



設定ガイド

Adaptive Server[®] Enterprise

15.7

[UNIX 版]

ドキュメント ID : DC35824-01-1570-01

改訂 : 2011 年 9 月

Copyright © 2011 by Sybase, Inc. All rights reserved.

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

このマニュアルの内容を弊社の書面による事前許可を得ずに、電子的、機械的、手作業、光学的、またはその他のいかなる手段によっても、複製、転載、翻訳することを禁じます。

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

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

Java および Java 関連の商標は、米国およびその他の国における Sun Microsystems, Inc. の商標または登録商標です。

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

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	概要	1
	Adaptive Server について	2
	システム固有の問題	2
	システムのユーザの役割	3
	環境変数	3
	Adaptive Server のデバイスとシステム・データベース	5
	マスタ・デバイス	5
	sybssystemdb 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	6
	データベース・デバイスとしての UNIX オペレーティング・ システム・ファイルの使用	8
	データベース・デバイスのロケーション、タイプ、サイズの決定	9
	クライアント/サーバ通信	11
	Adaptive Server の設定の変更	13
	英語以外の言語	13
	Adaptive Server の仕様	14
第 2 章	サーバの起動と停止	17
	概要	17
	サーバの起動要件	17
	サーバの起動	18
	サーバの起動パラメータ	18
	RUN_server_name ファイルの使用	19
	startserver コマンドの使用	19
	オペレーティング・システムの再起動時にサーバを起動する	20
	HP-UX の場合	20
	IBM RS/6000 の場合	20
	Sun Solaris と Linux の場合	21
	最初のインストール後の XP Server の起動	23
	サーバの停止	23
	Adaptive Server の停止	24
	Backup Server の停止	24
	kill コマンドの使用	25
	停止と共有メモリ・ファイル	26
	Linux のヒュージ・ページ	27

第 3 章	オペレーティング・システムの設定	29
	環境変数の確認.....	29
	srvbuild または srvbuildres を使用した新しいサーバの設定.....	30
	stty 設定の使用.....	33
	正しいパーミッションのリストア.....	33
	ファイル記述子とユーザ接続.....	34
	HP-UX の場合.....	34
	AIX の場合.....	34
	Linux の場合.....	34
	Sun Solaris の場合.....	35
	現在のソフト制限値とハード制限値の表示.....	35
	ソフト制限値を増やす方法.....	35
	ハード制限値を増やす方法.....	36
	サンプル・プログラム.....	36
	非同期ディスク I/O の有効化.....	37
	クライアント接続のタイムアウト時間の調整.....	40
	HP-UX 11 以降の場合.....	40
	IBM RS/6000 の場合.....	40
	Sun Solaris の場合.....	41
	Linux の場合.....	41
	ハードウェア・エラーのチェック.....	41
	HP-UX の場合.....	41
	IBM RS/6000 の場合.....	42
	Sun Solaris の場合.....	42
	Linux の場合.....	42
	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リング.....	42
	HP-UX の場合.....	43
	IBM RS/6000 の場合.....	43
	Sun Solaris と Linux の場合.....	44
	データベースの整合性の検査.....	44
第 4 章	Adaptive Server のデフォルト設定	47
	デフォルト設定.....	47
第 5 章	ネットワークを介する通信の設定	49
	Adaptive Server で使用するディレクトリ・サービス・エントリの 決定方法.....	50
	クライアントのディレクトリ・サービスの使用方法.....	51
	ディレクトリ・サービス・エントリの作成.....	51
	サポートされているディレクトリ・ドライバ.....	52
	interfaces ファイルの内容.....	52
	異機種間環境と同機種間環境.....	53
	interfaces ファイルのフォーマットについて.....	54
	interfaces ファイルのエントリの要素.....	55

	マスタ interfaces ファイルの作成	57
	dsedit または dscp を使用してマスタ interfaces ファイルを 作成する	57
	テキスト・エディタを使用したマスタ interfaces ファイルの作成	58
	複数のネットワークで使用する interfaces ファイルの設定	58
	複数のネットワーク・ハンドラ用にサーバを設定する	59
	クライアント接続の設定	59
	クエリ・ポート・バックアップの設定	61
	IPv6 のサポート	62
	IPv6 のインフラストラクチャ	63
	トラブルシューティング	65
	サーバが起動しない	65
	ESP 実行時のエラー	66
第 6 章	ディレクトリ・サービスとしての LDAP (Lightweight Directory Access Protocol) の使用	67
	概要	67
	LDAP ディレクトリ・サービスと Sybase interfaces ファイルの違い	68
	libtcl*.cfg ファイル	71
	LDAP ディレクトリ・サービスの有効化	72
	ディレクトリ・サービスへのサーバの追加	73
	複数のディレクトリ・サービス	75
	パスワードの暗号化	75
	パフォーマンス	76
	interfaces ファイルから LDAP へのマイグレート	77
第 7 章	Adaptive Server のローカライゼーションのカスタマイズ	79
	ローカライゼーション・サポートの概要	79
	言語モジュール	80
	サーバのデフォルトの文字セット	80
	サポートされている文字セット	82
	文字セット変換	86
	クライアント／サーバ間の変換	87
	ソート順	87
	利用できるソート順	88
	言語モジュール	90
	新しい言語モジュールのインストール	90
	メッセージ言語	90
	ローカライゼーション	91
	ローカライゼーションのディレクトリ	91
	ディレクトリについて	91
	charsets ディレクトリについて	92
	locales.dat ファイルについて	92

	ローカライゼーション設定の変更	94
	Adaptive Server のローカライゼーション	94
	Backup Server のローカライゼーション	95
	ソート順	97
	文字セット	98
	charset ユーティリティ	100
第 8 章	エラー・メッセージのロギングとイベントのロギング	101
	Adaptive Server エラー・ロギング	101
	エラー・ロギングの有効化と無効化	102
	エラー・ログのパスの設定	102
	Adaptive Server のエラー・ログのパス設定	102
	メッセージの管理	103
	ユーザ定義メッセージのロギング	103
	監査イベントのロギング	104
第 9 章	Adaptive Server データベースの管理	105
	データベース・デバイスの管理	105
	デバイスの要件	105
	データベース・デバイス用のファイルの作成	106
第 10 章	Adaptive Server へのオプション機能の追加	109
	監査の追加	109
	監査システムのデバイスとデータベース	109
	監査インストール環境の概要	110
	監査デバイスのインストール前の作業	110
	監査のインストール	111
	Transact-SQL 構文のオンライン・ヘルプのインストール	116
	オンライン構文ヘルプ：sp_syntax	116
	sybsyntax データベースのデフォルト・デバイス	117
	sybsyntax のインストール	118
	索引	121

概要

Sybase[®] Adaptive Server[®] Enterprise for UNIX 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

注意 このマニュアルは、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記述されています。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。

この章では、Adaptive Server の設定方法とカスタマイズの手順を説明します。

トピック名	ページ
Adaptive Server について	2
システム固有の問題	2
システムのユーザの役割	3
環境変数	3
Adaptive Server のデバイスとシステム・データベース	5
クライアント／サーバ通信	11
Adaptive Server の設定の変更	13
英語以外の言語	13
Adaptive Server の仕様	14

Adaptive Server について

Adaptive Server 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 (拡張ストアド・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。

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。

- 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* ファイルのエントリ構造
- 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

システム固有の問題については、このマニュアルで説明します。システム固有の問題の詳細については、使用しているプラットフォームの『インストール・ガイド』と『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用するログイン名は “sa” です。“sa” は UNIX のログインではありません。“sa” 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。このログインをおすすめしますが、必須ではありません。

環境変数

Sybase 製品を操作する上では、環境変数が正しく設定されていることが非常に重要です。

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、またはユーザの `.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。このマニュアルのインストール手順では、これらの環境変数をいつ設定するかについて説明します。

注意 インストーラは、インストール作業の一環として、`SYBASE.csh` および `SYBASE.sh` ファイルにあるこれらの環境変数を設定します。このファイルを基にして、これらの環境を設定することもできます。

- `DSLISITEN` – 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。`DSLISITEN` を設定せず、起動時にも Adaptive Server に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、*installed_components/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。

```
$SYBASE/$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.
```

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ために使用され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。

Adaptive Server には、次のデバイスが必要です。

- マスタ・デバイス – システム・データベースを格納する。
- `sybsystemdb` デバイス – 分散トランザクションに関する情報を格納する。
- `sysprocsdev` デバイス – システム・プロシージャを格納する。

マスタ・デバイス、`sybsystemdb` デバイス、`sysprocsdev` デバイスは、新しい Adaptive Server の作成時に作成されます。

マスタ・デバイス

マスタ・デバイスには、次のデータベースが入っています。

- `master` – Adaptive Server 全体のオペレーションを制御し、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` データベースは全部がマスタ・デバイスに入っていて、他のデバイスに拡張することはできません。
- `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。Adaptive Server では複数の `tempdb` がサポートされます。詳細については、『Transact-SQL ユーザーズ・ガイド』の「データベースおよびテーブルの作成」の章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 `pubs 3` サンプル・データベース – インストール時に `master` データベースに格納されます。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。

sybssystemdb デバイス

新規にインストールする場合は、マスタ・デバイスに **sybssystemdb** データベースも含まれています。**sybssystemdb** デバイスには、**sybssystemdb** 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。

DTM (分散トランザクション管理) 機能をサポートするには、**sybssystemdb** データベースが必要です。インストールの前に、**sybssystemdb** 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 デバイスには、**sybssystemprocs** データベースが格納されています。このデータベースには、Sybase が提供するほとんどのシステム・プロシージャが入っています。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。たとえば、**sp_configure** などがあります。リカバリ状況下で必要になるシステム・プロシージャは、**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 **sybssystemprocs** データベースを格納するため、**sybssystemprocs** 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

以降の項で説明するデバイスとデータベースはオプションです。

PCI (Pluggable
Component Interface)
データベース

PCI (Pluggable Component Interface) により、Adaptive Server にさまざまな機能を提供するプラグ可能ライブラリを追加できます。Adaptive Server 15.0.3 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

サンプル・データベースには、次のものがあります。

- **pubs2** データベースと **pubs3** データベース – 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 – フランス語のデータとドイツ語のデータが入っています。
- **jpubs** データベース – 日本語のデータが入っています。

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「第 3 章 インストール後のタスク」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 デバイスとデータベース

sybsecurity デバイスは、監査インストール・プロセス中に作成されます。**sybsecurity** デバイスは、**sybsecurity** データベースを格納します。また、システムに監査を設定するときに使用する監査システム・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。

Adaptive Server での監査の設定については、[「第 10 章 Adaptive Server へのオプション機能の追加」](#)を参照してください。監査システムのインストールと使用方法については、『セキュリティ管理ガイド』の「監査」の章を参照してください。

dbccdb データベース

dbcc (データベース一貫性チェッカ) には、データベースの論理的、物理的一貫性をチェックするコマンドが用意されています。**dbccdb** データベースには、**dbcc checkstorage** または **dbcc checkverify** を使用したときの **dbcc** の結果が格納されます。

dbcc checkstorage は、「ターゲット・データベース」の一貫性情報、オペレーション・アクティビティ、そのオペレーションの結果を **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** オペレーションの結果についてのレポートを生成したりする **dbcc** ストアド・プロシージャが格納されます。

dbccdb のインストールと使用方法については、『システム管理ガイド 第 2 巻』の「第 25 章 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

sybmgmtdb は Job Scheduler データベースです。

ジョブ、スケジュール、スケジュールされたジョブ、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は、ストアド・プロシージャによって行われます。ストアド・プロシージャを使用することで、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。

sybmgmtdb と Job Scheduler の詳細については、『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 UNIX オペレーティング・システム・ファイルの使用

UNIX 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、次のいずれかの方法で、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。

- disk init, disk reinit, または sp_deviceattr で directio パラメータを使用する。
- disk init で dsync 設定を指定する。

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、そのデバイスの directio を "true" に設定することはできません。デバイスの directio を有効にするには、dsync の設定を "false" にリセットします。

directio パラメータの使用

disk init, disk reinit, sp_deviceattr の directio パラメータを指定することにより、オペレーティング・システムのバッファ・キャッシュを回避して、データをディスクに直接転送できます。directio は、I/O の方法やパフォーマンス向上の効果の点ではロー・デバイスと同じですが、ロー・デバイスより使いやすく、ファイル・システム・デバイスの管理が容易です。

directio オプションのデフォルト値は、すべてのプラットフォームで false (オフ) に設定されます。

directio 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』を参照してください。

dsync オプションの使用

dsync オプションによって、Adaptive Server がファイル・システム上のデバイスからデータを確実に回復できるようになります。デフォルトでは、Adaptive Server はファイル・システム・デバイスの dsync を有効にします。ただし、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。disk init コマンド、disk reinit コマンド、sp_deviceattr コマンドを使用して、dsync を設定またはリセットできます。

注意 ロー・デバイスの場合、dsync オプションは無視されます。

デフォルトでは、新しいバージョンの Adaptive Server をインストールすると、すべてのファイル・システム・デバイスに対して dync が on に設定されます。

データベースを UNIX ファイル・システム・デバイスに格納している UNIX サーバ上で、ASE 12.0 以前のバージョンからアップグレードする場合、デフォルトでは、dsync は次のように設定されます。

- マスタ・デバイスに対しては on
- その他のすべてのデバイスに対しては off

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。また、『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。

警告！ アップグレード直後に dsync オプションが設定されていない場合、データが破損することがあります。

データベース・デバイスのロケーション、タイプ、サイズの決定

Adaptive Server には、複数のデータベース・デバイスが必要です。表 1-1 に、各デバイスの値のベースラインを示します。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 1-1: 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)
マスタ	システム・データベースを格納	2K ページの場合：	30MB
		24MB	60MB
		4K ページの場合：	120MB
		45MB	240MB
		8K ページの場合：	89MB
		16K ページの場合：	177MB

デバイス	目的	最小サイズ	最小サイズ (推奨)
sysprocsdev (または sybssystemprocs デバイ スともいう)	sybssystemprocs データベースを 格納	136MB	140MB (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)
sybssystemdb	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 または model データベ ースのサイズ (い ずれか大きい方)	固有の監査要件 によって異なる。 「 監査の追加 (109 ページ) を 参照。

Solaris、HP-UX、IBM AIX、Linux の場合

すべてのデータベースに対して、ロー・パーティションまたはファイルを使用できます。

Sybase Adaptive Server はデータの格納に対して、Network Appliance 社のファイラーでの NFS マウント・デバイス、または CIFS マウント・デバイス上でのデータベース・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。NetApp ファイラーを使用するための、オペレーティング・システムおよび Sybase Adaptive Server に対する変更の必要はありません。

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。

すべてのプラットフォームの場合

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。

次のガイドラインに従って、ロー・パーティション・デバイスを準備します。

- Sybase インストール・ソフトウェアが含ま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・システムの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。「[データベース・デバイス用のファイルの作成](#)
(106 ページ) を参照してください。
- パフォーマンスを最適にするには、Sybase ソフトウェアとマスタ・デバイ
スを含むすべてのディスク・デバイスを同じマシン上に置く。
- 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

- 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、[表 1-1 \(9 ページ\)](#) の推奨
サイズに基づいて各デバイスのロー・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が、
“sybase” ユーザにあるか確認します。

注意 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。

クライアント／サーバ通信

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。クライアントは 1 つ以上のサーバと通信でき、サーバはリモート・プロ
シージャ・コールによって別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。

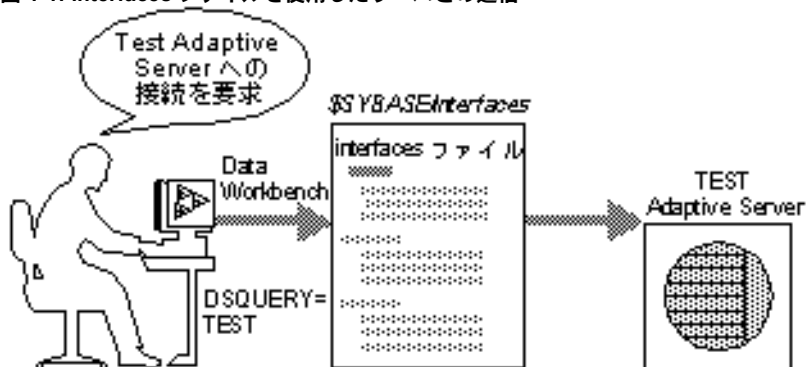
- interfaces ファイルに格納する。このファイルは、UNIX プラットフォームで *interfaces* という名前が付けられ、*\$SYBASE* インストール・ディレクトリに置かれます。
- LDAP サーバに格納する。

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・プログラムから特定のサーバに接続する場合、クライアント・プログラムは図 1-1 に示すように、ディレクトリ・サービスでそのサーバ名を検索して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 1-1: interfaces ファイルを使用したサーバとの通信



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、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。既存のサーバの新しい *interfaces* ファイル・エントリを作成する場合は、「第 6 章 ディレクトリ・サービスとしての LDAP (Lightweight Directory Access Protocol) の使用」を参照してください。

Adaptive Server の設定の変更

`sp_configure` を使用して、Adaptive Server の設定を変更します。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。

言語、文字セット、ソート順の設定については、『第 7 章 Adaptive Server のローカライゼーションのカスタマイズ』を参照してください。

Adaptive Server で高可用性機能を使用する設定については、『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。

Adaptive Server で分散トランザクション管理 (2 フェーズ・コミット) を実行するように設定する方法については、『システム管理ガイド 第 2 巻』を参照してください。

英語以外の言語

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

データベースの仕様

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 – 4TB • 4K ページ・サーバ – 8TB • 8K ページ・サーバ – 16TB • 16K ページ・サーバ – 32TB 	
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最大 4TB 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最大 4TB となる。
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		
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 = テーブル内のノンクラスタード・インデックスのすべての領域の合計、 および、ロギング用に 20% のオーバヘッド	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	

クエリの仕様

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 の各項には最大 50 個のユーザ・テーブルと 14 個のワーク・テーブル、union のすべての項には合計で最大 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るデータベース、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。
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	

プロシージャの仕様

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	

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のサイズによって異なります。表 1-2 に、APL (全ページ・ロック) テーブルのカラムとローの制限を示します。

表 1-2: APL (全ページロック) テーブル

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

表 1-3 に、DOL (データオンリーロック) テーブルのカラムとローの制限を示します。

表 1-3: データ・ロー・テーブルとデータ・ページ・テーブル

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

データベースの必要領域は、サーバの論理ページの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 と等しくなります。表 1-4 は、データベースごとの最小サイズを示します。

表 1-4: ページ・サイズによるデータベース要件

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
model データベース	3MB	6MB	12MB	24MB
tempdb データベース	4MB	6MB	12MB	24MB
sybssystemdb データベース	3MB	6MB	12MB	24MB
sybpcidb データベース	24MB	48MB	96MB	192MB

論理ページのサイズが大きくなると、格納できるデータ量も増えます。表 1-5 に、論理ページ・サイズごとの最大データ量を示します。

表 1-5: ページ・サイズによるテーブルのデータ数の制限

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

サーバの起動と停止

この章では、Adaptive Server、Backup Server、XP Server の起動と停止の方法について説明します。

トピック名	ページ
概要	17
サーバの起動	18
オペレーティング・システムの再起動時にサーバを起動する	20
最初のインストール後の XP Server の起動	23
サーバの停止	23

概要

この章で説明する方法は、オペレーティング・システムのクラッシュなどが原因でデータベースを管理するために停止したあとに、Adaptive Server と Backup Server を起動するために使用します。

XP Server はインストール処理では起動されません。XP コマンドが `isql` を使用して発行されたときにだけ、XP Server は 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。

サーバの起動要件

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 の配布ファイルへアクセスできる。
- 「第 1 章 概要」で説明している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。詳細については、『Sybase ソフトウェア資産管理ユーザーズ・ガイド』を参照。

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

`RUN_server_name` ファイルと `startserver` コマンドを使用して、コマンド・ラインから Adaptive Server または Backup Server を起動できます。起動オプションをカスタマイズできます。

`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。

サーバの起動パラメータ

デフォルトのサーバ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。

Backup Server のサーバ名には “_back” の文字列が付加されます。

表 2-1 に、Adaptive Server のデフォルトの起動パラメータを示します。

表 2-1: デフォルトの Adaptive Server 起動パラメータ

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

起動パラメータの変更

`$$SYBASE/$SYBASE_ASE/install/RUN_server_name` ファイルを直接編集しないかぎり、デフォルトの起動パラメータは変更できません。

`$$SYBASE/$SYBASE_ASE/install/RUN_server_name` ファイル内で、追加の起動パラメータも指定できます。

『ASE ユーティリティ・ガイド』の `dataserver` と `backupserver` の説明でリスト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* ファイルの使用**

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 では、*RUN_server_name* ファイルの名前は *RUN_servername* です。ここで、*servername* にはサーバの名前が入ります。

たとえば、Adaptive Server の名前が TEST の場合、その *RUN_server_name* ファイルの名前は *RUN_TEST* になります。

Backup Server の場合、*RUN_server_name* ファイルは *RUN_servername_back* という名前になります。ここで、*servername* にはサーバの名前が入ります。

警告！ `$$SYBASE/$$SYBASE_ASE/install` に作成された *RUN_server_name* 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、サーバを再起動するためにこのファイルが使用されます。別のロケーションに *RUN_server_name* ファイルが必要な場合、元の *RUN_server_name* 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、次のように入力します。

```
$$SYBASE/$$SYBASE_ASE/install/startserver [ -f RUN_server_name file ]
```

ここで、`$$SYBASE/$$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_servername`) 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` を使用して Adaptive Server を起動するには、マスタ・デバイス上での読み込み/書き込みパーミッションを持っている必要があります。`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にサーバを起動する

この項では、Adaptive Server および Backup Server が自動的に再起動するようにオペレーティング・システムを設定する方法について説明します。

運用システムでは、UNIX オペレーティング・システムが起動するときに、Adaptive Server と Backup Server を自動的に再起動してください。このように設定するには、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か、*SYBASE.csh* または *SYBASE.sh* を source コマンドを使用して実行する必要があります。

HP-UX の場合

HP-UX のバージョン 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。このためシステム管理者は、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されるスクリプトを作成する必要があります。

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。HP-UX テンプレート・ファイル */sbin/init.d/template* をもとにしてスクリプト・ファイルを作成できます。

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。ここで、*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するために使用されます。

IBM RS/6000 の場合

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。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"
```

ここで、*release_directory* は Sybase インストール・ディレクトリ (SYBASE 環境変数として指定されている) へのフル・パスで、*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* のあらゆるエントリより後になるようにします。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。

ネットワーク・オペレーションの起動が遅い場合、*/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、*/etc/inittab* に Backup Server の起動コマンドを追加します。このコマンドには、Backup Server の *RUN_server_name* ファイルのパスを使用します。

Sun Solaris と Linux の場合

ネットワーク・リソースがすべて使用可能であることを確認してから、Adaptive Server または Backup Server が自動的に起動するように設定します。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。*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の後にあることを確認します。ネットワーク・オペレーションの起動が遅い場合、*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。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。

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・システムを設定するには、次の手順に従います。

- 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
```

ここで、*\$\$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` ファイルの内容を見て、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を指定します。通常、Sun Solaris では 2 または 3、Linux では 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。ここで、`#` は手順 2 で取得したデフォルトの実行レベルです。

Sun Solaris では次のような構文を使用します。

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

Linux の場合：

`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。ここで、`#` 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
```

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

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 `syssservers` テーブルに自動的に追加されます。XP Server を Adaptive Server インストールとは別の構築セッションでインストールする場合は、XP Server のインストール・プロセス中に、関連する Adaptive Server の名前、システム管理者の名前、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・プロセスで上記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 `syssservers` 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` には Adaptive Server 名を指定します。

サーバの停止

サーバを停止するコマンドを発行するパーミッションを持つのはシステム管理者だけです。このコマンドを使用すれば、サーバの再起動時に自動リカバリで必要となる作業量が最小になります。

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 を停止するには、次の手順に従います。

- 1 `isql` を使用して、システム管理者権限を持つ Adaptive Server アカウントにログインします。

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

- 2 次のコマンドを入力して、サーバを停止します。

```
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を終了し、データベースごとにチェックポイントを実行し、新しいログインを無効にする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、また各データベース内のチェックポイントも実行しません。

注意 `shutdown with nowait` コマンドの使用はおすすめしません。このコマンドは、必要な場合以外は使用しないでください。

Backup Server の停止

Backup Server を停止するには、次の手順に従います。

- 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 次のコマンドを入力して、サーバを停止します。

```
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` コマンドの発行によって、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、また各データベース内のチェックポイントも実行しません。

Backup Server に `shutdown with nowait` コマンドを使用すると、不整合または不完全なダンプやロードが発生する可能性があります。このコマンドは必要な場合以外は使用しないでください。

`shutdown` コマンドの詳細については、『リファレンス・マニュアル：コマンド』を参照してください。

kill コマンドの使用

警告！ Adaptive Server と Backup Server では、`kill` コマンドは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を終了するので、この方法は使用しないでください。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
```

ここで、`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、4 つのエンジンを使用しているサーバのプロセスを示します。

```
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 が起動すると、`$$SYBASE/$$SYBASE_ASE` ディレクトリに `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 `MAXSHMSEGSIZE` 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。作成する追加共有メモリ・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 がクラッシュしたり、`kill -9` コマンドを使用して停止された場合、これらのファイルは削除されません。Adaptive Server がクラッシュしたり、`kill -9` コマンドを使用して停止された後で 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、Adaptive Server が以前に作成された共有メモリ・ファイルを上書きできる必要があるためです。

Adaptive Server または Backup Server に異常が発生して終了した場合も、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” になっているこれらの共有メモリ・セグメントを識別し、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。

Linux のヒュージ・ページ

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。これらのキャッシュ・ミスは非常に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。

注意 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。

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)のサイズが減ります。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・ページが既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、共有メモリにのみ使用されます。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るようにしてください。

オペレーティング・システムの設定

この章では、Adaptive Server をインストールまたはアップグレードした後で調整できるオペレーティング・システムの設定について説明します。特に明記されていないかぎり、この章の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。Adaptive Server 15.0.3 以降のインストーラでは、インストール・プロセス中に Adaptive Server の一部の設定を行うことができます。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。

トピック名	ページ
環境変数の確認	29
srvbuild または srvbuildres を使用した新しいサーバの設定	30
stty 設定の使用	33
正しいパーミッションのリストア	33
ファイル記述子とユーザ接続	34
非同期ディスク I/O の有効化	37
クライアント接続のタイムアウト時間の調整	40
ハードウェア・エラーのチェック	41
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リング	42
データベースの整合性の検査	44

環境変数の確認

「[環境変数](#)」(3 ページ) に示す Adaptive Server の環境変数が正しく設定されているかどうかを確認してください。

環境変数の現在の値を確認するには、オペレーティング・システムのプロンプトで次のコマンドを入力します。

```
env
```

srvbuild または srvbuildres 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。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。`srvbuildres` の詳細については、『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」の章を参照してください。

`srvbuild` を使用して新しいサーバを設定するには、次の手順に従います。

- 1 環境変数を設定するには、`source` コマンドを使用して、`$$SYBASE` の `SYBASE.csh` ファイルまたは `SYBASE.sh` ファイルを実行します。
- 2 `$$SYBASE/$SYBASE_ASE/bin/srvbuild` を実行します。
- 3 [Select Servers to Build] ウィンドウが表示されます。左側のボックスをクリックして、設定するサーバを選択します。各ボックスを選択すると、サーバ名を入力可能なサーバ名テキスト・フィールドが有効になります。

注意 選択可能な「サーバ・タイプ」のリストは、`$$SYBASE` に何をインストールしたかによって異なります。

- 4 構築するサーバを選択したら、[OK] をクリックします。設定時に選択した内容に応じて、情報が次の画面に表示されます。
- 5 [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 以降のバージョンでは、指定した個々のデバイス上で **tempdb** が作成される。**tempdb** デバイスとデータベースのデフォルト・サイズは 100MB。**tempdb** の次の属性を指定する。
 - **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、次の項目も設定する。

- **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る。

- Adaptive Server で使用可能な物理メモリ
- Adaptive Server で使用可能な CPU

設定ツールを使用して新しいサーバを作成した場合、ツールで設定されたデフォルト設定ではシステム上で使用できるすべてのリソースを最適に使用できないことがあります。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 Adaptive Server 15.0.3 インストーラを使用すると、基本的な設定の調整を、インストール後の作業としてではなく、インストール中に行うことができます。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。

- 6 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
- 7 [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
- 8 Job Scheduler を設定するには、[Job Scheduler type] 画面に次の情報を入力します。
 - 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
- 9 [Self Management の設定] 画面に、次の内容を入力します。
- 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 を設定した後に Self Management のユーザ・パスワードを変更する場合、次のコマンドを再実行する必要があります。

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

- 10 [サーバをビルドしてください!] をクリックして、サーバの設定を続行します。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。

stty 設定の使用

`stty tostop` オプションを設定すると、バックグラウンドの Adaptive Server は、端末への書き込みを試行すると同時に停止します。このエラーを回避するには、次のコマンドを実行してから Adaptive Server を起動します。

```
stty -tostop
```

Adaptive Server の出力をすべてファイルにリダイレクトする場合、`stty` の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。パーミッションが正しくないことに気づいた場合、`SYBASE/SYBASE_ASE/install` ディレクトリに保存されているスクリプト `setperm_all` 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 つのプロセスあたり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 の場合

カーネル・パラメータの `maxfiles` と `maxfiles_lim` が、任意の 1 プロセスで使用可能なファイル記述子の数を制御します。HP-UX での制限は、32 ビット・システムで 10,000、64 ビット・システムで 60,000 です。

現在のファイル記述子の値を取得するには、`ulimit -n` のように Korn シェルまたは Bourne シェルの `ulimit` コマンドを使用します。

AIX の場合

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 1 つのエンジンで最大 2000 のファイル記述子を使用できます。`open_max` の設定方法の詳細については、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` パラメータの値を取得するには、次のように Korn シェルまたは Bourne シェルの `ulimit` コマンドを使用します。

```
ulimit -n
```

Linux の場合

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。`ulimit` を使用してファイル記述子の数を設定できます。

Sun Solaris の場合

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せんが、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 エンジン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。

ユーザ接続の詳細については、『システム管理ガイド』を参照してください。

現在のソフト制限値とハード制限値の表示

現在のソフト制限値を表示するには、C シェルの場合、次のように入力します。

```
limit descriptors
```

Bourne シェルの場合、次のように入力します。

```
ulimit -n
```

現在のハード制限値を表示するには、C シェルの場合次のように入力します。

```
limit -h descriptors
```

Bourne シェルの場合、次のように入力します。

```
ulimit -Hn
```

ソフト制限値を増やす方法

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
```

Bourne シェルの場合、次のように入力します。

```
ulimit -Sn new_value
```

ここで *n* はソフト制限値の現在値で、*new_value* は増加後のソフト制限値を示します。

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

ハード制限値を増やすには、「[サンプル・プログラム](#)」(36 ページ) の例で示すようなプログラムを使用します。

❖ サンプル・プログラムを設定してハード制限値を増やす

- 1 ASCII テキスト・エディタを使用して、`file_name.c` (`file_name` にはファイルの名前を指定する) を作成します。「[サンプル・プログラム](#)」(36 ページ) の例に示すテキストを入力します。

- 2 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
```

ここで `file_name` は作成したソース・ファイルの名前、`program_name` はプログラムに付ける名前です。

- 3 プログラムのパーミッションと所有権を変更して、“root” 権限で実行されるようにします。

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

ここで `program_name` は、コンパイルしたプログラムの名前です。

- 4 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、“root” ユーザはこのプログラムを使用して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
```

ここで `program_name` はコンパイルしたプログラムの名前、`master_device_name` は Adaptive Server のマスタ・デバイスへのフル・パスです。オペレーティング・システムのプロンプトでコマンドを入力する代わりに、Adaptive Server の `RUN_server_name` ファイル内で `dataserver` コマンドラインの先頭に `program_name` を挿入することもできます。

サンプル・プログラム

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。

次の例は、ハード制限値を増やす場合に使用できるソース・コードを示します。

```
#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);
}
```

ユーザ接続の詳細については、『システム管理ガイド』を参照してください。

非同期ディスク I/O の有効化

UNIX システムおよび Linux システム上では、ブロック・デバイスを使用しないことをおすすめします。データベース・デバイスとしてブロック・デバイスを使用した場合、システム・クラッシュ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`
- `sp_deviceattr...directio` または `dsync` パラメータを `true`

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

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から、Adaptive Server を起動してください。

Linux では、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、すべてのプロセスによって予約されている現在の記述子の数 (aio-nr) を減算した値が、max online engines の値に max async i/os per engine の値を乗算した値以上である必要があります。

aio-max-nr の値と aio-nr の値を確認するには、次のように入力します。

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

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 を設定してください。

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 KAIO 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「max async i/os per engine」を参照してください。

ファイル・システム・
デバイスを使用する
HP-UX 11.31 以降

❖ **ファイル・システム・デバイスを使用して動作する HP-UX で非同期 I/O を有効にする**

下記の OS パラメータを最大値に設定してから、ファイル・システム・デバイスの非同期 I/O を有効にすることをおすすめします。

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

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。

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

3 このパラメータは静的であるため、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・システムを使用して動作する HP-UX で同時実行 I/O を有効にする**

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。

- 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。

ロー・パーティションを使用する HP-UX

HP-UX ユーザは、非同期 I/O を必ず有効にする必要があります。

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。ドライバのインストールについてのヘルプ情報は、オペレーティング・システム管理者または HP のテクニカル・サポートにお問い合わせください。

注意 Adaptive Server (または SQL Server) を停止してから、以下の指示に従って実行してください。

非同期 I/O を有効にするには、次の手順に従います。

- 1 SAM の [Kernel Configuration] メニューから、[Drivers] を選択し、*asyncdisk* の [Pending State] を [In] と設定して、ドライバを追加します。

または、*asyncdsk* サブシステム・キーワードを */stand/system* に追加することもできます。

- 2 カーネルを再構築して、システムをリブートします。
- 3 **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
```

ここで、

<uid> は、Adaptive Server をブートするユーザが使用するユーザ ID です。

<ugrp> は、ユーザ ID <uid> のユーザ・グループです。

- 4 UNIX プロンプトで、以下の文を “root” 権限で実行します。Adaptive Server と Backup Server を起動しているユーザのユーザ ID は、*/dev/async* ディレクタリの所有者でなければなりません。

IBM AIX の場合

非同期ディスク I/O を有効にします。

この手順は、IBM のユーザには必須です。

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 [Asynchronous I/O] の [Change/Show Characteristics] を選択します。

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、アクティブではなくなった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。現在のタイムアウト時間の値を表示するには、後の項で説明する、それぞれのプラットフォーム用のコマンドを使用します。

HP-UX 11 以降の場合

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。

```
/nnd -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。

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 **nnd -set** コマンドを使用します。

IBM RS/6000 の場合

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_keeptime
```

`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 分以上の値を推奨しています。

Sun Solaris の場合

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

タイムアウト時間を 15 分 (900,000 ミリ秒) に短縮するには、次のコマンドを入力します。

```
/usr/sbin/ndd -set /dev/tcp tcp_keepalive_interval 900000
```

Linux の場合

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。

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

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

データベースの破壊につながる可能性のある問題を示すハードウェア・エラー・メッセージには、次のような種類があります。

- 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 사용합니다。エラーが検出された場合は、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。この項の最初で説明した種類のハードウェア・エラーが検出された場合は、Sun Microsystems の診断ツール `sundiag` 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リング

『システム管理ガイド 第2巻』では、負荷とシステムの設定に対応して最適な数の Adaptive Server エンジン进行管理する方法が説明されています。最適な数を決めるには、システムと CPU の使用率をモニタします。

HP-UX の場合

HP-UX では、パフォーマンスをモニタするために多くのツールが用意されています。その一部について次に説明します。

- **sar** コマンド – 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** コマンド – 仮想メモリの使用状況をモニタする。
- **netstat** コマンド – ネットワーク・ステータスをモニタする。
- **ps** コマンド – 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** コマンド – 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 の場合

IBM RS/6000 では、パフォーマンスをモニタするために次のツールが用意されています。

- **iostat** コマンド – 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** コマンド – 仮想メモリの使用状況をモニタする。
- **netstat** コマンド – ネットワーク・ステータスをモニタする。
 - **netstat -v** で、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** コマンドは、現在のネットワーク・オプションを表示する。また、mbuf プールのチューニングにも使用する。
- **ps** コマンド – 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** コマンド – 実行が完了するまで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 の場合

Sun Solaris と Linux では、パフォーマンスをモニタするために次のツールが用意されています。

- **iotstat** コマンド – 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** コマンド – 仮想メモリの使用状況をモニタする。
- **netstat** コマンド – ネットワーク・ステータスをモニタする。
- **ps** コマンド – 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** コマンド – 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。

データベースの整合性の検査

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 データベースの整合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** では一貫性の検査が短時間で実行されますが、**dbcc checkalloc** または **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「第10章 データベースの一貫性の検査」の「**dbcc checkstorage** を使用するための準備」、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、この作業を行うために複数の **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
    end
end
```

```

    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* と呼ばれる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` データベースをチェックします。

```

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

```

次に、C シェル・スクリプトは `grep` コマンドを実行して、`dbcc` 出力にある 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* と呼ばれる出力ファイルに送られます。

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。

前述のサンプルのシェル・スクリプトと `isql` スクリプトは、インストール環境での必要性に合わせてカスタマイズできます。

このスクリプトを自動的に実行させるには、`crontab` ファイルを編集して、次のようなエントリを追加します。

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

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

Adaptive Server のデフォルト設定

Adaptive Server のインストールまたはアップグレードには、パラメータのデフォルト設定と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』と『システム管理ガイド 第 2 巻』を参照してください。

言語、文字セット、ソート順、オプション機能などの設定の詳細については、『システム管理ガイド 第 1 巻』を参照してください。

トピック名	ページ
デフォルト設定	47

デフォルト設定

表 4-1 に、Adaptive Server をインストールした後のデフォルト設定を示します。使用するコンピュータやデータベースに応じてこれらの設定を変更する必要がある場合もあります。

表 4-1: 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>Servername</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/servername.log</i>
イベント・ロギング	設定なし
国際化サポート (ローカライゼーション)	
• 言語	us_english
• 文字セット	HP = Roman8 IBM = ISO 8859-1 Sun = ISO 8859-1 Linux = iso_1
• 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 4-2 は、Backup Server と XP Server のデフォルト設定を示します。これらのサーバの詳細については、「概要」(1 ページ)を参照してください。

表 4-2: 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	<i>\$\$SYBASE/\$\$SYBASE_ASE/install/AdaptiveServername_back.log</i>
XP Server	名前	<i>\$\$SYBASE/\$\$SYBASE_ASE/ADAPTIVESERVERNAME_XP</i>
	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

ネットワークを介する通信の設定

Adaptive Server はネットワークを介して、ほかの 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。

トピック名	ページ
Adaptive Server で使用するディレクトリ・サービス・エントリの決定方法	50
クライアントのディレクトリ・サービスの使用方法	51
ディレクトリ・サービス・エントリの作成	51
サポートされているディレクトリ・ドライバ	52
interfaces ファイルの内容	52
異機種間環境と同機種間環境	53
interfaces ファイルのフォーマットについて	54
マスタ interfaces ファイルの作成	57
複数のネットワークで使用する interfaces ファイルの設定	58
IPv6 のサポート	62
トラブルシューティング	65

ディレクトリ・サービスには、サーバのネットワーク・ロケーションについての情報が入っています。ディレクトリ・サービスには、Adaptive Server、Backup Server、ネットワーク上にある他のサーバ製品のエントリがすべて入っています。

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。

ディレクトリ・サービスには、Backup Server と XP Server を含むすべてのサーバの名前とアドレスがリストされています。クライアント・プログラムを使用して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先を決定します。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。

表 5-1 に、サーバとクライアントの interfaces ファイルのタスクとトピックについての詳細が説明されている箇所を示します。

表 5-1: interfaces ファイルのタスクとトピックの参照先

interfaces ファイルの種類	タスクまたはトピック	参照先
UNIX のサーバまたはクライアント	複数の Adaptive Server インストール環境用のエントリを追加する	「第 7 章 Adaptive Server のローカライゼーションのカスタマイズ」
	複数のインストール環境で使用するマスタ interfaces ファイルの作成	「マスタ interfaces ファイルの作成」(57 ページ)
	複数のネットワーク用の設定	「複数のネットワークで使用する interfaces ファイルの設定」(58 ページ)
	リファレンス情報	「interfaces ファイルのフォーマットについて」(54 ページ)
PC クライアント	クライアントの設定	使用しているプラットフォームの『ASE インストール・ガイド』
	上級タスクについてのリファレンス情報と指示	使用している PC クライアント・プラットフォームの『Open Client/Server プログラマーズ・ガイド補足』または適切な Open Client のマニュアル
	リストされていないクライアント・プラットフォーム	使用している PC クライアント・プラットフォームの『Open Client/Server プログラマーズ・ガイド補足』または適切な Open Client のマニュアル

Adaptive Server で使用するディレクトリ・サービス・エントリの決定方法

Adaptive Server は、ディレクトリ・サービスを使用してクライアントから受信するアドレスを決定します。Adaptive Server の起動時に、次の手順が実行されます。

- 1 コマンド・ラインで **-s** オプションに指定されたサーバ名を調べます。コマンド・ラインでサーバ名が指定されていない場合は、次の処理が行われます。
- 2 DSLISTEN 環境変数の値を確認して、自身の名前を決定します。DSLISTEN 環境変数が設定されていない場合、サーバ名は SYBASE と見なされます。
- 3 上記の手順で見つけた名前と一致するエントリ名をディレクトリ・サービス内で検索します。
- 4 検出したディレクトリ・サービス・エントリに指定されているネットワーク情報を、クライアント接続の受信に使用します。

クライアントのディレクトリ・サービスの使用方法

クライアントがサーバに接続するとき、次の処理を実行します。

- プログラムを通して、または DSQUERY 環境変数を参照して、サーバの名前を決定する。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名を検索する。
- 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。

クライアント接続については、Open Client のマニュアルで詳細に説明しています。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』または適切な 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・エントリの作成

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。また、次の Sybase ユーティリティを使用してディレクトリ・サービスのネットワーク情報を編集することもできます。

- `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

次の3つのディレクトリ・ドライバがサポートされています。

- `interfaces` ドライバ
- ライトウェイト・ディレクトリ・サービス・ドライバ (LDAP)
- DCE (Distributed Computing Environment) で提供される CDS (Cell Directory Service)

この章の残りの部分では、`interfaces` ファイルについて説明し、サポートする UNIX プラットフォームごとに固有の設定情報を記述します。LDAP ドライバと Cell Directory Service の詳細、`interfaces` ファイルと LDAP ディレクトリ・サービスの比較については、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、ネットワーク上で動作しているすべてのサーバについてのネットワーク情報が入っています。これらのサーバには、Adaptive Server、Backup Server、XP Server に加えて Replication Server や Open Server などのサーバ・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) で構成されています。`interfaces` ファイルのエントリの具体的な構成については、「[interfaces ファイルのフォーマットについて](#)」(54 ページ) を参照してください。

`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・アプリケーションによって、ネットワークを介してクエリを受信するときに使用される。この情報は「リスナ・サービス」と呼ばれる。
- `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する。この情報は「クエリ・サービス」と呼ばれる。

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から接続要求を受信するため、サーバの `master` 行と `query` 行に指定されているネットワーク情報は同一です。

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` ファイルでは `master` 行と `query` 行の両方を指定する必要があります。

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つ作成して、これを各マシンにコピーできます。詳細については、「マスタ `interfaces` ファイルの作成」(57 ページ) を参照してください。

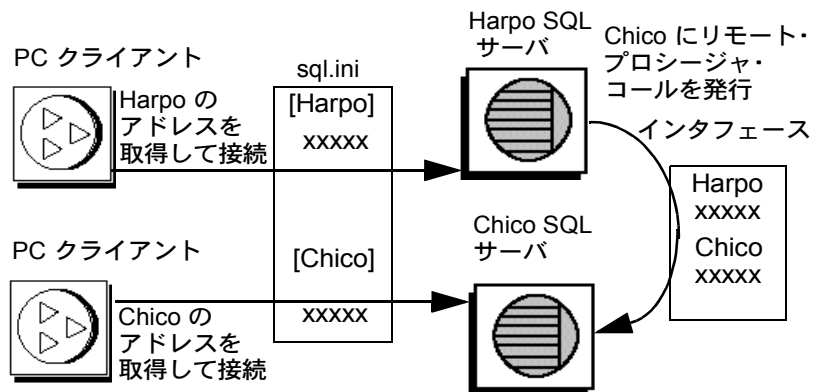
ホスト・マシンが複数のネットワークをサポートしている場合は、「複数のネットワークで使用する `interfaces` ファイルの設定」(58 ページ) を参照してください。

異機種間環境と同機種間環境

Adaptive Server とクライアントを同一のプラットフォームまたは異なるプラットフォームで実行できます。

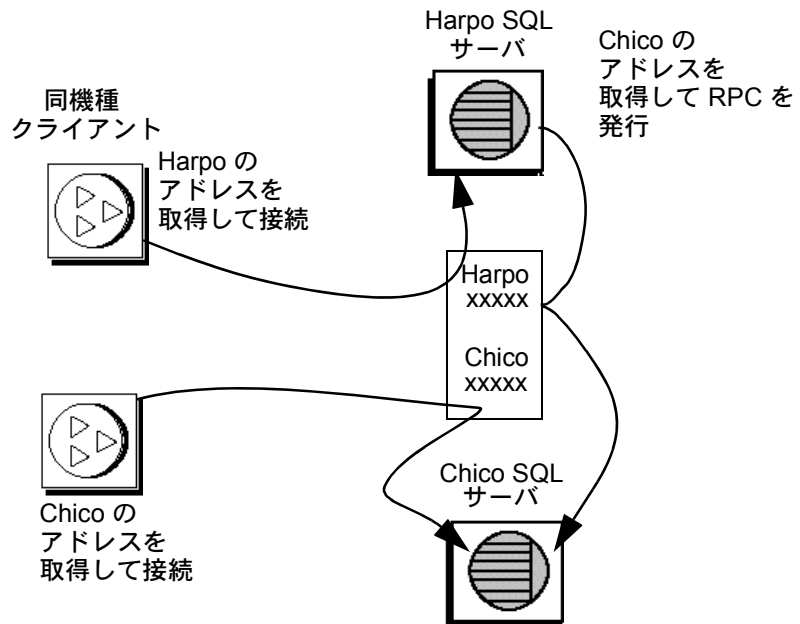
プラットフォームが異なる場合、各プラットフォームが `interfaces` ファイルに対して異なるフォーマットと異なる設定を要求する場合があります。図 5-1 は、クライアント PC が `interfaces` ファイル (`sql.ini`) 内のネットワーク情報を使用して UNIX 上で実行されている Adaptive Server に接続する方法と、Adaptive Server が `interfaces` ファイルを使用してリモート・プロシージャ・コールの実行中にほかのサーバに接続する方法を示します。

図 5-1: 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 上で実行されている場合、同じ `interfaces` ファイルが両方に対して有効です。図 5-2 は、同機種間環境で実行されているクライアントと Adaptive Server が1つの `interfaces` ファイルのコピーを使用して接続を確立する方法を示します。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 5-2: 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について

次の規則が、interfaces ファイルのエントリのフォーマットに適用されます。

- 各 Adaptive Server に対してエントリは 1 つしかないが、エントリ内には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。

TLI スタイルのエントリのフォーマットは次のとおりです。

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

次に例を示します。

```
harpo_server1
    master tli tcp /dev/tcp 0x000233337f00001
    query tli tcp /dev/tcp 0x000233337f00001
```

TCP スタイルのエントリのフォーマットは次のとおりです。

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

次に例を示します。

```
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

表 5-2 は、interfaces ファイルのエントリの要素を示します。

表 5-2: interfaces ファイルの要素

要素	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 秒。
<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 秒。
<i>service_type</i>	エントリによって定義されるサービスの種類。次のいずれかにする。 <ul style="list-style-type: none"> • マスタ • クエリ
<i>api</i>	ネットワークで使用できるアプリケーション・プログラミング・インタフェース。サポートされている値は tli。
<i>protocol</i>	ネットワーク・プロトコルの名前。使用できるプロトコル・タイプは次のとおり。 <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” が書き込まれる。

要素	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。 <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。</p> <p style="text-align: center;">/bin/hostname</p>
<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。</p> <p style="text-align: center;">/bin/hostname</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。</p> <p>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 プロトコルのエントリ用の <i>address</i>	<p>アドレスは、次の要素から構成される。</p> <ul style="list-style-type: none"> TLI のアドレス・プレフィクスは “<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 から 65535 の間のユニークな数にする。ネットワーク上の各マシンで <code>/etc/services</code> ファイルを確認して、使用中のポート番号を調べる。<code>/etc/services</code> の新しいセクションに “Sybase specific services” とラベルを付けて、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が指定されていれば、他のユーザがその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。
<i>port</i>	<p>1025 から 65535 の間のユニークなポート番号。ネットワーク上の各マシンの <code>/etc/services</code> ファイルを確認して、使用中のポート番号を調べる。<code>/etc/services</code> の新しいセクションに “Sybase specific services” とラベルを付けて、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が指定されていれば、他のユーザがその番号を使用するのを防ぐことができる。</p>
<i>ha_failover</i>	<p>ディレクトリ・サービスまたは <code>interfaces</code> ファイルに作成される高可用性のエントリ。</p>
フィルタ	<p>Adaptive Server は、ディレクトリ・サービスの <code>master</code> 行と <code>query</code> 行に追加するフィルタとして SSL (Secure Socket Layers) プロトコルをサポートする。SSL は機密情報を安全に送信するための標準。</p>

マスタ *interfaces* ファイルの作成

マスタ *interfaces* ファイルには、ネットワーク上にあるすべての Sybase サーバ用のエントリが格納されています。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、このファイルを使用できます。マスタ *interfaces* ファイルのコピーを配布すれば、ネットワーク上にあるすべての 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切な UNIX Sybase ディレクトリにコピーできます。

次の 2 とおりの方法のいずれかでマスタ・ファイルを作成できます。

- `dsedit` または `dscp` を使用する。
- テキスト・エディタを使用する。

`dsedit` または `dscp` を使用してマスタ *interfaces* ファイルを作成する

`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* ファイルを作成して、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` ユーティリティまたは `dscp` ユーティリティを使用する方が簡単です。`dsedit` または `dscp` を使用すると、*interfaces* ファイルのフォーマットの一貫性が保証されます。

`dsedit` または `dscp` を使用してマスタ *interfaces* ファイルを作成するには、次の手順に従います。

- 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 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て、この *interfaces* ファイルを編集します。
- 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。

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。

テキスト・エディタを使用したマスタ interfaces ファイルの作成

個々の interfaces ファイルから 1 つのマスタ interfaces ファイルを構成するには、次の手順に従います。

- 1 個々の interfaces ファイルを連結します。
- 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。

注意 *interfaces* ファイルを手動で編集する場合、各エントリの 2 行目以降の各行が<タブ>文字で始まっていることを確認してください。

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・エントリをユニークにする必要がある。*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 を受け入れることができる。デフォルト名を受け入れる場合、結合したファイル内の重複する SYBASE エントリを検索して名前を変更する。
- 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との組み合わせ。
- 元の *interfaces* ファイルがネットワーク上にマシンが 1 つしかないときに作成された場合、そのファイルのエントリのマシン名 (アドレス) の場所に “loghost” という語が指定されている場合がある。*loghost* がある場合、それをマシン名と置換する。

複数のネットワークで使用する 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す。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・インタフェース用のエントリを *interfaces* ファイルに追加します。

複数のネットワーク・ハンドラ用にサーバを設定する

複数のネットワーク・リスナを設定するには、次の手順に従います。

- 1 オペレーティング・システムのホスト・データベース内で、各ネットワーク・インタフェースに対して一意の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の各インタフェースに `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。

複数のネットワーク・ハンドラ用 `interfaces` ファイルの例

次の例は、2つのネットワーク・インタフェースを持つ `Adaptive Server` の `interfaces` ファイルを示します。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つの方法があります。

- 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。
- クライアントに対して異なる `DSQUERY` 名を使用する。すべてのマシンで `interfaces` ファイルは同じだが、`interfaces` ファイルには複数の `DSQUERY` 名が含まれている。

単一のネットワーク独立 DSQUERY 名の使用

クライアントに同一の DSQUERY 名を付けることが重要な場合、クライアントのネットワーク・アドレスに必要な変更を interfaces ファイル内に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、次の例のようになります。

```
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。

両方のファイル内に完全な “master” 行があります。Adaptive Server だけが “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 名を使用する

各ネットワーク・リスナで異なる DSQUERY 名を使用するには、次の手順に従います。

- 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、`dsedit` を使用して各ネットワークに1つずつ `sql.ini` ファイルのエントリをサーバに対して複数作成する。次の例では、`PRODUCTION_network1` と `PRODUCTION_network2` に1つずつエントリを作成する。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・エディタで `interfaces` ファイルを編集できる。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` ファイルにコピーする。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、クライアントが動作しているネットワークに対応した `DSQUERY` の値を使用します。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

複数のネットワーク・インタフェースは、ネットワークで障害が発生した場合のバックアップとしても使用できます。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。

クエリ・ポート・バックアップ用に `Adaptive Server` を設定するには、次の手順に従います。

- 1 `interfaces` ファイルのサーバ・エントリ内に、複数の “master” 行と “query” 行をインストールします。
- 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。

次に、通常の接続が失敗した場合だけに使用するバックアップ・ネットワークの設定方法の例を示します。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、PC クライアントの *interfaces* ファイルを適切な複数の “query” エントリで設定します。同機種間環境のクライアント *interfaces* ファイルの場合、Adaptive Server の *interfaces* ファイルのエントリすべてをクライアントの *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) の特徴を持ち、さらに IPv6 の一部の機能を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 の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 Enterprise 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 5-3 に、プラットフォームの実行時要件と特定の製品およびそのリリース・バージョンを示します。

表 5-3: IPv6 のサポート

プラットフォーム	IPv6 認識の Adaptive Server	IPv6 認識の Open Client/Server
Sun Solaris 8 32 ビット版および 64 ビット版	12.5.3a および 15.0	12.5 および 15.0
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	12.5 および 15.0
Microsoft Server 2003	12.5.3a および 15.0	12.5 および 15.0
Linux RHEL 3.0	15.0	12.5 および 15.0

XP Server、Backup Server、Replication Server、Open Switch などの 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 認識になります。Open Client 製品は、IPv6 認識ではありません。

Adaptive Server Enterprise 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 に対する 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 Enterprise 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

注意 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、1 つの Adaptive Server の設定では、DCL 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 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、ログに記録するように、トレース・フラグ 7815 を設定できます。

トラブルシューティング

この項では、サーバの起動不能の原因となるいくつかの一般的な状況について、その解決方法を説明します。

サーバが起動しない

サーバが起動せずに次のメッセージが表示されるときは、`interfaces` ファイル内で指定したポート番号が使用中の場合があります。

```
UI-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.
UI-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.
```

❖ ポートの割り当てを調べる

- 1 `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。
- 2 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、このポートはサーバに使用できません。別のプロセスがすでにこのポートを使用しています。

- 3 サーバ・ポートが使用中か検証するには、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。
手動でサーバを起動する方法については、使用しているプラットフォームの『インストールガイド』と『ASE ユーティリティ・ガイド』を参照してください。

❖ 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

- 1 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、プロセスを終了させる。
 - `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。
- 2 サーバを手動で起動して、ポート番号が使用できるか確認します。

手動でサーバを起動する方法の詳細については、使用しているプラットフォームの『インストール・ガイド』と『ASE 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

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.
```

ポート番号が別のプロセスで使用されている可能性があるため、XP Server を起動できません。前の項で説明した `netstat` コマンドを使用して、XP Server 用に指定したポート番号が使用中か確認します。

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。

- 1 Adaptive Server を再起動する。
- 2 先ほど試みた ESP を実行します。

XP Server は自動で起動します。

同じポート番号を使用するプロセスが見つかった場合は、次のいずれかを実行します。

- XP Server で新しいポート番号を使用するように `interfaces` ファイルを変更する。
- 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。

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 LDAP (Lightweight Directory Access Protocol) の使用

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。この章では、LDAP ディレクトリ・サービスを使用して接続を確立する方法について説明します。LDAP は Adaptive Server の機能ですが、ライセンスが必要です。

トピック名	ページ
概要	67
LDAP ディレクトリ・サービスと Sybase interfaces ファイルの違い	68
libtcl*.cfg ファイル	71
LDAP ディレクトリ・サービスの有効化	72
ディレクトリ・サービスへのサーバの追加	73
複数のディレクトリ・サービス	75
パスワードの暗号化	75
パフォーマンス	76
interfaces ファイルから LDAP へのマイグレート	77

概要

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

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

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 サーバの設定時に、次のアクセス制限を指定できます。

- 匿名認証 – すべてのユーザがあらゆる情報にアクセスできます。
- ユーザ名とパスワードによる認証 – Adaptive Server は、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異なります。

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

複数のディレクトリ・サービスが1つのサーバでサポートされる場合は、その検索の順序は `libtcl*.cfg` に指定されます。検索順は `dataserver` コマンドライン・オプションでは指定できません。詳細については、「[複数のディレクトリ・サービス](#)」(75 ページ) を参照してください。

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

LDAP ドライバは、LDAP サーバと組み合わせて使用するディレクトリ・サービスを実装します。LDAP ディレクトリは、以下の要素を実現するインフラストラクチャです。

- 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

表 6-1 は、Sybase interfaces ファイルと LDAP サーバの相違点を示します。

表 6-1: interfaces ファイルと LDAP ディレクトリ・サービスの違い

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格納できない	サーバのメタデータを格納できる

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。属性のリストについては、表 6-2 を参照してください。

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。LDAP ディレクトリ・サービスを使用してサーバに接続する場合は、`isql` ではなく、`isql_r` を使用してください。

表 6-2 に、Sybase LDAP ディレクトリ・エントリのリストを示します。

表 6-2: Sybase LDAP ディレクトリの定義

属性名	値のタイプ	説明
ditbase	<i>interfaces</i> ファイルまたは <i>libtcl.cfg</i>	オブジェクト・ツリーの DIT ベース。 <i>libtcl.cfg</i> ファイルが指定された場合は、 <i>interfaces</i> ファイルは無視される。 <i>libtcl.cfg</i> ファイルは、指定された接続用に <code>ct_con_prop()</code> で上書きできる。
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 または Sybase SQL 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 (大文字と小文字を区別して入力する) アドレス：そのプロトコル・タイプの有効なアドレス <p>注意 <code>dscp</code> は、この属性をトランスポート・タイプとトランスポート・アドレスに分割します。</p>
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、 <code>CS_RETRY_COUNT</code> にマッピングされる。 <code>CS_RETRY_COUNT</code> は、 <code>ct_connect</code> がサーバ名と対応するネットワーク・アドレスのシーケンスをリトライする回数を指定する。

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、ct_connect がアドレスのすべてのシーケンスをリトライするまでの遅延時間を秒単位で指定する。
sybaseHAServername (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 接続の LDAP エントリとフェールオーバー・マシンを示します。

```
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
sybaseHAServernam: secondary
```

LDAP ディレクトリ・サービスへのすべてのエントリーは、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。「[libtcl*.cfg ファイル](#)」(71 ページ) を参照してください。

上記の例のエントリーは、“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。このエントリーは、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` ファイルにあります。

同じディレクトリに、`sybase-schema.conf` ファイルもあります。このファイルには、同じスキーマが格納されていますが、Netscape 固有の構文を使用します。

LDAP では各属性の複数のエントリーをサポートしているため、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。表 6-2 の `sybaseAddress` を参照してください。

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。

このエントリは、**dsedit** を使用して編集できます。「[ディレクトリ・サービスへのサーバの追加](#)」(73 ページ) を参照してください。

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 サーバ名、ポート番号、DIT ベース、ユーザ名、パスワードを指定し、LDAP サーバへの接続を認証します。

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

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する必要があります。

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

LDAP を *libtcl.cfg* ファイルで指定した場合は、`interfaces` ファイルは使用されません。

注意 起動時に `-l` 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* ファイルを上書きして `interfaces` ファイルを使用します。

最も単純な形式では、*libtcl.cfg* ファイルは次のフォーマットになります。

```
[DIRECTORY]
ldap=libsybdldap.dll ldapurl
```

ここで、*ldapurl* は次のように定義されます。

```
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 サーバが読み込み専用アクセス可能な場合にだけ動作します。

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

libtcl.cfg ファイルでユーザ名とパスワードを LDAP URL への拡張機能として指定すると、接続時にパスワード認証が有効になります。

LDAP ディレクトリ・サービスの有効化

ディレクトリ・サービスを使用するには、次の手順に従います。

- 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 使用している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]* エントリにある LDAP URL 行の行頭から、コメント・マーカのセミコロン (;) を削除します。
- *[DIRECTORY]* エントリに LDAP URL を追加します。サポートされている LDAP URL 値については、[表 6-3](#) を参照してください。

警告！ LDAP URL は、1 行で記述する必要があります。

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

```
libtcl64.cfg
ldap=libsydbldap64.so ldap://host:port/ditbase??scope??bindname=username
password
```

次に例を示します。

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

“*one*” は検索の範囲であり、DIT ベースの 1 つ下のレベルのエントリが検索されます。

表 6-3 に、*ldapurl* 変数のキーワードの定義を示します。

表 6-3: *ldapurl* 変数

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)

- 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。
- 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。「[ディレクトリ・サービスへのサーバの追加](#)」(73 ページ)を参照してください。

ディレクトリ・サービスへのサーバの追加

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が、代わりに *dsedit* を使用することをおすすめします。LDAP サーバには、汎用ツールからは提供されない組み込みのセマンティック・チェックがあるからです。

各サーバ・エントリは、一連の属性で構成されています。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。受け入れられる値については、[表 6-2](#) を参照してください。

srvbuild を使用して、エントリの追加はできますが、修正と削除はできません。

❖ **dsedit を使用してディレクトリ・サービスにサーバ・エントリを追加する**

LDAP URL を *libtcl.cfg* ファイルに追加してから、LDAP サーバ・エントリを追加、削除、変更してください。「[libtcl*.cfg ファイル](#)」(71 ページ) を参照してください。

dsedit を使用してディレクトリ・サービスにサーバを追加するには、次の手順に従います。

- 1 *SYBASE.csh* または *SYBASE.sh* ファイルを source コマンドで使用し、環境変数を設定します。
- 2 cd を実行して、*\$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 [Add New Server Entry] 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。
 - セキュリティ・メカニズム – オプション。高可用性フェールオーバーサーバを使用している場合は、その名前を入力します。
- 7 [Add New Network Transport] をクリックして次の操作を実行します。
 - ドロップダウン・リストからトランスポート・タイプを選択します。
 - ホスト名を入力します。
 - ポート番号の入力。
- 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 URL `http://host:port/ditbase??one` を入力します。

次に例を示します。

```
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、Sybase の Web サイト (<http://www.sybase.com/support/manuals/>) で Generic Collections の Open Client/Server 11.1.x の『Open Client/Server Configuration Guide』を参照してください。

複数のディレクトリ・サービス

LDAP サービスは、実際のサーバか、他の LDAP サービスへのゲートウェイかに関係なく、どのタイプも LDAP サーバと呼ばれます。

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。

次に例を示します。

```
[DIRECTORY]
ldap=libsydbldap.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 ドライバへの接続にフェールオーバーします。この接続も失敗すると、*huey:11389* 上の LDAP サーバに接続しようとします。DIT ベースのフォーマットはベンダによって異なります。

注意 詳細については、Sybase の Web サイト (<http://www.sybase.com/support/manuals/>) で『Open Client Client-Library/C プログラマーズ・ガイド』と『Open Client Client-Library/C リファレンス・マニュアル』を参照してください。

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の暗号化用の `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されたパスワードを暗号値に置換して表示する単純なアルゴリズムを提供します。この値は、パスワードと置き換えられます。`pwdcrypt` は `$$SYBASE/$$SYBASE_OCS/bin` にあります。

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

```
bin/pwdcrypt
```

要求されたら、パスワードを 2 度入力します。

`pwdcrypt` が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。

```
0x01312a775ab9d5c71f99f05f7712d2cded2i8d0ae1ce78868d0e8669313
d1bc4c706
```

標準的な ASCII 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、ファイル・エンタリが次のように表示されます。

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

パスワードを、暗号化した文字列に置き換えます。

```
ldap=libsybdlldap.so
ldap://seashore/dc=sybase,dc=com??one??bindname=uid=Manager,dc=sybase,dc=com?
0x01312a775ab9d5c71f99f05f7712d2cded2i8d0ae1ce78868d0e8669313d1bc4c706
```

警告！ パスワードが暗号化されている場合でも、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 を起動したときに行われるので、パフォーマンスに違いがある場合はログイン時にわかります。通常のシステム負荷では、パフォーマンスの低下を感じることはありません。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

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ことはできません。以前のリリースの Adaptive Server から Adaptive Server バージョン 15.0 に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。

サーバ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できます。

- 1 サーバを停止します。「[第 2 章 サーバの起動と停止](#)」を参照してください。
- 2 `$$SYBASE/$SYBASE_OCS/config/libtcl.cfg` 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。「[LDAP ディレクトリ・サービスの有効化](#)」(72 ページ)を参照してください。
- 3 `dsedit` を使用して、ディレクトリ・サービスにサーバ・エントリを追加します。「[ディレクトリ・サービスへのサーバの追加](#)」(73 ページ)を参照してください。
- 4 サーバを再起動します。

Adaptive Server のローカライゼーションのカスタマイズ

この章では、言語、文字セット、ソート順の設定を含めた各言語に対応したインストールのための、Sybase のローカライゼーション・サポートについて説明します。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。

トピック名	ページ
ローカライゼーション・サポートの概要	79
文字セット変換	86
ソート順	87
言語モジュール	90
ローカライゼーション	91
ローカライゼーション設定の変更	94

ローカライゼーション・サポートの概要

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合させることです。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。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を定義したファイルが付属しています。

Sybase は、次の地域の主要な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・メッセージの変換 – 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

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。

サーバの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ていて、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 の文字セットや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または 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。

srvbuild によって新しい Adaptive Server が設定されると、Adaptive Server をアメリカ英語以外の言語にローカライズして、デフォルト以外の文字セットやソート順を使用するかをたずねるメッセージ・ボックスが表示されます。

srvbuild は、次のデフォルトで Adaptive Server を作成します。

- 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、配布メディアから使用している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インストールされます。次の操作ができます。

- [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、言語、文字セット、ソート順のデフォルトを変更するために使用される GUI ユーティリティです。

デフォルトでは、IBM、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 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 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 のデフォルトとして、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。次のガイドラインを考慮して、新しいデフォルトの文字セットを選択します。

- 変換エラーやオーバー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 を指定するとデータ変換の量を最小化することができます。

- 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。通常、これは Unicode (UTF-8) になります。

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または 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。

サポートされている文字セット

Adaptive Server では、次の言語、スクリプト、文字セットがサポートされています。

- アラビア語 – 表 7-1 (83 ページ) を参照。
- バルト語 – 表 7-2 (83 ページ) を参照。
- 中国語 (簡体字) – 表 7-3 (83 ページ) を参照。
- 中国語 (繁体字) – 表 7-4 (83 ページ) を参照。
- キリル語 – 表 7-5 (84 ページ) を参照。
- 東欧言語 – 表 7-6 (84 ページ) を参照。
- ギリシャ語 – 表 7-7 (84 ページ) を参照。
- ヘブライ語 – 表 7-8 (84 ページ) を参照。
- 日本語 – 表 7-9 (84 ページ) を参照。
- 韓国語 – 表 7-10 (85 ページ) を参照。
- タイ語 – 表 7-11 (85 ページ) を参照。
- トルコ語 – 表 7-12 (85 ページ) を参照。

- Unicode (650 を超える言語をサポート) – 表 7-13 (85 ページ) を参照。
- ベトナム語 – 表 7-14 (85 ページ) を参照。
- 西欧言語 – 表 7-15 (86 ページ) を参照。

下記の表は、それぞれの文字セットと、Unilib 変換 (Unilib 要求カラム) が必要かどうかについての情報を示しています。

- チェック・マーク (x) – Unilib 変換が必要な文字セット。
- チェック・マークなし – Unilib 変換と組み込み変換のどちらかを使用する文字セット。

詳細については、「文字セット変換」(86 ページ) を参照してください。

表 7-1 に、アラビア語の文字セットを示します。

表 7-1: アラビア語の文字セット

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

表 7-2 に、バルト語の文字セットを示します。

表 7-2: バルト語の文字セット

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

表 7-3 に、中国語 (簡体字) の文字セットを示します。

表 7-3: 中国語 (簡体字) の文字セット

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
gb18030	X	PRC 18030 標準

表 7-4 に、中国語 (繁体字) の文字セットを示します。

表 7-4: 中国語 (繁体字) の文字セット

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

表 7-5 にキリル語の文字セットを示します。

表 7-5: キリル語の文字セット

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
kz1048		カザフスタン・キリル語

表 7-6 に、東欧言語の文字セットを示します。

表 7-6: 東欧言語の文字セット

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

表 7-7 に、ギリシャ語の文字セットを示します。

表 7-7: ギリシャ語の文字セット

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

表 7-8 に、ヘブライ語の文字セットを示します。

表 7-8: ヘブライ語の文字セット

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

表 7-9 に、日本語の文字セットを示します。

表 7-9: 日本語文字セット

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

表 7-10 に、韓国語の文字セットを示します。

表 7-10: 韓国語の文字セット

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

表 7-11 に、タイ語の文字セットを示します。

表 7-11: タイ語のクライアント文字セット

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

表 7-12 に、トルコ語の文字セットを示します。

表 7-12: トルコ語の文字セット

文字セット	Unilib の必要性	説明
cp857		IBM PC トルコ語
cp1254		Microsoft Windows トルコ語
iso88599		ISO 8859-9 Latin-5 トルコ語
turkish8		HP TURKISH8

表 7-13 に、Unicode の文字セットを示します。

表 7-13: Unicode の文字セット

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

表 7-14 に、ベトナム語の文字セットを示します。

表 7-14: ベトナム語の文字セット

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

表 7-15 に、西欧言語の文字セットを示します。

表 7-15: 西欧言語の文字セット

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 – U.S. コード・セット
cp850		IBM CP850 – ヨーロッパ・コード・セット
cp860	X	PC ボルトガル語
cp858	X	cp850 ヨーロッパ言語のサポート付き
cp1252	X	Microsoft Windows US (ANSI)
iso_1		ISO 8859-1 Latin-1
roman8		HP ROMAN8
iso15	X	ISO 8859-15 Latin-1 ヨーロッパ言語のサポート付き
roman9	X	HP ROMAN8 ヨーロッパ言語のサポート付き

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。次の必要条件を考慮して、文字セットを選択します。

- 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する必要があります。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 変換を有効にするには、`sp_configure` を使用して `enable unicode conversions` をオンにします。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。

Adaptive Server または Backup Server のいずれかが、クライアントの言語と文字セットをサポートしていないと、サーバは警告メッセージを表示します。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にも、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、同じ言語に対する文字セット間、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、自動文字セット変換は、次の西欧言語の文字セット間で実行できます。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できます。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。サポートされている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。

クライアント／サーバ間の変換

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ければ、ユーザのデフォルト言語または Adaptive Server のデフォルト言語を使用します。
- 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表示し、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

それぞれの文字セットには、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) の中にあります。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。ただし、サーバは一度に1つのソート順だけをサポートすることを考慮して、すべてのクライアントで動作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または 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。

利用できるソート順

Adaptive Server が文字データの整列、比較や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) の中にあります。

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
```

ローカライゼーション・ファイルの詳細については、「[ローカライゼーションのディレクトリ](#)」(91 ページ) を参照してください。

表 7-16 に、インストール時に指定できるソート順と、インストールした後で `sqlloc` ユーティリティを使用して指定できるソート順を示します。

表 7-16: Adaptive Server で利用できるソート順

ソート順の名前	説明
バイナリ順	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。 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。バイナリ順では要求に合わない場合は、インストール時や、インストールした後で <code>sqlloc</code> ユーティリティを使用し、他のソート順を指定できる。
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)	大文字と小文字を区別する。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。

ソート順の名前	説明
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)	大文字と小文字を区別しない辞書のソート順。大文字と小文字は等しく処理され、ソート結果では両者が混在する。
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)	大文字と小文字を区別しない辞書のソート順。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)	状況によって照合の優先度が異なる、大文字と小文字を区別しない辞書のソート順。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。 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。
	注意 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いては、このソート順を選択しないでください。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、大きなテーブルでのパフォーマンスが低下する場合があります。
代替辞書順 (大文字と小文字を区別する)	大文字と小文字を区別する代替辞書ソート順。小文字をソートしてから大文字をソートする。 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)	大文字と小文字、およびアクセントを区別しない代替辞書のソート順。 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)	大文字と小文字を区別しないが、大文字が優先される代替辞書のソート順。 いくつかの西欧言語で使用する。
スペイン語辞書順 (大文字と小文字を区別する)	大文字と小文字を区別するスペイン語辞書のソート順。 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)	大文字と小文字を区別しないスペイン語辞書のソート順。 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)	大文字と小文字、およびアクセントを区別しないスペイン語辞書のソート順。 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)	大文字と小文字、およびアクセントを区別しないが、大文字が優先度される辞書のソート順。 スカンジナビア語で使用する。

利用できるソート順を確認するには、**sqlloc** を使用して、使用する文字セットのソート順を表示します。utf_8 の Unicode ソート順の詳細については、『システム管理ガイド 第 1 巻』の「第 7 章 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 のエラー・メッセージをアメリカ英語 (us_english) 以外の言語で表示する場合は、適切な言語モジュールをインストールしてください。

新しい言語モジュールをインストールすると、その新しい言語をサポートする言語を定義したファイルが、Sybase インストール・ディレクトリに自動的にロードされます。ディレクトリの詳細については、「[ローカライゼーションのディレクトリ](#)」(91 ページ) を参照してください。

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、手で追加の言語モジュールをインストールする必要があります。

新しい言語モジュールをインストールするには、次の手順に従います。

- 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してください。
- 2 言語を再設定し、必要に応じて Adaptive Server の文字セットやソート順も再設定します。詳細については、「[ローカライゼーション設定の変更](#)」(94 ページ) を参照してください。

メッセージ言語

メッセージ言語として、Adaptive Server にはデフォルトでアメリカ英語がインストールされています。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中に、デフォルト言語としてアメリカ英語以外の言語を指定できるが、指定する言語の言語モジュールがインストールされていることが必要。
- 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のロードが必要。ロード完了後、クライアントが使用する言語用に Adaptive Server 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をインストールして、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け取ります。

ローカライゼーション

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を使用します。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される、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・プロセス中、あるいは再設定中に、デフォルト以外の言語、文字セット、ソート順を指定できます。

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、次のディレクトリがあります。

- *locales*
- *charsets*

次の表に、ローカライゼーション・ファイルの構造を示します。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/	<i>charsets</i>	<i>charset_name</i>	*.srt ファイル
		<i>charset_name...</i>	<i>charset.loc</i>
		<i>Unicode</i>	*.uct ファイル
	<i>locales</i>	<i>language_name</i>	<i>charset_name</i>
		<i>language_name...</i>	<i>charset_name...</i>
		<i>locales.dat</i>	
		<i>message</i>	<i>language_name</i>
			<i>language_name...</i>

ディレクトリについて

`$$SYBASE/locales` ディレクトリには、利用できる各言語の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、現地の日付や時刻、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 内の各ファイルには、文字セットの定義と、その文字セットで利用できるすべてのソート順の定義など、特定の文字セットに関連する情報があります。

locales.dat ファイルについて

locales.dat ファイルは、次のように編集できます。

- プラットフォームのデフォルトの言語または文字セットを変更する
- プラットフォームのロケール名と Sybase の言語や文字セット名の間の新しい関連付けを行う

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
```

各要素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て、*\$\$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の場所を識別する文字セットの名前。

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* を使用するように指定します。

```
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。接続のプロセスは次の手順に従います。

- 1 クライアント・アプリケーションは、起動すると、オペレーティング・システムのロケール設定をチェックしてから *locales.dat* ファイルをチェックし、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。フランス語のロケール・エントリの例を次に示します。

```
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、既に *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。

locales.dat ファイルの編集

編集を開始する前に元のファイルをコピーし、編集後のファイルで問題が発生する場合に備えます。

locales.dat ファイルの編集は、次の手順に従います。

- 1 *locales.dat* ファイルのコピーを、テキスト・エディタで開きます。
- 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 では、*[sun_svr4]* です。
 - HP では、*[hp ux]* です。
 - IBM では、*[aix]* です。

- 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* の値を、オペレーティング・システムに必要な値と一致させてください。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ないと、アプリケーションは正しく実行されません。

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 が文字セット ROMAN8 を使用しているときは、使用している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。

ローカライゼーション設定の変更

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のローカライゼーションを使用します。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される、ソート順を定義したファイル
- *us_english* のシステム・メッセージ・ファイル

インストール・プロセス中と再設定中に、デフォルト以外の言語、文字セット、ソート順を指定できます。

Adaptive Server のローカライゼーション

各言語は、モジュールごとに約 2MB のデータベース領域を使用します。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、それから他の言語を追加します。

注意 複数の言語を Adaptive Server にインストールしたいが、**master** データベースのサイズが複数言語の管理に十分でない場合は、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 第2巻』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、source コマンドを使用して、*SYBASE.csh* または *SYBASE.sh* のいずれかを実行します。
- 2 サーバ上の 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 [Localize] を選択します。
- 4 Adaptive Server の選択ウィンドウから、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。選択できる言語は、デフォルトの文字セットでサポートされている言語だけです。

[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 [Localization Summary] ウィンドウに、選択した設定オプションが示されます。[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。

Backup Server のローカライゼーション

Backup Server の言語と文字セットは、*RUN_<backup_server_name>* ファイルを修正することにより変更できます。*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 へ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 を設定するには、次の手順に従います。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。

- 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。

`charset` を使用するには、あらかじめサーバを起動し、システム管理者権限を持っている必要があります。ソート順の *file name* を次のように使用します。

```
$$SYBASE/$$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。表 7-17 (97 ページ) を参照してください。*character_set* を使用する文字セットの Sybase 名と置き換えます。表 7-18 (99 ページ) を参照してください。

- 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。このユーティリティの詳細については、「[charset ユーティリティ](#)」(100 ページ) を参照してください。

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してい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 ソート順の ID を使って、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。表 7-17 (97 ページ) を参照してください。*character_set* を使用する文字セットの Sybase 名と置き換えます。表 7-18 (99 ページ) を参照してください。

- 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 システムでは通常のプロセスによってサーバをリブートします。通常、`$$SYBASE/$$SYBASE_ASE/install` から任意の `RUN_xxx` スクリプトを呼び出します。
- 7 サーバを起動し、すべてのシステム・インデックスを再構築してから停止します。サーバを再起動して、安定した状態になるまで待ちます。

ソート順

表 7-17 に、利用できるソート順を示します。使用している言語が表示されず、その言語固有のソート順がない場合は、バイナリ・ソート順を使用します。

表 7-17: 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	<i>binary.srt</i>	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)	<i>cyrdict.srt</i>	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)	<i>cyrnocs.srt</i>	64
英語	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)	<i>dictiona.srt</i>	51
フランス語	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)	<i>nocase.srt</i>	52
ドイツ語	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)	<i>nocasepr.srt</i>	53
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、およびアクセント記号を区別しない)	<i>noaccent.srt</i>	54
英語	代替辞書順 (大文字と小文字を区別する)	<i>altdict.srt</i>	45
フランス語	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)	<i>altnoacc.srt</i>	39
ドイツ語	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)	<i>altnocsp.srt</i>	46
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)	<i>elldict.srt</i>	65
ギリシャ語	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)	<i>elldict.srt</i>	65
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)	<i>hundict.srt</i>	69
ハンガリー語	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)	<i>hunnoac.srt</i>	70
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しない)	<i>hunnoacs.srt</i>	71
ロシア語	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)	<i>rusdict.srt</i>	58
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 (大文字と小文字を区別しない、アクセント記号を区別する)	<i>rusnocs.srt</i>	59
スカンジナビア語	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)	<i>scandict.srt</i>	47
これらのソート順は CP 850 のみで使用される。	辞書順 (大文字と小文字を区別しない、優先度を付けた順位)	<i>scannocsp.srt</i>	48

言語またはスクリプト	ソート順	ファイル名	ID
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)	<i>espdict.srt</i>	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)	<i>espnocs.srt</i>	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)	<i>espnoac.srt</i>	57
タイ語	辞書順	<i>dictionary.srt</i>	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)	<i>turdict.srt</i>	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)	<i>turnoac.srt</i>	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)	<i>turnocs.srt</i>	74

文字セット

表 7-18 に、利用できる文字セットとその Sybase 名を示します。

表 7-18: 文字セットの Sybase 名

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
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	eucens
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

文字セット	Sybase 名
ISO 8859-15	iso15
カザフスタン・キリル語	kz1048
Koi8	koi8
ROMAN8	roman8
ROMAN9	roman9
シフト JIS	sjis
TIS 620	tis620
TURKISH8	turkish8
UTF-8	utf8

charset ユーティリティ

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 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できるのは、インストール時だけです。

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 7-19: charsets 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る。
-P	コマンド・ラインに“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。
-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。指定しない場合は、charset は SYBASE ディレクトリの interfaces ファイルを使用する。
-v	Sybase バージョン文字列を印刷してから、終了させる。他のオプションを同時に指定しない。
sort_order	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 sort_order 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、charset.loc を使用して文字セット・ファイルの名前を特定する。
charset	Adaptive Server が使用する文字セットのディレクトリを指定する。

エラー・メッセージのロギングとイベントのロギング

この章では、Adaptive Server のエラー・ロギング機能の使用方法を説明します。

トピック名	ページ
Adaptive Server エラー・ロギング	101
エラー・ログのパスの設定	102
メッセージの管理	103

Adaptive Server エラー・ロギング

Adaptive Server は起動するたびに、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。

`$$SYBASE/$SYBASE_ASE/install/server_name.log`

このファイルの役割と特性は、次のとおりです。

- 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、次のことができます。

- `sp_errorlog` を使用してエラー・ログのパスを動的に変更する。Adaptive Server で使用されない古いエラー・ログは移動して、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 第 1 巻』の「第 12 章 システムの問題の診断」を参照してください。

エラー・ロギングの有効化と無効化

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。「[ユーザ定義メッセージのロギング](#)」(103 ページ) を参照してください。

エラー・ログのパスの設定

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` コマンド内で `-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、サーバごとに異なるエラー・ログ・ファイル名を指定します。

Adaptive Server のエラー・ログのパス設定

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。

たとえば、エラー・ログのパスを

`$$SYBASE/$$SYBASE_ASE/install/ASE_2K.log` から `$$SYBASE` ディレクトリに変更するには、次のように入力します。

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

`RUN_server_name` ファイルの使用方法については、「[サーバの起動と停止](#)」(17 ページ) および『[ユーティリティ・ガイド](#)』を参照してください。

メッセージの管理

イベント・ロギングが有効になっているときは、次の方法でその機能を管理できます。

- `sp_addmessage` を使用してユーザ・メッセージを追加するか、`sp_altermessage` を使用して特定のメッセージを Adaptive Server のエラー・ログに書き込むかどうかを制御します。

`sp_addmessage` と `sp_altermessage` の完全な構文については、『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。
- 設定パラメータを使用して、監査イベントのログを取るかどうかを指定します。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`) に関係しています。

ユーザ定義メッセージのロギング

Adaptive Server エラー・ログにユーザ定義メッセージのログを取るかどうかを指定できます。Adaptive Server では、次のメッセージに対してこの設定ができます。

- 新しいメッセージ (`sp_addmessage`)
- 既存のメッセージ (`sp_altermessage`)

これらのコマンドとそのパラメータの詳細については、『リファレンス・マニュアル：プロシージャ』の「`sp_addmessage`」と「`sp_altermessage`」を参照してください。

新しいメッセージ

新しいユーザ定義メッセージを `sysusermessages` に追加するときは、`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には、`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

監査イベントのロギング

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。ただし、`sp_configure` パラメータを使用して Adaptive Server のエラー・ログにログインなど監査イベントのログを取るよう指定できます。

使用可能なパラメータと値は次のとおりです。

- `log audit logon success` を 1 に設定 – 成功した Adaptive Server へのログインのログを取る

```
sp_configure "log audit logon success", 1
```

- `log audit logon failure` を 1 に設定 – 失敗した Adaptive Server へのログインのログを取る

```
sp_configure "log audit logon failure", 1
```

- どちらかのパラメータを 0 に そのメッセージ・タイプ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
```

`sp_configure` 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。

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。

トピック名	ページ
データベース・デバイスの管理	105

データベース・デバイスの管理

「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を指します。

デバイスの要件

Adaptive Server のデバイスとデータベースには、以下の制約があります。

- 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。指定可能な最小ディスク・サイズは 1 MB であるため、使用可能な最小デバイスは 1 MB または 256 の論理ページより「大きく」なります。
 - 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 4TB.
 - 4K ページ・サーバの最大データベース・サイズは 8TB
 - 8K ページ・サーバの最大データベース・サイズは 16TB
 - 16K ページ・サーバの最大データベース・サイズは 32TB.
- 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 にはアクティブなすべてのデータベース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` コマンド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・システムの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できます。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例を次に示します。

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


この例では、"size = 2048" は、2048 の「仮想」ページをデバイスに割り当てるようにコマンドに指示しています。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。ここで、N は使用するデバイス番号です。デバイス番号を指定する場合は、このサーバの他のデバイスで使用されていない番号を使用してください。どのデバイス番号が既に使用されているかを調べるには、`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、そのデバイスのサイズを大きくすることができます。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 パラメータにはデバイスの最終的なサイズを指定する点が異なります。

`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。

`sp_helpdevice` コマンドと `disk init` コマンドの詳細については、『システム管理ガイド 第1巻』と『リファレンス・マニュアル: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。

Adaptive Server へのオプション機能の追加

この章では、Adaptive Server にオプションの機能を追加する方法について説明します。

トピック名	ページ
監査の追加	109
Transact-SQL 構文のオンライン・ヘルプのインストール	116

使用しているシステムに Sybase 製品をインストールしたら、設定や管理について、その製品のマニュアルを参照してください。

監査の追加

監査は、データベース管理システムのセキュリティの重要な機能です。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。監査証跡はシステムへの侵入やリソースの誤用を発見するのに使用します。システム・セキュリティ担当者は、監査証跡を詳細に調べ、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。

監査システムのデバイスとデータベース

監査システムはいくつかのコンポーネントで構成されています。主要なコンポーネントは次のとおりです。

- **sybsecurity** デバイスと **sybsecurity** データベース。監査情報を保存する。
- 監査証跡。設定時間に決定する監査デバイスとテーブルで構成される。
- **syslogs** トランザクション・ログ・デバイス。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 sybsecurity データベースを格納します。sybsecurity データベースは、監査設定プロセスの一部として作成されます。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・テーブルと、サーバワイドな監査オプションを追跡するためのシステム・テーブルおよび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、どの監査テーブルを現在のものにするかを設定したり、変更したりできます。

Adaptive Server を監査用に設定する場合、使用している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。指定できるシステム・テーブルは最高 8 つです (sysaudits_01 から sysaudits_08 まで)。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

監査用に設定するとき、トランザクション・ログを含む syslogs システム・テーブル用に別のデバイスを指定します。すべてのデータベースにある syslogs テーブルには、そのデータベース内で実行されたトランザクションのログが記録されています。

監査インストール環境の概要

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。詳細については、『セキュリティ管理ガイド』を参照。
- auditnit ユーティリティを使用する。監査のインストール前に実行する作業と auditinit ユーティリティの使い方は次項を参照。

監査デバイスのインストール前の作業

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します。この情報はあとで必要になります。

Sybase では次のことをおすすめします。

- システムを必要最低限の数の監査デバイスで設定する。これには最低 3 つのデバイスが必要です。sp_addauditable を使用すれば、あとでデバイスを追加できます。詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

- 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、スレッシュホールド上限値も共有することになります。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 つの監査テーブルを使用すると、たとえ 1 つのデバイスが壊れても、システム・セキュリティ担当者が新しいスレッシュホールド・プロシージャをインストールして、直るまで壊れた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して (最高 8 つまで) 監査能力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。このようにすると、監査機能を強化したいが、追加できるデバイスがほとんどないかゼロの場合、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。詳細については、「[データベース・デバイスとしての UNIX オペレーティング・システム・ファイルの使用](#)」(8 ページ)を参照してください。

監査のインストール

❖ Adaptive Server の監査を有効に設定する

- 1 Sybase 環境変数を設定していない場合は、`source` コマンドを使用して、`SYBASE.csh` または `SYBASE.sh` のいずれかを実行します。

- 2 UNIX プロンプトで `auditinit` を開始します。

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

`auditinit` によって、次のメニューが表示されます。

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

- 3 [Configure a Server Product] を選択します。
- 4 Adaptive Server を選択します。
- 5 [Configure an Existing Sybase Server] 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 Server Configuration] 画面で [Configure Auditing] 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は変更できます。各メニューを終了するときは、[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 [Configure Auditing] 画面で [Configure Auditing:] を選択します。

`auditinit` によって [Configure Auditing] メニューが再表示され、[Configure Auditing:] に “yes” と表示されます。

- 10 Adaptive Server を再起動して、変更を有効にします。

#### ❖ 監査テーブルのデバイスを作成する

- 1 [Configure Auditing] 画面で [Add a Device for Audit Table(s)] を選択します。

`auditinit` によって、次の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 device (Meg):
4. Device size for auditing:
```

- 2 [Sybsecurity Physical Device Name:] を選択します。

監査テーブルのデバイスを作成するには、次の手順に従います。

- 1 「[監査デバイスのインストール前の作業](#)」(110 ページ) でロケーションを決定した物理デバイス (ファイル・システムまたはロー・パーティション) の「フル・パス」を入力します。

```
Enter the physical name of the device to use for the audit
database (default is " "):
```

```
/dev/path_to_partition
```

`path_to_partition` は、デバイスのロー・パーティションまたはファイル名へのパスです。

- 2 [Return] キーを押して警告を確認します。

`auditinit` によって [Add/Change a New Device for Auditing] メニューが再表示され、デバイスの物理名が表示されます。

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1. sybsecurity physical device
name: /secret1/sybase_dr/install/aud1.dat
2. Logical name of the device:
3. Size of the device:
4. Device size for auditing:
```

- 3 このメニューの残りの項目に進みます。

注意 [Size of the Device:] の値は、[Device Size for Auditing:] の値以上になるようにしてください。[Device Size for Auditing:] は、デバイスのサイズと同じにします。Sybase の監査ガイドラインに従う場合は、[Device Size for Auditing:] に表示された値を変更する必要はありません。

- 4 [Ctrl + A] を押して設定を受け入れます。`auditinit` は [Configure Auditing] メニューに戻り、作成したデバイスを表示します。

```
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
```

- 5 複数の監査デバイスを追加するには、手順 1 ~ 6 を繰り返します。

追加できるデバイスは最高 8 つまでです。Sybase では 3 つ以上の監査テーブルを追加することをおすすめします。

デバイスを追加すると、`auditinit` は [Configure Auditing] メニューに戻って、作成したデバイスをすべて表示します。

```
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

❖ 監査データベース・トランザクション・ログのデバイスを作成する

- 1 [Configure Auditing] - [Add a Device for the Audit Database Transaction Log] を選択します。

auditinit は [Add/Change a New Device for Auditing] メニューを表示します。

```
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 Physical Device Name:] を選択します。

auditinit は物理名を入力するようプロンプトを表示し、デフォルトがある場合はそれを表示します。

```
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] キーを押して警告を確認します。

auditinit によって [Add/Change a New Device for Auditing] メニューが再表示され、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。次のこと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 にすることをすすめます。
 - **auditinit** は、[Add/Change a New Device for Auditing] メニューの [Size of the Device:] と [Device Size for Auditing:] の両方にサイズを表示します。

- [Device Size for Auditing:] 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・サイズと等しくなっています。デバイスのサブセットだけを使用したい場合は、[Size of the Device:] の値を編集できます。
- 6 [Ctrl+A] を押して、[Add/Change a New Device for Auditing] で表示された設定を受け入れます。

auditinit は [Configure Auditing] メニューに戻り、作成したデバイスを表示します。

```
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 name	Physical name Size	Segment name	Table	
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 監査設定を実行する準備ができれば、[Ctrl + A] を押します。**auditinit** は [Sybase Server Configuration] 画面に戻ります。

- 8 [Ctrl + A] を再度押します。**auditinit** プロンプトが表示されます。

```
Execute the Sybase Server Configuration now?
```

- 9 “y”(yes) を入力します。

auditinit は監査をインストールするタスクを開始します。インストールが正常に完了すると、次の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

監査をインストールしても、システム・セキュリティ担当者が次のコマンドを使用して有効にしないと監査は開始されません。

```
sp_configure 'auditing', 1
```

詳細については、『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**

- 1 [Configure Auditing] - [Delete a Device Entry] 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・エントリを変更する**

- 1 [Configure Auditing] - [Change a Device Entry] を選択します。
- 2 変更するデバイスの番号を入力します。

auditinit は、[Add/Change a New Device for Auditing] メニューに、選択したデバイスの情報を表示します。

```
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

この項では、Transact-SQL 構文のオンライン・ヘルプをインストールする方法を説明します。

オンライン構文ヘルプ： **sp_syntax**

`$$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。このデータは `sp_syntax` を使用して検索できます。`sp_syntax` の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

`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、表 10-1 に示す `sp_syntax` スクリプトが 1 つ以上入っています。

表 10-1: `sp_syntax` インストール・スクリプト

スクリプト	製品
<code>ins_syn_cl</code>	Open Client Client-Library™
<code>ins_syn_esql</code>	Embedded SQL™
<code>ins_syn_os</code>	Open Server
<code>ins_syn_sql</code>	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。このスクリプトを実行すると、`sybsyntax` データベースの SQL に関する部分がインストールされます。

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。最初に行うスクリプトでは、`sybsyntax` データベースと必要なテーブルおよびインデックスを作成します。それ以降に行う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。以前に行われたスクリプトを実行すると、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て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` スクリプトは矛盾します。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・デバイス

`sybsyntax` データベースは、少なくとも `model` データベースと同じ大きさの領域をデバイスに必要とします。デフォルトでは、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上に、`sybsyntax` インストール・スクリプトが `sybsyntax` データベースをインストールします。

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するときに、`sp_diskdefault` を使用していない場合は、スクリプトによってマスタ・デバイス上に `sybsyntax` がインストールされます。この設定はお勧めできません。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。

マスタ・デバイスに `sybsyntax` をインストールしないようにするには、次のいずれかの方法に従います。

- `sp_diskdefault` を使用して、デフォルトのデバイスをマスタ・デバイス以外に指定します。`sp_diskdefault` の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。
- 実行する `sybsyntax` インストール・スクリプトを修正して、異なるデバイスを指定します。次の項を参照してください。

sybsyntax のインストール

実行する sybsyntax インストール・スクリプトごとに、次の手順を実行します。

- 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。この情報はあとで必要になります。
- 2 オリジナル・スクリプトのコピーを作成します。編集したスクリプトに問題が発生したときは、このコピーを使用します。
- 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・デバイスの詳細については、「[sybsyntax データベースのデフォルト・デバイス](#)」(117 ページ) を参照してください。
 - 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

```

- このセクション全体をコメントした後、次の行をスクリプトに追加します。

```
create database sybsyntax on device_name
```

ここで、*device_name* は、sybsyntax をインストールするデバイス名です。

- 4 次のようなコマンドでスクリプトを実行します。

```
isql -Usa -Ppassword -Sservername < $SYBASE/$SYBASE_ASE/scripts/ins_syn_sql
```

ここで、*sa* はシステム管理者のユーザ ID です。*password* はシステム管理者のパスワードです。*servername* はデータベースをインストールする Adaptive Server 名です。

DSQUERY 環境変数を *servername* に設定している場合は、サーバ名を SDSQUERY に置き換えることができます。

- 5 インストールした **sybsyntax** データベースが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。次に例を示します。

```
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

記号

/etc/services ファイル 56

A

Adaptive Server

- interfaces ファイル内の名前 55
 - UNIX コマンド・ラインからの起動 18
 - エラー・ログのパス 102
 - オペレーティング・システムを使用した起動 20
 - カスタマイズ機能 47
 - 起動スクリプト 21
 - クライアント間の変換 87
 - クライアントの通信 49
 - 言語、変更 81
 - ソート順 81
 - 停止 24
 - データベース・デバイスの稼働条件 9
 - デフォルト設定 47, 48
 - デフォルトのソート順 81
 - デフォルトの文字セット 81
 - 文字セット 86
 - 文字セット、変更 81
- auditinit ユーティリティ 7

B

Backup Server

- Adaptive Server のデフォルト 48
- UNIX コマンド・ラインからの起動 18
- オペレーティング・システムを使用した起動 20
- 設定 91, 94
- デフォルト設定 48
- 文字セット 87, 95

C

- charsets ディレクトリ 88, 91
- 説明 92
- common.loc ファイル 92
- create database コマンド、作成されるシステム・テーブル 5

D

- dataserver ユーティリティ 26
- dbcc checkstorage、データベース 7
- dbcc エラー・メッセージ 45
- dbccdb データベース 7
- Dec-Kanji 文字セット 87
- directio 8
- disk init コマンド 11
- dscp ユーティリティ 12
- マスタ interfaces ファイルの作成 57
- dsedit
 - LDAP サーバの追加 74
- dsedit ユーティリティ 12
- マスタ interfaces ファイルの作成 57
- DSLISTEN 環境変数 3
- DSQUERY 環境変数 4
- クライアント接続 59
- 説明 51
- 名前 60
- 複数のネットワーク、異なる値を使用 60

E

- ESP。「ESP (拡張ストアド・プロシージャ)」参照
- EUC-JIS 文字セット 87

索引

H

HP-UX

- netstat コマンド 43
- ps コマンド 43
- sar コマンド 43
- time コマンド 43
- vmstat コマンド 43
- タイムアウト時間 40
- ネットワーク・プロトコル 55

I

I/O モニタ 43

IBM RS/6000

- iostat コマンド 43
- netstat -v コマンド 43
- netstat コマンド 43
- no -a コマンド 43
- ps コマンド 43
- time コマンド 43
- vmstat コマンド 43
- システムのモニタ 43
- タイムアウト時間 40
- ネットワーク・プロトコル 55
- ハードウェア・エラー・メッセージ 41

interfaces ファイル 11

- Adaptive Server が使用 53
- Adaptive Server の名前 55
- API 要素 55
- delay_interval 要素 55
- device 要素 56
- dscp を使用してマスタ・ファイルを作成 57
- dsedit を使用してマスタ・ファイルを作成 57
- ether プレースホルダ 55
- host 要素 56
- loghost プレースホルダ 58
- machine 要素 56
- network 要素 55
- port 要素 56
- protocol 要素 55
- retry_attempt 要素 55
- servername 要素 55
- service_type 要素 55
- SPX 用の address 要素 56
- 異機種間環境 51

エントリ内のユニークな要素 58

クエリ・サービス・タイプ 55

クエリ・ポート・バックアップの設定 61

クライアントが使用 49, 50

クライアントとサーバのバージョン、違い 52

作成、経験の浅い方 57

自動作成 49

スペース 54

説明 49

タブ文字 54

テキスト・エディタを使用したマスタ・ファイルの
作成 57

デバッグ・サービス・タイプ 55

デフォルトのロケーション 49

同機種間環境 51

内容 51

複数のネットワーク 51, 58

複数のネットワーク・リスナ 59

マスタ・サービス・タイプ 55

ロケーション 49

interfaces ファイル内の address 要素

TCP プロトコル・エントリ 56

interfaces ファイル内の API 要素

説明 55

interfaces ファイル内の delay_interval 要素 55

interfaces ファイル内の device 要素 56

interfaces ファイル内の ether プレースホルダ 55

interfaces ファイル内の host 要素 56

interfaces ファイル内の loghost 58

interfaces ファイル内の machine 要素 56

interfaces ファイル内の network 要素 55

interfaces ファイル内の port 要素 56

interfaces ファイル内の retry_attempts 要素 55

interfaces ファイル内の servername 要素 55

interfaces ファイル内の service_type 要素 55

interfaces ファイル内のスペース 54

interfaces ファイル内のタブ文字 54

interpubs サンプル・データベース 7

iostat コマンド

IBM RS/6000 43

Sun Solaris 44

iso-Latin1 文字セット 81

J

jpubs サンプル・データベース 7

K

KEEPALIVE オプション、TCP/IP 40
kill コマンド 25, 26

L

LD_LIBRARY_PATH 環境変数 4
LD_LIBRARY_PATH64 環境変数 4
LDAP
 interfaces ファイルとの比較 68
 libtcl.cfg で指定 71
 アクセス制限 68
 エントリ例 70
 サーバの追加 73
 定義 67
 ディレクトリ・スキーマ 70
 ディレクトリの定義 69
 複数のディレクトリ・サービス 75
 有効化 72
LDAP サーバ
 dsedit を使用した追加と修正 74
LDAP ライブラリ
 環境変数 73
 ロケーション 73
ldapurl
 キーワード 73
 定義 71
 例 71
LIBPATH 環境変数 4
libtcl*.cfg ファイル 71
 フォーマット 71
 目的 71
 ロケーション 71
libtcl*.cfg
 password 75
loc ファイル 91
locales ディレクトリ 91
locales.dat ファイル 92

M

maxfiles カーネル・パラメータ 34
maxfiles_lim カーネル・パラメータ 34
mbuf プール 43
model データベース 5

N

NCR
 システムのモニタ 43
netstart -v コマンド (IBM RS/6000) 43
netstat コマンド
 HP-UX 43
 IBM RS/6000 43
 Sun Solaris 44
no -a コマンド (IBM RS/6000) 43

P

PATH 環境変数 4
PCI (Pluggable Component Interface) 6
PID。「プロセス ID」参照
PPID。「親プロセス ID」参照
protocol
 interfaces ファイル内の要素 55
ps コマンド
 HP-UX 43
 IBM RS/6000 43
 Sun Solaris 44
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 75
 ロケーション 75

R

Roman8 文字セット 81
run control ディレクトリ 22
runserver ファイル 19

S

sar コマンド
 HP-UX 43
setenv コマンド 29
setperm_all コマンド 33
shutdown コマンド 24, 25
sllloc ユーティリティ 88
SMIT
 「System Management Interface Tool」参照
SPX ネットワーク・プロトコル 12, 55
srt ファイル 87

索引

srvbuild ユーティリティ 12
startserver ユーティリティ 19
stty 設定 33
Sun Solaris
 iostat コマンド 44
 netstat コマンド 44
 ps コマンド 44
 time コマンド 44
 vmstat コマンド 44
 タイムアウト時間 41
 ネットワーク・プロトコル 55
sundiag システム診断ツール 42
SYBASE 環境変数 4
Sybase のグローバルライゼーション・サポート 79, 90, 94
SYBASE_ASE 環境変数 4
SYBASE_OCS 環境変数 4
SYBASE_SYSAM 環境変数 4
sybsecurity
 データベース 7, 110
 デバイス 7, 10
sybsyntax データベース 116
sybssystemdb
 デバイス 10
 目的 5
sybssystemprocs データベース 6
sysprocsdev デバイス
 最小サイズ 10
 目的 5, 6
System Management Interface Tool (SMIT) 40

T

TCP/IP 40, 55
 KEEPALIVE オプション 40
TCP/IP ネットワーク・プロトコル 12
tempdb データベース 5
time コマンド
 HP-UX 43
 IBM RS/6000 43
 Sun Solaris 44
TLI プロトコル 55

U

Unicode
 文字変換 83
UNIX
 ネットワーク・プロトコル 55
 ハードウェア・エラー・メッセージ 41
UnixWare
 ネットワーク・プロトコル 55
us_english 言語 81

V

vmstat コマンド
 HP-UX 43
 IBM RS/6000 43
 Sun Solaris 44

W

Windows ソケット
 デフォルト・ソケット 47, 48

X

XP Server
 デフォルト設定 48
XP Server、起動 17
xp_cmdshell コマンド 17

あ

アクセント付き文字 13, 88
アラビア語の文字セット 83

い

異機種間環境 82, 86
 interfaces ファイル 53
 説明 53
インストール環境のカスタマイズ 47

え

- エラー・ロギング
 - 設定 102
- エラー・ログのパス 47, 102
 - 設定 102
- エラー、dbcc メッセージ内 45

お

- オペレーティング・システム
 - 管理者 3
 - リソース 42
- 親プロセス ID (PPID) 26
- オンライン構文ヘルプ 116

か

- 拡張スタアド・プロシージャ (ESP) 2
- 環境変数
 - DSLISTEN 3
 - DSQUERY 4, 51, 60
 - LD_LIBRARY_PATH 4
 - LD_LIBRARY_PATH64 4
 - LIBPATH 4
 - PATH 4
 - SYBASE 4
 - SYBASE_ASE 4
 - SYBASE_OCS 4
 - SYBASE_SYSAM 4
- 韓国語の文字セット 85
- 監査
 - auditinit ユーティリティを使用したインス
トール 110
 - installsecurity スクリプトを使用したインス
トール 110
 - グローバル・オプション 110
 - 追跡用のテーブル 110
 - データベース 110
 - データベース・デバイスの推奨サイズ 10
 - デバイス 110
 - プロセス 110
- 監査システム 109
- 監査証跡
 - 概要 109
 - システム監査テーブル 110

管理者

- Sybase システム 3
- オペレーティング・システム 3

き

- 起動スクリプト 20
- 共有メモリファイル 26
- ギリシャ語の文字セット 84
- キリル語の文字セット 84

く

- クエリ
 - サービスの種類 52, 55
 - 仕様 15
 - ポート・バックアップの設定 61
- クライアント
 - Adaptive Server の通信 49
 - DSQUERY 59
 - アプリケーションと locales.dat ファイル 93
 - サーバ間の変換 87
 - デフォルトの文字セット 82
 - ファイル・サーバ 60
- クライアントと Adaptive Server との通信 49
- クライアントの interfaces ファイル
 - 異機種 53
 - クライアントとサーバのバージョンの違い 52
 - 同機種 53
- グローバルライゼーション・サポート、
Sybase 47, 79, 90, 94

け

- 権限 11
- 言語
 - 指定言語でレポートされるエラー 91
 - 変換サポート 79
 - 変更 94
 - メッセージの選択 90
- 言語モジュール 90
 - 新規インストール 90
 - デフォルト 47
 - 翻訳されたシステム・メッセージ 80
 - メモリ要件 94
 - ローカライゼーション・ファイル 80

索引

現在のファイル記述子の表示 33
現地の日付、時刻、通貨のフォーマット 92

こ

コード変換
文字セット間 86
国際化システム
Sybase サポート 79
サポート 79
コマンド
disk init 11
iostat 43, 44
maxfiles 34
maxfiles_lim 34
netstat 43, 44
netstat -v 43
no -a 43
ps 43, 44
sar 43
setenv 29
setperm_all 33
time 43, 44
vmstat 43, 44

さ

サーバ
名前の要件 55
サーバの起動
UNIX コマンド・ラインからの起動 18
サーバの追加、LDAP 73
サーバの名前の要件 55
サービスの種類
クエリ 52, 55
デバッグ 55
マスタ 55
リスナ 52
サイズ
sysystemprocs データベース、アップグレードに必要な最小サイズ 14

作成

dscp ユーティリティを使用したマスタ interfaces ファイルの作成 57
dsedit を使用したマスタ interfaces ファイルの作成 57
interfaces ファイル 51, 57
interfaces ファイルの自動作成 49
経験の浅い方向けの interfaces ファイルの作成方法 57
テキスト・エディタを使用したマスタ interfaces ファイルの作成 57
参照整合性制約 15

し

辞書のソート順 88
スカンジナビア語 89
スペイン語 89
システム監査テーブル 110
システム・データベース
格納 9
システムのモニタ
IBM RS/6000 43
NCR 43
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 80
シフト JIS 文字セット 87
照合順。「ソート順」参照 87

す

スカンジナビア語辞書のソート順 89
スクリプト
C シェル 44
管理 44
起動 20, 21
サンプル管理 44
スペイン語辞書のソート順 89

せ

セキュリティ。「監査」参照
設定
Backup Server 91, 94
文字セット 95
設定、デフォルト 48

そ

- ソート順 87
 - Adaptive Server のデフォルト 81
 - 大文字と小文字 88
 - 辞書 88
 - 定義ファイル 87
 - データベース 88
 - バイナリ 88
 - 変更 82, 94
 - 文字セット 88
- ソート順における大文字と小文字 88

た

- タイ語の文字セット 85

ち

- 中国語の文字セット 83

て

- ディレクトリ
 - charsets 88, 92
 - run control 22
 - ローカライゼーション 91
- ディレクトリ・スキーマ、LDAP 70
- データの変換 79
- データベース 88
 - dbccdb 7
 - model 5
 - pcidb 6
 - sybsecurity 7
 - sybssystemprocs 6, 14
 - tempdb 5
 - サイズ 14
 - サンプル 7
 - 仕様 14
 - デバイス 105
 - マスタ 5, 6
- データベース・デバイス
 - sybsyntax 117
 - sybssystemdb 5
 - sysprocsdev 5, 6
 - マスタ 5
 - ロー・パーティションの準備 10

- テーブルの仕様 14

- デバイス
 - ファイル 105
- デバッグ・サービス・タイプ 55
- デフォルト
 - Adaptive Server の言語 81
 - Adaptive Server の文字セット 81
 - インストールされる文字セット 81
 - 言語、変更 81
 - ソート順 81
 - 文字セット、変更 81
- デフォルト・サーバ名としての \$SYBASE 環境変数 58

と

- ドイツ語サンプル・データベース 7
- 同機種間環境
 - interfaces ファイル 53
 - 説明 53
- ドキュメント
 - 翻訳された Adaptive Server のマニュアル 80
- トルコ語の文字セット 85

な

- 名前付きパイプ
 - デフォルト・パイプ 48

に

- 日本語サンプル・データベース 7

ね

- ネットワーク
 - DSQUERY 60
 - interfaces ファイル 49
 - 障害 62
 - ステータスのモニタ 43
 - バックアップ接続 61
 - 複数 51
- ネットワーク・サポート
 - デフォルト設定 47, 48

索引

ネットワーク・プロトコル

- Digital UNIX 55
- HP-UX 55
- IBM RS/6000 55
- SPX 12
- Sun Solaris 55
- TCP/IP 12
- UnixWare 55

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 41

- IBM RS/6000 41
- UNIX 41

パーミッション 33

- リストア 33

バイナリ・ソート順 88

パスワードの暗号化

- libtcl*.cfg 75
- pwdcrypt 75

バス、エラー・ログ 102

バッファの仕様 15

ひ

非同期 I/O (AIO)

- HP ドライバ 39

- 有効化 39, 40

非同期 I/O の有効化 39, 40

ふ

ファイル

- common.loc 92
- locales.dat 92
- runserver 19
- 共有メモリ 26

- ソート順定義 (.srt) ファイル 87

- デバイス・ファイル 105

- ローカライズされたエラー・メッセージ (.loc) 91

- ローカライゼーション 80

ファイル記述子

- 現在のパーミッションの表示 33

ファイル記述子の制限値 35

ファイル・サーバ 60

フォーマット、現地の日付、時刻、通貨 92

複数のインストール環境

- interfaces ファイルの作成 51, 57

- interfaces ファイルへの影響 53

複数のディレクトリ・サービス

- LDAP 75

複数のネットワーク

- interfaces ファイル 51, 58

- ネットワークの障害時のバックアップとして使用 61

プラットフォーム固有のロケール名 92

フランス語サンプル・データベース 7

プロシージャの仕様 15

プロシージャ、Sybase 拡張ストアド・プロシージャ 2

プロセス ID (PID) 26

プロトコル

- SPX 55

- TCP/IP 55

へ

ヘブライ語の文字セット 84

変換、Unicode 文字 83

変更

- 言語 94

- ソート順 94

- 文字セット 81, 94

ほ

ポート番号と interfaces ファイル 59

ホスト名

- 決定 56

翻訳されたメッセージ

- エラー (.loc ファイル) 91

- システム 80

ま

マスタ

- interfaces ファイル 51, 57

- サービスの種類 55

- データベース 5

- デバイス 5, 9

め

- メッセージ
 - 言語の選択 90
 - ハードウェア・エラー 41

も

- 文字型デバイス
 - I/O パフォーマンスの向上 39
- 文字セット 87
 - アクセント記号付きの文字 13
 - アメリカ英語 13
 - 異機種間環境 86
 - クライアントによる選択 82
 - コード変換 86
 - 設定 95
 - ソート順 88
 - データベース 88
 - デフォルト 80
 - 変換 86
 - 変更 81, 82, 94
- 文字セット間での変換 86
- モニタリング
 - I/O 43
 - オペレーティング・システム・リソース 42
 - 仮想メモリの使用状況 43
 - ネットワーク・ステータス 43

ゆ

- ユーザ接続 34
- ユーザ定義メッセージ 103
- ユーティリティ
 - silloc 88

ら

- ラテン語の文字セット 84

り

- リスナ・サービス 52

ろ

- ローカライゼーション 79
 - 一般的な情報 92
 - 設定の変更 94
- ローカライゼーション・サポート 47
- 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」参照
- ログイン
 - sa 3
 - sybase 3
 - スーパーユーザ 3
 - ルート 3

