



SQL Anywhere® Studio 新機能ガイド

パート番号 : DC03921-01-0902-01

改訂 : 2005 年 3 月

版權

Copyright © 2005 iAnywhere Solutions, Inc., Sybase, Inc. All rights reserved.

ここに記載されている内容を iAnywhere Solutions, Inc.、Sybase, Inc. またはその関連会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、SYBASE のロゴ、Adaptive Server、AnswerBase、Anywhere、EIP、Embedded SQL、Enterprise Connect、Enterprise Portal、GainMomentum、iAnywhere、jConnect MASS DEPLOYMENT、Netimpact、ObjectConnect、ObjectCycle、OmniConnect、Open ClientConnect、Open ServerConnect、PowerBuilder、PowerDynamo、Powersoft、Quickstart Datamart、Replication Agent、Replication Driver、SQL Anywhere、SQL Central、SQL Remote、Support Plus、SWAT、Sybase IQ、Sybase System 11、Sybase WAREHOUSE、SyBooks、XA-Library は米国法人 Sybase, Inc. の登録商標です。Backup Server、Client-Library、jConnect for JDBC、MainframeConnect、Net-Gateway、Net-Library、Open Client、Open Client/Server、S-Designor、SQL Advantage、SQL Debug、SQL Server、SQL Server Manager、Sybase Central、Watcom、Web.SQL、XP Server は米国法人 Sybase, Inc. の商標です。

ここに記載されている上記以外の社名および製品名は、各社の商標または登録商標場合があります。

目次

	はじめに	vii
	SQL Anywhere Studio のマニュアル	viii
	表記の規則	xii
	Adaptive Server Anywhere サンプル・データベース	xv
	CustDB サンプル・データベース	xvi
	詳細情報の検索／フィードバックの提供	xvii
1	SQL Anywhere Studio バージョン 9 へようこそ	1
	概要	2
	Adaptive Server Anywhere の新機能の概要	5
	Ultra Light の新機能の概要	7
	Mobile Link の新機能の概要	9
	管理ツールの新機能の概要	10
	新機能のサンプル	11
2	バージョン 9.0.2 の新機能	17
	バージョン 9.0.2 の新機能	18
	バージョン 9.0.2 での動作の変更	47
3	バージョン 9.0.1 の新機能	57
	バージョン 9.0.1 の新機能	58
	バージョン 9.0.1 での動作の変更	84
4	バージョン 9.0.0 の新機能	91
	バージョン 9.0.0 の新機能	92
	バージョン 9.0 での動作の変更	125
5	バージョン 8.0.2 の新機能	139
	バージョン 8.0.2 の新機能	140

	バージョン 8.0.2 での動作の変更	156
6	バージョン 8.0.1 の新機能	163
	バージョン 8.0.1 の新機能.....	164
	バージョン 8.0.1 での動作の変更	174
7	バージョン 8.0.0 の新機能	179
	バージョン 8 の新機能	180
	バージョン 8 での動作の変更	217
8	バージョン 7.0.3 の新機能	231
	新しい機能	232
	動作の変更	233
9	バージョン 7.0.2 の新機能	235
	バージョン 7.0.2 の新機能.....	236
	バージョン 7.0.2 での動作の変更	241
10	バージョン 7.0.1 の新機能	243
	バージョン 7.0.1 の新機能.....	244
11	バージョン 7.0.0 の新機能	253
	バージョン 7.0.0 の新機能.....	254
	バージョン 7.0.0 での動作の変更	269
12	バージョン 6.0.3 の新機能	275
	バージョン 6.0.3 の新機能.....	276
	バージョン 6.0.3 での動作の変更	286
13	バージョン 6.0.2 の新機能	289
	バージョン 6.0.2 の新機能.....	290
	バージョン 6.0.2 での動作の変更	294
14	バージョン 6.0.1 の新機能	295
	バージョン 6.0.1 の新機能.....	296
	SQL Remote の新機能	301
	動作の変更	302

15	ソフトウェアとデータベースのアップグレード	305
	Adaptive Server Anywhere のアップグレード	306
	Mobile Link のアップグレード	316
	Ultra Light アプリケーションのアップグレード	321
	SQL Remote のアップグレード	323
16	バージョン 5 アプリケーションのアップグレード	325
	バージョン 5 のアップグレードの知識	326
	SQL Anywhere バージョン 5 アーキテクチャの復習	329
	Embedded SQL アプリケーションのアップグレード	333
	ODBC アプリケーションのアップグレード	340
	互換性ライブラリの使用	347
	dbclient コマンド情報の取得	354
	データベースのアップグレード	356
	バージョン 5 SQL Remote インストール環境のアップグレード	357
	索引	359

はじめに

このマニュアルの内容 このマニュアルでは、SQL Anywhere Studio バージョン 9 とそれ以前のリリースに含まれる新機能について説明します。

対象読者 このマニュアルは、SQL Anywhere Studio の以前のリリースのユーザーで、今回のリリースで新しく追加された事項と変更された事項を知りたい方を対象としています。

SQL Anywhere Studio のマニュアル

このマニュアルは、SQL Anywhere のマニュアル・セットの一部です。この項では、マニュアル・セットに含まれる各マニュアルと使用法について説明します。

SQL Anywhere Studio のマニュアル

SQL Anywhere Studio のマニュアルは、各マニュアルを 1 つの大きなヘルプ・ファイルにまとめたオンライン形式、マニュアル別の PDF ファイル、および有料の製本版マニュアルで提供されます。SQL Anywhere Studio のマニュアルは、次の分冊マニュアルで構成されています。

- **『SQL Anywhere Studio の紹介』** このマニュアルでは、SQL Anywhere Studio のデータベース管理と同期テクノロジーの概要について説明します。また、SQL Anywhere Studio を構成する各部分について説明するチュートリアルも含まれています。
- **『SQL Anywhere Studio 新機能ガイド』** このマニュアルは、SQL Anywhere Studio のこれまでのリリースのユーザを対象としています。ここでは、製品の今回のリリースと以前のリリースで導入された新機能をリストし、アップグレード手順を説明しています。
- **『Adaptive Server Anywhere データベース管理ガイド』** このマニュアルでは、データベースおよびデータベース・サーバの実行、管理、設定について説明しています。
- **『Adaptive Server Anywhere SQL ユーザーズ・ガイド』** このマニュアルでは、データベースの設計と作成の方法、データのインポート・エクスポート・変更の方法、データの検索方法、ストアド・プロシージャとトリガの構築方法について説明します。
- **『Adaptive Server Anywhere SQL リファレンス・マニュアル』** このマニュアルは、Adaptive Server Anywhere で使用する SQL 言語の完全なリファレンスです。また、Adaptive Server Anywhere のシステム・テーブルとシステム・プロシージャについても説明しています。
- **『Adaptive Server Anywhere プログラミング・ガイド』** このマニュアルでは、C、C++、Java プログラミング言語を使用してデータベース・アプリケーションを構築、配備する方法について

て説明します。Visual Basic や PowerBuilder などのツールのユーザは、それらのツールのプログラミング・インタフェースを使用できます。また、Adaptive Server Anywhere ADO.NET データ・プロバイダについても説明します。

- **『Adaptive Server Anywhere SNMP Extension Agent ユーザーズ・ガイド』** このマニュアルでは、Adaptive Server Anywhere SNMP Extension Agent の設定を行い、SNMP 管理アプリケーションとともに使用して Adaptive Server Anywhere データベースを管理する方法について説明します。
- **『Adaptive Server Anywhere エラー・メッセージ』** このマニュアルでは、Adaptive Server Anywhere エラー・メッセージの完全なリストを、その診断情報とともに説明します。
- **『SQL Anywhere Studio セキュリティ・ガイド』** このマニュアルでは、Adaptive Server Anywhere データベースのセキュリティ機能について説明します。Adaptive Server Anywhere 7.0 は、米国政府から TCSEC (Trusted Computer System Evaluation Criteria) の C2 セキュリティ評価を授与されています。このマニュアルには、Adaptive Server Anywhere の現在のバージョンを、C2 基準を満たした環境と同等の方法で実行することを望んでいるユーザにとって役に立つ情報が含まれています。
- **『Mobile Link 管理ガイド』** このマニュアルでは、モバイル・コンピューティング用の Mobile Link データ同期システムについてあらゆる角度から説明します。このシステムによって、Oracle、Sybase、Microsoft、IBM の単一データベースと、Adaptive Server Anywhere や Ultra Light の複数データベースの間でのデータ共有が可能になります。
- **『Mobile Link クライアント』** このマニュアルでは、Adaptive Server Anywhere リモート・データベースと Ultra Light リモート・データベースの設定を行い、これらを同期させる方法について説明します。
- **『Mobile Link サーバ起動同期ユーザーズ・ガイド』** このマニュアルでは、Mobile Link のサーバによって開始される同期について説明します。サーバによって開始される同期とは、統合データベースから同期の開始を可能にする Mobile Link の機能です。

- **『Mobile Link チュートリアル』** このマニュアルでは、Mobile Link アプリケーションの設定と実行の手順を示す各種のチュートリアルについて説明します。
- **『QAnywhere ユーザーズ・ガイド』** このマニュアルでは、Mobile Link QAnywhere について説明します。Mobile Link QAnywhere は、従来のデスクトップ・クライアントやラップトップ・クライアントだけでなく、モバイル・クライアントや無線クライアント用のメッセージング・アプリケーションの開発と展開を可能にするメッセージング・プラットフォームです。
- **『Mobile Link およびリモート・データ・アクセスの ODBC ドライバ』** このマニュアルでは、Mobile Link 同期サーバから、または Adaptive Server Anywhere リモート・データ・アクセスによって、Adaptive Server Anywhere 以外の統合データベースにアクセスするための ODBC ドライバの設定方法について説明します。
- **『SQL Remote ユーザーズ・ガイド』** このマニュアルでは、モバイル・コンピューティング用の SQL Remote データ・レプリケーション・システムについて、あらゆる角度から説明します。このシステムによって、Adaptive Server Anywhere または Adaptive Server Enterprise の単一データベースと Adaptive Server Anywhere の複数データベースの間で、電子メールやファイル転送などの間接的リンクを使用したデータ共有が可能になります。
- **『SQL Anywhere Studio ヘルプ』** このマニュアルには、Sybase Central や Interactive SQL、その他のグラフィカル・ツールに関するコンテキスト別のヘルプが含まれています。これは、製本版マニュアル・セットには含まれていません。
- **『Ultra Light データベース・ユーザーズ・ガイド』** このマニュアルは、Ultra Light 開発者を対象としています。ここでは、Ultra Light データベース・システムの概要について説明します。また、すべての Ultra Light プログラミング・インタフェースに共通する情報を提供します。
- **Ultra Light のインタフェースに関するマニュアル** 各 Ultra Light プログラミング・インタフェースには、それぞれに対応するマニュアルを用意しています。これらのインタフェースは、RAD (

ラピッド・アプリケーション開発)用の Ultra Light コンポーネントとして提供されているものと、C、C++、Java 開発用の静的インタフェースとして提供されているものがあります。

このマニュアル・セットの他に、PowerDesigner と InfoMaker には、独自のオンライン・マニュアル(英語版)がそれぞれ用意されています。

マニュアルの形式

SQL Anywhere Studio のマニュアルは、次の形式で提供されています。

- **オンライン・マニュアル** オンライン・マニュアルには、SQL Anywhere Studio の完全なマニュアルがあり、SQL Anywhere ツールに関する印刷マニュアルとコンテキスト別のヘルプの両方が含まれています。オンライン・マニュアルは、製品のメンテナンス・リリースごとに更新されます。これは、最新の情報を含む最も完全なマニュアルです。

Windows オペレーティング・システムでオンライン・マニュアルにアクセスするには、[スタート] – [プログラム] – [SQL Anywhere 9] – [オンライン・マニュアル] を選択します。オンライン・マニュアルをナビゲートするには、左ウィンドウ枠で HTML ヘルプの目次、索引、検索機能を使用し、右ウィンドウ枠でリンク情報とメニューを使用します。

UNIX オペレーティング・システムでオンライン・マニュアルにアクセスするには、SQL Anywhere のインストール・ディレクトリに保存されている HTML マニュアルを参照してください。

- **PDF 版マニュアル** SQL Anywhere の各マニュアルは、Adobe Acrobat Reader で表示できる PDF ファイルで提供されています。

PDF 版マニュアルは、オンライン・マニュアルまたは Windows の [スタート] メニューから利用できます。

- **製本版マニュアル** 製本版マニュアルをご希望の方は、ご購入いただいた販売代理店または弊社営業担当までご連絡ください。

表記の規則

この項では、このマニュアルで使用されている書体およびグラフィック表現の規則について説明します。

SQL 構文の表記規則

SQL 構文の表記には、次の規則が適用されます。

- **キーワード** SQL キーワードはすべて次の例に示す **ALTER TABLE** のように大文字で表記します。

ALTER TABLE [*owner*.]*table-name*

- **プレースホルダ** 適切な識別子または式で置き換えられる項目は、次の例に示す *owner* や *table-name* のように表記します。

ALTER TABLE [*owner*.]*table-name*

- **繰り返し項目** 繰り返し項目のリストは、次の例に示す *column-constraint* のように、リストの要素の後ろに省略記号 (ピリオド 3 つ ...) を付けて表します。

ADD column-definition [*column-constraint*, ...]

複数の要素を指定できます。複数の要素を指定する場合は、各要素間をカンマで区切る必要があります。

- **オプション部分** 文のオプション部分は角カッコで囲みます。

RELEASE SAVEPOINT [*savepoint-name*]

この例では、角カッコで囲まれた *savepoint-name* がオプション部分です。角カッコは入力しないでください。

- **オプション** 項目リストから 1 つだけ選択するか、何も選択しなくてもよい場合は、項目間を縦線で区切り、リスト全体を角カッコで囲みます。

[**ASC** | **DESC**]

この例では、ASC と DESC のどちらか 1 つを選択しても、どちらも選択しなくてもかまいません。角カッコは入力しないでください。

-
- **選択肢** オプションの中の1つを必ず選択しなければならない場合は、選択肢を中カッコで囲み、縦棒で区切ります。

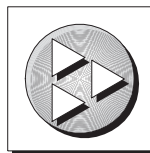
[QUOTES { ON | OFF }]

QUOTES オプションを使用する場合は、ON または OFF のどちらかを選択する必要があります。角カッコと中カッコは入力しないでください。

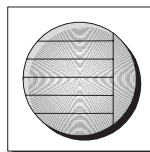
グラフィック・アイコン

このマニュアルでは、次のアイコンを使用します。

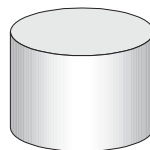
- クライアント・アプリケーション



- Sybase Adaptive Server Anywhere などのデータベース・サーバ



- データベース。高度な図では、データベースとデータベースを管理するデータ・サーバの両方をこのアイコンで表します。



- レプリケーションまたは同期のミドルウェア。ソフトウェアのこれらの部分は、データベース間のデータ共有を支援します。たとえば、Mobile Link 同期サーバ、SQL Remote Message Agent などがあげられます。



- プログラミング・インタフェース



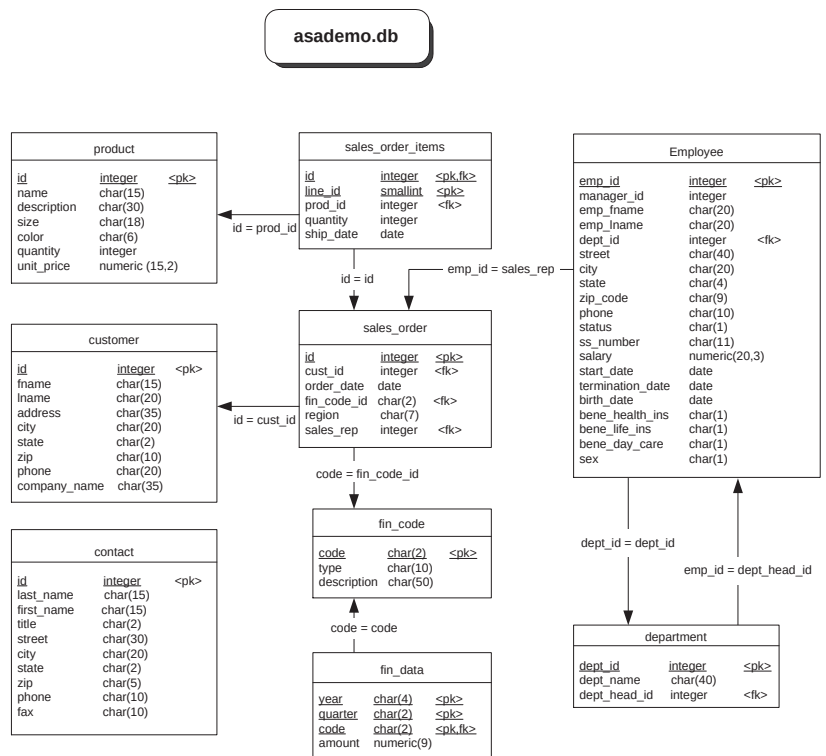
Adaptive Server Anywhere サンプル・データベース

このマニュアルでは、多くの例で Adaptive Server Anywhere サンプル・データベースが使用されています。

サンプル・データベースは、**asademo.db** という名前のファイルに保存され、SQL Anywhere ディレクトリに置かれています。

サンプル・データベースは小規模の企業の例です。データベースには、この企業の内部情報 (従業員、部署、経理) とともに、製品情報や販売情報 (受注、顧客、連絡先) が入っています。データベースに含まれる情報はすべて架空のものです。

次の図は、サンプル・データベース内のテーブルと各テーブル間の関係を示しています。



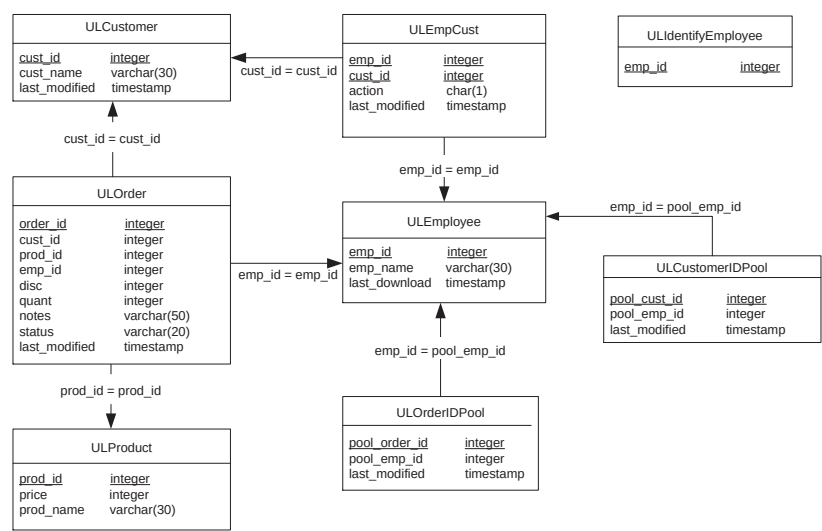
CustDB サンプル・データベース

Mobile Link と Ultra Light のマニュアルでは、多くの例で Ultra Light のサンプル・データベースが使用されています。

Ultra Light サンプル・データベースのリファレンス・データベースは、*custdb.db* という名前のファイルに保存され、SQL Anywhere ディレクトリのサブディレクトリ *Samples¥UltraLite¥CustDB* に置かれています。全面的にこのデータベースを使用して構築したアプリケーションも提供されています。

サンプル・データベースは、あるハードウェア販売会社の販売管理データベースです。データベースには、この販売会社の顧客、製品、営業戦力に関する情報が入っています。

次の図は、CustDB データベース内のテーブルと、各テーブル間の関係を示しています。



詳細情報の検索／フィードバックの提供

詳細情報の検索

詳しい情報やリソース (コード交換など) については、iAnywhere Developer Network (<http://www.iAnywhere.com/developer/>) を参照してください。

ご質問がある場合や支援が必要な場合は、次に示す iAnywhere Solutions ニュースグループのいずれかにメッセージをお寄せください。

ニュースグループにメッセージをお送りいただく際には、ご使用の SQL Anywhere Studio バージョンのビルド番号を明記し、現在発生している問題について詳しくお知らせくださいますようお願いいたします。バージョン情報は、コマンド・プロンプトで **dbeng9 -v** と入力して確認できます。

ニュースグループは、ニュース・サーバ forums.sybase.com にあります (ニュースグループにおけるサービスは英語でのみの提供となります)。以下のニュースグループがあります。

- [sybase.public.sqlanywhere.general](#)
- [sybase.public.sqlanywhere.linux](#)
- [sybase.public.sqlanywhere.mobilink](#)
- [sybase.public.sqlanywhere.product_futures_discussion](#)
- [sybase.public.sqlanywhere.replication](#)
- [sybase.public.sqlanywhere.ultralite](#)
- [iAnywhere.public.sqlanywhere.qanywhere](#)

ニュースグループに関するお断り

iAnywhere Solutions は、ニュースグループ上に解決策、情報、または意見を提供する義務を負うものではありません。また、システム・オペレータ以外のスタッフにこのサービスを監視させて、操作状況や可用性を保証する義務もありません。

iAnywhere Solutions のテクニカル・アドバイザーとその他のスタッフは、時間のある場合にかぎりニュースグループでの支援を行います。こうした支援は基本的にボランティアで行われるため、解決策や情報を定期的に提供できるとはかぎりません。支援できるかどうかは、スタッフの仕事量に左右されます。

フィードバック

このマニュアルに関するご意見、ご提案、フィードバックをお寄せください。

マニュアルに関するご意見、ご提案は、SQL Anywhere ドキュメンテーション・チームの iasdoc@ianywhere.com 宛てに電子メールでお寄せください。このアドレスに送信された電子メールに返信はいたしません。お寄せいただいたご意見、ご提案は必ず読ませていただきます。

マニュアルまたはソフトウェアについてのフィードバックは、上記のニュースグループを通してお寄せいただいてもかまいません。

第 1 章

SQL Anywhere Studio バージョン 9 へようこそ

この章の内容

この章では、SQL Anywhere Studio バージョン 9 の新機能のうち、上位レベルの概要を説明します。

新機能の完全なリストについては、次の項目を参照してください。

- ◆ [「バージョン 9.0.2 の新機能」17 ページ](#)
- ◆ [「バージョン 9.0.1 の新機能」57 ページ](#)
- ◆ [「バージョン 9.0.0 の新機能」91 ページ](#)

概要

SQL Anywhere Studio は、完全なリレーショナル・データベース管理システムです。その主なコンポーネントは次のとおりです。

- **Adaptive Server Anywhere** Adaptive Server Anywhere は、高い信頼性、すぐに使える高性能、幅広い SQL 機能を備えた、自己管理型のリレーショナル・データベースです。Windows、Windows CE、Linux、NetWare、各種 UNIX の環境で使用できます。
- **Ultra Light** Ultra Light は、Palm OS、Pocket PC、Java ベースのデバイスなど、小型デバイス用のデータベース・システムです。Ultra Light データベース管理システムは、制約の多い環境用に特別に設計されたものであり、トランザクション処理のフル・サポート、開発モデルの選択、エンタープライズ・データ・ストアとの同期を実現します。
- **Mobile Link** Mobile Link は、リレーショナル・データベース間で情報を共有するための同期テクノロジーです。物理的に個別のデータベースを 1 つの情報システムに統合し、システム全体にわたってトランザクションの整合性を維持します。
- **SQL Remote** SQL Remote は、特にメッセージ・ベースの同期を目的とする、同期テクノロジーです。ダイヤルアップ・リンク、電子メール、FTP システムを使用してデータベースに接続できます。

SQL Anywhere Studio は、SMB (Small and Medium Business) マーケット、モバイル・コンピューティング、埋め込みシステムで多く利用されています。

- **SMB サーバ** Adaptive Server Anywhere は、セルフチューニングが可能で管理がしやすく、スケーラブルで、セキュリティに優れたデータベース・サーバであるため、SMB データベース・サーバとして適しています。ハンドヘルド・デバイスから、何百もの同時接続によるアクセスを受ける数十ギガバイトのデータベースにいたるまで、あらゆる状況に対応できます。

- **モバイル・コンピューティング** Adaptive Server Anywhere はリソースが限られた環境での動作に優れており、Windows CE デバイスはもちろん、デスクトップやラップトップでの使用にも適しています。Ultra Light は、Windows CE と Palm OS デバイス用の、さらに省スペースなデータベースを提供します。Mobile Link は、複数のリモート・データベースが 1 つの統合システムにまとめられた、スケーラブルでセキュリティに優れた同期システムを提供します。
- **埋め込みシステム** Adaptive Server Anywhere と Ultra Light は自己管理機能を備えているので、埋め込みシステムのようにまったく管理が行われない環境での使用に適しています。Mobile Link の「サーバから開始する同期」の機能 (このリリースの新機能) を使用すると、組み込みデータベースの同期を中央のロケーションから実行できます。

バージョン 9

SQL Anywhere Studio バージョン 9 は、その特徴である使いやすさと自己管理機能を維持する一方で、製品には多数の新機能が組み込まれています。

Web サービスと .NET

SQL Anywhere Studio の今回のリリースには、XML と Web サービスの機能が導入されています。Web サーバは、データベースに直接組み込まれています。柔軟なインタフェースを介して、データを XML として取得したり、XML データをデータベースに保存したりできます。Web サービスとして、Microsoft Visual Studio .NET との統合も提供されています。

バージョン 9 は、Microsoft .NET アーキテクチャを利用しています。XML と Web サービスの機能、Adaptive Server Anywhere 用の ADO.NET プロバイダ、Mobile Link 同期用の .NET スクリプト機能、ハンドヘルド・データベース・アプリケーション開発用の Ultra Light .NET コンポーネントを備えている SQL Anywhere は、.NET で動作するように開発されています。

小型デバイス用アプリケーションの短期開発

Ultra Light .NET コンポーネントは、Ultra Light コンポーネント・パッケージの 1 つで、小型デバイス用の強力なデータベース・アプリケーションの開発を促進します。その他のコンポーネントは、Pocket IE、eMbedded Visual Basic、AppForge MobileVB、AppForge Crossfire、Borland JBuilder など、各種ツールによるアプリケーション短期開発との統合を実現します。

Mobile Link 同期

Mobile Link 同期では、ユーザによる設定が可能な双方向同期を使用して、Ultra Light または Adaptive Server Anywhere データベースをエンタープライズ・システムに連結します。今回のリリースでは、この強化機能が、サーバから開始する同期と、ファイルベースのダウンロードという機能として提供されています。

パフォーマンス

バージョン 9 では、Adaptive Server Anywhere データベースのパフォーマンスがさらに向上しています。キャッシュ管理の向上とその他の内部変更によって、多数のユーザと多くのプロセッサがスケーラビリティを持ち、オプティマイザの革新的なアルゴリズムによって、複雑なクエリを適切に処理できます。さらに、Linux、HP-UX、64 ビット Windows オペレーティング・システムで、ソフトウェアの 64 ビット・バージョンが使用可能であるか、使用が予定されています。

データベースのパフォーマンスを最大限に高めることがぜひとも必要な場合は、グラフィカルなインデックス・コンサルタントを使用することでパフォーマンスを簡単にチューニングできます。これは、再設計された Sybase Central 管理ツールに含まれる 1 つの機能にすぎません。外観も新しくなり、これまで以上に簡単に多くの情報が得られます。また、クエリ・エディタや再設計されたストアド・プロシージャ・デバッガなど、強力な設計ツールと管理ツールのセットが用意されています。

Adaptive Server Anywhere の新機能の概要

次に示すのは、今回のリリースにおける、Adaptive Server Anywhere データベースの新機能の概要です。

- **スケーラビリティとパフォーマンス** Adaptive Server Anywhere の今回のリリースでは、スケーラビリティとパフォーマンスが大幅に進歩しました。キャッシュ管理システムが再設計され、多数のユーザによる同時実行性が大きく向上しています。Adaptive Server Anywhere は、これまでも数百の同時ユーザの環境で動作していましたが、今回のリリースでは、処理できる規模のレベル向上が期待できます。

クエリ処理も、一段と進歩しています。革新的な新しいアルゴリズムによって、以前のリリースでかかっていた時間の何分の一かで、多くのクエリを実行できます。

パフォーマンスの向上を目的とする変更の多くはデータベース・サーバの内部に組み込まれていますが、新しいインデックス・コンサルタントを使用すると、管理者は適切なインデックスを選択して、パフォーマンスを簡単に最適化できるようになります。

- **Web サービス** 今回のリリースでは、データベースに Web サーバが直接組み込まれています。このため、通常のクライアント／サーバ・ネットワーク・プロトコルに加えて、HTTP を介して Adaptive Server Anywhere データベースにアクセスできるようになりました。Adaptive Server Anywhere は、SOAP 要求のサポートを含めて Web サービスを提供します。Microsoft Visual Studio .NET との密接な統合によって、Web サービス・アプリケーションを簡単に開発できます。
- **データベース内の XML** クライアント・アプリケーションは、結果セットを XML として取得したり、XML データを取り出し、それをリレーショナル形式のデータベースに保存したりできるようになりました。開発者は XML データのフォーマットを制御できるため、ほかのシステムとの統合作業が楽になります。

- **.NET 統合** Adaptive Server Anywhere バージョン 9 は、Microsoft .NET アーキテクチャで動作するように作成されています。Web サービスと XML 機能に加えて、ADO .NET プロバイダによって、Visual Studio .NET からのデータ・アクセスが可能です。
- **SQL の豊富な新機能** Adaptive Server Anywhere の SQL 言語が拡張され、多くのデータ操作が可能になりました。ROLLUP クエリなどの単純な OLAP 機能を実行したり、再帰ユニオン・クエリを実行して階層的な「部品表」データを取得したり、INTERSECT や EXCEPT などの集合演算を実行したりできます。

Adaptive Server Anywhere の新機能の完全なリストについては、
「[Adaptive Server Anywhere の新機能](#)」92 ページを参照してください。

Ultra Light の新機能の概要

Ultra Light は、Palm OS や Windows CE が動作している小型デバイス用のリレーショナル・データベースです。このバージョンでは、リレーショナル・データ・ストレージの利点を、より多くのアプリケーション開発者が活用できるようになりました。

- **Ultra Light コンポーネント** コンポーネントを使うと、一般的なプログラミング言語やツールと統合ができます。Ultra Light では、次の開発プラットフォームを対象とするコンポーネントが用意されています。
- **ActiveX** コンポーネントは、Ultra Light 開発を、Microsoft eMbedded Visual Basic と Pocket IE (JavaScript を使用) に統合します。
- **AppForge MobileVB** コンポーネントは、Ultra Light 開発を、Visual Basic の強力な拡張版に統合します。AppForge MobileVB によって、Palm OS と Windows CE デバイスの両方において、Visual Basic を使用したアプリケーションの短期開発が可能になります。
- **.NET** コンポーネントは、C# または Visual Basic による Ultra Light 開発を、.NET Framework と .NET Compact Framework に統合します。
- **Native Java** コンポーネントは、小型デバイス用の Java 開発モデルを提供します。
- **C++ インタフェース** によって、C++ 開発者は、その他の Ultra Light コンポーネントの機能にアクセスできます。これには、複数のアプリケーションからのアクセスをサポートする、Ultra Light ランタイムのバージョンへのアクセスも含まれます。
- **動的 SQL** Ultra Light コンポーネントは、直接的なデータ・アクセスを行うテーブルベースの API を備えていますが、マルチテーブル・ジョインなど、柔軟な操作を行うために動的 SQL もサポートしています。

Ultra Light の新機能の完全なリストについては、「[Ultra Light の新機能](#)」120 ページを参照してください。

Mobile Link の新機能の概要

- **サーバ起動同期** ソフトウェアの以前のリリースでは、クライアントが同期を開始する必要がありました。今回のリリースではサーバから同期を開始できるようになったため、緊急な変更をリモート・データベースに送信することができます。サーバ起動同期は、クライアント・データベースが埋め込みシステムで動作している場合にも適しています。
- **ファイルベースのダウンロード** 同期の更新をファイルとして配布できるようになったため、同期の変更をオフラインで配布できます。
- **より選択的な同期** Adaptive Server Anywhere クライアントは、アップロード専用とダウンロード専用の同期を選択できるようになりました。また、同期処理を制御する新しいスクリプトが追加されました。

Mobile Link の新機能の完全なリストについては、「[Mobile Link の新機能](#)」115 ページを参照してください。

管理ツールの新機能の概要

Sybase Central は、Adaptive Server Anywhere データベースと Mobile Link 統合データベースの管理ツールです。今回のリリースには、多くの新機能が含まれています。

- **インタフェースの再設計** インタフェースが再設計され、必要な情報を簡単に取得できるようになりました。
- **ストアド・プロシージャ・デバッガ** Sybase Central にストアド・プロシージャ・デバッガが組み込まれ、使いやすいように再設計されました。
- **起動の高速化** 起動が高速化され、Sybase Central と Interactive SQL へのアクセスがさらに便利になりました。

管理ツールの新機能の完全なリストについては、「[Adaptive Server Anywhere の新機能](#)」92 ページを参照してください。

新機能のサンプル

以下を始めとするサンプルは、<http://ianywhere.codexchange.sybase.com/servlets/ProjectDocumentList?folderID=0> にあります。

Ultra Light のサンプル

Ultra Light のサンプルは、<http://ianywhere.codexchange.sybase.com/servlets/ProjectDocumentList?folderID=283> にあります。

UL.NET サンプル

このサンプルは、新しい Ultra Light .NET コンポーネントを紹介するものです。このサンプルでは、.USM (Ultra Light スキーマ・ファイル) に基づいて Ultra Light データベースを作成します。スキーマ・ファイルには、テーブル、カラム、インデックスなど、データベース情報が格納されています。Ultra Light データベースのスキーマを直接変更しないでください。代わりに、アプリケーションに組み込まれた Ultra Light 機能を使用してスキーマ・ファイル (通常は拡張子 .USM) を修正し、修正したファイルからデータベース・スキーマをアップグレードします。データベースの作成後、Mobile Link を介してデータがダウンロードされます。最後に、結果セットをリスト・ボックスに表示する SQL クエリが実行されます。

ULFoundation

ULFoundation は、新しい Ultra Light アプリケーションの開発を開始するベース・プラットフォームとして、かつ、学習ツールとして開発されました。このプロジェクトの目的は、Ultra Light に直接関係するアプリケーションの部分を、開発者に理解してもらうことです。プラットフォーム固有の機能は、最小限に抑えられています。コードは、可能なかぎり、プラットフォームに依存しないように記述されています。なお、基本の ULFoundation プログラムへの新機能の追加を通じて操作を説明するモジュールを、新たに開発中です。これは、Ultra Light の動作方法を紹介する効果的なツールであり、さまざまなアプリケーションで使用できます。ULFoundation は、さまざまな方法で設定できます。テキスト・ファイルの指示に従うことをおすすめしますが、その指示に従うことが必須ではありません。

Ultra Light 動的 SQL サンプル

このサンプルは、Ultra Light コンポーネント・パッケージに追加された動的 SQL を紹介するものです。このサンプルの場合、アプリケーションは Ultra Light .NET API を使用して記述されています。サンプルの中で、アプリケーションは、作成済みの Ultra Light データベースと一緒に、CE デバイスに配備されます。このサンプルでは、ユーザは任意の有効な SQL SELECT クエリを入力できます。クエリ結果はリスト・ボックスに表示されます。

Ultra Light エンジン・サンプル

このサンプルでは、2つのアプリケーション (ULapp1 と ULapp2) が、新しいマルチプロセス・アクセス機能を使用して、デバイス上の同じデータベースにアクセスします。最初のアプリケーションはデータベースを作成します。Ultra Light エンジンが起動し、2番目のアプリケーションを開くことができます。2番目のアプリケーションも、新しく作成されたデータベースに接続します。2つのアプリケーションのうち、少なくとも1つが開いているかぎり、エンジンは開いたままです。両方のアプリケーションが閉じると、エンジンは自動的に停止します。

Mobile Link のサンプル

Mobile Link のサンプルは、<http://ianywhere.codexchange.sybase.com/servlets/ProjectDocumentList?folderID=319> にあります。

.NET リモータイング・サンプル

.NET リモータイングを使用すると、アプリケーションのコンポーネントが1台のコンピュータにすべてある場合と世界中に分散している場合のいずれでも、広範囲な分散型アプリケーションを簡単に作成できます。また、同じコンピュータ上の別のプロセス内のオブジェクトや、ネットワークを経由してアクセス可能な別のコンピュータ上のプロセスのオブジェクトを使用する、クライアント・アプリケーションを作成できます。.NET リモータイングを使用すると、同じプロセス内の別のアプリケーション・ドメインとやり取りすることもできます。

詳細については、<http://msdn.microsoft.com/library/default.asp?url=/library> を参照してください。

次の Visual Basic.NET と C# のサンプルでは、Mobile Link を使用して、.NET リモータイング・サービスを起動します。`server.cs/servclass.cs` ファイルは、単純なりモータイング・サービスを作成します。この

サービスは、1095 にローカル・ポートを開き、要求を待機します。リモーディング・サービスのメソッドが起動されると、リモーディング・サーバはコンソールにメッセージを通知します。

ASP.NET サンプル

ASP.NET は、Web アプリケーションと分散型アプリケーションの作成、配備、実行を目的とする、.NET Framework のコンポーネントです。ASP.NET は、.ASMX ファイルの使用を介した Web サービスのサポートを提供します。Web サービスは、メッセージの交換によってコンピュータの連動を実現するためのプロトコルです。Web サービスは、プラットフォームやプログラミング言語にまたがって相互運用するための XML、SOAP、WSDL などの標準プロトコルに基づいています。

このサンプルは、Web サービスの起動を介して、Mobile Link と ASP.NET 間の統合を行います。このサンプルの中で、ファイル *sample.asmx* は、IIS home/root ディレクトリ (*c:\inetpub\wwwroot* など) にコピーされるサンプルの Web サービスです。同期の際、Mobile Link サーバは ASP.NET の起動を試みます。次に、ASP.NET が IIS home/root ディレクトリ内のファイル名 *data.bin* への書き込みを試みます。

ファイル同期サンプル

このサンプルは、リモート・データベースに適用される変更のリストから成るファイルを送信する、Mobile Link サーバの機能を紹介するものです。リモート・データベースはこのファイルを受け取り、その内容を適用し、変更が適用されたことの確認を求めます。

Adaptive Server Anywhere のサンプル

Adaptive Server Anywhere のサンプルは、<http://ianywhere.codexchange.sybase.com/servlets/ProjectDocumentList?folderID=282> にあります。

ADO.NET - Simple XML サンプル

Simple XML サンプルは、ADO.NET を介して Adaptive Server Anywhere から XML データを取得するサンプル・コードを提供します。

ASP.NET チュートリアル

このチュートリアルは、ADO.NET を使用する Adaptive Server Anywhere 9.0 データベースに、ASP.NET を使用してアクセスする方法の概要を理解することが目的です。この例では、Visual Basic.NET と

共に、Visual Studio.NET 2002 または 2003 を開発ツールとして使用できます。このチュートリアルでは、ブラウザの中で Adaptive Server Anywhere データベースに対してクエリを実行し、結果をリスト・ボックスに返す、サンプルの ASP.NET アプリケーションを作成するための手順を、すべて説明します。

HTTP サーバ - Show Table サンプル

この Show Table サンプルは、次のようなデータベース情報を表示する、Web ベースの管理ツールの例です。

- テーブル・リスト
- テーブルの内容
- 使用できる Web サービス
- データベースの履歴

このサンプルは、XML スタイル・シートを使用して HTML と XML データに対処する、HTTP サーバの機能を紹介するものです。デモンストレーションの中で、2 つの Adaptive Server Anywhere Web サービスが作成されます。

1. `show_html_table`
2. `show_xml_table`

HTTP サーバ - 単純 選択サンプル

このサンプルでは、SQL クエリを受け取り、結果セットを HTML または XML フォーマットで返す Web サービスを、Adaptive Server Anywhere の中で簡単に作成できることを示します。このサービスを作成するために実行される SQL コマンドには、次のようなものがあります。

- `CREATE SERVICE qhtml TYPE 'html' AUTHORIZATION ON USER DBA SECURE OFF`
- `CREATE SERVICE qxml TYPE 'xml' AUTHORIZATION ON USER DBA SECURE OFF`

ユーザが URL (`http://localhost/websample/qhtml?select * from sysusers` など) を入力すると、Adaptive Server Anywhere Web サーバは URL を受け取り、文字列を解析します。

クエリは実行され、qhtml サービスが指定するように、結果は HTML フォーマットで返されます。

HTTP サーバ - SOAP データセッ ト・サンプル

このサンプルでは、Adaptive Server Anywhere から SOAP 要求として返されたデータを、.NET DataGrid コンポーネントの中で簡単に表示できることを示します。SOAP は、非集中型、つまり分散型環境の中で情報を交換するための、コンパクトなプロトコルです。XML ベースのプロトコルであり、次の 3 つの部分で構成されます。

- メッセージの内容とその処理方法を記述するフレームワークを定義するエンベロープ
- アプリケーションで定義されたデータ型のインスタンスを表現するためのエンコーディング・ルールのセット
- リモート・プロシージャの呼び出しと応答を表す規則

SOAP は、ほかのさまざまなプロトコルと組み合わせて使用できます。ただし、このマニュアルでは、HTTP と HTTP 拡張フレームとの組み合わせの場合にかぎって、SOAP の使い方を説明します。

DataGrid は、Visual Studio.NET 2003 の中で使用できる、一般的なデータ・バウンド・リスト・コントロールです。DataGrid は、データ・ソースの項目をテーブル形式で表示します。これらの項目は、選択、ソート、編集することができます。アプリケーションからの SOAP 要求は Adaptive Server Anywhere 9.0 のサンプル・データベースに接続し、結果セットが .NET DataSet オブジェクトとして返されます。次に、DataSet が DataGrid コンポーネントの DataSource として使用されます。ソース・コードを使用して始めからデモンストレーションを実行するには、ファイル *Walkthrough.doc* を参照してください。

第 2 章

バージョン 9.0.2 の新機能

この章の内容

この章では、SQL Anywhere Studio バージョン 9.0.2 で導入された新機能と変更された動作の概要について説明します。

バージョン 9.0.2 の新機能

この項では、SQL Anywhere Studio バージョン 9.0.2 のコンポーネントに導入された新機能について説明します。

Adaptive Server Anywhere の新機能

この項では、Adaptive Server Anywhere バージョン 9.0.2 の新機能について説明します。新機能について主要なものからそうでないものまですべてを示し、各機能の詳細についての参照先も記述しています。

SQL の強化

- **ネイティブの UNIQUEIDENTIFIER データ型**

UNIQUEIDENTIFIER データ型が、BINARY(16) に基づいて定義されるドメインではなく、ネイティブのデータ型になりました。この結果、Adaptive Server Anywhere では、必要に応じて型変換が自動的に行われるため、STRTOUUID 変換関数と UUIDTOSTR 変換関数を使用して UNIQUEIDENTIFIER 値を処理する必要はありません。

このリリースより前に作成されたデータベースで UNIQUEIDENTIFIER データ型を使用するには、データベースをアンロードして再ロードすることによってそのデータベースのファイル・フォーマットをアップグレードする必要があります。

詳細については、『ASA SQL リファレンス・マニュアル』>「UNIQUEIDENTIFIER データ型 [バイナリ]」を参照してください。

- **RESOLVE UPDATE トリガの CONFLICT 関数** 競合解決トリガの中で CONFLICT 関数を使用して、SQL Remote 統合データベースに対して実行した UPDATE において、競合の原因が特定のカラムであるかどうかを判断できるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』>「CONFLICT 関数 [その他]」を参照してください。

- **プロシージャ・プロファイリングの強化** プロファイリング情報が、sa_server_option ストアド・プロシージャを使用して、ユーザ単位と接続単位でフィルタできるようになりました。

詳細については、『ASA SQL ユーザーズ・ガイド』>「プロシージャ・プロファイリングの有効化」と『ASA SQL リファレンス・マニュアル』>「sa_server_option システム・プロシージャ」を参照してください。

- **作成または変更前にリモート・サーバのテストが可能** Sybase Central の[リモート・サーバ作成]ウィザードに[テスト接続]ボタンが追加され、リモート・サーバを作成する前に、リモート・サーバ定義に指定されている接続情報を使用して正しく接続できるかどうかをテストできるようになりました。

Sybase Central の[リモート・サーバ]プロパティ・シートにも[テスト接続]ボタンが追加され、プロパティを変更したときにリモート・サーバに正しく接続できるかどうかをテストできます。

詳細については、『ASA SQL ユーザーズ・ガイド』>「Sybase Central を使用したリモート・サーバの作成」を参照してください。

- **INPUT 文と OUTPUT 文に ESCAPES 句を指定できる** ESCAPES 句を使用すると、データベース・サーバが文字を特殊文字として認識し、解釈することを指定できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「INPUT 文 [Interactive SQL]」と『ASA SQL リファレンス・マニュアル』>「OUTPUT 文 [Interactive SQL]」を参照してください。

- **別の接続からメッセージを受信して WAITFOR がウェイクアップできる** WAITFOR 文は、MESSAGE 文を使用して別の接続からメッセージを受信するときに、ウェイクアップできるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』>「WAITFOR 文」を参照してください。

- **抽出テーブルがクエリ・プランに表示される** 抽出テーブルが、クエリ実行プランにノードとして表示されるようになりました。

詳細については、『ASA SQL ユーザーズ・ガイド』>「抽出テーブル」を参照してください。

- **ALTER DOMAIN 文** ALTER DOMAIN 文を使用して、ユーザー定義のドメインとデータ型の名前を変更できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「ALTER DOMAIN 文」を参照してください。

- **プロシージャの NO RESULT SET 句** 外部環境が、ストアード・プロシージャが結果セットを返さないことを理解する必要がある場合、ストアード・プロシージャの NO RESULT SET の宣言を使用することができます。

詳細については、『ASA SQL リファレンス・マニュアル』>「CREATE PROCEDURE 文」を参照してください。

- **インデックス作成時にカラム統計が更新される** CREATE INDEX 文には、インデックス付けされたカラムの、カラム統計が更新されるという副次的な効果があります。

詳細については、『ASA SQL リファレンス・マニュアル』>「CREATE INDEX 文」を参照してください。

プログラミング・インタフェースの強化

- **PHP モジュール** SQLAnywhere PHP モジュールを使用すると、PHP スクリプト言語から Adaptive Server Anywhere データベースにアクセスできます。

詳細については、『ASA プログラミング・ガイド』>「SQLAnywhere PHP モジュールの使用」を参照してください。

- **Web サービス・クライアント** Adaptive Server Anywhere は、Web サービス・プロバイダとして動作だけでなく、Web サービス・クライアントとしても動作できるようになりました。このため、インターネット上で使用できる標準の

Web サービスに加えて、Adaptive Server Anywhere Web サービスにアクセスするストアド・プロシージャとストアド関数を作成できます。

詳細については、『ASA データベース管理ガイド』> 「Web サービス・クライアント関数とプロシージャの作成」を参照してください。

- **複数の Web サービス・フォーマットのサポート** DISH サービスで提供される WSDL ファイルのフォーマットと、SOAP 応答の一部として返されるデータ・ペイロードのフォーマットを、クライアント・アプリケーションに合わせて選択できるようになりました。Microsoft .NET 用の DNET、自動的にインタフェースを生成するクライアント用の CONCRETE、汎用的な XML フォーマットの中から選択できます。

詳細については、『ASA データベース管理ガイド』> 「SOAP および DISH Web サービスの作成」を参照してください。

- **ODBC_DESCRIBE_BINARY_AS_VARBINARY オプション**

このオプションを使用すると、すべての BINARY カラムと VARBINARY カラムを、BINARY としてまたは VARBINARY として、アプリケーションに対して記述するかどうかを選択できます。

詳細については、『ASA データベース管理ガイド』> 「ODBC_DESCRIBE_BINARY_AS_VARBINARY [データベース]」を参照してください。

- **新しい PREFETCH オプション値** PREFETCH オプションに、ALWAYS という新しい値が追加されました。この値は、SENSITIVE カーソル・タイプと、プロキシ・テーブルを含むカーソルであっても、カーソルの結果がプリフェッチされることを意味します。

詳細については、『ASA データベース管理ガイド』> 「PREFETCH オプション [データベース]」を参照してください。

- **db_locate_servers_ex 関数** この関数を使用すると、特定のホスト上のすべての Adaptive Server Anywhere データベース・サーバをリストする、dblocate -n オプションで表示される情報に、プログラムからアクセスできます。

詳細については、『ASA プログラミング・ガイド』> 「db_locate_servers_ex 関数」を参照してください。

管理の強化

- **SNMP エージェント** SNMP (Simple Network Management Protocol) アプリケーションから、Adaptive Server Anywhere をモニタできるようになりました。

詳細については、『ASA SNMP Extension Agent ユーザーズ・ガイド』> 「ASA SNMP Extension Agent ユーザーズ・ガイド」を参照してください。

- **デッドロックのレポート** LOG_DEADLOCKS という新しいデータベース・オプションと、sa_report_deadlocks という新しいシステム・ストアド・プロシージャを使用して、デッドロックに関係する接続についての情報を取得できるようになりました。LOG_DEADLOCKS オプションをオンにすると、データベース・サーバは、デッドロックに関する情報を内部バッファに記録します。sa_report_deadlocks を呼び出すことによって、この内部バッファからデッドロック情報を取得できます。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「ブロックされているユーザの判別」を参照してください。

- **新しい照合** 今回のリリースには、次の照合が追加されています。
 - **1252SWEFIN** スウェーデン語とフィンランド語をサポートします。スウェーデン語とフィンランド語のシステムで照合が指定されていない場合、データベース・サーバは、新規データベースのデフォルトの照合として 1252SWEFIN を選択します。

- **1255HEB** ヘブライ語をサポートします。ヘブライ語の Windows システムで照合が指定されていない場合、データベース・サーバは、新規データベースのデフォルトの照合として 1255HEB を選択します。
- **1256ARA** アラビア語をサポートします。アラビア語の Windows システムで照合が指定されていない場合、データベース・サーバは、新規データベースのデフォルトの照合として 1255HEB を選択します。
- **950ZHO_HK、950ZHO_TW** 中国語をサポートします。950ZHO_HK は、Windows 中国語 (繁体字) 文字セット cp950 に加えて、香港補足文字セット (HKSCS) をサポートします。950ZHO_TW は、Windows 中国語 (繁体字) 文字セット cp950 はサポートしますが、HKSCS はサポートしません。順序は、中国語 (繁体字) のバイト単位の順序に基づきます。これらの照合が旧式の 950TWN 照合に優先します。
- **1252SPA** スペイン語をサポートします。スペイン語の Windows システムで照合が指定されていない場合、データベース・サーバは、新規データベースのデフォルトの照合として 1252SPA を選択します。
- **874THAIBIN** タイ語をサポートします。Windows システムと UNIX システムの両方で、タイ語の照合として、この照合の使用が推奨されます。

詳細については、『ASA データベース管理ガイド』> 「提供されている照合と推奨する照合」を参照してください。

- **サービス作成 (dbsvc) ユーティリティの新しいオプション**

サービス作成ユーティリティ (dbsvc) では、次のオプションが新しくサポートされるようになりました。

- **-cm オプション** このオプションは、指定されたサービスを作成するときに使用されたコマンドを表示します。これは、サービスを配備したり、元の状態に復元したりするときに便利です。

- **-sd オプション** このオプションを使用すると、Windows サービス・マネージャに表示される、サービスの説明を指定できます。
- **-sn オプション** このオプションを使用すると、Windows サービス・マネージャに表示される、サービスの名前を指定できます。

詳細については、『ASA データベース管理ガイド』> 「サービス作成ユーティリティのオプション」を参照してください。

- **データ・ソース (dbdsn) ユーティリティの新しいオプション**

データ・ソース・ユーティリティ (dbdsn) では、次のオプションが新しくサポートされるようになりました。

- **-cm オプション** このオプションは、指定されたデータ・ソースを作成するときに使用されたコマンドを表示します。これは、データ・ソースを配備したり、元の状態に復元したりするときに便利です。
- **Driver 接続パラメータ** Windows でデータ・ソース・ユーティリティ (dbdsn) を使用してデータベース・ソースを作成する場合、Driver 接続パラメータを使用して、ODBC データ・ソースのドライバを指定できます。UNIX の場合は、Driver 接続パラメータを指定しないと、ASANY9 環境変数の設定に基づく Adaptive Server Anywhere ODBC ドライバのフル・パスを使用して、Driver エントリが自動的に追加されます。

詳細については、『ASA データベース管理ガイド』> 「データ・ソース・ユーティリティ オプション」を参照してください。

- **ディスクが満杯の場合のコールバック・サポート** -fc データベース・サーバ・オプションを使用すると、ファイル・システムが満杯の状態になったときに、ユーザへの通知を行い、おそらく適切に対処できるコールバック関数が含まれる DLL を指定することができます。

詳細については、『ASA データベース管理ガイド』> 「-fc サーバ・オプション」を参照してください。

- **[データベース検証] ウィザードの強化** Sybase Central で [データベース検証] ウィザードを使用してデータベースを検証する場合、ウィザードには、検証処理全体の進行状況に加えて、検証中の現在のテーブルも表示されるようになりました。また、チェックサムが有効なデータベースの場合は、テーブルとチェックサムを同時に検証できます。

詳細については、『ASA データベース管理ガイド』> 「データベースの検証」を参照してください。

- **Sybase Central でのテーブル・データのアンロード** [データのアンロード] ダイアログを使用して、Sybase Central 内の複数のテーブルから、1 回の操作でデータをアンロードできるようになりました。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「[データのアンロード] ダイアログの使用」を参照してください。

- **sa_index_density と sa_index_levels への新しいカラムの追加**

sa_index_density および sa_index_levels ストアド・プロシージャから返される結果セットに、TableId、IndexId、IndexType という 3 つの新しいカラムが追加されました。これらのストアド・プロシージャの動作を以前の動作に戻すには、ストアド・プロシージャをいったん削除し、以前のバージョンのソフトウェアの結果セットに含まれていたカラムを指定して、ストアド・プロシージャを再作成します。

詳細については、『ASA SQL リファレンス・マニュアル』> 「sa_index_density システム・プロシージャ」と『ASA SQL リファレンス・マニュアル』> 「sa_index_levels システム・プロシージャ」を参照してください。

- **BACKUP 文と RESTORE DATABASE 文の HISTORY オプション** HISTORY オプションを使用すると、BACKUP 文と RESTORE DATABASE 文による処理を *backup.syb* ファイルに記録するかどうかを制御できます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「BACKUP 文」と『ASA SQL リファレンス・マニュアル』> 「RESTORE DATABASE 文」を参照してください。

- **Windows ユーザ・グループを使用する統合化ログインのサポート** Windows NT/2000/XP 上の個々のユーザに対して統合化ログインを作成できることに加えて、Windows NT/2000/XP 上のユーザ・グループに対しても統合化ログイン・マッピングを作成できるようになりました。データベースをアップグレードしてからこの機能を使用することをおすすめします。

詳細については、『ASA データベース管理ガイド』> 「Windows ユーザ・グループ用の統合化ログインの作成」を参照してください。

- **要求ログのサイズの管理** -zn データベース・サーバ・オプションを使用すると、保持する要求ログ・ファイル数を指定できます。

詳細については、『ASA データベース管理ガイド』> 「-zn サーバ・オプション」を参照してください。

- **バックアップによってファイルの名前が変更されるときにトランザクション・ログの末尾の空きページが削除される** トランザクション・ログ・ファイルは、パフォーマンスを向上するために固定サイズで増分します。バックアップの一環でトランザクション・ログの名前が変更される場合、ログの末尾にある空きページが削除され、ディスク領域が解放されます。
- **リモート・サーバ接続を明示的に閉じることが可能** 以前のリリースでは、Adaptive Server Anywhere からリモート・サーバへの接続は、ユーザと Adaptive Server Anywhere との接続が切断された場合のみ、切断されていました。ALTER SERVER 文の新しい CONNECTION CLOSE 句を使用して、Adaptive Server Anywhere とリモート・サーバとの接続を明示的に切断できるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「ALTER SERVER 文」を参照してください。

セキュリティの強化

- **初期化ファイルを dbfhide で難読化できる** ファイル非表示ユーティリティ (dbfhide) を使用して、Adaptive Server Anywhere とそのユーティリティが使用する .ini ファイルを難読化できるようになりました。

詳細については、『ASA データベース管理ガイド』> 「.ini ファイルの内容の非表示」を参照してください。

- **FIPS 承認セキュリティ** Windows CE を除く、サポートされているすべての Windows プラットフォームで、Certicom の FIPS 140-2 承認ソフトウェアで保護された、安全な通信を使用できるようになりました。

詳細については、『SQL Anywhere Studio セキュリティ・ガイド』> 「トランスポート・レイヤ・セキュリティを使用するデータベース・サーバの起動」を参照してください。

サポートされている 32 ビット Windows プラットフォームでは、Certicom の FIPS 140-2 承認ソフトウェアを使用した、強力なデータベース暗号化も使用できます。

詳細については、『SQL Anywhere Studio セキュリティ・ガイド』> 「データベースの暗号化」を参照してください。

その他の機能強化

- **新しい接続プロパティ** 次の接続プロパティが追加されました。
 - ClientPort
 - LoginTime
 - ServerPort

詳細については、『ASA データベース管理ガイド』> 「接続レベルのプロパティ」を参照してください。

- **イベント・ビューア・メッセージの適切なフォーマット**

Adaptive Server Anywhere データベースを配備する場合は、イベント・ビューアのメッセージのフォーマットを制御するレジストリ・エントリを設定する必要があります。

詳細については、『ASA プログラミング・ガイド』> 「データベース・サーバの配備」を参照してください。

- **LOG_DEADLOCKS オプション** このオプションを使用すると、データベース・サーバが、デッドロックに関する情報を内部バッファに記録するかどうかを制御できます。このオプションを `sa_report_deadlocks` プロシージャで使用すると、デッドロックに関する情報を取得できます。

詳細については、『ASA データベース管理ガイド』>
「LOG_DEADLOCKS オプション [データベース]」を参照してください。

- **ROLLBACK_ON_DEADLOCK オプション** このオプションを使用すると、デッドロックが発生したときに、トランザクションを自動的にロールバックするかどうかを制御できます。

詳細については、『ASA データベース管理ガイド』>
「ROLLBACK_ON_DEADLOCK [データベース]」を参照してください。

- **TEMP_SPACE_LIMIT_CHECK オプション** このオプションを使用すると、接続がテンポラリ・ファイル領域の割り当てを越えて要求したときの動作を制御できます。

詳細については、『ASA データベース管理ガイド』>
「TEMP_SPACE_LIMIT_CHECK オプション [データベース]」を参照してください。

- **新しいシステム・ストアド・プロシージャ** 各種の新しいシステム・ストアド・プロシージャが追加されました。

- **sa_rowgenerator プロシージャ** `sa_rowgenerator` システム・プロシージャは、RowGenerator テーブルの代替として提供されており、指定された開始値と終了値の間のローから成る結果セットを返します。

このプロシージャを使用すると、範囲内の各値に対応するローから成る結果セットを生成したり、結果セット内の既知の行数に対してテスト・データを生成したりできます。

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

- **sa_send_udp ストアド・プロシージャ** このプロシージャは、指定されたアドレスに UDP パケットを送信し、Mobile Link サーバ起動同期で使用することで、リスナ・ユーティリティ (*dblsn.exe*) をウェイクアップできます。

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

- **sa_verify_password ストアド・プロシージャ** このプロシージャは、現在のユーザのパスワードを確認するために、sp_password ストアド・プロシージャによって使用されます。

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

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

- **Windows CE の最大キャッシュ・サイズ** SQL Anywhere Studio の以前のリリースでは、Windows CE の最大キャッシュ・サイズは 32 MB でした。この制限が取り除かれて、キャッシュ・サイズは、デバイス上で使用可能なメモリ量によって制限されるようになりました。
- **UNIX 用の新しいデータベース・サーバ・オプション** UNIX 用に次のデータベース・サーバ・オプションが追加されました。
- **-uc** UNIX 上でデータベース・サーバをコンソール・モードで起動します。

詳細については、『ASA データベース管理ガイド』> 「-uc サーバ・オプション」を参照してください。

- **-ui X Windows** がサポートされている Linux と Solaris でデータベース・サーバを起動する場合に、[サーバ起動オプション] ダイアログとサーバ・メッセージ・ウィンドウの表示を試みます。使用可能な表示を見つけれない場合は、サーバはコンソール・モードで起動します。

詳細については、『ASA データベース管理ガイド』> 「-ui サーバ・オプション」を参照してください。

- **-ux X Windows** がサポートされている Linux と Solaris でデータベース・サーバを起動する場合に、[サーバ起動オプション] ダイアログとサーバ・メッセージ・ウィンドウを表示します。

詳細については、『ASA データベース管理ガイド』> 「-ux サーバ・オプション」を参照してください。

Mobile Link の新機能

次に、バージョン 9.0.2 で導入したソフトウェアに加えられた変更と追加を示します。

- **新しいリダイレクタ** Apache 用の新しいネイティブなリダイレクタが用意されています。これに加えて、M-Business Anywhere リダイレクタも用意されています。両方とも、Windows、Solaris、Linux で使用できます。

詳細については、『Mobile Link 管理ガイド』> 「Apache リダイレクタ」と『Mobile Link 管理ガイド』> 「M-Business Anywhere リダイレクタ」を参照してください。

- **指定のホストを無視するようにプロトコルを設定可能** 新しい ignore オプションを使用すると、Mobile Link 同期サーバに接続するときに、サーバによって無視されるホストを指定できます。

『Mobile Link 管理ガイド』> 「-x オプション」の ignore を参照してください。

- **Mobile Link サーバがビジーな場合のクライアントの同期待機の回避** サーバがビジーな場合に、クライアントが同期を待ち続けることを回避できるようになりました。

詳細については、『Mobile Link 管理ガイド』> 「-x オプション」の **backlog** を参照してください。

- **統合データベースに格納されるバージョン** SQL Anywhere Studio のバージョンとビルド番号が、Mobile Link システム・テーブル `ml_property` に格納されるようになりました。エンタリは、`component_name` が **ML**、`property_set_name` が **server_info**、`property_name` が **release_version**、`property_value` が **version.build** の形式で、9.0.2.1234 などがその例です。

Mobile Link システム・テーブルの詳細については、『Mobile Link 管理ガイド』> 「ml_property」を参照してください。

- **Mobile Link 同期サーバがサポートする新しい uniqueidentifier データ型** UNIQUEIDENTIFIER データ型が、BINARY(16) に基づいて定義されるドメインではなく、ネイティブのデータ型になりました。この結果、Mobile Link リモート・データベースでは、必要に応じて型変換が自動的に行われるため、文字列から UUID への変換関数と UUID から文字列への変換関数を使用して UNIQUEIDENTIFIER 値を処理する必要はありません。

サポートされている統合データベースへのこのデータ型のマッピングについては、『Mobile Link 管理ガイド』> 「データ型の変換」を参照してください。

セキュリティの強化

- **FIPS 承認セキュリティ・ストリーム** Windows デバイスで、Certicom の FIPS 140-2 承認ソフトウェアが適用された安全な通信を使用できるようになりました。

詳細については、『Mobile Link 管理ガイド』> 「トランスポート・レイヤ・セキュリティを使用する Mobile Link 同期サーバの起動」を参照してください。

- **出力ログ内での接続オプションの表示** 接続文字列と接続オプションが、出力ログに表示されるようになりました。このとき、パスワードはアスタリスクで表示されます。

Mobile Link クライアントの強化

- **旧式のセキュリティ機能「Mobile Link の動作の変更」51 ページ**を参照してください。
- **Ultra Light 用の新しい同期設定ツール** Ultra Light Schema Painter が、Adaptive Server Anywhere 統合データベース用のデータベース・テーブルとトリガに加えて、Mobile Link 同期スクリプトも生成できるようになりました。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「Ultra Light スキーマ・ペインタ」と『Ultra Light データベース・ユーザーズ・ガイド』> 「チュートリアル：Ultra Light データベースの使用」を参照してください。
- **リモート・データベースの削除と再作成の簡単化** Adaptive Server Anywhere クライアント・サブスクリプションの最初の同期が、いつでも動作するようになりました。

詳細については、『Mobile Link クライアント』> 「進行オフセット」を参照してください。
- **Mobile Link への接続が失敗すると新しい dbmlsync フックが呼び出される** 新しく追加された `sp_hook_dbmlsync_connect_failed` イベント・フックを使用すると、同期接続が失敗した場合のリカバリの方法をプログラムできます。

詳細については、『Mobile Link クライアント』> 「`sp_hook_dbmlsync_connect_failed`」を参照してください。
- **Mobile Link クライアントと HTTP インフラストラクチャの統合の向上** プロキシ・サーバか Web サーバまたはその両方が、RFC 2617 Basic または Digest 認証を要求する場合に、HTTP を使用して同期できるようになりました。

次の項目を参照してください。

- ◆ 『Mobile Link クライアント』> 「`http_password`」
- ◆ 『Mobile Link クライアント』> 「`http_userid`」
- ◆ 『Mobile Link クライアント』> 「`http_proxy_password`」
- ◆ 『Mobile Link クライアント』> 「`http_proxy_userid`」

また、2つの新しいクライアント接続パラメータを使用して、カスタム HTTP ヘッダとカスタム cookie を指定できるようになりました。セッション cookie に対処するために、HTTP クライアントは、サーバ応答で受信するすべての Set-Cookie および Set-Cookie2 HTTP ヘッダを認識し、これらの cookie を、この後のすべての HTTP 要求と共に送り返します。cookie の名前が既存の cookie に一致する場合、クライアントは、古い値を新しい値に置換します。同期と同期の間で cookie は記憶されません。同期の最後に破棄されます。

詳細については、『Mobile Link クライアント』>
「custom_header」と『Mobile Link クライアント』>
「set_cookie」を参照してください。

- **接続エラーの検出の支援** Mobile Link クライアントは、無効な接続パラメータが指定された場合に警告メッセージを出力するようになりました。
- **ミラー・ログのロケーション** dbmsync が、リモート・データベースとは異なるマシンで実行している場合、または、ミラー・ログが、ミラー・トランザクション・ログとは異なるディレクトリにある場合、この新しい拡張オプションを使用して古いミラー・ログのロケーションを指定すると、dbmsync は、古いログ・ファイルを自動的に削除できます。

詳細については、『Mobile Link クライアント』>
「MirrorLogDirectory (mld) 拡張オプション」を参照してください。

サーバ起動同期の強化

- **接続起動同期機能の強化** Windows Listener は、_BEST_IP_CHANGED_ に加えて、内部メッセージ _IP_CHANGED_ も生成するため、接続に変更があった場合に同期を起動できるようになりました。

詳細については、『Mobile Link サーバ起動同期ユーザーズ・ガイド』> 「接続起動同期」を参照してください。

- **Listener ポスト・アクションの強化** Listener ポスト・アクションを指定する場合、オプションで Windows メッセージ ID を使用して、ウィンドウ・メッセージを指定できるようになりました。また、ウィンドウ・クラスの代わりにウィンド

ウ・タイトルをオプションで使用できるようになりました。メッセージやタイトルに、スペースや句読点などの英数字以外の文字が含まれる場合は、ウィンドウ・クラス名やメッセージを単一引用符で囲むこともできます。

詳細については、『Mobile Link サーバ起動同期ユーザーズ・ガイド』> 「Listener ユーティリティ」の **post** を参照してください。

- **新しいアクション変数** 各種の新しいアクション変数が追加されました。
 - `$request_id`
 - `$best_ip`
 - `$best_adapter_name`
 - `$best_adapter_mac`
 - `$best_network_name`

詳細については、『Mobile Link サーバ起動同期ユーザーズ・ガイド』> 「action 変数」を参照してください。

- **デバイス・サポートの増加** Palm Listener は、Kyocera 7135 Smartphone と Treo 600 Smartphone をサポートするようになりました。

詳細については、『Mobile Link サーバ起動同期ユーザーズ・ガイド』> 「Palm デバイス用 Listener」を参照してください。

SQL Remote の新機能

次に、バージョン 9.0.2 で導入したソフトウェアに加えられた変更と追加を示します。

- **ミラー・ログのロケーション** dbremote が、リモート・データベースとは異なるマシンで実行している場合、または、ミラー・ログが、ミラー・トランザクション・ログとは異なる

ディレクトリにある場合、新しい `-ml` オプションを使用して古いミラー・ログのロケーションを指定すると、`dbremote` は、古いログ・ファイルを自動的に削除できます。

詳細については、『SQL Remote ユーザーズ・ガイド』>「Message Agent」を参照してください。

- **RESOLVE UPDATE トリガの CONFLICT 関数** 競合解決トリガの中で `CONFLICT` 関数を使用して、SQL Remote 統合データベースに対して実行した `UPDATE` において、競合の原因が特定のカラムであるかどうかを判断できるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』>「`CONFLICT` 関数 [その他]」を参照してください。

Ultra Light の新機能

次に、バージョン 9.0.2 で導入したソフトウェアに加えられた変更と追加を示します。

コンポーネントの新しい機能

- **Ultra Light.NET の ADO.NET インタフェース** Ultra Light.NET は、新しい `iAnywhere.Data.UltraLite` ネームスペースで、ADO.NET プログラミング・インタフェースをサポートするようになりました。ADO.NET は、Ultra Light に対して業界標準のインタフェースを提供し、大規模なアプリケーションの、Adaptive Server Anywhere への簡単な移行パスも提供します。

以前の Ultra Light.NET インタフェース (`iAnywhere.UltraLite` ネームスペース) に代わって ADO.NET インタフェースを使用してください。以前のインタフェースは推奨されなくなりました。

詳細については、『Ultra Light.NET ユーザーズ・ガイド』>「チュートリアル : Ultra Light.NET アプリケーションの構築」と『Ultra Light.NET ユーザーズ・ガイド』>「`iAnywhere.Data.UltraLite` ネームスペース」を参照してください。

- **Ultra Light for MobileVB の強化** Ultra Light for MobileVB は、AppForge Crossfire を使用して、Visual Basic .NET プログラミングをサポートするようになりました。

詳細については、『Ultra Light for MobileVB ユーザーズ・ガイド』> 「Ultra Light for MobileVB ユーザーズ・ガイド」を参照してください。

- **Ultra Light for M-Business Anywhere の強化** Ultra Light for M-Business Anywhere には、次のような強化が行われています。
 - Ultra Light for M-Business Anywhere は、クライアント／サーバ Ultra Light エンジンをサポートするようになりました。アプリケーションは、DatabaseManager.runtimeType プロパティを使用して、エンジンまたはランタイム・ライブラリが使用されているかどうかを調べることができます。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「Ultra Light エンジンの使用」を参照してください。

- Ultra Light for M-Business Anywhere アプリケーションは、1 回の操作でデータと Web コンテンツの両方を同期できるようになりました。

詳細については、『UltraLite for M-Business Anywhere User's Guide』> 「One-button synchronization」を参照してください。

- Mobile Link リダイレクタを使用して、1 つの M-Business Anywhere サーバを通じて、データと Web コンテンツの両方を同期できるようになりました。ファイアウォールの外部からの同期の場合、これによって、アクセス可能であることが必要なポート数が少なくて済みます。

詳細については、『UltraLite for M-Business Anywhere User's Guide』> 「Synchronizing data via M-Business Anywhere」と『Mobile Link 管理ガイド』> 「M-Business Anywhere リダイレクタ」を参照してください。

- Windows XP 上の M-Business Anywhere 5.5 が、プラットフォームとしてサポートされるようになりました。接続パラメータの `databaseOnDesktop` と `schemaOnDesktop` が、この環境をサポートします。

詳細については、『UltraLite for M-Business Anywhere User's Guide』> 「Class ConnectionParms」を参照してください。

- API に追加されたメソッドを使用すると、カラム名ではなく、カラム ID を使用して、データに関する情報を収集できます。

詳細については、『UltraLite for M-Business Anywhere User's Guide』> 「Class ResultSetSchema」と『UltraLite for M-Business Anywhere User's Guide』> 「Class TableSchema」を参照してください。

詳細については、『UltraLite for M-Business Anywhere User's Guide』> 「UltraLite for M-Business Anywhere User's Guide」を参照してください。

- **Native Ultra Light for Java の強化** Native Ultra Light for Java には、次のような強化が行われています。
 - 名前だけでなくカラム ID によってカラム・スキーマ情報にアクセス可能。
 - 新しい `SyncProgressData ErrorMessage` プロパティと、同期エラーのレポートの向上。
 - `PreparedStatement.[get]Plan` の追加。
 - `ResultSet` と `ResultSetSchema` が使用されている間、`PreparedStatement` は有効。

詳細については、『Native Ultra Light for Java ユーザーズ・ガイド』> 「Native Ultra Light for Java API リファレンス」を参照してください。

- **Ultra Light.NET コンポーネントの強化** 次の機能が Ultra Light.NET でサポートされます。これらの機能は、ADO.NET インタフェース (iAnywhere.Data.UltraLite ネームスペース) の一部として使用することをおすすめします。
 - 新しい ULCursorSchema.Name、ULResultSetSchema.Name 読み込み専用プロパティ。
 - 新しい ULSyncProgressData ErrorMessage プロパティと、同期エラーのレポートの向上。
 - ULCommand.Plan 読み込み専用プロパティ。

詳細については、『Ultra Light.NET ユーザーズ・ガイド』> 「iAnywhere.Data.UltraLite ネームスペース」を参照してください。

- **Palm 開発者はバージョンに依存しないプレフィクス・ファイルを使用できる** 以前のリリースでは、Ultra Light プレフィクス・ファイルは、開発している Palm OS のバージョンに依存していました。どのバージョンの Palm OS でも *ulpalmos.h* を使用できるようになりました。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「Ultra Light plug-in for CodeWarrior の使用」を参照してください。

- **Palm 開発者が拡張モードを使用できる** CodeWarrior は、グローバル・データのメモリの使用を向上する「**拡張モード**」と呼ばれるコード生成モードをサポートします。Ultra Light ランタイム・ライブラリの拡張モード・バージョンを使用できるようになりました。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「拡張モード・アプリケーションの構築」を参照してください。

- **信用された証明書を永続的な記憶領域から取得できる** ソフトウェアの以前のリリースでは、安全な同期の信用された証明書は、データベース・スキーマに埋め込まれていました。

Windows および Windows CE プラットフォームでは、信用された証明書を外部に保存できるようになり、`trusted_certificates` オプションを使用してアクセスできます。

詳細については、『Mobile Link クライアント』>
「`trusted_certificates`」と『Mobile Link クライアント』>
「Security Parameters 同期パラメータ」を参照してください。

SQL とランタイム の強化

- **動的 SQL の強化** Ultra Light の動的 SQL のサポートには、次のような強化が行われています。
- **クエリ最適化の向上** ソフトウェアの以前のバージョンでは、テーブルがアクセスされる順序は、クエリ内でのテーブルの出現順序でした。このバージョンでは、テーブルに効率よくアクセスできる順序になるようにクエリが最適化されます。データベース内で適切なインデックスが定義されているかぎり、オプティマイザはクエリの実行パフォーマンスの向上を支援します。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「クエリの最適化」を参照してください。
- **クエリ・プランの表示** クエリ・アクセス・プランには、わかりやすいように、インデックス番号ではなくインデックス名が含まれるようになりました。アクセス・プランは、たとえば、新しい Ultra Light Interactive SQL ユーティリティから表示できます。
- **IF 式と CASE 式** この2つの条件式が追加されて、Ultra Light がサポートする式の範囲が拡張されました。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「IF 式」と『Ultra Light データベース・ユーザーズ・ガイド』> 「CASE 式」を参照してください。
- **テーブル名に所有者名を指定できる** Ultra Light テーブルは、所有者を持ちません。既存の SQL とプログラムで生成された SQL の便宜上、`owner.table-name` がサポートされるようになりました。Ultra Light はこれを受け取りますが、`owner` は無視します。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「動的 SQL」を参照してください。

- **UNIQUEIDENTIFIER データ型の導入** UNIQUEIDENTIFIER データ型が、BINARY(16) に基づいて定義されるドメインではなく、ネイティブのデータ型になりました。この結果、Ultra Light では、必要に応じて型変換が自動的に行われるため、文字列から UUID への変換関数と UUID から文字列への変換関数を使用して UNIQUEIDENTIFIER 値を処理する必要はありません。

詳細については、『ASA SQL リファレンス・マニュアル』> 「UNIQUEIDENTIFIER データ型 [バイナリ]」を参照してください。

- **Ultra Light クエリ・プランの記述の強化** Ultra Light Interactive SQL で表示できる Ultra Light クエリ・プランの記述が強化されて理解しやすくなり、パフォーマンスの問題を適切に診断できるようになりました。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「クエリの最適化」を参照してください。

管理の強化

- **Ultra Light Interactive SQL ユーティリティ** Ultra Light データベースに対する SQL 文のテストと、Ultra Light データの変更を行う目的で、Ultra Light Interactive SQL ユーティリティが提供されています。このユーティリティはクエリ・プランも表示するので、パフォーマンスの問題を診断できます。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「Ultra Light Interactive SQL ユーティリティ」と『Ultra Light データベース・ユーザーズ・ガイド』> 「チュートリアル : Ultra Light データベースの使用」を参照してください。

- **データベース管理のコマンド・ライン・ユーティリティ** 一連のコマンド・ライン・ユーティリティによって、Windows マシン上の Ultra Light ファイルを対象とするデータベース管理タスクが簡単になりました。これらのユーティリティは、アプリケーション開発時に特に効果的です。

新しい各ユーティリティは、ulconv ユーティリティが提供するタスクのサブセットを実行します。ソフトウェアの将来のバージョンでは、ulconv ユーティリティが、これらの新しい単一タスク・ユーティリティに置き換わる予定です。

詳細については、次の項目を参照してください。

- ◆ 『Ultra Light データベース・ユーザーズ・ガイド』> 「ulcreate ユーティリティ」
- ◆ 『Ultra Light データベース・ユーザーズ・ガイド』> 「ulload ユーティリティ」
- ◆ 『Ultra Light データベース・ユーザーズ・ガイド』> 「ulsync ユーティリティ」
- ◆ 『Ultra Light データベース・ユーザーズ・ガイド』> 「ulunload ユーティリティ」

同期の強化

- **Mobile Link クライアントと HTTP インフラストラクチャの統合の向上** 2つの新しいクライアント接続パラメータを使用して、カスタム・ヘッダとカスタム cookie を指定できるようになりました。

詳細については、『Mobile Link クライアント』> 「custom_header」と『Mobile Link クライアント』> 「set_cookie」を参照してください。

- **Schema Painter からの同期スクリプトの生成** Ultra Light Schema Painter は、Adaptive Server Anywhere 統合データベースの同期スクリプトを生成できるようになりました。この機能を使用すると、Ultra Light アプリケーションを同期アーキテクチャに簡単に拡張できます。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「Ultra Light スキーマ・ペインタ」と『Ultra Light データベース・ユーザーズ・ガイド』> 「チュートリアル：Ultra Light データベースの使用」を参照してください。

- **参照整合性違反の同期通知** 参照整合性違反をレポートする同期コールバック関数がサポートされるようになりました。現在は、参照整合性に違反するローが通知されずに削除されます。

詳細については、『Mobile Link 管理ガイド』>「参照整合性と同期」を参照してください。

QAnywhere の新機能

- **フェールオーバー・サーバ** QAnywhere エージェントは、Mobile Link サーバ接続プロトコル・オプションを 1 つだけでなく、リストで持つことができるようになりました。

『QAnywhere ユーザーズ・ガイド』>「フェールオーバー・メカニズムの設定」を参照してください。
- **エミュレータのサポート** Pocket PC 2002 と Pocket PC 2003 の QAnywhere クライアント・アプリケーションは、x86 エミュレータをサポートするようになりました。これらのエミュレータでは、QAnywhere Agent の "scheduled" ポリシーのみがサポートされます。
- **サーバ・メッセージ・ストアとしてサポートされる新しい RDBMS** サポートされているすべての Mobile Link 統合データベースを、サーバ・メッセージ・ストア (Adaptive Server Anywhere、Adaptive Server Enterprise、Microsoft SQL Server、Oracle、DB2) として、QAnywhere アプリケーションの中で使用できるようになりました。
- **.NET Compact Framework の QAnywhere .NET クライアント・ライブラリによるメッセージ・リスナのサポート** .NET Compact Framework の QAnywhere .NET クライアント・ライブラリが、メッセージ・リスナをサポートするようになりました。

転送ルールの強化

- **リモート・メッセージ・ストア・プロパティの同期** リモート・メッセージ・ストア・プロパティを設定すると、そのプロパティがサーバ・メッセージ・ストアに同期され、転送ルールで使用できます。

詳細については、『QAnywhere ユーザーズ・ガイド』>「カスタムのクライアント・ストア・プロパティ」を参照してください。

- **メッセージ・ストア・プロパティの強化** `ias_Network` プロパティに、詳細なネットワーク情報にアクセスできるフィールドが設定されました。

詳細については、『QAnywhere ユーザーズ・ガイド』> 「クライアント・ストア・プロパティ」を参照してください。

さらに、カスタマイズされたメッセージ・ストア・プロパティを作成できます。

詳細については、『QAnywhere ユーザーズ・ガイド』> 「カスタムのクライアント・ストア・プロパティ」を参照してください。

- **メッセージ削除のルール** メッセージ・ストア内でのメッセージの持続性について、転送ルールを指定できるようになりました。クライアント側とサーバ側でメッセージを削除できます。

詳細については、『QAnywhere ユーザーズ・ガイド』> 「転送ルール」を参照してください。

QAnywhere Agent の強化

- **接続文字列** `qaagent -c` オプションで接続文字列を指定して、ローカル・メッセージ・ストアを起動できるようになりました。Adaptive Server Anywhere 接続文字列パラメータを使用できます。

詳細については、『QAnywhere ユーザーズ・ガイド』> 「-c オプション」を参照してください。

- **クワイエット・モード** QAnywhere Agent は 2 種類のクワイエット・モードをサポートしており、これによって、一部の Windows CE デバイスで生じる問題を回避できます。

詳細については、『QAnywhere ユーザーズ・ガイド』> 「-q オプション」および『QAnywhere ユーザーズ・ガイド』> 「-qi オプション」を参照してください。

- **QAstop ユーティリティ** `-qi` オプションを指定して QAnywhere Agent をクワイエット・モードで起動した場合は、新しい `qastop` ユーティリティを使用して QAnywhere Agent を停止する必要があります。

詳細については、『QAnywhere ユーザーズ・ガイド』> 「-qi オプション」を参照してください。

- **冗長性の強化** -o または -ot オプションで出力・ログ・ファイルの名前を指定し、-os および -ot オプションで出力ファイルのサイズを調整できるようになりました。また、以前の -verbose オプションが -v オプションに置き換えられました。-v を使用すると、ログ出力を詳細に制御できます。

次の項目を参照してください。

- ◆ 『QAnywhere ユーザーズ・ガイド』> 「-o オプション」
- ◆ 『QAnywhere ユーザーズ・ガイド』> 「-ot オプション」
- ◆ 『QAnywhere ユーザーズ・ガイド』> 「-on オプション」
- ◆ 『QAnywhere ユーザーズ・ガイド』> 「-os オプション」
- ◆ 『QAnywhere ユーザーズ・ガイド』> 「-v オプション」
- **リモート・メッセージ・ストアとして使用するデータベースの初期化** 新しい qaagent -si オプションを使用して、リモート・メッセージ・ストアを設定できるようになりました。詳細については、『QAnywhere ユーザーズ・ガイド』> 「-si オプション」を参照してください。
- **バージョン 9.0.1 からのアップグレード** QAnywhere Agent の新しい -su オプションを使用すると、バージョン 9.0.1 のリモート・メッセージ・ストアを 9.0.2 にアップグレードできます。

詳細については、『QAnywhere ユーザーズ・ガイド』> 「-su オプション」を参照してください。

QAnywhere Mobile Link システム・テーブル

すべての QAnywhere Mobile Link システム・テーブルは、ml_qa_user_group によって所有されるようになりました。以前は、DBO によって所有されていました。

2 つの新しい Mobile Link システム・テーブルが追加されました。詳細については、次の項目を参照してください。

- 『Mobile Link 管理ガイド』> 「ml_qa_delivery」
- 『Mobile Link 管理ガイド』> 「ml_qa_delivery_client」

各種の Mobile Link システム・テーブルのスキーマに変更が加えられています。詳細については、次の項目を参照してください。

- 『Mobile Link 管理ガイド』> 「ml_qa_global_props」
- 『Mobile Link 管理ガイド』> 「ml_qa_global_props_client」
- 『Mobile Link 管理ガイド』> 「ml_qa_repository」
- 『Mobile Link 管理ガイド』> 「ml_qa_repository_client」
- 『Mobile Link 管理ガイド』> 「ml_qa_repository_props」
- 『Mobile Link 管理ガイド』> 「ml_qa_repository_client」

9.0.2 クライアントの場合、次の Mobile Link システム・テーブルは生成されません。

- ml_qa_repository_staging_client
- ml_qa_status_staging_client

9.0.2 サーバの場合、次の Mobile Link システム・テーブルは生成されません。

- ml_qa_repository_content

マニュアルの強化

この項では、バージョン 9.0.2 の Adaptive Server Anywhere マニュアルの体裁、編成、またはナビゲーションの強化について説明し、主な変更についてはすべて説明します。

新しいマニュアル

既存の機能の説明が、次に示すさまざまな分野で強化されています。

- **SNMP エージェントのマニュアル** Adaptive Server Anywhere SNMP エージェントを説明する新しいマニュアルが追加されました。

詳細については、『ASA SNMP Extension Agent ユーザーズ・ガイド』> 「Adaptive Server Anywhere SNMP Extension Agent」を参照してください。

- **Windows CE の手引き** Windows CE ユーザの手引きとなる章が追加されました。

詳細については、『SQL Anywhere Studio の紹介』> 「SQL Anywhere Studio for Windows CE」を参照してください。

- **Mobile Link 同期クライアントに対する DBTools インタフェース** DBTools の dbmlsync の使用方法に関するサンプルとその他の情報が追加されました。

詳細については、『Mobile Link クライアント』> 「dbmlsync の DBTools インタフェース」を参照してください。

- **QAnywhere の強化** QAnywhere のマニュアルが拡張され、JMS メッセージング・システムと Mobile Link データ同期にメッセージングを統合する方法に関して、新しい情報が追加されました。また、QAnywhere アプリケーションの設定に関する情報が強化されました。

詳細については、『QAnywhere ユーザーズ・ガイド』> 「QAnywhere ユーザーズ・ガイド」を参照してください。

- **サーバ起動同期の SDK** SDK のマニュアルが拡張され、Palm Listener SDK の項が新しく追加されました。

『Mobile Link サーバ起動同期ユーザーズ・ガイド』> 「Mobile Link Listener SDK」を参照してください。

マニュアルの強化

- **Mobile Link の再編成** Mobile Link のマニュアルが、クライアント・ガイド、管理ガイド、チュートリアル・マニュアルに再編成されました。クライアント・ガイドには、Adaptive Server Anywhere クライアントの説明に加えて、以前は『Ultra Light データベース・ユーザーズ・ガイド』に記載されていた、Ultra Light クライアント用の同期パラメータと同期接続パラメータの説明も含まれています。
- **Ultra Light API と QAnywhere API のリファレンス** Ultra Light.NET、Ultra Light C++ API、QAnywhere .NET、QAnywhere C++ API の情報が、他のマニュアルと同じ形式で提供されるようになりました。この結果、HTML ベースのマニュアルと同様、PDF でも提供されます。

バージョン 9.0.2 での動作の変更

この項では、SQL Anywhere Studio バージョン 9.0.2 のコンポーネントに導入された動作の変更について説明します。また、現在のソフトウェアではサポートされているが、SQL Anywhere Studio の今後のメジャー・リリースからはサポートされない、廃止される機能についても説明します。

廃止される機能は変更の可能性があります

廃止される機能のリストはあくまでも予定であって完全なものとは限らず、変更の可能性があります。

Adaptive Server Anywhere の動作の変更

非推奨機能とサポートされなくなった機能

ここでは、サポートを終了した機能、今後使用することを推奨しない機能の中で、既存のアプリケーションに影響する可能性があるものがリストされています。

- **MIN_TABLE_SIZE_FOR_HISTOGRAM オプションの削除**

データベース・サーバは、MIN_TABLE_SIZE_FOR_HISTOGRAM オプションを使用しなくなりました。ソフトウェアの以前のバージョンでは、このオプションを使用して、ヒストグラムの作成対象となる最小テーブル・サイズを指定できました。現在のバージョンでは、Adaptive Server Anywhere は、5 つ以上のローから成るすべてのテーブルに対して、自動的にヒストグラムを作成します。CREATE STATISTICS 文を使用すると、サイズに関係なく、すべてのテーブルに対してヒストグラムを作成できます。

詳細については、『ASA SQL ユーザーズ・ガイド』>「カラム統計の更新」を参照してください。

- **廃止されたデータベース・オプション** 次のデータベース・オプションは、サポートされなくなりました。

- TRUNCATE_DATE_VALUES

- **ASSUME_DISTINCT_SERVERS**
- **古いデータベース・フォーマットの廃止** SQL Anywhere Studio の今後のメジャー・リリースでは、ソフトウェアの旧バージョンで作成されたデータベースはサポートされません。移行ツールは提供されます。
- **UNIX 用の非スレッド型 DBTools ライブラリの廃止** UNIX 用の非スレッド型 DBTools ライブラリ (*libdbtool9.so*) は廃止されます。現在のソフトウェアでは完全にサポートされていますが、SQL Anywhere Studio の今後のメジャー・リリースではサポートされません。
- **950TWN 照合の非サポート** 950ZHO_HK 照合と 950ZHO_TW 照合が 950TWN 照合に優先します。

詳細については、『ASA データベース管理ガイド』> 「提供されている照合と推奨する照合」を参照してください。

その他の動作の変更

ここでは、これまでのバージョンとは異なる動作をリストにして説明します。

- **トランザクション・ログを削除する場合のトランザクション・ログ・ユーティリティ (dblog) の制限** -n オプションを使用してトランザクション・ログを削除する場合は、対応するトランザクション・ログ・オフセットを無視するオプション (Log Transfer Manager の場合は -il、SQL Remote の場合は -ir、dbmlsync の場合は -is) も指定する必要があります。
- 詳細については、『ASA データベース管理ガイド』> 「トランザクション・ログ・ユーティリティのオプション」を参照してください。
- **クワイエット・モードで動作するデータベース・ユーティリティ** -q オプション (クワイエット・モード) が指定された状態で次のいずれかの操作を実行する場合は、-y オプションも指定する必要があります。
 - サービス作成 (dbsvc) ユーティリティでサービスの変更または削除を行う

- データ・ソース (dbdsn) ユーティリティでデータ・ソースの変更または削除を行う
- 消去 (dberase) ユーティリティでファイルを消去する
- ログ変換 (dbtran) ユーティリティでトランザクション・ログを変換する
- **ECC_TLS 暗号化または RSA 暗号化を使用する場合は証明書名とパスワードの指定が必要** 証明書のデフォルト値である `certificate_password` パラメータと `trusted_certificates` パラメータが削除されました。これらのデフォルト値は、SQL Anywhere Studio インストール環境の `win32` ディレクトリに提供されているサンプル証明書を利用していました。サンプル証明書は、テストと開発の目的でのみ効果的ですが、セキュリティは備えていません。

また、`-ec all` サーバ・オプションもサポートされなくなりました。

詳細については、『ASA データベース管理ガイド』> 「`-ec` サーバ・オプション」を参照してください。

- **-xs サーバ・オプションの変更** `-xs all` サーバ・オプションは、HTTP ポートと HTTPS ポートの両方で、Web 要求の受信がサポートされなくなりました。

詳細については、『ASA データベース管理ガイド』> 「`-xs` サーバ・オプション」を参照してください。

- **デフォルト・ポートを使用していない場合、Mac OS X、HP-UX、Tru64 上のネットワーク・データベース・サーバに対して TCP/IP ポート番号を指定する必要がある** Mac OS X、HP-UX、または Tru64 でデータベース・サーバを起動する場合、デフォルト・ポート (2638) が使用中またはデフォルト・ポートを使用しないのであれば、`ServerPort [PORT]` プロトコル・オプションを使用してポート番号を指定する必要があります。

詳細については、『ASA データベース管理ガイド』> 「`ServerPort` プロトコル・オプション [PORT]」を参照してください。

- **アンロード・ユーティリティ (dbunload) でアンロードおよび再ロードされるデータベースの dbspace ファイル名が変更される** アンロード・ユーティリティ (dbunload) の -an オプションでデータベースをアンロードおよび再ロードした場合、新しいデータベースの dbspace ファイル名は、末尾に **R** が追加されます。これは、新しい dbspace ファイルが、元の dbspace ファイルと同じディレクトリに配置された場合の、名前の競合を回避するために行われます。Sybase Central の [データベース・アンロード] ウィザードを使用してデータをアンロードおよび再ロードした場合も、dbspace ファイル名には R が追加されます。

詳細については、『ASA データベース管理ガイド』> 「アンロード・ユーティリティのオプション」を参照してください。

- **LONG VARCHAR 値を返すプロパティ関数** 次の関数はこれまで VARCHAR(254) 値を返していました。現在は、VARCHAR(*maxpropsize*) 値を返します。ここで、*maxpropsize* は、サーバに対して指定されている最大ページ・サイズに基づきます。
 - CONNECTION_PROPERTY
 - DB_EXTENDED_PROPERTY
 - DB_PROPERTY
 - EVENT_PARAMETER
 - PROPERTY
- **STRTOUUID 関数の変更** 以前のリリースでは、STRTOUUID は、無効な UUID 値が渡されると NULL を返していました。現在は、CONVERSION_ERROR オプションが OFF に設定されていないかぎり、変換エラーを返します。CONVERSION_ERROR オプションが OFF の場合は、NULL を返します。

Mobile Link の動作の変更

ここでは、これまでのバージョンとは異なる動作をリストにして説明します。

セキュリティ動作の変更

- **HTTPS を使用するために廃止された HTTP+TLS セキュリティ** HTTP で接続しているクライアントを対象とするトランスポート・レイヤ・セキュリティは廃止されました。HTTP でトランスポート・レイヤ・セキュリティを使用するには、HTTPS を使用する必要があります。

サーバ側のセキュリティの詳細については、『Mobile Link 管理ガイド』> 「トランスポート・レイヤ・セキュリティを使用する Mobile Link 同期サーバの起動」を参照してください。

クライアント側のセキュリティの詳細については、『Mobile Link 管理ガイド』> 「トランスポート・レイヤ・セキュリティを使用する Mobile Link クライアントの設定」を参照してください。

- **Mobile Link で ECC_TLS 暗号化または RSA 暗号化を使用する場合は証明書名とパスワードの指定が必要** 証明書のデフォルト値である `certificate_password` 同期パラメータと `trusted_certificates` 同期パラメータが削除されました。これらのデフォルト値は、SQL Anywhere Studio インストール環境の `win32` ディレクトリに提供されているサンプル証明書を利用していました。サンプル証明書は、テストと開発の目的でのみ効果的ですが、セキュリティは備えていません。

詳細については、『Mobile Link 管理ガイド』> 「-x オプション」を参照してください。

Mobile Link のその他の動作の変更

- **UDP 受信のポーリング間隔がない** Listener の UDP 接続のポーリング間隔がなくなりました。Listener は、メッセージを即時に処理します。

詳細については、『Mobile Link サーバ起動同期ユーザズ・ガイド』> 「Listener」の **-l オプション**を参照してください。

- **Treo 180 Smartphone と Kyocera 6035 Smartphone における Mobile Link Palm Listener サポートの廃止** Palm Listener のサポート・デバイスの詳細については、『Mobile Link サーバ起動同期ユーザズ・ガイド』> 「Palm デバイス用 Listener」を参照してください。

SQL Remote の動作の変更

非推奨機能とサポートされなくなった機能

ここでは、サポートを終了した機能、今後使用することを推奨しない機能の中で、既存のアプリケーションに影響する可能性があるものがリストされています。

- **SQL Remote for Adaptive Server Enterprise の廃止** SQL Anywhere Studio の今後のメジャー・リリースでは、SQL Remote for Adaptive Server Enterprise がなくなります。Mobile Link は、Adaptive Server Enterprise データベースと Adaptive Server Anywhere データベース間のデータ同期に対して、より柔軟でスケーラブルなソリューションを提供します。

その他の動作の変更

ここでは、これまでのバージョンとは異なる動作をリストにして説明します。

- **抽出 (dbxtract) ユーティリティ** dbxtract でリモート・データベースを抽出するときに -q オプション (クワイエット・モード) を指定する場合は、既存のコマンド・ファイルが確認されずに自動的に置換されるように、-y オプションも指定する必要があります。

詳細については、『SQL Remote ユーザズ・ガイド』> 「抽出ユーティリティ」を参照してください。

- **IPM_Receive メッセージ制御パラメータ** MAPI IPM_Receive メッセージ制御パラメータのデフォルト値が、YES に変更されました。この値を YES に設定すると、IPC メッセージと IPM メッセージの両方が SQL Remote に選択されます。

詳細については、『SQL Remote ユーザズ・ガイド』> 「MAPI メッセージ・システム」を参照してください。

Ultra Light の動作の変更

Ultra Light の今後のメジャー・リリースでは、業界標準 API を使用した開発が強化され、本来の静的型インタフェースに代わってコンポーネント・モデルを使用した開発が強化されます。これらの変更は、Ultra Light を使用したアプリケーション開発が簡単になるなど、ユーザにとってさまざまな利点をもたらします。

この計画の結果、このリリースでは各種の Ultra Light API が使用されなくなりました。つまり、現在のソフトウェアでは引き続き完全にサポートされていますが、今後のメジャー・リリースではサポートされません。今後のメジャー・リリースでは、旧式のインタフェースを使用するアプリケーションの移行を支援するツールが提供されます。

ここにリストされている非推奨機能とサポートされなくなった機能は、あくまでも予定であり、変更の可能性があります。

非推奨機能とサポートされなくなった機能

次の機能は、使用が推奨されないか、サポートが終了しました。

- **静的インタフェースの廃止** SQL Anywhere Studio の今後のメジャー・リリースでは、静的型 C++ API または静的型 Java API をサポートしません。Embedded SQL インタフェースは使用できますが、現在生成されているコード・メカニズムを通じては使用できません。
- **ADO.NET に置き換えられる Ultra Light.NET コンポーネント・インタフェース** このリリースでは、Ultra Light.NET は、新しい iAnywhere.Data.UltraLite ネームスペースでの ADO.NET 開発をサポートします。ADO.NET は、業界標準インタフェースの利点を備えており、大規模なアプリケーションを Adaptive Server Anywhere に簡単に移行することができます。Ultra Light.NET コンポーネント API (iAnywhere.UltraLite ネームスペース) は、このリリースでは使用が推奨されず、今後のメジャー・リリースでは提供されません。
- **JDBC に置き換えられる Native Ultra Light for Java コンポーネント・インタフェース** 現在の Native Ultra Light for Java インタフェースは、JDBC インタフェースに置き換えられることが予定されています。

その他の動作の変更

ここでは、これまでのバージョンとは異なる動作をリストにして説明します。

- **ダウンロード時の参照整合性に基づく削除に対して警告が出力される** Ultra Light は、ダウンロード時に参照整合性を維持するために、必要に応じてローを自動的に削除します。このように削除されたローごとに、警告が出力されるようになりました。

詳細については、『Mobile Link 管理ガイド』>「参照整合性と同期」を参照してください。

- **Native Ultra Light for Java の動作の変更** `Cursor.getRowCount()` メソッドが `int` を返すようになりました。ただし、アプリケーションの変更は必要ありません。
- **Ultra Light.NET コンポーネントの動作の変更**
`Cursor.getRowCount()` メソッドが `int` を返すようになりました。ただし、アプリケーションの変更は必要ありません。
- **無効な同期パラメータの処理** 以前のリリースでは、Ultra Light ランタイムは、無効な同期パラメータはすべて無視していました。このため、スペルミスのパラメータは無視され、デフォルト値が代わりに使用されていました。

このリリースでは、無効なパラメータが検出されると同期が失敗し、SQL コードの `SQLE_UNRECOGNIZED_OPTION` が設定されます。エラー・コールバックが設定されている場合は、無効なパラメータごとにそのエラー・コールバックが呼び出されます。重複は、引き続き無視されます。

- **セキュリティ機能を備えた同期のための新しいライブラリ** 同期用のセキュリティ・オプションが、別個のライブラリに移動されました。暗号化された同期で `ULSecureCerticomTLSStream` または `ULSecureRSATLSStream` セキュリティ・オプションを使用する場合は、対応する静的ライブラリに対して個別にリンクするか、個々の DLL を提供する必要があります。

詳細については、『Mobile Link クライアント』>「Security 同期パラメータ」を参照してください。

- **Ultra Light for MobileVB と Crossfire の統合** 既存のプロジェクトで、ソフトウェアの以前のバージョンの Crossfire と統合されている Ultra Light for MobileVB を使用している場合は、Interop.UltraLiteAFLib.dll への参照を、iAnywhere.UltraLiteForAppForge.dll に変更する必要があります。

詳細については、『Ultra Light for MobileVB ユーザーズ・ガイド』> 「チュートリアル：AppForge Crossfire のサンプル・アプリケーション」を参照してください。

QAnywhere の動作の変更

非推奨機能とサポートされなくなった機能

- **QAnywhere Agent オプション** 次の QAnywhere Agent (qaagent) オプションが廃止され、置換されました。

廃止された qaagent オプション	置換後の qaagent オプション
-agent_id <i>id</i>	-id <i>id</i>
-dbauser <i>user</i>	-c "UID=<i>user</i>"
-dbeng <i>name</i>	-c "ENG=<i>name</i>"
-dbfile <i>filename</i>	-c "DBF=<i>filename</i>"
-dbname <i>name</i>	-c "DBN=<i>name</i>"
-ek <i>key</i>	-c "DBKEY=<i>key</i>"
-password <i>password</i>	-c "PWD=<i>password</i>"
-sv	-c "Start={ <i>dbeng9</i> <i>dbsrv9</i> }"
-verbose	-v[<i>levels</i>]

また、次の qaagent オプションは不要となり、廃止されました。

- ◆ **-e**
- ◆ **-rb**

詳細については、『QAnywhere ユーザーズ・ガイド』>「QAnywhere Agent の構文」を参照してください。

- **QAnywhere Agent はクライアント・メッセージ・ストアを作成しない** メッセージ・ストア・データベースを各自で作成してから、qaagent を実行する必要があります。QAnywhere が必要とするシステム・オブジェクトでデータベースを初期化する、-si という新しいオプションが用意されています。

詳細については、『QAnywhere ユーザーズ・ガイド』>「クライアント・メッセージ・ストアの設定」を参照してください。

その他の動作の変更

- **QAnywhere クライアント・ライブラリ・メッセージのオーバーヘッドの縮小** 通常のメッセージに対して、ias_MessageType プロパティは設定されなくなりました。ネットワーク・ステータスと、システム・キューに送信される他のシステム・メッセージに対しては引き続き設定されます。

第3章

バージョン 9.0.1 の新機能

この章の内容

この章では、SQL Anywhere Studio バージョン 9.0.1 で導入された新機能と変更された動作の概要について説明します。

バージョン 9.0.1 の新機能

この項では、SQL Anywhere Studio バージョン 9.0.1 のコンポーネントに導入された新機能について説明します。

Adaptive Server Anywhere の新機能

この項では、Adaptive Server Anywhere バージョン 9.0.1 の新機能について説明します。新機能について主要なものからそうでないものまですべてを示し、各機能の詳細についての参照先も記述しています。

OLAP の強化

- **OLAP クエリの拡張** 一連のオンライン分析処理 (OLAP) 機能を使用して、データベース内のデータをより詳細に分析できます。強化機能には、CUBE と GROUPING SETS を使用して小計ローを結果セットに柔軟に追加する機能や、移動平均および他の高度な機能を提供するウィンドウ機能などが含まれます。

これらの OLAP 機能のサポートは、クエリ・エディタに組み込まれています。これらを使用して、ROLLUP、CUBE、GROUPING SETS 演算を使用するクエリを構築できます。

詳細については、次の項目を参照してください。

- ◆ 『ASA SQL ユーザーズ・ガイド』> 「OLAP の使用」
- ◆ 『ASA SQL ユーザーズ・ガイド』> 「ROLLUP の使用」
- ◆ 『ASA SQL ユーザーズ・ガイド』> 「CUBE の使用」
- ◆ 『ASA SQL ユーザーズ・ガイド』> 「GROUPING SETS の使用」
- ◆ 『ASA SQL ユーザーズ・ガイド』> 「OLAP 関数の活用」
- ◆ 『ASA SQL ユーザーズ・ガイド』> 「CUBE の使用」

- **新しい統計関数** 複数の統計関数が追加されました。

詳細については、次の項を参照してください。

- ◆ 『ASA SQL リファレンス・マニュアル』> 「COS 関数 [数値]」

- ◆ 『ASA SQL リファレンス・マニュアル』> 「COVAR_POP 関数 [集合]」
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「COVAR_SAMP 関数 [集合]」
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「CUME_DIST 関数 [集合]」
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「DENSE_RANK 関数 [集合]」
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「PERCENT_RANK 関数 [集合]」
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「RANK 関数 [集合]」
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「REGR_AVGX [集合]」
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「REGR_AVGY [集合]」
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「REGR_COUNT [集合]」
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「REGR_INTERCEPT [集合]」
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「REGR_R2 [集合]」
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「REGR_SLOPE [集合]」
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「REGR_SXX [集合]」
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「REGR_SXY [集合]」
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「REGR_SYY [集合]」
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「ROW_NUMBER 関数 [集合]」
- **新しい文字列関数** 次の文字列関数が追加されました。
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「BASE64_DECODE 関数 [文字列]」
 - ◆ 『ASA SQL リファレンス・マニュアル』> 「BASE64_ENCODE 関数 [文字列]」

- ◆ 『ASA SQL リファレンス・マニュアル』> 「COMPRESS 関数 [文字列]」
- ◆ 『ASA SQL リファレンス・マニュアル』> 「DECOMPRESS 関数 [文字列]」
- ◆ 『ASA SQL リファレンス・マニュアル』> 「DECRYPT 関数 [文字列]」
- ◆ 『ASA SQL リファレンス・マニュアル』> 「ENCRYPT 関数 [文字列]」
- ◆ 『ASA SQL リファレンス・マニュアル』> 「HASH 関数 [文字列]」

文の強化

- **LOAD TABLE の強化** LOAD TABLE 文には、作成される統計を制限するための句が用意され、テーブルをより速くロードできるようになりました。また、この文には、ファイルの先頭の数行を無視するための SKIP オプションもあります。

詳細については、『ASA SQL リファレンス・マニュアル』> 「LOAD TABLE 文」を参照してください。

- **SELECT ...INTO *base-table*** この新しい SELECT 構文は、ベース・テーブルを作成し、このテーブルにクエリのデータを入力します。

詳細については、『ASA SQL リファレンス・マニュアル』> 「SELECT 文」を参照してください。

- **SQL 文の変数の拡張サポート** 一部の文は、複数のロケーションで定数と同じように変数を使用可能にすることによって、より柔軟に作成されています。これは、変数を宣言して使用できるストアド・プロシージャやバッチで特に役に立ちます。その結果、以前には、より煩雑な方法でのみ使用できた機能を EXECUTE IMMEDIATE で使用できます。

次の文には、このような変数の拡張サポートがあります。

- SELECT 文の TOP 句では、定数や整数変数が参照できます。『ASA SQL リファレンス・マニュアル』> 「SELECT 文」を参照してください。

- BACKUP 文 *backup-directory* と *archive-root*。『ASA SQL リファレンス・マニュアル』> 「BACKUP 文」を参照してください。
- RESTORE 文 *filename*、*archive-root*、新しい *dbspace-name*。『ASA SQL リファレンス・マニュアル』> 「RESTORE DATABASE 文」を参照してください。
- LOAD TABLE 文 *filename*。『ASA SQL リファレンス・マニュアル』> 「LOAD TABLE 文」を参照してください。
- UNLOAD 文と UNLOAD TABLE 文 *filename*。詳細については、『ASA SQL リファレンス・マニュアル』> 「UNLOAD 文」と『ASA SQL リファレンス・マニュアル』> 「UNLOAD TABLE 文」を参照してください。
- **SET 文の強化** SET 文は、Microsoft SQL Server との互換性のためにオプション ANSI_NULLS (ANSINULLS オプションと同じ)に対応できるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「SET 文 [T-SQL]」を参照してください。

- **ALTER TABLE 文の拡張** ALTER TABLE は、デフォルト値をもつ NOT NULL カラムを空でないテーブルに追加できるようになりました。この機能によって、既存のテーブルを変更するときの柔軟性が増しました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「ALTER TABLE 文」を参照してください。

- **ALTER VIEW 文の強化** ALTER VIEW 文は、基本となるテーブルのカラムを変更するときにビュー定義を再作成するための RECOMPILE 句をサポートできるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「ALTER VIEW 文」を参照してください。

- **MESSAGE 文の強化** MESSAGE 文には、FOR CONNECTION 句が追加されました。

また、MESSAGE 文には、DEBUG ONLY 句も追加されました。DEBUG_MESSAGES オプションを ON に設定すると、DEBUG ONLY 句を含む MESSAGE 文のあるすべてのストア・プロシージャとトリガに対してデバッグ・メッセージが表示されます。

詳細については、『ASA SQL リファレンス・マニュアル』>「MESSAGE 文」と『ASA データベース管理ガイド』>「DEBUG_MESSAGES オプション [データベース]」を参照してください。

セキュリティの強化

- **データベース・ページ・チェックサム** データベース・ページ・チェックサムは、データベース・ページがディスクで修正されたかどうかを判断するのに使用します。チェックサムを有効にしてデータベースを作成した場合、チェックサムはページがディスクに書き込まれる前に計算されます。ページがディスクから読み出されるときに、チェックサムは再計算されて、保存されているチェックサムと比較されます。値が異なる場合は、ページがディスクで変更されたか、破損しています。この機能を使用するには、チェックサムを有効にして既存のデータベースをアンロードしてからデータベースに再ロードします。データベースのチェックサムが有効かどうかを確認するには、Checksum プロパティを使用します。

チェックサムを有効にしてデータベースを作成する方法については、『ASA SQL リファレンス・マニュアル』>「CREATE DATABASE 文」、『ASA データベース管理ガイド』>「初期化ユーティリティ」、および『ASA データベース管理ガイド』>「データベース・レベルのプロパティ」を参照してください。

また、チェックサムを使用してデータベースを検証することもできます。

詳細については、『ASA SQL リファレンス・マニュアル』>「VALIDATE CHECKSUM 文」、『ASA データベース管理ガイド』>「検証ユーティリティ」、および『ASA SQL リファレンス・マニュアル』>「sa_validate システム・プロシージャ」を参照してください。

パフォーマンスの強化

複雑なクエリを含む広範なタスクのパフォーマンスを向上させるために多くの強化が行われています。これらの強化の一部は、単に内部的な強化です。その他の強化は、次のとおりです。

- **並列インデックス・スキャン** ハードウェアまたはソフトウェア RAID アレイなど、複数のディスクのスピンドルがあるボリュームでは、クエリ・オプティマイザは、インデックスを並列的に使用してテーブルをスキャンできるようになりました。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「並列インデックス・スキャン」を参照してください。

- **クラスタード・ハッシュ GROUP BY アルゴリズム** Adaptive Server Anywhere クエリ・オプティマイザは、パフォーマンスを向上させるために、HAVING 句がローのごく一部を返す GROUP BY クエリの特定のクラスにとって特に役に立つ新しいアルゴリズムを使用できます。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「クラスタード・ハッシュ GROUP BY アルゴリズム」と『ASA データベース管理ガイド』> 「OPTIMIZATION_WORKLOAD オプション [データベース]」を参照してください。

- **データベース・サーバのキャッシュ・ウォーミング** キャッシュ・ウォーミングをサポートするために、3 つの新しいデータベース・サーバのコマンド・ライン・オプションが追加されました。データベースが最後に起動したときに参照したデータベース・ページとともにデータベース・サーバのキャッシュを事前にロードすることによって、データベースに対して実行される初期クエリの実行時間を短縮できるよう、キャッシュ・ウォーミングが設計されています。キャッシュ・ウォーミングを使用することによって、データベースが起動するたびにデータベースに対して同じクエリが実行される時のパフォーマンスを向上させることができます。

詳細については、『ASA データベース管理ガイド』> 「-cc サーバ・オプション」、『ASA データベース管理ガイド』> 「-cr サーバ・オプション」、『ASA データベース管理ガイド』>

「-cv サーバ・オプション」、および『ASA SQL ユーザーズ・ガイド』>「キャッシュ・ウォーミングの使用」を参照してください。

- **オブティマイザに関するヒント WITH (XLOCK)** は、FROM 句の新しいテーブル・ヒント機能です。XLOCK は、ヒントが指定されたテーブルの文によって処理されるローを排他的にロックすることを指定します。影響を受けるローは、トランザクションの終わりまでロックされたままになります。この機能はすべての独立性レベルで動作します。

WITH INDEX ヒントは、クエリの最適化中にオブティマイザに特定のインデックスを使用させます。これは高度な機能で、正しく使用しないとパフォーマンスが低下する可能性があります。このため、この機能は経験のあるユーザのみ使用してください。

詳細については、『ASA SQL リファレンス・マニュアル』>「FROM 句」と『ASA SQL ユーザーズ・ガイド』>「頻繁に検索するカラムにインデックスを使う」を参照してください。

- **NetWare の内部実行スレッドのデフォルトのスタック・サイズの増加** NetWare の内部実行スレッドのデフォルトのスタック・サイズは、128 K に増加されました。

詳細については、『ASA データベース管理ガイド』>「-gss サーバ・オプション」を参照してください。

プログラミング・インタフェースの強化

- **Perl インタフェース** Perl DBI モジュールの Perl の新しい DBD::ASAny ドライバを使用して、Perl スクリプトから Adaptive Server Anywhere データベースにアクセスして修正できます。

詳細については、『ASA プログラミング・ガイド』>「DBD::ASAny Perl インタフェース」を参照してください。

- **InstallShield プロジェクト** SQL Anywhere Studio には、InstallShield Merge Module Projects と Object Projects が組み込まれました。これらのプロジェクトを使用して Merge Modules と Objects を生成し、現在マシンにインストールされているソフトウェアを再配備できます。SQL Anywhere の前のバージョン

ンには、Merge Modules と Objects が含まれています。これらを使用して元のソフトウェアを再配備することもできますが、EBF を適用した後の再配備としては便利な手段ではありません。

詳細については、『ASA プログラミング・ガイド』> 「InstallShield を使用した配備」を参照してください。

管理の強化

- **BACKUP の強化** BACKUP 文には、イメージのバックアップ用として ON EXISTING ERROR 句が組み込まれました。この句を指定すると、バックアップがすでに存在するときにファイルを作成するとエラーが発生します。

BACKUP 文のアーカイブのバックアップが、以前はイメージのバックアップでのみ使用可能だったオプションをサポートできるよう強化されています。

詳細については、『ASA SQL リファレンス・マニュアル』> 「BACKUP 文」を参照してください。

バックアップ・ユーティリティは、サーバ・マシン上でバックアップを作成できるようになりました。以前は、バックアップはクライアント・マシン上でのみ作成できました。

詳細については、『ASA データベース管理ガイド』> 「バックアップ・ユーティリティ」を参照してください。

- **アンロード・ユーティリティ (dbunload) の強化** アンロード・ユーティリティは、データベースをアンロードするときにビューの依存性を自動的に処理できるようになりました。ソフトウェアの以前のバージョンで、ビュー定義を *reload.sql* ファイルに複数回出力するために使用されていた *-j* オプションは、使用されなくなりました。アンロード・ユーティリティは、他のビューに依存するビュー定義のアンロードを自動的に処理します。

また、アンロード・ユーティリティを使用して、新しいデータベースにアンロードするときのデータベース・ページ・サイズを変更できます。

詳細については、『ASA データベース管理ガイド』> 「dbunload コマンド・ライン・ユーティリティを使用したデータベースのアンロード」を参照してください。

- **サーバ検索 (dblocate) ユーティリティの強化** サーバ検索 [dblocate] ユーティリティでは、ホスト名または IP アドレスを指定し、データベース・サーバの検索対象を特定のマシンに制限できるようになりました。また、IP アドレスがマシン名に解析されないよう指定する -n オプションをサポートできるようになったため、パフォーマンスが向上します。

詳細については、『ASA データベース管理ガイド』> 「サーバ検索ユーティリティ」を参照してください。

- **Adaptive Server Anywhere コンソール・ユーティリティによる統合化ログインのサポート** Windows NT/2000/XP で Adaptive Server Anywhere コンソール [dbconsole] ユーティリティに接続する場合、[接続] ダイアログでは、統合化ログインを使用してデータベースに接続できます。

詳細については、『SQL Anywhere Studio ヘルプ』> 「[接続] ダイアログ : [ID] タブ」を参照してください。

- **データベース・サーバを再起動せずに要求ログ・ファイルを変更可能** データベース・サーバの起動時には、-zs サーバ・オプションを使用して、要求ログ・ファイルのサイズを指定できます。sa_server_option システム・プロシージャを使用すると、データベース・サーバを再起動せずに、要求ログ・ファイルのサイズを変更できます。

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

- **プロファイリング・システム・トリガに追加された追加情報** sa_procedure_profile システム・プロシージャと sa_procedure_profile_summary システム・プロシージャは、データベースでプロシージャ・プロファイリングがオンになっている場合、システム・トリガに関する追加情報を返すようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』>「sa_procedure_profile システム・プロシージャ」と『ASA SQL リファレンス・マニュアル』>「sa_procedure_profile_summary システム・プロシージャ」を参照してください。

- **新しいシステム・テーブル** データベースが起動されたプラットフォームとソフトウェアの異なるバージョンに関する情報を管理する新しいシステム・テーブルが追加されました。

詳細については、『ASA SQL リファレンス・マニュアル』>「SYSHISTORY システム・テーブル」を参照してください。

- **新しい照合** リトアニア語 (1257LIT、ANSI コード・ページ 1257) をサポートする照合、トルコ語 (1254TRKALT) をサポートする照合の 2 つの照合が使用可能になりました。トルコ語の照合では、I-dot と I-no-dot は区別されません。

詳細については、『ASA データベース管理ガイド』>「提供されている照合と推奨する照合」と『ASA データベース管理ガイド』>「代替トルコ語照合 1254TRKALT」を参照してください。

- **DEDICATED_TASK オプション** このオプションを指定すると、要求処理タスクは 1 つの接続からの要求の処理専用になります。この事前に確立された接続を使用すると、データベース・サーバが応答しなくなったときにこのサーバのステータスに関する情報を収集できます。

詳細については、『ASA データベース管理ガイド』>「DEDICATED_TASK オプション [データベース]」を参照してください。

Interactive SQL の強化

- **Interactive SQL を使用すると、ファイルの読み込みと書き込みに使用されるコードを指定できます。** Interactive SQL の READ、INPUT、OUTPUT 文は、ファイルの読み込みと書き込みに使用される文字コードを指定できるオプションのコード句をサポートできるようになりました。DEFAULT_ISQL_ENCODING オプションが追加されることで、後続の READ、INPUT、OUTPUT 文に使用される文字コードを指定できます。

詳細については、『ASA データベース管理ガイド』> 「DEFAULT_ISQL_ENCODING オプション [Interactive SQL]」、『ASA SQL リファレンス・マニュアル』> 「READ 文 [Interactive SQL]」、『ASA SQL リファレンス・マニュアル』> 「INPUT 文 [Interactive SQL]」、および『ASA SQL リファレンス・マニュアル』> 「OUTPUT 文 [Interactive SQL]」を参照してください。

また、Interactive SQL の [インポート] ウィザードと [エクスポート] ウィザードを使用するときにファイルの読み込みと書き込みに使用される文字コードも指定できます。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「データのインポートとエクスポート」と『SQL Anywhere Studio ヘルプ』> 「[エクスポート] ウィザード」を参照してください。

- **Interactive SQL による統合化ログインのサポート** Windows NT/2000/XP で Interactive SQL に接続する場合、[接続] ダイアログでは、統合化ログインを使用してデータベースに接続できます。

詳細については、『SQL Anywhere Studio ヘルプ』> 「[接続] ダイアログ: [ID] タブ」を参照してください。

- **InteractiveSQL では、結果セットの表示に使用されるフォントを設定できます。** Interactive SQL の [結果] ウィンドウ枠内に表示されるデータのフォント、フォント・スタイル、ポイント・サイズを選択できます。

Interactive SQL の [結果] ウィンドウ枠の設定については、『SQL Anywhere Studio ヘルプ』> 「[オプション] ダイアログ: [結果] タブ」を参照してください。

- **Interactive SQL を使用すると、ファイルのブラウズに使用される初期フォルダを指定できます。** Interactive SQL でファイルをブラウズする場合、初期ディレクトリとして現在のディレクトリ (オペレーティング・システムによって定義されているディレクトリ) と、ファイルが最後に開かれたフォルダのどちらを使用するかを指定できます。

詳細については、『SQL Anywhere Studio ヘルプ』> 「[オプション] ダイアログ : [一般] タブ」を参照してください。

Sybase Central の強化

- **Sybase Central では、結果セットの表示に使用されるフォントを設定できます。** テーブルを選択するときに、Sybase Central の [データ] タブに表示されるデータのフォント、フォント・スタイル、ポイント・サイズを選択できます。

Sybase Central の [データ] タブの設定については、『SQL Anywhere Studio ヘルプ』> 「[プラグインの環境設定] ダイアログ : [テーブル・データ] タブ」を参照してください。

- **[リモート・サーバ作成] ウィザードが現在のユーザの外部ログインの作成をサポート** [リモート・サーバ作成] ウィザードでは、現在のユーザの外部ログインを作成することによって、リモート・サーバを作成する前に外部ログインを作成する必要がなくなりました。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「Sybase Central を使用したリモート・サーバの作成」を参照してください。

- **Sybase Central による統合化ログインのサポート** Windows NT/2000/XP で Sybase Central に接続する場合、[接続] ダイアログでは、統合化ログインを使用してデータベースに接続できます。

詳細については、『SQL Anywhere Studio ヘルプ』> 「[接続] ダイアログ : [ID] タブ」を参照してください。

- **Sybase Central の [ビュー] メニューを使用してカラムをソート可能** Sybase Central の [ビュー] メニューの [ソート] 項目を使用すると、右ウィンドウ枠にあるカラム見出しをクリックする代わりに、右ウィンドウ枠でカラムをソートできます。
- **[外部キー] プロパティ・シートから外部キー設定を変更可能**

Sybase Central の外部キー設定は、[外部キー] プロパティ・シートから変更できます。

詳細については、『SQL Anywhere Studio ヘルプ』> 「[設定の変更] ダイアログ」を参照してください。

- **[プロキシ・テーブル] ウィザードでプライマリ・キー・カラム情報を表示可能** 以前は、[プロキシ・テーブル] ウィザードを使用してプロキシ・テーブルを作成する場合、リモート・テーブルのプライマリ・キーに属しているカラムを確認できませんでした。しかし、プライマリ・キーに属しているカラムをウィザードで確認できるようになりました。
- **ユーティリティ・ウィザードをキャンセル可能** [データベース・アップグレード] ウィザード、[データベース・バックアップ] ウィザード、[データベース・リストア] ウィザード、[データベース検証] ウィザード、[データベース圧縮] ウィザード、[データベース展開] ウィザード、および [バックアップ・イメージ作成] ウィザードは、キャンセルできます。これらのウィザードには、操作が成功したか失敗したかについてのステータス情報が表示されるメッセージ・ダイアログがあります。
- **Sybase Central はサービスの作成と編集時に domain\user の形式のアカウント名をサポート** [サービス作成] ウィザードと [サービス] プロパティ・シートでは、サービスの作成と編集時に **domain\user** の形式のアカウント名を入力できるようになりました。アカウント名は、[サービス] プロパティ・シートまたは [サービス作成] ウィザードの [アカウント] タブの [その他のアカウント] フィールドに入力できます。
- **データベース・サーバは Linux プラットフォーム上で非同期 I/O を使用** Linux 上でデータベース・サーバを実行するときに、データベース・サーバは可能であれば、デフォルトで非同期 I/O を使用します。-ua データベース・サーバ・オプションでは、非同期 I/O の使用をオフにできます。

詳細については、『ASA データベース管理ガイド』> 「-ua サーバ・オプション」を参照してください。

その他の機能強化

- **OPENXML による等号述部のサポート** OPENXML 関数では、XPath 式で等号述部を使用できます。この機能によって、属性値を使用して XML ドキュメント内のノードを検索できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「OPENXML 関数 [文字列]」を参照してください。

- **TransactionStartTime 接続プロパティ** このプロパティは、COMMIT または ROLLBACK の後にデータベースが最初に修正された時間を返します。

詳細については、『ASA データベース管理ガイド』>「接続レベルのプロパティ」を参照してください。

- **UserAppInfo プロパティ** このプロパティは、AppInfo 接続パラメータで指定された接続文字列の一部を返します。

詳細については、『ASA データベース管理ガイド』>「接続レベルのプロパティ」を参照してください。

- **ConsoleLogFile サーバ・プロパティ -o オプションを指定すると、このプロパティは、データベース・サーバ・ウィンドウからのメッセージがログに記録されるファイル名を返します。**

詳細については、『ASA データベース管理ガイド』>「サーバ・レベルのプロパティ」を参照してください。

- **UNIX プラットフォームの DriveType データベース・プロパティ** DriveType データベース・プロパティは、UNIX プラットフォームに対応できるよう拡張されています。

詳細については、『ASA データベース管理ガイド』>「データベース・レベルのプロパティ」を参照してください。

- **接続 ID は 1 から始まり、データベース・サーバへの新しい接続ごとに 1 ずつ増加** データベース・サーバが起動すると、サーバへの接続ごとに 1 から始まる接続 ID が割り当てられます。この接続番号は、サーバへの新しい接続ごとに 1 ずつ増加します。接続 ID は、-z サーバ出力と LOGFILE 接続パラメータ出力のログに記録されます。また、接続 ID は、

CONNECTION_PROPERTY、NEXT_CONNECTION、NEXT_DATABASE、および DROP CONNECTION 関数と要求のロギングによっても使用されます。

詳細については、『ASA SQL リファレンス・マニュアル』>「CONNECTION_PROPERTY 関数 [システム]」、『ASA SQL リファレンス・マニュアル』>「NEXT_CONNECTION 関数 [システム]」、『ASA SQL リファレンス・マニュアル』>「NEXT_DATABASE 関数 [システム]」、および『ASA SQL ユーザーズ・ガイド』>「要求ロギング」を参照してください。

- **NetWare と UNIX 上でのキャッシュ管理の向上** -c を使用して指定されたキャッシュ・サイズが UNIX または NetWare 上で使用可能なメモリ量よりも大きい場合、データベース・サーバは、使用可能なメモリ量に基づいて最大キャッシュ・サイズを計算できるようになりました。

3 つの環境でのデータベース・サーバによる最大キャッシュ・サイズの計算方法については、『ASA データベース管理ガイド』>「-c サーバ・オプション」を参照してください。

- **ODBC_DISTINGUISH_CHAR_AND_VARCHAR オプション**

ODBC_DISTINGUISH_CHAR_AND_VARCHAR オプションは、Adaptive Server Anywhere ODBC ドライバによる CHAR カラムの記述方法を制御します。

詳細については、『ASA データベース管理ガイド』>「ODBC_DISTINGUISH_CHAR_AND_VARCHAR オプション [データベース]」を参照してください。

Mobile Link の新機能

次に、バージョン 9.0.1 で導入したソフトウェアに加えられた変更と追加を示します。

- **QAnywhere メッセージ** Mobile Link QAnywhere には、アプリケーション対アプリケーションのメッセージ機能が用意されています。この機能によって、Windows または Windows CE

オペレーティング・システムで動作しているさまざまなデバイス上のリモート・アプリケーションとメッセージを交換するアプリケーションを作成できます。

詳細については、『QAnywhere ユーザーズ・ガイド』>「QAnywhere ユーザーズ・ガイド」を参照してください。

- **外部認証** Mobile Link のユーザ認証が強化され、LDAP サーバと POP3 電子メール・サービスなどの他の外部ソースを使用してユーザを簡単に認証できるようになりました。

詳細については、『Mobile Link クライアント』>「外部サーバに対する認証」を参照してください。

- **新しい Mobile Link システム・テーブル** 新しい Mobile Link システム・テーブルは複数あります。新しい Mobile Link システム・テーブルが存在するということは、Adaptive Server Anywhere データベースをアップグレードし、その他の統合データベースのアップグレード・スクリプトを実行する必要があります。

詳細については、『Mobile Link 管理ガイド』>「Mobile Link システム・テーブル」を参照してください。

- **設定可能なスクリプト・バージョン** 新しい ml_property Mobile Link システム・テーブルを使用すると、スクリプト・バージョンのプロパティを格納できます。

詳細については、『Mobile Link 管理ガイド』>「ml_property」と『Mobile Link 管理ガイド』>「ml_add_property」を参照してください。

- **iAnywhere ODBC ドライバ** Windows で DB2 対応の iAnywhere ODBC ドライバを使用できるようになりました。これは Wire Protocol ドライバであるため、DB2 クライアント・ソフトウェアは必要ありません。

詳細については、『Mobile Link およびリモート・データ・アクセスの ODBC ドライバ』>「iAnywhere Solutions ODBC ドライバの概要」を参照してください。

- **IBM DB2 設定スクリプトのバージョン番号** Mobile Link 同期サーバの複数のバージョンが同じ DB2 サーバ・インスタンスを使用できるようになりました。これは、Mobile Link が DB2 ストアド・プロシージャに対して使用する 2 つの Java クラスに SQL Anywhere Studio バージョン番号が使用されるようになったためです。9.0.1 リリースでは、これらは `SyncDB2_901.class` と `SyncDB2Long_901.class` と呼ばれます。

詳細については、『Mobile Link 管理ガイド』> 「IBM DB2 統合データベース」を参照してください。

- **新しい -us サーバ・オプション** 新しいサーバ・オプションを使用すると、Mobile Link が不要なテーブル・スクリプトを呼び出せなくなるため、パフォーマンスが向上します。

詳細については、『Mobile Link 管理ガイド』> 「-us オプション」を参照してください。

- **ActiveSync プロバイダによるアクティビティ・ログ・ファイルの生成** ActiveSync プロバイダは、アクティビティのログを生成できるようになりました。

詳細については、『Mobile Link クライアント』> 「ActiveSync プロバイダ・インストール・ユーティリティ」を参照してください。

Adaptive Server Anywhere クライアントの強化

- **Adaptive Server Anywhere クライアントのアプリケーション統合の向上** dbmlsync の新しい統合コンポーネントを使用すると、Windows プラットフォーム上で Adaptive Server Anywhere リモート・データベースと統合するアプリケーションをより簡単にカスタマイズしやすい方法で作成できます。

詳細については、『Mobile Link クライアント』> 「dbmlsync 統合コンポーネント」を参照してください。

- **失敗したダウンロードの再開** Adaptive Server Anywhere と Ultra Light の両方のリモートデータベースでダウンロードが失敗したときに、長時間のデータ再送を防ぐことができるようになりました。ダウンロードの部分的再送の後、失敗したダウンロードを再開できる場合があります。

詳細については、『Mobile Link 管理ガイド』> 「失敗したダウンロードの再開」を参照してください。

- **Adaptive Server Anywhere クライアントのトランザクション・レベルのアップロード** リモート・データベース上のトランザクションをアップロードに保持できるようになりました。また、この処理は同期単位で行うことができます。

詳細については、『Mobile Link クライアント』> 「-tu オプション」を参照してください。

- **拡張オプションの新しい指定方法**
sp_hook_dbmlsync_set_extended_options という新しいフックを使用すると、次に行われる同期の動作をプログラムでカスタマイズできます。

詳細については、『Mobile Link クライアント』> 「sp_hook_dbmlsync_set_extended_options」を参照してください。

サーバ起動同期

- **自動デバイス追跡** 自動デバイス追跡によって、通知プロセスが簡単になります。

詳細については、『Mobile Link サーバ起動同期ユーザーズ・ガイド』> 「デバイス・トラッキング」を参照してください。

- **Sybase Central の設定** Sybase Central Mobile Link プラグインを使用して、通知を設定できるようになりました。

詳細については、『Mobile Link サーバ起動同期ユーザーズ・ガイド』> 「プロパティの設定」を参照してください。

- **オプションの配信確認** メッセージを受信したときに確認が自動的に統合データベースに送信されるよう通知を設定できるようになりました。

詳細については、『Mobile Link サーバ起動同期ユーザーズ・ガイド』> 「Listener ユーティリティ」を参照してください。

- **新しい Listener オプション** Listener には、複数のチャネルを一度に受信する機能など、複数の新しいコマンド・ライン・オプションが用意されています。

詳細については、『Mobile Link サーバ起動同期ユーザーズ・ガイド』> 「Listener」を参照してください。

- **Palm の設定** Palm デバイスを簡単に設定するための Palm Listener 設定ユーティリティ (dblsncfg) が用意されました。

詳細については、『Mobile Link サーバ起動同期ユーザーズ・ガイド』> 「Palm デバイス用 Listener」を参照してください。

Mobile Link モニタ

- **モニタ・データのエクスポート** モニタ・データをリレーショナル・データベースまたは Excel ファイルにエクスポートできるようになりました。
- **モニタの強化情報** テーブルにどのカラムを表示するかをカスタマイズできるようになりました。また、以前は[同期]プロパティ・シートからのみ使用可能だった情報が含まれる新しいカラムや、モニタ・セッションの同期をユニークに識別する新しいカラムが用意されました。
- **ソート機能の向上** モニタ・テーブルのソート機能が向上しました。データが追加または更新されときのソート順を管理できるようになりました。
- **ユーザ・インタフェースの強化** チャート内の選択項目をズーム・イン、ズーム・アウト、またはズームするための新しいメニューとツール・バー・ボタンが用意されました。また、一時停止によってチャートを自動的にスクロールするかどうかを制御できるようになりました。

詳細については、『Mobile Link 管理ガイド』> 「Mobile Link モニタ」を参照してください。

新しい Web サーバのサポート

- **リダイレクタによる Apache Web サーバのサポート** Apache Web サーバ用の新しいネイティブなリダイレクタが用意されています。Apache Web サーバを使用する場合、Tomcat は必要ありません。

詳細については、『Mobile Link 管理ガイド』> 「リダイレクタによる Web サーバを経由した同期」を参照してください。

Ultra Light の新機能

Ultra Light 9.0.1 には、次の新機能が導入されています。

- **Ultra Light.NET コントロール** コントロールのセットが Visual Studio.NET 2003 ツールボックスに追加され、接続パラメータを指定しやすくなったり、Ultra Light.NET アプリケーションから同期をモニタしやすくなります。

詳細については、『Ultra Light.NET ユーザーズ・ガイド』>「チュートリアル：Ultra Light.NET アプリケーションの構築」を参照してください。

- **Ultra Light for M-Business Anywhere** 以前は AvantGo M-Business Server と呼ばれていた新しいコンポーネントが iAnywhere M-Business Anywhere で使用できます。

詳細については、『Ultra Lite for M-Business Anywhere User's Guide』>「Ultra Lite for M-Business Anywhere User's Guide」を参照してください。

- **動的 SQL の CREATE と DROP 文** CREATE/DROP TABLE と CREATE/DROP INDEX 文は、動的 SQL で使用できるようになりました。Ultra Light コンポーネントのユーザは、これらの文を使用して、Ultra Light データベースのスキーマを変更できます。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』>「CREATE INDEX 文」と『Ultra Light データベース・ユーザーズ・ガイド』>「CREATE TABLE 文」を参照してください。

- **動的 SQL からのトランザクションの制御** COMMIT と ROLLBACK 文は、動的 SQL で使用できるようになりました。Ultra Light コンポーネントのユーザは、これらの文によって、SQL 文を使用してトランザクションを制御できます。これらの文は、接続オブジェクトに対してコミットとロールバック・メソッドの代わりとして使用できます。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』>「COMMIT 文」と『Ultra Light データベース・ユーザーズ・ガイド』>「ROLLBACK 文」を参照してください。

- **動的 SQL SELECT の強化** WHERE 句または HAVING 句の探索条件でサブクエリを使用できます。これらは、FROM 句の抽出テーブルとしても使用できます。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』>「SELECT 文」と『Ultra Light データベース・ユーザーズ・ガイド』>「動的 SQL の探索条件」を参照してください。

HAVING 句がサポートされるようになりました。『Ultra Light データベース・ユーザーズ・ガイド』>「SELECT 文」を参照してください。

- **Ultra Light に対する ODBC インタフェース** Ultra Light は、ODBC プログラミング・インタフェースのサブセットをサポートできるようになりました。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』>「チュートリアル：ODBC を使用したアプリケーションの構築」と『Ultra Light C/C++ ユーザーズ・ガイド』>「Ultra Light ODBC API リファレンス」を参照してください。

- **C++ インタフェースの混在** Ultra Light C/C++ ベースのインタフェース (Embedded SQL、静的型 C++ API、C++ コンポーネント) は、同じアプリケーションで使用できます。

中でも注目すべきは、C++ コンポーネントの動的 SQL が既存の Embedded SQL または静的型 C++ API アプリケーションに追加されたことや、Embedded SQL を使用して SQL を主に C++ コンポーネント・ベースのアプリケーションで実行できることです。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』>「Ultra Light C/C++ インタフェースの結合」を参照してください。新機能には、『Ultra Light C/C++ ユーザーズ・ガイド』>

「db_start_database 関数」(Embedded SQL) と『Ultra Light C/C++ ユーザーズ・ガイド』> 「StartDatabase メソッド」が含まれています。

- **Ultra Light C++ コンポーネントの CodeWarrior ステーションナリ** ステーションナリ **Palm OS Ultra Light C++ コンポーネント・アプリケーション**は、Ultra Light plug-in for CodeWarrior の一部として提供されています。これは、Palm OS 用の CodeWarrior を使用した C++ コンポーネント・アプリケーションの構築を支援します。

Ultra Light plug-in for CodeWarrior のファイルは、Ultra Light のインストール中にディスクにコピーされますが、プラグインは、追加のインストール手順を実行しなければ使用できません。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「Metrowerks CodeWarrior で Ultra Light アプリケーションを開発する」を参照してください。

- **Ultra Light C/C++ のエラー処理の向上** エラー・コールバックは、すべての Ultra Light C/C++ インタフェースでサポートされるようになりました。コールバックを使用すると、アプリケーションにすべてのエラーを通知できるため、開発者は開発時に貴重な情報を得ることができます。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「ULRegisterErrorCallback 関数」と『Ultra Light C/C++ ユーザーズ・ガイド』> 「ULRegisterErrorCallback のコールバック関数」を参照してください。

- **Ultra Light コンポーネントはエンジンを使用可能** 複数のアプリケーションからの接続を受け入れられる Ultra Light データベース・エンジンは、Ultra Light コンポーネントの代替導入オプションとして使用できるようになりました。

このオプションは、Ultra Light for MobileVB または Ultra Light ActiveX には使用できません。

Ultra Light エンジンの詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「Ultra Light エンジンの使用」を参照してください。

- **データベース変換ツール** `ulconv` ユーティリティは、Ultra Light データベース上で多くの処理を行うためのコマンド・ライン・ツールです。これには、XML ファイルへのデータベースのアンロード、XML ファイルからの新しいデータベースのロード、データベース・フォーマットの変換などの処理があります。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「`ulconv` ユーティリティ」を参照してください。

- **同期プロGRESS・イベントの追加** エラーが発生し、ダウンロードされた変更内容がロールバックされる場合、同期オブザーバは追加イベントを使用できます。

詳細については、使用している API の `ULSyncState` 構造体またはオブジェクトを参照してください。

- **スキーマのアップグレードの監視** スキーマのアップグレードには時間がかかることがあります。新しいスキーマのアップグレード・イベントには、アプリケーションがスキーマのアップグレードの進行状況を監視するためのメカニズムが採用されています。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「スキーマのアップグレードの監視」を参照してください。

- **再起動可能なダウンロード** Ultra Light は、同期 observer を介して、通信エラーやユーザによるキャンセルが原因で失敗したダウンロードを再起動できるようになりました。

詳細については、『Mobile Link 管理ガイド』> 「失敗したダウンロードの再開」を参照してください。

- **新しい Windows CE プラットフォームのサポート** Ultra Light は、Smartphone 2002 プラットフォームをサポートできるようになりました。このプラットフォームでは、ActiveSync 同期はサポートされません。

Ultra Light は、V4T ("thumb") モードの ARM チップ上の Windows CE 4.1 もサポートしています。

詳細については、『SQL Anywhere Studio の紹介』> 「Ultra Light ターゲット・プラットフォーム」を参照してください。

- **マルチデータベースのサポート** Ultra Light コンポーネントは、異なるデータベース・ファイル名または作成者 ID を指定して複数の接続要求を発行することによって、1つのアプリケーションから複数のデータベースに対応できます。

この機能のため、接続パラメータにはいくつかの拡張機能があります。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「DatabaseName 接続パラメータ」を参照してください。

- **ULPalmLaunch と ULPalmExit は不要** Ultra Light は、アプリケーションを閉じるときのステータス管理をサポートしやすくする追加の接続関連プリミティブをサポートできるようになりました。これらの新しい機能のため、Palm OS アプリケーションには、ULPalmLaunch や ULPalmExit などの Palm 固有の特別なプリミティブがなくなりました。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「Ultra Light Palm アプリケーションの状態の保存」、『Ultra Light for MobileVB ユーザーズ・ガイド』> 「Ultra Light Palm アプリケーションのステータスの管理」、および『Ultra Light C/C++ ユーザーズ・ガイド』> 「データベースへの接続」を参照してください。

- **Ultra Light データベース・プロパティ** Ultra Light データベースのプロパティは、Ultra Light コンポーネント・アプリケーションで使用できるようになりました。大文字と小文字の区別、照合、およびグローバル・オートインクリメントの値に使用されるデータベース ID はすべて、API に応じて Connection オブジェクトのプロパティまたはメソッドとして使用できます。

マニュアルの強化

この項では、バージョン 9.0.1 の Adaptive Server Anywhere マニュアルの体裁、編成、またはナビゲーションの強化について説明し、主な変更についてはすべて説明します。

新しいマニュアル

- **Mobile Link サーバ起動同期** Mobile Link サーバ起動同期機能は、適切なマニュアルに移動されました。

詳細については、『Mobile Link サーバ起動同期ユーザーズ・ガイド』>「Mobile Link サーバ起動同期ユーザーズ・ガイド」を参照してください。

- **QAnywhere メッセージ** Mobile Link の新しいメッセージ・アプリケーションである QAnywhere について説明する新しいマニュアルが作成されました。

詳細については、『QAnywhere ユーザーズ・ガイド』>「QAnywhere ユーザーズ・ガイド」を参照してください。

- **新しい Mobile Link チュートリアル** Java および .NET スクリプト論理を Adaptive Server Anywhere リモートデータベースで使用方法について説明する新しいチュートリアルが作成されました。

詳細については、『Mobile Link チュートリアル』>「チュートリアル：Adaptive Server Anywhere での Java 同期論理」と『Mobile Link チュートリアル』>「チュートリアル：Adaptive Server Anywhere での .NET 同期論理」を参照してください。

- **dbmlsync 統合コンポーネントについて説明する新しい章** 詳細については、『Mobile Link クライアント』>「dbmlsync 統合コンポーネント」を参照してください。
- **Ultra Light C/C++ マニュアルの統合** Ultra Light C/C++ インタフェース (Embedded SQL、静的型 C++ API、および C++ コンポーネント) のマニュアルは、ODBC インタフェースとともに 1 冊のマニュアルに統合されました。

『Ultra Light C/C++ ユーザーズ・ガイド』>「Ultra Light C/C++ ユーザーズ・ガイド」を参照してください。

- **Ultra Light for M-Business Anywhere** 新しい Ultra Light for M-Business Anywhere コンポーネントには独自のマニュアルが用意されました。

『UltraLite for M-Business Anywhere User's Guide』> 「UltraLite for M-Business Anywhere User's Guide」を参照してください。

マニュアルの強化

- **『Adaptive Server Anywhere 入門』と『SQL Anywhere Studio の紹介』の統合** 『Adaptive Server Anywhere 入門』と『SQL Anywhere Studio の紹介』は 1 冊のマニュアルに統合されました。新しいマニュアルは、『SQL Anywhere Studio の紹介』と呼ばれます。
- **PDF マニュアルの強化** 一部のユーザにとって、PDF は、特に概念的なマニュアルの場合、HTML ベースのオンライン・マニュアルの代わりとして役に立ちます。オンライン・マニュアルの PDF バージョンは、現在、デフォルトでインストールされており、Windows の [スタート] メニューまたはオンライン・マニュアルから各トピックの上部の PDF 項目をクリックすることによってアクセスできます。

PDF ファイルのクリッカブル機能は、マニュアル内だけでなく、他のマニュアルや Web サイト上の情報にもリンクしています。これらの機能の動作は、ファイルをブラウザのプラグインから読み込むか、または Acrobat Reader から直接読み込むかによって異なります。最善の方法は、Acrobat Reader を直接使用する方法です。

- **ヘルプに関するヘルプ** 新しい項では、Windows の HTML ヘルプ・フォーマットのオンライン・マニュアルと PDF フォーマットのオンライン・マニュアルの違いについて説明します。また、この項では、さまざまなヘルプ機能を使用してマニュアル内をナビゲーションし、目的の情報にアクセスする方法についても説明します。

詳細については、『SQL Anywhere Studio の紹介』> 「オンラインでの情報のアクセス」を参照してください。

バージョン 9.0.1 での動作の変更

この項では、SQL Anywhere Studio バージョン 9.0.1 のコンポーネントに導入された動作の変更について説明します。

Adaptive Server Anywhere の動作の変更

非推奨機能とサポートされなくなった機能

ここでは、サポートを終了した機能、今後使用することを推奨しない機能の中で、既存のアプリケーションに影響する可能性があるものがリストされています。

- **MDSR 暗号化の廃止** 以前は、Adaptive Server Anywhere では、強力な暗号化として MDSR と AES の両方がサポートされていました。現在、強力な暗号化としては唯一 AES 暗号化がサポートされています。この変更は、初期化ユーティリティ、抽出ユーティリティ、またはアンロード・ユーティリティで暗号化を使用する場合、**-ea** オプションが必要ないことを意味します。また、Create Database 文と Create Encrypted File 文の両方から *algorithm* パラメータが削除されます。
- **ライト・ファイルの廃止** このリリースでは、ライト・ファイルの使用は推奨しません。

詳細については、『ASA データベース管理ガイド』> 「ライト・ファイル・ユーティリティ (旧式)」と 『ASA SQL リファレンス・マニュアル』> 「CREATE WRITEFILE 文 (旧式)」を参照してください。
- **圧縮データベース・ファイルの廃止** このリリースでは、圧縮データベース・ファイルの使用は推奨しません。

詳細については、『ASA データベース管理ガイド』> 「圧縮ユーティリティ (旧式)」と 『ASA SQL リファレンス・マニュアル』> 「CREATE COMPRESSED DATABASE 文 (旧式)」を参照してください。

- **アンロード・ユーティリティ (dbunload) の -j オプションの廃止** アンロード・ユーティリティが強化された結果、アンロード・ユーティリティの -j オプションはサポートされなくなりました。
- **言語ユーティリティ (dblang) の -d オプションの廃止** Adaptive Server Anywhere のレジストリ設定を変更するために使用されていた言語ユーティリティの -d オプションは、サポートされなくなりました。

その他の動作の変更

ここでは、これまでのバージョンとは異なる動作をリストにして説明します。

- **CURRENT_TIMESTAMP と CURRENT_USER 特別値の追加** Microsoft SQL Server との互換性を保つために、CURRENT_TIMESTAMP (CURRENT_TIMESTAMP と同じ) と CURRENT_USER (CURRENT_USER と同じ) 特別値が追加されました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「CURRENT_TIMESTAMP 特別値」と 『ASA SQL リファレンス・マニュアル』> 「CURRENT_USER 特別値」を参照してください。

- **データベース・サーバがすでに実行している場合、db_start_engine は SQLCODE 0 を返す** db_start_engine は、データベース・サーバがすでに実行している場合、0 以外の値を返し、SQLCODE を 0 に設定するようになりました。以前は、db_start_engine は 0 以外の値を返しましたが、SQLCODE は SQLE_ENGINE_ALREADY_RUNNING に設定されていました。

詳細については、『ASA プログラミング・ガイド』> 「db_start_engine 関数」を参照してください。

- **データベースがすでに実行している場合、db_start_database は 0 以外の値と SQLCODE 0 を返す** db_start_database は、データベースがすでに実行している場合、0 以外の値を返し、SQLCODE を 0 に設定するようになりました。以前は、db_start_database は 0 (失敗を示す) を返し、SQLCODE を SQLE_ALIAS_CLASH に設定していました。

詳細については、『ASA プログラミング・ガイド』> 「db_start_database 関数」と『Ultra Light C/C++ ユーザーズ・ガイド』> 「db_start_database 関数」を参照してください。

- **検証ユーティリティのデフォルト・アルゴリズムの変更** 検証ユーティリティ (dbvalid) は、エクスプレス・チェック (-fx オプション) アルゴリズムをデフォルトで使えるようになりました。このエクスプレス・チェック・アルゴリズムによって、テーブルがサーバのキャッシュに完全には収まらない、複数のインデックスがある大規模なテーブルに対する処理速度が大幅に早くなりました。Adaptive Server Anywhere の以前のバージョンで使用されていた検証アルゴリズムを使用する場合、-fn オプションを指定できます。

詳細については、『ASA データベース管理ガイド』> 「dbvalid コマンド・ライン・ユーティリティを使用したデータベースの検証」を参照してください。

- **[接続] ダイアログでのマルチバイト文字の入力** Windows のセキュリティ上の慣習に準拠するには、Input Method Editor (IME) を使用して、Sybase Central、Interactive SQL、および Adaptive Server Anywhere コンソール・ユーティリティで使われる [接続] ダイアログの [パスワード] フィールドに日本語や他のアジア諸国のマルチバイト文字を入力することはできません。

既存のデータベースのパスワードにこれらの文字が含まれる場合、パスワードは、[接続] ダイアログと [ODBC の設定] の両ダイアログの [詳細] タブの [追加接続パラメータ] フィールドに入力できます。ただし、[詳細] タブにパスワードを入力する場合、パスワードは隠されず、プレーン・テキストで表示されるので注意してください。データベースをアップグレードする場合は、マルチバイト文字が含まれないようにパスワードを変更することをおすすめします。

- **DECLARE LOCAL TEMPORARY TABLE 文には所有者名を指定できない** ソフトウェアの以前のバージョンでは、DECLARE LOCAL TEMPORARY TABLE に所有者名を指定するときに所有者が現在のユーザと異なっていた場合、同じ名前をもつテンポラリ・テーブルを複数作成できました。現在では、所有者名を指定すると、構文エラーが発生します。

- **MIN_TABLE_SIZE_FOR_HISTOGRAM オプションのデフォルト設定の変更** MIN_TABLE_SIZE_FOR_HISTOGRAM オプションを使用して、ヒストグラムが作成されるテーブルの最小サイズを指定します。このデフォルト値は、100 ローに変更されました。ソフトウェアの以前のバージョンでは、デフォルト値は 1000 ローでした。ソフトウェアの以前のバージョンで作成したデータベースでは、SET OPTION 文を使用してこの設定を変更できます。
- **NULL 定数データ型変換の変更** 以前のバージョンでは、NULL 定数を CHAR、VARCHAR、LONG VARCHAR、BINARY、VARBINARY、または LONG BINARY データ型に変換するときに長さが指定されていない場合、カラムのサイズは 32767 に初期化されていました。現在、このサイズは 0 に初期化されます。

たとえば、以前、次のクエリは、長さが 32767 として記述されたカラムを返していました。

```
SELECT CAST( NULL AS CHAR )
-- This now returns a CHAR(0) column

SELECT 'abc'
UNION ALL
SELECT NULL
-- This now returns a CHAR(3) column

SELECT ''
UNION ALL
SELECT NULL
-- This now returns a CHAR(0) column

SELECT IF 1=1 THEN 'abc' ELSE NULL ENDIF
-- This now returns a CHAR(3) column
```

- **ORDER BY 句に序数を使用するときの UPDATE 文とエラー**
序数を使用する ORDER BY 句が含まれる UPDATE 文は、構文エラーを返すようになりました。
- **識別子に関する制限** 識別子には二重円記号または二重引用符を使用できなくなりました。識別子に円記号を使用できるのは、エスケープ文字として使用する場合のみです。

詳細については、『ASA SQL リファレンス・マニュアル』> 「識別子」を参照してください。

- **EXECUTE IMMEDIATE 文のデフォルト設定の WITH RESULT SET 句** WITH RESULT SET ON 句を指定すると、EXECUTE IMMEDIATE 文が結果セットを返すことができます。デフォルト設定は WITH RESULT SET OFF です。

詳細については、『ASA SQL リファレンス・マニュアル』> 「EXECUTE IMMEDIATE 文 [SP]」を参照してください。

Mobile Link の動作の変更

ここでは、これまでのバージョンとは異なる動作をリストにして説明します。

- **フックが呼び出される順序の変更** イベント・フックが呼び出される順序が変更されました。つまり、アップロード・イベントはインクリメンタル更新ごとに繰り返される同期シーケンスの一部にすぎないため、インクリメンタル・アップロードがより効率的になりました。

詳細については、『Mobile Link クライアント』> 「同期イベント・フックの順序」を参照してください。

- **Adaptive Server Anywhere クライアントのパブリケーションのローワイズ分割の修正** WHERE 句が含まれるパブリケーションは、WHERE 条件を満たすローのみをレプリケートするようになりました。バージョン 8.0.0 ～ 9.0.0 には、WHERE 句が不定の値と評価されたときにローをレプリケートするバグがありました。たとえば、パブリケーション WHERE 句に "WHERE val = 1" とあった場合、val が NULL のローもレプリケートされました。このバグは、SQL Remote クライアントと ASA Mobile Link クライアントの両方に影響しました。

Ultra Light の動作の変更

ここでは、これまでのバージョンとは異なる動作をリストにして説明します。

- **Palm OS のステータス管理** Ebedded SQL または静的型 C++ API が使用された Palm OS アプリケーションの場合、ULPalmExit (ULData::PalmExit) と ULPalmLaunch (ULData::PalmLaunch) 関数は、ステータスや同期情報を管理する必要がなくなり、使用されなくなりました。また、ULData と ULConnection Reopen メソッドも使用されなくなりました。

Palm OS 上のアプリケーションの初期化、接続、終了関数の順序は他のアプリケーションと同じようになりました。ULSetSynchInfo メソッドは、HotSync 同期を制御します。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』>「Palm アプリケーションに HotSync 同期を追加する」を参照してください。

- **Palm OS 3.0 のサポート終了** このリリースでサポートされる最も古いバージョンは、Palm OS 3.5 です。
- **ULEnableGenericSchema 関数 (旧式)** スキーマのアップグレードが必要な Ultra Light C/C++ アプリケーションは、ULEnableGenericSchema を呼び出す必要がなくなりました。代わりに、ULRegisterSchemaUpgradeObserver 関数を使用します。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』>「スキーマのアップグレードの監視」を参照してください。

- **Ultra Light コンポーネントのテーブル API** Table オブジェクトの Delete メソッドは、削除後にローを自動的に再表示しなくなりました。以前の動作を維持するには、削除オペレーション後に Relative(0) を使用してローを再フェッチします。
- **Native Ultra Light for Java によるカラム ID とパラメータ ID のキャストは不要** カラム ID とパラメータ ID を受け入れていたすべてのメソッド、short 型のパラメータを受け入れていた一部のメソッドは、整数を受け入れるよう変更されました。これによって、コードに数値定数をキャストする必要がなくな

なりました。たとえば、`table.getString( (short)1 );` の代わりに、`table.getString( 1 );` を使用できるようになりました。

この変更の結果、9.0.1 ソフトウェアに対応させるために Native Ultra Light for Java アプリケーションは再コンパイルします。ただし、コードの変更は必要ありません。

第4章

バージョン 9.0.0 の新機能

この章の内容

この章では、SQL Anywhere Studio バージョン 9 で導入された新機能と変更された動作の概要について説明します。

バージョン 9.0.0 の新機能

この項では、SQL Anywhere Studio バージョン 9.0 に導入された新機能について説明します。

Adaptive Server Anywhere の新機能

この項では、Adaptive Server Anywhere バージョン 9.0 の新機能について説明します。新機能について主要なものからそうでないものまですべてを示し、各機能の詳細について記述されている参照先を合わせて示します。

新機能のハイライト

- **XML のサポート** Adaptive Server Anywhere 9.0 では、XML ドキュメントの保存、リレーショナル・データを XML としてエクスポート、XML のインポート、リレーショナル・データのクエリから XML を返すなど、XML を広範囲にわたってサポートします。
- **FOR XML 句 RAW、AUTO、EXPLICIT の 3 つのモード**で SELECT 文に FOR XML 句を指定できるようになり、XML ドキュメントのクエリ結果を得ることができます。各モードでは、生成される XML の形式をそれぞれ異なるレベルで制御できます。

詳細については、『ASA SQL ユーザーズ・ガイド』>「クエリ結果を XML として取得する」と『ASA SQL リファレンス・マニュアル』>「SELECT 文」を参照してください。

- **FOR_XML_NULL_TREATMENT オプション**

FOR_XML_NULL_TREATMENT オプションを使用すると、FOR XML 句を含むクエリからの NULL 値の返し方を制御できます。

詳細については、『ASA データベース管理ガイド』>「FOR_XML_NULL_TREATMENT オプション [データベース]」を参照してください。

- **OPENXML プロシージャ** 詳細については、『ASA SQL リファレンス・マニュアル』>「OPENXML 関数 [文字列]」を参照してください。
- **SQL/XML のサポート** SQL/XML は、SQL と XML を組み合わせて使用する場合の仕様を記述した規格草案です。SQL/XML サポートの一部として、Adaptive Server Anywhere は、XML ドキュメントをデータベースに保存できる XML データ型を備えています。

詳細については、『ASA SQL リファレンス・マニュアル』>「XML データ型 [文字]」を参照してください。

Adaptive Server Anywhere では、リレーショナル・データから XML ドキュメントを生成する手段として、FOR XML 句のほかに次の SQL/XML 関数もサポートしています。

- **XMLAGG 関数** この集合関数は、XML 要素のコレクションから XML 要素のフォレストを生成します。

詳細については、『ASA SQL リファレンス・マニュアル』>「XMLAGG 関数 [文字列]」を参照してください。

- **XMLCONCAT 関数** この関数は、渡された XML 値を連結して XML 要素のフォレストを生成します。

詳細については、『ASA SQL リファレンス・マニュアル』>「XMLCONCAT 関数 [文字列]」を参照してください。

- **XMLELEMENT 関数** この関数は XML 要素を生成します。オプションで、要素の内容、属性、属性の内容を指定できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「XMLELEMENT 関数 [文字列]」を参照してください。

- **XMLFOREST 関数** この関数は、XML 要素のフォレストを生成します。

詳細については、『ASA SQL リファレンス・マニュアル』> 「XMLFOREST 関数 [文字列]」を参照してください。

- **XMLGEN 関数** この関数は、XQuery コンストラクタに基づいて XML 値を生成します。

詳細については、『ASA SQL リファレンス・マニュアル』> 「XMLGEN 関数 [文字列]」を参照してください。

- **データベース内の HTTP サーバ** Adaptive Server Anywhere データベース・サーバが Web サーバとして動作できるようになりました。このため、Adaptive Server Anywhere データベースと任意の Web ブラウザだけを使用して、Web ベースのアプリケーションを作成し、実行できます。

この機能によって、データベース・サーバは、標準の SOAP 要求に加えて、標準の HTTP 要求と HTTPS 要求も処理します。使用できるサービス・タイプは、HTTP、HTTPS、XML、RAW、SOAP、DISH です。DISH は SOAP サービスのハンドラです。

このリリースより前に作成されたデータベースでこの拡張機能を利用するには、アップグレード・ユーティリティを使用してデータベースをアップグレードする必要があります。

詳細については、『ASA データベース管理ガイド』> 「Web サービスの使用」を参照してください。

- **インデックス・コンサルタント** インデックス・コンサルタントは、適切なインデックス選択を支援するためのツールです。単一のクエリまたは操作のセットを分析して、データベースに追加するインデックスや削除するインデックスを推奨します。

このリリースより前に作成されたデータベースでこの拡張機能を利用するには、アップグレード・ユーティリティを使用してデータベースをアップグレードする必要があります。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「インデックス・コンサルタントの概要」を参照してください。

- **64 ビット・バージョンが使用できる** Itanium II チップ搭載の Windows Server 2003 では、ソフトウェアの 64 ビット通常版を使用できます。64 ビットの Linux と HP-UX オペレーティング・システムでは、クライアント/サーバ限定版を使用できます。

サポートされるプラットフォームの詳細については、『SQL Anywhere Studio の紹介』> 「SQL Anywhere Studio がサポートしているプラットフォーム」を参照してください。

SQL の強化

- **選択の前に WITH 句で共通テーブル式を指定できる** 共通テーブル式は、SELECT 文のスコープにのみ存在するテンポラリなビュー定義です。再帰的または非再帰的です。共通テーブル式を使用すると、より簡潔にクエリを記述できる場合があります。また、1つのクエリの中で複数レベルの集合を実行できます。共通テーブル式を使用できるのは、トップレベルの SELECT 文、ビュー定義内のトップレベルの SELECT 文、または INSERT 文内のトップレベルの文の中だけです。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「共通テーブル式」を参照してください。

- **特殊な形式の共通テーブル式を使用して再帰ユニオンを実行できる** 再帰的な共通テーブル式によって、再帰クエリを記述できます。階層的なデータ構造や有向グラフを表すテーブルに対してクエリを実行する場合に、この機能は特に効果的です。どの再帰共通テーブル式も、最初に実行される初期サブクエリと、再帰サブクエリを含みます。再帰サブクエリの FROM 句に指定する必要があるビューへの参照は、前の反復処理でビューに追加されたローを参照します。クエリを実行するデータ構造が何層にもわたる場合は特に、再帰処理を停止する条件の指定に注意してください。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「再帰共通テーブル式」を参照してください。

- **INTERSECT 演算と EXCEPT 演算のサポート** これらの演算では、複数の結果セットの共通部分と差異が計算されます。これは、UNION 演算を補うものです。

詳細については、次の項目を参照してください。

- 『ASA SQL ユーザーズ・ガイド』> 「EXCEPT と INTERSECT の使用」
- 『ASA SQL ユーザーズ・ガイド』> 「集合演算子と NULL」
- 『ASA SQL リファレンス・マニュアル』> 「EXCEPT 演算」
- 『ASA SQL リファレンス・マニュアル』> 「INTERSECT 演算」
- **SELECT 文にストアド・プロシージャの結果セットを指定できる** SELECT 文の中で、ベース・テーブルやビューを指定できる位置に、ストアド・プロシージャ・コールを指定できるようになりました。

ストアド・プロシージャ・コールの統計情報を保存する場合は、アップグレード・ユーティリティを使用してデータベースをアップグレードする必要があります。統計情報がない場合、ストアド・プロシージャ・コールの結果をジョインしようとしても、適切なプランが得られない可能性があります。

詳細については、『ASA SQL リファレンス・マニュアル』> 「FROM 句」を参照してください。

- **オンライン分析処理機能の追加** 使用可能な SQL 言語に、さまざまな OLAP 機能が追加されました。
- **ROLLUP 演算** クエリに GROUP BY 句が指定されている場合、ROLLUP 演算によって結果セットに小計のローが追加されます。各小計ローには、GROUP BY の結果セットに含まれるロー・セットの合計が示されます。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「ROLLUP の使用」を参照してください。

- **LIST 関数で順序付きのリストを使用できる** LIST 関数の機能が拡張されて、ソートされた項目のリストを出力できるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』
> 「LIST 関数 [集合]」を参照してください。

- **その他の集合関数** 標本ベースと母集団ベースの標準偏差と分散を計算する関数が追加されました。

詳細については、『ASA SQL リファレンス・マニュアル』
> 「集合関数」を参照してください。

- **CREATE INDEX 文で組み込み関数のインデックスを作成できる** この機能は、テーブルに新しい計算カラムを追加し、そのカラムのインデックスを作成できる便利な方法です。

詳細については、『ASA SQL リファレンス・マニュアル』>
「CREATE INDEX 文」と『ASA SQL ユーザーズ・ガイド』>
「インデックスの作成」を参照してください。

- **すべてのコンテキストで ORDER BY 句を使用できる** 以前のリリースでは、ビュー定義、サブクエリ、または UNION 演算内の多くの SELECT 文では ORDER BY 句を使用できませんでしたが、その制限が取り除かれました。

SELECT 文に ORDER BY 句を指定すると、FIRST 句や TOP 句と一緒に使用する場合などは特に、ビュー定義や集合演算の結果が影響を受けることがあります。それ以外のコンテキストでは、ORDER BY 句を指定しても、オペレーションに違いは生じません。

- **SELECT 文の TOP 句に START AT を指定できる** START AT を指定すると、結果セットを明示的に制限するクエリの中で、さらに柔軟性を高めることができます。

詳細については、『ASA SQL リファレンス・マニュアル』>
「SELECT 文」を参照してください。

- **制約に名前を指定できる** 検査制約、一意性制約、参照整合性制約に、名前を指定できるようになりました。このため、テーブル全体の制約を変更するのではなく、個々の制約を変更することで、テーブルやカラムの制約を変更できます。

このリリースより前に作成されたデータベースでこの拡張機能を利用するには、データベースをアンロードして再ロードすることによってそのデータベースのファイル・フォーマットをアップグレードする必要があります。

詳細については、『ASA SQL リファレンス・マニュアル』> 「ALTER TABLE 文」、『ASA SQL リファレンス・マニュアル』> 「CREATE TABLE 文」、および『ASA SQL ユーザーズ・ガイド』> 「テーブル制約とカラム制約の使い方」を参照してください。

- **FROM 句の中でラテラル抽出テーブルによる外部参照が可能** FROM 句の中で、抽出テーブルとストアド・プロシージャからの外部参照を指定できるようになりました。外部参照の実行を指定するには、LATERAL キーワードを使用します。

詳細については、『ASA SQL リファレンス・マニュアル』> 「FROM 句」を参照してください。

- **EXECUTE IMMEDIATE 文でエスケープ文字を柔軟に処理できる** 新しい WITH ESCAPES OFF オプションを使用すると、エスケープ文字の処理を抑制できます。この機能によって、ファイル・パスを含む動的な文を簡単に作成できるようになります。

詳細については、『ASA SQL リファレンス・マニュアル』> 「EXECUTE IMMEDIATE 文 [SP]」を参照してください。

- **EXECUTE IMMEDIATE 文が結果セットを返すクエリをサポートする** この機能によって、ストアド・プロシージャの中で、より動的に文を作成できるようになりました。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「プロシージャでの EXECUTE IMMEDIATE 文の使用」と『ASA SQL リファレンス・マニュアル』> 「EXECUTE IMMEDIATE 文 [SP]」を参照してください。

- **CREATE FUNCTION 文と ALTER FUNCTION 文で Transact-SQL 構文を使用できる** Transact-SQL 構文を使用して、呼び出し元の環境へスカラ値を返すユーザ定義関数を定義できるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「CREATE FUNCTION 文」を参照してください。

- **複数のローの挿入時にオートインクリメント・カラムの値を使用できる** value-sensitive (キーセット駆動型) カーソルを使用してローを挿入する場合、新しく挿入されたローはカーソル結果セットの最後に表示されます。

このため、直前に挿入されたローのオートインクリメント・カラムの値は、カーソル内の最後の行を選択することによって確認できます。たとえば、Embedded SQL の場合、FETCH ABSOLUTE -1 *cursor-name* を使用することで値を取得できます。

詳細については、『ASA プログラミング・ガイド』> 「カーソルによるローの変更」を参照してください。

- **リモート・データ・アクセスで UUID/GUID カラムを処理できる** リモート・データ・アクセスで、SQL Server のユニーク識別子カラムを管理できるようになりました。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「データ型変換：Microsoft SQL Server」と『ASA SQL リファレンス・マニュアル』> 「UNIQUEIDENTIFIERSTR データ型 [文字]」を参照してください。

- **リモート・データ・アクセスでリモート接続に名前を指定できる** ODBC を介してのリモート・データ・アクセス接続に名前を指定できるようになりました。このため、特定の接続を切断することができます。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「リモート・データ・アクセスの接続の管理」を参照してください。

- **式のデータ型を返す新しい関数の追加** EXPRTYPE 関数は、式のデータ型を返します。

詳細については、『ASA SQL リファレンス・マニュアル』> 「EXPRTYPE 関数 [その他]」を参照してください。

- **EXIT 文の強化** Interactive SQL の EXIT 文は、Interactive SQL に終了コードを設定できるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』>
「EXIT 文 [Interactive SQL]」を参照してください。

- **OUTPUT 文に ASIS キーワードを指定できる** ASIS を指定すると、エスケープ処理をせずに値がそのままファイルに書き込まれます。

詳細については、『ASA SQL リファレンス・マニュアル』>
「OUTPUT 文 [Interactive SQL]」を参照してください。

- **インデックスと外部キーを変更できる** ALTER INDEX 文を使用すると、インデックスと外部キーの名前を変更できます。また、プライマリまたは外部キー・インデックスに加えて、ユーザ作成インデックスについても、インデックスのタイプをクラスタードまたはノンクラスタードに変更できます。

このリリースより前に作成されたデータベースでクラスタード・インデックスを利用するには、データベースをアンロードして再ロードすることによってそのデータベースのファイル・フォーマットをアップグレードする必要があります。

詳細については、『ASA SQL リファレンス・マニュアル』>
「ALTER INDEX 文」を参照してください。

- **クエリの中で異なる集合を複数使用できる** 集合関数の引数として DISTINCT *column-name* を指定できます。ソフトウェアの以前のバージョンでは、DISTINCT 引数を伴う集合関数は、クエリに 1 つだけ指定できました。現在のバージョンでは、複数の集合関数を使用できます。以前のバージョンでは次のクエリを使用できませんでしたが、バージョン 9 では有効です。

```
SELECT count( DISTINCT first_name ),  
       count( DISTINCT last_name )  
FROM contact
```

- **サポートされるすべての言語のイベント・スケジュールで、英語のフルスペルの曜日名と省略形の曜日名が認識される** イベントを作成する場合、データベース・サーバでは、Adaptive Server Anywhere がサポートしているすべての言語において、英語のフルスペルの曜日名と省略形の曜日名の両方

が認識されます。以前のバージョンでは、英語以外の言語のスケジュールでは、フルスペルの英語の曜日名を指定する必要がありますがありました。

詳細については、『ASA SQL リファレンス・マニュアル』>「CREATE EVENT 文」を参照してください。

- **プロシージャ・テキストを隠してロジックの機密を保持** SET HIDDEN オプションを使用して、ストアド・プロシージャ、関数、トリガ、ビューに含まれるロジックを隠すことができます。これによって、ストアド・プロシージャ、関数、トリガ、ビュー内のロジックを明らかにせずに、アプリケーションやデータベースを配布できます。

このリリースより前に作成されたデータベースでこの拡張機能を利用するには、アップグレード・ユーティリティを使用してデータベースをアップグレードする必要があります。

詳細については、『ASA SQL ユーザーズ・ガイド』>「プロシージャ、関数、トリガ、ビューの内容を隠す」を参照してください。

管理とスケーラビリティの強化

- **より詳細なリターン・コードを返す検証ユーティリティ** 検証ユーティリティ (dbvalid) は、異常の発生原因を示す、より詳細なリターン・コードを返します。

詳細については、『ASA データベース管理ガイド』>「dbvalid コマンド・ライン・ユーティリティを使用したデータベースの検証」を参照してください。

- **2つの新しいサーバ・プロパティ** 2つの新しいサーバ・プロパティが追加されました。CommandLine は、サーバを起動するときに使用したコマンド・ラインを表示し、CompactPlatformVer は、PlatformVer サーバ・プロパティの縮小バージョンを表示します。

詳細については、『ASA データベース管理ガイド』>「サーバ・レベルのプロパティ」を参照してください。

- **新しい `sp_remote_primary_keys` ストアド・プロシージャ** リモート・データ・アクセスを使用するリモート・テーブルに関するプライマリ・キー情報を取得するために、`sp_remote_primary_keys` というストアド・プロシージャが追加されました。

このリリースより前に作成されたデータベースでこの拡張機能を利用するには、アップグレード・ユーティリティを使用してデータベースをアップグレードする必要があります。

詳細については、『ASA SQL リファレンス・マニュアル』>「`sp_remote_primary_keys` システム・プロシージャ」を参照してください。

- **接続の通信リンク名を返す新しい `connection_property`** `CommNetworkLink` という新しい接続プロパティが、接続の通信リンクの名前を返します。

詳細については、『ASA データベース管理ガイド』>「接続レベルのプロパティ」を参照してください。

- **NetWare が完全な文字セット変換をサポート** バージョン 8.x の NetWare では、シングルバイト対シングルバイトの変換がサポートされていました。バージョン 9.0 では、ほかのプラットフォームでサポートされるすべての文字セットが、NetWare でもサポートされます。

詳細については、『ASA データベース管理ガイド』>「-ct サーバ・オプション」を参照してください。

- **アンロード・ユーティリティでカラム・リストをアンロードできる** アンロード・ユーティリティ (`dbunload`) で、`reload.sql` ファイルに生成される、`LOAD TABLE` 文のカラム・リストをアンロードできるようになりました。これによって、テーブル内のカラムの順序を簡単に変更できます。

詳細については、『ASA データベース管理ガイド』>「アンロード・ユーティリティ」を参照してください。

- **LDAP によるデータベース・サーバの登録** データベース・サーバを LDAP サーバに登録できるようになったので、クライアントと検索ユーティリティ (`dblocate`) は、LDAP サーバに

対してデータベース・サーバの検索を実行できます。このため、WAN やファイアウォールを経由して動作しているクライアントは、IP アドレスを指定することなくデータベース・サーバを検索できます。LDAP は TCP/IP でネットワーク・サーバ上でのみ使用されます。

詳細については、『ASA データベース管理ガイド』> 「LDAP サーバを使用した接続」または『ASA データベース管理ガイド』> 「LDAP プロトコル・オプション [LDAP]」を参照してください。

- **接続数が多い場合の処理の向上** 200 を超える接続がある場合、大量の接続を適切に処理するために、活性タイムアウト値が自動的に増加します。

詳細については、『ASA データベース管理ガイド』> 「-tl サーバ・オプション」と『ASA データベース管理ガイド』> 「LivenessTimeout 接続パラメータ [LTO]」を参照してください。

- **要求ログのフィルタリングとホスト変数のサポート** 要求ログへの出力をフィルタリングできるようになり、特定の接続や特定のデータベースの要求だけを入れることができます。また、ホスト変数値を要求ログに出力できるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「sa_server_option システム・プロシージャ」、『ASA SQL ユーザーズ・ガイド』> 「パフォーマンスのモニタリングと改善」、『ASA SQL リファレンス・マニュアル』> 「sa_get_request_times システム・プロシージャ」、および『ASA データベース管理ガイド』> 「-zr サーバ・オプション」を参照してください。

- **BACKUP 文と dbbackup でログ・コピーの名前を変更できる** BACKUP 文とバックアップ・ユーティリティ [dbbackup] を使用して、ログ・コピーの名前を変更できます。

詳細については、『ASA データベース管理ガイド』> 「バックアップ・ユーティリティ」と『ASA SQL リファレンス・マニュアル』> 「BACKUP 文」を参照してください。

- **START DATABASE 文でチェックポイントでのログ・トランケーションと読み込み専用モードを指定できる START DATABASE** 文を使用すると、有効なチェックポイントでのログ・トランケーション、または読み込み専用モードを指定して、データベースを起動できます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「START DATABASE 文 [Interactive SQL]」を参照してください。

- **Adaptive Server Anywhere における異なる監査オプションのサポート** Adaptive Server Anywhere の以前のバージョンでは、監査のオンとオフを選択できました。現在のバージョンでは、監査するオプションを指定できるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「sa_disable_auditing_type」または『ASA SQL リファレンス・マニュアル』> 「sa_enable_auditing_type」を参照してください。

- **event_parameter 関数に渡せる 3 つの新しい値** 3 つの新しい値を event_parameter 関数に渡すことができます。ScheduleName は、イベントを発生したスケジュールの名前を返します。AppInfo は、イベントを引き起こした接続の connection_property ('AppInfo') の値を返します。DisconnectReason は、接続が終了した原因を示す文字列を返します。

詳細については、『ASA SQL リファレンス・マニュアル』> 「EVENT_PARAMETER 関数 [システム]」を参照してください。

- **ネットワーク・サーバに同時に接続しているユーザ数を示す新しいサーバ・プロパティ** 新しい LicensesInUse プロパティによって、ネットワーク・サーバに現在同時に接続しているユーザ数が判断されます。各同時接続ユーザは、接続数ではなく、サーバに接続しているユニークなクライアント・ネットワーク・アドレスの数で判断されます。たとえば、3 台のクライアント・マシンがサーバに接続していて、各クライアント・マシンが 2 つの接続を持つ場合、プロパティ ('LicensesInUse') を選択すると "3" が返されます。

詳細については、『ASA データベース管理ガイド』> 「サーバ・レベルのプロパティ」を参照してください。

- **サービス作成 [dbsvc] ユーティリティでサービスの起動と停止が可能** サービス作成 [dbsvc] ユーティリティに、2つの新しいオプションが追加されました。1つは **Dbsvc -u** `<service_name>` で、`service_name` で指定されるサービスを起動します。もう1つは **dbsvc -x** `<service_name>` で、`service_name` で指定されるサービスを停止します。

詳細については、『ASA データベース管理ガイド』> 「dbsvc コマンド・ライン・ユーティリティを使用したサービスの管理」を参照してください。

- **LocalOnly プロトコル・オプション [LOCAL] をサポートするネットワーク・サーバ** サーバで、LocalOnly プロトコル・オプション [LOCAL] を使用できます。LocalOnly プロトコル・オプションを YES に設定してサーバを実行すると、ネットワーク・サーバは、接続や CPU の制限を受けないパーソナル・サーバとして動作します。

詳細については、『ASA データベース管理ガイド』> 「LocalOnly プロトコル・オプション [LOCAL]」を参照してください。

- **Address Windowing Extensions 使用時の最小データベース・サーバ・キャッシュ・サイズの変更** Windows 2000、Windows XP、Windows Server 2003 で Address Windowing Extensions (AWE) を使用する場合は、データベース・サーバ・キャッシュの最小サイズが 2 MB になりました。以前のリリースにおける AWE 使用時の最小キャッシュ・サイズは 3 GB ～ 256 MB でした。

詳細については、『ASA データベース管理ガイド』> 「-cw サーバ・オプション」を参照してください。

- **ドライブ・タイプを指定する新しいデータベース・プロパティ** データベース・ファイルがあるドライブについての情報を提供する、新しい DriveType データベース・プロパティが追加されました。

詳細については、『ASA データベース管理ガイド』> 「データベース・プロパティ」を参照してください。

- **Adaptive Server Anywhere NetWare の高速化** NetWare 版の Adaptive Server Anywhere サーバは、CLIB ではなく LibC を使用するようになりました。LibC は C ランタイム・ライブラリで、これまでの CLIB ライブラリよりも、NetWare オペレーティング・システムの新しいカーネルとの相互関係に優れています。NetWare 版のすべてのクライアント側ソフトウェア (dblib、dbisql、dbconsole、dbremote など) は、引き続き CLIB を使用します。これには NetWare の最大ファイル・サイズを NTFS と同じサイズに増加できるという利点があります。複数の CPU を使用し (ある場合)、TCP と SPX が Winsock を使用することで、以前のバージョンよりも高速になります。

詳細については、『ASA データベース管理ガイド』> 「Adaptive Server Anywhere の制限」と [「バージョン 9.0 での動作の変更」125 ページ](#)を参照してください。

- **NetWare における外部関数の強化** NetWare の外部関数または外部ストアド・プロシージャは、名前が競合することなく複数の NLM を使用できるようになりました。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「外部関数のプロトタイプ」を参照してください。

- **エラー・メッセージの言語を接続ごとに指定可能** データベース・サーバへの各接続において、データベース・サーバから出力されるエラー・メッセージやさまざまな文字列の言語を要求できるようになりました。接続が使用する言語は、サーバが使用する言語とは別です。データベース・サーバでは、接続が要求する言語を使用して日付文字列を解釈することもできます。
- **プロセッサ・タイプを識別する 2 つの新しいサーバ・プロパティの追加** 2 つの新しいサーバレベル・プロパティが追加されました。1 つは ProcessorArchitecture であり、プロセッサ・タイプを識別します。もう 1 つは NativeProcessorArchitecture であり、プロセッサがエミュレートされるプラットフォーム上で、ネイティブのプロセッサ・タイプを識別します。

詳細については、『ASA データベース管理ガイド』> 「サーバ・レベルのプロパティ」を参照してください。

- データベースのパスワードの大文字と小文字の区別はデータベースの大文字と小文字の区別とは無関係 CREATE DATABASE 文、初期化 [dbinit] ユーティリティ、および [データベース作成] ウィザードを使用して、パスワードの大文字と小文字を区別するかどうかを指定します。パスワードの大文字と小文字を区別する設定は、文字列の比較に使用されるデータベースの大文字と小文字を区別する設定とは関係ありません。新しい CaseSensitivePasswords データベース・プロパティを使用して、データベースのパスワードの大文字と小文字を区別する設定を確認できます。**

詳細については、『ASA SQL リファレンス・マニュアル』> 「CREATE DATABASE 文」と『ASA データベース管理ガイド』> 「初期化ユーティリティ」を参照してください。

パフォーマンスの強化 (クエリの最適化)

次に示す新機能はクエリの最適化を強化する機能ですが、これらの機能を使用するのにユーザの操作は必要ありません。これらの機能は、ユーザが介入しなくても実行されます。クエリ実行プランを調べると、このような最適化の効果を確認できます。

最適化の強化機能を使用するためにデータベースをアップグレードする必要はありませんが、バージョン 9 ソフトウェアで作成されたデータベースが最も効率的に処理されます。

- コストベースのサブクエリの最適化** オプティマイザは、サブクエリが使用できる最適化の範囲を大幅に拡張しました。以前のリリースでは、サブクエリは、セマンティックなクエリの最適化の中でジョインとして記述し直すか、クエリの残りの部分とは別に最適化されていました。現在のバージョンでは、ジョインとして記述し直すには複雑すぎるサブクエリも、クエリの一部としてそのまま最適化できるようになりました。
- バッファされたローをフェッチすることで逐次スキャンのパフォーマンスを向上** 逐次テーブル・スキャンでデータベース・ページからローを読み込む場合、ローをバッファにコピーしてから、ユーザにローを返します。クエリの複雑さによりですが、これによって時間が大幅に短縮されます。

- **効率的に実行される TOP N クエリ** TOP N 句を使用するクエリの実行アルゴリズムが新しくなり、実行速度が速くなりました。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「上位 N のソート」を参照してください。

- **ヒストグラムに保持する頻度を決定する新しいアルゴリズム**
以前のバージョンでは、カラム・ヒストグラムは、選択性が 1% を超える値に対して単一バケットを作成していました。現在のバージョンでは単一バケットの条件が緩められ、代わって、ヒストグラムは単一バケットの最小数を保持しようとしています。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「オプティマイザの推定とヒストグラム」を参照してください。

- **現在キャッシュされているクエリ・プラン数を表示する QueryCachedPlans プロパティ** 新しい QueryCachedPlans プロパティを使用すると、特定の接続やすべての接続において、現在キャッシュされているクエリ実行プランの数を表示できます。QueryCachePages、QueryOptimized、QueryBypassed、QueryReused と組み合わせて使用することで、MAX_PLANS_CACHED オプションの最適な設定を判断できます。

詳細については、『ASA データベース管理ガイド』> 「接続レベルのプロパティ」と『ASA データベース管理ガイド』> 「データベース・プロパティ」を参照してください。

- **プロシージャの文でプランが高速にキャッシュされる** アクセス・プランがキャッシュされる文のスコープが拡張されて、プロシージャによって結果セットが呼び出し元の環境に返される、ストアド・プロシージャ内のクエリも含まれるようになりました。この強化機能によって、いくつかの文は再び最適化しなくても済みます。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「アクセス・プランのキャッシュ」を参照してください。

パフォーマンスの強化 (サーバ・オペレーション)

- **各インデックスが更新されるたびにインデックスの統計情報が維持される** 各インデックスが更新されるたびに、カタログ・テーブルのインデックスを含めてすべてのインデックスの統計情報が維持され、実質的にパフォーマンスを損なうことなくオブティマイザに対して正確な統計情報を提供します。統計は、インデックスの各統計に対して1つのローという形式で SYSATTRIBUTE で維持されます。

このリリースより前に作成されたデータベースでこの拡張機能を利用するには、データベースをアンロードして再ロードすることによってそのデータベースのファイル・フォーマットをアップグレードする必要があります。

詳細については、『ASA SQL リファレンス・マニュアル』>「SYSATTRIBUTE システム・テーブル」を参照してください。

- **パフォーマンス・モニタ統計値の追加** [通信 : 使用中のライセンシス]、[接続数] という2つの新しいパフォーマンス・モニタ統計値が追加され、使用中の接続数を追跡できるようになりました。

詳細については、『ASA データベース管理ガイド』>「通信の統計値」と『ASA データベース管理ガイド』>「その他の統計値」を参照してください。

- **UNLOAD 文と UNLOAD TABLE 文への APPEND { ON | OFF } オプションの追加** 新しい APPEND オプションを使用すると、アンロードしたデータを指定のファイルの最後に追加できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「UNLOAD 文」または『ASA SQL リファレンス・マニュアル』>「UNLOAD TABLE 文」を参照してください。

- **テンポラリー・テーブルを NOT TRANSACTIONAL として宣言できる** NOT TRANSACTIONAL を使用すると、テーブルは COMMIT や ROLLBACK の影響を受けません。テーブルにアクセスするプロシージャが COMMIT なしで繰り返し呼び出される場合に、この拡張機能は便利です。

詳細については、『ASA SQL リファレンス・マニュアル』>「CREATE TABLE 文」と『ASA SQL リファレンス・マニュアル』>「DECLARE LOCAL TEMPORARY TABLE 文」を参照してください。

- **継続的なインデックス統計情報** 候補となるインデックスの物理的プロパティに関して正確な統計情報を管理することにより、オプティマイザはコストに基づいて、使用するインデックスを容易に選択できます。統計情報は **SYSATTRIBUTE** にあり、インデックスが更新されるたびに保持されます。また、**VALIDATE** 文は、指定のインデックスの統計情報が正確であるかどうかを検証し、正確でない場合はエラーを生成します。これによって、実質上パフォーマンス・コストをかけることなく、正確な統計がオプティマイザに提供されます。

このリリースより前に作成されたデータベースでこの拡張機能を利用するには、データベースをアンロードして再ロードすることによってそのデータベースのファイル・フォーマットをアップグレードする必要があります。

詳細については、『ASA SQL リファレンス・マニュアル』>「**SYSATTRIBUTE** システム・テーブル」と『ASA SQL リファレンス・マニュアル』>「**VALIDATE INDEX** 文」を参照してください。

- **OPTIMISTIC_WAIT_FOR_COMMIT オプションの追加** このオプションは、トランザクションがプライマリ・ローの前に外部ローを追加した場合に、バージョン 5.x のロック動作を模倣するためのものです。汎用的なオプションではありませんが、バージョン 5.x のアプリケーションをバージョン 8.x 以降に移行する場合に役立ちます。

詳細については、『ASA データベース管理ガイド』>「**WAIT_FOR_COMMIT** オプション [データベース]」を参照してください。

- **拡張プロパティ関数の追加** 新しい **db_extended_property()** 関数は、プロパティ固有の文字列パラメータをオプションで指定できる点を除いて、**db_property()** 関数に似ています。

詳細については、『ASA SQL リファレンス・マニュアル』>「DB_EXTENDED_PROPERTY 関数 [システム]」を参照してください。

- **2つの新しいプロパティの追加** **FileSize** と **FreePages** という2つの新しいプロパティが追加されました。各プロパティとも、オプションの引数で、プロパティが要求される DB 領域を指定できます。

詳細については、『ASA データベース管理ガイド』>「データベース・プロパティ」を参照してください。

- **サーバのクワイエット・モードの強化** サーバのクワイエット・モードとエラー・ログのスイッチが強化され、さまざまなメッセージの出力を抑制できるようになりました。また、-q オプションは -qw オプションに変更され、-Q オプションは -qi オプションに変更されました。

開発ツールと管理ツール

- **Adaptive Server Anywhere プラグインの変更** Sybase Central の Adaptive Server Anywhere プラグインの構成が変更されました。これまでプロパティ・シート、ダイアログ・ボックス、左ウィンドウ枠内のフォルダに表示されていた情報の多くが、右ウィンドウ枠内のタブに表示されるようになりました。たとえば、外部キーの情報を表示する場合、現在のバージョンでは、左ウィンドウ枠で外部キーを持つテーブルを選択し、右ウィンドウ枠で [外部キー] タブを選択します。以前のバージョンでは、左ウィンドウ枠に [外部キー] フォルダがありました。

この他にも、プラグインには次に示すようないくつかの変更が加えられています。

- テーブル・エディタは別のウィンドウではありません。テーブルは、Sybase Central の右ウィンドウ枠で直接編集します。
- 一度に複数のウィンドウを開きたい場合、Sybase Central の右ウィンドウ枠または別のコード・エディタのウィンドウでストアド・プロシージャ、関数、トリガ、およびイベントを編集できます。

- ツールバーのボタンに、選択されているオブジェクト固有のオプションが含まれるようになりました。
- SQL 文のログとサーバ・メッセージは、Sybase Central のメイン・ウィンドウに直接表示できます (サーバ・メッセージ・ウィンドウに同じ情報が表示されます)。この情報を表示するには、Sybase Central で、[ビュー] – [設計の詳細] を選択します。Sybase Central のメイン・ウィンドウの下部に、[設計の詳細] ウィンドウ枠が表示されます。
- Adaptive Server Anywhere プラグインにはさまざまな新しいウィザードが追加されており、その指示に従うことで、テーブル、ユニークな制約、Web サービスを作成できます。
- **Adaptive Server Anywhere プラグインにおけるクリップボード・サポートの強化** Adaptive Server Anywhere プラグインにおけるクリップボード・サポートが強化され、Sybase Central 内の大半のオブジェクトを、Interactive SQL やテキスト・エディタなど、ほかのアプリケーションにコピーして貼り付けることができます。オブジェクトをほかのアプリケーションにコピーする場合は、選択するオブジェクトに応じて、オブジェクトの名前または SQL が表示されます。たとえば、Sybase Central の中でインデックスをコピーし、それをテキスト・エディタに貼り付ける場合、そのインデックスの CREATE INDEX 文が表示されます。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「Sybase Central におけるデータベース・オブジェクトのコピー」を参照してください。

- **デバッガの変更** ストアド・プロシージャと Java クラスの両方をデバッグできるデータベース・オブジェクト・デバッガが、Sybase Central に統合されました。ユーザ・インタフェースは再設計されています。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「データベースでのデバッグ論理」を参照してください。

- **Adaptive Server Anywhere Console ユーティリティの強化** インタフェースの変更、複数接続、ソート、ドラッグ・アンド・ドロップのサポートなど、Adaptive Server Anywhere コンソール・ユーティリティが多くの点で強化されました。
- **Sybase Central と Interactive SQL の高速起動** Windows 上で Sybase Central または Interactive SQL を起動するときの、アプリケーションの起動時間の短縮を目的とする高速ランチャが、Sybase Central と Interactive SQL に組み込まれています。Adaptive Server Anywhere 9.0 を実行すると、2つのバックグラウンド・プロセスが開始します。これは *dbisqlg.exe* のインスタンスと *scjview.exe* のインスタンスであり、Interactive SQL と Sybase Central の高速ランチャ・プロセスにそれぞれ相当します。これらの実行プログラムは両方とも、ユーザーがログインすると起動します。

詳細については、『SQL Anywhere Studio ヘルプ』> 「[オプション] ダイアログ : [一般] タブ」を参照してください。

- **Interactive SQL の構文を強調表示するエディタ** Interactive SQL の [オプション] ダイアログを使用すると、Interactive SQL の [SQL 文] ウィンドウ枠に入力される構文をどのように表示するかを設定できます。

詳細については、『SQL Anywhere Studio ヘルプ』> 「[オプション] ダイアログ : [エディタ] タブ」を参照してください。

- **Interactive SQL からの印刷** Interactive SQL の [SQL 文] ウィンドウ枠やグラフィカル・プランの内容を印刷できます。

詳細については、『SQL Anywhere Studio の紹介』> 「Interactive SQL メイン・ウィンドウの説明」を参照してください。

- **グラフィカルなプランの強化** グラフィカルなクエリ・アクセス・プランの表示が、多くの点で強化されました。
 - あるオペレータから別のオペレータへ渡されるローの数は、線の太さを変えることで表されます。

- 低速なオペレーションは、赤色の枠線で強調表示されます。
- 統計情報の表示が拡張され、再編成されています。
- アクセス・プランを印刷できます。
- **データベース・ユーティリティで @filename パラメータを使用できる** Interactive SQL [dbisql]、言語ユーティリティ [dblang]、Adaptive Server Anywhere コンソール・ユーティリティ [dbconsole] を除くすべてのデータベース管理ユーティリティでは、@file 構文を使用して、ファイルに含まれるパラメータを指定できるようになりました。ファイル名はコマンド・ラインの任意の位置に指定でき、ファイルに含まれるパラメータがその位置に挿入されます。複数のファイルの指定も可能であり、ファイル指定子をコマンド・ライン・スイッチと一緒に使用することもできます。なお、@file 構文は再帰的でないことに注意してください。

詳細については、『ASA データベース管理ガイド』> 「@data サーバ・オプション」を参照してください。

- **Interactive SQL で結果の横にロー番号を表示できる** Interactive SQL には、結果の横にロー番号を表示するオプションがあります。このオプションは、Interactive SQL の [オプション] ダイアログの [結果] タブで設定します。

詳細については、『SQL Anywhere Studio ヘルプ』> 「[オプション] ダイアログ : [結果] タブ」を参照してください。

- **Interactive SQL を .SQL ファイルのデフォルト・エディタとして設定できる** Windows プラットフォームにおいて、.SQL ファイルをダブルクリックすると Interactive SQL がファイルを開くように、ファイルの関連付けを作成できます。

詳細については、『ASA データベース管理ガイド』> 「Interactive SQL ユーティリティ」を参照してください。

- **Interactive SQL の [コマンド履歴] ダイアログの強化**

Interactive SQL の [コマンド履歴] ダイアログでは、ウィンドウで複数のコマンドを選択できることに加えて、コマンドをコピーしたり削除したりできます。コマンド履歴は、Interactive SQL セッション間で保持されます。

詳細については、『SQL Anywhere Studio の紹介』> 「SQL 文の印刷」を参照してください。

- **警告メッセージの W プレフィクス** 9.0 より以前のバージョンでは、警告メッセージとエラー・メッセージのプレフィクスはすべて、I または E でした。バージョン 9.0 では、警告メッセージのプレフィクスは W です。この変更は、dbmlsrv9、dbmlsync、dbremote、ssremote、dbltm、ssqueue に影響します。

Mobile Link の新機能

次に、バージョン 9.0.0 で導入したソフトウェアに加えられた変更と追加を示します。

- **サーバ起動同期** サーバ起動同期の機能を使用すると、統合データベースから Mobile Link 同期を起動できます。つまり、データ更新をリモート・データベースに送信できます。Mobile Link コンポーネント (Notifier) は、同期を起動する統合データベース内の変更の種類と、更新されたメッセージを受信するリモートの選択方法を決定するための、プログラム可能なオプションを備えています。リモート・コンポーネント (リスナ) では、リモートの応答方法を決定します。

詳細については、『Mobile Link サーバ起動同期ユーザーズ・ガイド』> 「サーバ起動同期の概要」を参照してください。

- **ファイルベースのダウンロード** ダウンロードをファイルとして処理できるようになりました。このため、電子メール、ftp、ディスク、マルチキャストによるファイル配布など、ファイルを任意の方法で配布できます。このリリースの場合、この機能を使用できるのは Adaptive Server Anywhere リモート・データベースだけです。

詳細については、『Mobile Link 管理ガイド』> 「ファイルベースのダウンロード」を参照してください。

Mobile Link 同期 サーバの強化

- **接続スクリプト `begin_publication` と `end_publication` の追加** 2つの新しいスクリプトが追加されました。これらが使用されるケースの1つは、ファイルベースのダウンロードの実装です。

詳細については、『Mobile Link 管理ガイド』> 「`begin_publication` 接続イベント」と『Mobile Link 管理ガイド』> 「`end_publication` 接続イベント」を参照してください。

- **接続スクリプト `authenticate_parameters` の追加** カスタム認証を行うための新しいスクリプトが追加されました。認証の中でこの新しいスクリプトが起動されてから、`begin_synchronization` スクリプトが起動されます。

詳細については、『Mobile Link 管理ガイド』> 「`authenticate_parameters` 接続イベント」を参照してください。

- **文字列に埋め込まれたブランクを削除するオプションの追加**
`dbmlsrv9 -b` オプションを指定すると、同期の際、VARCHAR または LONG VARCHAR データ型のカラムの文字列から、後続ブランクが削除されます。

詳細については、『Mobile Link 管理ガイド』> 「`-b` オプション」を参照してください。

- **.old 拡張子を持つ新しいログ・ファイルを開始するオプション** `dbmlsrv9 -on` オプションを指定すると、Mobile Link サーバ・ログが使用するディスク領域について、厳しい制限を課すことができます。

詳細については、『Mobile Link 管理ガイド』> 「`-on` オプション」を参照してください。

- **ログの進行状況オフセット** Mobile Link 同期サーバでは、進行状況のオフセット、最終アップロード時間、最終ダウンロード時間をレポートできます。この情報を入手するには、`dbmlsrv9` の `-vp` オプションまたは `-v+` オプションを使用します。

詳細については、『Mobile Link 管理ガイド』> 「-v オプション」を参照してください。

- **.NET と Java 同期論理におけるエラーと警告の処理** Mobile Link 同期サーバでエラーと警告を処理するための論理を追加できます。

詳細については、『Mobile Link 管理ガイド』> 「Java での Mobile Link サーバ・エラーの処理」と『Mobile Link 管理ガイド』> 「.NET での Mobile Link サーバ・エラーの処理」を参照してください。

- **Mobile Link システム・テーブルへの追加** ml_user テーブルと ml_subscription テーブルに、last_upload_time と last_download_time という 2 つの新しいカラムが追加されました。デフォルトは NOT NULL で、デフォルト時刻は 1900 年 1 月 1 日 00:00:00 です。

また、ml_subscription には、subscription_id カラムも追加されました。publication_name カラムには、パブリケーション名が格納されます。

詳細については、『Mobile Link 管理ガイド』> 「ml_user」と『Mobile Link 管理ガイド』> 「ml_subscription」を参照してください。

Adaptive Server Anywhere クライアントの強化

- **アップロードのみの同期処理** アップロードのみの同期処理を選択できるようになりました。

詳細については、dbmlsync に関する『Mobile Link クライアント』> 「-uo オプション」を参照してください。

- **ダウンロードのみの同期処理** ダウンロードのみの同期処理を選択できるようになりました。

詳細については、『Mobile Link クライアント』> 「-ds オプション」と『Mobile Link クライアント』> 「DownloadOnly (ds) 拡張オプション」を参照してください。

- **ウィンドウ・メッセージによる同期の起動** ウィンドウ・メッセージを `dbas_synchronize` として登録し、これを `dbmlsync` のトップレベル・ウィンドに送信することで、`dbmlsync` を起動し、同期処理を実行できます。
- **起動時に dll をロード (Windows CE)** 新しい `dbmlsync -pd` オプションを使用すると、起動時にロードする DLL を指定できます。Windows CE 上で `dbmlsync` を使用する場合は、このオプションを使用してください。

詳細については、『Mobile Link クライアント』> 「-pd オプション」を参照してください。

- **スキーマをアップグレードまたは変更する手段の追加**

`dbmlsync` の `-i` オプションと `SiteScriptName (sn)` 拡張オプションに代わって、`sp_hook_dbmlsync_schema_upgrade` フック・ストアド・プロシージャが追加されました。

詳細については、『Mobile Link クライアント』> 「`sp_hook_dbmlsync_schema_upgrade`」を参照してください。

- **Mobile Link のリターン・コード** `dbmlsync` セッションの中で複数の同期処理を実行している場合は特に、その同期処理の成功や失敗を追跡し、記録することを支援するために、`sp_hook_dbmlsync_process_return_code` という新しいクライアント・イベント・フック・プロシージャが追加されました。また、`sp_hook_dbmlsync_abort` フックの `#hook_dict` テーブルには、新しい値 (リターン・コード) が設定されます。

詳細については、『Mobile Link クライアント』> 「`sp_hook_dbmlsync_process_return_code`」と『Mobile Link クライアント』> 「`sp_hook_dbmlsync_abort`」を参照してください。

- **スケジューリングの強化** スケジュールを指定する場合、新しい `HoverRescanThreshold (hrt)` 拡張オプションまたは `sp_hook_dbmlsync_log_rescan` フックを使用することで、ログをスキャンする時間を短縮できます。

詳細については、『Mobile Link クライアント』> 「HoverRescanThreshold (hrt) 拡張オプション」と『Mobile Link クライアント』> 「sp_hook_dbmlsync_log_rescan」を参照してください。

英語以外の言語でも、スケジュールの中で省略形の英語の曜日名を使用できるようになりました。以前のバージョンでは、英語以外の言語のスケジュールでは、フルスペルの英語の曜日名を指定する必要がありました。

スケジュール構文に、2つの新しいキーワードが追加されました。キーワードの1つは **INFINITE** です。これは、dbmlsync に対して、次の同期が通知されるまでいつまでも待つことを指定します。もう1つのキーワードは **0** です。日付としての **0** はその月の最後の日を指定します。

詳細については、『Mobile Link クライアント』> 「Schedule (sch) 拡張オプション」を参照してください。

Ultra Light クライアントの強化

- **HotSync conduit のトラブルシューティング支援の強化**
HotSync ログにトラブルシューティング情報を記録するように、HotSync conduit を設定できます。

詳細については、『Mobile Link クライアント』> 「HotSync ログ・ファイル」を参照してください。

パフォーマンスとモニタリングの強化

- **スキーマの変更がない場合の dbmlsync のパフォーマンス向上**
dbmlsync はデフォルトで、各同期の前にスキーマ情報をロードしてなくなりました。これによって、低速なハンドヘルド・デバイスにおける同期が、一般に 20 秒早くなります。

詳細については、『Mobile Link クライアント』> 「-sc オプション」を参照してください。

- **Windows CE 上での dbmlsync のパフォーマンスの向上**
dbmlsync は、Windows CE 上で dbtool9.dll を使用しなくなりました。つまり、メモリの使用量が少なくなります。
- **Mobile Link モニタのコマンド・ライン・オプション**
Mobile Link モニタは、さまざまなオプションを指定して、コマンド・ラインから起動できるようになりました。

詳細については、『Mobile Link 管理ガイド』> 「Mobile Link モニタの起動」を参照してください。

- **リダイレクタの強化** 新しく追加された LOG_LEVEL パラメータで、冗長性レベルを制御できます。

詳細については、『Mobile Link 管理ガイド』> 「リダイレクタのプロパティの設定 (すべてのバージョン共通)」を参照してください。

- **活性の向上** TCP/IP を介して接続している場合、切断された接続がこれまでより早く検出されるようになりました。接続が切断されたときに Mobile Link ワーカー・スレッドが迅速に解放されるため、スループットが向上します。

その他

- **警告メッセージの W プレフィクス** 9.0 より以前のバージョンでは、警告メッセージとエラー・メッセージのプレフィクスはすべて、I または E でした。バージョン 9.0 では、警告メッセージのプレフィクスは W です。この変更は、dbmlsrv9、dbmlsync、dbremote、ssremote、dbltm、ssqueue に影響します。

SQL Remote の新機能

SQL Remote バージョン 9.0 には、次の新機能が追加されています。

- **警告メッセージの W プレフィクス** 9.0 より以前のバージョンでは、警告メッセージとエラー・メッセージのプレフィクスはすべて、I または E でした。バージョン 9.0 では、警告メッセージのプレフィクスは W です。この変更は、dbmlsrv9、dbmlsync、dbremote、ssremote、dbltm、ssqueue に影響します。

Ultra Light の新機能

Ultra Light 開発は、2 種類のプログラミング・インタフェースを使用して行うことができます。

- **Ultra Light コンポーネント** Ultra Light コンポーネントは、アプリケーション短期開発ツールのユーザに、Ultra Light データベースと同期の機能をもたらします。これらのコンポーネ

ントには、サポートされている開発ツールごとに使い慣れたインタフェースが用意されています。Ultra Light コンポーネントは、簡単なテーブル・ベースのデータ・アクセス・インタフェースのほか、より複雑なクエリ用として動的 SQL も備えています。

Ultra Light コンポーネントは、バージョン 8.0.2 で導入されました。

コンポーネントの概要については、『Ultra Light データベース・ユーザーズ・ガイド』>「Ultra Light の概要」を参照してください。

- **静的開発モデル** Embedded SQL、静的型 C++ API、静的型 Java API は引き続き使用可能です。マニュアルの中では、コンポーネントと区別するために静的インタフェースとして参照されています。

特に次の点に注意してください。

- 「**Native Ultra Light for Java**」は、C/C++ Ultra Light Runtime を使用する Ultra Light コンポーネントです。Ultra Light の「**静的型 Java API**」は、以前のリリースで使用可能な pure Java ソリューションであり、コンパイル時にクエリを指定する必要があります。
- 「**Ultra Light for C++**」は、コンポーネント・インタフェースです。Ultra Light の「**静的型 C++ API**」は、以前のリリースで使用可能な静的インタフェースであり、コンパイル時にクエリを指定する必要があります。
- 「**Embedded SQL**」は静的インタフェースであり、コンパイル時にクエリを指定する必要があります。

次に、バージョン 9.0 で導入したソフトウェアに加えられた変更と追加を示します。

- **新しいコンポーネント** AppForge MobileVB、eMbedded Visual Basic、Java のコンポーネントに加えて、次のコンポーネントが導入されました。

- **Ultra Light .NET** Visual Studio .NET 環境を使用する開発用のコンポーネント。このコンポーネントで構築したアプリケーションは、.NET Compact Framework (バージョン 1.05.0000 以上) をサポートするデバイスに配備できます。

詳細については、『Ultra Light.NET ユーザーズ・ガイド』>「Ultra Light.NET の概要」を参照してください。

- **C++ コンポーネント** C++ コンパイラを使用する開発用のコンポーネント。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』>「C/C++ 開発者用 Ultra Light の概要」を参照してください。

- **Pocket IE のサポート** eMbedded Visual Basic コンポーネントが ActiveX コンポーネントにアップグレードされました。JScript を使用する開発、つまり、Windows CE デバイス上の Pocket IE から実行するアプリケーションに対するサポートが追加されました。

詳細については、『Ultra Light ActiveX ユーザーズ・ガイド』>「Ultra Light ActiveX の概要」を参照してください。

- **動的 SQL** Ultra Light コンポーネントでは、バージョン 8.0.2 で提供されていたテーブルベースのデータ・アクセス・インタフェースに加えて、マルチテーブル・ジョインなど、より複雑なクエリに対して動的 SQL を使用できるようになりました。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』>「動的 SQL」を参照してください。

- **接続パラメータ** Ultra Light コンポーネント (C++ を除く) の接続パラメータは、単一の文字列ではなく個々のプロパティとして公開されています。このため、接続の問題をデバッグしやすくなり、より簡単に接続を管理できるようになりました。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』>「接続パラメータ」を参照してください。

- **MobileVB コンポーネントのドラッグ・アンド・ドロップ**
MobileVB コンポーネントをフォームにドラッグできるようになりました。コンポーネントのプロパティは、コードと同様、設計環境で設定できます。

詳細については、『Ultra Light for MobileVB ユーザーズ・ガイド』> 「Ultra Light for MobileVB のアーキテクチャ」を参照してください。

- **マルチプロセス・アクセス** C++ コンポーネントは、複数のプロセスからのアクセスをサポートします。このモデルを使用するアプリケーションを開発するには、個々の Ultra Light データベース・エンジンとアプリケーションを、別の Ultra Light ランタイム・ライブラリにリンクする必要があります。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「アプリケーションのコンパイルとリンク」を参照してください。

- **同時同期処理** 以前のリリースでは、同期中のデータへのフル・アクセスは禁止されていました。現在のリリースでは、同期のダウンロード・フェーズでは、データにフル・アクセスできます。アップロード・フェーズでは、読み込み専用アクセスが可能です。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「Ultra Light での同時実行性について」を参照してください。

- **Palm OS の強化** Palm OS 上での Ultra Light コードの構造が再編成され、Palm データベース・セグメントを適切に使用できるようになりました。
- **エラー情報の拡張** Ultra Light コンポーネントを使用して構築されたアプリケーションに対して、より多くのエラー情報が提供されます。
- **Windows NT/2000/XP 上で使用可能なユニコード・ライブラリ** Embedded SQL と静的型 C++ API アプリケーション用に、ユニコード・バージョンの Ultra Light ランタイム・ライブラリが用意されています。このバージョンを使用するのは Ultra

Light コンポーネントです。このライブラリを使用すると、Ultra Light データベース・ファイルは、Windows CE とデスクトップ・オペレーティング・システムとの互換性を持ちます。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「Ultra Light の文字セット」を参照してください。

- **配備プラットフォームとして Windows XP をサポート**

Windows XP への Ultra Light アプリケーションの配備がサポートされるようになりました。

バージョン 9.0 での動作の変更

この項では、SQL Anywhere Studio バージョン 9.0 のコンポーネントに導入された動作の変更について説明します。

Adaptive Server Anywhere の動作の変更

ここでは、これまでのバージョンとは異なる動作をリストにして説明します。

- **データベース内の Java オブジェクトのサポート終了** Java オブジェクトとしてのデータの保存はサポートされなくなりました。Java ストアド・プロシージャは引き続きサポートされます。

データベース内の Java の最新サポートについては、『ASA プログラミング・ガイド』> 「データベースにおける Java の概要」を参照してください。

- **Windows 環境用の新しいギリシャ語照合の追加** 以前のバージョンに OEM/DOS 文字セットのギリシャ語照合はありましたが、Windows 用のギリシャ語照合 1253ELL が新しく追加されました。ギリシャ語の Windows 環境でデータベースを新しく作成する場合、照合を指定しないと、1253ELL が自動的に選択されます。

詳細については、『ASA データベース管理ガイド』> 「提供されている照合と推奨する照合」を参照してください。

- **新しい接続制限** データベース・サーバは、接続制限より 1 つ余分に DBA 接続を許可します。これは、意図的なサービス拒否や不用意なサービス拒否があった場合に、接続してそれらの接続を切断できるようにするためです。

詳細については、『ASA データベース管理ガイド』> 「-gm サーバ・オプション」を参照してください。

- **単一のプロセッサに限定されたパーソナル・データベース・サーバ** ソフトウェアの以前のバージョンでは、パーソナル・データベース・サーバは、要求を処理するために最大 2 つの CPU を使用していました。現在、パーソナル・サーバは単一のプロセッサに限定されています。
- **FROM 句で指定されているテーブル式への参照を、ネストしている外部ジョインの ON 句の中で使用できる** 以前のリリースでは、ON 句内の外部参照が可能でした。現在のリリースでは、このような外部参照を LATERAL キーワードで指定する必要があります。このような制限を課すことで外部参照であることが明白になり、SQL/99 規格にも準拠します。

次のクエリは、LATERAL キーワードを使用せずに外部参照 (強調表示されている部分) を指定しているため、現在のバージョンでは正しくありません。

```
SELECT *  
  FROM T1,  
        T2 LEFT OUTER JOIN  
                ( T3 LEFT OUTER JOIN T4 ON  
T1.col1 = T2.col2 )  
        ON T1.col2 = T2.col2
```

詳細については、『ASA SQL リファレンス・マニュアル』> 「FROM 句」を参照してください。

- **非修飾テーブル参照に一致するものが複数ある場合、構文エラーとして報告される** 以前のリリースでは、所有者名の指定がないテーブルへの参照 (非修飾テーブル参照) がクエリに含まれていて、そのテーブルの可能性のあるものが複数ある場合、最初に見つかったものが使用されていました。現在のリリースでは、非修飾テーブル参照はエラーになります。『ASA エラー・メッセージ』> 「テーブル名 '%1' はあいまいです。」を参照してください。
- **NULL エスケープ文字を含む LIKE 演算子は NULL に評価される** NULL エスケープ文字を含む LIKE 演算子は NULL に評価されます。以前のリリースでは、NULL エスケープ文字を含む LIKE 演算子は、エスケープ文字がないものとして評価されていました。新しい動作は ISO/ANSI 規格に一致します。

- **プロパティと統計値の削除** ServerIdleWaits データベース・プロパティ、TaskSwitch 接続プロパティ、CurrTaskSwitch 接続プロパティが削除されました。また、各プロパティに対応するパフォーマンス・モニタ統計値の Context Switches、Server Idle Waits/sec、Request Queue Waits/sec も削除されました。
- **INSERT/UPDATE/DELETE 実行時のカラム統計値の更新**
INSERT 文、UPDATE 文、または DELETE 文を実行して大量のデータが変更された場合、統計値が更新されるようになりました。
- **リカバリ時には統計値が更新されない** リカバリ時、または単純な DELETE 文や UPDATE 文の実行時には、統計値は更新されなくなりました。単純な文とは、最適化されず、サーバが直接実行する文です。
- **正しいデータ型で表示されるヒストグラム範囲**
sa_get_histogram() システム・プロシージャとヒストグラム [dbhist] ユーティリティは、これまではハッシュ値で出力範囲を表示していました。現在のバージョンでは、出力されるヒストグラムの範囲は対応するカラム内のデータに一致し、正しいデータ型で表示されます。

詳細については、『ASA データベース管理ガイド』> 「ヒストグラム・ユーティリティ」と『ASA SQL リファレンス・マニュアル』> 「sa_get_histogram システム・プロシージャ」を参照してください。
- **リモート・データベースあたり 1 人の統合ユーザのみ可能** 同じリモート・データベースに複数の統合ユーザを定義できなくなりました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「GRANT CONSOLIDATE 文 [SQL Remote]」または『ASA SQL リファレンス・マニュアル』> 「REVOKE CONSOLIDATE 文 [SQL Remote]」を参照してください。
- **明示的に指定されていない場合、CommLinks 接続パラメータは共有メモリを使用する** CommLinks 接続パラメータが指定されていない接続は、常に共有メモリを介して接続を試みるようになりました。

詳細については、『ASA データベース管理ガイド』>
「CommLinks 接続パラメータ [LINKS]」を参照してください。

- **CommLinks 接続パラメータは、常に共有メモリ・プロトコルを最初に試行する** CommLinks=all を指定すると、常に共有メモリ・プロトコルを使用して接続が試行され、その後に別のプロトコルを使用して接続が試行されます。

詳細については、『ASA データベース管理ガイド』>
「CommLinks 接続パラメータ [LINKS]」を参照してください。

- **接続エラーによるプロセスの中止** 以前のバージョンでは、接続が確立されるまで、CommLinks 接続パラメータにリストされている接続プロトコルが 1 つずつ試行されました。現在のバージョンでは、プロセスの最中に接続エラーが発生すると、リストされているプロトコルがすべて試行されたかどうかにかかわらず、接続プロセスが即時に中止されます。

詳細については、『ASA データベース管理ガイド』>
「CommLinks 接続パラメータ [LINKS]」を参照してください。

- **PREVENT_ARTICLE_PKEY_UPDATE のデフォルト値の変更** プライマリ・キー値の更新を回避するために、PREVENT_ARTICLE_PKEY_UPDATE データベース・オプションのデフォルト値が ON に変更されました。新しいデフォルト設定によって、パブリケーションに含まれるプライマリ・キーにおける、プライマリ・キーの更新は認められません。値を OFF に設定することで、この機能を無効にできます。

詳細については、『ASA データベース管理ガイド』>
「PREVENT_ARTICLE_PKEY_UPDATE オプション [データベース]」を参照してください。

- **非確定として扱われる関数** RAND 関数、NEWID 関数、GET_IDENTITY 関数は、非確定として扱われます。この結果、これらの関数はクエリの実行時にキャッシュされません。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「関数のキャッシュ」を参照してください。

- **パフォーマンス・メッセージにデータベース名を表示** エンジン・パフォーマンス・アドバイス・メッセージにデータベース名が表示されるようになりました。これは、複数のデータベースを実行しているときに特に便利です。また、*Note* という文字列で始まるメッセージは、これがアドバイス・メッセージであることを表します。
- **Adaptive Server Anywhere バージョン 9.0 以前を使用している NetWare クライアントはアップグレードが必要** Adaptive Server Anywhere における NetWare サポートの強化の結果、Adaptive Server Anywhere バージョン 9.0 以前を使用している NetWare クライアントは、特定の EBF をインストールしていないかぎり、共有メモリを使用して 9.0 サーバに接続できません。7.0.4.3400、8.0.0.2358、8.0.1.3088、8.0.2.4095 より以前のビルド番号のクライアントは、どれも 9.x サーバを検出できません。
- **ALTER DATABASE CALIBRATE の構文の変更** ALTER DATABASE CALIBRATE TEMPORARY DBSPACE の構文が、ほかの類似の文の構文に合わせて、ALTER DATABASE CALIBRATE DBSPACE TEMPORARY に変更されました。

詳細については、『ASA SQL リファレンス・マニュアル』>「ALTER DATABASE 文」を参照してください。

- **積極的に行われる動的キャッシュ・サイズ決定** 新しいデータベースを起動した後、またはファイルのサイズがきわめて大きくなった場合に、動的キャッシュ・サイズ決定が積極的に行われてキャッシュのサイズを変更します。この変更が加えられる前は、最高でも 1 分間に 1 回だけ統計値がサンプリングされ、キャッシュのサイズが変更されていました。現在のバージョンでは、データベースを起動した後、またはファイルがきわめて大きくなった場合に、5 秒から 30 秒ごとに統計値がサンプリングされ、キャッシュのサイズが変更されます。

詳細については、『ASA SQL ユーザーズ・ガイド』>「パフォーマンス向上へのキャッシュの使用」を参照してください。

- **インタフェースとメッセージの言語の決定** ASLANG と ASCHARSET という 2 つの環境変数が、インタフェース (Sybase Central や Interactive SQL) とメッセージで使用される言語を制御します。ASLANG は言語を指定し、ASCHARSET は文字セットを指定します。

詳細については、『ASA データベース管理ガイド』> 「ASLANG 環境変数」または『ASA データベース管理ガイド』> 「ASCHARSET 環境変数」を参照してください。

- **返されるローが rowcount 設定で制限される** rowcount 設定によって、カーソルが返す先頭からのローが制限されます。現在のバージョンでは、絶対フェッチを使用して結果の先頭に移動できなくなりました。

このような動作が必要な場合は、TOP N / START AT という新しい機能を使用することで、その動作を実現できます。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「上位 N のソート」を参照してください。

今後サポートを終了する予定の機能およびサポートを終了した機能

Adaptive Server Anywhere -d サーバ・オプションのサポート終了

Adaptive Server Anywhere の NetWare サポートが強化された結果、-d サーバ・オプションはサポートを終了しました。

NetWare 4.x のサポート終了 Adaptive Server Anywhere の NetWare サポートが強化された結果、Adaptive Server Anywhere は、NetWare バージョン 5.1 SP6 以上、またはバージョン 6.0 SP3 以上でのみ動作します。適切なサービス・パックをインストールしてください。適切なサービス・パックがインストールされていないと、Adaptive Server Anywhere サーバからエラー・メッセージが表示されます。

SQLLOCALE 環境変数のサポート終了 SQLLOCALE 環境変数に代わって、ASLANG と ASCHARSET という 2 つの新しい環境変数が使用されるようになりました。

詳細については、「[バージョン 9.0 での動作の変更](#)」125 ページを参照してください。

Mobile Link の動作の変更

ここでは、これまでのバージョンとは異なる動作をリストにして説明します。

- **dbmlsync の -i オプションと SiteScriptName 拡張オプションのサポート終了** dbmlsync -i と dbmlsync -e sc は、サポートされなくなりました。代わりに、sp_hook_dbmlsync_schema_upgrade という新しいフックが使用されます。

詳細については、『Mobile Link クライアント』>
「sp_hook_dbmlsync_schema_upgrade」を参照してください。

- **ダウンロード確認がデフォルトでは OFF** Adaptive Server Anywhere リモートの場合、SendDownloadAck 拡張オプションのデフォルト値は OFF です。Ultra Light の場合、ul_synch_info 構造体の send_download_ack フィールドのデフォルト値は ul_false です。

バージョン 9 にアップグレードする場合、ダウンロード・トランザクションのコミット前に、リモートがダウンロードを適用したことをアプリケーションが確認する必要がある場合は、このオプションを明示的に ON にしてください。

詳細については、『Mobile Link クライアント』>
「SendDownloadACK (sa) 拡張オプション」と『Mobile Link クライアント』> 「Send Download Acknowledgement 同期パラメータ」を参照してください。

- **Windows CE デバイス上でデフォルトでは動作しない dbmlsync フックがある** dbmlsync の LockTables 拡張オプションが変更され、テーブルを共有モードでロックするか、または排他モードでロックするかを指定できるようになりました。LockTables のデフォルト設定は ON であり、Windows CE 以外のすべてのプラットフォームで、テーブルを共有モードでロックし続けます。しかし、Windows CE デバイスの場合、

ON は、テーブルを排他モードでロックすることを意味します。この変更によって、Windows CE アプリケーションのパフォーマンスが大幅に向上します。

dbmsync のイベント・フックである
sp_hook_dbmsync_download_com_error、
sp_hook_dbmsync_download_fatal_sql_error、
sp_hook_dbmsync_download_log_ri_violation はすべて、個別の
接続で実行されます。これらのフックが排他モードでロック
されている同期テーブルへのアクセスを試みても、正しく実
行できません。Windows CE 上での配備の中でこのいずれかの
フックを使用する場合は、LockTables を SHARE に設定する
必要があります。

詳細については、『Mobile Link クライアント』> 「LockTables
(It) 拡張オプション」、『Mobile Link クライアント』>
「sp_hook_dbmsync_download_com_error」、『Mobile Link クラ
イアント』> 「sp_hook_dbmsync_download_fatal_sql_error」、
および 『Mobile Link クライアント』>
「sp_hook_dbmsync_download_log_ri_violation」を参照してく
ださい。

- **Mobile Link 同期サーバのエラー・コード** Mobile Link 同期
サーバから、エラーに関する詳しい情報が出力されるよう
になりました。Mobile Link サーバ・エラー・コードは -10001 か
ら始まり、すべてが -10000 未満です。dbmsync の場合、エ
ラーは GUI と出力ファイルに出力されます。Ulra Light の場
合、エラーは ul_synch_info 構造体の中の文字列として有効で
す。

詳細については、『ASA エラー・メッセージ』> 「Mobile Link
同期サーバのエラー・メッセージ」を参照してください。

- **今後サポートを終了する予定のアップロード・カーソル**

upload_cursor、new_row_cursor、old_row_cursor の各スクリプ
トは今後サポートを終了する予定です。アップロード・スト
リームには文ベースのスクリプトを使用してください。

詳細については、『Mobile Link 管理ガイド』> 「ローをアップ
ロードするスクリプトの作成」を参照してください。

- **サポート終了予定の -zac と -zec** Mobile Link 同期サーバの、カーソルベースのスクリプトを生成する -zac と -zec オプションは廃止される予定です。-za と -ze を引き続き使用することで、文ベースのスクリプトを自動的に生成できます。

詳細については、『Mobile Link 管理ガイド』> 「-za オプション」と『Mobile Link 管理ガイド』> 「-ze オプション」を参照してください。

- **-zd の削除** Mobile Link 同期サーバの、last_download タイムスタンプを最後に渡すことを指定する -zd オプションが削除されました。この結果、このパラメータは常に最初に渡されるようになりました。

- **サポート終了予定の mlxtract** mlxtract ユーティリティは今後サポートを終了する予定です。

詳細については、『Mobile Link クライアント』> 「リモート・データベースの作成」を参照してください。

- **end_synchronization スクリプトが常に呼び出される** 9.0 より以前のバージョンでは、同期が失敗したときに end_synchronization スクリプトが呼び出されないことがありました。現在のバージョンでは、begin_synchronization スクリプトが呼び出されていれば、スクリプトは常に呼び出されます。つまり、同期が成功したかどうかに関わらず、end_synchronization スクリプトに設定したクリーンアップ・アクティビティは実行されます。

また、end_synchronization スクリプトには sync_ok という新しいパラメータが追加されています。このパラメータは、同期の成功 (1) または失敗 (0) を表します。

詳細については、『Mobile Link 管理ガイド』> 「end_synchronization 接続イベント」と『Mobile Link 管理ガイド』> 「end_synchronization テーブル・イベント」を参照してください。

- **ストリーム dll と共有オブジェクトの名前変更** ストリーム dll と共有オブジェクトの名前が変更され、Adaptive Server Anywhere との統一が図られました。次の表は、変更の詳細を示します。

以前の名前	新しい名前
dbhttp9	dbmlhttp9
dbhttps9	dbmlhttps9
dbjrsa9	dbmljrsa9
dbjtls9	dbmljtls9
dbrsa9	dbmlrsa9
dbsock9	dbmlsock9
dbtls9	dbmltls9

詳細については、『Mobile Link 管理ガイド』> 「Mobile Link アプリケーションの配備」を参照してください。

- **ScoutSync のサポート終了** ScoutSync はサポートされなくなりました。
- **同期のたびにスキーマ情報は再ロードされない** 9.0 より以前のバージョンでは、同期のたびにデータベースからスキーマ情報が再ロードされていました。現在のバージョンでは、dbmlsync の起動時にのみ、スキーマ情報は再ロードされます。dbmlsync -sc オプションを使用すると、以前の動作に戻すことができます。-sc オプションを使用していない場合は、スキーマ変更がリモート・データベースに加えられる前に、dbmlsync をシャット・ダウンする必要があります。シャット・ダウンせずにスキーマ変更を加えると、同期エラーが発生する可能性があります。または、dbmlsync が予期せぬ動作をすることがあります。

詳細については、『Mobile Link クライアント』> 「-sc オプション」を参照してください。

- **主要なスクリプトがない場合は同期が中止される** 9.0 より以前のバージョンでは、特定のスクリプトがないためにデータの損失を引き起こす可能性がある場合でも、同期は続けられました。現在のバージョンでは、このような場合に Mobile Link は中止するようになりました。dbmlsrv9 -fr オプションを使用すると、失敗の代わりにエラーを出力できます。

詳細については、『Mobile Link 管理ガイド』> 「-fr オプション」を参照してください。

- **keep_alive 同期パラメータは常に ON** TCP/IP と HTTP プロトコルの keep_alive 同期パラメータは無視され、常に ON に設定されています。これは、以前のデフォルト設定でした。TCP/IP 接続の活性を制御するには、liveness_timeout パラメータを使用します。

詳細については、『Mobile Link クライアント』> 「CommunicationAddress (adr) 拡張オプション」の liveness_timeout パラメータまたは『Mobile Link 管理ガイド』> 「-x オプション」を参照してください。

Ultra Light の動作の変更

ここでは、これまでのバージョンとは異なる動作をリストにして説明します。

- **サポートされるプラットフォームの変更** Ultra Light 配備プラットフォームのサポートは、次のように変更されました。
- **ScoutSync のサポート終了** ScoutSync 同期ソフトウェアはサポートされなくなりました。
- **VxWorks のサポート終了** VxWorks オペレーティング・システムはサポートされなくなりました。
- **pure Java Ultra Light では JDK 1.1.8 が必要** pure Java 静的型開発モデルでは、JDK 1.1.4 以上ではなく、JDK 1.1.8 以上が必要です。

- **Palm OS の変更** Palm OS 用の Ultra Light アーキテクチャの変更によって、最新のデバイスにおけるパフォーマンスが向上します。その結果、以前のリリースよりも多くの動的メモリが必要です。きわめて小さなデータベースを除いて、Palm OS バージョン 3.5 以上と 4 MB 以上のメモリを使用することをおすすめします。
- **MobileBuilder と PRC ツールのサポート中止** PenRight! MobileBuilder プラットフォーム上での Ultra Light の開発はサポートされなくなりました。GNU PRC Tool チェーンを使用しての開発もサポートされなくなりました。

- **開発プラットフォームの変更** Ultra Light コンポーネントのアプリケーション開発は、Windows NT/2000/XP でのみサポートされます。Windows 98 SE 上での静的インタフェースを使用している開発もサポートされます。それ以外の Windows 95/98/Me ファミリー・メンバは、開発目的ではサポートされません。

サポートされる Metrowerks CodeWarrior のバージョンは 8 と 9 です。

- **マニュアル内の用語の変更** Ultra Light コンポーネントの説明では、異なるインタフェースを区別するために、新しい名前が必要です。以前の Ultra Light インタフェース (Embedded SQL、C++ API、Java API) は、使用するクエリをコンパイル時に指定する必要があるために、「静的インタフェース」という名前が付けられています。コンポーネントは、「動的 SQL」へのアクセスを提供します。
- **Windows NT/2000/XP 上の Ultra Light ランタイム・ライブラリ** ActiveX と MobileVB コンポーネントは、Windows 上のユニコード・ランタイム・ライブラリを使用します。このランタイム・ライブラリは、Windows 版のバージョン 8.0.2 Ultra Light データベース (.udb) ファイルと互換性がありますが、ほかの Windows オペレーティング・システム上で構築されたバージョン 8.0.2 Ultra Light データベース・ファイルとの互換性はありません。
- **file_name パラメータ** ソフトウェアの以前のバージョンでは、デスクトップ上の Ultra Light データベース・ファイル名を指定するための file_name パラメータは、platform-specific パラ

メータが指定されていない場合は、デバイス上のファイル名を指定するためにも使用されていました。現在のバージョンでは、`file_name` パラメータは、デスクトップ・オペレーティング・システムを除いては無視されるようになりました。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「データベース識別パラメータ」を参照してください。

- **静的型 Java API の変更** 静的型 Java API が変更されました。`JdbcDatabase` オブジェクトにあった次のメソッドが、`JdbcConnection` オブジェクトに移動されました。

- `countUploadRows`
- `getLastDownloadTimeDate`
- `getLastDownloadTimeLong`

明示的な `JdbcManager` オブジェクトを持たないアプリケーションが使用できるように、`JdbcConnection` に `grant` メソッドと `revoke` メソッドが追加されました。

詳細については、『Ultra Light 静的型 Java ユーザーズ・ガイド』> 「`JdbcConnection` クラス」を参照してください。

- **エラー・コードの変更** 一部の Ultra Light エラー・コードが、より具体的で実用的な値に変更されました。アプリケーションの中で個々のエラー・コードをテストしている場合は、アップグレード後に新しいコードを確認してください。

たとえば、データベースに接続するときに `SQLE_DATABASE_NOT_FOUND` (または同等の Ultra Light インタフェースの 1 つ) をチェックしている場合は、これを `SQLE_ULTRALITE_DATABASE_NOT_FOUND` に変更する必要があります。

エラー・コードのリストについては、使用しているインタフェース内の SQL エラー・オブジェクトを参照してください。

- **Embedded SQL の UL_STORE_PARMS の変更**

UL_STORE_PARMS マクロは、EXEC SQL CONNECT 文の中で評価されるようになりました。データベースは dbinit 呼び出しの際に起動されるのではなく、接続時に起動されます。したがって、UL_STORE_PARMS は、複数の接続を使用している場合は、異なる回数だけ評価されます。また、EXEC SQL CONNECT 文の前に UL_STORE_PARMS を定義する必要があります。

第 5 章

バージョン 8.0.2 の新機能

この章の内容

この章では、SQL Anywhere Studio バージョン 8.0.2 で導入された新機能と変更された動作の概要について説明します。

バージョン 8.0.2 の新機能

この項では、SQL Anywhere Studio バージョン 8.0.2 のコンポーネントに導入された新機能について説明します。

Adaptive Server Anywhere の新機能

この項では、Adaptive Server Anywhere バージョン 8.0.2 の新機能について説明します。新機能について主要なものからそうでないものまですべてを示し、各機能の詳細についての参照先も記述しています。

- 新機能のハイライト**
- **クラスタード・インデックスのサポート** テーブルにクラスタード・インデックスを作成すると、そのテーブル内のローは、インデックス内での表示順序とほぼ同じ順序で格納されます。LOAD TABLE 文を使用して、クラスタされた順序でテーブルを情報とともにロードできます。情報をテーブルに挿入すると、テーブルのクラスタの特性が劣化します。REORGANIZE TABLE 文を使用して、クラスタ順序を回復できます。クラスタード・インデックスを使用すると、パフォーマンスが向上します。

このリリースより前に作成されたデータベースでクラスタード・インデックスを使用するには、データベースをアンロードして再ロードすることによってそのデータベースのファイル・フォーマットをアップグレードする必要があります。

詳細については、『ASA SQL ユーザーズ・ガイド』>「クラスタード・インデックスの使用」を参照してください。

- **ユニークな識別子のサポート** Adaptive Server Anywhere は、ユニークな識別子 (UUID と GUID) をサポートします。UUID (ユニバーサル・ユニーク識別子) と GUID (グローバル・ユニーク識別子) は、同期環境内の異なるデータベースにわたってローをユニークに識別するメカニズムです。

詳細については、『ASA SQL ユーザーズ・ガイド』>「NEWID デフォルト」を参照してください。

- **ON EXISTING 句での既存のローの更新** テーブルにプライマリ・キーがあるかぎり、INSERT 文の ON EXISTING 句を使用して既存のローを新しい値で更新できます。

詳細については、『ASA SQL ユーザーズ・ガイド』>「INSERT によるデータの変更」または『ASA SQL リファレンス・マニュアル』>「INSERT 文」を参照してください。

- **Windows CE での BACKUP 文のサポート** Adaptive Server Anywhere では、Windows CE プラットフォームで動作するデータベースのイメージ・バックアップを作成したり、データベースのトランザクション・ログの名前を変更またはトランケートしたりできます。

詳細については、『ASA データベース管理ガイド』>「バックアップの種類」または『ASA SQL リファレンス・マニュアル』>「BACKUP 文」を参照してください。

- **グラフィカルなプランの強化** グラフィカルなプランが拡張されて、より多くの情報が含まれるようになり、外観が新しくなりました。

詳細については、『ASA SQL ユーザーズ・ガイド』>「グラフィカルなプラン」を参照してください。

- **ワーク・テーブルの明示的な使用** プランでのワーク・テーブルができるかぎり最後まで使用されないようになりました。ワーク・テーブルが使用されているときは、グラフィカルなプランに明示的に表示されます。

詳細については、『ASA SQL ユーザーズ・ガイド』>「グラフィカルなプラン」または『ASA SQL ユーザーズ・ガイド』>「クエリ処理中のワーク・テーブルの使用」を参照してください。

- **新しいジョインの追加** このリリースに追加された新しいジョインには、ネスト・ループ・セミジョイン、ネスト・ループ非セミジョイン、ハッシュ・セミジョイン、ハッシュ非セミジョインがあります。

詳細については、『ASA SQL ユーザーズ・ガイド』>「ジョイン・アルゴリズム」を参照してください。

関数の強化

- **特定のカーソル・タイプの SQL クエリのプランを取得** PLAN、EXPLANATION、GRAPHICAL_PLAN の各関数を使用し、カーソル・タイプに基づいて SQL クエリのプランを取得できます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「GRAPHICAL_PLAN 関数 [その他]」、『ASA SQL リファレンス・マニュアル』> 「EXPLANATION 関数 [その他]」、または『ASA SQL リファレンス・マニュアル』> 「PLAN 関数 [その他]」を参照してください。

Interactive SQL でのこれらのプラン・オプションの設定については、『SQL Anywhere Studio ヘルプ』> 「[オプション] ダイアログ : [プラン] タブ」を参照してください。

- **文字セット変換関数** 新しい関数 CSCONVERT を使用して、文字列の文字セットを変換できます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「CSCONVERT 関数 [文字列]」を参照してください。

- **変数テスト関数** 新しい関数 VAREXISTS を使用して、特定の名前のユーザ定義変数が作成または宣言されているかどうかをテストできます。このテストを実行してから、必要に応じて変数を作成し、その変数を安全に使用できます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「VAREXISTS 関数 [その他]」を参照してください。

文の強化

- **プロシージャ・テキストを隠して論理の機密を保持** SET HIDDEN オプションを使用して、ストアド・プロシージャ、関数、トリガ、ビューに含まれる論理を隠すことができます。これによって、ストアド・プロシージャ、関数、トリガ、ビュー内の論理を明らかにせずに、アプリケーションやデータベースを配布できます。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「プロシージャ、関数、トリガ、ビューの内容を隠す」を参照してください。

- **LOAD TABLE で 1 バイトより長いデリミタが使用可能** LOAD TABLE 文は、255 バイトまでのデリミタをサポートするようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「LOAD TABLE 文」を参照してください。

- **Adaptive Server Enterprise と Microsoft SQL Server の互換性を提供する新しい文** DEALLOCATE 文を使用して、カーソルに関連付けられているリソースを解放できます。この文は、Adaptive Server Enterprise と Microsoft SQL Server の間の互換性のために用意されています。

詳細については、『ASA SQL リファレンス・マニュアル』>「DEALLOCATE 文」を参照してください。

- **dblog ユーティリティと同様の動作をする ALTER DATABASE 文** ALTER DATABASE 文を使用して、データベース・ファイルに関連付けられているトランザクション・ログとミラー・ログの名前を変更できます。以前は、これらのログの名前の変更はトランザクション・ログ (dblog) ユーティリティでのみ行うことができました。

詳細については、『ASA SQL リファレンス・マニュアル』>「ALTER DATABASE 文」を参照してください。

- **LOAD TABLE がグローバル・テンポラリ・テーブルとローカル・テンポラリ・テーブルの両方で使用可能** Adaptive Server Anywhere は、宣言されたローカル・テンポラリ・テーブルで LOAD TABLE 文をサポートするようになりました。以前は、グローバル・テンポラリ・テーブルのみがサポートされていました。

詳細については、『ASA SQL リファレンス・マニュアル』>「LOAD TABLE 文」を参照してください。

- **SET 文を使用した変数値の割り当てが可能** Transact-SQL プロシージャで SET 文を使用して、変数に値を割り当てることができるようになりました。
- **INSERT 文での WITH AUTO NAME のサポート** INSERT 文で WITH AUTO NAME を指定すると、SELECT リスト内の項目の名前によって、値と挿入先カラムの関連付けが決定されます。

詳細については、『ASA SQL リファレンス・マニュアル』>「INSERT 文」を参照してください。

- **EXIT 文の強化** Interactive SQL の EXIT 文は、Interactive SQL に終了コードを設定できるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』>「EXIT 文 [Interactive SQL]」を参照してください。

- **FROM 句のクエリに対する最適化ゴールの指定**
FASTFIRSTROW テーブル・ヒントを使用すると、OPTIMIZATION_GOAL オプションを first-row に設定しなくてもクエリの最適化ゴールを設定できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「FROM 句」を参照してください。

セキュリティの強化

- **新しいユーティリティを使用してファイルの内容を隠す** 設定ファイル (別名コマンド・ファイル) には、パスワードが含まれることがあります。強化されたセキュリティ機能として、Adaptive Server Anywhere にはファイル非表示ユーティリティという新しいユーティリティがあります。このユーティリティを使用すると、単純な暗号で設定ファイルの内容を隠すことができます。

詳細については、『ASA データベース管理ガイド』>「ファイル非表示ユーティリティ」を参照してください。

- **Certicom 暗号化の変更** ECC_TLS と RSA_TLS という 2 種類の Certicom 暗号化をサポートするようにセキュリティが強化されました。以前のバージョンの Adaptive Server Anywhere で Certicom 暗号化と呼ばれていた暗号化は、ECC_TLS 暗号化という名前に変更されました。Certicom パラメータは現在も受け入れられますが、ECC_TLS 暗号化と同等です。Adaptive Server Anywhere は、RSA_TLS 暗号化もサポートするようになりました。

詳細については、『ASA データベース管理ガイド』>「-ec サーバ・オプション」または『ASA データベース管理ガイド』>「Encryption 接続パラメータ [ENC]」を参照してください。

パフォーマンスの強化

- 新しい接続パラメータによるネットワークの応答性の向上**
 LazyClose と PrefetchOnOpen ネットワーク接続パラメータを使用すると、遅延時間の長いネットワークまたは多くの要求を処理するアプリケーションのあるネットワークで、パフォーマンスが向上します。

 これらのパラメータについては、『ASA データベース管理ガイド』> 「LazyClose 接続パラメータ [LCLOSE]」と『ASA データベース管理ガイド』> 「PreFetchOnOpen 接続パラメータ」を参照してください。
- Windows NT/2000/XP で分散読み込みが可能** 以前は、大きなテーブルの逐次スキャンでは、ページが 64 K バッファにコピーされてからキャッシュにコピーされていました。現在は、Windows NT サービス・パック 2 以上または Windows 2000/XP の環境で実行し、ページ・サイズが 4 K 以上であれば、分散読み込みによってページがキャッシュに直接コピーされるため、時間が節約され、パフォーマンスが向上します。

 詳細については、『ASA SQL ユーザーズ・ガイド』> 「適切なページ・サイズの使用」を参照してください。
- 要求のロギングでの時間単位の向上** プロシージャ・プロファイリングまたは要求のロギングを使用して取得される時間は、1 ミリ秒の単位になりました。この変更は、主に Windows オペレーティング・システムで稼働しているサーバに影響します。
- 複数バージョンのパフォーマンス・モニタの実行** 複数バージョンの Adaptive Server Anywhere を同時に実行している場合は、複数バージョンの Windows パフォーマンス・モニタも同時に実行できます。

Windows パフォーマンス・モニタの詳細については、『ASA SQL ユーザーズ・ガイド』> 「Windows パフォーマンス モニタからのデータベース統計値のモニタリング」を参照してください。

その他の機能強化

- レジストリ設定によるサーバの temp フォルダの変更** Windows CE プラットフォームでは、レジストリを使用して、サーバが使用するテンポラリ・ディレクトリを指定できます。

詳細については、『ASA データベース管理ガイド』> 「Windows CE でのレジストリ設定」を参照してください。

- **新しい iAnywhere JDBC ドライバ** 信頼性とパフォーマンスに優れたこの JDBC ドライバは、ODBC データ・ソースと Command Sequence クライアント／サーバ・プロトコルの利点を利用します。これは、jConnect JDBC ドライバに代わるものです。

iAnywhere JDBC ドライバの詳細については、『ASA プログラミング・ガイド』> 「iAnywhere JDBC ドライバの使用」を参照してください。

JDBC ドライバの選択については、『ASA プログラミング・ガイド』> 「JDBC ドライバの選択」を参照してください。

- **トリガは、トリガが起動する原因となったアクションを識別できる** トリガが UPDATE、INSERT、DELETE のいずれの操作によって起動したかに応じて、異なるアクションを実行できるようになりました。この機能によって、単一トリガ内の異なるイベント間で論理を共有できるとともに、アクションに固有の方法でアクションを実行できます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「トリガ・オペレーション条件」を参照してください。

- **TDS 固有でなくなった RETURN_DATE_TIME_AS_STRING** すべての接続で、RETURN_DATE_TIME_AS_STRING オプションを使用できるようになりました。

このオプションの詳細については、『ASA データベース管理ガイド』> 「RETURN_DATE_TIME_AS_STRING オプション [データベース]」を参照してください。

- **DB 領域への領域追加時に単位の指定が可能** 特定のサイズ (ページ、キロバイト、メガバイト、ギガバイト、またはテラバイトの単位) でデータベース・ファイルを拡張できます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「ALTER DBSPACE 文」を参照してください。

- **sa_make_object システム・プロシージャ** このシステム・プロシージャを SQL スクリプトで使用して、オブジェクトの骨格となるインスタンスが存在することを確認できます。その後で、ALTER 文を実行して実際の定義を提供できます。

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

- **Microsoft SQL Server と互換性のある新しいグローバル変数** 新しいグローバル変数が導入され、Microsoft SQL Server との互換性が提供されています。@@fetch_status グローバル変数は @@sqlstatus グローバル変数と同じですが、最後のフェッチのステータスを別の値で返す点が異なります。

詳細については、『ASA SQL リファレンス・マニュアル』>「グローバル変数」を参照してください。

- **NetWare でサポートされる文字セット変換** NetWare は、文字セット変換をサポートするようになりました。

詳細については、『ASA データベース管理ガイド』>「-ct サーバ・オプション」を参照してください。

- **インストールされている Java クラスのバージョンを情報ユーティリティがレポート** dbinfo ユーティリティと a_db_info 構造体は、データベースにインストールされている Java クラスのバージョンをレポートするようになりました。

詳細については、『ASA データベース管理ガイド』>「情報ユーティリティ」と『ASA プログラミング・ガイド』>「a_db_info 構造体」を参照してください。

- **フェッチ操作の警告の抑制** バージョン 8.0 以降のデータベース・サーバでは、それよりも前のバージョンのソフトウェアに比べて多様なフェッチ警告が返されます。[Adaptive Server Anywhere 9 の ODBC 設定] ダイアログでは、データベース・サーバから返される警告メッセージを抑制し、以前のバージョンのソフトウェアを使用して配備されたアプリケーションに対して、警告メッセージが適切に処理されるようにできます。

詳細については、『SQL Anywhere Studio ヘルプ』>
「[ODBC 設定] ダイアログ : [ODBC] タブ」を参照してください。

- **プライマリ・キー・カラムの更新の制御** 新しい
prevent_article_pkey_update オプションを ON に設定すると、パブリケーションに含まれるテーブルのプライマリ・キー・カラムの更新が禁止されます。このオプションを使用することで、特にレプリケーションと同期の環境におけるデータの整合性が確保されます。

詳細については、『ASA データベース管理ガイド』>
「PREVENT_ARTICLE_PKEY_UPDATE オプション [データベース]」を参照してください。

Mobile Link の新機能

次に、バージョン 8.0.2 で導入したソフトウェアに加えられた変更と追加を示します。

- **.NET のサポート** Mobile Link では、同期スクリプトで Visual Studio .NET プログラミング言語の記述がサポートされるようになりました。

詳細については、『Mobile Link 管理ガイド』> 「.NET での同期スクリプトの作成」、『Mobile Link 管理ガイド』> 「-sl dnet オプション」、『Mobile Link 管理ガイド』>
「ml_add_dnet_connection_script」、および 『Mobile Link 管理ガイド』> 「ml_add_dnet_table_script」を参照してください。

- **起動クラス** 最初の同期の前、Mobile Link サーバの起動時に JVM または CLR を実行する Java コードと .NET コードを記述できるようになりました。

詳細については、『Mobile Link 管理ガイド』> 「ユーザ定義起動クラス」を参照してください。

- **UUID を使用したユニークなプライマリ・キーの管理** リモート・データベースでユニークなプライマリ・キーを管理する新しい方法が、ユニバーサル・ユニーク ID (UUID、別名 GUID) とともに導入されました。

詳細については、『Mobile Link 管理ガイド』> 「UUID を使用したユニークなプライマリ・キーの管理」を参照してください。

- **参照整合性違反の新しい処理方法** ダウンロード時の参照整合性違反の管理に役立つ 2 つの新しいクライアント・ストアド・プロシージャ `sp_hook_dbmlsync_download_ri_conflict` と `sp_hook_dbmlsync_download_log_ri_conflict` が導入されました。

詳細については、『Mobile Link クライアント』> 「`sp_hook_dbmlsync_download_ri_violation`」と『Mobile Link クライアント』> 「`sp_hook_dbmlsync_download_log_ri_violation`」を参照してください。

- **リモート・テーブルのすべてのローを削除する簡単な方法** すべてのプライマリ・キー・カラムが NULL である 1 つのローを `download_delete_cursor` に含めると、リモート・テーブル内のすべてのデータを削除できます。

詳細については、『Mobile Link 管理ガイド』> 「`download_delete_cursor` スクリプトの作成」を参照してください。

パフォーマンスとモニタリングの強化

- **Mobile Link モニタ** グラフィカル・ツールである Mobile Link モニタが導入され、各同期処理にかかった時間を Mobile Link ユーザまたはワーカ・スレッドでソートして表示できるようになりました。

詳細については、『Mobile Link 管理ガイド』> 「Mobile Link モニタ」を参照してください。

- **dbmlsync にアップロードされるロー数の推定が可能** 新しい `dbmlsync` コマンド・ライン・オプション `-urc` が作成されました。アップロードされるロー数を推定することで、同期のパフォーマンスを改善できます。

詳細については、『Mobile Link クライアント』> 「-urc オプション」を参照してください。

- **HTTP/HTTPS 接続の継続を指定可能** **persistent** オプションを使用して、同期のすべての HTTP 要求に同じ接続を使用するように Mobile Link に指定できます。この設定によって、パフォーマンスを向上できます。この設定は、Mobile Link に直接接続しているときにだけ使用してください。プロキシやリダイレクタなどの中間エージェントを介して接続するときには使用しないでください。

詳細については、『ASA SQL リファレンス・マニュアル』> 「CREATE SYNCHRONIZATION USER 文 [Mobile Link]」を参照してください。

- **警告メッセージの新しい制御方法** 3つの新しい dbmlsrv9 コマンド・ライン・オプション -zw、-zwd、-zwe が作成されました。-zw を使用すると、レポートする警告メッセージのレベルを制御できます。-zwd を使用すると、特定の警告コードを無効にできます。-zwe を使用すると、-zw で無効にされている特定の警告コードを有効にできます。

詳細については、『Mobile Link 管理ガイド』> 「-zw オプション」、『Mobile Link 管理ガイド』> 「-zwd オプション」、および『Mobile Link 管理ガイド』> 「-zwe オプション」を参照してください。

- **新しい冗長ロギング・オプション** dbmlsync -v コマンド・ライン・オプションが変更され、拡張されました。現在は、-v を単独で使用すると冗長性が最小になります。冗長性を最大にするには、-v+ を使用します。ログに記録された情報を詳細に調整するために指定できる新しいレベルもいくつかあります。これらのオプションは、拡張オプションとしても使用できます。

詳細については、『Mobile Link クライアント』> 「-v オプション」を参照してください。

接続の強化

- **ping のサポート** リモート・データベースは、Mobile Link 同期サーバに対して ping を実行できるようになりました。

詳細については、『Mobile Link クライアント』> 「-pi オプション」と『Mobile Link クライアント』> 「Ping 同期パラメータ」を参照してください。

- **新しい同期ストリーム** Mobile Link では、HTTPS プロトコルがサポートされるようになりました。この新しいストリームは、RSA 暗号化を使用して HTTP over SSL/TLS を実装します。また、他の HTTPS サーバと互換性があります。

詳細については、『Mobile Link 管理ガイド』> 「-x オプション」と『ASA SQL リファレンス・マニュアル』> 「CREATE SYNCHRONIZATION USER 文 [Mobile Link]」を参照してください。

- **新しい buffer_size オプション** buffer_size オプションを使用して、固定長の HTTP メッセージの最大バッファ・サイズを指定できるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「CREATE SYNCHRONIZATION USER 文 [Mobile Link]」を参照してください。

- **Mobile Link クライアントの自動ダイヤル** Pocket PC 2002 または Windows デスクトップ・コンピュータで稼働する Mobile Link クライアントは、ダイヤルアップ・ネットワーク接続を通じて接続できるようになりました。スケジュールを使用すると、リモート側で自動的に同期を実行できます。新しい同期ストリーム・パラメータは、**network_name**、**network_connect_timeout**、**network_leave_open** です。

詳細については、『ASA SQL リファレンス・マニュアル』> 「CREATE SYNCHRONIZATION USER 文 [Mobile Link]」を参照してください。

新しい Web サーバのサポート

- **サーブレット・リダイレクタ** Mobile Link では、Apache Tomcat など、Java サーブレット API 2.2 をサポートする Web サーバがサポートされるようになりました。

詳細については、『Mobile Link 管理ガイド』> 「リダイレクタによる Web サーバを経由した同期」を参照してください。

- セキュリティの強化**
- **RSA 暗号パッケージ・プログラムのサポート** 既存の楕円曲線暗号化に加え、RSA 暗号化を同期セキュリティに使用できるようになりました。gencert ユーティリティと readcert ユーティリティでは、RSA 証明書と楕円曲線証明書がサポートされます。

詳細については、『Mobile Link 管理ガイド』>「Mobile Link トランSPORT・レイヤ・セキュリティ」を参照してください。
 - **gencert による生成済み証明書要求への署名が可能** 証明書生成ユーティリティ gencert には、生成済み証明書要求に署名できる新しいコマンド・ライン・オプションがあります。

詳細については、『Mobile Link 管理ガイド』>「証明書生成ユーティリティ」を参照してください。

SQL Remote の新機能

SQL Remote バージョン 8.0.2 に、次の新機能が追加されています。

- **統合データベースに送信されるエラー・ログ** リモート・サイトでのエラーのトラブルシューティングを向上するために、統合データベースでログ情報を収集できます。

詳細については、『SQL Remote ユーザーズ・ガイド』>「リモート・サイトでのトラブルシューティング・エラー」を参照してください。

Ultra Light の新機能

Ultra Light 8.0.2 には、次の新機能が導入されています。

- **Ultra Light コンポーネント** Ultra Light データベース・テクノロジーを、使いやすい方法で新しい開発プラットフォームから使用できるようになりました。Ultra Light コンポーネントは、eMbedded Visual Basic、AppForge MobileVB、Java のユーザに Ultra Light テクノロジーを提供します。Java のコンポーネントは、このマニュアルで説明する Ultra Light for Java の代わりに使用で

きます。このコンポーネントは 100% pure Java の実装ではありませんが、パフォーマンスを向上させるためにネイティブ・クラスを使用します。

Ultra Light コンポーネントのマニュアルはオンライン・マニュアルで提供されています。まずは、『Ultra Light データベース・ユーザーズ・ガイド』>「Ultra Light データベース・ユーザーズ・ガイド」を参照してください。

- **Ultra Light データベースのアップグレード** 新しいバージョンのアプリケーションを配備するときに、Ultra Light データベースのスキーマを新しいアプリケーションのスキーマにアップグレードするかどうかを選択できるようになりました。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』>「ULEnableGenericSchema 関数 (旧式)」を参照してください。

9.0.1 では、ULEnableGenericSchema は ULRegisterSchemaUpgradeObserver に置き換わっています。『Ultra Light C/C++ ユーザーズ・ガイド』>「ULRegisterSchemaUpgradeObserver 関数」を参照してください。

- **スレッド対応の Java ランタイム** Ultra Light Java ランタイムはスレッド対応になり、マルチスレッドの Ultra Light アプリケーションを開発できるようになりました。
- **Ultra Light データベース・ファイルの削除** ULDropDatabase 関数を使用して、Ultra Light データベース・ファイルをアプリケーションから削除できます。

詳細については、次の項目を参照してください。

- **Embedded SQL** : 『Ultra Light C/C++ ユーザーズ・ガイド』>「ULDropDatabase 関数」
- **C++ API** : 『Ultra Light C/C++ ユーザーズ・ガイド』>「Drop メソッド」
- **Java** : 『Ultra Light 静的型 Java ユーザーズ・ガイド』>「drop メソッド」

- **ユニバーサル・ユニーク識別子** Ultra Light データベースで、Adaptive Server Anywhere の UNIQUEIDENTIFIER データ型を使用できるようになりました。このデータ型は、ユニバーサル・ユニーク識別子 (UUID または GUID) の格納に使用される BINARY(16) です。NEWID 関数をデフォルト値として使用する UNIQUEIDENTIFIER カラムは、GLOBAL AUTOINCREMENT の代わりに使用でき、インストールされている Mobile Link 全体を通じてプライマリ・キーがユニークであることを保証します。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「NEWID デフォルト」を参照してください。

- **同期の新しいセキュリティ・オプション** このリリースでは、2 つの新しいセキュリティ機能のある同期プロトコルが導入されました。HTTPS は、トランスポート・レイヤ・セキュリティ・プロトコルで実装した HTTP です。RSA は、HTTP または TCP/IP ネットワーク上で使用するトランスポート・レイヤ・セキュリティ暗号化の 1 つの形式です。

これらのセキュリティ・オプションでは、Certicom テクノロジーが使用されます。Certicom テクノロジーを使用するには、別途ライセンスの SQL Anywhere Studio セキュリティ・オプションを手する必要があります。このセキュリティ・オプションは、輸出規制対象品目です。このオプションの詳細については、『SQL Anywhere Studio の紹介』> 「SQL Anywhere Studio へようこそ」を参照してください。

RSA 暗号化の詳細については、『Mobile Link クライアント』> 「Security 同期パラメータ」を参照してください。Java から RSA 暗号化を使用する方法については、『Ultra Light 静的型 Java ユーザーズ・ガイド』> 「同期パラメータの初期化」を参照してください。

HTTPS 同期の詳細については、『Mobile Link クライアント』> 「Stream Type 同期パラメータ」と『Mobile Link クライアント』> 「HTTPS プロトコル・オプション」を参照してください。

- **最終ダウンロード時間のリセット** たとえばアプリケーションをクリーンな状態に設定するために、以前にダウンロードしたデータを再同期するには、最終ダウンロード・タイムスタンプをリセットします。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「ULResetLastDownloadTime 関数」と『Ultra Light C/C++ ユーザーズ・ガイド』> 「ResetLastDownloadTime メソッド」を参照してください。

- **前回の同期のトラブルシューティング** 前回の同期の成功または失敗に関する情報を取得する関数を使用できるようになりました。この機能は、HotSync を使用する Palm OS アプリケーションで、同期がアプリケーションの外部で実行される場合に特に役立ちます。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「GetSynchResult メソッド」と『Ultra Light C/C++ ユーザーズ・ガイド』> 「ULGetSynchResult 関数」を参照してください。この機能は Ultra Light Java アプリケーションにはまだ使用できません。

- **サイズの小さいファイルをたくさん生成** -x オプションを使用すると、Ultra Light ジェネレータは、C/C++ プロジェクトについて、ファイルサイズが小さなファイルをたくさん書き出します。このオプションは、生成されたコードが大きすぎてコンパイラが単一のファイルで処理できない場合に役立ちます。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「Ultra Light ジェネレータ」を参照してください。

- **強化された同期 observer** 同期 observer 関数が拡張されました。より多くのステータスとフィールドがインタフェースに追加され、応答性に優れ、情報の多い同期ダイアログを設計できます。

バージョン 8.0.2 での動作の変更

この項では、SQL Anywhere Studio バージョン 8.0.2 のコンポーネントに導入された動作の変更について説明します。

Adaptive Server Anywhere の動作の変更

ここでは、これまでのバージョンとは異なる動作をリストにして説明します。

- **Windows CE 2.11 のサポート終了** Windows CE 2.11 プラットフォームはサポートされなくなりました。
- **SH3 チップと SH4 チップのサポート終了** SH3 チップと SH4 チップを使用する Windows CE デバイスはサポートされなくなりました。

サポートされているプラットフォームのリストについては、『SQL Anywhere Studio の紹介』>「オペレーティング・システムのバージョン」と『SQL Anywhere Studio の紹介』>「Windows オペレーティング・システムと NetWare オペレーティング・システム」を参照してください。

- **OPTIMIZATION_GOAL 設定** OPTIMIZATION_GOAL オプションのデフォルト設定は、**first-row** ではなく **all-rows** です。これは、一部のクエリで選択されている実行プランに影響するため、パフォーマンス特性が変わります。

詳細については、『ASA データベース管理ガイド』>「OPTIMIZATION_GOAL オプション [データベース]」を参照してください。

- **Windows オペレーティング・システムでの xp_cmdshell による コマンド・ウィンドウの表示** xp_cmdshell が新しいウィンドウを起動するかどうかを制御できるようになりました。この動作変更は、バージョン 8.0.2 以降で作成またはアップグレードされたデータベースに適用されます。それより古いデータベースの場合は、コマンド・ウィンドウを表示しない以前の動作が維持

されます。この新しい動作は、Adaptive Server Enterprise や Microsoft SQL Server などの他のデータベースと互換性があります。

`xp_cmdshell` の呼び出しで 2 番目のパラメータを指定すると、コマンド・ウィンドウを非表示にできます。

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

- データベース・サーバで使用されている言語に関係なく、完全なスペルの英語の曜日名が認識される** イベントを作成するときに、データベース・サーバが使用している言語 (ドイツ語、中国語など) に関係なく、完全なスペルの英語の曜日名がデータベース・サーバによって認識されます。したがって、再ロード・スクリプトのイベント定義が、異なる言語で実行されているサーバで認識されます。

省略形の英語の曜日名 (Mon、Tue など) を使用するイベントは、英語以外の言語で実行されているサーバでは認識されません。

詳細については、『ASA SQL リファレンス・マニュアル』>「CREATE EVENT 文」を参照してください。

- オプション設定の検証** 最小値と最大値を持つ整数オプションが検証されるようになりました。オプションを無効な値に設定すると、『ASA エラー・メッセージ』>「不正なオプション '%1' を設定しています。」というエラーが発生します。

無効なオプション設定を含むデータベースをアンロードして再ロードした場合は、最も近い有効値に設定されます。

影響するオプションを次に示します。角カッコは上限値と下限値を示します。

オプション	範囲
ISOLATION_LEVEL	[0, 3]
PRECISION	[0, 127]
SCALE	[0, 127]

オプション	範囲
NEAREST_CENTURY	[0, 100]
MAX_HASH_SIZE	[2, 64]
MAX_WORK_TABLE_HASH_SIZE	[2, 64]
FIRST_DAY_OF_WEEK	[1, 7]
DEFAULT_TIMESTAMP_INCREMENT	[1, 60 000 000]

- **名前変更されたジョイン** グラフィカル・プランとドキュメントの両方で2つのジョインの名前が変更されました。ネスト・ループ・ジョイン NOT EXISTS (JNE) は、ネスト・ループ非セミジョイン (JNLA) と呼ばれるようになり、ネスト・ループ・ジョイン EXISTS (JE) はネスト・ループ・セミジョイン (JNLS) と呼ばれるようになりました。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「ジョイン・アルゴリズム」を参照してください。

廃止されサポートされなくなった機能

このリストには、サポートを終了した機能の中で既存のアプリケーションに影響のあるものがまとめられています。

- **Windows で使用されなくなった -d サーバ・オプション** -d オプションを NetWare で使用すると、DFS (Direct File System) I/O ではなく POSIX I/O が強制的に使用されます。Windows では、このオプションはコマンド・ラインで使用できますが、無視されます。

詳細については、『ASA データベース管理ガイド』> 「-d サーバ・オプション」を参照してください。

Mobile Link の動作の変更

ここでは、これまでのバージョンとは異なる動作をリストにして説明します。

- **シリアル通信プロトコルのサポート終了** シリアル・プロトコルはサポートされなくなりました。代わりに、HTTP、HTTPS、または TCP/IP を使用できます。
- **Certicom からの証明書の発行終了** Certicom からトランスポート・レイヤ・セキュリティ証明書を取得できなくなりました。ただし、Certicom reqtool ユーティリティを使用して証明書要求を生成し、VeriSign や Entrust Technologies などの他のさまざまなソースから証明書を購入することはできます。

詳細については、<http://www.verisign.com/> または http://www.entrust.com/certificate_services/index.htm を参照してください。

- **dbmlsrv の使用されなくなったオプション -vw** 警告メッセージを抑制するために使用していた -vw dbmlsrv コマンド・ライン・オプションは使用されなくなりました。代わりに、-zw または -zwd を使用できます。

詳細については、『Mobile Link 管理ガイド』> 「-zw オプション」と『Mobile Link 管理ガイド』> 「-zwd オプション」を参照してください。

- **dbmlsync オプション -v の動作変更** -v dbmlsync コマンド・ライン・オプションが変更され、拡張されました。現在は、-v を単独で使用する冗長性が最小になります。

詳細については、『Mobile Link クライアント』> 「-v オプション」を参照してください。

- **同期サーバで使用されている言語に関係なく完全な長さの英語の曜日名が認識される** Mobile Link のユーザ、パブリケーション、サブスクリプションのスケジュールを作成するとき、または dbmlsync コマンド・ラインでスケジュール情報を指定する場合、英語以外の言語で実行している同期サーバでスケジュールを認識するには、完全な長さの形式の英語の曜日名 (Monday など) を使用してください。

省略形の英語の曜日名 (Mon など) を使用するスケジュールは、英語以外の言語で実行されている同期サーバでは認識されません。

詳細については、『ASA SQL リファレンス・マニュアル』>「CREATE SYNCHRONIZATION USER 文 [Mobile Link]」を参照してください。

- **dbmlsync での長いデータのサポート強化** DBMLSync は、アップロード・ストリームを構築するときに、はるかに効率的な方法で BLOB を処理するようになりました。BLOB は、断片化されてメモリに読み込まれるため、長い BLOB の処理能力が使用可能メモリによって制限されなくなりました。一度に複数のパブリケーションの同期を行う場合は、BLOB データが一度に格納され、アップロード・ストリーム間で共有されます。出力ログは、BLOB のサイズとその最初の 32 バイトを出力するようになりました。
- **HTTP オプション use_cookies の削除** use_cookies オプションは削除されました。このオプションを設定した場合は無視されます。Mobile Link では、cookie が必要な状況が自動的に検出されます。

Ultra Light の動作の変更

ここでは、これまでのバージョンとは異なる動作をリストにして説明します。

- **Windows CE 2.11 のサポート終了** Windows CE 2.11 プラットフォームはサポートされなくなりました。
- **SH3 チップと SH4 チップのサポート終了** SH3 チップと SH4 チップを使用する Windows CE デバイスはサポートされなくなりました。

サポートされているプラットフォームのリストについては、『SQL Anywhere Studio の紹介』>「オペレーティング・システムのバージョン」と『SQL Anywhere Studio の紹介』>「Windows オペレーティング・システムと NetWare オペレーティング・システム」を参照してください。

- **シリアル通信プロトコルのサポート終了** シリアル・プロトコルはサポートされなくなりました。シリアル同期は、主に Palm Computing Platform 上のクライアントから使用されていました。これらのクライアントでは、代わりに HotSync 同期を使用できます。
- **VxWorks でのトランスポート・レイヤ・セキュリティなし** 同期にトランスポート・レイヤ・セキュリティを提供する Certicom ライブラリは、VxWorks オペレーティング・システムでサポートされなくなりました。
- **VxWorks 5.5 はサポートされない** サポートされている VxWorks オペレーティング・システムのバージョンは VxWorks 5.3 と 5.4 です。

バージョン 9 における VxWorks の非サポート

バージョン 9 では VxWorks プラットフォームはいっさいサポートされません。

- **Certicom ライブラリには JDK 1.2 が必要** このリリースでは Certicom セキュリティ・ライブラリが更新されました。Java アプリケーション用の新しいライブラリには、JDK 1.1.4 ではなく JDK 1.2 が必要です。

第 6 章

バージョン 8.0.1 の新機能

この章の内容

この章では、Adaptive Server Anywhere バージョン 8.0.1 で導入された新機能と変更された動作の概要について説明します。

バージョン 8.0.1 の新機能

この項では、SQL Anywhere Studio バージョン 8.0.1 のコンポーネントに導入された新機能について説明します。

Adaptive Server Anywhere の新機能

この項では、Adaptive Server Anywhere バージョン 8.0.1 の新機能について説明します。新機能について主要なものからそうでないものまですべてを示し、各機能の詳細についての参照先も記述しています。

- **テーブル・ページに予約する領域の指定** テーブル・ページで予約領域にする空き領域の割合を指定すると、テーブルの断片化を減らすことができます。

詳細については、『ASA SQL ユーザーズ・ガイド』>「テーブルの断片化」と『ASA SQL リファレンス・マニュアル』>「ALTER TABLE 文」を参照してください。

このリリースより前に作成されたデータベースに対して割り付ける領域の割合を指定するには、データベースをアンロードして再ロードすることによってそのデータベースのファイル・フォーマットをアップグレードする必要があります。

- **新しいシステム・テーブル** 2つの新しいシステム・テーブル SYSATTRIBUTE と SYSATTRIBUTENAME が追加されました。

詳細については、『ASA SQL リファレンス・マニュアル』>「SYSATTRIBUTE システム・テーブル」と『ASA SQL リファレンス・マニュアル』>「SYSATTRIBUTENAME システム・テーブル」を参照してください。

- **sa_disk_free_space システム・プロシージャ** このプロシージャを使用すると、DB 領域、テンポラリ・ファイル、トランザクション・ログ、トランザクション・ログ・ミラーで使用可能な領域を決定できます。

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

- **sa_flush_statistics システム・プロシージャ** データベース管理者は、このプロシージャを使用することで、データベース・サーバのキャッシュのみに存在するコスト・モデル統計情報を確実にフラッシュできます。

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

- **サーバのメッセージ・ウィンドウの内容を取得する新しい方法** サーバのウィンドウから情報を返す新しいシステム・プロシージャと新しい3つのプロパティが用意されました。

詳細については、『ASA SQL リファレンス・マニュアル』>「sa_get_server_messages システム・プロシージャ」と、『ASA データベース管理ガイド』>「サーバ・レベルのプロパティ」の MessageText、MessageTime、MessageWindowSize を参照してください。

- **ANSI 以外の文と等価な ANSI 文の確定** REWRITE 関数は、新しい引数 ANSI を受け付けるようになりました。これにより、REWRITE 関数は SELECT 文、UPDATE 文、または DELETE 文に対して等価な ANSI 文を返します。

詳細については、『ASA SQL リファレンス・マニュアル』>「REWRITE 関数 [その他]」を参照してください。

- **UPDATE 文が変数への代入を許可** UPDATE 文の SET 句は、テーブルの更新だけでなく、変数への値の代入にも使用できるようになりました。この機能は、Adaptive Server Enterprise と互換性があります。

詳細については、『ASA SQL リファレンス・マニュアル』>「UPDATE 文」を参照してください。

- **オートインクリメントの代替手段** GET_IDENTITY 関数は、オートインクリメント・カラムへの ID 番号割り付けの代替手段として提供されます。

詳細については、『ASA SQL リファレンス・マニュアル』>「GET_IDENTITY 関数 [その他]」を参照してください。

- **識別子を角カッコで区切る** 識別子は、角カッコで区切ることができます。角カッコは、QUOTED_IDENTIFIER オプションの設定に関わらず、どのような場合でも使用できます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「識別子」を参照してください。

- **FROM 句での独立性レベルの設定** WITH table-hint 引数を使用すると、特定の SELECT 文、UPDATE 文、または DELETE 文に対してテーブルまたはビューのロック方法を指定できます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「FROM 句」を参照してください。

- **[データベース移行] ウィザード** [データベース移行] ウィザードを使用すると、Sybase Central から Adaptive Server Anywhere データベースにリモート・テーブルをマイグレートできます。

ターゲット・データベースがバージョン 8.0.0 またはそれよりも前の場合、外部キーはマイグレートできません。外部キーをマイグレートするには、ターゲット・データベースをアンロードして再ロードすることにより、データベースのファイル・フォーマットをアップグレードする必要があります。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「Adaptive Server Anywhere へのデータベースの移行」を参照してください。

- **Sybase Central からのバージョン 5.x または 6.x のデータベースのアンロード** Sybase Central では、バージョン 5.x または 6.x のデータベースに接続し、[データベース・アンロード] ウィザードを使用してデータベース・ファイル・フォーマットをアップグレードできるようになりました。これを行うには、バージョン 8.0.0 以降のサーバ上でデータベースを実行する必要があります。

詳細については、「[Adaptive Server Anywhere のアップグレード](#)」[306 ページ](#)を参照してください。

- **[データベース・アップグレード]ウィザードでのデータベースのバックアップとシャットダウン** Sybase Central の [データベース・アップグレード] ウィザードを使用して、メインのデータベース・ファイル、トランザクション・ログ、DB 領域など、すべてのデータベース・ファイルをバックアップできるようになりました。このウィザードでは、アップグレードの完了時にデータベースをシャットダウンすることもできます。

詳細については、「[データベースのアップグレード](#)」307 ページを参照してください。

- **sa_migrate の強化** sa_migrate プロシージャにはオプションの引数 *migrate_fkeys* が追加されています。この引数でリモート・データベースからテーブルをマイグレートするときに、外部キー・マッピングをマイグレートするかどうかを指定できます。これまでのリリースでは、sa_migrate プロシージャを使用すると、外部キー・マッピングは常にマイグレートされていました。

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

このリリースより前に作成されたデータベースでこの機能を使用するには、データベースをアンロードして再ロードすることによってそのデータベースのファイル・フォーマットをアップグレードする必要があります。

- **新しい SORT_COLLATION データベース・オプション**
SORT_COLLATION データベース・オプションを使用すると、ORDER BY 式に対して SORTKEY 関数が暗黙的に使用されます。このオプションの値を、有効な照合名または照合 ID に設定すると、ORDER BY 句の文字列式は、SORTKEY 関数が呼び出されたものとして扱われます。

詳細については、『ASA データベース管理ガイド』>「SORT_COLLATION オプション [データベース]」を参照してください。

- **IP アドレス／ポートを使用してサーバに接続**

VerifyServerName=NO プロトコル・オプションを使用すると、IP アドレスかポートさえ分かっているならば、サーバ名の検証を行わずに Adaptive Server Anywhere クライアントから Adaptive Server Anywhere サーバに接続できます。VerifyServerName パラメータは、DoBroadcast=NONE を指定した場合にのみ使用します。

詳細については、『ASA データベース管理ガイド』>

「VerifyServerName プロトコル・オプション [VERIFY]」を参照してください。

- **新しい LocalOnly プロトコル・オプションによるブロードキャストの管理** LocalOnly プロトコル・オプションを使用すると、ローカル・マシン上のサーバ (存在する場合) のみに接続できます。LocalOnly=YES を設定すると、サーバから他のマシンへのブロードキャスト応答が無視された場合を除き、通常のブロードキャスト・メカニズムが使用されます。

詳細については、『ASA データベース管理ガイド』> 「LocalOnly プロトコル・オプション [LOCAL]」を参照してください。

- **カーソルの固定に使用するキャッシュの割合の指定**

PINNED_CURSOR_PERCENT_OF_CACHE オプションを使用すると、カーソルを固定するために使用するキャッシュの割合を指定できます。この制限値を低く設定するとメモリが少ない環境のパフォーマンスが向上します。

詳細については、『ASA データベース管理ガイド』>

「PINNED_CURSOR_PERCENT_OF_CACHE オプション [データベース]」を参照してください。

- **データベース・ファイルとログ・ファイルの断片化をモニタ**

DBFileFragments と LogFileFragments データベース・プロパティを使用すると、ファイルの断片化をモニタできます。通常、トランザクション・ログ・ファイルの断片化は大きな問題にはなりませんが、データベース・ファイルの断片化はパフォーマンス低下の原因となることがあるので、その場合にはディスク断片化解除ユーティリティを使用してください。

詳細については、『ASA データベース管理ガイド』> 「データベース・レベルのプロパティ」を参照してください。

- **新しい接続プロパティ** 2つの接続プロパティが新しく追加されました。LivenessTimeout は接続の活性タイムアウト、IdleTimeout は接続のアイドル・タイムアウトをそれぞれ返します。

詳細については、『ASA データベース管理ガイド』> 「接続レベルのプロパティ」を参照してください。

- **新しいサーバ・プロパティ** 新しい IdleTimeout サーバ・プロパティは、デフォルトのアイドル・タイムアウト値を返します。

詳細については、『ASA データベース管理ガイド』> 「サーバ・レベルのプロパティ」を参照してください。

- **非決定的関数** 基本データを修正する関数、またはクエリの実行中に変更される可能性のある基本データに依存する関数を NOT DETERMINISTIC と宣言できます。このように宣言された関数は、クエリ実行中に呼び出されるたびに再評価されます。このように宣言されない関数では、パフォーマンス向上のために関数の値がキャッシュされ、再利用されます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「CREATE FUNCTION 文」を参照してください。

- **バックアップで、すべてのトランザクションが完了するのを待つ** デフォルトで、BACKUP 文では、開いているトランザクションが完了するのを待たずに、トランザクション・ログ名の変更やトランザクション・ログのトランケートが行われます。WAIT AFTER END 句を指定すると、バックアップに含まれるすべてのトランザクションを確実に完了できるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「BACKUP 文」を参照してください。

Mobile Link の新機能

次に、バージョン 8.0.1 で導入したソフトウェアに加えられた変更と追加を示します。

- **完全なエラー・コンテキストの報告** 同期中にエラーが発生した場合、Mobile Link 同期サーバは、出力ファイルに完全なエラー・コンテキストを示すようになりました。

詳細については、『Mobile Link 管理ガイド』> 「-o オプション」を参照してください。

- **ユーザ ID マッピング** Mobile Link では、データベース・ユーザ ID の検索や、Mobile Link ユーザ名のユーザ ID へのマッピングが簡単になりました。

詳細については、『Mobile Link 管理ガイド』> 「modify_user 接続イベント」を参照してください。

- **クライアント・オプションとしてアドレスとタイプを設定** Mobile Link クライアントでは、コマンド・ラインで通信タイプとアドレスを指定して Mobile Link 同期サーバに接続できるようになりました。

詳細については、『Mobile Link クライアント』> 「dbmlsync 拡張オプション」を参照してください。

- **Mobile Link が発行する ODBC 文のログ** Mobile Link に対しては、Mobile Link が発行するすべての ODBC 文について ODBC 出力ファイルにログを取るように指定できます。

詳細については、『Mobile Link 管理ガイド』> 「-t オプション」を参照してください。

- **ダウンロード・タイムスタンプの修正** 2つの新しいイベントで、最終ダウンロード・タイムスタンプまたは次の最終ダウンロード・タイムスタンプを修正できます。

詳細については、『Mobile Link 管理ガイド』> 「modify_last_download_timestamp 接続イベント」と『Mobile Link 管理ガイド』> 「modify_next_last_download_timestamp 接続イベント」を参照してください。

- **タイムスタンプ競合への自動許容** このオプションを使用すると、統合データベースとリモート・データベース間でタイムスタンプの競合が発生した場合、競合を検出するために、最小精度よりも高い精度のタイムスタンプ値を使用できます。

詳細については、『Mobile Link 管理ガイド』> 「-zp オプション」を参照してください。

SQL Remote の新機能

SQL Remote バージョン 8.0.1 に、次の新機能が追加されています。

- **SMTP ユーザ認証** SMTP/POP メッセージ・システムを使用しているときに、SMTP サーバでの別個のユーザ認証のためのパラメータが用意されています。

詳細については、『SQL Remote ユーザーズ・ガイド』> 「SMTP メッセージ・システム」を参照してください。

Ultra Light の新機能

Ultra Light 8.0.1 には、次の新機能が導入されています。

- **CodeWarrior 8 のサポート** このリリースでは、CodeWarrior バージョン 8 をサポートします。
- **マルチスレッド・アプリケーションのサポート** Ultra Light アプリケーションは、マルチスレッド・アプリケーションをサポートするプラットフォーム上でマルチスレッド化できるようになりました。
- **Pocket PC 2002 のサポート** Pocket PC 2002 が、サポートされるプラットフォームのリストに追加されています。
- **JDBC ResultSet メソッドの追加** ResultSet.findColumn と ResultSet.getType メソッドがサポートされるようになりました。

詳細については、『Ultra Light 静的型 Java ユーザーズ・ガイド』> 「Ultra Light での JDBC 機能」を参照してください。

- **Ultra Light Java からの情報へのアクセス**

JdbcConnection.getLastIdentity メソッド、**getLastDownloadTime** メソッド、**JdbcDatabase.countUploadRows** メソッドを使用すると、有用な情報にアクセスできます。これらの機能は、以前は C/C++ アプリケーションのみで使用可能でした。

詳細については、『Ultra Light 静的型 Java ユーザーズ・ガイド』> 「JdbcConnection クラス」を参照してください。

- **Ultra Light Java でのユーザ認証** Java バージョンの Ultra Light で、ユーザ認証がサポートされるようになりました。

詳細については、『Ultra Light 静的型 Java ユーザーズ・ガイド』> 「アプリケーションへのユーザ認証の追加」、『Ultra Light 静的型 Java ユーザーズ・ガイド』> 「JdbcSupport クラス」、および『Ultra Light 静的型 Java ユーザーズ・ガイド』> 「JdbcDatabase クラス」を参照してください。

- **HotSync 同期の進捗状況の表示** デスクトップ・コンピュータの [HotSync Progress] ダイアログには、Ultra Light アプリケーションとの同期の進捗状況が表示されます。

- **HotSync の設定** Palm Desktop から HotSync conduit を設定できます。

詳細については、『Mobile Link クライアント』> 「Mobile Link HotSync コンジットの設定」を参照してください。

- **Ultra Light アプリケーションからのスクリプトの自動生成**

Ultra Light アプリケーションでは、同期スクリプトを自動的に生成できるように、Mobile Link 同期サーバへのカラム名を用意するようになりました。

- **C++ API からカラムの SQL データ型を取得** **GetColumnSQLType** メソッドは、カラムのデータ型を返します。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「GetColumnSQLType メソッド」を参照してください。

- **同期中のチェックポイント (オプション)** 大量の更新をダウンロードする同期は、Ultra Light データベースのサイズが大幅に増加する原因となる場合があります。この増加は、同期中にチェックポイントを実行することで制限できます。新しく **checkpoint_store** 同期パラメータを指定して、チェックポイントを制御します。デフォルトでは、チェックポイントは実行されません。

詳細については、『Mobile Link クライアント』> 「Checkpoint Store 同期パラメータ」を参照してください。

バージョン 8.0.1 での動作の変更

この項では、SQL Anywhere Studio バージョン 8.0.1 のコンポーネントに導入された動作の変更について説明します。

Adaptive Server Anywhere の動作の変更

ここでは、これまでのバージョンとは異なる動作をリストにして説明します。

- **名前を変更するトランザクション・ログ・ファイルの新しい命名規則** バックアップ中に変更されるトランザクション・ログ・ファイルの最後の 2 桁の数字は、2 つの文字に変更されました。たとえば、最初のバックアップが 2000 年 12 月 10 日である場合、ログ・ファイル名は `00121001.log` でなく、`001210AA.log` となります。最初の 2 桁は年、次の 2 桁は月、その次の 2 桁は日付を示し、最後の 2 文字によって、同じ日に実行された複数のバックアップを識別します。この方法を使用すると、1 日に可能なバックアップ数が 100 から 676 に増えます。
- **LOAD TABLE による計算カラムの再計算** LOAD TABLE は、計算カラムを検出し、テーブルに挿入されている各ローに対して計算カラムを評価するようになりました。
- **DBCONSOLE で接続の再接続ができるようになった** 以前の DBConsole セッションでは 1 つの接続のみが可能でした。接続は、アプリケーションを終了することなく、切断と再接続ができるようになりました。

廃止されサポートされなくなった機能

このリストには、サポートを終了した機能の中で既存のアプリケーションに影響のあるものがまとめられています。

- **使用されなくなった DEBUG 接続パラメータ** DEBUG 接続パラメータが使用されなくなりました。LOG パラメータを使用すると、デバッグ情報を含むログ・ファイルを作成できます。バージョン 8.0.1 からは、`DEBUG=YES;LOG=filename` が行ってきたことを `LOG=filename` が行います。

詳細については、『ASA データベース管理ガイド』> 「接続パラメータ」を参照してください。

- **使用されなくなった AGENT 接続パラメータ** AGENT 接続パラメータが使用されなくなりました。CommLinks パラメータを適切なプロトコル・オプションと共に使用すると、AGENT パラメータと同じ動作になります。

詳細については、『ASA データベース管理ガイド』> 「接続パラメータ」を参照してください。

- **Port 接続プロパティが削除** port 接続パラメータが削除されました。
- **Adaptive Server Anywhere 変換ドライバが削除** 変換ドライバは使用しないことをおすすめします。文字セット変換は、サーバが自動的に処理します。
- **SharedMemory を最初に試行** LINKS= 接続パラメータで指定したポートは、指定した順序で接続が試みられていました。今後は、sharedmemory (shmem) ポートが指定されている場合は、最初にこのポートへの接続が試みられ、その後は指定されている順序で接続が試みられるようになりました。
- **グローバル・オートインクリメント** デフォルト値が 0 から 2147483647 に変更されました。GLOBAL_DATABASE_ID を 0 に設定し、1 から始まる値を生成できるようになりました。

Mobile Link の動作の変更

- **タイムスタンプ不一致の通知** 統合データベースとリモート・データベースのタイムスタンプが異なる場合、Mobile Link 同期サーバによって、同期ごとの警告がログに記録されます。
- **グローバル・オートインクリメント** デフォルト値が 0 から 2147483647 に変更されました。GLOBAL_DATABASE_ID を 0 に設定し、1 から始まる値を生成できるようになりました。

GLOBAL_DATABASE_ID が設定されていないか、デフォルト値に設定されている場合、グローバル・オートインクリメント値を生成しようとすると、結果は NULL になります。この値を NULL 入力不可のプライマリ・キー・カラムに挿入しようとすると、通常はエラーが発生します。これは、GLOBAL_DATABASE_ID オプションが設定されていないことを示します。

GLOBAL_DATABASE_ID に 0 を設定できないようにすると、1 から始まる値ではなく、カラムに指定された分割サイズから始まる値が生成されます。

詳細については、『ASA データベース管理ガイド』> 「GLOBAL_DATABASE_ID オプション [データベース]」を参照してください。

- **dbmlstop によるソフト・シャットダウン** デフォルト (-w、-f、-h、-t のいずれも設定されていない場合) では、dbmlstop によってソフト・シャットダウンが行われます。これは、現在の同期が完了すると、新しい接続が受け入れられずに停止することを意味します。

詳細については、『Mobile Link 管理ガイド』> 「Mobile Link 停止ユーティリティ」を参照してください。

Ultra Light の動作の変更

- **Palm データベースのバックアップ** 以前のリリースでは、ULUtil アプリケーションを使用してデータベースをバックアップすると、その後 HotSync 操作を実行するたびにデータベースがバックアップされました。

ほとんどの Ultra Light データは、同期によって効果的にバックアップされます。明示的なバックアップの最も一般的な用途は、配備用の初期データベースの作成であるため、ほとんどの場合、HotSync での継続的なバックアップは不要な動作です。現在のバージョンでは、Ultra Light アプリケーションが起動するたびに、その後の HotSync 操作でのバックアップが無効になります。

HotSync が実行されるたびにデータベースのバックアップを明示的に要求する場合は、`UL_STORE_PARMS` マクロで **`palm_all_backup`** パラメータを設定します。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「`UL_STORE_PARMS` マクロ」を参照してください。

廃止されサポートされなくなった機能

Ultra Light による、ScoutSync テクノロジを使用した Palm Computing Platform での同期サポートは終了しました。バージョン 8.0.x では、ScoutSync バージョン 3.6 までは引き続きサポートされますが、SQL Anywhere Studio の今後のメジャー・リリースからは ScoutSync はサポートされません。

第7章

バージョン 8.0.0 の新機能

この章の内容

この章では、Adaptive Server Anywhere バージョン 8.0.0 で導入された新機能と変更された動作の概要について説明します。

バージョン 8 の新機能

この項では、SQL Anywhere Studio バージョン 8 に導入された新機能について説明します。

Adaptive Server Anywhere の新機能

この項では、Adaptive Server Anywhere バージョン 8.0 の新機能について説明します。新機能について主要なものからそうでないものまですべてを示し、各機能の詳細についてのマニュアル内の参照先も記述しています。

このマニュアルの製本版を使用していて、SQL Anywhere Studio のマニュアル・セット全巻をお持ちでない場合、各機能の詳細についてはオンライン・マニュアルを参照してください。

一部の新機能では、データベースをバージョン 8 にアップグレードするか、データベースをアンロードして再ロードすることでデータベースのファイル・フォーマットをアップグレードする必要があります。特定の機能にアクセスするためにデータベースまたはファイル・フォーマットのアップグレードが必要な場合は、稼動条件が下記に記載されています。

これらのタスクを実行する方法については、「[Adaptive Server Anywhere のアップグレード](#)」306 ページを参照してください。

Adaptive Server Anywhere の新機能は、次の見出しの下にグループ分けされています。

- 「クエリ処理とデータベース・パフォーマンス」181 ページ
- 「セキュリティ」184 ページ
- 「SQL の機能」185 ページ
- 「開発ツールと管理ツール」187 ページ
- 「アプリケーション開発」190 ページ
- 「管理機能とトラブルシューティング」191 ページ

- [「クライアント／サーバ接続」196 ページ](#)
- [「データベース内の Java」198 ページ](#)
- [「マニュアル」199 ページ](#)
- [「その他」200 ページ](#)

クエリ処理とデータベース・パフォーマンス

- **クエリ処理の向上** このバージョンでは、クエリ実行エンジンとオブティマイザが強化されています。これによって、パフォーマンス、特に複雑なクエリのパフォーマンスが大幅に向上しています。Adaptive Server Anywhere のクエリ処理は、次の点が強化されています。
 - ジョインの内部処理の高度化。
 - 代替アクセス・プランへのアクセスに使用されるオブティマイザのコスト・モデルの向上。
 - 実行モデルの向上。

変更の大半は、内部の変更です。詳細な説明については、『ASA SQL ユーザーズ・ガイド』>「クエリの最適化と実行」を参照してください。

これらの変更により、これまでのように結果の実体化が必ずしも非効率的とは言えなくなります。テンポラリ・ワーク・テーブルを使用することによって、クエリを非常に効率的に実行できる場合もあります。詳細については、『ASA SQL ユーザーズ・ガイド』>「クエリ処理中のワーク・テーブルの使用」を参照してください。

オブティマイザは、コストに基づいてインデックスを選択するようになり、これまでのリリースのように述語の選択性だけに頼ることはなくなりました。

強化されたクエリ処理のほとんどは、データベースのアップグレードを必要としません。このリリースより前に作成されたデータベースで新しいコスト・モデルを使用するには、データ

ベースをアンロードして再ロードすることによってそのデータベースのファイル・フォーマットをアップグレードする必要があります。

- **新しいインデックス・タイプ** 複数のカラム・インデックスと幅の広いカラムを含むインデックスについて、パフォーマンスを向上させる新しいタイプのインデックスが追加されました。新しいインデックスは、圧縮 B ツリー・インデックスといいます。

Adaptive Server Anywhere は、インデックスの幅 (インデックス内の全カラムの幅の合計) に基づいて、適切なタイプのインデックスを自動的に作成します。圧縮 B ツリー・インデックスは、インデックスの幅が 10 バイト以上で、ページ・サイズの 1/8 未満から最大 256 バイトまでのときに作成されます。それ以外の場合は、ハッシュ B ツリー・インデックスが作成されます。

CREATE INDEX 文の WITH HASH SIZE 句は、廃止されました。

インデックスの詳細については、『ASA SQL ユーザーズ・ガイド』> 「インデックスのタイプ」を参照してください。

このリリースより前に作成されたデータベースで新しいタイプのインデックスを使用するには、データベースをアンロードして再ロードすることによってそのデータベースのファイル・フォーマットをアップグレードする必要があります。

新しい制約が追加されています。外部キー・インデックスのサイズとタイプは、対応するプライマリ・キー・インデックスと同じである必要があります。

dbunload は、デフォルト (WITH HASH SIZE 10) で指定された場合は、ハッシュ・サイズ指定を省略するようになりました。

- **新しいデータベース・オプション OPTIMIZATION_GOAL** クエリ処理の最適化の対象を、最初のローを迅速に返すこと、または完全な結果セットを返すコストを最小限に抑えることのどちらかに指定します。デフォルトは、最初のローの最適化です。

詳細については、『ASA データベース管理ガイド』> 「OPTIMIZATION_GOAL オプション [データベース]」を参照してください。

- **テーブル・スキャンのパフォーマンスの強化** Adaptive Server Anywhere 8.0 で作成された 2 K、4 K、または 8 K ページのデータベースでは、テーブルの逐次スキャンを必要とするクエリのパフォーマンスが強化されました。大きなテーブルに対しては、ページ・マップとも呼ばれるビットマップが Adaptive Server Anywhere によって作成されます。特定のテーブルのデータを含むすべてのページがビットマップにリストされます。この機能によって、1 回の I/O 操作だけで大きなテーブルを検索できます。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「テーブルとページのサイズ」を参照してください。

このリリースより前に作成されたデータベースでこの拡張機能を利用するには、データベースをアンロードして再ロードすることによってそのデータベースのファイル・フォーマットをアップグレードする必要があります。

- **チェックポイント・ログの記憶領域の向上** チェックポイント・ログは、データベース・ファイルの最後にある連続ページ内に格納されるようになりました。これによって、チェックポイント・ログ内の項目に対して逐次スキャンと複数ページの書き込みが可能になり、パフォーマンスが向上しました。

チェックポイント・ログの詳細については、『ASA データベース管理ガイド』> 「チェックポイントとチェックポイント・ログ」を参照してください。

このリリースより前に作成されたデータベースでこの拡張機能を利用するには、データベースをアンロードして再ロードすることによってそのデータベースのファイル・フォーマットをアップグレードする必要があります。

- **プランのキャッシュ** Adaptive Server Anywhere では、クエリと、ストアド・プロシージャ、ユーザ定義関数、トリガで実行される INSERT 文、UPDATE 文、DELETE 文の実行プランがキャッシュされます。キャッシュできるプランの最大数は、MAX_PLANS_CACHED オプションで指定します。プランのキャッシュを無効にするには、このオプションを 0 に設定します。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「アクセス・プランのキャッシュ」を参照してください。

- **デフォルトの I/O コスト・モデルの上書き** ALTER DATABASE 文に CALIBRATE 句を指定すると、デフォルトの I/O コスト・モデルを上書きできるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「ALTER DATABASE 文」を参照してください。

- **新しいデータベース・オプション MAX_PLANS_CACHED** キャッシュに格納される実行プランの最大数を設定します。

詳細については、『ASA データベース管理ガイド』> 「MAX_PLANS_CACHED オプション [データベース]」を参照してください。

- **新しいデータベース・オプション MIN_TABLE_SIZE_FOR_HISTOGRAM** このオプションは、ヒストグラムを作成するテーブルの最小サイズを設定します。ヒストグラムには、1つのカラムでの値の分散状況に関する情報が格納されます。オプティマイザはこの情報をもとに、効率的な実行プランを選択します。

セキュリティ

- **TCP/IP に対する強力な暗号化** Adaptive Server Anywhere は、Solaris、Linux、NetWare、サポートされるすべての Windows OS (Windows CE を除く) の TCP/IP ポートで証明書による暗号化をサポートするようになりました。強力な暗号化によって、クライアントとサーバ間で交換されるネットワーク・パケットの機密性と整合性が保護されます。この暗号化は、トランスポート・レイヤ・セキュリティ (TLS: Transport Layer Security) とも呼ばれています。

ユーザは、Adaptive Server Anywhere の前のバージョンで使用されていた -e コマンド・ライン・オプションの代わりに、データベース・サーバの -ec コマンド・ライン・オプションを使用して、サーバの接続パラメータを設定できます。クライアントの接続パラメータは、暗号化接続パラメータと一緒に設定できます。

詳細については、『ASA データベース管理ガイド』> 「-ec サーバ・オプション」と『ASA データベース管理ガイド』> 「Encryption 接続パラメータ [ENC]」を参照してください。

この機能を使用するには、クライアントとサーバの両方でバージョン 8 ソフトウェアを使用する必要があります。データベースをアップグレードする必要はありません。

- **データベース・ファイルの強力な暗号化** 盗難の危険性が高いノートブック・コンピュータやラップトップ・コンピュータのセキュリティを強化するために、データベース・ファイル自体を強力に暗号化できるようになりました。

詳細については、次を参照してください。

- 『ASA データベース管理ガイド』> 「dbinit コマンド・ライン・ユーティリティを使用したデータベースの作成」
- 『ASA データベース管理ガイド』> 「-ek データベース・オプション」
- 『ASA データベース管理ガイド』> 「-ep サーバ・オプション」
- 『ASA SQL リファレンス・マニュアル』> 「CREATE DECRYPTED FILE 文」

暗号化データベース・ファイルを作成するには、バージョン 8 ソフトウェアを使用する必要があります。

SQL の機能

- **全外部ジョイン** 全外部ジョインがサポートされるようになりました。さらに、キーワード OUTER が、右、左、全外部ジョインのためのオプションになりました。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「外部ジョイン」を参照してください。

- **CASE 文** ANSI 規格では、2 つの形式の CASE 文を使用できます。Adaptive Server Anywhere 8.0 は、両方の構文をサポートします。

詳細については、『ASA SQL リファレンス・マニュアル』>「CASE 文」を参照してください。

- **WAITFOR 文** 指定された時間の間、または指定の時間になるまで現在の接続処理を遅らせます。

詳細については、『ASA SQL リファレンス・マニュアル』>「WAITFOR 文」を参照してください。

- **RAISERROR 文によって接続を拒否する** この文は、接続の拒否または制限に使用できるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』>「RAISERROR 文 [T-SQL]」を参照してください。

- **タイム・ゾーンの調整** タイム・ゾーン間の日付／時刻値の調整を簡単に行えるように、次の新機能が追加されています。

- **CURRENT UTC TIMESTAMP** サーバのタイム・ゾーン調整値を使用することでタイム・ゾーン値を調整します。
- **DEFAULT UTC TIMESTAMP** INSERT にデフォルト値を指定し、更新されたカラムをその値に設定します。
- **TimeZoneAdjustment プロパティ** 新しいローカル時間を表示するために、協定世界時 (UTC: Coordinated Universal Time) に加算する必要がある分数を返します。
- **TIME_ZONE_ADJUSTMENT オプション** 1 つの接続でタイム・ゾーンの調整を修正できます。

- **新しい照合関数** SORTKEY 関数は、文字データのソートに使用できる値を生成します。SORTKEY 関数を使用すれば、Adaptive Server Anywhere のデフォルトの照合よりも細かいソートを実行できます。

COMPARE 関数を使用すれば、代替照合規則に基づいて 2 つの文字列を直接比較できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「SORTKEY 関数 [文字列]」と『ASA SQL リファレンス・マニュアル』>「COMPARE 関数 [文字列]」を参照してください。

- **ERRORMSG 関数** 新しい SQL 関数 ERRORMSG を使用して、エラー・メッセージを取得できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「ERRORMSG 関数 [その他]」を参照してください。

- **データ型変換関数** ISDATE 関数と ISNUMERIC 関数は、文字列をそれぞれ日付または数値に変換できるかどうかをテストします。

詳細については、『ASA SQL リファレンス・マニュアル』>「ISDATE 関数 [データ型変換]」と『ASA SQL リファレンス・マニュアル』>「ISNUMERIC 関数 [その他]」を参照してください。

開発ツールと管理ツール

- **アクセシビリティ機能** SQL Anywhere Studio は、米国連邦リハビリテーション法 508 条に準拠しています。ユーザ・インタフェースとマニュアルはこの法律の規定に従って作成されています。このソフトウェアには、アクセシビリティ・ツールの使用を可能にするアクセシビリティ有効化コンポーネントが含まれています。アクセシビリティ有効化コンポーネントは、デフォルトではインストールされません。

詳細については、『SQL Anywhere Studio の紹介』>「SQL Anywhere Studio へようこそ」を参照してください。

- **クエリ・エディタ** Interactive SQL にグラフィカルなクエリ・エディタが追加されました。クエリ・エディタを使用すると、SQL コードを使用せずに SELECT 文を作成または編集できます。クエリ・エディタを開くには、Interactive SQL で [ツール] – [クエリの編集] をクリックします。

詳細については、『SQL Anywhere Studio ヘルプ』>「クエリ・エディタの概要」を参照してください。

- **Interactive SQL と Sybase Central でデータ編集が可能**
Interactive SQL 結果セットを編集するか、Sybase Central でテーブルとビューを編集することによって、データベースを更新できます。ローの値をコピー、編集、挿入、削除できます。

Sybase Central に表示されるデータを、クリップボードにコピーできます。

詳細については、『SQL Anywhere Studio の紹介』> 「Interactive SQL でのテーブル値の編集」を参照してください。

- **Interactive SQL での SQL エスケープ構文処理のサポート**

Interactive SQL では、JDBC ドライバで実装されている関数のライブラリへのアクセスを可能にする JDBC エスケープ構文をサポートするようになりました。

詳細については、『ASA プログラミング・ガイド』> 「JDBC エスケープ構文の使用」を参照してください。

- **プロシージャのプロファイル表示** Sybase Central の [プロファイル] タブでは、ストアド・プロシージャ、関数、イベント、トリガの呼び出し回数と実行回数についての情報が表示されます。また、プロシージャの各行の実行速度についての情報も参照できます。プロファイリング情報は、Sybase Central と SQL のストアド・プロシージャを使用して入手できます。

Sybase Central でプロシージャのプロファイリング情報を参照する方法については、『ASA SQL ユーザーズ・ガイド』> 「データベース・プロシージャのプロファイリング」を参照してください。

SQL ストアド・プロシージャを使用してプロシージャのプロファイリング情報を取得する方法については、『ASA SQL リファレンス・マニュアル』> 「sa_procedure_profile_summary システム・プロシージャ」および『ASA SQL リファレンス・マニュアル』> 「sa_procedure_profile システム・プロシージャ」を参照してください。

この機能を使用するには、データベースをアップグレードする必要があります。

- **アクセス・プラン情報表示の向上** アクセス・プランの表示方法に、グラフィカル表示と統計付きグラフィカル表示という 2 つの新しい方法が追加されました。これらの新しい方法ではクエリの処理コストに関する情報がさらに詳細に表示されるので、クエリのサブセットのコストを調べることができます。デフォルトのアクセス・プランは、グラフィカルなプランに設定され

ようになりました。長いプランと短いプランは、Adaptive Server Enterprise で使用される Ariadne 構文に基づいて作成され、新しい省略形が追加されています。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「アクセス・プランの解釈」を参照してください。

- **[結果] ウィンドウ枠でのクエリ実行プランの表示** Interactive SQL の [結果] ウィンドウ枠に、[結果] タブが追加されました。[結果] タブにはクエリの結果が、[プラン] タブにはクエリの実行プランが表示されます。以前は、クエリ実行プランは Interactive SQL の [メッセージ] ウィンドウ枠に表示されていました。

詳細については、『SQL Anywhere Studio ヘルプ』> 「[オプション] ダイアログ: [結果] タブ」を参照してください。

- **[結果] ウィンドウ枠での Ultra Light プランの表示** Interactive SQL の [結果] ウィンドウ枠に、[Ultra Light プラン] タブが追加されました。このタブには、Ultra Light プランの最適化方法が XML フォーマットの文字列で表示されます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「GRAPHICAL_ULPLAN 関数 [その他]」を参照してください。

- **OUTPUT 文を使用した XML のエクスポート** クエリ結果を XML 形式でエクスポートできます。出力には、埋め込み DTD が含まれます。バイナリ値は、2 桁の 16 進数文字列として表されるバイナリ・データとして CDATA ブロック内にエンコードされます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「OUTPUT 文 [Interactive SQL]」を参照してください。

- **Interactive SQL のバッチ・オプション** `-codepage` と `-onerror` コマンド・ライン・オプションを使用して、Interactive SQL でのバッチ・ファイルの実行を制御する機能が追加されました。また、`-d1` コマンド・ライン・オプションを使用すると、バッチ・ファイルのデバッグに役立つフィードバックも提供されます。

詳細については、『ASA データベース管理ガイド』> 「Interactive SQL ユーティリティ」を参照してください。

アプリケーション開発

- **新しいカーソル・タイプ** Adaptive Server Anywhere のカーソルが強化されて、整理されたセマンティックを提供したり、キーセット駆動型カーソルなどの新しいタイプのカーソルに適合するようになり、また、新しいクエリ最適化機能を最大限に利用できるようになりました。

詳細については、『ASA プログラミング・ガイド』> 「Adaptive Server Anywhere のカーソル」を参照してください。

- **長いカラムのフェッチの向上** 1 回の操作でフェッチできるデータ量が 32 KB から、設定可能な値までに増えました。デフォルトは 256 KB です。ODBC では、SQL_ATTR_MAX_LENGTH 文属性を使用してこの値を設定できます。Embedded SQL では、DT_LONGVARCHAR 型と DT_LONGBINARY 型を使用します。

詳細については、『ASA プログラミング・ガイド』> 「データの取り出し」と『ASA プログラミング・ガイド』> 「長い値の送信と取り出し」を参照してください。

- **データベースのプロパティを取得する新しい埋め込み SQL 関数** 関数 `db_get_property` を使用すると、データベースのプロパティを取得できます。

詳細については、『ASA プログラミング・ガイド』> 「db_get_property 関数」を参照してください。データベース・プロパティの詳細については、『ASA データベース管理ガイド』> 「データベース・プロパティ」を参照してください。

- **BLOCKING_TIMEOUT オプション** 新しい BLOCKING_TIMEOUT オプションを使用すると、ロックを取得するまでにトランザクションがどの程度の期間待つかを制御できます。

詳細については、『ASA データベース管理ガイド』> 「BLOCKING_TIMEOUT オプション [データベース]」を参照してください。

- **RETURN_DATE_TIME_AS_STRING オプション**
RETURN_DATE_TIME_AS_STRING オプションでは、jConnect と Open Client を介して、日付、時刻、タイムスタンプの値がどのように戻されるかを制御できます。

詳細については、『ASA データベース管理ガイド』> 「RETURN_DATE_TIME_AS_STRING オプション [データベース]」を参照してください。

管理機能とトラブルシューティング

バージョン 8 では、上記の Sybase Central に対する管理機能の強化の他に、次の管理機能も強化されています。

- **アクセスを中断させない場合のテーブル・パフォーマンスの向上** データベースへのアクセスを中断させないという稼働条件のためにデータベース全体の再構築が不可能なときは、REORGANIZE TABLE 文を使用してパフォーマンスを向上させることができます。この文を使って、テーブル内のローの断片化を解除したり、DELETE によって散在したインデックスを圧縮したりします。また、インデックス・ツリーに含まれるレベル数を減らす他に、テーブルとそのインデックスを格納するページの合計数も減らします。

プライマリ・キー、外部キー、またはインデックスに基づいてテーブルを再編成するには、データベースを Adaptive Server Anywhere バージョン 7 以上にしてください。

詳細については、『ASA SQL リファレンス・マニュアル』> 「REORGANIZE TABLE 文」を参照してください。
- **高速データベース検証** データベースの検証にかかる時間を短縮する新しいタイプの検証チェックが追加されました。このオプションは、小さなサイズのキャッシュを使用して大きなデータベースを検証する場合に、特に有用です。影響を受けるツールには、sa_validate システム・プロシージャ、検証ユーティリティ (dbvalid)、VALIDATE TABLE 文などがあります。

詳細については、『ASA データベース管理ガイド』> 「データベース検証時のパフォーマンスの改善」を参照してください。

このリリースより前に作成されたデータベースでこの機能を使用するには、データベースをアンロードして再ロードすることによってそのデータベースのファイル・フォーマットをアップグレードする必要があります。

- **バックアップで未処理トランザクションの完了を待機する必要なし** バックアップ命令でトランザクション・ログをトランケートするか名前を変更する必要がある場合、コミットされていないトランザクションは新しいトランザクション・ログに持ち越されます。つまり、サーバがバックアップを開始する前に、未処理のトランザクションをコミットまたはロールバックする必要はありません。

詳細については、『ASA データベース管理ガイド』> 「dbtran コマンド・ライン・ユーティリティを使用したトランザクション・ログの変換」と『ASA データベース管理ガイド』> 「バックアップの内部メカニズム」を参照してください。

このリリースより前に作成されたデータベースでこの機能を使用するには、データベースをアンロードして再ロードすることによってそのデータベースのファイル・フォーマットをアップグレードする必要があります。

- **断片化統計の取得** ファイル、テーブル、インデックスの断片化は、どれもパフォーマンスを低下させる可能性があります。Adaptive Server Anywhere 8.0 では、Windows NT 上でデータベースを起動すると、各 DB 領域でのファイルのフラグメント数に関する情報が自動的にサーバに表示されます。

データベース管理者は、新しいシステム・プロシージャ `sa_table_fragmentation` と `sa_index_density` を使用して、データベースのテーブルとインデックスで発生した断片化に関する情報を取得できます。

ファイルの断片化の詳細については、『ASA SQL ユーザーズ・ガイド』> 「ファイルの断片化」を参照してください。

テーブルの断片化の詳細については、『ASA SQL ユーザーズ・ガイド』> 「テーブルの断片化」と『ASA SQL リファレンス・マニュアル』> 「sa_table_fragmentation システム・プロシージャ」を参照してください。

インデックスの断片化の詳細については、『ASA SQL ユーザーズ・ガイド』>「インデックスの断片化」と『ASA SQL リファレンス・マニュアル』>「sa_index_density システム・プロシージャ」を参照してください。

- **接続のために準備された最新の SQL 文を取得する** データベース・サーバの `-z1` コマンド・ライン・オプションを有効にすると、サーバ上のデータベースへのそれぞれの接続に対して準備された最新の SQL 文を取得できます。この機能は、`remember_last_statement` 設定で `sa_server_option` ストアド・プロシージャを使用して有効にすることもできます。

この機能が有効になっている場合、**LastStatement** プロパティ関数はサーバ上のデータベースへの現在の接続用に準備された最新の SQL 文を返し、`sa_conn_activity` システム・プロシージャはすべての接続用に準備された最新の SQL 文を返します。

詳細については、『ASA データベース管理ガイド』>「`-z1` サーバ・オプション」、『ASA SQL リファレンス・マニュアル』>「`sa_conn_activity` システム・プロシージャ」、および『ASA SQL リファレンス・マニュアル』>「`sa_server_option` システム・プロシージャ」を参照してください。

- **`-cw` コマンド・ライン・オプション** このサーバ・オプションを使用すると、Windows 2000、Windows XP、Windows Server 2003 で最大 64 GB のキャッシュ・サイズを使用できます。

詳細については、『ASA データベース管理ガイド』>「`-cw` サーバ・オプション」を参照してください。

- **`-qp` オプション** このサーバ・オプションを使用すると、データベース・サーバ・ウィンドウにパフォーマンスに関するメッセージが表示されないようにできます。

詳細については、『ASA データベース管理ガイド』>「`-qp` サーバ・オプション」を参照してください。

- **サーバ・ログのデバッグ機能の向上** 接続デバッグにロギングされる情報が改善されて、接続試行中の部分に関するより詳細なコンテキストが提供され、CONN: プレフィックスが削除され、TCP/IP メッセージ数が増加されました。

- **データベースが保持できるプロシージャが増加** システム・テーブル SYSPROCEDURE、SYSPROCPARM、SYSPROCPERM、SYSTRIGGER のプライマリ・キーが、SMALLINT から UNSIGNED INT に変更されました。この変更によって、データベースが保持できるプロシージャ数が増加しています。

データベースが保持できるプロシージャ数の詳細については、『ASA データベース管理ガイド』> 「サイズと数の制限」を参照してください。

この機能を使用するには、データベースのファイル・フォーマットをアップグレードする必要があります。

- **クエリ・パフォーマンスのモニタリング** クエリ・パフォーマンスを測定する新しいシステム・プロシージャとユーティリティが追加されました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「sa_get_request_profile システム・プロシージャ」、『ASA SQL リファレンス・マニュアル』> 「sa_get_request_times システム・プロシージャ」、および『ASA SQL ユーザーズ・ガイド』> 「クエリのパフォーマンスのモニタリング」を参照してください。

- **新しい診断プロパティ** これらのプロパティを使用すると、接続、データベース、現在のデータベース・サーバについての情報を取得できます。今回のリリースには、次の接続プロパティが追加されています。

- UtilCmdsPermitted プロパティ
- TempTablePages プロパティ
- LastStatement プロパティ
- PacketSize プロパティ
- Max_plans_cached プロパティ
- QueryCachePages プロパティ
- QueryLowMemoryStrategy プロパティ

詳細については、『ASA データベース管理ガイド』> 「接続レベルのプロパティ」を参照してください。

今回のリリースには、次のデータベース・プロパティが追加されています。

- DBFileFragments プロパティ
- LogFileFragments プロパティ
- BlobArenas プロパティ
- SeparateForeignKeys プロパティ
- VariableHashSize プロパティ
- TableBitMaps プロパティ
- FreePageBitMaps プロパティ
- SeparateCheckpointLog プロパティ
- Histograms プロパティ
- LargeProcedureIDs プロパティ
- PreserveSource プロパティ
- TransactionsSpanLogs プロパティ
- Capabilities プロパティ
- TempTablePages プロパティ
- CompressedBTrees プロパティ
- ProcedurePages プロパティ
- QueryCachePages プロパティ
- QueryLowMemoryStrategy プロパティ

詳細については、『ASA データベース管理ガイド』> 「データベース・レベルのプロパティ」を参照してください。

今回のリリースには、次のサーバ・プロパティが追加されています。

- MachineName プロパティ
- IsJavaAvailable プロパティ

- PlatformVer プロパティ

詳細については、『ASA データベース管理ガイド』> 「サーバ・レベルのプロパティ」を参照してください。

- **パフォーマンス・モニタ統計値の追加** 今回のリリースでは、複数のパフォーマンス・モニタ統計値が追加されています。

詳細については、『ASA データベース管理ガイド』> 「データベース・パフォーマンスに関する統計情報」を参照してください。

- **ログイン・プロシージャによって接続を拒否する**

LOGIN_PROCEDURE オプションを使用すると、新しい接続ごとにストアド・プロシージャを呼び出すようにできます。このプロシージャによって、データベース接続を拒否することもできます。

詳細については、『ASA データベース管理ガイド』> 「LOGIN_PROCEDURE オプション [データベース]」を参照してください。

- **dbsvc の強化** Windows サービスを管理するための dbsvc コマンド・ライン・ユーティリティが拡張され、システム・コマンドの **net start** と **net stop** を使ってサービスを開始または停止するときに使用するサービス名をリストしたり、他のサービスやグループへの依存性を処理したりすることができるようになりました。

詳細については、『ASA データベース管理ガイド』> 「サービス作成ユーティリティ」を参照してください。

- **ストアド・プロシージャのソース・フォーマットを保持** スペースや改行を含むソース・フォーマットがコメントとしてデータベースに格納されるようになりました。このコメントは、プロシージャ・プロファイリングに使用されます。

クライアント／サーバ接続

- **向上したバッファ・サイズのネゴシエーション** クライアントとサーバの両方で、バッファ・サイズを個別に指定できるようになりました。

この機能を使用するには、クライアントとサーバの両方でバージョン 8 ソフトウェアを使用する必要があります。データベースをアップグレードする必要はありません。

- **通信の圧縮** 新しいタイプの通信の圧縮によって、無線ネットワーク、一部のモデム、シリアル・リンク、一部の WAN などの、帯域幅が限定されたネットワークでデータを転送する場合のパフォーマンスを向上させることができます。

詳細については、『ASA データベース管理ガイド』> 「パフォーマンス改善のための通信圧縮設定の調整」を参照してください。

この機能を使用するには、クライアントとサーバの両方でバージョン 8 ソフトウェアを使用する必要があります。データベースをアップグレードする必要はありません。

- **dbping の強化** *dbping* ユーティリティに、接続問題の診断を支援するオプションが追加されました。これらの機能には、ODBC を使用して接続する機能と、接続時の接続、データベース、サーバのプロパティをレポートする機能が含まれています。

詳細については、『ASA データベース管理ガイド』> 「Ping ユーティリティ」を参照してください。

- **TDS のデバッグを抑制するオプション**

SUPPRESS_TDS_DEBUGGING オプションは、TDS デバッグ情報をサーバ・ウィンドウに表示するかどうかを制御します。

詳細については、『ASA データベース管理ガイド』> 「SUPPRESS_TDS_DEBUGGING オプション [データベース]」を参照してください。

- **PrefetchBuffer 接続パラメータ** この接続パラメータを使うと、プリフェッチされたローを格納するメモリの最大容量を指定できます。

詳細については、『ASA データベース管理ガイド』> 「PrefetchBuffer 接続パラメータ [PBUF]」を参照してください。

- **PrefetchRows 接続パラメータ** PrefetchRows 接続パラメータを使うと、データベースのクエリ時にプリフェッチされるローの最大数を指定できます。状況によっては、クライアントによってデータベース・サーバからプリフェッチされるローの数を増やすと、クエリのパフォーマンスが向上することがあります。

詳細については、『ASA データベース管理ガイド』> 「PrefetchRows 接続パラメータ [PROWS]」を参照してください。

- **クライアントでアイドル・タイムアウトを指定可能** IDLE 接続パラメータを使用すると、各クライアントでアイドル・タイムアウトを指定できます。これまでは、サーバへのすべての接続で、-ti サーバ・コマンド・ライン・オプションを使用して共通のアイドル・タイムアウトを指定していました。

詳細については、『ASA データベース管理ガイド』> 「Idle 接続パラメータ [IDLE]」を参照してください。

データベース内の Java

データベース内の Java には、次の新機能が含まれています。

- **Java 2 のサポート** データベース内の Java は、Java 2 (JDK 1.2 と 1.3) と Java のクラスを使用できるようになりました。

詳細については、『ASA プログラミング・ガイド』> 「データベースを Java 実行可能にする」を参照してください。

この機能を使用するには、ALTER DATABASE を使用するか、*dbupgrad* コマンド・ライン・ユーティリティで -jdk オプションを指定してデータベースをアップグレードする必要があります。

- **JDBC 2.0** データベース内の Java クラスでは、JDBC 2.0 インタフェースを使用してデータにアクセスできるようになりました。

詳細については、『ASA プログラミング・ガイド』> 「データベース内の JDBC の機能」を参照してください。

この機能を使用するには、ALTER DATABASE を使用するか、*dbupgrad* コマンド・ライン・ユーティリティで -jdk オプションを指定してデータベースをアップグレードする必要があります。

- **診断プロシージャ** 新しいシステム・プロシージャ **sa_java_loaded_classes** は、Java 仮想マシンによってロードされたすべてのクラスをリストします。

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

この機能を使用するには、データベースをアップグレードする必要があります。

- **セキュリティ・マネージャ** 組み込みセキュリティ・マネージャを使用するか、ユーザ独自の実装で、セキュリティ関連の Java 機能に対するアクセスを制御できます。

詳細については、『ASA プログラミング・ガイド』> 「Java のセキュリティ管理」を参照してください。

マニュアル

Adaptive Server Anywhere マニュアル・セットに、情報の検索、アクセス、使用がより短時間で行える新機能がいくつか追加されました。

- **マニュアルの再構成** 前回のリリース以降、マニュアル・セットは、次の 2 点で大幅な変更が行われています。
 - 『レプリケーションおよび同期ガイド』は、2 冊に分割され、2 種類の同期テクノロジーを別々に説明しています。新しいマニュアルは『*Mobile Link 同期ユーザーズ・ガイド*』と『*SQL Remote ユーザーズ・ガイド*』です。
 - Adaptive Server Anywhere の『ユーザーズ・ガイド』、『プログラミング・インタフェース』、『リファレンス・マニュアル』は、『データベース管理ガイド』、『SQL ユーザーズ・ガイド』、『SQL リファレンス・マニュアル』、『プログラミング・ガイド』に置き換えられました。データベースに関するエラー・メッセージは、適切なマニュアルに移動されました。新しい構成によって、各マニュアルは、製本版として扱いやすいサイズになりました。

- **新しいコンテキスト別ヘルプ** Sybase Central、Interactive SQL、Adaptive Server Anywhere デバッガ、クエリ・エディタなどのすべてのユーザ・インタフェース・ツールは、プラットフォームを問わない共通のコンテキスト別ヘルプ・システムを共有しています。このシステムは、オンライン・マニュアルと完全にリンクしています。
- **オンライン・マニュアルの強化** オンライン・マニュアルの HTML ヘルプ・バージョンに、SQL Anywhere の Web リンク、チュートリアル、プロシージャなどにすぐにアクセスできるメニュー・バーが追加されました。

その他

- **ハイバネーション後も持続する接続** Embedded SQL、ODBC、または OLE DB クライアントからの接続は、コンピュータがハイバネーション状態にある間も持続されるようになりました。これまでは、同一マシン上のクライアントとサーバ間の TCP/IP 接続は、指定された活性またはアイドル・タイムアウト時間より長くハイバネーション状態にあったマシンが復帰すると、切断されていました。
- **現在のライセンス情報の表示** dblic ユーティリティでは、サーバを起動せずに、サーバの実行プログラムに関する現在のライセンス情報を表示できる引数を使用できるようになりました。

詳細については、『ASA データベース管理ガイド』> 「ライセンス・ユーティリティ」を参照してください。

- **カスタム照合の照合ラベルと照合名の表示** dbinfo コマンド・ライン・ユーティリティは、カスタム照合用の照合ラベルと照合名を返すようになりました。また、*dbtools.h* の *a_db_info* 構造体に *collationnamebuffer* と *collationnamebufsize* という 2 つのフィールドが追加されました。

詳細については、『ASA データベース管理ガイド』> 「dbinfo コマンド・ライン・ユーティリティを使用したデータベース情報の取得」と『ASA プログラミング・ガイド』> 「a_dbtools_info 構造体」を参照してください。

- **sp_remote_tables システム・プロシージャ** sp_remote_tables ストアド・プロシージャに、新しい引数 **tabletype** が追加されました。この引数は、リモート・テーブルのタイプを返します。

引数 **tabletype** の詳細については、『ASA SQL リファレンス・マニュアル』> 「sp_remote_tables システム・プロシージャ」を参照してください。

- **-ct コマンド・ライン・オプション** -ct コマンド・ライン・オプションを使用すると、文字セット変換を有効または無効に設定できます。文字セット変換はデフォルトでは有効に設定されるようになりました。変換を無効に設定するには、-ct- を指定します。文字セット変換を有効にするには、-ct+ と指定します。

詳細については、『ASA データベース管理ガイド』> 「-ct サーバ・オプション」を参照してください。

- **リモート・テーブルの外部キー情報の取得** 2つの新しいストアド・プロシージャ sp_remote_exported_keys と sp_remote_imported_keys を使用して、リモート・テーブルの外部キーと対応するプライマリ・キーに関する情報を取得できるようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「sp_remote_exported_keys システム・プロシージャ」と『ASA SQL リファレンス・マニュアル』> 「sp_remote_imported_keys システム・プロシージャ」を参照してください。

- **xp_sendmail** SMTP や MAPI を経由して電子メールを送信するための拡張ストアド・プロシージャが追加されました。詳細については、『ASA SQL リファレンス・マニュアル』> 「xp_startsmtp システム・プロシージャ」と『ASA SQL リファレンス・マニュアル』> 「xp_stopsmtp システム・プロシージャ」を参照してください。

xp_sendmail ストアド・プロシージャには、任意の長さのメッセージを指定できます。このプロシージャに対する LONG VARCHAR パラメータの長さは、システムで利用できるメモリ容量によって制限されます。

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

- **Log Transfer Manager のための Replication Server 12 の機能**
LTM 設定ファイル内の **qualify_table_owner** パラメータは、プライマリ・データベースとレプリケーション・データベースで、異なるテーブル名、所有者、カラム名を使用できる Replication Server 12 の機能をサポートします。

詳細については、『ASA データベース管理ガイド』>「LTM 設定ファイル」を参照してください。

- **ASANYSH8 環境変数** 新しい環境変数 ASANYSH8 が追加されました。Interactive SQL、Sybase Central、Adaptive Server Anywhere コンソール・ユーティリティ、デバグガでは、この環境変数を使用して、共有コンポーネントのディレクトリを検索します。

ASANYSH8 環境変数の詳細については、『ASA データベース管理ガイド』>「環境変数の設定」を参照してください。

Mobile Link の新機能

バージョン 8.0 で導入されたソフトウェアの変更と追加を次に示します。

柔軟性

- **Java 同期論理** SQL 言語の代わりまたはそれに加えて、Java で記述した同期スクリプトを実装できるようになりました。これらのスクリプトは、Mobile Link Java 環境を使用して外部 JRE の中で実行できます。

詳細については、『Mobile Link 管理ガイド』>「Java による同期スクリプトの作成」を参照してください。

- **パブリケーションを使用した同期** Mobile Link クライアント内のすべてのデータを、同時に同期させる必要はなくなりました。その代わりに、データをパブリケーションに編成し、パブリケーションごとに別々に同期できます。パブリケーションと同期サブスクリプションには、以前の構文よりも簡単で精度の高い新しい構文が用意されています。

詳細については、『Mobile Link クライアント』>「Adaptive Server Anywhere クライアント」を参照してください。

- **Mobile Link 同期を処理する Web サーバの設定** ファイアウォールの後ろ側で、Mobile Link 同期サーバとの HTTP 同期を実行できるようになりました。標準的な Web サーバ用の Web サーバ・プラグインによって、Web サーバ経由の HTTP 同期を実行できます。

詳細については、『Mobile Link 管理ガイド』>「リダイレクタによる Web サーバを経由した同期」を参照してください。

- **Windows CE クライアントでの ActiveSync のサポート** Adaptive Server Anywhere と Ultra Light Windows CE Mobile Link のクライアントは、いずれも Windows CE ActiveSync 同期ソフトウェアを使用できます。

詳細については、『Mobile Link クライアント』>「ActiveSync 同期の使用」を参照してください。

- **クライアントのコマンド・ライン機能の強化** CREATE/ALTER SYNCHRONIZATION SUBSCRIPTION 文とコマンド・ラインの両方で、拡張オプションを指定できます。

詳細については、『Mobile Link クライアント』>「Mobile Link 同期クライアント」を参照してください。

- **拡張オプションをデータベース内に格納できる** CREATE/ALTER SYNCHRONIZATION SUBSCRIPTION 文を使用して、拡張オプションと接続パラメータをデータベースに保存し、サブスクリプション、ユーザ、またはパブリケーションと関連付けることができます。Dbmlsync は、この情報をデータベースから読み込みます。

詳細については、『Mobile Link クライアント』>「Mobile Link 同期クライアント」を参照してください。

パフォーマンス

- **文ベースのアップロード** Mobile Link では、カーソルベースのアップロードよりも直感的に扱えるだけでなく処理速度も速い、文ベースのアップロードを行えるようになりました。文ベースのアップロードでは、**upload_insert**、**upload_delete**、

upload_update、**upload_new_row_insert**、**upload_old_row_insert** の各イベントが使用されます。競合の解決には、**upload_fetch** スクリプトが使用されます。

詳細については、『Mobile Link 管理ガイド』> 「ローをアップロードするスクリプトの作成」を参照してください。

- **マルチプロセッサの管理** 使用するプロセッサの最大数を設定するオプションが Mobile Link に新しく追加されました。-zt オプションを使用して、Mobile Link 同期サーバで使用されるリソースをより細かく制御できます。また、複数のプロセッサを使用した場合の ODBC の問題の検出や解決の際にも役立ちます。

詳細については、『Mobile Link 管理ガイド』> 「-zt オプション」を参照してください。

- **ダウンロード確認のオプション** Mobile Link 同期クライアントは、ダウンロード確認なしで同期できるようになりました。したがって、Mobile Link 同期サーバのワーカ・スレッドは、クライアントがダウンロードを適用するまで待つ必要がなく、早期にワーカ・スレッドを解放して次の同期を実行できます。ダウンロード確認は、オプションになりました。ダウンロード確認を省略することで、スループットが向上します。特に、処理の遅いクライアントでは顕著な効果が現れます。ダウンロード確認を省略した場合、統合データベース側では次の同期が開始されるまでダウンロードが成功したかどうかを知ることはできません。

詳細については、『Mobile Link クライアント』> 「dbmlsync 拡張オプション」を参照してください。

- **ダウンロード・ストリームのバッファ** Mobile Link 同期サーバは、ダウンロード・キャッシュにダウンロード・ストリームをバッファするようになりました。ダウンロード・トランザクションをコミットするのにクライアントからの確認は必要ないので、バッファされたダウンロード・ストリームは、コミット後にクライアントに送信されます。これにより、これまでのようにネットワークの遅延によってダウンロード・トランザクションの処理が滞ることもなくなります。

詳細については、『Mobile Link 管理ガイド』> 「-d オプション」を参照してください。

ダウンロード・ストリームは、Adaptive Server Anywhere クライアント側でもバッファできます。使用可能なバッファのサイズは、*dbmlsync* DownloadBufferSize 拡張オプションを使って設定できます。

詳細については、『Mobile Link クライアント』> 「dbmlsync 拡張オプション」を参照してください。

- **接続スクリプトとテーブル・スクリプトのバルク・ロード** テーブルと version_id の特定のペアに対して初めて接続スクリプトまたはテーブル・スクリプトを要求すると、すべてのスクリプトがキャッシュにバルク・ロードされます。すべてのスクリプトを個別に取得するのではなく一括して取得することによって、パフォーマンスが向上します。
- **Mobile Link 同期サーバのシャットダウン処理の強化** Mobile Link 同期サーバが完全にシャットダウンを完了してから処理を実行するように、*dbmlstop* に通知することができます。また、*dbmlstop* を使用して、特定の Mobile Link 同期サーバの名前を指定して停止できます。

詳細については、『Mobile Link 管理ガイド』> 「Mobile Link 停止ユーティリティ」を参照してください。

- **接続タイムアウト** 指定された時間にわたって使用されなかった Mobile Link データベース接続は、サーバによって自動的に切断されるようになりました。タイムアウトは、-ct (接続タイムアウト) コマンド・ライン・オプションを使用して設定できます。

詳細については、『Mobile Link 管理ガイド』> 「-ct オプション」を参照してください。

- **同時アップローダの最大数オプション** -wu コマンド・ライン・オプションを使用して、同時アップロードを許可するワーカ・スレッドの最大数を設定できます。この結果、一部の配備ではスループットが増大します。

詳細については、『Mobile Link 管理ガイド』> 「-wu オプション」を参照してください。

セキュリティ

- **Mobile Link ユーザ認証** パスワードに基づくユーザ認証システムによって、Mobile Link のインストール環境のセキュリティがさらに強化されます。さらに、`-zu` を使用すると、`authenticate_user` スクリプトが定義されていない場合にユーザを自動的に追加できます。これによって、ユーザのスキーマ情報を Mobile Link の認証として使用できます。

詳細については、『Mobile Link クライアント』> 「Mobile Link ユーザの認証」を参照してください。

- **Mobile Link ユーザの管理** `dbmluser` ユーティリティが拡張され、システムからユーザを削除したり、システムにユーザを追加したりできるようになりました。このユーティリティについてはその他にも拡張機能が追加されています。`dbmluser` コマンド・ライン・オプション `-pf`、`-pp`、`-pu` は廃止され、それぞれ `-f`、`-p`、`-u` に置き換えられました。

詳細については、『Mobile Link 管理ガイド』> 「Mobile Link ユーザ認証ユーティリティ」を参照してください。

レポート機能の強化

- **統計スクリプト** Mobile Link に、同期統計を追跡するためのスクリプトが追加されました。これらの同期統計を収集すると、その後の同期のパフォーマンスの監視に使用できます。

詳細については、『Mobile Link 管理ガイド』> 「`synchronization_statistics` 接続イベント」、『Mobile Link 管理ガイド』> 「`synchronization_statistics` テーブル・イベント」、『Mobile Link 管理ガイド』> 「`upload_statistics` 接続イベント」、および『Mobile Link 管理ガイド』> 「`upload_statistics` テーブル・イベント」を参照してください。

- **詳細なネットワーク・エラー情報** Mobile Link 同期のサーバとクライアントは、詳細なエラー情報を表示できるようになり、発生したすべてのエラーを、エラー・コードを参照して解決できるようになりました。エラーをレポートしたネットワーク・レイヤ、実行されたネットワーク操作、エラー内容、システム固有のエラー・コードを確認できます。

- **エラー発生時に Mobile Link 同期サーバに送信される Remote Adaptive Server Anywhere の出力ログ** 同期問題のトラブルシューティングは、リモート・ログと Mobile Link 同期サーバ・ログの両方を検査するのが最も簡単な方法です。この新機能では、クライアント側でエラーが発生したときに、ASA リモートの出力ログが Mobile Link 同期サーバに送信されます。

詳細については、『Mobile Link 管理ガイド』> 「-e オプション」を参照してください。

- **ログ・メッセージがワーカ・スレッドを識別する** Mobile Link 同期サーバ・ログに表示されるメッセージは、そのメッセージをロギングしたワーカ・スレッドを示すようになりました。これによって、1 人のユーザが同時に同期させようとしたために発生したメッセージを識別できます。また、1 人のユーザが続けて 2 回同期させたときのメッセージも識別できます。

- **冗長ロギング** Mobile Link 同期サーバの -v コマンド・ライン・オプションに追加の変更子を使用すると、Mobile Link 同期サーバのロギングを設定できます。

詳細については、『Mobile Link 管理ガイド』> 「-v オプション」を参照してください。

- **無視されたローがクライアントにレポートされる** スクリプトがないために、Mobile Link 同期サーバがアップロードされたローを無視した場合、クライアントにメッセージが返されます。このメッセージは、Adaptive Server Anywhere クライアントからの警告として表示され、Ultra Light クライアントの ignored_rows 同期パラメータにも表示されます。

詳細については、『Mobile Link クライアント』> 「Ignored Rows 同期パラメータ」を参照してください。

使いやすさ

- **最終ダウンロード・タイムスタンプ** 前回のダウンロードのタイムスタンプが Mobile Link クライアント・データベースに自動的に書き込まれます。

- **自動同期スクリプトの生成** スナップショットを使用した同期に適したスクリプトを生成するように、Mobile Link に指示できます。これらのスクリプトの作成とアクティブ化は、`-za` オプションで制御します。

詳細については、『Mobile Link 管理ガイド』> 「スクリプトの自動生成」を参照してください。

- **サンプル同期スクリプトの生成** サンプル同期スクリプトを生成するように、Mobile Link に指示できます。`-ze` コマンド・ライン・オプションを使用して、サンプル・スクリプトを生成するかどうかを指定します。

詳細については、『Mobile Link 管理ガイド』> 「サンプル・スクリプトの生成」を参照してください。

適合性

- **一般的な RDBMS のサポート** Mobile Link は、統合データベースとして、Oracle 8i と 9i、Microsoft SQL Server 7、Microsoft SQL Server 2000、IBM DB2などをサポートします。

詳細については、『Mobile Link 管理ガイド』> 「Mobile Link でサポートされる ODBC ドライバ」を参照してください。

- **TCP/IP ストリームでの活性検出** Mobile Link の同期中に使用される TCP/IP ベースのストリームは、クライアントとサーバの両側で、活性チェックを有効にする **keep_alive** と呼ばれる新しいパラメータを利用できるようになりました。

詳細については、『Mobile Link 管理ガイド』> 「`-x` オプション」を参照してください。

Ultra Light の新機能

Ultra Light 8.0 には、次の新機能が導入されています。

セキュリティ

- **ユーザの認証** これまでのリリースでは、Ultra Light データベースには、アクセスを管理するためのユーザ認証メカニズムがありませんでした。今回のリリースには、ユーザ認証メカニズムが組み込まれています。多くのリレーショナル・データベース

管理システムのユーザ ID とは異なり、Ultra Light のユーザ ID は、テーブルやその他のデータベース・オブジェクトの所有権を意味しません。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「Ultra Light のユーザ認証」を参照してください。

- **データベースの暗号化** データベースを暗号化することで、データのセキュリティを強化することができます。以下の 2 つの方法があります。
 - **強力な暗号化** 強力な暗号化アルゴリズムを使用してデータベースを暗号化すると、最高レベルのセキュリティを得ることができます。このセキュリティには、パフォーマンスの低下が伴います。この暗号化は、キーを使用するものであり、AES 128 ビット・アルゴリズムを使用しています。
 - **データベースの難読化 (Obfuscation)** データベースを読みにくくすることで、データのセキュリティを強化することができます。難読化しない場合、16 進数エディタなどのツールを使ってデータベース内のデータを表示できます。難読化は、データを表示しようとする程度の軽い不正アクセスは防げますが、強力な暗号化のような厳重な保護は行いません。難読化では、強力な暗号化に伴うパフォーマンスの低下は発生しません。
- **Ultra Light Java アプリケーションの安全な同期** Certicom トランスポート・レイヤ・セキュリティを使用した安全な同期は、これまでは C/C++ の Ultra Light アプリケーションでのみ行えました。今回のリリースでは、Ultra Light Java アプリケーションでも行えるようになりました。

詳細については、『Ultra Light 静的型 Java ユーザーズ・ガイド』> 「トランスポート・レイヤ・セキュリティ」を参照してください。

同期

- **ActiveSync 同期** Windows CE デバイス上で動作する Ultra Light アプリケーションは、ActiveSync を使用して同期させることができます。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』>「アプリケーションへの ActiveSync 同期の追加」を参照してください。

- **同期の柔軟性の向上** 同期させるデータをさらに効率よく柔軟に選択できるようにする新機能がいくつか追加されました。

- パブリケーションを使用すると、データを複数のセットに分割し、セットごとに別々に同期させることができます。処理速度が遅い接続リンクなどでは、これによって時間が重要なデータを効率的に同期でき、時間が重要ではない他のデータは都合のよいときに同期できます。

- ダウンロード専用同期を使用すると、読み込み専用テーブルを Ultra Light データベースに追加して、それらを効率的に同期できます。

詳細については、『Mobile Link クライアント』>「Ultra Light データベースへの読み込み専用テーブルの組み込み」を参照してください。

- テーブル内のデータが変更されているか否かに関係なく、毎回同期するようにテーブルをマークできます。この機能を使用すると、同期を制御する Ultra Light クライアント上のユーザ設定情報を管理できます。

詳細については、『Mobile Link クライアント』>「同期を制御するクライアント固有のデータの使用」を参照してください。

- **グローバル・オートインクリメントのデフォルト・カラム値** この機能を使って、同期中のデータベース内のプライマリ・キーの一意性を簡単に管理できます。

詳細については、『Mobile Link クライアント』>「デフォルトのグローバル・オートインクリメント・カラムの宣言」を参照してください。

- **Ultra Light ジェネレータの制御の追加** `ulgen` と `sqlpp` 実行プログラム用のコマンド・ライン・オプションが追加されました。
- **スクリプトのバージョン** 生成された同期スクリプトに、スクリプトのバージョンを関連付けることができます。
- **ログ・クエリ実行プラン** 生成されたクエリのクエリ実行プランをエクスポートし、Interactive SQL で表示できます。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「Ultra Light ジェネレータ」と 『Ultra Light データベース・ユーザーズ・ガイド』> 「SQL プリプロセッサ」を参照してください。

- **エラー・レポート** `ul_synch_info` 構造の `stream_error` フィールドを使って、同期エラーの原因を判断できます。

詳細については、『Mobile Link クライアント』> 「Stream Error 同期パラメータ」を参照してください。

データベース管理

- **既存のデータベースの再使用** Ultra Light のこれまでのリリースでは、データベース・アプリケーションが変更された場合は、データベースを再構築して同期させる必要がありました。今回のリリースより、データベースのスキーマが変更されないかぎり、新バージョンのアプリケーションで Ultra Light データベースを続けて使用できるようになりました。クエリを変更した場合は、そのクエリが新しいカラムを参照することで、生成済みのデータベースのスキーマが変更されないかぎり、新しいデータベースを必要としません。
- **データベースの断片化解除** Ultra Light の記憶領域は、空き領域を効率的に再使用するよう設計されているので、通常の状態では明示的な断片化解除を必要としません。領域要件が極端に厳しいアプリケーションのために、明示的な断片化解除機能が用意されています。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「Ultra Light データベースのデフラグメンテーション」を参照してください。

- **ページ・サイズを選択** デフォルトの 4 KB ページに代わって 2 KB ページ・サイズを選択できます。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』>
「UL_STORE_PARMS マクロ」を参照してください。

開発用の機能

- **CodeWarrior 7 のサポート** CodeWarrior 用の Ultra Light プラグインは、CodeWarrior バージョン 7 をサポートするようになりました。
- **eMbedded Visual C++** このツールを使用した開発がサポートされ、CustDB サンプル・アプリケーション用の eMbedded Visual C++ プロジェクトが用意されました。
- **Palm OS 4.0 とファイル・ベースのデータ記憶装置** Ultra Light は Palm Computing Platform のバージョン 4.0 をサポートします。Palm ではバージョン 4.0 以降、さまざまな補助記憶装置スキームが導入されています。Palm 4.0 デバイスの拡張カードではファイル・ベースの Ultra Light データ・ストアを使用できます。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』>
「ULEnableFileDB 関数」を参照してください。

- **Palm Computing Platform の同期機能の向上** Palm Computing Platform の HotSync と ScoutSync による同期には、新しい簡略化された同期メカニズムが導入されています。新しい同期メカニズムには、これまでの同期メカニズムと比べて次のような長所があります。
 - 起動と終了が速い。
 - 同期の実行時に Palm デバイス上に追加の記憶装置が不要。
 - アプリケーションを複数回同期でき、そのたびに起動する必要がない。
 - ストリーム・パラメータの指定が不要。

ULPalmDBStream 関数と **ULConduitStream** 関数は廃止されました。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』>
「Palm アプリケーションに HotSync 同期を追加する」を参照してください。

- **Palm Computing Platform での配備の簡略化** Ultra Light データベースの初期コピーをエンド・ユーザに配備できるので、最初の同期操作でユーザごとにデータの初期コピーをダウンロードする必要がありません。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』>「Palm アプリケーションの配備」を参照してください。

- **Palm セグメントの処理方法の向上** Palm Computing Platform 用のアプリケーションを開発するときは、限定されたサイズのセグメントにアプリケーション・コードを分割する必要があります。

このソフトウェアの以前のバージョンのセグメンテーション方法では、ユーザは Ultra Light で生成されるコードのセグメンテーションを制御できず、割り当てられるセグメントが多くなりすぎる傾向がありました (これはパフォーマンスの低下を招きます)。新しいメカニズムでは、生成されるセグメントの数が減少するとともに、ユーザがセグメントの割り当てを制御できます。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』>「マルチセグメント・コード生成の有効化」を参照してください。

- **Embedded SQL の LONG 値** DECL_LONGVARCHAR と DECL_LONGBINARY を使用して、LONG 値 (32 KB から 64 KB の間) にホスト変数を使用できます。
- **リファレンス・データベースでのアナライザのフック** Ultra Light ジェネレータは、分析処理の前後にストアド・プロシージャを呼び出すようになりました。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』>「SQL プリプロセッサ」と『Ultra Light データベース・ユーザーズ・ガイド』>「Ultra Light ジェネレータ」を参照してください。

- **クエリ・プラン情報** Ultra Light ジェネレータで、Ultra Light アプリケーションでのクエリに使用されるアクセス・プランを出力できるようになりました。また、Interactive SQL から Ultra Light に使用されるアクセス・プランを表示することもできます。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「Ultra Light ジェネレータ」を参照してください。

- **スクリプトのバージョン制御** Ultra Light ジェネレータのコマンド・ラインで同期に使用されるスクリプト・バージョンを指定できます。

詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「Ultra Light ジェネレータ」を参照してください。

- **SQL と API 機能の追加** Ultra Light アプリケーションに対して、次の機能を使用できるようになりました。

- **@@identity のサポート** Ultra Light で @@identity グローバル変数がサポートされるようになりました。この機能は、グローバル・オートインクリメントのデフォルト・カラム値に関連して使用できます。C++ API では、**ULConnection::GetLastIdentity()** メソッドを使用します。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「GetLastIdentity メソッド」を参照してください。

- **テーブル内のローの数** **ULTable::GetRowCount()** メソッドを使用すると、C++ API プログラミング・インタフェースからテーブル内のローの数を判断できます。Embedded SQL ユーザは、引き続き **SELECT COUNT(*) FROM table-name** 文を使用します。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「GetRowCount メソッド」を参照してください。

- **テーブル内のすべてのローの削除** **ULTable::DeleteAllRows()** メソッドを使用すると、C++ API プログラミング・インタフェースからテーブル内のすべてのローを削除できます。Embedded SQL ユーザは、引き続き **DELETE FROM table-name** 文を使用します。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「DeleteAllRows メソッド」を参照してください。

- **影響を受けたローの数** Embedded SQL から SQLCOUNT マクロを使用して、最新の INSERT、UPDATE、または DELETE 文によって影響を受けたローの数を調べることができます。
- **アップロードされるローの数** 同期する必要があるローの数を調べることができます。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「ULCountUploadRows 関数」と 『Ultra Light C/C++ ユーザーズ・ガイド』> 「CountUploadRows メソッド」を参照してください。

- **最終ダウンロード時間** Ultra Light アプリケーションから前回パブリケーションをダウンロードした時間を取得できます。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「ULGetLastDownloadTime 関数」と 『Ultra Light C/C++ ユーザーズ・ガイド』> 「GetLastDownloadTime メソッド」を参照してください。

- **カーソル処理の追加** C++ API の ULTable クラスに、結果セット内のローを検索するためのメソッド (FindFirst、FindNext、FindPrevious、FindLast) が追加されました。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』> 「ULTable クラス」を参照してください。

- **DUMMY システム・テーブルからのクエリ** SELECT ... FROM DUMMY 形式のクエリがサポートされるようになりました。
- **複数テーブルの更新** 複数のテーブルに対するカーソルによって、複数のテーブルを修正する更新を受け付けることができるようになりました。
- **Embedded SQL での LONG データ型の処理の向上** DECL_LONGVARCHAR と DECL_LONGBINARY ホスト変数型を使用すると、1 回の操作で 32 KB を超えるデータを送信または検索ができます。

詳細については、『Ultra Light C/C++ ユーザーズ・ガイド』
> 「Embedded SQL のデータ型」を参照してください。

SQL Remote の新機能

- **イベント・フック・プロシージャ** レプリケーション・プロセスのカスタマイズを可能にする一連のイベント・フック・プロシージャが追加されました。指定された名前のストアド・プロシージャを作成すれば、Message Agent がレプリケーションの実行中に行うアクションに対してさまざまな時点でカスタマイズを追加できます。

詳細については、『SQL Remote ユーザーズ・ガイド』> 「SQL Remote イベント・フック・プロシージャ」を参照してください。

バージョン 8 での動作の変更

この項では、SQL Anywhere Studio バージョン 8 のコンポーネントに導入された動作の変更をリストします。

Adaptive Server Anywhere の動作の変更

ここでは、これまでのバージョンとは異なる動作について説明します。

新たに廃止され、サポートされなくなった機能のリストについては、「[廃止されサポートされなくなった機能](#)」225 ページを参照してください。

- **データベース内の Java は、別々にライセンス設定が可能** この結果、データベース作成時のデフォルト動作では、データベース内の Java のサポートが除外されます。

Ultra Light ジェネレータが外部 Java VM を使用するように変更されているので、データベース内の Java は、Ultra Light リファレンス・データベースで必要ではなくなりました。

詳細については、『SQL Anywhere Studio の紹介』> 「SQL Anywhere Studio へようこそ」を参照してください。

- **集合関数と外部参照** Adaptive Server Anywhere のバージョン 8 は、サブクエリでの集合関数の使用法を明確に規定した新しい SQL/99 標準に準拠しています。この変更は、Adaptive Server Anywhere の以前のバージョン用に記述された文の動作に影響を与えます。以前は正しかったクエリでエラー・メッセージが生成され、結果セットが変わる可能性があります。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「集合関数と外部参照」を参照してください。

- **ユーザが提供する選択性推定** Adaptive Server Anywhere では、アクセス・プランの選択を支援する選択性推定を明示的に指定できます。これらの推定は、ソフトウェアが選択したアクセス・プランが不適切な場合のパフォーマンス問題を回避するときに

最も有効でした。新しい `USER_ESTIMATES` 接続オプションは、ユーザが提供した選択性推定をオプティマイザが使用するか無視するかを制御します。

パフォーマンス問題を回避するために選択性推定を使用している場合は、明示的な推定が不正確になり、オプティマイザが不適切なプランを選択する可能性があるため、`USER_ESTIMATES` オプションの設定を `OFF` にすることをおすすめします。今回のバージョンでは、内部ジョイン・アルゴリズムなどのクエリ処理の強化が追加されているため、クエリのパフォーマンスが大幅に向上しています。

ユーザが提供する選択性推定の詳細については、『ASA データベース管理ガイド』>「`USER_ESTIMATES` オプション [データベース]」と『ASA SQL リファレンス・マニュアル』>「明示的な選択性推定」を参照してください。

- **ローの順序** バージョン 8.0 では、クエリ処理の向上による副作用として、ローの順序の確定性が低下しています。`ORDER BY` 句がない場合は、Adaptive Server Anywhere が最も効率のよい順序でローを返します。これは、結果セットの提示が、最後にアクセスしたローとその他の要因によって変化する可能性があることを意味します。特定の順序でローが返されるようにする唯一の方法は `ORDER BY` を使用することです。

`LIST` 関数は、この変更によって特に影響を受ける関数の 1 つです。

- **アクセス・プランの変更** Adaptive Server Anywhere の今回のリリースで選択されるアクセス・プランは、以前のリリースほどインデックスを使用しない傾向があります。テーブル・スキャンの効率を向上させ、アクセス・プランのコスト比較時に選択の幅があるコスト・モデルを使用することによって、インデックスの利便性をこれまでのバージョンよりも正確に評価できるようになりました。
- **カーソルの変更** カーソルの強化の副効用として、このバージョンのカーソルは、定義された規格に以前よりも近い動作をするようになりました。このため、Adaptive Server Anywhere が ODBC やその他のインタフェースの要求に合致する動作を行うように、一部のカーソルの反応が変更される場合があります。

たとえば、Embedded SQL の SCROLL カーソルではプリフェッチが実行できなくなったため、値の変更がカーソルに反映されません。

この変更は、SQL_SUCCESS へのリターン・コードのみをチェックし、SQL_SUCCESS_WITH_INFO のチェックはしていない既存のアプリケーションに影響します。

SQL_SUCCESS_WITH_INFO に対してチェックを行うアプリケーションでは、カーソルの動作が要求された動作と異なる場合は、警告を受け取ります。この警告は、SQLCODE=121、SQLSTATE 01S02 です。

insensitive カーソルは更新できません。

詳細については、『ASA プログラミング・ガイド』> 「insensitive カーソル」を参照してください。

- **ストアド・プロシージャの格納** ストアド・プロシージャは、書かれたとおりに格納されるようになりました。Adaptive Server Anywhere は、ストアド・プロシージャの内部表現を作成し、これをプロファイリングに使用します。
- **挿入時の OPEN CURSOR はサポートされない** INSERT 文でカーソルを開く機能は削除されました。更新可能なカーソルは、業界標準と同じように SELECT 文を使用して開くことができます。
- **ユーザ定義関数** ユーザ定義関数のパラメータと戻り値は、キャッシュされるようになりました。SQL 文の中で関数が複数回使用される場合は、キャッシュされたパラメータ値は、関数を再評価するのではなく、使用されたキャッシュ済みの結果になります。以前のリリースでは、ユーザ定義関数が必要になるたびに再評価されていました。この新しい動作によってパフォーマンスが向上し、さらに一貫性のある結果が得られますが、以前のリリースのソフトウェアと比較すると結果が異なっている場合があります。
- **NUMBER(*) 関数の変更** NUMBER 関数は、不確定な動作を回避するために使用を制限されています。NUMBER は、クエリの選択リストの中で、結果セットのローに連番を振ることを目的としていますが、この使い方はまだ許可されています。

NUMBER 関数は、-1 の値を使用して絶対フェッチを実行した後、カーソルを逆向きに移動するなどの動作を行った場合に、以前のバージョンでは発生しなかった負の値を戻すことがあります。この新しい動作は、ISO/ANSI フェッチ・オフセットに対応しています。

NUMBER 関数を WHERE 句または HAVING 句など、多くの状況で使用すると、エラーが発生するようになりました。

詳細については、『ASA SQL リファレンス・マニュアル』>「NUMBER 関数 [その他]」を参照してください。

- **カスタム照合の変更** 以前は、照合ユーティリティの -d オプションは 3 つのパラメータを受け付けていましたが、現在使用できるパラメータは 2 つだけになりました。**cust-map-file** パラメータは、現在は受け付けられません。照合ユーティリティの構文は、次のとおりです。

```
dbcollat -d coll-defn-file custom-file
```

さらに、現在はスクリプト・ファイル *collsqmp.sql* と *custmap.sql* は存在しないため、それぞれ組み込み照合とカスタム照合で使用できません。

新しく作成されたデータベースでは、SYSCOLLATIONMAPPINGS テーブルに照合マッピングを持つローが 1 つだけ格納されます。Adaptive Server Anywhere の以前のバージョンで作成されたデータベースでは、このテーブルに組み込み照合のためのローが格納されています。

詳細については、『ASA データベース管理ガイド』>「照合ユーティリティ」と『ASA SQL リファレンス・マニュアル』>「SYSCOLLATIONMAPPINGS システム・テーブル」を参照してください。

- **トリガ名の変更** トリガ名は、データベース全体でユニークなものにする必要はなくなりました。ユニークである必要があるのは、トリガが適用されるテーブルの中だけです。したがって、DROP TRIGGER と COMMENT ON TRIGGER の構文が変更され、テーブルも指定した場合には所有者だけを指定できるようにな

りました。つまり、所有者だけでトリガを修飾する古いスクリプトでは、「テーブルが見つかりません」のエラーが発生します。

- **サンプル・データベース内のアドレスの変更** Adaptive Server Anywhere 9.0 のサンプル・データベース内のアドレスは、以前のリリースのものとは異なります。
- **内部 JDBC ドライバの JAR ファイル名の変更** 内部 JDBC ドライバのクラスは ASAJDBC ではなく ASAJRT という名前の JAR ファイルでインストールされるようになりました。
- **RESTORE DATABASE 文のパーミッション** RESTORE DATABASE 文を実行するのに、ユーティリティ・データベースに接続する必要はなくなりました。RESTORE DATABASE 文を実行するために必要なパーミッションは、-gu コマンド・ライン・オプションによって制御されます。

詳細については、『ASA SQL リファレンス・マニュアル』>「RESTORE DATABASE 文」を参照してください。

- **TDS 接続では空文字列を NULL 文字列で返す** TDS_EMPTY_STRING_IS_NULL オプションは、TDS 接続でサーバが空文字列を返す場合、ブランク文字 1 文字を含む文字列または NULL 文字列のどちらで返すかを制御します。

詳細については、『ASA データベース管理ガイド』>「TDS_EMPTY_STRING_IS_NULL オプション [データベース]」を参照してください。

- **COMMENT 文の変更** 以前は、COMMENT ON INDEX の構文には、インデックスの所有者名がオプションで含まれていました。インデックス名に所有者とテーブルをオプションに入れることができるようになりました。COMMENT ON INDEX の構文は次のとおりです。

COMMENT ON INDEX [[owner.]table.]index-name IS comment

詳細については、『ASA SQL リファレンス・マニュアル』>「COMMENT 文」を参照してください。

- **文字セット変換がデフォルトで有効** Adaptive Server Anywhere の以前のバージョンでは、文字セット変換はデフォルトで無効に設定され、有効にするには、`-ct` コマンド・ライン・オプションを指定する必要がありました。文字セット変換はデフォルトで有効になりましたが、`-ct` コマンド・ライン・オプションを使用すると無効にできます。

サーバが接続の文字セットとデータベースの文字セットが異なると判断すると、サーバはその接続との間で送受信されるすべての文字列に対して文字セット変換を適用します。

サーバがデータベースと接続の文字セットが等しいと判断すると、サーバはその接続に対する文字セット変換を無効にします。

ほとんどの場合、文字セット変換は有効に設定しておいてください。設定を無効にするのは、バイナリ・データがデータベースに挿入され、文字データとしてフェッチされた場合 (あるいはその逆の場合) です。サーバは文字データに対してのみ文字セット変換を適用するため、このような場合には、入力されたとおりのデータが返されることがあります。この問題を回避するには、アプリケーションでバイナリ型を使って文字データの送信やフェッチを行わないようにします。

詳細については、『ASA データベース管理ガイド』> 「`-ct` サーバ・オプション」と『ASA データベース管理ガイド』> 「データベース・サーバで文字セット変換をオフにする」を参照してください。

- **CONVERT、TIMESTAMP_FORMAT、DATE_FORMAT**
TIMESTAMP_FORMAT または DATE_FORMAT オプションを使用するときに、大文字と小文字の混在する文字記号 (Mmm など) を指定すると、Adaptive Server Anywhere は使用されている言語に適切な大文字または小文字を選択します。また、CONVERT 関数は、使用されている言語に適切な大文字または小文字に文字日付を変換するようになりました。たとえば、英語では May、フランス語では mai が正しい大文字と小文字の用法です。

詳細については、『ASA データベース管理ガイド』> 「DATE_FORMAT オプション [互換性]」、『ASA データベース管理ガイド』> 「TIMESTAMP_FORMAT オプション [互換性]」、および 『ASA SQL リファレンス・マニュアル』> 「CONVERT 関数 [データ型変換]」を参照してください。

- **3 値的ブール論理への変更** 2 値的ブール論理は、*expr* = NULL の場合にのみ適用されます。ここで、*expr* はベース・カラムまたはベース・カラムについての式を指します。それ以外の場合は、3 値的論理が適用されます。ANSINULL オプションは、クエリの WHERE 句のこの特別なケースにのみ影響するようになりました。
- **Sybase Central と Interactive SQL が COMMLINKS 接続パラメータを受け入れる** Adaptive Server Anywhere の以前のバージョンでは、Sybase Central と Interactive SQL (*dbisql* ユーティリティ) は、COMMLINKS 接続パラメータを無視していました。Sybase Central と Interactive SQL は、このパラメータを受け入れるようになりました。

この変更により、一部の接続文字列の動作が、Adaptive Server Anywhere の以前のバージョンでの動作とは異なる場合があります。具体的には、COMMLINKS=tcPIP を指定しないと、Interactive SQL と Sybase Central はネットワーク上のサーバを探しません。

詳細については、『ASA データベース管理ガイド』> 「CommLinks 接続パラメータ [LINKS]」を参照してください。

- **クライアントは SQLLOCALE 環境変数を見捨てる** クライアントは CharSet 接続パラメータを使用して接続に使用する文字セットを指定できます。Adaptive Server Anywhere の以前のバージョンでは、CharSet 接続パラメータの指定がない場合、クライアントのデフォルト文字セットの変更には、SQLLOCALE 環境変数の CHARSET パラメータが使用されていました。クライアントはこの SQLLOCALE 環境変数を見捨てるようになりました。
- サポートされない文字セットを使用すると接続が失敗するクライアントは CharSet 接続パラメータを使用して接続に使用する文字セットを指定できます。ただし、サーバが要求された文字

セットをサポートしていないと、接続は失敗します。Adaptive Server Anywhere の以前のバージョンでは、クライアントがサポートされていない文字セットを要求すると、警告は表示されましたが、接続は確立されました。クライアントが文字セットを指定せず、サーバがクライアントのローカル文字セットをサポートしていない場合、接続は確立されますが、文字セットがサポートされていないことを知らせる警告が表示されます。

この動作は、バージョン 6.x、バージョン 7.x、バージョン 8 データベース・サーバに接続するバージョン 8 クライアントで発生します。

- **デフォルト・パケット・サイズの変更** クライアント／サーバ通信のデフォルト・パケット・サイズが、1024 バイトから 1460 バイトに変更されました。

パケット・サイズの詳細については、『ASA データベース管理ガイド』> 「CommBufferSize 接続パラメータ [CBSIZE]」と『ASA データベース管理ガイド』> 「-p サーバ・オプション」を参照してください。

- **dbdsn ユーティリティが管理するのは Adaptive Server Anywhere データ・ソースのみ** Adaptive Server Anywhere ODBC データ・ソースを管理する dbdsn コマンド・ライン・ユーティリティは、Adaptive Server Anywhere データ・ソースのみを管理するように明示的に制限されるようになりました。

- **LOGIN_PROCEDURE オプションには DBA 権限が必要** LOGIN_PROCEDURE オプションは、DBA 権限を持つユーザだけが設定できます。Adaptive Server Anywhere の以前のバージョンでは、このオプションは DBA 権限を持たなくても設定できました。DBA 権限を持つユーザは、他のユーザの設定も変更できますが、DBA 権限を持たないユーザは自分の設定の変更もできません。この変更により、DBA は、必要に応じて、ユーザの接続時に確実に共通のプロシージャを実行することができます。

詳細については、『ASA データベース管理ガイド』> 「LOGIN_PROCEDURE オプション [データベース]」を参照してください。

- **ESTIMATE_SOURCE が返す新しい値** ESTIMATE_SOURCE 関数は、以前よりも詳細な値を返します。

詳細については、『ASA SQL リファレンス・マニュアル』>「ESTIMATE_SOURCE 関数 [その他]」を参照してください。

廃止されサポートされなくなった機能

このリストには、サポートを終了した機能の中で既存のアプリケーションに影響のあるものがまとめられています。

- **NetWare 4.10 のサポート終了** Novell NetWare バージョン 4.11 以降は引き続きサポートされています。バージョン 3.x と 4.10 はサポートされていません。
- **NetBIOS のサポート終了** NetBIOS ポートは、サポートされなくなりました。NetBIOS を使用する場合は、TCP/IP または SPX に切り替えてください。
- **IPX のサポート終了** IPX ポートは、サポートされなくなりました。IPX を使用する場合は、SPX または TCP/IP に切り替えてください。
- **使わなくなった照合** 次の照合は、サポートされなくなりました。別の照合に置き換えられた照合は、その旨明記されています。

使用されなくなった照合	置き換えられた照合
437	437LATIN1
850	850LATIN1
852	852LATIN2
860	860LATIN1
863	863LATIN1
865	865NOR
SJIS	932JPN
SJIS2	932JPN
WIN_LATIN1	1252LATIN1

使用されなくなった照合	置き換えられた照合
WIN_LATIN5	1254TRK
Internal	850LATIN1
437EBCDIC	

- e オプションのサポート終了** データ・ソース・ユーティリティでクライアント／サーバ通信を暗号化するために使用されていた `-e` コマンド・ライン・オプションと `-e` オプションは、サポートされなくなりました。代わりに、`-ec` オプションが使用されます。サーバ側の `-ec simple` は、Adaptive Server Anywhere の以前のバージョンの `-e` と同じ暗号化アルゴリズムを使用します。
- NONE パラメータの廃止** ISQL_PLAN オプションの NONE パラメータは、サポートされなくなりました。クエリ最適化プランは、[結果] ウィンドウ枠の [プラン] タブに表示されるようになりました。[プラン] タブをクリックすると、常にプランが表示されます。以前は、プランは [メッセージ] ウィンドウ枠に表示されていました。
- WITH HASH SIZE n 句の廃止** WITH HASH SIZE 句は、サポートされなくなりました。
- MAX_WORK_TABLE_HASH_SIZE オプションの廃止** MAX_WORK_TABLE_HASH_SIZE オプションはサポートされなくなりました。
- MAX_HASH_SIZE オプションの廃止** MAX_HASH_SIZE オプションは、サポートされなくなりました。
- SATMP 環境変数の廃止** Adaptive Server Anywhere の UNIX バージョンで使用されていた、テンポラリ・ファイルの保存ディレクトリを示す SATMP 環境変数は、サポートされなくなりました。UNIX では、ASTMP 環境変数を使用してテンポラリ・ファイルの保存場所を示すことができます。

詳細については、『ASA データベース管理ガイド』> 「ASTMP 環境変数」を参照してください。

- **dbtran -id オプションの削除** 新しいバージョンでは、dbtran コマンド・ライン・ユーティリティの -id コマンド・ライン・オプションが廃止されました。

Mobile Link の動作の変更

- **Mobile Link Adaptive Server Anywhere クライアントの設定**
Mobile Link クライアントは、同期定義ではなく、パブリケーションと同期サブスクリプションを使用して設定されるようになりました。
- 詳細については、『Mobile Link クライアント』>「Adaptive Server Anywhere クライアント」を参照してください。
- **最後ダウンロード・タイムスタンプ・パラメータでスクリプトを変更する** 多数のスクリプトに対応する 1 つの新しいパラメータが追加され、タイムスタンプ・ベースの同期の実装が容易になりました。この新しいパラメータは、多数のスクリプトに対する最初のパラメータとして指定されるため、既存のスクリプトは中断されます。既存スクリプトの使用を続行するには、-zd Mobile Link 同期サーバ・コマンド・ライン・オプションを指定して、最終ダウンロード・タイムスタンプを最後のパラメータとして指定するように動作を変更してください。
 - **Mobile Link のシャットダウン** 以前は、dbmlstop コマンドを使用して、リモート接続から Mobile Link 同期サーバをシャットダウンできました。現在は、Mobile Link 同期サーバと同じマシンからの dbmlstop 要求だけが、Mobile Link 同期サーバをシャットダウンするようになりました。dbmlstop にサーバを停止させることを許可する -zs オプションは、必要なくなりました。
 - **TCP/IP ストリーム内の活性検出デフォルト設定の変更**
keep_alive のデフォルト設定は、1 (ON) になりました。
 - **Mobile Link で dbmluser 情報を非表示にできる** dbmluser コマンド・ライン・ユーティリティを使用したときに表示された、タイムスタンプ、著作権、その他の Mobile Link 同期サーバ・メッセージなどの情報は、デフォルトでは表示されなくなりました。

- **Mobile Link ユーザ認証** Mobile Link ユーザ認証を使用しない場合は、Mobile Link 同期サーバ・コマンドで `-zu+` オプションを使用してください。
- **デフォルトのログ拡張子が .mls に変更** 各ファイルには、DDMMYYNN.MLS という名前が付けられるようになりました。DD は日、MM は月、YY は西暦年を表します。NN は、1 から始まるファイルの連続番号です。

Ultra Light の動作の変更

- **Palm アプリケーションに必要なコード変更** Palm Computing Platform バージョン 4.x に標準のレコード・ベースのデータベース記憶領域を使用するか、ファイル・ベースの拡張カード記憶装置を使用するかをコードで指定する必要があります。関数呼び出しを 1 つ追加してから、ULPalmLaunch (Embedded SQL の場合) または ULData.PalmLaunch (C++ API の場合) を呼び出してください。追加する関数呼び出しは次のとおりです。

```
ULEnablePalmRecordDB( &sqlca );
```

または

```
ULEnableFileDB( & sqlca );
```

レコード・ベースの記憶装置を使用する場合は

ULEnablePalmRecordDB を、ファイル・ベースの記憶装置を使用する場合は **ULEnableFileDB** を指定します。デバイスがファイル・ベースの記憶装置をサポートしていない場合、ULPalmLaunch によって SQLCODE -82 が設定されます。

Ultra Light による以下のプラットフォームのサポートは終了しました。

- **DOS ターゲット・プラットフォーム** DOS は、サポート対象でなくなりました。
- **Metrowerks CodeWarrior 5 開発プラットフォーム** Ultra Light を開発するには、CodeWarrior 6 が必要になりました。

- **Palm 2.x のサポート終了** Ultra Light は、PalmPilot Professional などの Palm OS 2.x デバイスのための開発をサポートしなくなりました。バージョン 3.0 以上が必要です。
- **ULPalmDBStream と ULConduitStream の廃止** Palm Computing Platform での HotSync または ScoutSync による同期に新しい同期ストリームが導入されたことで、**ULPalmDBStream** 関数と **ULConduitStream** 関数は使用されなくなりました。これらの関数は受け入れられますが、効果はありません。
- **Ultra Light ジェネレータは、外部 Java VM を使用する** Ultra Light アナライザはデータベース・エンジンの外部で実行されるようになったので、Java が有効化されていないリファレンス・データベースに対しても使用できます。
- **Ultra Light JDBC パッケージ名の変更** Ultra Light JDBC 関数のパッケージ名は、**com.sybase.asa.ultralite.jdbc** から **ianywhere.ultralite.jdbc** に変更されました。したがって、Ultra Light アプリケーションに使用される `import` 文を変更する必要があります。
- **すべての変更は、ダウンロード同期の前にコミットが必要** 同期させる前にすべての変更をコミットするという規則が、ダウンロード専用同期にも適用されるようになりました。
- **dbmlsync StreamCompression 拡張オプションの廃止** このオプションは無視されます。

Adaptive Server Anywhere の動作の変更のいくつかは、アプリケーションに影響を及ぼす可能性もあるので、その内容についても確認してください。詳細については、「[Adaptive Server Anywhere の動作の変更](#)」[217 ページ](#)を参照してください。

第 8 章

バージョン 7.0.3 の新機能

この章の内容

この章では、SQL Anywhere Studio バージョン 7.0.3 で導入された新機能と変更された動作の概要について説明します。

新しい機能

この項では、Adaptive Server Anywhere バージョン 7.0.3 の新機能について説明します。新機能の一覧を挙げ、各機能の詳細が説明されている参照先も示します。

- **ブランク埋め込みと大文字と小文字の区別のためのデータベース・プロパティ** 文字列比較時にデータベースでブランク埋め込みを使用するか (**BlankPadding**)、またはデータベースで大文字と小文字を区別するか (**CaseSensitive**) を決める 2 つのプロパティが新しく追加されました。

詳細については、『ASA データベース管理ガイド』> 「データベース・レベルのプロパティ」を参照してください。

- **C2 セキュリティ・モード用のサーバ・プロパティ** データベース・サーバが `-sc` オプションを使って起動されたかどうかを判定する、新しい C2 サーバ・プロパティが追加されました。`-sc` オプションは、C2 基準を満たした環境で使うためのものです。

詳細については、『ASA データベース管理ガイド』> 「サーバ・レベルのプロパティ」を参照してください。

- **ログイン・プロシージャで接続のブロックが可能**
login_procedure オプションを使用すると、新しい接続ごとにストアド・プロシージャを呼び出すようにすることができます。このプロシージャによって、データベース接続を拒否することもできます。

詳細については、『ASA データベース管理ガイド』> 「LOGIN_PROCEDURE オプション [データベース]」を参照してください。

- **UNIX 上で FileDSN をサポート** ODBC データ・ソース用の FileDSN 接続パラメータが UNIX でサポートされるようになりました。

動作の変更

ここでは、これまでのバージョンとは異なる動作について説明します。

- **LOAD TABLE のセマンティックの変更** カラム・リストが指定されたときの LOAD TABLE コマンドのセマンティックが改善されました。カラム・リストには、ファイル内の各カラムを出現順に指定しなければなりません。リストにないカラム名は、カラムの NULL 入力属性やデータ型、デフォルト動作などにより、NULL、ゼロ、空文字列、またはデフォルト値のいずれかに設定されます。

LOAD TABLE によって無視される入力ファイル内のカラムは、カラム名 **filler()** を使用して指定できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「LOAD TABLE 文」を参照してください。

第 9 章

バージョン 7.0.2 の新機能

この章の内容

この章では、Adaptive Server Anywhere バージョン 7.0.2 で導入された新機能と変更された動作の概要について説明します。

バージョン 7.0.2 の新機能

この項では、SQL Anywhere Studio バージョン 7.0.2 のコンポーネントに導入された新機能について説明します。

Adaptive Server Anywhere の新機能

この項では、Adaptive Server Anywhere バージョン 7.0.2 の新機能について説明します。新機能について主要なものからそうでないものまですべてを示し、各機能の詳細についての参照先も記述しています。

- **動的キャッシュ・サイズ決定** Windows 95/98 では、データベース・サーバの負荷とその他のシステム・メモリへの要求に応じて、データベース・サーバのキャッシュ・サイズが増減します。この機能を使用すると、さまざまな状況で明示的にキャッシュ・サイズを選択する必要がなくなり、パフォーマンスが向上します。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「動的キャッシュ・サイズ決定 (Windows NT/2000/XP、Windows 95/98/Me)」を参照してください。

- **現在のライセンス情報の表示** ライセンス [dblic] ユーティリティでは、サーバを起動せずに、サーバの実行プログラムに関する現在のライセンス情報を表示できる引数を使用できるようになりました。

詳細については、『ASA データベース管理ガイド』> 「ライセンス・ユーティリティ」を参照してください。

- **照合の追加** ロシア語とウクライナ語 (1251CYR、ANSI コード・ページ 1251) をサポートする照合、トルコ語 (1254TRK、ANSI コード・ページ 1254) をサポートする照合、ドイツ語 (1252DEU、ANSI コード・ページ 1252) の一部のユーザをサポートするための特別な照合の 3 つの照合が使用可能になりました。

ドイツ語では、1252LATIN1 照合を使用することをおすすめします。1252DEU は特別な照合です。この照合は、ソートや比較の特性を理解した上で使用するようにしてください。

使用可能な照合の完全なリストについては、『ASA データベース管理ガイド』> 「照合の選択」を参照してください。

- **Interactive SQL のリターン・コード** Interactive SQL をコマンド・プロンプトから実行したときに、セッション中の動作の成功または失敗を示すプログラム終了コードを設定できるようになりました。

詳細については、『ASA データベース管理ガイド』> 「Interactive SQL ユーティリティ」を参照してください。

- **DELETE_OLD_LOGS の強化** DELETE_OLD_LOGS データベース・オプションは、レプリケーション環境でオフライン・トランザクション・ログを管理するときに使用されます。このオプションが、処理済みのトランザクション・ログを削除するときに細かい制御ができるように強化されました。

詳細については、『ASA データベース管理ガイド』> 「DELETE_OLD_LOGS オプション [レプリケーション]」を参照してください。

- **接続のトラブルシューティングと強化** クライアント／サーバ通信のトラブルシューティングとチューニングを強化するために、次の変更が行われました。

- クライアントのデバッグ・ログ・ファイルに、APPINFO 文字列が追加されました。

詳細については、『ASA データベース管理ガイド』> 「AppInfo 接続パラメータ [APP]」を参照してください。

- 新しい2つの接続パラメータを使用すると、ローのプリフェッチをチューニングできます。

詳細については、『ASA データベース管理ガイド』> 「PrefetchRows 接続パラメータ [PROWS]」と『ASA データベース管理ガイド』> 「PrefetchBuffer 接続パラメータ [PBUF]」を参照してください。

- これまで、**ConnectionName** 接続パラメータの値は、ODBC クライアントに対して上書きされていました。今回のリリースでは、ODBC クライアントから **ConnectionName** パラメータを使用できるようになりました。

接続パラメータのリストについては、『ASA データベース管理ガイド』>「接続パラメータ」を参照してください。

- **言語ユーティリティ** 言語 [dblang] ユーティリティを使用して、Adaptive Server Anywhere のメッセージと Sybase Central インタフェース要素の言語レジストリについてレポートと修正ができます。

詳細については、『ASA データベース管理ガイド』>「言語ユーティリティ」を参照してください。

- **dbspawn の強化** プロセス生成 [dbspawn] ユーティリティでは、データベース・サーバのオペレーティング・システム・プロセス ID をオプションでレポートできます。

詳細については、『ASA データベース管理ガイド』>「プロセス生成ユーティリティ」を参照してください。

- **週の最初の曜日オプション** 週の最初の曜日のデフォルトが7になりました。これは日曜日に当たります。この値は、平日の値を取得するときの DATEPART の結果に影響します。週の最初の曜日は、Transact-SQL SET 文の DATEFIRST オプションを使用して変更できます。また、SET OPTION FIRST_DAY_OF_WEEK=*n* を使用して、週の最初の曜日を永続的に設定できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「SET 文 [T-SQL]」または『ASA データベース管理ガイド』>「FIRST_DAY_OF_WEEK オプション [データベース]」を参照してください。

- **新しいマイグレーション・ツール** 新しいストアド・プロシージャのセット sa_migrate を使用して、Oracle、DB2、Microsoft SQL Server、Sybase Adaptive Server Enterprise、Sybase Adaptive Server Anywhere、Microsoft Access のリモート・データベースを Adaptive Server Anywhere にマイグレート (インポート) できます。

詳細については、『ASA SQL ユーザーズ・ガイド』>「Adaptive Server Anywhere へのデータベースの移行」を参照してください。

- **イベント・ハンドラ** Adaptive Server Anywhere は、ある時点で実行されている特定のイベント・ハンドラのインスタンス数を決定できるようになりました。この機能を使用して、イベント・ハンドラが一度に 1 つだけ実行されるように制限できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「EVENT_PARAMETER 関数 [システム]」を参照してください。

- **新しい接続プロパティ** 新しい接続プロパティによって、イベント・ハンドラを実行するために使用する内部接続を互いに区別できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「CONNECTION_PROPERTY 関数 [システム]」を参照してください。

- **dbdsn によるユーザ指定子とシステム指定子のサポート** データ・ソース [dbdsn] ユーティリティは、u (ユーザ) と s (システム) オプションをサポートするようになりました。

詳細については、『ASA データベース管理ガイド』>「データ・ソース・ユーティリティ」を参照してください。

- **@filename ファイル内のコメントのサポート** Adaptive Server Anywhere は、@filename ファイルにあるコメント行をサポートするようになりました。

詳細については、『ASA データベース管理ガイド』>「@data サーバ・オプション」を参照してください。

- **タイムスタンプ・トランケート・オプション** Adaptive Server Anywhere 以外のデータベースとの互換性を高めるために、タイムスタンプ値をトランケートできるようになりました。

詳細については、『ASA データベース管理ガイド』>「TRUNCATE_TIMESTAMP_VALUES オプション [データベース]」を参照してください。

- **ライセンス情報の取得** 使用中の Adaptive Server Anywhere に関する正確なライセンス情報を取得できるように、エンジン・プロパティが追加されました。

詳細については、『ASA データベース管理ガイド』> 「サーバ・レベルのプロパティ」を参照してください。

- **オートインクリメント値のリセット** `sa_reset_identity` システム・プロシージャを使用すると、次のローのオートインクリメント値をリセットできます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「`sa_reset_identity` システム・プロシージャ」を参照してください。

Mobile Link の新機能

- **アップロード・ストリームを適用するスレッドの最大数** データベース競合を極力避けるために、`-wu` コマンド・ライン・オプションを使用して、同時アップロードできるワーカ・スレッドの最大数を設定できるようになりました。アップロード要求は、要求を受け付けた順序で処理されます。
- 詳細については、『Mobile Link 管理ガイド』> 「Mobile Link 同期サーバのオプション」を参照してください。

バージョン 7.0.2 での動作の変更

この項では、SQL Anywhere Studio バージョン 7.0.2 のコンポーネントに導入された動作の変更について説明します。

Adaptive Server Anywhere の動作の変更

ここでは、これまでのバージョンとは異なる動作について説明します。

- **エイリアスは最初の参照よりも前に定義する必要あり** SQL Anywhere の以前のバージョンでは、エイリアスが定義される前に、そのエイリアスを SELECT リストで参照できました。今回のリリースでは、そのような操作を行うと、エラー「エイリアス alias 名の定義は、最初の参照前に記述する必要があります。」が発生します。このエラーを避けるために、SELECT リストを並べ替えて、エイリアスが定義されてから最初の使用が行われるようにする必要があります。

第 10 章

バージョン 7.0.1 の新機能

この章の内容

この章では、Adaptive Server Anywhere バージョン 7.0.1 で導入された新機能と変更された動作の概要について説明します。

バージョン 7.0.1 の新機能

この項では、SQL Anywhere Studio バージョン 7.0.1 のコンポーネントに導入された新機能について説明します。

Adaptive Server Anywhere の新機能

この項では、Adaptive Server Anywhere バージョン 7.0.1 の新機能について説明します。新機能について主要なものからそうでないものまですべてを示し、各機能の詳細についての参照先も記述しています。

- **新しいサービス作成ユーティリティ** データベース・サーバを NT 環境下でサービスとして実行させることで、データベースはそれが実行されているマシンに制限されることなく実行を続けることができます。これまでは、サービスの追加は、Sybase Central から [新しいサービスの作成] ウィザードを使用して行っていました。Adaptive Server Anywhere のバージョン 7 では、サービス作成 [dbsvc] ユーティリティを使用して、Windows NT 上で ASA サービスを管理することもできるようになりました。さまざまなオプションを使用して、サービスの追加と削除、すべての ASA サービスのリスト、または特定のサービスの詳細表示ができます。この機能は、インストール環境でサービスの作成を埋め込むときに特に便利です。

サービス作成ユーティリティの詳細については、『ASA データベース管理ガイド』> 「サービス作成ユーティリティ」を参照してください。

- **Windows CE 3.0 のサポート** Adaptive Server Anywhere は、Windows CE 2.11 の他に、以下のプロセッサ上で動作する Windows CE 3.0 もサポートするようになりました。
 - MIPS
 - Hitachi SH3
 - ARM

Windows CE 2.11 は広範囲のプラットフォームでサポートされます。詳細については、『SQL Anywhere Studio の紹介』>「Windows オペレーティング・システムと NetWare オペレーティング・システム」を参照してください。

Windows CE 3.0 のサポートによって、CE 上の OLE DB ドライバは、追加ソフトウェアをインストールしなくても動作します。

- **Embedded SQL の強化** 新しい関数 **db_locate_servers** は、TCP/IP 上で受信している Adaptive Server Anywhere データベース・サーバの場所をプログラムによって検出します。

詳細については、『ASA プログラミング・ガイド』>「db_locate_servers 関数」を参照してください。

新しいコールバック関数 **DB_CALLBACK_CONN_DROPPED** を使用すると、データベース・サーバの接続が切断されときのロジックを追加できます。

詳細については、『ASA プログラミング・ガイド』>「db_register_a_callback 関数」を参照してください。

- **接続レベルのデバッグと Logfile 接続パラメータ** クライアント側の接続パラメータである DBG と LOG は接続に固有となったので、1 つのアプリケーションからでも、接続ごとにデバッグ情報を設定できます。

詳細については、『ASA データベース管理ガイド』>「Logfile 接続パラメータ [LOG]」を参照してください。

- **新しいデータベース・プロパティ** Replication Agent または LTM のユーザのために、**LTMGeneration** プロパティが追加されました。このプロパティは、主にテクニカル・サポートを行うために使用されます。

詳細については、『ASA データベース管理ガイド』>「データベース・レベルのプロパティ」を参照してください。

- **新しい配備機能** InstallShield Professional 5.5 以降を使用している場合は、新しい SQL Anywhere Studio InstallShield Template Projects を使用して、作成したアプリケーションを配備できま

す。これによって、テンプレート・プロジェクトの全部またはインストールに適用する部分だけを使用して、アプリケーションのインストール・プログラムを迅速に構築できます。

詳細については、『ASA プログラミング・ガイド』> 「InstallShield を使用した配備」を参照してください。

- **Backup 文の新機能** Backup 文を使用するときに、ディレクトリに空の文字列を指定することで、ログをコピーしないで名前を変更したり、トランケートしたりできます。これは、スペースが問題となるレプリケーション環境では特に便利です。この機能をトランザクション・ログ・サイズのイベント・ハンドラと一緒に使用して、ログが所定のサイズに達したときに名前を変更したり、DELETE_OLD_LOGS オプションと一緒に使用して、ログがなくなかったときに削除したりすることができます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「BACKUP 文」を参照してください。

Mobile Link の新機能

次に、バージョン 7.0.1 で導入したソフトウェアに加えられた変更と追加を示します。

- **ユーザの認証** パスワードに基づくユーザ認証システムによって、Mobile Link のインストール環境のセキュリティがさらに強化されます。

詳細については、『Mobile Link クライアント』> 「Mobile Link ユーザの認証」を参照してください。

- **トランスポート・レイヤ・セキュリティ・マニュアルの増補** トランスポート・レイヤ・セキュリティのマニュアルが改訂され、この強力なセキュリティ・メカニズムで構築可能なさまざまなアーキテクチャの説明が加わりました。

詳細については、『Mobile Link 管理ガイド』> 「Mobile Link トランスポート・レイヤ・セキュリティ」を参照してください。

- **同期と同期関連プロセスのカスタマイズ** Adaptive Server Anywhere 同期クライアント *dbmlsync* が、一連のイベントをサポートするようになりました。Adaptive Server Anywhere データベースにストアド・プロシージャを追加して、イベント・ベースのアクションをプログラムできます。これによって、同期処理に柔軟性が加わります。同期のスケジュール機能もその 1 つです。

詳細については、『Mobile Link クライアント』>「クライアント同期処理のカスタマイズ」を参照してください。

- **同期の最適化** 同期処理の以下の点を最適化できます。
 - Ultra Light クライアント・アプリケーションでは、同期にアップロードだけを含み、ダウンロード・フェーズを行わないように指定できます。
アップロードだけが必要な場合は、このオプションによって、同期にかかる時間が短縮されます。
 - Adaptive Server Anywhere クライアントでは、インクリメンタル・アップロード・オプションを指定して、多くのアップロードを行う際に必要となるメモリを少なくできます。
詳細については、『Mobile Link クライアント』>「同期のカスタマイズ」を参照してください。
 - Adaptive Server Anywhere クライアントでは、同期中にローの同時修正が可能です。
詳細については、『Mobile Link クライアント』>「同期中の同時実行性」を参照してください。
- **同期のスケジュール** *dbmlsync* ユーティリティの拡張オプションまたは同期定義を使用すると、スケジュールに基づいて同期が実行できます。
詳細については、『Mobile Link クライアント』>「同期のスケジュール」を参照してください。
- **Adaptive Server Anywhere クライアントの同期ユーティリティの強化** *dbmlsync* ユーティリティの機能が、一部強化されました。

- `-mp` と `-mn` オプションを使用すると、Mobile Link パスワードの指定または変更ができます。
- `-n` オプションを繰り返し指定すると、複数の同期定義を同期化できます。
- `-v` オプションで、より便利な情報を得ることができるようになりました。同期定義内に設定されているオプションがその例です。
- `-r` オプションが拡張され、クライアントと統合データベース内に記録された進行状況インジケータが一致しない場合のアップロードをさらに柔軟にできるようになりました。
- `-x` オプションで、トランザクション・ログの名前を変更し、再起動できます。このオプションは、クライアント側のバックアップが不要になるように、クライアントのデータのバックアップとして統合データベースを使用する場合に役に立ちます。
- コマンド・ラインで接続パラメータを指定しない場合、接続パラメータと起動オプションを指定できるダイアログを *dbmlsync* が表示します。
- *dbmlsync* ウィンドウでは、同期の進行状況が表示され、同期をキャンセルできます。

詳細については、『Mobile Link クライアント』> 「Mobile Link 同期クライアント」を参照してください。

- **Mobile Link 同期サーバの新しいオプション** Mobile Link 同期サーバでは、オプションが追加されました。

詳細については、『Mobile Link 管理ガイド』> 「Mobile Link 同期サーバのオプション」を参照してください。

- **新しいスクリプト・イベント** ODBC Driver Manager で発生するエラーを処理し報告するための新しいスクリプトが追加されました。これによって、同期テクニックを設計するときの柔軟性が増加します。

詳細については、次の項目を参照してください。

- 『Mobile Link 管理ガイド』> 「handle_odbc_error 接続イベント」
- 『Mobile Link 管理ガイド』> 「prepare_for_download 接続イベント」
- 『Mobile Link 管理ガイド』> 「report_odbc_error 接続イベント」
- **dbmlsync 機能へのインタフェース** プログラミング言語 C を使用している開発者は、*dbmlsync* ユーティリティの機能をアプリケーションに追加できます。

詳細については、『Mobile Link クライアント』> 「アプリケーションからの同期の開始」を参照してください。

SQL Remote の新機能

SQL Remote バージョン 7.0.1 に、次の新機能が追加されています。

- **Novell NetWare 上のメッセージ・リンクの追加** Novell NetWare で FTP と SMTP/POP メッセージ・リンクを使用できるようになりました。
- **冗長モードの強化** Message Agent の冗長モードで、ユーザ ID とアスタリスクに置換されたパスワードを含むすべての接続情報が出力されるようになりました。

Ultra Light の新機能

Ultra Light 7.0.1 には、次の新機能が導入されています。

- **Palm Computing Platform 用の新しい同期ストリーム** 今回のリリースでは、現行の **ULPalmDBStream** 同期ストリームに加え、新しい同期ストリームを Palm Computing Platform に使用できます。この新しいストリームは **ULConduitStream** と呼ばれ、ほとんどの状況で、HotSync 同期のパフォーマンスを大幅に向上させることができます。

この機能は使用されなくなりました

バージョン 8.0.0 から導入された新しい conduit ベースの同期ストリームが、**ULPalmDBStream** と **ULConduitStream** の両方に取って代わります。

- **同期のモニタとキャンセル** 同期状況を表示し、同期をキャンセルできる機能を Ultra Light アプリケーションに構築できます。
- **Mobile Link でのユーザ認証** Mobile Link 同期独自のユーザ認証スキームが使用されるようになりました。このスキームを利用するために、Ultra Light 同期パラメータにパスワード用のフィールドとメソッドが追加されました。

詳細については、『Mobile Link クライアント』> 「Ultra Light 同期クライアントのネットワーク・プロトコルのオプション」を参照してください。

- **セキュリティ機能を備えた同期のための新しいプラットフォーム** Hitachi SH4 チップを搭載した Windows CE、Intel x86 チップまたは Windows VxSim エミュレータを搭載した VxWorks などの広範囲のターゲット・プラットフォームから、トランスポート・レイヤのセキュリティ機能を同期に使用できるようになりました。

詳細については、次の項目を参照してください。

- 『SQL Anywhere Studio の紹介』> 「SQL Anywhere Studio がサポートしているプラットフォーム」
- 『Mobile Link クライアント』> 「Ultra Light 同期クライアントのネットワーク・プロトコルのオプション」
- 『Ultra Light C/C++ ユーザーズ・ガイド』> 「Windows CE での同期」

バージョン 9 における VxWorks の非サポート

バージョン 9 では VxWorks プラットフォームはいっさいサポートされません。

- **非同期テーブル** Ultra Light データベースに含めるものの同期しないテーブルをリファレンス・データベースの中に入れることができます。これらのテーブルは、同期以外はリモート・データベース内の他のテーブルと同じように使用できます。

詳細については、『Mobile Link クライアント』>「Ultra Light データベースへの非同期テーブルの組み込み」を参照してください。

- **Windows CE エミュレータのサポートの強化** Ultra Light アプリケーションを Windows CE x86 エミュレータで実行できるようになりました。
- **同期の最適化** クライアント・アプリケーションでは、同期にアップロードだけを含み、ダウンロード・フェーズを行わないように指定できます。アップロードだけが必要な場合、特に低速な通信リンクを使用する場合は、このオプションによって同期にかかる時間を短縮できます。

詳細については、『Mobile Link クライアント』>「Ultra Light 同期クライアントのネットワーク・プロトコルのオプション」を参照してください。

- **HTTP バージョンの自動検出** Mobile Link 同期サーバは、各クライアントが使用している HTTP バージョンを検出してそれを使用するようになりました。この機能により、Mobile Link 同期サーバの `-x` オプションで使用されていた **version** パラメータが不要になります。

Mobile Link 同期サーバ・コマンド・ラインの詳細については、『Mobile Link 管理ガイド』>「Mobile Link 同期サーバのオプション」を参照してください。

- **クライアント・ポートの指定** 同期時にクライアントが使用するポートの範囲を、クライアント側で指定できます。この機能は、ファイアウォールの内側にあるクライアントからファイアウォールの外側にある Mobile Link 同期サーバに対して同期するときに便利です。

詳細については、『Mobile Link クライアント』>「Ultra Light 同期クライアントのネットワーク・プロトコルのオプション」を参照してください。

第 11 章

バージョン 7.0.0 の新機能

この章の内容

この章では、Adaptive Server Anywhere バージョン 7.0.0 で導入された新機能と変更された動作の概要について説明します。

バージョン 7.0.0 の新機能

マニュアルの基本フォーマットは HTML ヘルプです。HTML ヘルプのホームページからは、新機能、Sybase への問い合わせ方法、その他、本リリースの導入時に必要な情報に簡単にアクセスできます。

使用するマシンに Internet Explorer 4.0 も HTML ヘルプもインストールされていない場合は、HTML ヘルプではなく Windows ヘルプをインストールして使用してください。Windows ヘルプには HTML ヘルプのホームページは表示されませんが、その他の内容は同じです。

Windows ヘルプを使用する場合は、『*Adaptive Server Anywhere 入門*』の第 1 章に Adaptive Server Anywhere の新機能が記載されています。また、『*Ultra Light 開発者用ガイド*』と『*レプリケーションおよび同期ガイド*』の第 1 章に、それぞれのテクノロジーの新機能に関する情報が記載されています。

Adaptive Server Anywhere の新機能

この項では、Adaptive Server Anywhere バージョン 7.0 の新機能について説明します。新機能について主要なものからそうでないものまですべてを示し、各機能の詳細についてのマニュアル内の参照先も記述しています。

このマニュアルの製本版を使用していて、SQL Anywhere Studio のマニュアル・セット全巻をお持ちでない場合、各機能の詳細についてはオンライン・マニュアルを参照してください。オンライン・マニュアルで情報を検索するには、目次を開き、参照したい項目を入力します。

管理機能と操作性の強化

- ・ **データベースのタスク・スケジューリングとイベント処理** データベースに操作のスケジュールを追加できるようになりました。この機能は、自動バックアップ、サマリ・テーブルに入力する定期レポートなどのタスクを実行するのに便利です。

データベース・サーバに対して、特定のイベントが発生したときにイベント・ハンドラを起動するように指示できます。イベントにはデータベース・ファイルまたはトランザクション・ログ・ファイルを格納するドライブのディスク領域の閾値超過、接続試行の失敗などが含まれます。

イベント・ハンドラの作成と変更には Sybase Central を使用でき、デバッグには Adaptive Server Anywhere デバッガを使用できます。

詳細については、『ASA データベース管理ガイド』>「スケジュールとイベントの使用によるタスクの自動化」と『ASA SQL リファレンス・マニュアル』>「CREATE EVENT 文」を参照してください。

- **更新された Sybase Central** Sybase Central が書き直され、重要な新機能が追加されました。特に重要なのは、Windows オペレーティング・システムだけではなく、サポートされているすべてのプラットフォームから Sybase Central を使用できるようになったことです。
- **更新された Interactive SQL** Interactive SQL [dbisql] ユーティリティが強化され、サポートされているすべてのプラットフォームからウィンドウ・アプリケーションとして使用できるようになりました。
- **新しい検証機能** データベースの追加検証機能として、VALIDATE INDEX 文が新しく加わり、VALIDATE TABLE 文が強化されました。この文は、検証 [dbvalid] ユーティリティと sa_validate システム・プロシージャの両方で呼び出すことができ、どちらの場合でも新しい検証機能を使用できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「VALIDATE INDEX 文」と『ASA SQL リファレンス・マニュアル』>「VALIDATE TABLE 文」を参照してください。

- **ロックのトラブルシューティング** 新しいシステム・プロシージャ sa_locks を使うと、データベースのロック情報を通知できます。ロック状態が検出されると、関連する接続プロセス情報を AppInfo 接続プロパティを使って表示できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「sa_locks システム・プロシージャ」と『ASA データベース管理ガイド』>「AppInfo 接続パラメータ [APP]」を参照してください。

- **結果セットのアンロード** 新しい UNLOAD SQL 文を使用すると、クエリ結果セットをカンマ区切りのテキスト・ファイルにアンロードできます。

詳細については、『ASA SQL リファレンス・マニュアル』>「UNLOAD 文」を参照してください。

- **データベースのバックアップ・コピーの検証** WAIT BEFORE START 句を使ってデータベースのバックアップを作成すると、読み取り専用モードで開き、検証ができるバックアップ・コピーが作成されます。

詳細については、『ASA SQL リファレンス・マニュアル』>「BACKUP 文」を参照してください。

- **デフォルトのグローバル・オートインクリメント** この機能を使用すると、SQL Remote レプリケーション環境にあるすべてのデータベースにユニークな整数キーを簡単に生成できます。

分散コンピューティング・アーキテクチャとの統合

- **分散トランザクションと 3 層コンピューティング** 分散トランザクションには、単一のトランザクションで複数のサーバを使用する操作が含まれます。トランザクション・サーバは、分散トランザクションのコミットとロールバック動作を制御します。

Adaptive Server Anywhere のこのリリースは、DTC (Microsoft Distributed Transaction Coordinator) で管理される分散トランザクションにサーバとして追加できます。Sybase Enterprise Application Server や Microsoft Transaction Server などの製品は、DTC をトランザクションの調整に使用できるので、DTC をサポートすることにより、Adaptive Server Anywhere をこれらの製品を使った 3 層コンピューティングに使用できます。

詳細については、『ASA プログラミング・ガイド』>「3 層コンピューティングと分散トランザクション」を参照してください。

COM の統合

- **OLE DB プロバイダ** OLE DB は Microsoft が提供するデータ・アクセス・モデルです。OLE DB は、Component Object Model (COM) インタフェースを使用します。ODBC と違って、OLE DB は、データ・ソースが SQL クエリ・プロセッサを使用することを仮定していません。これまでも Microsoft が提供する OLE DB/ODBC ブリッジを使用し、OLE DB 経由で Adaptive

Server Anywhere にアクセスすることは可能でしたが、今回のリリースから Adaptive Server Anywhere に OLE DB プロバイダが付属するようになりました。このプロバイダには、以下のような利点があります。

- OLE DB は、今後予定されている Windows CE 版への主要データ・アクセス・ポイントとなります。
- カーソルによる更新など、OLE DB/ODBC ブリッジを使用している場合には利用できない機能がいくつかあります。
- Adaptive Server Anywhere の OLE DB プロバイダを使用する場合、配備に ODBC は必要ありません。

詳細については、『ASA プログラミング・ガイド』> 「OLE DB と ADO プログラミング・インタフェース」を参照してください。

コネクティビティの強化

- **Java コネクティビティの機能の向上** jConnect を使用して Java アプリケーションから Adaptive Server Anywhere に接続する場合、かつては ODBC と Embedded SQL アプリケーションでのみ使用可能だったデータベース・サーバの自動スタート、プロトコル・オプションを使ったネットワーク通信の詳細な制御などのさまざまな機能を使用できるようになりました。
- **TCP/IP コネクティビティ** TCP/IP を使ったクライアント／サーバ接続の確立が簡単になりました。クライアントは、サーバがデフォルトのポート番号 (2638) 以外のポートで実行していても、接続を試行する際にポート番号を指定する必要がなくなりました。データベース・サーバを起動したとき、デフォルトのポート番号がすでに使用されていたら、サーバはオペレーティング・システムから未使用のポートを取得します。

ファイアウォールを通過する接続を試み (UseUDP=NO を使用)、データベース・サーバがポート 2638 を使用していない場合は、今までどおりポート番号を指定する必要があります。この方法の詳細については、『ASA データベース管理ガイド』> 「ファイアウォール経由の接続」を参照してください。

サーバ検索 [dblocate] ユーティリティによって、ネットワーク上で TCP/IP を実行するすべての Adaptive Server Anywhere データベース・サーバが表示されます。詳細については、『ASA データベース管理ガイド』> 「サーバ検索ユーティリティ」を参照してください。

- **SPX コネクティビティ** データベースへの接続に SPX プロトコルを使用できます。この機能は、IPX/SPX をプライマリ・ネットワーク・プロトコルとして使用する Novell NetWare 環境では特に役に立ちます。IPX よりも SPX の使用をおすすめします。

クライアント側の SPX の詳細については、『ASA データベース管理ガイド』> 「CommLinks 接続パラメータ [LINKS]」を参照してください。サーバ側の SPX の詳細については、『ASA データベース管理ガイド』> 「-x サーバ・オプション」を参照してください。SPX で使用するネットワーク・プロトコル・オプションの詳細については、『ASA データベース管理ガイド』> 「ネットワーク・プロトコル・オプション」を参照してください。

パフォーマンスの強化

- **動的キャッシュ・サイズ決定** Windows NT と UNIX では、データベース・サーバの負荷とその他のシステム・メモリへの要求に応じて、データベース・サーバのキャッシュ・サイズが増減します。この機能を使用すると、さまざまな状況で明示的にキャッシュ・サイズを選択する必要がなくなり、パフォーマンスが向上します。Windows 95/98 にもキャッシュのサイズ変更機能が実装されていますが、基本機能に限られます。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「パフォーマンス向上へのキャッシュの使用」を参照してください。

- **インデックス処理の強化** インデックスに格納される情報量である「ハッシュ・サイズ」を、より柔軟に制御できるようになり、インデックスの選択性が高くなりました。また、プライマリ・キー・インデックスと外部キー・インデックスのアーキテクチャが変更になりました。

複数カラムのインデックス付け、または多くのローで最初の文字セットまたは数字セットが同じであるカラムのインデックス付けについては、ハッシュ・サイズを制御することで、インデックスの選択性が高くなり、パフォーマンスが向上します。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「インデックスの使用」、『ASA SQL リファレンス・マニュアル』> 「CREATE INDEX 文」、および 『ASA SQL リファレンス・マニュアル』> 「CREATE TABLE 文」を参照してください。

インデックス・レベル数を確認する方法については、『ASA SQL リファレンス・マニュアル』> 「sa_index_levels システム・プロシージャ」を参照してください。

これまでのリリースでは、プライマリ・キーと外部キーには、すべてのプライマリ・キー値とそれに関連するすべての外部キー値を示す単一のインデックスが自動的に関連付けられました。このアーキテクチャは、状況によってはパフォーマンスが低下することがありました。新しいインデックス編成では、これらのインデックスを分離するので、状況によってパフォーマンスが向上します。

キー・インデックスの詳細については、『ASA SQL ユーザーズ・ガイド』> 「キーを使ったクエリのパフォーマンス改善」を参照してください。

可変ハッシュ・サイズのインデックスと個別キー・インデックス機能を活用するには、データベースをアンロードし、再ロードする必要があります。アップグレード [dbupgrad] ユーティリティの実行では不十分です。

- **文字列拡張用の個別記憶領域** 255 文字を超える値の物理記憶領域が再編成されました。1 つのテーブルに対して割り付けられるページが、2 つのセットに分割されるようになりました。最初のセットには、ローだけが格納されます。ローのカラム値に 255 文字を超える文字列が含まれている場合は、文字列のプレフィックス (255 文字まで) と文字列拡張に対する参照だけがローに格納されます。255 文字を超える文字列では、文字列拡張はテーブル・ページの 2 番目のセットに割り付けられます。この変更により、長い値を格納するテーブル・スキャンを要求するクエリのパフォーマンスが向上します。これは、テーブルの逐次スキャンで、最初のセットに含まれているページだけをスキャンすれば済むからです。

この機能を使用するには、データベースを一度アンロードして、再ロードする必要があります。

- **新しいデータベースのページ・サイズ** 1 K、2 K、4 K のページ・サイズに加えて、8 K、16 K、または 32 K のページ・サイズを持つデータベースを作成できるようになりました。

ページ・サイズを大きくすると、特にデータベースのサイズが大きい場合など、状況によってパフォーマンスが向上する場合があります。ただし、ページ・サイズを大きくすると、追加メモリが必要になるので、対費用効果を考えて実行してください。

詳細については、『ASA データベース管理ガイド』> 「dbinit コマンド・ライン・ユーティリティを使用したデータベースの作成」と『ASA SQL リファレンス・マニュアル』> 「CREATE DATABASE 文」を参照してください。

テーブルごとのインデックス数と、それがページ・サイズによってどう変わるかについては、『ASA データベース管理ガイド』> 「サイズと数の制限」を参照してください。

- **オプティマイザのチューニング** OPTIMIZATION_GOAL オプションを使うと、オプティマイザに対して、クエリの最初のローを返すまでの時間、またはすべてのローを返すのに要する総時間を最適化するように指示できます。デフォルトでは、最初のローを最適化します。PowerBuilder DataWindow などの完全な結果セットを要求するアプリケーションを使用する場合は、このオプションの設定を変更してください。

詳細については、『ASA データベース管理ガイド』> 「OPTIMIZATION_GOAL オプション [データベース]」を参照してください。

- **オプティマイザの強化** オプティマイザがさらに強化され、内部テンポラリ・テーブルを使用し、プライマリ・キーと外部キーのインデックスを使用するクエリのパフォーマンスを最適化できるようになりました。これらの強化された機能を使用するのにユーザによるアクションは必要ありません。

その他の機能強化

- **多数のユーザとその他の識別子** システム・テーブルでデータベース・オブジェクトを区別する多くの識別子が、SMALLINT から UNSIGNED INTEGER に変更されました。この変更により、絶対制限値に違反しないでデータベースに格納できるオブジェクトの数が増えました。

- **イメージとドキュメントの挿入とエクスポート** 2つの新しいシステム外部関数を使って、ファイルの内容の読み込みと書き込みができます。これらの関数は、Interactive SQL などの環境からテーブルに、イメージやドキュメントなどを直接挿入できます。

詳細については、『ASA SQL ユーザーズ・ガイド』>「ドキュメントとイメージの挿入」、『ASA SQL リファレンス・マニュアル』>「xp_read_file システム・プロシージャ」、および『ASA SQL リファレンス・マニュアル』>「xp_write_file システム・プロシージャ」を参照してください。

- **外部関数用の新しいインタフェース** 外部ライブラリを参照するストアド・プロシージャとユーザ定義の関数用のインタフェースが新しくなりました。新しいインタフェースでは、さまざまなオペレーティング・システム (UNIX を含む) とさまざまなデータ型を使用でき、255 バイトまでのデータを返すという制限が取り除かれ、NULL が引数の有効な値として認識されます。これまでのインタフェースもサポートしていますが、新規の開発環境では使用しないでください。

詳細については、『ASA SQL ユーザーズ・ガイド』>「外部呼び出しを使ったプロシージャと関数の作成」を参照してください。

- **START DATABASE、STOP DATABASE、STOP ENGINE 文** これらの文は、これまで Interactive SQL からのみ使用可能でした。現在はどのアプリケーションからでも使用できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「START DATABASE 文 [Interactive SQL]」、『ASA SQL リファレンス・マニュアル』>「STOP DATABASE 文」、および『ASA SQL リファレンス・マニュアル』>「STOP ENGINE 文」を参照してください。

- **更新と削除における FIRST 句と TOP 句** FIRST 句と TOP 句は、WHERE 句の条件を満たす最初の 1 つ以上のロー・セットのみを更新または削除するために使用できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「DELETE 文」と『ASA SQL リファレンス・マニュアル』>「UPDATE 文」を参照してください。

- **テーブルの明示的ロック** LOCK TABLE 文を使用すると、現在の独立性レベルに関わりなく、テーブル・レベルの同時実行性を直接制御できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「LOCK TABLE 文」を参照してください。

- **Transact-SQL 外部ジョイン式** WHERE 句のジョイン演算子 (*と=*) を使用すると、Transact-SQL 構文を使用したいユーザが外部ジョインを指定できるようになります。これまでのリリースでは、そのようなジョインで使用できるのはカラム名だけでした。現在では、ジョイン演算子の双方が同じテーブルを参照するかぎり、どの式でも使用できます。たとえば、次のクエリも現在は使用可能です。

```
select *  
  from customer, sales_order  
 where substr( customer.id, 1, 1 ) *=  
        substr( sales_order.cust_id, 1, 1)
```

- **変数を参照できるストアド・プロシージャのカーソル** ストアド・プロシージャとユーザ定義の関数では、次の構文を使って変数にカーソルを宣言できます。

DECLARE *cursor-name* **CURSOR USING** *variable-name*

ここで、*variable-name* は、カーソルの SELECT 文を含む文字列変数です。

詳細については、『ASA SQL リファレンス・マニュアル』>「DECLARE CURSOR 文 [ESQL] [SP]」を参照してください。

- **データベース・サーバの追加プロパティ** 次のプロパティが追加されました。
 - **PageSize** データベース・サーバは、起動時から終了するまで、同じページ・サイズを使用します。このページ・サイズは、データベース・サーバによってマウントできる最大のページ・サイズです。このページ・サイズは、PageSize サーバ・レベル・プロパティ関数を使用してチェックできるようになりました。

```
select property( 'PageSize' )
```


- **AppInfo** この関数を使用すると、クライアント・アプリケーションの識別情報を取得できます。これは接続プロパティです。

```
select connection_property( 'AppInfo' )
```

詳細については、『ASA データベース管理ガイド』> 「AppInfo 接続パラメータ [APP]」を参照してください。

- **IsRuntimeServer** この関数は、データベース・サーバが機能が制限されたデスクトップ・ランタイム・パーソナル・データベース・サーバである場合は、YES を返します。それ以外の場合は、NO を返します。

- **ログ・トランケーション・ポイント** レプリケーション固有のログ・オフセットに対するプロパティが追加されました。プロパティ **LMTTrunc**、**RemoteTrunc**、**SyncTrunc** は、それぞれ Replication Agent、SQL Remote、Mobile Link **dbmlsync** レプリケーション用の確認された最小のログ・オフセットを返します。これらのオフセットは、トランザクション・ログをトランケートする場所を示すので、トランケーション・ポイントとも呼ばれます。プロパティ **CurrentRedoPos** は、ログ・ファイルの現在のオフセットを返します。そこから次のデータベース操作のログが作成されます。

すべてのプロパティ関数のリストとそれらにアクセスする方法については、『ASA データベース管理ガイド』> 「データベース・プロパティ」を参照してください。

- **コミット前の参照整合性チェック** 新しいシステム・プロシージャ (**sa_check_commit**) を使用すると、データベースへの変更をコミットする前に参照整合性の不一致をチェックできます。

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

- **SQL 関数の強化** 以下の関数が追加または強化されました。
- **REPLACE 関数** この新しい関数は、すべての部分文字列を別の部分文字列に置き換えます。

詳細については、『ASA SQL リファレンス・マニュアル』>「REPLACE 関数 [文字列]」を参照してください。

- **LIST 関数の強化** LIST 関数がオプションで 2 番目の値を受け入れられるようになりました。これは、リスト項目を区切るデリミタ文字列です。

詳細については、『ASA SQL リファレンス・マニュアル』>「LIST 関数 [集合]」を参照してください。

- **出力リダイレクションの変更** Interactive SQL の出力リダイレクション機能が拡張され、3 つの新しい Interactive SQL 文と、[ファイル] メニューに [エクスポート] オプションが加わりました。

OUTPUT TO 文を使って、[結果] ウィンドウ枠の内容を新しいファイルにリダイレクトできるようになりました。APPEND 句を加えると、内容が既存のファイルの最後に追加され、また VERBOSE 句を加えると、出力に [メッセージ] ウィンドウ枠の内容が入ります。

これまでのバージョンでは、Interactive SQL で出力をリダイレクトできるのは、記号 >#、>>#、>&、>>& を使用した場合にかぎられました。これらの記号は現在でも使用できますが、新しい Interactive SQL 文を使った方が、出力をより詳細に指定でき、読みやすいコードが提供されています。

詳細については、『Adaptive Server Anywhere ユーザーズ・ガイド』の『ASA SQL ユーザーズ・ガイド』>「クエリ結果のエクスポート」を参照してください。

- **Embedded SQL の強化** 指定した現在の接続文字列でデータベース・サーバにアクセスできるかどうかをテストするための新しい関数 `db_string_ping_server` が導入されました。

詳細については、『ASA プログラミング・ガイド』>「db_string_ping_server 関数」を参照してください。

- **新しい LOAD TABLE/UNLOAD TABLE フォーマット**

UNLOAD TABLE 文にデータを BCP フォーマットで出力できる新しいフォーマットが追加され、LOAD TABLE 文に Adaptive Server Enterprise が生成した blob を含む BCP 出力ファイルをインポートするための新しいフォーマットが追加されました。

詳細については、『ASA SQL リファレンス・マニュアル』>「LOAD TABLE 文」または『ASA SQL リファレンス・マニュアル』>「UNLOAD TABLE 文」を参照してください。

- **デフォルトの最終タイムスタンプ** 新しいグローバル変数

@@dbts は、DEFAULT TIMESTAMP を使ってカラム用に生成された最終値を表す TIMESTAMP 値を返します。

詳細については、『ASA SQL リファレンス・マニュアル』>「グローバル変数」を参照してください。

- **トラブルシューティングの強化** データベース・サーバの起動時、-zr オプションを使用すると、サーバが実行する操作のログをファイルに記録できます。sa_server_option プロシージャを使用すると、サーバの実行中に同じ動作を制御できます。

詳細については、『ASA SQL リファレンス・マニュアル』>「sa_server_option システム・プロシージャ」と『ASA データベース管理ガイド』>「-zr サーバ・オプション」を参照してください。

- **NetWare でのアーカイブのバックアップ** アーカイブのバックアップ・フォーマットが、NetWare でもサポートされるようになりました。テープへのアーカイブのバックアップには NetWare 5 が必要です。

詳細については、『ASA SQL リファレンス・マニュアル』>「BACKUP 文」を参照してください。

- **dbtran の追加フィルタ** コマンド・バージョンのログ変換 [dbtran] ユーティリティを使用して、出力をさらにフィルタできるようになりました。

詳細については、『ASA データベース管理ガイド』>「ログ変換ユーティリティのオプション」を参照してください。

- **テーブル・トランケーションの高速化** バージョン 7.0 データベースの外部キー付きテーブルに対する TRUNCATE TABLE 文の実行がより高速になりました。
- **イベント・ログ・メッセージを非表示にする** データベース・サーバを Windows NT サービスとして実行する場合、レジストリ・エントリを変更すれば、イベント・ログ・メッセージを非表示にできます。

詳細については、『ASA データベース管理ガイド』> 「Windows イベント・ログ・メッセージを出力しない」を参照してください。

SQL Remote の新機能

SQL Remote バージョン 7.0 には、次の新機能が追加されています。

- **グローバルに固有のプライマリ・キーを使用する** Adaptive Server Anywhere データベースの DEFAULT GLOBAL AUTOINCREMENT カラムを各データベースの GLOBAL_DATABASE_ID オプション設定とともに使用すると、複数の Adaptive Server Anywhere データベースを対象とする SQL Remote 全体に対して固有のプライマリ・キーを使用できます。この方法は、一部にユーザ設定が必要となるプライマリ・キー・プールよりも便利です。

詳細については、『SQL Remote ユーザーズ・ガイド』> 「グローバル・オートインクリメント・デフォルト・カラム値の使用」を参照してください。

- **dbxtract の内部アンロード** 抽出 [dbxtract] ユーティリティは、処理が遅い OUTPUT 文の代わりに、Adaptive Server Anywhere バージョン 7.0 にデフォルトとして組み込まれた UNLOAD 文を使用するようになりました。内部 (サーバ側) と外部 (クライアント側) のアンロードとロードの組み合わせを選択するためのオプションが組み込まれました。

オプションの完全なリストについては、『SQL Remote ユーザーズ・ガイド』> 「抽出ユーティリティ」を参照してください。

Mobile Link と Ultra Light の新機能

バージョン 6.0.3 以降のソフトウェアへの変更点と追加点を次に示します。

- **Adaptive Server Anywhere クライアント** Mobile Link テクノロジーが、Ultra Light アプリケーションに加え、Adaptive Server Anywhere もクライアントとしてサポートするようになりました。

詳細については、『Mobile Link クライアント』>「Adaptive Server Anywhere クライアント」を参照してください。

- **Adaptive Server Anywhere クライアント・データベースを作成する *mlxtract*** *mlxtract* は、Adaptive Server Anywhere データベースを、Mobile Link クライアントとしての使用に適するように作成します。このとき、Adaptive Server Anywhere リファレンス・データベースをテンプレートとして使用します。
- **同期スクリプト・バージョン** 同期スクリプトは、各スクリプトにスクリプト・バージョン名を割り当てればグループ化できるようになりました。この機能を使用すると、Mobile Link 同期サーバは、何種類ものアプリケーションを同期しているときや、同じアプリケーションでもバージョンが違うものを同期しているときも、それぞれ別々に応答できます。
- **新しいデータ型** Mobile Link テクノロジーを使用して LONG BINARY と LONG VARCHAR データ型をレプリケートできるようになりました。
- **新しい HotSync conduit** 新しい HotSync conduit を使用すると、中央の Mobile Link 同期サーバと HotSync を同期できます。Mobile Link 同期サーバが HotSync マネージャと同じマシンにある必要がなくなりました。
- **ScoutSync conduit** Palm Computing Platform 用の Ultra Light アプリケーションが、Riverbed Technologies の ScoutSync テクノロジーを使用して同期できるようになりました。

- **report_error スクリプト** この新しいスクリプトは、同期中のエラーをレポートするのに便利です。また、**report_error** スクリプトを使用すると、**handle_error** スクリプトの動作もより簡単にデバッグできます。**report_error** スクリプトのパラメータは、**handle_error** スクリプトと同じですが、例外として、最初のパラメータは、**handle_error** が返したアクション・コードになります。

バージョン 7.0.0 での動作の変更

この項では、使わなくなった機能、サポートしなくなった機能、これまでのバージョンとは異なる動作について説明します。

Adaptive Server Anywhere の動作の変更

廃止されサポートされなくなった機能

このリストには、サポートを終了した機能の中で既存のアプリケーションに影響のあるものがまとめられています。

- Windows 3.x と Windows CE 2.0 のサポート終了** Windows 3.1 と Windows 3.11 はサポートされなくなりました。Windows CE 2.0 はサポートされなくなりました。
- DDE プロトコルのサポート終了** DDE プロトコルは、16 ビットの Windows 3.x アプリケーションから同じマシン上にある Windows 95/98 データベース・サーバに接続するのに使用されてきましたが、旧バージョンで使用していた Windows 3.x アプリケーションは TCP/IP を使ってバージョン 7.0 データベース・サーバと通信できるので、現在は DDE プロトコルを使用する必要がありません。
- 旧式になった IPX プロトコル** IPX による通信は現在のリリースではまだサポートされていますが、SPX プロトコルを使用することをおすすめします。SPX のプロトコル・オプションは IPX と同じで、パフォーマンスが向上します。IPX については、将来のリリースではサポートが打ち切られる予定です。

デフォルトでは、データベース・サーバとクライアント・ソフトウェアの両方で IPX プロトコルを使用するには、`-x` オプションまたは **CommLinks** 接続パラメータを使用して明示的に指定する必要があります。デフォルトでは、SPX プロトコルが使用されます。

クライアント側から SPX を使用方法については、『ASA データベース管理ガイド』> 「CommLinks 接続パラメータ [LINKS]」を参照してください。サーバ側から SPX を使用方法については、『ASA データベース管理ガイド』> 「`-x` サーバ・オプション」を参照してください。

- **旧式になったネットワーク・プロトコル・オプション** Broadcast と CommAutoStop プロトコル・オプションは使用可能ですが、効果はありません。Adaptive Server Anywhere の将来のバージョンではサポートが打ち切られる予定です。
- **dbclient 互換性実行プログラム廃止** バージョン 6 では、dbcli6.exe ユーティリティを使い、簡単にバージョン 5 クライアント接続方式と互換性を持つことができました。バージョン 7 には、これと同等のユーティリティはありません。

動作の変更

このリストには、既存の機能の中で動作に変更があり、アプリケーションに影響のあるもの、または開発やデータベース管理に影響のあるものがまとめられています。

- **Interactive SQL の変更** Interactive SQL の新バージョンには、前バージョンからいくつか変更になった点があります。もともとこれは対話型のツールなので、説明が必要な変更はほとんどありません。

INPUT 文と OUTPUT 文で、サポートするフォーマットが変更になり、以下のフォーマットがサポートされるようになりました。

- **INPUT** ASCII、DBASE、DBASEII、DBASEIII、EXCEL、FIXED、FOXPRO、LOTUS
- **OUTPUT** ASCII、DBASE、DBASEII、DBASEIII、EXCEL、FIXED、FOXPRO、HTML、LOTUS、SQL
- **サーバのネーム・スペース変更** ネットワーク上で、TCP/IP を使用して同じ名前を持つ複数のデータベース・サーバを実行できなくなりました。以前は、ポートが違えば、同じ名前の複数のサーバを実行できました。
- **DELETE_OLD_LOGS が ON の場合、ミラーリングされたログは削除** これまで、古いトランザクション・ログのプライマリ・コピーが削除されても、それからミラーリングされたトランザクション・ログは削除されませんでした。
- **ODBC SQLDescribeCol の動作** @@identity フィールドでの SQLDescribeCol 呼び出しは、SQL_BIGINT を返すようになりました。これまでのバージョンでは、SQL_INTEGER が返されました。

- **更新の制約** 新しく ANSI_UPDATE_CONSTRAINTS オプションが追加されました。このオプションを CURSORS または STRICT に設定すると、更新対象が ANSI 標準に準拠するものに制約されます。このオプションを OFF に設定すると、従来の動作と同じで、更新可能な範囲が広がります。

詳細については、『ASA データベース管理ガイド』> 「ANSI_UPDATE_CONSTRAINTS オプション [互換性]」と『ASA SQL リファレンス・マニュアル』> 「UPDATE 文」を参照してください。

- **識別子の長さ制限** 長い識別子の取り扱い方がこれまでよりも整合性を持つようになりました。128 バイトを超える識別子は、名前を指定するデータベース・オブジェクトの型によって、許可されることもされないこともありましたが、現在は、128 バイトを超える識別子を定義しようとすると、すべてエラーが返されます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「識別子」を参照してください。

- **jConnect 接続** jConnect 経由で Adaptive Server Anywhere データベース・サーバの名前付きデータベースに接続するのに REMOTEPWD フィールドを使用する場合、本ソフトウェア付属の jConnect バージョン 4.2 以上では、別の形式でフィールドを割り当てる必要があります。

詳細については、『ASA プログラミング・ガイド』> 「サーバの URL の指定」を参照してください。

- **ユーザ定義のエラー** プロシージャとトリガ内で、99000 から 99999 までの範囲にユーザ定義のエラーを例外として複合文で宣言できます。これらのエラーは、SIGNAL 文を使用して処理します。

詳細については、『ASA SQL リファレンス・マニュアル』> 「BEGIN 文」と『ASA SQL リファレンス・マニュアル』> 「SIGNAL 文」を参照してください。

- **LOAD TABLE と UNLOAD TABLE のセキュリティ** データベース・サーバ・オプションが追加され、LOAD TABLE 文と UNLOAD TABLE 文を実行するのに必要なパーミッションを制御できるようになりました。

詳細については、『ASA データベース管理ガイド』> 「-gl サーバ・オプション」を参照してください。

- **トリガの @@identity** オートインクリメント・カラムを持つテーブル (T1) INSERT トリガがあり、そのトリガが同様にオートインクリメント・カラムを持つ別のテーブル (T2) への挿入を実行する場合、これまでは、挿入が完了した後、T1 に割り当てたオートインクリメント値の取得はできませんでした。その時点で、@@identity の値は、T2 に割り当てた値になりました。しかし、@@identity の動作が変更になり、値にアクセスできるようになりました。

トリガ内の @@identity の新しい動作については、『ASA SQL リファレンス・マニュアル』> 「@@identity グローバル変数」を参照してください。

- **Embedded SQL DECL_FIXCHAR** これまでのリリースでは、SQL プリプロセッサが DECL_FIXCHAR 型を配列に変換していました。たとえば、DECL_FIXCHAR(12) は、char name_fixchar[12] に変換されていました。

現在のリリースでは、SQL プリプロセッサが DECL_FIXCHAR 宣言を DECL_FIXCHAR(12) name_fixchar に変換し、参照フォームは name_fixchar.array[i] の形式になります。

SQL Remote の動作の変更

次の動作がバージョン 7.0 で変更されました。

- **dbxtract は内部アンロードを使用する** dbxtract はデフォルトで、OUTPUT 文を使用してクライアント側のデータをアンロードする代わりに、UNLOAD 文を使用してサーバ側のデータをアンロードします。従来のリリースの -i と -x のオプションの代わ

りに `-ii`、`--ix`、`-xi`、`-xx` オプションを使うと、使用する内部オペレーションと外部オペレーションの組み合わせを選択できます。

オプションの完全なリストについては、『SQL Remote ユーザーズ・ガイド』>「抽出ユーティリティ」を参照してください。

Mobile Link と Ultra Light の動作の変更

- **新しいテーブル名とスクリプト名** 同期スクリプトに関連する情報が含まれている統合データベースのテーブルに、新しい名前が付けました。以前は、こうしたテーブルの名前は、プレフィクス `ul_` で始まっていました。このプレフィクスが、`ml_` に変わりました。古い統合データベースは、バージョン 7.0 との互換性を持たせるためにアップグレードする必要があります。

同様に、テーブル・スクリプトの追加を容易にするストアド・プロシージャの名前が、`sp_table_script` から `ml_add_table_script` に変更されました。また、接続スクリプトの追加を容易にするストアド・プロシージャの名前が、`sp_connection_script` から `ml_add_connection_script` に変更されました。

DB2 上では、これらの名前は 18 文字にトランケートされます。

- **バージョン名を必要とする同期スクリプト** 同期スクリプトには、スクリプト・バージョン名の割り当てが必要になりました。スクリプト・バージョン名によって、Mobile Link 同期サーバは異なるクライアントを別々に処理できます。

第 12 章

バージョン 6.0.3 の新機能

この章の内容

この章では、Adaptive Server Anywhere 6.0.3 で導入された新しい機能と変更された動作の概要について説明します。

バージョン 6.0.3 の新機能

この項では、SQL Anywhere Studio バージョン 6.0.3 のコンポーネントに導入された新機能について説明します。

Adaptive Server Anywhere の新機能

Adaptive Server Anywhere バージョン 6.0.3 では、バグの修正に加え、ソフトウェアとマニュアルに新しい機能が組み込まれました。

- **ストアド・プロシージャと Java デバッガ** 従来のバージョンで提供された Java デバッガが、アップグレードされました。新しいバージョンのデバッガは、データベース内の Java クラスだけでなく、SQL ストアド・プロシージャとトリガもデバッグできるようになりました。

デバッガの使用方法については、『ASA SQL ユーザーズ・ガイド』> 「データベースでのデバッグ論理」を参照してください。

- **読み込み専用データベース** データベース・サーバを起動したときに、データベースを読み込み専用に指定できます。この機能により、CD-ROM などの読み込み専用メディアにデータベースを容易に展開できます。

読み込み専用データベースの場合、**ReadOnly** データベース・プロパティは ON を返し、読み込み専用モードで実行されていないデータベースの場合は OFF が返されます。

読み込み専用データベースの詳細については、『ASA データベース管理ガイド』> 「-r サーバ・オプション」を参照してください。

- **計算カラムの拡張** 計算カラムを、これまで以上に柔軟に使用できるようになりました。空でないテーブルに計算カラムを追加し、計算カラムと対応する式を変更できます。計算カラムはいくつかの条件下で計算し直され、常に信頼性の高い値を保持します。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「計算カラムの使用」と『ASA SQL ユーザーズ・ガイド』> 「計算カラムの挿入と更新」を参照してください。

構文の詳細については、『ASA SQL リファレンス・マニュアル』> 「ALTER TABLE 文」を参照してください。

- **ユーロ通貨のサポート** ユーロ通貨記号を含む照合が追加されました。これらは、1252LATIN1 と ISO9LATIN1 照合です。

詳細については、『ASA データベース管理ガイド』> 「1252LATIN1 照合」と『ASA データベース管理ガイド』> 「ISO9LATIN1 照合」を参照してください。

- **照合の追加** 852POL (OEM Code Page 852 ラテン文字 2、ポーランド語順)、1250POL (Windows Latin2 Code Page 1250、ポーランド語順)、1250Latin2 (Windows Latin2 Code Page 1250)、932JPN (日本語)、936ZHO (EUC_CHINA に類似した言語)、950TAI (EUC_TAIWAN に類似した言語) など、これまでになかった照合が、提供される照合のリストに追加されました。

完全なリストについては、『ASA データベース管理ガイド』> 「照合の選択」を参照してください。

- **新しい Windows CE プラットフォーム** SH4 と ARM プロセッサが、Windows CE 2.1x でサポートされるようになりました。

サポートされているデバイス、チップ、オペレーティング・システムのバージョンのリストについては、『SQL Anywhere Studio の紹介』> 「Windows オペレーティング・システムと NetWare オペレーティング・システム」を参照してください。

- **ALTER TABLE の拡張** ALTER TABLE 文が拡張され、カラムにデフォルト値の設定と削除を行う、SQL/92 と互換性のある句を提供できるようになりました。これらの句は、既存の MODIFY 句の代わりに使用できます。

```
ALTER column-name SET DEFAULT default-value
| ALTER column-name DROP DEFAULT
```

詳細については、『ASA SQL リファレンス・マニュアル』> 「ALTER TABLE 文」を参照してください。

- **LOAD TABLE の拡張** LOAD TABLE 文を使ってテーブルの指定したカラムをロードできるようになりました。また、再構築に関する問題を処理する新しい CHECK CONSTRAINTS オプションが追加されました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「LOAD TABLE 文」を参照してください。

- **ファイアウォール経由の容易な接続** ファイアウォールの内外の接続を容易にするプロトコル・オプションのセットが組み込まれました。

詳細については、『ASA データベース管理ガイド』> 「ファイアウォール経由の接続」を参照してください。

- **BACKUP 文の拡張** トランザクション・ログのバックアップ・コピーの名前を *YYMMDDnn.log* という形式のファイル名に変更できる MATCH キーワードが追加されました。このキーワードを使用すれば、データを上書きせずに同じ文を複数回実行できます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「BACKUP 文」を参照してください。

- **容易なアンロードと再ロード** アンロード [dbunload] ユーティリティが強化され (-ar オプション)、データベースがレプリケーションの対象であるかどうかに関わらず、使用できるデータベースのアンロードと再ロードを 1 回の操作でできるようになりました。

詳細については、『ASA データベース管理ガイド』> 「アンロード・ユーティリティのオプション」と『ASA SQL ユーザーズ・ガイド』> 「データベースの再構築」を参照してください。

- **テンポラリ・ファイルのロケーション** テンポラリ・ファイルのロケーションを判断するとき、データベース・サーバは新しい環境変数 ASTMP をチェックします。これにより、システムのテンポラリ・ディレクトリ以外のディレクトリにテンポラリ・ファイルを置くことができます。

詳細については、『ASA データベース管理ガイド』> 「ASTMP 環境変数」を参照してください。

- **新しいシステム・プロシージャ** 新しいシステム・プロシージャを使用すると、DBA ユーザはデータベース・サーバのオプション (sa_server_option) を無効にし、データベース・サーバ・キャッシュ (sa_flush_cache) をフラッシュできます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「システム・プロシージャとカタログ・ストアド・プロシージャ」を参照してください。

- **文字セット変換のチューニング** 新しい CharSet 接続パラメータを使用して、個々の接続に対し、文字セット変換で使用するアプリケーション・ロケールを制御できるようになりました。

詳細については、『ASA データベース管理ガイド』> 「CharSet 接続パラメータ [CS]」を参照してください。

- **再編成されたパフォーマンス・モニタの統計** Windows NT パフォーマンス・モニタで利用できる統計が、エリアに編成されました。いくつかの統計が追加され、利用頻度の少なかった統計は削除されました。

使用可能な統計のリストについては、『ASA SQL ユーザーズ・ガイド』> 「Windows パフォーマンス モニタからのデータベース統計値のモニタリング」を参照してください。

- **ユーティリティ・データベースのデータベース・プロパティ** ユーティリティ・データベースに対して、テーブル指定のない SELECT 文を実行できるようになりました。この機能は、主にデータベースと接続プロパティの検索に使用されます。

詳細については、『ASA データベース管理ガイド』> 「ユーティリティ・データベースの使用」を参照してください。

- **新しいデータベース・プロパティ** プロパティ関数を使用して次のプロパティを利用できます。

- **IsNetworkServer** ネットワーク・データベース・サーバに接続されると YES を返し、パーソナル・データベース・サーバに接続されると NO を返します。

詳細については、『ASA データベース管理ガイド』> 「サーバ・レベルのプロパティ」を参照してください。

- **DefaultCollation** 新しい **DefaultCollation** プロパティを使用して、データベースを作成するときに使用するデフォルト照合を検索できます。

詳細については、『ASA データベース管理ガイド』> 「デフォルトの照合の検索」と『ASA データベース管理ガイド』> 「サーバ・レベルのプロパティ」を参照してください。

- **MultiByteCharSet** **MultiByteCharSet** データベース・プロパティを使用すると、データベースがマルチバイト照合またはシングルバイト照合のどちらを使用しているかをチェックできます。

このプロパティの詳細については、『ASA データベース管理ガイド』> 「データベース・レベルのプロパティ」を参照してください。

- **内部 JDBC における JDBC 2.0 関数のサポート** 内部サーバ側 JDBC ドライバは、JDBC 2.0 インタフェースからの関数をサポートするようになりました。サーバ側の Java アプリケーションは、スクロール可能で更新可能な結果セットやバッチ更新などの機能を使用できるようになりました。これに伴い、Interactive SQL から Java メソッドによる結果セットにアクセスできるようになりました。

詳細については、『ASA プログラミング・ガイド』> 「データベース内の JDBC の機能」を参照してください。

- **Java クラスで main メソッドを使用する** SQL から Java クラスの main メソッドを実行できます。

詳細については、『ASA プログラミング・ガイド』> 「main メソッドの呼び出し」を参照してください。

- **ユーザ定義関数に Java クラスを使用する** SQL ユーザ定義関数に、Java メソッドを含めることができます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「CREATE FUNCTION 文」を参照してください。

- **Java メソッドを使用するストアド・プロシージャの拡張** Java メソッドに含まれるストアド・プロシージャに、OUT と INOUT パラメータを使用できます。

詳細については、『ASA プログラミング・ガイド』> 「Java からストアド・プロシージャを経由して値を返す」を参照してください。

- **データベース内のマルチスレッド Java クラス**
java.lang.thread パッケージへのサポートが追加されました。

詳細については、『ASA プログラミング・ガイド』> 「Java アプリケーションでのスレッドの使用」を参照してください。

- **Java からファイルへのアクセス** データベースのクラスからファイルへのアクセスを可能にするクラスを含め、java.io パッケージのすべてのクラスに対するサポートが追加されました。新しいオプションが導入されましたが、セキュリティの理由により、この機能は DBA だけが設定できるようになっています。

この機能は Windows NT でのみ提供されます。

詳細については、『ASA データベース管理ガイド』> 「JAVA_INPUT_OUTPUT オプション [データベース]」を参照してください。

- **CONVERT 関数の拡張** CONVERT 関数がサポートする日付と時刻のスタイルが拡張されました。

詳細については、『ASA SQL リファレンス・マニュアル』> 「CONVERT 関数 [データ型変換]」を参照してください。

- **データベース・サーバの起動ダイアログ** 32 ビットの Windows オペレーティング・システムでは、データベース・サーバを引数なしで開始すると、データベース・ファイルと追加のパラメータを指定するウィンドウが表示されます。

詳細については、『ASA データベース管理ガイド』> 「サーバの起動」を参照してください。

- **ログ変換 [dbtran] ユーティリティの強化** ログ変換 [dbtran] ユーティリティを使うと、トランザクション・ログ・オペレーションのフィルタリングによりオペレーションのサブセットを分離できます。

詳細については、『ASA データベース管理ガイド』> 「dbtran コマンド・ライン・ユーティリティを使用したトランザクション・ログの変換」を参照してください。

- **トランザクション・ログ [dblog] ユーティリティの強化** トランザクション・ログ [dblog] ユーティリティを使用して、オフセット情報を含めて追加のサマリ情報が表示されるようになりました。

詳細については、『ASA データベース管理ガイド』> 「dblog コマンド・ライン・ユーティリティを使用したログ・ファイルの管理」を参照してください。

- **プロセス生成 [dbspawn] ユーティリティの強化** プロセス生成 [dbspawn] ユーティリティに `-f` オプションが追加され、すでに実行中のサーバがあっても、サーバを強制的に起動できるようになりました。このオプションは、`db_start_engine Embedded SQL` 関数だけで使用され、`ForceStart` 接続パラメータを指定しません。

詳細については、『ASA データベース管理ガイド』> 「プロセス生成ユーティリティ」と『ASA プログラミング・ガイド』> 「db_start_engine 関数」を参照してください。

- **Replication Agent をデーモンとして実行する** UNIX オペレーティング・システムでは、`-ud` オプションを指定することによって、Replication Agent をデーモンとして実行できます。

詳細については、『ASA データベース管理ガイド』> 「Log Transfer Manager」を参照してください。

SQL Remote の新機能

SQL Remote バージョン 6.0.3 では、バグの修正に加え、次の新しい機能が追加されました。このリストには、Adaptive Server Anywhere の機能で特に SQL Remote に関係のあるものも含まれています。

- **UNIX による FTP と SMTP/POP のサポート** UNIX オペレーティング・システムでサポートされるメッセージ・システムの範囲が拡張され、FTP と SMTP/POP もサポートされるようになりました。

サポートされているオペレーティング・システムとメッセージ・システムのリストについては、『SQL Remote ユーザーズ・ガイド』> 「サポートされるプラットフォームとメッセージ・リンク」を参照してください。
- **データベースに格納されるメッセージ・リンクのオプション** 各メッセージ・システムで SQL Remote の動作を制御するメッセージ・リンク・パラメータは、レジストリではなく、データベースに格納されるようになりました。これによって、メッセージ・リンク・パラメータに関連する配備と管理が簡略化されました。

詳細については、『SQL Remote ユーザーズ・ガイド』> 「メッセージ・タイプ制御パラメータの設定」、『SQL Remote ユーザーズ・ガイド』> 「SET REMOTE OPTION 文」、および『SQL Remote ユーザーズ・ガイド』> 「sp_link_option プロシージャ」を参照してください。
- **日付と時間のレプリケーション・フォーマット** データベース・オプションを指定して、日付と時間をレプリケートするときに使用するフォーマットを SQL Remote に指示できるようになりました。使用するオプションは SR_time_format、SR_date_format、SR_timestamp_format です。

詳細については、『SQL Remote ユーザーズ・ガイド』> 「日付と時刻のレプリケーション」と『SQL Remote ユーザーズ・ガイド』> 「SQL Remote のオプション」を参照してください。

- **Message Agent と SQL Remote Open Server をデーモンとして実行する** UNIX オペレーティング・システムでは、`-ud` オプションを使用して、この 2 つのアプリケーションをデーモンとして実行できます。

詳細については、『SQL Remote ユーザーズ・ガイド』> 「Message Agent」と『SQL Remote ユーザーズ・ガイド』> 「SQL Remote Open Server」を参照してください。

- **Adaptive Server Anywhere データベースの容易なアンロードと再ロード** アンロード `[dbunload]` ユーティリティが強化され (`-ar` オプション)、データベースがレプリケーションの対象であるかどうかに関わらず、使用できるデータベースのアンロードと再ロードを 1 回の操作でできるようになりました。

詳細については、『ASA データベース管理ガイド』> 「アンロード・ユーティリティのオプション」と『ASA SQL ユーザーズ・ガイド』> 「データベースの再構築」を参照してください。

- **トランザクション・ログ `[dblog]` 出力の強化** トランザクション・ログ `[dblog]` ユーティリティを使用して、オフセット情報を含めて追加のサマリ情報が表示されるようになりました。

詳細については、『ASA データベース管理ガイド』> 「`dblog` コマンド・ライン・ユーティリティを使用したログ・ファイルの管理」を参照してください。

- **ログ変換 `[dbtran]` ユーティリティの強化** ログ変換 `[dbtran]` ユーティリティを使うと、トランザクション・ログ・オペレーションのフィルタリングによりオペレーションのサブセットを分離できます。この機能を使用できるのは、SQL Remote の管理者だけです。

詳細については、『ASA データベース管理ガイド』> 「`dbtran` コマンド・ライン・ユーティリティを使用したトランザクション・ログの変換」を参照してください。

Mobile Link と Ultra Light の新機能

バージョン 6.0.2 以降のソフトウェアへの変更点と追加点を次に示します。

- **新しいデータ型** real データ型と double データ型が完全にサポートされるようになりました。
- **文字セット変換** Mobile Link 同期サーバは、アップロードされたすべての文字をユニコードに変換し、ユニコード ODBC API を使用して統合データベースに渡すようになりました。逆に、ダウンロードされたすべての文字をユニコードから Ultra Light アプリケーションの文字セットに変換します。統合データベース・サーバでの文字セット変換が結果に影響を及ぼすがありますが、新しいシステムでは、複数のプラットフォームでより一貫した動作を行えます。

詳細については、『Mobile Link 管理ガイド』>「ODBC ドライバの文字セット変換の制御」を参照してください。

- **Windows NT サービスとして実行する Mobile Link 同期サーバ**
Mobile Link 同期サーバをサービスとして実行する場合は、Windows NT ワークステーションをログオフしても実行し続けるように設定できます。

詳細については、『Mobile Link 管理ガイド』>「現在のセッション以外での Mobile Link の実行」を参照してください。

- **DB2 設定スクリプトの提供** IBM DB2 を統合データベースとしてより簡単に使用できるようにするため、使用可能なセット・スクリプトに DB2 設定スクリプトが追加されました。

設定スクリプトのリストについては、『Mobile Link 管理ガイド』>「統合データベースの設定」を参照してください。

バージョン 6.0.3 での動作の変更

この項では、SQL Anywhere Studio バージョン 6.0.3 のコンポーネントに導入された動作の変更について説明します。

Adaptive Server Anywhere の動作の変更

- **追加カラムにデフォルト値を設定する** デフォルト値のあるカラムが追加された場合、その追加カラム全体にデフォルト値が設定されます。従来のリリースでは、カラムに NULL が設定されました。
- **参照整合性アクションのパーミッション** プライマリ・テーブルが変更されると、カスケード削除や更新などの参照整合性アクションがセカンダリ・テーブルで実行されます。参照整合性アクションは、システム・トリガを使って行われます。このトリガは、セカンダリ・テーブルの所有者のパーミッションを取得して実行されるようになりました。以前は、プライマリ・テーブルの所有者のパーミッションを取得してトリガが実行されていました。新しい動作では、カスケード・オペレーションを異なる所有者のテーブル間で実行するのに、追加のパーミッションを取得する必要はありません。
- **datediff、MONTHS、YEARS 関数** 2つの日付間の月数は、2つの日付の間にある月の第1日目の数として計算されます。たとえば、1月25日と2月2日の差異は1(ヶ月)に、1月1日と1月31日の差異は0(ヶ月)になります。年数は、2つの日付の間に存在する1月1日の数として計算されるようになりました。

これにより、新旧の関数で結果として得られる数字が1異なることがあります。この変更は、Adaptive Server Enterprise との互換性のために行われました。

より短い時間単位の場合は、DATEDIFF 関数に対するオーバーフロー値が使われるようになりました。従来のバージョンでは、制限を越えると間違った答がでました。

詳細については、『ASA SQL リファレンス・マニュアル』>「DATEDIFF 関数 [日付と時刻]」を参照してください。

- **デフォルトのページ・サイズ** データベースのデフォルト・ページ・サイズが 2048 バイトになりました。これは、多くのユーザにとって便利です。
- **デフォルト・データベース照合** データベースを作成するときに使うデフォルト照合が変更されました。このデフォルト照合は使用するオペレーティング・システムの設定によって異なります。

デフォルト照合の検索方法については、『ASA データベース管理ガイド』> 「デフォルトの照合の検索」を参照してください。

- **SQL プリプロセッサのデフォルト照合** 照合を明示的に指定しない場合、Embedded SQL プリプロセッサは、ロケール情報を使用してデフォルト照合を選択するようになりました。ロケール情報がない場合は、850LATIN1 を使用します。使用する照合はバナーの後にレポートされます。従来の動作では 850 を使用しました。

プリプロセッサの詳細については、『ASA プログラミング・ガイド』> 「SQL プリプロセッサ」を参照してください。

- **サーバ名の長さの強制適用** サーバ名は起動時にチェックされ、最長 40 文字にトランケートされます。NetBIOS では、16 文字にトランケートされます。クライアント側では、EngineName パラメータの値も 40 文字にトランケートされます。

詳細については、『ASA データベース管理ガイド』> 「EngineName 接続パラメータ [ENG]」を参照してください。

- **Agent 接続パラメータ** Agent 接続パラメータの動作が変更されました。このパラメータの意味がバージョン 5 からバージョン 6 で変更されたのは、*dbclient* 実行プログラムが不要になったためです。パラメータの意味が変更されて、バージョン 6 の環境でより便利になりました。

Agent 接続パラメータは、バージョン 8.0.1 からは使用されなくなりました。

SQL Remote の動作の変更

SQL Remote バージョン 6.0.3 では、次の動作が変更されました。

- **データベースに格納されるメッセージ・リンク・パラメータ** デフォルトでは、新しいバージョンのソフトウェアで最初に Message Agent を実行するときに、メッセージ・リンク・パラメータがデータベースに移動します。データベースの外部にある変更前のロケーションで、これらのパラメータを明示的に探すソフトウェアを使用している場合には、エラーが発生します。External_remote_options データベース・オプションを ON に設定すると、従来のバージョンどおりの動作となります。
- **パスワードの格納** メッセージ・リンクのパスワードを入力したときに、従来のバージョンのソフトウェアでは、パスワードが格納されませんでした。新しいバージョンではパラメータがデータベースに保管されるようになったため、保存されたパスワードはディスク上に格納されず、セキュリティが向上しました。パスワードはデフォルトで保存されるようになりました。Save_remote_passwords オプションを OFF に設定すると、従来のバージョンと同じ動作となります。

第 13 章

バージョン 6.0.2 の新機能

この章の内容

この章では、Adaptive Server Anywhere 6.0.2 で導入された新しい機能と変更された動作の概要について説明します。

バージョン 6.0.2 の新機能

この項では、SQL Anywhere Studio バージョン 6.0.2 のコンポーネントに導入された新機能について説明します。

Adaptive Server Anywhere の新機能

Adaptive Server Anywhere バージョン 6.0.2 では、バグの修正に加え、ソフトウェアとマニュアルに新しい機能が組み込まれました。

相互参照

製本版のマニュアルは、メンテナンス・リリースごとに更新されているわけではありません。この項の相互参照は製本版のマニュアルでは有効でない場合があります。最新の情報については、オンライン・マニュアル (英語版) を参照してください。

- **Ultra Light 展開オプション** このバージョンでは、PalmPilot や Windows CE などの小型デバイス用 Ultra Light データベースを開発できます。

詳細については、『*Ultra Light 開発者用ガイド*』を参照してください。

- **SQL 文のバックアップとリストア** BACKUP と RESTORE を SQL 文として追加することにより、SQL スクリプトを使用したサーバ側のバックアップとバックアップの自動化が可能になります。

BACKUP 文によって、テープへの直接バックアップが可能になります。

詳細については、『*ASA SQL リファレンス・マニュアル*』> 「BACKUP 文」を参照してください。

- **セキュリティ機能** 新しいセキュリティ機能が追加されました。
 - **監査** データベース管理者は、AUDITING オプションをオンにすることで、データベースで実行されたアクティビティを追跡できます。アクティビティのレコードはトラン

ザクション・ログに保持されます。監査をオンにすると、ログイン試行、すべてのイベントの正確なタイムスタンプ、すべてのパーミッション・チェック、DBA 権限を必要とするすべてのアクションを含むため、トランザクション・ログに保存されるデータ量が増加します。

詳細については、『SQL Anywhere Studio セキュリティ・ガイド』> 「データベース・アクティビティの監査」を参照してください。

- **パスワードの最小長** データベース管理者は、パスワードが簡単に解読されないように、パスワードの最小長を指定できます。

詳細については、『ASA データベース管理ガイド』> 「MIN_PASSWORD_LENGTH オプション [データベース]」を参照してください。

- **サーバの検出** 接続のトラブルシューティング用にユーティリティが提供されています。

詳細については、『ASA データベース管理ガイド』> 「Ping ユーティリティ」を参照してください。

- **jConnect 接続によるデータベースの開始** jConnect を介した Java アプリケーションからの接続を含む TDS データベース接続を実行すると、サーバ上でデータベースを起動できます。

詳細については、『ASA プログラミング・ガイド』> 「サーバの URL の指定」を参照してください。

- **ODBC 3.51** ODBC ドライバは ODBC 3.51 に更新されました。このバージョンの ODBC では Unicode アプリケーションもサポートしています。

詳細については、『ASA プログラミング・ガイド』> 「ODBC 準拠」を参照してください。

- **使用できる JOIN 構文の制御** これまでのリリースでは、あいまいなジョイン句を持つマルチテーブル・クエリが許可されていましたが、今回のリリースでは、オプションを設定してこのようなクエリを許可しないようにできます。

詳細については、『ASA データベース管理ガイド』>
「EXTENDED_JOIN_SYNTAX オプション [データベース]」を参
照してください。

- **管理ユーティリティの強化** 新しい機能を提供するオプション
が、管理ユーティリティに追加されました。
- **トランザクション・ログ [dbtran] ユーティリティ** 新しい
-a オプションを使用すると、*dbtran* は、個々のオペレー
ションを、その発生時にトランザクション・ログ・ファイ
ルに記録します。これによりトランザクション・ログの出
力が読みやすくなります。これは主に監査を目的として追
加されました。

詳細については、『ASA データベース管理ガイド』>
「dbtran コマンド・ライン・ユーティリティを使用したトラ
ンザクション・ログの変換」を参照してください。
- **トランザクション・ログ [dbtran] ユーティリティ** *dbtran*
は、ログ・ファイルではなく、稼働中のデータベース・
サーバに対して実行できます。この機能を使用するとトラ
ンザクション・ログに直接アクセスする必要がないので、
トランザクション・ログのセキュリティを強化するために
追加されました。

詳細については、『ASA データベース管理ガイド』>
「dbtran コマンド・ライン・ユーティリティを使用したトラ
ンザクション・ログの変換」を参照してください。
- **Log Transfer Manager [dbltm] ユーティリティのロギング**
メッセージのロギングをチューニングするための新しいオ
プションが追加されました。

詳細については、『ASA データベース管理ガイド』> 「Log
Transfer Manager」を参照してください。
- **新しい Log Transfer Manager [dbltm] ユーティリティのオ
プション** 新しいオプションとして、バックアップされた
トランザクションのみをレプリケートするオプション
(**backup_only**) と、すべてのデータのレプリケート完了と
ともに停止するオプション (**continuous**) が加わりました。

詳細については、『ASA データベース管理ガイド』> 「LTM 設定ファイル」を参照してください。

SQL Remote の新機能

SQL Remote バージョン 6.0.2 では、バグの修正に加え、次の新しい機能が追加されました。

- **パフォーマンスの強化** トランザクション・ログのスキャンとメッセージの送信を行うための Adaptive Server Anywhere Message Agent (*dbremote*) オペレーショナル・モデルの強化により、レプリケーションのターンアラウンド・タイムが大幅に短縮されました。

あるサイトにデータを入力してから、そのレプリケーションを別のサイトに入力するまでの最小遅延時間は、従来のバージョンではおよそ 10 分という制約がありました。新しいオペレーショナル・モデルでは、状況によって最小遅延時間を秒単位で設定することができます。

Message Agent のメッセージ送信処理は、継続モードで実行されている場合、必要なデータがコミットされるのを待っている間アクティブなトランザクション・ログの終わりに置かれ(「ホーバー」状態)、そのたびにトランザクション・ログをスキャンし直しません。これによってより頻繁なポーリングが可能になり、レプリケーションにかかる時間が大幅に削減されます。

詳細については、『SQL Remote ユーザーズ・ガイド』> 「Message Agent のパフォーマンスのチューニング」を参照してください。

- **SQL Remote メッセージのロギング** メッセージのロギングをチューニングするための新しいオプションが追加されました。

詳細については、『SQL Remote ユーザーズ・ガイド』> 「Message Agent」を参照してください。

バージョン 6.0.2 での動作の変更

この項では、SQL Anywhere Studio のコンポーネントでの動作の変更について説明します。

Adaptive Server Anywhere の動作の変更

ここでは、これまでのバージョンとは異なる動作について説明します。

- **Java のデバッグに必要なパーミッション** Java デバッガを使用するには、DBA 権限を持っているか、SA_DEBUG グループのメンバシップを付与されている必要があります。SA_DEBUG グループは、6.0.2 より前に作成されたデータベースには存在しません。このような古いデータベースでは、任意のユーザが Java デバッガを使用できます。SA_DEBUG グループは、セキュリティ上のループ・ホールを防止するために追加されました。

詳細については、『ASA SQL ユーザーズ・ガイド』> 「Java デバッガの稼働条件」を参照してください。

- **デフォルト・パケット・サイズの変更** クライアント／サーバ通信のデフォルト・パケット・サイズが、512 バイトから 1000 バイトに変更されました。この変更によって、マルチロー・フェッチと大きなローのフェッチのパフォーマンスが向上します。同時に必要なメモリも増加します。

パケット・サイズの詳細については、『ASA データベース管理ガイド』> 「CommBufferSize 接続パラメータ [CBSIZE]」と 『ASA データベース管理ガイド』> 「-p サーバ・オプション」を参照してください。

第 14 章

バージョン 6.0.1 の新機能

この章の内容

この章では、Adaptive Server Anywhere 6.0.1 で導入された新しい機能の概要について説明します。

バージョン 6.0.1 の新機能

この項では、Adaptive Server Anywhere バージョン 6.0.1 の新機能について説明します。新機能の一覧を挙げ、各機能の詳細が説明されている参照先も示します。

Adaptive Server Anywhere for Windows CE

Microsoft Windows CE は、ハンドヘルド・コンピュータ・デバイス用と特定のタスクを実行するために特別に開発された埋め込みデバイス用オペレーティング・システムです。

バージョン 6.0.1 以降の Adaptive Server Anywhere は、Windows CE でも使用できます。Adaptive Server Anywhere Windows CE 版には、次の特長があります。

- **完全な機能を備えたデータベース** Adaptive Server Anywhere の他のバージョンが持つすべての SQL 機能を、Windows CE 版でも使用できます。これらの機能には、トランザクション処理、参照整合性の確保、プロシージャとトリガなどが含まれます。

Java 機能とリモート・データ・アクセス機能は、Windows CE では使用できません。
- **デスクトップからの管理** PC へのネットワーク接続または直接接続が可能なデバイスで Windows CE を実行している場合は、PC 上で実行している Sybase Central から Windows CE データベースを管理できます。
- **ODBC と Embedded SQL アプリケーション** クライアント・アプリケーションの開発には、これらのインタフェースのどちらでも使用できます。
- **SQL Remote レプリケーション** Windows CE ActiveSync による同期との互換性を保つために、SQL Remote ファイル・リンクが実装されています。

リモート・データ・アクセス

リモート・データ・アクセス機能は、外部データ・ソースがあたかもローカル・データベース上に保存されているかのようにアクセスすることを可能にします。

リモート・データ・アクセスの詳細については、『ASA SQL ユーザーズ・ガイド』> 「リモート・データへのアクセス」と『ASA SQL ユーザーズ・ガイド』> 「リモート・データ・アクセスのサーバ・クラス」を参照してください。

文字セット変換

文字セット変換が追加され、クライアント・アプリケーションとデータベース・サーバとの間で文字セットが渡されるときに、異なる文字セット間で自動的に文字列が変換されるようになりました。これにより、混合文字セット環境での柔軟性が高くなりました。

文字セット変換は、同じ文字を異なる値で表す文字セット間でも実行できます。これを可能にするには、文字セット間にある程度の互換性が必要になります。たとえば、EUC-JIS とシフト JIS の文字セット間では文字セット変換を実行できますが、EUC-JIS と OEM Code Page 850 の間では実行できません。

文字セット変換を有効にするには、新しい `-ct` オプションを使用して、データベース・サーバを起動してください。次に例を示します。

```
dbeng9 -ct asademo.db
```

多くの場合、文字セット変換機能は自動的に実行され、ユーザの介入はほとんど必要ありません。

文字セット変換機能については、『ASA データベース管理ガイド』> 「文字セット変換を使用してデータベース・サーバを起動する」を参照してください。詳細については、『ASA データベース管理ガイド』> 「文字セット変換の知識」を参照してください。

新しい Java 機能

Java サポートにはいくつかの変更が加えられました。拡張機能には次のようなものがあります。

- **圧縮 jar ファイル** 圧縮された jar ファイルと zip ファイルをデータベースにインストールできるようになりました。ただし、Sun JDK に付属の *jar* ユーティリティは使用しないでください。それ以外の zip ユーティリティを使用すれば適切なファイルが生成されます。
- **Java プロシージャからの結果セット** 1つのストアード・プロシージャに Java メソッドをラップできます。これによって、呼び出し元の環境に 1つの結果セット、または複数の結果セットを返すことができます。

この機能の詳細については、『ASA プログラミング・ガイド』>「Java メソッドから返される結果セット」を参照してください。

- **デフォルトの内部接続** 内部 JDBC オペレーション用にデータベース接続を確立する場合、次の URL の使用をおすすめします。

```
jdbc:default:connection
```

バージョン 6.0.0 では、この接続を確立するのに空の文字列を使用していました。新バージョンでも空の文字列を使用できますが、優先されません。SQLJ1 が推奨する標準に対応する新しい URL をお使いください。

追加の新機能

Adaptive Server Anywhere バージョン 6.0.1 のその他の新しい機能には、以下のものが含まれます。

- **jConnect 4.0** この製品に含まれる jConnect が、バージョン 4.0 に更新されました。

- **自動スタート接続パラメータ** このパラメータを使用すると、ネットワーク接続が成功しなかった場合にはパーソナル・サーバを起動しないように設定できます。

詳細については、『ASA データベース管理ガイド』> 「AutoStart 接続パラメータ [ASTART]」を参照してください。

- **MESSAGE 文** MESSAGE 文の拡張機能によって、メッセージをクライアント、サーバ・ウィンドウ、またはログ・ファイルに送信できます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「MESSAGE 文」を参照してください。

- **メッセージ・コールバック** Windows Embedded SQL アプリケーションは、メッセージ・コールバック関数を登録することによって、要求が処理されている間でもサーバから受信したメッセージを処理できます。

詳細については、『ASA プログラミング・ガイド』> 「要求管理の実装」を参照してください。

- **オペレーティング・システムのスレッド制御の強化** 新しいデータベース・サーバのオプション (-gx) は、使用中のオペレーティング・システムのスレッド数を制御します。既存の -gt オプションは、同時に使用できるスレッド数を制御し、実際に使用できる CPU の数を制御します。

詳細については、『ASA データベース管理ガイド』> 「データベース・サーバ」を参照してください。

- **接続プロパティ・システム・プロシージャ** 2つのシステム・プロシージャが追加されました。これらは、接続情報のクエリに使用できます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「sa_conn_properties_by_conn システム・プロシージャ」と『ASA SQL リファレンス・マニュアル』> 「sa_conn_properties_by_name システム・プロシージャ」を参照してください。

- **NULLIF 関数** CASE 式の省略形を提供します。NULLIF は 2 つの式の値を比較します。1 番目の式と 2 番目の式が等しい場合、NULLIF は NULL を返します。1 番目の式と 2 番目の式が異なる場合、NULLIF は 1 番目の式を返します。NULLIF 関数は、いくつかの CASE 式の簡単な作成方法を提供します。

詳細については、『ASA SQL リファレンス・マニュアル』>「その他の関数」を参照してください。

SQL Remote の新機能

いくつかの機能が SQL Remote に追加されました。

- **最小化された Message Agent** `-q` オプションを使用すると、Message Agent を最小化されたウィンドウで起動できます。
- **Message Agent のメッセージの再送信要求** Message Agent が、失われたメッセージをどの時点で再送信するように要求するかは、`-rp` オプションを使用してユーザが設定できるようになりました。

これらのオプションの詳細については、『SQL Remote ユーザーズ・ガイド』>「Message Agent」と『SQL Remote ユーザーズ・ガイド』>「受信メッセージのポーリングのチューニング」を参照してください。

- **ステابل・キューのクリーニング** Adaptive Server Enterprise では、Message Agent の新しい `-fq` オプションを使用すると、確認済みのメッセージをステابل・キューからクリーニングでき、管理しやすくなります。

詳細については、『SQL Remote ユーザーズ・ガイド』>「Message Agent」を参照してください。

動作の変更

ここでは、バージョン 6.0.0 と 6.0.1 とで異なる動作について説明します。

Java システム・テーブルの変更 Java クラス情報を記録するために使用していたシステム・テーブル (SYSJAR、SYSJARCOMPONENT、SYSJAVACLASS) には、SMALLINT プライマリ・キーがありました。これらのデータ型は、INTEGER プライマリ・キーを使用するよう変更されました。この変更によって、より多くの Java クラスをデータベースに格納でき、データベースの Java クラスへのより多くの変更が可能になりました。

この変更は、新しいデータベースとアップグレード [dbupgrad] ユーティリティを使用して 6.0.1 以降のリリースからアップグレードされたデータベースに対して有効です。

-x コマンド・ラインのカッコ これまでのリリースでは、-x オプションを使用するデータベース・サーバ・コマンド・ラインは、オプションを囲むのに大カッコ ({}) を使用していました。これは CommLinks 接続パラメータの文字列にも適用されます。次に例を示します。

```
dbsrv9 -x tcpip{Localhost=demo}
```

大カッコを使用する既存のコマンド・ラインもサポートされてはいますが、オプションを囲むのにはカッコ () の使用をおすすめします。カッコの使用によって、データベース・サーバ・コマンド・ラインでのマルチバイト文字セットの識別子と、CommLinks 接続パラメータがサポートされます。

```
dbsrv9 -x tcpip(Localhost=demo)
```

Transact-SQL と jConnect 接続の ANSINULL デフォルト Adaptive Server Enterprise のデフォルト動作に合わせて ON に変更されました。

データベース・サーバ -v オプション バージョン 6 より前は、このオプションはトランザクション・ログに冗長出力を生成しました。この動作は変更され、現在では、-v はバージョン情報を提供するのに使用されます。

データベース・サーバ -gss オプション -gs サーバ・オプションはスタック・サイズの設定に使用されていましたが、動作が複雑でした。現在では、-gs オプションは使用せず、-gss によって同じ機能がより簡潔な方法で提供されます。

詳細については、『ASA データベース管理ガイド』> 「データベース・サーバ」を参照してください。

Interactive SQL での文字セット変換 以前は、CHAR_OEM_TRANSLATION オプションを DETECT に設定すると、Interactive SQL はデータベースから照合ラベルをフェッチして、OEM から ANSI への文字セット変換をオンにするかどうかを決定していました。照合ラベルが ANSI コード・ページを示す文字列で開始されている場合、変換はオフになります。そうでない場合、変換はオンになります。オプションを DETECT に設定すると、Interactive SQL により、データベースの照合ラベルとディスプレイ変換設定を示すメッセージがステータス・ウィンドウに表示されます。

新しい動作は次のとおりです。オプションを DETECT に設定すると、Interactive SQL は、サーバから CharacterSet 接続プロパティを取得します。この文字セットは、サーバがこの接続上のすべての文字列の送信に使用するものです。文字セットが ANSI コード・ページを示す場合、OEM から ANSI への変換はオフになります。そうでない場合、変換はオンになります。新しいメッセージが表示され、データベースの照合ラベル、この接続の通信に使用される文字セット、ディスプレイ変換設定が示されます。

第 15 章

ソフトウェアとデータベースのアップグレード

この章の内容

この章では、ソフトウェアとデータベースをアップグレードする方法について説明します。

Adaptive Server Anywhere のバージョン 5 アプリケーションをアップグレードする場合には、追加の考慮事項があります。詳細については、「[バージョン 5 アプリケーションのアップグレード](#)」325 ページを参照してください。

Adaptive Server Anywhere のアップグレード

複数バージョンの実行

SPX を使用しているバージョンが 1 つだけであるという条件で、複数バージョンの Adaptive Server Anywhere (たとえば、バージョン 8.0 とバージョン 9.0) を同時に実行できます。

既存のソフトウェアとの互換性

Adaptive Server Anywhere データベース・サーバは、それまでのバージョンで使っていたデータベースも処理できるので、既存のアプリケーションを引き続き使用するためにデータベースをアップグレードする必要はありません。

以前のバージョンの Adaptive Server Anywhere データベース・サーバを新しいバージョンのデータベースに対しては使用しないでください。

Adaptive Server Anywhere ソフトウェア・バージョン 6.0.0 以降では、クライアント・ソフトウェアをアップグレードしなくても、データベース・サーバをアップグレードできます。

- バージョン 9 のデータベース・サーバは、バージョン 6.0.0 以降のソフトウェアを使用するクライアント・アプリケーションからの接続をサポートします。
- 現在のバージョンの Sybase Central には、以前のバージョンのデータベースとデータベース・サーバを管理する次の機能が用意されています。
 - バージョン 7 以降のサーバで動作するバージョン 7 以降のデータベースを完全に管理します。
 - バージョン 8 以降のデータベース・サーバで動作するバージョン 5 または 6 のデータベースには、[データベース・アンロード] ウィザードを使用してデータベース・ファイル・フォーマットをアップグレードする目的でのみ接続できます。
 - バージョン 7 以前のデータベース・サーバで動作するバージョン 6 以前のデータベースはサポートされません。

バージョン 5 以前のクライアント／サーバのインストール環境をアップグレードする方法については、「[バージョン 5 アプリケーションのアップグレード](#)」325 ページを参照してください。

アップグレードをする理由

ソフトウェアの現在のバージョンで提供される新機能を使用する場合は、データベースをアップグレードしてください。

バージョン 9 では、次の 2 種類のアップグレードが可能です。

- **データベースのアップグレード** アップグレード [dbupgrad] ユーティリティを使用してデータベースをアップグレードします。この処理では、システム・テーブルが更新され、新しいデータベース・オプションが提供されます。また、システム・プロシージャも追加または変更されます。

データベースのアップグレードは、既存のデータベース・ファイルそのものに対して実行できます。データベースをアンロードして再ロードする必要はありません。

詳細については、「[データベースのアップグレード](#)」307 ページを参照してください。

- **データベース・ファイル・フォーマットのアップグレード** 新しいインデックス・タイプなどのように、新しいデータベース・ファイル・フォーマットを必要とする機能もあります。アップグレード [dbupgrad] ユーティリティを使用しても、データベース・ファイルのフォーマットは変更されません。

データベース・ファイルのフォーマットをアップグレードするには、データベースをアンロードしてから再ロードします。

詳細については、「[データベース・ファイル・フォーマットのアップグレード](#)」310 ページを参照してください。

データベースのアップグレード

データベースをアップグレードすると、バージョン 9 の機能を利用できるようにシステム・テーブル、システム・プロシージャ、データベース・オプションが追加、変更されます。ディスクへのデータの保存やそのデータへのアクセスなどに使用されるファイル・フォーマット

トは変更されません。そのため、ソフトウェアの最新バージョンで提供される機能と強化されたパフォーマンスのすべてを利用できるわけではありません。

データベース・ファイル・フォーマットのアップグレードについては、「[データベース・ファイル・フォーマットのアップグレード](#)」310 ページを参照してください。

❖ データベースをアップグレードするには、次の手順に従います (Sybase Central の場合)。

- 1 ソフトウェアをアップグレードする場合の一般的な対応策を行います。

詳細については、「[標準アップグレードを行う前の一般的な注意事項](#)」313 ページを参照してください。

- 2 アップグレード対象のデータベースを稼働しているバージョン9 データベース・サーバを起動します。
- 3 Sybase Central からアップグレードするデータベースに接続します。
- 4 データベースを右クリックし、ポップアップ・メニューから [データベースのアップグレード] を選択します。[データベースのアップグレード] ウィザードが表示されます。
- 5 ウィザードの指示に従います。
- 6 アップグレードしたデータベースを使用する前に、データベースを停止し、トランザクション・ログを保存します。

❖ データベースをアップグレードするには、次の手順に従います (コマンド・ラインの場合)。

- 1 ソフトウェアをアップグレードする場合の一般的な対応策を行います。

詳細については、「[標準アップグレードを行う前の一般的な注意事項](#)」313 ページを参照してください。

- 2 アップグレードするデータベースに対して、排他アクセスを持っていることを確認してください。
- 3 システム・パスには、ほかのユーティリティより前にバージョン 9 のユーティリティが置かれていることを確認してください。

詳細については、「[ユーティリティの使用](#)」314 ページを参照してください。

- 4 データベースに対して、次のようにアップグレード [dbupgrad] ユーティリティを実行します。

```
dbupgrad -c "connection-string"
```

connection-string で DBA 権限を使用して、アンロードするデータベースに接続します。

詳細については、『ASA データベース管理ガイド』> 「dbupgrad コマンド・ライン・ユーティリティを使用したデータベースのアップグレード」を参照してください。

- 5 アップグレードしたデータベースを使用する前に、データベースを停止し、トランザクション・ログを保存します。

❖ データベースをアップグレードするには、次の手順に従います (SQL の場合)。

- 1 Interactive SQL または SQL 文を実行できる別のアプリケーションからデータベースに接続します。
- 2 ALTER DATABASE 文を実行します。JAVA 句を使って、データベース内の Java アプリケーションで使用される JDK レベルをアップグレードするかどうかを指定します。

たとえば、次の文は、JDK レベルを変更せずにデータベースをアップグレードします。

```
ALTER DATABASE UPGRADE
```

詳細については、『ASA SQL リファレンス・マニュアル』> 「ALTER DATABASE 文」を参照してください。

データベース・ファイル・フォーマットのアップグレード

この項では、データベースのアンロードと再ロードを使用して、データベースのファイル・フォーマットをアップグレードする方法について説明します。

警告

大規模なデータベースのアンロードや再ロードは時間がかかり、大量のディスク領域が必要になることがあります。この処理では、アンロードされたデータと新しいデータベース・ファイルを保持するため、データベースの約 2 倍のディスク領域にアクセスする必要があります。

特別な注意事項

SQL Remote レプリケーションで使用するデータベースや、Mobile Link インストール環境内のリモート・データベースのファイル・フォーマットをアップグレードするときに、ユーティリティを使用する場合は、必ず `-ar` または `-an` オプションを使用してください。このオプションを使用すると、トランザクション・ログのオフセットが、以前のデータベースと同じになるように新しいデータベースにも設定されます。Sybase Central では、次の指示に従うと、このオプションを確実に使用できます。

プロキシ・テーブルを持つデータベースをアンロードして再ロードする場合は、ユーザがローカル・データベースとリモート・データベースの両方に対して同じパスワードを持つ場合でも、外部ログインを作成して、ローカル・ユーザをリモート・ユーザにマッピングしてください。外部ログインがない場合は、リモート・サーバに接続できないため再ロードが失敗します。

レプリケーションに関係するバージョン 5 データベースのアンロードと再ロードを行う場合は、この項の指示ではなく、『SQL Remote ユーザーズ・ガイド』> 「レプリケーションに参加しているデータベースのアンロードと再ロード」の指示に従ってください。

アンロードと再ロードの手順

❖ **データベース・ファイル・フォーマットをアップグレードするには、次の手順に従います (Sybase Central の場合)。**

- 1 ソフトウェアをアップグレードする場合の一般的な対応策を行います。

詳細については、「[標準アップグレードを行う前の一般的な注意事項](#)」313 ページを参照してください。

- 2 アップグレード対象のデータベースに対してバージョン 9 データベース・サーバを起動します。次に例を示します。

```
dbeng9 mydatabase.db
```

- 3 Sybase Central を起動します。

[スタート] メニューから、[プログラム] – [SQL Anywhere 9] – [Sybase Central] を選択します。

- 4 アップグレードするデータベースに接続します。

バージョン 5.x またはバージョン 6.x のデータベースに接続すると、[データベース・アンロード] ウィザードでデータベースをアンロードして再ロードするかどうかを尋ねるダイアログが表示されます。[はい] をクリックします。ウィザードが完了すると、データベースへの接続が閉じられます。

- 5 データベースを右クリックし、ポップアップ・メニューから [データベースのアンロード] を選択します。

[データベース・アンロード] ウィザードが表示されます。

- 6 このウィザードの最初のページを読んで、[次へ] をクリックします。

- 7 接続しているデータベースのアンロードを選択します。[次へ] をクリックします。

- 8 [新しいデータベースへのアンロードと再ロード] を選択します。[次へ] をクリックします。

- 9 データベースの新しいファイル名を指定します。[次へ] をクリックします。

新しいデータベースのページ・サイズを指定できます。デフォルトのページ・サイズは 2048 バイトです。

必要に応じてデータベース・ファイルを暗号化できます。暗号化を行うと、データベースを起動するたびに、暗号化キーが必要になります。

データベース・ファイルの暗号化の詳細については、『SQL Anywhere Studio セキュリティ・ガイド』> 「データベースの暗号化」を参照してください。

- 10 [構造とデータをアンロード] を選択します。[次へ] をクリックします。
- 11 [すべてのデータベース・オブジェクトをアンロード] を選択します。[次へ] をクリックします。
- 12 アンロード／再ロードが完了したときに新しいデータベースに接続するかどうかを指定します。
- 13 [完了] をクリックすると、処理が開始します。新しいデータベースを調べて、アップグレードが正常に完了したことを確認する必要があります。

❖ データベース・ファイル・フォーマットをアップグレードするには、次の手順に従います (コマンド・ラインの場合)。

- 1 ソフトウェアをアップグレードする場合の一般的な対応策を行います。

詳細については、[「標準アップグレードを行う前の一般的な注意事項」 313 ページ](#)を参照してください。

- 2 アンロードと再ロードを行うデータベースに対して、排他アクセスを持っていることを確認してください。他のユーザは接続できません。
- 3 システム・パスには、ほかのユーティリティより前にバージョン 9 のユーティリティが置かれていることを確認してください。

詳細については、[「ユーティリティの使用」 314 ページ](#)を参照してください。

- 4 -ar オプションを使用してアンロード [dbunload] ユーティリティを実行し、新しいデータベースを作成します。

```
dbunload -ar -c "connection-string"
```

connection-string で DBA 権限を使用して、アンロードするデータベースに接続します。このコマンドは、既存のデータベースをアップグレード後のデータベースに置き替えます。-ar オプションを使用するには、パーソナル・サーバに接続するか、アンロード [dbunload] ユーティリティと同じマシン上にあるネットワーク・サーバに接続します。

アンロード [dbunload] ユーティリティのその他のオプションについては、『ASA データベース管理ガイド』> 「dbunload コマンド・ライン・ユーティリティを使用したデータベースのアンロード」を参照してください。

- 5 再ロードしたデータベースを使用する前に、データベースを停止し、トランザクション・ログを圧縮します。

アンロードと再ロードのときに (大文字と小文字を「区別する」から「区別しない」に変更するなど) データベースの特性を変更する場合、手順はもう少し複雑になります。詳細については、『ASA SQL ユーザーズ・ガイド』> 「データベースの再構築」を参照してください。

標準アップグレードを行う前の一般的な注意事項

どのようなアプリケーションをアップグレードする場合でも、事前に行うべき作業がいくつかあります。Adaptive Server Anywhere をアップグレードする場合も同じです。

- **動作の変更のチェック** 記述されているバージョン 9.0 の動作変更が既存のアプリケーションに影響を与えないことを確認してから、アップグレードしてください。

詳細については、「[Adaptive Server Anywhere の動作の変更](#)」125 ページを参照してください。

- **テスト** 事前にバージョン 9 環境でアプリケーションのテストを徹底的に行ってから、実際の運用に使用するアプリケーションをアップグレードしてください。
- **バックアップ** 既存のソフトウェアとデータのバックアップを取ってから、アップグレードを行ってください。また、データベースのアップグレード前にさかのぼるリカバリはできないので、アップグレード後にバックアップを取り、その後のリカバリに備えてください。

Ultra Light データベースや、Mobile Link または SQL Remote インストール環境の Adaptive Server Anywhere リモート・データベースなど、同期に関連するデータベースについては、未処理の変更をすべて同期させてからアップグレードを行ってください。

- **アップグレード手順のテスト** 多くのエンド・ユーザをアップグレードする場合は、事前にアップグレード手順を十分にテストしてください。

SQL Anywhere はさまざまな設定で使用されるので、すべての場合に対応できるアップグレード手順といったものではありません。

ユーティリティの使用

Adaptive Server Anywhere の複数のバージョンが同じマシンにインストールされている場合は、ユーティリティを使用するとき、システムのパスに注意してください。セットアップでは、最も新しくインストールされたバージョンの実行プログラムのディレクトリがシステム・パスの最後に追加されるため、ソフトウェアの最新バージョンをインストールしたにもかかわらず、以前にインストールしたバージョンを実行している場合があります。

たとえば、パス内で Adaptive Server Anywhere バージョン 7 の実行プログラムのディレクトリがバージョン 9 の実行プログラムのディレクトリよりも前にある場合に **dbinit** コマンドを使用すると、バージョン 7 のツールを使用することになり、その結果、バージョン 7 のデータベースが作成されてしまいます。

❖ **バージョン 9.0 ユーティリティを使用するには、次の手順に従います。**

- コマンドを実行する前に、現在のディレクトリをバージョン 9.0 実行プログラム・ディレクトリに変更します。

または

バージョン 9.0 実行プログラム・ディレクトリが、バージョン 9.0 以前の実行プログラム・ディレクトリより前になるようにシステム・パスを変更します。

Mobile Link のアップグレード

既存の Mobile Link インストール環境をアップグレードするには、次の順序でコンポーネントをアップグレードします。

1. Mobile Link サーバを停止します。
2. 統合データベースをアップグレードします。

詳細については、「[統合データベースのアップグレード](#)」317 ページを参照してください。

3. Mobile Link 同期サーバをアップグレードします。

詳細については、「[Mobile Link 同期サーバのアップグレード](#)」318 ページを参照してください。

4. Mobile Link サーバを起動します。
5. Mobile Link クライアントをアップグレードします。

Adaptive Server Anywhere リモート・データベースの詳細については、「[Adaptive Server Anywhere Mobile Link クライアントのアップグレード](#)」319 ページを参照してください。Ultra Light アプリケーションの詳細については、「[Ultra Light アプリケーションのアップグレード](#)」321 ページを参照してください。

新しい Mobile Link クライアントは以前のバージョンの Mobile Link 同期サーバとは互換性ありませんが、新しい Mobile Link 同期サーバは以前のバージョンのクライアントと互換性があります。

アップグレードを行う前に、影響がありそうな動作の変更を確認し、一般的なアップグレード前の対応策を実行してください。

詳細については、「[バージョン 8 での動作の変更](#)」217 ページと「[標準アップグレードを行う前の一般的な注意事項](#)」313 ページを参照してください。

統合データベースのアップグレード

統合データベース内の Mobile Link システム・テーブルをアップグレードしないと、新しい Mobile Link 同期サーバを使用できません。

Adaptive Server Anywhere 統合データベースを使用している場合は、通常のアップグレード処理を実行して Mobile Link システム・テーブルをアップグレードし、この項で説明する手順は実行しないでください。

Adaptive Server Anywhere データベースをアップグレードする方法については、「[データベースのアップグレード](#)」307 ページを参照してください。

Mobile Link バージョン 6 からの場合と Mobile Link バージョン 7 からの場合では、アップグレード手順が少しだけ異なります。

❖ 統合データベースをアップグレードするには、次の手順に従います (DB2 以外の場合)。

- 1 Mobile Link バージョン 6 からアップグレードする場合にかぎり、セットアップ SQL スクリプト `MobiLink¥setup¥syncxxx.sql` を実行して、Mobile Link のシステム・テーブルとストアード・プロシージャを作成します。この場合、xxx は `ora` (Oracle)、`ase` (ASE)、または `mss` (Microsoft SQL Server) です。
- 2 データベース管理システムのアップグレード・スクリプトのロケーションを検索します。

アップグレード・スクリプトは、SQL Anywhere ディレクトリの下に `MobiLink/upgrade/version` ディレクトリにあります。このディレクトリは、アップグレード対象の Mobile Link のバージョンを示しています。

- 3 データベース管理システムにアップグレード・スクリプトを適用します。

データベース管理システムの管理ユーティリティまたは Interactive SQL ユーティリティを使用して、SQL スクリプトを適用します。

❖ 統合データベースをアップグレードするには、次の手順に従います (DB2 の場合)。

- 1 *MobiLink¥setup¥SyncDB2Long_901.class* ファイルを DB2 サーバ・マシン上の *SQLLIB¥FUNCTION* ディレクトリにコピーします。この場合は通常、インスタンスを再起動する必要があります。詳細については、使用している DB2 のマニュアルを参照してください。
- 2 Mobile Link バージョン 6 からアップグレードする場合にかぎり、セットアップ SQL スクリプト *MobiLink¥setup¥syncdb2long.sql* を実行して、Mobile Link のシステム・テーブルとストアド・プロシージャを作成します。
- 3 DB2 アップグレード・スクリプトのロケーションを検索します。

アップグレード・スクリプトは *upgrade_db2tolong.sql* という名前です。SQL Anywhere ディレクトリの下に *MobiLink/upgrade/version* ディレクトリにあります。ディレクトリは、アップグレード対象の Mobile Link のバージョンを示しています。

- 4 *upgrade_db2tolong.sql* をコピーして、そのコピーを修正します。スクリプトの先頭にある **CONNECT** 文を変更して、接続するインスタンスで動作できるようにします。コピーした SQL スクリプトを統合データベースに適用します。

未定義の **ml_add_user** スクリプトを削除した場合にエラーが発生しても、無視してかまいません。

Mobile Link 同期サーバのアップグレード

以前のバージョンの代わりにバージョン 9 の Mobile Link 同期サーバを使用する前に、影響がありそうな動作の変更を確認してください。

動作の変更のリストについては、「[Mobile Link の動作の変更](#)」227 ページを参照してください。

Adaptive Server Anywhere Mobile Link クライアントのアップグレード

運用環境では、統合データベースと Mobile Link 同期サーバの両方をアップグレードしてから、Adaptive Server Anywhere リモート・データベースをアップグレードします。

検討すべきアップグレードは複数あります。

- ソフトウェアのアップグレード
- リモート・データベース自体のアップグレード
- アプリケーション全体のアップグレード

いずれの場合も、リモート・データベースのデータをバックアップする 1 つの方法として、未処理の操作をすべて同期させてからアップグレードしてください。

ソフトウェアのアップグレード

dbmsync Mobile Link クライアントと Adaptive Server Anywhere データベース・エンジンは同時にアップグレードすることをおすすめします。リモート・データベースをアップグレードしてから、新しい *dbmsync* ユーティリティを実行してください。

Mobile Link バージョン 9 クライアントでは、同期を実行するために Mobile Link バージョン 9 同期サーバが必要です。Mobile Link バージョン 9 クライアントは、バージョン 9 より以前の Mobile Link 同期サーバとは同期しません。Mobile Link のアップグレードの詳細については、「[Mobile Link のアップグレード](#)」316 ページを参照してください。

リモート・データベースのアップグレード

Adaptive Server Anywhere リモート・データベースをアップグレードするには、アップグレード [dbugrad] ユーティリティを実行します。より完全なアップグレードを行うために、データベース・ファイル・フォーマットをアップグレードすることもできます。

いずれのアップグレード操作も、Adaptive Server Anywhere データベースの場合と同じ手順で実行できます。手順については、「[データベースのアップグレード](#)」307 ページと「[データベース・ファイル・フォーマットのアップグレード](#)」310 ページを参照してください。

アプリケーションのアップグレード

Mobile Link アプリケーションの新しいバージョンを配備する場合、同期スクリプトには新しいバージョンの名前を使用することをおすすめします。たとえば、既存のアプリケーションで **v1** というスクリプトのバージョンを使用している場合は、アップグレードしたアプリケーションでは **v2** という名前のスクリプト・バージョンを使用します。この2つのスクリプト・バージョンは同時に使用できます。これにより、リモート・データベースを一度にではなく、段階的にアップグレードすることが容易になります。

バージョン 9.0.0 以降では、*dbmlsrv -zd* オプションは削除されています。配備環境で *-zd* オプションを使用している場合、アップグレードを行うには、最後のダウンロード・タイムスタンプが最初のパラメータとして採用されるようダウンロード・スクリプトを変更します。

Ultra Light アプリケーションのアップグレード

バージョン 9 にアップグレードするには、バージョン 9 ソフトウェアで Ultra Light アプリケーションを再構築します。この場合、バージョン 9 Ultra Light ジェネレータを実行して、バージョン 9 ライブラリにリンクします。

バージョン 9 Ultra Light ジェネレータを実行するには、コマンド・ラインに明示的にパスを含めるか、システム・パスで以前のバージョンのディレクトリよりも前にバージョン 9 ディレクトリが記述されていることを確認します。

追加された Palm コード・セグメンテーションを利用するには、リファレンス・データベースをアップグレードします。リファレンス・データベースをアップグレードしないと、Palm コードはすべて同じセグメントに配置されます。Palm Computing Platform 以外のプラットフォーム用に開発する場合は、リファレンス・データベースをバージョン 9 にアップグレードしなくても、バージョン 9 アプリケーションを生成できます。

Palm C/C++ 開発

バージョン 8 またはバージョン 9 ソフトウェアを使用している場合は、`db_init` (Embedded SQL) または `ULData::Open` (静的型 C++ API) を呼び出す前に、関数呼び出しを追加して、使用する記憶タイプを指定してください。追加する関数呼び出しは次のとおりです。

```
ULEnablePalmRecordDB( &sqlca );
```

または

```
ULEnableFileDB( &sqlca );
```

レコード・ベースの主記憶装置を使用する場合は

ULEnablePalmRecordDB を、ファイル・ベースの補助記憶装置を使用する場合は **ULEnableFileDB** を呼び出します。

配備されたデータベースのアップグレード

すべての変更が同期されていることを確認してから、配備された Ultra Light データベースをアップグレードしてください。Palm アプリケーションの場合、アプリケーションのバージョンごとに新しい作成者 ID を使用し、新しいアプリケーションをインストールしてから、

古いアプリケーションとデータベースを Palm デバイスから削除することをおすすめします。同じ作成者 ID を使用する場合は、まず既存のアプリケーションを削除してください。

Ultra Light ソフトウェアは、いくつかの条件に基づいてデータベースをアップグレードできます。詳細については、『Ultra Light データベース・ユーザーズ・ガイド』> 「Ultra Light データベース・スキーマのアップグレード」と 『Ultra Light データベース・ユーザーズ・ガイド』> 「Ultra Light ソフトウェアのアップグレード」を参照してください。

Ultra Light バージョン 9 アプリケーションでは、同期を実行するために Mobile Link バージョン 9 同期サーバが必要です。Ultra Light バージョン 9 アプリケーションは、バージョン 9 より以前の Mobile Link 同期サーバとは同期しません。Mobile Link のアップグレードの詳細については、[「Mobile Link のアップグレード」316 ページ](#)を参照してください。

Mobile Link アプリケーションの新しいバージョンを配備する場合、同期スクリプトには新しいバージョンの名前を使用することをおすすめします。これにより、リモート・データベースを一度にではなく、段階的にアップグレードすることが容易になります。

SQL Remote のアップグレード

既存の SQL Remote インストール環境をバージョン 6 以降からアップグレードする場合は、各データベース・サーバをその Message Agent (*dbremote* と *ssremote*) より前、または同時にアップグレードします。Message Agent のアップグレードには特に決まった順序はありません。

バージョン 5 のユーザは、この項の指示ではなく、「[バージョン 5 SQL Remote インストール環境のアップグレード](#)」357 ページの指示に従ってください。

- **データベースのアップグレードは不要** データベースをアップグレードすることは必須ではありませんが、最高のパフォーマンスを実現するためには、データベースをアンロードしてから再ロードして、データベース・ファイル・フォーマットをアップグレードすることをおすすめします。すべてのデータベースを同時にアップグレードする必要はありません。

データベースのアップグレードについては、「[データベースのアップグレード](#)」307 ページと「[データベース・ファイル・フォーマットのアップグレード](#)」310 ページを参照してください。

- **ソフトウェア・アップグレードは一度に 1 サイトずつ行う** 以前のバージョンの Message Agent では、COMPRESSION データベース・オプションが -1 (マイナス 1) の値に設定されていれば、バージョン 9 Message Agent とメッセージを交換できます。インストール環境全体のソフトウェアを同時にアップグレードする必要はありません。
- **Message Agent とデータベース・サーバは別々にアップグレードできる** データベース・サーバは Message Agent よりも先にアップグレードできます。ただし、パフォーマンスの面からは、Message Agent をデータベース・サーバと同時にアップグレードすることをおすすめします。

バージョン 5 アプリケーションのアップグレード

この章の内容

クライアント／サーバ通信プロトコルは、SQL Anywhere バージョン 5 と Adaptive Server Anywhere バージョン 9 の間で変更されました。この章では、バージョン 5 から新しいバージョンにアップグレードするユーザを対象にアップグレード手順について説明します。

この章では、SQL Anywhere バージョン 5.0 と 5.5 のユーザと Watcom SQL 4.0 のユーザを対象に通信プロトコルの変更を追加するためのアップグレード手順について説明します。特に、クライアント／サーバ・インストールの手順ごとのアップグレードにおける問題点について説明します。

バージョン 5 のアップグレードの知識

データベース・アプリケーションと DBMS は、いくつかのコンポーネントで構成されています。規模の大きい SQL Anywhere インストール環境の場合、一般に、いくつかのアプリケーションを実行する多くのクライアント・マシンが複数のサーバに接続されています。

このようなシステムをアップグレードする場合、どのコンポーネントをどのような順序でアップグレードするかを決定する必要があります。この章は、このような決定を行ってアップグレードを実施するのに役立ちます。

SQL Anywhere バージョン 5

この章では、SQL Anywhere のバージョン 5.0 とバージョン 5.5 の両方を「SQL Anywhere バージョン 5」と記載します。

バージョン 5 アプリケーションをアップグレードするとき、このソフトウェアの新しいバージョンでの接続の仕組みを理解しておくと参考になります。詳細については、『ASA データベース管理ガイド』>「データベースへの接続」を参照してください。

システムを構成するコンポーネント

現在 SQL Anywhere バージョン 5 を実行している場合、次のコンポーネントの一部または全部がシステムに含まれることになります。

- **アプリケーション** SQL Anywhere コンポーネント以外に使用するアプリケーションです。
- **接続パラメータ** SQL Anywhere バージョン 5.0 の接続パラメータです。接続パラメータは、ODBC データ・ソースなどからアセンブルされます。
- **ドライバ・マネージャ** ODBC アプリケーションのための ODBC ドライバ・マネージャです。

- **ODBC ドライバ** ODBC アプリケーションのための SQL Anywhere バージョン 5 の ODBC ドライバです。ネットワーク・アプリケーションの場合、ODBC ドライバはクライアント・マシン上にあります。
- **インタフェース・ライブラリ** SQL Anywhere バージョン 5 のインタフェース・ライブラリは、ODBC と Embedded SQL アプリケーションによって使用されます。ネットワーク・アプリケーションの場合、インタフェース・ライブラリはクライアント・マシン上にあります。
- **SQL Anywhere クライアント** ネットワーク・アプリケーションの `dbclient.exe` 実行プログラムとそのコマンド・ラインです。コマンド・ラインでは、サーバ名と一連のネットワーク・プロトコル・オプションを指定できます。この値はバッチ・ファイルまたは ODBC データ・ソース StartLine パラメータに保管されます。ネットワーク・アプリケーションの場合、SQL Anywhere クライアントはクライアント・マシン上にあります。
- **データベース・サーバ** SQL Anywhere バージョン 5 のデータベース・サーバです。ネットワーク・アプリケーションの場合、これはクライアント・コンポーネントとは別のマシン上にあることもあります。
- **データベース** SQL Anywhere バージョン 5 のデータベースです。これはデータベース・サーバと同じマシン上にあります。

データベースのアップグレードは不要

新しいバージョンのデータベース・サーバを既存のアプリケーションで使用するためにデータベースをアップグレードする必要はありません。

アップグレードに関する主な問題

バージョン 5 からのアップグレードに関する主な問題は、クライアント／サーバ通信プロトコルの変更に起因するものです。この変更のために、バージョン 5 のインタフェース・ライブラリは新しいバージョンのデータベース・サーバとは通信できません。

この問題の対策として、Adaptive Server Anywhere には、互換性ライブラリの DLL が含まれています。この DLL により、バージョン 5 と現在のバージョンの両方のデータベース・サーバとの通信が可能になります。

動作の変更

また、このマニュアルにある動作変更リストで、バージョン 5 より後のバージョンの変更が、作成済みのアプリケーションに影響する可能性があるかどうかを確認してください。

さらに、Transact-SQL 外部ジョインを使用するアプリケーションについては、新しいバージョンと次のような非互換性があることも注意してください。

Transact-SQL 外部ジョインの NULL 入力テーブルは、別の通常のジョインまたは外部ジョインには使用できません。たとえば、次のクエリでは、テーブル S は `R.x *= S.x` の NULL 入力テーブルですが、別のジョインに使用されています。

```
-- invalid query
SELECT *
FROM R, S, T
WHERE R.x *= S.x
AND S.y = T.y
```

データベースをアップグレードする時期

ソフトウェアの現在のバージョンの使用を開始するときにバージョン 5 データベース自体をアップグレードする必要はありません。データベースのアップグレードが必要なのは、バージョン 5 よりも後に導入された多数の機能を利用する場合です。

データベースのアップグレードについては、「[ソフトウェアとデータベースのアップグレード](#)」305 ページを参照してください。

複数のバージョンの実行

SQL Anywhere Studio ソフトウェアは、別々のディレクトリにインストールすれば、2 つの異なるバージョンを実行できるように設計されています。これは SQL Anywhere セットアップ・プログラムのデフォルトの動作です。

SQL Anywhere バージョン 5 アーキテクチャの復習

ここでは、SQL Anywhere バージョン 5 データベースに接続する SQL Anywhere バージョン 5 アプリケーションのアーキテクチャについて復習します。

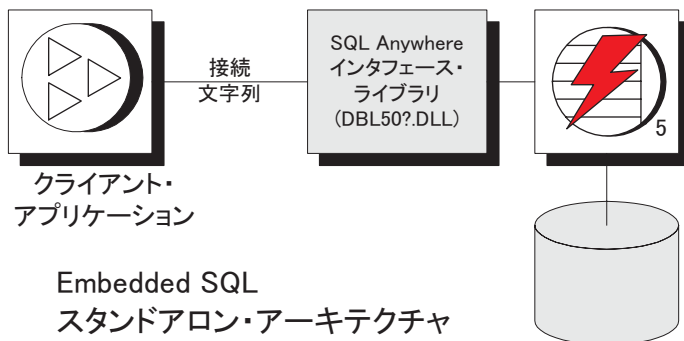
この情報は、新しいバージョンにアップグレードするときに必要な変更を理解するのに役立ちます。SQL Anywhere バージョン 5 アーキテクチャに精通している場合には、この項を読む必要はありません。

バージョン 5 のスタンドアロン・コンポーネント (Embedded SQL)

SQL Anywhere バージョン 5 をパーソナル・サーバとして、Embedded SQL クライアント・アプリケーションとともに使用している場合、マシンには次のコンポーネントがあります。

- SQL Anywhere バージョン 5 データベース
- SQL Anywhere バージョン 5 データベース・エンジン (パーソナル・データベース・サーバ)
- SQL Anywhere バージョン 5 インタフェース・ライブラリ
- SQL Anywhere バージョン 5.0 接続文字列

次の図は、これらのコンポーネントがどのように関連しているかを示しています。



この図では、*dbi50?.dll* の疑問符はオペレーティング・システムを示す 1 文字を表します。Windows オペレーティング・システムでは、インタフェース・ライブラリの名前は *dlb50t.dll* になります。

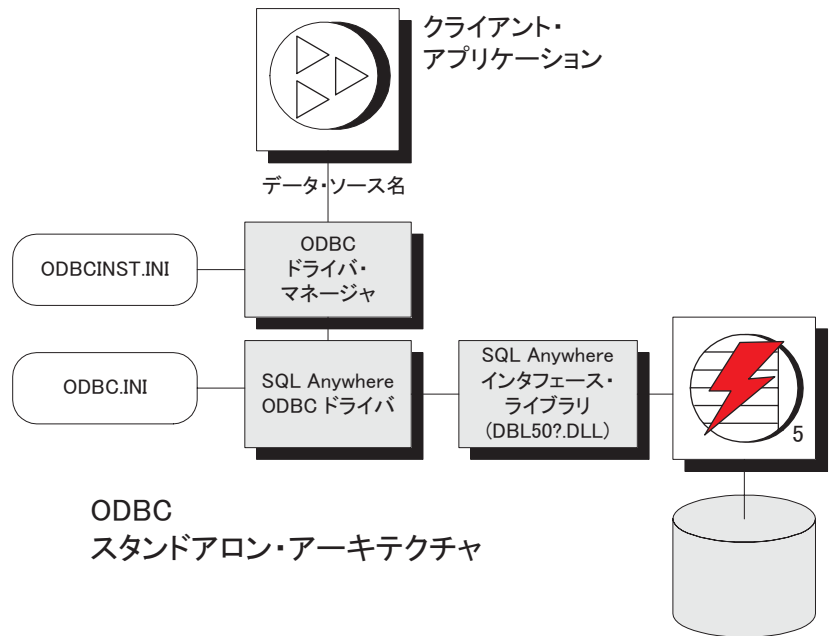
Embedded SQL クライアント・アプリケーションは、インタフェース・ライブラリを呼び出します。インタフェース・ライブラリは名前で参照されるので、*dbi50t.dll* という名前はクライアント・アプリケーションの一部です。

バージョン 5 のスタンドアロン・コンポーネント (ODBC)

SQL Anywhere バージョン 5 をパーソナル・サーバとして、ODBC クライアント・アプリケーションと共に使用している場合、マシンには次のコンポーネントがあります。

- SQL Anywhere バージョン 5 データベース
- SQL Anywhere バージョン 5 データベース・エンジン
- SQL Anywhere バージョン 5 インタフェース・ライブラリ
- SQL Anywhere バージョン 5 ODBC ドライバ
- ODBC ドライバ・マネージャ
- SQL Anywhere バージョン 5 接続記述。これは ODBC データ・ソースか、またはアプリケーションからの接続文字列です。

次の図は、これらのコンポーネントがどのように関連しているかを示しています。クライアント・アプリケーションは、データ・ソース名を ODBC ドライバ・マネージャに渡します。ODBC ドライバ・マネージャは、ODBCINST.INI の該当するドライバを検索します。ドライバは ODBC.INI の接続情報を検索し、インタフェース・ライブラリを介して、SQL Anywhere バージョン 5 データベース・エンジンに接続します。



バージョン 5 のクライアント／サーバ・コンポーネント

ネットワーク・サーバとして SQL Anywhere を使用している場合、サーバ・マシンには、次のコンポーネントがあります。

- SQL Anywhere バージョン 5 データベース
- SQL Anywhere バージョン 5 データベース・サーバ

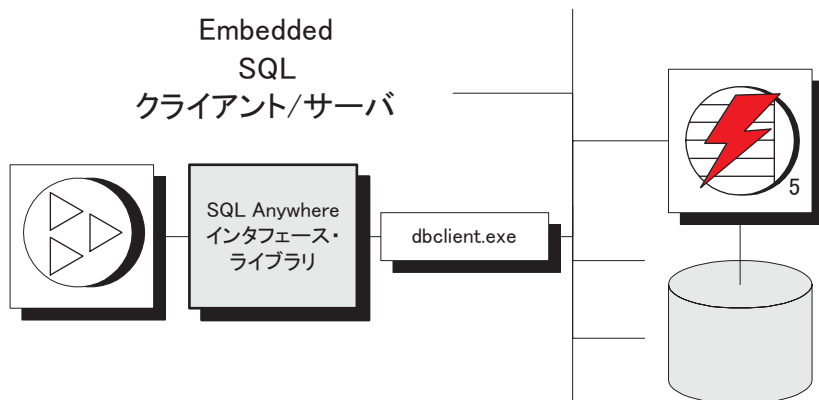
クライアント・マシンには、次のコンポーネントがあります。

- SQL Anywhere バージョン 5 クライアント実行プログラム
- SQL Anywhere バージョン 5 インタフェース・ライブラリ
- アプリケーションが ODBC を使用する場合、SQL Anywhere ODBC ドライバ

- SQL Anywhere バージョン 5 接続記述。これは ODBC データ・ソースか、またはアプリケーションからの接続文字列です。

データ・ソースには、SQL Anywhere クライアント・コマンド・ラインに対応する接続情報が開始オプションに含まれていることがあります。また、アプリケーション内に接続文字列が含まれていたり、特定のパラメータとオプションでクライアントを起動するバッチ・ファイルが提供されている場合もあります。

次の図は、バージョン 5 Embedded SQL クライアント／サーバ接続のアーキテクチャを示しています。ODBC アプリケーションの場合、ODBC ドライバ・マネージャと ODBC ドライバはアプリケーションとインタフェース・ライブラリの間に位置しています。



Embedded SQL アプリケーションのアップグレード

データベース・サーバとインタフェース・ライブラリをアップグレードすることによって、SQL Anywhere バージョン 5 スタンドアロン・アプリケーションをアップグレードして、新しいバージョンのデータベース・サーバを使用できます。データベースまたはクライアント・アプリケーション自体をアップグレードする必要はありません。

アップグレード手順では、「[互換性ライブラリ](#)」を使用します。互換性ライブラリは、Embedded SQL アプリケーションがバージョン 5 とそれ以降のバージョンの両方のデータベース・サーバで動作できるようにするダイナミック・ライブラリ (PC では DLL、UNIX では共有ライブラリ) です。

互換性ライブラリの詳細については、「[互換性ライブラリの使用](#)」347 ページを参照してください。

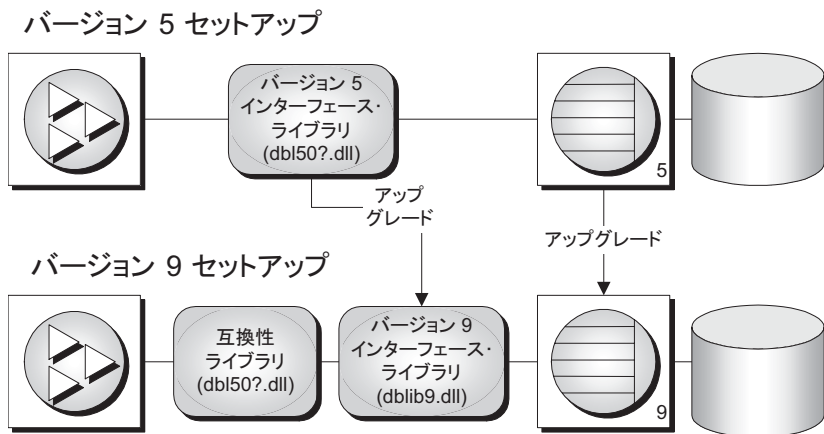
StartLine 接続パラメータ

アプリケーション接続文字列が `dbclient.exe` または `dbeng50.exe` コマンド・ラインを明示的に提供する StartLine パラメータを使用していて、これがアプリケーションにハード・コードされている場合、アップグレードには特別な注意が必要です。

StartLine 接続パラメータを使用するアプリケーションのアップグレードについては、「[開始パラメータと互換性ライブラリ](#)」352 ページを参照してください。

アップグレードされたコンポーネント

次の図は、アップグレードするためにセットアップで行う必要のある変更を示します。



互換性ライブラリの詳細については、[「互換性ライブラリの使用」347 ページ](#)を参照してください。

Embedded SQL アプリケーションのアップグレード手順

この項では、種類の異なる Embedded SQL アプリケーションのアップグレード手順について説明します。

- 組み込みデータベースを使用するアプリケーションは、パーソナル・データベース・サーバ (バージョン 5 では *dbeng50.exe*、バージョン 9 では *dbeng9.exe*) を使用するスタンドアロン・アプリケーション。
- クライアント／サーバ・アプリケーションは、ネットワークを介してネットワーク・データベース・サーバに接続する。バージョン 5 では、これらのアプリケーションは *dbclient.exe* 実行プログラムを使用します。

❖ **組み込みデータベースを使用する Embedded SQL アプリケーションをアップグレードするには、次の手順に従います。**

- 1 **ソフトウェアの現在のバージョンをインストールします。** 新しいバージョンのソフトウェアには、バージョン 5 アプリケーションを引き続き使用できるようにするコンポーネントが含まれています。現在のソフトウェアは、バージョン 5 ソフトウェアの上にインストールすることも、別のディレクトリにインストールすることもできます。

新しい実行ディレクトリは、システム・パスではバージョン 5 実行ディレクトリより前に置かれます。

- 2 **アプリケーションが互換性 DLL を使用していることを確認します。** 必要であれば、互換性ライブラリ `dbi50?.dll` を Adaptive Server Anywhere 実行ディレクトリからアプリケーションが見つけやすい場所にコピーしてください。

たとえば、アプリケーションのロード・モジュールと同じディレクトリに互換性ライブラリをコピーします。`dbi50?o.dll` ファイルは、Adaptive Server Anywhere 実行ディレクトリにインストールされます。このディレクトリは、互換性ライブラリがこのライブラリを必要とするときにパスになければなりません。

この段階では、バージョン 5 アプリケーションは引き続き従来どおり機能します。ただし、直接バージョン 5 インタフェース・ライブラリからではなく、互換性ライブラリを介してデータベースに接続することになります。

この段階で問題がある場合、アプリケーションがインタフェース・ライブラリをどのようにして見つけるかをチェックする必要があります。詳細については、「[ファイル・ロケーションと互換性ライブラリ](#)」351 ページを参照してください。

- 3 **新しい接続記述を作成します。** アプリケーションが設定ファイル、バッチ・ファイル、またはシステム・レジストリから接続パラメータを取得する場合は、現在のバージョンのデータベース・サーバを使用した新しい記述を準備してください。たとえば、Sybase Central は接続記述をシステム・レジストリに保管します。

- DBF パラメータを使用してデフォルト・データベース・サーバを起動する接続文字列を保管している場合、バージョン 5 データベース・サーバではなく、新しいデータベース・サーバが互換性ライブラリによって自動的に起動されます。この場合には、新しい接続記述は必要ありません。
- *dbeng50.exe* を指定した *StartLine* パラメータを使用する接続文字列を保管している場合、*dbeng9.exe* を指定したものと置き換えます。アプリケーションにハード・コードの接続文字列が含まれている場合、この時点で追加の手順が必要になります。

StartLine パラメータのアップグレードについては、「[開始パラメータと互換性ライブラリ](#)」352 ページを参照してください。

- データベース・サーバが、バッチ・ファイルや Windows NT サービスの使用などその他の方法で起動される場合、バージョン 5 ではなく、新しいバージョンのデータベース・サーバが起動するように再設定します。

- 4 **新しい接続記述を使用します。** この手順では、現在のバージョンのソフトウェアをすべて使用していて、アップグレードは完了しています。データベース自体は、既存のアプリケーションで機能するようにアップグレードする必要はありません。

❖ Embedded SQL クライアント／サーバ・アプリケーションをアップグレードするには、次の手順に従います。

- 1 **アップグレードの準備を行います。** この手順は、クライアント・マシンごとに実行します。ソフトウェアの現在のバージョンをインストールして、アップグレードの準備を行います。

新しい実行ディレクトリが、システム・パスのバージョン 5 実行ディレクトリより前に置かれていることを確認してください。

- 2 アプリケーションが互換性 DLL を使用していることを確認します。** この手順は、クライアント・マシンごとに実行します。必要であれば、互換性ライブラリ *dbi50?.dll* を Adaptive Server Anywhere 実行ディレクトリからアプリケーションが見つけやすい場所にコピーしてください。

たとえば、アプリケーションのロード・モジュールと同じディレクトリに互換性ライブラリをコピーします。*dbi50?.dll* ファイルは、Adaptive Server Anywhere 実行ディレクトリにインストールされます。このディレクトリは、互換性ライブラリがこのライブラリを必要とするときにパスになければなりません。

この段階では、バージョン 5 アプリケーションは引き続き従来どおり機能します。ただし、直接バージョン 5 インタフェース・ライブラリからではなく、互換性ライブラリを介してデータベースに接続することになります。

この段階で問題がある場合、アプリケーションがインタフェース・ライブラリをどのようにして見つけるかをチェックする必要があります。詳細については、「[ファイル・ロケーションと互換性ライブラリ](#)」351 ページを参照してください。

- 3 新しい接続記述を作成します。** この手順は、クライアント・マシンごとに実行します。アプリケーションが設定ファイル、バッチ・ファイル、またはシステムのレジストリから接続パラメータを取得する場合は、新しいバージョンのデータベース・サーバを使用する新しい記述を準備してください。この記述は、サーバをアップグレードするときに使用されます。

dbclient.exe を指定した StartLine 接続パラメータを使用する接続文字列を保管している場合、これを新しいものと置き換えます。新しい接続記述にはすべての *dbclient* 情報がパラメータ・セットとして含まれるようにしてください。アプリケーションにハード・コードの接続文字列が含まれている場合、この時点で追加の手順が必要になります。

StartLine パラメータのアップグレードについては、「[開始パラメータと互換性ライブラリ](#)」352 ページを参照してください。

dbclient コマンド情報を取得する接続記述の作成方法については、「[dbclient コマンド情報の取得](#)」354 ページを参照してください。

4 データベース・サーバをアップグレードします。 この手順はサーバ・マシンで実行します。

- すべてのソフトウェア・アップグレードに共通することですが、アップグレードする前に、データベースをバックアップします。
- サーバ・マシンに Adaptive Server Anywhere をインストールします。
- データベースで新しいデータベース・サーバを起動します。

5 新しい接続記述を使用します。 この手順は、クライアント・マシンごとに実行します。新しいバージョンのデータベース・サーバに接続するには、新しい接続記述を使用する必要があります。この手順では、現在のバージョンのソフトウェアをすべて使用していて、アップグレードは完了しています。データベース自体は、既存のアプリケーションで機能するようにアップグレードする必要はありません。

Adaptive Server Anywhere でバージョン 5 ユーティリティを使用する方法

バージョン 5 データベース・ユーティリティの場合、接続文字列は対話形式で入力されます。Interactive SQL などのバージョン 5 データベース・ユーティリティは、次の順序でインタフェース・ライブラリを検索する Embedded SQL アプリケーションです。

1. 現在のディレクトリ
2. バージョン 5 実行ディレクトリ
3. システム・パス

これらのアプリケーションの場合、互換性ライブラリがシステム・パスでバージョン 5 インタフェース・ライブラリより前にあっても、バージョン 5 互換性ライブラリが検出されます。

❖ **Adaptive Server Anywhere バージョン 9 でバージョン 5 ユーティリティを使用するには、次の手順に従います。**

- 1 インタフェース・ライブラリ・ファイルのバックアップ・コピーを作成します。
- 2 互換性ライブラリをバージョン 9 実行ディレクトリからバージョン 5 ディレクトリにコピーします。たとえば、Windows 95 と Windows NT の場合、*db150t.dll* ファイルをバージョン 9 インストール環境の *win32* サブディレクトリからバージョン 5 インストール環境の *win32* サブディレクトリにコピーします。
- 3 これで、バージョン 5 ユーティリティはバージョン 5 とバージョン 9 の両方のデータベース・サーバで実行できるようになります。

ODBC アプリケーションのアップグレード

バージョン 5 ODBC アプリケーションは、次の方法でアップグレードできます。

- バージョン 5 ODBC データ・ソースを現在の ODBC データ・ソースと置き換える。この方法は完全なアップグレードであり、この項では、この方法について説明します。
- 互換性ライブラリを使用して、現在のバージョンのデータベース・サーバに接続する。この方法を選択した場合、アプリケーションは引き続きバージョン 5 ODBC ドライバを使用するので、これは完全なアップグレードではありません。この方法でアップグレードする手順は、Embedded SQL アプリケーションの場合と同じです。

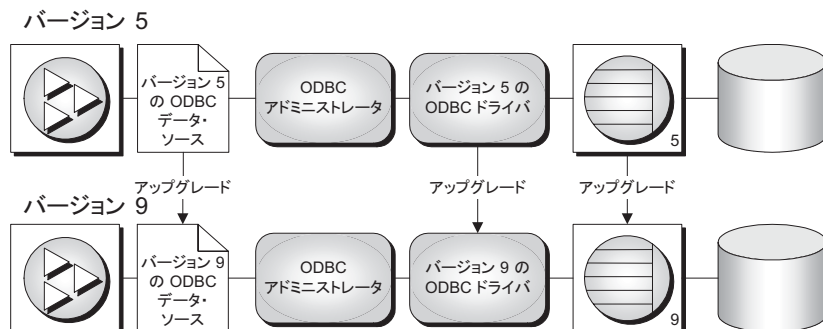
この方法でのアップグレードについては、[「Embedded SQL アプリケーションのアップグレード」333 ページ](#)を参照してください。

StartLine パラメータ

ODBC データ・ソースに、バージョン 5 スタンドアロン・エンジン (*dbeng50.exe*) または SQL Anywhere クライアント (*dbclient.exe*) を指定した Startline が含まれている場合は、互換性ライブラリを使用してアップグレードすることはできません。

アップグレードされたコンポーネント

次の図は、ODBC スタンドアロン・アプリケーションをアップグレードするときに行うべき変更を示しています。



データ・ソース

ODBC データ・ソースは、どの ODBC ドライバを使用するかを指定します。ODBC データ・ソースを Adaptive Server Anywhere バージョン 9 データ・ソースとして作成する場合には、バージョン 9 ODBC ドライバを使用します。

データ・ソースは最新のバージョンが必要

ODBC アプリケーションには、ODBC ドライバと同じバージョンの ODBC データ・ソースが必要です。

ODBC アプリケーションのアップグレード手順

この項では、種類の異なるいくつかの ODBC アプリケーションのアップグレード手順について説明します。

- 組み込みデータベースを使用するアプリケーションは、パーソナル・データベース・サーバ (バージョン 5 では *dbeng50.exe*、バージョン 9 では *dbeng9.exe*) を使用するスタンドアロン・アプリケーション。
- クライアント／サーバ・アプリケーションは、ネットワークを介してネットワーク・データベース・サーバに接続する。バージョン 5 では、これらのアプリケーションは *dbclient.exe* 実行プログラムを使用します。

アプリケーションによっては、使用する ODBC データ・ソース名の変更が可能です。一方、固定したデータ・ソース名を使用するアプリケーションもあります。どちらのアプリケーションもアップグレードできます。

❖ 組み込みデータベースを使用する ODBC アプリケーションをアップグレードするには、次の手順に従います。

- 1 **現在のバージョンのソフトウェアをインストールします。** ソフトウェアの現在のバージョンには、バージョン 5 ODBC アプリケーションが引き続き動作できるようにするコンポーネントが含まれています。現在のソフトウェアは、バージョン 5 ソフトウェアの上にインストールすることも、別のディレクトリにインストールすることもできます。

このインストールによってアプリケーションが影響を受けることはありません。

- 2 **バージョン 9 ODBC データ・ソースを作成します。** どのような変更を行うかは、データ・ソースで使用する接続パラメータによって異なります。

- DBF パラメータを使用してデフォルト・データベース・サーバを起動する場合、新しいデータ・ソースに古いデータ・ソースと同じ接続パラメータを使用できます。
- *dbeng50.exe* を指定した StartLine パラメータを使用する接続文字列を保管している場合、*dbeng9.exe* を指定したものと置き換えます。

StartLine パラメータのアップグレードについては、「[開始パラメータと互換性ライブラリ](#)」352 ページを参照してください。

- データベース・サーバが、バッチ・ファイルや Windows NT サービスの使用などその他の方法で起動される場合、バージョン 5 ではなく、新しいバージョンのデータベース・サーバが起動するように再設定します。

データ・ソースの作成については、『ASA データベース管理ガイド』> 「ODBC データ・ソースの使用」を参照してください。

- 3 新しいデータ・ソースを使用します。** この手順では、現在のバージョンのソフトウェアをすべて使用していて、アップグレードは完了しています。データベース自体は、既存のアプリケーションで機能するようにアップグレードする必要はありません。

アプリケーションによっては、データ・ソース名がハード・ワイヤされている場合があります。この場合には、バージョン 5 データ・ソースを同じ名前のバージョン 9 データ・ソースと置き換える必要があります。バージョン 5 データ・ソースを削除するのではなく、名前を変更することをおすすめします。

❖ ODBC クライアント／サーバ・アプリケーションをアップグレードするには、次の手順に従います。

- 1 アップグレードの準備を行います。** この手順は、クライアント・マシンごとに実行します。ソフトウェアの現在のバージョンをインストールして、アップグレードの準備を行います。

このインストールによってアプリケーションが影響を受けることはありません。

- 2 バージョン 9 ODBC データ・ソースを作成します。** この手順は、クライアント・マシンごとに実行します。どのような変更を行うかは、データ・ソースで使用する接続パラメータによって異なります。

- DBF パラメータを使用してデフォルト・データベース・サーバを起動する場合、新しいデータ・ソースに古いデータ・ソースと同じ接続パラメータを使用できます。
- *dbclient.exe* を指定した StartLine 接続パラメータを使用する接続文字列を保管している場合、これを新しいものと置き換えます。新しい接続記述にはすべての *dbclient* 情報がパラメータ・セットとして含まれるようにしてください。

さい。アプリケーションにハード・コードの接続文字列が含まれている場合、この時点で追加の手順が必要になります。

StartLine パラメータのアップグレードについては、「[開始パラメータと互換性ライブラリ](#)」352 ページを参照してください。

dbclient コマンド情報を取得する接続記述の作成方法については、「[dbclient コマンド情報の取得](#)」354 ページを参照してください。

- データベース・サーバが、バッチ・ファイルや Windows NT サービスの使用などその他の方法で起動される場合、バージョン 5 ではなく、新しいバージョンのデータベース・サーバが起動するように再設定します。

バージョン 9 データ・ソースの作成については、『ASA データベース管理ガイド』>「ODBC データ・ソースの使用」を参照してください。

3 データベース・サーバをアップグレードします。 この手順はサーバ・マシンで実行します。

- すべてのソフトウェア・アップグレードに共通することですが、アップグレードする前に、データベースをバックアップします。
- サーバ・マシンに Adaptive Server Anywhere をインストールします。
- データベースでバージョン 9 データベース・サーバを起動します。

4 新しいデータ・ソースを使用します。 この手順は、クライアント・マシンごとに実行します。新しいバージョンのデータベース・サーバに接続するには、新しい接続記述を使用する必要があります。この手順では、現在のバージョンのソフトウェアをすべて使用していて、アップグレードは完了しています。データベース自体は、既存のアプリケーションで機能するようにアップグレードする必要はありません。

PowerBuilder ユーザと InfoMaker ユーザのためのアップグレードに関する注意事項

Sybase PowerBuilder と InfoMaker のユーザは、Adaptive Server Anywhere バージョン 9 のすべての機能を使用できるようにするためにいくつかの変更を行ってください。

pbodb80.ini ファイル

PowerBuilder と InfoMaker では、ODBC データ・ソース情報を保持するのに、*pbodb80.ini* という名前のファイルを使用します。ファイル名にある 80 は、使用しているバージョンによって異なります。このファイルは、各 ODBC ドライバに、DDL 構文、デフォルト DBParm オプション、有効な機能名と特殊データ型などを提供します。

使用している *pbodb80.ini* ファイルに Adaptive Server Anywhere セクションがない場合、PowerBuilder と InfoMaker はデフォルトでコア構文を使用します。このため、これらのツールを使用して実行できる操作は制限されます。たとえば、プライマリ・キーと外部キーの作成、変更、削除はできません。

pbodb80.ini ファイルのアップグレード

PowerBuilder と InfoMaker のすべての機能を使用できるようにするには、*pbodb80.ini* ファイルをアップグレードする必要があります。

❖ *pbodb80.ini* ファイルをアップグレードするには、次の手順に従います。

- 1 既存の *pbodb80.ini* ファイルのバックアップ・コピーを作成します。
- 2 既存の Sybase SQL Anywhere セクションと同じ情報が入っているファイルの作業用コピーに、次の Adaptive Server Anywhere セクションを追加します。

```
[Adaptive Server Anywhere]
PBSyntax='WATCOM50_SYNTAX'
PBDateTime='STANDARD_DATETIME'
PBFunctions='WATCOM_FUNCTIONS'
PBDefaultValues='autoincrement,current
date,current time,current
timestamp,timestamp,null,user'
PBDefaultCreate='YES'
PBDefaultAlter='YES'
```

```
PBDefaultExpressions='YES'
DelimitIdentifier='YES'
PBDateTimeInvalidInSearch='NO'
PBTimeInvalidInSearch='YES'
PBQualifierIsOwner='NO'
PBSpecialDataTypes='WATCOM_SPECIALDATATYPES'
IdentifierQuoteChar='"'
PBSystemOwner='sys,dbf,rs_systabgroup'
PBUseProcOwner='YES'
SQLSrvrTSName='YES'
SQLSrvrTSQuote='YES'
SQLSrvrTSDelimit='YES'
ForeignKeyDeleteRule='Disallow if Dependent Rows
Exist (RESTRICT),Delete any Dependent Rows
(CASCADE),Set Dependent Columns to NULL (SET NULL)'
TableListType='GLOBAL TEMPORARY'
```

互換性ライブラリの使用

互換性ライブラリは、Embedded SQL アプリケーションがバージョン 5 とバージョン 9 の両方のデータベース・サーバで動作できるようにするダイナミック・ライブラリ (PC では DLL、UNIX では共有ライブラリ) です。この項では、互換性ライブラリがどのように機能するかを説明します。

この項の対象読者

SQL Anywhere バージョン 5 を Adaptive Server Anywhere バージョン 9 にアップグレードしていて、バージョン 9 サーバで稼働させる必要のある既存の Embedded SQL アプリケーションがある場合には、この項をお読みください。

バージョン 5 Embedded SQL インタフェース・ライブラリ

SQL Anywhere バージョン 5 アプリケーションを実行するすべてのクライアント・マシンには、ネットワーク接続であるか、パーソナル・サーバ接続であるかに関わらず、SQL Anywhere バージョン 5 インタフェース・ライブラリがあります。

Windows オペレーティング・システムでは、このライブラリは *dbi50t.dll* という名前の DLL です。

バージョン 5 クライアント・アプリケーションがインタフェース・ライブラリを見つける方法

バージョン 5 クライアント・アプリケーションは、次のいずれかの方法でインタフェース・ライブラリを見つけます。

- **ODBC アプリケーション** ODBC アプリケーションは SQL Anywhere ODBC ドライバを使用して SQL Anywhere データベースに接続します。SQL Anywhere バージョン 5 ODBC ドライバは、バージョン 5 Embedded SQL インタフェース・ライブラリの関数を呼び出します。

- **Embedded SQL アプリケーション** Windows オペレーティング・システムの Embedded SQL アプリケーションは、インタフェース・ライブラリを呼び出します。これらの呼び出しでは、インタフェース・ライブラリは名前 (Windows 95 と Windows NT では、*dbi50t.dll*) で参照されます。

SQL Anywhere バージョン 5 ODBC ドライバは、Embedded SQL アプリケーションです。

互換性ライブラリの使用

互換性ライブラリは、バージョン 9 のクライアント・ソフトウェアの一部として、オプションでインストールされます。これによって 2 つのインタフェース・ライブラリに対する同時サポートが提供されます。インストール環境をチェックして、互換性ライブラリがインストールされていることを確認してください。

セットアップ・プログラムは、アプリケーションがバージョン 5 インタフェース・ライブラリではなく互換性ライブラリを呼び出すことを確認する必要があります。

セットアップ・プログラムは、次の手順を実行して、アプリケーションがインタフェース・ライブラリを呼び出すことを確認します。

- 互換性ライブラリは、SQL Anywhere バージョン 5 インタフェース・ライブラリと同じ名前です。たとえば、Windows NT では、互換性ライブラリは *dbi50t.dll* という名前です。
- 互換性ライブラリは、他のバージョン 9 ソフトウェアと同じディレクトリにインストールされます。
- バージョン 9 インストール・ディレクトリは、システム・パスではバージョン 5 ディレクトリより前に置かれます。これによって、アプリケーションはバージョン 5 インタフェース・ライブラリより先に互換性ライブラリを見つけます。

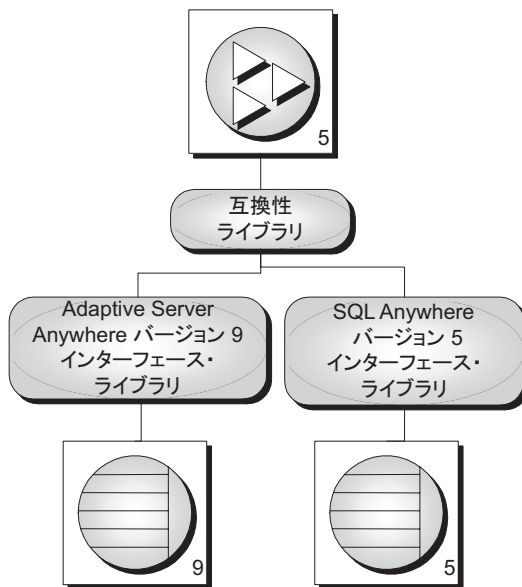
- バージョン 5 インタフェース・ライブラリは、互換性ライブラリと同じディレクトリにインストールされますが、名前は **dbi50to.dll** になります。互換性ライブラリは、バージョン 5 サーバにアクセスするときに、このインタフェース・ライブラリを呼び出します。
- バージョン 9 インタフェース・ライブラリは、互換性ライブラリと同じディレクトリにインストールされます。Windows NT では、名前は **dblib9.dll** です。互換性ライブラリは、バージョン 9 サーバにアクセスするときに、このインタフェース・ライブラリを呼び出します。

互換性ライブラリを使用して問題が発生した場合には、パス内のディレクトリの順序をチェックし、パス内でバージョン 9 がバージョン 5 より前に置かれていることを確認してください。

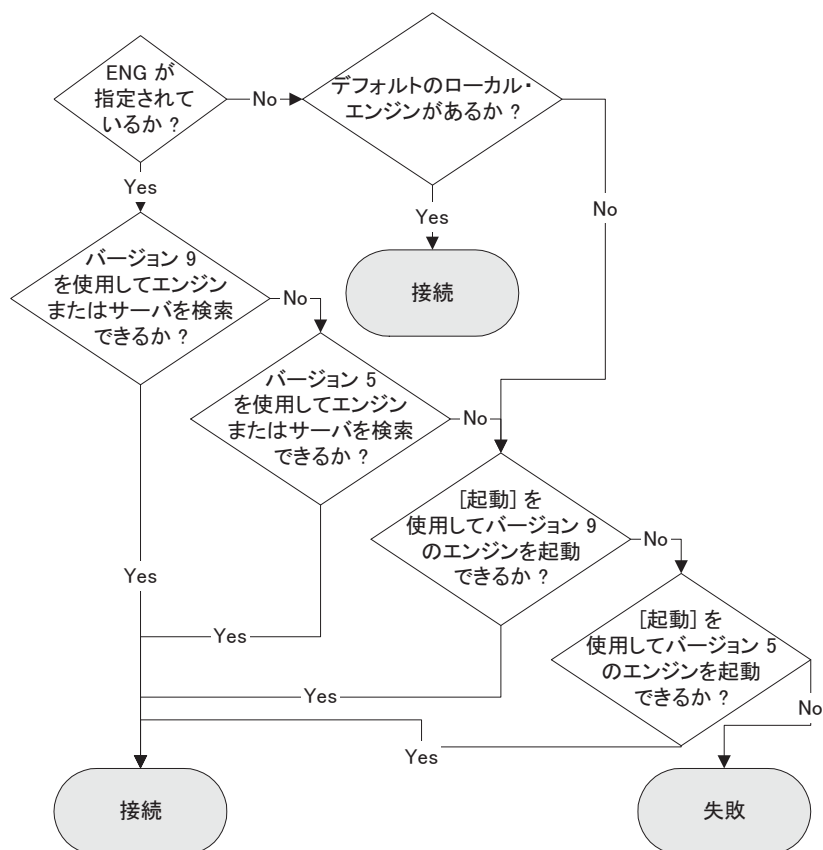
互換性ライブラリはどのように機能するか

互換性ライブラリは、与えられた接続文字列を使用して、Adaptive Server Anywhere バージョン 9 インタフェース・ライブラリを使用する Adaptive Server Anywhere に接続しようとします。これに失敗すると、SQL Anywhere バージョン 5 ライブラリを使用する SQL Anywhere データベースに接続しようとします。

次の図は、互換性ライブラリが SQL Anywhere バージョン 5 と Adaptive Server Anywhere バージョン 9 の両方のデータベース・サーバとの通信をどのようにして可能にするかを示しています。各囲み内の右下の数字は、ソフトウェア・コンポーネントのバージョン番号です。



次の図は、互換性ライブラリがサーバに接続するとき使用するアルゴリズムを示しています。



ファイル・ロケーションと互換性ライブラリ

互換性ライブラリとバージョン 5 インタフェース・ライブラリのファイル名は同じ (*dbi50.dll*) です。アプリケーションが互換性ライブラリを使用するには、DLL を検索するときに、バージョン 5 インタフェース・ライブラリよりも先に互換性ライブラリを見つけなければなりません。

アプリケーションが、バージョン 5 インタフェース・ライブラリよりも先に互換性ライブラリを確実に見つけるようにするには、アプリケーションがどのようにして DLL を検索するかを理解してください。

DLL の検索

バージョン 9 インストール・プログラムは、システム・パスでバージョン 5 ディレクトリよりも前にバージョン 9 実行ディレクトリを置くとはかぎりません。したがって、システム・パスで *dbi50.dll* を検索するアプリケーションは、バージョン 5 インタフェース・ライブラリよりも先に互換性ライブラリが検出されるように変更しなければならない場合もあります。

バージョン 5 ODBC ドライバは、バージョン 5 インタフェース・ライブラリと同じディレクトリにあるため、バージョン 9 ライブラリではなく、バージョン 5 インタフェース・ライブラリを探します。バージョン 5 ODBC アプリケーションが、バージョン 9 データベース・サーバに接続できるようにするには、バージョン 5 インターフェイス・ライブラリの名前を変更するか、バージョン 9 の互換性ライブラリと *dbi50to.dll* をコピーしてバージョン 5 ディレクトリに入れます。

どちらのライブラリが見つかったかを調べる方法

どちらのライブラリが見つかったかは、次の方法で調べられます。

- バージョン 9 データベース・サーバに接続を試みる。バージョン 5 インタフェース・ライブラリを使用していると、接続できません。
- 接続文字列に **StartLine** は指定しないで、データベース・ファイル・パラメータ (**DBF**) を指定する。バージョン 5 インタフェース・ライブラリが見つかった場合は、**SQL Anywhere** バージョン 5 スタンドアロン・データベース・エンジンが起動されます。互換性ライブラリが見つかった場合は、バージョン 9.0 パーソナル・データベース・サーバが起動されます。

開始パラメータと互換性ライブラリ

StartLine 接続パラメータが含まれている接続文字列を使用するアプリケーションでは、アップグレード中にいくつかの問題があります。

StartLine パラメータは、データベース・エンジンまたは **SQL Anywhere** クライアント実行プログラムの起動を明示的に指示します。サンプル **StartLine** パラメータは次のとおりです。

- **スタンドアロン・アプリケーション** バージョン 5 スタンドアロン・アプリケーションのサンプル StartLine パラメータは、次のとおりです。

```
dbeng50.exe -c 8M
```

- **ネットワーク・クライアント・アプリケーション** バージョン 5 ネットワーク・クライアント・アプリケーションのサンプル StartLine パラメータは、次のとおりです。

```
dbclient.exe -x tcpip
```

ソフトウェアの現在のバージョンでは、**dbeng50.exe** 実行プログラムに代わってパーソナル・データベース・サーバ **dbeng9.exe** が提供されています。**dbclient.exe** 実行プログラムは必要ありません。

❖ StartLine パラメータをアップグレードするには、次の手順に従います。

- アップグレード手順は、接続パラメータがどこに保管されているかによって異なります。
 - 接続パラメータがアプリケーションの外に保管されている場合には、**dbeng50.exe** ではなく適切な **dbeng9.exe** 実行プログラム名を使用するよう、接続パラメータを変更する必要があります。
- dbclient** コマンド・ライン上のすべての情報は、他の接続パラメータ用に書き直すことができます。詳細については、『ASA データベース管理ガイド』> 「接続パラメータとネットワーク・プロトコル・オプション」を参照してください。
- 接続パラメータが、アプリケーションにハード・ワイヤされている場合には、アプリケーションを新しい接続文字列でリンクし直します。

クライアント・コマンド・ラインと接続パラメータには、さまざまな設定が考えられます。どの設定の場合も、配備する前に徹底的にテストしてください。

dbclient コマンド情報の取得

データベース・サーバに接続するバージョン 5 クライアント・アプリケーションは、SQL Anywhere クライアントである *dbclient.exe* という名前の実行プログラムを介して接続します。クライアント実行コマンド・ラインには、サーバを検出するのに必要な次のような情報が含まれています。

- **デフォルト・サーバ名** クライアント・コマンド・ラインのサーバ名は、デフォルト・サーバ名です。クライアント実行プログラムが実行されているときは、アプリケーションはデフォルト・サーバに接続するためにサーバ名を指定する必要はありません。
- **ネットワーク・プロトコル・オプション** プロトコル・オプション・セットと一緒に使用するネットワーク・プロトコルのリストには、クライアント実行プログラムが、サーバを見つけるときにどこを調べるかが指定されています。
- **クライアント／サーバ通信チューニング** 最適なパフォーマンスを得るために、パラメータ・セットを使って、パケット・サイズ、バッファ・サイズなどをチューニングできます。

バージョン 9 では、この情報は、ほかの接続情報と一緒に ODBC データ・ソースに保管されます。クライアント実行プログラムがなくなったので、クライアント・コマンド・ラインはありません。バージョン 9 では、Embedded SQL アプリケーションは ODBC データ・ソースを接続パラメータのソースとして使用できます。

クライアント・コマンド・ラインのスコープ

バージョン 5 クライアント実行プログラムは、一度に 1 つだけ実行できます。また、複数のアプリケーションで使用でき、複数のサーバへの接続を処理します。したがって、コマンド情報はそのマシンのグローバル情報です。

クライアント・コマンド情報を取得する方法

バージョン 9 にアップグレードするときには、バージョン 5 *dbclient* コマンド情報を、バージョン 9 ODBC または Embedded SQL アプリケーションで使えるような方法で取得します。これは、次のいずれかの方法で取得します。

- **情報を ODBC データ・ソースに置く** 情報をデータ・ソースに置くことができれば、バージョン 9 ODBC または Embedded SQL アプリケーションで使用できます。
- **SQLCONNECT 環境変数を使用する** SQLCONNECT 環境変数には、接続文字列が含まれています。環境変数は、どの接続パラメータを使用するかを決めるプロセスの早い段階で検索されます。SQLCONNECT の設定値を使用して接続パラメータを無効にできます。

クライアント・コマンド情報を取得する方法は、情報がどこに入っているかによって異なります。

コマンド情報がある場所

既存のコマンド・ライン情報は次のいずれかの場所にあります。

- **ODBC データ・ソース** ODBC データ・ソースには、クライアント実行コマンドを保持できる START パラメータが入っています。
- **接続文字列として** アプリケーションでクライアント情報を (たとえば、初期化ファイルから) 取得して、その情報を接続文字列に START パラメータとして提供することができます。
- **バッチ・ファイル** 起動プロセスの一部としてクライアント実行コマンド・ラインが含まれているバッチ・ファイルを使用できます。
- **アイコンの下** デスクトップ画面のアイコンの下にあるクライアント実行コマンド・ラインを使用できます。

情報を取得する方法

- **ODBC データ・ソースから** ODBC データ・ソース・アップグレードでは、ODBC データ・ソース START パラメータ内の情報を取得します。
- **バッチ・ファイルから、またはアイコンの下** `dbclient.exe` パラメータを CommLinks 接続パラメータに移動できます。
- **ハード・ワイヤされた接続文字列** アプリケーション内にハード・ワイヤされた (つまり編集できない) 接続文字列がある場合は、アプリケーションのソースを変更して再コンパイルします。

データベースのアップグレード

Adaptive Server Anywhere のいくつかの新しい機能を使用するには、データベース・ファイルをアップグレードする必要があります。

警告

アップグレードの前に必ずデータベースのバックアップを行ってください。

サポートされるバージョン

データベースは、次に挙げるバージョンのソフトウェアから現在のバージョンのフォーマットにアップグレードできます。

- **SQL Anywhere バージョン 5** バージョン 5.0 と 5.5 を含む、すべてのパッチ・レベル。
- **Watcom SQL バージョン 4** すべてのパッチ・レベル。
- **Watcom SQL バージョン 3.2** パッチ・レベル **e** 以上のみ。

アップグレード [dbupgrad] ユーティリティの役割

アップグレード [dbupgrad] ユーティリティは次の作業を行います。

- 新しいシステム・テーブルを追加する
- 既存のシステム・テーブルに新しいカラムを追加する
- データベース・オプションを追加する

データベースをアップグレードする方法については、「[ソフトウェアとデータベースのアップグレード](#)」305 ページを参照してください。

バージョン 5 SQL Remote インストール環境のアップグレード

SQL Remote インストール環境には、1 つの統合データベースと多数のリモート・データベースが含まれています。また、各サイトには、Message Agent があります。

それぞれのサイトでは、Message Agent がメッセージの送受信を行います。メッセージは SQL 文の形式で、データベース・サーバがその SQL 文の実行を処理します。

SQL Remote のアップグレード要件は、次のとおりです。

- **データベースのアップグレードは不要** データベースをバージョン 9 用にアップグレードすることは必須ではありませんが、最高のパフォーマンスを実現するためには、データベースをアンロードしてから再ロードして、データベース・ファイル・フォーマットをアップグレードすることをおすすめします。すべてのデータベースを同時にアップグレードする必要はありません。

データベースをアンロードおよび再ロードする方法については、『SQL Remote ユーザーズ・ガイド』>「レプリケーションに参加しているデータベースのアンロードと再ロード」を参照してください。データベース・バージョンをアップグレードする方法については、「[データベースのアップグレード](#)」307 ページを参照してください。

- **ソフトウェア・アップグレードは一度に 1 サイトずつ行う** バージョン 5 Message Agent は、COMPRESSION データベース・オプションが -1 (マイナス 1) の値に設定されていれば、バージョン 9 Message Agent とメッセージを交換できます。インストール環境全体のソフトウェアを同時にアップグレードする必要はありません。

- **Message Agent とサーバは別々のアップグレードが可能**

Message Agent は Embedded SQL アプリケーションです。このため、互換性ライブラリを使用すれば、データベース・サーバを Message Agent より先にアップグレードできます。ただし、パフォーマンスの面からは、Message Agent をデータベース・サーバと同時にアップグレードすることをおすすめします。

Message Agent をデータベース・サーバより先にアップグレードすることはできません。新しいクライアント・アプリケーションはバージョン 5 サーバでは機能しないからです。

レプリケーションはトランザクション・ログに基づいていますが、データベースをアンロードして再ロードすると、古いトランザクション・ログが使えなくなります。このため、レプリケーションが関係するときは、バックアップをきちんと実行することが特に重要です。

例

以下に、アップグレードの一例を示します。

1. 統合データベース・サーバと Message Agent をアップグレードします。すべてのメッセージがリモート・サイトのバージョン 5 ソフトウェアと互換性を持つように、COMPRESSION データベース・オプションを -1 に設定してください。
2. リモート・データベース・サーバと Message Agent をアップグレードします。COMPRESSION データベース・オプションを -1 以外の値に設定すれば、統合データベース・サーバに送信されるメッセージに対して圧縮機能とエンコード機能を利用できます。
3. すべてのリモート・データベース・サーバと Message Agent をアップグレードし終わったら、統合サイトの COMPRESSION データベース・オプションを -1 以外の値に設定します。

索引

記号

- \$best_adapter_mac
 - 9.0.2 の新機能 33
 - \$best_adapter_name
 - 9.0.2 の新機能 33
 - \$best_ip
 - 9.0.2 の新機能 33
 - \$best_network_name
 - 9.0.2 の新機能 33
 - \$request_id
 - 9.0.2 の新機能 33
 - .NET
 - バージョン 9 の新機能 3
 - .NET Compact Framework
 - バージョン 9 の新機能 3
 - .NET のサポート
 - 8.0.2 の新機能 148
 - @@dbts グローバル変数
 - バージョン 7.0.0 の新機能 265
 - @@fetch_status グローバル変数
 - 8.0.2 の新機能 147
 - @@identity グローバル変数
 - バージョン 7.0.0 での動作の変更 272
 - バージョン 8.0.0 での Ultra Light の新機能 214
 - @filename オプション
 - バージョン 7.0.2 の新機能 239
 - _IP_CHANGED_
 - 9.0.2 の新機能 33
 - 1252SPA
 - バージョン 9.0.2 の新機能 22
 - 1252SWEFIN 照合
 - バージョン 9.0.2 の新機能 22
 - 1254TRKALT 照合
 - バージョン 9.0.1 の新機能 67
 - 1255HEB 照合
 - バージョン 9.0.2 の新機能 22
 - 1256ARA 照合
 - バージョン 9.0.2 の新機能 22
 - 3 層コンピューティング
 - バージョン 7.0.0 の新機能 256
 - 508 条
 - 準拠 187
 - 64 ビット・ソフトウェア
 - バージョン 9.0.0 の新機能 95
 - 874THAIBIN
 - バージョン 9.0.2 の新機能 22
 - 9.0.2 で廃止される Ultra Light.NET コンポーネント・インタフェース 53
 - 950TWN 照合
 - バージョン 9.0.2 での廃止 48
 - 950ZHO_HK
 - バージョン 9.0.2 の新機能 22
 - 950ZHO_TW 照合
 - バージョン 9.0.2 の新機能 22
- ## A
- a_db_info 構造体
 - 8.0.2 の新機能 147
 - ActiveSync
 - バージョン 8.0.0 での Mobile Link の新機能 203
 - バージョン 8.0.0 での Ultra Light の新機能 209
 - ActiveSync プロバイダのアクティビティ・ログ
 - 9.0.1 の新機能 74
 - Adaptive Server Anywhere
 - SMB サーバ 2
 - サンプル 13
 - バージョン 9 の新機能の概要 5

Adaptive Server Anywhere コンソール・ユーティリティ [dbconsole]
バージョン 9.0.1 の強化 66

Adaptive Server Anywhere の新機能
バージョン 9.0.2 18

Adaptive Server Anywhere の動作の変更
バージョン 9.0.2 47

ADO.NET
バージョン 9.0.2 の Ultra Light サポート 35
バージョン 9 の新機能 3

AGENT 接続パラメータ
バージョン 6.0.3 での動作の変更 287
バージョン 8.0.1 で使用されなくなった機能 175

ALTER DATABASE 文
8.0.2 の新機能 143
バージョン 8.0.0 の新規調整機能 184

ALTER DBSPACE 文
8.0.2 の新機能 146

ALTER EVENT 文
バージョン 7.0.0 の新機能 254

ALTER DOMAIN 文
バージョン 9.0.2 の新機能 20

ALTER INDEX 文
9.0.0 の新機能 100

ALTER TABLE 文
バージョン 6.0.3 の新機能 277
バージョン 9.0.1 の強化 61

ALTER VIEW 文
バージョン 9.0.1 の強化 61

ANSI_NULLS オプション
バージョン 9.0.1 の強化 61

ANSI_UPDATE_CONSTRAINTS オプション
バージョン 7.0.0 の新機能 270

ANSINULL オプション
バージョン 6.0.1 での動作の変更 302

AppForge Crossfire
バージョン 9.0.2 での Ultra Light の動作の変更 55

AppForge CrossFire
バージョン 9.0.2 の新機能 36

AppForge MobileVB

8.0.2 の Ultra Light コンポーネントの新機能 152

AppInfo プロパティ
バージョン 7.0.0 の新機能 262

ASANYSH8 環境変数
バージョン 8.0.0 の新機能 202

ASSUME_DISTINCT_SERVERS オプション
バージョン 9.0.2 での廃止 47

ASTART 接続パラメータ
バージョン 6.0.1 の新機能 298

ASTMP 環境変数
バージョン 6.0.3 の新機能 278

authenticate_parameters
バージョン 9.0.0 の新機能 116

AutoStart 接続パラメータ
バージョン 6.0.1 の新機能 298

AWE キャッシュ
バージョン 9.0.0 の新機能 105

B

backlog プロトコル・オプション
9.0.2 の新機能 30

BACKUP 文
8.0.1 の新機能 169
8.0.2 の新機能 141
バージョン 6.0.2 の新機能 290
バージョン 6.0.3 の新機能 278
バージョン 7.0.0 の新機能 256
バージョン 7.0.1 の新機能 246
バージョン 9.0.1 の強化 65
バージョン 9.0.1 の変数の強化サポート 60
バージョン 9.0.2 の新機能 25

BASE64_DECODE 関数
バージョン 9.0.1 の新機能 59

BASE64_ENCODE 関数
バージョン 9.0.1 の新機能 59

BCP フォーマット
バージョン 7.0.0 の新機能 264

begin_publication
バージョン 9.0.0 の新機能 116

- BlankPadding プロパティ
バージョン 7.0.3 の新機能 232
- BLOB
8.0.2 での Mobile Link [dbmlsync] の変更点 160
- BlobArenas プロパティ
バージョン 8.0.0 の新機能 194
- BLOCKING_TIMEOUT オプション
バージョン 8.0.0 の新機能 190
- buffer_size ストリーム・パラメータ
8.0.2 での Mobile Link [HTTP、HTTPS] の新機能 151
- b オプション
9.0.0 での Mobile Link [dbmlsrv9] の新機能 116
- ## C
- C++ インタフェースの混在
9.0.1 の Ultra Light の新機能 78
- C2 プロパティ
バージョン 7.0.3 の新機能 232
- Capabilities プロパティ
バージョン 8.0.0 の新機能 194
- CaseSensitivePasswords プロパティ
バージョン 9.0.0 の新機能 107
- CaseSensitive プロパティ
バージョン 7.0.3 の新機能 232
- CASE 式
9.0.2 の Ultra Light の新機能 39
- CASE 文
バージョン 8.0.0 の新機能 185
- cc オプション
バージョン 9.0.1 のデータベース・サーバの新機能 63
- Certicom
2002 年 7 月で証明書の発行終了 159
8.0.2 の新機能 144
- CHAR_OEM_TRANSLATION オプション
バージョン 6.0.1 での動作の変更 303
- CharSet 接続パラメータ
バージョン 6.0.3 の新機能 279
- Checksum プロパティ
バージョン 9.0.1 の新機能 62
- ClientPort プロパティ
バージョン 9.0.2 の新機能 27
- CodeWarrior
サポートされるバージョン 136
バージョン 8.0.0 でサポートされていないバージョン 5 228
バージョン 8.0.0 での Ultra Light の新機能 212
バージョン 8.0.1 でサポートされるバージョン 8 171
- CodeWarrior ステーションナリ
9.0.1 の Ultra Light C++ コンポーネントの新機能 79
- CodeXchange
Adaptive Server Anywhere サンプル 13
Mobile Link サンプル 12
Ultra Light サンプル 11
説明 11
- com.sybase.asa.ultralite.jdbc パッケージ
バージョン 8.0.0 では名前変更 229
- COMMENT 文
バージョン 8.0.0 での動作の変更 221
- COMMIT 文
9.0.1 の Ultra Light 動的 SQL の新機能 77
- CommLinks 接続パラメータ
バージョン 6.0.1 での動作の変更 302
バージョン 8.0.0 での動作の変更 223
- COMPARE 関数
バージョン 8.0.0 の新機能 186
- CompressedBTrees プロパティ
バージョン 8.0.0 の新機能 194
- COMPRESS 関数
バージョン 9.0.1 の新機能 59
- conduit の設定
8.0.1 での Ultra Light の新機能 172
- CONFLICT 関数
バージョン 9.0.2 の SQL Remote の新機能 35
バージョン 9.0.2 の新機能 18
- CONNECTION_PROPERTY 関数
バージョン 9.0.2 での動作の変更 50

- ConnectionName 接続パラメータ
 - バージョン 7.0.2 の新しい ODBC 機能 237
 - ConsoleLogFile プロパティ
 - バージョン 9.0.1 の新機能 71
 - CONVERT 関数
 - バージョン 6.0.3 の新機能 281
 - バージョン 8.0.0 の新機能 222
 - CON 接続パラメータ
 - バージョン 7.0.2 の新しい ODBC 機能 237
 - CORR 関数
 - バージョン 9.0.1 の新機能 58
 - countUploadRows メソッド (JdbcConnection クラス)
 - 8.0.1 での Ultra Light の新機能 171
 - COVAR_POP 関数
 - バージョン 9.0.1 の新機能 58
 - COVAR_SAMP 関数
 - バージョン 9.0.1 の新機能 58
 - CPU
 - バージョン 9.0.0 での動作の変更 126
 - CREATE DATABASE 文
 - バージョン 9.0.0 の新機能 107
 - バージョン 9.0.1 の強化 62
 - CREATE EVENT 文
 - 8.0.2 での動作変更 157
 - バージョン 7.0.0 の新機能 254
 - CREATE FUNCTION 文
 - 8.0.1 の新機能 169
 - CREATE INDEX 文
 - 9.0.1 の Ultra Light 動的 SQL の新機能 77
 - CREATE SYNCHRONIZATION DEFINITION 文
 - バージョン 8.0.0 では廃止 227
 - CREATE SYNCHRONIZATION SITE 文
 - バージョン 8.0.0 では廃止 227
 - CREATE SYNCHRONIZATION SUBSCRIPTION 文
 - 8.0.2 でのスケジュールの変更 159
 - バージョン 8.0.0 での Mobile Link の新機能 202
 - CREATE SYNCHRONIZATION TEMPLATE 文
 - バージョン 8.0.0 では廃止 227
 - CREATE SYNCHRONIZATION USER 文
 - 8.0.2 でのスケジュールの変更 159
 - バージョン 8.0.0 での Mobile Link の新機能 202
 - CREATE TABLE 文
 - 9.0.1 の Ultra Light 動的 SQL の新機能 77
 - CrossFire 36
 - cr オプション
 - バージョン 9.0.1 のデータベース・サーバの新機能 63
 - CSCONVERT 関数
 - 8.0.2 の新機能 142
 - CS 接続パラメータ
 - バージョン 6.0.3 の新機能 279
 - ct オプション
 - バージョン 8.0.0 での動作の変更 201
 - CUBE 演算
 - バージョン 9.0.1 の新機能 58
 - CUME_DIST 関数
 - バージョン 9.0.1 の新機能 58
 - CURRENT_TIMESTAMP
 - バージョン 9.0.1 での動作の変更 85
 - CURRENT_USER
 - バージョン 9.0.1 での動作の変更 85
 - CURRENT UTC TIMESTAMP
 - バージョン 8.0.0 の新機能 186
 - custom_header プロトコル・オプション
 - 9.0.2 の Ultra Light の新機能 41
 - 9.0.2 の新機能 32
 - cv オプション
 - バージョン 9.0.1 のデータベース・サーバの新機能 63
 - cw オプション
 - バージョン 8.0.0 の新機能 193
- ## D
- DATE_FORMAT オプション
 - バージョン 8.0.0 での新しい動作 222
 - DATEDIFF 関数

- バージョン 6.0.3 での動作の変更 286
- DATEPART 関数
 - バージョン 7.0.2 での動作の変更 238
- DB_EXTENDED_PROPERTY 関数
 - バージョン 9.0.2 での動作の変更 50
- db_get_property 関数
 - バージョン 8.0.0 の新機能 190
- db_locate_servers_ex 関数
 - バージョン 9.0.2 の新機能 22
- db_locate_servers 関数
 - バージョン 7.0.1 の新機能 245
- DB_PROPERTY 関数
 - バージョン 9.0.2 での動作の変更 50
- EVENT_PARAMETER 関数
 - バージョン 9.0.2 での動作の変更 50
- db_string_ping_server 関数
 - バージョン 7.0.0 の新機能 264
- dbauser オプション
 - 9.0.2 で廃止された QAnywhere オプション 55
- dbbackup ユーティリティ
 - バージョン 9.0.1 の強化 65
- dbclient
 - アップグレード 354
- dbclient ユーティリティ
 - バージョン 7.0.0 では使用不可 270
- dbcollat ユーティリティ
 - バージョン 8.0.0 での動作の変更 220
- dbconsole ユーティリティ
 - バージョン 9.0.1 の強化 66
- DBD::ASAny
 - バージョン 9.0.1 の新しい Perl ドライバ 64
- dbdsn ユーティリティ
 - バージョン 8.0.0 での動作の変更 224
 - バージョン 9.0.2 での動作の変更 48
 - バージョン 9.0.2 の新機能 24
- dbeng オプション
 - 9.0.2 で廃止された QAnywhere オプション 55
- dberase ユーティリティ
 - バージョン 9.0.2 での動作の変更 48
- dbfhide ユーティリティ
 - 8.0.2 の新機能 144
 - バージョン 9.0.2 の新機能 26
- DBFileFragments プロパティ
 - 8.0.1 の新機能 168
 - バージョン 8.0.0 の新機能 194
- dbfile オプション
 - 9.0.2 で廃止された QAnywhere オプション 55
- dbinfo ユーティリティ
 - バージョン 8.0.0 の新機能 200
- dbinit ユーティリティ
 - バージョン 8.0.0 での動作の変更 217
 - バージョン 9.0.0 の新機能 107
 - バージョン 9.0.1 の強化 62
- dbisqlg.exe
 - バージョン 9.0.0 の新機能 113
- dbisql ユーティリティ
 - バージョン 8.0.0 の新機能 187
- dbl50t.dll
 - ロケーション 351
- dblic ユーティリティ
 - バージョン 8.0.0 の新機能 200
- dblocate ユーティリティ
 - バージョン 9.0.1 の強化 66
- dblog ユーティリティ
 - バージョン 9.0.2 での動作の変更 48
- dbmlsrv9
 - バージョン 7.0.1 の新機能 248
 - バージョン 8.0.0 での動作の変更 227
- dbmlstop ユーティリティ
 - バージョン 8.0.0 での Mobile Link の新しい動作 205
 - バージョン 8.0.0 での動作の変更 227
 - バージョン 8.0.1 での動作の変更 176
- dbmlsync 統合コンポーネント
 - 9.0.1 の新機能 74
- dbmlsync ユーティリティ
 - バージョン 7.0.0 の新機能 267
 - バージョン 7.0.1 の新機能 246
 - バージョン 7.0.1 のプログラミング・インタフェース 249
 - バージョン 8.0.0 での Mobile Link の新機能 203
- dbmluser ユーティリティ
 - バージョン 8.0.0 での動作の変更 227

- dbname オプション
 - 9.0.2 で廃止された QAnywhere オプション 55
- dbping ユーティリティ
 - バージョン 8.0.0 の新機能 197
- dbremote
 - バージョン 9.0.2 の新機能 34
- dbspaces
 - バージョン 9.0.2 での動作の変更 50
- dbsvc ユーティリティ
 - バージョン 7.0.1 の新機能 244
 - バージョン 8.0.0 の新機能 196
 - バージョン 9.0.2 での動作の変更 48
 - バージョン 9.0.2 の新機能 23
- DBTools インタフェース
 - バージョン 9.0.2 での libdbtool9.so の廃止 48
- dbtran ユーティリティ
 - バージョン 9.0.2 での動作の変更 48
- dbuleng9
 - 9.0.1 の Ultra Light の新機能 79
- dbunload ユーティリティ
 - バージョン 9.0.1 の強化 65
- dbvalid ユーティリティ
 - バージョン 8.0.0 の新機能 191
 - バージョン 9.0.1 での動作の変更 86
 - バージョン 9.0.1 の強化 62
- dbxtract ユーティリティ
 - バージョン 7.0.0 の新機能 266
 - バージョン 9.0.2 での動作の変更 52
- dc オプション
 - 9.0.1 の新機能 dbmsync 74
- DDE プロトコル
 - バージョン 7.0.0 ではサポート対象外 269
- DEALLOCATE 文
 - 8.0.2 の新機能 142
- DEBUG_MESSAGES オプション
 - バージョン 9.0.1 の強化 61
- DEBUG 接続パラメータ
 - バージョン 7.0.1 の新機能 245
 - バージョン 8.0.1 で使用されなくなった機能 174
- DECL_FIXCHAR マクロ
 - バージョン 7.0.0 での動作の変更 272
- DECL_LONGBINARY マクロ
 - バージョン 8.0.0 での Ultra Light の新機能 215
- DECL_LONGVARCHAR マクロ
 - バージョン 8.0.0 での Ultra Light の新機能 215
- DECLARE CURSOR 文
 - バージョン 7.0.0 の新機能 262
- DECLARE LOCAL TEMPORARY TABLE 文
 - バージョン 9.0.1 での動作の変更 86
- DECOMPRESS 関数
 - バージョン 9.0.1 の新機能 59
- DECRYPT 関数
 - バージョン 9.0.1 の新機能 59
- DEDICATED_TASK オプション
 - バージョン 9.0.1 の新機能 67
- DEFAULT_ISQL_ENCODING オプション
 - バージョン 9.0.1 の新機能 67
- DefaultCollation プロパティ
 - バージョン 6.0.3 の新機能 279
- DEFAULT UTC TIMESTAMP
 - バージョン 8.0.0 の新機能 186
- DELETE_OLD_LOGS オプション
 - バージョン 7.0.2 の新機能 237
- DeleteAllRows 関数
 - バージョン 8.0.0 での Ultra Light の新機能 214
- DENSE_RANK 関数
 - バージョン 9.0.1 の新機能 58
- DOS
 - バージョン 8.0.0 でサポートされていない Ultra Light プラットフォーム 228
- DOUBLE データ型
 - バージョン 6.0.3 での Mobile Link の新機能 285
 - バージョン 6.0.3 での Ultra Light の新機能 285
- download_delete_cursor
 - 8.0.2 での新しい動作 160
- DownloadOnly dbmsync 拡張オプション
 - 9.0.0 での Mobile Link の新機能 117
- Driver 接続パラメータ
 - バージョン 9.0.2 の新機能 24

- DriveType プロパティ
バージョン 9.0.1 の新機能 71
- DROP INDEX 文
9.0.1 の Ultra Light 動的 SQL の新機能 77
- DROP TABLE 文
9.0.1 の Ultra Light 動的 SQL の新機能 77
- ds dbmlsync 拡張オプション
9.0.0 での Mobile Link の新機能 117
- ds オプション
9.0.0 での Mobile Link [dbmlsync] の新機能 117
- ## E
- ec オプション
8.0.2 の新機能 144
バージョン 9.0.2 でのサーバ・オプションの動作の変更 49
- ek オプション
9.0.2 で廃止された QAnywhere オプション 55
- Embedded SQL
Ultra Light 9.0.2 での廃止 53
- eMbedded Visual Basic
8.0.2 の Ultra Light コンポーネントの新機能 152
- eMbedded Visual C++
バージョン 8.0.0 での Ultra Light の新機能 212
- Encryption 接続パラメータ
8.0.2 の新機能 144
- ENCRYPT 関数
バージョン 9.0.1 の新機能 59
- ENC 接続パラメータ
8.0.2 の新機能 144
- end_publication
バージョン 9.0.0 の新機能 116
- ERRORMSG 関数
バージョン 8.0.0 の新機能 186
- ESTIMATE_SOURCE 関数
バージョン 8.0.0 での新しい動作 224
- EVENT_PARAMETER 関数
バージョン 7.0.2 の新機能 238
- EXCEPT 演算
バージョン 9.0.0 の新機能 95
- EXECUTE IMMEDIATE 文
9.0.0 の拡張 98
バージョン 9.0.1 での動作の変更 88
- EXIT 文
8.0.2 の新機能 143
バージョン 9.0.0 の拡張 99
- EXPLANATION 関数
8.0.2 の新機能 141
- EXPRTYPE 関数
バージョン 9.0.0 の新機能 99
- EXTENDED_JOIN_SYNTAX オプション
バージョン 6.0.2 の新機能 291
- e オプション
9.0.2 で廃止された QAnywhere オプション 55
バージョン 8.0.0 では廃止 225
- ## F
- FileDataSourceName 接続パラメータ
バージョン 7.0.3 の UNIX 新機能 232
- FILEDSN 接続パラメータ
バージョン 7.0.3 の UNIX 新機能 232
- FipsMode プロパティ
バージョン 9.0.2 の新機能 27
- fips オプション
バージョン 9.0.2 の新機能 27
- FIRST_DAY_OF_WEEK オプション
バージョン 7.0.2 の新機能 238
- FIRST 句
バージョン 7.0.0 の新機能 261
- FOR_XML_NULL_TREATMENT オプション
バージョン 9.0.0 の新機能 92
- FOR XML 句
バージョン 9.0.0 の新機能 92
- FreePageBitMaps プロパティ
バージョン 8.0.0 の新機能 194
- FROM 句
8.0.2 の新機能 144
独立性レベルの設定 166

- バージョン 9.0.1 の強化 64
- fr オプション
 - Mobile Link [dbmlsrv9] の新機能 135
- FTP メッセージ・タイプ
 - バージョン 6.0.3 での SQL Remote の新機能 283
 - バージョン 7.0.1 の NetWare の新機能 249

G

- gencert ユーティリティ
 - バージョン 8.0.2 での強化 152
- GET_IDENTITY
 - 8.0.1 の新機能 165
- GetColumnSQLType メソッド
 - 8.0.1 での Ultra Light の新機能 172
- getLastDownloadTime メソッド (JdbcConnection クラス)
 - 8.0.1 での Ultra Light の新機能 171
- getLastIdentity メソッド (JdbcConnection クラス)
 - 8.0.1 での Ultra Light の新機能 171
- GLOBAL_DATABASE_ID オプション
 - バージョン 8.0.1 での動作の変更 175
- GNU GCC ツール
 - バージョン 7.0.1 での Ultra Light の新機能 250
- GRAPHICAL_PLAN 関数
 - 8.0.2 の新機能 141
- GROUPING SETS 演算
 - バージョン 9.0.1 の新機能 58
- gs オプション
 - バージョン 6.0.1 での動作の変更 302
- GUID
 - 8.0.2 の新機能 140

H

- handle_odbc_error
 - バージョン 7.0.1 の新機能 248
- HASH 関数

- バージョン 9.0.1 の新機能 59
- Histograms プロパティ
 - バージョン 8.0.0 の新機能 194
- HotSync conduit
 - バージョン 9.0.0 でのトラブルシューティングの支援 119
- HotSync conduit の設定
 - 8.0.1 での Ultra Light の新機能 172
- [HotSync progress]
 - 8.0.1 での Ultra Light の新機能 172
- HoverRescanThreshold dbmlsync 拡張オプション
 - バージョン 9.0.0 の新機能 118
- hrt dbmlsync 拡張オプション
 - バージョン 9.0.0 の新機能 118
- HTTP
 - 9.0.2 で廃止された Mobile Link セキュリティ 51
- http_password プロトコル・オプション
 - 9.0.2 の新機能 32
- http_proxy_password プロトコル・オプション
 - 9.0.2 の新機能 32
- http_proxy_userid プロトコル・オプション
 - 9.0.2 の新機能 32
- http_userid プロトコル・オプション
 - 9.0.2 の新機能 32
- HTTPS
 - 8.0.2 での Mobile Link の新機能 151
- HTTPS 同期
 - 8.0.2 での Ultra Light の新機能 154
- HTTP クライアント
 - バージョン 9.0.2 の新機能 20
- HTTP サーバ
 - バージョン 9.0.2 の新機能 21
- HTTP 同期
 - バージョン 7.0.1 での Ultra Light の新機能 251

I

- iAnywhere.Data.UltraLite ネームスペース
 - バージョン 9.0.2 の ADO.NET インタフェース

- 35
- iAnywhere JDBC ドライバ
8.0.2 の新機能 146
- IBM DB2
バージョン 9.0.1 での設定スクリプト名の変更 74
- IdleTimeout プロパティ
8.0.1 の新機能 169
- Idle 接続パラメータ
バージョン 8.0.0 の新機能 198
- id オプション
dbtran でのサポート終了 226
- IF 式
9.0.2 の Ultra Light の新機能 39
- ignore パラメータ
9.0.2 の新機能 30
- IMAGE データ型
バージョン 7.0.0 での Mobile Link の新機能 267
バージョン 7.0.0 での記憶領域の強化 259
- InfoMaker
Adaptive Server Anywhere との使用 345
- INI ファイル
バージョン 9.0.2 の新機能 26
- Input Method Editor
バージョン 9.0.1 での動作の変更 86
- INPUT 文
バージョン 9.0.1 の強化 67
- INSERT 文
8.0.2 の新機能 140, 143
- InstallShield テンプレート
バージョン 7.0.1 の新機能 245
- InstallShield プロジェクト
バージョン 9.0.1 の新機能 64
- Interactive SQL
8.0.2 の新機能 143
9.0.2 の Ultra Light の新機能 40
バージョン 5 338
バージョン 7.0.2 の新機能 237
バージョン 8.0.0 でのバッチ処理の新機能 189
バージョン 8.0.0 の新機能 187
- バージョン 9.0.1 の強化 68
- Interop.UltraLiteAFLib.dll
バージョン 9.0.2 での Ultra Light の動作の変更 55
- INTERSECT 演算
バージョン 9.0.0 の新機能 95
- IPM_Receive 制御パラメータ
バージョン 9.0.2 での動作の変更 52
- IPX
バージョン 7.0.0 では旧式 269
- IPX プロトコル
バージョン 8.0.0 ではサポート終了 225
- ISDATE 関数
バージョン 8.0.0 の新機能 187
- IsFipsAvailable プロパティ
バージョン 9.0.2 の新機能 27
- IsJavaAvailable プロパティ
バージョン 8.0.0 の新機能 195
- IsNetworkServer プロパティ
バージョン 6.0.3 の新機能 279
- ISNUMERIC 関数
バージョン 8.0.0 の新機能 187
- ISQL_PLAN オプション
バージョン 8.0.0 での動作の変更 226
- IsRuntimeServer プロパティ
バージョン 7.0.0 の新機能 263
- ## J
- java.io
バージョン 6.0.3 の新機能 281
- java.lang.thread パッケージ
バージョン 6.0.3 の新機能 281
- java.math.BigInteger クラス
バージョン 7.0.0 の新機能 265
- java.net.PlainDatagramSocketImpl クラス
バージョン 7.0.1 の新機能 246
- Java
バージョン 6.0.1 の新機能 298
バージョン 8.0.0 でのセキュリティ・マネージャの新機能 199

Java 2

バージョン 8.0.0 の新機能 198

Java 同期論理

バージョン 8.0.0 での Mobile Link の新機能
202

jConnect

バージョン 6.0.1 の新機能 298

jConnect ドライバ

バージョン 7.0.0 の新機能 257

JDBC

バージョン 6.0.3 の新機能 280

バージョン 8.0.0 の新機能 198

JDBC JAR ファイル

バージョン 8.0.0 での動作の変更 221

JDBC ResultSet メソッド

8.0.1 での Ultra Light の新機能 171

JDBC エスケープ構文

バージョン 8.0.0 の新機能 188

JDK

Ultra Light で必要なバージョン 135

バージョン 8.0.0 の新機能 198

K

keep_alive

Mobile Link バージョン 9.0.0 では常に ON
135

L

LargeProcedureIDs プロパティ

バージョン 8.0.0 の新機能 194

LastStatement プロパティ

バージョン 8.0.0 の新機能 194

LazyClose 接続パラメータ

8.0.2 の新機能 144

LCLOSE 接続パラメータ

8.0.2 の新機能 144

LDAP 認証

9.0.1 の Mobile Link の強化 73

libdbtool9.so

バージョン 9.0.2 での廃止 48

LINKS 接続パラメータ

バージョン 6.0.1 での動作の変更 302

バージョン 8.0.0 での動作の変更 223

Linux

バージョン 9.0.1 の新機能 70

LIST 関数

9.0.0 の拡張 96

順序 218

バージョン 7.0.0 の新機能 264

LivenessTimeout プロパティ

8.0.1 の新機能 169

LOAD TABLE 文

8.0.2 の新機能 142, 143

バージョン 6.0.3 の新機能 277

バージョン 7.0.3 での動作の変更 233

バージョン 9.0.1 の強化 60

バージョン 9.0.1 の変数の強化サポート 60

LocalOnly プロトコル・オプション

8.0.1 の新機能 168

LOCAL プロトコル・オプション

8.0.1 の新機能 168

LOCK TABLE 文

バージョン 7.0.0 の新機能 261

LOG_DEADLOCKS オプション

バージョン 9.0.2 の新機能 22, 28

LOG_LEVEL

バージョン 9.0.0 でのリダイレクタの新機能
120

LogFileFragments プロパティ

8.0.1 の新機能 168

バージョン 8.0.0 の新機能 194

Logfile 接続パラメータ

バージョン 7.0.1 の新機能 245

LOGIN_PROCEDURE オプション

バージョン 7.0.3 の新機能 232

バージョン 8.0.0 での動作の変更 224

バージョン 8.0.0 の新機能 196

LoginTime プロパティ

バージョン 9.0.2 の新機能 27

Log Transfer Manager ユーティリティ [dbltm]

バージョン 6.0.3 の新機能 282

LOG 接続パラメータ

バージョン 7.0.1 の新機能 245

LONG BINARY データ型

バージョン 7.0.0 での Mobile Link の新機能
267

バージョン 7.0.0 での記憶領域の強化 259

LONG VARCHAR データ型

バージョン 7.0.0 での Mobile Link の新機能
267

バージョン 7.0.0 での記憶領域の強化 259

LTM

バージョン 8.0.0 の新機能 202

LTMGeneration プロパティ

バージョン 7.0.1 の新機能 245

LTMTrunc プロパティ

バージョン 7.0.0 の新機能 263

M

MachineName プロパティ

8.0.0 の新機能 195

MAX_HASH_SIZE オプション

バージョン 8.0.0 では廃止 226

MAX_PLANS_CACHED オプション

バージョン 8.0.0 の新機能 184

Max_plans_cached プロパティ

バージョン 8.0.0 の新機能 194

MAX_WORK_TABLE_HASH_SIZE オプショ
ン

バージョン 8.0.0 では廃止 226

M-Business Anywhere

9.0.1 の Ultra Light の新機能 77

9.0.2 の Ultra Light の強化 36

M-Business Anywhere リダイレクタ

バージョン 9.0.2 の新機能 30

MDSR 暗号化

バージョン 9.0.1 での廃止 84

Message Agent

バージョン 9.0.2 の新機能 34

MESSAGE 文

バージョン 6.0.1 の新機能 299

バージョン 9.0.1 の強化 61

MIN_PASSWORD_LENGTH オプション

バージョン 6.0.2 の新機能 291

MIN_TABLE_SIZE_FOR_HISTOGRAM オプ
ション

バージョン 8.0.0 の新機能 184

バージョン 9.0.1 での動作の変更 87

バージョン 9.0.2 での廃止 47

MirrorLogDirectory dbmlsync 拡張オプション

バージョン 9.0.2 の新機能 33

ml_qa_delivery

9.0.2 の新しい QAnywhere Mobile Link システ
ム・テーブル 44

ml_qa_delivery_client

9.0.2 の新しい QAnywhere Mobile Link システ
ム・テーブル 44

ml_qa_repository_content

9.0.2 で削除された QAnywhere Mobile Link シス
テム・テーブル 45

ml_qa_repository_staging_client

9.0.2 で削除された QAnywhere Mobile Link シス
テム・テーブル 45

ml_qa_status_staging_client

9.0.2 で削除された QAnywhere Mobile Link シス
テム・テーブル 45

ml_subscription

バージョン 9.0.0 で変更された Mobile Link シ
ステム・テーブル 117

ml_user

バージョン 9.0.0 で変更された Mobile Link シ
ステム・テーブル 117

mlxtract ユーティリティ

バージョン 7.0.0 の新機能 267

Mobile Link

アップグレード 316

サンプル 12

バージョン 9 の新機能の概要 9

Mobile Link ASA クライアントのローワイズ
分割

9.0.1 の Mobile Link の新しい動作 88

Mobile Link が発行する ODBC 文のログ

8.0.1 の新機能 170

Mobile Link システム・テーブル

- 9.0.1 の新しいテーブル 73
- アップグレード 317
- バージョン 7.0.0 での動作の変更 273
- Mobile Link 同期サーバ
 - アップグレード 318
 - バージョン 6.0.3 の新機能 285
 - バージョン 8.0.0 での動作の変更 227
- Mobile Link の新機能
 - バージョン 9.0.2 30
- Mobile Link の動作の変更
 - バージョン 9.0.2 51
- Mobile Link モニタ
 - 8.0.2 の新機能 149
- Mobile Link ユーザ名
 - バージョン 8.0.0 での動作の変更 227
- modify_last_download_timestamp
 - 8.0.1 の新機能 170
- modify_next_last_download_timestamp
 - 8.0.1 の新機能 170
- MONTHS 関数
 - バージョン 6.0.3 での動作の変更 286
- MultiByteCharSet プロパティ
 - バージョン 6.0.3 の新機能 280

N

- Native Ultra Light for Java
 - 9.0.2 の Ultra Light の強化 37
- NetBios プロトコル
 - バージョン 8.0.0 ではサポート終了 225
- NetWare
 - バージョン 8.0.0 でサポートされていないバージョン 4.10 225
 - バージョン 9.0.1 の新機能 64
- network_leave_open
 - 8.0.2 での Mobile Link [dbmlsrv8] の新機能 151
- network_name
 - 8.0.2 での Mobile Link [dbmlsrv8] の新機能 151
- network_open_timeout

- 8.0.2 での Mobile Link [dbmlsrv8] の新機能 151
- new_row_cursor
 - バージョン 9.0.0 でサポートを終了予定の機能 132
- Notifier
 - 9.0.0 での Mobile Link の新機能 115
- NULL
 - バージョン 8.0.0 での空文字列の動作の変更 221
 - バージョン 9.0.1 での動作の変更 87
- NULLIF 関数
 - バージョン 6.0.1 の新機能 299
- NUMBER 関数
 - バージョン 8.0.0 での動作の変更 219

O

- ODBC
 - 9.0.1 の Ultra Light の新機能 78
 - バージョン 5 の互換性 351
- ODBC_DESCRIBE_BINARY_AS_VARBINARY オプション
 - バージョン 9.0.2 の新機能 21
- ODBC_DISTINGUISH_CHAR_AND_VARCHAR オプション
 - バージョン 9.0.1 の新機能 72
- [ODBC 設定] ダイアログ
 - バージョン 9.0.1 での動作の変更 86
- ODBC データ・ソース
 - バージョン 9.0.2 の新機能 24
- old_row_cursor
 - バージョン 9.0.0 でサポートを終了予定の機能 132
- OLE DB プロバイダ
 - バージョン 7.0.0 の新機能 257
- on オプション
 - 9.0.0 での Mobile Link [dbmlsrv9] の新機能 116
- OPENXML 関数
 - バージョン 9.0.0 の新機能 93
 - バージョン 9.0.1 の強化 71

OPEN 文

バージョン 8.0.0 での動作の変更 219

OPTIMIZATION_GOAL オプション

8.0.2 で変更されたデフォルト設定 156

バージョン 7.0.0 の新機能 260

バージョン 8.0.0 の新機能 182

OPTIMIZATION_WORKLOAD オプション

バージョン 9.0.1 の新機能 63

ORDER BY 句

9.0.0 の拡張 97

ORDER BY と UPDATE 文

バージョン 9.0.1 での動作の変更 87

OUTPUT 文

バージョン 8.0.0 の新機能 189

バージョン 9.0.0 の拡張 100

バージョン 9.0.1 の強化 67

P

PacketSize プロパティ

バージョン 8.0.0 の新機能 194

PageSize プロパティ

バージョン 7.0.0 の新機能 262

Palm Computing Platform

Ultra Light アプリケーションのアップグレード
321バージョン 2 はバージョン 8.0.0 ではサポート
終了 228バージョン 8.0.0 での Ultra Light の新機能
212

バージョン 8.0.0 で必要なコードの変更 228

Palm OS

9.0.1 でサポートされる最も古いバージョン
89

-password オプション

9.0.2 で廃止された QAnywhere オプション 55

PBUF 接続パラメータ

バージョン 8.0.0 の新機能 197

PCTFREE

8.0.1 の新機能 164

PDF フォーマット

9.0.1 の強化 83

オンライン・マニュアル 83

PERCENT_RANK 関数

バージョン 9.0.1 の新機能 58

Perl

バージョン 9.0.1 の新しい DBD::ASAny ドライ
バ 64

persistent

8.0.2 での Mobile Link [HTTP、HTTPS] の新機
能 150

PHP

バージョン 9.0.2 の新機能 20

ping ユーティリティ [dbping]

バージョン 6.0.2 の新機能 291

PINNED_CURSOR_PERCENT_OF_CACHE

オプション

8.0.1 の新機能 168

-pi オプション

8.0.2 での Mobile Link [dbmsync] の新機能
150

PLAN 関数

8.0.2 の新機能 141

PlatformVer プロパティ

バージョン 8.0.0 の新機能 195

Pocket PC

8.0.1 でサポートされる 2002 171

post

9.0.2 の Mobile Link [dblsn] の強化 33

PowerBuilder

Adaptive Server Anywhere との使用 345

PRC ツール

バージョン 7.0.1 での Ultra Light の新機能
250

PrefetchBuffer 接続パラメータ

バージョン 8.0.0 の新機能 197

PreFetchOnOpen 接続パラメータ

8.0.2 の新機能 144

PrefetchRows 接続パラメータ

バージョン 8.0.0 の新機能 197

PREFETCH オプション

バージョン 9.0.2 の強化 21

prepare_for_download

バージョン 7.0.1 の新機能 248

PreserveSource プロパティ
バージョン 8.0.0 の新機能 194

ProcedurePages プロパティ
バージョン 8.0.0 の新機能 194

PROPERTY 関数
バージョン 9.0.2 での動作の変更 50

PROWS 接続パラメータ
バージョン 8.0.0 の新機能 197

Q

QAnywhere
9.0.1 の Mobile Link の新機能 72

QAnywhere の新機能
バージョン 9.0.2 42

QAnywhere の動作の変更
バージョン 9.0.2 55

qastop ユーティリティ
QAnywhere Agent 43

-qp オプション
バージョン 8.0.0 の新機能 193

qualify_table_owner パラメータ
バージョン 8.0.0 の新機能 202

QueryCachePages プロパティ
バージョン 8.0.0 の新機能 194

QueryLowMemoryStrategy プロパティ
バージョン 8.0.0 の新機能 194

R

RAISERROR 文
バージョン 8.0.0 の新機能 186

RANK 関数
バージョン 9.0.1 の新機能 58

-rb オプション
9.0.2 で廃止された QAnywhere オプション 55

READ 文
バージョン 9.0.1 の強化 67

REAL データ型
バージョン 6.0.3 での Mobile Link の新機能 285

バージョン 6.0.3 での Ultra Light の新機能 285

REGR_AVGX 関数
バージョン 9.0.1 の新機能 58

REGR_AVGY 関数
バージョン 9.0.1 の新機能 58

REGR_COUNT 関数
バージョン 9.0.1 の新機能 58

REGR_INTERCEPT 関数
バージョン 9.0.1 の新機能 58

REGR_R2 関数
バージョン 9.0.1 の新機能 58

REGR_SLOPE 関数
バージョン 9.0.1 の新機能 58

REGR_SXX 関数
バージョン 9.0.1 の新機能 58

REGR_SXY 関数
バージョン 9.0.1 の新機能 58

REGR_SYY 関数
バージョン 9.0.1 の新機能 58

release_version
9.0.2 の新機能 31

remember_last_statement
バージョン 8.0.0 の新機能 193

REMOTEPWD フィールド
バージョン 7.0.0 での動作の変更 271

RemoteTrunc プロパティ
バージョン 7.0.0 の新機能 263

REORGANIZE TABLE 文
バージョン 8.0.0 の新機能 191

REPLACE 関数
バージョン 7.0.0 の新機能 263

report_error イベント
バージョン 7.0.0 の新機能 267

report_odbc_error
バージョン 7.0.1 の新機能 248

RequestLogNumFiles プロパティ
バージョン 9.0.2 の新機能 26

RESTORE DATABASE 文
バージョン 6.0.2 の新機能 290

バージョン 8.0.0 での動作の変更 221

バージョン 9.0.1 の変数の強化サポート 60

バージョン 9.0.2 の新機能 25
 RETURN_DATE_TIME_AS_STRING オプション
 バージョン 8.0.2 の新機能 146
 バージョン 8.0.0 の新機能 190
 REWRITE 関数
 8.0.1 の新しい引数 ANSI 165
 ROLLBACK_ON_DEADLOCK オプション
 バージョン 9.0.2 の新機能 28
 ROLLBACK 文
 9.0.1 の Ultra Light 動的 SQL の新機能 77
 ROLLUP 演算
 バージョン 9.0.0 の新機能 96
 バージョン 9.0.1 の新機能 58
 ROW_NUMBER 関数
 バージョン 9.0.1 の新機能 58
 RSA
 8.0.2 での Mobile Link の新機能 151

S

sa_check_commit システム・プロシージャ
 バージョン 7.0.0 の新機能 263
 sa_conn_activity システム・プロシージャ
 バージョン 8.0.0 の新機能 193
 SA_DEBUG グループ
 バージョン 6.0.2 の新機能 294
 sa_disk_free_space システム・プロシージャ
 8.0.1 の新機能 164
 sa_flush_statistics システム・プロシージャ
 8.0.1 の新機能 164
 sa_get_request_profile システム・プロシージャ
 バージョン 9.0.2 の新機能 26
 sa_get_request_times システム・プロシージャ
 バージョン 9.0.2 の新機能 26
 sa_get_server_messages システム・プロシージャ
 8.0.1 の新機能 165
 sa_index_density システム・プロシージャ
 バージョン 8.0.0 の新機能 192
 バージョン 9.0.2 の新機能 25
 sa_index_levels システム・プロシージャ
 バージョン 9.0.2 の新機能 25
 sa_java_loaded_classes システム・プロシージャ
 バージョン 8.0.0 の新機能 198
 sa_locks システム・プロシージャ
 バージョン 7.0.0 の新機能 255
 sa_make_object システム・プロシージャ
 8.0.2 の新機能 146
 sa_migrate システム・プロシージャ
 バージョン 7.0.2 の新機能 238
 バージョン 8.0.1 の新機能 167
 sa_report_deadlocks システム・プロシージャ
 バージョン 9.0.2 の新機能 22
 sa_rowgenerator システム・プロシージャ
 バージョン 9.0.2 の新機能 28
 sa_send_udp システム・プロシージャ
 バージョン 9.0.2 の新機能 29
 sa_server_option システム・プロシージャ
 バージョン 6.0.3 の新機能 278
 バージョン 7.0.0 の新機能 265
 バージョン 9.0.1 の強化 66
 バージョン 9.0.2 の新機能 26
 sa_table_fragmentation システム・プロシージャ
 バージョン 8.0.0 の新機能 192
 sa_validate システム・プロシージャ
 バージョン 8.0.0 の新機能 191
 バージョン 9.0.1 の新機能 62
 sa_verify_password システム・プロシージャ
 バージョン 9.0.2 の新機能 29
 SATMP 環境変数
 バージョン 8.0.0 では廃止 226
 Schema Painter
 9.0.2 の強化 41
 scjview.exe
 バージョン 9.0.0 の新機能 113
 ScoutSync
 サポート終了 135
 使用されなくなった機能 177
 バージョン 9.0 での削除 134

- SELECT 文
 - バージョン 7.0.2 での動作の変更 241
- SeparateCheckpointLog プロパティ
 - バージョン 8.0.0 の新機能 194
- SeparateForeignKeys プロパティ
 - バージョン 8.0.0 の新機能 194
- ServerPort プロパティ
 - バージョン 9.0.2 の新機能 27
- set_cookie プロトコル・オプション
 - 9.0.2 の Ultra Light の新機能 41
 - 9.0.2 の新機能 32
- SET 文
 - 8.0.2 の新機能 143
- SiteScriptName
 - サポート終了 131
- Smartphone 2002
 - 9.0.1 の Ultra Light の新機能 80
- SMB
 - SMB のデータベースとしての Adaptive Server Anywhere データベース 2
- SMTP/POP メッセージ・タイプ
 - バージョン 6.0.3 での SQL Remote の新機能 283
 - バージョン 7.0.1 の NetWare の新機能 249
- SMTP システム・プロシージャ
 - バージョン 8.0.0 の新機能 201
- SMTP メッセージ・タイプ
 - 8.0.1 の新機能 171
- SNMP エージェント
 - バージョン 9.0.2 の新機能 22
- SOAP クライアント
 - バージョン 9.0.2 の新機能 20
- SOAP サーバ
 - バージョン 9.0.2 の新機能 21
- SORT_COLLATION データベース・オプション
 - 8.0.1 の新機能 167
- SORTKEY 関数
 - バージョン 8.0.0 の新機能 186
- sp_hook_dbmlsync_connect_failed ストアド・プロシージャ
 - 9.0.2 の新機能 32
- sp_hook_dbmlsync_download_ri_violation ストアド・プロシージャ
 - 8.0.2 の新機能 149
- sp_hook_dbmlsync_schema_upgrade ストアド・プロシージャ
 - バージョン 9.0.0 の新機能 118
- sp_remote_exported_keys システム・プロシージャ
 - バージョン 8.0.0 の新機能 201
- sp_remote_imported_keys システム・プロシージャ
 - バージョン 8.0.0 の新機能 201
- sp_remote_tables システム・プロシージャ
 - バージョン 8.0.0 の新機能 200
- SPX
 - バージョン 7.0.0 の新機能 258
- SQL/XML
 - バージョン 9.0.0 の新機能 93
- SQL Anywhere
 - こんぼーねんと 2
- SQLAnywhere PHP モジュール
 - バージョン 9.0.2 の新機能 20
- SQL Anywhere Studio
 - マニュアル viii
- SQLE_DATABASE_NOT_FOUND エラー
 - 9.0.0 での Ultra Light の動作の変更 137
- SQLLOCALE 環境変数
 - バージョン 8.0.0 での動作の変更 223
- SQL Remote
 - ASE サポートの廃止 52
 - アップグレード 323
 - バージョン 6.0.1 の新機能 301
- SQL Remote の新機能
 - バージョン 9.0.2 34
- SQL Remote の動作の変更
 - バージョン 9.0.2 52
- SR_DATE_FORMAT
 - バージョン 6.0.3 での SQL Remote の新機能 283
- SR_TIME_FORMAT
 - バージョン 6.0.3 での SQL Remote の新機能 283

SR_TIMESTAMP_FORMAT

バージョン 6.0.3 での SQL Remote の新機能
283

ssqueue

9.0.2 での廃止 52

ssremote

9.0.2 での廃止 52

START AT

バージョン 9.0.0 の新機能 97

START DATABASE 文

バージョン 7.0.0 の新機能 261

STOP DATABASE 文

バージョン 7.0.0 の新機能 261

STOP ENGINE 文

バージョン 7.0.0 の新機能 261

StreamCompression

バージョン 8.0.0 では廃止 229

STRTOUID 関数

バージョン 9.0.2 での動作の変更 50

SUPPRESS_TDS_DEBUGGING オプション

バージョン 8.0.0 の新機能 197

-sv オプション

9.0.2 で廃止された QAnywhere オプション 55

Sybase Central

バージョン 8.0.0 の新機能 187

バージョン 9.0.1 の新機能 69

バージョン 9 の新機能の概要 10

古いデータベースの管理 306

Sybase Central からのバージョン 5.x データベースのアンロード

8.0.1 の新機能 166

Sybase Central からのバージョン 6.x データベースのアンロード

8.0.1 の新機能 166

SyncTrunc プロパティ

バージョン 7.0.0 の新機能 263

SYSATTRIBUTENAME システム・テーブル

8.0.1 の新機能 164

SYSATTRIBUTE システム・テーブル

8.0.1 の新機能 164

SYSCOLLATIONMAPPINGS システム・テーブル

バージョン 8.0.0 での動作の変更 220

SYSHISTORY システム・テーブル

バージョン 9.0.1 の新機能 67

-s オプション

バージョン 9.0.1 の新しいバックアップ・ユーティリティ・オプション 65

T**TableBitMaps プロパティ**

バージョン 8.0.0 の新機能 194

TCP/IP

バージョン 7.0.0 の新機能 257

バージョン 8.0.0 での暗号化 184

TEMP_SPACE_LIMIT_CHECK オプション

バージョン 9.0.2 の新機能 28

TempTablePages プロパティ

バージョン 8.0.0 の新機能 194

TEXT データ型

バージョン 7.0.0 での Mobile Link の新機能
267

バージョン 7.0.0 での記憶領域の強化 259

TIME_ZONE_ADJUSTMENT オプション

バージョン 8.0.0 の新機能 186

TIMESTAMP_FORMAT オプション

バージョン 8.0.0 での新しい動作 222

TimeZoneAdjustment プロパティ

バージョン 8.0.0 の新機能 186

TOP 句

バージョン 7.0.0 の新機能 261

バージョン 9.0.1 の変数の強化サポート 60

TransactionsSpanLogs プロパティ

バージョン 8.0.0 の新機能 194

TransactionStartTime プロパティ

バージョン 9.0.1 の新機能 71

Transact-SQL

外部ジョインの動作の変更 328

TRIGGER EVENT 文

バージョン 7.0.0 の新機能 254

TRUNCATE_DATE_VALUES オプション

バージョン 9.0.2 での廃止 47

TRUNCATE_TIMESTAMP_VALUES オプション
バージョン 7.0.2 の新機能 239

TRUNCATE TABLE 文
バージョン 7.0.0 でのパフォーマンスの強化
265

-tu オプション
9.0.1 の新機能 dbmlsrv9 75

U

-ua オプション
バージョン 9.0.1 の新機能 70

-uc オプション
バージョン 9.0.2 の新機能 29

-ui オプション
バージョン 9.0.2 の新機能 29

UL_STORE_PARMS マクロ
9.0.0 での Ultra Light の動作の変更 138

ul_stream_error 構造体
バージョン 8.0.0 での Ultra Light の新機能
211

ulconv ユーティリティ
9.0.1 の Ultra Light の新機能 80

ULCountUploadRows 関数
バージョン 8.0.0 での Ultra Light の新機能
215

ulcreate ユーティリティ
9.0.2 の Ultra Light の新機能 40

ULEnableFileDB 関数
バージョン 8.0.0 での動作の変更 228

ULEnableGenericSchema 関数
8.0.2 での Ultra Light の新機能 153
9.0.1 で使用されなくなった機能 89

ULEnablePalmRecordDB 関数
バージョン 8.0.0 での動作の変更 228

ulgen ユーティリティ
バージョン 8.0.0 での動作の変更 229

ULGetRowCount 関数
バージョン 8.0.0 での Ultra Light の新機能
214

ulload ユーティリティ

9.0.2 の Ultra Light の新機能 40

ULPalmExit 関数
9.0.1 で使用されなくなった機能 89

ULPalmLaunch 関数
9.0.1 で使用されなくなった機能 89

ulpalmos.h
9.0.2 のプレフィクス・ファイルの強化 38

ULRegisterErrorCallback 関数
9.0.1 の Ultra Light C/C++ の新機能 79

ULSecureCerticomTLSStream
バージョン 9.0.2 での Ultra Light の動作の変更
54

ulsync ユーティリティ
9.0.2 の Ultra Light の新機能 40

ULTable 214

Ultra Light.NET
9.0.2 の強化 38
ADO.NET インタフェースの推奨 53
バージョン 9.0.2 の ADO.NET インタフェース
35

Ultra Light.NET コントロール
9.0.1 の新機能 77

Ultra Light
9.0.0 でのプラットフォーム・サポートの変更
135
サンプル 11
バージョン 6.0.2 の新機能 290
バージョン 9 の新機能 3
バージョン 9 の新機能の概要 7

Ultra Light Embedded SQL
Ultra Light 9.0.2 での廃止 53

Ultra Light for M-Business Anywhere
9.0.1 の新機能 77

Ultra Light Interactive SQL
9.0.2 の新機能 40

Ultra Light エンジン
バージョン 9.0.2 の Ultra Light for M-Business
Anywhere のサポート 36
9.0.1 の Ultra Light の新機能 79

Ultra Light コンポーネント
8.0.2 の新機能 152
9.0.1 でのテーブル API の動作の変更 89

- Ultra Light ジェネレータ
 - 8.0.2 での Ultra Light の新機能 155
 - バージョン 8.0.0 での Ultra Light の新機能 210
 - バージョン 8.0.0 での動作の変更 229
 - Ultra Light 静的インタフェース
 - 用語 121
 - Ultra Light 静的型 C++ API
 - 9.0.2 での廃止 53
 - Ultra Light 静的型 Java API
 - 9.0.2 での廃止 53
 - Ultra Light データベース・コンバータ
 - 9.0.1 の Ultra Light の新機能 80
 - Ultra Light の新機能
 - バージョン 9.0.2 35
 - Ultra Light の動作の変更
 - バージョン 9.0.2 53
 - ulunload ユーティリティ
 - 9.0.2 の Ultra Light の新機能 40
 - UNIQUEIDENTIFIER データ型
 - 9.0.2 での強化 18
 - UNLOAD TABLE 文
 - バージョン 9.0.1 の変数の強化サポート 60
 - UNLOAD 文
 - バージョン 7.0.0 の新機能 255
 - バージョン 9.0.1 の変数の強化サポート 60
 - uo dbmsync 拡張オプション
 - 9.0.0 での Mobile Link の新機能 117
 - uo オプション
 - 9.0.0 での Mobile Link [dbmsync] の新機能 117
 - 9.0.0 での Mobile Link の新機能 117
 - UPDATE 文
 - バージョン 6.0.3 の新しい Java 機能 280
 - upload_cursor
 - バージョン 9.0.0 でサポートを終了予定の機能 132
 - UploadOnly dbmsync 拡張オプション
 - 9.0.0 での Mobile Link の新機能 117
 - urc オプション
 - 8.0.2 での Mobile Link [dbmsync] の新機能 149
 - use_cookies
 - 8.0.2 で削除された Mobile Link HTTP オプション 160
 - UserAppInfo プロパティ
 - バージョン 9.0.1 の新機能 71
 - us オプション
 - 9.0.1 の新機能 dbmsync 74
 - UtilCmdsPermitted プロパティ
 - バージョン 8.0.0 の新機能 194
 - UUID
 - 8.0.2 での Mobile Link の新機能 148
 - 8.0.2 での Ultra Light の新機能 153
 - 8.0.2 の新機能 140
 - ux オプション
 - バージョン 9.0.2 の新機能 29
- ## V
- V4T モード
 - 9.0.1 の Ultra Light の新機能 80
 - VALIDATE CHECKSUM 文
 - バージョン 9.0.1 の新機能 62
 - VALIDATE INDEX 文
 - バージョン 7.0.0 の新機能 255
 - VALIDATE TABLE 文
 - バージョン 7.0.0 の新機能 255
 - VAREXISTS 関数
 - 8.0.2 の新機能 142
 - VariableHashSize プロパティ
 - バージョン 8.0.0 の新機能 194
 - verbose オプション
 - 9.0.2 で廃止された QAnywhere オプション 55
 - VERIFY プロトコル・オプション
 - 8.0.1 の新機能 168
 - VerifyServerName プロトコル・オプション
 - 8.0.1 の新機能 168
 - Visual Basic .NET
 - AppForge CrossFire の Ultra Light サポート 36
 - vp オプション
 - 9.0.0 での Mobile Link [dbmsrv9] の新機能 116

-vw オプション

8.0.2 での Mobile Link [dbmlsrv8] の変更点
159

VxWorks

サポート終了 135

-v オプション

6.0.1 でのデータベース・サーバの動作の変更
302

8.0.2 で強化された Mobile Link [dbmlsync] の機能 150

8.0.2 での Mobile Link [dbmlsync] の変更点
159

W

WAITFOR 文

バージョン 8.0.0 の新機能 186

バージョン 9.0.2 の新機能 19

Web サービス

バージョン 9.0.2 の新機能 21

バージョン 9 の新機能 3

Web サービス・クライアント

バージョン 9.0.2 の新機能 20

Windows 3.x

バージョン 7.0.0 ではサポート対象外 269

Windows CE

2.11 のサポート終了 156, 160

8.0.2 の新機能 141

dll をロードする 9.0.0 での dbmlsync の新機能
118

SH3 チップのサポート終了 156, 160

SH4 チップのサポート終了 156, 160

デフォルトの dbmlsync 排他モード 131

バージョン 6.0.1 の新機能 296

バージョン 6.0.3 の新機能 277

バージョン 7.0.1 の新機能 244

バージョン 9.0.2 の新機能 29

Windows CE エミュレータ

バージョン 7.0.1 での Ultra Light の新機能
251

Windows グループ

バージョン 9.0.2 の新機能 26

Windows パフォーマンス モニタ

8.0.2 の新機能 145

WITH (XLOCK) 機能

バージョン 9.0.1 の強化 64

WITH AUTO NAME

8.0.2 の新機能 143

WITH HASH SIZE 句

バージョン 8.0.0 では廃止 226

WITH table-hint

8.0.1 の新機能 166

X

XML

バージョン 8.0.0 の新機能 189

バージョン 9.0.0 の新機能 92

バージョン 9 の新機能 3

XMLAGG 関数

バージョン 9.0.0 の新機能 93

XMLCONCAT 関数

バージョン 9.0.0 の新機能 93

XMLELEMENT 関数

バージョン 9.0.0 の新機能 93

XMLFOREST 関数

バージョン 9.0.0 の新機能 93

XMLGEN 関数

バージョン 9.0.0 の新機能 93

XML データ型

バージョン 9.0.0 の新機能 93

xp_cmdshell システム・プロシージャ

8.0.2 での動作変更 156

xp_read_file システム・プロシージャ

バージョン 7.0.0 の新機能 260

xp_sendmail システム・プロシージャ

バージョン 8.0.0 の新機能 201

xp_startsmtp システム・プロシージャ

バージョン 8.0.0 の新機能 201

xp_stopsmtp システム・プロシージャ

バージョン 8.0.0 の新機能 201

xp_write_file システム・プロシージャ

バージョン 7.0.0 の新機能 260

- xs オプション
バージョン 9.0.2 でのサーバ・オプションの動作の変更 49
- x オプション
バージョン 6.0.1 での動作の変更 302

Y

YEARS 関数

- バージョン 6.0.3 での動作の変更 286

Z

- zac option
サポート終了予定の Mobile Link [dbmlsrv9] の機能 132
- zd オプション
バージョン 9 で削除された Mobile Link [dbmlsrv] の機能 133
- zec option
サポート終了予定の Mobile Link [dbmlsrv9] の機能 132
- zl オプション
バージョン 8.0.0 の新機能 193
- zn オプション
バージョン 9.0.2 の新機能 26
- zwd オプション
8.0.2 での Mobile Link [dbmlsrv8] の新機能 150
- zwe オプション
8.0.2 での Mobile Link [dbmlsrv8] の新機能 150
- zw オプション
8.0.2 での Mobile Link [dbmlsrv8] の新機能 150

あ

- アーカイブのバックアップ
バージョン 7.0.0 での NetWare の新機能 265

- バージョン 9.0.1 の強化 65
- アイコン
マニュアルで使用 xiii
- アクセシビリティ
バージョン 8.0.0 の新機能 187
- アクセス・プラン
9.0.0 の新機能 113
バージョン 8.0.0 での動作の変更 218
バージョン 8.0.0 の新機能 188
- 圧縮
バージョン 8.0.0 の新機能 197
- 圧縮 B ツリー・インデックス
バージョン 8.0.0 の新機能 182
- 圧縮データベース
バージョン 9.0.1 での廃止 84
- アップグレード
8.0.2 での Ultra Light の新機能 153
Adaptive Server Anywhere 306
dbclient 354
Mobile Link 316
Mobile Link システム・テーブル 317
Mobile Link 同期サーバ 318
SQLRemote 323
SQL Remote バージョン 5 357
Ultra Light データベース 321
Ultra Light バージョン 9 321
互換性ライブラリ 347
スタンドアロン・アプリケーション 333
注意事項 313
データベース 307, 356
データベース・ファイル・フォーマット 310
同期スクリプト 319
統合データベース 317
バージョン 5 コンポーネント 326
バージョン 5 データベース 328
リモート・データベース 319
- アップロード・ストリーム
バージョン 7.0.2 の Mobile Link の新機能 240
バージョン 8.0.0 での Mobile Link の新機能 205
- アップロード専用同期
バージョン 7.0.1 での Ultra Light の新機能 251

アップロードのみの同期処理
9.0.0 での Mobile Link の新機能 117

アラビア語
バージョン 9.0.2 の新機能 22

暗号化
8.0.2 の新機能 144
バージョン 8.0.0 での Ultra Light の新機能 209
バージョン 8.0.0 の新機能 184
バージョン 9.0.2 での動作の変更 49

アンロード・ユーティリティ [dbunload]
データベース・ファイルのアップグレード 312
バージョン 6.0.3 の新機能 278
バージョン 9.0.1 で使用されなくなった機能 85
バージョン 9.0.1 の強化 65

い

イベント処理
バージョン 7.0.0 の新機能 254

イベントの処理
バージョン 7.0.0 の新機能 254

イベント・フック
9.0.1 の Mobile Link の新しい動作 88
バージョン 7.0.1 の新機能 246

イベント・ログ
バージョン 7.0.0 の新機能 266

イメージのバックアップ
バージョン 9.0.1 の強化 65

インデックス
8.0.2 の新機能 140
バージョン 7.0.0 の新機能 258
バージョン 8.0.0 の新機能 182

インデックス・コンサルタント
バージョン 9 の新機能 4

インデックス・ヒント
バージョン 9.0.1 の新機能 64

インポート
バージョン 7.0.0 の新機能 260
[インポート] ウィザード

バージョン 9.0.1 の強化 68

う

ウィザード
[データベース・アップグレード] 308
[データベース・アンロード] 310

ウィンドウ機能
バージョン 9.0.1 の新機能 58

埋め込みコンピューティング
Adaptive Server Anywhere 3
Mobile Link 3
Ultra Light 3

え

エイリアス
トラブルシューティング 241
バージョン 7.0.2 での動作の変更 241

エイリアスを参照
参照より前に定義する 241

[エクスポート] ウィザード
バージョン 9.0.1 の強化 68

エミュレータ
9.0.2 の QAnywhere のサポート 42

エラー・コード
9.0.0 での Ultra Light の動作の変更 137

エラー・コンテキスト報告
8.0.1 の新機能 169

エラー処理
9.0.1 の Ultra Light C/C++ の強化 79
バージョン 8.0.0 での Mobile Link の新機能 206

エラー報告
バージョン 8.0.0 での Mobile Link の新機能 206
バージョン 8.0.0 での Ultra Light の新機能 211

エラー・メッセージ
バージョン 8.0.0 の新機能 186

エラー・ログ

8.0.2 での SQL Remote の新機能 152

お

大文字と小文字の区別

バージョン 9.0.0 の新機能 107

オンライン・ヘルプ

バージョン 8.0.0 の新機能 199

オンライン・マニュアル

PDF フォーマット 83

か

カーソル

バージョン 8.0.0 での Ultra Light の新機能
215

バージョン 8.0.0 での動作の変更 218

バージョン 8.0.0 の新機能 190

外部関数

バージョン 7.0.0 の新機能 261

[外部キー] プロパティ・シート

バージョン 9.0.1 の強化 69

外部参照集合関数

ユーザ定義動作のバージョン 8.0.0 での変更
217

外部ジョイン

Transact-SQL 動作の変更 328

バージョン 7.0.0 での Transact-SQL の新機能
262

外部ストアド・プロシージャ

バージョン 7.0.0 の新機能 261

拡張モード

9.0.2 の Ultra Light の新機能 38

カスケード削除

バージョン 9.0.2 での Ultra Light の動作の変更
54

カスタム照合

バージョン 8.0.0 での動作の変更 220

活性

バージョン 8.0.0 での Mobile Link の新機能
208

空文字列

バージョン 8.0.0 での動作の変更 221

監査

バージョン 6.0.2 の新機能 290

関数

8.0.2 の新機能 141, 142

関数のインデックス

バージョン 9.0.0 の新機能 97

管理機能の強化

バージョン 7.0.0 254

管理ツール

バージョン 9 の新機能の概要 10

き

キーセット駆動型カーソル

バージョン 8.0.0 の新機能 190

規格

508 条準拠 187

規格と互換性

508 条準拠 187

規則

表記 xi

起動クラス

8.0.2 での Mobile Link の新機能 148

キャッシュ

バージョン 7.0.0 の新機能 258

バージョン 7.0.2 の新機能 236

バージョン 9.0.1 の強化 72

キャッシュ・ウォーミング

バージョン 9.0.1 の新機能 63

キャッシュ・サイズ

バージョン 9.0.1 の強化 72

競合解決トリガ

バージョン 9.0.2 の SQL Remote の新機能 35

バージョン 9.0.2 の新機能 18

共通テーブル式

バージョン 9.0.0 の新機能 95

強力な暗号化

バージョン 8.0.0 での Ultra Light の新機能
209

バージョン 8.0.0 の新機能 184
バージョン 9.0.2 の新機能 27

く

クエリ・エディタ
 バージョン 8.0.0 の新機能 187
 バージョン 9.0.1 の強化 58
クエリ最適化
 9.0.2 の Ultra Light の新機能 39
クエリ実行プラン
 9.0.2 の Native Ultra Light for Java 機能 37
クエリの最適化
 バージョン 8.0.0 の新機能 181
クエリの処理
 バージョン 8.0.0 の新機能 181
クエリ・プラン
 バージョン 8.0.0 での Ultra Light の新機能 211
 バージョン 8.0.0 の新機能 188
組み込みデータベース
 Adaptive Server Anywhere 3
 Ultra Light 3
クライアント・オプションとしてのアドレスとタイプの設定
 8.0.1 の新機能 170
クライアント／サーバ通信
 バージョン 8.0.0 の新機能 196, 197
クライアント・ポートの指定
 バージョン 7.0.1 での Ultra Light の新機能 251
クラスタード・インデックス
 8.0.2 の新機能 140
クラスタード・ハッシュ GROUP BY
 バージョン 9.0.1 の新機能 63
グラフィカルなプラン
 8.0.2 の新機能 141
ワーク・テーブル 141
グローバル・オートインクリメント
 バージョン 7.0.0 の新機能 256, 266
 バージョン 8.0.0 での Ultra Light の新機能 210

 バージョン 8.0.1 での動作の変更 175
グローバル・テンポラリ・テーブル
 8.0.2 の新機能 143
グローバル変数
 8.0.2 の新機能 147

け

警告メッセージ
 9.0 での W プレフィックスの新機能 120
計算カラム
 LOAD TABLE による再計算 174
 バージョン 6.0.3 の新機能 276
結果
 バージョン 9.0.1 の Interactive SQL の強化 68
 バージョン 9.0.1 の Sybase Central の強化 69
限界
 バージョン 8.0.0 の新機能 193
言語ユーティリティ [dblang]
 バージョン 7.0.2 の新機能 238
 バージョン 9.0.1 で使用されなくなった機能 85
検証
 バージョン 7.0.0 の新機能 255
 バージョン 8.0.0 の新機能 191
検証ユーティリティ [dbvalid]
 バージョン 9.0.1 での動作の変更 86
 バージョン 9.0.1 の強化 62

こ

更新
 8.0.2 の新機能 140
高速ランチャ
 バージョン 9.0.0 の新機能 113
コールバック関数
 9.0.2 の Ultra Light の同期の強化 41
 バージョン 9.0.2 の新機能 24
互換性
 クライアント／サーバ 306
 データベースとデータベース・サーバ 306

- 問題 306
- 互換性ライブラリ
 - ODBC アプリケーション 351
 - アップグレード 347
 - インストール 348
 - 概要 327
 - ロケーション 351
- コスト・モデル
 - バージョン 8.0.0 の新機能 181
- コマンド・ライン・ユーティリティ
 - アップグレード 314
 - 複数バージョン 314
- [コマンド履歴] ダイアログ :9.0.0 の新機能 115
- コメント
 - バージョン 8.0.0 での動作の変更 221
- コンソール・ユーティリティ [dbconsole]
 - バージョン 9.0.1 の強化 66
- さ**
- サーバ・エラー・コード
 - バージョン 9.0.0 での Mobile Link の動作の変更 132
- サーバから開始する同期
 - バージョン 9 の新機能 4
- [サーバ起動オプション] ダイアログ
 - バージョン 9.0.2 の新機能 29
- サーバ起動同期
 - 9.0.0 での Mobile Link の新機能 115
- サーバ検索ユーティリティ (dblocate)
 - バージョン 9.0.1 の強化 66
- サーバのログイン
 - バージョン 8.0.0 の新機能 193
- サーバ名
 - バージョン 6.0.3 での最大長の変更 287
- サーバ・メッセージ・ウィンドウ
 - バージョン 9.0.2 の新機能 29
- サービス
 - バージョン 9.0.1 の強化 70
- [サービス作成] ウィザード
 - バージョン 9.0.1 の強化 70
- サービス作成ユーティリティ [dbsvc]
 - バージョン 7.0.1 の新機能 244
 - バージョン 9.0.2 での動作の変更 48
 - バージョン 9.0.2 の新機能 23
- [サービス] プロパティ・シート
 - バージョン 9.0.1 の強化 70
- 再起動可能なダウンロード
 - 9.0.1 の Ultra Light の新機能 80
- 再帰ユニオン
 - バージョン 9.0.0 の新機能 95
- 最終ダウンロード時間
 - バージョン 8.0.0 での Ultra Light の新機能 215
- 最終ダウンロード・タイムスタンプ
 - バージョン 8.0.2 での Ultra Light の新機能 154
 - バージョン 8.0.0 での Mobile Link の新機能 207
- 最終ダウンロードのタイムスタンプ
 - 常に最初に渡される 133
- 最初のローの最適化オプション
 - バージョン 8.0.0 の新機能 182
- 作成者 ID
 - Palm Computing Platform アプリケーション 321
- サブクエリ
 - 9.0.1 の Ultra Light の新機能 78
- サブスクリプション
 - バージョン 8.0.0 での Mobile Link の新機能 202
- サポート
 - ニュースグループ xvii
- サポートされるプラットフォーム
 - Ultra Light 9.0.0 の変更 135
- 参照整合性
 - 9.0.2 の Ultra Light の同期の強化 41
 - バージョン 9.0.2 での Ultra Light の動作の変更 54
- 参照整合性アクション
 - バージョン 6.0.3 での動作の変更 286
- サンプル
 - Adaptive Server Anywhere 13

Mobile Link 12
Ultra Light 11
説明 11
サンプル証明書
バージョン 9.0.2 での動作の変更 49
サンプル・データベース
バージョン 8.0.0 での動作の変更 221

し

識別子
8.0.1 での角カッコによる区切り 165
バージョン 7.0.0 での最大長の変更 271
バージョン 9.0.1 での動作の変更 87
システム・データ・ソース
バージョン 7.0.2 の新機能 239
システム・テーブル
バージョン 7.0.0 の新機能 260
システム・トリガ
バージョン 6.0.3 での動作の変更 286
システム・パス
ユーティリティ 314
失敗したダウンロードの再開
9.0.1 の新機能 dbmlsync 74
集合関数
バージョン 9.0.0 の新機能 97
集合関数外部参照
バージョン 8.0.0 での動作の変更 217
週の最初の曜日
バージョン 7.0.2 での動作の変更 238
終了コード
バージョン 7.0.2 の Interactive SQL の新機能 237
バージョン 9.0.0 での Mobile Link [dbmlsync] の拡張 118
出力リダイレクション
バージョン 7.0.0 の新機能 264
ジョイン
8.0.2 の新機能 141
ジョイン・アルゴリズム
バージョン 8.0.0 の新機能 181
消去ユーティリティ [dberase]

バージョン 9.0.2 での動作の変更 48
照合
バージョン 6.0.3 での動作の変更 287
バージョン 6.0.3 の新機能 277
バージョン 7.0.2 の新機能 236
バージョン 8.0.0 では廃止 225
バージョン 8.0.0 の新機能 200
バージョン 9.0.1 の新機能 67
バージョン 9.0.2 の新機能 22
使用されなくなった機能
バージョン 8.0.1 174, 177
バージョン 8.0.2 158
証明書
取得場所 159
バージョン 9.0.2 での動作の変更 49
初期化ユーティリティ [dbinit]
バージョン 9.0.0 の新機能 107
バージョン 9.0.1 の強化 62
シリアル・プロトコル
8.0.2 から Mobile Link でのサポート終了 158
8.0.2 から Ultra Light でのサポート終了 160
新機能
Ultra Light バージョン 8.0.0 208
バージョン 6.0.1 296
バージョン 6.0.2 290, 293
バージョン 6.0.3 276, 283
バージョン 7.0.0 254, 266, 267
バージョン 7.0.1 244, 246, 249
バージョン 7.0.2 236
バージョン 7.0.3 232
バージョン 8.0.0 180
バージョン 8.0.0 Adaptive Server Anywhere 180
バージョン 8.0.0 Mobile Link 202
バージョン 8.0.1 164
バージョン 8.0.2 140
バージョン 9.0.0 92
バージョン 9.0.1 58
バージョン 9.0.2 18

す

- スウェーデン語
 - バージョン 9.0.2 の新機能 22
- スキーマのアップグレード
 - 9.0.1 の Ultra Light の強化 80
- スキーマのアップグレードの監視
 - 9.0.1 の Ultra Light の強化 80
- スクリプト
 - アップグレード 319
- スクリプトの自動生成
 - 8.0.1 での Ultra Light の新機能 172
- スクリプトの生成
 - バージョン 8.0.0 での Mobile Link の新機能 207
- スクリプト・バージョン
 - バージョン 9.0.1 で設定可能 73
- スケジューリング
 - 9.0.0 での Mobile Link [dbmlsync] の変更 118
 - バージョン 7.0.0 の新機能 254
 - バージョン 7.0.1 の Mobile Link の新機能 247
 - バージョン 9.0.0 の新機能 101
- スケジュール
 - 8.0.2 での Mobile Link [dbmlsync] の変更点 159
 - 8.0.2 での動作変更 157
- スケジュールされたイベント
 - 8.0.2 での動作変更 157
- スタック・サイズ
 - バージョン 6.0.1 での動作の変更 302
- ストアド・プロシージャ
 - バージョン 6.0.3 の新しい Java 機能 280
 - バージョン 8.0.0 での動作の変更 219
 - バージョン 8.0.0 の新機能 188
- ストアド・プロシージャ・ソース・フォーマット
 - バージョン 8.0.0 の新機能 196
- ストアド・プロシージャの結果セット
 - バージョン 9.0.0 の新機能 96
- スペイン語
 - バージョン 9.0.2 の新機能 22
- スレッド
 - 8.0.2 での Ultra Light Java の新機能 153

- バージョン 6.0.1 の新機能 299
- バージョン 6.0.3 での Java の新機能 281
- バージョン 7.0.2 の Mobile Link の新機能 240

せ

- 静的インタフェース
 - 定義 121
- 制約
 - バージョン 9.0.0 の新機能 98
- セキュリティ
 - 8.0.2 での Ultra Light の新機能 154
 - バージョン 8.0.0 での Ultra Light の新機能 209
 - バージョン 9.0.2 での Ultra Light の動作の変更 54
 - バージョン 9.0.2 の新機能 27
- セキュリティ・プロトコル・オプション
 - 9.0.2 で廃止された HTTP 51
- セグメント
 - バージョン 8.0.0 での Ultra Light の新機能 213
- 接続 ID
 - バージョン 9.0.1 の強化 71
- [接続] ダイアログ
 - バージョン 9.0.1 での動作の変更 86
- 接続タイムアウト
 - バージョン 8.0.0 での Mobile Link の新機能 205
- 全外部ジョイン
 - バージョン 8.0.0 の新機能 185

た

- タイ語
 - バージョン 9.0.2 の新機能 22
- タイムスタンプ競合への自動対応
 - 8.0.1 の新機能 170
- タイム・ゾーン
 - バージョン 8.0.0 の新機能 186
- ダウンロード確認

バージョン 8.0.0 での Mobile Link の新機能 204
バージョン 9.0.0 での Mobile Link のデフォルト変更 131
ダウンロード・ストリーム
バージョン 8.0.0 での Mobile Link の新しい動作 204
ダウンロード専用同期
バージョン 8.0.0 での Ultra Light の新機能 210
バージョン 8.0.0 での動作の変更 229
タスク・スケジューリング
バージョン 7.0.0 の新機能 254
断片化
バージョン 8.0.0 の新機能 192
断片化解除
バージョン 8.0.0 での Ultra Light の新機能 211
バージョン 8.0.0 の新機能 191

ち

チェックサム
バージョン 9.0.1 の新機能 62
チェックポイント・ログ
バージョン 8.0.0 で向上した記憶領域 183
バージョン 8.0.0 の新機能 183
逐次テーブル・スキャン
バージョン 8.0.0 の新機能 182
中国語
バージョン 9.0.2 の新機能 22
抽出テーブル
9.0.1 の Ultra Light の新機能 78
抽出ユーティリティ
バージョン 7.0.0 での動作の変更 272
バージョン 7.0.0 の新機能 266
バージョン 9.0.2 での動作の変更 52

つ

通信の暗号化

バージョン 9.0.2 の新機能 27

て

ディスクが満杯
バージョン 9.0.2 の新機能 24
ディスク領域
バージョン 9.0.2 の新機能 24
データ・ソース
バージョン 9.0.2 の新機能 24
データ・ソース・ユーティリティ [dbdsn]
バージョン 7.0.2 の新機能 239
バージョン 9.0.2 での動作の変更 48
バージョン 9.0.2 の新機能 24
[データのアンロード] ダイアログ
バージョン 9.0.2 の新機能 25
データベース
アップグレード 306, 307, 356
バージョン 5 のアップグレード 328
ファイル・フォーマットのアップグレード 310
[データベース圧縮] ウィザード
バージョン 9.0.1 の強化 70
[データベース・アップグレード] ウィザード
使用 308
バージョン 8.0.1 の新機能 166
バージョン 9.0.1 の強化 70
[データベース・アンロード] ウィザード
使用 310
[データベース移行] ウィザード
バージョン 8.0.1 の新機能 166
データベース管理ユーティリティ
バージョン 5 338
[データベース検証] ウィザード
バージョン 9.0.1 の強化 70
バージョン 9.0.2 の強化 25
[データベース展開] ウィザード
バージョン 9.0.1 の強化 70
データベース内の Java
バージョン 8.0.0 での動作の変更 217

バージョン 9.0.0 での動作の変更 125

データベースのアップグレード
バージョン 9.0.2 で廃止される古いフォーマット 48

データベースの暗号化
バージョン 8.0.0 での Ultra Light の新機能 209

バージョン 9.0.2 の新機能 27

データベースの移行
バージョン 7.0.2 の新機能 238

データベースの再構築
バージョン 9.0.2 での動作の変更 50

データベースの作成
バージョン 8.0.0 での動作の変更 217

[データベース・バックアップ] ウィザード
バージョン 9.0.1 の強化 70

データベース・ファイル
バージョン 8.0.0 での暗号化 185

バージョン 8.0.0 の新機能 185

データベース・ファイルの削除
8.0.2 での Ultra Light の新機能 153

データベース・ファイル・フォーマット
アップグレード 310

バージョン 8.0.0 の新機能 183

バージョン 9.0.2 で廃止される古いフォーマット 48

データベース・プロパティ
9.0.1 の Ultra Light の新機能 81

[データベース・リストア] ウィザード
バージョン 9.0.1 の強化 70

テーブル所有者
9.0.2 の Ultra Light の新機能 39

テーブル・スキャン
バージョン 8.0.0 の新機能 182

テーブル・ヒント
8.0.2 の新機能 144

テクニカル・サポート
ニュースグループ xvii

デッドロックのレポート
バージョン 9.0.2 の新機能 22

デバッグ
バージョン 6.0.2 の新機能 276

デバッグ
動作の変更 286

デフォルト値
バージョン 6.0.3 での動作の変更 286

デフォルトの .SQL ファイル・エディタ
バージョン 9.0.0 の新機能 114

テンポラリ・ファイル
バージョン 6.0.3 の新機能 278

と

同期
バージョン 7.0.1 の Mobile Link の新機能 247

バージョン 8.0.0 での Ultra Light の新機能 210

同期 observer
8.0.2 での Ultra Light の新機能 155

同期エラーのレポート
9.0.2 の新しい Native Ultra Light for Java 機能 37

同期オブザーバ
9.0.1 の Ultra Light の強化 80

同期結果
8.0.2 での Ultra Light の新機能 155

同期コールバック関数
9.0.2 の Ultra Light の新機能 41

同期サイト
バージョン 8.0.0 では廃止 227

同期スクリプト
バージョン 7.0.0 での動作の変更 273

バージョン 8.0.0 での Mobile Link の新機能 207

同期テンプレート
バージョン 8.0.0 では廃止 227

同期の RSA 暗号化
8.0.2 での Ultra Light の新機能 154

同期の定義
バージョン 8.0.0 では廃止 227

同期パラメータ
バージョン 9.0.2 での Ultra Light の動作の変更 54

統計情報

- バージョン 9.0.2 での動作の変更 47
- 統計情報付きのグラフィカルなプラン
 - 8.0.2 の新機能 141
- 統計スクリプト
 - バージョン 8.0.0 での Mobile Link の新機能 206
- 統合化ログイン
 - バージョン 9.0.1 の強化 66
 - バージョン 9.0.2 の新機能 26
- 統合データベース
 - アップグレード 317
- 動作の変更
 - バージョン 6.0.1 302
 - バージョン 6.0.2 294
 - バージョン 6.0.3 286, 288
 - バージョン 7.0.0 269, 272, 273
 - バージョン 7.0.2 241
 - バージョン 7.0.3 233
 - バージョン 8.0.0 217, 227, 228
 - バージョン 8.0.1 174
 - バージョン 8.0.2 156
 - バージョン 9.0.0 125
 - バージョン 9.0.1 84
 - バージョン 9.0.2 47
- 同時アップロード
 - バージョン 8.0.0 での Mobile Link の新機能 205
- 動的キャッシュ・サイズ決定
 - バージョン 7.0.0 の新機能 258
 - バージョン 7.0.2 の新機能 236
- 独立性レベル
 - FROM 句での設定 166
- トラブルシューティング
 - エイリアスは参照より前に定義する 241
 - コマンド・ライン・ユーティリティ 314
 - バージョン 7.0.2 の新機能 237
- トランザクション
 - 9.0.1 の Ultra Light 動的 SQL の新機能 77
- トランザクション・ログ
 - バージョン 9.0.2 での動作の変更 48
 - バージョン 9.0.2 の新機能 26
- トランザクション・ログのオフセット

- データベース・ファイルのアップグレード 310
- トランザクション・ログ・ユーティリティ [dblog]
 - バージョン 6.0.3 での SQL Remote の新機能 284
 - バージョン 6.0.3 の新機能 282
 - バージョン 9.0.2 での動作の変更 48
- トランスポート・レイヤ・セキュリティ 証明書の取得場所 159
 - バージョン 7.0.1 の Mobile Link の新機能 246
 - バージョン 7.0.1 の新機能 250
 - バージョン 9.0.2 の新機能 27
- トリガ
 - 8.0.2 の新機能 146
 - バージョン 8.0.0 での動作の変更 220

な

- 名前付き制約
 - バージョン 9.0.0 の新機能 98

に

- ニュースグループ
 - テクニカル・サポート xvii

ね

- ネイティブな Apache リダイレクタ
 - バージョン 9.0.2 の新機能 30
- ネスト・ループ・セミジョイン
 - 8.0.2 の新機能 141
- ネスト・ループ非セミジョイン
 - 8.0.2 の新機能 141
- ネットワーク通信パラメータ
 - 8.0.2 の新機能 144

は

バージョン

同期スクリプトのアップグレード 319

バージョン 6.0.1

動作の変更 302

バージョン 6.0.1 の新機能 295

バージョン 6.0.2

新機能 290, 293

バージョン 6.0.2 の新機能 289

バージョン 6.0.3

新機能 283

動作の変更 288

バージョン 6.0.3 の新機能 275

バージョン 7.0.0

新機能 266, 267

動作の変更 269, 272, 273

バージョン 7.0.0 の新機能 253

バージョン 7.0.1

新機能 244, 246, 249

バージョン 7.0.1 の新機能 243

バージョン 7.0.2

新機能 236

動作の変更 241

バージョン 7.0.2 の新機能 235

バージョン 7.0.3

新機能 232

動作の変更 233

バージョン 7.0.3 の新機能 231

バージョン 8.0.0

Adaptive Server Anywhere の新機能 180

Mobile Link の新機能 202

Ultra Light の新機能 208

新機能 180

動作の変更 217, 227, 228

バージョン 8.0.0 の新機能 179

バージョン 8.0.1

新機能 164

動作の変更 174

バージョン 8.0.1 の新機能 163

バージョン 8.0.2

新機能 140

動作の変更 156

バージョン 8.0.2 の新機能 139

バージョン 9.0.0 の新機能 91

バージョン 9.0.1

新機能 58

動作の変更 84

バージョン 9.0.1 の新機能 57

バージョン 9.0.2

新機能 18

動作の変更 47

バージョン 9.0.2 の新機能 17

バージョン 9

Adaptive Server Anywhere の新機能の概要 5

Mobile Link の新機能の概要 9

Ultra Light の新機能の概要 7

概要 2

管理ツールの新機能の概要 10

新機能 92

動作の変更 125

バージョンごとの変更箇所

6.0.1 302

6.0.2 290, 293

6.0.3 283

7.0.0 266, 267

7.0.1 244, 246, 249

7.0.2 236, 241

7.0.3 232, 233

8.0.0 180, 202, 208, 217

8.0.1 164, 174

8.0.2 140, 156

9.0.0 92, 125

9.0.1 58, 84

9.0.2 18, 47

パーソナル・サーバ

バージョン 9.0.0 での動作の変更 126

廃止された機能

バージョン 9.0.0 130

バージョン 9.0.1 84

バージョン 9.0.2 47

変更の可能性 47

ハイパネーション

バージョン 8.0.0 の新機能 200

配備

Ultra Light データベースのアップグレード

- 321
- イベント・ビューア・メッセージのレジストリ・エントリ 27
- バージョン 8.0.0 での Ultra Light の新機能 212
- パケット・サイズ
 - バージョン 6.0.2 での動作の変更 294
 - バージョン 8.0.0 での動作の変更 224
- パス
 - 設定 269
- パスワード
 - バージョン 6.0.3 での SQL Remote の動作の変更 288
 - バージョン 9.0.1 での動作の変更 86
- バックアップ
 - バージョン 8.0.0 の新機能 192
 - [バックアップ・イメージ作成]ウィザード
 - バージョン 9.0.1 の強化 70
- バックアップ・ユーティリティ [dbbackup]
 - バージョン 9.0.1 の強化 65
- ハッシュ・サイズ
 - バージョン 7.0.0 の新機能 258
 - バージョン 8.0.0 では廃止 226
- ハッシュ・ジョイン
 - バージョン 8.0.0 の新機能 181
- ハッシュ・セミジョイン
 - 8.0.2 の新機能 141
- ハッシュ非セミジョイン
 - 8.0.2 の新機能 141
- パフォーマンス
 - 8.0.2 の新機能 145
 - データベース・ファイルのアップグレード 310
 - バージョン 8.0.0 での Mobile Link の新機能 203, 204
 - バージョン 8.0.0 での新機能 181
 - バージョン 9 の新機能 4
- パフォーマンス・モニタ
 - 8.0.2 の新機能 145
 - バージョン 6.0.3 の新機能 279
- パブリケーション
 - バージョン 8.0.0 での Mobile Link の新機能 202

- バージョン 8.0.0 での Ultra Light の新機能 210

ひ

- ヒストグラム
 - バージョン 9.0.2 での動作の変更 47
- 非同期 I/O
 - バージョン 9.0.1 の新機能 70
- 非同期テーブル
 - バージョン 7.0.1 での Ultra Light の新機能 250
- ビューの依存性
 - バージョン 9.0.1 の強化 65
- 表記
 - 規則 xi
- 標準偏差
 - バージョン 9.0.0 の新機能 97

ふ

- ファイアウォール
 - バージョン 6.0.3 の新機能 278
 - バージョン 7.0.0 の新機能 257
- ファイル
 - バージョン 7.0.0 の新機能 260
- ファイル非表示ユーティリティ [dbfhide]
 - 8.0.2 の新機能 144
 - バージョン 9.0.2 の新機能 26
- ファイル・フォーマット
 - アップグレード 310
 - バージョン 8.0.0 の新機能 183
- ファイルベースのダウンロード
 - 9.0.0 での Mobile Link の新機能 115
- フィードバック
 - 提供 xvii
 - マニュアル xvii
- フィンランド語
 - バージョン 9.0.2 の新機能 22
- ブール論理
 - バージョン 8.0.0 での動作の変更 223

フェッチ操作
 8.0.2 の新機能 147
 複数バージョン
 Adaptive Server Anywhere 314
 プランのキャッシュ
 バージョン 8.0.0 の新機能 183
 プランのグラフィカル表示
 バージョン 8.0.0 の新機能 188
 バージョン 9.0.0 の新機能 113
 プレフィクス・ファイル
 9.0.2 の強化 38
 [プロキシ・テーブル作成] ウィザード
 バージョン 9.0.1 の強化 70
 プロシージャ
 8.0.2 の新機能 142
 プロセス生成ユーティリティ [dbspawn]
 バージョン 6.0.3 の新機能 282
 バージョン 7.0.2 の新機能 238
 ブロック・ネストされたループ・ジョイン
 バージョン 8.0.0 の新機能 181
 分散
 バージョン 9.0.0 の新機能 97
 分散トランザクション
 バージョン 7.0.0 の新機能 256
 分散読み込み
 8.0.2 の新機能 145
 文ベースのアップロード
 バージョン 8.0.0 での Mobile Link の新機能 203
 へ
 並列インデックス・スキャン
 バージョン 9.0.1 の新機能 63
 ページ・サイズ
 バージョン 6.0.3 での動作の変更 286
 バージョン 7.0.0 の新機能 259
 ヘブライ語
 バージョン 9.0.2 の新機能 22
 変換
 文字セット 142
 変数

8.0.2 の新機能 147

ほ

ホーバー・モード
 バージョン 6.0.2 での SQL Remote の新機能 293

ま

マージ・ジョイン
 バージョン 8.0.0 の新機能 181
 マニュアル
 SQL Anywhere Studio viii
 バージョン 8.0.0 の新機能 199
 マニュアルの強化
 バージョン 9.0.2 45
 マルチスレッド・アプリケーション
 8.0.1 での Ultra Light の新機能 171
 マルチデータベースのサポート
 9.0.1 の Ultra Light の新機能 81

め

メッセージ
 9.0.1 の Mobile Link の新機能 72
 メッセージのロギング
 バージョン 6.0.2 での SQL Remote の新機能 293
 メッセージ・リンク
 バージョン 6.0.3 での SQL Remote の動作の変更 288
 メッセージ・リンク・オプション
 バージョン 6.0.3 での SQL Remote の新機能 283

も

文字セット
 バージョン 6.0.3 での Mobile Link の新機能

285
バージョン 8.0.0 での動作の変更 223
変換 297
文字セットの変換
8.0.2 の新機能 142
バージョン 8.0.0 での動作の変更 221
モバイル・コンピューティング
Adaptive Server Anywhere 3
Mobile Link 3
Ultra Light 3

ゆ

ユーザ ID マッピング
8.0.1 の新機能 170
ユーザが提供する選択性推定
バージョン 8.0.0 での動作の変更 217
ユーザ定義関数
バージョン 6.0.3 の新しい Java 機能 280
バージョン 8.0.0 での動作の変更 219
非確定 169
ユーザ・データ・ソース
バージョン 7.0.2 の新機能 239
ユーザ認証
8.0.1 での Ultra Light Java の新機能 172
バージョン 7.0.1 での Mobile Link の新機能 246
バージョン 8.0.0 での Mobile Link の新機能 205
バージョン 8.0.0 での Ultra Light の新機能 208
ユーティリティ
8.0.2 の新機能 147
ユーティリティ・データベース
バージョン 6.0.3 の新機能 279
ユーロ通貨
バージョン 6.0.3 の新機能 277
ユニークな識別子
8.0.2 の新機能 140

よ

要求のロギング
8.0.2 の新機能 145
バージョン 9.0.1 の強化 66
バージョン 9.0.2 の新機能 26
要求レベル・ログ vii
要求ログ
バージョン 9.0.2 の新機能 26
読み込み専用データベース
バージョン 6.0.3 の新機能 276

ら

ライセンス
バージョン 7.0.2 の新機能 236
バージョン 8.0.0 の新機能 200
ライセンス・ユーティリティ [dblic]
バージョン 7.0.2 の新機能 236
ライト・ファイル
バージョン 9.0.1 での廃止 84
ラテラル抽出テーブル
バージョン 9.0.0 の新機能 98

り

リスナ
9.0.0 での Mobile Link の新機能 115
リターン・コード
バージョン 7.0.2 の Interactive SQL の新機能 237
バージョン 9.0.0 での Mobile Link [dbmlsync] の拡張 118
リダイレクタ
8.0.2 での Java リダイレクタの新機能 151
9.0.1 での Apache Web サーバのサポート 76
[リモート・サーバ作成] ウィザード
バージョン 9.0.1 の強化 69
バージョン 9.0.2 の強化 19
リモート・データ・アクセス
バージョン 6.0.1 の新機能 297

リモート・データベース
アップグレード 319

れ

レプリケーション・エージェント
バージョン 8.0.0 の新機能 202
連邦リハビリテーション法
508 条 187

ろ

ローカル・テンポラリ・テーブル
8.0.2 の新機能 143
ローの順序
バージョン 8.0.0 での動作の変更 218
ロギング
バージョン 8.0.0 での Mobile Link の新しい動
作 207
ログ変換ユーティリティ [dbtran]
バージョン 6.0.2 の新機能 292
バージョン 6.0.3 での SQL Remote の新機能
284
バージョン 6.0.3 の新機能 281
バージョン 9.0.2 での動作の変更 48

わ

ワーク・テーブル
8.0.2 の新機能 141
バージョン 8.0.0 の新しい動作 181
ワンボタンでの同期
9.0.2 の Ultra Light for M-Business Anywhere の
新機能 36

