



設定ガイド

Adaptive Server[®] Enterprise

15.7 ESD #2

UNIX 版

ドキュメント ID：DC35824-01-1572-01

改訂：2012 年 7 月

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

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

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

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

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

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

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

IBM および Tivoli は、International Business Machines Corporation の米国およびその他の国における登録商標です。

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：Adaptive Server について	1
システム固有の問題	1
システムのユーザの役割	2
環境変数	2
 第 2 章：Adaptive Server のデバイスとシステム・データベース	5
master デバイス	5
sybsystemdb デバイス	6
sysprocsdev デバイス	6
オプションのデバイスとデータベース	7
プラグ可能コンポーネント・インタフェース (PCI)	
データベース	7
サンプル・データベース	7
sybsecurity デバイスと sybsecurity データベース	8
dbccdb データベース	8
sybmgmtldb データベース	8
データベース・デバイスとしてのオペレーティング・システム・ファイルの使用	9
directio パラメータ	9
dsync パラメータ	9
データベース・デバイスのサイズの確認	10
ロー・パーティションまたはファイルのサポート	11
ロー・パーティション・デバイスの準備	11
ロー・パーティションの選択	12

第 3 章：クライアント/サーバ通信	13
第 4 章：Adaptive Server の設定の変更について	15
第 5 章：英語以外の言語	17
第 6 章：Adaptive Server の仕様	19
データベースの仕様	19
テーブルの仕様	20
プロシージャの仕様	21
クエリの仕様	21
最大カラム・サイズ	22
ページ・サイズによるデータベース要件	22
ページ・サイズによるテーブルのデータ量の制限	23
第 7 章：サーバの起動と停止	25
サーバの起動	25
サーバの起動パラメータ	26
RUN_server_name ファイル	26
startserver コマンドの使用	27
オペレーティング・システムの再起動時のサーバの起動	27
HP-UX 用のシステム再起動スクリプトの作成	28
IBM RS/6000 用のシステム再起動スクリプトの作成	28
Sun Solaris および Linux 用のシステム再起動スクリプトの作成	29
最初のインストール後の XP Server の起動	31
サーバの停止	32

Adaptive Server の停止	32
Backup Server の停止	33
kill コマンドの使用	33
停止と共有メモリ・ファイル	34
 第 8 章：Linux のヒュージ・ページ	37
 第 9 章：オペレーティング・システムの設定	39
環境変数の確認	39
stty 設定の使用	39
正しいパーミッションのリストア	39
ファイル記述子とユーザ接続	40
HP-UX のファイル記述子とユーザ接続	40
AIX のファイル記述子とユーザ接続	40
Linux のファイル記述子とユーザ接続	40
Sun Solaris のファイル記述子とユーザ接続	41
現在のソフト制限値とハード制限値の表示	41
ソフト制限値を増やす方法	41
ハード制限値を増やす方法	42
非同期ディスク I/O の有効化	43
Linux での非同期ディスク I/O の有効化	43
IBM AIX での非同期ディスク I/O の有効化	44
HP-UX での非同期ディスク I/O の有効化	44
ファイル・システム・デバイスの非同期 I/O の有効化	45
ファイル・システムの非同期 I/O の有効化	45
クライアント接続のタイムアウト時間の調整	46
HP-UX のクライアント接続タイムアウトの調整	46
IBM RS/6000 のクライアント接続タイムアウトの調整	46
Linux のクライアント接続タイムアウトの調整	47

ハードウェア・エラーのチェック	47
オペレーティング・システム・リソースの使用状況のモ ニタ	48
HP-UX のオペレーティング・システム・リソースの モニタ	48
IBM RS/6000 のオペレーティング・システム・リ ソースのモニタ	48
Sun Solaris と Linux のオペレーティング・システ ム・リソースのモニタ	49
データベースの一貫性の検査	49
 第 10 章：srvbuild を使用した新しいサーバの設定	53
 第 11 章：Adaptive Server のデフォルト設定	57
デフォルト設定	57
 第 12 章：ネットワークを介する通信の設定	59
Adaptive Server が使用するディレクトリ・サービス・エ ントリの特定期間	59
クライアントのディレクトリ・サービスの使用方法	60
ディレクトリ・サービスのエントリの作成	60
サポートされているディレクトリ・ドライバ	61
interfaces ファイルの内容	61
interfaces ファイルのフォーマット	63
interfaces ファイルのエントリの要素	64
マスタ interfaces ファイルの作成	66
dsedit または dscp を使用したマスタ interfaces ファ イルの作成	66
テキスト・エディタを使用したマスタ interfaces ファイルの作成	67

複数のネットワーク用の interfaces ファイルの設定	67
複数のネットワーク・ハンドラ用のサーバの設定	68
クライアント接続の設定	68
ネットワークに依存しない単一の DSQUERY 名の管理	69
異なる DSQUERY 名の使用	69
クエリ・ポート・バックアップの設定	70
IPv6 のサポート	71
IPv6 インフラストラクチャ	72
 第 13 章：トラブルシューティング	75
サーバが起動しない場合のトラブルシューティング	75
ESP 実行時のエラーのトラブルシューティング	76
 第 14 章：ディレクトリ・サービスとしての Lightweight Directory Access Protocol (LDAP) の 使用	77
LDAP ディレクトリ・サービスと Sybase interfaces ファ イルの違い	78
libtcl*.cfg ファイル	81
LDAP ディレクトリ・サービスの有効化	82
LDAP URL 変数のキーワード	83
ディレクトリ・サービスへのサーバの追加	83
複数のディレクトリ・サービス	84
パスワードの暗号化	85
パフォーマンス	86
interfaces ファイルから LDAP へのマイグレート	86
 第 15 章：ローカライゼーション・サポート	87
言語モジュール	88

デフォルトの文字セット	88
サポートされる文字セット	90
アラビア語の文字セット	90
バルト語の文字セット	90
中国語 (簡体字) の文字セット	90
中国語 (繁体字) の文字セット	91
キリル語の文字セット	91
東欧言語の文字セット	91
ギリシャ語の文字セット	92
ヘブライ語の文字セット	92
日本語の文字セット	92
韓国語の文字セット	93
タイ語の文字セット	93
トルコ語の文字セット	93
Unicode の文字セット	94
ベトナム語の文字セット	94
西欧言語の文字セット	94
文字セット名	95
文字セットのロード	97
言語定義ファイルのソート順	98
文字セット変換	100
サーバ／クライアント間の変換	101
ソート順	102
文字セットで利用できるソート順	102
言語モジュール	104
新しい言語モジュールのインストール	104
メッセージ言語	105
ローカライゼーションのディレクトリ	105
locales ディレクトリについて	106
charsets ディレクトリについて	106
locales.dat ファイルのエントリで使用するフォー マット	106

クライアント・アプリケーションの locales.dat ファイルの使用方法	107
Adaptive Server のローカライゼーション設定の変更	107
Adaptive Server でのその他の文字セットの設定	108
locales.dat ファイルの編集	109
Backup Server のローカライゼーション設定の変更 ..	110
 第 16 章：Adaptive Server エラー・ロギング	111
エラー・ログのパス設定	111
Adaptive Server エラー・ログのパスの設定	112
メッセージの管理	112
ユーザ定義メッセージのロギング	112
監査イベントのロギング	113
 第 17 章：データベース・デバイスの管理	115
デバイスの要件	115
データベース・デバイス用のファイルの作成	116
 第 18 章：データベース管理システムの監査	119
監査システムのデバイスとデータベース	119
監査のインストール	120
監査デバイスに関するインストール前の推奨事項	120
Adaptive Server での監査の設定	121
監査データベースのトランザクション・ログ用のデ バイスの作成	122
監査の有効化	124
デバイス・エントリの削除	124
デバイス・エントリの変更	124

第 19 章：Transact-SQL 構文のオンライン・ヘルプの インストール	127
sybsyntax データベースのデフォルトのデバイス	128
sybsyntax のインストール	128
索引	131

第 1 章

Adaptive Server について

Adaptive Server[®] Enterprise は、クライアント・アプリケーションとユーザ・インタフェース機能とは独立して、データ管理とトランザクション機能を実行します。

Adaptive Server には、次の機能もあります。

- 複数のデータベースと複数のユーザの管理
- ディスク上にあるデータのロケーションの記録
- 物理データ記憶領域への論理データ記述のマッピングの保守
- メモリ内でのデータ・キャッシュとプロシージャ・キャッシュの保守

Adaptive Server は、次のように補助プログラムを使用して専用のタスクを実行します。

- Backup Server — データベースのロード、ダンプ、バックアップ、リストアを管理する。
- XP Server — ESP (拡張ストア・プロシージャ) を保存する。これにより、Adaptive Server がオペレーティング・システム・コマンドとユーザ定義コマンドを実行できるようになる。

注意：これらの手順は、Adaptive Server がインストールされ稼働していることを前提としています。Adaptive Server の概要、および Adaptive Server のインストールと起動の詳細については、使用しているプラットフォームの『インストール・ガイド』を参照してください。

Adaptive Server Enterprise は、全機能を搭載した Adaptive Server であり、次のプラットフォームで稼働します。

- HP-UX
- IBM AIX
- Sun Solaris
- Linux

システム固有の問題

Adaptive Server は、さまざまなハードウェアおよびオペレーティング・システム・プラットフォーム上で動作します。システム固有の問題が Adaptive Server の基本機能に影響することはありませんが、異なるプラットフォーム間では、

次のような相違点があります。

第 1 章：Adaptive Server について

- Adaptive Server の設定
- Adaptive Server の動作を可能にしたり、パフォーマンスを強化したりするためのオペレーティング・システムへの変更
- interfaces ファイル内のエントリの構造
- データベース・デバイス選択時のオプション
- 日常のシステム管理タスクを単純化または自動化するためのオペレーティング・システム・コマンドまたはユーティリティ
- Adaptive Server のパフォーマンスを監視するためのオペレーティング・システム・ユーティリティ

使用しているプラットフォームの『インストール・ガイド』およびリリース・ノートを参照してください。

システムのユーザの役割

Adaptive Server のインストールと設定のプロセスでは、さまざまなユーザの役割が定義されます。それぞれに異なる責任と権限が与えられます。

次に示すユーザの役割を見ると、使用しているシステムに Adaptive Server を統合する方法が明確になります。

- オペレーティング・システム管理者 — オペレーティング・システムを管理するユーザ。スーパーユーザまたは root の権限を持っています。
- システム管理者 — Adaptive Server のシステム管理を担当するユーザ。ユーザ・アカウントの作成、データベースのパーミッションの割り当て、新しいデータベースの作成を行います。インストール時にシステム管理者が使うログイン名は "sa" です。"sa" は UNIX のログインではありません。"sa" ログインは Adaptive Server に固有のものであり、**isql** コマンドで Adaptive Server にログインするときに使用します。
- "sybase" ログイン — すべての Sybase[®] インストール・ディレクトリとファイルを所有する UNIX ログイン。これらのディレクトリやファイルのパーミッションを設定したり、Adaptive Server のインストールやアップグレードを行ったりするときに使用します。

環境変数

Sybase 製品が動作するには、システム環境変数を正しく設定することが重要です。

環境変数をユーザの環境に設定するときは、対話的に設定する方法と、ユーザの .login ファイルと .cshrc ファイル (C シェルの場合) または .profile ファイル (Bourne シェルの場合) に環境変数を含める方法があります。

注意： インストーラは、SYBASE.csh ファイルと SYBASE.sh ファイルでこれらの環境変数を設定します。これらのファイルを情報源として読み込み、ユーザの環境を設定します。

- **DSLISTEN** — Adaptive Server 起動時に名前が指定されていない場合に、クライアントの接続要求を受信するために Adaptive Server が使用する名前を定義します。DSLISTEN を設定せず、Adaptive Server 起動時にも名前を指定しなかった場合は、SYBASE がデフォルトで Adaptive Server 名として設定されます。
- **DSQUERY** — コマンド・ライン・オプションで Adaptive Server 名が指定されていない場合に、クライアント・プログラムが接続しようとする Adaptive Server の名前を定義します。DSQUERY を設定せず、コマンド・ライン・オプションで Adaptive Server 名も指定しなかった場合は、クライアントは SYBASE に対して接続を試みます。
- **SYBASE** — Sybase インストール・ディレクトリのパスを定義します。インストール・プログラムでは、インストール時に指定したリリース・ディレクトリを指すように SYBASE 環境変数を設定します。
- **SYBASE_ASE** — Adaptive Server コンポーネントのサブディレクトリを定義します。
- **SYBASE_OCS** — Open Client™ が設定されるサブディレクトリを定義します。
- **SYBASE_SYSAM** — ライセンス管理ソフトウェアのディレクトリを指します。
- **PATH** — 実行プログラムを検索するディレクトリ・パスを指定します。Sybase 実行プログラムは、インストールされたコンポーネントの /bin ディレクトリにあります。ソース・ファイル SYBASE.csh または SYBASE.sh を使用すると、PATH の前に次のパスが付加されます。

```
$SYBASE/$SYBASE_ASE/bin:$SYBASE/$SYBASE_OCS/bin
```

- **LD_LIBRARY_PATH** — この変数は、共有ライブラリを検索するディレクトリを指定します。Sybase 共有ライブラリは、インストールされたコンポーネントの /lib ディレクトリにあります。

ソース・ファイル SYBASE.csh または SYBASE.sh を使用すると、

LD_LIBRARY_PATH 環境変数の前に次のパスが付加されます。

```
$SYBASE/$SYBASE_ASE/lib:$SYBASE/$SYBASE_OCS/  
lib:$SYBASE/SYBASE_FTS/lib,etc.
```

- **LD_LIBRARY_PATH64** — この変数は、64 ビット版の Sun プラットフォームで共有ライブラリを検索するディレクトリを指定します。
- **LIBPATH** — この変数は、IBM RS/6000 プラットフォームでライブラリを検索するディレクトリを指定します。
ソース・ファイル SYBASE.csh または SYBASE.sh を使用すると、**LIBPATH** 環境変数の前に次のパスが付加されます。

第 1 章：Adaptive Server について

```
$SYBASE/$SYBASE_ASE/lib:$SYBASE/$SYBASE_OCS/  
lib: $SYBASE/SYBASE_FTS/lib, etc.
```

- **SHLIB_PATH** — この変数は、HP-UX プラットフォームでライブラリを検索するディレクトリを指定します。
ソース・ファイル SYBASE.csh または SYBASE.sh を使用すると、LIBPATH 環境変数の前に次のパスが付加されます。

```
$SYBASE/$SYBASE_ASE/lib:$SYBASE/$SYBASE_OCS/  
lib: $SYBASE/SYBASE_FTS/lib, etc.
```

参照：

- 環境変数の確認 (39 ページ)

Adaptive Server のデバイスとシステム・データベース

デバイスとは、データベースやデータベース・オブジェクトを格納するファイルまたはディスクの一部のことです。デバイスは、ロー・ディスク・パーティションまたはオペレーティング・システム・ファイルを使用して初期化できます。

新しい Adaptive Server の作成時に次のデバイスが作成されます。

- `master` — システム・データベースを格納します。
- `sybsystemdb` — 分散トランザクションについての情報を格納します。
- `sysprocsdev` — システム・プロシージャを格納します。

master デバイス

`master` デバイスは、`master` データベース、`tempdb` データベース、`model` データベース、およびサンプル・データベースを格納します。

- `master` — Adaptive Server の動作を制御し、すべてのユーザ、ユーザ・データベース、デバイス、オブジェクト、システム・テーブル・エントリについての情報を格納します。`master` データベースは全体が `master` デバイ스에格納され、他のデバイスに拡張することはできません。
- `model` — 新しいユーザ・データベース用のテンプレートを提供します。`model` データベースには、必須システム・テーブルがあります。このテーブルは、**create database** コマンドによって新しいユーザ・データベースにコピーされます。
- `tempdb` — Adaptive Server のテンポラリ・データベースの作業領域です。Adaptive Server では複数の `tempdb` データベースがサポートされます。『Transact-SQL ユーザーズ・ガイド』の「データベースおよびテーブルの作成」を参照してください。Adaptive Server が起動するたびに、`tempdb` データベースはクリアされ、`model` データベースから再構築されます。
- `pubs2` および `pubs3` サンプル・データベース — インストール時に `master` デバイ스에格納されますが、インストール後はユーザ定義デバイスに移動してください。

注意： リカバリを正常に行うためには、マスタ・デバイスにほかのシステム・データベース、ユーザ・データベース、またはユーザ・オブジェクトを作成しないことをおすすめします。

sybsystemdb デバイス

sybsystemdb デバイスは、sybsystemdb データベースを格納します。このデータベースは、処理中のトランザクションについての情報を格納し、リカバリ中にも使用されます。

新規インストールの場合、master デバイスにも sybsystemdb データベースが格納されます。

DTM (分散トランザクション管理) 機能をサポートするには、sybsystemdb データベースが必要です。インストールの前に、sybsystemdb をサポートするための十分な空き領域がデフォルト・セグメント上にあることを確認してください。

sysprocsdev デバイス

sysprocsdev デバイスは、sybsystemprocs データベースを格納します。このデータベースは、Sybase が提供するほとんどのシステム・プロシージャを格納します。

システム・プロシージャとは、システム・タスクを実行する SQL 文とフロー制御文の集まりです。リカバリ状況下で必要になるシステム・プロシージャ (sp_configure など) は、master データベース内に格納されています。

注意： sysprocsdev は、このデバイスに使用されるデフォルトのシステム名です。ただし、このデバイスは sybsystemprocs データベースを格納するため、sybsystemprocs デバイスと呼ばれることもあります。

オプションのデバイスとデータベース

オプションのデバイスとデータベースには、プラグ可能コンポーネント・インタフェース (PCI) データベース、サンプル・データベース、sybsecurity デバイスおよびデータベース、dbccdb データベースが含まれます。

プラグ可能コンポーネント・インタフェース (PCI) データベース

プラグ可能コンポーネント・インタフェース (PCI) を使用すると、Adaptive Server にさまざまな機能を提供するライブラリを追加できます。Adaptive Server 15.0.3 以降には、プラグ可能コンポーネントとして Java サポート (プラグ可能コンポーネント・アダプタ/Java 仮想マシン) が組み込まれています。

sybpcidb データベースには、PCI およびプラグ可能コンポーネント・アダプタ/Java 仮想マシン (PCA/JVM) プラグインに必要な設定情報が格納されます。

srvbuildres を使用して Adaptive Server で PCI を有効にするには、これらのユーティリティが使用する PCI/Java 関連のプロパティをリソース・ファイルに追加します。次の値を入力します。

```
sqlsrv.do_configure_pci: yes
sqlsrv.sybpcidb_device_physical_name:/device_path
sqlsrv.sybpcidb_device_size: USE_DEFAULT
sqlsrv.sybpcidb_database_size: USE_DEFAULT
```

サンプル・データベース

利用可能なサンプル・データベースは、pubs2、pubs3、interpubs、および jpubs です。

- pubs2 データベースと pubs3 データベースは、Adaptive Server の学習ツールとして提供されています。Adaptive Server のマニュアルに記載されている例の大半では、pubs2 データベースを使用しています。その他の例では、pubs3 データベースを使用しています。英語版の Adaptive Server ではどちらも使用できます。
- interpubs データベースには、フランス語のデータとドイツ語のデータが格納されています。
- jpubs データベースには、日本語のデータが格納されています。

サンプル・データベース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』の「インストール後の作業」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。

sybsecurity デバイスと sybsecurity データベース

sybsecurity デバイスは監査インストール・プロセスの一環として作成され、sybsecurity データベースを格納します。また、監査を設定するときに使用するシステム・プロシージャも格納します。

監査システムは、Adaptive Server の監査証跡にシステムのセキュリティ情報を記録します。この監査証跡を使用して、Adaptive Server やシステム・リソースの使用状況をモニタできます。

『セキュリティ管理ガイド』の「監査」を参照してください。

参照：

- 監査のインストール (120 ページ)
- 第 18 章、「データベース管理システムの監査」 (119 ページ)
- データベース・デバイスのサイズの確認 (10 ページ)

dbccdb データベース

データベース一貫性チェッカ (**dbcc**) のコマンドを使用すると、データベースの論理および物理の一貫性をチェックできます。dbccdb データベースには、**dbcc checkstorage** または **dbcc checkverify** の結果が格納されます。

dbcc checkstorage は、「ターゲット・データベース」の設定情報、操作アクティビティ、その操作の結果を dbccdb データベースに記録します。このデータベースには、dbccdb の作成と管理を行ったり、**dbcc checkstorage** 操作の結果についてのレポートを生成したりする **dbcc** ストアド・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「データベースの一貫性の検査」を参照してください。

sybmgmtdb データベース

ジョブ、スケジュール、およびスケジュールされたジョブについての情報、Job Scheduler タスクで内部処理のために必要なデータはすべて、sybmgmtdb データベースに格納されます。

sybmgmtdb データベースのデータへのほとんどのアクセスには、ストアド・プロシージャを使用します。そのため、GUI、JS Agent、コマンド・ライン・インタフェースからデータを利用できます。sybmgmtdb データベースのデータに直接アクセスするのは Job Scheduler タスクだけです。

『Job Scheduler ユーザーズ・ガイド』を参照してください。

データベース・デバイスとしてのオペレーティング・システム・ファイルの使用

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、デバイスへの書き込みが物理メディア上で直接行われるようにします。

次の操作を行います。

- **disk init**、**sp_deviceattr**、および **disk reinit** で **directio** を指定する
- **dsync with disk init**

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** を false にリセットします。

directio パラメータ

disk init、**disk reinit**、**sp_deviceattr** に **directio** パラメータを指定すると、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、Adaptive Server の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。

デフォルトでは、**directio** オプションはすべてのプラットフォームで true (オン) に設定されます。ロー・デバイスの場合、**directio** と **dsync** は無視されます。

『システム管理ガイド 第 1 巻』を参照してください。

参照：

- **dsync** パラメータ (9 ページ)
- 監査デバイスに関するインストール前の推奨事項 (120 ページ)

dsync パラメータ

dsync パラメータを指定すると、ファイル・システム上のデバイスからデータをリカバリすることができます。

デフォルトでは、ファイル・システム・デバイスの **dsync** は無効になります。

dsync は、書き込み処理の多いファイル・システム・デバイスのパフォーマンスを低下させることがあります。**dsync** を設定またはリセットするには、**disk init** コマンド、**disk reinit** コマンド、**sp_deviceattr** コマンドを使用します。**dsync** を false (オフ) に設定した場合、キャッシュ I/O が使用されます。

注意： ロー・デバイスの場合、**dsync** と **directio** は無視されます。

アップグレード直後に、ファイル・システム・デバイスに対して **dsync** または **directio** が設定されていること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_deviceattr**」および「**sp_helpdevice**」も参照してください。

参照：

- **directio** パラメータ (9 ページ)
- 監査デバイスに関するインストール前の推奨事項 (120 ページ)

データベース・デバイスのサイズの確認

Adaptive Server では、各データベース・デバイスの基準値が規定されています。これらの値の最終変更については、リリース・ノートを参照してください。

表 1：Adaptive Server のデータベース・デバイス

デバイス	目的	最小サイズ	最小サイズ (推奨)
master	システム・データベース を格納	2K ページの場合：24MB 4K ページの場合：45MB 8K ページの場合：89MB 16K ページの場合：177MB	30MB 60MB 120MB 240MB
sysprocsdev (sybsystemprocs デ バイスともいう)	sybsystemprocs データベースを格納	160MB	172MB (この 値に、作成し たストアド・ プロシージャ を保持するた めの領域を追 加する)

デバイス	目的	最小サイズ	最小サイズ (推奨)
sybsystemdb	トランザクション処理	2K ページの場合：3MB 4K ページの場合：6MB 8K ページの場合：12MB 16K ページの場合：24MB	5 ～ 24MB
sybsecurity (オプション)	監査に必要	10MB と model データベースのサイズのうち、大きい方	固有の監査要件によって異なる

参照：

- sybsecurity デバイスと sybsecurity データベース (8 ページ)
- 監査のインストール (120 ページ)
- 第 18 章、「データベース管理システムの監査」 (119 ページ)

ロー・パーティションまたはファイルのサポート

すべてのデータベースにロー・パーティションまたはファイル・システムを使用できます。

Adaptive Server はデータの格納について、NetApp ファイラーで NFS マウント・デバイスおよび CIFS マウント・デバイス上のデータベース・デバイスをサポートします。NetApp ファイラーは、ロー・デバイスと同じパフォーマンスとデータ整合性を提供します。オペレーティング・システムや Adaptive Server を変更する必要はありません。

NFS マウント・デバイスは、Solaris、HP-UX、IBM AIX、Linux の各システム上での動作が確認されています。

ロー・パーティション・デバイスの準備

ロー・パーティション・デバイスを準備する際には、ベスト・プラクティスに従ってパフォーマンスを向上させ、ファイルの障害を回避します。

選択したディスクに、パーティションの再設定が必要になることがあります。サポートが必要な場合は、オペレーティング・システム管理者に連絡してください。

- Sybase インストール・ソフトウェアが格納されているパーティションで、データベース・デバイスを初期化しない。初期化すると、そのパーティションの既存のファイルがすべて破壊される。
- Sybase が使用するロー・パーティションは、ファイル・システムやスワップ領域など、オペレーティング・システムがその他の目的で使用するようにマウントすることはできない。
- Sybase 設定ユーティリティまたは **disk init** コマンドを使用してパーティションの一部をデータベース・デバイスとして初期化すると、そのパーティション全体が別の目的で使用できなくなる。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には、**disk resize** を使用する。
- パフォーマンスを最適にするには、Sybase ソフトウェアとすべてのディスク・デバイス (master デバイスなど) を同じマシン上に配置する。
- パーティション・マップが格納されているパーティションの使用を防ぐために、シリンダ 0 を使用しない。

ロー・パーティションの選択

ロー・デバイスのサイズとロー・デバイスが使用可能かどうかを確認します。

1. 使用可能なロー・パーティションを調べます。
2. ロー・パーティションのサイズを決定します。
3. 使用可能なロー・パーティションのリストから、各デバイスのロー・パーティションを選択します。
4. オペレーティング・システム管理者に、選択したパーティションが使用可能であることを確認します。
5. そのロー・パーティションに対する読み込み権限および書き込み権限を "sybase" ユーザが持っていることを確認します。

ロー・パーティションの選択方法の詳細については、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。

Adaptive Server は、他の Adaptive Server、Open Server™ アプリケーション (Backup Server など)、ネットワーク上のクライアント・ソフトウェアと通信します。リモート・プロシージャ・コールを使用して、クライアントは 1 つ以上のサーバと通信し、サーバは他のサーバと通信できます。

Sybase 製品間で対話するには、他の製品がネットワーク上のどこにあるかを各製品が認識する必要があります。既知のサーバの名前とアドレスはすべて、ディレクトリ・サービス・ファイルにリストされます。この情報をディレクトリ・サービス・ファイルに格納するには、次の 2 つの方法があります。

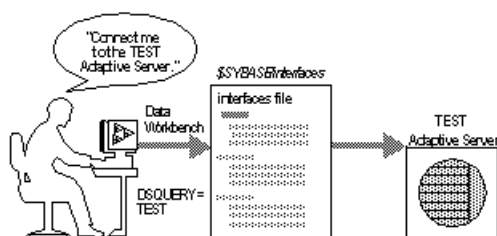
- interfaces ファイルに格納する。このファイルは、UNIX プラットフォームで interfaces という名前が付けられ、\$SYBASE インストール・ディレクトリに置かれます。
- LDAP サーバに格納する。

Adaptive Server またはクライアント・ソフトウェアをインストールしたら、ディレクトリ・サービスにリストされているネットワーク上のどのサーバにも接続できます。

クライアント・プログラムを使用して特定のサーバと接続する場合、クライアント・プログラムはディレクトリ・サービスでサーバ名を検索し、そのサーバに接続します。サーバ名は、DSQUERY 環境変数を使用して指定できます。

TCP/IP ネットワーク上では、クライアントは、接続する Adaptive Server、Open Server、または Backup Server をポート番号で識別します。また、サーバがクライアントからの接続を受信する場所もポート番号によって識別されます。サーバでは、これらの 2 つのサービス (「クエリ・サービス」および「リスナ・サービス」) に 1 つのポートを使用します。

図 1 : interfaces ファイルを使用したサーバとの通信



第 3 章：クライアント/サーバ通信

インストール時に、**srvbuild** ユーティリティを使用して新しいサーバを作成および設定します。**srvbuild** プロセスによって、新しい Adaptive Server、Backup Server、XP Server の **interfaces** ファイルにエントリが追加されます。

dsedit および **dscp** を使用して既存の **interfaces** ファイル・エントリを修正する方法については、『ユーティリティ・ガイド』を参照してください。

参照：

- 第 14 章、「ディレクトリ・サービスとしての Lightweight Directory Access Protocol (LDAP) の使用」(77 ページ)

Adaptive Server の設定の変更について

Adaptive Server の設定を変更するには、**sp_configure** を使用します。

複数のサーバの設定を変更するには、スクリプトに **sp_configure** 情報を入力します。**sp_configure** の使用方法の詳細については、『システム管理ガイド 第 1 巻』と『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。

次の設定を行うこともできます。

- 言語、文字セット、ソート順を設定する。
- 高可用性機能を使用するように Adaptive Server を設定する。「高可用性システムにおける Sybase フェールオーバーの使用」を参照してください。
- 分散トランザクション管理 (2 フェーズ・コミット) を実行するように Adaptive Server を設定する。『システム管理ガイド 第 2 巻』を参照してください。

参照：

- Adaptive Server のローカライゼーション設定の変更 (107 ページ)

英語以外の言語で **srvbuild** を実行する場合、入力には us_english 文字セットでサポートされる文字セットを使用します。

注意： us_english 文字セットでは、チルド (~) やウムラウト (ü) などのアクセント記号はサポートされていません。このため、**srvbuild** はこれらの文字を使用する文字セットをサポートしません。

言語、文字セット、ソート順の詳細については、使用しているプラットフォームの『インストール・ガイド』を参照してください。**srvbuild** 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。

Adaptive Server の仕様には、Adaptive Server の稼働条件についてのデータベース、テーブル、クエリ、プロシージャ、および拡張された制限機能の情報が含まれます。

データベースの仕様

データベースの仕様では、データベース・サイズやサーバあたりのデータベース・デバイス数など、各項目の要件を定義します。

項目	稼働条件	注意
Adaptive Server あたりのデータベース数	サーバあたり最大 32,767 のデータベース	
最大データベース・サイズ	<ul style="list-style-type: none"> 2K ページ・サーバ - 8TB 4K ページ・サーバ - 16TB 8K ページ・サーバ - 32TB 16K ページ・サーバ - 64TB 	Adaptive Server は、論理ページ範囲の上限で 256 個の論理ページ ID (割り付けることも使用することもできない) を予約するため、上記のサイズは、実際の使用可能領域の量よりも若干大きくなっている。このオーバヘッドにより、実際の使用可能領域の量は、記載されている各ページ・サイズの論理ページ・サイズに 256 をかけた数値分少なくなる (たとえば、2K サーバの実際の使用可能サイズは $8TB - (256 \times 2K)$)。
sybsystemprocs データベースの最小許容量	136MB	アップグレードに必要なサイズ
データベース・デバイスの最大サイズ (ディスク・パーティション)	2^{42} (4TB)	オペレーティング・システムがサポートするファイルのサイズが 4TB までの場合、Adaptive Server がサポートするファイル・システムのデバイスも 4TB まで。

項目	稼働条件	注意
サーバあたりのデータベース・デバイスの最大数	2 ³¹	
データベースあたりのデバイスまたはデバイス区画の最大数	無制限	使用可能なメモリ容量による制限を受ける。
データベースあたりのセグメントの最大数	31	
サーバあたりのログイン ID の最大数	2147516416	
データベースあたりのユーザの最大数	2146484223	
データベースあたりのグループの最大数	1032193	

テーブルの仕様

テーブルの仕様では、テーブルあたりのインデックス数、ロー数、カラム数など、各項目の要件を定義します。

項目	稼働条件	注意
データベースあたりのユーザ・オブジェクト数	2 ³¹ - 255	
テーブルあたりのインデックス数	250 (1 つのクラスタード・インデックス)	
テーブルあたりのロー数	使用可能なディスク容量による制限を受ける。	最大 2 ³²
複合インデックスあたりのカラム数	31	
クラスタード・インデックスの作成	1.2*(x + y) x = テーブル内のデータ領域の総合計、 y = テーブル上のすべてのノンクラスタード・インデックスの領域の合計、 およびロギング用に 20% のオーバーヘッド	ソートされているデータの場合は、テーブル・サイズ の約 20% が必要。
オブジェクト名の最大サイズ	255	

プロシージャの仕様

プロシージャの仕様では、バッファとプロシージャ・バッファの数、ストアド・プロシージャあたりの最小メモリ容量などの項目について説明します。

項目	稼働条件	注意
バッファとプロシージャ・バッファの数	設定可能	メモリ量と共有メモリ・セグメントの最大サイズによって制限される。
ストアド・プロシージャあたりの最小メモリ容量	2K	
ストアド・プロシージャあたりの最大パラメータ数	2048	

クエリの仕様

クエリの仕様では、"union" クエリ内のテーブルの最大数など、各項目の要件を定義します。

項目	稼働条件	注意
1 つのクエリ (union のないクエリまたは 1 つのクエリの union の各項) に関与するテーブルの最大数	64	ユーザ・テーブルの最大数は 50。これには、結果テーブルの他に、ビュー (ビュー自身はカウントされない)、相関、セルフジョインによって参照されているテーブルを含む。最大ワーク・テーブルは 46。
"union" クエリ内のテーブルの最大数	256	union の各項に最大 50 個のユーザ・テーブルと 14 個のワーク・テーブル、union のすべての項全体で最大 256 個のテーブルを含む。
1 つのトランザクションに関与するデータベースの最大数	無制限	トランザクションが開始されたデータベース、トランザクション中に変更されたすべてのデータベース、結果またはワーク・テーブルに使用される tempdb を含む。
1 つのクエリに関与するデータベースの実際の数	16	クエリの対象となる各データベースの各オカレンスと、結果またはワーク・テーブルに使用される tempdb を含む。

項目	稼働条件	注意
1 つのクエリに対して参照整合性制約を持つテーブルの最大数	192	

最大カラム・サイズ

Adaptive Server の拡張された制限機能は、テーブルのタイプとデータベースの論理ページ・サイズによって異なります。

全ページロック (APL) テーブルのカラムとローの制限は次のとおりです。

APL テーブルの制限	カラム数	カラム・サイズ 2K ページ	カラム・サイズ 4K ページ	カラム・サイズ 8K ページ	カラム・サイズ 16K ページ
固定長カラム	1024	1960 バイト	4008 バイト	8104 バイト	16296 バイト
可変長カラム	254	1948 バイト	3988 バイト	8068 バイト	16228 バイト

データオンリーロック (DOL) テーブルのカラムとローの制限は次のとおりです。

DOL テーブルの制限	カラム数	カラム・サイズ 2K ページ	カラム・サイズ 4K ページ	カラム・サイズ 8K ページ	カラム・サイズ 16K ページ
固定長カラム	1024	1958 バイト	4006 バイト	8102 バイト	16294 バイト
可変長カラム	1024	1954 バイト	4002 バイト	8098 バイト	16290 バイト

ページ・サイズによるデータベース要件

データベースの必要領域は、サーバの論理ページ・サイズによって異なります。model データベースが下記の最小サイズよりも大きい場合、データベースの最小サイズは model データベースと等しくなります。

それぞれのデータベースのページ・サイズによるデータベース要件は次のとおりです。

データベース	2K ページ	4K ページ	8K ページ	16K ページ
master データベース	13MB	26MB	52MB	104MB

データベース	2K ページ	4K ページ	8K ページ	16K ページ
model データベース	3MB	6MB	12MB	24MB
tempdb データベース	4MB	6MB	12MB	24MB
sybsystemdb データベース	3MB	6MB	12MB	24MB
sybpcidb データベース	24MB	48MB	96MB	192MB

ページ・サイズによるテーブルのデータ量の制限

論理ページ・サイズが大きくなると、格納できるデータ量も増えます。

ページ・サイズによるテーブルのデータ量の制限は次のとおりです。

テーブル	2K ページ	4K ページ	8K ページ	16K ページ
インデックス・キーあたりのバイト数	600	1250	2600	5300
ユーザに見えるローの長さ DOL テーブル	1958	4006	8102	16294
ユーザに見えるローの長さ APL テーブル	1960	4008	8104	16296

Adaptive Server、Backup Server、XP Server の起動と停止を行います。

XP Server はインストール・プロセスでは起動されません。XP Server は、**isql** を使用して XP コマンドを発行したときに Adaptive Server によって起動されます。

Sybase Control Center を使用して、サーバを手動または自動で起動したり停止したりできます。Sybase Control Center の詳細については、『システム管理ガイド 第 1 巻』とオンライン・ヘルプを参照してください。

参照：

- interfaces ファイルから LDAP へのマイグレート (86 ページ)
- ディレクトリ・サービスへのサーバの追加 (83 ページ)
- Adaptive Server エラー・ログのパスの設定 (112 ページ)

サーバの起動

コマンド・ラインから Adaptive Server または Backup Server を起動します。

Adaptive Server または Backup Server は、`RUN_server_name` ファイルと **startserver** コマンドを使用して起動できます。`RUN_server_name` を編集して起動オプションをカスタマイズすることもできます。

サーバを起動するには、以下の要件を満たすユーザ・アカウントが必要です。

- データベース・サーバ (実行権限が必要) とデータベース・デバイス (読み込み/書き込み権限が必要) にアクセスできる。
- Adaptive Server の配布ファイルにアクセスできる。
- システム環境変数が設定されている。
- SySAM ライセンスにアクセスできる。『Sybase ソフトウェア資産管理ユーザーズ・ガイド』を参照してください。

コンピュータにサーバをインストールするときに、インストール・プログラムによって `interfaces` ファイルが作成され、システム環境変数が設定されます。

サーバの起動パラメータ

Adaptive Server または Backup Server を起動するには、起動パラメータを正しいロケーションおよびデバイス名に設定します。

デフォルトの起動パラメータは、`$SYBASE/$SYBASE_ASE/install/RUN_server_name` に格納されています。`server_name` は、インストールしたサーバの名前です。

デフォルトの起動パラメータを変更するには、`$SYBASE/$SYBASE_ASE/install/RUN_server_name` ファイルを編集する以外に方法がありません。`$SYBASE/$SYBASE_ASE/install/RUN_server_name` ファイル内で追加の起動パラメータを指定することもできます。

Backup Server 名には "_BAK" の文字列が付加されます。

表 2 : Adaptive Server のデフォルトの起動パラメータ

スイッチ	説明
-d \$SYBASE/data/master.dat	マスタ・デバイス・ファイルのロケーション
-s server_name	Adaptive Server の名前
-e \$SYBASE/\$SYBASE_ASE /install/errorlog	エラー・ログ・ファイルのロケーションと名前
-M \$SYBASE/\$SYBASE_ASE	共有メモリファイルを格納するディレクトリ
-N \$SYBASE/\$SYBASE_ASE/sysam/<srv_name>.properties	ライセンス・キャッシュ・ファイルのロケーションと名前

『ユーティリティ・ガイド』の **dataserver** と **backupserver** の説明に記載されている有効なサーバ・コマンド・ライン・オプションは、いずれも追加の起動パラメータとして指定できます。

RUN_server_name ファイル

新しい Adaptive Server または Backup Server を作成するたびに、**srvbuild** プログラムによって、サーバの再起動に必要な情報が格納された `RUN_server_name` ファイルが作成されます。

`RUN_server_name` ファイルは `$SYBASE/$SYBASE_ASE/install` ディレクトリに作成されます。

Adaptive Server の場合、`RUN_servername` ファイルの名前は `RUN_servername` です。`servername` はサーバの名前です。たとえば、Adaptive Server の名前が `TEST` の場合、その `RUN_servername` ファイルの名前は `RUN_TEST` になります。

Backup Server の場合、`RUN_servername` ファイルの名前は `RUN_servername_back` です。

警告！ `$SYBASE/$SYBASE_ASE/install` に作成された `RUN_servername` ファイルは削除しないでください。インストール環境をカスタマイズしたときに、このファイルによってサーバが再起動されます。別のロケーションに `RUN_servername` ファイルが必要な場合、元の `RUN_servername` ファイルを新しいロケーションにコピーしてください。

startserver コマンドの使用

コマンド・ラインからサーバを起動するには、**startserver** を使用します。

前提条件

マスタ・デバイスに対する読み込み/書き込みパーミッションが必要です。

手順

コマンド・ラインからサーバを起動するには、次のように入力します。
`$SYBASE/$SYBASE_ASE/install/startserver [-f RUN_server_name file]`

`$SYBASE/$SYBASE_ASE/install/startserver` は **startserver** ユーティリティのフル・パス名、`RUN_server_name file` は `RUN_server_name` ファイルのフル・パス名 (通常は `$SYBASE/$SYBASE_ASE/install/RUN_server_name`) です。

Adaptive Server の名前が `SYBASE` の場合、`RUN_server_name` ファイルの指定はオプションになります。

startserver の詳細については、『ASE ユーティリティ・ガイド』を参照してください。

オペレーティング・システムの再起動時のサーバの起動

Adaptive Server と Backup Server が自動的に再起動するようにオペレーティング・システムを設定します。

運用システムでは、UNIX オペレーティング・システムの起動時に Adaptive Server と Backup Server が自動的に再起動するように設定してください。そのためには、オペレーティング・システムの起動スクリプトにサーバのエントリを作成します。

注意： 起動スクリプトは、サーバの起動前に、必要なすべての Sybase 環境変数を設定する必要があります。つまり、SYBASE.csh または SYBASE.sh を情報源として読み込む必要があります。

HP-UX 用のシステム再起動スクリプトの作成

オペレーティング・システムの起動時または停止時に実行するスクリプトを作成します。

1. システム管理者は、オペレーティング・システムの起動時または停止時に実行するスクリプトを作成する必要があります。
HP-UX テンプレート・ファイル /sbin/init.d/template を基にしてスクリプト・ファイルを作成できます。HP-UX バージョン 10.0 以降では、/etc/rc ファイル内のコマンドを編集できません。**rc (1M)** の詳細については、HP-UX のマニュアル・ページを参照してください。
2. 起動と停止のスクリプトを作成したら、/sbin/init.d ディレクトリに保存します。
このディレクトリ内の実行スクリプトには、/sbin/rcn.d ディレクトリへのシンボリック・リンクがあります。*n* は、システムの実行レベルです。/
sbin/rcn.d 内のリンクされたスクリプトは、実行スクリプトのシーケンスを制御します。

IBM RS/6000 用のシステム再起動スクリプトの作成

運用システムでは、UNIX オペレーティング・システムが再起動するときに、必ず Adaptive Server が自動的に再起動するように設定できます。

前提条件

Adaptive Server が起動する前に、すべてのネットワーク・リソースが使用可能になっているようにしてください。使用できないリソースがある場合、Adaptive Server が起動しません。また、ネットワークが立ち上がっていない場合も、サーバは起動しません。

手順

Adaptive Server を自動的に再起動するには、**startserver** コマンドを /etc/inittab ファイルに追加します。

/etc/inittab に追加する Adaptive Server 起動コマンドの推奨フォーマットを次に示します。

```
"sybase:2:wait:/release_directory/install/startserver -f ¥  
RUN_servername /dev/console 2>&1"
```

`release_directory` は Sybase インストール・ディレクトリへのフル・パス (SYBASE 環境変数として指定されている)、`RUN_servername` は起動するサーバの `RUN_server_name` ファイルです。

`/etc/inittab` ファイル内で、Adaptive Server を起動するエントリが `/etc/rc.tcpip` と `/etc/rc.nfs` のエントリより後になるようにします。

ネットワーク・オペレーションの開始に時間がかかる場合、`/etc/rc.tcpip` ファイル内のコマンドが正しい順序で並んでいても、Adaptive Server が起動しないことがあります。`RUN_server_name` ファイル内のサーバ起動コマンドの前に **sleep** コマンドを挿入して、Adaptive Server が起動するまでの待ち時間を指定できます。**sleep** コマンドの形式は次のとおりです。

```
sleep seconds_of_rest
```

注意： オペレーティング・システムの起動時に Backup Server を再起動するには、Backup Server の起動コマンドを `/etc/inittab` に追加します。このコマンドでは、Backup Server の `RUN_server_name` ファイルのパスを使用します。

Sun Solaris および Linux 用のシステム再起動スクリプトの作成

オペレーティング・システムの起動時または停止時に Adaptive Server または Backup Server を自動的に起動または停止するスクリプトを作成します。

前提条件

すべてのネットワーク・リソースが使用可能であることを確認します。ネットワークが立ち上がっていない場合、サーバは起動しません。必ず、`rc` ディレクトリ内で、サーバを再起動するエントリがネットワーク・オペレーションを開始するコマンドより後になるようにしてください。ネットワーク・オペレーションの開始に時間がかかる場合、`rc` ディレクトリ内のコマンドが正しい順序で並んでいても、サーバが起動しないことがあります。

手順

作成したリンク先の `RUN_server_name` ファイルの最初に **sleep** コマンドを挿入して、サーバが起動するまでの待ち時間を指定できます。

1. 起動スクリプト・ファイルを次のような内容で作成します。

```
$SYBASE/$SYBASE_ASE/install/startserver      -f $SYBASE/
$SYBASE_ASE/install/RUN_servername
```

`$SYBASE/$SYBASE_ASE/install/startserver` は **startserver** ユーティリティのフル・パス名、`$SYBASE/$SYBASE_ASE/install/`

RUN_servername はサーバの RUN_server_name ファイルのフル・パス名です。

2. 次のような構文を使用して、スクリプトを /etc/init.d ディレクトリにコピーします。

```
cp script_name /etc/init.d/script_name
```

3. /etc/inittab ファイルの内容を調べて、使用しているオペレーティング・システムのデフォルトの実行レベルを確認します。**initdefault** エントリで、デフォルトの実行レベルが指定されています。通常、Sun Solaris の場合は 2 または 3、Linux の場合は 5 です。
4. **ln** コマンドを使用して、RUN_server_name ファイルから適切な run control (rc) ディレクトリ rc# へのハード・リンクを作成します。#は、手順 2 で取得したデフォルトの実行レベルです。

次のような Sun Solaris 構文を使用します。

```
ln /etc/init.d/script_name /etc/rc#.d/S##script_name
```

```
ln /etc/init.d/script_name /etc/rc#.d/S##script_name
```

次のような Linux 構文を使用します。

ln -s を使用して、スクリプト名から適切な rc (run control) ディレクトリ rc# へのシンボリック・リンクを作成します。#は実行レベルです。次のような構文を使用します。

```
ln -s /etc/init.d/script_name /etc/rc#.d/S##script_name
```

このリンクを作成するコマンドを入力するときに、スクリプト・ファイル名の前に大文字の“S”と2桁のシーケンス番号を追加します。“S”は起動ファイルを示します。rc ディレクトリ内のファイルは番号順に実行されるため、シーケンス番号が必要です。このファイルは最後に実行するので、ASCII 順ですべての既存の番号より後のシーケンス番号を使用します。

rc ディレクトリに対して **ls** コマンドを実行して、既存のシーケンス番号を確認できます。次に例を示します。

```
ls /etc/rc3.d/S*
```

次のような結果が返されます。

```
/etc/rc3.d/S10syslog
```

```
/etc/rc3.d/S15nfs.server
```

```
/etc/rc3.d/S21rfs
```

スクリプト名が **sybstart** の場合、次のように入力します。

```
ln /etc/init.d/sybstart /etc/rc3.d/S77sybstart
```

この例では、77 の代わりに 27 より大きい任意の数字を指定できます。

Linux の場合：

rc ディレクトリに対して **ls** コマンドを実行して、既存のシーケンス番号を確認できます。次に例を示します。

```
ls /etc/rc5.d/S*
```

次のような結果が返されます。

```
/etc/rc5.d/S12syslog
/etc/rc5.d/S14nfslock
/etc/rc5.d/S27ypbind
```

訳文不要スクリプト名が **sybstart** の場合、次のように入力します。

```
ln -s /etc/init.d/sybstart /etc/rc5.d/S99sybstart
```

この例では、99 の代わりに 27 より大きい任意の数字を指定できます。

最初のインストール後の XP Server の起動

XP Server についての情報を **sysservers** テーブルに追加します。

同一の構築セッションで Adaptive Server と XP Server をインストールする場合は、**srvbuild** によって、XP Server についての情報が Adaptive Server の **syservers** テーブルに自動的に追加されます。Adaptive Server インストールとは別の構築セッションで XP Server をインストールする場合は、XP Server のインストール中に、関連する Adaptive Server の名前、およびシステム管理者の名前とパスワードを入力するように要求されます。この情報を使用して、Adaptive Server は XP Server を起動します。

インストール中にこれらの情報を入力しないと、XP Server を実行できません。拡張ストアド・プロシージャ (ESP) を実行すると、次のようなエラー・メッセージが表示されます。

```
Msg 11018, Level 16, State 1:
Procedure 'xp_cmdshell', Line 2
XP Server must be up for ESP to execute.
```

この必須情報を **syservers** テーブルに手動で追加するには、次のように入力します。

```
sp_addserver SERVERNAME_XP, NULL, SERVERNAME_XP
```

servername は Adaptive Server の名前です。

サーバの停止

shutdown コマンドを発行できるのは、システム管理者のみです。**shutdown** コマンドを使用すると、サーバの再起動時における自動リカバリの作業量が最小限に抑えられます。

Adaptive Server または Backup Server の停止方法としては、Transact-SQL の **shutdown** コマンドの使用をおすすめします。

Adaptive Server の停止

Adaptive Server を停止できるのは、システム管理者のみです。

1. **isql** を使用して、システム管理者権限で Adaptive Server アカウントにログインします。

```
isql -Usa -Ppassword -Sserver_name
```

2. 次のように入力します。

```
1> shutdown
2> go
```

shutdown のデフォルトでは **with wait** オプションが使用されます。このオプションを使用すると、Adaptive Server は SQL 文またはプロシージャの実行の終了、各データベースのチェックポイントの実行、新しいログインの無効化などの停止作業を行うことができます。

shutdown コマンドを発行すると、次のようなメッセージが **sterr** ファイルに出力されます。

```
Server SHUTDOWN by request.The SQL Server is terminating this
process.
CT-LIBRARY error:
```

これは正常な動作です。Adaptive Server がプロセスの完了を待っていることを示すメッセージが表示されているときに Adaptive Server をすぐに停止する必要がある場合は、**shutdown with nowait** を使用できます。このコマンドは、現在実行中の文が終了するのを待たず、各データベースのチェックポイントも実行しません。

注意： **shutdown with nowait** コマンドは、必要な場合以外は使用しないことをおすすめします。

Backup Server の停止

Backup Server を停止できるのは、システム管理者のみです。

1. **isql** を使用して、システム管理者権限でサーバにログインします。
2. 次のように入力します。

```
1> shutdown SYB_BACKUP
2> go
```

Backup Server を停止したら、最低 30 秒間待ってから再起動してください。

shutdown コマンドを発行すると、次のようなメッセージが `stderr` ファイルに出力されます。

```
Backup Server: 3.48.1.1: The Backup Server will go down
immediately.
Terminating sessions.
```

これは正常な動作です。Adaptive Server または Backup Server がプロセスの完了を待っていることを示すメッセージが表示されているときに Adaptive Server または Backup Server をすぐに停止する必要がある場合は、**shutdown with nowait** を使用できます。このコマンドは、現在実行中の文が終了するのを待たず、各データベースのチェックポイントも実行しません。

注意： **shutdown with nowait** コマンドは、必要な場合以外は使用しないことをおすすめします。

『リファレンス・マニュアル：コマンド』を参照してください。

kill コマンドの使用

kill コマンドを使用して、Adaptive Server および Backup Server のプロセスを停止できます。

警告！ **kill** コマンドは、Adaptive Server と Backup Server を停止する最後の手段として使用してください。

できるかぎり、Transact-SQL の **shutdown** コマンドまたは **shutdown with nowait** コマンドを使用してください。**-9** フラグを指定して **kill** コマンドを実行しないでください。この方法を使用すると、データベースに対する変更がすべてデータベース・デバイスに書き込まれたことを確認するチェックポイントを実行せずにサーバが終了します。また、Adaptive Server が対応する共有メモリ・ファイルとネットワーク・ハンドラを削除せずに終了する場合があります。

Adaptive Server と Backup Server はバックグラウンド・プロセスであるため、所有者または root ユーザは UNIX の **kill** コマンドを使用して、オペレーティング・システムから強制終了できます。構文は次のとおりです。

```
kill pid
```

pid は、**showserver** コマンドで示される **dataserver** または **backupserver** プロセスのプロセス ID です。特定の Adaptive Server のエンジンを 1 つでも強制終了すると、そのサーバのすべてのエンジンが強制終了されます。

複数の Adaptive Server が同一システム上で稼働している場合、強制終了するエンジンが正しい Adaptive Server と対応していることを確認します。Adaptive Server が複数のエンジン (CPU) を使用するように設定されている場合、エンジンはそれぞれオペレーティング・システムのプロセスと対応しています。マルチエンジン・サーバを強制終了する正しい方法は、エンジンのプロセス ID を 0 に指定することです。

次の **showserver** 出力は、Adaptive Server がプロセス・カーネル・モードで稼働している場合の、4 つのエンジンを持つサーバのプロセスを示しています。デフォルトのスレッド・モードの場合、それぞれのエンジンはプロセスではないため、**showserver** では 1 つのプロセスのみが示されます。

showserver

UID	PID	PPID	C	STIME	TTY	TIME	COMD
jorge	3320	1	80	10:31:40	pts/4	302:15	dataserver -dteamster
jorge	3321	3320	80	10:31:45	pts/4	324:47	dataserver -ONLINE:1
jorge	3322	3320	80	10:31:45	pts/4	326:02	dataserver -ONLINE:2
jorge	3323	3320	80	10:31:45	pts/4	328:56	dataserver -ONLINE:3

この例は、オペレーティング・システムのプロセス ID (PID) が 3320、3321、3322、3323 である 4 つの実行中の **dataserver** プロセスを示しています (**dataserver** は、実行形式の Adaptive Server プログラムです)。

dataserver の子エンジン・プロセスには、**-ONLINE:** 引数があります。

各子エンジンの親プロセス ID (PPID) は、親のプロセス ID (PID) と同じです。前に示した例では、親サーバの PID が 3320 になっています。親プロセスによって生成された他の 3 つのエンジンも PPID は同じです。

PPID に関連性がなく、複数の **dataserver** プロセスがある場合は、複数の Adaptive Server がシステム上で稼働しています。

停止と共有メモリ・ファイル

Adaptive Server または Backup Server を停止する際に選択した方法が共有メモリ・セグメントに影響することがあります。

Adaptive Server が起動すると、`$SYBASE/$SYBASE_ASE directory に SERVER_NAME.krg` ファイルが作成され、Adaptive Server が使用する共有メモリ・セグメントについての情報が格納されます。

Adaptive Server に設定されているメモリ・サイズがオペレーティング・システムの `MAXSHMSEG` パラメータよりも大きい場合、Adaptive Server は共有メモリ・

セグメントを追加作成します。追加作成する共有メモリ・セグメントごとに、`SERVER_NAME.srg[N]` (N の範囲は $0 \sim N$) という名前の追加ファイルが `$$SYBASE/$$SYBASE_ASE` に作成されます。

Adaptive Server が正常に停止すると、共有メモリ・ファイルは自動的に削除されます。Adaptive Server で障害が発生した場合や、**kill -9** コマンドを使用して Adaptive Server を停止した場合、これらのファイルは削除されません。障害が発生した後や、**kill -9** コマンドを使用して停止した後で Adaptive Server を再起動するには、これらのファイルに対する読み込み/書き込みパーミッションが必要です。これは、以前に作成された共有メモリ・ファイルを Adaptive Server が上書きできなければならないためです。

Adaptive Server または Backup Server を強制終了した場合も、共有メモリ・セグメントが残ります。**ipcs** コマンドと **ipcrm** コマンドを使用して、"NATTACH" カウントが 0 を示している共有メモリ・セグメントを特定し、削除してください。

ipcs と **ipcrm** の詳細については、UNIX の man ページを参照してください。

Adaptive Server では、x64 および P-series Linux バージョンのヒュージ・ページをサポートしています。

CPU キャッシュ TLB (Translation Lookaside Buffer) には、仮想ページ・アドレスから物理ページ・アドレスへの変換に関する情報が保管され、物理メモリへのバイト・アクセスごとに変換 (キャッシュ・ミス) が必要となります。キャッシュ・ミスはコストがかかりますが、Linux マシンで「ヒュージ・ページ」を有効にすると、TLB ヒットを向上させることができます。

ヒュージ・ページでは、物理アドレス空間をカバーするために使用するページが少ないので、「トランケーション・ポイントの書き換え」(仮想アドレスから物理アドレスへのマッピング)のサイズが小さくなります。そのため、TLB 内で必要なエントリが減り、システムのパフォーマンスが向上します。

Adaptive Server バージョン 15.0.3 以降では、デフォルトでヒュージ・ページを使用して共有メモリを割り付けます。ただし、システムに十分なヒュージ・ページがない場合、またはヒュージ・ページを使用するように設定されていない場合には、Adaptive Server は通常のページを使用し、次のメッセージをエラー・ログに書き込みます。

```
Could not allocate memory using Huge Pages. Allocated using regular pages. For better performance, reboot the server after configuring enough Huge Pages
```

Adaptive Server は、共有メモリを 256MB の最も近い倍数に調整します。たとえば、800MB の共有メモリを指定して Adaptive Server を設定した場合、1GB に丸められます (Linux の一部のバージョンでは、サイズが **Hugepagesize** の倍数でないと、ヒュージ・ページを割り付けることはできません)。

次のように `/proc/meminfo` をチェックして、Linux でヒュージ・ページがすでに設定されていることを確認してから、Adaptive Server を起動してください。

```
cat /proc/meminfo
....
HugePages_Total:    32
HugePages_Free:     32
Hugepagesize:       16384 kB
```

注意： ヒュージ・ページ用に割り付けたメモリは、共有メモリのみに使用されます。割り付けたヒュージ・ページが多すぎると、Adaptive Server が物理ページを過度にスワップすることがあります。ヒュージ・ページは、必要な数だけ割り付けてください。

Adaptive Server をインストールまたは更新した後、設定を調整します。

Adaptive Server の一部の設定は、インストール中に行うことができます。特に明記されていないかぎり、設定情報は、サポートされている UNIX プラットフォームすべてに適用できます。詳細については、『インストール・ガイド』を参照してください。

環境変数の確認

Adaptive Server の環境変数が正しく設定されていることを確認するには、オペレーティング・システムのプロンプトで **env** を使用します。

参照：

- 環境変数 (2 ページ)

stty 設定の使用

バックグラウンドの Adaptive Server が端末に出力を書き込まないようにするには、**stty tostop** を設定します。

バックグラウンドの出力を停止するには、Adaptive Server を起動する前に次のコマンドを入力します。

```
stty -tostop
```

Adaptive Server のすべての出力をファイルにリダイレクトする場合、**stty** 設定を変更する必要はありません。

正しいパーミッションのリストア

Sybase ソフトウェアのファイルとディレクトリには、インストール時に正しいアクセス・パーミッションが設定されます。必要に応じて、\$SYBASE/\$SYBASE_ASE/install にある setperm_all スクリプトを使用して正しいパーミッションをリストアできます。

ファイル記述子とユーザ接続

Adaptive Server によって使用されるユーザ接続の数は、オペレーティング・システムで Adaptive Server が使用できるファイル記述子の数を超えることはできません。

Adaptive Server のユーザ接続を設定する場合、システム管理者は 1 つのプロセスで使用可能なファイル記述子の数を考慮に入れる必要があります。オープン可能なファイル記述子のほとんどはユーザ接続で使用できます。Adaptive Server によってファイルとデバイスのオープンに使用されるものはごく一部です。

HP-UX のファイル記述子とユーザ接続

現在のファイル記述子の値を検索または変更します。

カーネル・パラメータの **maxfiles** と **maxfiles_lim** によって、1 つのプロセスで使用可能なファイル記述子の数が制御されます。64 ビット・システムの場合、制限値は 60,000 です。

現在のファイル記述子の値を検索するには、次のコマンドを使用します。

```
ulimit -n
```

AIX のファイル記述子とユーザ接続

現在のファイル記述子の値を検索または変更します。

1 プロセスあたりのファイル記述子の数は、オペレーティング・システムのパラメータ **open_max** によって決まります。**open_max** のデフォルト値は 32767 です。Adaptive Server は、**open_max** の値に関係なく、最大 65534 個のファイル記述子を使用できます。AIX 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。

現在の **open_max** の値を取得するには、次のコマンドを使用します。

```
ulimit -n
```

Linux のファイル記述子とユーザ接続

現在のファイル記述子の値を検索または変更します。

1 プロセスあたりのファイル記述子の数は 10,000 に制限されています。ファイル記述子の数を設定するには、**ulimit** を使用します。Adaptive Server は、最大 40000 個のファイル記述子を使用できます。

Sun Solaris のファイル記述子とユーザ接続

現在のファイル記述子の値を検索または変更します。

Sun Solaris では、ファイル記述子に対してソフト制限値とハード制限値の両方を設定できます。ソフト制限値はハード制限値を上限としてユーザが増やせますが、ハード制限値を増やせるのは root パーミッションを持ったユーザだけです。ソフト制限値によって、Adaptive Server エンジンでオープン可能なファイル記述子の数が決まります。制限値は 10,000 です。

オープン可能なファイル記述子のほとんどはユーザ接続で使用できます。Adaptive Server によってファイルとデバイスのオープンに使用されるものはごく一部です。

『システム管理ガイド 第 1 巻』の「設定パラメータ」を参照してください。

現在のソフト制限値とハード制限値の表示

C シェルおよび Bourne シェルの現在のソフト制限値とハード制限値を表示できません。

現在のソフト制限値またはハード制限値を表示する手順は次のとおりです。

- C シェルのソフト制限値を表示するには、次のように入力します。
`limit descriptors`
- Bourne シェルのソフト制限値を表示するには、次のように入力します。
`ulimit -n`
- C シェルのハード制限値を表示するには、次のように入力します。
`limit -h descriptors`
- Bourne シェルのハード制限値を表示するには、次のように入力します。
`ulimit -Hn`

ソフト制限値を増やす方法

C シェルおよび Bourne シェルのソフト制限値を増やします。

- ソフト制限値を増やすには、C シェルの場合次のように入力します。
`limit descriptors n`
- ソフト制限値を増やすには、Bourne シェルの場合次のように入力します。
`ulimit -Sn new_value`

n はソフト制限値の現在値、*new_value* は増加後のソフト制限値です。

注意： 上記のコマンドを `RUN_server_name` ファイル内で使用して、ハード制限値とソフト制限値を増やすことができます。 `RUN_server_name` ファイルは

Bourne シェル・スクリプトなので、RUN_server_name ファイル内では必ず Bourne シェル用のコマンドを使用してください。

ハード制限値を増やす方法

プログラムを設定してハード制限値を増やします。

1. file_name.c を作成します (file_name は、ファイルに付ける名前です)。
2. 次のようなテキストを入力します。状況に応じて適宜変更を加えてください。 .

注意：これはサンプル・スクリプトです。必要に応じて変更してください。

```
#include <sys/time.h>
#include <sys/resource.h>
#include <sys/types.h>
/*
** define MAX_CONNECTIONS to a number less than
** 10000. The number defined will then become the maximum
** number of connections allowed by an Adaptive Server.
*/
#define MAX_CONNECTIONS 9999
extern int errno;

main(argc,argv)
char **argv;
{
    struct rlimit rlp;
    uid_t uid;

    rlp.rlim_cur = MAX_CONNECTIONS;
    rlp.rlim_max = MAX_CONNECTIONS;
    /* set the number of open file descriptors to
    MAX_CONNECTIONS */
    if (setrlimit (RLIMIT_NOFILE,&rlp) == -1)
    {
        perror("setrlimit");
        exit(1);
    }

    /* reset the user id to disable superuser
    privileges */
    uid = getuid();
    setuid(uid);
    /* run the program indicated as arguments to
    this program */
    execv(++argv, argv);
}
```

3. 次のように入力してファイルをコンパイルします。

```
cc file_name.c -o program_name
```

program_name は、コンパイルされたファイルの名前です。

4. プログラムのパーミッションと所有権を変更します。

```
chmod 755 program_name
chown root program_name
```

5. root ユーザは、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することにより、ユーザ接続の数を増やして Adaptive Server を起動できます。

```
# program_name dataserver -d master_device_name
```

master_device_name は、Adaptive Server のマスタ・デバイスのフル・パスです。また、Adaptive Server の *RUN_server_name* ファイル内で **dataserver** コマンド・ラインの前に *program_name* を追加することもできます。

『システム管理ガイド 第 1 巻』の「設定パラメータ」を参照してください。

非同期ディスク I/O の有効化

Linux、IBM、HP-UX で非同期ディスク I/O を有効にします。

UNIX システムと Linux システムでは、データベース・デバイスとしてブロック・デバイスを使用しないことをおすすめします。ブロック・デバイスを使用した場合、システムの障害によってデータの整合性が失われるおそれがあります。

UNIX プラットフォームでロー・デバイスを使用する場合は、次のように設定できません。

- **disk init...directio** または **dsync** パラメータを **true** に設定する
- **sp_deviceattr...directio** または **dsync** パラメータを **true** に設定する

これらのいずれかが **true** に設定されていると、Adaptive Server は次のようなメッセージを返します。

```
You cannot set directio option for raw device '/dev/raw/raw235' or
You cannot set attribute dsync for raw device
```

Linux での非同期ディスク I/O の有効化

Linux で非同期 I/O を有効にします。

前提条件

Adaptive Server を起動する前に、カーネル非同期 I/O に十分なシステム・リソースがあることを確認してください。

システム全体の予約可能な I/O 記述子の総数 (aio-max-nr) から、すべてのプロセスで予約されている現在の記述子の数 (aio-nr) を引いた値が、**max async i/os per engine** の値以上でなければなりません。

手順

1. aio-max-nr と aio-nr の値を確認するには、次のコマンドを使用します。

```
cat /proc/sys/fs/aio-max-nr
cat /proc/sys/fs/aio-nr
```

2. 予約可能な記述子の数を変更するには、**sysctl** オペレーティング・システム・コマンドを使用して fs.aio-max-nr を設定します。詳細については、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。

起動時に I/O 記述子の数が不足している場合や、エンジンがオンラインになっている場合、Adaptive Server はエラー・ログに次のようなメッセージを返すことがあります。

```
kernel KAIIO not initialized because the requested number of async
I/Os(%d) will exceed the resources available on the operating
system.
```

```
kernel Kernel asynchronous I/O not initialized. The io_setup()
system call returned %d.
```

『システム管理ガイド 第 1 巻』の「設定パラメータ」の「**max async i/os per engine**」を参照してください。

IBM AIX での非同期ディスク I/O の有効化

IBM AIX で、カーネル・パラメータを調整して非同期 I/O を有効にするには、SMIT (System Management Interface Tool) を使用します。

1. UNIX プロンプトで smit と入力します。
2. [Devices] メニューから [Asynchronous I/O] を選択します。
3. [Change/Show Characteristics of Asynchronous I/O] を選択します。

HP-UX での非同期ディスク I/O の有効化

文字型 (ロー) デバイスやブロック・デバイス上の I/O パフォーマンスを向上させるには、SAM から HP の非同期 I/O ドライバをインストールして非同期 I/O を有効にします。

1. Adaptive Server を停止します。

2. SAM の [Kernel Configuration] メニューから、[Drivers] を選択し、*asyncdisk* の保留状態を [In] に設定して、ドライバを追加します。または、*asyncdisk* サブシステム・キーワードを */stand/system* に追加します。
3. カーネルを再構築して、システムを再起動します。
4. `userid root` を使用して、次のコマンドを実行します。

```
#/etc/mknod/dev/async c 101 4
#chmod 0660/dev/async
#chown <uid> /dev/async
#/etc/setprivgrp <ugrp> MLOCK
```

各パラメータの意味は次のとおりです。

- `<uid>` は、Adaptive Server を再起動するユーザが使用するユーザ ID です。このユーザは、`/dev/async` ディレクトリの所有者でなければなりません。
- `<ugrp>` は、`<uid>` ユーザ ID のユーザ・グループです。

ファイル・システム・デバイスの非同期 I/O の有効化

HP-UX でファイル・システム・デバイスの非同期 I/O を有効にします。

前提条件

次のオペレーティング・システム・パラメータを最大値に調整することをおすすめします。

```
# kctune aio_max_ops=0x100000
# kctune aio_proc_threads=2048
```

手順

1. `allow sql server async i/o` 設定パラメータを有効にします。
2. `enable hp posix async i/o` 設定パラメータを有効にします。このパラメータは静的です。

```
sp_configure 'enable hp posix async i/o', 1
```

3. Adaptive Server を再起動します。
『システム管理ガイド 第 1 巻』の「設定パラメータ」を参照してください。

ファイル・システムの非同期 I/O の有効化

HP-UX でファイル・システムの非同期 I/O を有効にします。

前提条件

HP-UX ファイル・システムでパフォーマンスを向上させるために、VxFS 同時実行 I/O (CIO) を有効にすることをおすすめします。VxFS 同時実行 I/O は、HP-UX 11.31 の OnlineJFS (VxFS-Full) バージョン 5.0.1 以降で使用できます。

手順

OnlineJFS がインストールされていて有効になっているかどうかを確認するには、次のように入力します。

```
# vxlicrep | grep -i onlinejfs
```

または

```
# swlist -l product |grep -i onlinejfs
```

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。

クライアント接続のタイムアウト時間の調整

Adaptive Server では、TCP/IP プロトコルの **KEEPALIVE** オプションを使用して、非アクティブなクライアントを検出します。

クライアントへの接続が所定の時間 (タイムアウト時間) 非アクティブであった場合、オペレーティング・システムは **KEEPALIVE** パケットを一定間隔で送信します。これらのパケットに対してクライアント・マシンから応答がない場合、オペレーティング・システムはクライアントが応答しなくなったことを Adaptive Server に通知します。その後、Adaptive Server はそのクライアントの接続を終了します。

KEEPALIVE のデフォルトのタイムアウト時間は 2 時間 (7,200,000 ミリ秒) です。現在の時刻の値を表示するには、使用しているプラットフォームに適したコマンドを使用してください。

HP-UX のクライアント接続タイムアウトの調整

現在のタイムアウト時間を表示または変更します。

1. 現在のタイムアウト時間を表示するには、次のコマンドを入力します。

```
/ndd -get/set /dev/tcp tcp_keepalive_interval
```

tcp_keepalive_interval パラメータは、接続が切断されたかどうかをシステムがチェックするまで、アイドル状態の接続をアクティブなまま保持する時間 (秒単位) を指定します。

2. タイムアウト時間を変更するには、**net tune -s** コマンドまたは **ndd -set** コマンドを使用します。

IBM RS/6000 のクライアント接続タイムアウトの調整

現在のタイムアウト時間を表示し、必要に応じて調整します。

現在のタイムアウトの値を表示するには、次のコマンドを入力します。

```
/usr/sbin/no -o tcp_keepidle
```

tcp_keepidle パラメータは、接続が切断されたかどうかをシステムがチェックするまで、アイドル状態の接続をアクティブなまま保持する時間 (0.5 秒単位) を指定します。デフォルトは 14,400 X 0.5 秒 (7,200 秒つまり 2 時間) です。

IBM では、15 分以上の値を使用することを推奨しています。

Linux のクライアント接続タイムアウトの調整

現在のタイムアウト時間を表示または変更します。

1. タイムアウトの値を表示または変更するには、次のコマンドを入力します。

```
/sbin/sysctl -e net.ipv4.tcp_keepalive_time
```

2. 必要に応じて、タイムアウト時間を調整します。

たとえば、タイムアウト時間を 15 分 (900 秒) に短縮するには、次のコマンドを入力します。

```
/sbin/sysctl -w net.ipv4tcp_keepalive_time=900
```

ハードウェア・エラーのチェック

ハードウェア・エラーを定期的にチェックします。

次のようなエラー・メッセージは、データベースの破壊につながる可能性のある問題を示します。

- ディスクの読み込みエラー、書き込みエラー、またはリトライ・エラー
- タイムアウト
- システム障害
- メモリに関するあらゆる種類の問題
- HP-UX の場合

/var/adm/syslog/syslog.log ファイルを定期的にチェックします。このファイルは直接表示できますが、HP-UX の **dmesg** コマンドを使用する方法もあります。詳細については、HP-UX 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。

- IBM RS/6000 の場合

errpt コマンドには、一定の基準を満たすイベントにレポートを限定するオプションがいくつか用意されています。**errpt** コマンドを定期的に使用します。エラーが表示された場合は、診断ツール **diag** を使用してメモリとディスクをチェックします。または、SMIT (System Management Interface Tool) を使用して、**errpt** コマンドを実行します。このコマンドを実行すると、大量の出力が生成される場合があります。

- Sun Solaris の場合

/var/adm/messages ファイルを定期的にチェックします。前述のようなハードウェア・エラーが表示された場合は、Sun Microsystems の診断ツール

sunddiag を使用してメモリとディスクをチェックします。詳細については、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。

- **Linux の場合**

`/var/log/messages` ファイルを定期的にチェックします。詳細については、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。

オペレーティング・システム・リソースの使用状況のモニタ

オペレーティング・システムのパフォーマンスをモニタします。

『システム管理ガイド 第2巻』の「マルチプロセッサ・サーバの管理」では、負荷とシステム設定に応じた最適な Adaptive Server エンジンの数について説明しています。

HP-UX のオペレーティング・システム・リソースのモニタ

HP-UX のツールを使用して、パフォーマンスをモニタします。

- **sar** — 各ディスクとコントローラに対する I/O スループットの相対比率と絶対比率をレポートする。
- **vmstat** — 仮想メモリの使用状況をモニタする。
- **netstat** — ネットワーク・ステータスをモニタする。
- **ps** — 個々のプロセスの累積 CPU 時間と CPU 使用率のスナップショットを表示する。
- **time** — 実行が完了するまでに使用されたさまざまなユーザ・リソース、システム・リソース、リアルタイム・リソースを確認する場合に役立つ。

これらのツールの詳細については、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。

IBM RS/6000 のオペレーティング・システム・リソースのモニタ

IBM RS/6000 のツールを使用して、パフォーマンスをモニタします。

- **iostat** — 端末とハード・ディスクの入出力の量、および CPU 時間の使用状況をレポートする。
- **vmstat** — 仮想メモリの使用状況をモニタする。
- **netstat** — ネットワーク・ステータスをモニタする。
 - **netstart -v** — 送信統計と受信統計を表示する。ネットワーク・トラフィックに十分なバッファが設定されているか判断するときにも使用する。
 - **no -a** — 現在のネットワーク・オプションを表示する。mbuf プールのチューニングにも使用する。

- **ps** — 個々のプロセスの累積 CPU 時間と CPU 使用率のスナップショットを表示する。
- **time** — 実行が完了するまでに使用されたさまざまなユーザ・リソース、システム・リソース、リアルタイム・リソースを確認する。

これらのツールの詳細については、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。

Sun Solaris と Linux のオペレーティング・システム・リソースのモニタ

Sun Solaris と Linux のツールを使用して、パフォーマンスをモニタします。

- **iostat** — 端末とハード・ディスクの入出力の量、および CPU 時間の使用状況をレポートする。
- **vmstat** — 仮想メモリの使用状況をモニタする。
- **netstat** — ネットワーク・ステータスをモニタする。
- **ps** — 個々のプロセスの累積 CPU 時間と CPU 使用率の正確なスナップショットを表示する。このコマンドは、データサーバ、エンジン、プロセスの負荷を確認するときに役立つ。
- **time** — 実行が完了するまでに使用されたさまざまなユーザ・リソース、システム・リソース、リアルタイム・リソースを確認する場合に役立つ。

これらのツールの詳細については、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。

データベースの一貫性の検査

dbcc チェックを実行してデータベースのバックアップを行うことで、Adaptive Server データベースの一貫性とリカバリ性を維持します。

dbcc checkalloc または **dbcc checkdb** の実行所要時間より一貫性の検査の実行所要時間が短い場合は、**dbcc checkstorage** を使用して大きなデータベースで一貫性の検査を定期的に行ってください。**dbcc checkstorage** は一貫性の検査を短時間で実行しますが、**dbcc checkalloc** や **dbcc checkdb** ほど詳細な検査ではありません。『システム管理ガイド 第 2 巻』の「dbcc checkstorage を使用するための準備」と「データベースの一貫性の検査」、および『リファレンス・マニュアル：コマンド』を参照してください。

次の C シェル・スクリプトの例を実行して、複数の **isql** スクリプトを呼び出すこともできます。

```
#!/bin/csh -f
if ( -e dbcc_mail.out) then
    rm dbcc_mail.out
```

```
endif
foreach i (*.dbcc)
isql -Usa -Ppassword < $i > dbcc_out
if ( `grep -c 'Msg 25[0-9][0-9]' dbcc_out` ) then
    echo "There are errors in" $i >> dbcc_mail.out
    cat dbcc_out >> dbcc_mail.out
else
    echo "Backing up " $i:r >> dbcc_mail.out
    isql -Usa -Ppassword < $i:r.backup
endif
end
mail -s "Backup Report" jjones < dbcc_mail.out
```

スクリプトの最初のセット(各データベースに1つずつあり、ファイル名に .dbcc が付く)は、各データベースに対して **dbcc checkalloc** と **dbcc checkdb** を実行し、dbcc_out という出力ファイルにメッセージを送信します。

たとえば、スクリプト master.dbcc は、**dbcc** を実行して master データベースを検査します。

```
dbcc checkalloc (master)
go
dbcc checkdb (master)
go
```

その後、Cシェル・スクリプトは **grep** コマンドを実行して、**dbcc** 出力内の 2500 番台のエラー・メッセージを検索します。**grep** コマンドの結果は、dbcc_mail.out という出力ファイルに送信されます。

次に、このスクリプトは、2500 番台のエラーが発生しなかった各データベースについて **isql** バックアップ・スクリプトを呼び出し、"Backing up *database_name*" という行を dbcc_mail.out に追加します。たとえば、スクリプト master.backup は master データベースをバックアップします。

```
use master
go
dump database master to master_dump
go
```

適切な **dump transaction** コマンドをスクリプトに追加できます。

2500 番台のエラー・メッセージがある場合、スクリプトはデータベースをバックアップしません。スクリプトの最後で、dbcc_mail.out がシステム管理者 "jjones" にメールで送信されます。これによって、システム管理者は重大な **dbcc** エラーと正常なバックアップの記録を得ることができます。

前述のシェル・スクリプトと **isql** スクリプトの例は、インストール環境のニーズに合わせてカスタマイズできます。

スクリプトを自動的に実行するには、`crontab` ファイルを編集し、次のようなエントリを追加します。この場合、`dbcc_ck` という C シェル・スクリプトが毎日午前 2 時に実行されます。

```
00 02 * * * /usr/u/sybase/dbcc_ck 2>&1
```

この例では、**`dbcc_ck`** という C シェル・スクリプトが毎朝午前 2 時に実行されます。

srvbuild を使用した新しいサーバの設定

キー設定属性にデフォルト値またはユーザ指定の値を使用して、新しいサーバを設定する場合は、**srvbuild** を使用します。

注意： リソース・ファイルを使用して、非 GUI モードで新しいサーバを設定する場合は、**srvbuildres** を使用します。『ユーティリティ・ガイド』の「ユーティリティ・コマンド・リファレンス」を参照してください。

1. \$SYBASE 内の SYBASE.csh ファイルまたは SYBASE.sh ファイルを情報源として読み込み、環境変数を設定します。
2. \$SYBASE/\$SYBASE_ASE/bin/srvbuild を実行します。
3. 設定するサーバ・タイプをクリックします。各タイプを選択すると、サーバ名が有効になり、サーバ名を入力できます。[[OK]] をクリックします。

使用可能なサーバ・タイプは、\$SYBASE に何をインストールしたかによって異なります。

次に表示される画面は、選択したサーバ・タイプによって異なります。

4. [Adaptive Server type] 画面で、次の内容を選択します。
 - Adaptive Server アプリケーションのタイプ — 新しいサーバをどのように使用するかに基づいて、次の項目を選択します。
 - MIXED — OLTP と DSS の両方に使用する場合。
 - OLTP — オンライン・トランザクション処理に使用する場合。通常、複雑ではない小さなトランザクションの割合が高い場合に使用される。
 - DSS — 意思決定支援システムに使用する場合。通常、これらのシステムでは、更新処理がほとんど発生せず、複雑で大規模なクエリを持つ。
 - サーバのページ・サイズ — Adaptive Server アプリケーションのタイプによって異なります。次のいずれかを選択します。
 - 4K — MIXED および OLTP の場合。
 - 8K — DSS の場合
 - マスタ・デバイスのパス
 - マスタ・デバイスのサイズ
 - マスタ・データベースのサイズ
 - Sybssystemprocs デバイスのパス
 - Sybssystemprocs デバイスのサイズ

- Sybssystemprocs データベースのサイズ
- エラー・ログ
- トランSPORT・タイプ
- ホスト名
- ポート番号

[Adaptive Server の詳細属性の編集] タブに、次の情報を入力します。

- Adaptive Server 設定ファイル
 - sybssystemdb デバイスのパス
 - sybssystemdb デバイスのサイズ
 - sybssystemdb データベースのサイズ
 - 共有メモリ・ファイル・ディレクトリ
 - デフォルトの Backup Server
 - **tempdb** 情報 — Adaptive Server 15.0.3 ESD #1 以降では、指定した個々のデバイス上に tempdb が作成されます。tempdb デバイスとデータベースのデフォルト・サイズは 100MB です。tempdb の次の属性を指定します。
 - tempdb デバイスのパス
 - tempdb デバイスのサイズ
 - tempdb データベースのサイズ
 - Adaptive Server での PCI の有効化
- Adaptive Server で PCI を有効にする場合は、次の項目も設定します。
- sybpcidb デバイスのパス
 - sybpcidb デバイスのサイズ
 - sybpcidb データベースのサイズ
- Adaptive Server の設定値の最適化。サーバの設定後、次の値の入力を要求する **srvbuild** プロンプトが表示されます。
 - Adaptive Server で使用可能な物理メモリ
 - Adaptive Server で使用可能な CPU

デフォルト設定では、システム上で使用可能なリソースをすべて最適に使用できるとはかぎりません。システムのリソース使用量に基づいて、新しい値を入力します。指定した値がサーバに割り付けることができるリソース量より大きい場合、最適化に失敗し、サーバが起動しなくなることがあります。使用可能な物理メモリと使用可能な CPU の現在のデフォルト入力値は、物理メモリおよび CPU の数の 80% です。

注意：バージョン 15.0.3 以降のインストーラでは、インストール後の作業としてではなく、インストール中に基本設定を調整できます。詳細については、『インストール・ガイド』を参照してください。

5. Backup Server を設定するには、[BACKUP SERVER の設定] 画面で、次の内容を選択します。
 - エラー・ログ
 - テープ設定ファイル
 - 言語
 - 文字セット
 - ネットワーク接続の最大数
 - サーバ接続の最大数
 - トランスポート・タイプ
 - ホスト名
 - ポート番号
6. [XP Server type] 画面で、次の内容を選択します。
 - トランスポート・タイプ
 - ホスト名
 - ポート番号
7. Job Scheduler を設定する場合は、次の項目を指定します。
 - sybmgmtdb デバイスのパス
 - sybmgmtdb デバイスのサイズ
 - sybmgmtdb データベースのサイズ
 - トランスポート・タイプ
 - ホスト名
 - ポート番号
8. [Self Management の設定] で次の情報を入力します。
 - Self Management の有効化
 - Self Management ユーザの名前
 - Self Management ユーザのパスワード

Self Management を設定した後で Self Management のユーザ・パスワードを変更する場合は、次のコマンドを実行します。

```
sp_addexternlogin loopback, <Self Management user name>, <Self Management user name>, <new Self Management user password>
```

9. [サーバをビルドしてください] をクリックして、設定を続行します。選択したサーバのビルド中は、**srvbuild** により、設定のステータスが表示されます。サーバの設定が完了したら、[終了] をクリックして **srvbuild** を終了します。

Adaptive Server のデフォルト設定

Adaptive Server のインストールまたは更新時には一部のデフォルトのパラメータや設定が自動的に設定され、いくつかの補助プログラムが用意されています。

この「デフォルト」の Adaptive Server のインストールとテストを行った後、システムの必要性に応じて設定を変更したり、他のオプション機能をインストールしたりする必要があります。

Adaptive Server と Backup Server の設定については、『システム管理ガイド 第 1 巻』の「設定パラメータ」を参照してください。

言語、文字セット、ソート順、およびオプション機能の設定については、『システム管理ガイド 第 1 巻』の「文字セット、ソート順、言語の設定」を参照してください。

デフォルト設定

使用するコンピュータやデータベースのニーズに応じて、デフォルト設定を行うことが必要になる場合もあります。

表 3 : Adaptive Server のパラメータのデフォルト設定

項目	デフォルト値
名前	<i>Servename</i>
トランスポート・タイプ	TCP/IP
ポート番号	5000
エラー・ログのパス	<i>\$SYBASE/\$SYBASE_ASE/install/servename.log</i>
イベント・ロギング	設定なし
言語	us_english
文字セット	HP = Roman8 IBM = ISO 8859-1 Sun = ISO 8859-1 Linux = iso_1

項目	デフォルト値
ソート順	Linux – バイナリ順
ログイン・セキュリティ・モード	標準

表 4：Backup Server と XP Server のデフォルト設定

サーバ	項目	デフォルト値
Backup Server	名前	<i>AdaptiveServername_back</i>
	ネットワーク・サポート	(TCP/IP)
	ソケット番号	5001
	エラー・ログのパス	\$SYBASE/\$SYBASE_ASE/install/AdaptiveServername_back.log
XP Server	名前	\$SYBASE/\$SYBASE_ASE/ADAPTIVESERVERNAME_XP
	ネットワーク・サポート	(TCP/IP)
	ソケット番号	5002
	エラー・ログのパス	N/A

Adaptive Server はネットワーク上の他の Adaptive Server、Open Server アプリケーション、クライアント・ソフトウェアと通信できます。

リモート・プロシージャ・コールを介して、クライアントは、1 つまたは複数のサーバと、サーバは他のサーバと通信できます。

ディレクトリ・サービスには、Adaptive Server や Backup Server など、ネットワーク上のすべてのサーバ製品のネットワーク・ロケーションに関する情報が格納されます。

Sybase のクライアント／サーバ環境では、クライアントがネットワーク上のサーバのロケーションを知っていて、サーバがクライアントの言語または文字セットをサポートしている場合、クライアントは Adaptive Server に接続できます。クライアントが接続を開始する場合、クライアントはディレクトリ・サービスの中から接続先サーバのネットワーク・ロケーションを検索します。

クライアント・プログラムを使用して特定のサーバと接続する場合、クライアント・プログラムはディレクトリ・サービスでサーバ名を検索し、そのサーバに接続します。

サーバには、ネットワーク情報も必要です。サーバは起動時に `interfaces` ファイルの内容を調べて、クライアントの接続要求の受信場所を特定します。さらに、Adaptive Server が他の Adaptive Server に対してリモート・プロシージャ・コールを実行している場合、クライアントとしても機能できます。

Adaptive Server が使用するディレクトリ・サービス・エントリの特定方法

Adaptive Server は、ディレクトリ・サービスを使用して、クライアント接続を受信するアドレスを特定します。

Adaptive Server を起動すると、次の手順が実行されます。

1. コマンド・ラインの **-s** オプションで指定されたサーバ名を探します。コマンド・ラインでサーバ名が指定されていない場合、`DSLISTEN` 環境変数の値をチェックして自身の名前を確認します。`DSLISTEN` 環境変数が設定されていない場合は、サーバ名を `SYBASE` とします。
2. ディレクトリ・サービス内で、上記の手順で見つけた名前と一致するエントリを探します。

3. 検出したディレクトリ・サービス・エントリで指定されているネットワーク情報を使用して、クライアント接続を受信します。

クライアントのディレクトリ・サービスの使用方法

クライアントはディレクトリ・サービスを使用して、サーバへの接続方法を特定します。

クライアントがサーバに接続する際、次の手順が実行されます。

- プログラムを通して、または DSQUERY 環境変数を参照して、サーバの名前を確認する。アプリケーション・ユーザが DSQUERY を設定していない場合、サーバ名のランタイム値はデフォルトの SYBASE 環境変数になる。
- ディレクトリ・サービス内で、サーバの名前と一致するエントリを探す。
- ディレクトリ・サービス・エントリで指定されているネットワーク情報を使用して、サーバに接続する。クライアントが 1 回で接続できない場合、ディレクトリ・サービスで示されている遅延間隔とリトライ回数に従って接続を試行し続ける。一致するエントリが見つからない場合、クライアントの標準エラー・ファイルにエラー・メッセージが書き込まれる。複数のネットワークをサポートしている場合、クライアントはそのサーバの 2 番目のネットワーク・アドレス・エントリの情報を使用して接続を試行する。

クライアント接続の詳細については、使用しているクライアント・プラットフォームの『Open/Client プログラマーズ・ガイド補足』または該当する Open/Client のマニュアルを参照してください。

ディレクトリ・サービスのエントリの作成

Sybase ユーティリティを使用して、ディレクトリ・サービス内のネットワーク情報を編集します。

インストール・プログラム **srvbuild** では、各サーバ・インストール環境のディレクトリ・サービス・エントリが自動的に作成されます。ディレクトリ・サービス内のネットワーク情報を編集するには、次の Sybase ユーティリティを使用します。

- **dsedit** — X-Windows の GUI ユーティリティ
- **dscp** — UNIX コマンド・ライン・ユーティリティ

これらのユーティリティの使用方法の詳細については、『ASE ユーティリティ・ガイド』を参照してください。

サポートされているディレクトリ・ドライバ

Sybase では、interfaces、Lightweight Directory Services (LDAP)、Cell Directory Services (CDS) の各ディレクトリ・ドライバをサポートしています。

- interfaces ドライバ
- LDAP ドライバ
- 分散コンピューティング環境 (DCE) で提供される CDS

使用しているプラットフォームの『Open Client/Server 設定ガイド』を参照してください。

interfaces ファイルの内容

interfaces ファイルには、ネットワーク上のすべてのサーバに関するネットワーク情報が格納されます。これらのサーバには、Adaptive Server、Backup Server、XP Server に加えて、Replication Server などのその他のサーバ・アプリケーションや Open Server アプリケーションが含まれます。

interfaces ファイル内のネットワーク情報には、サーバ名、ホスト・マシンのネットワーク名またはネットワーク・アドレス、サーバがクエリを受信するポート、オブジェクトまたはソケット番号 (ネットワーク・プロトコルによって異なる) が含まれます。

interfaces ファイルの各エントリには、次の 2 種類の行を指定できます。

- master 行 – サーバ・アプリケーションがネットワークを介してクエリを受信するときに使用します。この情報は「リスナ・サービス」と呼ばれます。
- query 行 – クライアント・アプリケーションがネットワークを介してサーバに接続するときに使用します。この情報は「クエリ・サービス」と呼ばれます。

サーバは、クライアントが接続要求に使用するのと同じポートで接続要求を受信するため、サーバの master 行と query 行で指定されているネットワーク情報は同じです。

サーバは他のサーバに対してクライアントとして動作する場合もあるため、サーバの interfaces ファイルには master 行と query 行の両方が必要です。

クライアントの interfaces ファイルには master 行は必要ありません。クライアントの interfaces ファイルは query 行だけで正しく機能します。

サイトに複数のインストール環境がある場合

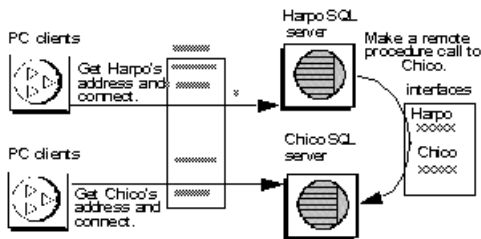
Adaptive Server のインストール環境が複数ある場合、ネットワーク上で動作しているすべてのサーバについての情報を、各サーバの `interfaces` ファイルに格納してください。

すべてのサーバ製品を同じプラットフォームで実行している場合、マスタ `interfaces` ファイルを 1 つ作成し、そのファイルを各マシンにコピーできます。

異機種間環境と同機種間環境

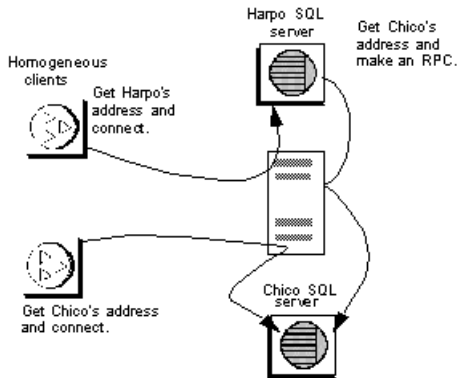
プラットフォームが異なる場合、`interfaces` ファイルのフォーマットと設定を、プラットフォームごとに変えることが必要になる場合があります。たとえば、PC クライアントは自身の `interfaces` ファイル (`sql.ini`) 内のネットワーク情報を使用して、UNIX で実行されている Adaptive Server に接続し、Adaptive Server は自身の `interfaces` ファイルを使用して、リモート・プロシージャ・コールで別のサーバに接続します。

図 2：異機種間環境でのネットワーク接続の確立



クライアントとサーバの両方が UNIX で実行されている場合、同じ `interfaces` ファイルが両方に有効です。たとえば、同機種間環境で実行されているクライアントと Adaptive Server は、1 つの `interfaces` ファイルのコピーを使用して接続を確立できます。2 つの Adaptive Server が同一のオペレーティング・システム上で実行されているため、同じ `interfaces` ファイルまたは同じファイルの完全に同一なコピーを使用できます。

図 3：同機種間環境でのネットワーク接続の確立



interfaces ファイルのフォーマット

interfaces ファイルのフォーマットでは、TLI スタイルのエントリと TCP スタイルのエントリがサポートされます。

以下のルールが適用されます。

- 各 Adaptive Server のエントリは 1 つだけであるが、エントリ内に複数の行がある場合がある。
- *servername* 行に続く各行は、スペースまたはタブ文字で開始する。
- 行の各要素は 1 つのスペースで区切る。
- 各エントリは空白行で区切る。
- 行頭にシャープ記号 (#) を追加し、行末に改行を追加して、interfaces ファイルにコメントを追加できる。

interfaces ファイルのエントリのフォーマットには、TLI と TCP の 2 つがあります。

注意：スレッド・モードでは、TLI インタフェースはサポートされません。

TLI スタイルのエントリでは次のフォーマットを使用します。

```
servername retry_attempts delay_interval<newline>
<tab>service_type api protocol device address filter<newline>
<tab>ha_failover servername<newline>
```

例：

```
harpo_server1
master tli tcp /dev/tcp 0x0002333337f00001
query tli tcp /dev/tcp 0x0002333337f00001
```

第 12 章：ネットワークを介する通信の設定

TCP スタイルのエントリでは次のフォーマットを使用します。

```
servername retry_attempts delay_interval<newline>
<tab>service_type protocol network machine port filter<newline>
<tab>ha_failover servername<newline>
```

例：

```
chico_server1
```

```
master tcp ether chico 5678 ssl
```

```
query tcp ether chico 5678 ssl
```

interfaces ファイルのエントリの要素

interfaces ファイルは、サーバの属性を定義するエントリで構成されます。

コンポーネント	値
<i>servername</i>	Adaptive Server や Backup Server の名前。サーバ名の要件は次のとおり。 <ul style="list-style-type: none">• 名前の長さは 30 文字以内• 最初の文字は英字 (ASCII 文字の a-z、A-Z)• 2 文字目以降は、英字、数字、またはアンダースコア (_)
<i>retry_attempts</i> (オプション)	最初に失敗した後で、クライアントがサーバへの接続を試行する回数。デフォルトは 0。
<i>delay_interval</i> (オプション)	何秒おきに接続を試行するか。デフォルトは 0。
<i>service_type</i>	エントリによって定義されるサービスの種類。次のいずれかを指定する必要がある。 <ul style="list-style-type: none">• master• query
<i>api</i>	ネットワークで使用可能なアプリケーション・プログラミング・インタフェース。サポートされる値は "tli" のみ。
<i>protocol</i>	次のプロトコル・タイプのみを使用可能。 <ul style="list-style-type: none">• TCP/IP、"tcp" と表す。
<i>network</i>	ネットワークの名前。Adaptive Server では現在使用されていない。 srvbuild によって、プレースホルダとして "ether" が入力される。

コンポーネント	値
<i>host</i>	<p>サーバのホスト・マシンのネットワーク名またはネットワーク・アドレス。</p> <ul style="list-style-type: none"> TCP/IP の場合、ホスト名またはインターネット・アドレスを使用する。エントリの最大サイズは 32 バイト。 マシンのホスト名を調べるには、そのマシンにログインし、次のように入力する。 <code>/bin/hostname</code>
<i>machine</i>	<p>サーバのホスト・マシンのネットワーク名またはネットワーク・アドレス。</p> <p>ホスト名またはインターネット・アドレスを使用できる。エントリの最大サイズは 32 バイト。</p> <p>マシンのホスト名を調べるには、そのマシンにログインし、次のように入力する。 <code>/bin/hostname</code></p>
<i>device</i>	<p>ネットワーク・デバイスの終了ポイント。</p> <p>TCP ネットワークの場合、ネットワーク・ソフトウェアを提供しているベンダによって異なる。デバイス名については、ベンダ固有のマニュアルを確認する。TCP プロトコル・スイートのさまざまなプロトコルに対応した複数のストリーム・デバイスをネットワークが提供している場合がある。TCP ストリーム・デバイスを選択する。一般的な TCP ストリーム・デバイスは <code>/dev/tcp</code>。</p>
TLI プロトコル・エントリ用の <i>address</i>	<p>アドレスは次の要素で構成される。</p> <ul style="list-style-type: none"> アドレス・プレフィクス。TLI の場合は "<code>%x</code>"。 ネットワーク・タイプは常に 0002。 ポート番号は、4 桁の 16 進数に変換。1025 ~ 65535 のユニークな番号でなければならない。ネットワーク上の各マシンの <code>/etc/services</code> ファイルをチェックして、使用中のポート番号を確認する。<code>/etc/services</code> ファイル内の "Sybase specific services" という新しいセクションに Adaptive Server のポート番号を入力する。このエントリを作成しなくてもオペレーティング・システムは正常に動作するが、ファイルにポート番号を入力しておけば、他のユーザがそのポート番号を使用するのを防ぐことができる。 8 桁の 16 進数に変換したホスト・マシンの IP ネットワーク・ノード・アドレス。 末尾の 16 桁の 0。オプション。

コンポーネント	値
<i>port</i>	1025～65535 のユニークなポート番号。ネットワーク上の各マシンの /etc/services ファイルをチェックして、使用中のポート番号を確認する。/etc/services ファイル内の "Sybase specific services" という新しいセクションに Adaptive Server のポート番号を入力する。このエントリを作成しなくてもオペレーティング・システムは正常に動作するが、ファイルにポート番号を入力しておけば、他のユーザがそのポート番号を使用するのを防ぐことができる。
<i>ha_failover</i>	ディレクトリ・サービスまたは interfaces ファイルに作成される高可用性のエントリ。
<i>filter</i>	Adaptive Server は、ディレクトリ・サービスの master 行と query 行に追加するフィルタとして SSL (Secure Socket Layers) をサポートする。SSL は機密情報を安全に送信するための標準。

マスタ interfaces ファイルの作成

マスタ interfaces ファイルには、ネットワーク上のすべての Sybase サーバのエントリが格納されます。

マスタ interfaces ファイルは、ネットワークに接続しているすべてのサーバとクライアントで使用できます。マスタ interfaces ファイルのコピーを配布することによって、ネットワーク上のすべての Sybase 製品が互いに対話できるようになります。すべての Adaptive Server のエントリが格納されている interfaces ファイル (マスタ・ファイル) のコピーを配布すれば、ネットワーク上の同機種間環境で簡単に interfaces ファイル間の一貫性を管理できます。

ファイルの 1 つのバージョンに対してすべての変更を行い、次に更新したマスタ・ファイルをすべての適当な UNIX Sybase ディレクトリにコピーできます。

マスタ・ファイルを作成するには、次のいずれかを使用します。

- **dsedit** または **dscp**
- テキスト・エディタ

dsedit または **dscp** を使用したマスタ interfaces ファイルの作成

dsedit ユーティリティまたは **dscp** ユーティリティを使用してマスタ interfaces ファイルを作成し、すべてのサーバに配布できます。

Sybase 製品の使用経験があまりないユーザの場合、テキスト・エディタを使用するよりも **dsedit** または **dscp** を使用の方が簡単です。また、**dsedit** または **dscp** を

使用した場合でも、`interfaces` ファイルのフォーマットの一貫性を確保できません。

1. 最も完全で最新の情報が格納されている `interfaces` ファイルを選択します。
2. 更新された最新の Sybase インストール環境で **dsedit** または **dscp** セッションを開始します。
3. このファイルにリストされていない Adaptive Server または Backup Server のエントリを追加します。
『ASE ユーティリティ・ガイド』を参照してください。

テキスト・エディタを使用したマスタ `interfaces` ファイルの作成

ASCII テキスト・エディタを使用して、連結した `interfaces` ファイルを変更します。

1. 個々の `interfaces` ファイルをすべて連結します。
2. そのファイルのコピーを作成します。
3. ASCII テキスト・エディタを使用して、連結したファイルのコピーを修正します。

`interfaces` ファイルを手動で編集する場合は必ず、それぞれのエントリについて、2 行目以降の各行が `<tab>` 文字で始まるようにしてください。

編集したファイル内で、次の要素がユニークで正しく指定されている必要があります。

- *servername* — `interfaces` ファイル内の各サーバ・エントリはユニークでなければなりません。**srvbuild** セッション中に、サーバの名前を入力するか、デフォルトのサーバ名 **SYBASE** を受け入れています。結合したファイル内で重複する **SYBASE** エントリを探し、それらの名前を変更します。
- ホスト・マシンのネットワーク名またはネットワーク・アドレスと Adaptive Server のポート番号またはオブジェクト番号の組み合わせ。
- ネットワーク上にマシンが 1 つしかないときに元の `interfaces` ファイルが作成された場合、そのファイルのエントリでマシン名 (アドレス) の代わりに `"loghost"` という語が指定されている場合があります。*loghost* が指定されている場合は、マシン名に置き換えます。

複数のネットワーク用の `interfaces` ファイルの設定

プラットフォームによっては、Adaptive Server は複数のネットワークに対応できません。この場合、Adaptive Server は複数のネットワーク・インタフェースを介してクライアントからの接続要求を受信できます。各ネットワーク・インタフェースのエントリを `interfaces` ファイルに追加する必要があります。

複数のネットワーク・ハンドラ用のサーバの設定

複数のネットワーク・リスナを設定します。

1. オペレーティング・システムのホスト・データベースで、各ネットワーク・インタフェースにユニークなホスト名を定義します。
2. テキスト・エディタを使用して `interfaces` ファイルを開き、サーバが受信する追加インタフェースごとに1つずつ、Adaptive Server の "master" 行のコピーを追加します。
3. 各行にユニークなホスト名を指定し、各ネットワーク・インタフェースのネットワーク・ハンドラを設定します。
4. インタフェース内のポート番号は同一である必要はありませんが、同一にすることもできます。これらポート番号の名前と数値の範囲は、プライマリ・ネットワーク・インタフェースと同じ規則に従います。

次の例は、ネットワーク・インタフェースが2つある Adaptive Server の `interfaces` ファイルを示しています。サーバのホスト・マシンは、コーポレート・ネットワークでは `serv_corpnet` と呼ばれ、エンジニアリング・ネットワークでは `serv_engnet` と呼ばれます。

```
# PRODUCTION server with two network listeners
PRODUCTION<tab>3<tab>3<newline>
<tab>master tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_ENGNET 5479
<tab>query tcp ether SERV_CORPNET 4559
```

Adaptive Server を再起動すると、サーバの `DSLISTEN` の値に対応するエントリの各 master 行に対してネットワーク・ハンドラ・プロセスが生成されます。各インタフェースで確立される接続は、ピアとして同等に処理されます。

クライアント接続の設定

Adaptive Server のクライアントが `interfaces` ファイルをスキャンしてサーバ名を検索するときに、クライアントはサーバのエントリ内で最初に検出した "query" エントリを使用します。このため、複数のネットワーク接続を使用するクライアントの設定は、サーバ・ポートの設定よりも複雑になります。

次の2つのオプションがあります。

- すべてのクライアントで `DSQUERY` 名を使用する。違うマシンの `interfaces` ファイルには、違うネットワーク名が格納されている。
- 各クライアントで異なる `DSQUERY` 名を使用する。すべてのマシンの `interfaces` ファイルは同じだが、`interfaces` ファイルには複数の `DSQUERY` 名が格納されている。

ネットワークに依存しない単一の DSQUERY 名の管理

すべてのネットワーク上のすべてのクライアントで使用する DSQUERY 名を 1 つ保持し、各ネットワークの interfaces ファイルを適宜変更することができます。

クライアントで同一の DSQUERY 名を使用することが重要な場合、interfaces ファイル内でクライアントのネットワーク・アドレスに必要な変更を加えることができます。各ネットワークのクライアント・ファイル・サーバに別々の Sybase インストール・ディレクトリおよび別個の interfaces ファイルをインストールすれば、ユーザが正しいネットワーク・アドレスに接続できます。クライアントが使用する DSQUERY 名を変更する代わりに、すべてのネットワークのすべてのクライアントで単一の DSQUERY 名を使用して、各ネットワークの interfaces ファイルを必要に応じて変更します。

この方法は、次のことを前提にしています。

- Sybase インストール環境のクライアントから各ネットワーク上で読み取り可能なファイルを、システム管理者が完全に制御している。
- 異なるネットワーク上にある Sybase インストール環境の間で、(少なくとも) interfaces ファイルは共有もコピーもされていない。

"engineering" ネットワーク上の interfaces ファイルは、次のようになります。

```
PRODUCTION<tab>3<tab>3<newline>
<tab>query tcp ether SERV_ENGNET 5470
<tab>master tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_ENGNET 5479
```

"corporate" ネットワーク上の interfaces ファイルは、次の例のようになります。

```
PRODUCTION<tab>3<tab>3<newline>
<tab>query tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_ENGNET 5479
```

各ファイル内の "query" 行は、使用するネットワークによって異なります。

両方のファイルに完全な "master" エントリがあります。Adaptive Server のみが "master" 行を使用するため、これが可能になります。サーバのホスト・マシンから両方のネットワークを認識できると仮定した場合(両方のホスト名が交換可能な場合)、Adaptive Server の起動時にどちらの interfaces ファイルが使用されていても問題はありません。

異なる DSQUERY 名の使用

ネットワーク・リスナごとに異なる DSQUERY 名を使用できます。

1. 追加するサーバ名を選択します。

元のサーバ名とネットワーク名を連結できます。たとえば、サーバの名前が PRODUCTION の場合、PRODUCTION_network1 と PRODUCTION_network2 という名前を選択できます。

2. 次のいずれかを実行します。

- PC クライアントの場合、**sqledit** を使用して、各ネットワークに 1 つずつ、サーバの `sql.ini` ファイルのエントリを複数作成します。次の例では、PRODUCTION_network1 と PRODUCTION_network2 に 1 つずつエントリを作成します。詳細については、クライアント・プラットフォームの Open Client のマニュアルを参照してください。
- UNIX クライアントの場合、ASCII テキスト・エディタを使用して `interfaces` ファイルを編集します。サーバの `interfaces` ファイルから、各ネットワークのサーバ名の行と "master" 行をクライアントの `interfaces` ファイルにコピーします。各エントリに適切なサーバ名を追加して、"master" を "query" に変更する。

各ネットワーク上のクライアントは、クライアントが配置されているネットワークに対応した DSQUERY の値を使用する必要があります。次の例では、PRODUCTION_network1 または PRODUCTION_network2 を使用できます。

```
# Client entry for PRODUCTION on network1
PRODUCTION_network1<tab>3<tab>3<newline>
<tab>query tcp ether serv_corpnet 4559
# Client entry for PRODUCTION on network2
PRODUCTION_network2<tab>3<tab>3<newline>
<tab>query tcp ether serv_engnet 5479
```

クエリ・ポート・バックアップの設定

クライアントが 2 つのネットワークを介して 1 つのサーバに接続している場合、最初のネットワークが停止したときは 2 番目のネットワークを介してクライアントは接続を確立できます。

1. `interfaces` ファイル内のサーバ・エントリに複数の "master" 行と "query" 行を挿入します。
2. Adaptive Server は両方のポートで接続を受信します。Adaptive Server に接続するためにホスト名とポート番号を検索するクライアントは、接続が確立されるまで各 "query" 行のポートに対して順番に接続を試行します。

次の例は、通常の接続に失敗した場合にのみ使用するバックアップ・ネットワークの設定方法を示しています。プライマリ・ネットワークは「コーポレート・ネットワーク」、バックアップ・ネットワークは「エンジニアリング・ネットワーク」です。

```
# PRODUCTION server with two network listeners
PRODUCTION<tab>3<tab>3<newline>
```

```
<tab>master tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_ENGNET 5479
<tab>query tcp ether SERV_CORPNET 4559
<tab>query tcp ether SERV_ENGNET 5479
```

3. Open Client のマニュアルの説明に従って、適切な複数の "query" エントリを使用して PC クライアントの interfaces ファイルを設定します。同機種間環境のクライアントの interfaces ファイルの場合、Adaptive Server のファイル・エントリ全体をクライアントの interfaces ファイルにコピーできます。
4. コーポレート・ネットワークが無効の場合、またはホスト・マシンのコーポレート・ネットワーク・インタフェースに障害が発生したり、ネットワーク障害によってシャット・ダウンされた場合、セカンダリ・ポートに接続されます。

IPv6 のサポート

Adaptive Server では、IPv6 技術がサポートされます。

IPv6 の用語は次のとおりです。

- リンクローカル・アドレス — 1つのリンク経由だけで使用できる IPv6 アドレス。
- サイトローカル・アドレス — 1つのサイト内だけで使用できる IPv6 アドレス。
- グローバル・アドレス — インターネット全体にわたって使用できる IPv6 アドレス。

Interfaces ファイルでも IPv6 をサポートしています。次に interfaces ファイルのエントリの例を示します。

```
RHAM0_10778_XP query tcp sun-ether fd77:55d:
459d9:169:250:56ff:feb3:4246 19560 master tcp sun-ether
fd77:55d:59d9:169:250:56ff:feb3:4246 19560
```

IPv6 アプリケーションのタイプは次のとおりです。

- IPv6-unaware (非認識) — IPv6 アドレスを処理できないアプリケーション。
- IPv6-aware (認識) — IPv4 アドレスを持たないノードと通信できるアプリケーション。API が実際のアドレスの内容とフォーマットを隠す場合など、これはアプリケーションに対して透過的になることがあります。
- IPv6-enabled (有効化) — IPv6-aware (認識) の機能があるだけでなく、IPv6 の一部の機能も利用できるアプリケーション。
- IPv6-required (要求) — IPv6 の機能を必要とし、IPv4 経由では動作しないアプリケーション。

IPv6 インフラストラクチャ

デュアル・スタック・インフラストラクチャは、IPv4 と IPv6 の両方を実装します。Adaptive Server を IPv6 認識サーバとして使用する場合は、このインフラストラクチャの実装をおすすめします。

Sybase アプリケーションは IPv6 に対応します。

表 5 : IPv6 のサポート

プラットフォーム	Adaptive Server の IPv6 認識	Open Client/Server の IPv6 認識
Sun Solaris 8 32 ビット版および 64 ビット版	12.5.3a および 15.0	15.7
HP-UX 11i(v1) 32 ビット版および 64 ビット版	12.5.3a および 15.0	15.7
Microsoft Server 2003	12.5.3a および 15.0	15.7
Linux RHEL 3.0	15.0	15.7

XP Server、Backup Server、Replication Server、Open Switch など、Open Client/Server ベースの多くの Sybase 製品は、ネットワーク・ソケット処理に対して IPv6 を認識するレイヤ構成の Open Client トランスポート制御層 (DB-Library ベース) により、自動的に IPv6 認識になっています。Open Client 製品は、IPv6 認識ではありません。

Adaptive Server の場合、Adaptive Server 内の一部のサード・パーティ・コンポーネントがまだ IPv6 認識ではないため、IPv6 認識の定義は複雑です。サポートされているプラットフォームとバージョンに関して、IPv6 認識である Adaptive Server Enterprise の機能メカニズムを次に示します。

- 接続ハンドラ
- RPC メカニズム
- Job Scheduler Task/Agent セッション接続
- ネットワーク・ホスト API
- sybsendmsg の UDP メッセージのサポート
- コンポーネント統合サービス接続
- ホスト/名前解決
- XML URL 接続ハンドラ
- クライアント・アドレス・データの監査

Adaptive Server の次の機能メカニズムは、現時点で IPv6 をサポートしていませんが、今後のバージョンでサポートする予定です。

- Java サポート
- ライセンス管理サーバ
- LDAP ドライバ

デフォルトでは、Adaptive Server は IPv6 非認識です。

IPv6 認識の処理を行う場合は、Adaptive Server を起動する前に、インフラストラクチャが正しく設定されていることを確認します。オペレーティング・システムが正しく設定されていれば、DCL エントリを追加することにより、IPv6 接続ハンドラを設定し、有効にできます。通常、単一の Adaptive Server 設定では、DCL 内に最大 32 の接続ハンドラを割り当てることができます。

たとえば、サイト・ローカル設定で 2 つのドメインを名前で管理している場合は、次のように表示されます。

```
sybase.com - being responsible for all IPv4 networking applications
v6.sybase.com - being responsible for all IPv6 networking applications
```

ポート 17100 のホスト "revival" にある "SYBASE" という DCL エントリは、次のようになります。

```
SYBASE
master tcp ether revival.sybase.com 17100
query tcp ether revival.sybase.com 17100
master tcp ether revival.v6.sybase.com 17100
query tcp ether revival.v6.sybase.com 17100
```

この例では、Adaptive Server が IPv6 認識で起動するときに、2 つの接続ハンドラを作成します。1 つは受信する IPv4 クライアント接続要求をポート 17100 で待機し、もう 1 つは受信する IPv6 クライアント接続要求をポート 17100 で待機します。

注意： Adaptive Server の起動時に、IPv4 と IPv6 のクライアント・アドレス接続要求およびホスト／名前検索を取得してログに記録するように、トレース・フラグ 7815 を設定します。

サーバが起動しない原因となる一般的な状況のトラブルシューティングについて説明します。

サーバが起動しない場合のトラブルシューティング

サーバが起動しない場合、interfaces ファイルで指定されたポート番号が使用可能であることを確認します。

サーバが起動せず、次のメッセージが表示される場合は、interfaces ファイルで指定されたポート番号が使用中である可能性があります。

```
00:00000:00002:2003/09/22 12:37:23.63 kernel network name
SERV_CORPNET, type ether, port 4559, filter NONE
00:00000:00002:2003/09/22 12:37:23.65 kernel ninit: bind, Address
already in use
00:00000:00002:2003/09/22 12:37:23.68 server Error: 1602, Severity:
18, State: 2
00:00000:00002:2003/09/22 12:37:23.68 server Unable to initialize
network 0
00:00000:00002:2003/09/22 12:37:23.68 kernel ninit: All master
network
listeners have failed. Shutting down.
00:00000:00002:2003/09/22 12:37:23.68 kernel ueshutdown: exiting
00:00000:00016:2003/09/22 16:11:35.46 server SQL Server shutdown by
request.
```

1. ポートの割り当てを調べます。

- a) interfaces ファイルを調べて、サーバに割り当てたポート番号を確認します。
- b) 次のように入力して、別のプロセスが同じポート番号を使用していないかどうかを確認します。

```
netstat -a
```

netstat の出力にそのポート番号がローカル・アドレスとして表示された場合、そのポートをサーバに使用することはできません。別のプロセスがすでにこのポートを使用しています。

- c) サーバ・ポートが使用中であることを確認するには、サーバを手動で起動します。割り当てられたポート番号がすでに使用中の場合、サーバは起動しません。

使用しているプラットフォームのインストール・ガイドおよび『ASE ユーティリティ・ガイド』を参照してください。

2. 終了したはずのサーバ・プロセスがポート番号を開放しない場合は、次のいずれかを実行します。

- オペレーティング・システムの **kill** コマンドを使用して、プロセスを終了する。
 - `interfaces` ファイルを修正して、サーバに別のポート番号を使用する。
3. サーバを手動で起動して、ポート番号が使用可能であることを確認します。
使用しているプラットフォームのインストール・ガイドおよび『ユーティリティ・ガイド』を参照してください。

ESP 実行時のエラーのトラブルシューティング

ESP の実行時にエラーが表示される場合、同じポート番号を使用しているプロセスがないかどうかを確認します。

前提条件

netstat コマンドを使用して、XP Server に指定されたポート番号が使用中でないかどうかを確認します。

手順

ESP (拡張ストアド・プロシージャ) を実行しようとしたときに、次のようなエラーが表示される場合があります。

```
00:00000:00008:1997/09/10 12:52:53.03 kernel XP Server failed to
start. Try bringing up XP Server manually. Check SQL Server
documentation for more information on how to bring XP Server up.
```

1. 同じポート番号を使用しているプロセスがない場合、次の手順に従います。
 - a) Adaptive Server を再起動します。
 - b) 先ほど試みた ESP を実行します。
XP Server は、自動で起動します。
2. 同じポート番号を使用しているプロセスが見つかった場合、次の手順に従います。
 - a) XP Server で新しいポート番号を使用するように `interfaces` ファイルを変更します。
 - b) XP Server に割り当てられたポート番号を使用するプロセスを停止する。
 - c) Adaptive Server を再起動し、前に試行した ESP を実行します。XP Server は、自動で起動します。

ディレクトリ・サービスとしての Lightweight Directory Access Protocol (LDAP) の使用

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。

Adaptive Server では、ディレクトリ・サービスを使用してインターネット上でのクライアントと RPC との接続を確立しています。接続を確立するには、LDAP ディレクトリ・サービスを使用します。LDAP は Adaptive Server の機能ですが、ライセンスが必要です。

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、データ・フォーマット情報も含めた読み込み、書き込み、クエリ、サーバ応答のクライアント要求などのオペレータです。

LDAP サーバは、次の情報を格納したり取り出したりします。

- 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

LDAP サーバでは次のアクセス制限を設定できます。

- 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、UNIX プラットフォームのデフォルトのユーザ名とパスワードを使用します。
 - `$SYBASE/$SYBASE_OCS/config/libtcl.cfg` (32 ビット・プラットフォーム)
 - `$SYBASE/$SYBASE_OCS/config/libtcl64.cfg` (64 ビット・プラットフォーム)

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： ユーザ認証のために LDAP サーバに渡される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。

LDAP サーバが libtcl.cfg or libtcl64.cfg ファイル (まとめて libtcl*.cfg ファイル) で指定されている場合は、LDAP サーバからのみサーバ情報にアクセスできます。Adaptive Server は interfaces ファイルを無視します。

1 台のサーバで複数のディレクトリ・サービスがサポートされている場合は、検索順序が libtcl*.cfg で指定されます。dataserver コマンド・ライン・オプションを使用して検索順序を指定することはできません。

『セキュリティ管理ガイド』を参照してください。

参照：

- 第 3 章、「クライアント/サーバ通信」 (13 ページ)

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

LDAP サーバで使用するディレクトリ・サービスは LDAP ドライバによって実装されます。

LDAP ディレクトリでは、次の情報が提供されます。

- 従来の Sybase interfaces ファイルのネットワークベース版
- 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

interfaces 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

LDAP ディレクトリ・サービスは、Sybase interfaces ファイルより多くの属性をサポートしています。この属性には、サーバのバージョンやサーバのステータスなどを含めることができます。

注意：LDAP だけが、リエントラント・ライブラリでサポートされています。**isql**ではなく、**isql_r**を使用してください。

表 6：Sybase LDAP ディレクトリの定義

属性名	値のタイプ	説明
ditbase	interfaces ファイルまたは libtcl.cfg	オブジェクト・ツリーの DIT ベース。 libtcl.cfg ファイルが指定されている場合は、interfaces ファイルが無視される。 ct_con_prop() を使用して、libtcl.cfg ファイルで指定された接続を無効にできる。
dn	文字列	識別名。オブジェクトを識別するユニークな名前にする必要がある。
sybaseVersion	整数	サーバのバージョン番号。
sybaseServername	文字列	サーバの名前。
sybaseService	文字列	サービスの種類：Sybase Adaptive Server
sybaseStatus	整数	ステータス：1=アクティブ、2=停止、3=失敗、4=不明。
sybaseAddress	文字列	各サーバのアドレス。次の項目を含む。 <ul style="list-style-type: none"> プロトコル：TCP、NAMEPIPE、SPX、DECNET (エントリでは大文字と小文字が区別される) アドレス：そのプロトコル・タイプに有効な任意のアドレス dscp は、この属性をトランスポート・タイプとトランスポート・アドレスに分割する。
sybaseSecurity (オプション)	文字列	セキュリティ OID (オブジェクト ID)。
sybaseRetryCount	整数	この属性は、CS_RETRY_COUNT にマッピングされる。CS_RETRY_COUNT は、サーバ名に関連付けられたネットワーク・アドレスのシーケンスを ct_connect がリトライする回数を指定する。

第 14 章：ディレクトリ・サービスとしての Lightweight Directory Access Protocol (LDAP) の使用

属性名	値のタイプ	説明
sybaseRetryDelay	整数	この属性は、CS_LOOP_DELAY にマッピングされる。CS_LOOP_DELAY はアドレスのシーケンス全体をリトライする前に ct_connect が待機する遅延秒数を指定する。
sybaseHAservname (オプション)	文字列	フェールオーバー保護用のセカンダリ・サーバ。

従来の interfaces ファイルは、TCP 接続のフェールオーバー・マシンで次のように表示されます。

```
master tcp ether huey 5000
query tcp ether huey 5000
hafailover secondary
```

次の例は、TCP およびフェールオーバー・マシンの LDAP エントリを示しています。

```
dn: sybaseServername=foobar, dc=sybase, dc=com
objectClass: sybaseServer
sybaseVersion: 1500
sybaseServername: foobar
sybaseService: ASE
sybaseStatus: 4
sybaseAddress: TCP#1#foobar 5000
sybaseRetryCount: 12
sybaseRetryDelay: 30
sybaseHAservname: secondary
```

LDAP ディレクトリ・サービスへのすべてのエントリは、エンティティと呼ばれます。各エンティティは DN (識別名) を持ち、それぞれの DN に基づいて階層ツリー構造内に格納されます。このツリーは、「ディレクトリ情報ツリー (DIT)」と呼ばれます。クライアント・アプリケーションは、DIT ベースを使用してエンティティの格納場所を指定します。

上記の例のエントリは、“foobar” という名前の Adaptive Server がポート番号 5000 の TCP 接続で受信していることを示します。このエントリは、12 (回) のリトライ回数と 30 (秒) のリトライ遅延時間も指定しています。サーバが応答するアドレスをクライアントが検出すると、クライアントとサーバ間でログイン・ダイアログが開始されます。

UNIX では、Sybase の LDAP ディレクトリ・スキーマのリストは、\$SYBASE/\$SYBASE_OCS/config ディレクトリの sybase.schema というファイルにあります。同じディレクトリ内に、sybase-schema.conf というファイルもあります。このファイルは、同じスキーマを格納していますが、Netscape 固有の構文を使用しています。

第 14 章：ディレクトリ・サービスとしての Lightweight Directory Access Protocol (LDAP) の使用

LDAP では各属性の複数のエントリをサポートしているので、各アドレス属性は単一サーバのアドレス (プロトコル、アクセス・タイプ、アドレスを含む) を持つ必要があります。

次の例は、異なる接続プロトコルの 2 つのアドレスで受信している Windows サーバの LDAP エントリを示します。

```
sybaseAddress = TCP#1#TOEJAM 4444
sybaseAddress = NAMEPIPE#1#¥pipe¥sybase¥query
```

注意： アドレス・フィールドの各エントリは、# 文字で区切ります。

これらのエントリは、**dsedit** を使用して編集できます。

すべての Sybase 製品でプラットフォームに関係なく互換性を保つため、プロトコルおよびアドレス属性フィールドはプラットフォームと製品に依存しないフォーマットにしてください。

libtcl*.cfg ファイル

libtcl*.cfg ファイルを使用して、LDAP サーバへの接続の認証に必要な LDAP サーバ名、ポート番号、DIT ベース、ユーザ名およびパスワードを指定します。

libtcl*.cfg ファイルの目的は、Open Client/Open Server および Open Client/Open Server ベースのアプリケーションに、ドライバ、ディレクトリ、セキュリティ・サービスなどの設定情報を提供することです。**dsedit** や **srvbuild** などの 32 ビット・ユーティリティは libtcl.cfg ファイルで設定情報を検索し、64 ビット・アプリケーションは libtcl64.cfg ファイルを設定情報に使用します。

32 ビットと 64 ビットのアプリケーション間の互換性を確保するには、libtcl.cfg ファイルと libtcl64.cfg ファイルの両方を編集します。

デフォルトの libtcl.cfg ファイルは、\$SYBASE/\$SYBASE_OCS/config にあります。

libtcl.cfg ファイルで LDAP が指定されている場合、interfaces ファイルは使用されません。

注意： 起動時に **-i** オプションを使用する Open Client/Open Server アプリケーションは、libtcl.cfg ファイルを無視して interfaces ファイルを使用します。

最も単純な形式を使用した場合、libtcl.cfg ファイルのフォーマットは次のようになります。

```
[DIRECTORY]
```

第 14 章：ディレクトリ・サービスとしての Lightweight Directory Access Protocol (LDAP) の使用

```
ldap=libsybdldap.dll ldapurl
```

ここで、*ldapurl* は次のように定義されています。

```
ldap://host:port/ditbase
```

次の LDAP エントリは、上記と同じ属性を使用していますが、匿名接続であり、LDAP サーバで読み込み専用アクセスが許可される場合にのみ機能します。

```
ldap=libsybdldap.dll ldap://ldaphost/d=sybase,dc=com
```

接続時のパスワード認証を有効にするには、LDAP URL の拡張機能として *libtcl.cfg* ファイルでユーザ名とパスワードを指定します。

LDAP ディレクトリ・サービスの有効化

LDAP ディレクトリ・サービスを有効にします。

1. “Security and directory services” ライセンス・パッケージである ASE_SECDIR をインストールします。
2. プラットフォームの UNIX ロード・ライブラリ・パス環境変数に、LDAP ライブラリのロケーションを追加します。
3. ディレクトリ・サービスを使用するように *libtcl.cfg* ファイルを設定します。

標準的な ASCII テキスト・エディタを使用して、次のように修正します。

- *libtcl.cfg* ファイルの [DIRECTORY] エントリで、LDAP URL 行の行頭からセミコロン (;) のコメント・マーカを削除します。
- [DIRECTORY] エントリに LDAP URL を追加します。

警告！ LDAP URL は、1 行で記述する必要があります。

```
libtcl.cfg
ldap=libsybdldap.so ldap://host:port/ditbase??scope??
bindname=username
password

libtcl64.cfg
ldap=libsybdldap64.so ldap://host:port/ditbase??scope??
bindname=username
password
```

例：

```
[DIRECTORY]
ldap=libsybdldap.so ldap:///huey:11389/dc=sybase,dc=com??one??
bindname=cn=Manager,dc=sybase,dc=com secret
```

第 14 章：ディレクトリ・サービスとしての Lightweight Directory Access Protocol (LDAP) の使用

“one”は、エントリを取得する検索範囲が DIT ベースの 1 レベル下であることを示します。

4. 必要なサード・パーティ・ライブラリが、適切な環境変数で指定されていることを確認します。Netscape LDAP SDK ライブラリは、\$SYBASE/\$SYBASE_OCS/lib3p または lib3p64 にあります。UNIX ロード・ライブラリ・パス環境変数で、このディレクトリを指定する必要があります。
5. **dscp** または **dsedit** を使用して、LDAP サーバにサーバ・エントリを追加します。

LDAP URL 変数のキーワード

libtcl.cfg ファイルの DIRECTORY セクションを設定します。

キーワード	説明	デフォルト
<i>host</i> (必須)	LDAP サーバを実行しているマシンのホスト名または IP アドレス	なし
<i>port</i>	LDAP サーバが受信しているポート番号	389
<i>ditbase</i> (必須)	デフォルトの DIT ベース	なし
<i>username</i>	認証するユーザの DN (識別名)	NULL (匿名認証)
<i>password</i>	認証されるユーザのパスワード	NULL (匿名認証)

ディレクトリ・サービスへのサーバの追加

dsedit を使用して新しいサーバ・エントリを追加します。

各サーバ・エントリは、一連の属性で構成されます。サーバ・エントリを追加または修正するときは、サーバ属性についての情報を要求されます。属性のいくつかはデフォルトで提供されますが、その他はユーザが入力する必要があります。提供されたデフォルト値は、角カッコ “[]” で囲まれて表示されます。入力可能な値については、「Sybase LDAP ディレクトリの定義」を参照してください。

srvbuild を使用してエントリを追加できますが、エントリの変更や削除を行うことはできません。

警告！ ほとんどの LDAP サーバには、ディレクトリ・エントリを追加するための **ldapadd** ユーティリティがあります。代わりに、**dsedit** を使用することをおすすめします。このユーティリティには、汎用ツールにはないセマンティック・チェックが組み込まれているためです。

第 14 章：ディレクトリ・サービスとしての Lightweight Directory Access Protocol (LDAP) の使用

1. `SYBASE.csh` または `SYBASE.sh` を情報源として読み込み、環境変数を設定します。
2. `cd` コマンドを使用して `$SYBASE/$SYBASE_OCS/bin` に移動します。
3. `dsedit` を実行します。
4. サーバの一覧から [LDAP] を選択して、[OK] をクリックします。
5. [新しいサーバ・エントリを追加する] をクリックします。
6. 次のように入力します。
 - サーバ名 – 必須です。
 - セキュリティ・メカニズム – オプションです。高可用性フェールオーバー・サーバが存在する場合はその名前です。
7. [新しいネットワーク・トランスポートを追加する] をクリックして、次の操作を実行します。
 - トランスポート・タイプを選択します。
 - ホスト名を入力します。
 - ポート番号を入力します。
8. [OK] を 2 回クリックして、`dsedit` を終了します。
サーバ・エントリを表示するには、Netscape で次の URL を入力します。

`http://host:port/ditbase??one.`

例：

`ldap://huey:11389/dc=sybase,dc=com??one`

注意： Microsoft Internet Explorer では、LDAP URL は認識されません。

`dscp` の詳細については、『Open Client/Server 設定ガイド』を参照してください。

参照：

- `interfaces` ファイルから LDAP へのマイグレート (86 ページ)
- 第 7 章、「サーバの起動と停止」 (25 ページ)

複数のディレクトリ・サービス

高可用性フェールオーバー保護用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。LDAP サービスは、実際のサーバであるか、他の LDAP サービスへのゲートウェイであるかにかかわらず、どのようなタイプであっても LDAP サーバと呼ばれます。

例：

```
[DIRECTORY]
```

```
ldap=libsybdldap.so ldap://test:389/dc=sybase,dc=com  
ldap=libsybdldap.so ldap://huey:11389/dc=sybase,dc=com
```

この例では、*test:389*への接続が失敗すると、指定された DIT ベースを持つ DCE ドライバに接続がフェールオーバーされます。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。

注意： 詳細については、『Open Client Client-Library/C プログラマーズ・ガイド』および『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。

パスワードの暗号化

libtcl.cfg ファイルのエントリは、人間が判読できるフォーマットになっています。Sybase では、基本的なパスワード暗号化のための **pwdcrypt** ユーティリティを提供しています。

pwdcrypt は、キーボードから入力したパスワードの代わりに使用できる暗号値を生成する単純なアルゴリズムです。**pwdcrypt** は、`$SYBASE/$SYBASE_OCS/bin` にあります。

1. `$SYBASE/$SYBASE_OCS` ディレクトリから、次のように入力します。

```
bin/pwdcrypt
```

2. 要求されたら、パスワードを 2 度入力します。

pwdcrypt は暗号化されたパスワードを生成します。次に例を示します。

```
0x01312a775ab9d5c71f99f05f7712d2cded2i8d0ae1ce78868d0e8669313d1bc  
4c706
```

3. 標準的なテキスト・エディタを使用して、暗号化されたパスワードをコピーして libtcl.cfg ファイルに貼り付けます。暗号化の前には、ファイル・エントリは次のように表示されます。

```
ldap=libsybdldap.so  
ldap://ldaphost/dc=sybase,dc=com??one??  
bindname=uid=Manager,dc=sybase,  
dc=com?password
```

4. パスワードを、暗号化した文字列に置き換えます。

```
ldap=libsybdldap.so  
ldap://ldaphost/dc=sybase,dc=com??one??  
bindname=uid=Manager,dc=sybase,dc=com?  
0x01312a775ab9d5c71f99f05f7712d2cded2i8d0ae1ce78868d0e8669313d1bc  
4c706
```

警告！ パスワードが暗号化された場合も、ファイル・システム・セキュリティを使用してパスワードを保護してください。

パフォーマンス

LDAP サーバを使用した場合のパフォーマンスは、`interfaces` ファイルを使用した場合よりも遅くなることがあります。これは、LDAP サーバがネットワークに接続してデータを取り出すのに時間がかかるためです。

この接続は、Adaptive Server の起動時に行われるため、パフォーマンスが低下した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。高いシステム・ロードで接続が多い場合、特に短時間に繰り返し接続するような場合は、LDAP サーバを使用したときと従来の `interfaces` ファイルを使用したときとは、全体のパフォーマンスに違いがでることがあります。

interfaces ファイルから LDAP へのマイグレート

Adaptive Server をアップグレードしたら、LDAP サービスを使用するようにサーバを設定します。

`interfaces` ファイルを使用する既存のサーバを、LDAP を使用するサーバに直接アップグレードする方法はありません。Adaptive Server のバージョンをアップグレードする場合は、『インストール・ガイド』を参照してください。

1. サーバを停止します。
2. `$SYBASE/$SYBASE_OCS/config/libtcl.cfg` ファイルまたは `libtcl64.cfg` ファイルを編集して、ディレクトリ・サービスを追加します。
3. `dsedit` を使用して、サーバ・エントリをディレクトリ・サービスに追加します。
4. サーバを再起動します。

参照：

- ディレクトリ・サービスへのサーバの追加 (83 ページ)
- 第 7 章、「サーバの起動と停止」 (25 ページ)

Adaptive Server では、世界各国の顧客や異機種間環境で使用する顧客向けに、ローカライゼーションをサポートしています。

ローカライゼーションとは、アプリケーションをある特定の言語または地域の稼働条件に適応させることです。これには、その国の言語に翻訳されたシステム・メッセージや、その国で使用している日付、時刻、通貨の正しいフォーマットの提供も含まれます。

デフォルトでは、Adaptive Server と Backup Server の設定には英語のロケール設定が使用されます。この設定には次のファイルが含まれます。

- 西欧の文字セットを定義したファイル
- 西欧の文字セットで使用する、ソート順を定義したファイル
- アメリカ英語のシステム・メッセージ・ファイル

インストール中または再設定中に、別の言語、文字セット、ソート順を指定できます。

Adaptive Server では、世界各国の顧客や異機種間環境で使用する顧客向けに、ローカライゼーションをサポートしています。サポートする内容は次のとおりです。

- データ処理のサポート — Adaptive Server には、さまざまな言語で使用する文字を処理するための、文字セットとソート順の定義ファイルが付属しています。

Sybase は、次の地域の主要言語をサポートします。

- 西欧
- 東欧
- 中東
- ラテン・アメリカ
- アジア
- システム・メッセージの翻訳 — Adaptive Server には次の言語のモジュールがあります。
 - ブラジル系ポルトガル語
 - 中国語 (簡体字)
 - フランス語
 - ドイツ語
 - 日本語
 - 韓国語
 - ポーランド語

- スペイン語
- タイ語
- マニュアルの翻訳 — 次の言語に翻訳されたマニュアルが利用できます。
 - 中国語 (簡体字)
 - フランス語
 - ドイツ語
 - 日本語
 - 韓国語
 - ポーランド語
 - スペイン語

言語モジュール

Adaptive Server は、ローカライズされたソフトウェア・メッセージを別々の言語モジュールに格納します。

言語モジュールをインストールすると、適切な場所に格納された新しい言語をサポートするメッセージ、文字セット、ソート順を定義したファイルを、インストール・プログラムがロードします。

Adaptive Server と Backup Server をインストールすると、デフォルトでは英語のシステム・メッセージがインストールされます。

デフォルトの文字セット

デフォルトの文字セットとは、データがコード化され、Adaptive Server データベースに格納されている文字セットです。

デフォルトの言語と文字セットの変更

警告！ 新しい Adaptive Server の文字セットとソート順の変更をすべて行ってから、ユーザ・データベースの作成や Sybase が提供するデータベースの変更を行ってください。データまたはデータ構造が Adaptive Server に追加されてから文字セットやソート順を変更すると、時間のかかる追加手順が必要になる場合があります。データを追加した後で文字セットやソート順を変更する場合は、『システム管理ガイド 第 1 巻』を参照してください。

srvbuild を使用して新しい Adaptive Server を設定した後、Adaptive Server を `us_english` 以外の言語にローカライズするかどうか、デフォルト以外の文字セットやソート順を使用するかどうかを尋ねられます。

デフォルトの Adaptive Server は次の設定を使用します。

- us_english 言語
- iso_1 文字セット (HP-UX プラットフォームでは Roman8 を使用)
- バイナリ・ソート順

有効な言語オプションは、配布メディアからシステムにアンロードされた言語モジュールによって異なります。

デフォルトでは、すべての文字セットが配布メディアからコピーされます。

次の操作を実行できます。

- [No] をクリックして、デフォルトを受け入れます。
- [Yes] をクリックして、デフォルトを変更します。
sqlloc メニューが表示されます。**sqlloc** は、デフォルトの言語、文字セット、ソート順を変更する場合に使用する GUI ユーティリティです。

デフォルトでは、IBM および (SUN) Solaris システムに Adaptive Server と Backup Server をインストールすると、西欧言語をサポートする ISO 8859-1 の文字セット・ファイルがインストールされます。

デフォルトでは、HP システムに Adaptive Server と Backup Server をインストールすると、西欧言語をサポートする Roman8 の文字セット・ファイルがインストールされます。

サーバのデフォルトの文字セットの変更

Adaptive Server のデフォルトには任意の文字セットを選択できます。これにはプラットフォームのデフォルトと異なる文字セットも含まれます。新しいデフォルトの文字セットを選択する際は、次のガイドラインに留意します。

- 変換エラーやオーバヘッドを防止するため、クライアントが使用している文字セットに基づいてデフォルトの文字セットを決定する。
 たとえば、ほとんどのクライアントが ISO 8859-1 を使用している場合、ISO 8859-1 を指定することでデータ変換の量を最小化できます。
- 使用しているサーバが異機種言語環境で稼働している場合は、必要とされるすべての文字セットで動作する文字セットを選択します。多くの場合、これは Unicode (UTF-8) です。

警告！ 新しい Adaptive Server のデフォルトの文字セットとソート順の変更をすべて行ってから、ユーザ・データベースの作成や Sybase が提供するデータベースの変更を行ってください。データやデータ構造が Adaptive Server に追加されてから文字セットやソート順を変更すると、不正な動作が発生する可能性があります。データを追加した後で文字セットやソート順を変更する場合は、『システム管理ガイド 第 1 巻』を参照してください。

サポートされる文字セット

Adaptive Server は多数の言語、スクリプトおよび文字セットをサポートしています。

アラビア語の文字セット

Adaptive Server はアラビア語の文字セットをサポートしています。

- X – Unilib[®] 変換を必要とします。
- X なし – Unilib 変換と組み込み変換のどちらかを使用します。

文字セット	Unilib の必要性	説明
cp864	X	PC アラビア語
cp1256	X	Microsoft Windows アラビア語
iso88596	X	ISO 8859-6 ラテン語／アラビア語

バルト語の文字セット

Adaptive Server はバルト語の文字セットをサポートしています。

- X – Unilib[®] 変換を必要とします。
- X なし – Unilib 変換と組み込み変換のどちらかを使用します。

文字セット	Unilib の必要性	説明
cp1257	X	Microsoft Windows バルト語

中国語 (簡体字) の文字セット

Adaptive Server は中国語 (簡体字) の文字セットをサポートしています。

- X – Unilib[®] 変換を必要とします。
- X なし – Unilib 変換と組み込み変換のどちらかを使用します。

文字セット	Unilib の必要性	説明
eucgb	X	EUC GB コード化 = 中国語 (簡体字) の文字セット
cp936	X	Microsoft 中国語 (簡体字) の文字セット

文字セット	Unilib の必要性	説明
gb18030	X	RC 18030 標準

中国語 (繁体字) の文字セット

Adaptive Server は中国語 (繁体字) の文字セットをサポートしています。

- X – Unilib[®] 変換を必要とします。
- X なし – Unilib 変換と組み込み変換のどちらかを使用します。

文字セット	Unilib の必要性	説明
cp950	X	PC (Microsoft) 中国語 (繁体字)
euccns	X	EUC CNS コード化 = 中国語 (繁体字) の文字セット
big5	X	Big 5 中国語 (繁体字)
big5hk	X	Big 5 HKSCS の拡張付き

キリル語の文字セット

Adaptive Server はキリル語の文字セットをサポートしています。

- X – Unilib[®] 変換を必要とします。
- X なし – Unilib 変換と組み込み変換のどちらかを使用します。

文字セット	Unilib の必要性	説明
cp855		IBM PC キリル語
cp866		PC ロシア語
cp1251		Microsoft Windows 3.1 キリル語
iso88595		ISO 8859-5 ラテン語／キリル語
koi8		KOI-8 キリル語
mac_cyr		Macintosh キリル語
kz1048		カザフスタン・キリル語

東欧言語の文字セット

Adaptive Server は東欧言語の文字セットをサポートしています。

- X – Unilib[®] 変換を必要とします。

- X なし — Unilib 変換と組み込み変換のどちらかを使用します。

文字セット	Unilib の必要性	説明
cp852		PC 東欧言語
cp1250		Microsoft Windows 3.1 東欧言語
iso88592		ISO 8859-2 Latin-2
mac_ee		Macintosh 東欧言語

ギリシャ語の文字セット

Adaptive Server はギリシャ語の文字セットをサポートしています。

- X — Unilib[®] 変換を必要とします。
- X なし — Unilib 変換と組み込み変換のどちらかを使用します。

文字セット	Unilib の必要性	説明
cp869		IBM PC ギリシャ語
cp1253		MS Windows ギリシャ語
greek8		HP GREEK8
iso88597		ISO 8859-7 ラテン語／ギリシャ語
macgrk2		Macintosh ギリシャ語

ヘブライ語の文字セット

Adaptive Server はヘブライ語の文字セットをサポートしています。

- X — Unilib[®] 変換を必要とします。
- X なし — Unilib 変換と組み込み変換のどちらかを使用します。

文字セット	Unilib の必要性	説明
cp1255	X	Microsoft Windows ヘブライ語
iso88598	X	ISO 8859-8 ヘブライ語

日本語の文字セット

Adaptive Server は日本語の文字セットをサポートしています。

- X — Unilib[®] 変換を必要とします。

- X なし — Unilib 変換と組み込み変換のどちらかを使用します。

文字セット	Unilib の必要性	説明
cp932	X	IBM J-DBCS:CP897 + CP301 (シフト JIS)
deckanji		Digital UNIX JIS コード化
eucjis		EUC JIS コード化
sjis		シフト JIS (拡張なし)

韓国語の文字セット

Adaptive Server は韓国語の文字セットをサポートしています。

- X — Unilib[®] 変換を必要とします。
- X なし — Unilib 変換と組み込み変換のどちらかを使用します。

文字セット	Unilib の必要性	説明
eucksc	X	EUC KSC 韓国語コード化 = CP949

タイ語の文字セット

Adaptive Server はタイ語の文字セットをサポートしています。

- X — Unilib[®] 変換を必要とします。
- X なし — Unilib 変換と組み込み変換のどちらかを使用します。

文字セット	Unilib の必要性	説明
tis620	X	TIS-620 標準タイ語
cp874	X	Microsoft Windows タイ語

トルコ語の文字セット

Adaptive Server はトルコ語の文字セットをサポートしています。

- X — Unilib[®] 変換を必要とします。
- X なし — Unilib 変換と組み込み変換のどちらかを使用します。

文字セット	Unilib の必要性	説明
cp857		IBM PC トルコ語

文字セット	Unilib の必要性	説明
cp1254		Microsoft Windows トルコ語
iso88599		ISO 8859-9 Latin-5 トルコ語
macturk		Macintosh トルコ語
turkish8		HP TURKISH8

Unicode の文字セット

Adaptive Server は Unicode の文字セット (650 を超える言語に対応) をサポートしています。

- X — Unilib[®] 変換を必要とします。
- X なし — Unilib 変換と組み込み変換のどちらかを使用します。

文字セット	Unilib の必要性	説明
utf8	X	Unicode UTF-8 コード化

ベトナム語の文字セット

Adaptive Server はベトナム語の文字セットをサポートしています。

- X — Unilib[®] 変換を必要とします。
- X なし — Unilib 変換と組み込み変換のどちらかを使用します。

文字セット	Unilib の必要性	説明
cp1258	X	Microsoft Windows ベトナム語

西欧言語の文字セット

Adaptive Server は西欧言語の文字セットをサポートしています。

- X — Unilib[®] 変換を必要とします。
- X なし — Unilib 変換と組み込み変換のどちらかを使用します。

文字セット	Unilib の必要性	説明
ascii8	X	US ASCII、8 ビット・データ、ISO 646
cp437		IBM CP437 — US コード・セット
cp850		IBM CP850 — ヨーロッパ・コード・セット

文字セット	Unilib の必要性	説明
cp860	X	PC ポルトガル語
cp863	X	IBM PC カナダ系フランス語コード・ページ
cp1252	X	Microsoft Windows US (ANSI)
iso_1		ISO 8859-1 Latin-1
mac		Macintosh の標準コード化
roman8		HP ROMAN8
iso 885915	X	ISO 8859-15 Latin-1 ヨーロッパ言語のサポート付き

文字セット名

利用できる文字セットとその Sybase 名を示します。

文字セット	Sybase 名
ASCII 8	acsii_8
Big 5	big5
Big 5HK	big5hk
CP 437	cp437
CP 850	cp850
CP 852	cp852
CP 855	cp855
CP 857	cp857
CP 858	cp858
CP 860	cp860
CP 864	cp864
CP 866	cp866
CP 869	cp869
CP 874	cp874
CP 932	cp932
CP 936	cp936

文字セット	Sybase 名
CP 950	cp950
CP 1250	cp1250
CP 1251	cp1251
CP 1252	cp1252
CP 1253	cp1253
CP 1254	cp1254
CP 1255	cp1255
CP 1256	cp1256
CP 1257	cp1257
CP 1258	cp1258
DEC Kanji	deckanji
EUC-CNS	euccns
EUC-GB	eucgb
EUC-JIS	eucjis
EUC-KSC	eucksc
GB 18030	gb18030
GREEK8	greek8
ISO 8859-1	iso_1
ISO 8859-2	iso88592
ISO 8859-5	iso88595
ISO 8859-6	iso88596
ISO 8859-7	iso88597
ISO 8859-8	iso88598
ISO 8859-9	iso88599
ISO 8859-15	iso15
カザフスタン・キリル語	kz1048
Koi8	koi8
ROMAN8	roman8

文字セット	Sybase 名
ROMAN9	roman9
シフト JIS	sjis
TIS 620	tis620
TURKISH8	turkish8
UTF-8	utf8

文字セットのロード

charset ユーティリティを使用して、Adaptive Server に文字セットとソート順をロードします。**charset** を使用してデフォルトの文字セットとソート順をロードできるのは、インストール時のみです。

Adaptive Server のデフォルトの文字セットとソート順を変更する場合は、『システム管理ガイド：第 1 巻』を参照してください。

構文

```
charset
```

```
[ -U username ]
```

```
[ -P password ]
```

```
[ -S server ]
```

```
[ -I interfaces ]
```

```
[ -v version ]
```

```
sort_order
```

```
[ charset ]
```

表 7 : charset のキーワードとオプション

キーワードとオプション	説明
-U	“sa” としてオペレーティング・システムにログインしていない場合は、コマンド・ラインに “-Usa” または “/username = sa” を指定する必要があります。
-P	コマンド・ラインで “sa” パスワードを指定する。指定しない場合は、“sa” パスワードの入力が要求される。

キーワード とオプション	説明
-S	サーバの名前を指定する。指定しない場合は、 charset が DSQUERY 環境変数を使ってサーバ名を特定する。DSQUERY 環境変数がない場合は、 charset は “SYBASE” という名前のサーバへ接続しようとする。
-I	使用する interfaces ファイルを指定する。指定しない場合、 charset は SYBASE ディレクトリの interfaces ファイルを使用する。
-v	Sybase のバージョンを出力して、終了する。他のオプションを同時に指定しない。
<i>sort_order</i>	charset を使ってデフォルトの文字セットとソート順をロードする場合、Adaptive Server が使用するソート順のファイル名を指定する <i>sort_order</i> パラメータは必須。追加の文字セットをロードする場合は、 <i>charset.loc</i> を使用して文字セット・ファイルの名前を特定する。
<i>charset</i>	Adaptive Server が使用する文字セットのディレクトリを指定する。

言語定義ファイルのソート順

利用できるソート順を示します。

記載されていない言語には、言語固有のソート順はありません。バイナリ・ソート順を使用します。

表 8：利用できるソート順

言語またはスクリプト	ソート順	ファイル名	ID
すべての言語	バイナリ順	binary.srt	50
キリル語	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)	cyrdict.srt	63
	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)	cyrnocs.srt	64
英語 フランス語 ドイツ語	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)	dictiona.srt	51

言語またはスクリプト	ソート順	ファイル名	ID
これらのソート順はすべての西欧の文字セットで使用される。	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)	<code>nocase.srt</code>	52
	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)	<code>nocasepr.srt</code>	53
	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)	<code>noaccent.srt</code>	54
英語 フランス語 ドイツ語 これらのソート順は CP 850 のみで使用される。	代替辞書順 (大文字と小文字を区別する)	<code>altdict.srt</code>	45
	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)	<code>altnoacc.srt</code>	39
	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)	<code>altnocsp.srt</code>	46
ギリシャ語 このソート順は ISO 8859-7 のみで使用される。	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)	<code>elldict.srt</code>	65
ハンガリー語 これらのソート順は ISO 8859-2 のみで使用される。	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)	<code>hundict.srt</code>	69
	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)	<code>hunnoac.srt</code>	70
	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)	<code>hunnocs.srt</code>	71
ロシア語 このソート順は CP 855 を除いたすべてのキリル語の文字セットで使用される。	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)	<code>rusdict.srt</code>	58
	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)	<code>rusnocs.srt</code>	59

言語またはスクリプト	ソート順	ファイル名	ID
スカンジナビア語 これらのソート順は CP 850 のみで 使用される。	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)	scandict.srt	47
	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)	scannocp.srt	48
スペイン語	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)	espdict.srt	55
	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)	espnocs.srt	56
	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)	espnoac.srt	57
タイ語	辞書順	dictionary.srt	51
トルコ語 これらのソート順は ISO 8859-9 のみで 使用される。	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)	turdict.srt	72
	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)	turnoac.srt	73
	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)	turnocs.srt	74

文字セット変換

Backup Server は、クライアントの言語と Adaptive Server の文字セットを使用して、Adaptive Server にメッセージを渡します。Adaptive Server はこのメッセージを変換し、クライアントの言語と文字セットでメッセージを発行します。

文字セットを選択する際は、次の要件に留意します。

- 異機種間環境では、Adaptive Server と Backup Server は、稼働するプラットフォームや使用する文字セットが異なる複数のクライアントとの通信を必要と

する場合があります。データの整合性を維持するため、サーバは異なる文字セット間でコード変換を行います。

- 組み込みコード変換を使用するには、クライアントが使用しているすべての文字セットについて、文字セット定義ファイルをサーバにインストールします。組み込み変換は、多くの文字セットをサポートします。
- Unilib 変換のサポートは、Sybase がサポートするすべての文字セットで利用できます。Unilib 変換を有効化するには、**sp_configure** を使用し、**enable unicode conversions** をオンにする必要があります。『システム管理ガイド：第 1 巻』を参照してください。

Adaptive Server または Backup Server のいずれかが、クライアントの言語または文字セットをサポートしていない場合は、サーバから警告メッセージが出力されます。Backup Server の文字セットが Adaptive Server の文字セットと互換性がない場合も、エラーが発生します。デフォルトでは、Unicode 変換が有効です。

文字セット変換は、同じ言語の文字セット間、または同じ言語グループ内の文字セット間でのみ実行できます。

たとえば、次の西欧言語の文字セット間では自動文字セット変換を実行できます。ASCII 8、CP 437、CP 850、CP 860、CP 863、CP 1252、ISO 8859-1、ISO 8859-15、ROMAN8。同様に、次の日本語の文字セット間でも変換が可能です。CP 932、EUC-JIS、シフト JIS、DEC-Kanji。

ただし、西欧言語の文字セットと日本語の文字セットの間では、コード変換はできません。実行可能な変換の詳細については、『システム管理ガイド：第 1 巻』を参照してください。

サーバ／クライアント間の変換

Adaptive Server がクライアントの言語または文字セットをサポートしていない場合、クライアントはそのサーバと接続できますが、文字変換は実行されません。

ローカライズされたクライアント・アプリケーションを Adaptive Server に接続する場合、サーバは、そのクライアントの言語と文字セットをサポートしているかどうかをチェックします。

- Adaptive Server がその言語をサポートしていれば、すべての文字セット変換が自動的に実行され、クライアントの言語と文字セットを使用してメッセージを表示します。
- Adaptive Server がその言語をサポートしていない場合は、ユーザのデフォルト言語または Adaptive Server のデフォルト言語が使用されます。
- Adaptive Server がその文字セットをサポートしていなければ、クライアントに対して警告を出力し、変換機能をオフにして、言語をアメリカ英語に設定します。

ソート順

文字セットのそれぞれには、1つ以上のソート順 (照合順) が定義されています。ソート順は、ソート順定義ファイル (.srt ファイル) にあります。ソート順定義ファイルは、文字セット定義ファイルに付属しており、同じディレクトリに格納されています。

サイトでの必要に応じて、データのソート順を選択できます。サーバが同時にサポートできるソート順は 1 つのみであるため、すべてのクライアントで機能するソート順を選択します。

警告！ 新しい Adaptive Server のデフォルトの文字セットとソート順の変更をすべて行ってから、ユーザ・データベースの作成や Sybase が提供するデータベースの変更を行ってください。データやデータ構造体が Adaptive Server に追加されてから文字セットやソート順を変更すると、不正な動作が発生する場合があります。データを追加した後の文字セットやソート順の変更については、『システム管理ガイド：第 1 巻』を参照してください。

文字セットで利用できるソート順

Adaptive Server が文字データの整列、比較、インデックスの付与に使用する照合順は、ソート順によって決定されます。それぞれの文字セットには、1つ以上のソート順が定義されています。

ソート順は、文字セット定義ファイルに付属しているソート順定義ファイル (.srt ファイル) に格納されています。

注意： 利用できるソート順は、Adaptive Server にインストールされている文字セットによって異なります。

使用している言語の .srt ファイルを調べれば、その文字セットで利用できるソート順がわかります。ソート順は、次のロケーションに格納されています。

```
$SYBASE/charsets/<charset_name>/*.srt
```

ソート順は、インストール時に指定することも、後で **sqlloc** ユーティリティを使用して指定することもできます。

ソート順の名前	説明
バイナリ順	<p>文字セットの数値バイトの値に従って、すべてのデータをソートする。バイナリ順では、すべて ASCII の大文字をソートしてから小文字をソートする。アクセント付き文字または表意文字 (マルチバイト文字) は、それぞれの標準の順序 (任意の場合もあり) でソートされる。</p> <p>すべての文字セットでは、デフォルトとしてバイナリ順を使用する。バイナリ順がニーズに合わない場合は、インストール時に、または <code>sqlloc</code> ユーティリティを使用して他のソート順を指定する。</p>
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)	アクセント付き文字も含めて、大文字をソートしてから小文字をソートする。各種のアクセント付き文字を認識し、対応するアクセントなしの文字の後にソートする。
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)	大文字と小文字は同様に処理され、ソート結果には両者が混在する。
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)	アクセント記号は無視される。
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)	<p>大文字で書かれた語が小文字で書かれた同一語として扱われる。</p> <p>大文字と小文字が区別されるのは、order by 句を使用する場合のみ。order by 句を使用すると、大文字がソートされてから小文字がソートされる。</p> <p>インストール環境で、order by 句を使用したときに大文字をソートしてから小文字をソートし、それ以外は区別しないで処理する必要がある場合を除き、このソート順を選択しない。この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、大きなテーブルでパフォーマンスが低下する場合がある。</p>
代替辞書順 (大文字と小文字を区別する)	いくつかの西欧言語で使用する。
代替辞書順 (大文字と小文字を区別しない、アクセントを区別しない)	いくつかの西欧言語で使用する。
代替辞書順 (大文字と小文字を区別しない、大文字を優先する)	いくつかの西欧言語で使用する。

ソート順の名前	説明
スペイン語辞書順 (大文字と小文字を区別する)	スペイン語およびほとんどのラテン・アメリカの言語用ロケールで使用する。
スペイン語辞書順 (大文字と小文字を区別しない)	スペイン語およびほとんどのラテン・アメリカの言語用ロケールで使用する。
スペイン語辞書順 (大文字と小文字を区別しない、アクセントを区別しない)	スペイン語およびほとんどのラテン・アメリカの言語用ロケールで使用する。
スカンジナビア語辞書順 (大文字と小文字を区別する)	大文字と小文字を区別する辞書のソート順。 スカンジナビア語で使用する。
スカンジナビア語辞書順 (大文字と小文字を区別しない、大文字を優先する)	スカンジナビア語で使用する。

使用する予定の文字セットのソート順を表示するには、**sqlloc** を使用します。
『システム管理ガイド 第 1 巻』の「文字セット、ソート順、言語の設定」を参照してください。

言語モジュール

Adaptive Server は、ローカライズされたソフトウェア・メッセージをそれぞれの言語モジュールに格納します。

言語モジュールをインストールすると、適切な場所に格納された新しい言語をサポートするメッセージ、文字セット、ソート順を定義したファイルを、インストール・プログラムがロードします。

Adaptive Server と Backup Server をインストールすると、デフォルトでは英語のシステム・メッセージがインストールされます。

新しい言語モジュールのインストール

Adaptive Server のフル・インストールでは、すべての言語コンポーネントが自動的にインストールされます。フル・インストールを選択しなかった場合は、必要に応じて手動で追加の言語モジュールをインストールします。

1. 配布メディアから、言語モジュール・ソフトウェアをロードします。このソフトウェアは、Adaptive Server をロードしたディレクトリと同じディレクトリにロードする必要があります。

2. Adaptive Server の言語を再設定し、必要に応じて文字セットとソート順も再設定します。

メッセージ言語

Adaptive Server のデフォルトでは、アメリカ英語がメッセージの言語としてインストールされます。

言語モジュールには、次の規則が適用されます。

- Adaptive Server のインストールまたは再設定時に、アメリカ英語以外の言語をデフォルト言語として指定できます。指定する言語の言語モジュールもインストールされていることを確認します。
- クライアントがアメリカ英語以外の Adaptive Server メッセージを必要とする場合は、その言語用の言語モジュールをロードする必要があります。ロード後に、Adaptive Server でクライアントが使用する言語を設定できます。
- Adaptive Server がクライアントの言語をサポートしていない場合は、クライアントは Adaptive Server のデフォルト言語でメッセージを受け取る。
たとえば、クライアントの言語がラテン系言語の場合、スペイン語の言語モジュールがインストールされます。Adaptive Server のデフォルト言語としてスペイン語を指定すると、クライアントはスペイン語でメッセージを受信します。

ローカライゼーションのディレクトリ

Sybase のローカライゼーション設定には、locales と charsets のディレクトリがあります。

次の表は、ローカライゼーション・ファイルの構造を示しています。表に示されているのは、ローカライゼーション・ファイルの一部です。

%SYBASE%/ または \$SYBASE/	charsets	charset_name	*.srt ファイル
		charset_name...	charset.loc
		unicode	*.uct ファイル
	locales	language_name	charset_name
		language_name...	charset_name...
		locales.dat	
		message	language_name
			language_name...

locales ディレクトリについて

\$SYBASE/locales ディレクトリには、利用できる各言語に対応するサブディレクトリがあります。各言語のサブディレクトリには、その言語で利用できるそれぞれの文字セットのサブディレクトリがあります。

- これらのサブディレクトリにある .loc ファイルを使用して、Adaptive Server と Backup Server は、指定された文字セットでコード化された指定の言語でエラーをレポートします。
各サブディレクトリには、各種の .loc ファイルが用意されています。これらのファイルの大部分には、指定の製品やユーティリティで使用する、変換済みのエラー・メッセージが格納されています。
- 各言語の utf8 サブディレクトリ内の common.loc ファイルには、現地の日付、時刻、通貨のフォーマットなど、ローカライズされた情報が含まれており、すべての製品で使用されます。
- locales.dat ファイルには、プラットフォーム固有のロケール名を Sybase の言語と文字セットの組み合わせに関連付けるエントリが含まれます。

charsets ディレクトリについて

\$SYBASE/charsets/charset_name 内のファイルには、利用できるソート順の定義など、特定の各文字セットに関連する情報が格納されています。

locales.dat ファイルのエントリで使用するフォーマット

locales.dat ファイルを編集すると、プラットフォームのデフォルトの言語や文字セットを変更したり、プラットフォームのロケール名と Sybase の言語および文字セット名との新しい関連付けを追加したりすることができます。

locales.dat ファイル内の各エントリにより、プラットフォーム固有のロケール定義が Sybase の言語と文字セットの組み合わせにリンクされます。各エントリのフォーマットは次のとおりです。

```
locale = platform_locale, syb_language, syb_charset
```

各パラメータの意味は次のとおりです。

- *platform_locale* は、ロケールのプラットフォーム固有のキーワード。受け入れられる値の詳細については、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。
ロケールがそのサイトのデフォルトとして定義されている場合、*platform_locale* は "default" になります。
- *syb_language* は、\$SYBASE/locales/language_name 内で使用される言語ディレクトリの名前。

- `syb_charset` は、文字セットの変換方法を決定し、`$SYBASE/locales/language_name/charset_name` 内でクライアントのメッセージ・ファイルのディレクトリ・ロケーションを特定する文字セット。

たとえば、次のエントリでは、デフォルトのロケールで言語に `us_english` を使用し、文字セットに `iso_1` を使用するよう指定します。

```
locale = default, us_english, iso_1
```

クライアント・アプリケーションの `locales.dat` ファイルの使用方法

クライアント・アプリケーションは、`locales.dat` ファイルを使用して、使用する言語と文字セットを識別します。

1. クライアント・アプリケーションは、起動時にオペレーティング・システムのロケール設定と `locales.dat` ファイルをチェックして、その設定が Adaptive Server に適切なものであるかどうかを確認します。たとえば、フランス語のロケール・エントリは次のようになります。

```
locale = fr_FR, french, iso_1
```

2. クライアントが Adaptive Server と接続すると、言語と文字セットについての情報がログイン・レコードにある Adaptive Server に渡されます。
3. Adaptive Server は続いて、次の情報を使用します。
 - `iso_1` などの文字セットについての情報。この情報をもとに、クライアントの文字セットを識別し、文字データをこの文字セットに変換できるかどうかを確認します。
 - 言語 (前述の例ではフランス語) と文字セットについての情報。この情報をもとに、クライアントの言語を使用したメッセージがあるかどうかを確認します。

注意： Adaptive Server ソフトウェアが使用するいくつかのロケール・エントリは、すでに `locales.dat` ファイルで定義されています。定義されたエントリが要求に合わない場合は、修正したり、新しいロケール・エントリを追加したりできます。

Adaptive Server のローカライゼーション設定の変更

各言語モジュールは、約 2MB のデータベース領域を使用します。別の言語を追加する場合は、必要に応じて、**alter database** コマンドを使用して master データベースのサイズを増やしてください。

複数の言語を Adaptive Server にインストールする場合、master データベースのサイズが複数の言語を管理できるほど大きくなければ、トランザクション・ログが満杯になる可能性があります。master データベースは、マスタ・デバイス上で

のみ拡張できます。詳細については、『システム管理ガイド』を参照してください。

1. Sybase 環境変数を設定していない場合は、SYBASE.csh または SYBASE.sh を情報源として読み込みます。
2. サーバで Adaptive Server のローカライゼーションを設定するには、次のように **sqlloc** を起動します。

```
$SYBASE/$SYBASE_ASE/bin/sqlloc
```

3. [既存のサーバのローカライズ] を選択します。
4. サーバを選択します。
5. ユーザ名とパスワードを指定します。ユーザには sa 権限が必要です。
6. 次に、以下の項目を指定します。
 - デフォルト言語
 - デフォルトの文字セット
 - デフォルトのソート順
7. インストールする他の言語を選択します。デフォルトの文字セットでサポートされている言語のみを選択できます。

[Add and Remove Languages] ウィンドウに、Sybase で利用できるすべての言語が表示されます。

8. [ローカライズの概要] ウィンドウに、選択した設定オプションの概要が表示されます。[OK] をクリックして、選択内容を確定します。

[Status Output] ウィンドウに、インストールの完了が通知されます。

参照：

- 第 4 章、「Adaptive Server の設定の変更について」 (15 ページ)

Adaptive Server でのその他の文字セットの設定

使用している言語の文字セットとソート順で Adaptive Server を設定します。

前提条件

サーバが稼働している必要があります。また、システム管理者権限が必要です。

手順

1. **charset** ユーティリティを使用して、デフォルトの文字セットとソート順をロードします。

```
$SYBASE/$SYBASE_ASE/bin/charset -Usa -Ppassword  
-Sserver_name sort_order_file character_set
```

`sort_order_file` をソート順のファイル名と置き換えます。

`character_set` を使用する文字セットの Sybase 名と置き換えます。

2. **charset** ユーティリティを使用して、追加する文字セットをロードします。

Adaptive Server の組み込み文字セット変換を使用する場合は、クライアントのプラットフォームで使用するすべての文字セットの定義ファイルをロードする必要があります。Unilib 文字セット変換を使用する場合は、ロードの必要はありません。

3. **isql** を使用して、サーバに "sa" としてログインし、master データベースを選択します。

```
1> use master
2> go
```

4. サーバに新しい文字セットとソート順を設定します。

```
1> sp_configure "default sortorder_id",
2> sort_order_id, "character_set"
3> go
```

`sort_order_id` を使用するソート順の ID と置き換えます。`character_set` を使用する文字セットの Sybase 名と置き換えます。

5. サーバを停止して、再設定のプロセスを開始します。
6. UNIX システムの通常のプロセスを使用してサーバを再起動します。通常、`$SYBASE/$SYBASE_ASE/install` から `RUN_xxx` スクリプトのいずれかを呼び出します。
7. サーバが起動し、すべてのシステム・インデックスを再構築した後、停止します。サーバを再起動して、安定した状態になるまで待ちます。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。

locales.dat ファイルの編集

テキスト・エディタを使用して、`locales.dat` ファイルを編集します。

前提条件

編集後のファイルで問題が発生した場合に備えて、元のファイルのコピーを作成します。

手順

1. メモ帳などのテキスト・エディタを使用して、`locales.dat` ファイルのコピーを開きます。
2. 次のような角カッコで囲まれたセクションを探します。

- Sun Solaris の場合、[sun_svr4]
 - HP の場合、[hp_ux]
 - IBM の場合、[aix]
3. 使用する言語 (*syb_language*) と文字セット (*syb_charset*) の組み合わせを指定したエントリが、このセクションにあるかどうかを確認します。

注意： *platform_locale* の値は、オペレーティング・システムに必要な値と一致していなければなりません。システムの設定ファイルで行われているロケール定義が Sybase のロケール定義と一致していない場合、アプリケーションは正しく稼働しません。

たとえば、Open Client のメッセージをフランス語で表示する場合、Adaptive Server が Roman8 文字セットを使用しているときには、プラットフォームの *locales.dat* エントリを調べて次のエントリを探します。

```
locale = fr_FR, french, roman8
```

4. 必要なエントリを追加するか、既存のエントリを修正します。
5. 変更があった場合はその内容を保存し、テキスト・エディタを終了します。

Backup Server のローカライゼーション設定の変更

RUN_<backup_server_name> ファイルを使用して、Backup Server をローカライズします。

backupserver コマンドの引数の詳細については、『ユーティリティ・ガイド』を参照してください。

Adaptive Server は、起動すると、Adaptive Server エラー・ログと呼ばれるローカルのエラー・ログ・ファイルに情報を書き込みます。

Adaptive Server エラー・ログへの記録は常に有効になっています。ただし、特定のユーザ定義メッセージを作成したり変更したりする場合は、そのメッセージをログから省略するように設定できます。

エラー・ログ・ファイル `$SYBASE/$SYBASE_ASE/install/server_name.log` は次のように動作します。

- 起動するたびに、起動の成功／失敗に関する情報を格納する。
- オペレーション中にサーバによって生成されたエラー・メッセージと情報メッセージを記録する。
- サーバ・プロセスを停止するまで開いたままになる。
- Adaptive Server からの起動メッセージを格納する。

エラー・ログが大きくなりすぎた場合は、次の操作を実行できます。

- **sp_errorlog** を使用してパスを動的に変更する。Adaptive Server で使用されなくなった古いエラー・ログを移動して、使用可能なディスク領域を増やすことができます。
- Adaptive Server を停止して、記録されたメッセージを削除する。

エラー・ログ・フォーマットの説明については、『システム管理ガイド』の「システムの問題の診断」を参照してください。

エラー・ログのパス設定

新しく Adaptive Server を設定すると、インストール・プログラムが Sybase インストール・ディレクトリにエラー・ログのロケーションを設定します。Backup Server には独自のエラー・ログがあります。

各サーバのエラー・ログのデフォルト・ロケーションは次のとおりです。

- Adaptive Server : `$SYBASE/$SYBASE_ASE/install/servername.log`
- Backup Server: `$SYBASE/$SYBASE_ASE/install/servername_back.log`

起動時に、Adaptive Server のエラー・ログ・ファイルの名前とロケーションをコマンド・ラインからリセットできます。**dataserver** コマンドで **-e** 起動パラメータと値を指定して Adaptive Server を起動します。

注意：複数の Adaptive Server が、同一のエラー・ログを共有することはできません。複数の Adaptive Server をインストールする場合は、サーバごとにユニークなエラー・ログ・ファイル名を指定します。

Adaptive Server エラー・ログのパスの設定

RUN_server_name ファイルを編集して、エラー・ログのパスを変更します。

エラー・ログのパスは、\$SYBASE/\$SYBASE_ASE/install/

RUN_server_name ファイルを編集して変更できます。次に例を示します。

```
$SYBASE/ASE-15_0/bin/dataserver -d/Devices/ASE_2K.dat -sASE_2K -i/  
ASE_150 -e$SYBASE/ASE_2K.log -M/ASE_150
```

RUN_server_name ファイルの使用方法については、『ユーティリティ・ガイド』を参照してください。

参照：

- 第 7 章、「サーバの起動と停止」(25 ページ)

メッセージの管理

メッセージをログに書き込むかどうかを制御したり、ユーザ・メッセージを追加したりすることにより、イベント・ロギングを管理します。

イベント・ロギングが有効な場合、次の操作を実行できます。

- sp_addmessage** を使用してユーザ・メッセージを追加したり、**sp_altermessage** を使用して特定のメッセージを Adaptive Server エラー・ログに書き込むかどうかを制御したりする。
『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。
- 設定パラメータを使用して、監査イベントをログに書き込むかどうかを指定する。監査イベントは、Adaptive Server へのユーザのログインの成功 (**log audit logon success**) または失敗 (**log audit logon failure**) に関連します。

ユーザ定義メッセージのロギング

ユーザ定義メッセージを Adaptive Server エラー・ログに書き込むことができます。

Adaptive Server では、次のメッセージを管理できます。

- 新しいメッセージ — 新しいユーザ定義メッセージを `sysusermessages` に追加する場合は、**sp_addmessage** に **with_log** オプションを指定します。このパラメータは、メッセージが表示されるたびにそのメッセージのログを取るよう Adaptive Server を設定します。
- 既存のメッセージ — 既存のユーザ定義メッセージを変更する場合は、**sp_altermessage** に **with_log** オプションを指定します。このパラメータで、次のようにそのメッセージの報告ステータスを切り替えます。
 - TRUE — ロギングを有効にする
 - FALSE — ロギングを無効にする

『リファレンス・マニュアル：プロシージャ』の「**sp_addmessage**」および「**sp_altermessage**」を参照してください。

監査イベントのロギング

sp_configure のパラメータを使用して、ログインなどの監査イベントを Adaptive Server エラー・ログに書き込むかどうかを指定します。

デフォルトでは、Adaptive Server は監査イベントのログを取りません。イベントをログに書き込むには、**sp_configure** で次のパラメータと値を指定します。

- **log audit logon success** を 1 に設定 — 成功した Adaptive Server へのログインのロギングを有効にする

```
sp_configure "log audit logon success", 1
```
- **log audit logon failure** を 1 に設定 — 失敗した Adaptive Server へのログインのロギングを有効にする

```
sp_configure "log audit logon failure", 1
```
- どちらかのパラメータを 0 に設定 — その種類のメッセージのロギングを無効にする

```
sp_configure "log audit logon success", 0
sp_configure "log audit logon failure", 0
```

『システム管理ガイド：第 1 巻』を参照してください。

「データベース・デバイス」とは、Adaptive Server データベースおよびデータベース・オブジェクトを格納するディスク、またはディスクの一部のことです。

Adaptive Server データベースの管理では、日常の作業と、パフォーマンスおよびチューニングなどを検討します。

- ほとんどの管理作業については、『システム管理ガイド 第 1 巻』と『システム管理ガイド 第 2 巻』で詳細に説明しています。
- パフォーマンスの問題については、『パフォーマンス&チューニング・シリーズ』で詳細に説明しています。

デバイスの要件

Adaptive Server のデバイスとデータベースには、最大値と最小値が適用されます。

- デバイスの最大サイズは 4TB。
- 使用可能な最小デバイス・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。データベースは、256 の論理ページを 1 つのグループとして領域を管理します。また、指定可能な最小ディスク・サイズは 1MB です。そのため、使用可能な最小デバイスは、1MB または 256 論理ページのうちの大きい方です。
 - 2K ページ・サーバの最小デバイス・サイズは 1MB
 - 4K ページ・サーバの最小デバイス・サイズは 1MB
 - 8K ページ・サーバの最小デバイス・サイズは 2MB
 - 16K ページ・サーバの最小デバイス・サイズは 4MB
- データベース・デバイスの最大数は 2,147,483,647 です。ただし、Adaptive Server のメモリには各デバイスの説明を保持する必要があるため、この数は実際にはシステムのメモリによって制限されます。使用しているオペレーティング・システムによって、1 つのプログラムが同時に開くことができるデバイスの数も制限されます。
- データベースには最大 2,147,483,648 の論理ページを格納できるため、最大サイズは論理ページ・サイズによって決まります。
 - 2K ページ・サーバの最大データベース・サイズは 8TB
 - 4K ページ・サーバの最大データベース・サイズは 16TB
 - 8K ページ・サーバの最大データベース・サイズは 32TB

- 16K ページ・サーバの最大データベース・サイズは 64TB
- 最小データベース・サイズは、インストール環境の model データベースのサイズです。
- 各データベースは、1 つ以上のディスク区分にある、1 つ以上のデータベース・デバイスに格納されます。1 つのデータベース内のディスク区分の最大数は 8,388,608 です。ただし、Adaptive Server ではアクティブなデータベースすべての説明を保持する必要があるため、この数は実際にはオペレーティング・システムのメモリによっても制限されます。

注意： デフォルトでは、**srvbuild** は `$SYBASE/data` ディレクトリにデバイスを作成します。

データベース・デバイス用のファイルの作成

disk init を使用して、新しいデータベース・デバイスを作成できます。

ロー・パーティションまたはオペレーティング・システム・ファイルを指定できます。ロー・パーティションを使用する場合は、パーティションのフル・パスを指定する必要があります。オペレーティング・システム・ファイルを使用する場合は、フル・パスまたは相対パスを使用します。パス名は、サーバの現在の作業ディレクトリが基準になります。

すべてのデータベース・デバイスに関して、フル・パスを指定することをおすすめします。**disk init** を使用してパス名を指定する場合は、環境変数を使用しないでください。

disk init を使用してデータベース・デバイスを作成する場合の例は次のとおりです。

```
disk init name = "user_device1",  
physname = "/work/data7/device1.dat",  
size = 2048
```

"size = 2048" は、2048 の「仮想」ページをデバイスに割り付けます。仮想ページは 2048 バイトであるため、このコマンドにより 4MB のデバイスが作成されます。

この例のコマンドではデバイス番号を指定せず、代わりにサーバがデバイス番号を選択できるようにしています。特定の番号を特定のデバイスに割り当てる必要がない場合、この方法を使用することをおすすめします。明示的なデバイス番号が必要な場合は、パラメータ `"vdevno = N"` を使用します。*N* は、使用するデバイス番号です。このサーバ上の他のデバイスで使用されているデバイス番号を指定することはできません。すでに使用されているデバイス番号を調べるには、**sp_helpdevice** を使用します。

既存のデータベース・デバイスが小さすぎるのがわかった場合は、**disk resize** コマンドを使用して、デバイスのサイズを大きくします。このコマンドは、**disk init** と同じ "name" パラメータと "size" パラメータを使用します。ただし、size パラメータでは、デバイスの最終的なサイズを指定します。

disk resize コマンドを使用すると、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。disk resize コマンドを使用してサイズを増やすことができるのは、ロー・パーティションとファイル・システム上のデバイスです。デバイスの最小増加量は、1MB と 1 アロケーション・ユニットのどちらか大きい方です。

注意： デバイスのサイズをどのくらい増やすことができるかは、オペレーティング・システムの制約によって制限されます。たとえば、UNIX ロー・パーティションのフル定義サイズを既に割り当てている場合は、そのパーティション上のデバイスを大きくすることはできません。

『システム管理ガイド 第 1 巻』 および 『リファレンス・マニュアル：コマンド』 を参照してください。

デバイス・ファイ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：物理データベースのチューニング』を参照してください。

セキュリティ関連のシステム・アクティビティは、監査証跡に記録されます。監査証跡を使用して、システムへの侵入やリソースの悪用を検出できます。

システム・セキュリティ担当者は、監査証跡を詳細に調べることで、データベース内のオブジェクトへのアクセスのパターンを調べ、特定ユーザの作業を監視できます。監査レコードはユーザごとに追跡できるため、監査システムはユーザによるシステムの誤用に対する抑止になります。

システム・セキュリティ担当者は、監査システムを管理し、監査の開始と停止、監査オプションの設定、監査データの処理を実行できる唯一のユーザです。

参照：

- sybsecurity デバイスと sybsecurity データベース (8 ページ)
- 監査のインストール (120 ページ)
- データベース・デバイスのサイズの確認 (10 ページ)

監査システムのデバイスとデータベース

監査システムの主要コンポーネントは、sybsecurity デバイスと sybsecurity データベースです。

- sybsecurity デバイスと sybsecurity データベースは、監査情報を格納します。
- 監査証跡は、設定時に指定した複数の監査デバイスと監査テーブルで構成されます。
- syslogs トランザクション・ログ・デバイスは、トランザクション・ログを格納します。

sybsecurity デバイスとデータベース

sybsecurity デバイスは、監査設定プロセスの一環として作成される sybsecurity データベースを格納します。sybsecurity データベースは、model データベース内のすべてのシステム・テーブルに加えて、サーバ全体の監査オプションを追跡するためのシステム・テーブルと監査証跡用のシステム・テーブルを格納します。

監査証跡のためのテーブルとデバイス

Adaptive Server は、sysaudits_01 から sysaudits_08 までのシステム・テーブルに監査証跡を格納します。「現在の監査テーブル」は、常に 1 つしかありませ

ん。Adaptive Server は、現在の監査テーブルにすべての監査データを書き込みます。システム・セキュリティ担当者は、**sp_configure** を使用して、現在の監査テーブルの設定や変更を行うことができます。

Adaptive Server で監査を設定する際には、インストール環境に合わせて監査テーブルの数を決定します。監査証跡には最低 2 つまたは 3 つのシステム・テーブルを使用し、各システム・テーブルはマスタ・デバイスとは独立した独自のデバイスに保存します。こうしておけばスレッショルド・プロシージャを使用して現在の監査テーブルを自動的に保管し、いっぱいになったら新しい空のテーブルに切り替えて次の監査レコードを保管できます。

syslogs システム・テーブルのデバイス

監査の設定時に、トランザクション・ログを格納する *syslogs* システム・テーブル用に別のデバイスを指定する必要があります。*syslogs* テーブルはすべてのデータベースに存在し、データベースで実行されたトランザクションのログを格納します。

監査のインストール

auditinit ユーティリティまたは **installsecurity** スクリプトを使用して、監査をインストールします。

Adaptive Server に初めて監査をインストールする場合、次の 2 つの方法があります。

- **installsecurity** スクリプトを使用する。『セキュリティ管理ガイド』を参照してください。
- **auditinit** ユーティリティを使用する。

参照：

- *sybsecurity* デバイスと *sybsecurity* データベース (8 ページ)
- 第 18 章、「データベース管理システムの監査」 (119 ページ)
- データベース・デバイスのサイズの確認 (10 ページ)

監査デバイスに関するインストール前の推奨事項

インストール前の推奨事項を実行してから、監査をインストールしてください。

- *sybsecurity*、*syslogs*、*sysaudits* の各テーブル・デバイス用に用意するデバイスのロケーションを決定する。この情報はあとで必要になります。
- システムで必要最低限の数の監査デバイスを設定する。少なくとも 3 つのデバイスを設定する必要があります。**sp_addauditable** を使用すると、後で監査デ

バイスを追加できます。『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。

- 監査テーブルとデバイスを 1 対 1 の割合でインストールする。同一デバイスを共有するテーブルは、スレッショルド上限値も共有します。これらのテーブルは、同じデバイス上にあるので、そのデバイスがいっぱいになってしまうと、続けて使用することができません。
- デバイスごとに 1 つずつ監査テーブルをインストールする。これによって、監査レコードを失うことなくスムーズに実行する監査システムを設定できます。2 つの監査テーブルを使用すると、1 つがいっぱいになったら、もう 1 つの方に切り替えることができます。さらに、3 つ目の監査テーブルを使用すると、1 つのデバイスで障害が発生した場合、システム・セキュリティ担当者は新しいスレッショルド・プロシージャをインストールして、障害が発生したデバイスが修復されるまでそのデバイスを省略するようにデバイスのローテーションを変更できます。
- デバイスをテーブルより大きくする。監査テーブルとデバイスを 3 つしか使用しない場合、テーブルとデバイスのサイズは同じでかまいません。監査テーブルとデバイスを追加することにより、監査機能を強化できるからです。テーブルやデバイスが上限値 (6 から 8) 近くで作業する場合は、デバイスをテーブルより十分に大きくします。監査機能を強化する必要があるものの、追加できるデバイスがほとんどまたはまったくない場合、後でテーブルのサイズをデバイスのサイズに合わせて増やすことができます。

ファイル・システム・デバイスを使用する場合、**dsync** 属性をオンに設定するか、そのデバイスで **directio** 属性を使用します。

参照：

- **directio** パラメータ (9 ページ)
- **dsync** パラメータ (9 ページ)

Adaptive Server での監査の設定

auditinit コマンドを使用して、Adaptive Server で監査を設定します。

1. Sybase 環境変数を設定していない場合は、SYBASE.csh ファイルまたは SYBASE.sh ファイルを情報源として読み込みます。
2. UNIX プロンプトで、次のように入力します。

```
$SYBASE/$SYBASE_ASE/install/auditinit
```

auditinit によって次のように表示されます。

```
AUDITINIT
1. Release directory: /usr/u/sybase
2. Configure a Server product
```

3. [サーバ製品の設定] を選択します。
4. [Adaptive Server] を選択します。
5. [既存の Sybase サーバの設定] を選択します。
6. 設定するサーバを選択します。
7. 選択したサーバに SA パスワードを入力します。
8. [Sybase サーバの設定] 画面で [監査機能の設定] を選択します。

auditinit のメニューに従って作業を進めるときに、表示されるデフォルト値を変更できます。各メニューを終了するときには、[Ctrl + A] を押してデフォルト値や変更した値を受け入れて、次のメニューに移ります。

```
CONFIGURE AUDITING
1. Configure auditing:  no
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry

List of devices for the audit tables:
Logical name      Physical name      Segment name      Table name      Size

Device for the audit database transaction log:
Logical name      Physical name      Segment name      Table
name      Size
```

9. [監査機能の設定] 画面で [監査機能の設定] を選択します。
10. Adaptive Server を再起動して、変更を有効にします。

監査データベースのトランザクション・ログ用のデバイスの作成

監査トランザクション・ログ用のデバイスを追加します。

1. [監査機能の設定] メニューから [監査データベースのトランザクション・ログ・デバイスの追加] を選択します。

auditinit で [監査用の新しいデバイスの追加/変更] メニューが表示されます。

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1. sybsecurity physical device name:
2. Logical name of the device:
3. Size of the new device (Meg):
4. Device size for auditing:
```

2. [sybsecurity 物理デバイス名] を選択します。

auditinit で物理名の入力が必要で、デフォルトが表示されます (使用可能な場合)。

```
Enter the physical name of the device to use for the sybsecurity
database (default is''):
/dev/path_to_partition
```

path_to_partition は、デバイスのロー・パーティションへのパスです。

3. 物理デバイスのフル・パス名を入力します。
4. [Return] キーを押します。

auditinit で [監査用の新しいデバイスの追加/変更] メニューが表示され、デバイスの物理名に選択した値が表示されます。

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1.sybsecurity physical device name:
   /secret1/sybase_dr/install/auditlog.dat
2.Logical name of the device:
3.Size of the device:
4.Device size for auditing:
```

5. このメニューの残りの項目に進みます。その際、次の点に注意してください。
 - Sybase では、トランザクション・ログのサイズを最低 2MB にすることをおすすめします。
 - **auditinit** では、[監査用の新しいデバイスの追加/変更] メニューの [デバイスのサイズ] と [監査機能用のデバイスのサイズ] の両方にサイズが表示されます。
 - [監査機能用のデバイスのサイズ] のデフォルト値は、監査タスクのログにデバイス全体を使用するという前提に基づいて、デバイスのサイズと等しくなっています。デバイスのサブセットのみを使用し、[デバイスのサイズ] の値を編集します。
6. [Ctrl+A] を押して、[監査用の新しいデバイスの追加/変更] メニューに表示される設定を受け入れます。

auditinit が [監査機能の設定] メニューに戻り、作成したすべてのデバイスが表示されます。

```
CONFIGURE AUDITING
1. Configure auditing: yes
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry

List of devices for the audit tables:
Logical name      Physical name      Segment name      Table
name             Size
6. Audit_01'      /secret1/sybase_   dr/install/
aud1.dat' sysaudits_01      5
7. Audit_02'      /secret1/sybase_   dr/install/
aud2.dat' sysaudits_02      5
8. auditlog       /secret1/.../
auditlog.dat logsegment syslogs           2
```

7. 監査の設定を実行する準備ができたなら、[Ctrl+A] を押します。
8. [Sybase サーバの設定] 画面で、もう一度 [Ctrl+A] を押します。次のようなメッセージが表示されます。

```
Execute the Sybase Server Configuration now?
```

9. "y" (yes) を入力します。

auditinit によって、監査のインストール作業が実行されます。インストールが正常に完了すると、次のようなメッセージが表示されます。

```
Running task: install auditing capabilities.  
.....Done  
Auditing capability installed.  
Task succeeded: install auditing capabilities.  
Configuration completed successfully.  
Press <return> to continue.
```

監査の有効化

sp_configure を使用して、監査を有効にします。

前提条件

監査を有効にするには、SSO パーミッションが必要です。

手順

監査を有効にするには、次のように入力します。

```
sp_configure 'auditing', 1
```

『セキュリティ管理ガイド』を参照してください。

デバイス・エントリの削除

監査デバイスを削除します。

1. [監査機能の設定] メニューから [デバイス・エントリの削除] を選択します。
2. 削除するデバイスの番号を入力します。
3. [Return] キーを押します。

デバイス・エントリの変更

監査用のデバイスの名前またはサイズを変更します。

1. [監査機能の設定] メニューから [デバイス・エントリの変更] を選択します。
2. 変更するデバイスの番号を入力します。

[監査用の新しいデバイスの追加/変更] メニューに、選択したデバイスに関する情報が表示されます。

```
ADD/CHANGE A NEW DEVICE FOR AUDITING  
1. sybsecurity physical device name:  
   /secret1/sybase_dr/install/audlog  
2. Logical name of the device: aud.log
```

```
3. size of the new device (Meg): 5
```

```
4. Device size for auditing:5
```

3. 変更する残りのエントリをそれぞれ選択します。
4. [Ctrl + A] を押して新しいエントリを保存します。

Transact-SQL 構文のオンライン・ヘルプのインストール

\$SYBASE/\$SYBASE_ASE/scripts ディレクトリには、構文ヘルプ・データベース sybsyntax をインストールするためのスクリプトが格納されています。

このデータを取得するには、**sp_syntax** 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。

scripts ディレクトリには、サーバに含まれている Sybase 製品に応じて、**sp_syntax** スクリプトが 1 つ以上格納されています。

表 9 : sp_syntax インストール・スクリプト

スクリプト	製品
ins_syn_cl	Open Client Client-Library™
ins_syn_esql	Embedded SQL™
ins_syn_os	Open Server
ins_syn_sql	Transact-SQL

ins_syn_sql スクリプトは、すべての Adaptive Server インストール環境に含まれます。このスクリプトには、Transact-SQL、システム・プロシージャ、Sybase ユーティリティの構文情報が含まれています。sybsyntax データベースの SQL に関する部分をインストールするには、このスクリプトを実行します。

これらのスクリプトは、使用しているサーバでの Sybase 情報の必要度に応じてインストールできます。最初に実行するスクリプトでは、sybsyntax データベースと必要なテーブルおよびインデックスが作成されます。それ以降に実行するスクリプトはすべて、データベースにある既存の情報に追加されます。スクリプトを再実行すると、以前にインストールした情報ローがデータベース内のテーブルから削除された後、再インストールされます。

警告！ ins_syn_cl スクリプトと ins_syn_os スクリプトは競合します。両方のスクリプトを実行すると、エラーが発生します。

sybsyntax データベースのデフォルトのデバイス

デフォルトでは、sybsyntax インストール・スクリプトは、デフォルトのデータベース・デバイスとして指定されたデバイスに sybsyntax データベースをインストールします。

sp_diskdefault を使用して、デフォルト・ディスクとしてインストールされているマスタ・デバイスのステータスを変更したり、別のデフォルト・デバイスを指定したりしていない場合、sybsyntax はマスタ・デバイスにインストールされます。この設定を使用しないことをおすすめします。本来は master データベースの今後の拡張に使用できるよう残して置くべき貴重な領域を sybsyntax が使用してしまうためです。

sybsyntax がマスタ・デバイスにインストールされないようにするには、次のいずれかを実行します。

- **sp_diskdefault** を使用して、マスタ・デバイス以外のデフォルト・デバイスを指定する。**sp_diskdefault** については、
- 実行するそれぞれの sybsyntax インストール・スクリプトを修正して、別のデバイスを指定する。

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。

sybsyntax のインストール

sybsyntax インストール・スクリプトは、データベースと必要なテーブルおよびインデックスをインストールします。

1. sybsyntax データベースを格納するデバイスのタイプ (ロー・パーティション、論理ボリューム、オペレーティング・システム・ファイルなど) とロケーションを決定します。
2. 元のスクリプトのコピーを作成します。編集したスクリプトで問題が発生した場合に備えて、必ずこのコピーにアクセスできるようにしておいてください。
3. 必要に応じて、テキスト・エディタを使用してスクリプトを編集し、デフォルト・デバイスを、マスタ・デバイスから手順 1 で作成したデバイスに変更します。

デフォルト・デバイスを指定する次のセクションをコメントにします。

```
/* create the database, if it does not exist */  
if not exists (select name from sysdatabases  
where name = "sybsyntax")
```

```

begin
  /* create the sybsyntax table if it doesn't exist */
  /* is the space left on the default database
  devices > size of model? */
  if (select sum (high-low +1) from sysdevices where status
    & 1 = 1) - (select sum(size) from sysusages, sysdevices
    where vstart >= sysdevices.low
    and vstart <= sysdevices.high
    and sysdevices.status &1 = 1) >
    (select sum(sysusages.size) from sysusages
    where dbid = 3)
  begin
    create database sybsyntax
  end
  else
  begin
    print "There is not enough room on the default
    devices to create the sybsyntax database."
  return
  end
end

```

4. このセクション全体をコメントにした後、次のような行をスクリプトに追加します。

```
create database sybsyntax on device_name
```

device_name は、sybsyntax をインストールするデバイスの名前です。

5. 次のスクリプトを実行します。

```
isql -Usa -Ppassword -Sservername < $SYBASE/$SYBASE_ASE/scripts/
ins_syn_sql
```

sa はシステム管理者のユーザ ID、*password* はシステム管理者のパスワード、*servername* はデータベースをインストールする Adaptive Server です。

DSQUERY 環境変数を *servername* に設定している場合は、サーバ名を \$DSQUERY に置き換えることができます。

6. sybsyntax データベースがインストールされ、正しく動作していることを確認するには、**isql** を使用して、データベースをインストールしたサーバにログインし、**sp_syntax** を実行します。

```
isql -Usa -Ppassword -Sservername

1> sp_syntax "select"
2> go
```

Adaptive Server は、“select” という単語またはその単語の一部を含むコマンド・リストを表示します。

索引

A

Adaptive Server

- common.loc 106
- interfaces ファイル内の名前 64
- locates.dat 106
- UNIX コマンド・ラインからの起動 25
- エラー・ログのパス 112
- オペレーティング・システムとともに起動 27
- カスタマイズ機能 57
- 起動スクリプト 29
- クライアントの通信 59
- 言語、変更 88
- ソート順 88
- 停止 32
- データベース・デバイスの要件 10
- デフォルト設定 57
- デフォルトのソート順 88
- デフォルトの文字セット 88
- 変換、クライアント間 101
- 文字セット 100
- 文字セット、変更 88

auditinit ユーティリティ 7, 8

B

Backup Server

- UNIX コマンド・ラインからの起動 25
- オペレーティング・システムとともに起動 27
- 設定 87
- デフォルト、Adaptive Server 57
- デフォルト設定 57
- 文字セット 100, 110

C

charsets ディレクトリ 102, 105
説明 106

D

dbcc checkstorage、データベース 7, 8

dbcc エラー・メッセージ 49

dbccdb データベース 7, 8

Dec-Kanji 文字セット 100

directio 9

disk init コマンド 11

dscp ユーティリティ 13

マスタ interfaces ファイルの作成 66

dsedit ユーティリティ 13

マスタ interfaces ファイルの作成 66

DSLISN 環境変数 2

DSQUERY 環境変数

クライアント接続 68

説明 60

名前 69

複数のネットワーク、異なる値の使用 69

E

ESP。「拡張ストアド・プロシージャ (ESP)」を
参照 1

EUC-JIS 文字セット 100

H

HP-UX

netstat コマンド 48

ps コマンド 48

sar コマンド 48

time コマンド 48

vmstat コマンド 48

タイムアウト時間 46

ネットワーク・プロトコル 64

I

I/O モニタリング 48

IBM RS/6000

iostat コマンド 48

netstat -v コマンド 48

netstat コマンド 48

no -a コマンド 48

- ps コマンド 48
- time コマンド 48
- vmstat コマンド 48
- システムのモニタ 48
- タイムアウト時間 46
- ネットワーク・プロトコル 64
- interfaces ファイル
 - Adaptive Server の名前 64
 - Adaptive Server、使用 61
 - API 要素 64
 - delay_interval 要素 64
 - device 要素 64
 - dscp を使用したマスタ・ファイルの作成 66
 - dsedit を使用したマスタ・ファイルの作成 66
 - host 要素 64
 - loghost プレースホルダ 67
 - machine 要素 64
 - network 要素 64
 - port 要素 64
 - protocol 要素 64
 - retry_attempt 要素 64
 - servername 要素 64
 - service_type 要素 64
 - SPX 用の address 要素 64
 - 異機種間環境 60
 - エントリ内のユニークな要素 67
 - クエリ・ポート・バックアップの設定 70
 - クライアントが使用 59
 - クライアント・バージョンとサーバ・バージョン、違い 61
 - 作成、経験の浅い方 66
 - 自動作成 59
 - 使用、クライアント 59
 - スペース 63
 - 説明 59
 - テキスト・エディタを使用したマスタ・ファイルの作成 66
 - デフォルト・ロケーション 59
 - 同機種間環境 60
 - 内容 60
 - 複数のネットワーク 60, 67
 - 複数のネットワーク・リスナ 68
 - ロケーション 59
 - interfaces ファイル内の address 要素
 - TCP プロトコル・エントリ 64
 - interfaces ファイル内の API 要素
 - 説明 64
 - interfaces ファイル内の delay_interval 要素 64
 - interfaces ファイル内の device 要素 64
 - interfaces ファイル内の host 要素 64
 - interfaces ファイル内の loghost 67
 - interfaces ファイル内の machine 要素 64
 - interfaces ファイル内の network 要素 64
 - interfaces ファイル内の port 要素 64
 - interfaces ファイル内の retry_attempts 要素 64
 - interfaces ファイル内の servername 要素 64
 - interfaces ファイル内の service_type 要素 64
 - interfaces ファイル内のスペース 63
 - interfaces ファイル内のタブ文字 63
 - interpubs サンプル・データベース 7
 - iostat コマンド
 - IBM RS/6000 48
 - Sun Solaris 49
 - iso-Latin1 文字セット 88
- J**
 - jpubs サンプル・データベース 7
- K**
 - KEEPALIVE オプション、TCP/IP 46
 - kill コマンド 33, 34
- L**
 - LD_LIBRARY_PATH 環境変数 2
 - LDAP
 - interfaces ファイルとの違い 78
 - libtcl.cfg での指定 81
 - アクセス制限 77
 - エントリ例 78
 - ディレクトリ・スキーマ 78
 - ディレクトリの定義 78
 - 複数のディレクトリ・サービス 84
 - 有効化 82

LDAP ライブラリ
環境変数 82
ロケーション 82

ldapurl
キーワード 82
定義 81
例 81

LIBPATH 環境変数 2
libtcl*.cfg ファイル 81
フォーマット 81
目的 81
ロケーション 81
libtcl*.cfg ファイル
パスワード 85
locales ディレクトリ 105

M

master
デバイス 5, 10
maxfiles カーネル・パラメータ 40
maxfiles_lim カーネル・パラメータ 40
mbuf プール 48
model データベース 5

N

NCR
システムのモニタ 48
netstart -v コマンド (IBM RS/6000) 48
netstat コマンド
HP-UX 48
IBM RS/6000 48
Sun Solaris 49
no -a コマンド (IBM RS/6000) 48

P

PATH 環境変数 2
PID。「プロセス ID」を参照 33
PPID。「親プロセス ID」を参照 33
protocol
interfaces ファイル内の要素 64
ps コマンド
HP-UX 48
IBM RS/6000 48

Sun Solaris 49
pubs2 サンプル・データベース 7
pubs3 サンプル・データベース 7
pwdcrypt
パスワードの暗号化 85
ロケーション 85

R

roman8 文字セット 88
run control ディレクトリ 29

S

sar コマンド
HP-UX 48
setenv コマンド 39
setperm_all コマンド 39
shutdown コマンド 32, 33
sllloc ユーティリティ 102
SMIT (System Management Interface Tool) 44
SPX ネットワーク・プロトコル 13, 64
srt ファイル 102
srvbuild ユーティリティ 13
startserver ユーティリティ 27
stty 設定 39
Sun Solaris
iostat コマンド 49
netstat コマンド 49
ps コマンド 49
time コマンド 49
vmstat コマンド 49
ネットワーク・プロトコル 64
sundiag システム診断ツール 47
Sybase
グローバルライゼーション・サポート 105
SYBASE 環境変数 2
Sybase のグローバルライゼーション・サポート
87
SYBASE_OCS 環境変数 2
SYBASE_SYSAM 環境変数 2
sybmgmtldb 8
sybsecurity
データベース 7, 8, 119
デバイス 7, 8, 10
sybsyntax データベース 127

索引

sybsystemdb 5
 デバイス 10
sybsystemprocs データベース 6
sysprocsdev デバイス 5
 最小サイズ 10
 目的 6
System Management Interface Tool (SMIT) 44

T

TCP/IP 46, 64
 KEEPALIVE オプション 46
TCP/IP ネットワーク・プロトコル 13
tempdb データベース 5
time コマンド
 HP-UX 48
 IBM RS/6000 48
 Sun Solaris 49

U

Unicode
 文字変換 90-94
UNIX
 ネットワーク・プロトコル 64
 ハードウェア・エラー・メッセージ 47
UnixWare
 ネットワーク・プロトコル 64
us_english 言語 88

V

vmstat コマンド
 HP-UX 48
 IBM RS/6000 48
 Sun Solaris 49

W

Windows ソケット
 デフォルトのソケット 57

X

XP Server
 デフォルト設定 57

あ

アクセント記号付き文字 17, 102
アラビア語の文字セット 90

い

異機種間環境 88, 100
 interfaces ファイル 61
 説明 61
インストール環境のカスタマイズ 57

え

エラー、dbcc メッセージ内 49
エラー・ロギング
 設定 111
エラー・ログのパス 57, 111, 112
 設定 112

お

オペレーティング・システム
 管理者 2
親プロセス ID (PPID) 33
オンライン構文ヘルプ 127

か

拡張ストアド・プロシージャ (ESP) 1
可変長カラム、最大サイズ 22
カラム・サイズ 22
環境変数
 DSLISTEN 2
 DSQUERY 60, 69
 LD_LIBRARY_PATH 2
 LIBPATH 2
 PATH 2
 SYBASE 2
 SYBASE_OCS 2
 SYBASE_SYSAM 2
韓国語の文字セット 90, 93
監査
 auditinit ユーティリティを使用したインス
 トール 120

installsecurity スクリプトを使用したインストール 120
 グローバル・オプション 119
 追跡用のテーブル 119
 データベース 119
 データベース・デバイスの推奨サイズ 10
 デバイス 119
 プロセス 119
 監査証跡
 概要 119
 監査システム 119
 システム監査テーブル 119

き

起動スクリプト 28
 共有メモリ・ファイル 34
 ギリシャ語の文字セット 90, 92
 キリル語の文字セット 90, 91

く

クエリ
 サービスの種類 61
 ポート・バックアップの設定 70
 クライアント
 Adaptive Server の通信 59
 DSQUERY 68
 アプリケーションと locales.dat ファイル 107
 デフォルトの文字セット 88
 ファイル・サーバ 69
 変換、サーバ間 101
 クライアントと Adaptive Server との通信 59
 クライアントの interfaces ファイル
 異機種 61
 クライアント・バージョンとサーバ・バージョンの違い 61
 同機種 61
 グローバリゼーション・サポート、Sybase 57, 87, 105

け

権限 11

言語

 翻訳サポート 87
 メッセージの選択 105
 言語モジュール 105
 新規インストール 104
 デフォルト 57
 翻訳されたシステム・メッセージ 87
 メモリ要件 107
 ローカライゼーション・ファイル 88, 104
 現在のファイル記述子の表示 39
 現地の日付、時刻、通貨のフォーマット 106

こ

コード変換
 文字セット間 100
 国際化システム
 Sybase のサポート 87
 固定長カラム、最大サイズ 22
 コマンド
 iostat 48, 49
 maxfiles 40
 maxfiles_lim 40
 netstat 48, 49
 netstat -v 48
 no -a 48
 ps 48, 49
 sar 48
 setenv 39
 setperm_all 39
 time 48, 49
 vmstat 48, 49

さ

サーバ
 名前の要件 64
 サーバの起動
 UNIX コマンド・ラインから 25
 サーバの追加、LDAP 83
 サーバの名前の要件 64
 サービスの種類
 クエリ 61
 リスナ 61
 作成
 dscp ユーティリティを使用したマスタ interfaces ファイル 66

索引

dseditを使用したマスタ interfaces ファイル
66
interfaces ファイル 60, 66
interfaces ファイルの自動 59
経験の浅い方向けの interfaces ファイル
66
テキスト・エディタを使用したマスタ
interfaces ファイル 66

し

辞書のソート順 102
スカンジナビア語 102
スペイン語 102
システム監査テーブル 119
システム・データベース
格納 10
システムの再起動スクリプト 28
システムのモニタ
IBM RS/6000 48
NCR 48
システム・プロシージャ、格納ロケーション
6
システム・メッセージ、翻訳 87
シフト JIS 文字セット 100
照合順。タグ。「ソート順」を参照 102

す

スカンジナビア語辞書のソート順 102
スクリプト
C シェル 49
管理 49
起動 28, 29
サンプル管理 49
システムの再起動 28
スペイン語辞書のソート順 102

せ

西欧言語の文字セット 94
セキュリティ。「監査」参照 119
設定
Backup Server 87
文字セット 110
設定、デフォルト 57

全ページロックのカラム・サイズ 22

そ

ソート順 102
Adaptive Server のデフォルト 88
大文字と小文字 102
辞書 102
定義ファイル 102
データベース 102
バイナリ 102
変更 88
文字セット 102
ソート順における大文字と小文字 102

た

タイ語の文字セット 90, 93
タブ文字、interfaces ファイル 63

ち

中国語 (簡体字) の文字セット 90
中国語 (繁体字) の文字セット 91
中国語の文字セット 90

て

ディレクトリ
charsets 102, 106
run control 29
ローカライゼーション 105
ディレクトリ・スキーマ、LDAP 78
データオンリーロックのカラム・サイズ 22
データの変換 87
データベース 102
dbccdb 7
master 6
model 5
pcidb 7
sybsecurity 7, 8
sybsystemprocs 6
tempdb 5
サンプル 7
デバイス 115

データベース・デバイス
 master 5
 sybsyntax 128
 sysprocsdev 6
 ロー・パーティションの準備 11
 データベースの仕様
 クエリ 21
 サイズ 19, 20
 参照整合性制約 21
 データベース 19, 20
 テーブルの仕様 19, 20
 バッファの仕様 21
 プロシージャの仕様 21
 デバイス
 ファイル 115
 デフォルト
 Adaptive Server の言語 88
 Adaptive Server の文字セット 88
 インストールされる文字セット 88
 言語、変更 88
 ソート順 88
 文字セット、変更 88
 でふおるとさーばめいとしての \$SYBASE かん
 きょうへんすう 67

と

ドイツ語サンプル・データベース 7
 東欧言語の文字セット 91
 同機種間環境
 interfaces ファイル 61
 説明 61
 トルコ語の文字セット 90, 93

な

名前付きパイプ
 デフォルト・パイプ 57

に

日本語
 サンプル・データベース 7
 文字セット 92

ね

ネットワーク
 DSQUERY 69

interfaces ファイル 59
 障害 70
 ステータスのモニタ 48
 デフォルト設定 57
 バックアップ接続 70
 複数 60
 ネットワーク・サポート
 デフォルト設定 57
 ネットワーク・プロトコル
 Digital UNIX 64
 HP-UX 64
 IBM RS/6000 64
 SPX 13
 Sun Solaris 64
 TCP/IP 13
 UnixWare 64

は

パーティション・マップ
 障害の回避 11
 ハードウェア・エラー・メッセージ 47
 UNIX 47
 パーミッション 39
 リストア 39
 バイナリ・ソート順 102
 パス、エラー・ログ 111
 パスワードの暗号化
 libtcl*.cfg 85
 pwddcrypt 85
 バルト語の文字セット 90

ひ

非同期 I/O (AIO)
 HP ドライバ 44
 有効化 44
 非同期 I/O の有効化 44

ふ

ファイル
 runserver 26
 共有メモリ 34
 ソート順定義 (.srt) ファイル 102
 デバイス・ファイル 115

索引

- ローカライゼーション 88, 104
- ファイル記述子
 - 現在の表示 39
- ファイル記述子の制限値 41
- ファイル・サーバ 69
- 複数のインストール環境
 - 1つの interfaces ファイルの作成 60, 66
 - interfaces ファイルへの影響 61
- 複数のディレクトリ・サービス
 - LDAP 84
- 複数のネットワーク
 - interfaces ファイル 60, 67
 - ネットワーク障害時のバックアップとして使用 70
- プラグ可能コンポーネント・インタフェース (PCI) 7
- フランス語サンプル・データベース 7
- プロシージャ、Sybase 拡張ストアド・プロシージャ 1
- プロセス ID (PID) 33
- プロトコル
 - SPX 64
 - TCP/IP 64

へ

- ベトナム語の文字セット 94
- ヘブライ語の文字セット 90, 92
- 変換、Unicode 文字 90-94

ほ

- ポート番号と interfaces ファイル 68
- 翻訳されたメッセージ
 - システム 87

ま

- マスタ
 - interfaces ファイル 60, 66
- マニュアル
 - 翻訳された Adaptive Server 87

め

- メッセージ
 - 言語の選択 105

- ハードウェア・エラー 47

も

- 文字型デバイス
 - I/O パフォーマンスの向上 44
- 文字セット 100
 - アクセント記号付き文字 17
 - 異機種間環境 100
 - 英語 17
 - クライアントによる選択 88
 - コード変換 100
 - 設定 110
 - ソート順 102
 - データベース 102
 - デフォルト 88
 - 変換 100
 - 変更 88
- 文字セット間の変換 100
- モニタリング
 - I/O 48
 - 仮想メモリの使用状況 48
 - ネットワーク・ステータス 48

ゆ

- ユーザ接続 40
- ユーティリティ
 - slloc 102

ら

- ラテン文字セット 90

り

- リスナ・サービス 61

ろ

- ローカライズされたエラー・メッセージ 106
- ローカライゼーション・サポート 57, 87
- ロー・パーティション
 - 使用可能 11
 - データベース・デバイスの最小サイズ 11

ロー・ブロック・デバイス
「文字型デバイス」を参照 44
ログイン
sa 2

スーパーユーザ 2
論理ページ・サイズと最大カラム・サイズ 22

