sp_dropremotelogin` を使用して、古いマッピングを削除します。

`sp_addremotelogin` を使用して、リモート・ログインを追加します。

```
sp_addremotelogin remoteserver [, loginame
    [, remotename]]
```

ローカル名が `master.syslogins` に登録されていない場合は、`create login` を使って Adaptive Server ログインとして追加してから、リモート・ログインを追加してください。

`sp_addremotelogin` を実行できるのは、システム管理者だけです。『リファレンス・マニュアル：プロシージャ』を参照してください。

リモート・ログインを特定のローカル名にマップする方法

次の例は、リモート・システムの“pogo”という名前前のログインを、ローカル・ログイン名“bob”にマップします。このユーザは“pogo”としてリモート・システムにログインします。ユーザ“pogo”が GATEWAY からリモート・プロシージャ・コールを実行するとき、ローカル・システムはリモート・ログイン名を“bob”にマップします。

```
create logn bob with password itsA8secret
sp_addremotelogin GATEWAY, bob, pogo
```

すべてのリモート・ログインを1つのローカル名にマップする方法

次の例は、すべてのリモート・ログイン名をローカル名“albert”にマップするエントリを作成します。前の項で説明したように、個別のマッピングを持つ名前以外のすべての名前が“albert”にマップされます。たとえば、“pogo”を“bob”にマップして、それ以外のすべてのログインを“albert”にマップした場合も、“pogo”は“bob”にマップされたままになります。

```
create login albert with password itsA8secret
sp_addremotelogin GATEWAY, albert
```

`sp_addremotelogin` を使用してリモート・サーバのすべてのユーザを同じローカル名にマップする場合は、`sp_remotoption` を使用して、それらのユーザに“trusted” オプションを指定します。たとえば、“albert”にマップされるサーバ GATEWAY のユーザすべてを trusted にするには、次のように指定します。

```
sp_remotoption GATEWAY, albert, NULL, trusted, true
```

trusted が指定されていないログインは、そのローカル・サーバで RPC を実行することはできません。ただし、リモート・サーバへのログイン時に、ローカル・サーバのパスワードを指定すれば、ローカル・サーバ上で RPC を実行できます。これらのユーザは、Open Client Client-Library を使用するとき、`ct_remote_pwd` ルーチンを使用してサーバ対サーバ接続のパスワードを指定することができます。isql と bcp は、ユーザが RPC 接続用のパスワードを指定することを許可しません。`sp_remotoption` の詳細については、「[リモート・ユーザのパスワードの検査](#)」(294 ページ)を参照してください。

警告！ 複数のリモート・ログインを1つのローカル・ログインにマップしないでください。サーバ上での個人の責任が不明確になります。監査対象のアクションの追跡によって特定できるのはローカル・サーバのログインだけであり、リモート・サーバ上の個々のログインを特定することはできません。

ネットワークベース・セキュリティの使用

ユーザが「統一化ログイン」を使用してリモート・サーバにログインする場合は、そのログインにはローカル・サーバでも trusted が指定されている必要があります。そうでない場合は、ユーザはリモート・サーバにログインするときにサーバのパスワードを指定する必要があります。

警告！ `sp_remotoption` の trusted モードを使用すると、このような trusted ユーザのパスワードが検証されないため、サーバのセキュリティは低くなります。

ローカル・サーバのリモート・ログイン名の保持

リモート・ユーザが、ローカル・サーバの使用時にリモート・ログイン名を保持できるようにするには、次の手順に従います。

- 1 `create login` を使用して、リモート・サーバからの各ログインに対するログインを作成します。
- 2 `sp_addremotelogin` を使用して、サーバ全体に対する1つのエントリを `master.sysremotelogins` に作成します。リモート・ログイン名の値は `null`、`suid` の値は `-1` とします。次に例を示します。

```
sp_addremotelogin GATEWAY
```

リモート・ユーザ・ログインのマッピング例

次の文は、`master.syssservers` に記録されているローカル・サーバとリモート・サーバの情報を表示します。

```
select srvid, srvname from syssservers
srvid  srvname
-----
0      SALES
1      CORPORATE
2      MARKETING
3      PUBLICATIONS
4      ENGINEERING
```

SALES サーバはローカル・サーバです。それ以外のサーバは、リモート・サーバです。

次の文は、`master.sysremotelogins` に保管されているリモート・サーバとユーザに関する情報を表示します。

```
select remoteserverid, remoteusername, suid
from sysremotelogins
remoteserverid  remoteusername  suid
-----
1                joe                1
1                nancy              2
1                NULL              3
3                NULL              4
4                NULL              -1
```

この結果における `remoteserverid` の値を、前の結果の `srvid` の値と照合すれば、`remoteusername` がどのサーバで有効であるかがわかります。たとえば最初の結果では、`srvid 1` は、CORPORATE サーバを示します。2 番目の結果では、`remoteserverid 1` は、同じサーバを示します。したがって、リモート・ユーザ・ログイン名 “joe” と “nancy” は、CORPORATE サーバ上で有効です。

次の文は、`master.syslogins` のエントリを表示します。

```
select suid, name from syslogins
suid    name
-----
1      sa
2      vp
3      admin
4      writer
```

この3つのクエリの結果から、次のことがわかります。

- リモートの CORPORATE サーバ (`sruvid` と `remoteserverid` が 1) 上のリモート・ユーザ名 “joe” (`suid` 1) は、“sa” ログイン (`suid` 1) にマップされます。
- リモートの CORPORATE サーバ (`sruvid` と `remoteserverid` が 1) 上のリモート・ユーザ名 “nancy” (`suid` 2) は、“vp” ログイン (`suid` 2) にマップされます。
- CORPORATE サーバの他のログイン (`remoteusername` “NULL” は、“admin” ログイン (`suid` 3) にマップされます。
- PUBLICATIONS サーバ (`sruvid` と `remoteserverid` が 3) のログインはすべて “writer” ログイン (`suid` 4) にマップされます。
- ENGINEERING サーバ (`sruvid` と `remoteserverid` が 4) のすべてのログインは、そのリモート・ユーザ名 (`suid` -1) によって `master.syslogins` での検索が行われます。
- `sysremotelogins` には、`remoteserverid` が MARKETING サーバを示すエントリはありません。したがって、MARKETING サーバにログインしたユーザが、そのサーバからリモート・プロシージャ・コールを実行することはできません。

リモート・ユーザのマップの手順と個々のストアド・プロシージャに対してパーミッションを設定する機能によって、どのリモート・ユーザがローカル・プロシージャにアクセスできるかを管理できます。たとえば、CORPORATE サーバからの “vp” ログインが特定のローカル・プロシージャを実行できるようにして、それ以外の CORPORATE のすべてのログインについては “admin” ログインがパーミッションを所有しているプロシージャを実行できるようにします。

注意 通常は、リモート・サーバのユーザのパスワードはローカル・サーバのパスワードと一致していなければなりません。

リモート・ユーザのパスワードの検査

システム・セキュリティ担当者は、`sp_remoteoption` を使用して、リモート・ユーザがローカル・サーバにログインするときにパスワードの検査を行うかどうかを設定できます。デフォルトでは、パスワードを検査します (“untrusted”モード)。trusted モードでは、ローカル・サーバはログイン・アカウントについてのユーザ・アクセスの確認を行わずに、他のサーバやフロントエンド・アプリケーションからのリモート・ログインを許可します。

引数を指定して `sp_remoteoption` を実行すると、指定したユーザのモードを変更できます。

```
sp_remoteoption [remoteserver, loginame, remotename,  
optname, {true | false}]
```

次の例は、ユーザ “bob” に trusted モードを設定します。

```
sp_remoteoption GATEWAY, pogo, bob, trusted,  
true
```

untrusted モードを使用した場合の影響

untrusted モードを使用した場合の影響は、ユーザのクライアント・プログラムによって異なります。isql や一部のユーザ・アプリケーションでは、リモート・サーバとローカル・サーバでログインのパスワードが同じでなければなりません。Open Client アプリケーションの場合は、ローカル・ログインのパスワードがサーバごとに異なってもかまわないこともあります。

“untrusted” モードでパスワードを変更するには、アクセスするすべてのリモート・システム上でパスワードを変更してから、ローカル・サーバのパスワードを変更してください。ローカル・サーバのパスワードを先に変更すると、リモート・プロシージャ・コールを発行してリモート・サーバ上で `sp_password` を実行するときに、パスワードが一致しないこととなります。

リモート・サーバ上のパスワードを変更する構文は次のとおりです。

```
remote_server...sp_password caller_passwd, new_passwd
```

ローカル・サーバでの構文は次のとおりです。

```
sp_password caller_passwd, new_passwd
```

『セキュリティ管理ガイド』の「第5章 Adaptive Server のログイン・アカウントとデータベース・ユーザの管理」を参照してください。

リモート・ログイン情報の取得

`sp_helpremotelogin` は、サーバ上のリモート・ログインについての情報を表示します。次の例では、リモート・ログイン“pogo”は、ローカルではログイン名“bob”にマッピングされていることがわかります。他のリモート・ログインはすべてリモート名をそのまま使います。

```
sp_helpremotelogin
```

server	remote_user_name	local_user_name	options
GATEWAY	**mapped locally**	**use local name**	untrusted
GATEWAY	pogo	bob	untrusted

リモート・ログインの設定パラメータ

表 7-2 は、RPC に影響を与える設定パラメータを示します。これらの設定パラメータはすべて `sp_configure` を使用して設定されます。また、Adaptive Server を再起動しないと有効になりません。

表 7-2: RPC に影響を与える設定パラメータ

設定パラメータ	デフォルト
allow remote access	1
number of remote logins	20
number of remote sites	10
number of remote connections	20
remote server pre-read packets	3

個々の設定パラメータの詳細については、「[第5章 設定パラメータ](#)」を参照してください。

トピック名	ページ
データベース・デバイス	297
disk init コマンドの使用	298
disk init の構文	299
デバイス情報の表示	305
デバイスの削除	307
デフォルト・デバイスの指定	307
disk resize コマンドによるデバイスのサイズ拡大	308

データベース・デバイス

データベース・デバイスには、データベースを構成するオブジェクトが格納されます。「デバイス」という語は、1つの物理デバイスを指すとは限りません。データベースやそのオブジェクトを保管するための、ディスクの一部(ディスク・パーティションなど)やファイル・システム内のファイルをデバイスと呼ぶこともあります。

データベース・デバイスまたはファイルをデータベース記憶領域として使用するには、Adaptive Server によって認識されるように準備する必要があります。この処理を「初期化」と呼びます。

初期化されたデータベース・デバイスに対しては、次の処理が実行できます。

- `create database` コマンドと `alter database` コマンドで使用できるようにデフォルト・デバイス・プールに割り当てる。
- ユーザ・データベースが使用できる領域のプールに割り当てる。
- ユーザ・データベースに割り当てて、1つまたは複数のデータベース・オブジェクトの保管に使用する。
- データベースのトランザクション・ログを保管するために割り当てる。

disk init コマンドの使用

システム管理者は、**disk init** コマンドを使用して、新しいデータベース・デバイスを初期化します。このコマンドは次の処理を行います。

- 指定された物理ディスク・デバイスまたはオペレーティング・システム・ファイルをデータベース・デバイス名にマッピングする。
- **master.sysdevices** に新しいデバイスを登録する。
- そのデバイスをデータベース記憶領域として使用できるように準備する。

注意 **disk init** を実行する前に、プラットフォームの『ASE インストール・ガイド』を参照して、データベース・デバイスの選択方法と、Adaptive Server で使用するためにそのデータベース・デバイスを準備する方法を確認してください。Sybase データベースのパフォーマンスを最高にするために、コンピュータのディスク・パーティションの再設定が必要な場合もあります。

disk init コマンドは、データベース・デバイスを複数の「アロケーション・ユニット」に分割します。アロケーション・ユニットは、256 論理ページから構成される 1 つのグループです。アロケーション・ユニットのサイズは、サーバに設定されている論理ページ・サイズ (2、4、8、16K) によって異なります。**disk init** コマンドは、各アロケーション・ユニットの最初のページをアロケーション・ページとして初期化します。そのアロケーション・ユニットにデータベースがある場合は、その情報がこのページに保管されます。

警告！ **disk init** コマンドの実行後は、必ず **master** データベースのダンプを行ってください。このようにすれば、**master** が損傷した場合でも、簡単かつ確実にリカバリできます。『システム管理ガイド 第 2 巻』の「第 13 章 システム・データベースのリストア」を参照してください。

disk init の構文

disk init の構文については、『リファレンス・マニュアル：コマンド』を参照してください。

論理デバイス名の指定

device_name は、有効な識別子でなければなりません。この名前は、**create database** コマンドと **alter database** コマンド、およびセグメントを管理するシステム・プロシージャで使用します。論理デバイス名は、Adaptive Server だけが認識するもので、サーバが実行されるオペレーティング・システムには認識されません。

物理デバイス名の指定

データベース・デバイスの *physname* には、ロー・ディスク・パーティション (UNIX) の名前、外部デバイスの名前、またはオペレーティング・システム・ファイルの名前を指定します。プラットフォームが PC の場合は、通常、*physname* にはオペレーティング・システム・ファイルの名前を指定します。

デバイス番号の選択

Adaptive Server は **disk init vdevno** パラメータを受け付けますが、このパラメータは必須ではありません。vdevno を指定する場合、1 ~ 2,147,483,647 の識別子から現在未使用のものを選択できます (仮想デバイス ID 0 は **master** デバイスで使用しています)。たとえば、vdevno = 33 を指定すると、デバイスに仮想デバイス ID 33 が割り当てられます。vdevno を指定しない場合、sysdevices に登録されている最大の vdevno より大きい数字が選択されます。

作成可能なデータベース・デバイス数は、**number of devices** 設定パラメータによって制限されます。Adaptive Server の初期設定では、デバイス数は 10 です。sp_configure を使用して、**number of devices** の値を変更します。

インストールされているシステム上で同時に使用可能なデバイス数が、オペレーティング・システムによって制限されている場合もあります。オペレーティング・システムでは、各 Sybase デバイスが 1 つのオープン・ファイルと見なされます。

Adaptive Server は、次に使用できるデータベース・デバイス識別番号を自動的に指定します。これが、vdevno (仮想デバイス番号、virtual device number) です。disk init コマンドを発行するときに、この番号を指定する必要はありません。

手動で vdevno を選択する場合は、Adaptive Server によって使用されるデバイス間でユニークな番号を指定してください。デバイス番号 0 は、マスタ・デバイスを表します。有効な番号は 1 ~ 2,147,483,647 です。この値の範囲内で未使用の vdevno を選択できます。

既に `vdevno` に使用されている番号を確認するには、`sp_helpdevice` からのレポートの `vdevno` カラムを調べるか、または次のクエリを実行し、現在使用中のデバイス番号をすべてリストします。

```
select vdevno from master..sysdevices
      where status & 2=2
```

ここで、`status & 2=2` は物理ディスクを示します。

デバイス・サイズの指定

デバイスのサイズを指定するには、キロバイトを示す場合は 'k' または 'K'、メガバイトを示す場合は 'm' または 'M'、ギガバイトを示す場合は 'g' または 'G'、テラバイトを示す場合は 't' または 'T' を使用します。`disk init` コマンドと `create database` コマンドのどちらも、必ず単位指定子を入力することをおすすめします。これは、実際に割り付けられるページ数との混同を避けるためです。単位指定子は、一重引用符、二重引用符、または角カッコで囲んでください。

理論上は、最大 2,147,483,647 個のディスク・デバイスを作成し、各デバイスに最大 2,147,483,648 の 2K ブロックを割り当てることができます。最大インストール可能サイズが、データベース・サイズ、ハードウェア、およびオペレーティング・システムの事実上の制限になります。

次のガイドラインが、`disk init` の構文に適用されます。

- `disk init` または `disk reinit` の `size` 引数で単位を指定しなかった場合、`size` はデフォルトで仮想ページの数と解釈されます。たとえば、`size = 15000` と入力すると、Adaptive Server では 15,000 仮想ページと解釈されます。仮想ページは 1 ページあたり 2048 バイトです。
- `disk resize` コマンドを使用して既存のデータベース・デバイスのサイズを増やすことはできますが、減らすことはできません。
- 新しいデータベースを作成するための新しいデバイスの最小サイズ (`size`) は、サーバが使用する論理ページ・サイズによって異なる。詳細については、[表 8-1](#) を参照。

表 8-1: 最小データベース・サイズ

論理ページ・サイズ	最小データベース・サイズ
2K	3MB
4K	6MB
8K	12MB
16K	24MB

データベースを `model` データベースより小さくすることはできません。`model` データベースが上記の最小サイズより大きい場合、そのサイズが最小データベース・サイズになります。

Adaptive Server はアロケーション・ユニット (256 論理ページから成るグループ) のデータベース領域を割り付けおよび管理します。`create database` を使用して作成可能な最小データベースは 1 MB であるため、使用可能なデータベース・デバイスの最小サイズは 1 MB と 256 論理ページのどちらか大きい方になります (論理ページ・サイズが 2K または 4K の場合は 1 MB、論理ページ・サイズが 8K の場合は 2 MB、論理ページ・サイズが 16K の場合は 4 MB)。

デバイスのサイズを決めるときは、この 256 ページ単位のグループを考慮すると、領域の無駄を避けることができます。たとえば、インストールされているシステムで 16k の論理ページ・サイズを使用している場合、デバイス・サイズを `size = '31M'` に指定すると、アロケーション・ユニットが 4MB になるため、デバイスの最後の 3MB が無駄になります。

ロー・デバイスを初期化する場合は、プラットフォームの『ASE インストール・ガイド』を参照して、オペレーティング・システムで使用可能なデバイス・サイズを調べてください。使用可能な合計サイズは、使用するプラットフォームの最大値までです。いったん Adaptive Server 用にディスクを初期化すると、そのロー・デバイスの領域は別の目的では使用できなくなります。

`disk init` は、`size` を使用して `sysdevices.high` の最終仮想ページ番号の値を計算します。`sysdevices.high` および `sysdevices.low` の値は、2K バイトのブロック (Adaptive Server の物理ディスク管理の単位) で構成される仮想ページのページ番号です。この値は、インストール環境の論理ページ・サイズとは異なることがあります。

注意 `size` パラメータで指定したブロック数が物理デバイスにない場合、`disk init` コマンドは実行できません。オプションの `vstart` パラメータを使用する場合は、`vstart` パラメータと `size` パラメータで指定したブロック数の合計が物理デバイス上に必要です。このブロック数を確保できない場合は、このコマンドは異常終了します。

dsync 設定の指定 (オプション)

UNIX のオペレーティング・システム・ファイル上で初期化されたデバイスの場合に、`dsync` 設定は、そのファイルへの書き込みをバッファリングするかどうかを制御します。`dsync` 設定がオンの場合、Adaptive Server は UNIX の `dsync` フラグを使用してデータベース・デバイス・ファイルを開きます。このフラグを使用すると、デバイス・ファイルへの書き込みが物理記憶メディアに対して直接行われるようになり、システム障害が発生した場合もデバイス上の Adaptive Server のデータをリカバリすることができます。

`dsync` がオフの場合は、UNIX のファイル・システムによってデバイス・ファイルへの書き込みがバッファリングされることもありますが、デバイス上のデータのリカバリは保証されません。`dsync` は、データ整合性が要求されていない場合にのみオフにしてください。

注意 ロー・パーティション上で初期化されたデバイスの場合、`dsync` 設定は無視されます。代わりに、データベース・デバイスへの書き込みは、物理メディアに対して直接行われます。

パフォーマンスへの `dsync` の影響

データベース・デバイス・ファイルで `dsync` 設定を使用すると、いくつかのパフォーマンスのトレードオフが発生します。

- Adaptive Server は、HP-UX のオペレーティング・システム・ファイルに対する非同期 I/O をサポートしません。
- データベース・デバイス・ファイルに `dsync` オプションが設定されている場合は、デバイス・ファイルに書き込む Adaptive Server エンジン、書き込みオペレーションが完了するまで待機します。このため、更新操作中は、パフォーマンスが低下する場合があります。
- `dsync` がオンの場合、データベース・デバイス・ファイルへの書き込みオペレーションは、以前のバージョンの Adaptive Server (`dsync` がサポートされていない場合) より遅くなる場合があります。これは、Adaptive Server が、キャッシュ・データを UNIX ファイル・システムのバッファにコピーする代わりに、データをディスクに書き込まなくてはならないからです。

最高の書き込みパフォーマンスが要求される場合 (ただし、システム障害後のデータ整合を必要としない場合)、`dsync` をオフにすると、以前のバージョンの Adaptive Server と同様のパフォーマンスを得ることができます。たとえば、専用のデバイス・ファイル上に `tempdb` を保管する場合に、`dsync` を使用しているときのパフォーマンスが許容範囲内でない場合は、`dsync` を無効に設定します。

- 読み込み操作の応答時間は、一般的に、デバイスを UNIX オペレーティング・システム・ファイル上に保管する方が、ロー・パーティション上に保管するよりも良くなります。デバイス・ファイルからのデータは、Adaptive Server のキャッシュだけでなく、UNIX ファイル・システムのキャッシュも利用できるため、物理ディスクにアクセスせずにより多くの読み込みを実行できる可能性があります。

dsync の制限事項

dsync を使用するときは、次の制限事項が適用されます。

- マスタ・デバイスの場合、dsync は、常に true に設定され、変更することはできません。
- sp_deviceattr プロシージャを使用してデバイス・ファイルの dsync 設定を変更するときは、変更内容を有効にするために Adaptive Server を再起動します。
- バージョン 12.x より前の Adaptive Server からアップグレードしたときに dsync が true に設定されるのは、マスタ・デバイス・ファイルのみです。他のデバイス・ファイルの dsync 設定を変更するには、sp_deviceattr を使用してください。
- ロー・パーティションに保管されるデータベース・デバイスに対する dsync 設定は無視されます。ロー・パーティションに保管されるデバイスへの書き込みは、常に物理メディアに対して直接行われます。
- directio パラメータと dsync パラメータは互いに排他的です。デバイスの dsync を true に設定した場合、同じデバイスの directio を true に設定することはできません。デバイスの directio を有効にするには、dsync の設定を false に変更しておく必要があります。

directio によるオペレーティング・システム・バッファの回避

disk init、disk reinit、sp_deviceattr の directio パラメータを指定することにより、オペレーティング・システムのバッファ・キャッシュを回避して、データをディスクに直接転送することができます。directio は、I/O の方法やパフォーマンス向上の効果の点ではロー・デバイスと同じですが、ロー・デバイスより使いやすく、ファイル・システム・デバイスの管理が容易です。directio は、マスタ・デバイスに設定することはできません。directio は、静的パラメータであるため、このパラメータを有効にするには、Adaptive Server を再起動します。

注意 directio は、すべてのプラットフォームで使用できるわけではありません。サポートされていないプラットフォームで、directio パラメータを使用して disk init を発行すると、Adaptive Server は、No such parameter: 'directio' というメッセージを発行します。

`directio` オプションのデフォルト値は、すべてのプラットフォームで `false` (オフ) に設定されます。

注意 リカバリが重要ではないデータベース (`tempdb` など) に使用するデバイスの `dsync` は、デフォルトで `false` に設定されている場合があります。このようなデバイスの `directio` を有効にすると、パフォーマンスが低下することがあるため、デバイスの用途を十分に検討してから `directio` を有効にしてください。

次の例では、`directio` を使ってデータをディスクに直接書き込む “`user_disk`” という名前のデバイスが作成されます。

```
disk init
name = "user_disk",
physname = "/usr/u/sybase/data/userfile1.dat",
size = 5120, directio= true
```

UNIX オペレーティング・システムのファイルでディスクの 10MB を初期化するには、次のように入力します。

```
disk reinit
name = "user_disk",
physname = "/usr/u/sybase/data/userfile1.dat",
size = 5120, directio= true
```

デフォルトでは、既存のすべてのデバイスの `directio` が無効に設定されているので、`sp_deviceattr` を使用して有効にします。

```
sp_deviceattr device_name, directio, [true | false]
```

たとえば、次のコマンドを実行すると、“`user_disk`” デバイスの `directio` によるディスク書き込みが有効になります。

```
sp_deviceattr user_disk, directio, true
```

disk init のその他のオプション・パラメータ

`vstart` は、Adaptive Server がデータベース・デバイスの使用を始める開始仮想アドレス、つまりオフセットです。`vstart` に指定できる単位指定子は、`k` または `K` (キロバイト)、`m` または `M` (メガバイト)、`g` または `G` (ギガバイト)、`t` または `T` (テラバイト) です。オフセットのサイズは、`vstart` の値の入力状態によって異なります。

- 単位を指定しなかった場合、`vstart` は 2K ページをその開始アドレスとして使用します。たとえば、`vstart = 13` と指定すると、Adaptive Server では `13 × 2K` ページが開始アドレスのオフセットとして使用されます。
- 単位の値を指定すると、`vstart` は、入力した値を開始アドレスとして使用します。たとえば、`vstart = "13M"` と指定すると、Adaptive Server では 13MB で開始アドレスのオフセットが設定されます。

`vstart` のデフォルト値 (および通常は優先値) は 0 です。指定したデバイスで、`vstart + size` の合計ブロックが利用できない場合は、`disk init` コマンドは失敗します。

オプションの `cntrltype` キーワードは、ディスク・コントローラを指定します。デフォルト値は 0 です。システム管理者からの指示があった場合にだけ、このオプションを再設定してください。

注意 ディスクの初期化を実行するには、Adaptive Server を起動したユーザが、初期化するデバイスに対する適切なオペレーティング・システム・パーミッションを所有している必要があります。

デバイス情報の表示

`sp_helpdevice` を実行すると、`sysdevices` テーブル内のデバイスに関する情報が表示されます。

デバイス名なしで `sp_helpdevice` を実行すると、Adaptive Server で使用可能なすべてのデバイスがリストされます。デバイス名を指定すると、そのデバイスに関する情報がリストされます。次の例では、`sp_helpdevice` を使用してマスター・デバイスに関する情報を表示します。

```

                                sp_helpdevice master
device_name  physical_name  description
-----
master      d_master              special, default disk, physical disk, 30 MB

status      cntrltype    vdevno    vpn_low    vpn_high
-----
3           0            0         0          10239

```

`master.sysdevices` 内の各ローが表す情報は、次のとおりです。

- データベースのバックアップに使用されるダンプ・デバイス (テープ、ディスク、またはファイル)。
- データベース記憶領域として使用されるデータベース・デバイス。

`sysdevices` の初期内容は、オペレーティング・システムによって異なります。通常、`sysdevices` には次のエントリが含まれます。

- マスタ・デバイスのエントリ (1つ)。
- `sysystemprocs` データベースのエントリ (1つ)。これは `pubs2` や `sysyntax` のような追加のデータベースの保管用や、ユーザ・データベースとログ用に使用できます。
- テープのダンプ・デバイスのエントリ (2つ)。

監査プログラムをインストールした場合には、`sybsecurity` 用の別のデバイスもあります。

`vpn_low` と `vpn_high` の各カラムは、デバイスに割り当てられているページ番号を表します。ダンプ・デバイスの場合、これらのカラムはデバイスのメディア容量を表します。

`status` フィールドは、デバイスのタイプ、ユーザがデータベース・デバイスを指定しないで `create database` コマンドまたは `alter database` コマンドを発行したときにディスク・デバイスがデフォルトの記憶デバイスとして使用されるかどうか、ディスク・ミラーリング情報、および `dsync` 設定値を表します。

表 8-2: `sysdevices` 内のステータス・ビット

ビット	意味
1	デフォルトのディスク (<code>create database</code> コマンドまたは <code>alter database</code> コマンドでロケーションが指定されない場合に使用できる)
2	物理ディスク
4	論理ディスク (使用しない)
8	スキップ・ヘッダ (テープ・ダンプ・デバイスで使用)
16	ダンプ・デバイス
32	逐次書き込み
64	デバイスがミラーリングされている
128	読み込みがミラーリングされている
256	セカンダリ・ミラーリング側のみ
512	ミラーリング使用可能
2048	内部で使用 (<code>disk unmirror, side = retain</code> の後に設定)
4096	ミラーリングを解除する必要があるプライマリ・デバイス (内部で使用)
8192	ミラーリングを解除する必要があるセカンダリ・デバイス (内部で使用)
16384	UNIX ファイル・デバイスは、 <code>dsync</code> 設定を使用する (物理メディアに直接書き込みが行われる)

ダンプ・デバイスと `sp_addumpdevice` の詳細については、『システム管理者ガイド 第2巻』の「第11章 バックアップおよびリカバリ・プランの作成」を参照してください。

デバイスの削除

データベース・デバイスとダンプ・デバイスを削除するには、`sp_dropdevice` を使用します。

```
sp_dropdevice logicalname
```

データベースが使用しているデバイスは削除できません。まず、データベースを削除してください。

`sp_dropdevice` は、`sysdevices` からデバイス名を削除します。`sp_dropdevice` を実行しても、オペレーション・システム・ファイルは削除されるのではなく、Adaptive Server からアクセスできなくなるだけです。`sp_dropdevice` を使用した後でファイルを消去するには、オペレーティング・システムのコマンドを使用してください。

デフォルト・デバイスの指定

Adaptive Server のユーザによるデータベースの作成時に使用されるデフォルト・データベース・デバイスのプールを作成するには、デバイスを初期化した後で `sp_diskdefault` システム・プロシージャを実行します。`sp_diskdefault` を実行すると、`sysdevices` 内で、そのデバイスにデフォルト・デバイスのマークが付けられます。ユーザがデータベース・デバイスを指定せずにデータベースを作成 (または変更) すると、デフォルト・ディスク領域プールから新しいディスク領域が割り付けられます。

`sp_diskdefault` の構文は次のとおりです。

```
sp_diskdefault logicalname, {defaulton | defaultoff}
```

ユーザ・デバイスを追加した後、デフォルト領域のプールからマスタ・デバイスを削除するには、`defaultoff` オプションを使用します。

```
sp_diskdefault master, defaultoff
```

次のコマンドは、`sprocdev` (`sybssystemprocs` データベースが格納されているデバイス) を、デフォルト・デバイスとして指定します。

```
sp_diskdefault sprocdev, defaulton
```

デフォルト・デバイスは複数指定することもできます。その場合、デバイスは `sysdevices` テーブルに表示される順番 (つまりアルファベット順) に使用されます。最初のデフォルト・デバイスがいっぱいになると、2 番目以降のデフォルト・デバイスが順に使用されます。

注意 初期化したデータベース・デバイスを、デフォルト・デバイス・プールではなく、特定のデータベースまたはデータベース・オブジェクトに割り当てることもできます。たとえば、テーブルのサイズが特定のデバイスのサイズを超えないようにする場合です。

デフォルト・デバイスと非デフォルト・デバイスの選択

`sp_diskdefault` を利用すると、ユーザによるデータベースの作成や変更が可能な状態を維持しながら、パフォーマンスとリカバリを考慮した上で領域の使用について計画を立てることができます。

次のデバイスはデフォルト・デバイスとして使用しないでください。

- マスタ・デバイス
- `sybsecurity` 用のデバイス
- ログ専用のデバイス
- 高パフォーマンスのデータベースが常駐するデバイス

`sybssystemprocs` を保持しているデバイスは、他のユーザ・データベース用に使用できません。

注意 ディスク・ミラーリングまたはセグメントを使用している場合、どのデバイスをデフォルトのリストに追加するかを決めるときには、注意が必要です。通常、ミラーリングするデバイスや、セグメント上にオブジェクトが配置されるデータベースには、デフォルト領域の一部ではなく、特定のデバイスを割り付けてください。

disk resize コマンドによるデバイスのサイズ拡大

`disk resize` コマンドを使用すれば、新しいデバイスを初期化することなく、データベース・デバイスのサイズを動的に増やすことができます。たとえば、`/sybase/testdev.dat` に 10MB の追加領域が必要になった場合には、`disk resize` コマンドを実行して、必要な領域をデバイスに割り付けることができます。この追加領域は、`create database` コマンドと `alter database` コマンドで使用できます。

`disk resize` コマンドを使用してサイズを拡大できるのは、ロー・パーティション上のデバイスとファイル・システムのデバイスです。デバイス領域の拡大量として指定できる最小値は、1 MB または 1 アロケーション・ユニットのいずれか大きい方です。

ページ・サイズ	アロケーション・ユニットのサイズ	最小追加サイズ
2K	0.5MB	1MB
4K	1MB	1MB
8K	2MB	2MB
16K	4MB	4MB

ダンプとロードに使用するデバイスには、**disk resize** コマンドは使用できません。

デバイスに設定されたプロパティは、サイズを拡大した後も設定は解除されずに残ります。つまり、デバイスに **dsync** が設定されている場合、このデバイスのサイズを拡大した後も **dsync** は設定されたままになります。また、デバイスのサイズを拡大する前に設定されていたアクセス権限もそのまま残ります。

sa_role を持つユーザが **disk resize** コマンドを実行すると、次の処理が行われます。

- **master...sysdevices** の **high** の値を更新する
- データベース記憶領域の追加領域を作成する

disk resize コマンドの監査証跡を使用すると、デバイスのサイズが変更された回数を追跡できます。サイズ変更対象のデバイスは常にオンラインになり、サイズ変更の処理中でもユーザはデバイスを使用できます。

ディスクのサイズ変更には、次のような要件があります。

- デバイスが **disk init** コマンドで初期化されている。
- **device_name** は、有効な論理デバイス名である。
- サイズ変更操作の実行中はミラーリングを無効にする。サイズ変更操作の完了後に、ミラーリングを再開できます。

この例で、デバイス **testdev** の設定は次のとおりです。

```
sp_helpdevice testdev
device_name  physical_name      description
  status  cntrltype  vdevno      vpn_low      vpn_high
-----  -
testdev     /sybase/dev/testdev.dat  special, dsync on, directio off, physical disk,
10.00MB    16386      0           1           0           5119
```

disk resize コマンドを使用してデバイス **testdev** のサイズを 4MB 増やすには、次のように入力します。

```
disk resize
name = "test_dev",
size = "4M"
```

testdev.dat のサイズは、14MB に拡大されています。

```
sp_helpdevice testdev
device_name  physical_name      description
  status  cntrltype  vdevno      vpn_low      vpn_high
-----  -
testdev     /sybase/dev/testdev.dat  special, dsync on, directio off, physical disk,
14.00MB    16386      0           1           0           7167
```

disk resize の構文については、『リファレンス・マニュアル：コマンド』を参照してください。

ディスク領域の不足

ディスクの物理的な初期化中にディスク領域不足のエラーが発生した場合は、エラーが発生する前の時点での利用可能な最大サイズにデータベース・デバイスが拡張されます。

たとえば、4K の論理ページを使用するサーバでデバイスのサイズを 40MB 増やそうとしたときに、空きディスク領域が 39.5MB しかない場合は、デバイスは 39.5MB 分だけ拡張されます。

disk resize コマンドで、デバイスのサイズの縮小はできません。

トピック名	ページ
sp_dboption プロシージャの使用	311
データベース・オプションの説明	312
データベースの各オプションの表示	313

データベース・オプションは、データベースの動作について次のような制御を行います。

- トランザクションの動作
- テーブル・カラムのデフォルトの設定
- ユーザ・アクセスの制限
- リカバリと bcp 操作の実行
- ログの動作

システム管理者とデータベース所有者は、データベース・オプションを使用して、データベース全体の設定を行うことができます。この点では、データベース・オプションは、サーバ全体に影響する `sp_configure` パラメータや、現在のセッションまたはストアド・プロシージャにだけ影響する `set` オプションとは異なります。

sp_dboption プロシージャの使用

データベース全体の設定内容を変更するには、`sp_dboption` を使用します。オプションは変更されるまで有効なままです。`sp_dboption` プロシージャの機能は、次のとおりです。

- パラメータが指定されないときは、データベース・オプションの全リストを表示する。
- パラメータが指定されたときは、データベース・オプションを変更する。

ユーザ・データベースに対してのみ、オプションを変更できます。`master` データベースのオプションは変更できません。ユーザ・データベース内のデータベース・オプションを変更(または、データベース・オプションのリストを表示)するには、`master` データベースを使用している状態で `sp_dboption` を実行します。

構文は次のとおりです。

```
sp_dboption [dbname, optname, {true | false}]
```

注意 `model` のデータベース・オプションを変更しても、Adaptive Server を再起動したときに、`tempdb`、または現在のユーザが定義した複数のテンポラリ・データベースには影響を与えません。これらの変更は、`model` データベースを変更した後に作成されたデータベースにのみ表示されます。Adaptive Server を再起動すると、テンポラリ・データベースに含まれていたオブジェクトとデータはクリアされますが、データベース・オプションはリセットされません。

データベース・オプションの説明

`master` データベースに対するアクセス権を持つすべてのユーザは、パラメータを付けずに `sp_dboption` を実行すれば、データベース・オプションのリストを表示できます。`sp_dboption` からのレポートは、次のように表示されます。

```
sp_dboption
Settable database options.
-----
abort tran on log full
allow nulls by default
allow wide dol rows
async log service
auto identity
dbo use only
ddl in tran
delayed commit
enforce dump tran sequence
full logging for all
full logging for alter table
full logging for reorg rebuild
full logging for select into
identity in nonunique index
no chkpt on recovery
no free space acctg
read only
select into/bulkcopy/pllsort
single user
trunc log on chkpt
trunc. log on chkpt.
unique auto_identity index
```


特定のデータベースに設定されているオプションに関するレポートを表示する場合は、そのデータベース内で `sp_helpdb` システム・プロシージャを実行します。

各データベース・オプションの詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

データベースの各オプションの表示

特定のデータベースに設定されているオプションを確認するには、`sp_helpdb` を使用します。`sp_helpdb` の出力では、アクティブなオプションが “status” カラム内に表示されます。

次の出力例は、`mydb` データベースでは `read only` オプションが `on` になっていることを示します。

```

                                sp_helpdb mydb
name          db_size  owner  dbid  created          status
-----
mydb          20.0 MB  sa     5     Mar 05, 2005    read only

device_fragments      size      usage              created          free kbytes
-----
master                10.0 MB  data and log      Mar 05 2005    1792

device                segment
-----
master                default
master                logsegment
master                system

```

すべてのデータベースのオプションをまとめて表示するには、データベースを指定しないで `sp_helpdb` を発行します。

```

                                sp_helpdb
name          db_size  owner  dbid  created          status
-----
master        48.0 MB  sa     1     Apr 12, 2005    mixed log and data
model         8.0 MB  sa     3     Apr 12, 2005    mixed log and data
pubs2         20.0 MB  sa     6     Apr 12, 2005    select into/
               bulkcopy/pllsort, trunc log on chkpt, mixed log and data
sybssystemdb   8.0 MB  sa     5     Apr 12, 2005    mixed log and data
sybssystemprocs 112.0 MB  sa     4     Apr 12, 2005    trunc log on chkpt,
               mixed log and data
tempdb        8.0 MB  sa     2     Apr 12, 2005    select into/
               bulkcopy/pllsort, trunc log on chkpt, mixed log and data

```

sysoptions による現在設定されているスイッチの表示

`sysoptions.number` カラムには、現在設定されているスイッチのスイッチ ID が含まれています。テーブルが追加されています。`sysoptions` で以下のスイッチが表示されます。

- `runserver` ファイルに `-T` フラグが設定されたトレース・フラグ
- `dbcc traceon(flag_number)` または `set switch serverwide on` が設定されたトレース・フラグ
- 特定のシステム・プロセス ID (`spid`) に対して `set switch on` が設定されたトレース・フラグおよびスイッチ

`sysoptions` には、ユーザが参照可能なスイッチのみが表示されます。ユーザは他の `spid` によりプライベートに設定されているスイッチを表示できません。`number` の値は、スイッチ以外のすべてのオプション・カテゴリに対して `Null` です。

`sysoptions` はスイッチの以下の情報を表示します。

- `spid` – 現在のセッションの `spid`。
- `name` – スイッチの名前。名前のないスイッチが設定されると、`name` には文字列に変換されたスイッチ番号が入ります。
- `category` – 文字列 `Switch` を指定します。
- `currentsetting` – スイッチが設定されている場合は 1、スイッチが設定されていない場合は 0 が設定されます。
- `defaultsetting` – 0 が設定されます。
- `scope` – スイッチ設定がサーバワイドかそのセッションのみかを示します。値は次のとおりです。
 - 0 – スイッチが設定されていない。
 - 8 – スイッチがサーバワイドで設定されている。
 - 16 – スイッチは現在の `spid` でプライベートに設定されている。
 - 24 – スイッチがサーバワイドでプライベートに設定されている。
- `number` – スイッチ ID が整数として含まれます。

次のクエリで、スイッチのすべての `sysoptions` エントリを表示します。

```
select * from sysoptions
where category = 'Switch'
```

次のクエリで、現在のセッションのスイッチを表示します。

```
select * from sysoptions
where category = 'Switch'
and currentsetting = '1'
```

次はトレース・フラグ 3604 を設定した後、**sysoptions** 出力を表示します。

spid	name	scope	category	currentsetting	defa
ultsetting			number		
11	print_output_to_client	0	8	Switch	1
					3604

この章では、Adaptive Server の国際化とローカライゼーション・サポートについて説明します。

トピック名	ページ
国際化とローカライゼーションの概要	317
国際化されたシステムの利点	318
サンプル国際化システム	319
国際化システムの要素	321
サーバの文字セットの選択	321
ソート順の選択	330
システム・メッセージ用言語の選択	338
サーバの設定：例	339
文字セット、ソート順、メッセージ言語の変更	341
サポートされていない言語の日付文字列のインストール	352
国際化ファイルとローカライゼーション・ファイル	354

国際化とローカライゼーションの概要

「国際化」とは、アプリケーションを複数の言語や文化的慣例に対応させることをいいます。

国際化されたアプリケーションでは、実行時に外部ファイルを使用して、言語固有の情報を表示します。このようなアプリケーションは、言語固有のコードを含んでいないので、コードに変更を加えることなくどのようなネイティブ言語の環境にも配備できます。ソフトウェア製品の 1 つのバージョンを複数の言語や地域に適応させることができ、設計変更を行わなくても各地域の要件や習慣に適合させることが可能です。このようなソフトウェア開発のアプローチは、アプリケーションのライフタイム全体を通して大幅な時間と費用の節約につながります。

「ローカライゼーション」とは、国際化された製品を特定の言語 (たとえばスペイン語) または地域の要件に合うように適応させることで、これには、その国の言語に翻訳されたシステム・メッセージ、ユーザ・インタフェースの翻訳版、その国で使用されている正しいフォーマットでの日付、時間、通貨の表示が含まれます。ソフトウェア製品の 1 つのバージョンについて、いくつものローカライズ版を作成することができます。

Sybase は、国際化とローカライゼーションの両方をサポートしています。Adaptive Server には、西欧、東欧、中東、ラテン・アメリカ、アジアにおける主な商用語の、データ処理サポートに必要な文字セット定義ファイルとソート順定義ファイルがあります。

Sybase 言語モジュールは、中国語 (簡体字)、フランス語、ドイツ語、日本語、韓国語、ブラジルで使用するポルトガル語、スペイン語の各言語に翻訳されたシステム・メッセージとフォーマットを提供します。デフォルトでは、Adaptive Server には英語のメッセージ・ファイルが付属しています。

この章では、文字セットと言語モジュールについて簡単に説明するとともに、Adaptive Server のデフォルトの文字セット、ソート順、またはメッセージ言語を変更するために必要な手順を説明します。

国際化されたシステムの利点

アプリケーションを他国でも使用できるように設計することは、きわめて面倒な仕事に思えます。プログラマたちは、国際化とは各国の文化や言語上の慣習によって必要となる部分を個々にハードコードすることだと考えがちです。

しかし、もっと良いアプローチがあります。それは、国際化されたアプリケーション、つまり実行時にローカル・コンピューティング環境を調べて使用する言語を決定し、その言語に関する情報が記述されているファイルを読み込むアプリケーションを作成することです。

国際化されたアプリケーションであれば、同じアプリケーションをどの国でも使用できます。このアプローチには次のような利点があります。

- アプリケーションを 1 つだけ作成すればよく、各国語版を個々に作成する必要はありません。
- アプリケーションの提供先の国が増えたときも、アプリケーションに変更を加える必要はありません。その国のローカライゼーション・ファイルを添付するだけで済みます。
- すべてのサイトで機能と動作を統一できます。

サンプル国際化システム

国際化されたシステムでは、国際化されたクライアント・アプリケーション、ゲートウェイ、サーバをさまざまなネイティブ言語環境の複数のプラットフォームで実行できます。

たとえば、次のようなコンポーネントで構成される国際化システムもあります。

- ニューヨーク、メキシコシティ、パリにある発注処理アプリケーション (Client-Library アプリケーション)
- ドイツにある在庫管理サーバ (Adaptive Server)
- フランスにある発注遂行アプリケーション (Adaptive Server)
- 日本にある中央会計アプリケーション (Adaptive Server と相互稼働する Open Server アプリケーション)

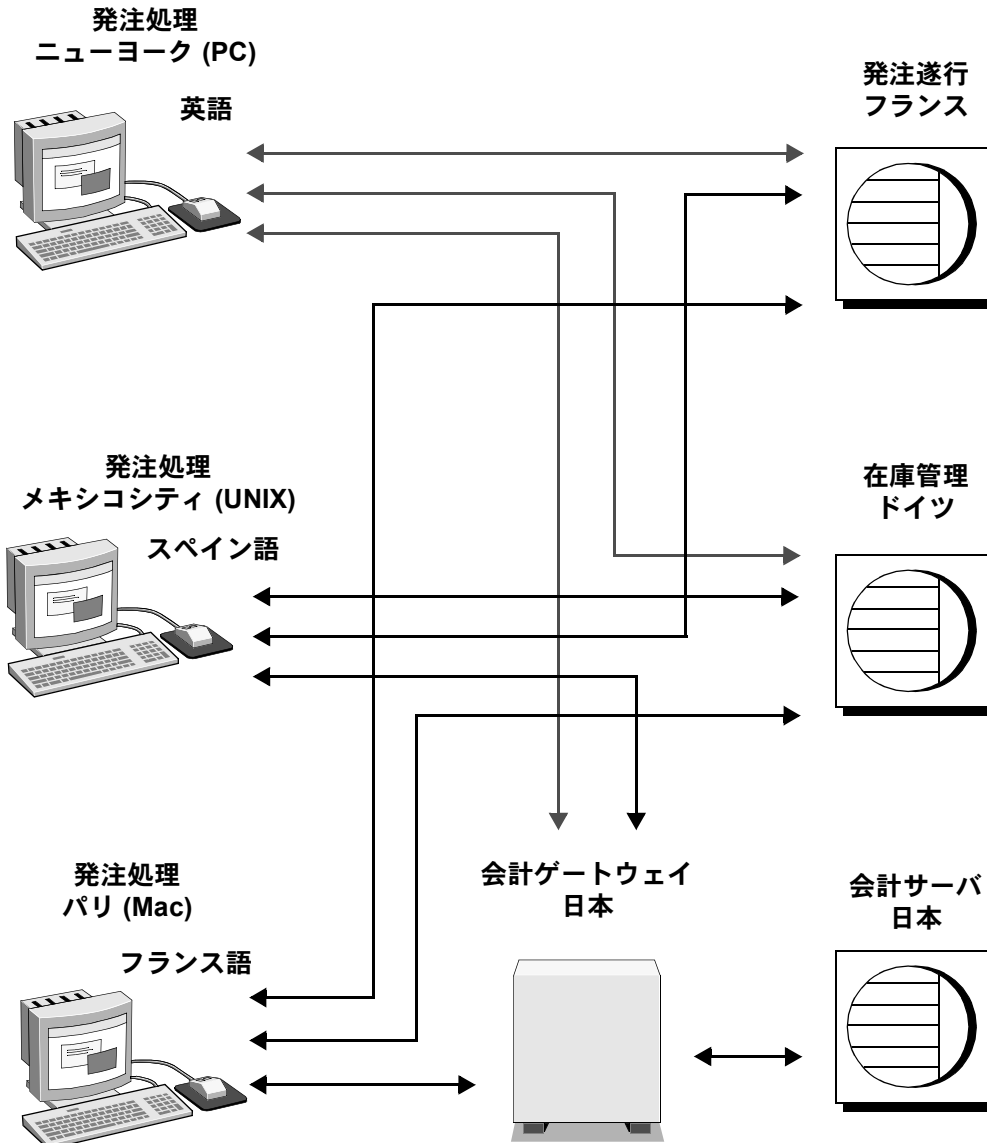
このシステムの発注処理アプリケーションは、以下の処理を実行します。

- 在庫管理サーバに対してクエリを発行し、注文された商品の在庫があるかどうかを調べる。
- 発注遂行サーバに発注データを送る。
- 財務情報を会計アプリケーションに送る。

在庫管理サーバと発注遂行サーバはクエリに応答し、会計アプリケーションは財務データを収集してレポートを作成します。

このシステムを図で表すと次のようになります。

図 10-1: 国際化システムの例



この例では、すべてのアプリケーションとサーバがそれぞれの国の言語と文字セットを使用して、入力データを受け取り、メッセージを出力します。

国際化システムの要素

国際化環境のサーバ言語を設定するときには、以下に挙げる 3 つの要素について操作が可能です。これらの要素について検討したうえで、クライアント／サーバ・ネットワークの構築計画を入念に作成してください。

- 文字セット – サーバがクライアント・サーバとの間でデータを送受信するときに使用する言語。すべてのクライアント・サーバについて言語上の必要事項を慎重に計画、分析したうえで、文字セットを選択してください。
- ソート順 – ソート順オプションは選択した言語および文字セットによって異なります。
- システム・メッセージ – Sybase が提供する言語のいずれか 1 つでメッセージが表示されます。提供された言語の中にサーバ言語がない場合は、システム・メッセージはデフォルト言語の英語で表示されます。

以下の項では、これらの各要素について詳しく説明します。

サーバの文字セットの選択

サーバ内では、データは特別なコードでコード化されます。たとえば、“a” という文字は 10 進数の “97” にコード化されます。「文字セット」とは、特定の文字集合（アルファベット、数字、記号、非表示の制御文字を含む）と、それらが割り当てられた数値、つまりコードのことです。文字セットには、一般にその言語の全字母用の文字が含まれます。たとえば、英語で使用されるラテン・アルファベット、またはロシア語、セルビア語、ブルガリア語などの言語で使用されるキリル文字のような書体です。言語のサブセット（たとえば西欧言語）をサポートする、プラットフォーム固有の文字セットを「ネイティブ文字セット」といいます。Adaptive Server に付属する文字セットは、Unicode UTF-8 を除いてすべてネイティブの文字セットです。

「スクリプト」とは、書記体系のことで、たとえばラテン語、日本語、アラビア語などの言語を書き表すときにその言語を特徴付けるすべての要素の集合です。1 つのアルファベットまたはスクリプトでサポートされる言語に応じて、1 つの文字セットは 1 つ以上の言語をサポートします。たとえば、ラテン・アルファベットは西欧言語をサポートします（表 10-1 (322 ページ) のグループ 1 を参照）。それに対し、日本語というスクリプトがサポートするのは 1 つの言語（日本語）だけです。このように、グループ 1 の文字セットはさまざまな言語をサポートしますが、多くの文字セット（グループ 101 など）は 1 つの言語しかサポートしません。

1 つの文字セットによって処理される 1 つ以上の言語を、「言語グループ」と呼びます。言語グループには、複数の言語が属することも 1 つの言語しか属さないこともあります。ネイティブ文字セットとは、特定の言語グループに属する言語の文字の、プラットフォーム固有のコード化です。

クライアント/サーバ・ネットワークでは、複数の言語でのデータ処理をサポートできます。ただし、それらの言語はすべて同じ言語グループに属していなければなりません(表 10-1 (322 ページ) を参照)。たとえば、サーバ内のデータをグループ 1 の文字セットでコード化する場合は、フランス語、ドイツ語、イタリア語など、グループ 1 に属する言語のデータであれば同じデータベースに格納できます。ただし、異なる言語グループに属する言語のデータを同じデータベースに格納することはできません。たとえば、日本語のデータはフランス語やドイツ語のデータとは一緒に保存できません。

前述のネイティブ文字セットと異なり、「Unicode」は国際文字セットで、日本語、中国語、ロシア語、フランス語、ドイツ語など世界中の 650 以上もの言語をサポートします。Unicode を使用すると、プラットフォームに関係なく、同じサーバ内で多数の言語グループのさまざまな言語を一緒に使用できます。詳細については、「Unicode」(324 ページ) を参照してください。

すべての文字セットがラテン・スクリプトと英語をサポートするため、それぞれの文字セットは少なくとも 2 つの言語 (英語ともう 1 つ別の言語) をサポートします。

多くの言語は複数の文字セットによってサポートされます。ある言語に対してどの文字セットをインストールするかは、クライアントのプラットフォームとオペレーティング・システムに応じて決定します。

Adaptive Server は次の言語と文字セットをサポートします。

表 10-1: サポートされる言語と文字セット

言語グループ	言語	文字セット
グループ 1	西欧 : アルバニア語、カタロニア語、デンマーク語、オランダ語、英語、フェロー語、フィンランド語、フランス語、ガリシア語、ドイツ語、アイスランド語、アイルランド語、イタリア語、ノルウェー語、ポルトガル語、スペイン語、スウェーデン語	ASCII 8、CP 437、CP 850、CP 860、CP 863、CP 1252 ^a 、ISO 8859-1、ISO 8859-15、Macintosh Roman、ROMAN8、ROMAN9、ISO-15、CP 858
グループ 2	東欧 : クロアチア語、チェコ語、エストニア語、ハンガリー語、ラトヴィア語、リトアニア語、ポーランド語、ルーマニア語、スロヴァキア語、スロヴェニア語 (および英語)	CP 852、CP 1250、ISO 8859-2、Macintosh Central European
グループ 4	バルト語 (および英語)	CP 1257
グループ 5	キリル : ブルガリア語、ベラルーシ語、マケドニア語、ロシア語、セルビア語、ウクライナ語 (および英語)	CP 855、CP 866、CP 1251、ISO 8859-5、Koi8、Macintosh Cyrillic
グループ 6	アラビア語 (および英語)	CP 864、CP 1256、ISO 8859-6
グループ 7	ギリシャ語 (および英語)	CP 869、CP 1253、GREEK8、ISO 8859-7、Macintosh Greek
グループ 8	ヘブライ語 (および英語)	CP 1255、ISO 8859-8
グループ 9	トルコ語 (および英語)	CP 857、CP 1254、ISO 8859-9、Macintosh Turkish、TURKISH8
グループ 101	日本語 (および英語)	CP 932 DEC Kanji、EUC-JIS、Shift-JIS
グループ 102	簡体字中国語 (PRC) (および英語)	CP 936、EUC-GB、GB18030

言語グループ	言語	文字セット
グループ 103	繁体字中国語 (ROC) (および英語)	Big 5、CP 950 ^b 、EUC-CNS、Big 5 HKSCS
グループ 104	韓国語 (および英語)	EUC-KSC、CP949
グループ 105	タイ語 (および英語)	CP 874、TIS 620
グループ 106	ベトナム語 (および英語)	CP 1258
Unicode	650 以上の言語	UTF-8

a. CP 1252 は、0x80 ~ 0x9F のコード・ポイントが CP 1252 内の文字にマップされることを除き、ISO 8859-1 と同一。
b. CP 950 は Big 5 と同一。

注意 すべての文字セットが英語をサポートするのは、どの文字セットにも最初の 128 (10 進数) 文字にラテン・アルファベット (“ASCII-7” として定義される) が含まれているからです。先頭の 128 文字より後の文字は文字セットによって異なり、別のネイティブ言語の文字をサポートするために使用されます。たとえば、CP 932 と CP 874 のどちらも、コード・ポイント 0 ~ 127 は英語とラテン・アルファベットをサポートしますが、CP 932 のコード・ポイント 128 ~ 255 は日本語文字をサポートし、CP 874 のコード・ポイント 128 ~ 255 はタイ語の文字をサポートします。

欧州通貨記号「ユーロ」をサポートする文字セットは、CP 1252 (西欧)、CP 1250 (東欧)、CP 1251 (キリル)、CP 1256 (アラビア語)、CP 1253 (ギリシャ語)、CP 1255 (ヘブライ語)、CP 1254 (トルコ語)、CP 874 (タイ語)、iso15、roman9、CP858 です。Unicode UTF-8 は、次の言語もサポートします。

- 繁体字中国語 (Windows および Solaris プラットフォーム)
- アラビア語、ヘブライ語、タイ語、ロシア語 (Linux プラットフォーム)

注意 iso_1 と ISO 8859-1 は名前が異なるだけで、同じ文字セットです。

異なる言語グループの言語を混在させるためには、Unicode を使用する必要があります。サーバの文字セットが Unicode であれば、650 を超える数の言語を 1 つのサーバでサポートでき、あらゆる言語グループの言語を混在させることができます。

Unicode

Unicode は、世界のすべての言語を同じデータ・セット内でコード化することを可能にした初めての文字セットです。Unicode が導入される前は、たとえば中国語でデータを保存するには、この言語に適した文字セットを選択しなければならず、他のほとんどの言語は排除されていました。同じデータ・セット内で複数の文字セットを混在させること、つまり、さまざまな言語を混在させることは不可能でした。

Sybase は `unichar`、`univarchar`、`unitext` の 3 つのデータ型で Unicode をサポートします。これらのデータ型は、Unicode UTF-16 でデータをコード化して保存します。

UTF-16 は、Unicode スカラ値を単一の 16 ビット値で表すコード化です (まれに、16 ビット値のペアで表すこともあります)。この 3 つのコード化は、あらゆる Unicode 文字を表すことができるという点では同じです。サーバのデフォルト文字セットとしての UTF-16 ではなく、データ型としての UTF-16 が選択されたことにより、既存のデータベース・アプリケーションを容易に、段階的に移行できます。

Adaptive Server は、SQL クエリでの Unicode リテラルと、UTF-8 のさまざまなソート順をサポートしています。

Adaptive Server が使用する文字セット・モデルは、サーバ全体で使用される、単一の設定可能な文字セットに基づいています。「文字」データ型 (`char`、`varchar`、`nchar`、`nvarchar`、`text`) のいずれかを使用して Adaptive Server に保存されるデータはすべて、この文字セットで記述されているものと見なされます。ソート順は、各国語に変換されたサーバ・メッセージの集合である言語モジュールと同様に、この文字セットを使用して定義されます。

クライアント・アプリケーションは、接続ダイアログの中で自身のネイティブ文字セットと言語を宣言します。設定が正しく行われていれば、サーバは、サーバの文字セットとクライアントの文字セットの間でそれ以降のすべての文字データの変換を試みます (文字データには、データベースに保存されるすべてのデータのほかに、クライアントのネイティブ言語でのサーバ・メッセージも含まれます)。このことが機能するには、サーバの文字セットとクライアントの文字セットが互に変換可能であることが条件です。文字が他方の文字セットで定義されていない場合は、正しく機能しません。たとえば、日本語に使用される文字セットである SJIS と、ロシア語などのキリル文字言語に使用される KOI8 の場合です。このように変換不可能なことがある場合に、Unicode を使用します。Unicode は、他のすべての文字セットの文字の定義が含まれる、文字のスーパーセットと考えることができます。

Unicode データ型である `unichar`、`univarchar`、および `unitext` は、従来の文字セット・モデルからは完全に独立しています。クライアントは、その他の文字データの送受信とは無関係に、Unicode データを送受信します。

文字セットのインストール

Adaptive Server バージョン 12.5.1 以降では、4 バイト形式の UTF-8 がサポートされています。この形式は、16 ビット値のペア (「サロゲート・ペア」) として UTF-16 で表される、使用頻度の低い Unicode 文字を表すのに使用されます。バージョン 12.5.1 より前の Adaptive Server では、3 バイト形式の UTF-8 だけがサポートされていました。バージョン 12.5.1 より前の Adaptive Server に UTF-8 文字セットがインストールされている場合は、4 バイト形式の UTF-8 を使用できるように文字セットを再インストールする必要があります。

設定パラメータ

Unicode の UTF-16 コード化には、使用頻度の低い文字を表すための 16 ビット値のペア (「サロゲート・ペア」) が含まれています。Adaptive Server には、サロゲート・ペアの整合性を保証するための機能が組み込まれています。このチェック機能をオフにするには設定パラメータ “enable surrogate processing” を 0 に設定します。このようにすると、パフォーマンスはやや向上しますが、サロゲート・ペアの整合性は保証されなくなります。

Unicode では、「正規化」も定義されています。正規化とは、単一の文字について可能なすべての表現を、単一の表現に変換するプロセスです。基本文字とそれに続く分音記号の組み合わせには、多くの場合、等価な事前結合済み文字がありますが、両者のビット・パターンは異なっています。たとえば、次の 2 つのシーケンスは等価です。

```
0x00E9 -- é (LATIN SMALL LETTER E WITH ACUTE)
0x00650301 -- e (LATIN SMALL LETTER E), ´ (COMBINING ACUTE ACCENT)
```

enable unicode normalization 設定パラメータは、Adaptive Server が受け取った Unicode データを正規化するかどうかを制御します。

default Unicode sortorder を “binary” に設定し、**enable Unicode normalization** 設定パラメータを 1 に設定すると、パフォーマンスが飛躍的に向上します。この組み合わせのとき、Adaptive Server は Unicode データの性質についていくつかの仮定を行いますが、その仮定を利用するようにコードが実装されています。

関数

char 型のパラメータを受け取る関数は、すべて **unichar** 型も受け入れるようにオーバーロードされています。複数のパラメータを持つ関数が呼び出されたときに、指定されたパラメータのいずれかが **unichar** 型であるときは、**unichar** 型以外のパラメータはすべて **unichar** 型に暗黙的に変換されます。

enable surrogate processing が 1 (デフォルト) に設定されているとき、サロゲート・ペアの整合性を保証するために、文字列関数ではサロゲート・ペアを分割するような処理は回避されます。位置は、サロゲート・ペアの開始位置を指すように変更されます。

`unicar` 型のサポートを補完するために関数が追加されています。追加された関数の中に、`to_unichar()` と `uscalar()` があります。これらの機能は、`char()` と `ascii()` に似ています。関数 `uhighsurr()` と `ulowsurr()` を利用すると、ユーザ・コードでサロゲート・ペアを明示的に扱うことができます。

関数で `unitext` を使用する場合には制限があります。詳細については、各関数の「使用法」のセクションにある制限の説明を参照してください。

unicar カラムの使用

`isql` ユーティリティまたは `bcp` ユーティリティを使用するとき、Unicode 値は 16 進形式で表示されます。ただし、`Jutf8` フラグを使用してクライアントの文字セットが UTF-8 であることを指定した場合を除きます。この場合、サーバから受信した Unicode データはすべて UTF-8 に変換されます。次に例を示します。

```
% isql -Usa -P -Jiso_1
1> select unicode_name from people where unicode_name = 'Jones'
2> go

unicode_name
-----|
0x004a006f006e00650073
(1 row affected)
```

これが、次のようになります。

```
% isql -Usa -P -Jutf8
1> select unicode_name from people where unicode_name = 'Jones'
2> go

unicode_name
-----
Jones
(1 row affected)
```

これによって、アドホック・クエリが実行しやすくなります。端末ウィンドウによってはすべての Unicode 文字を表示できないものもありますが、ASCII 文字を扱う単純なテストは非常に簡単になります。

unitext の使用

可変長の `unitext` データ型は、Unicode 文字で最大 1,073,741,823 文字 (2,147,483,646 バイト) まで保持できます。`unitext` は、`text` データ型を使用できる場所であれば、同じセマンティックで使用できます。`unitext` カラムは、Adaptive Server のデフォルト文字セットとは関係なく、UTF-16 エンコーディングで保管されます。

Open Client の相互運用性

Open Client ライブラリでデータ型 `cs_unichar` がサポートされるようになりました。このデータ型は、短い整数の配列として宣言されたユーザ変数にバインドできます。Open Client のこのデータ型は、サーバの `unichar` 型、`unitext` 型、および `univarchar` 型への直接のインタフェースとなります。

Java の相互運用性

内部 JDBC ドライバにより、SQL コンテキストと Java コンテキストの間で `unichar` データが効率的に転送されます。

SQL から Java への場合は、クラス `java.sql.ResultSet` の多数の “get” メソッドを使用して、結果セットのカラムからデータを取り出します。この get メソッドは、`unichar`、`unitext`、または `univarchar` として定義されたカラムに対して機能します。メソッド `getString()` は、変換を実行する必要がないので、特に効率的といえます。

Java から SQL への場合は、クラス `java.sql.PreparedStatement` の `setString()` メソッドを使用します。内部 JDBC ドライバは、`unichar`、`unitext`、または `univarchar` として定義された SQL パラメータに Java の文字列データを直接コピーします。

外部 JDBC ドライバ (`jConnect`) は、内部ドライバと同様のシームレスなインタフェースをサポートするように変更されました。

制限事項

Adaptive Server の以前のリリースには Unicode ベースの言語パーサがなく、新しい Unicode データ型の使用には制限がありました。新しいデータ型を使用するにはサーバのデフォルト文字セットを UTF-8 として設定する必要がありましたが、この制限は、Adaptive Server バージョン 12.5.1 以降では取り除かれ、サーバのデフォルト文字セットと関係なく Unicode のデータ型を使用できるようになりました。

サーバのデフォルト文字セットの選択

サーバを設定するときに、デフォルト文字セットを指定する必要があります。デフォルトの文字セットとは、サーバがデータを格納および操作するときに使用する文字セットのことです。デフォルト文字セットは各サーバに 1 つだけ設定できます。

デフォルトでは、インストール・ツールはオペレーティング・システムのネイティブ文字セットをサーバのデフォルト文字セットと見なします。ただし、Adaptive Server がサポートする文字セットであれば、サーバのデフォルト文字セットとして選択できます (表 10-1 (322 ページ) を参照)。

たとえば、AIX が稼働する IBM RS/6000 にサーバをインストールする場合に、西欧言語の 1 つを選択すると、ISO 8859-1 がデフォルト文字セットと見なされます。

Unicode サーバをインストールする場合は、デフォルト文字セットとして UTF-8 を選択します。

Unicode サーバ以外の場合は、クライアント・システムの大部分が使用するプラットフォームを特定し、そのプラットフォームの文字セットをサーバのデフォルト文字セットとして使用します。

これには次に挙げる 2 つの利点があります。

- 文字セット間でマッピング不可能な文字を最小限にできる。

通常は、2 つの文字セット間で文字が完全に一対一で対応することはないため、データの消失が起こる可能性があります。このことは通常はごく小さな問題でしかありません。変換されない文字のほとんどは、プラットフォームに固有であるか、あまり使用されない特殊文字だからです。

- 必要な文字セット変換を最小限にできる。

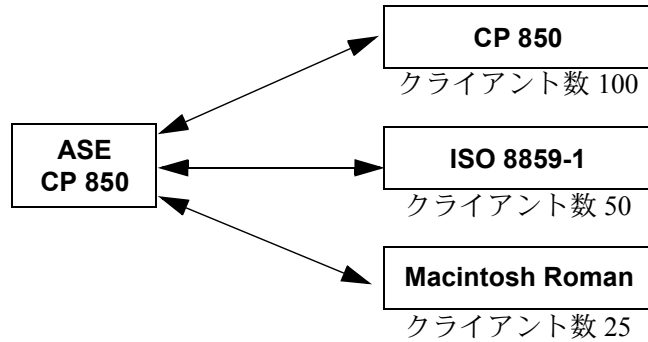
クライアント・システムの文字セットとサーバのデフォルト文字セットが異なる場合は、データの整合性を保つためにデータを変換する必要があります。文字セット変換によるパフォーマンスの低下は大きくありませんが、できるだけ変換が少なくすむデフォルト文字セットを選択するのが賢明です。

たとえば、ほとんどのクライアントが CP 850 を使用している場合は、サーバで CP 850 を指定します。これは、サーバを HP-UX システム (グループ 1 の言語のネイティブ文字セットが ROMAN8) にインストールする場合でも同じです。

注意 データベースを作成したり、Sybase 提供のデータベースに変更を加えたりする前に、デフォルトとして使用する文字セットを決定するよう強くおすすめします。

次の例 (図 10-2) では、175 のクライアントすべてが同じ Adaptive Server にアクセスします。これらのクライアントのプラットフォームも使用している文字セットも 1 つではありません。これらのクライアントが相互稼働するためには、クライアント/サーバ・システム内のすべての文字セットが同じ言語グループ (表 10-1 (322 ページ) を参照) に属している必要があります。Adaptive Server のデフォルト言語は CP 850 であり、最も多くのクライアントがこの文字セットを使用しています。したがって、文字セット変換の量が最小になり、サーバの動作効率が最大になります。

図 10-2: 同じ言語グループ内の異なる文字セットを使用するクライアント



サーバのデフォルト文字セットを選択するときには、一般的に使用されている文字セットをプラットフォームおよび言語別にまとめた次の表を参考にしてください。

表 10-2: 一般的な西欧クライアント・プラットフォーム

プラットフォーム	言語	文字セット
Win 95、98	米語、西欧語	CP 1252
Win NT 4.0	米語、西欧語	CP 1252
Win 2000	米語、西欧語	CP 1252
Sun Solaris	米語、西欧語	ISO 8859-1
HP-UX 10、11	米語、西欧語	ROMAN8
IBM AIX 4.x	米語、西欧語	ISO 8859-1

表 10-3: 一般的な日本語クライアント・プラットフォーム

プラットフォーム	言語	文字セット
Win 95、98	日本語	CP 932 for Windows
Win NT 4.0	日本語	CP 932 for Windows
Win 2000	日本語	CP 932 for Windows
Sun Solaris	日本語	EUC-JIS
HP-UX 10、11	日本語	EUC-JIS
IBM AIX 4.x	日本語	EUC-JIS

表 10-4: 一般的な中国語クライアント・プラットフォーム

プラットフォーム	言語	文字セット
Win 95、98	中国語 (簡体字)	CP 936 for Windows
Win NT 4.0	中国語 (簡体字)	CP 936 for Windows
Win 2000	中国語 (簡体字)	CP 936 for Windows
Sun Solaris	中国語 (簡体字)	EUC-GB
HP-UX 10、11	中国語 (簡体字)	EUC-GBS
IBM AIX 4.x	中国語 (簡体字)	EUC-GB

ソート順の選択

文字が同じでも言語によってソート順が異なります。たとえば、英語では *Cho* の位置が *Co* より前になりますが、スペイン語では逆になります。ドイツ語では、 β は 1 つの文字ですが、辞書では二重文字 *ss* として扱われ、ソートされます。アクセント記号付き文字は特別な順序でソートされます。たとえば、*aménité* の位置は *amène* より前となり、アクセント記号がない場合とは逆になります。したがって、文字を正しくソートするためには、言語に固有のソート順が必要です。

各文字セットには、Adaptive Server がデータの並べ替えに使用するソート順が 1 つ以上定義されています。ソート順は、特定の言語または言語グループと特定の文字セットに関連付けられています。英語、フランス語、ドイツ語では、同じソート順を使用できます。これらの言語では、*A*、*a*、*B*、*b* のように、同じ文字は同じ順序でソートされるからです。特定の言語に固有の文字もあります。たとえば、アクセント記号付き文字 *é*、*à*、および *á* はフランス語では使用されますが、英語やドイツ語では使用されません。そのため、これらの文字のソート順に関して矛盾は発生しません。ただし、これはスペイン語には当てはまりません。二重文字である *ch* と *ll* のソート方法が異なります。したがって、これらの 4 言語すべてを同じ文字セットがサポートしますが、英語、フランス語、ドイツ語で使用されるソート順とスペイン語で使用されるソート順は異なります。

また、ソート順は特定の文字セットに関連付けられます。そのため、英語、フランス語、ドイツ語用のソート順は ISO 8859-1 に 1 セットあり、CP 850 にも 1 セットあります。特定の文字セットで使用できるソート順は、その文字セット・ディレクトリのソート順定義ファイル (*.srt) に定義されています。文字セットと、その文字セットで利用できるソート順のリストについては、[表 10-5 \(333 ページ\)](#) を参照してください。

ソート順の使用

ソート順は以下のときに使用されます。

- インデックスの作成
- インデックス付きテーブルへのデータ格納
- `order by` 句の指定

ソート順の種類

すべての文字セットで、バイナリ・ソート順は必ず使用できます。このソート順では、文字セットの各文字に割り当てられたコード（「バイナリ」コード）の算術値だけに基づいてデータがソートされます。バイナリ・ソート順が適しているのは、各文字セットの最初の 128 文字 (ASCII 英語) とアジア系言語です。バイナリ・ソート順によって正しくソートされない可能性が最も高いのは、複数の言語をサポートする文字コード (グループ 1 や Unicode など) の場合です。このような文字セットを使用する場合は、別のソート順を選択してください。

文字セットによっては、以下に挙げる辞書ソート順も利用できます。

- 辞書順、大文字と小文字、およびアクセント記号を区別する — 大文字と小文字を異なる文字としてソートします。各種のアクセント記号付き文字を認識し、対応するアクセント記号なしの文字より後になるようにソートします。

- 辞書順、大文字と小文字を区別しない、アクセント記号を区別する – データを辞書順でソートしますが、大文字と小文字を異なる文字として認識しません。大文字と小文字は等しく処理され、ソート結果では両者が混在します。名前のテーブルでエントリの重複を避けたい場合に便利です。
- 辞書順、大文字と小文字を区別しない、アクセント記号を区別する、優先度を付けた順位 – 項目の同等性を評価する際に大文字と小文字の違いを認識しません。単語が大文字でも小文字でも同じ単語と見なされます。他の条件がすべて等しい場合は、大文字が優先されます (位置が先になる)。
優先度付きで大文字と小文字を区別しないソート順を使用すると、**order by** 句で指定されたカラムがテーブルのクラスタード・インデックスのキーと一致した場合に、大きなテーブルでのパフォーマンスが低下することがあります。**order by** 句で、大文字と小文字の区別だけが異なる文字列を、大文字が先、小文字が後になるようにソートする必要がある場合を除いて、このソート順は選択しないでください。
- 辞書順、大文字と小文字、およびアクセント記号を区別しない – アクセント記号付き文字を、対応するアクセント記号なしの文字と同じものとして扱います。ソート結果では、アクセント記号付き文字と対応するアクセント記号なしの文字が混在します。

デフォルト・ソート順の選択

Sybase サーバでサポートされるデフォルト・ソート順は一度に 1 つだけです。すべてのユーザが同じ言語を使用している場合、またはユーザの言語がすべて同じソート順を使用する場合は、そのソート順を選択してください。たとえば、ユーザがフランス語のデータを扱っていて、フランス語のソート順を望んでいる場合は、フランス語の辞書ソート順からいずれかを選択します。ユーザが複数言語のデータを扱っていて、それらの言語が同じソート順を使用する場合は (たとえば、英語、フランス語、ドイツ語の場合など)、そのソート順を選択すれば、どの言語を使用するユーザにも対応できます。

ただし、ユーザが複数の言語を使用しており、それらの言語が異なるソート順を必要とする (たとえば、フランス語とスペイン語) 場合は、それらのソート順のどちらかをデフォルトとして選択しなければなりません。たとえば、フランス語のソート順を選択すると、スペイン語のユーザにとっては、二重文字の *ch* と *ll* が正しい順序でソートされません。インストール時には、デフォルトでサーバのソート順としてバイナリ・ソート順が設定されます。

sortkey 関数を使用して、言語ごとに代替ソート順を設定することができます。このソート順は、各ユーザが使用する言語に合わせて動的に選択できます。**sortkey** 関数はデフォルト・ソート順とは関連しておらず、両方が同じサーバに共存できます。**sortkey** 関数で設定するソート順は、範囲と詳細さの点でデフォルト・ソート順メカニズムが提供するソート順より優れています。詳細については、『リファレンス・マニュアル:ビルディング・ブロック』の「**sortkey**」と「**compare**」を参照してください。

表 10-5: 利用できるソート順

言語またはスクリプト	文字セット	ソート順
すべての言語	UTF-8	複数文字コードのソート順については、 表 10-7 を参照
キリル :ブルガリア語、ベラルーシ語、マケドニア語、ロシア語、セルビア語、ウクライナ語	CP 855、CP 866、CP 1251、ISO 8859-5、Koi8、Macintosh Cyrillic	辞書順 (大文字と小文字、およびアクセント記号を区別する)
東欧 :チェコ語、スロバキア語	CP 852、ISO 8859-2、CP 1250	辞書順 (大文字と小文字、およびアクセント記号を区別する) 辞書順 (大文字と小文字を区別しない、アクセント記号を区別する) 辞書順 (大文字と小文字、およびアクセント記号を区別する、優先度を付けた順位) 辞書順 (大文字と小文字、およびアクセント記号を区別しない)
英語、フランス語、ドイツ語	ASCII 8、CP 437、CP850、CP 860、CP 863、CP 1252a、ISO 8859-1、ISO 8859-15、Macintosh Roman、ROMAN8、ROMAN9、ISO 15	辞書順 (大文字と小文字、およびアクセント記号を区別する) 辞書順 (大文字と小文字を区別しない、アクセント記号を区別する) 辞書順 (大文字と小文字、およびアクセント記号を区別する、優先度を付けた順位) 辞書順 (大文字と小文字、およびアクセント記号を区別しない)
英語、フランス語、ドイツ語	CP 850、CP 858	代替辞書順 (大文字と小文字を区別する) 代替辞書順 (大文字と小文字を区別し、アクセント記号を区別しない) 代替辞書順 (大文字と小文字を区別する、優先度を付けた順位)
ギリシャ語	ISO 8859-7	辞書順 (大文字と小文字、およびアクセント記号を区別する)
ハンガリー語	ISO 8859-2	辞書順 (大文字と小文字、およびアクセント記号を区別する) 辞書順 (大文字と小文字を区別しない、アクセント記号を区別する) 辞書順 (大文字と小文字、およびアクセント記号を区別しない)
日本語	EUCJIS、SJIS、DECKANJI	汎用 (大文字と小文字を区別しない、辞書順)
カザフ語	87	50
ロシア語	CP 866、CP 1251、ISO 8859-5、Koi8、Macintosh Cyrillic	辞書順 (大文字と小文字、およびアクセント記号を区別する) 辞書順 (大文字と小文字を区別しない、アクセント記号を区別する)

言語またはスクリプト	文字セット	ソート順
スカンジナビア語	CP 850	辞書順 (大文字と小文字、およびアクセント記号を区別する) 辞書順 (大文字と小文字を区別しない、優先度を付けた順位)
中国語 (簡体字)	EUC-GB、GB-18030、CP936	汎用 (大文字と小文字を区別しない、辞書順)
スペイン語	ASCII 8、CP 437、CP850、CP 860、CP 863、CP 1252、ISO 8859-1、ISO 8859-15、Macintosh Roman、ROMAN8	辞書順 (大文字と小文字、およびアクセント記号を区別する) 辞書順 (大文字と小文字を区別しない、アクセント記号を区別する) 辞書順 (大文字と小文字、およびアクセント記号を区別しない)
タイ語	CP 874、TIS 620	辞書順
トルコ語	ISO 8859-9	辞書順 (大文字と小文字、およびアクセント記号を区別する) 辞書順 (大文字と小文字、およびアクセント記号を区別しない) 辞書順 (大文字と小文字を区別しない、アクセント記号を区別する)
西欧語	CP 1252	辞書順 (大文字と小文字を区別しない、大文字と小文字を区別する、優先度順、アクセント記号を区別しない、スペイン語辞書、大文字と小文字を区別しないスペイン語、アクセントを区別しないスペイン語)

使用している言語がこの表にない場合は、その言語固有のソート順はありません。この場合は、バイナリ・ソート順を使用してください。また、**sortkey** 関数の使用でニーズが満たされるかどうかを調べてください。この表が示すように、多くの言語には複数のソート順があります。

中国語ピンイン・ソート順

ピンイン (かつての「漢語ピンイン」) では、ローマ字を使用して標準的な中国語の発音を表現します。ピンインは、標準中国語を読み書きする際に漢字を使わずに読み方をローマ字で表現したものです。ピンインでは、アクセント記号を使用して標準中国語の 4 音記号を表します。

以前のバージョンの Adaptive Server では、簡体字中国語 (GB) ソート順 (gbpinyin および gbpinyinocs) が使われており、Unilib 文字セットを使用した場合、GB 文字セットを使用しているデータベースのパフォーマンスが著しく低下していました。

Adaptive Server バージョン 15.0.3 では、自動的に gbpinyin および gbpinyinocs ソート順が使用されるため、処理ステップが省略されてパフォーマンスが大幅に向上します。

以前のバージョンの `size of unilib cache` 設定パラメータのデフォルト・サイズは 268KB でした。バージョン 15.0.3 では、デフォルト・サイズが 302KB に増えました。

ASCII および `gbpinyin` データにアクセスするクエリのパフォーマンスが向上しました。ただし、データ・セットに他の文字が混じっている場合は、パフォーマンスが改善しない場合があります。

Adaptive Server で `gbpinyin` および `gbpinyinocs` ソート順を使用するように設定する方法については、『システム管理ガイド』の「第 9 章 文字セット、ソート順、言語の設定」を参照してください。

中国語と日本語文字セットの大文字と小文字を区別しないソート順の選択

2 つのストアド・プロシージャを使用して、大文字と小文字を区別しないソート順を選択します。

- `sp_helpsort`
- `sp_configure`

`sp_helpsort`

`sp_helpsort` は、大文字と小文字を区別しない利用できるソート順を示します。

```
sp_helpsort
-----
Name                               ID
-----
nocase_eucgb                       52
nocase_cp936                       52
nocase_gb18030                     52
nocase_eucjis                      52
nocase_sjis                        52
nocase_deckanji                    52
```

`sp_configure`

大文字と小文字を区別しないソート順に切り替えるには、次のように入力します。

```
sp_configure 'default sortorder id', 52
```

デフォルト Unicode ソート順の選択

デフォルト Unicode ソート順は、サーバのデフォルト文字セットのソート順とは別のものです。この設定パラメータは静的パラメータであるため、変更した場合は、サーバの再起動と、`unichar` データのインデックス再構築が必要です。このソート順の識別には、一意性を保証するために、数値パラメータではなく文字列パラメータを使用します。

表 10-6 は、使用可能なデフォルトの Unicode ソート順を示します。

表 10-6: デフォルトの Unicode ソート順

名前	ID	説明
defaultml	20	デフォルト Unicode マルチ言語順
thaidict	21	タイ語辞書順
iso14651	22	ISO14651 標準順
utf8bin	24	UTF-8 バイナリと一致する UTF-16 順
binary	25	バイナリ・ソート
altnoacc	39	代替 (アクセント記号を区別しない)
altdict	45	代替 (辞書順)
altnocsp	46	代替 (大文字と小文字を区別しない、優先度を付けた順位)
scandict	47	スカンジナビア語 (辞書順)
scannocp	48	スカンジナビア語 (大文字と小文字を区別しない、優先度を付けた順位)
bin_utf8	50	UTF-8 バイナリ・ソート順
dict	51	汎用 (辞書順)
nocase	52	汎用 (大文字と小文字を区別しない、辞書順)
nocasep	53	汎用 (大文字と小文字を区別しない、優先度を付けた順位)
noaccent	54	汎用 (アクセント記号を区別しない、辞書順)
espdict	55	スペイン語 (辞書順)
espnoacs	56	スペイン語 (大文字と小文字を区別しない、辞書順)
espnoac	57	スペイン語 (アクセント記号を区別しない、辞書順)
rusnoacs	59	ロシア語 (大文字と小文字を区別しない、辞書順)
cyrnoacs	64	キリル語 (大文字と小文字を区別しない、辞書順)
elldict	65	ギリシャ語 (辞書順)
hunct	69	ハンガリー語 (辞書順)
hunnoacs	70	ハンガリー語 (アクセント記号を区別しない、辞書順)
hunnoacs	71	ハンガリー語 (大文字と小文字を区別しない、辞書順)
turknoacs	73	トルコ語 (アクセント記号を区別しない、辞書順)

表 10-7 に、ロード可能なソート順を示します。

表 10-7: ロード可能なソート順

名前	ID	説明
cp932bin	129	CP932 のバイナリ順と一致する順
dynix	130	中国語 (簡体字) 順
gb3213bn	137	GB2312 のバイナリ順と一致する順
cyrdict	140	共通キリル語辞書順
turdict	155	トルコ語辞書順
euckscbn	161	EUCKSC のバイナリ順と一致する順
gbpinyin	163	中国語 (簡体字) 順
rusdict	165	ロシア語辞書順
sjisbin	179	SJIS のバイナリ順と一致する順
eucjisbn	192	EUCJIS のバイナリ順と一致する順
big5bin	194	BIG5 のバイナリ順と一致する順

Adaptive Server でこのソート順リストを表示するには、`sp_helpsort` を使用します。『リファレンス・マニュアル：プロシージャ』を参照してください。

`$$SYBASE/collate/Unicode` ディレクトリにある外部ファイルを使用して、ソート順を追加することができます。ソート順の名前と照合 ID は `syscharsets` に保存されます。外部 Unicode ソート順の名前が `syscharsets` に保存されていなくても、デフォルト Unicode ソート順を設定できます。

注意 外部 Unicode ソート順は Sybase から提供されます。外部 Unicode ソート順を新たに作成することはできません。

Unicode データに関連付けられるソート順は、従来の文字データに関連付けられるソート順とは完全に別のものです。Unicode データ型を使用する関係式はすべて、Unicode ソート順を使用して実行されます。これには、Unicode データと非 Unicode データの両方が関係する混合モードの式も含まれます。たとえば、次のクエリでは、`varchar` 型の文字定数 `'Mü'` が暗黙的に `unichar` にキャストされ、比較は Unicode ソート順に従って実行されます。

```
select * from authors where unicode_name > 'Mü'
```

他のすべての比較演算子、および連結演算子 `+`、演算子 `in`、演算子 `between` についても同様です。既に説明したように、目標は、既存のデータベース・アプリケーションとの互換性を保つことです。

等号によるテーブル・ジョイン (等価ジョイン) には、特に注意が必要です。このジョインは、通常、関連するカラムに定義されているインデックスを利用するように最適化されます。`unichar` 型カラムと `char` 型カラムとをジョインするときは、`char` 型カラムの変換が必要です。また、文字のソート順と Unicode のソート順は異なるので、オプティマイザは `char` 型カラムのインデックスを無視します。

Adaptive Server バージョン 12.5.1 以降では、サーバのデフォルト文字セットが UTF-8 に設定されていれば、サーバのデフォルト・ソート順 (**char** 型データのための) を前述のソート順のいずれかに設定できます。これより前のバージョンでは、UTF-8 に対して正常に機能するソート順はバイナリ・ソート順 “bin_utf8” (ID=50) だけでした。UTF-8 での **char** データのソート順を、**unicar** のソート順に対応するように選択することもできます。ただし、このことは必須ではありません。

Unicode のバイナリ・ソート順を選択するときは、注意が必要です。“binary” という名前のソート順は、**unicar** データ (UTF-16) のソート順として最も効率的であり、そのためデフォルトのソート順となっています。このソート順は、Unicode スカラ値に基づいています。つまり、32 ビットのサロゲート・ペアの順序は、16 ビットの Unicode 値の後になります。“utf8bin” というソート順は、UTF-8 の **char** データのデフォルトの (最も効率的な) バイナリ・ソート順 (“bin_utf8”) と一致するように設計されています。したがって、**unicar** 型に “binary”、UTF-8 **char** 型に “binary” を選択するか、**unicar** 型に “bin_utf8”、UTF-8 **char** 型に “bin_utf8” を選択することをおすすめします。最初の組み合わせは **unicar** 型のソート効率が高く、後者は **char** 型の効率が高くなります。UTF-8 **char** のソート順として “utf8bin” を選択することは避けてください。このソート順は “bin_utf8” と同じですが、効率が悪いからです。

システム・メッセージ用言語の選択

Adaptive Server のインストール環境では、さまざまな言語のメッセージ・ファイルを備えた言語モジュールを使用できます。Adaptive Server で使用できるメッセージの言語モジュールには、英語、中国語 (簡体字)、フランス語、ドイツ語、日本語、韓国語、ブラジルで使用するポルトガル語、スペイン語のものがあります。この他の言語をクライアントが使用する場合は、システム・メッセージはデフォルト言語の英語で表示されます。

同じサーバから出力されるメッセージの言語をクライアントごとに選択できます。たとえば、あるクライアントはフランス語、別のクライアントはスペイン語、さらに別のクライアントはドイツ語で、それぞれメッセージを表示することが可能です。ただし、そのためには、各クライアントが使用する言語がすべて同じ言語グループに属していなければなりません。たとえば、フランス語、スペイン語、ドイツ語はすべて言語グループ 1 に、一方、日本語は、日本語以外の言語を含んでいない言語グループ 101 に属します。したがって、サーバ言語が日本語である場合、システム・メッセージは日本語と英語のどちらかでしか表示できません。すべての言語グループがメッセージを英語で表示できます。サーバ全体のデフォルト言語も設定されており、ユーザが言語を選択していない場合は、そのデフォルト言語が使用されます。Unicode を選択した場合は、サポートされる言語であればどれでもシステム・メッセージを表示できます。

システム・メッセージの言語を選択するには、次の 2 つの方法があります。

- ユーザ・プロファイルにおいて言語を選択する
- `locales.dat` ファイルに言語を入力する

表 10-8 は、サポートされるシステム・メッセージ言語とその言語グループを示します。各ユーザは、セッションごとにシステム・メッセージ言語を 1 つだけ選択できます。

表 10-8: サポートされるシステム・メッセージ

言語グループ	システム・メッセージ言語	文字セット
グループ 1	フランス語、ドイツ語、スペイン語、ブラジルで使用するポルトガル語	ASCII 8、CP 437、CP 850、CP 860、CP 863、CP 1252、ISO 8859-1、ISO 8859-15、Macintosh Roman、ROMAN8
グループ 2	ポーランド語	CP 1250、CP 852、ISO 8859-2
グループ 101	日本語	CP 932、DEC Kanji、EUC-JIS、Shift-JIS
グループ 102	簡体字中国語 (PRC)	CP 936、EUC-GB、GB18030
グループ 104	韓国語	EUC-KSC、CP 949
グループ 105	タイ語	CP 874、TIS 620
Unicode	フランス語、ドイツ語、スペイン語、ブラジルで使用するポルトガル語、日本語、中国語 (簡体字)、韓国語	UTF-8
その他すべての言語グループ	英語	

クライアントでのメッセージの表示に使用されるすべての言語の言語モジュールをインストールしてください。言語モジュールのファイルは Adaptive Server インストール・ディレクトリの `locales` サブディレクトリにあり、「ローカライゼーション・ファイル」と呼ばれるファイル・グループの一部です。ローカライゼーション・ファイルとソフトウェア・メッセージ・ディレクトリ構造の詳細については、「[ローカライゼーション・ファイルの種類](#)」(356 ページ)を参照してください。

サーバの設定：例

この項では、設定オプションとその設定手順について説明します。以下の手順はあくまでも例であり、実際の設定作業に役立つよう概念と方法を例示したものです。

スペイン語版サーバ

この例では、すべてのクライアントが同じ言語を使用する場合に新しいサーバを設定する方法について説明します。必要な作業は次のとおりです。

- 1 サーバ言語を選択します。この例ではスペイン語です。表 10-1 (322 ページ) に示すように、スペイン語は言語グループ 1 に属しています。使用するプラットフォームに基づいて、言語グループ 1 の文字セットを選択します。最も多くのクライアントが使用する文字セットを選択するようおすすめます。将来、他の国にも事業を拡大し、各国の言語に対応する必要があると思われる場合は、Unicode のインストールを検討してください (「[サーバの文字セットの選択](#)」(321 ページ) を参照)。
- 2 サーバにスペイン語の言語モジュールをインストールします。これによって、クライアントにはシステム・メッセージがスペイン語で表示されます。
- 3 デフォルト・ソート順を選択します。表 10-5 (333 ページ) に示すように、スペイン語にはバイナリ・ソート順の他に 3 つのソート順があります。その中からソート順を 1 つ選択します。
- 4 サーバを再起動します。

アメリカ企業の日本法人

この例では、クライアントは日本にあり、データの入力およびソートとシステム・メッセージの受信には日本語を使用しますが、データの送信先となるサーバは英語だけを理解するユーザがアクセスします。

- 1 サーバのデフォルト文字セットを選択します。言語グループ 101 (日本語) の文字セットをインストールした場合は、同じサーバで日本語と英語の両方のデータをサポートできます。
- 2 システム・メッセージを日本語で表示するために、日本語の言語モジュールをインストールします。
- 3 ソート順を選択します。表 10-5 (333 ページ) に示すように、日本語にはバイナリ・ソート順しかありません。したがって、英語と日本語のどちらのクライアントでもバイナリ・ソート順がデフォルト・ソート順となります。両クライアントのユーザに対する解決策として `sortkey` 関数の使用を検討してください。
- 4 日本人ユーザがデフォルトで日本語のメッセージを要求するように設定されていることを確認します。言語グループ 101 の文字セットを使用するのに加え、日本語の言語モジュールをインストールしているので、日本のクライアントにはメッセージが日本語で表示され、アメリカのクライアントはメッセージ言語として英語または日本語を選択できます。

クライアントが複数の国にある日本企業

この会社は日本にあり、フランス、ドイツ、スペインにクライアントがあります。このケースでは、同じサーバにヨーロッパ系の言語とアジア系の言語を併存させる必要があります。

- 1 デフォルトのサーバ言語と文字セットを選択します。この会社の本社は日本にあり、クライアントのほとんどが日本にあるため、デフォルトのサーバ言語は日本語でなければなりません。しかし、フランス、ドイツ、スペインのクライアントがデータをそれぞれのネイティブ言語で送受信できるようにすることも必要です。表 10-1 (322 ページ) に示すように、日本語は言語グループ 101 に属しているのに対し、フランス語、ドイツ語、スペイン語は言語グループ 1 に属しています。必要な言語がすべて同じ言語グループに属してはいないので、これらの言語を同じサーバでサポートする唯一の方法は、デフォルト文字セットとして Unicode を選択することです。
- 2 日本語、フランス語、ドイツ語、スペイン語の言語モジュールをインストールします。
- 3 バイナリ・ソート順を選択します。これが Unicode 文字セットで使用できる唯一のソート順だからです (ただし、各ユーザの好みに従ってデータをソートできるように、アプリケーション・コードに `sortkey` 関数を追加することも検討します)。
- 4 システム・メッセージのデフォルト言語として日本語を選択します。他の国のクライアントでは、それぞれのネイティブ言語をメッセージ言語として選択できます。

文字セット、ソート順、メッセージ言語の変更

サーバの設定が完了した後も、システム管理者は Adaptive Server のデフォルト文字セット、ソート順、メッセージ言語を変更できます。ソート順は特定の文字セットに基づいて作成されるので、文字セットを変更すると通常はソート順も変更されます。しかし、1つの文字セットに対して利用できるソート順は複数あるので、文字セットを変更しなくてもソート順を変更できます。

Adaptive Server のデフォルトのソート順、文字セット、プライマリ・ソート順のテーブルを表示するには、次のように入力します。

```
sp_helpsort
```

デフォルト文字セットの変更

Adaptive Server の「デフォルト文字セット」は 1 つだけ指定できます。この文字セットを使用してデータベース内にデータが保管されます。Adaptive Server をインストールするときに、デフォルト文字セットを指定します。

警告！ Adaptive Server のデフォルト文字セットを変更する場合は、次の指示に従い、十分に注意して実行してください。デフォルト文字セットを変更する前にバックアップを行うよう強くおすすめします。

Adaptive Server のデフォルト文字セットを変更するときは、既存のデータを新しいデフォルト文字セットに変換する必要があります。変換が不要なのは以下の場合だけです。

- サーバ内にユーザ・データがない。
- サーバ内のユーザ・データが破壊されてもかまわない。
- サーバ内のデータが ASCII-7 だけを使用しているという絶対の確信がある。このような場合は、デフォルトを変更する前にデータをサーバからコピー・アウトする必要はありません。

それ以外の場合は、以下の手順で必ず既存のデータを変換してください。

- 1 `bcp` を使用してデータをコピー・アウトします。
- 2 デフォルト文字セットを変更します。
- 3 データ変換用のフラグを指定して `bcp` を実行し、データをサーバにコピー・インします。

`bcp` を使用したデータのコピー方法の詳細については、『ユーティリティ・ガイド』を参照してください。

警告！ データを別の文字セット (特に UTF-8) に変換すると、変換後のデータが大きくなりすぎ、割り当てられたカラム・サイズに入り切らなくなることがあります。データが入りきらないカラムは作成し直す必要があります。

既存のデータの文字セットと新しいデフォルト文字セットとの間のコード変換がサポートされている必要があります。サポートされていないと、変換エラーが発生して、データは正しく変換されません。サポートされている文字セット変換の詳細については、「[第 11 章 クライアント／サーバの文字セット変換の設定](#)」を参照してください。

文字セット間で変換がサポートされていても、文字セット間に多少相異があることや、他の文字セットに相当する文字がないなどの理由から、エラーが発生することがあります。不完全な文字や無効な文字のあるデータのローを、データベースにコピー・インしないでください。

リソース・ファイルを使ったソート順の変更

リソース・ファイルを使用して、Adaptive Server の文字セットを変更できます。サンプル・リソース・ファイル *sqlloc.rs* は、*\$SYBASE/ASE-12_5/init/sample_resource_files/* にあります。

Adaptive Server 12.5.1 インストール環境のリソース・ファイルは次のようになります。

```
sybinit.release_directory: USE_DEFAULT
sqlsrv.server_name: PUT_YOUR_SERVER_NAME_HERE
sqlsrv.sa_login: sa
sqlsrv.sa_password:
sqlsrv.default_language: USE_DEFAULT
sqlsrv.language_install_list: USE_DEFAULT
sqlsrv.language_remove_list: USE_DEFAULT
sqlsrv.default_characteraset: USE_DEFAULT
sqlsrv.characteraset_install_list: USE_DEFAULT
sqlsrv.characteraset_remove_list: USE_DEFAULT
sqlsrv.sort_order: USE_DEFAULT
# An example sqlloc resource file...
# sybinit.release_directory: USE_DEFAULT
# sqlsrv.server_name: PUT_YOUR_SERVER_NAME_HERE
# sqlsrv.sa_login: sa
# sqlsrv.sa_password:
# sqlsrv.default_language: french
# sqlsrv.language_install_list: spanish,german
# sqlsrv.language_remove_list: USE_DEFAULT
# sqlsrv.default_characteraset: cp437
# sqlsrv.characteraset_install_list: mac,cp850
# sqlsrv.characteraset_remove_list: USE_DEFAULT
# sqlsrv.sort_order: dictionary
```

デフォルト・ソート順の変更

Adaptive Server の「デフォルトのソート順」は 1 つだけ設定できます。これはデータを並べ替えるために使用される照合順序です。特定の Adaptive Server の文字データのソート順を変更するときは次の点に注意してください。同じ組織に複数の Adaptive Server がある場合は、すべて同じソート順を使用することをおすすめします。ソート順を 1 つにすることで一貫性が保たれ、分散処理の管理が簡単になります。

デフォルト・ソート順を変更すると、インデックスの再構築が必要になる場合があります。詳細については、「[文字セット、ソート順、メッセージ言語の再設定](#)」(344 ページ) を参照してください。

文字セット、ソート順、メッセージ言語の再設定

この項では、Adaptive Server のデフォルトの文字セット、ソート順、メッセージ言語を変更する前と後に行う手順の要約を説明します。新しいサーバの文字セット、ソート順、メッセージ言語を設定する方法については、プラットフォームの『Adaptive Server Enterprise 設定ガイド』を参照してください。

文字セットまたはソート順を変更する前と後に、Adaptive Server のすべてのデータベースのバックアップを作成してください。次の条件が当てはまる場合は、データベースをバックアップした後に `bcp` を使用してデータをコピー・インおよびコピー・アウトしてください。

- データベース内に文字データがあり、そのデータを新しい文字セットに変換する必要がある。このような場合は、サーバのデフォルト文字セットの変更後にデータのデータベース・ダンプをロードしないでください。ロードされるデータは新しい文字セットに基づくものと解釈され、データは破損した状態になります。
- 変更するのはデフォルト・ソート順だけで、デフォルト文字セットは変更しない。このような場合は、ソート順を変更する前に作成したダンプからデータベースをロードすることはできません。ロードしようとすると、エラー・メッセージが表示され、ロードはアポートされます。
- デフォルトの文字セットを変更するが、新旧どちらかのソート順がバイナリでない。このような場合は、文字セットを変更する前に作成したデータベース・ダンプをロードすることはできません。

新旧どちらの文字セットもバイナリ・ソート順を使用し、両方の文字セット間の変換を必要としない場合を除き、デフォルト文字セットとソート順を再設定した後にデータベース・ダンプからデータを再ロードすることはできません。詳細については、「[デフォルト文字セットの変更](#)」(342 ページ)を参照してください。

Unicode の例

次の例では、`xpubs` という名前の架空のデータベースで `univarchar` 型のカラムを使用するように変更します。

スキーマ

インストール時にすべてデフォルトを選択して設定されたサーバ (文字セットは “iso_1”、デフォルト・ソート順は ID 50 の “binary_iso_1”) 上で、次のスクリプトを使用してデータベースを作成したとします。

```
> create database xpubs
> go
> use xpubs
> go
> create table authors (au_id int, au_lname varchar(255),
au_fname varchar(255))
> go
> create index au_idx on authors(au_lname, au_fname)
> go
```

その後で、一連の挿入と更新によって、データがサーバにロードされました。

UTF-8 への変換

Unicode を使用するには、初めに、データを抽出して UTF-8 形式に変換します。

```
% bcp xpubs..authors out authors.utf8.bcp -c -Jutf8 -Usa -P
```

次に、UTF-8 をデフォルト文字セットとしてサーバにインストールします。

```
% charset -Usa -P binary.srt utf8
% isql -Usa -P
> sp_configure 'default sortorder id', 50, 'utf8'
> go
> shutdown
> go
```

サーバを再起動すると、デフォルト文字セットが変更され、システム・テーブルのインデックスが再作成されます。もう一度サーバを再起動して、データを再ロードします。

```
% isql -Usa -P
> sp_dboption xpubs, 'select into', true
> go
> use xpubs
> go
> checkpoint
> go
> delete from authors
> go
> quit

% bcp xpubs..authors in authors.utf8.bcp -c -Jutf8 -Usa -P
```

選択したカラムを `unichar` にマイグレートする

作業データベースのデフォルト文字セットが UTF-8 に設定されている場合は、選択したカラムを `univarchar` に簡単に変換できます。

```
% isql -Usa -P
> use xpubs
> go
> alter table authors modify au_lname univarchar(255),
au_fname univarchar(255)
> go
```

カラムが新しいデータ型に変更され、データが適切に変換され、インデックスが再作成されます。

`unitext` へ、または `unitext` からのマイグレーション

現時点では、`alter table modify` コマンドには、`text`、`image`、または `unitext` カラムを指定できません。`text` カラムを `unitext` カラムにマイグレートするには、`bcp` を実行し、`unitext` カラムのテーブルを作成し、再度 `bcp` を実行して新しいテーブルにデータを挿入します。この方法でマイグレートするには、`bcp` の呼び出し時に `-Jutf8` オプションを指定する必要があります。

準備手順

インストール・プログラムを実行して `Adaptive Server` を再設定する前に、次の手順を実行してください。

- 1 すべてのユーザ・データベースと `master` データベースをダンプします。`model` または `sysystemprocs` の内容を変更した場合は、これらもダンプします。
- 2 言語モジュールがロードされていない場合はロードしてください (詳細は、プラットフォームの『`Adaptive Server Enterprise` 設定ガイド』を参照してください)。
- 3 `Adaptive Server` のデフォルト文字セットを変更する場合で、現在使用しているデータベースに `ASCII-7` 以外のデータが含まれているときは、`bcp` を使用してデータベースの既存のデータをコピー・アウトしてください。

言語モジュールをロードした後で、`Adaptive Server` のインストール・プログラムを実行して以下の操作を行うことができます。

- `Adaptive Server` に組み込まれているメッセージ言語と文字セットをインストールまたは削除する。
- デフォルトのメッセージ言語または文字セットを変更する。
- 別のソート順を選択する。

インストール・プログラムの使用方法については、プラットフォームの『Adaptive Server Enterprise 設定ガイド』を参照してください。

注意 文字セットやソート順を変更するには、サーバで管理されているデータベースをすべてオープンしておく必要があります。オープンしているデータベースの数が不足していると、ソート順を変更してから再起動したときに、次のメッセージがエラー・ログに記録され、サーバのソート順は以前のソート順に戻ります。

```
The configuration parameter 'number of open databases'
must be at least as large as the number of databases,
in order to change the character set or sort order." Re-
start Adaptive Server, use sp_configure to increase
'number of open databases' to at least %d, then re-
configure the character set or sort order
```

言語、文字セット、またはソート順を再設定する必要がある場合は、『ユーティリティ・ガイド』で説明している `sqlloc` ユーティリティを使用してください。Windows を使用している場合は、『設定ガイド』の「第 3 章 Adaptive Server のデフォルト設定」で説明している `Server Config` ユーティリティを使用してください。

追加の言語をインストールしたけれども、Adaptive Server の文字セットやソート順は変更していない場合は、以上で再設定のプロセスは終了です。

Adaptive Server のデフォルト文字セットを変更した場合で、現在のデータベースに ASCII-7 以外のデータがあるときは、`bcp` を使用してデータをデータベースにコピー・インします。このとき、変換のためのフラグを指定してください。

Adaptive Server のデフォルトのソート順または文字セットを変更した場合は、「[文字セット、ソート順、メッセージ言語の再設定](#)」(344 ページ) を参照してください。

ユーザのデフォルト言語の設定

追加の言語がインストールされると、クライアント・プログラムを実行するユーザは、`create login` を実行してその言語をデフォルト言語として設定することや、`locales.dat` に定義されているエントリを使用してクライアント・マシンの `LANG` 変数を設定することができます。

再設定後のリカバリ

Adaptive Server が停止して再起動するたびに、各データベースのリカバリが自動的に実行されます。自動リカバリについては、『システム管理者ガイド 第2巻』の「第11章 バックアップおよびリカバリ・プランの作成」を参照してください。

リカバリが完了すると、新しいソート順と文字セットの定義がロードされます。

ソート順を変更すると Adaptive Server はシングルユーザ・モードに切り替わります。このとき、システム・テーブルに対して必要な更新を加えることができますが、他のユーザはサーバを使用できなくなります。文字ベースのインデックスを持つシステム・テーブルについては、ソート順の変更によって壊れたインデックスがあるかどうかのチェックが自動的に行われます。テーブルの文字ベースのインデックスは、必要に応じて新しいソート順の定義を使用して自動的に再構築されます。

システムのインデックスが再構築された後で、文字ベースのユーザ・インデックスについて、`sysindexes` システム・テーブル内に「suspect (疑わしい)」というマークが付けられます。疑わしいインデックスを持つユーザ・テーブルについては、`sysobjects` テーブル内で「read only (読み込み専用)」というマークが付けられます。これによって、疑わしいインデックスのチェックと、必要に応じて再構築が実行されるまでは、このようなテーブルへの更新とインデックスの使用はできなくなります。

範囲分割されたユーザ・テーブルの文字ベースのパーティション・キーがチェックされ、ソート順の変更または文字セットの変更によってパーティションが破損している可能性がある場合は、「suspect (疑わしい)」のマークが付きます。

次に、設定情報を保持するディスク領域にある古い情報が、新しいソート順情報に置き換えられます。その後、Adaptive Server が停止します。次のセッションを開始するときは、システム情報は完全に正確な状態になっています。

`sp_indsuspect` による壊れたインデックスの検索

Adaptive Server の停止後に再起動して `sp_indsuspect` を実行すると、どのユーザ・テーブルのインデックス再構築が必要かを調べることができます。

```
sp_indsuspect [tab_name]
```

この場合、`tab_name` はチェックするテーブルの名前です。`tab_name` を指定しないで `sp_indsuspect` を実行すると、現在のデータベースにあるテーブルのうち、ソート順の変更時に「suspect (疑わしい)」というマークが付けられたインデックスを持つすべてのテーブルのリストが作成されます。

次の例は、mydb データベースで `sp_indsuspect` を実行した結果、疑わしいインデックスが 1 つ見つかったことを示しています。

```
sp_indsuspect

Suspect indexes in database mydb
Own.Tab.Ind (Obj_ID, Ind_ID) =
dbo.holdings.h_name_ix(160048003, 2)
```

ソート順を変更した後のインデックスの再構築

`dbcc reindex` によって、`dbcc checktable` の「高速」バージョンが実行され、ユーザ・テーブルのインデックスの整合性が検査されます。詳細については、『システム管理ガイド 第 2 巻』の「第 10 章 データベースの一貫性の検査」を参照してください。`dbcc reindex` は、使用されているソート順が新しいソート順と一致しないインデックスを削除して、再構築します。`dbcc reindex` が最初のインデックス関連エラーを発見すると、メッセージが表示され、ソート順の違うインデックスは再構築されません。Adaptive Server でのソート順を変更した後に、システム管理者またはテーブル所有者は `dbcc reindex` を実行してください。

```
dbcc reindex ({table_name | table_id})
```

`sp_indsuspect` によって表示された、疑わしいインデックスを持つすべてのテーブルに対してこのコマンドを実行してください。次に例を示します。

```
dbcc reindex(titles)

One or more indexes are corrupt. They will be rebuilt.
```

これは、`dbcc reindex` で、`titles` テーブルに 1 つまたは複数の疑わしいインデックスが発見されたことを示しています。`dbcc reindex` は、該当するインデックスに対して削除と再作成を実行します。

テーブルのインデックスが正しい場合やテーブルにインデックスがない場合は、`dbcc reindex` を実行したときにインデックスの再構築は行われませんが、代わりに、メッセージが表示されます。テーブルに破壊されたデータが含まれているという疑いがある場合は、このコマンドはアボートされます。この場合、`dbcc checktable` の実行を指示するエラー・メッセージが表示されます。

`dbcc reindex` が正常終了したときは、そのテーブルのインデックスの「suspect (疑わしい)」マークはすべて削除されています。テーブルの「read-only (読み込み専用)」マークも削除され、テーブルを更新できるようになります。インデックスを再構築する必要があるかどうかに関係なく、この 2 つのマークは削除されます。

`dbcc reindex` を使用して、システム・テーブルのインデックスを再構築することはできません。システム・インデックスは、Adaptive Server がソート順を変更して再起動した後で、自動リカバリの一部として必要に応じて検査され再構築されます。

文字セットを変更した後の text データのアップグレード

Adaptive Server の文字セットをマルチバイト文字セットに変更した場合は、`dbcc fix_text` を使用して `text` 型の値をアップグレードしてください。

`text` 型の値は複数のページにわたるほどの大きさになることがあるため、Adaptive Server はページ境界を越える文字データを処理する必要があります。これには、個々の `text` ページに関する追加の情報が必要です。システム管理者またはテーブル所有者は、`text` データが含まれる各テーブルについて、`dbcc fix_text` を実行して、必要な新しい値を計算する必要があります。

`text` データが含まれているすべてのテーブルの名前を調べるには、次のクエリを使用してください。

```
select sysobjects.name
from sysobjects, syscolumns
where syscolumns.type = 35
and sysobjects.id = syscolumns.id
```

システム管理者またはテーブル所有者は、必要となる新しい値を計算するために、`dbcc fix_text` を実行してください。

`dbcc fix_text` の構文は次のとおりです。

```
dbcc fix_text (table_name | table_id)
```

指定するテーブルは、現在のデータベースになければなりません。

`dbcc fix_text` は、指定されたテーブルをオープンし、`text` 値ごとに必要な文字統計値を計算し、適切なページ・ヘッダ・フィールドに追加します。この処理は、テーブルにある `text` 値のサイズによっては時間がかかることがあります。`dbcc fix_text` によって多数のログ・レコードが生成され、トランザクション・ログが満杯になることがあります。`dbcc fix_text` は一連の小さなトランザクションで更新を行うので、ログが満杯になっても、失われる作業はわずかです。

ログ・スペースが不足した場合は、ログを消去してください。詳細については、『システム管理ガイド 第2巻』の「第12章 ユーザ・データベースのバックアップとリストア」を参照してください。その後、元の `dbcc fix_text` が停止したときに更新していたのと同じテーブルを指定して、`dbcc fix_text` を再起動してください。マルチバイトの `text` 値のそれぞれに、その `text` 値がアップグレードされたかどうかを示す情報が含まれています。したがって、`dbcc fix_text` は、前回までに処理されていない `text` 値だけをアップグレードします。

データベースのログが別のセグメントに保管されている場合は、スレッショルドを使用してログのクリアを管理できます。『システム管理ガイド 第2巻』の「第16章 スレッショルドによる空き領域の管理」を参照してください。

`dbcc fix_text` がテキスト・ページに対する必要なロックを取得できない場合は、次のように問題点がレポートされ、処理は続行します。

```
Unable to acquire an exclusive lock on text page 408. This text
value has not been recalculated. In order to recalculate those
TEXT pages you must release the lock and reissue the dbcc
fix_text command.
```

文字セットを変更した後の text 値の検索

マルチバイトの文字セットに変更した後で **text** 値を検索しようとしたとき、**dbcc fix_text** をまだ実行していない場合は、コマンドは正しく実行されず、次のエラー・メッセージが生成されます。

```
Adaptive Server is now running a multi-byte character set, and
this TEXT column's character counts have not been recalculated
using this character set. Use dbcc fix_text before running this
query again.
```

ソート順または文字セットを変更した後でエラーが発生した場合は、『ASE トラブルシューティング&エラー・メッセージ・ガイド』の「ソート順またはデフォルトの文字セットを手動で変更する方法」を参照してください。

疑わしいパーティションの処理

パーティションに「suspect (疑わしい)」のマークが付けられる理由は次の 2 つです。

- 範囲分割テーブルのソート順または文字セットが変更された
- ハッシュ分割されたテーブルについてプラットフォーム間でダンプとロードが行われた

テーブルに疑わしいパーティションがあるとマークされた場合、次の処理が行われます。

- このテーブルのすべての更新およびカーソル・アクティビティが保留されます。
- **partition by** を除き、**alter table** コマンドは使用できません。疑わしいパーティションのあるテーブルについて、**create index** および **drop index** は使用できません。
- 疑わしいパーティションのあるテーブルに対して、**select** コマンドを使用できます。ただし、オプティマイザは、破損の可能性のある **partition** 条件の使用を回避するために、このようなテーブルをラウンドロビン方式で分割されたテーブルとして処理します。

疑わしいパーティションを含んでいるテーブルの修正

- ソート順の変更後に `partition` 条件の修正が必要な場合は、`alter table` コマンドと一緒に `partition by` オプションを使用して疑わしいパーティションのあるテーブルを再分割してください。
- パーティション条件の修正が必要ない場合は、`reorg rebuild table` コマンドを使用してテーブルを再構築し、パーティション間にデータ・ローだけを再配分します。
- テーブルのパーティションだけでなくインデックスも「疑わしい」とされた場合は、`partition by` または `reorg rebuild` を使用して疑わしいインデックスとパーティションの両方を修正してください。

プラットフォーム間のダンプおよびロード操作時の疑わしいパーティションの処理

- 初めての `online database` コマンドの実行中にエンディアン・タイプ異なる2つのプラットフォームで `load database` を実行すると、ハッシュ分割に「suspect (疑わしい)」のマークが付けられます。
- `unichar` または `varchar` 分割キーで内部生成されたパーティション条件を持つラウンドロビン分割のグローバル・クラスタード・インデックスは、「suspect (疑わしい)」のマークが付けられます。
- データベースがオンラインになったら、`sp_post_xpload` を使用して疑わしい分割およびインデックスを修正します。

サポートされていない言語の日付文字列のインストール

`sp_addlanguage` を使用すると、言語モジュールが用意されていない言語での曜日と月の名前をインストールできます。`sp_addlanguage` を使用して、次のものを定義します。

- 言語の名前とその名前のエイリアス (オプション)
- 月のフルネームのリストと省略名のリスト
- 曜日のフルネームのリスト
- 日付の入力フォーマット (たとえば、月/日/年)
- 最初の曜日の番号

次にイタリア語の情報を追加する例を示します。

```
sp_addlanguage italian, italiano,  
"gennaio, febbraio, marzo, aprile, maggio, giugno, luglio, agosto, settembre, ottobre,  
novembre, dicembre",  
"genn, feb, mar, apr, mag, giu, lug, ago, sett, ott, nov, dic",  
"lunedì, martedì, mercoledì, giovedì, venerdì, sabato, domenica",  
dmy, 1
```


`sp_addlanguage` では、正確なデータの入力規則が適用されます。月の名前、月の省略名、曜日のリストはカンマで区切ります。途中にスペースや行送り (改行) を入れないでください。また、指定する要素の数が正確でなければなりません (月の文字列は 12 個、曜日の文字列は 7 個です)。

日付フォーマットとして有効な値は、`mdy`、`dmy`、`ymd`、`ydm`、`myd`、`dym` です。`dmy` と指定すると、日付の順は「日/月/年」となります。このフォーマットは、データの入力だけに影響します。出力フォーマットを変更するときは、`convert` 関数を使用してください。

サーバとクライアントでの日付の解釈

通常、日付値はクライアント側で分解されます。ユーザが日付値を選択すると、Adaptive Server はその値を内部フォーマットでクライアントに送ります。クライアントは、そのクライアント上の *locales* ディレクトリのデフォルト言語サブディレクトリにある *common.loc* ファイルと他のローカライゼーション・ファイルを使用して、内部フォーマットを文字データに変換します。たとえば、ユーザのデフォルト言語がスペイン語の場合、Adaptive Server は */locales/spanish/char_set* ディレクトリにある *common.loc* ファイルを探します。このファイルの情報を使用して、たとえば `12 febrero 2007` と表示します。

Adaptive Server の言語モジュールが提供されていないイタリア語がユーザのデフォルト言語であり、イタリア語の日付値が追加されているとします。クライアントは、サーバに接続するときにイタリア語の *common.loc* ファイルを探しますが、そのファイルは見つかりません。クライアントはエラー・メッセージを表示し、そのサーバに接続します。その後、ユーザが日付値を選択すると、日付は米語フォーマットで表示されます。`sp_addlanguage` で追加された日付値を表示するには、`convert` 関数を使用して、サーバで日付を文字データに強制的に変換してください。

次のクエリで生成される結果セットでは、日付は米語フォーマットです。

```
select pubdate from titles
```

次のクエリの場合は、月の名前がイタリア語で返されます。

```
select convert(char(19),pubdate) from titles
```

国際化ファイルとローカライゼーション・ファイル

国際化ファイルの種類

特定の言語でデータ処理をサポートするためのファイルを「国際化ファイル」と呼びます。Adaptive Server に付属する国際化ファイルには、いくつかの種類があります。

表 10-9: 国際化ファイル

ファイル	ロケーション	目的と内容
<i>charset.loc</i>	<i>charsets</i> ディレクトリにある各文字セットのサブディレクトリ	英数字、句読点、オペランド、大文字、小文字などの文字の辞書的属性を定義する、文字セット定義ファイル。Adaptive Server がデータを正しく処理するために必要。
<i>*.srt</i>	<i>charsets</i> ディレクトリにある各文字セットのサブディレクトリ	文字、数字、特殊文字のソート順を定義する。合字や発音区別符号などの言語特有の規則も考慮される。
<i>*.xlt</i>	<i>charsets</i> ディレクトリにある各文字セットのサブディレクトリ	bcp や isql などのユーティリティで使用される、端末固有の文字を変換するファイル。 <i>.xlt</i> ファイルの使用方法の詳細については、「 第 11 章 クライアント/サーバの文字セット変換の設定 」と『ユーティリティ・ガイド』を参照。

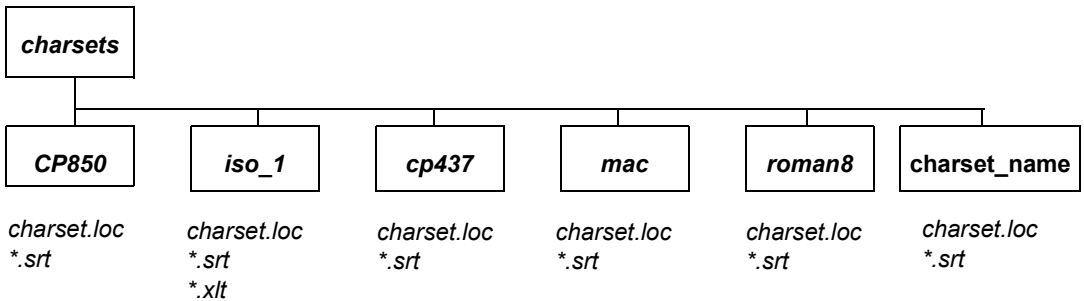
警告！ 国際化ファイルは変更できません。新しい端末定義またはソート順のインストールが必要な場合は、最寄りの Sybase 社または販売代理店に連絡してください。

文字セットのディレクトリ構造

図 10-3 は、Adaptive Server とともに提供される西欧言語の文字セットのディレクトリ構造を示します。*charsets* ディレクトリには、文字セットごとに別のサブディレクトリがあります。それぞれの文字セット (*cp850* など) のサブディレクトリには、文字セットとソート順の定義ファイルおよび端末固有のファイルがあります。

追加の文字セットをロードした場合も、その文字セットのサブディレクトリが *charsets* ディレクトリの下に作成されます。

図 10-3: charsets ディレクトリの構造



次のグローバル変数には、文字セットの情報が格納されています。

- `@@char_convert` - 文字セット変換が有効でない場合は 0 が格納される。文字セット変換が有効な場合は 1 が格納される。
- `@@client_csname` - クライアントの文字セット名。クライアント文字セットが初期化されていない場合は NULL に設定され、初期化されている場合は、その接続の文字セット名が格納される。
- `@@client_csid` - クライアントの文字セット ID。クライアント文字セットが一度も初期化されていない場合、-1 に設定される。一度でも初期化された場合、`syscharsets` から返された接続のためのクライアント文字セット ID に設定される。
- `@@client_csexpansion` - サーバの文字セットをクライアントの文字セットに変換するとき使用する拡張係数を返す。
- `@@maxcharlen` - Adaptive Server のデフォルト文字セット中の 1 文字の最大長 (バイト)。
- `@@ncharsize` - 現在のサーバのデフォルト文字セット中の 1 文字セットの最大長 (バイト単位)。
- `@@unicharsize` - 2 に等しい。

ローカライゼーション・ファイルの種類

Adaptive Server の言語モジュールごとに、以下のローカライゼーション・ファイルが用意されています。

- *locales.dat* – *locales* ディレクトリ内。デフォルト・メッセージ言語とデフォルト文字セットを識別するためにクライアント・アプリケーションが使用する。
- *server.loc* – *\$\$SYBASE/\$SYBASE_ASE/locales* ディレクトリ内の言語サブディレクトリの下に文字セット・サブディレクトリ内。ローカル言語に翻訳されたソフトウェア・メッセージ。Sybase 製品は製品固有の **.loc* ファイルを持つ。内容が翻訳されていない場合は、そのソフトウェアのメッセージまたはテキストはローカル言語ではなく米語で表示される。
- *common.loc* – *locales* ディレクトリの言語ディレクトリと文字セットのディレクトリ内。各国の言語での月名とその短縮形、および各国で使われる日付、時間、通貨のフォーマットに関する情報。

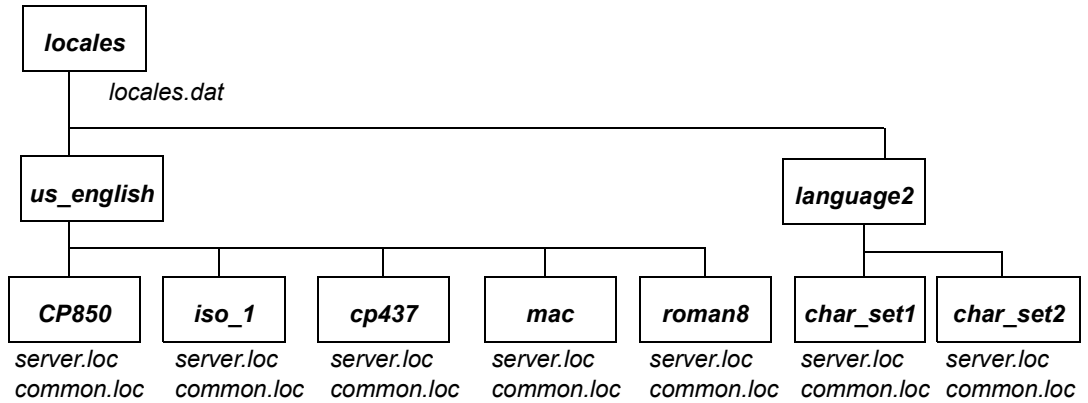
dataserver、*sqlloc*、*syconfig* などが使用するすべての Adaptive Server 関連ロケール・ファイルは *\$\$SYBASE/\$SYBASE_ASE/locales* にあります。すべての Open Client/Server 関連ロケール・ファイル (*ctlib*、*ctisql*、*ctbcp*、*optdiag*、*installjava* など) は *\$\$SYBASE/locales* にあります。

警告！ ローカライゼーション・ファイルは変更できません。このファイル内の情報を変更する必要がある場合は、最寄りの Sybase 社または販売代理店に連絡してください。

ソフトウェア・メッセージのディレクトリ構造

図 10-4 は、ローカライゼーション・ファイルがどのように配置されているかを示します。*locales* ディレクトリ内には、インストールされた言語ごとにサブディレクトリがあります。*us_english* サブディレクトリは必ずあります (PC プラットフォームでは、このディレクトリの名前は *english* です)。インストール中に、Adaptive Server 上にインストールする言語を選択するときに、サポートされているソフトウェア・メッセージ言語のリストが画面に表示されます。言語を追加するための言語モジュールをインストールすると、その言語のサブディレクトリが作成されます。それぞれの言語のサブディレクトリには、サポートされている文字セットのサブディレクトリがあります。たとえば、*cp850* は *us_english* で使用可能な文字セットです。Sybase 製品のソフトウェア・メッセージ・ファイルは、文字セット・サブディレクトリにあります。

図 10-4: メッセージ・ディレクトリ構造



メッセージ言語とグローバル変数

次のグローバル変数には、言語に関する情報が格納されています。

- `@@langid` – 現在使用している言語のローカル言語 ID (`syslanguages.langid` で指定されている) が格納される。
- `@@language` – 現在使用している言語の名前 (`syslanguages.name` で指定されている) が格納される。

クライアント／サーバの文字セット変換の設定

トピック名	ページ
文字セット変換	359
サポートする文字セット変換	360
変換タイプの選択	363
文字セット変換の有効化と無効化	366
文字セット変換のエラー処理	367
変換とデータ長の変更	367
ユーティリティ・プログラムのための文字セットの指定	369
表示およびファイル文字セットのコマンド・ライン・オプション	369

文字セット変換

異機種間環境では、Adaptive Server は、異なるプラットフォームで稼働し、異なる文字セットを使用しているクライアントと通信することがあります。異なる文字セットが同じ言語をサポートしていることがあります(たとえば、ISO 8858-1 と CP 850 はグループ 1 の言語をサポートしています)、それらの文字セットは同じ文字を異なる方法でコード化する場合があります。たとえば、ISO 8859-1 では、*à* という文字は 16 進の *0xE0* にコード化されます。しかし、CP 850 では、この文字は 16 進数の *0x85* にコード化されます。

クライアント／サーバ間のデータの整合性を維持するには、データを文字セット間で変換する必要があります。これは、マシンや文字セットの種類が変わっても、“a” が常に “a” となるようにするためです。このプロセスを「文字セット変換」と呼びます。

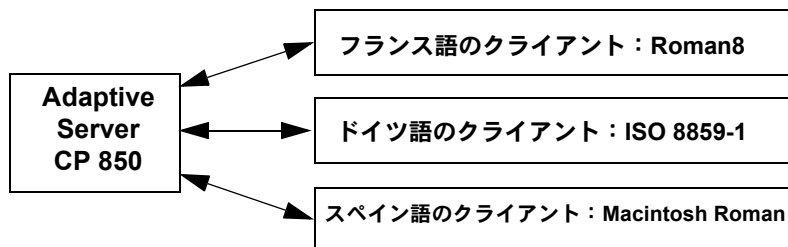
サポートする文字セット変換

文字セット変換は、1 対の文字セット間で行われます。それぞれのクライアント／サーバ・システムでサポートする変換は、そのサーバとクライアントで使用する文字セットにより異なります。サーバがネイティブな文字セットをデフォルトとして使用する場合と、Unicode UTF-8 を使用する場合とでは、異なるタイプの文字セット変換が行われます。

ネイティブな文字セットでの変換

Adaptive Server では、同じ言語グループに属するネイティブな文字セット間での変換がサポートされています。あるネイティブな文字セットがサーバのデフォルトとして設定されている場合、クライアントの文字セットも同じ言語グループに属している必要があります。図 11-1 は、西欧のサーバ／クライアント・システムの例を示しています。この例では、クライアントの文字セットと Adaptive Server のデフォルトの文字セットはすべて言語グループ 1 に属しています。クライアントの文字セットとサーバのデフォルトの文字セットの間で正しくデータが変換されています。クライアントはすべて同じ言語グループに属しているため、どのクライアントから送信されたデータであっても、サーバ上のデータはすべてのクライアントで表示できます。

図 11-1: サーバとクライアントの文字セットが同じ言語グループに属している場合の文字セット変換

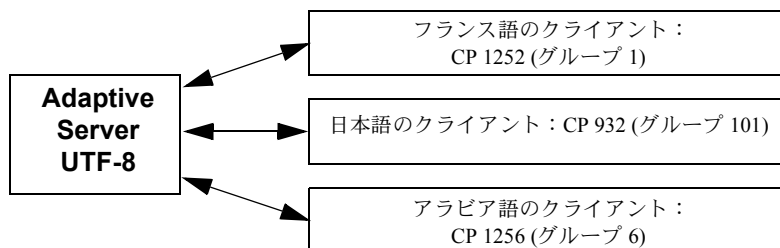


言語グループとサポートされる文字セットの一覧については、表 10-1 (322 ページ) を参照してください。

Unicode システムでの変換

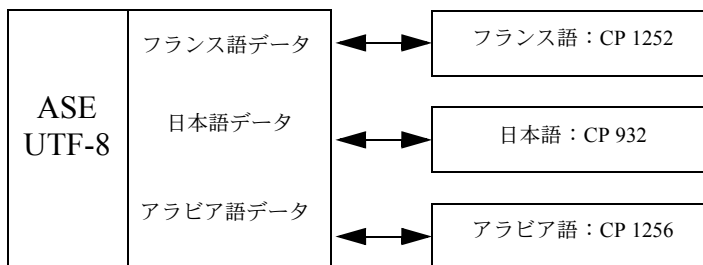
Adaptive Server では、Sybase がサポートするすべてのネイティブな文字セットと UTF-8 との間の文字セット変換もサポートされています。Unicode システムでは、サーバのデフォルトの文字セットが UTF-8 であるため、クライアントの文字セットは、どの言語グループのネイティブ文字セットであってもかまいません。したがって、日本語のクライアント (グループ 101)、フランス語のクライアント (グループ 1)、アラビア語のクライアント (グループ 6) はいずれも、同じサーバとの間でデータの送受信が可能です。それぞれのクライアントからのデータは、各クライアントとサーバ間での受け渡し時に、正しく変換されます。

図 11-2: Unicode システムでの文字セット変換



それぞれのクライアントで表示できるのは、その文字セットでサポートされている言語のデータだけです。したがって、日本語のクライアントは、サーバ上の日本語のデータはすべて表示できますが、アラビア語やフランス語のデータは表示できません。同様に、フランス語のクライアントは、フランス語の他に、文字セットでサポートされている西欧言語を表示できますが、日本語やアラビア語は表示できません。

図 11-3: Unicode データの表示



追加文字セットである ASCII-7 は、Unicode をはじめ すべての文字セットのサブセットであり、したがって、すべての言語グループのすべての文字セットと互換性があります。Adaptive Server またはクライアントのどちらかの文字セットが ASCII-7 ならば、7 ビットの ASCII 文字をクライアントとサーバ間で送受信するときに変更も変換も必要ありません。

サーバの文字セットを ASCII-7 に設定することはおすすめしません。ただし、各クライアントに対して、それぞれのネイティブな文字セットの最初の 128 文字だけを使用するように制限すると、同じ互換性が得られます。

Adaptive Server 直接変換

Adaptive Server 直接変換は、同じ言語グループの2つのネイティブな文字セット間での変換です。たとえば、CP 437 と CP 850 は両方ともグループ 1 の言語グループに属しているため、Adaptive Server はこれらの間での変換をサポートします。Adaptive Server 直接変換は、同じ言語グループの数多くの (すべてではない) ネイティブの文字セット間で行われます (表 11-1 (364 ページ) 参照)。

Unicode 変換

Unicode 変換は、すべてのネイティブな文字セットについて行われます。2つのネイティブな文字セット間の Unicode 変換では、Unicode が中間文字セットとして使用されます。たとえば、サーバのデフォルトの文字セット (CP 437) とクライアントの文字セット (CP 860) との間で変換する場合、最初に CP 437 が Unicode に変換され、次に Unicode が CP 860 に変換されます。

Unicode 変換は、サーバのデフォルトの文字セットが UTF-8 またはネイティブな文字セットである場合に使用されます。Unicode 変換を使用するには、サーバ設定でそのことを明示的に指定してください (サーバのデフォルトの文字セットが UTF-8 でない場合)。

以前のバージョンの Adaptive Server では直接変換が使用されており、現在でも文字セット変換のデフォルトの方法となっています。ただし、Unicode 変換では、文字セット変換が簡単になり、複雑でなくなります。Adaptive Server 直接変換は引き続きサポートされていますが、Sybase では、すべての文字セットの変換を完全にサポートできるようにするために Unicode 変換も使用しており、新しい直接変換を追加する予定はありません。

Unicode 非文字の許可

15.7 よりも前のバージョンの Adaptive Server では、utf-8 デフォルト文字セット内の `unichar`、`univarchar`、`unitext`、`char`、`varchar`、および `text` データ型は Unicode 非文字を受け入れませんでした (コード・ポイントは永久に内部使用のために予約されています)。

Adaptive Server バージョン 15.7 では、`enable functionality group` 設定パラメータまたは `enable permissive unicode` 設定パラメータを有効にすることにより、Unicode 非文字を無視することができます。

この機能を有効にしない場合、Adaptive Server は以前のバージョンと同様にこれらの非文字を拒否します。

Unicode 標準の詳細については、Unicode Consortium Web サイト (<http://www.unicode.org/>) を参照してください。

この機能を有効にすると、次の内部の Unicode 非文字が検出されなくなります。

- パラメータ：
 - `univarchar` および `unitext` (UTF-16) データ型で示されるもの
 - `varchar` および `text` (UTF-8) データ型で示されるもの
 - 動的 SQL 文のパラメータ
 - パラメータ化した言語の文の入力
- サーバの文字セットが UTF-8 の場合の文字列リテラル
- サーバの文字セットに関係なく、エスケープされた文字列リテラル (プレフィクスが `U&` のもの)
- `unicar` (UTF-16) と `varchar` (UTF-8) 間の変換プロセス (両方向)

さらに、Unicode 非文字は、有効な Unicode 文字よりも前にソートされる比較などの単純な式に使用することができます。

以下の関数へのパラメータとして、Unicode 非文字を使用できます。

<code>ascii()</code>	<code>lower()</code>	<code>sortkey()</code>
<code>char_length()</code>	<code>ltrim()</code>	<code>soundex()</code>
<code>charindex()</code>	<code>patindex()</code>	<code>str_replace()</code>
<code>compare()</code>	<code>replicate()</code>	<code>stuff()</code>
<code>datalength()</code>	<code>reverse()</code>	<code>substring()</code>
<code>difference()</code>	<code>right()</code>	<code>upper()</code>
<code>left()</code>	<code>rtrim()</code>	<code>uscalar()</code>
<code>len()</code>		

注意 この機能は、`enable surrogate handling` 設定パラメータで有効にされた UTF-16 サロゲート処理に影響しません。『システム管理ガイド』の「設定パラメータ」を参照してください。

変換タイプの選択

実際のクライアント／サーバ・システムでどちらの変換が利用可能かを判断するには、[表 11-1 \(364 ページ\)](#) を参照してください。

非 Unicode クライアント／サーバ・システム

非 Unicode システムでは、サーバ／クライアントの文字セットはネイティブな文字セットなので、Adaptive Server 直接変換を使用できます。

しかし、いくつかの文字セットについては、Adaptive Server 直接変換が用意されていないことがあります。その場合は、Unicode 変換を使用してください。

- クライアント／サーバ・システムのすべての文字セットが、表 11-1 のカラム 1 にある場合は、Adaptive Server 直接変換を使用してください。すべての文字セットが、同じ言語グループに属している必要があります。
- クライアント／サーバ・システムの文字セットが、すべて表 11-1 のカラム 2 にある場合、またはカラム 1 と 2 の組み合わせである場合は、Unicode 変換を使用するようにサーバを設定してください。この場合も、すべての文字セットが同じ言語グループに属している必要があります。

たとえば、サーバのデフォルトの文字セットが CP 850 で、クライアントの文字セットが ISO 8859-1 と ROMAN 8 のいずれかであるとします。表 11-1 を参照すると、CP 850 とクライアントの文字セットの間では直接変換が行われることがわかります。次に、CP 1252 を使用するクライアントを追加したとします。CP 1252 と CP 850 (サーバのデフォルト文字セット)の間には直接変換がないので、CP 1252 と CP 850 の間の変換には Unicode 変換を使用する必要があります。文字セットが混在している (ある文字セットは Adaptive Server 直接変換を使用でき、ある文字セットは Unicode 変換を使用する必要がある) ときは、Adaptive Server 直接変換と Unicode 変換を組み合わせることを指定できます。

Unicode クライアント／サーバ・システム

サーバのデフォルトの文字セットとして Unicode UTF-8 を使用している場合は、変換はすべて UTF-8 と、クライアント・システムで使用されているネイティブ文字セットとの間で行われます。Unicode システムでは、Unicode 変換だけが使用されます。

表 11-1: 文字セットの変換方法

言語グループ	カラム 1 – Adaptive Server 直接変換と Unicode 変換	カラム 2 – Unicode 変換のみ
グループ 1	CP 437, CP 850, ISO 8859-1, Macintosh Roman	CP 860, CP 1252, ISO 8859-15, CP 863
グループ 2	CP 852, CP 1250, CP 8859-1, Macintosh Central European	ISO 8859-2
グループ 4	変換不要 (1 つの文字セットだけをサポート)	
グループ 5	CP 855, CP 866, CP 1251, ISO 8859-5, Koi8, Macintosh Cyrillic	
グループ 6		CP 864, CP 1256, ISO 8859-6
グループ 7	CP 869, CP 1253, GREEK8, ISO 8859-7, Macintosh Greek	
グループ 8		CP 1255, ISO 8859-8
グループ 9	CP 857, CP 1254, ISO 8859-9, Macintosh Turkish, TURKISH8	
グループ 101	DEC Kanji, EUC-JIS, Shift-JIS	CP 932

言語グループ	カラム 1 – Adaptive Server 直接変換と Unicode 変換	カラム 2 – Unicode 変換のみ
グループ 102		CP 936、EUC-GB、GB18303
グループ 103		Big 5、CP 950、EUC-CNS
グループ 104		EUCKSC、CP 949
グループ 105		CP 874、TIS 620
グループ 106	変換不要 (1 つの文字セットだけをサポート)	
Unicode	変換不要 (1 つの文字セットだけをサポート)	

サーバの設定

デフォルトでは、Adaptive Server は直接変換を使用して異なる文字セット間でデータを変換します。Unicode 変換を使用するには、`sp_configure` を使用して `enable unicode conversions` オプションを 1 または 2 に設定します。

- `sp_configure` “enable unicode conversions” を 1 に設定した場合

この設定では、Adaptive Server 直接変換または Unicode 変換を使用します。Adaptive Server は最初に、サーバの文字セットとクライアントの文字セットの間で使用できる Adaptive Server 直接変換があるかどうかをチェックします。直接変換がある場合は直接変換を使用し、ない場合は Unicode 変換を使用します。

クライアント/サーバ・システムの文字セットが、表 11-1 のカラム 1 と 2 の両方に分かれている場合に、この設定を使用してください。

- `sp_configure` “enable unicode conversions” を 2 に設定した場合

この設定では、Unicode 変換だけを使用します。Adaptive Server は、使用可能な Adaptive Server 直接変換があるかどうかをチェックしないで、Unicode 変換を使用します。

クライアント/サーバの変換によってデータ長の変更が生じる場合に、この設定を使用してください (「[変換とデータ長の変更](#)」(367 ページ) 参照)。

すべての文字セットが表 11-1 のカラム 2 に該当する場合は、`enable unicode conversions` を 2 に設定して、常に Unicode 変換が使用されるようにします。

Adaptive Server バージョン 15.0 以降では、`enable unicode conversions` のデフォルト値は 1 です。

サーバのデフォルトの文字セットが UTF-8 である場合は、サーバは自動的に Unicode 変換だけを使用します。

文字セット変換の有効化と無効化

接続を要求するクライアントは、その文字セットを Adaptive Server に知らせます。Adaptive Server はクライアントの文字セットをデフォルトの文字セットと比較し、2 つの名前が同一であれば、変換を要求しません。名前が異なる場合は、Adaptive Server のデフォルトの文字セットとクライアントの文字セットの間の変換をサポートしているかどうかを判断します。サポートしていない場合は、エラー・メッセージをクライアントに送り、ログイン処理を続行します。サポートしている場合は、文字セット変換が自動的に有効になります。サーバのデフォルトの文字セットが UTF-8 である場合は、Unicode 変換が自動的に使用されます。デフォルトがネイティブな文字セットの場合、ユーザが Unicode 変換を要求していないかぎり、サーバは Adaptive Server 直接変換を使用します。

文字セット変換は、サーバ・レベルで無効にできます。以下のような場合に無効にします。

- すべてのクライアントがサーバのデフォルトと同じ文字セットを使用しているため、変換する必要がない場合。
- クライアントの文字セットとサーバのデフォルトの文字セットとの変換がサポートされていない場合。
- コード化を変更しないで、データをサーバに保存する場合。

サーバ・レベルで文字セット変換を無効にするには、`disable character set conversion` パラメータを 1 に設定します。

また、クライアント・セッション内で `set char_convert` コマンドを使用することによって、文字セット変換を接続レベルでも制御できます。`set char_convert off` は、特定のクライアントとサーバの間の変換を無効にします。クライアントとサーバが同じ文字セットを使用している場合は変換が不要になるため、`set char_convert off` を使用してください。`set char_convert on` を実行すると、変換は再び有効になります。

変換できない文字

次の場合に、一部の文字が変換されないことがあります。

- 文字が、変換元の文字セットには存在している (コード化されている) が、ターゲットの文字セットには存在しない場合。たとえば、Macintosh の文字セット中にある OE の合字 (コード・ポイント 0xCE) です。この文字は、ISO 8859-1 文字セットには存在しません。Macintosh から ISO 8859-1 文字セットに変換するデータに OE の合字が存在すると、変換エラーが起こります。
- 文字が、変換元の文字セットにもターゲットの文字セットにも存在するが、その文字を表すためのバイト数が変換元の文字セットとターゲットの文字セットとで異なる場合。

たとえば、1 バイトのアクセント記号付き文字 (á や è など) は、UTF-8 では 2 バイト文字であり、2 バイトのタイ語の文字は UTF-8 では 3 バイト文字になります。この制限を回避するには、`enable unicode conversion` オプションを 1 または 2 に設定します。

文字セット変換のエラー処理

ある文字がクライアントの文字セットに存在してサーバの文字セットには存在しない、またはその逆の場合は、Adaptive Server の文字セット変換の変換エラーがレポートされます。Adaptive Server は、サーバへの入力時に正常に変換されたデータをクライアントが取り出すときに、クライアントの文字セットに正常に再変換できることを保証しなければなりません。そのためには、データベースに疑わしいデータが保管されることがないようにする必要があります。

データの入力中に変換エラーが発生すると、次のメッセージが生成されます。

```
Msg 2402, Severity 16 (EX_USER):
Error converting client characters into server's
character set. Some character(s) could not be converted.
```

変換エラーが発生すると、挿入文や更新文が含まれているクエリは実行できません。エラーが発生した場合は、データを見直して問題の文字を探し、置換してください。

Adaptive Server からクライアントにデータを送信するときに変換エラーが検出された場合は、疑わしい文字のバイトは ASCII の疑問符 (?) に置き換えられます。クエリ・バッチは最後まで実行されます。文が完了すると、Adaptive Server は次のメッセージを送信します。

```
Msg 2403, Severity 16 (EX_INFO):
WARNING! Some character(s) could not be converted into
client's character set. Unconverted bytes were changed
to question marks ('?').
```

変換とデータ長の変更

場合によっては、サーバの文字セットとクライアントの文字セットの間でデータを変換した結果、データ長に変更が生じることがあります。たとえば、一方のシステムの文字セットでは 1 つの文字を表すために 1 バイトを使用し、もう一方のシステムの文字セットでは 1 つの文字につき 2 バイトが必要になる場合に変更が生じます。

文字セット変換によってデータ長に変更が生じた場合、次の 2 つの状態が考えられます。

- 次の例のように、データ長が減少する。
 - ギリシャ語またはロシア語での、マルチバイトの UTF-8 からシングルバイトのギリシャ語またはロシア語文字セットへの変換
 - 日本語での、EUC-JIS の 2 バイト半角カタカナ文字から Shift-JIS のシングルバイトの文字への変換
- 次の例のように、データ長が増加する。
 - シングルバイトのタイ語文字から UTF-8 でのマルチバイトのタイ語文字への変換
 - Shift-JIS での日本語のシングルバイト文字から EUC-JIS での 2 バイト半角カタカナ文字への変換

システムとアプリケーションの設定

クライアント／サーバ・システムの中に UTF-8 を使用している部分がある場合や、日本語文字セットを使用している場合は、文字セット変換の結果としてデータ長の変更が生じる可能性があります。データ長の変更を処理できるようにサーバを設定してください。また、場合によっては、クライアントでもデータ長の変更を処理できるように設定する必要があります。

- 1 Unicode 変換を使用するようにサーバを設定します。「[サーバの設定](#)」(365 ページ)を参照してください。サーバとクライアントの間でデータ長が増加する場合は、手順 2 と 3 も行ってください。
- 2 クライアントは、11.1 以降の Open Client を使用している必要があります。クライアントが CS_LONGCHAR データを処理できることを、Open Client の `ct_capability` を使用して、接続時にサーバに知らせる必要があります。

`capability` パラメータは CS_DATA_LCHAR に設定し、`value` パラメータは CS_TRUE に設定してください。`connection` は、CS_CONNECTION 構造体へのポインタです。

```
CS_INT capval = CS_TRUE
ct_capability(connection, CS_SET, CS_CAP_RESPONS,
              CS_DATA_LCHAR, &capval)
```

- 3 変換によってデータ長が増加した場合、`char` および `varchar` のデータは、クライアントの文字セットに変換され、CS_LONGCHAR データとしてクライアントに送信されます。CS_LONGCHAR として受信したデータを抽出するようにクライアント・アプリケーションをコーディングする必要があります。

ユーティリティ・プログラムのための文字セットの指定

Sybase のユーティリティ・プログラムは、クライアント・プラットフォームのデフォルトの文字セットが、クライアントの使用している文字セットと同一だと見なします。ただし、クライアントの文字セットが、プラットフォームの文字セットと異なる場合もあります。このため、コマンド・ラインでクライアントの文字セットを指定しなければならないことがあります。isql、bcp、defncopy の各ユーティリティのコマンド・ラインのオプションとしてクライアントの文字セットを指定し、LANG 変数の設定や *locales.dat* の設定を一時的に無効にします。

-J *charset_name* (UNIX と PC の場合) を指定すると、クライアントの文字セットは *charset_name* であるとして設定されます。

クライアントの文字セットのコマンド・ライン・フラグを省略すると、プラットフォームのデフォルトの文字セットが使用されます。『ユーティリティ・ガイド』を参照してください。

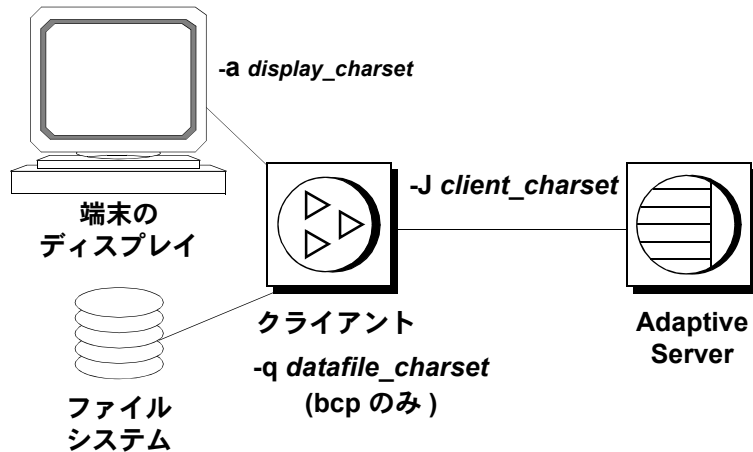
表示およびファイル文字セットのコマンド・ライン・オプション

この章では、クライアントと Adaptive Server の間の文字セットの変換について説明していますが、次の 2 つについても文字セットの変換が必要になることがあります。

- クライアントと端末間
- クライアントとファイル・システム間

図 11-4 は、スタンドアロン・ユーティリティ isql、bcp、defncopy で利用できるパスとコマンド・ライン・オプションを示しています。

図 11-4: 文字セットの変換が必要な場所



`-J` または `/clientcharset` コマンド・ライン・オプションによって、クライアントが Adaptive Server との文字データの送受信に使用する文字セットを指定します。

表示文字セットの設定

クライアントを実行する端末の文字セットがクライアントの文字セットとは異なる場合は、`-a` コマンド・ライン・オプションを使用してください。図 11-4 では、変換に必要な文字セット変換ファイル (`.xlt` ファイル) を指定するために、`-a` オプションと `-J` オプションを一緒に使用しています。

`-a` を使用するとき `-J` を省略できるのは、クライアントの文字セットがデフォルトの文字セットと同じ場合だけです。

ファイル文字セットの設定

クライアントの文字セットとは異なる文字セットを使用するファイル・システムとの間で文字データをコピーするために `bcp` を実行する場合は、`-q` コマンド・ライン・オプションを使用してください。図 11-4 では、変換に必要な文字セット変換ファイル (`.xlt` ファイル) を指定するために、`-q` オプションまたは `/filecharset` オプションと、`-J` オプションまたは `/clientcharset` オプションを一緒に使用しています。

この章では、システムの問題の診断および解決方法について説明します。

トピック名	ページ
Adaptive Server のエラー・メッセージ	371
Adaptive Server エラー・ロギング	375
Backup Server のエラー・ロギング	382
プロセスの強制終了	384
ハウスキーピング機能	388
サーバの停止	392
既知の問題についての情報	393

Adaptive Server のエラー・メッセージ

Adaptive Server は、問題を検出すると、次の内容を含むエラー・メッセージを表示します。

- エラー・メッセージをユニークに識別する「メッセージ番号」
- 問題の種類と重大度レベルを示す 10～24 の範囲の「重大度レベル番号」
- 「エラー・ステータス番号」。エラーが発生した Adaptive Server コードの行を特定できます。
- 「エラー・メッセージ」。発生した問題の内容を通知します。その問題の解決方法が提示されることもあります。

エラー・ログ・フォーマットの説明については、使用しているプラットフォームの『設定ガイド』を参照してください。

たとえば、存在しないテーブルにアクセスしようとすると、次のような情報が表示されます。

```
select * from publisher

Msg 208, Level 16, State 1:
publisher not found. Specify owner.objectname or use
sp_help to check whether the object exists (sp_help may
produce lots of output).
```

1つのクエリに対して複数のエラー・メッセージが発行される場合もあります。バッチまたはクエリで複数のエラーが発生すると、Adaptive Server は通常は最初のエラーだけをレポートします。あとのエラーは、次のバッチまたはクエリの実行時にレポートされます。

エラー・メッセージは `master..sysmessages` に保管されており、Adaptive Server のバージョンが新しくなるたびに更新されます (数千のローがあります)。次は、その最初の部分です (Adaptive Server のデフォルト言語が `us_english` の場合)。

```
select error, severity, description
from sysmessages
where error >=101 and error <=106
and langid is null
```

```
error severity description
-----
```

```
101      15 Line %d: SQL syntax error.
102      15 Incorrect syntax near '%.*s'.
103      15 The %S_MSG that starts with '%.*s' is too long.
          Maximum length is %d.
104      15 Order-by items must appear in the select-list if
          the statement contains set operators.
105      15 Unclosed quote before the character string '%.*s'.
106      16 Too many table names in the query. The maximum
          allowable is %d.
```

(6 rows affected)

`sysmessages` を問い合わせで、エラー・メッセージのカスタム・リストを生成できます。

- サーバで複数の言語がサポートされている場合、`sysmessages` には言語ごとに各メッセージが保管されています。`langid` カラムは、`us_english` では `NULL` で、サーバにインストールされている他の言語の場合は `syslanguages.langid` に一致します。
- `sqlstate` カラムは、ANSI SQL92 で定義されている、エラー条件と例外の `SQLSTATE` 値を保管します。
- 17000 以上のメッセージ番号は、システム・プロシージャのエラー・メッセージとメッセージ文字列です。

エラー・ログのフォーマット

Adaptive Server はエラー・メッセージに次のフォーマットを使用します。

```
instance_ID:thread_ID:family_ID:spid date time {server | kernel} message
```

各要素の意味は次のとおりです。

- *instance_ID* – クラスタ環境で実行しているインスタンス。Adaptive Server がノンクラスタード環境で実行している場合、この値は 00 です。
- *thread_ID* – エラーが発生したスレッド。常に 4 桁で表示されます (たとえば、10 は 0010 と表示されます)。

注意 プロセス・モードでは、Adaptive Server は各ログ・エントリに関連したエンジンを表示します。エンジン番号は 4 桁の数で表されます。オンラインのエンジンが 1 つだけの場合は、00 が表示されます。

- *family_ID* – (発生した) スレッドのファミリ ID。
 - 逐次処理では、00000 が表示されます。
 - 並列処理では、(発生した) スレッドの親の SPID 番号が表示されます。
- *spid* – (発生した) スレッドの SPID。
 - 逐次処理では、メッセージを生成したスレッドの SPID 番号が表示される。スレッドがシステム・タスクの場合は、SPID 値は 00000 になります。
 - 並列処理では、(発生した) スレッドの SPID 番号が表示されます。
- *date* – yyyy/mm/dd の形式の日付。これを利用すると、エラー・メッセージを日付順にソートできます。
- *time* – 24 時間形式で表された時刻。時刻には、秒および 1/100 秒までが含まれます。
- *server | kernel* – これは、Sybase 製品の保守契約を結んでいるサポート・センタ専用のエントリです。
- *message* – エラー・メッセージ。

次に例を示します。

```
00:0024:00000:00000:2010/04/27 10:28:07.82 kernel Thread 24 (LWP 24390) of Threadpool
syb_default_pool online as engine 0
```

- *instance_ID* = 0 – サーバはクラスタ環境用に設定されていません。
- *thread_ID* = 0024 – スレッド番号が 24 のため、Adaptive Server はエンジン 0 をオンラインにしました。
- *family_ID* = 00000 – プロセスは逐次モードで実行しています。

- `spid=00000` – これはシステム・タスクです。
- `date` と `time=2010/04/27 10:28:07.82`
- `server | kernel=kernel`

エラー・メッセージおよびメッセージ番号

メッセージ番号 (`error`) と言語 ID (`langid`) の組み合わせによって、各エラー・メッセージがユニークに識別されます。同じメッセージ番号で異なる言語 ID のメッセージは、翻訳されたメッセージであることを示します。

```
select error, description, langid
from sysmessages
where error = 101
```

error	description	langid
101	Line %d: SQL syntax error.	NULL
101	Ligne %1!: erreur de syntaxe SQL.	1
101	Zeile %1!: SQL Syntaxfehler.	2

(3 rows affected)

エラー・メッセージのテキストは、問題の説明です。この説明には、行番号、データベース・オブジェクトのタイプ (テーブル、カラム、ストアド・プロシージャなど) の参照、特定のデータベース・オブジェクト名などが含まれることもあります。

`sysmessages` の `description` フィールドには、パーセント記号 (%) に 1 文字以上の文字列が続くことがあります。これはプレースホルダの役割を果たすもので、Adaptive Server が問題を検出してそのエラー・メッセージを生成するときにはデータが代入されます。“%d” は数値のプレースホルダです。“%S_MSG” はデータベース・オブジェクトの種類を表すプレースホルダです。引用符で囲まれた“%. *s” は、特定のデータベース・オブジェクト名のプレースホルダです。表 12-1 (375 ページ) に、プレースホルダとその意味を示します。

たとえば、メッセージ番号 103 の `description` フィールドは次のとおりです。

```
The %S_MSG that starts with '%.*s' is too long. Maximum length
is %d.
```

実際に表示されるエラー・メッセージは次のようになります。

```
The column that starts with 'title' is too long. Maximum length
is 80.
```

Sybase 製品の保守契約を結んでいるサポート・センタにエラーをレポートするときは、番号、オブジェクト・タイプ、およびオブジェクト名もレポートしてください (「エラーのレポート」 (382 ページ) を参照)。

エラー・メッセージ・テキスト内の変数

表 12-1 は、エラー・メッセージ・テキストに使用される記号とその意味を示しています。

表 12-1: エラー・テキストに使用される記号

記号	意味
%d、%D	10 進数
%x、%X、%*x、%lx、%04x、%08lx	16 進数
%s	null で終了する文字列
%.*s、%*s、%*.s	文字列 (通常は特定のデータベース・オブジェクト名)
%S_type	Adaptive Server で定義されている構造体
%c	1 つの文字
%f	浮動小数点数
%ld	長い 10 進数
%lf	倍精度浮動小数点数

Adaptive Server エラー・ロギング

Adaptive Server から出力されるエラー・メッセージは、ユーザの画面だけに送信されます。

致命的なエラー・メッセージ (重大度レベル 19 以上) からのスタック・トレース、およびカーネルからのエラー・メッセージは、エラー・ログ・ファイルにも送信されます。このファイルの名前については、プラットフォームの『Adaptive Server Enterprise 設定ガイド』または『ASE ユーティリティ・ガイド』を参照してください。

注意 エラー・ログ・ファイルの所有者は、Adaptive Server をインストールしたユーザ (またはエラー・ログが削除された後に Adaptive Server を起動したユーザ) です。オペレーティング・システム・レベルでのエラー・ログのパーミッションまたは所有権に問題があると、Adaptive Server が正常に起動できないことがあります。

エラー・ログが存在しない場合は、Adaptive Server によってエラー・ログが作成されます。エラー・ログのロケーションは、起動時に `runserver` ファイル内の `errorlogfile` パラメータとして、またはコマンド・ラインで指定します。別のロケーションを選択しない場合は、Sybase インストーラ・ユーティリティは、`$$SYBASE/$SYBASE_ASEinstall` をエラー・ログのロケーションとして `runserver` ファイルを設定します。`runserver` ファイルまたはコマンド・ラインでロケーションを指定しなければ、エラー・ログは Adaptive Server を起動したディレクトリに作成されます。エラー・ログのロケーションの指定については、『ASE ユーティリティ・ガイド』の `dataserver` を参照してください。

注意 エラー・ログをすぐに参照できるように、Adaptive Server を常に同じディレクトリから起動するか、`runserver` ファイルまたはエラー・ログ・フラグを使用してください。

サーバを起動したときに、サーバ上の各データベースの起動およびリカバリが正常に行われたか失敗したかは、エラー・ログ内のメッセージによってわかります。エラー・ログ・ファイルには、以降の致命的エラーのメッセージとすべてのカーネル・エラー・メッセージが追加されていきます。古いメッセージや不要なメッセージを削除してエラー・ログのサイズを小さくするには、Adaptive Server の停止中に「削除」を行ってください。

重大度レベル

メッセージの重大度レベルは、Adaptive Server が検出した問題の種類と重大度を示すものです。整合性を保つため、Adaptive Server はエラー状態が発生したときに `sysmessages` からメッセージを表示しますが、処理は内部テーブルに従って行います。エラーに対応するメッセージの重大度がそれぞれ異なることがあるため、開発するアプリケーションまたはプロシージャで Adaptive Server のメッセージや重大度レベルを参照する場合に、想定した動作が異なっている可能性があります。

警告！ Adaptive Server のエラー番号に基づいて、独自のエラー番号とメッセージを作成できます (たとえば、Adaptive Server の値に 20,000 を追加します)。ただし、`sysmessages` システム・テーブルにある、Adaptive Server によって提供されるシステム・メッセージを変更することはできません。

ユーザ定義のエラー・メッセージを `sysusermessages` に追加するには、`sp_addmessage` を使用します。『リファレンス・マニュアル：プロシージャ』を参照してください。

重大度レベルが 17 以上の問題が発生した場合は、ユーザは必ずシステム管理者に連絡します。システム管理者には、そのエラーを解決し、その発生頻度を監視する責任があります。

その問題の影響がデータベース全体に及ぶ場合、システム管理者は、データベース一貫性チェッカ (dbcc) を使用して損傷の範囲を判断しなければならない場合があります。dbcc によって、削除すべきオブジェクトを識別できることがあります。dbcc によって損傷を修復できることもあります。データベースを再ロードする必要が生じることもあります。

詳細については、『システム管理ガイド 第 2 巻』の次の章を参照してください。

- dbcc については、『システム管理ガイド 第 2 巻』の「第 10 章 データベースの一貫性の検査」を参照してください。
- ユーザ・データベースのロードについては、『システム管理ガイド 第 2 巻』の「第 12 章 ユーザ・データベースのバックアップとリストア」を参照してください。
- システム・データベースのロードについては、『システム管理ガイド 第 2 巻』の「第 13 章 システム・データベースのリストア」を参照してください。

重大度レベル 10 ~ 18

重大度レベルが 10 ~ 16 のエラー・メッセージは、問題の原因がユーザ・エラーである場合に生成されます。これらは、ユーザが解決できるエラーです。重大度レベル 17 と 18 のエラーでは、ユーザのセッションは停止されません。

重大度レベルが 17 以上のエラー・メッセージが発生したときは、システム管理者またはデータベース所有者へのレポートが必要です。

重要度レベル 10：ステータス情報

重大度レベルが 10 のメッセージは、エラーではありません。特定のコマンドの実行後に追加情報を表示するためのもので、通常はメッセージ番号や重大度レベルは表示されません。たとえば `create database` コマンドを実行すると、要求した領域のどの程度が新しいデータベースに割り付けられたかを示すメッセージが表示されます。

重大度レベル 11：指定されたデータベース・オブジェクトが見つからない

重大度レベル 11 のメッセージは、コマンドで指定されたオブジェクトを Adaptive Server が見つけないことを示します。

主な原因には、データベース・オブジェクト名の入力ミス、オブジェクトの所有者名の指定もれ、現在のデータベースの誤認などがあります。オブジェクト名が正しく入力されていることを確認し、オブジェクトの所有者がユーザ自身または “dbo” でない場合は所有者名を指定します。また、現在のデータベースが正しいことを確認してください。

重大度レベル 12：不正データ型の検出

重大度レベル 12 のメッセージは、データ型に問題があることを示します。たとえば、カラムに正しくないデータ型の値を入力しようとした場合や、比較するカラムどうしのデータ型が異なり、互換性もない場合です。

比較の問題を解決するには、`select` 文で `convert` 関数を使用してください。詳細については、『リファレンス・マニュアル：ビルディング・ブロック』または『Transact-SQL ユーザーズ・ガイド』を参照してください。

重大度レベル 13：ユーザ・トランザクションの構文エラー

重大度レベル 13 のメッセージは、現在のユーザ定義トランザクションに問題があることを示します。たとえば、`begin transaction` を発行しないで `commit transaction` コマンドを発行した場合や、定義されていないセーブポイントにトランザクションをロールバックしようとした場合です（セーブポイント名の入カミスの場合もあります）。

重大度レベル 13 はデッドロックを示すこともあります。このとき、デッドロック・ビクティムとなったプロセスはロールバックされます。ユーザは、コマンドを初めから実行し直す必要があります。

重大度レベル 14：コマンド実行のパーミッションが不十分

重大度レベル 14 のメッセージは、コマンドの実行やデータベース・オブジェクトへのアクセスに必要なパーミッションがユーザにないことを示します。データベース・オブジェクトの所有者、データベースの所有者、またはシステム管理者に連絡して、問題のコマンドやオブジェクトを使用するためのパーミッションを取得してください。

重大度レベル 15：SQL 文の構文エラー

重大度レベル 15 のメッセージは、コマンドの構文に誤りがあることを示します。このエラー・メッセージのテキストには、誤りのある行番号およびその付近のキーワードが含まれています。

重大度レベル 16：その他のユーザ・エラー

重大度レベル 16 のエラー・メッセージのほとんどは、他のカテゴリに属さない、致命的でない誤りがあったことを示します。重大度レベルが 16 以上の場合は、ソフトウェアまたはハードウェアのエラーを示すこともあります。

たとえば、制約に違反する方法でビューを更新しようとした場合です。また、コマンドの中でカラム名を修飾せずに使用したとき、同じ名前のカラムがそのコマンドで使用する別のテーブルにもある場合に、このカテゴリのエラーとなります。Adaptive Server は、ユーザがどのテーブルを使用したいのかは判断できません。コマンドの構文と作業データベース・コンテキストを確認してください。

通常は重大度レベルが 17 以上となるメッセージでも、`dbcc checktable` または `dbcc checkalloc` によって発生した場合は、次のオブジェクトのチェックに進むことができるように重大度レベルは 16 となります。`dbcc` ユーティリティの実行時に重大度レベル 16 の 2500 ~ 2599 のエラー・メッセージが表示された場合は、『ASE トラブルシューティング&エラー・メッセージ・ガイド』を参照してください。

注意 重大度レベル 17 および 18 は、通常はエラー・ログにレポートされません。ユーザには、重大度レベル 17 および 18 のエラーが発生した場合はシステム管理者に連絡するように指示してください。

重大度レベル 17：リソース不足

重大度レベル 17 のエラー・メッセージは、コマンドの実行によって Adaptive Server のリソース不足が発生したか、システム管理者が設定した制限を超えたことを示します。ユーザは作業を続行できますが、実行できないコマンドもあります。

システムの制限には、同時にオープンできるデータベースの数や、Adaptive Server への接続数などがあります。制限はシステム・テーブルに保管され、`sp_configure` コマンドを使用してチェックできます。設定パラメータの変更の詳細については、「[第 5 章 設定パラメータ](#)」を参照してください。

重大度レベル 17 のエラー・メッセージが示すエラーが領域不足である場合は、データベース所有者がこの問題を解決できます。その他の重大度レベル 17 のエラー・メッセージについては、解決するのはシステム管理者です。

重大度レベル 18：致命的でない内部エラーが検出された

重大度レベル 18 のエラー・メッセージは、内部ソフトウェアのバグを示します。ただし、コマンドは最後まで実行され、Adaptive Server との接続は維持されます。ユーザは実行中の作業を続行できますが、実行できないコマンドもあります。重大度レベル 18 のエラーが発生する状況には、クエリのアクセス・パス決定の理由が正当なものではないことを Adaptive Server が検出した場合があります。

このようなメッセージを表示する問題が発生しても、ユーザの作業は中断されないため、その問題がユーザからレポートされないことがあります。ただし、システム管理者が問題をレポートできるようにするため、重大度レベル 18 以上のエラー・メッセージが表示された場合は必ずシステム管理者に報告するように、ユーザに指示を与えてください。

重大度レベル 19 ~ 26

致命的な問題が発生すると、重大度レベル 19 以上のエラー・メッセージが生成されます。このとき、Adaptive Server とユーザとの接続は切断されます (重大度レベルが高い場合は Adaptive Server が停止することがあります)。作業を続けるには、ユーザはクライアント・プログラムを再起動しなければなりません。

致命的なエラーが発生すると、プロセスは、停止する前にいったん静止状態となり、発生した問題についての情報を記録します。その後、プロセスは強制終了されて消滅します。

ユーザの接続が切断されたとき、ユーザは再接続して作業を再開できないこともあります。この範囲の重大度レベルの問題の影響は、1 ユーザの 1 プロセスにとどまらず、データベース内のプロセス全体に及ぶこともあり得ます。場合によっては、システム管理者が Adaptive Server を再起動する必要があります。これらの問題は、必ずと言うわけではありませんが、データベースやそのオブジェクトに損傷を与えることがあります。また逆に、データベースやそのオブジェクトの損傷によりこれらの問題が発生する場合もあります。ハードウェアの故障が原因の問題もあります。

カーネルからのエラー・メッセージは、エラー・ログ・ファイルに送信されます。

重大度レベル 19：リソースでの Adaptive Server の致命的なエラー

重大度レベル 19 のエラー・メッセージは、設定可能でない内部制限値を超えたこと、および Adaptive Server が正常にリカバリできないことを示します。ユーザは、Adaptive Server に再接続する必要があります。

重大度レベル 20：現在のプロセスでの Adaptive Server の致命的なエラー

重大度レベル 20 のエラー・メッセージは、コマンドのバグが Adaptive Server によって検出されたことを示します。この問題が影響するのは現在のプロセスだけです。また、データベースが損傷を受けている可能性はほとんどありません。dbcc 診断を実行してください。ユーザは、Adaptive Server に再接続する必要があります。

重大度レベル 21：データベース・プロセスでの Adaptive Server の致命的なエラー

重大度レベル 21 のエラー・メッセージは、現在のデータベースでのすべてのプロセスに影響を与えるバグが Adaptive Server によって検出されたことを示します。ただし、データベース自体が損傷を受けている可能性はほとんどありません。Adaptive Server を再起動して、dbcc 診断を実行してください。ユーザは、Adaptive Server に再接続する必要があります。

重大度レベル 22：Adaptive Server の致命的なエラー、テーブルの整合性の損傷

重大度レベル 22 のエラー・メッセージは、メッセージに示されたテーブルまたはインデックスが、以前にソフトウェアまたはハードウェアの問題によって損傷を受けたことを示します。

まず、Adaptive Server を再起動して `dbcc` を実行し、データベース内の他のオブジェクトも損傷を受けているかどうかを調べてください。`dbcc` のレポート内容にかかわらず、ディスク自体には問題がなく、キャッシュ内だけに問題が存在する場合があります。この場合は、Adaptive Server を再起動すれば問題は解消されます。

再起動しても問題が解決できない場合は、ディスクにも問題があります。エラー・メッセージに示されたオブジェクトを削除すると、問題が解決できる場合があります。たとえば、ノンクラスタード・インデックスの中に長さ 0 のローが見つかったというメッセージが表示された場合は、テーブル所有者はそのインデックスを削除して作り直します。

Adaptive Server は、リカバリ中に検出した疑わしいページまたはインデックスをオフラインにします。`sp_setsuspect_granularity` を使用して、リカバリ中に疑わしいと判断されたものがデータベース全体なのか、または個々のページのみなのかを確認してください。『リファレンス・マニュアル：プロシージャ』の「`sp_setsuspect_granularity`」を参照してください。

ユーザは、Adaptive Server に再接続する必要があります。

重大度レベル 23：致命的なエラー、データベースの整合性の損傷

重大度レベル 23 のエラー・メッセージは、以前にソフトウェアまたはハードウェアの問題によって発生した損傷が原因で、データベース全体の整合性が失われた可能性があることを示します。Adaptive Server を再起動して、`dbcc` 診断を実行してください。

データベース全体に問題の可能性があることをエラー・メッセージが示しているも、実際にはキャッシュだけの損傷で、ディスク自体には問題がないことがあります。その場合は、`startserver` を使用して Adaptive Server を再起動すると、問題が解決されます。

重大度レベル 24：ハードウェア・エラーまたはシステム・テーブルの損傷

重大度レベル 24 のエラー・メッセージは、メディア障害または(まれに) `sysusages` の矛盾を示します。この場合、システム管理者がデータベースを再ロードする必要があります。また、ハードウェアの購入元に連絡する必要がある場合もあります。

重大度レベル 25：Adaptive Server 内部エラー

重大度レベル 25 のエラーは画面には表示されません。Adaptive Server 内部エラーとして処理されます。

重大度レベル 26：ルール・エラー

重大度レベル 26 のエラー・メッセージは、内部的なロックまたは同期の規則に違反していることを示します。Adaptive Server をいったん停止して、再起動する必要があります。

エラーのレポート

Sybase サポート・センタにエラーをレポートするときは、次の情報をレポートしてください。

- メッセージ番号、重大度レベル番号、ステータス番号。
- エラー・メッセージに含まれているすべての数値、データベース・オブジェクトのタイプ、またはデータベース・オブジェクト名。
- メッセージが生成されたときの状況、つまりそのとき実行していたコマンド。エラー・ログのハードコピーを提出すると解決に役立ちます。

Backup Server のエラー・ロギング

Adaptive Server と同じように、Backup Server も、エラー・ログが存在していなければエラー・ログを作成します。エラー・ログのロケーションは、起動時に `runserver` ファイル内の `error_log_file` パラメータとして、またはコマンド・ラインで指定します。インストール時に別のロケーションが選択された場合を除いて、Sybase インストーラは、`$$SYBASE/install` をエラー・ログのロケーションとして `runserver` ファイルを設定します。`runserver` ファイルまたはコマンド・ラインでロケーションを指定しなければ、エラー・ログは Backup Server を起動したディレクトリに作成されます。エラー・ログに出力されるメッセージを制限するには、`backupserver -V` オプション (Windows NT では `bcksvr -V`) を使用します。詳細については、『ASE ユーティリティ・ガイド』の Backup Server に関する項を参照してください。

Backup Server のエラー・メッセージの形式は次のとおりです。

```
MMM DD YYYY: Backup Server:N.N.N.N: Message Text
```

Backup Server のメッセージ番号は、N.N.N.N のようにピリオドで区切られた 4 つの整数で構成されています。N.N.N の形式のメッセージは、Open Server によって送信されるものです。

Backup Server のエラー・メッセージの 4 つのコンポーネントは、*major.minor.severity.state* です。

- *major* コンポーネントは、通常、エラーが発生した Backup Server コードの機能領域を示します。
 - 1 - システム・エラー
 - 2 - Open Server のイベント・エラー
 - 3 - Backup Server のリモート・プロシージャ・コール・エラー
 - 4 - I/O サービス・レイヤ・エラー
 - 5 - ネットワークのデータ転送エラー
 - 6 - ボリューム処理エラー
 - 7 - オプションの解析エラー

major カテゴリ 1～6 のエラーの原因は、Backup Server の内部エラーまたはさまざまなシステムの問題が考えられます。*major* カテゴリ 7 のエラーの原因は、ほとんどがダンプ・コマンドまたはロード・コマンドに指定したオプションの誤りです。

- *minor* 番号は、*major* カテゴリの中で順番に割り当てられます。
- *severity* は、次のいずれかです。
 - 1 - 情報。ユーザのアクションは不要です。
 - 2, 3 - セッションに致命的な影響を与える可能性のある、予期されない状態が発生しました。エラーは、使用状況、環境、または内部論理によって、またはこれらの要因が組み合わされて発生しました。
 - 4 - Backup Server の実行に致命的な影響を与える、予期されない状態が発生しました。Backup Server をただちに終了してください。
- *state* コードは、コード内のエラー・レポートのインスタンスに 1 対 1 で対応しています。Backup Server のエラーについて Sybase の保守契約を結んでいるサポート・センタに連絡する必要がある場合、*state* コードはエラーの正確な原因を判断するのに役立ちます。

プロセスの強制終了

プロセスとは、Adaptive Server によって実行される実行単位のことです。プロセスには、開始時にユニークなプロセス ID が割り当てられ、この ID 番号を `spid` といいます。この処理を「初期化」と呼びます。この ID 番号は、各プロセスについてのその他の情報とともに `master.sysprocesses` に保管されます。並列プロセス環境で実行されるプロセスは、子プロセスを作成します。子プロセスには、それぞれ独自の `spid` があります。`spid` を作成して割り当てるプロセスには、Adaptive Server の起動、ログイン・タスク、チェックポイント、ハウスキーピング・タスクなどがあります。`sp_who` を実行すると、ほとんどの情報を確認できます。

シングルエンジンのサーバ上で `sp_who` を実行すると、`sp_who` プロセスが「実行中」であり、他のすべてのプロセスは「実行可能」またはスリープ状態にあることが表示されます。マルチエンジンのサーバでは、エンジンごとに「実行中」のプロセスが 1 つ存在します。

`kill` コマンドは、進行中のプロセスを強制終了します。プロセスを強制終了する必要がある状況としては、プロセスが他のユーザの作業を妨害しているときに、そのプロセスの実行責任者に連絡がとれない場合があります。プロセスが保持しているロックによって、データベース・オブジェクトへのアクセスがブロックされている場合や、多数のスリープ中のプロセスによって、使用可能なユーザ接続が占有されている場合もあります。システム管理者は、以下を待っているプロセスを含む、ほとんど実行中または「実行可能な」プロセスを強制終了できます。

- `waitfor` コマンドなどでのアラーム
- ネットワークの送信または受信
- ロック
- ファミリ内の他のプロセスからの同期メッセージ

終了していないすべてのトランザクションを正常にロールバックして、プロセスが使用しているすべてのシステム・リソースを解放できる場合にかぎり、プロセスの強制終了を実行できます。ファミリの一部であるプロセスでは、子プロセスを強制終了すると、ファミリ内のすべてのプロセスも強制終了されます。しかし、最も簡単な方法は親プロセスを強制終了することです。プロセスのファミリの場合は、子プロセスのステータスが `sync sleep` ならば、`kill` コマンドがより速く検出されます。

表 12-2 は、`sp_who` がレポートするステータス値と `kill` コマンドの効果が反映されるタイミングを示しています。

表 12-2: `sp_who` がレポートするステータスの値

ステータス	意味	kill コマンドの効果
<code>recv sleep</code>	ネットワーク読み込みの待機中。	即時
<code>send sleep</code>	ネットワーク送信の待機中。	即時
<code>alarm sleep</code>	次のようなアラーム待ち。 <code>waitfor delay "10:00"</code>	即時
<code>lock sleep</code>	ロック取得の待機中。	即時
<code>sync sleep</code>	ファミリー内の他のプロセスからの同期メッセージ待ち。	即時 (ファミリー内の他のプロセスも強制終了可能な状態にする)
<code>sleeping</code>	ディスク I/O またはその他のリソース待ち。おそらく、プロセスは実行中であるが非常に大量のディスク I/O が行われていることを示す。	通常、「ウェイクアップ」すると直ちに強制終了される。プロセスによってはウェイクアップしないものがあり、クリアするにはサーバの再起動が必要。
<code>runnable</code>	実行可能なプロセスのキュー内にある。	即時
<code>running</code>	サーバ・エンジンの 1 つで実行中。	即時
<code>infected</code>	サーバが重大なエラー状態を検出した。この状態が発生することはほとんどない。	<code>kill</code> コマンドの実行はすすめられない。プロセスをクリアするためにサーバを再起動しなければならない可能性が高い。
<code>background</code>	ユーザ・プロセスによってではなく Adaptive Server によって実行される、スレッシュールド・プロシージャなどのプロセス。	即時。細心の注意を払って <code>kill</code> を実行すること。バックグラウンド・プロセスを強制終了する前に、 <code>sysprocesses</code> を十分に確認することをすすめる。
<code>log suspend</code>	ログでラストチャンス・スレッシュールドに達したために中断されているプロセス。	即時

`kill` コマンドを発行できるのはシステム管理者だけです。このコマンドの使用パーミッションを譲渡することはできません。

構文は次のとおりです。

`kill spid`

2 つ以上のプロセスを同時に強制終了することはできませんが、連続する `kill` コマンドをバッチとして実行できます。次に例を示します。

```
1> kill 7
2> kill 8
3> kill 9
4> go
```

kill コマンドの取り消しはできません。また、ユーザ定義のトランザクション内で実行することはできません。spid は数値定数です。変数は使用できません。次は、sp_who の出力の例を示しています。

```
sp_who
fid  spid  status      loginame  origname  hostname      blk_spid  dbname
      tempdbname  cmd                block_xloid  threadpool
-----
0    1    recv sleep   howard    howard          svr30eng      0    master
      tempdb    AWAITING COMMAND  0    syb_default_pool
0    2    sleeping    NULL      NULL            0    master
      tempdb    NETWORK HANDLER  0    syb_default_pool
0    3    sleeping    NULL      NULL            0    master
      tempdb    DEADLOCK TUNE    0    syb_default_pool
0    4    sleeping    NULL      NULL            0    master
      tempdb    MIRROR HANDLER  0    syb_default_pool
0    5    sleeping    NULL      NULL            0    master
      tempdb    CHECKPOINT SLEEP 0    syb_default_pool
0    6    sleeping    NULL      NULL            0    master
      tempdb    HOUSEKEEPER      0    syb_default_pool
0    7    recv sleep   bill      bill            bigblue       0    master
      tempdb    AWAITING COMMAND 0    syb_default_pool
0    8    recv sleep   wilbur    wilbur          hazel         0    master
      tempdb    AWAITING COMMAND 0    syb_default_pool
0    9    recv sleep   joan      joan            luv2work      0    master
      tempdb    AWAITING COMMAND 0    syb_default_pool
0    10   running     foote     foote           svr47hum      0    master
      tempdb    SELECT           0    syb_default_master

(10 rows affected, return status = 0)
```

この例のプロセス 2～6 は、強制終了が不可能です。これらはシステム・プロセスです。ログイン名が NULL であることとホスト名がないことから、システム・プロセスであることがわかります。NETWORK HANDLER、MIRROR HANDLER、HOUSEKEEPER、および CHECKPOINT SLEEP (まれに CHECKPOINT のこともある) は常に sp_who 出力に表示されます。監査が使用可能であると、AUDIT PROCESS が表示されます。

プロセス 1、8、9、10 は、ステータス値が“recv sleep”、“send sleep”、“alarm sleep”、“lock sleep”であるので、強制終了が可能です。

`sp_who` の出力では、“recv sleep” が Adaptive Server ユーザのものでコマンドの結果を見るために一時停止させられているのか、あるいはユーザが PC などの端末を再起動したためにプロセスが中断状態で残っているのかはわかりません。このようなプロセスの情報を得るには、`sysprocesses` テーブルを問い合わせます。たとえば、次のクエリでは、プロセス 8 が使用しているホスト・プロセス ID とクライアント・ソフトウェアが表示されています。

```
select hostprocess, program_name
      from sysprocesses
     where spid = 8

hostprocess program_name
-----
3993          isql
```

このクエリと、`sp_who` の結果から得られたユーザおよびホストについての情報を参考にして、オペレーティング・システム・レベルでプロセスを追跡できます。

statusonly を指定した kill の使用

`kill ...statusonly` コマンドは、ロールバック・ステータスであるサーバ・プロセス ID (`spid`) の進捗状況についてレポートします。指定した `spid` は強制終了されません。`statusonly` レポートには、ロールバックの完了率と完了までにかかる推定時間 (秒単位) が表示されます。ロールバックの進捗状況を追跡するには、`kill...with statusonly` を複数回実行する必要があります。

```
kill spid with statusonly
```

`spid` は、停止するプロセスの番号です。

たとえば、次の例は `spid` 番号 13 のロールバック・プロセスについてレポートします。

```
kill 13 with statusonly
spid: 13 Transaction rollback in progress. Estimated rollback completion: 17% Estimated
time left: 13 seconds
```

`kill...statusonly` を発行したときに、指定した `spid` のロールバックが既に完了している場合、または指定した `spid` がロールバックできない場合は、`kill...statusonly` から次のメッセージが返されます。

```
Status report cannot be obtained. KILL spid:nn is not in
progress.
```

sp_lock によるブロック・プロセスの調査

前述の `sp_who` の他に、システム・プロシージャ `sp_lock` も、他のプロセスをブロックしているプロセスの特定に利用できます。`sp_who` のレポートの `blk_spid` カラムに、別のプロセスがブロックされて、ロック取得のために待機中であることが示されている場合に、`sp_lock` を実行すると、ブロックしているプロセスについての情報を表示できます。たとえば、前述の `sp_who` の出力にあるプロセス 10 は、プロセス 7 によってブロックされています。プロセス 7 についての情報を表示するには、次を実行します。

```
sp_lock 7
```

Adaptive Server でのロックの詳細については、『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』を参照してください。

ハウスキーピング機能

ハウスキーピング・タスクには、次の重要な機能があります。

- ハウスキーピングを構成するタスクには、ハウスキーピング・ウォッシュ、ハウスキーピング・ガーベジ・コレクション、ハウスキーピング・チャオアの 3 つがあります。`sp_who` はこの 3 つのタスクを、HK WASH、HK GC、HK CHORES として出力にすべて表示します。

```
sp_who
fid  spid  status      loginame      origname      hostname
blk_spid dbname tempdbname  cmd           block_xloid  threadpool
----  -
-----
-----
0      8      sleeping    NULL          NULL          luv2work
0      0      master      tempdb        HK WASH       0            syb_default_pool
0      9      sleeping    NULL          NULL          NULL
0      0      master      tempdb        HK GC         0            syb_default_pool
0      10     sleeping    NULL          NULL          NULL
0      0      master      tempdb        HK CHORES     0            syb_default_pool
```

- ハウスキーピング関連のすべてのシステム・タスクを自動的に再起動します。これらのシステム・タスクが予期せず終了しても、サーバを再起動する必要はありません。
- システム管理者は、すべてのハウスキーピング・タスクの優先度を変更できます。

`sp_showpsex` も `sp_who` と同様に、3 つのハウスキーピングの名前をすべて認識します。

`sp_who` と `sp_showpsex` の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

ハウスキーピング・ウォッシュ

バッファのウォッシュは必須のタスクではなく、有効にした場合、アイドル時にのみ実行されます。このタスクを実行しないようにするには、`housekeeper free write percent` 設定パラメータを使用します。ハウスキーピング・タスクのうち、ハウスキーピング・ウォッシュ・タスクだけがこの設定パラメータを使用します。

ハウスキーピング・チョア

ハウスキーピング・チョア・タスクは、アイドル時にのみ実行されます。共通の設定パラメータはありません。このタスクは、次のような雑多な処理を管理します。

- テーブル統計情報をフラッシュする。
- アカウント統計情報をフラッシュする。
- 分散トランザクションのタイムアウトを処理する。この処理を行わないようにするには、`dtm detach timeout period` 設定パラメータを使用します。
- ライセンスの使用状況を調べる。この処理を行わないようにするには、`license information` 設定パラメータを使用します。

ハウスキーピング・ガーベジ・コレクション

ガーベジ・コレクションには、「消極的」と「積極的」の2つの形態があります。これらの名前は、空ページを探す2種類のテストの性質を表しています。

- 消極的ガーベジ・コレクションは、コストをかけずに空ページを探すテストを指します。このテストは、トランザクションが長時間実行されている間は効果がないことがあり、空ページが累積していく可能性があります。消極的ガーベジ・コレクションは低コストで実行できますが、パフォーマンスを低下させることがあります。パフォーマンスに影響を与えるものには、割り付けられたテーブル領域の断片化と、クエリの実行時に調べる必要がある空ページの累積があります。
- 積極的ガーベジ・コレクションは、より高度な方法で空ページを検索するテストを指します。このテストは、ページ内の削除されたローを1つずつ調べて、そのローを削除したトランザクションがコミットされているかどうかを判断するので、消極的ガーベジ・コレクション・テストよりもコストが高くなります。

`delete` コマンドとハウスキーピング・ガーベジ・コレクション・タスクについて、積極的ガーベジ・コレクションを行うか消極的ガーベジ・コレクションを行うかを設定するには、`enable housekeeper GC` 設定パラメータを使用します。

積極的ハウスキーピング・ガーベジ・コレクションの場合は、タスクによってハウスキーピング対象リストを検査する頻度は、アプリケーションによって空ページが生成される速さと一致するように自動的に調整されます。

ユーザの優先度での実行

ハウスキーピング・ガーベジ・コレクション・タスクは通常、一般ユーザの優先度レベルで動作するので、CPU 時間に関して通常のユーザ・タスクと競合することになります。したがって、ハウスキーピングによる空きページの削除を上回る速さで、空きページのリストが増大することはありません。しかし、Adaptive Server がスレッド・モードに設定されている場合、`sp_bindexeclass 'sv'` オブジェクト・タイプを使用してハウスキーピング・ウォッシュ・タスクのサーバ・ワイドの優先度を変更します。優先度を `EC1`、`EC2`、`EC3` レベルに設定するか、または新しいユーザ作成の実行クラスを定義します。この例ではハウスキーピング・ウォッシュ・タスクの優先度を最高の `EC1` に設定します。

```
sp_bindexeclass 'HK WASH', 'sv', NULL, 'EC1'
```

Adaptive Server がスレッド・モードまたはプロセス・モードに設定されている場合、`sp_setpsex` を使用してセッションのハウスキーピング・タスクの優先度レベルを設定します。

この例では、現在のセッションのハウスキーピング・ウォッシュ・タスク (`spid 8`) の優先度レベルを `HIGH` に設定します。

```
sp_setpsex 8, 'priority', 'HIGH'
```

『リファレンス・マニュアル：プロシージャ』を参照してください。

enable housekeeper GC の設定

ガーベジ・コレクション・タスクに関して Adaptive Server を設定するときの構文は次のとおりです。

```
sp_configure "enable housekeeper GC", value
```

`enable housekeeper GC` 設定パラメータの有効な値は次のとおりです。

- 0 - ハウスキーピング・ガーベジ・コレクション・タスクは実行しませんが、`delete` コマンドによる消極的ガーベジ・コレクションは実行できるようにします。`reorg reclaim_space` を使用して、空ページの割り付けを解除する必要があります。これは、パフォーマンスへの影響が最も少なく、最も低コストのオプションですが、累積した空ページの量が増えるとパフォーマンス上の問題が発生する可能性があります。この値を使用することはおすすめしません。

- 1 – ハウスキーピング・ガーベジ・コレクション・タスクと `delete` コマンドの両方で、消極的ガーベジ・コレクションを実行できます。これはデフォルトの値です。アプリケーションで許容される以上の空ページが累積する場合は、オプション 4 または 5 の使用を検討してください。 `optdiag` ユーティリティを使用すると、空ページの統計情報を取得できます。
- 2 – 今後のために予約済み。
- 3 – 今後のために予約済み。
- 4 – ハウスキーピング・ガーベジ・コレクション・タスクと `delete` コマンドの両方で、積極的ガーベジ・コレクションを実行できます。このオプションを選択すれば効果が最も高くなりますが、`delete` コマンドのコストは最も高くなります。このオプションは、データオンリーロック・テーブルに対する一連の削除を 1 つのバッチで実行する場合に理想的です。
- 5 – ハウスキーピング・タスクでは積極的ガーベジ・コレクションを実行でき、`delete` では消極的ガーベジ・コレクションを実行できます。オプション 4 を選択した場合よりも、削除のコストは低くなります。このオプションは、同時トランザクションによって削除が行われる場合に適しています。

たとえば、次のように入力します。

```
sp_configure "enable housekeeper GC", 4
```

reorg コマンドの使用法

ガーベジ・コレクションの効果が最も高くなるのは、`enable housekeeper GC` を 4 または 5 に設定した場合です。このパラメータ値を 5 に設定することをおすすめしますが、パフォーマンスを考慮する上でこのパラメータを 4 または 5 に設定できないこともあります。その場合は、空ページが累積したときに、影響を受けているテーブルに対して `reorg` を実行してください。空ページに関する統計情報を取得するには、`optdiag` ユーティリティを使用します。

サーバが停止またはクラッシュすると、ハウスキーピング・ガーベジ・コレクション・タスクがまだ処理していないページ割り付け解除要求は失われます。このような、空ではあるけれどもハウスキーピング・ガーベジ・コレクション・タスクが割り付けを解除していないページは、`reorg` によって削除されるまでは、割り付け済みのままになります。

『システム管理ガイド 第 2 巻』の「第 9 章 `reorg` コマンドの使用法」を参照してください。

サーバの停止

システム管理者は、Adaptive Server または Backup Server を停止させることができます。使用する構文は次のとおりです。

```
shutdown [backup_server_name] [with {wait|nowait}]
```

`shutdown` コマンドのデフォルトは `with wait` です。したがって、`shutdown` と `shutdown with wait` の結果はまったく同じです。

Adaptive Server の停止

サーバ名を指定しないで `shutdown` を実行した場合は、使用中の Adaptive Server が停止します。`shutdown` コマンドが発行されると、Adaptive Server は次の処理を行います。

- 1 システム管理者以外はログインできないようにします。
- 2 個々のデータベースでチェックポイントを実行して、変更されたページをメモリからディスクにフラッシュします。
- 3 現在実行中の SQL 文やプロシージャの終了を待ちます。

このようにして、`shutdown` は Adaptive Server の再起動時に自動リカバリが行わなければならない作業量を最小にします。

`with nowait` オプションを指定した場合は、Adaptive Server はただちに停止します。ユーザ・プロセスはアポートされ、`shutdown with nowait` の後のリカバリの時間は長くなります。`shutdown with nowait` コマンドを発行する前に `checkpoint` コマンドを発行すると、リカバリ時間を短縮するのに役立ちます。

Backup Server の停止

Backup Server を停止するには、その Backup Server の名前を含めます。

```
shutdown SYB_BACKUP
```

デフォルトは `with wait` であるため、進行中のすべてのダンプやロードが終了してから Backup Server のプロセスが停止します。`shutdown` コマンドが発行された後は、その Backup Server で新しいダンプ・セッションやロード・セッションを開始することはできません。

使用している Adaptive Server からアクセスできる Backup Server の名前を調べるには、`sp_helpserver` を実行します。`name` カラムの値を `shutdown` コマンドで使用してください。Backup Server を停止できるのは、次の状態のときだけです。

- 使用している Adaptive Server の `syssservers` にリストされている
- 実行するユーザのローカルの `interfaces` ファイルにリストされている

Backup Server を `syssservers` に追加するには、`sp_addserver` を使用します。

アクティブなダンプおよびロードのチェック

`shutdown` コマンドを実行する前に Backup Server 上のアクティビティを確認するには、その Backup Server 上で `sp_who` コマンドを実行します。

```
SYB_BACKUP...sp_who
spid      status      loginame      hostname      blk      cmd
-----
1         sleeping    NULL          NULL          0        CONNECT HANDLER
2         sleeping    NULL          NULL          0        DEFERRED HANDLER
3         runnable    NULL          NULL          0        SCHEDULER
4         runnable    NULL          NULL          0        SITE HANDLER
5         running     sa            heliotrope    0        NULL
```

Backup Server での `nowait` の使用

`shutdown backup_server with nowait` コマンドは、現在のアクティビティに関係なく Backup Server を停止します。これは、重大な問題が発生した場合にだけ使用してください。これを使用すると、ダンプまたはロードが不完全になったり、一貫性のない状態で残ることがあります。

ログまたはデータベースのダンプの実行中に `shutdown with nowait` を使用する場合は、ダンプ終了を示すメッセージを確認してください。このメッセージを受け取っていない場合、またはダンプが終了したかどうかは確かでない場合は、次回のダンプではトランザクション・ダンプではなく、`dump database` を使用してください。このようにすれば、信頼すべきダンプの一貫性が失われているという疑いを持たなくて済みます。

`shutdown with nowait` を実行したときに何らかのロードが実行中であり、ロードの終了を示すメッセージを受け取っていない場合は、そのデータベースに対して `load transaction` コマンドを発行できなくなることがあります。そのデータベースを使用する前に、完全なデータベース一貫性チェック (`dbcc`) を実行してください。 `load database` から始まる一連のロード・コマンドをすべて発行し直さなければならない場合があります。

既知の問題についての情報

『リリース・ノート』には、Adaptive Server と Backup Server に関する既知の問題や互換性についての重要な情報があります。あらかじめ『リリース・ノート』を参照しておくことによって、このような問題の解決に費やす時間を節約し、無用な推測を避けることができます。

索引

記号

- “ ” (引用符)
 - パラメータ値を囲む 10
- ?? (疑問符)
 - 疑わしい文字 367
- % (パーセント記号)
 - エラー・メッセージのプレースホルダ 374
- @@char_convert グローバル変数 355
- @@client_csexpansion グローバル変数 355
- @@client_csid グローバル変数 355
- @@client_csname グローバル変数 355
- @@langid グローバル変数 357
- @@language グローバル変数 357
- @@max_connections グローバル変数 209
- @@maxcharlen グローバル変数 355
- @@ncharsize グローバル変数 355

数字

- 2 フェーズ・コミット
トランザクション 26
- 7 ビット ASCII 文字データ、文字セット変換 361

A

- abstract plan cache 設定パラメータ 76
- abstract plan dump 設定パラメータ 76
- abstract plan load 設定パラメータ 77
- abstract plan replace 設定パラメータ 77
- Adaptive Server への接続 14
- additional network memory 設定パラメータ 77
- allow backward scans 設定パラメータ 80
- allow nested triggers 設定パラメータ 81
- allow procedure grouping 設定パラメータ 81
- allow remote access 設定パラメータ 81
- allow resource limits 設定パラメータ 82
- allow sendmsg 設定パラメータ 82
- allow sql server async i/o 設定パラメータ 83
- allow updates to system tables 設定パラメータ 12, 83

- allow updates 設定パラメータ (現在の allow updates to system tables) 12
- alter database コマンド
 - システム・テーブル 280
 - データベース・デバイスの省略 306, 307
- ASCII 文字
 - 文字セット変換 361
- audit queue size 設定パラメータ 85
- auditing 設定パラメータ 86
- automatic cluster takeover 設定パラメータ 86
- average cap size 設定パラメータ 85

B

- Backup Server
 - tape retention in days 設定パラメータ 260
 - エラー・メッセージ 382
 - 停止 392
- bcp (バルク・コピー・ユーティリティ)
 - ソート順の変更 344
 - 文字セット変換 369
- Big 5
 - CP 950 との類似性 323

C

- caps per ccb 設定パラメータ 87
- charset.loc ファイル 354
- charsets ディレクトリ 354
- checktable オプション、dbcc 349
- CIPC large message pool size 設定パラメータ 88
- CIPC regular message pool size 設定パラメータ 89
- cis bulk insert batch size 設定パラメータ 89
- cis connect timeout 設定パラメータ 90
- cis cursor rows 設定パラメータ 90
- cis idle connectin timeout 設定パラメータ 91
- cis packet size 設定パラメータ 91
- cis rpc handling 設定パラメータ 92
- Closed Problem Reports 393
- cluster heartbeat interval 設定パラメータ 92

索引

cluster heartbeat retries 設定パラメータ 93
cluster vote timeout 設定パラメータ 93, 94
cntrltype オプション
 disk init 305
column default cache size 設定パラメータ 94
common.loc ファイル 356
compression info pool size 設定パラメータ 95
compression memory size 設定パラメータ 94, 95
configuration file 設定パラメータ 95, 117, 244, 245
cost of a cpu unit 設定パラメータ 97
cost of a logical io 設定パラメータ 96
cost of a physical io 設定パラメータ 96
CP 1252
 ISO 8859-1 との類似性 322
CP 950
 Big 5 との類似性 323
cp437 文字セット 103
cp850 文字セット 103
CPR ファイル 393
cpu accounting flush interval 設定パラメータ 97
cpu grace time 設定パラメータ 98
create database コマンド
 default database size 設定パラメータ 104
 model データベース 23
 システム・テーブル 7
 データベース・デバイスの省略 306, 307
create index コマンド 275, 281
create procedure コマンド 12
create table コマンド 275
cs_connection コマンド、number of user connections 210
current audit table 設定パラメータ 99

D

database size 設定変数 50
dbcc (データベース一貫性チェック) 39
 使用するとき 381
 データベースの損傷 377, 381
DB-Library プログラム
 number of user connections 210
“dbo” ユーザ名 2, 4
dbprocess コマンド、number of user connections 210
deadlock checking period 設定パラメータ 100
deadlock pipe active 設定パラメータ 101
deadlock pipe max messages 設定パラメータ 101
deadlock retries 設定パラメータ 102
deckanji 文字セット 103
default character set id 設定パラメータ 103

default database size 設定パラメータ 104
default exp_row_size percent 設定パラメータ 104
default fill factor percent 設定パラメータ 105
default language id 設定パラメータ 106
default network packet size 設定パラメータ 106
default sortorder id 設定パラメータ 108
default XML sortorder 設定パラメータ 108
default セグメント 277
defaulton | defaultoff オプション、sp_diskdefault 307
defncopy ユーティリティ・コマンド
 『ASE ユーティリティ・ガイド』参照
 文字セット変換 369
delete コマンド 389
disable character set conversions 設定パラメータ 109
disable disk mirroring 設定パラメータ 110
disable varbinary truncation 設定パラメータ 110
disk i/o structures 設定パラメータ 111
disk init コマンド 274, 279, 280, 298–305
disk mirror コマンド 275
disk reinic コマンド
 「disk init コマンド」参照
disk resize 274, 308
 構文 309
 最小サイズ 308
 使用 309
 ディスク領域の不足 310
 デバイスの縮小 310
 ミラーリング 309
DMA object pool size 設定パラメータ 111
drop logins オプション、sp_dropserver 289
dsync オプション
 disk init 306
dtm detach timeout period 設定パラメータ 112, 389
dtm lock timeout period 設定パラメータ 112
dump database コマンド
 disk init 298
 master データベース 38
 model データベース 23
dump on conditions 設定パラメータ 113
dynamic allocation on demand 設定パラメータ 114

E

enable backupserver HA 設定パラメータ 114
enable cis 設定パラメータ 115, 117, 118, 130
enable DTM 設定パラメータ 116
enable encrypted columns 設定パラメータ 116
enable functionality group 設定パラメータ 118
enable HA 設定パラメータ (高可用性向け) 119

enable housekeeper GC 設定パラメータ 120, 390
 enable hp posix async i/o 設定パラメータ 122
 enable i/o fencing 設定パラメータ 122
 enable inline default sharing 設定パラメータ 119, 127
 enable java 設定パラメータ 117, 123
 enable job scheduler 設定パラメータ 123
 enable ldap user auth 設定パラメータ 124
 enable literal autoparam 設定パラメータ 124
 enable logins during recovery 設定パラメータ 124, 125
 enable merge join 設定パラメータ 125
 enable metrics capture 設定パラメータ 125
 enable monitoring 設定パラメータ 126
 enable pam user auth 設定パラメータ 126
 enable query tuning mem limit 設定パラメータ 128
 enable real time messaging 設定パラメータ 128
 enable rep agent threads 設定パラメータ 129
 enable row level access control 設定パラメータ 129
 enable semantic partitioning 設定パラメータ 129
 enable sort-merge join and JTC 設定パラメータ 130
 enable SQL debugger 設定パラメータ 130
 enable stmt cache monitoring 設定パラメータ 131
 enable surrogate processing 設定パラメータ 131
 enable unicode conversion 設定パラメータ 132
 enable unicode conversions 設定パラメータ 365
 enable unicode normalization 設定パラメータ 132
 enable webservices 設定パラメータ 133
 enable xact coordination 設定パラメータ 133
 enable xml 設定パラメータ 134
 engine memory log size 設定パラメータ 134
 errorlog pipe active 設定パラメータ 134
 errorlog pipe max messages 設定パラメータ 135
 esp execution priority 設定パラメータ 135
 esp execution stacksize 設定パラメータ 136
 esp unload dll 設定パラメータ 136
 eucjis 文字セット 103
 event buffers per engine 設定パラメータ 137
 event log computer name 設定パラメータ 138
 event logging 設定パラメータ 138
 executable code size + overhead 設定パラメータ 139
 extended cache size 設定パラメータ 139

F

FIPS login password encryption 設定パラメータ 140
 fix_text オプション、dbcc 350

G

global async prefetch limit 設定パラメータ 140
 global cache partition number 設定パラメータ 141
 guest ユーザ
 作成 50
 サンプル・データベース 27
 データベース 50

H

heap memory per user 設定パラメータ 141
 histogram tuning factor 設定パラメータ 142
 housekeeper free write percent 設定パラメータ 143, 389

I

i/o accounting flush interval 設定パラメータ 145
 i/o batch size 設定パラメータ 146
 i/o polling process count 設定パラメータ 146
 IBM 文字セット 103
 identity burning set factor 設定パラメータ 147
 identity grab size 設定パラメータ 148
 identity reservation size 設定パラメータ 149
 idle migration timeout 設定パラメータ 149
 ID、ユーザ
 システム・プロシージャ 12
installhasvss スクリプト 120
insthasv スクリプト 120
 interfaces ファイル 14
 ISO 8859-1
 CP 1252 との類似性 322
 iso_1 文字セット 103
 isql ユーティリティ・コマンド
 number of user connections 210
 システム管理 5
 ステータスおよび情報メッセージ 377
 パスワード 294
 文字セット変換 369

J

Java 設定パラメータ 245
 job scheduler interval 設定パラメータ 150
 job scheduler tasks 設定パラメータ 150
 js job output width 設定パラメータ 151

索引

K

kernel mode 設定パラメータ 151
kernel resource memory 設定パラメータ 152
kill statusonly パラメータ 387
kill コマンド 384–388
kill コマンド、変更 387

L

LDAP

interfaces ファイルとの比較 17
 アクセス制限 16
 定義 15
 複数のディレクトリ・サービス 17
libtcl.cfg ファイル 15
libtcl.cfg ファイルのディレクトリ・サービス 15
license information 設定パラメータ 152
license information、設定パラメータ 389
local オプション、*sp_addserver* 286
locales ディレクトリ 339
locales.dat ファイル 356
lock address spinlock ratio 設定パラメータ 153
lock hashtable size 設定パラメータ 154
lock scheme 設定パラメータ 155
lock shared memory 設定パラメータ 155
lock spinlock ratio 設定パラメータ 156
lock table spinlock ratio 設定パラメータ 156
lock timeout pipe active 設定パラメータ 157
lock wait period 設定パラメータ 158
log audit logon failure 設定パラメータ 158
log audit logon success 設定パラメータ 159
log on オプション
 create database 280
logsegment ログ記憶領域 277

M

Macintosh 文字セット 103, 366
master データベース 7, 21–22, 37
 sysdevices テーブル 305
 オプション設定の変更 311
 作成 277
 システム・テーブルのキー 8
 「ディスク・ミラーリング」「システム・テーブル」
 参照
 バックアップ 37, 50
max async i/os per engine 設定パラメータ 159

max async i/os per server 設定パラメータ 160
max buffers per lava operator 設定パラメータ 175
max cis remote connections 設定パラメータ 162
max concurrently recovered db 設定パラメータ 162, 188
max memory 設定パラメータ 163
max native threads per engine 設定パラメータ 164
max nesting level 設定パラメータ 164
max network packet size 設定パラメータ 165
max number network listeners 設定パラメータ 167
max online engines 設定パラメータ 168
max online Q engines 168
max parallel degree 設定パラメータ 169
max pci slots 設定パラメータ 170
max repartition degree 設定パラメータ 171
max resource granularity 設定パラメータ 172
max roles enabled per user 設定パラメータ 182
max scan parallel degree 設定パラメータ 172
max SQL text monitored 設定パラメータ 173
max transfer history 設定パラメータ 174
maximum dump conditions 設定パラメータ 174
mci memory size 設定パラメータ 221
memory alignment boundary 設定パラメータ 177
memory per worker process 設定パラメータ 177
messaging memory 設定パラメータ 178
metrics elap max 設定パラメータ 178
metrics exec max 設定パラメータ 178
metrics lio max 設定パラメータ 179
metrics pio max 設定パラメータ 179
Microsoft 文字セット 103
min pages for parallel scan 179
minimum pages for a parallel scan 設定パラメータ 179
minimum password length 設定パラメータ 180
mnc_full_index_filter 設定パラメータ 181
model データベース 50
model データベース 23
 サイズ 104, 300
 作成 277
 システム・テーブルのキー 8
MSDTC 116
msg confidentiality reqd 設定パラメータ 182
msg integrity reqd 設定パラメータ 182

N

nested trigger 設定パラメータ (現在の allow nested triggers) 81
net password encryption reqd 設定パラメータ 183
net password encryption オプション 288
number of alarms 設定パラメータ 183
number of aux scan descriptors 設定パラメータ 184

number of backup connections 設定パラメータ 187
 number of cbs 設定パラメータ 187
 number of devices 設定パラメータ 188
 number of disk tasks 設定パラメータ 189
 number of dtx participants 設定パラメータ 190
 number of histogram steps 設定パラメータ 192
 number of index trips 設定パラメータ 193
 number of large i/o buffers 設定パラメータ 194
 number of locks 設定パラメータ 195
 number of mailboxes 設定パラメータ 196
 number of messages 設定パラメータ 196
 number of network tasks 設定パラメータ 197
 number of oam trips 設定パラメータ 197
 number of open databases 設定パラメータ 198
 number of open indexes 設定パラメータ 200
 number of open objects 設定パラメータ 201
 number of pre-allocated extents 設定パラメータ 204
 number of Q engines at startup 206
 number of remote connections 設定パラメータ 206
 number of remote logins 設定パラメータ 191, 207
 number of remote sites 設定パラメータ 207
 number of sort buffers 設定パラメータ 207
 number of user connections 設定パラメータ 72, 208–210
 number of worker processes 設定パラメータ 211

O

o/s file descriptors 設定パラメータ 212
 object lockwait timing 設定パラメータ 212
 open index hash spinlock ratio 設定パラメータ 213
 open index spinlock ratio 設定パラメータ 213
 open object spinlock ratio 設定パラメータ 214
 optimization timeout limit 設定パラメータ 215, 216

P

page lock promotion HWM 設定パラメータ 216
 page lock promotion LWM 設定パラメータ 217, 236
 page lock promotion PCT 設定パラメータ 218
 partition groups 設定パラメータ 220
 partition spinlock ratio 設定パラメータ 220
 per object statistics active 設定パラメータ 221
 per object statistics active 設定パラメータ 221
 performance monitoring option 設定パラメータ 223
 permission cache entries 設定パラメータ 224
 plan text pipe active 設定パラメータ 225
 plan text pipe max messages 設定パラメータ 225
 print deadlock information 設定パラメータ 225

print recovery information 設定パラメータ 226
 process wait events 設定パラメータ 228
 prod-consumer overlap factor 228
 public メンバシップ 51
 pubs2 データベース
 image 情報 28
 管理 27
 pubs3 データベース
 管理 27

Q

quorum heartbeat interval 設定パラメータ 228
 quorum heartbeat retries 設定パラメータ 229
 quoted identifier enhancement 設定パラメータ 229

R

read committed with lock 設定パラメータ 230
 read only データベース・オプション 348
 recovery interval in minutes 設定パラメータ 230–232
 長時間実行トランザクション 231
 remote server pre-read packets 設定パラメータ 233
 reorg reclaim_space コマンド 390
 reorg コマンド
 手動で実行 391
 reorg コマンドの手動実行 391
 restricted decrypt permission 設定パラメータ 233
 retaindays オプション
 dump database 260
 dump transaction 260
 Roman8 文字セット 103
 row lock promotion HWM 設定パラメータ 235
 row lock promotion LWM 設定パラメータ 236
 row lock promotion PCT 設定パラメータ 237
 RPC。「リモート・プロシージャ・コール」参照
 rtm thread idle wait period 設定パラメータ 237
 runnable process search count 設定パラメータ 238

S

secure default login 設定パラメータ 240
 segmap カラム、sysusages テーブル
 変更するプロシージャ 280
 select for update 設定パラメータ 241
 select into/bulkcopy/pllsort データベース・オプション
 model データベース 23

索引

- select on syscomments.text column 設定パラメータ 241
- send doneinproc tokens 241
- server.loc ファイル 356
- server_name.cfg、設定ファイルのデフォルト名 58
- session migration timeout 設定パラメータ 242
- set スイッチ
 - 現在の設定の表示 314
- shared memory starting address 設定パラメータ 243
- shutdown コマンド 392–393
- size of auto identity column 設定パラメータ 243
- size of global fixed heap 設定パラメータ 244
- size of process object fixed heap 設定パラメータ 244
- size of shared class heap 設定パラメータ 245
- size of unilib cache 設定パラメータ 245
- sjis (シフト JIS) 文字セット。「日本語文字セット」参照
- sp_addlanguage システム・プロシージャ 352
- sp_addremotelogin システム・プロシージャ 290–292
- sp_addsegment システム・プロシージャ
 - sysusages 280
- sp_addserver システム・プロシージャ 285–287
- sp_adduser システム・プロシージャ 23
- sp_configure システム・プロシージャ 62
 - 「個々の設定パラメータ名」参照
 - リモート・ログイン 295
- sp_countmetadata システム・プロシージャ 199, 200, 202, 203
- sp_dboption システム・プロシージャ 311–313
- sp_deviceatrr システム・プロシージャ 274, 303
- sp_diskdefault システム・プロシージャ 274, 307–308
- sp_dropdevice システム・プロシージャ 307
- sp_dropremotelogin システム・プロシージャ 290
- sp_dropsegment システム・プロシージャ
 - sysusages 280
- sp_dropserver システム・プロシージャ 289
- sp_extendsegment システム・プロシージャ
 - sysusages 280
- sp_helpconfig システム・プロシージャ 198, 200, 201
- sp_helpdb システム・プロシージャ 11
 - データベース・オプション情報 313
- sp_helpdevice システム・プロシージャ 11, 305
- sp_helpindex システム・プロシージャ 11
- sp_helpjoins システム・プロシージャ 8
- sp_helpkey システム・プロシージャ 8
- sp_helppremotelogin システム・プロシージャ 295
- sp_helpserver システム・プロシージャ 289
- sp_helpstext システム・プロシージャ 11
- sp_indsuspect システム・プロシージャ 348, 349
- sp_modifylogin システム・プロシージャ 347
- sp_monitorconfig システム・プロシージャ
 - number of open databases の設定 199
 - number of open indexes の設定 201
 - number of open objects の設定 202, 204
- sp_remoteoption システム・プロシージャ 294
- sp_serveroption システム・プロシージャ 287
- sp_showpsexec システム・コマンド、ハウスキーピングの出力 388
- sp_who、ハウスキーピングの出力 388
- #spdevtab テンポラリー・テーブル 11
- #spindtab テンポラリー・テーブル 11
- SPR ファイル 393
- sproc optimize timeout limit 設定パラメータ 246
- spt_committab テーブル 11
- spt_monitor テーブル 11
- spt_values テーブル 11
- SQL batch capture 設定パラメータ 246
- sql server clock tick length 設定パラメータ 247
- sql text pipe active 設定パラメータ 248
- sql text pipe max messages 設定パラメータ 248
- .srt ファイル 354
- srvname カラム、syssservers テーブル 287
- srvnetname カラム、syssservers テーブル 287
- stack guard size 設定パラメータ 249
- stack size 設定パラメータ 252
- start mail session 設定パラメータ 253
- start xp server during reboot 設定パラメータ 253
- startup delay 設定パラメータ 254
- statement pipe active 設定パラメータ 255
- statement pipe max messages 設定パラメータ 255
- statement statistic active 設定パラメータ 256
- statement statistics active 設定パラメータ 256
- streamlined dynamic SQL 設定パラメータ 256
- strict dtm enforcement 設定パラメータ 256, 257
- Sun 文字セット 103
- suspend audit when device full 設定パラメータ 257
- syb_sendmsg port number 設定パラメータ 258
- Sybase Central、システム管理作業での使用 6
- sybsecurity データベース 26
- sybssystemdb データベース 26
- sybssystemprocs データベース 10, 12, 24
 - 「データベース」参照
- sysconfigures テーブル 75
- syscurconfigs テーブル 75
- sysdevices テーブル 279, 305
 - disk init 280
 - sp_dropdevice 307
 - sp_helpdevice 305
 - ステータス・ビット 306
- sysdevices 内の status ビット 306
- sysindexes テーブル 281, 348

syslogs テーブル
 修正 9
sysmessages テーブル 372, 374
sysobjects テーブル 348
sysremotelogins テーブル 292
syssegments テーブル 281
sys.servers テーブル 283, 284, 285, 289
 sp_helpserver 289
 srvname カラム 287
 srvnetname カラム 287
system セグメント 277
systemwide password expiration 設定パラメータ 259
sysusages テーブル 280
 破損 381, 382

T

tape retention in days 設定パラメータ 260
tcp no delay 設定パラメータ 261
tempdb データベース 24–25
 サイズ 24
 作成 277
 「データベース」参照
text prefetch size 設定パラメータ 261
text 値、*dbcc fix_text* によるアップグレード 350
text データ型
 マルチバイト文字セット 350
 文字セットの変更 350
time slice 設定パラメータ 262
timeouts オプション、*sp_serveroption* 287
total data cache size 設定パラメータ 262
transfer utility memory size 設定パラメータ 264
trunc log on chkpt データベース・オプション
 recovery interval in minutes 231
trusted モード
 リモート・ログイン 294
txn to pss ratio 設定パラメータ 265

U

unicar データ型 324
 Unicode 322, 324–327
 unicar データ型 324
 univarchar データ型 324
 UTF-16 324
 文字セット 323
unified login required 267
univarchar データ型 324

UNIX プラットフォーム、ロー・ディスク・パーティ
 ション 299
untrusted モード、リモート・ログイン 294
upgrade version 設定パラメータ 267
us_english 言語 106
use security services 設定パラメータ 267
user log cache size 設定パラメータ 268
user log cache spinlock ratio 設定パラメータ 269
 UTF-16 324

V

vstart オプション
 disk init 304

W

wait event timing 設定パラメータ 270
with nowait オプション、**shutdown** 392, 393
workload manager size 設定パラメータ 270

X

X/Open XA 116
xact 270
.xlt ファイル 354
 XP Server
 解放、メモリ 136
 優先度 135
xp_cmdshell context 設定パラメータ 271
xp_cmdshell システム拡張ストアド・プロシージャ 12

あ

アイコン 46
 空き領域
 「サイズ」「領域の割り付け」参照
 不足 379
 空き領域。「記憶領域の管理」参照
 値の比較
 データ型の問題 378
 アドレス、サーバ 14
 アプリケーション開発 210
 誤り、ユーザ。「エラー」「重大度レベル、エラー」参照
 アラビア語の文字セットのサポート 322
 暗号化、パスワード 140

索引

い

- 一貫性
 - データベースの検査 39
- インストール
 - サンプル・データベース 27
- インストール、サーバ
 - interfaces ファイル 15
 - インストール後のステータス 277
- インデックス
 - default fill factor percent の割合 105
 - 疑わしい 348, 381
 - オブジェクト・アロケーション・マップ 197
 - 再構築 348
 - ソート順の変更 349
 - 文字セットの変更 349
 - 文字ベース 348
- インデックス記述子
 - オープンできる最大数 200

う

- ウォッシュ、ハウスキーピング・タスク 143
- 疑わしいパーティションの処理 351
- 疑わしいパーティション、プラットフォーム間のダン
プとロード 352
- 上書き、データベース・オプション 50
- 運用サーバ 32

え

- エイリアス
 - サーバ 286
- エラー
 - 「エラー・ログ」「エラー・メッセージ」参照
 - サーバの応答 371-382
 - ステータス番号 371
 - 致命的 380-382
 - 複数 372
 - 文字変換 366
 - ユーザ 377, 377-379
 - レポート 382
 - ロギング 375
 - ログをとる情報の種類 13
- エラーのバックトレース。「エラー・ログ」参照

- エラー・メッセージ 374-382
 - 重大度レベル 376-382
 - 致命的なエラー 380-382
 - 番号 374
 - 変更、サーバが発行するエラー・メッ
セージ 356, 376
 - 文字変換 367
 - ユーザ定義 376
 - ユーザ定義の作成 376
- エラー・メッセージ内の変数 375
- エラー・ログ 40, 380
 - 作成と所有権 375
 - 消去 376
 - フォーマット 373
 - ロケーション 13
- エンジン
 - 数 168

お

- 欧州通貨記号
 - 文字セット 323
- 応答時間 262
- オーバーフロー・エラー
 - サーバ・スタック 250
- オーバーフロー・スタック (stack guard size 設定パラ
メータ) 249
- オブジェクト
 - アイコン 46
 - 「データベース・オブジェクト」参照
 - ナビゲート 47
- オブジェクト所有者。「データベース・オブジェクト所
有者」参照
- オプション
 - サーバ 287
 - データベース 311-313
 - リモート・サーバ 287
 - リモート・ログイン 294
- オペレータの役割 3
- オペレーティング・システム・コマンド
 - 実行開始時 12

か

- カーソル
 - ロー・カウント、設定 90
- カーネル
 - エラー・メッセージ 375, 380
- ガーベジ・コレクション
 - 消極的テスト 389
 - 積極的テスト 389
 - 設定、積極的 390
 - ハウスキーピング・ユーティリティ 389
- ガーベジ・コレクションの妨害
 - 空ページの累積 390
- 開発用サーバ 32
- 書き込み操作
 - 物理的な 277
- 拡張 UNIX 文字セット 103
- 拡張ストアド・プロシージャ
 - 設定パラメータ 135-272
- 仮想
 - アドレス 304
 - ページ番号 301
- 空ページの累積 390
- カラム名
 - 修飾されない名前 378
- 韓国語
 - 文字セットのサポート 323
- 監査
 - sybsecurity* データベース 26
 - キュー、サイズ 85
- 監査証跡 26
 - エラー・メッセージのスタックトレース 375
- 漢字。「日本語文字セット」参照

き

- キー、テーブル
 - システム・テーブル 8
- 記憶領域の管理 273
 - インストール時のデフォルト設定 277
 - コマンドの概要 274
 - システム・テーブル 278-281
 - データベース・デバイスの初期化 297-306
 - デフォルト・データベース・デバイス 307-308
 - 問題 34-36, 276
 - 「領域」参照 273
- 期限切れのパスワード 260
- 既知の問題 393

疑問符 (??)

- 疑わしい文字 367
- キャッシュ・パーティション
 - 設定 139, 141
- キャッシュ、データ
 - データベースの整合性のエラー 381
- キャッシュ、プロシージャ 227
- ギリシャ語
 - 文字セットのサポート 322
- キリル文字セットのサポート 322
- 記録の保管 41-42
 - 管理 42
 - システム 42
 - 設定 41
 - 連絡先 41

<

- クエリ
 - 変換エラー、クエリへの影響 367
- クライアント
 - 文字セット変換 369
- グループ
 - public 51
 - 言語 322

け

- 計算式
 - ユーザの要件 210
- 言語
 - サーバ 322
 - 文字セットによるサポート 321
- 言語グループ 321, 322
- 言語デフォルト 106
 - us_english* 106
 - ユーザ情報の変更 347
- 言語、代替 354
 - サポートされている言語 318
 - 日付フォーマット、サポートされていない言語 352
 - 「文字セット」「*charset.loc* ファイル」「日本語文字セット」参照
 - ローカライゼーション・ファイル 338-357
- 現在のデータベース 377
- 検索
 - データベース・オブジェクト 377
- 検証、ユーザ・アクセス 288, 291

索引

こ

高可用性

- enable HA の設定 119
- installhasvss スクリプト 120
- insthasv スクリプト 120

降順スキャン

- デッドロック 80

更新

- allow updates to system tables 設定パラメータ 12
- text の更新、文字セット変更後 350
- 「変更」参照

構造

- 国際化ファイルのディレクトリ 354
- ローカライゼーション・ファイルのディレクトリ 356

構文

- disk resize 309
- エラー 378

コード化、文字 359

- コール、リモート・プロシージャ 283-295
- タイムアウト 287

国際化

- サンプル・システム 319
- 定義 317
- ファイル 354
- 文字セットのディレクトリ構造 354
- 利点 318

国際言語のサポート。「文字セット」「言語」参照

コピー、選択データ

- 「insert コマンド」「select コマンド」参照

コマンド

- delete 389
- disk resize 308
- reorg reclaim_space 390

か

サーバ

- interfaces ファイル 14
- インストール 33, 277
- エラー・メッセージ 375
- エラー・メッセージの重大度レベル 376-382
- 構文エラー 378
- 終了 392
- シングルユーザ・モード 84
- スケジューラ 262
- 接続 14

設定パラメータの値 57

- ソート順の一貫性 343
- 致命的でない内部エラー 379
- 致命的なエラー 380-382

停止 392

名前 286, 287

パスワード 288, 294

パフォーマンスのモニタリング 71

「プロセス (サーバのタスク)」「リモート・サーバ」参照

ユーザ接続 210

リモート 285-291

ローカル 286

サーバ・エイリアス 286

サーバ情報オプション。「情報 (サーバ)」参照

サーバのリポート

「再起動、サーバ」参照

再起動、サーバ

インデックスの再構築 348

同じディレクトリ 376

再設定後 348

システム・テーブル 348

テンポラリ・テーブル 25

最小

サイズ、disk resize 308

サイズ

dbcc fix_text トランザクション 350

model データベース 104, 300

tempdb データベース 24

新しいデータベース 23

エラー・ログ 13

「領域」参照

最適化目標と設定パラメータ 215

削除

サーバ 289

ダンプ・デバイス 307

データベース・デバイス 307

デフォルト領域のプールからマスタ・デバイスを削除 307

ファイル 307

ユーザ 51

リモート・ログイン 289, 290

作成

guest ユーザ 50

master データベース 277

model データベース 277

tempdb データベース 277

システム・テーブル 7
 システム・プロシージャ 12
 ストアド・プロシージャ 12
 セグメント 277
 データベース 49
 データベース・オブジェクト 275
 ユーザ 51
 ユーザ定義のエラー・メッセージ 376
 サフィックス名、テンポラリ・テーブル 25

し

時間

ロック取得 158

時刻値

表示フォーマット 356

システム拡張ストアド・プロシージャ 12
 システム・カタログ。「システム・テーブル」参照
 システム管理作業

Sybase Central を使用して実行 6

システム管理者 1-5

エラーに対する責任 376, 379-382

基礎作業 31-42

システム問題の解決 376, 379

システム・セキュリティ担当者 3

システム・データベース 19-26

システム・テーブル 7-8

create database 7, 280

dbcc reindex 349

インデックスの再構築 349

行える変更 12

キー 8

記憶領域管理の関連性 278-281

クエリ 8, 12

更新 9, 12

個々のテーブル名参照

サーバの再起動 348

作成 7

ストアド・プロシージャ 8, 12

破損 381, 382

ユーザ・データベース用 23

システム・プロシージャ 10-12

作成 12

使用 10

「情報(サーバ)」「ストアド・プロシージャ」および各
 プロシージャ名参照

テーブル 11

テンポラリ・テーブル 25

リモート・サーバの管理 285-289

システム・メッセージ。「エラー・メッセージ」
 参照 371

システム問題

System Problem Reports (SPR) 393

「エラー」参照

サーバの応答 371-382

重大度レベル 10 ~ 18 377-379

重大度レベル 19 ~ 24 380-381

実行

ESP と XP Server の優先度 135

重大度レベル、エラー 371, 376

Backup Server 383

レベル 10 ~ 18 (ユーザ・エラー) 377

レベル 19 ~ 24 (致命的) 380

終了

Backup Server 392

サーバ 392

障害、メディア 381

消極的ガーベジ・コレクション 389

譲渡、所有権

「データベース・オブジェクト、所有権」参照

情報(サーバ)

エラー・メッセージ 374-382

設定パラメータ 62

ダンプ・デバイス 305

データベース・オプション 312-313

データベース・デバイス 305

デバイス 305

問題 375

リモート・サーバ 289

リモート・サーバ・ログイン 295

使用方法

disk resize 309

情報メッセージ(サーバ)。「エラー・メッセージ」「重大
 度レベル」参照

初期化

データベース・デバイス 297-305

シングルユーザ・モード 84, 348

索引

す

スイッチ、現在の set 設定の表示 314
数(量)
 エンジン 168
 オープン・オブジェクト 201
 サーバ上でオープンしているデータベース 198
 データベース・デバイス 188
 ユーザ接続 (@@max_connections) 209
 ロック 195
 ロック取得の秒数 158
スキャン記述子 184-186
スクリプト 321
スタンドアロン・ユーティリティおよび文字
 セット 369
ステータス
 情報メッセージ(レベル10) 377
ストアド・プロシージャ
 作成 12
 システム・テーブルの変更 12
 「データベース・オブジェクト」「システム・プロ
 シージャ」参照
 トリガ。「トリガ」参照
 パーミッション 293
 プロシージャ・キャッシュ 227
 リモート・ユーザ・アクセス 293
スピンロック
 ロック・ハッシュ・テーブル 156
スペイン語
 文字セットのサポート 322

せ

西欧
 文字セットのサポート 322
脆弱な時間帯 84
静的設定パラメータ 58
セーブポイント
 エラー(レベル13) 378
セキュリティ&ディレクトリサービスの必要性 140
セグメント 281
 logsegment 277
 syssegments テーブル 281
 system セグメント 277
 作成 277
 「データベース・デバイス」「領域の割り付け」参照
 デフォルト 277

積極的ガベージ・コレクション 389
 優先レベル 390
積極的ハウスキーピング 389
接続
 interfaces ファイル 14
 最大ユーザ数 209
 ディレクトリ・サービス 15
設定(サーバ)
 ソート順 341-349
 メッセージ言語 341-346
 文字セット 341
設定のリセット
 「設定パラメータ」「reconfigure コマンド」参照
設定パラメータ 75-269
 dtm detach timeout period 389
 housekeeper free write percent 389
 max native threads per engine 164
 rtm thread idle wait period 237
 デフォルト設定値 57
 表示、値 62
 ヘルプ情報 61
 変更 295
 リモート・ログイン 81, 295
設定パラメータ、max transfer history 174
設定ファイル
 指定、起動時 63
 設定値の格納 58
 デフォルト名とデフォルト・ロケーション 58

そ

ソート順
 default sortorder id 108
 default XML sortorder 108
 一貫性、サーバ内 343
 新規インストール 354
 定義ファイル 354
 番号 108
 変更 343-347
 変更後のインデックスの再構築 349
速度(サーバ)
 システムのパフォーマンス 277
その他のユーザ・エラー 378

た

- ダーティ・ページ 231
- タイ語
 - 文字セットのサポート 323
- 代替言語。「言語、代替」参照
- ダンプ・デバイス
 - sysdevices* テーブル 279
 - 関連情報 305
 - 削除 307
- ダンプ、データベース 37
- 端末
 - インストール、新しい端末定義 354
 - 文字セット変換 369

ち

- チェックポイント・プロセス 231
 - recovery interval* パラメータ 232
 - trunc log on chkpt* データベース・オプション 231
- 致命的なエラー
 - エラー・メッセージ 380-382
 - カーネルからのバックトレース 375, 380
 - 重大度レベル 19 以上 380-382
- 中国語 (簡体字)
 - 文字セットのサポート 322
- 中国語 (繁体字)
 - 文字セットのサポート 323
- チューニング
 - パフォーマンスのモニタリング 71
- 直接更新
 - システム・テーブル 83

つ

- 追加
 - 月 352
 - データベース・デバイス 210, 298-305
 - 日付文字列 352
 - ユーザをデータベースに 210
 - リモート・サーバ 285-289
 - リモート・ログイン 290-292
- 通貨
 - ローカル・フォーマット 356
- 月の値
 - 代替言語 352

て

- 停止、サーバ 392
- ディスク I/O
 - 設定パラメータ 191
 - データベース・ロードとディスク I/O 162, 188, 194
- ディスク・コントローラ 305
- ディスク・デバイス
 - 「データベース・デバイス」「ダンプ・デバイス」
 - 「領域の割り付け」参照
- ディスク・ミラーリング
 - sysdevices* テーブル内のステータス 306
 - 無効化 110
 - 有効化 110
 - リカバリ 276
- ディスク領域の不足
 - disk resize* 310
- ディスク。「データベース・デバイス」「デバイス」
- 「ダンプ・デバイス」参照
- ディレクトリ構造
 - *.loc* ファイル 357
 - 国際化ファイル 354
 - 文字セット 354
 - ローカライゼーション・ファイル 357
- データ・キャッシュ
 - データベースの整合性のエラー 381
 - パーティションの設定 139, 141
- データ辞書。「システム・テーブル」参照
- データベース
 - guest* ユーザ 50
 - エラーの影響 381
 - オープンしている数 198
 - オプション 311-313
 - サイズ 23
 - 作成 49
 - システム 19
 - 新規 23
 - 整合性の考慮事項 381
 - ダンプ 37
 - 「データベース・オブジェクト」「ユーザ・データベース」参照
 - デフォルト 22
 - デフォルトの記憶領域 20, 307
 - バックアップ 23, 37, 50
 - ロード、ソート順の変更後 344
 - ロード、文字セットの変更後 344

索引

- データベース・オブジェクト
 - アクセス・パーミッション 5
 - エラーの影響 381
 - オープンできる最大数 201
 - 検索 377
 - 個々のオブジェクト名参照
 - 作成 22, 275
 - 所有権 4
 - デバイスへの割り当て 276
 - ユーザ作成を制御 22
- データベース・オブジェクト所有者 4
 - 作業 4
 - 「データベース所有者」参照
 - パーミッション 5
- データベース・オプション 311-313
 - 設定 313
 - 設定の表示 313
 - リスト作成 312
- データベース管理 1-5
- データベース所有者 4
 - エラーに対する責任 377, 379
 - 作業 4
 - パーミッション 4
 - ログイン名 2, 4
- データベース・デバイス 297
 - 関連情報 305
 - サーバが使用できる数 188
 - 削除 307
 - 初期化 297-305
 - 追加 298-305
 - 「ディスク・ミラーリング」「ダンプ・デバイス」「マスタ・デバイス」参照
 - デフォルト 307-308
 - 名前 278, 299
 - 配置、オブジェクト 277
 - フラグメント 280
 - 領域。「セグメント」「領域の割り付け」参照
- テーブル
 - 2台のディスク間での分割 277
 - dbcc checktable 349
 - インデックスなし 349
 - 疑わしいパーティションを含んでいる、修正 352
 - 疑わしい、修正 352
 - オブジェクト・アロケーション・マップ 197
 - システム・プロシージャ 11
 - 整合性の損傷 381
 - 「データベース・オブジェクト」「システム・テーブル」参照
 - テンポラリ 24
 - 読み込み専用 348
- テーブル・エディタ 52
- テーブル所有者。「データベース・オブジェクト所有者」参照
- テスト・サーバ 32-33
- デッドロック 378
 - 降順スキャン 80
- デバイス 297
 - number of user connections 209, 210
 - 削除 307
 - 情報リスト 305
 - 初期化 297-305
 - 追加 298-305
 - 「データベース・デバイス」「ダンプ・デバイス」「マスタ・デバイス」参照
 - 物理名 299
 - 別のデバイスの使用 276
- デバイスの縮小、disk resize 310
- デバイス・フラグメント 280
- デバイス名 299
 - sysdevices のリスト 280
- デフォルト
 - 「データベース・オブジェクト」参照
- デフォルト設定
 - インストール時のシステム・データベース 278
 - 言語 106
 - 設定パラメータ 57
 - ソート順 108
 - データベース 22
 - パーミッション 23
 - 変更、ソート順 343-349
 - 変更、文字セット 342-351
 - 文字セット ID 番号 103
- デフォルト・データベース・デバイス
 - 指定 307
- 転送されたロー
 - default exp_row_size 設定パラメータによる削減 104
- テンポラリ・テーブル 24

と

- ドイツ語
 - 文字セットのサポート 322
- 東欧
 - 文字セットのサポート 322
- 統計
 - ハウスキーピングのフラッシュ 144
 - フラッシュ、ハウスキーピング・タスク 144
- 動的設定パラメータ 58
- トランザクション
 - 2 フェーズ・コミット 26
 - エラー 378
 - 長時間実行 231
 - リカバリ 231
- トランザクション・ログ
 - alter database 280
 - create database 280
 - trunc log on chkpt オプション 231
 - 消去 350
 - デバイスの配置 276, 280
- トリガ
 - 「データベース・オブジェクト」「ストアド・プロシージャ」参照
 - ネスト 81
- トルコ語
 - 文字セットのサポート 322

な

- 内部エラー、致命的でないもの 379
- ナビゲート
 - オブジェクト 47
- 名前
 - カラム、コマンド 378
 - サーバ 287
 - システム拡張ストアド・プロシージャ 12
 - システム・プロシージャ 10
 - リモート・サーバ 285
 - リモート・ユーザ 290
 - リモート・ユーザのマッピング 290

に

- 日本語文字セット 103
- sjis (シフト JIS) 103
- 「言語、代替」参照
- サポート 322

ね

- ネットワーク
 - interfaces ファイル 14
 - 接続 14
 - ソフトウェア 34
 - ディレクトリ・サービス 15

の

- ノンストップ・リカバリ 276

は

- パーセント記号 (%)
 - エラー・メッセージのプレースホルダ 374
- パーティション、疑わしい 351
- パーティション、疑わしい、テーブルの修正 352
- ハードウェア
 - エラー 381, 382
- パーミッション
 - disk init 305
 - master データベース 22
 - model データベース 23
 - tempdb データベース 25
 - 上書き 51
 - オブジェクト 5
 - グループとユーザ 51
 - ストアド・プロシージャ 293
 - データベース・オブジェクト所有者 5
 - データベース所有者 4
 - デフォルト 23
 - 否定 378
 - 不十分 (レベル 14) 378
 - リモート・ユーザ 293
- バイト
 - 文字 366

索引

ハウスキーピング・ガーベジ・コレクション 389
ハウスキーピング・タスク
 設定 143
 統計のフラッシュ 144
 領域の再利用 120
ハウスキーピング・チョア 389
 設定パラメータ `license information` 389
ハウスキーピング・ユーティリティ
 3つのタスク 388
 ウォッシュ 389
 ウォッシュ・タスク 143
 機能 388
 ハウスキーピング・ウォッシュ、ハウスキーピング・ガーベジ・コレクション、ハウスキーピング・チョア 388
バケット、ネットワーク
 サイズ、設定 165-167
パスワード
 ネットワーク間での暗号化 288
 リモート・ユーザ 288, 294
バックアップ 37-40
 master データベース 50
 ヒント 37-40
ハッシュ・バケット(ロック) 156
パフォーマンス
 default fill factor percent の影響 106
 ESP と XP Server の優先度 135
 監査キュー・サイズ 85
 速度 277
 ディスク・ミラーリング 276
 領域の割り付け 277
バルト語の文字セットのサポート 322
番号
 エラー・メッセージ 374
 ステータス・ビット (`sysdevices`) 306
 ソート順 108

ひ

日付
 エラー・メッセージのフォーマット 373
 代替言語 352
 日付部分の追加 352
 表示フォーマット 356

日付部分
 代替言語 352
非同期 I/O
 制限、サーバ要求 159
非同期プリフェッチ
 設定 140
ビュー
 「データベース・オブジェクト」参照
ピンイン
 gbpinyin ソート順と gbpinyinocs ソート順 334
 size of unilib cache 設定パラメータ 334, 335
 アクセント記号の使用 334
 中国語の発音の表現 334

ふ

ファイル
 Closed Problem Reports (CPR) 393
 interfaces 14
 libtcl.cfg ファイル 15
 System Problem Reports (SPR) 393
 エラー・ログ 13, 375
 国際化 354
 削除 307
 文字セットの変換 (`.xlt`) 354
 ローカライゼーション 356
ファイル記述子 209
 オペレーティング・システムに設定されたプロセス
 当たりの最大数 212
フィルファクタ
 default fill factor percent 設定パラメータ 105
フォーマット
 日付、時間、通貨 356
 ロケール、サポートしていない 352-353
複数のディレクトリ・サービス
 LDAP 17
不十分なパーミッション 378
物理リソース、管理
 「記憶領域の管理」参照
付与
 アクセス・パーミッション 4
 オブジェクト作成のパーミッション 4
フラグメント、デバイス領域 280
プラットフォーム間のダンプとロード、疑わしいパー
 ティションの処理 352

フランス語
 文字セットのサポート 322
 プレースホルダ
 エラー・メッセージのパーセント (%) 記号 374
 プロシージャ・キャッシュ 227, 381
 プロシージャ・コール
 「リモート・プロシージャ・コール」参照
 プロシージャ。「ストアド・プロシージャ」「システム・
 プロシージャ」参照
 プロセス ID、ステータス 387
 プロセス (サーバのタスク) 384, 388
 強制終了 384-388
 分割
 ディスク 299
 テーブルを 2 台のディスクに分割 277
 分散トランザクション管理 (DTM) 26
 分散トランザクション処理 (DTP) 26
 分離されているトランザクション 112

へ

ページ、データ 298
 ダーティ 231
 ベトナム語
 文字セットのサポート 323
 ヘブライ語
 文字セットのサポート 322
 変換
 「文字セット」参照
 変更
 「更新」参照
 システム・テーブル、危険度 9, 12
 設定パラメータ 71, 295
 データベース・オプション 311-313

ま

マスタ・デバイス 20, 299, 306
 sp_diskdefault 307
 「データベース・デバイス」参照
 デフォルト領域のプールからの削除 307
 マッピング
 デバイス名を物理名にマッピング 298
 リモート・ユーザ 289-293

マルチ言語文字セット 103
 マルチバイト文字セット 350
 default character set id 設定パラメータ 103
 非互換 366
 変更 351

め

命名
 サーバ 286
 メール・セッション、開始 253
 メタデータ・キャッシュの設定パラメータ 68-214
 メッセージ
 エラー 13, 374-382
 起動 13
 言語設定 318
 システム 374-382
 致命的なエラー 13
 ユーザ定義 376

メモリ

number of open databases 199
 解放、XP Server から 136
 監査レコード 85
 「領域の割り付け」参照
 メモリ・ダンプのスレッド数、決定 191

も

文字
 変換できない文字 366
 文字セット 103
 ID 番号 103
 Unicode 323
 アラビア語 322
 インデックスの再構築、文字セットの
 設定後 348-351
 欧州通貨記号 323
 韓国語 323
 ギリシャ語 322
 キリル・スクリプト 322
 クライアントと端末間の変換 369
 クライアントとファイル・システム間の変換 369
 言語グループ 322
 異なる文字セットにおけるコード化 359
 サーバ/クライアント間の変換 360-361
 サポートされる変換パス 360-367

索引

西欧語 322
タイ語 323
中国語 (簡体字) 322
中国語 (繁体字) 323
定義 321
定義ファイル 354
デフォルト 327
東欧 322
トルコ語 322
日本語 322
「日本語文字セット」参照
バルト語 322
ベトナム語 323
ヘブライ語 322
変換エラー 366
変換ファイル、端末固有 354, 370
変更 341
マルチバイト 350
マルチバイト、変更 351
文字セットの変更後の text 値のアップグレード 350
ロシア語 322
文字セットのバイナリ・ソート順
文字セットの変更とデータベース・ダンプ 344
文字セット変換 359, 367-368
モニタリング
spt_monitor テーブル 11
SQL テキスト 173
Windows NT パフォーマンス・モニタ 246
モニタリング・テーブル
設定オプション 68

や

役割、システム
オペレータ 3
システム管理者 2
システム・セキュリティ担当者 3

ゆ

ユーザ
guest 50
number of user connections 210
エラーの原因 377, 377-379
削除 51
作成 51
シングルユーザ・モード 84
リモート 289-293
ユーザ ID
値 1、データベース所有者 12
ユーザ・エラー 377, 377-379
ユーザ・オブジェクト。「データベース・オブジェクト」参照
ユーザ接続
割り付けられているメモリ 209-210
ユーザ・データベース
master データベース制御 21
システム・テーブル 23
ユーザ定義メッセージ 376
ユーザとの同一化。「setuser コマンド」参照
ユーザの誤り。「エラー」「重大度レベル、エラー」参照
ユーザ・パーミッションの上書き 51
ユーザ、オブジェクト。「データベース・オブジェクト所有者」参照
優先度
XP Server 135
ユーティリティ・コマンド
『ASE ユーティリティ・ガイド』参照
文字セット 369
ユーティリティ、ハウスキーピング、積極的 389

よ

曜日
代替言語 352
読み込み
物理的な 277

ら

ライセンス・セキュリティ&ディレクトリサービスが必要 140
ラテン・アルファベット 323

り

- リカバリ
 - master* データベース 37, 298
 - 再設定後 344
 - 設定パラメータ 230-232
 - ソート順の変更 344
 - ノンストップ 276
 - バックアップを計画 23
 - 領域の割り付け 276
 - ロード、データベース 344
- リスト作成
 - データベース・オプション 312
- リソースの制限値
 - 設定 82
- リソース不足エラー (レベル 17) 379
- リターン・ステータス
 - システム・プロシージャ 11
- リモート・サーバ 285-289
 - オプション 287
 - 削除 289
 - 情報 289
 - 追加 285-289
 - 名前 285
- リモート・プロシージャ・コール 283-295
 - 設定パラメータ 295
- リモート・ログイン
 - trusted* モードと *untrusted* モード 291
 - オプション 294
 - 削除 289, 290
 - 設定パラメータ 81, 295
 - タイムアウト 287
 - 追加 290-292
- 領域の再利用
 - enable housekeeper GC* 設定パラメータ 120
- 領域の不足。「領域」参照
- 領域の割り付け
 - sysusages* テーブル 280
 - コマンドの概要 274
 - 「データベース・デバイス」「セグメント」「記憶領域の管理」参照
 - リカバリ/パフォーマンス 276
- リンク、ページ
 - 「ページ、データ」参照

る

- ルール
 - 「データベース・オブジェクト」参照

れ

- レベル、重大度
 - 「重大度レベル、エラー」参照
- レポート
 - エラー 377, 379, 382
 - 「情報 (サーバ)」参照

ろ

- ローカライゼーション 317, 318
 - 「言語、代替」参照
 - ファイル 356
- ローカルおよびリモート・サーバ。「リモート・サーバ」参照
- ローカル・サーバ 286
- ロード用 50
- ロード、データベース
 - number of large i/o buffers* 設定パラメータ 110, 162, 188, 194
- ロールバック、プロセス
 - サーバ・スタック容量 252
 - リカバリ・インターバル 230
- ロー・ロック・プロモーション・スレッシュホールド
 - sp_configure* を使用した設定 235, 237
- ロー、テーブル
 - sysindexes* 281
- ロギング
 - Windows NT のイベント・ログ 138, 139
 - 失敗したログイン 159
 - 成功したログイン 159
- ログイン
 - “*dbo*” ユーザ名 2, 4
 - データベース・オブジェクト所有者 4
 - ロック 51
- ログインのロック 51
- ログイン・パスワードの暗号化 140
- ログ・ファイル。「エラー・ログ」参照
- ロシア語
 - 文字セットのサポート 322

索引

- ロック 51
 - dbcc コマンド 350
 - 数量 195
- ロック・スキーム
 - サーバワイドなデフォルト 155
 - デフォルト 155
- ロック・タイムアウト
 - サーバワイドの設定 158
- ロック・ハッシュ・テーブル
 - サイズの設定 154
- ロック・ハッシュ・バケット 156
- ロック・プロモーション・スレッシュールド
 - sp_configure を使用した設定 216–237
- 論理
 - ページ・サイズ 31

わ

- 割り付け
 - ページ 298
 - ユニット 298
 - ユニット。「サイズ」「領域の割り付け」参照