SQL Anywhere® 16
変更点とアップグレード

バージョン 16.0

2013 年 2 月

Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

バージョン 16.0
2013 年 2 月

© 2013 SAP AG or an SAP affiliate company.All rights reserved.

次の条件に従うかぎり、このマニュアルの全部または一部を使用、印刷、再生、配布することができます。1) マニュアルの全部または一部にかかわらず、すべてのコピーにこの情報またはマニュアル内のその他の版権と商標の表示を含めること。2) マニュアルに変更を加えないこと。3) SAP 以外の人間がマニュアルの著者または情報源であるかのように示す行為をしないこと。ここに記載された情報は事前の通知なしに変更されることがあります。

SAP AG およびディストリビュータが販売しているソフトウェア製品には、他のソフトウェアベンダー独自のソフトウェアコンポーネントが含まれているものがあります。国内製品の仕様は変わることがあります。

これらの資料は SAP AG および関連会社 (SAP グループ) が情報のみを目的として提供するものであり、いかなる種類の表明または保証も行うものではなく、SAP グループはこの資料に関する誤りまたは脱落について責任を負わないものとします。SAP グループの製品およびサービスに関する保証は、かかる製品およびサービスに付属している明確な保証文書がある場合、そこで明記されている保証に限定されます。ここに記載されているいかなる内容も、追加保証を構成するものとして解釈されるものではありません。

ここに記載された SAP および他の SAP 製品とサービス、ならびに対応するロゴは、ドイツおよび他の国における SAP AG の商標または登録商標です。http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark を参照してください。

目次

はじめに vii

バージョン 16.0 の新機能 1

製品全体の新機能 1
製品全体の動作の変更 7
製品全体の廃止予定機能および削除された機能 7
SQL Anywhere の新機能 9
SQL Anywhere の動作の変更 29
SQL Anywhere の廃止予定機能とサポート終了機能 38
Mobile Link の新機能 40
Mobile Link の動作の変更 49
QAnywhere の新機能 53
Relay Server の新機能 53
Relay Server の動作変更 55
Ultra Light の新機能 56
Ultra Light の動作の変更と廃止予定機能 65
管理ツールの新機能 67

バージョン 12.0.1 の新機能 73

12.0.1 での追加事項：新機能と動作の変更 73
SQL Anywhere の新機能 84
パフォーマンスの強化 90
プロパティとパフォーマンスモニタの統計値 90
SQL Anywhere の動作の変更 94
SQL Anywhere の廃止予定機能とサポート終了機能 95
Mobile Link の新機能 96
Mobile Link の動作の変更 101
SQL Remote の新機能 103
Ultra Light の新機能 104
Ultra Light の動作の変更と廃止予定機能 110
管理ツールの新機能 111

マニュアルの強化 116

バージョン 12.0.0 の新機能 117

製品全体の新機能 117
製品全体の動作の変更 117
SQL Anywhere の新機能 119
SQL Anywhere の動作の変更 146
SQL Anywhere の廃止予定機能とサポート終了機能 153
Mobile Link の新機能 155
Mobile Link の動作の変更 165
SQL Remote の新機能 170
Ultra Light の新機能 171
Ultra Light の動作の変更と廃止予定機能 175
Ultra Light J の新機能 178
Ultra Light J の動作の変更と廃止予定機能 183
管理ツールの新機能 185
マニュアルの強化 195

バージョン 11.0.1 の新機能 197

SQL Anywhere の新機能 197
SQL Anywhere の動作の変更と廃止予定機能 203
Mobile Link の新機能 207
Mobile Link の動作の変更と廃止予定機能 209
Ultra Light の新機能 209
Ultra Light の動作の変更と廃止予定機能 210
Ultra Light J の新機能 210
Ultra Light J の動作の変更と廃止予定機能 211
管理ツールの新機能 211
管理ツールの動作の変更と廃止予定機能 212
製品全体の新機能 213
マニュアルの強化 214

バージョン 11.0.0 の新機能 215

SQL Anywhere 215
Mobile Link 257
SQL Remote 262
Ultra Light 263
Sybase Central と Interactive SQL 270
マニュアルの強化 279
製品全体の機能 280

SQL Anywhere 16 へのアップグレード方法 283

データベースの再構築 (アンロード／再ロード)、データベースのアップグレード、クライアントライブラリの更新を必要とする機能 283
SQL Anywhere サーバのアップグレード 289
アップロードと再ロードの注意事項 291
複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法 292
データベースの再構築 (アンロード／再ロード) 292
バージョン 10 以降のデータベースの再構築手順 294
バージョン 9 以前のデータベースの再構築手順 297
バージョン 10 以降のデータベースのアップグレード手順 304
データベースミラーリングシステムでのアップグレードと再構築 308
トラブルシューティング：データベースのアップグレード 316
Mobile Link のアップグレード 319
Ultra Light のアップグレード 331
SQL Remote のアップグレード 334
SQL Anywhere モニタのアップグレードおよびリソースの移行 335

索引 339

はじめに

このマニュアルでは、SQL Anywhere 16 とそれ以前のバージョンに含まれる新機能について説明します。

バージョン 10.0.1 以前の SQL Anywhere の新機能と動作変更については、ftp://ftp2.ianywhere.jp/public/tech/dbwnja10.pdf を参照してください。

バージョン 16.0 の新機能

バージョン 10 以前の SQL Anywhere の新機能と動作変更については、ftp://ftp2.ianywhere.jp/public/tech/dbwnja10.pdf を参照してください。

製品全体の新機能

次に、バージョン 16.0 で導入された製品全体の追加機能を示します。サポートされているプラットフォームおよびバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

新しいセキュリティモデル：ロールベースのアクセス制御 (RBAC)

SQL Anywhere 16.0 には、以前の権限およびパーミッションセキュリティモデルに代わって、ロールベースおよび権限ベースの新しいセキュリティモデルが導入されています。新しいロールベースのセキュリティモデルにより、ユーザが実行できる権限付きのタスクのきめ細かい制御が可能になり、アクセス制御の管理が簡素化されます。

以前からの SQL Anywhere の利用者がバージョン 16.0 にアップグレードする場合、データベースのアップグレード後もアプリケーションが引き続き機能するように下位互換が保証されています。モデル間の相違点、アップグレード時に自動的に実行される内容、SQL Anywher を使用するアプリケーションに実行すべき内容 (GRANT 文と REVOKE 文の呼び出しの更新など) について、よく理解できるように特別な章を設けています。「ロールベースのセキュリティへのアップグレード」『SQL Anywhere サーバ データベース管理』を参照してください。

チュートリアルについては、「チュートリアル：ロールと権限の付与 (Sybase Central の場合)」『SQL Anywhere サーバ データベース管理』および「チュートリアル：ロールと権限の付与 (SQL の場合)」『SQL Anywhere サーバ データベース管理』を参照してください。

以下に、新しいロールベースのセキュリティモデルをサポートするために提供される新機能と変更機能の概要を示します。

- **ロールベースのセキュリティモデルの概要**　より精度の高い機能を提供することによりセキュリティを強化するために、システム権限とロールの完全なセットを追加しました。システムで実行できる権限付き操作のために、システム権限が作成されています。ロールは権限を組み合わせて論理グループにします。SQL Anywhere には事前定義済みのロールが多数用意されていますが、ユーザが独自のロールを作成することもできます。「ユーザのセキュリティ (ロールと権限)」『SQL Anywhere サーバ データベース管理』を参照してください。

  アップグレード中に、権限、パーミッション、グループはロール、権限、ユーザ拡張ロールに置き換わります。「ロールベースのセキュリティへのアップグレード」『SQL Anywhere サーバ データベース管理』を参照してください。

- **新しい TRUNCATE オブジェクトレベル権限**　指定のテーブルやマテリアライズドビューをトランケートするための、新しいオブジェクトレベル権限 TRUNCATE が追加されています。この権限は、以前のリリースのソフトウェアでは、オブジェクトレベルのパーミッションとして存在していませんでした。「オブジェクトレベル権限」『SQL Anywhere サーバ データベース管理』を参照してください。

- **新しい LOAD オブジェクトレベル権限**　特定のテーブルをロードするための、新しいオブジェクトレベル権限 LOAD が追加されています。この権限は、以前のリリースのソフトウェアでは、オブジェクトレベルのパーミッションとして存在していませんでした。「オブジェクトレベル権限」『SQL Anywhere サーバ データベース管理』を参照してください。

- **ユーザアカウントの自動ロック解除**　root_auto_lock_time (ルートログインポリシーのみ) ログインポリシーオプションと auto_unlock_time ログインポリシーオプションを使用して、ログインの失敗によってデータベースからロックアウトされたユーザ用に自動ロック解除時間を指定できます。これらのオプションは、CREATE LOGIN POLICY 文と ALTER LOGIN POLICY 文で使用できます。「ユーザアカウントの自動ロック解除」『SQL Anywhere サーバ データベース管理』を参照してください。

- **システムプロシージャ**　以下に、ロールベースのセキュリティをサポートするために追加されたシステムプロシージャを示します。

  - **sp_objectpermission システムプロシージャ**　このプロシージャは、指定のオブジェクト、dbspace、またはユーザ名に付与されるオブジェクト権限に関するレポートを生成します。このシステムプロシージャを取得するには、既存のデータベースを再構築してください。「sp_objectpermission システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **sp_displayroles システムプロシージャ**　指定されたシステム権限、システムロール、ユーザ定義ロール、またはユーザ名に付与されたすべてのロールを返すか、またはロールの階層ツリー全体を表示します。「sp_displayroles システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **sp_has_role システムプロシージャ**　プロシージャのインボーカに、指定されたシステム権限またはユーザ定義ロールが付与されているかどうかを返します。「sp_has_role システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **sp_proc_priv システムプロシージャ**　プロシージャの実行に必要なシステム権限のリストを返します。「sp_proc_priv システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **sp_auth_sys_role_info システムプロシージャ**　以前のバージョンの SQL Anywhere から、それに対応する互換ロールへの、権限のマッピングを返します。「sp_auth_sys_role_info システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **sp_sys_priv_role_info システムプロシージャ**　基本となるシステムロールへのシステム権限のマッピングを返します。「sp_sys_priv_role_info システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **データベースオプション**　以下に、ロールベースのセキュリティをサポートするために追加されたシステムオプションを示します。

  - **min_role_admins オプション**　ロールに必要な管理者の最小数を設定します。「min_role_admins オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

- **db_publisher オプション**　このオプションはデータベースパブリッシャのユーザ ID を格納します。設定方法は他のデータベースオプションと同様ですが、GRANT PUBLISH 文や REVOKE PUBLISH 文を使用して設定することもできます。「db_publisher オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **データベースサーバオプション**　ロールベースのセキュリティをサポートするため、次のデータベースサーバオプションの動作が変更されています。

  - **-gu データベースサーバオプション**　値を DBA に設定した場合、SERVER OPERATOR 権限を持つユーザのみがデータベースを作成または削除できます。「-gu データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

  - **-gk データベースサーバオプション**　値を DBA に設定した場合、SERVER OPERATOR 権限を持つユーザのみが、dbstop ユーティリティを使用してデータベースサーバをシャットダウンできます。「-gk データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

  - **-gd データベースサーバオプション**　値を DBA に設定した場合、SERVER OPERATOR 権限を持つユーザのみがデータベースを起動できます。「-gd データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

  - **-gl データベースサーバオプション**　値を DBA に設定した場合、LOAD ANY TABLE 権限または ALTER ANY TABLE 権限を持つユーザのみが LOAD 文を実行できます。UNLOAD 文を実行するには SELECT ANY TABLE 権限が必要です。「-gl データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **新しい SQL 文と変更された SQL 文**　以下に、ロールベースのセキュリティをサポートするために追加、または変更された SQL 文を示します。

  - **GRANT 文の変更**　ロールと権限を付与できるように GRANT 文が強化されています。「GRANT 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

    権限、パーミッション、およびグループのメンバーシップを付与するための古い構文は、サポートされたままですが、廃止される予定です。「GRANT 文 (権限とグループ) (廃止予定)」『SQL Anywhere サーバ データベース管理』を参照してください。

  - **REVOKE 文の変更**　ロールと権限を呼び出せるように REVOKE 文が強化されています。「REVOKE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

    権限、パーミッション、およびグループのメンバーシップを付与するための古い構文は、サポートされたままですが、廃止される予定です。「REVOKE 文 (権限とグループ) (廃止予定)」『SQL Anywhere サーバ データベース管理』を参照してください。

  - **新しい CREATE ROLE 文**　ロールの作成または置き換え、ユーザからロールへの変換、またはロールでのロール管理者の管理を行います。「CREATE ROLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **新しい ALTER ROLE 文**　互換ロール (SYS_AUTH_ で始まるロール) をユーザ定義ロールに移行し、互換ロールを削除します。「ALTER ROLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

○ **新しい DROP ROLE 文**　データベースからユーザ定義ロールまたは互換ロールを削除するか、ユーザ拡張ロールを通常のユーザに変換します。「DROP ROLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

○ **新しい GRANT ROLE SYS_RUN_REPLICATION_ROLE 文**　この文は、レプリケーションの実行に必要なロールを付与します。「GRANT ROLE SYS_RUN_REPLICATION_ROLE 文 [Mobile Link] [SQL Remote]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

○ **新しい GRANT ROLE SYS_REPLICATION_ADMIN_ROLE 文**　この文は、レプリケーションの管理に必要なロールを付与します。「GRANT ROLE SYS_REPLICATION_ADMIN_ROLE 文 [Mobile Link] [SQL Remote]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

○ **GRANT PUBLISH 文と REVOKE PUBLISH 文の変更**　以前は、GRANT PUBLISH 文と REVOKE PUBLISH 文は ISYSAUTHORITY システムテーブル内のデータベースパブリッシャ情報を更新していました。しかし、データベースパブリッシャユーザ ID は、ロールベースのセキュリティの一部として、db_publisher データベースオプションとして格納されるようになりました。現在、これらの文は ISYSUSERAUTHORITY を更新するのではなく、db_publisher データベースオプションの値を更新します。「GRANT PUBLISH 文 [SQL Remote]」『SQL Anywhere サーバ SQL リファレンス』と「REVOKE PUBLISH 文 [SQL Remote]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

● **カタログの変更**　以下に、ロールベースのセキュリティをサポートするために行われたカタログの変更を示します。変更されているのはカタログテーブルですが、アクセスできるのは対応するシステムビューのみであるため、ビューも示します。

カタログの項目	変更
ISYSROLEGRANT システムテーブル/「SYSROLEGRANT システムビュー」『SQL Anywhere サーバ SQL リファレンス』	新規。ロールのメンバーシップとメンバーシップのタイプに関する情報を格納します。
「SYSROLEGRANTS 統合ビュー」『SQL Anywhere サーバ SQL リファレンス』	新規。SYSROLEGRANT と同じですが、2 つのカラム role_name と grantee_name が追加されています。
ISYSROLEGRANTEXT システムテーブル/「SYSROLEGRANTEXT システムビュー」『SQL Anywhere サーバ SQL リファレンス』	新規。SET USER 権限と CHANGE PASSWORD 権限のための構文拡張を格納します。
「SYSGROUP 互換ビュー」『SQL Anywhere サーバ SQL リファレンス』	このビューは以前はシステムビューでしたが、現在は互換ビューです。
「SYSGROUPS 互換ビュー」『SQL Anywhere サーバ SQL リファレンス』	このビューは以前は統合ビューでしたが、現在は互換ビューです。

カタログの項目	変更
「SYSUSER システムビュー」『SQL Anywhere サーバ SQL リファレンス』	新しいカラム：dual_password
「SYSUSERAUTHORITY 互換ビュー (旧式)」『SQL Anywhere サーバ SQL リファレンス』	このビューはシステムビューから互換ビューに変更されています。基本となる ISYSAUTHORITY システムテーブルは削除されていますが、このビューは互換性のために保持されています。
「SYSTABLEPERM システムビュー」『SQL Anywhere サーバ SQL リファレンス』	新しいカラム：loadauth および truncateauth
「SYSTABAUTH 統合ビュー」『SQL Anywhere サーバ SQL リファレンス』	新しいカラム：loadauth および truncateauth

- **ロールと権限をサポートするための SQL Anywhere プラグインの変更**　新しいロールベースのセキュリティモデルをサポートするため、Sybase Central には多くの変更が加えられています。テーブルの作成など、権限付きのデータベースタスクを実行する場合、SQL 文を使用する場合よりも、Sybase Central での実行のためにより多くの権限が必要となることがあります。Sybase Central では、使用中に、表示のためのフォルダ (ビュー、ユーザなど) の移植などの、追加の権限付きタスクが実行されるためです。

  タスクの実行に必要な権限が不明な場合、Sybase Central でのタスクの実行に関する文書を参照してください。

LDAP ユーザ認証サポート

SQL Anywhere では LDAP ユーザ認証がサポートされるようになりました。以下に、LDAP ユーザ認証をサポートするために追加または変更された内容を示します。 「LDAP 認証」『SQL Anywhere サーバ データベース管理』を参照してください。

- **ルートログインポリシーの変更**　LDAP ユーザ認証をサポートするため、ルートログインポリシーに次のログインポリシーオプションが追加されています。

  - ldap_primary_server
  - ldap_secondary_server
  - ldap_auto_failback_period
  - ldap_failover_to_std
  - ldap_refresh_dn

  「ルートログインポリシー」『SQL Anywhere サーバ データベース管理』を参照してください。

- **システムプロシージャ**　LDAP ユーザ認証をサポートするため、次のシステムプロシージャが追加または強化されています。

  - 「sa_get_ldapserver_status システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』 (新規)
  - 「sa_get_user_status システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』 (強化)

  sa_get_user_status システムプロシージャにより、新しいカラム (user_dn、user_dn_cached_at、password_change_state、password_change_first_user、password_change_second_user) がレポートされます。

- **データベースサーバとデータベースオプション**　LDAP ユーザ認証をサポートするため、次のデータベースオプションが追加または強化されています。

  - 「login_mode オプション」『SQL Anywhere サーバ データベース管理』
  - 「trusted_certificates_file オプション」『SQL Anywhere サーバ データベース管理』

- **SQL 文**　LDAP ユーザ認証をサポートするため、次の SQL 文が追加されています。

  - 「CREATE LDAP SERVER 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「ALTER LDAP SERVER 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP LDAP SERVER 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「VALIDATE LDAP SERVER 文」『SQL Anywhere サーバ SQL リファレンス』

  LDAP ユーザ認証をサポートするため、次の SQL 文が強化されています。

  - 「ALTER USER 文」『SQL Anywhere サーバ SQL リファレンス』 (新しい REFRESH DN 構文)
  - 「CREATE LOGIN POLICY 文」『SQL Anywhere サーバ SQL リファレンス』 (新しいログインポリシーオプション)
  - 「ALTER LOGIN POLICY 文」『SQL Anywhere サーバ SQL リファレンス』 (新しいログインポリシーオプション)
  - 「COMMENT 文」『SQL Anywhere サーバ SQL リファレンス』 (LDAP サーバ設定オブジェクトにコメントを付加するための機能)

- **カタログの変更**　LDAP ユーザ認証をサポートするため、次のようにカタログが変更されています。

  - **SYSLDAPSERVER システムビュー (新規)**　SYSLDAPSERVER システムビューには、データベースで設定されている各 LDAP SERVER オブジェクトに対して 1 つのローが含まれます。 「SYSLDAPSERVER システムビュー」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **SYSUSER システムビュー**　SYSUSER システムビューには、LDAP ユーザ認証に関連する 2 つの新しいカラム user_dn と user_dn_cached_at があります。 「SYSUSER システムビュー」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

製品全体の動作の変更

次に、バージョン 16.0 で導入された製品全体の動作の変更を示します。サポートされているプラットフォームおよびバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

その他の製品全体の動作の変更

サンプル証明書の格納ディレクトリの変更　TLS 通信に使用されるサンプル X.509 証明書が、*bin32* および *bin64* ディレクトリから *C:¥Users¥Public¥Documents¥SQL Anywhere 16¥Samples¥Certificates* ディレクトリに移動されています。

新しいディレクトリには以下のファイルも追加されています。

- *rsaclient.id*
- *rsaclient_nopwd.id*
- *rsaroot.key*
- *rsaserver_nopwd.id*

詳細については、*C:¥Users¥Public¥Documents¥SQL Anywhere 16¥Samples¥Certificates¥readme.txt* を参照してください。

SQL Anywhere による TLS バージョン 1.1 のサポート　SQL Anywhere サーバおよびクライアントライブラリは、TLS 接続暗号化と HTTPS の両方について、TLS バージョン 1.1 をサポートするようになりました。TLS バージョン 1.1 は Mac OS X ではサポートされていません。

iAnywhere Solutions Oracle ODBC ドライバの名前の変更　バージョン 16.0 では、SQL Anywhere Oracle ODBC ドライバの名前は **SQL Anywhere 16 - Oracle** になっています。以前のバージョンでは、**iAnywhere Solutions 12 - Oracle** という名前でした。SQL Anywhere ODBC ドライバの参照用の名前は **SQL Anywhere 16** です。

製品全体の廃止予定機能および削除された機能

- **楕円曲線暗号化 (ECC)**　ECC 暗号化はサポートされなくなりました。この変更は、次の機能に影響を及ぼします。

  - IsEccAvailable サーバプロパティが削除されています。

  - -ec データベースサーバオプションは TLS_TYPE プロトコルオプションで ECC を受け付けなくなりました。-ec データベースサーバオプション用の TLS_TYPE プロトコルオプションはマニュアルから削除されましたが、下位互換のためにソフトウェアによるサポートは継続されます。

  - ENCRYPTION 接続パラメータは TLS-TYPE 引数で ECC を受け付けなくなりました。ENCRYPTION 接続用の TLS_TYPE 引数はマニュアルから削除されましたが、下位互換のためにソフトウェアによるこのオプションのサポートは継続されます。

  - 証明書作成ユーティリティ (createcert) は -ec オプションを受け付けなくなり、-t オプションは暗号化タイプで ecc を受け付けなくなりました。

○キーペアジェネレータユーティリティ (createkey) は RSA キーペアの作成のみを行います。**Choose encryption type** プロンプトと **Enter ECC curve** プロンプトは削除されています。

○SQL Anywhere モニタは ECC 暗号化をサポートしていません。ECC 暗号化を使用する、16 より以前のバージョンの Mobile Link サーバをモニタリングするには、以下を実行する必要があります。

●ECC を使用しないネットワークプロトコルオプションの 2 番目のセットを使用して Mobile Link サーバを起動します。

●モニタリング対象のリソースとして Mobile Link サーバを追加します。Mobile Link サーバリソースのポート番号を指定するプロンプトが表示されたら、ECC 暗号化に関連付けられていない番号を指定してください。

○Mobile Link の tls_type プロトコルオプションは tls_type オプションで ECC を受け付けなくなりました。Mobile Link の tls_type プロトコルオプションはマニュアルから削除されましたが、ソフトウェアによるサポートは継続されます。

○Mobile Link クライアントの e2ee_type プロトコルオプションは tls_type オプションで ECC を受け付けなくなりました。Mobile Link クライアントの e2ee_type プロトコルオプションはマニュアルから削除されましたが、ソフトウェアによるサポートは継続されます。

○Outbound Enabler tls_type オプションでは、ECC の選択がサポートされなくなりました。tls_type プロトコルオプションはマニュアルから削除されました。

○Ultra Light C/C++ と Ultra Light J API のメソッドのほか、ECC 設定に関連するプロパティが削除されています。

● **カラム名の送信**　Mobile Link と Ultra Light では、同期中に常にカラム名が送信されるようになりました。カラム名の送信を無効にしようとしても、無視されます。この変更は、次の機能に影響を及ぼします。

○Mobile Link では、dbmlsync 用の SendColumnNames (scn) 拡張オプションは廃止され、OFF に設定しても無視されます。SendColumnNames 拡張オプションはマニュアルから削除されましたが、ソフトウェアによるサポートは継続されます。

○Ultra Light および Ultra Light J クライアントは、Mibile Link サーバデータベースとの同期中に常にカラム名を送信します。カラム名の送信を停止するためのメソッドとアプローチは廃止され、無視されるようになりました。影響を受けるメンバーとパラメータは以下のとおりです。

- Send Column Names 同期パラメータ
- 同期プロファイルオプション
- SyncParms.setSendColumnNames メソッド [Ultra Light J]
- SyncParms.getSendColumnNames メソッド [Ultra Light J]
- ULSyncParms.SendColumnNames プロパティ [Ultra Light.NET]
- ul_sync_info 構造体内の send_column_names [Ultra Light C および Embedded SQL データ型]

SQL Anywhere の新機能

次に、SQL Anywhere バージョン 16.0 の新機能を示します。

サポートされるプラットフォームのリストに加えられた変更については、http://www.sybase.com/detail?id=1061806 を参照してください。

主な機能

次に、SQL Anywhere バージョン 16.0 で導入された主な機能を示します。

パフォーマンスの強化

SQL Anywhere オプティマイザは、クエリ最適化プロセス中に、左側が深いツリー型の構造ではないアクセスプランを考慮するよう強化されています。追加のアクセスプランを考慮することにより、オプティマイザでより効率的な最良プランが見つかり、SQL Anywhere 16.0 での一部の文の実行時間が大幅に短縮されます。「実行プランのコンポーネント」『SQL Anywhere サーバ SQL の使用法』を参照してください。

ROW データ型と ARRAY データ型のサポート

SQL Anywhere では ROWS と ARRAYS という複合データ型のサポートが追加されています。これらのデータ型を使用すると、リストの構造やその値のデータ型を定義できるため、より効率的にリストを保管できます。また、UNNEST 演算子を使用することで、リスト要素へのアクセスをより容易にします。この場合、直接アクセス、二重角カッコを使用したアクセス、または結果セットとしてのアクセスが可能です。VARCHAR カラム内の区切られた文字列としてリストを保管し、sa_split_list で解析する場合は、ARRAY データ型の使用を検討してください。ARRAYS は、それぞれの間に何らかの関連のあるさまざまなオブジェクトを保管する場合に非常に便利です。ROWS は、1 つのオブジェクトと関連する複数の値を保管する場合に便利です。

ROW コンストラクタ、ARRAY コンストラクタなどの複合データ型をサポートするため、新しい SQL 関数が追加されています。新しい演算子 UNNEST も追加されています。

SQL の関数とプロシージャは、IN、OUT、または INOUT 引数として ROW データ型と ARRAY データ型を受け付けます。ローと配列は、SQL ユーザ定義関数からの戻り値のタイプとして使用できます。

ROW 型と ARRAY 型は以下で使用できます。

- SELECT 文、INSERT 文、UPDATE 文、MERGE 文、DELETE 文
- Transact-SQL と Watcom-SQL のストアドプロシージャ
- ファンクション
- トリガ
- バッチ

現在、ROW、ARRAY、VARRAY、UNNEST は予約語です。

次の項を参照してください。

- 「複合データ型」『SQL Anywhere サーバ SQL リファレンス』
- 「複合型の比較」『SQL Anywhere サーバ SQL リファレンス』
- 「配列演算子」『SQL Anywhere サーバ SQL リファレンス』
- 「ARRAY コンストラクタ [複合]」『SQL Anywhere サーバ SQL リファレンス』
- 「ROW コンストラクタ [複合]」『SQL Anywhere サーバ SQL リファレンス』
- 「ARRAY_AGG 機能 [Aggregate]」『SQL Anywhere サーバ SQL リファレンス』
- 「ARRAY_MAX_CARDINALITY 関数 [複合]」『SQL Anywhere サーバ SQL リファレンス』
- 「CARDINALITY 関数 [複合]」『SQL Anywhere サーバ SQL リファレンス』
- 「TRIM_ARRAY 関数 [複合]」『SQL Anywhere サーバ SQL リファレンス』
- 「UNNEST 配列演算子」『SQL Anywhere サーバ SQL リファレンス』

OData のサポート

SQL Anywhere には、Web クライアントが SQL Anywhere データベースサーバと通信するための OData サーバが組み込まれています。Web クライアントは、HTTP サーバ内にある設定可能な OData プロデューサに要求を送信することができます。OData の概念はここでリレーショナルデータベースの操作に変換されます。OData サポートの一部として、次の機能が追加または変更されています。

- OData プロトコルバージョン 2 への準拠
- OData プロデューサ
- Embedded Jetty WebServer (代替 HTTP サーバのサポート)
- OData サーバユーティリティ
- OData サーバ停止ユーティリティ
- サービスユーティリティ (dbsvc) での OData Windows サービスの作成
- SQL Anywhere プラグインによる OData サービス管理のサポート

「OData のサポート」『SQL Anywhere サーバ プログラミング』を参照してください。

イベントのトレースのサポート

イベントのトレースでは、システム定義およびユーザ定義のトレースイベントに関する情報がイベントトレースターゲットに記録されます。トレースセッションは、ターゲットに記録される情報を収集するトレースイベント (データベースサーバソフトウェアまたは SQL アプリケーション内の特定の位置) で構成されています。ターゲットとは、データベースサーバがトレースイベントを記録する場所 (ファイルなど) です。

イベントのトレースは、運用環境で推奨されており、これを使用すると、ログに記録される情報をきめ細かく制御できます。データベースサーバとアプリケーションの両方についての、ユーザ定義とシステム定義の両方のトレースイベントを記録でき、トレースイベントをカスタマイズして、パフォーマンスの問題を特定することができます。

trace_system_event セキュリティ機能を使用すると、ユーザ定義のトレースイベントの作成を可能にするかどうかを制御できます。

次の項を参照してください。

- 「イベントのトレース」『SQL Anywhere サーバ データベース管理』
- 「sp_trace_events システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
- 「sp_trace_event_fields システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
- 「sp_trace_event_sessions システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
- 「sp_trace_event_session_events システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
- 「sp_trace_event_session_targets システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
- 「sp_trace_event_session_target_options システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
- 「CREATE TEMPORARY TRACE EVENT 文」『SQL Anywhere サーバ SQL リファレンス』
- 「CREATE TEMPORARY TRACE EVENT SESSION 文」『SQL Anywhere サーバ SQL リファレンス』
- 「ALTER TRACE EVENT SESSION 文」『SQL Anywhere サーバ SQL リファレンス』
- 「DROP TRACE EVENT 文」『SQL Anywhere サーバ SQL リファレンス』
- 「DROP TRACE EVENT SESSION 文」『SQL Anywhere サーバ SQL リファレンス』
- 「NOTIFY TRACE EVENT 文」『SQL Anywhere サーバ SQL リファレンス』
- 「イベントトレースデータ (ETD) ファイル管理ユーティリティ (dbmanageetd)」『SQL Anywhere サーバ データベース管理』
- 「セキュリティ機能」『SQL Anywhere サーバ データベース管理』

SAP HANA のサポート

Interactive SQL による SAP HANA データベースへの接続のサポート　Interactive SQL を使用して、SAP HANA データベースに接続できます。**[接続]** ウィンドウで、**[データベースタイプを変更]** をクリックし、次に **[SAP HANA]** をクリックします。「Interactive SQL」『SQL Anywhere サーバ データベース管理』を参照してください。

HANAODBC リモートデータアクセスサーバクラス　SAP HANA をバックエンドサーバとして使用するため、リモートデータアクセスのサポートに HANAODBC リモートデータアクセスクラスが含まれています。「リモートデータアクセスのサーバクラス」『SQL Anywhere サーバ SQL の使用法』を参照してください。

データベースミラーリングと読み込み専用スケールアウトの強化

ミラーまたはコピーノードでのイベントの実行　イベントを、ミラーリングおよび読み込み専用スケールアウトシステム内のミラーサーバとコピーノード上で実行できるようになりました。どのサーバでも実行できるイベントを作成するには、CREATE EVENT 文または ALTER EVENT 文で FOR ALL 句を指定します。「CREATE EVENT 文」『SQL Anywhere サーバ SQL リファレンス』と「ALTER EVENT 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

新しい MIRROR リモートデータアクセスサーバクラス　MIRROR サーバクラスは ODBC を通してリモート SQL Anywhere サーバに接続されます。ただし、リモートサーバの作成時には、SYS.SYSMIRRORSERVER カタログテーブルにあるミラーサーバ名が USING 句に含まれます。「サーバクラス MIRROR」『SQL Anywhere サーバ SQL の使用法』を参照してください。

実行中のミラーリング設定内での監視サーバの移動　「チュートリアル : 監視サーバの移動」『SQL Anywhere サーバ データベース管理』を参照してください。

ミラーサーバをコピーノードに変換する　「チュートリアル：パートナサーバをコピーノードに変換する」『SQL Anywhere サーバ データベース管理』を参照してください。

パートナをあるサーバから別のサーバに移動　「チュートリアル：パートナサーバの移動」『SQL Anywhere サーバ データベース管理』を参照してください。

1 台のコピーノードとし、同じデータベース用の監視ノードとすることも可能　「チュートリアル：1 台のサーバをコピーノードと監視サーバとして兼用する方法」『SQL Anywhere サーバ データベース管理』を参照してください。

動作中のサーバ上でミラーリングまたは読み込み専用スケールアウトデータベースを動的に起動
動作中のパートナ上でミラーデータベースまたはコピーノードを起動するには、START DATABASE 文と MIRROR ON 句を一緒に使用します。「START DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

SET MIRROR OPTION 文　SET MIRROR OPTION には promotion_time という新しいオプションが用意されています。このオプションを使用すると、親の接続が切れた後、プロンプトを表示しないでコピーノードをルートデータベースサーバに接続された状態にする時間を指定できます。max_disconnected_time オプションでは、前回コピーノードが親、代替の親、またはルートデータベースに接続されてからコピーノードが停止するまでの時間を指定できるようになりました。「SET MIRROR OPTION 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

データベースプロパティの強化

CopyNodeParent　読み込み専用スケールアウト設定に含まれるコピーノードの現在の親サーバ名を返します。CopyNodeParent データベースプロパティ『SQL Anywhere サーバ データベース管理』を参照してください。

PartnerState　パートナが定義されていない場合、PartnerState は NULL を返します。以前は disconnected を返していました。PartnerState データベースプロパティ『SQL Anywhere サーバ データベース管理』を参照してください。

LOAD TABLE の新しいデフォルトの WITH ROW LOGGING 句　WITH ROW LOGGING 句を指定すると、各ローのロードが INSERT 文としてトランザクションログに記録されます。ミラーリングされたデータベースで FROM filename-expression または USING FILE filename-expression を使用している場合、同期を行っているデータベースやデフォルトになっているデータベースには、このレベルのロギングをおすすめします。ただし、大量のデータをロードする場合は、このロギングタイプがパフォーマンスに影響したり、トランザクションログがかなり長くなったりする可能性があります。

管理の強化

- **アップグレード後のデータベースの再起動**　デフォルトで、データベースが停止して、アップグレード後に再起動されます。アップグレード後にデータベースを再起動するのではなく、停止するには、ALTER DATABASE 文の RESTART OFF 句を使用するか、アップグレードユーティリティ (dbupgrad) の -nrs オプションを使用します。アップグレードが完了すると、データベースのアップグレードにより、トランザクションログの名前が変更されます。「ALTER DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』と「アップグレードユーティリティ (dbupgrad)」『SQL Anywhere サーバ データベース管理』を参照してください。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

- **テーブルの所有者の変更**　ALTER TABLE 文を使用してテーブルの所有者を変更できるようになりました。テーブルの所有者の変更時に、明示的に付与された権限を保持または削除したり、外部キーを保持または削除したりすることができます。「ALTER TABLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **二重制御パスワード**　change_password_dual_control ログインポリシーオプションでは、ターゲットユーザのパスワードを変更するのに 2 人の管理者の関与が必要です。1 人目がパスワードの前半部分を指定し、2 人目が後半部分を指定します。二重制御パスワード変更を使用すると、パスワードの管理者に別のユーザのパスワードを知られないようにすることができます。

  ターゲットユーザは、そのログインポリシーで change_password_dual_control オプションが有効になっている必要があります。

  ログインポリシーで change_password_dual_control オプションが有効になったユーザに対しては、verify_password_function データベースオプションで指定される機能は呼び出されません。

  - 「二重制御パスワード」『SQL Anywhere サーバ データベース管理』
  - 「verify_password_function オプション」『SQL Anywhere サーバ データベース管理』
  - 「sa_verify_password システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』

データベース接続

次に、SQL Anywhere バージョン 16.0 でのデータベース接続の強化を示します。

- **MaxRequestVars (MAXVARS) プロトコルオプション**　このプロトコルオプションは、データベースサーバに送信される要求内で許可する HTTP 入力変数の最大数を指定します。「MaxRequestVars (MAXVARS) プロトコルオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **SuppressInfoForDataTypes ODBC 接続パラメータ**　この接続パラメータを使用すると、ODBC ドライバによって指定のデータ型に関する情報が返されません。「ODBC 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

バックアップとリカバリ

次に、SQL Anywhere バージョン 16.0 でのバックアップとリカバリの強化を示します。

- **バックアップユーティリティ (dbbackup) の強化**　バックアップユーティリティに次のオプションが追加されています。これらのオプションは必ず -s オプションと一緒に使用してください。「バックアップユーティリティ (dbbackup)」『SQL Anywhere サーバ データベース管理』を参照してください。

オプション	説明
-aw[-]	ライタの自動チューニングを有効にします。-aw- を指定すると、ライタの自動チューニングが無効になります。
-bc *コメント*	バックアップ履歴ファイルにコメントを記録します。
-h[-]	バックアップ履歴を有効にします。-h- を指定すると、バックアップ履歴が無効になります。
-wa	トランザクションが完了してトランザクションログの名前変更またはトランケートが行われるまで待機します。
-wb	アクティブなトランザクションがなくなるまで、データベースのバックアップを遅らせます。

セキュリティ

次に、SQL Anywhere バージョン 16.0 でのセキュリティの強化を示します。

- **未加工の暗号化**　ENCRYPT 関数では、データベースの外部で復号化できるように、データベースサーバ内のデータを暗号化し、未加工フォーマットに出力できます。DECRYPT 関数では、データベースサーバの外部で暗号化されたデータを複合化できます。 「ENCRYPT 関数 [文字列]」『SQL Anywhere サーバ SQL リファレンス』と「DECRYPT 関数 [文字列]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **臨時のパスワード変更による TDS RSA 暗号化のサポート**　新しい -tdsl データベースサーバオプションを使用して、データベースサーバがサポートする TDS ログイン要求のタイプを制限できます。TDS ログインモードで設定できるのは、すべてのログイン要求、RSA ログイン要求のみ、臨時ログイン要求による RSA のみのいずれかです。 「-tdsl データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **セキュリティ機能**

  - **セキュリティ機能キーのカスタマイズが可能**　次のシステムプロシージャを使用して、このユーザに割り当て可能なカスタマイズセキュリティ機能キーを作成し、表示することができるようになりました。

    - **sp_create_secure_feature_key システムプロシージャ**　新しいセキュリティ機能キーを作成します。「sp_create_secure_feature_key システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

- **sp_alter_secure_feature_key システムプロシージャ**　認証キー、機能リスト、またはその両方を変更して、以前に定義されたセキュリティ機能キーを変更します。「sp_alter_secure_feature_key システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sp_drop_secure_feature_key システムプロシージャ**　セキュリティ機能キーを削除します。「sp_drop_secure_feature_key システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sp_list_secure_feature_key システムプロシージャ**　定義されたセキュリティ機能キーのリストを返します。「sp_list_secure_feature_key システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sp_use_secure_feature_key システムプロシージャ**　指定したセキュリティ機能キーに関連するセキュリティ機能へのアクセスを許可します。「sp_use_secure_feature_key システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

○ **セキュリティ機能**　サーバと同じコンピュータ上のディレクトリとファイルをユーザが操作できないよう保護します。システムプロシージャに対応する次の機能が **local_io** 機能セットに追加されました。

- sp_list_directory
- sp_create_directory
- sp_copy_directory
- sp_move_directory
- sp_delete_directory
- sp_copy_file
- sp_move_file
- sp_delete_file

manage_server 機能セットを使用すると、データベースサーバに関連する機能にユーザがアクセスできないように設定できます。

create_trace_file 機能と create_trace_system_event 機能では、それぞれ、ユーザがイベントトレースターゲットとユーザ定義イベントを作成できないように設定できます。

○ **データベースサーバのセキュリティ機能キーの指定後に SQL を再実行**　Sybase Central でセキュリティ機能を使用する SQL を実行しようとする場合、データベースサーバのセキュリティ機能キーを指定してから SQL を再実行できます。この機能を使用するには、セキュリティ機能キーと一緒にデータベースサーバを起動 (-sk オプションを使用) する必要があります。

- **ディスクサンドボックス**　ディスクサンドボックス機能は、データベースからデータベースファイルのあるディレクトリへのファイルの読み込み／書き込み操作を制限します。

ディスクサンドボックス機能が有効な場合、相対パス名は、メインデータベースファイルがあるディレクトリに対する相対パスとみなされます。ディスクサンドボックス機能が有効でない場合、相対パス名は、データベースサーバの作業ディレクトリに対する相対パスとみなされます。「SQL Anywhere の動作の変更」29 ページを参照してください。

ディスクサンドボックス機能をサポートするため、次の機能が追加されています。

機能	説明
データベースサーバオプション (dbsrv16)	○ **-sbx データベースサーバおよびデータベースオプション**　このオプションは、データベースサーバ上で動作するすべてのデータベース、または単体のデータベースに対するディスクサンドボックスのデフォルト動作を設定します。「-sbx データベースサーバオプション」『SQL Anywhere サーバ データベース管理』と「-sbx データベースオプション」『SQL Anywhere サーバ データベース管理』を参照してください。
データベースオプション	○ **disk_sandbox database option**　このデータベースオプションは、データベースでのファイルの読み込み／書き込み操作を、メインデータベースファイルのあるディレクトリに制限するかどうかを制御します。「disk_sandbox オプション」『SQL Anywhere サーバ データベース管理』を参照してください。
文	○ **START DATABASE 文**　データベースでのファイルの読み込み／書き込み操作をメインデータベースファイルのあるディレクトリに制限するには、START DATABASE 文の実行時に DISKSANDBOX 句を指定します。「START DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

機能	説明
セキュリティ機能	○ **manage_disk_sandbox セキュリティ機能**　manage_disk_sandbox セキュリティ機能を制御するには、-sf データベースサーバオプションを使用します。manage_disk_sandbox セキュリティ機能を使用すると、ユーザによってディスクサンドボックス設定が変更されるのを防止できます。デフォルトで、manage_disk_sandbox セキュリティ機能は保護されます。 「-sf データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。 ○ **disk_sandbox セキュリティ機能**　disk_sandbox セキュリティ機能を制御するには、-sf データベースサーバオプションを使用します。disk_sandbox セキュリティ機能を使用すると、接続されたユーザが、その接続に対してディスクサンドボックスを有効または無効にできます。デフォルトで、disk_sandbox セキュリティ機能は保護されます。 「ディスクサンドボックス」『SQL Anywhere サーバ データベース管理』を参照してください。

機能	説明
システムプロシージャ	ディスクサンドボックス設定の変更を sa_server_option システムプロシージャまたは sa_db_option システムプロシージャで行うには、manage_disk_sandbox セキュリティ機能のためのセキュリティ機能キーを用意します。 ○ **sa_server_option システムプロシージャ**　データベースサーバの動作中に、データベースサーバのディスクサンドボックスのデフォルト動作を変更します。「sa_server_option システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。 ○ **sa_db_option システムプロシージャ**　データベースの動作中に、データベースのディスクサンドボックスの動作を変更します。「sa_db_option システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
プロパティ	○ **disk_sandbox 接続プロパティ**　ユーザ接続に対するディスクサンドボックス機能の設定が ON か OFF かを返します。disk_sandbox 接続プロパティ『SQL Anywhere サーバ データベース管理』を参照してください。 ○ **DiskSandbox データベースサーバプロパティ**　データベースサーバに対するディスクサンドボックス機能の設定が ON か OFF かを返します。DiskSandbox サーバプロパティ『SQL Anywhere サーバ データベース管理』を参照してください。 ○ **DiskSandbox データベースプロパティ**　データベースに対するディスクサンドボックス機能の設定が ON か OFF かを返します。DiskSandbox データベースプロパティ『SQL Anywhere サーバ データベース管理』を参照してください。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

データベースユーティリティ

次に、SQL Anywhere バージョン 16.0 でのデータベースユーティリティの強化を示します。

- **サーババックグラウンド起動ユーティリティ (dbspawn)**　dbspawn を使用して名前がユニークでないデータベースサーバを生成すると、新しいエラーと終了コード (EXIT_SERVER_NAME_IN_USE) が返されます。「ソフトウェアコンポーネントの終了コード」『SQL Anywhere サーバ プログラミング』を参照してください。

- **アンロードユーティリティ (dbunload)**　-ss オプションを使用すると、リロード SQL スクリプトファイルのカラム統計の生成が抑制されます。「アンロードユーティリティ (dbunload)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **証明書作成ユーティリティ (createcert) と証明書ビューアユーティリティ (viewcert)**

  ○createcert と viewcert で -p1 オプションを使用すると、暗号化されていない RSA プライベートキーのコード化に PKCS #1 を使用するよう指定できます。

  ○createcert で -sa オプションを使用すると、証明書の作成時または署名時に使用するシグネチャアルゴリズムを指定できます。

  ○viewcert で *@data* オプションを使用すると、設定ファイルまたは環境変数に格納されたコマンドラインの指定で viewcert を実行できます。

  「証明書作成ユーティリティ (createcert)」『SQL Anywhere サーバ データベース管理』と「証明書ビューアユーティリティ (viewcert)」『SQL Anywhere サーバ データベース管理』を参照してください。

データベースオプション

次に、SQL Anywhere バージョン 16.0 でのデータベースオプションの強化を示します。

- **auto_commit_on_create_local_temp_index オプション**　このオプションは、ローカルテンポラリテーブルにインデックスを作成する前にデータベースサーバが COMMIT を実行するかどうかを制御します。この機能を使用するには、既存のデータベースをアップグレードする必要があります。「auto_commit_on_create_local_temp_index オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **extern_login_credentials オプション**　リモート接続の試行で、ログインしたユーザの外部ログインクレデンシャルを使用するか、効果的なユーザの外部ログインクレデンシャルを使用するかを制御します。「extern_login_credentials オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **webservice_sessionid_name オプション**　このオプションは、セッション管理を使用するかどうかの判断のために SQL Anywhere Web サーバが使用する項目を再定義します。「webservice_sessionid_name オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

データベースサーバオプション

次に、SQL Anywhere バージョン 16.0 でのデータベースサーバオプションの強化を示します。

- **-al データベースサーバオプション**　指定したユーザに対して標準のユーザ認証を許可します。「-al データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **-al データベースオプション**　指定したデータベースの指定したユーザに対して標準のユーザ認証を許可します。「-al データベースオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **-gta データベースサーバオプション**　データベースサーバが使用できる論理プロセッサを設定します。「-gta データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **-uf データベースサーバオプション**　-uf データベースサーバオプションはすべてのオペレーティングシステムに適用されるようになりました。「-uf データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **-ufd データベースサーバオプション**　致命的なエラーまたはアサーションの失敗がデータベースで発生したときにデータベースサーバが実行するアクションを指定します。「-ufd データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **-xs データベースサーバオプション**　データベースサーバで複数のポートやプロトコルでの接続の受信を許可するために、-xs データベースサーバオプションを複数回指定できるようになりました。以前は -xs データベースサーバオプションは 1 回しか指定できなかったため、単一の -xs オプションのパラメータ内ですべてのポートとプロトコルを指定する必要がありました。「-xs データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

プロパティとパフォーマンスモニタの統計値

次に、SQL Anywhere バージョン 16.0 でのプロパティとパフォーマンスモニタの統計の強化を示します。

- **新しい接続プロパティ**　このリリースには、次の接続プロパティが追加されています。
  - auto_commit_on_create_local_temp_index 接続プロパティ『SQL Anywhere サーバ データベース管理』
  - extern_login_credentials 接続プロパティ『SQL Anywhere サーバ データベース管理』
  - NumLocalTempTables 接続プロパティ『SQL Anywhere サーバ データベース管理』
  - trusted_certificates_file 接続プロパティ『SQL Anywhere サーバ データベース管理』
  - webservice_sessionid_name 接続プロパティ『SQL Anywhere サーバ データベース管理』

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

- **新しいデータベースプロパティ**　このリリースには、次のデータベースプロパティが追加されています。

  - LastCommitRedoPos
  - LastSyncedRedoPos
  - LastWrittenRedoPos
  - UTCTimestampCatalog

- **新しいデータベースサーバプロパティ**　このリリースには、次のデータベースサーバプロパティが追加されています。

  - IsAesniAvailable
  - ProcessorAffinity

システムプロシージャとファンクション

次に、SQL Anywhere バージョン 16.0 で追加されたシステムプロシージャとファンクションの強化を示します。

- **sa_certificate_info システムプロシージャ**　このシステムプロシージャは、データベースに格納された証明書に関する情報を返します。「sa_certificate_info システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_cpu_topology システムプロシージャ**　このシステムプロシージャは、データベースサーバを実行しているコンピュータのプロセッサのトポロジに関する情報を返します。「sa_cpu_topology システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_db_option システムプロシージャ**　このシステムプロシージャを使用すると、データベースの実行中にデータベースオプションを上書きできます。「sa_db_option システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_parse_json システムプロシージャ**　このシステムプロシージャは、SQL の ROW 型と ARRAY 型を使用して JSON データの表現を返します。「sa_parse_json システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_server_option システムプロシージャ**　Windows と Linux では、sa_server_option システムプロシージャの ProcessorAffinity プロパティを使用して、データベースサーバの起動後にデータベースサーバで使用可能な論理プロセッサの数を変更できます。「sa_server_option システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ディレクトリとファイルのストアドプロシージャ**　新しいストアドプロシージャのセットを使用すると、ディレクトリアクセスサーバ、外部ログイン、ディレクトリアクセスのプロキシテーブルを設定しなくても、サーバからアクセス可能なディレクトリとファイルをリスト表示し、操作することができます。これらのシステムプロシージャはセキュリティ機能として有効にしたり無効にしたりできます。

- **sp_copy_directory システムプロシージャ**　このシステムプロシージャはディレクトリをコピーします。「sp_copy_directory システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
- **sp_copy_file システムプロシージャ**　このシステムプロシージャはファイルをコピーします。「sp_copy_file システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
- **sp_create_directory システムプロシージャ**　このシステムプロシージャはディレクトリを作成します。「sp_create_directory システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
- **sp_delete_directory システムプロシージャ**　このシステムプロシージャはディレクトリを削除します。「sp_delete_directory システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
- **sp_delete_file システムプロシージャ**　このシステムプロシージャはファイルを削除します。「sp_delete_file システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
- **sp_move_directory システムプロシージャ**　このシステムプロシージャはディレクトリを移動します。「sp_move_directory システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
- **sp_move_file システムプロシージャ**　このシステムプロシージャはファイルを移動します。「sp_move_file システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **xp_get_mail_error_code システムプロシージャ**　このシステムプロシージャは、xp_ mail プロシージャの使用時に発生した最新の SMTP または MAPI エラーに関する情報を返します。「xp_get_mail_error_code システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
- **xp_get_mail_error_text システムプロシージャ**　このシステムプロシージャは、xp_ mail プロシージャの使用時に発生した最新の SMTP エラーに関する情報を返します。「xp_get_mail_error_text システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
- **xp_getenv システムプロシージャ**　このシステムプロシージャは、環境変数の値を返します。「xp_getenv システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
- **xp_startsmtp システムプロシージャ**　このシステムプロシージャの trusted_certificates パラメータに、**file=***file-path* オプションと **cert_name=***certificate-name* オプションを指定できるようになりました。「xp_startsmtp システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
- **ファンクションとプロシージャのパラメータ表現での subselect とサブクエリのサポート**　ファンクションとプロシージャのパラメータ表現で subselect とサブクエリがサポートされます。

- **BINTOHEX 関数**　この関数は、バイナリ文字列を 16 進数で返します。「BINTOHEX 関数 [データ型変換]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
- **HEXTOBIN 関数**　この関数は、16 進文字列を LONG BINARY で返します。「HEXTOBIN 関数 [データ型変換]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

SQL 文

次に、SQL Anywhere バージョン 16.0 で導入された SQL 文の新機能と強化を示します。

ALTER TABLE 文　テーブルの所有者を変更できるようになりました。テーブルの所有者の変更時に、所有者のパーミッションを保持または削除したり、外部キーを保持または削除したりすることができます。「ALTER TABLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

CREATE CERTIFICATE 文と DROP CERTIFICATE 文　証明書をデータベースに格納できるようになりました。CREATE CERTIFICATE 文を使用すると、データベースに証明書を追加したり、データベースの証明書を置き換えたりできます。DROP CERTIFICATE 文を使用すると、データベースから証明書を削除できます。「CREATE CERTIFICATE 文」『SQL Anywhere サーバ SQL リファレンス』と「DROP CERTIFICATE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

CREATE EVENT 文と ALTER EVENT 文　イベントを、ミラーリングおよび読み込み専用スケールアウトシステム内のミラーサーバとコピーノード上で実行できるようになりました。どのサーバでも実行できるイベントを作成するには、CREATE EVENT 文または ALTER EVENT 文で FOR ALL 句を指定します。「CREATE EVENT 文」『SQL Anywhere サーバ SQL リファレンス』と「ALTER EVENT 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

CREATE FUNCTION 文 [Web サービス]　これで、CERTIFICATE 句の **certificate_name** オプションを使用して、データベースに保管されている証明書を指定できます。これで、キープアライブタイムアウト条件オプション **(kto)** を指定して、キープアライブ HTTP/HTTPS 接続をインスタンス化して、一定の時間の間、キャッシュすることができます。「CREATE FUNCTION 文 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

CREATE INDEX 文　文の実行時に WITH NULLS DISTINCT を指定できるようになりました。「CREATE INDEX 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

CREATE PROCEDURE 文 [Web サービス]　これで、CERTIFICATE 句の **certificate_name** オプションを使用して、データベースに保管されている証明書を指定できます。これで、キープアライブタイムアウト条件オプション **(kto)** を指定して、キープアライブ HTTP/HTTPS 接続をインスタンス化して、一定の時間の間、キャッシュすることができます。「CREATE PROCEDURE 文 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

CREATE SERVER 文　USING 句の DELIMITER オプションを使用して、ディレクトリアクセスサーバによって返されるファイル名のパスの区切り文字としてスラッシュ (/) または円記号 (¥) が使用されるようにできます。「CREATE SERVER 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

CREATE TEXT INDEX 文　マテリアライズドビューでテキストインデックスを作成できるようになりました。マテリアライズドビューでテキストインデックスを作成するには、IMMEDIATE

REFRESH 句を使用します。「CREATE TEXT INDEX 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

DROP REMOTE CONNECTION 文 DROP REMOTE CONNECTION 文はリモートサーバへのリモートデータアクセス接続を削除します。「DROP REMOTE CONNECTION 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

CREATE TEMPORARY TRACE EVENT SESSION 文 ユーザトレースイベントセッションを作成します。「CREATE TEMPORARY TRACE EVENT SESSION 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

CREATE TEMPORARY TRACE EVENT 文 データベースが停止するまで持続するユーザトレースイベントを作成します。「CREATE TEMPORARY TRACE EVENT 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

ALTER TRACE EVENT SESSION 文 トレースイベントのセッションへの追加およびセッションからの削除、ターゲットのセッションへの追加およびセッションからの削除、またはトレースセッションの開始および停止を実行します。「ALTER TRACE EVENT SESSION 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

DROP TRACE EVENT 文 ユーザ定義トレースイベントを削除します。「DROP TRACE EVENT 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

DROP TRACE EVENT SESSION 文 トレースイベントセッションを削除します。「DROP TRACE EVENT SESSION 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

LOAD TABLE 文 *load-option* 句では、以下を指定できるようになりました。

- **ALLOW ERRORS 句** ロード操作をロールバックするまでに許可するエラーの回数を指定できます。デフォルトは 0 です。
- **ROW LOG 句** ローの挿入または解析中にエラーが発生すると、データベースサーバはローをユーザにレポートするだけでなく、指定された場所に入力ローのイメージを書き込みます。
- **MESSAGE LOG 句** ローの挿入または解析中にエラーが発生すると、データベースサーバは、指定された場所にエラーを書き込みます。

NOTIFY TRACE EVENT 文 ユーザ定義トレースイベントをトレースセッションに記録します。「NOTIFY TRACE EVENT 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

TRY...CATCH 文 複合文の内部で発生するエラーに関する情報を取得するには、エラーハンドラとして (BEGIN CATCH...END CATCH と一緒に) BEGIN TRY...END TRY を使用できます。「TRY 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

エラーに関す情報を取得するため、次のファンクションとシステムプロシージャが追加されています。プロシージャとファンクションは、エラーハンドラの内部だけでなく、文内部の任意の場所で使用できます。

○「ERROR_LINE 関数 [その他]」『SQL Anywhere サーバ SQL リファレンス』
○「ERROR_MESSAGE 関数 [その他]」『SQL Anywhere サーバ SQL リファレンス』
○「ERROR_PROCEDURE 関数 [関数タイプ]」『SQL Anywhere サーバ SQL リファレンス』
○「ERROR_SQLCODE 関数 [その他]」『SQL Anywhere サーバ SQL リファレンス』
○「ERROR_SQLSTATE 関数 [その他]」『SQL Anywhere サーバ SQL リファレンス』
○「ERROR_STACK_TRACE 関数 [その他]」『SQL Anywhere サーバ SQL リファレンス』
○「sa_error_stack_trace システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sa_stack_trace システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「STACK_TRACE 関数 [その他]」『SQL Anywhere サーバ SQL リファレンス』

ENCRYPTED KEY 句または KEY 句を含む文で変数名を使用可能　ENCRYPTED KEY 句または KEY 句には文字列または変数名を指定できます。この変更は、次の文に影響を及ぼします。

○「ALTER DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』
○「CREATE DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』
○「CREATE DECRYPTED DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』
○「CREATE DECRYPTED FILE 文」『SQL Anywhere サーバ SQL リファレンス』
○「CREATE ENCRYPTED DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』
○「CREATE ENCRYPTED FILE 文」『SQL Anywhere サーバ SQL リファレンス』
○「DROP DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』
○「LOAD TABLE 文」『SQL Anywhere サーバ SQL リファレンス』
○「RESTORE DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』
○「START DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』
○「UNLOAD 文」『SQL Anywhere サーバ SQL リファレンス』
○ FROM 句の Openstring 式『SQL Anywhere サーバ SQL リファレンス』

データ型

次に、SQL Anywhere バージョン 16.0 でのデータ型の強化を示します。

● **後続 UNSIGNED キーワードのサポートの追加**　データベースサーバは、BIGINT UNSIGNED 型、INTEGER UNSIGNED 型、SMALLINT UNSIGNED 型、TINYINT UNSIGNED 型の宣言をサポートするようになりました。「数値データ型」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

プログラミングインタフェース

次に、SQL Anywhere バージョン 16.0 でのプログラミングインタフェースの強化を示します。

● **Visual Studio 2012 のサポートの追加**　SQL Anywhere .NET データプロバイダで Microsoft Visual Studio 2012 がサポートされるようになりました。「SQL Anywhere の .NET サポート」『SQL Anywhere サーバ プログラミング』を参照してください。

- **プロバイダでの .NET Compact Framework 3.5 のサポートの追加**　SQL Anywhere .NET データプロバイダで Compact Framework 3.5 がサポートされるようになりました。「SQL Anywhere .NET データプロバイダの機能」『SQL Anywhere サーバ プログラミング』を参照してください。

- **プロバイダでの .NET Framework 4.5 のサポートの追加**　SQL Anywhere .NET 4.0 データプロバイダで .NET Framework 4.5 がサポートされるようになりました。「SQL Anywhere .NET データプロバイダの機能」『SQL Anywhere サーバ プログラミング』を参照してください。

- **プロバイダでの Entity Framework 5.0 のサポートの追加**　SQL Anywhere .NET データプロバイダで Microsoft Entity Framework の最新リリースである Entity Framework 5.0 がサポートされるようになりました。「Entity Framework のサポート」『SQL Anywhere サーバ プログラミング』を参照してください。

- **OLE DB プロバイダによる DBTYPE_DBTIME2 データ型のサポート**　SQL Anywhere OLE DB プロバイダが DBTYPE_DBTIME2 データ型をサポートするようになりました。DBTYPE_DBTIME2 (145) とは、小数点以下の秒に対応した TIME データ型をサポートする OLE DB 型です (DBTYPE_DBTIME は小数点以下の秒をサポートしていません)。このデータ型をサポートすることにより、精度を落とすことなく、SQL Anywhere データベースと他のデータベース管理システム (SQL Anywhere を含む) 間での TIME 型のカラムの転送が容易に行えます。

- **更新された PHP ドライバのサポート**　5.3.18 までのバージョン、および 5.4.8 の PHP のサポートが追加されています。

カタログの変更

次に、SQL Anywhere バージョン 16.0 でのカタログの変更を示します。

- **新しい ISYSCERTIFICATE システムテーブルと SYSCERTIFICATE システムビュー**　ISYSCERTIFICATE システムテーブルの各ローには、テキスト PEM フォーマットの証明書が保管されています。SYSCERTIFICATE システムビューはこのシステムテーブル上のビューです。「ISYSCERTIFICATE システムテーブル」『SQL Anywhere サーバ SQL リファレンス』と「SYSCERTIFICATE システムビュー」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ISYSTABCOL と ISYSTAB 内の新しいカラム**　カラム nonmaterialized_value と start_schema が ISYSTABCOL に追加されています。カラム current_schema が ISYSTAB に追加されています。「SYSTAB システムビュー」『SQL Anywhere サーバ SQL リファレンス』と「SYSTABCOL システムビュー」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ISYSUSER 内の新しいカラム**　カラム user_type、user_dn、user_dn_cached_at、password_creation_time_utc、last_login_time_utc、dual_password が ISYSUSER に追加されています。「SYSUSER システムビュー」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **カタログでの UTC タイムスタンプの使用**　以前のリリースでは、カタログテーブルはデータベースサーバの場所に応じたローカルタイムスタンプを格納し、公開していました。現在、

タイムスタンプはそれに対応する UTC (協定世界時) ベースのタイムスタンプとして格納されるようになりましたが、ビューは UTC タイムスタンプとローカルタイムスタンプの両方を公開します。以下のシステムテーブルとビューのカラム数に依存するアプリケーションは更新する必要があります。

システムテーブル名	システムビュー名	追加されたカラム
ISYSTAB	SYSTAB	last_modified_at_utc
ISYSVIEW	SYSVIEW	mv_last_refreshed_at_utc, mv_known_stale_at_utc
ISYSOBJECT	SYSOBJECT	creation_time _utc
ISYSUSER	SYSUSER	password_creation_time, last_login_time
ISYSTEXTIDX	SYSTEXTIDX	last_refresh_utc
ISYSEXTERNENVOBJECT	SYSEXTERNENVOBJECT	update_time_utc
ISYSJAR	SYSJAR	update_time_utc
ISYSJAVACLASS	SYSJAVACLASS	update_time_utc
ISYSHISTORY	SYSHISTORY	first_time_utc, last_time_utc
ISYSREMOTEUSER	SYSREMOTEUSER	time_sent_utc, time_received_utc
	SYSREMOTEUSERS 統合ビュー	time_sent_utc, time_received_utc
ISYSCOLSTAT	SYSCOLSTAT	update_time_utc
	SYSCOLSTATS 統合ビュー	update_time_utc

UNIX/Linux の強化

次に、SQL Anywhere バージョン 16.0 での UNIX と Linux に関する強化を示します。

- **SQL Anywhere のフランス語版リソース (Linux 用)**　Linux で SQL Anywhere のフランス語版のリソースが利用できるようになりました。ローカライズは、パッケージ、インストーラ、マニュアル、ソフトウェアのユーザインタフェース、エラー／警告／情報メッセージなど、多くのコンポーネントに反映されます。「SQL Anywhere のローカライズ版」『SQL Anywhere サーバ データベース管理』を参照してください。

パフォーマンスの強化

次に、パフォーマンスの向上以外にユーザに見える変更がないバージョン 16.0 でのパフォーマンスの強化を示します。

- **AES 暗号化**　AES を使用してクライアント／サーバ通信を暗号化し、Windows および Linux 用の Intel/AMD プロセッサでデータベースファイルを暗号化する場合のパフォーマンスが向上しています。IsAesniAvailable データベースサーバのプロパティを問い合わせることによって、強化された AES アルゴリズムが使用されているかどうかをチェックできます。IsAesniAvailable サーバプロパティ『SQL Anywhere サーバ データベース管理』を参照してください。

- **テーブルへのカラムの追加**　ALTER TABLE...ADD COLUMN 文の使用時のパフォーマンスが向上しています。デフォルト値を評価すると定数になる大部分の状況で、ALTER TABLE 文の実行速度が以前のリリースよりも大幅に高速化しています。この機能を使用できるように、既存のバージョン 11 以降の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 10 以前のデータベースの場合、この機能を使用するには再構築してください。

その他

次に、SQL Anywhere バージョン 16.0 でのその他の強化を示します。

- **CONTAINS 探索条件の新しい表現と強化された表現**　CONTAINS 探索条件に新しい *before-expression* が追加されています。用語の相対敵名順序を重視する近接検索を実行するには、*before-expression* を使用します。「CONTAINS 探索条件」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  また、*near-expression* は検索文字列間の最小距離条件を受け入れるよう強化されています。「CONTAINS 探索条件」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **暗号化に関するクライアントメッセージ**　dbinit ユーティリティまたは CREATE DATABASE 文を使用してデータベースが作成された場合、データベースの暗号化のタイプを示すメッセージがクライアントに送信されるようになりました。暗号化を使用している場合は、使用されているアルゴリズムも表示されます。

- **グラフィカルなプランと長いプラン**　クエリ最適化プロセス中に収集される最適化統計が組み込まれるよう、グラフィカルなプランと長いプランが強化されています。「グラフィカルなプラン」『SQL Anywhere サーバ SQL の使用法』と「長いテキストプラン」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **CESU-8 のサポート**　SQL Anywhere で CESU-8 文字セット (エイリアスは ibm-9400) と CESU8BIN 照合がサポートされるようになりました。サポートされる文字セットの詳細リストを取得するには、dblang ユーティリティに -le+ オプションを指定します。サポートされている照合の詳細については、「代替照合」『SQL Anywhere サーバ データベース管理』を参照してください。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

- **予約語**　次に、SQL Anywhere バージョン 16.0 でデータベースに追加された予約語を示します。「予約語」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - array
  - json
  - row
  - rowtype
  - unnest
  - varray

- **識別子とエイリアス**　識別子とエイリアスに次の文字が使用できなくなりました。

  - 角カッコ
  - 逆引用符

> **注意**
> 16.0 より以前のバージョンのデータベースを再ロードしている場合は、識別子に含まれる角カッコおよび逆引用符をすべて削除してください。そうしないと、再ロードが失敗します。

識別子の文字制限の詳細については、「識別子」『SQL Anywhere サーバ SQL リファレンス』を参照してください。エイリアスの文字制限の詳細については、「SELECT 文」『SQL Anywhere サーバ SQL リファレンス』 の *select-list* 句を参照してください。

SQL Anywhere の動作の変更

次に、バージョン 16.0 で導入された SQL Anywhere の動作の変更を示します。サポートされているプラットフォームおよびバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

- **一部のシステムプロシージャのデフォルトセキュリティモデルの変更**　16.0 より以前のバージョンのシステムプロシージャの一部は、パーミッションが必要なデータベースでの操作を実行します。16.0 より以前のバージョンのセキュリティモデルではこれらのプロシージャはデファイナ (所有者) のパーミッションで実行しており、ユーザはプロシージャが実行するすべての操作のためのパーミッションではなく、プロシージャ自体を実行するためのパーミッションのみが必要でした。一部の例外として、DBA 権限も必要な場合がありました。

  バージョン 16.0 以降の時点で、これらのプロシージャのデフォルトセキュリティモデルが変更されます。デフォルトで、新しく作成したデータベースでは、これらのプロシージャはインボーカの権限によって実行されます。そのため、プロシージャを実行するには、インボーカがプロシージャのマニュアルに記載された権限を有している必要があります。インボーカには、プロシージャに対する EXECUTE 権限も必要ですが、これは、PUBLIC のメンバーであることにより継承されます。

  データベースの作成時に、CREATE DATABASE 文の新しい SYSTEM PROCEDURE AS DEFINER 句または初期化ユーティリティ (dbinit) の -pd オプションを使用して、古いモデルを使用するか新しいモデルを使用するかを制御できます。ただし、一部のプロシージャでは、セキュリティモデルの設定に関係なく、常にマニュアルに記載された権限が必要となります。これらのプロシージャのリストについては、「16.0 より前のバージョンのシステムプロシー

ジャを呼び出し側または定義者として実行する」『SQL Anywhere サーバ SQL の使用法』を参照してください。

データベースのアップグレード時のデフォルト動作では、設定済みのセキュリティモデルが保持されます。そのため、たとえば 12.0.1 データベースをアップグレードする場合、特に指定しなければ、アップグレードされたデータベースは古いセキュリティモデルを使用します。

使用するセキュリティモデルをアップグレード時に制御するには、ALTER DATABASE 文の SYSTEM PROCEDURE AS DEFINER 句またはアップグレードユーティリティ (dbupgrad) の -pd オプションを使用します。EXECUTE と特定の権限の両方が必ず必要となるシステムプロシージャの設定に関する同じ例外が、作成についてのアップグレードに適用されます。

注意
セキュリティモデルの決定 (インボーカとデファイナ) はユーザ定義のプロシージャのデフォルト動作に影響しません。このデフォルト動作はデファイナのままです。システムプロシージャのデフォルトをインボーカに変更しても、ユーザ定義のプロシージャのデフォルトはデファイナのままです。

- 「権限が必要な操作を実行するシステムプロシージャの変更」『SQL Anywhere サーバ データベース管理』
- 「16.0 より前のバージョンのシステムプロシージャを呼び出し側または定義者として実行する」『SQL Anywhere サーバ SQL の使用法』
- 「呼び出し側／定義者設定の影響を受けるプロシージャのリスト」『SQL Anywhere サーバ SQL の使用法』
- 「ALTER DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』
- 「CREATE DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』
- 「初期化ユーティリティ (dbinit)」『SQL Anywhere サーバ データベース管理』
- 「アップグレードユーティリティ (dbupgrad)」『SQL Anywhere サーバ データベース管理』

- **相対パスとディスクサンドボックス**　以前のリリースでは、相対パスのデフォルトは常にデータベースサーバの作業ディレクトリでした。現在、データベースに対してサンドボックスを有効にすると、パスはデータベースサーバのディレクトリではなく、データベースのディレクトリへの相対パスとなります。「ディスクサンドボックス」『SQL Anywhere サーバ データベース管理』を参照してください。

- **ビュー定義でサポートされている SELECT *** 　これまでのリリースでは、SELECT * は、CREATE VIEW 文のメインクエリでのみサポートされていました。現在は、メインクエリ、サブクエリ、派生テーブル、または、CREATE VIEW 文の subselect でサポートされています。「CREATE VIEW 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **パスワードの最小長の変更**　以前のリリースでは、パスワードのデフォルトの最小長は 0 文字でした。パスワードのデフォルトの最小長は 3 文字に変更されています。使用しているアプリケーションで 0 文字のパスワードが許可されている場合、SQL Anywhere のデータベースで次の文を実行し、以前のリリースに合わせてデフォルト設定を変更することができます。

```
SET OPTION PUBLIC.min_password_length=0;
```

詳細については、「min_password_length オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **ローカルテンポラリテーブルでインデックスを作成する場合のデフォルト動作の変更**　以前のリリースでは、データベースサーバは COMMIT の実行後にローカルテンポラリテーブルにインデックスを作成していました。現在、データベースサーバが COMMIT の実行後にローカルテンポラリテーブルにインデックスを作成するというのは通常の動作でなくなっています。この動作を制御するため、auto_commit_on_create_local_temp_index データベースオプションを設定することができます。「auto_commit_on_create_local_temp_index オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **新しいパケット最大サイズ**　パケットの最大サイズが 16000 バイトから 65535 バイトに増加されました。SQL Anywhere 12 以前のクライアントは、SQL Anywhere 16 のデータベースサーバへの接続時のサイズが 16000 バイトに制限されています。このデフォルトパケットサイズは変更されていません。「CommBufferSize (CBSIZE) 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **-ch データベースサーバオプションの制限の変更**　-ch オプションを使用して 64 MB 未満のキャッシュ最大サイズを指定した場合、データベースサーバによってキャッシュ最大サイズが 64 MB に調整されます。64 MB 未満のキャッシュ最大サイズが必要な場合 (この設定は推奨しません)、-chx オプションを使用します。この変更は Windows Mobile には適用されません。「-ch データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **インデックスの同時構築**　以前のリリースでは、インデックスの構築時に CREATE INDEX 文に EXCLUSIVE テーブルロックを指定する必要がありました。現在、操作の最初と最後の短時間だけ EXCLUSIVE テーブルロックを設定し、操作の大半の時間は SHARED ロックを設定すればよいため、インデックスの作成中に他の接続からテーブルデータへのアクセスが可能になっています。インデックス作成のための接続では、インデックスの作成が完了するまでテーブルへのアクセスがブロックされます。この機能を使用するには、既存のデータベースをアップグレードする必要があります。「CREATE INDEX 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **接続によりトランザクションログの適用が阻害される場合にミラーリング接続の削除が可能**　コピーノードまたはミラーデータベースへの接続は、トランザクションログの適用を阻害しているとき、一部の場合削除されます。たとえば、トランザクションログが変更または削除しようとしているプロシージャが接続で使用されている場合は、トランザクションログが適用されないようにブロックしている接続が削除され、サーバコンソールにメッセージが出力されます。「ミラーデータベースで実行されたクエリ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **プランのキャッシュの変更**　クエリ実行プランは、実行時間の長いクエリについてはキャッシュされません。これは、クエリの最適化を避けることのメリットが、クエリの総コストに比べて小さいためです。さらに、データベースサーバでは、ホスト変数の値に大きく左右されるクエリの再利用可能なクエリプランの再構築は行われません。

- **HTTP サーバへの要求変数のための新しいデフォルト**　以前のリリースでは、1 回の要求で送信できる HTTP 入力変数の数に制限はありませんでした。MaxRequestVars プロトコルオプションで HTTP 入力変数の数が制限できるようになりました。MaxRequestVars プロトコルオプションの変数の数のデフォルト値は 10000 です。「MaxRequestVars (MAXVARS) プロトコルオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **データベースアサーションの新しいデフォルト動作**　以前のリリースでは、データベースアサーション障害はデータベースサーバアサーション障害として処理され、データベースサーバがシャットダウンされるか、またはすべてのクライアント接続に対してエラーが返されました。現在、データベースアサーション障害はデータベースサーバアサーション障害とは区別して処理され、データベースサーバが動作を継続した状態でデータベースがシャットダウンされるようになりました。「-ufd データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **セキュリティ機能の動作変更**　セキュリティ機能キーの最小長が 6 文字になり、一部のセキュリティ機能はグローバルで無効化できず、接続ごとに単体でのみ無効化できるようになりました。「セキュリティ機能」『SQL Anywhere サーバ データベース管理』を参照してください。

- **出力での機密情報の難読化**　パスワードと暗号化キーを含む文が要求ログに出力されたり、診断トレーシングによって記録されたり、DESCRIBE 文によってカラム名として使用されたりする場合、パスワードと暗号化キーは難読化されます。この動作は、イベントトのトレース出力に記録される文、LastStatement 接続プロパティ、REWRITE 関数の出力にも適用されます。

  機密パラメータ、パスワード、キーは、次のファンクション、プロシージャ、文に対して非表示になります。

  - ENCRYPT 関数
  - DECRYPT 関数
  - sa_verify_password システムプロシージャ
  - sp_addlogin プロシージャ (Adaptive Server Enterprise 互換性プロシージャ)
  - sp_password システムプロシージャ
  - xp_startmail システムプロシージャ
  - xp_startsmtp システムプロシージャ
  - sp_create_secure_feature_key システムプロシージャ
  - sp_alter_secure_feature_key システムプロシージャ
  - sp_use_secure_feature_key システムプロシージャ
  - GRANT CONNECT 文
  - CREATE DATABASE 文
  - CREATE ENCRYPTED DATABASE 文
  - CREATE ENCRYPTED TABLE DATABASE 文
  - CREATE ENCRYPTED FILE 文
  - CREATE DECRYPTED DATABASE 文
  - CREATE DECRYPTED FILE 文
  - START DATABASE 文
  - DROP DATABASE 文
  - CREATE EXTERNLOGIN 文
  - SET TEMPORARY OPTION secure_feature_key = *key* 文

- **パーソナルサーバ (dbeng16) のライセンスの変更**　パーソナルサーバでは、1 つの CPU 上のコアは 4 つに制限されています。以前は、パーソナルサーバは CPU が 1 つに制限されていました。「サーバライセンス取得ユーティリティ (dblic)」『SQL Anywhere サーバ データベース管理』を参照してください。

データベースユーティリティの動作の変更

- **アップグレードユーティリティ (dbupgrad) の動作の変更**　デフォルトでは、データベースはアップグレード後に再起動されます。データベースが再起動されないようにするには、-nrs オプションを使用してアップグレード後にデータベースを停止します。アップグレードが完了すると、トランザクションログの名前が変更されます。「アップグレードユーティリティ (dbupgrad)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **サポートユーティリティ (dbsupport) の動作の変更**

  - **-cc 自動送信オプション**　従来は、-cc 自動送信オプションを指定した場合、クラッシュの発生時のみクラッシュレポートと診断情報が自動的に送信されていました。現在、-cc 自動送信オプションを設定した場合、クラッシュレポートが自動的に送信されるだけでなく、製品の改良に役立てるため、ソフトウェアの開発チームに診断情報が定期的に送信されるようになりました。

    このオプションは、インストール中にデフォルトで有効になります。「パフォーマンスデータの定期送信」『SQL Anywhere サーバ データベース管理』を参照してください。

  - **-cp 自動検出オプション**　Unix プラットフォームで -cp 自動検出オプションがサポートされるようになりました。Unix では、-cp 自動検出オプションを使用すると、HTTP_PROXY 環境変数によって設定されるプロキシサーバとポートを使用して dbsupport でプロキシサーバとポートを設定できます。「サポートユーティリティ (dbsupport)」『SQL Anywhere サーバ データベース管理』を参照してください。

  - **-ch オプション (旧式)**　dbsupport の -ch オプションと -ch- オプションはサポートされなくなりました。また、dbsupport の -e オプションでは -ch オプションを受け付けなくなりました。そのため、-ech は指定できません。「サポートユーティリティ (dbsupport)」『SQL Anywhere サーバ データベース管理』を参照してください。

システムプロシージャとファンクションの動作の変更

次に、SQL Anywhere バージョン 16.0 で追加されたシステムプロシージャとファンクションの変更点を示します。

- **他の接続の情報を返すために権限が必要**　バージョン 12 以前のデータベースの場合、接続の情報を返すために次のシステムプロシージャとファンクションを実行するためのパーミッションは不要です。

  バージョン 16 のデータベースの場合、誰でもこれらのシステムプロシージャとファンクションを使用して、現在の接続に関する情報を返すことができます。ただし、これらのシステム

プロシージャとファンクションを実行して他の接続に関する情報を返す場合は、SERVER OPERATOR、MONITOR、または DROP CONNECTION のシステム権限が必要です。

- 「CONNECTION_EXTENDED_PROPERTY 関数 [文字列]」『SQL Anywhere サーバ SQL リファレンス』
- 「CONNECTION_PROPERTY 関数 [システム]」『SQL Anywhere サーバ SQL リファレンス』
- 「NEXT_CONNECTION 関数 [システム]」『SQL Anywhere サーバ SQL リファレンス』
- 「sa_conn_activity システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
- 「sa_conn_compression_info システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
- 「sa_conn_info システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
- 「sa_conn_list システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
- 「sa_conn_options システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
- 「sa_conn_properties システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』

● **他のデータベースの情報を返すために権限が必要**　バージョン 12 以前のデータベースの場合、データベースの情報を返すために次のシステムプロシージャとファンクションを実行するためのパーミッションは不要でした。

バージョン 16 のデータベースの場合、誰でもこれらのシステムプロシージャとファンクションを使用して、現在のデータベースに関する情報を返すことができます。ただし、これらのシステムプロシージャとファンクションを実行して他のデータベースに関する情報を返す場合は、SERVER OPERATOR または MONITOR のシステム権限が必要です。

- 「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバ SQL リファレンス』
- 「DB_NAME 関数 [システム]」『SQL Anywhere サーバ SQL リファレンス』
- 「DB_PROPERTY 関数 [システム]」『SQL Anywhere サーバ SQL リファレンス』
- 「NEXT_DATABASE 関数 [システム]」『SQL Anywhere サーバ SQL リファレンス』
- 「sa_db_info システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
- 「sa_db_list システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
- 「sa_db_properties システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』

● **TRACEBACK 関数の強化**　TRACEBACK 関数はオブジェクト名と行番号による注釈がつけられた呼び出しスタックを返します。これにより、呼び出しスタック内でレポートされる実際の文を容易に探すことができます。「TRACEBACK 関数 [その他]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

● **SMTP および MAPI リターンコードの変更**　xp_startmail、xp_startsmtp、xp_sendmail、xp_stopmail、xp_stopsmtp システムプロシージャで使用されるリターンコードは、次の点が変更されています。

リターンコード	以前のリリースでの意味	バージョン 16.0 のデータベースでの意味	その他の情報
-1		未知のエラー	新しいエラーコード
0	成功	成功	

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

リターンコード	以前のリリースでの意味	バージョン 16.0 のデータベースでの意味	その他の情報
1		無効なパラメータが指定された	新しいエラーコード
2	xp_startmail または xp_startsmtp が失敗	メモリ不足	
3	xp_stopmail または xp_stopsmtp が失敗	xp_startmail または xp_startsmtp が呼び出されなかった	
4		ホスト名が不正	新しいエラーコード
5	xp_sendmail が失敗	接続エラー	
6		セキュリティで保護された接続のエラー	新しいエラーコード
7		MAPI 関数を使用できない	新しいエラーコード

xp_get_mail_error_code および xp_get_mail_error_text システムプロシージャはリターンコードに関する追加情報を返します。新しいシステムプロシージャとエラーコードを取得するには、既存のデータベースをアップグレードしてください。 次の項を参照してください。

- 「xp_get_mail_error_code システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
- 「xp_get_mail_error_text システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
- 「MAPI と SMTP 用のシステムプロシージャのリターンコード」『SQL Anywhere サーバ SQL リファレンス』

使用しているアプリケーションで更新後のリターンコードをチェックするように変更できない場合、SQL Anywhere の *scripts* サブディレクトリにあるバージョン 12 の *dbext.dll* ファイルと *use_old_dbextf.sql* スクリプトを使用することにより、データベースサーバでソフトウェアの以前のリリースのリターンコードが使用されるよう設定することができます。 「データベースサーバの配備」『SQL Anywhere サーバ プログラミング』を参照してください。

- **USER_NAME 関数と SUSER_NAME 関数による VARCHAR の返却**　USER_NAME 関数と SUSER_NAME 関数によって、LONG VARCHAR 型ではなく、VARCHAR 型のデータが返されるようになりました。これらの関数を使用するマテリアライズドビューを使用している場合、再構築する必要があります。

- **エラー処理**　ローの挿入または解析中に文でエラーが発生した場合に LOAD TABLE 操作を可能にするため、問題のあるローを ROW LOG に書き込み、エラーメッセージを MESSAGE

LOG に書き込むことができます。指定回数のエラーが発生すると、文の実行は失敗し、ロールバックされます。この機能をサポートするために、次の句が追加されています。

- ALLOW...ERRORS
- ROW LOG...
- MESSAGE LOG...

「LOAD TABLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

プログラムインタフェースの動作の変更

次に、SQL Anywhere バージョン 16.0 でのプログラミングインタフェースの動作の変更を示します。

- **メモリ不足の致命的なエラー**　以前のバージョンのクライアントライブラリでは、クライアントライブラリが動的メモリを取得できない場合に Windows 上で Out of memory ウィンドウが表示されました。これは、ユーザの応答を待ってアプリケーションを終了するものでした。クライアントライブラリのこの機能は無くなりました。現在、メモリ不足が発生すると、Windows のイベントログのエントリがイベントソース SQLANY16.0 (32-bit アプリケーション) または SQLANY64 16.0 (64-bit アプリケーション) の下に書き込まれます。イベントログのエントリの書き込み後に、このエラー状態を処理するための要求ルーチンの許可の実行が続行されます。この変更により、Windows サービスとして書き込まれるクライアントアプリケーションのハングが防止されます。アプリケーションがメモリ不足になった原因の特定はクライアントアプリケーション開発者の責任です。「イベントログメッセージのフォーマット」『SQL Anywhere サーバ プログラミング』を参照してください。

- **ODBC SQLTables 関数**　**SQLTables** 関数は、CREATE DATABASE 文によってインストールされた dbo が所有するシステムテーブルを、TABLE (結果セットの TABLE_TYPE カラム内) ではなく SYSTEM TABLE として分類するよう修正されています。dbo.jdbc_function_escapes や dbo.spt_jdatatype_info などのテーブルはシステムテーブルに分類されます。

  SQLTables 関数は、CREATE DATABASE 文によってインストールされた dbo が所有するシステムビューを、VIEW (結果セットの TABLE_TYPE カラム内) ではなく SYSTEM VIEW として分類するよう修正されています。dbo.systypes や dbo.sysusers などのビューはシステムビューに分類されます。

- **JDBC DatabaseMetaData.getTables メソッド**　DatabaseMetaData クラスの **getTables** メソッドは、CREATE DATABASE 文によってインストールされた dbo が所有するシステムテーブルを、TABLE (結果セットの TABLE_TYPE カラム内) ではなく SYSTEM TABLE として分類するよう修正されています。dbo.jdbc_function_escapes や dbo.spt_jdatatype_info などのテーブルはシステムテーブルに分類されます。

  getTables メソッドは、CREATE DATABASE 文によってインストールされた dbo が所有するシステムビューを、VIEW (結果セットの TABLE_TYPE カラム内) ではなく SYSTEM VIEW として分類するよう修正されています。dbo.systypes や dbo.sysusers などのビューはシステムビューに分類されます。

- **データベースツール C API の変更**　**script_full_path** フィールドは、**a_sync_db** 構造体から削除されています。バージョン 8.x 以降では、このフィールドの使用がサポートされていませ

ん。 a_sync_db 構造体 [データベースツール]『SQL Anywhere サーバ プログラミング』を参照してください。

a_backup_db 構造体に新しいフィールド (wait_before_start、wait_after_end、backup_comment、auto_tune_writers、backup_history) と列挙体が追加されています。 a_backup_db 構造体 [データベースツール]『SQL Anywhere サーバ プログラミング』を参照してください。

- **JSON オブジェクトへの識別子ルールとデータ型の制限の適用** JSON オブジェクト識別子は、データベースサーバで定義されている識別子ルールに準拠している必要があります。また、データベースサーバでは、基本となる ROW データ型および ARRAY データ型の場合と同じ制限が JSON データ型に課せられます。

SQL 文の動作の変更

次に、SQL Anywhere バージョン 16.0 での SQL の動作の変更を示します。

ALTER DATABASE 文 デフォルトで、データベースが停止して、アップグレード後に再起動されます。データベースが再起動されないようにするには、RESTART OFF 句を使用してアップグレード後にデータベースを停止します。「ALTER DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

ビュー定義でサポートされている SELECT * これまでのリリースでは、SELECT * は、CREATE VIEW 文のメインクエリでのみサポートされていました。現在は、メインクエリ、サブクエリ、派生テーブル、または、CREATE VIEW 文の subselect でサポートされています。 「CREATE VIEW 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

START DATABASE 文 START DATABASE 文の ON 句は廃止されています。START DATABASE 文は、現在のデータベースサーバ上でのみデータベースを起動できます。従来は、Interactive SQL で ON 句を使用して、使用するデータベースサーバを指定できました。 「START DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

ALTER SERVER 文 ALTER SERVER 文の CONNECTION CLOSE 句はサポートされていません。DROP REMOTE CONNECTION 文を使用します。 「DROP REMOTE CONNECTION 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

LOAD TABLE 文

- **STRIP ON 句** LOAD TABLE 文の STRIP ON 句はサポートされていません。代わりに、STRIP RTRIM 句を使用してください。 「LOAD TABLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

SYSUSER システムビューによるスタンドアロンロール情報の格納 従来、SYSUSER システムビューはシステム内のユーザのみを格納していました。現在、このビューにはスタンドアロンロール (ユーザと関連のないロール) も格納されるようになりました。ただし、これらのロールで意味を持つのは、user_id、object_id、user_name、user_type カラムのみです。 「SYSUSER システムビュー」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

SQL Anywhere の廃止予定機能とサポート終了機能

廃止予定機能

バージョン 16.0 で廃止された機能のリストを次に示します。

- **廃止された権限**　SQL Anywhere 16.0 には、以前の権限およびパーミッションセキュリティモデルに代わって、ロールベースおよび権限ベースの新しいセキュリティモデルが導入されています。新しいロールベースのセキュリティモデルにより、ユーザが実行できる権限付きのタスクのきめ細かい制御が可能になり、アクセス制御の管理が簡素化されます。従来のデータベース権限は廃止され、システムロールまたは互換ロールに置き換わっています。

  - DBA 権限
  - REMOTE DBA 権限
  - BACKUP 権限
  - VALIDATE 権限
  - PROFILE 権限
  - READCLIENTFILE 権限
  - WRITECLIENTFILE 権限
  - READFILE 権限
  - WRITEFILE 権限
  - RESOURCE 権限

  「権限、パーミッション、グループの変更」『SQL Anywhere サーバ データベース管理』を参照してください。

- **java_location オプションと java_main_userid オプション**　バージョン 11.0.0 以降では、ISYSEXTERNENV システムテーブルの導入に伴って java_location オプションと java_main_userid オプションが廃止されています。使用する Java VM を指定するのに java_location オプションを使用していたアプリケーションでは、代わりに ALTER EXTERNAL ENVIRONMENT 文を使用して ISYSEXTERNENV テーブル内で Java VM の場所を設定してください。java_main_userid オプションの代用となるものはありません。「SQL Anywhere 外部環境のサポート」『SQL Anywhere サーバ プログラミング』を参照してください。

- **secure_feature_key データベースオプション**　secure_feature_key データベースオプションは廃止される予定です。sp_use_secure_feature_key システムプロシージャを使用してください。「sp_use_secure_feature_key システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **PHP 関数**　次の PHP 関数は廃止される予定です。これらの各関数には、名前の先頭に sqlanywhere_ ではなく sasql_ の付いた、同等の機能を持つ新しい関数があります。

廃止予定の PHP 関数	同等の新しい PHP 関数
sqlanywhere_commit	「sasql_commit」『SQL Anywhere サーバ プログラミング』

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

廃止予定の PHP 関数	同等の新しい PHP 関数
sqlanywhere_connect	「sasql_connect」『SQL Anywhere サーバ プログラミング』
sqlanywhere_data_seek	「sasql_data_seek」『SQL Anywhere サーバ プログラミング』
sqlanywhere_disconnect	「sasql_disconnect」『SQL Anywhere サーバ プログラミング』
sqlanywhere_error	「sasql_error」『SQL Anywhere サーバ プログラミング』
sqlanywhere_errorcode	「sasql_errorcode」『SQL Anywhere サーバ プログラミング』
sqlanywhere_execute	
sqlanywhere_fetch_array	「sasql_fetch_array」『SQL Anywhere サーバ プログラミング』
sqlanywhere_fetch_field	「sasql_fetch_field」『SQL Anywhere サーバ プログラミング』
sqlanywhere_fetch_object	「sasql_fetch_object」『SQL Anywhere サーバ プログラミング』
sqlanywhere_fetch_row	「sasql_fetch_row」『SQL Anywhere サーバ プログラミング』
sqlanywhere_free_result	「sasql_free_result」『SQL Anywhere サーバ プログラミング』
sqlanywhere_identity	
sqlanywhere_num_fields	「sasql_num_fields」『SQL Anywhere サーバ プログラミング』
sqlanywhere_num_rows	「sasql_num_rows」『SQL Anywhere サーバ プログラミング』
sqlanywhere_pconnect	「sasql_pconnect」『SQL Anywhere サーバ プログラミング』
sqlanywhere_query	「sasql_query」『SQL Anywhere サーバ プログラミング』

廃止予定の PHP 関数	同等の新しい PHP 関数
sqlanywhere_result_all	「sasql_result_all」『SQL Anywhere サーバ プログラミング』
sqlanywhere_rollback	「sasql_rollback」『SQL Anywhere サーバ プログラミング』
sqlanywhere_set_option	「sasql_set_option」『SQL Anywhere サーバ プログラミング』

サポート終了機能

バージョン 16.0 でサポートが終了した機能のリストを次に示します。

- **SQL Anywhere JDBC 3.0 ドライバ**　SQL Anywhere JDBC 3.0 ドライバはこのソフトウェアリリースに含まれません。アプリケーションで *sajdbc.jar* を使用している場合は、*sajdbc4.jar* に切り替えてください。「JDBC サポート」『SQL Anywhere サーバ プログラミング』を参照してください。

- **JDBC ベースのリモートデータアクセスサーバクラス**　このソフトウェアリリースでは、JDBC ベースのリモートサーバクラスはサポートされていません。これらのクラスには、ASEJDBC、IQJDBC、SAJDBC のリモートサーバクラスが含まれます。「CREATE SERVER 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **OLE DB プロバイダによる Borland Delphi のサポート終了**　SQL Anywhere OLE DB プロバイダを Borland Delphi の古いバージョンと一緒に使用できなくなりました。Borland Delphi は DBTYPE_DBTIME2 データ型と DBTYPE_DBTIMESTAMPOFFSET データ型をサポートしていません。Embarcadero は 2006 年以前のバージョンの Delphi のサポートを終了しています。http://www.embarcadero.com/products/delphi/previous-versions を参照してください。

- **Address Windowing Extensions (AWE) はサポート対象外**　32-bit Windows での Address Windowing Extensions の使用はサポートされておらず、-cw と -cm のデータベースサーバオプションのサポートは終了しています。サイズの大きいキャッシュが必要な場合は、64 ビット版のオペレーティングシステムで 64 ビット版の SQL Anywhere データベースサーバを使用してください。

Mobile Link の新機能

次に、バージョン 16.0 で導入された Mobile Link の追加機能を示します。サポートされているオペレーティングシステムおよびバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

- **Mobile Link モニタを Mobile Link プロファイラに置き換え**　Mobile Link モニタは削除されました。Mobile Link プロファイラを使用して、開発およびテスト中の同期化に関する詳細なパフォーマンス情報を取得します。SQL Anywhere モニタを使用して、配備された Mobile Link の運用をモニタリングします。「Mobile Link プロファイラ」『Mobile Link サーバ管理』を参照してください。

- **Java と .NET データスクリプトからの SQL の返却廃止**　Java と .NET 同期スクリプトから、Mibile Link サーバによって SQL 文として解釈される文字列が返されなくなりました。Java と .NET から SQL を返すとエラーになり、同期に失敗します。

  統合データベースとの間でデータのアップロードとダウンロードを行う必要のある Java または .NET のダイレクトローハンドリングを使用することをお勧めします。「ダイレクトローハンドリング」『Mobile Link サーバ管理』を参照してください。

- **Mobile Link サーバの下位互換と上位互換**　Mobile Link クライアントのバージョン 16 の任意のビルドと Mobile Link サーバのバージョン 16 の任意のビルドとの同期ができるようになりました。従来は、Mobile Link サーバのビルドが Mobile Link クライアントのビルド以上でなければなりませんでした。

- **Oracle XMLTYPE データ型のサポート**　Oracle XMLTYPE データ型を使用したカラムとの同期がサポートされるようになりました。Oracle の XMLTYPE データ型『Mobile Link サーバ管理』を参照してください。

- **Mobile Link サーバによる統合化 LDAP 認証のサポート**　Lightweight Directory Access Protocol (LDAP) サーバに対する Mobile Link ユーザ認証が Mobile Link サーバで利用できるようになりました。LDAP 認証を実装には、次の新しい Mobile Link のストアドプロシージャを使用します。

  - **ml_add_ldap_server システムプロシージャ**　LDAP サーバの追加、削除、更新を行うには、このシステムプロシージャを使用します。「ml_add_ldap_server システムプロシージャ」『Mobile Link サーバ管理』を参照してください。

  - **ml_add_certificates_file ストアドプロシージャ**　Mobile Link サーバと LDAP サーバの間の暗号化 TLS 通信に必要な信頼できる証明書を設定するには、このシステムプロシージャを使用します。「ml_add_certificates_file システムプロシージャ」『Mobile Link サーバ管理』を参照してください。

  - **ml_add_user_auth_policy ストアドプロシージャ**　Mobile Link ユーザ認証ポリシーを設定するには、このシステムプロシージャを使用します。「ml_add_user_auth_policy システムプロシージャ」『Mobile Link サーバ管理』を参照してください。

- **新しい ml_model_drop システムプロシージャ**　動機モデルとそれに関連するスキーマを削除するには、このシステムプロシージャを使用します。削除できるのは、Sybase Central 向け Mobile Link 16.0 プラグインを使用して追加された同期モデルのみです。「ml_model_drop システムプロシージャ」『Mobile Link サーバ管理』を参照してください。

- **新しい ml_model_check_all_schema システムプロシージャ**　このプロシージャを使用して、配備された同期モデルに必要な各スキーマオブジェクトのステータスをチェックします。このストアドプロシージャは、配備されたすべての同期モデルの情報を返します。「ml_model_check_all_schema システムプロシージャ」『Mobile Link サーバ管理』を参照してください。

- **新しい ml_model_check_version_schema システムプロシージャ**　このプロシージャを使用して、配備された同期モデルに必要な各スキーマオブジェクトのステータスをチェックします。このストアドプロシージャは、指定したスクリプトバージョンの情報を返します。

「ml_model_check_all_schema システムプロシージャ」『Mobile Link サーバ管理』を参照してください。

- **Mobile Link サーバおよびクライアントによる TLS バージョン 1.1 のサポート**　Mobile Link サーバと Mobile Link クライアントで TLS バージョン 1.1 がサポートされるようになりました。ダイレクト TLS と HTTPS 同期の両方に適用されます。TLS バージョン 1.1 は Mac OS X ではサポートされていません。

- **ロック情報とブロック情報のレポート**　Mobile Link サーバはデフォルトで、Mobile Link サーバのログファイルに情報を記録するほか、長時間実行されているユーザ定義スクリプトのロック情報とブロック情報を Mobile Link プロファイラにレポートするようになりました。

  この機能のためにプロパティを設定するには、ml_add_property ストアドプロシージャを使用します。「ml_add_property システムプロシージャ」『Mobile Link サーバ管理』を参照してください。

- **イベントトレース用の -ts オプション**　トレースセッションを設定するには、mlsrv16、dbmlsync、dblsn ユーティリティと一緒に新しい -ts オプションを使用します。

  次の項を参照してください。

  - 「-ts mlsrv16 オプション」『Mobile Link サーバ管理』
  - 「-ts dbmlsync オプション」『Mobile Link クライアント管理』
  - 「-ts dblsn オプション」『Mobile Link サーバ起動同期』

統合データベース

次に、SQL Anywhere バージョン 16.0 での Mobile Link 用の統合データベースサポートの強化を示します。

推奨ドライバの詳細については、http://www.sybase.com/detail?id=1011880 を参照してください。

- **SQL Anywhere 16 統合データベースのサポート**　Mobile Link サーバにより、SQL Anywhere 16 統合データベースがサポートされるようになりました。「SQL Anywhere 統合データベース」『Mobile Link サーバ管理』を参照してください。

- **SAP HANA 1.0 データベースサーバのサポート**　Mobile Link サーバにより、SAP HANA データベースサーバのバージョン 1.00.32 以降がサポートされるようになりました。「SAP HANA の統合データベース」『Mobile Link サーバ管理』を参照してください。

- **SAP Sybase IQ 15.4 統合データベースのサポート**　Mobile Link サーバにより、SAP Sybase IQ 15.4 (ESD #1 以降) サーバ上で動作する統合データベースがサポートされるようになりました。「SAP Sybase IQ の統合データベース」『Mobile Link サーバ管理』を参照してください。

- **Microsoft SQL Server 2012 統合データベースのサポート**　Mobile Link サーバにより、Microsoft SQL Server 2012 サーバ上で動作する統合データベースがサポートされるようになりました。このサポートは、Microsoft Windows が動作する Mobile Link サーバのみに限定されます。「Microsoft SQL Server 統合データベース」『Mobile Link サーバ管理』を参照してください。

- **MySQL サーバ 5.5.X (X は 16 以降) 統合データベースのサポート**　Mobile Link サーバにより、バージョン 5.5.x (x は 16 以降) の MySQL サーバ上で動作する統合データベースがサポートされるようになりました。「MySQL 統合データベース」『Mobile Link サーバ管理』を参照してください。

- **Sybase Adaptive Server Enterprise 15.7 統合データベースのサポート**　Mobile Link サーバにより、Sybase ASE 15.7 サーバ上で動作する統合データベースがサポートされるようになりました。「Adaptive Server Enterprise 統合データベース」『Mobile Link サーバ管理』を参照してください。

- **IBM DB2 LUW 10.1 統合データベースのサポート**　Mobile Link サーバにより、IBM DB2 LUW 10.1 統合データベースがサポートされるようになりました。「IBM DB2 LUW 統合データベース」『Mobile Link サーバ管理』を参照してください。

- **Oracle 11g 統合データベースのサポート**　Mobile Link サーバにより、Oracle 11g 統合データベースがサポートされるようになりました。「Oracle 統合データベース」『Mobile Link サーバ管理』を参照してください。

Mobile Link サーバ

次に、SQL Anywhere バージョン 16.0 での Mobile Link サーバの強化を示します。

mlsrv16 の新機能

- **Mobile Link サーバによるマルチスレッドネットワーク処理のサポート**　Mobile Link サーバにより、ネットワークストリームを同時に処理する複数のネットワークワーカスレッドがサポートされるようになりました。ストリームスレッドを設定するには、新しい -wn mlsrv16 オプションを使用します。「-wn mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **mlsrv16 -x オプション用の header_limit オプション**　mlsrv16 -x オプション用の新しい header_limit オプションを使用すると、指定された要求で許容されるヘッダデータの最大量を指定できます。HTTP、HTTPS、OE プロトコルで使用できます。「-x mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **mlsrv16 -x オプション用の trusted_certificates オプション**　暗号化が有効になっている場合、trusted_certificates オプションを指定するとクライアントの検証が行われます。NetworkData.ClientCertificates API でこのオプションを使用すると、Mobile Link によりエンタープライズ証明書インフラストラクチャとの統合が強化できます。

  次の項を参照してください。

  - 「-x mlsrv16 オプション」『Mobile Link サーバ管理』
  - NetworkData インタフェース [Mobile Link サーバ Java]『Mobile Link サーバ管理』
  - NetworkData インタフェース [Mobile Link サーバ .NET]『Mobile Link サーバ管理』

Mobile Link の新しいスクリプト機能

- **新しい authentication_message 名前付きシステムパラメータ**　Mobile Link サーバにより、新しい authentication_message 名前付きシステムパラメータがサポートされるようになりし

た。名前付きパラメータは、authenticate_user、authenticate_user_hashed、authenticate_parameters スクリプトなどのあらゆるユーザ認証イベントで使用できます。

次の項を参照してください。

○「authenticate_user 接続イベント」『Mobile Link サーバ管理』
○「authenticate_user_hashed 接続イベント」『Mobile Link サーバ管理』
○「authenticate_parameters 接続イベント」『Mobile Link サーバ管理』

- **新しい script_version 名前付きシステムパラメータ**　Mobile Link サーバにより、現在の同期に使用されるスクリプトのバージョン文字列を Mobile Link サーバがこのパラメータに渡すよう指定するための、新しい script_version パラメータがサポートされるようになりました。この名前付きパラメータは基本的にあらゆる Mobile Link サーバの同期イベントに使用できますが、begin_connection、begin_connection_autocommit、end_connection イベントには使用できません。「名前付きスクリプトパラメータ」『Mobile Link サーバ管理』を参照してください。

Mobile Link サーバユーティリティ

次に、SQL Anywhere バージョン 16.0 での Mobile Link サーバユーティリティの強化を示します。

- **mluser ユーティリティの新しい -r オプション**　指定されたユーザ名とリモート ID の同期状態をリセットするには、mluser -u オプションと一緒にこのオプションを使用します。「Mobile Link ユーザ認証ユーティリティ (mluser)」『Mobile Link サーバ管理』を参照してください。

- **mluser ユーティリティの新しい -n オプション**　LDAP ユーザ認証のために Mobile Link ユーザを登録するにはこのオプションを使用します。「Mobile Link ユーザ認証ユーティリティ (mluser)」『Mobile Link サーバ管理』を参照してください。

- **Mobile Link 監視サーバの新しい停止ユーティリティ**　Mobile Link 監視サーバを停止するには、新しい mlarbstop ユーティリティを使用します。「Mobile Link 監視停止ユーティリティ (mlarbstop)」『Mobile Link サーバ管理』を参照してください。

- **mlreplay 用の新しい -sv オプション**　リプレイセッションのスクリプトバージョンをユーザ定義の値に置き換えるには、mlreplay の新しい -sv オプションを使用します。「Mobile Link Replay ユーティリティ (mlreplay)」『Mobile Link サーバ管理』を参照してください。

- **mlreplay の -sci オプションの新しいスクリプトバージョンフィールド**　新しいスクリプトバージョンフィールドは、mlreplay の -sci オプションで指定されるカンマ区切りのクライアントのシミュレーション情報ファイルの最終フィールドです。「Mobile Link Replay ユーティリティ (mlreplay)」『Mobile Link サーバ管理』を参照してください。

- **mlreplay の新しい活性タイムアウト設定**　Mobile Link Replay ユーティリティ (mlreplay) では、コマンドライン上で -x mlsrv16 オプションによって指定されたストリームオプションを使用して、活性タイムアウトを変更できます。「Mobile Link Replay ユーティリティ (mlreplay)」『Mobile Link サーバ管理』を参照してください。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

Mobile Link プロファイラ

Mobile Link プロファイラは同期のパフォーマンスに関する詳細情報を提供します。Mobile Link プロファイラでは、Mobile Link モニタで利用可能だったフェーズレベルのタイミングが取得できるほか、イベントレベルのタイミング、イベントとフェーズのサンプリング、Mobile Link サーバで長時間実行されている SQL 文が検出されたときの統合データベースのブロック情報を通して、詳細なパフォーマンス分析とボトルネックの特定が行えます。

プロファイリング結果はすべて SQL Anywhere データベースに保存されます。これは、Mobile Link プロファイラにロードしなおしたり、別のデータベースクライアントから問い合わせたりできます。Mobile Link プロファイラは、プロファイリングデータベースの作成、起動、接続、停止の処理を自動的に行います。

「Mobile Link プロファイラ」『Mobile Link サーバ管理』を参照してください。

Mobile Link モニタ

Mobile Link モニタは削除されました。Mobile Link プロファイラを使用して、開発およびテスト中の同期化に関する詳細なパフォーマンス情報を取得します。SQL Anywhere モニタを使用して、配備された Mobile Link の運用をモニタリングします。「Mobile Link プロファイラ」『Mobile Link サーバ管理』と「SQL Anywhere モニタ」『SQL Anywhere サーバ データベース管理』を参照してください。

Mobile Link クライアント

次に、SQL Anywhere バージョン 16.0 での Mobile Link クライアントの強化を示します。

- **ロールベースのアクセス制御 (RBAC) と同期**　新しい SQL Anywhere 16.0 のロールベースおよび権限ベースのセキュリティモデルでは、同期を実行するには SYS_RUN_REPLICATION_ROLE を付与し、同期に関連する管理タスクを実行するには SYS_REPLICATION_ADMIN_ROLE を付与する必要があります。これらの各ロールには、各種システム権限が含まれています。同期環境のセキュリティの制御を強化するために、ロールと権限を付与したり、取り消したりすることができます。

  次の項を参照してください。

  - 「ロールベースのアクセス制御と同期の場合のセキュリティに関する考慮事項」『Mobile Link クライアント管理』
  - 「新しいセキュリティモデル：ロールベースのアクセス制御 (RBAC)」1 ページ
  - 「ユーザのセキュリティ (ロールと権限)」『SQL Anywhere サーバ データベース管理』
  - 「GRANT ROLE SYS_REPLICATION_ADMIN_ROLE 文 [Mobile Link] [SQL Remote]」『SQL Anywhere サーバ SQL リファレンス』
  - 「GRANT ROLE SYS_RUN_REPLICATION_ROLE 文 [Mobile Link] [SQL Remote]」『SQL Anywhere サーバ SQL リファレンス』
  - 「REVOKE ROLE SYS_REPLICATION_ADMIN_ROLE 文 [Mobile Link] [SQL Remote]」『SQL Anywhere サーバ SQL リファレンス』
  - 「REVOKE ROLE SYS_RUN_REPLICATION_ROLE 文 [Mobile Link] [SQL Remote]」『SQL Anywhere サーバ SQL リファレンス』

- **Mobile Link クライアントの新しい trusted_certificate_name ネットワークプロトコルオプション**　trusted_certificate_name プロトコルオプションを使用すると、リモートデータベースに格納されている信頼できるルート証明書を指定できます。このオプションは、SQL Anywhere クライアントでのみ使用できます。「trusted_certificate_name」『Mobile Link クライアント管理』を参照してください。

- **HTTP のパフォーマンスの向上**　Mobile Link クライアントから発行される HTTP 要求が、以前のバージョンよりも少なくなりました。

Dbmlsync C++ API の変更

- **Connect メソッドに必要な新しい権限**　Connect メソッドで指定された userid に必要な権限が変更されました。 DbmlsyncClient.Connect メソッド [Dbmlsync C++]『Mobile Link クライアント管理』を参照してください。

- **DBSC_Event に新しい 'val3' 整数フィールドがあります**　DBSC_Event に新しい 'val3' 整数フィールドがあります。val3 フィールドの値の意味は、イベントのタイプに応じて異なります。 DBSC_Event 構造体 [Dbmlsync C++]『Mobile Link クライアント管理』を参照してください。

- **DBSC_EVENTTYPE_DOWNLOAD_COMMITTED には 2 つのパラメータがあります**　イベントの DBSC_Event 構造体では、ダウンロード中に挿入または更新されたローの数が val1 に、ダウンロードで削除されたローの数が val2 に含まれます。

- **DBSC_EVENTTYPE_UPLOAD_START イベントが追加されました**　新しいイベント DBSC_EVENTTYPE_UPLOAD_START が追加されました。イベントは、dbmlsync が Mobile Link サーバへのアップロードの送信を開始すると生成されます。

- **DBSC_EVENTTYPE_UPLOAD_SENT イベントが追加されました**　新しいイベント DBSC_EVENTTYPE_UPLOAD_SENT が追加されました。イベントは、dbmlsync が Mobile Link サーバへのアップロードの送信を完了すると生成されます。

- **DBSC_EVENTTYPE_DOWNLOAD_START イベントが追加されました**　新しいイベント DBSC_EVENTTYPE_DOWNLOAD_START が追加されました。このイベントは、dbmlsync がダウンロードの処理を開始すると生成されます。

Dbmlsync .NET API の変更

- **Connect メソッドに必要な新しい権限**　Connect メソッドで指定された userid に必要な権限が変更されました。 DbmlsyncClient.Connect メソッド [Dbmlsync .NET]『Mobile Link クライアント管理』を参照してください。

- **DBSC_Event に新しい 'val3' 整数フィールドがあります**　DBSC_Event 構造体に新しい 'val3' 整数フィールドがあります。val3 フィールドの値の意味は、イベントのタイプに応じて異なります。 DBSC_Event 構造体 [Dbmlsync .NET]『Mobile Link クライアント管理』を参照してください。

- **DBSC_EVENTTYPE_DOWNLOAD_COMMITTED には 2 つのパラメータがあります**　イベントの DBSC_Event 構造体では、ダウンロード中に挿入または更新されたローの数が val1 に、ダウンロードで削除されたローの数が val2 に含まれます。

- **DBSC_EVENTTYPE_UPLOAD_START イベントが追加されました**　新しいイベント DBSC_EVENTTYPE_UPLOAD_START が追加されました。イベントは、dbmlsync が Mobile Link サーバへのアップロードの送信を開始すると生成されます。
- **DBSC_EVENTTYPE_UPLOAD_SENT イベントが追加されました**　新しいイベント DBSC_EVENTTYPE_UPLOAD_SENT が追加されました。イベントは、dbmlsync が Mobile Link サーバへのアップロードの送信を完了すると生成されます。
- **DBSC_EVENTTYPE_DOWNLOAD_START イベントが追加されました**　新しいイベント DBSC_EVENTTYPE_DOWNLOAD_START が追加されました。このイベントは、dbmlsync がダウンロードの処理を開始すると生成されます。

SQL 文

次に、SQL Anywhere バージョン 16.0 で導入された SQL Anywhere クライアントとリモート Mobile Link データベースでの SQL の強化を示します。

- **ALTER SYNCHRONIZATION SUBSCRIPTION 文の変更**　ALTER SYNCHRONIZATION SUBSCRIPTION 文の ADDRESS 句、TYPE 句、SET SCRIPT VERSION 句で NULL を使用できるようになりました。 「ALTER SYNCHRONIZATION SUBSCRIPTION 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
- **ALTER SYNCHRONIZATION USER 文の変更**　ALTER SYNCHRONIZATION USER 文の ADDRESS 句と TYPE 句で NULL を使用できるようになりました。 「ALTER SYNCHRONIZATION USER 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
- **START SYNCHRONIZATION SCHEMA CHANGE 文の変更**　START SYNCHRONIZATION SCHEMA CHANGE の SET SCRIPT_VERSION 句で NULL を使用できるようになりました。「START SYNCHRONIZATION SCHEMA CHANGE 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

Sybase Central の Mobile Link プラグイン

次に、バージョン 16.0 で導入された Sybase Central の Mobile Link プラグインの新機能と動作の変更を示します。

Sybase Central の Mobile Link プラグインの新機能

次に、バージョン 16.0 で導入された Sybase Central の Mobile Link プラグインの新機能を示します。

- **同期モデルのテストツール**　同期モデルの新しいテストツールの **[展開]** タブを使用すると、同期モデルの展開をテストできます。 「展開前の同期モデルのテスト」『Mobile Link クイックスタート』を参照してください。

  使用例については、「チュートリアル : Mobile Link の概要」『Mobile Link クイックスタート』を参照してください。

- **プロジェクト作成ウィザード の簡略化**　プロジェクトが簡単に作成できるように、**プロジェクト作成ウィザード**が簡略化されています。このウィザードの重要な変更点は次のとおりです。

  - 統合データベースをプロジェクトと関連付ける必要があります。
  - **同期モデル** ページが削除されています。
  - プロジェクトの作成時に同期モデルが自動的に作成されます。
  - 各プロジェクトでリモートスキーマ名ごとに別の型を持つのではなく、1 つのリモートデータベースの型を持つようになりました。

- **簡略化された同期モデル展開ウィザード**　**同期モデル展開ウィザード**はシンプルで使いやすいツールです。同期モデルが展開されている場合、このウィザードにより、以前の展開で使用されていた値を使用して同期モデルを展開することができるようになりました。

- **簡略化されたテーブルマッピングエディタ**　テーブルマッピングエディタが再設計されています。重要な変更点は次のとおりです。

  - **[カラムマッピング]**、**[ダウンロードタイプ]**、**[削除のダウンロード]**、**[サブセットのダウンロード]**、**[ダウンロード削除サブセット]**、**[競合の処理]**、**[ステータス]** タブはすべて、**[詳細]** ウィンドウ枠に表示されます。
  - **[ダウンロードタイプ]**、**[削除のダウンロード]**、**[サブセットのダウンロード]**、**[ダウンロード削除サブセット]**、**[競合の検出]**、**[競合の解決]** オプションのカラムは、デフォルトで **[テーブルマッピング]** ウィンドウ枠に表示されなくなり、表示される場合は読み込み専用です。これらを表示するには、テーブルのヘッダを右クリックします。
  - **[詳細]** ウィンドウ枠の **[競合の処理]** タブは、競合の検出と解決のオプションで使用され、**[競合の解決]** タブの代わりとなるものです。
  - **[詳細]** ウィンドウ枠のタブは、ウィンドウ枠の下部ではなく、上部に表示されるようになりました。

- **新しい[Mobile Link サーバのコマンドラインのプロパティ] ウィンドウ**　新しい **[Mobile Link サーバのコマンドラインのプロパティ]** ウィンドウを使用すると、Mobile Link プロジェクトに取り込む Mobile Link サーバのコマンドラインを保存できます。

- **Mobile Link 16.0 プラグインを使用して追加した同期モデルを削除する機能**　同期モデルとそれに関連するスキーマを削除するには、同期モデル用の新しい **[統合データベースから削除]** コンテキストメニュー項目を使用します。削除できるのは、Sybase Central 向け Mobile Link 16.0 プラグインを使用して追加された同期モデルのみです。

- **統合データベースでの複数の同期モデルのサポート**　同期モデルを展開したときに、前にインストールしたスキーマと同じスキーマが保持されるようになりました。展開中にバージョン間の互換性の問題がある場合、展開の実行前に警告が表示されます。また、同じスクリプトバージョンに更新を再展開すると、新しいバージョンで必要なくなったスキーマは自動的に削除されます。

- **同期モデル用の競合の処理の向上**　競合の検出と解決の処理はすべて upload_update スクリプト内で行われるようになり、フェッチが必要なのは競合が発生した場合のみです。ローごとに行う必要はありません。そのため、更新が多数ある場合にパフォーマンスが大幅に改善されます。

- **同期モデル作成ウィザードの簡略化**　以前は、このウィザードには各種同期オプションを設定するためのオプションのページがありました。これらの同期オプションは **[マッピング]** タブや **[同期モデルのプロパティ]** ウィンドウでも設定できるようになったため、これらのオプションのページは削除されています。

- **LDAP サポートの追加**　MobiLink 16 プラグインには、Mobile Link に対する LDAP 認証をサポートする次のウィザードとプロパティシートが組み込まれています。

  - **LDAP サーバ作成ウィザード**
  - **ユーザ認証ポリシー作成ウィザード**
  - **[LDAP サーバのプロパティ]** ウィンドウ：
  - **認証ポリシーのプロパティ**

- **Mobile Link システムオブジェクトをデータベースから削除可能**　統合データベースから Mobile Link システム設定を削除するための新しいポップアップメニュー項目が利用できます。この項目を選択すると、統合データベースから Mobile Link のすべてのシステムオブジェクトが削除され、Mobile Link プロジェクトからデータベースが削除されます。

- **動的オブジェクトとプロパティの自動再表示**　リモートデータベースの集中管理のため、エージェントと展開済みリモートタスクの自動再表示がサポートされています。

- **ファイルへのリモートタスクのエクスポートが可能**　リモートタスクを、*.mlt* 拡張子の付いたファイルにエクスポートできるようになりました。「リモートタスクのエクスポート」『Mobile Link サーバ管理』を参照してください。

- **Oracle XMLTYPE データ型の制限付きサポート**　Sybase Central の Mobile Link プラグインでは、Oracle XMLTYPE が制限付きでサポートされるようになりました。「同期モデルの制限事項」『Mobile Link クイックスタート』を参照してください。

- **カラムのマッピングの警告**　同期モデルの作成時、または同期モデルへのテーブルマッピングの追加時に、統合データベース内の固定長の文字カラムがリモートテーブル内の可変長の文字カラムにマッピングされた場合に警告が表示されるようになりました。このチェックが行われるのは、統合データベースが Oracle、IBM DB2 LUW、Microsoft SQL Server、または Adaptive Server Enterprise の場合のみです。

Mobile Link の動作の変更

次に、バージョン 16.0 で導入された Mobile Link の動作変更を示します。サポートされているプラットフォームおよびバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

Mobile Link サーバの変更

次に、バージョン 16.0 で導入された Mobile Link サーバの動作変更を示します。

- **end_synchronization スクリプトの完了時に必ず COMMIT を発行**　同期の状態 (同期失敗の場合は 0、成功の場合は 1) を示すため、end_synchronization スクリプトに synchronization_ok

パラメータが渡されます。この変更の前は、synchronization_ok パラメータが 0 の場合は end_synchronization スクリプトがロールバックされていました。現在、end_synchronization スクリプトがすべて正常に実行されると、synchronization_ok パラメータの値に関係なく COMMIT が発行されるようになりました。

古い動作を実行するには、既存の end_synchronization スクリプトの本体を IF 文に組み込みます。次に例を示します。

```
IF {ml s.synchronization_ok} = 1 THEN
    -- existing script body here
END IF;
```

- **Mobile Link と Web ブラウザとのテスト接続**　HTTP または HTTPS の使用時には、Relay Server の有無にかかわらず、Web ブラウザを使用して、Mobile Link サーバが要求を受信しているかどうかを確認できます。Mobile Link サーバは **404 Not Found** エラーで応答します。これには Mobile Link サーバの主要バージョンも記載されています。 「Mobile Link 接続」『Mobile Link サーバ管理』を参照してください。

- **Linux での 32 ビット Mobile Link サーバのサポートを終了**　Linux で 64 ビットバージョンの Mobile Link サーバのみがサポートされるようになりました。

- **一部の統合データベースでの Windows 32 ビット Mobile Link サーバのサポートを終了**　次のタイプの統合データベースでは、Windows 32 ビット Mobile Link サーバはサポートされなくなりました。

  - Sybase Adaptive Server Enterprise
  - SAP HANA
  - SAP Sybase IQ
  - IBM DB2 LUW
  - MySQL

- **同期中に常にカラム名を送信**　Mobile Link では、同期中に常にカラム名が送信されるようになりました。 カラム名の送信 8 ページを参照してください。

Mobile Link の Notifier と Listener の変更

次に、バージョン 16.0 で導入された Mobile Link の Notifier と Listener の動作変更を示します。

- **エラーハンドラへの通知アドレス、件名、内容の制限つき表示**　Notifier のオプションのエラーハンドラを呼び出したときに、通知のアドレス、件名、内容の文字に推奨文字以外のものが含まれていた場合、アスタリスクに置き換えられます。「error_handler イベント」『Mobile Link サーバ起動同期』を参照してください。

- **デフォルトでの SMS リスニングの表示**　dblsn -ns オプションが削除され、SMS リスニングがデフォルトで表示されるようになっています。このオプションは、依然としてソフトウェアでサポートされていますが、指定されていても無視されます。 「Listener の配備のセキュリティ保護」『Mobile Link サーバ起動同期』を参照してください。

  SMS リスニングを有効にするには、-ls オプションを使用します。 「-ls dblsn オプション」『Mobile Link サーバ起動同期』を参照してください。

- **デフォルトでの UDP リスニングの表示**　dblsn -nu オプションが削除され、UDP リスニングがデフォルトで表示されるようになっています。このオプションは、依然としてソフトウェアでサポートされていますが、指定されていても無視されます。「Listener の配備のセキュリティ保護」『Mobile Link サーバ起動同期』を参照してください。

  UDP リスニングを有効にするには、-lu オプションを使用します。「-lu dblsn オプション」『Mobile Link サーバ起動同期』を参照してください。

- **新しい -ls dblsn オプション**　SMS リスニングを有効にするには、dblsn -ls オプションを使用します。「-ls dblsn オプション」『Mobile Link サーバ起動同期』を参照してください。

- **新しい -lu dblsn オプション**　UDP リスニングを有効にするには、dblsn -lu オプションを使用します。「-ls dblsn オプション」『Mobile Link サーバ起動同期』を参照してください。

Mobile Link クライアントの変更

次に、バージョン 16.0 で導入された Mobile Link クライアントの動作変更を示します。

- **サンプル証明書の格納ディレクトリの変更**　TLS 通信用のサンプル X.506 証明書の格納場所が、SQL Anywhere インストールディレクトリの bin32 および bin64 ディレクトリから *C:¥Documents and Settings¥All Users¥Documents¥SQL Anywhere 16¥Samples¥Certificates* ディレクトリに変わっています。マニュアルでは %SQLANYSAMP16%¥Certificates として示されます。

  詳細については、*C:¥Documents and Settings¥All Users¥Documents¥SQL Anywhere 16¥Samples¥Certificates¥readme.txt* を参照してください。

- **フックおよびストアドプロシージャ付きのロールと権限**　フックとストアドプロシージャを使用する場合、新しいロールと権限のセキュリティモデルの結果、次の変更が必要となります。

  ○フックを作成するには MANAGE REPLICATION システム権限が必要です。「SQL Anywhere クライアントのイベントフック」『Mobile Link クライアント管理』を参照してください。

  ○#hook_dict テーブルにアクセスするには、フックの所有者を SELECT ANY TABLE および UPDATE ANY TABLE システム権限のあるユーザとするか、CREATE PROCEDURE 文の SQL SECURITY INVOKER 句を使用してフックを定義してください。

  ○#hook_dict テーブルにアクセスするには、スクリプト化されたアップロードのストアドプロシージャの所有者を SELECT ANY TABLE および UPDATE ANY TABLE システム権限のあるユーザとするか、CREATE PROCEDURE 文の SQL SECURITY INVOKER 句を使用してフックが作成されていることを確認してください。「スクリプト化されたアップロードのストアドプロシージャ」『Mobile Link クライアント管理』を参照してください。

Sybase Central の Mobile Link プラグインの変更

次に、バージョン 16.0 で導入された Sybase Central の Mobile Link プラグインの動作の変更を示します。

- **同期モデルの通知サポートの終了**　[通知] タブを使用してサーバ起動同期を設定するための同期モデルサポートが終了しています。現在、サーバ起動同期は同期プロジェクトのリモートタスクを通してサポートされるようになっています。[通知] タブはなくなり、同期モデル展開ではサーバ起動同期がサポートされなくなりました。Notifier と Listener を通してサーバ起動同期を使用する場合、同期モデルとは別に行うことができます。

- **新しいリモートテーブルの所有者名では所有者 dbo を不使用**　所有者が **dbo** であるテーブルを持つ Microsoft SQL Server または Sybase Adaptive Server Enterprise の統合データベースと新しい SQL Anywhere リモートデータベースを使用して同期モデルを作成する場合、リモートテーブルの所有者は **dbo** となりません。代わりに、新しいリモートテーブルの所有者として統合データベースのユーザが使用されます。

Mobile Link の廃止予定機能とサポート終了機能

注意
廃止予定機能のリストはあくまでも予定であり、完全なものとは限らないため、変更される可能性があります。

- **SQL Anywhere 11.0.1 統合データベースのサポート**　バージョン 11.0.1 以前の SQL Anywhere のサポートは終了しました。

- **Sybase Adaptive Server Enterprise 15.0 統合データベースのサポート**　バージョン 15.0 以前の Sybase Adaptive Server Enterprise のサポートさは終了しました。

- **Microsoft SQL Server 2005 統合データベースのサポート**　バージョン 2005 以前の Microsoft SQL Server のサポートは終了しました。

- **SAP Sybase IQ 15.2 統合データベースのサポート**　バージョン 15.2 以前の SAP Sybase IQ のサポートは終了しました。

- **IBM DB2 LUW 9.5 統合データベースのサポート**　9.5 以前の IBM DB2 LUW のサポートは終了しました。

- **mlstop -f オプションの削除**　mlstop の -f コマンドラインオプションは削除されています。ただし、-f オプションを指定した場合、mlstop は -h オプションとして処理します。「Mobile Link 停止ユーティリティ (mlstop)」『Mobile Link サーバ管理』を参照してください。

- **mlreplay のコールバックの削除**　Mobile Link Replay ユーティリティ (mlreplay) から次のコールバックが削除されています。

  - FreeAllUploadRows
  - GetNumUploadTables
  - GetUploadTable
  - GetNumRows
  - GetRow

- **script_full_path フィールドの削除**　a_sync_db 構造体から script_full_path フィールドが削除されています。a_sync_db 構造体 [データベースツール]『SQL Anywhere サーバ プログラミング』を参照してください。

- **修正済みのローカウントに依存する競合の検出の削除**　アップロードスクリプトによって変更されたローの数を評価する、Mobile Link サーバに依存する競合の検出が削除されています。Mobile Link サーバでは、upload_fetch スクリプトに基づく競合の検出、または競合の検出の不使用がサポートされています。 「競合検出」『Mobile Link サーバ管理』を参照してください。

- **conflicted_inserts および conflicted_deletes アップロード統計の削除**　conflicted_inserts および conflicted_deletes アップロード統計が削除されています。

- **強制的な競合モードの削除**　廃止予定だった強制的な競合モードが Mobile Link から削除されています。

- **ml_add_column システムプロシージャが廃止予定**　ml_add_column システムプロシージャは廃止される予定です。カラム名は、常に送信されます。

- **ml_column システムテーブルが廃止予定**　ml_column システムテーブルが廃止される予定です。カラム名は、常に送信されます。

- **Mobile Link の tls_type および e2ee_type プロトコルオプションでオプションとしての ECC の受け付けを終了**　楕円曲線暗号化 (ECC)7 ページを参照してください。

QAnywhere の新機能

QAnywhere のサポートは、廃止され、削除されました。QAnywhere を管理するには、Sybase Central の以前のバージョンの QAnywhere に含まれる QAnywhere プラグインを使用する必要があります。以前のバージョンの QAnywhere の詳細については、http://dcx.sybase.com/index.html#1201en/sqlanywhere_en12/help_top_index.htm を参照してください。

Relay Server の新機能

次に、バージョン 16.0 で導入された Relay Server の追加機能を示します。サポートされているプラットフォームおよびバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

- **Microsoft IIS 8.0 のサポート**　Relay Server の設定に Microsoft IIS 8.0 を使用できるようになりました。 「Microsoft IIS 7.0、7.5、または 8.0 への Relay　Server の配備」『Relay Server』を参照してください。

- **Microsoft IIS 7.0、7.5、8.0 のクイック設定**　手動の手順の代替手段として、対話型クイック設定機能、*rs-setup.bat* が提供されています。*rs-setup.bat* は *%SQLANY16%/RelayServer/IIS/quicksetup_iis7or8* ディレクトリにあり、次のタスクを実行します。

  1. IIS をインストールし、設定します

  2. Relay Server が正しく動作することを確認するためのデモアプリケーションを作成します。

3. クイックリファレンスガイドを生成します

「Windows Server 2003 上の Microsoft IIS 6.0 への Relay Server コンポーネントの配備」『Relay Server』と「Microsoft IIS 7.0、7.5、または 8.0 への Relay Server の配備」『Relay Server』を参照してください。

- **Apache 2.4 のサポート** Apache 2.4 がサポートされるようになりました。

  「Linux 上の Apache への Relay Server コンポーネントの配備」『Relay Server』を参照してください。

- **Linux 上の Apache のクイック設定** クイック設定機能は次のタスクを実行します。

  1. Relay Server 用の Apache Web サーバを設定します。この手順を行うには、*install-dir/relayserver/quicksetup_apache* ディレクトリの *ap-setup.sh* スクリプトを実行します。
  2. Relay Servert テストサービスを作成して起動します。この手順を行うには、*install-dir/relayserver/quicksetup_apache* ディレクトリの *rs-test-setup.sh* スクリプトを実行します。
  3. クイックリファレンスガイドを生成します

  「Linux 上の Apache への Relay Server コンポーネントの配備」『Relay Server』を参照してください。

- **Outbound Enabler のトラブルシューティングの向上** トラブルシューティングを支援するため、Outbound Enabler のエラーメッセージに記載されるエラーコードと詳細な問題解決の提案が改良されています。

- **サーバ応答のフロー制御の向上** サーバ応答の過剰なバッファリングに関連した問題によって共有メモリリソースが枯渇することがあります。たとえば、クライアント数が多い場合やクライアントの読み込み速度が遅い場合、または HTTP の応答のサイズが大きい場合などに発生します。メモリ使用率を改善するため、Relay Server には新しい設定オプション **max_client_buffer** = *memory size* が用意されています。ここで、各クライアントのメモリバッファサイズの制限を指定できます。デフォルト値は 1 MB です。最大値は 4 GB です。「バックエンドファームセクション」『Relay Server』を参照してください。

- **重複 cookie の更新** クライアントが重複した標準 Cookie を複数の同時接続で共有している場合、タイムアウトエラーが発生することがあります。この問題は Relay Server ログに表示されます。**renew_overlapped_cookie** が on に設定されている場合、Relay Server はこのプロパティが明示的に有効に設定されているファームに対して cookie の重複を検出し、新しいアフィニティのバインドを作成することによって、重複している cookie を更新します。「バックエンドファームセクション」『Relay Server』を参照してください。

- **自動パススルーを使用したプロキシサポートの向上** プロキシは、CONNECT 要求によって非 SSL 終了トンネルを確立する HTTPS パススルーを使用します。この新しい機能を使用すると、プロキシのバッファリングを通して、Relay Server への Outbound Enabler 接続上でのデータストリーミングが可能になります。

- **up_pad_size 設定 (*rs.config* オプションセクション)** RSOE チャネルを通じてサイズの小さい要求がまばらにアップロードされる場合は、遅延を改善するために、この設定により最大

転送単位 (MTU) の埋め込みを挿入します。サーバ応答に関して、Outbound Enabler の遅延時間も改善されています。 「オプションセクション」『Relay Server』を参照してください。

- **Relay Server ログファイルのリモート管理** Relay Server ログファイルは、*rstool.jar* の *AdminChannel* クラスを通じてリモートから管理できます。

  - リモート設定の更新と取得
  - リモート Relay Server のログ抽出
  - ローカル Outbound Enabler のログ抽出
  - Relay Server のログアーカイブ

  「Relay Server ログファイルのリモート管理」『Relay Server』を参照してください。

Relay Server の動作変更

次に、バージョン 16.0 で導入された Relay Server の動作変更を示します。サポートされているプラットフォームおよびバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

- **メモリ消費を制限するための設定オプション** 設定ファイルのオプションセクションの **shared_mem** = *memory size* オプションで、Relay Server がステータスの追跡と拡張のために使用する共有メモリの最大量を指定できるようになりました。「オプションセクション」『Relay Server』を参照してください。

- **Relay Server のステータスページの詳細表示** Relay Server のステータスページが、システム全体のステータスのほか、特定のファームおよびバックエンドサーバのステータスも表示されるように強化されています。また、ステータスページの再表示間隔も設定できます。「Relay Server の [ステータス] ページ」『Relay Server』を参照してください。

- **Relay Server ロギングの変更** Relay Server のロギングで以下がサポートされるようになりました。

  - Relay Server と Outbound Enabler のログが、RFC 822 local differential format (+/- hhmm) を使用してミリ秒間隔でタイムスタンプをレポートするようになりました。

  - Relay Server と Outbound Enabler は、SAP パスポートヘッダを伝送する HTTP 要求を処理するときに、要求処理の冗長レベルを増加して、パスポートのトレースレベル要件を満たすとともに、関連するログの行にサフィックスを追加して、パスポートのキー情報をリストします。 「Relay Server のロギングと SAP パスポート」『Relay Server』を参照してください。

Sybase Central 用の Relay Server プラグインの変更

次に、バージョン 16.0 で導入された Sybase Central の Relay Server プラグインの動作の変更を示します。

- **[Relay Server ファームのプロパティ] 用の新しい [ネットワーク] タブ**　**[Relay Server ファームのプロパティ]** ウィンドウには、新しい **[ネットワーク]** タブがあります。ネットワークの遅延時間を短縮するために、このタブで埋め込みパケットの送信を有効にできます。
- **[バックエンドサーバファームのプロパティ] 用の新しい [詳細] タブ**　**[バックエンドサーバファームのプロパティ]** ウィンドウには、新しい **[詳細]** タブがあります。このタブで以下を設定できます。
  - **クライアント／サーバのアフィニティ**　以前は **[一般]** タブの **[結び付き]** でした。
  - **最大クライアントバッファサイズ**　新しい **max_client_buffer** オプションを設定できます。
  - **重複 cookie の更新**　クライアントデバイスが標準 Cookie を複数の同時重複接続で共有している場合、タイムアウトエラーが発生することがあります。この問題は Relay Server ログに表示されます。**重複 cookie の更新** が on に設定されている場合、Relay Server はこのプロパティが明示的に有効に設定されているファームに対して cookie の重複を検出し、新しいアフィニティのバインドを作成することによって、重複している cookie を更新します。

参照

- 「オプションセクション」『Relay Server』
- 「アフィニティ」『Relay Server』
- 「バックエンドファームセクション」『Relay Server』

Relay Server の廃止予定機能とサポート終了機能

> **注意**
> 廃止予定機能のリストはあくまでも予定であり、完全なものとは限らないため、変更される可能性があります。

- **Windows (Microsoft IIS) 版の 32 ビット Relay Server のサポート終了**　32 ビットバージョンの Windows 版 Relay Server はサポートされなくなったため、インストールされていません。
- **Linux 上の Apache 版 32 ビット Relay Server のサポート終了**　32 ビットバージョンの Windows 版 Relay Server はサポートされなくなったため、インストールされていません。
- **自動起動のサポート終了**　起動 (自動起動) オプションはは、Relay Server の設定ファイルのオプションセクションのプロパティとしてサポートされなくなりました。
- **ECC**　Outbound Enabler tls_type オプションでは、ECC の選択がサポートされなくなりました。tls_type プロトコルオプションはマニュアルから削除されました。

Ultra Light の新機能

次に、バージョン 16.0 で導入された Ultra Light の追加機能を示します。サポートされているプラットフォームおよびバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

一般的な機能

- **ダウンロードの新しい同期統計情報を利用可能**　Ultra Light では、syssyncresult システムテーブルでのダウンロード用の 3 種類の新しい同期統計情報として、ignored_updates、ignored_deletes、truncate_deletes がサポートされるようになりました。

  ignored_updates パラメータは、ダウンロードで受信した重複するローの数を示します。ignored_deletes パラメータは、すでに削除されているローのダウンロードで受信した削除済みのローの数を示します。truncate_deletes パラメータは、ダウンロード時にトランケート操作によって削除されたローの数を示します。このとき、download_delete_cursor スクリプト内で NULL カラム値が返されます。

  次の項を参照してください。

  - 「syssyncresult システムテーブル」『Ultra Light データベース管理とリファレンス』
  - ul_sync_stats_download 構造体 [Ultra Light C および Embedded SQL データタイプ]『Ultra Light C/C++ プログラミング』
  - ul_sync_stats_upload 構造体 [Ultra Light C および Embedded SQL データタイプ]『Ultra Light C/C++ プログラミング』
  - 「download_delete_cursor スクリプト」『Mobile Link サーバ管理』
  - ULSyncProgressData クラス [Ultra Light.NET]『Ultra Light .NET プログラミング』

- **BlackBerry スマートフォンでの ZLIB 圧縮のサポート**　Ultra Light J API で BlackBerry スマートフォンと Mobile Link サーバの同期を取る際、ZLIB データ圧縮がサポートされるようになりました。

  次の項を参照してください。

  - 「Mobile Link クライアントネットワークプロトコルオプション」『Mobile Link クライアント管理』
  - StreamHTTPParms.setZlibCompression メソッド [Ultra Light J]『Ultra Light Java プログラミング』
  - StreamHTTPParms.setZlibDownloadWindowSize メソッド [Ultra Light J]『Ultra Light Java プログラミング』
  - StreamHTTPParms.setZlibUploadWindowSize メソッド [Ultra Light J]『Ultra Light Java プログラミング』
  - StreamHTTPParms.zlibCompressionEnabled メソッド [Ultra Light J]『Ultra Light Java プログラミング』

- **BlackBerry スマートフォンでのエンドツーエンド暗号化 (E2EE) のサポート**　Ultra Light J API で BlackBerry スマートフォンと Mobile Link サーバの同期を取る際、エンドツーエンド暗号化がサポートされるようになりました。

次の項を参照してください。

- 「Mobile Link クライアントネットワークプロトコルオプション」『Mobile Link クライアント管理』
- StreamHTTPParms.getE2eePublicKey メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』
- StreamHTTPParms.setE2eePublicKey メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』
- StreamHTTPParms.E2EE_RSA 変数 [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』
- StreamHTTPParms.getE2eeType メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』
- StreamHTTPParms.setE2eeType メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』

● **BlackBerry スマートフォンでの HTTP 認証のサポート**　Ultra Light J API で BlackBerry スマートフォン用の HTTP 認証がサポートされるようになりました。

パスワードは HTTP ヘッダ内に示されますが、HTTPS プロトコルを使用してヘッダの暗号化とパスワード保護を行うことができます。

次の項を参照してください。

- StreamHTTPParms.setHTTPUserIdAndPassword メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』
- StreamHTTPParms.getHTTPUserId メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』
- StreamHTTPParms.getHTTPPassword メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』

● **BlackBerry スマートフォンでのカスタム HTTP ヘッダのサポート**　Ultra Light J API で BlackBerry スマートフォン用のカスタム HTTP ヘッダがサポートされるようになりました。

次の項を参照してください。

- StreamHTTPParms.addCustomHTTPHeader メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』
- StreamHTTPParms.getCustomHTTPHeaders メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』

● **BlackBerry スマートフォンでのインデックスハッシュのサポート**　Ultra Light Java Edition データベースのインデックスのハッシュサイズを調整するには、WITH MAX HASH SIZE 句を使用します。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

次の項を参照してください。

- 「最適なハッシュサイズ制限」『Ultra Light データベース管理とリファレンス』
- Connection.OPTION_MAX_HASH_SIZE 変数 [Ultra Light J]『Ultra Light Java プログラミング』
- 「CREATE TABLE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
- 「ALTER TABLE 文」『SQL Anywhere サーバ SQL リファレンス』
- 「CREATE INDEX 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
- 「ALTER INDEX 文」『SQL Anywhere サーバ SQL リファレンス』
- 「sysindex システムテーブル」『Ultra Light データベース管理とリファレンス』

- **BlackBerry スマートフォン用のインデックスのパフォーマンスの向上**　挿入、更新、削除をコミットするときに書き込まれるページ数が削減されるのでインデックスのパフォーマンスが向上しました。

  単一の挿入を行う場合に必要なページの読み取り数と書き込み数が少なくなります。ブロックでの挿入を実行した場合の CPU の負荷も軽減されます。

  BlackBerry スマートフォンとの同期を取る場合は、ダウンロードとダウンロード削除カーソルの同期サーバスクリプトをプライマリキーによってソートすることをおすすめします。「ローをダウンロードするスクリプト」『Mobile Link サーバ管理』を参照してください。

- **データベースに複数のアプリケーションが同時に接続した場合における Ultra Light BlackBerry スマートフォンのエラーレポートの向上**　複数のアプリケーションが 1 つの Ultra Light Java Edition データベースに同時にアクセスしようとした場合に、SQLE_FILE_IN_USE エラーがレポートされるようになりました。「Ultra Light および Ultra Light Java Edition データベースの制限事項」『Ultra Light データベース管理とリファレンス』を参照してください。

- **BlackBerry スマートフォン用の 256 ビット AES 暗号化および難読化の組み込み**　Ultra Light J API で ConfigPersistent.enableAesDBEncryption または ConfigPersistent.enableObfuscation メソッドを使用して、BlackBerry アプリケーションの暗号化または難読化を有効にできます。

  一部の Ultra Light Java Edition ユーティリティに -ek オプションが追加され、暗号化データベースへアクセスする際に必要な暗号化キーを指定できるようになりました。

  次の項を参照してください。

  - 「データベースセキュリティ」『Ultra Light データベース管理とリファレンス』
  - 「Ultra Light Java Edition のユーティリティ」『Ultra Light データベース管理とリファレンス』
  - ConfigPersistent.enableAesDBEncryption メソッド [Ultra Light J]『Ultra Light Java プログラミング』
  - ConfigPersistent.enableObfuscation メソッド [Ultra Light J]『Ultra Light Java プログラミング』
  - ConfigPersistent.setEncryptionKey メソッド [Ultra Light J]『Ultra Light Java プログラミング』
  - ConfigPersistent.getEncryptionKey メソッド [Ultra Light J]『Ultra Light Java プログラミング』
  - Connection.changeEncryptionKey メソッド [Ultra Light J]『Ultra Light Java プログラミング』

- **カスタムユーザ認証メッセージを取得可能**　Mobile Link サーバのカスタムユーザ認証同期スクリプトにより、同期の認証結果に関する追加情報が提供されるようになりました。

Ultra Light 管理システムに、この機能をサポートするための新しいシステムテーブル syssyncresult が追加されています。このシステムテーブルは ul_sync_result 構造体を表します。

Ultra Light Java Edition では、SyncResult.getAuthMessage メソッドを使用してカスタムユーザ認証情報を取得できます。

次の項を参照してください。

- 「syssyncresult システムテーブル」『Ultra Light データベース管理とリファレンス』
- ul_sync_result 構造体 [Ultra Light C および Embedded SQL データタイプ]『Ultra Light C/C++ プログラミング』
- SyncResult.getAuthMessage メソッド [Ultra Light J]『Ultra Light Java プログラミング』

- **Ultra Light によるローのキャッシュ**　Ultra Light でキャッシュのレイヤが追加され、最新デバイスの大容量メモリを有効に利用できるようになりました。アクセスの高速化のため、メモリへのデータベースページのキャッシュが行われるほか、割り当てられたメモリの一部が個々のローのキャッシュに割り当てられます。制御接続パラメータのキャッシュのための値は、ページキャッシュやローキャッシュを含めて、キャッシュのために割り当てられたメモリ全体を参照するようになりました。

  次の項を参照してください。

  - 「Ultra Light CACHE_SIZE 接続パラメータ」『Ultra Light データベース管理とリファレンス』
  - 「Ultra Light CACHE_MIN_SIZE 接続パラメータ」『Ultra Light データベース管理とリファレンス』
  - 「UltraLite CACHE_MAX_SIZE 接続パラメータ」『Ultra Light データベース管理とリファレンス』
  - 「Ultra Light cache_allocation オプション」『Ultra Light データベース管理とリファレンス』

プラットフォームとデバイス

- **Pocket PC 2003 での RSA TLS と HTTPS 同期のサポート**　Pocket PC 2003 で RSA TLS と HTTPS 同期がサポートされるようになりました。RSA FIPS 140-2 同期はサポートされていません。

プログラミングインタフェース

Ultra Light C/C++

- **同期ストリームのアップロード統計とダウンロード統計を個別にレポート**　統計レポートを強化するため、ul_sync_stats 構造体が 2 つの構造体、ul_sync_stats_download と ul_sync_stats_upload に分割されています。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

次の項を参照してください。

- ul_sync_stats_download 構造体 [Ultra Light C および Embedded SQL データタイプ]『Ultra Light C/C++ プログラミング』
- ul_sync_stats_upload 構造体 [Ultra Light C および Embedded SQL データタイプ]『Ultra Light C/C++ プログラミング』

● **iOS 用のワイド文字 (UTF-16/UCS-2) のデータストリーミングのサポート**　文字データストリーミングメソッドに iOS 用のワイド文字のサポートが追加されています。

次の項を参照してください。

- ULPreparedStatement.AppendParameterStringChunk メソッド [Ultra Light C++]『Ultra Light C/C++ プログラミング』
- ULResultSet.GetStringChunk メソッド [Ultra Light C++]『Ultra Light C/C++ プログラミング』
- ULResultSet.AppendStringChunk メソッド [Ultra Light C++]『Ultra Light C/C++ プログラミング』

● **アプリケーションでダウンロードの進行状況をモニタ可能**　アプリケーションでダウンロードの進行状況をモニタできるように、2 つの新しいフィールド current_download_row_count と total_download_row_count が ul_sync_status 構造体に追加されています。

次の項を参照してください。

- 「同期ステータス情報」『Ultra Light C/C++ プログラミング』
- ul_sync_status 構造体 [Ultra Light C および Embedded SQL データタイプ]『Ultra Light C/C++ プログラミング』

Ultra Light.NET

● **アプリケーションでダウンロードの進行状況をモニタ可能**　アプリケーションでダウンロードの進行状況をモニタできるように、2 つの新しいプロパティ CurrentDownloadRowCount と TotalDownloadRowCount が ULSyncProgressData クラスに追加されています。

次の項を参照してください。

- ULSyncProgressData.CurrentDownloadRowCount プロパティ [Ultra Light.NET]『Ultra Light .NET プログラミング』
- ULSyncProgressData.TotalDownloadRowCount プロパティ [Ultra Light.NET]『Ultra Light .NET プログラミング』

Ultra Light J

● **同期中のローの統計レポートの機能強化**　同期中に送受信された操作の回数とタイプを含む追加の詳細情報がレポートされます。

最初のテーブルに対して同期が SyncObserver.State.RECEIVING_TABLE 状態に入ると、ダウンロードで読み込まれるローの総数が SyncResult.getTotalDownloadRowCount メソッドにより返されます。

getSentRowCount メソッドは、getSentDeletes メソッド、getSentInserts メソッド、getSentUpdates メソッドに置き換えられています。

getReceivedRowCount メソッドは、getReceivedDeletes メソッド、getReceivedIgnoredDeletes メソッド、getReceivedIgnoredUpdates メソッド、getReceivedInserts メソッド、getReceivedTruncateDeletes メソッド、getReceivedUpdates メソッドに拡張されています。これらの新しいメソッドは、Ultra Light Java Edition のローカルデータベースに適用されたダウンロードの方法を示します。getReceivedRowCount メソッドはダウンロード中の現在のローカウントを示します。

getReceivedRowCount メソッドは long 値を返せるようになりました。

次の項を参照してください。

- SyncResult.getTotalDownloadRowCount メソッド [Ultra Light J]『Ultra Light Java プログラミング』
- SyncResult.getSentDeletes メソッド [Ultra Light J]『Ultra Light Java プログラミング』
- SyncResult.getSentInserts メソッド [Ultra Light J]『Ultra Light Java プログラミング』
- SyncResult.getSentUpdates メソッド [Ultra Light J]『Ultra Light Java プログラミング』
- SyncResult.getReceivedDeletes メソッド [Ultra Light J]『Ultra Light Java プログラミング』
- SyncResult.getReceivedIgnoredDeletes メソッド [Ultra Light J]『Ultra Light Java プログラミング』
- SyncResult.getReceivedIgnoredUpdates メソッド [Ultra Light J]『Ultra Light Java プログラミング』
- SyncResult.getReceivedInserts メソッド [Ultra Light J]『Ultra Light Java プログラミング』
- SyncResult.getReceivedTruncateDeletes メソッド [Ultra Light J]『Ultra Light Java プログラミング』
- SyncResult.getReceivedUpdates メソッド [Ultra Light J]『Ultra Light Java プログラミング』
- SyncResult.getReceivedRowCount メソッド [Ultra Light J]『Ultra Light Java プログラミング』

- **新しいメソッドを使用して Android スマートフォン上で準備文の入力パラメータを管理可能**　準備文のパラメータ数をカウントするための PreparedStatement.getParameterCount メソッドと、パラメータのドメインタイプを取得するための PreparedStatement.getParameterType メソッドが追加されています。

  次の項を参照してください。

  - PreparedStatement.getParameterCount メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』
  - PreparedStatement.getParameterType メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』

- **新しいメソッドを使用して Android スマートフォン上でエラーパラメータを管理可能**　エラーパラメータの数をカウントするための getParameterCount メソッドと、特定のエラーパラメータを返すための getParameter メソッドが追加されています。

次の項を参照してください。

- ULjException.getParameter メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』
- ULjException.getParameterCount メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』
- SQLInfo.getParameter メソッド [Ultra Light J]『Ultra Light Java プログラミング』
- SQLInfo.getParameterCount メソッド [Ultra Light J]『Ultra Light Java プログラミング』

- **新しいメソッドを使用して Android スマートフォン上でアップロードに必要なローの数をカウント可能**　DatabaseInfo.getNumberRowsToUpload(String, int) メソッドが追加されています。これは、Ultra Light C/C++ の ULConnection::CountUploadRows(const char * pubList, ul_u_long threshold) メソッドと同じ機能です。DatabaseInfo.getNumberRowsToUpload メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

- **Ultra Light のデータベース接続状態を Android スマートフォン上でレポート**　Android スマートフォン上で Connection.getState メソッドがサポートされるようになりました。Connection.getState メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

- **Android スマートフォン上で警告メッセージを取得**　Connection.getLastWarning メッセージを使用して SQLInfo オブジェクトを取得します。これにより、前回の SQL 文の実行中に発生した SQL コードと警告メッセージを取得できます。

  次の項を参照してください。

  - Connection.getLastWarning メソッド [Ultra Light J]『Ultra Light Java プログラミング』
  - SQLInfo インタフェース [Android] [Ultra Light J]『Ultra Light Java プログラミング』

- **Android スマートフォン上でのデータベース検証のサポート**　Android スマートフォン上で Ultra Light データベースを検証するには、Connection.validateDatabase メソッドを使用します。新しい ValidateDatabaseProgressData インタフェースと ValidateDatabaseProgressListener インタフェースにより、Android アプリケーションで検証操作の進行状況を追跡できます。

  次の項を参照してください。

  - 「Ultra Light データベースの検証」『Ultra Light データベース管理とリファレンス』
  - Connection.validateDatabase メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』
  - ValidateDatabaseProgressData インタフェース [Android] [Ultra Light J]『Ultra Light Java プログラミング』
  - ValidateDatabaseProgressData.StatusId インタフェース [Android] [Ultra Light J]『Ultra Light Java プログラミング』
  - ValidateDatabaseProgressListener インタフェース [Android] [Ultra Light J]『Ultra Light Java プログラミング』

- **Android スマートフォン上で暗号化キーの変更が可能**　Android スマートフォン上で Ultra Light データベースの暗号化キーを変更するには、Connection.changeEncryptionKey メソッドを使用します。

次の項を参照してください。

- 「データベースセキュリティ」『Ultra Light データベース管理とリファレンス』
- Connection.changeEncryptionKey メソッド [Ultra Light J]『Ultra Light Java プログラミング』

● **Android スマートフォン上での再起動可能なダウンロードのサポート**　Android スマートフォン上の Ultra Light で、通信エラーや SyncObserver オブジェクトによるアボートのために失敗したダウンロードを、部分的に再開できるようになりました。

次の項を参照してください。

- Connection.rollbackPartialDownload メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』
- SyncParms.setAdditionalParms メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』
- SyncParms.getAdditionalParms メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』
- SyncParms.setKeepPartialDownload メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』
- SyncParms.getKeepPartialDownload メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』
- SyncParms.setResumePartialDownload メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』
- SyncParms.getResumePartialDownload メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』
- SyncResult.getPartialDownloadRetained メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』

● **Android スマートフォン上での Ultra Light イベントのサポート**　Android スマートフォン上の Ultra Light で、テーブルの変更、コミット、同期完了のイベントがサポートされるようになりました。この機能をサポートするため、ULjEvent インタフェースが追加されています。

Android アプリケーションは Connection オブジェクトでイベントの登録と待機を行います。Connection は他のスレッドの Connection からトリガされるイベントの通知を取得できます。

次の項を参照してください。

- ULjEvent インタフェース [Android] [Ultra Light J]『Ultra Light Java プログラミング』
- Connection.registerForEvent メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』
- Connection.unregisterForEvent メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』
- Connection.waitForEvent メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』
- Connection.cancelWaitForEvent メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』

● **BlackBerry スマートフォン上でのインデックスの遅延ロードオプションの機能強化**　以前のバージョンでは、ロースコアのフラッシュサイズが 0 より大きい場合、インデックスの遅延ロードをオフにしても有効になりませんでした。遅延ロードがオフになるのは、インデック

スの遅延ロードをオフにし、なおかつロースコアのフラッシュサイズが 0 以下の場合のみでした。

今回より、setRowScoreFlushSize メソッドを使用してロースコアのフラッシュを 0 より大きい値に設定すると、遅延ロードがオンとしてマークされるようになりました。setLazyLoadIndexes メソッドを false に設定すると、ロースコアのフラッシュサイズは 0 に設定されます。

次の項を参照してください。

- ConfigPersistent.setRowScoreFlushSize メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』
- ConfigPersistent.setLazyLoadIndexes メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』

Ultra Light の動作の変更と廃止予定機能

次に、バージョン 16.0 で導入された Ultra Light の動作の変更と廃止予定機能を示します。サポートされているプラットフォームおよびバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

注意
廃止予定機能のリストはあくまでも予定であり、完全なものとは限らないため、変更される可能性があります。

動作の変更

- **ECC サポートの終了**　Ultra Light C/C++ と Ultra Light J API のメソッドのほか、ECC 設定に関連するプロパティが削除されています。楕円曲線暗号化 (ECC)7 ページを参照してください。

- **CustDB サンプル用の Visual Studio 2005 プロジェクトファイルの削除**　CustDB サンプル用の Visual Studio 2005 プロジェクトファイルがソフトウェアから削除されています。

  Visual Studio 2008 プロジェクトファイルが、*%SQLANYSAMP16%¥UltraLite¥CustDB* ディレクトリに格納されています。

  CustDB サンプル用の Visual Studio 2008 プロジェクトファイルをアップグレードするには、このファイルを Visual Studio 2010 で開きます。

- **BlackBerry スマートフォン用の暗号化制御の組み込み**　ConfigPersistent.setEncryption メソッドが削除されています。暗号化制御を行うには、Ultra Light Java Edition データベースを暗号化するためのカスタム暗号化コードを Ultra Light J API に適用する必要がありますが、この機能はサポートされなくなりました。

- **文のコミット時に Ultra Light Java Edition データベースのチェックポイントを自動実行**　次のメンバーが Ultra Light J API から削除されました。

  - ConfigPersistent.getAutoCheckpoint
  - ConfigPersistent.setAutoCheckpoint
  - Connection.checkpoint

- **BlackBerry スマートフォン用の非永続的なインデックスと非シャドウページングストアの削除**　非永続的なインデックスまたはシャドウページングをオフにした状態での Ultra Light Java Edition データベースの作成ができなくなりました。

  次のメンバーが Ultra Light J API から削除されました。

  - ConfigPersistent.hasPersistentIndexes
  - ConfigPersistent.setIndexPersistence
  - ConfigPersistent.setShadowPaging

- **BlackBerry スマートフォン用の終了時書き込み型の永続ストアのサポートが終了**　Ultra Light Java Edition データベースは終了時書き込み型のデータベースとして作成できなくなりました。

  次のメンバーが Ultra Light J API から削除されました。

  - ConfigPersistent.setWriteAtEnd
  - ConfigPersistent.writeAtEnd

- **Ultra Light J の getEncryptionKey メソッドおよび setEncryptionKey メソッドはそれぞれ、getDatabaseKey メソッドおよび setDatabaseKey メソッドと同じ**　Ultra Light J API に、新しい ConfigPersistent.getEncryptionKey メソッドと ConfigPersistent.setEncryptionKey メソッドが追加されています。

  次の項を参照してください。

  - ConfigPersistent.getEncryptionKey メソッド [Ultra Light J]『Ultra Light Java プログラミング』
  - ConfigPersistent.setEncryptionKey メソッド [Ultra Light J]『Ultra Light Java プログラミング』

- **Ultra Light Java Edition ユーティリティの名前の変更**　Ultra Light Java Edition のユーティリティの名前には、Ultra Light のユーティリティの名前と同様に小文字のみが使用されるようになりました。「Ultra Light Java Edition のユーティリティ」『Ultra Light データベース管理とリファレンス』を参照してください。

- **同期中に常にカラム名を送信**　カラム名の送信の制御はできなくなりました。カラム名の送信 8 ページを参照してください。

削除されたプラットフォーム

Ultra Light J API での J2ME スマートフォンのサポートは終了しました。

削除された API

- **Ultra Light for M-Business Anywhere の削除**　Ultra Light for M-Business Anywhere API のサポートは終了しました。

- **実装された Ultra Light C/C++ の削除**　*ulcpp11.cpp* ファイルと *uliface.h* ファイルで以前定義されていた、Ultra Light C/C++ API の古い実装のサポートは終了しました。

管理ツールの新機能

次に、バージョン 16.0 での管理ツールの強化を示します。サポートされているプラットフォームおよびバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

JRE 7 に基づく

Interactive SQL、Sybase Central、SQL Anywhere コンソールユーティリティ (dbconsole)、モニタなどの SQL Anywhere 管理ツールで JRE 7 (Java Runtime Environment 1.7.0) が使用されるようになり、これが必要となっています。

Linux/Solaris 用の 64 ビットのサポート

SQL Anywhere には、64 ビットの Linux および Solaris 用の 64 ビットの管理ツールが組み込まれています。これまでどおり、32 ビットの Linux および Solaris プラットフォーム用に 32 ビットバージョンの管理ツールも利用できますが、Mac OS X 用に 64 ビットバージョンの管理ツールも利用できます。

[接続] ウィンドウの変更

[接続] ウィンドウ用の [接続テスト] ツール　**[接続]** ウィンドウの **[接続テスト]** ツールを使用して、汎用 ODBC データベースと Ultra Light データベース用の情報に基づいて接続が正しくできるかどうかがテストできるようになりました。

ソフトウェア開発チームへのパフォーマンスデータの定期的送信

製品の改良に役立てるために、ソフトウェアには、パフォーマンスデータをソフトウェア開発チームにレポートする機能があります。従来、パフォーマンスデータはクラッシュの発生時に送信されるだけでした。インストール中、パフォーマンスデータをソフトウェア開発チームに送信するオプションがデフォルトで有効になります。また、次の機能も有効にすることができます。

- Interactive SQL で、**[ツール]** » **[オプション]** » **[サポート]** をクリック
- Sybase Central で、**[ヘルプ]** » **[SQL Anywhere]16** » **[更新チェッカーの設定]** をクリック
- サポートユーティリティ (dbsupport) で、-cc 自動送信オプションを指定

OEM ユーザ　デフォルトでは、管理ツールの展開時に、ソフトウェア開発チームにパフォーマンスデータを自動送信する機能は無効になっています。ただし、ユーザは、この機能を管理ツールで有効または無効にすることができます。ユーザのデータを送信するかどうかをユーザが変更できないようにするには、*OEM.ini* ファイルの showPerformanceDataUI オプションを false に設定します。

ユーザがパフォーマンスデータを SAP に送信できるようにするには、サポートユーティリティ (dbsupport) を -cc 自動送信オプションとともに使用する必要があります。

「パフォーマンスデータの定期送信」『SQL Anywhere サーバ データベース管理』を参照してください。

Sybase Central の新機能

次に、バージョン 16.0 で導入された Sybase Central の追加機能を示します。

Sybase Central プラグインの新機能

次に、バージョン 16.0 で導入された Sybase Central のプラグインの追加機能を示します。

SQL Anywhere プラグインの新機能

ER (実体関係) 図のレイアウトのインポートとエクスポート　Sybase Central から ER 図のレイアウトのインポートとエクスポートができます。「ER (実体関連) 図の表示」『SQL Anywhere サーバ データベース管理』を参照してください。

複数のユーザへのログインポリシーの割り当て　複数のユーザにログインポリシーを割り当てることができます。それには、**[ユーザ]** ペインでユーザを選択し、**[ファイル]**、**[ログインポリシーの設定]** を順にクリックします。また、特定のログインポリシーに割り当てられているすべてのユーザを確認することもできます。それには、ログインポリシーの **[プロパティ]** ウィンドウを開いて、**[ユーザ]** タブをクリックします。

マテリアライズドビューでのテキストインデックスのサポート　SQL Anywhere プラグインを使用して、マテリアライズドビューでテキストインデックスを作成し、管理することができます。

LDAP サーバ　LDAP サーバは SQL Anywhere プラグインで管理できます。

ロールとシステム権限のサポート　ロールと権限は SQL Anywhere プラグインで管理できます。

データベースへの証明書の格納　SQL Anywhere プラグインを使用して証明書を追加し、管理することができます。

Mobile Link プラグインの新機能

Mobile Link プラグインの機能の概要は、「Sybase Central の Mobile Link プラグイン」を参照してください。

Sybase Central の動作の変更と廃止予定機能

次に、バージョン 16.0 で導入された Sybase Central の変更を示します。

- **Sybase Central 16.0 へのバージョン変更**　SQL Anywhere 16 には、バージョン 16.0 の Sybase Central が含まれています。
- **Sybase Central の配備に *jh.jar* ファイルと *binXX/scvw[LL]1600.jar* ファイルは不要**　Sybase Central で JavaHelp はサポートされなくなったため、次のファイルはインストールに含まれず、Sybase Central の配備に必要な要素ではなくなっています。*jh.jar* と *binXX/scvw[LL]1600.jar* 。

SQL Anywhere プラグインの動作の変更

セキュリティ機能の例外発生時に secure_feature_key を自動要求　セキュリティ機能を使用する SQL 文を実行すると、SQL の実行前にセキュリティ機能キーを指定するよう要求されます。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

この機能を使用するには、セキュリティ機能キー (-sk) と一緒にサーバを起動する必要があります。

データベース作成ウィザードで作成されたユーティリティコマンドの表示　データベース作成ウィザードの最後には、新しいページが組み込まれています。このページには、dbinit ユーティリティコマンドが表示されます。これは、**[完了]** をクリックしたときに実行される文と同等です。**[完了]** をクリックするとユーティリティコマンドが実行されます。

または、コマンドをクリップボードにコピーし、[キャンセル] をクリックしてウィザードを終了してから、コマンドプロンプトでコマンドを実行します。「ウィザードによって生成された SQL 文とユーティリティコマンドの表示」『SQL Anywhere サーバ データベース管理』を参照してください。

Mobile Link プラグインの動作の変更

Mobile Link プラグインの変更の概要は、「Sybase Central の Mobile Link プラグイン」47 ページを参照してください。

Interactive SQL の新機能

次に、バージョン 16.0 で導入された Interactive SQL の追加機能を示します。

SAP HANA データベースのサポート　SAP HANA データベースに接続するには、Interactive SQL ユーティリティ (dbisql) の **-hana** コマンドラインオプションを使用します。または、Interactive SQL の **[接続]** ウィンドウで、**[データベースタイプを変更]** をクリックし、次に **[SAP HANA]** をクリックします。

SQL エディタでのテキスト補完、テーブルとカラム名の参照、プロシージャ名とソースの参照、インポートとエクスポートもサポートされています。

Interactive SQL 文の中でサポートされていないのは、CONNECT 文、SET CONNECTION 文、START SERVER 文です。

SAP HANA のプランビューアは新しい機能ですが、動作は他のプランビューアと同様です。**空間ビューア**はサポートされていません。

OEN による Interactive SQL の高速ランチャの無効化が可能　Interactive SQL の高速ランチャ機能を無効化して、ユーザが *OEM.ini* ファイルを使用してこの機能を再有効化するのを防止できます。「管理ツールの設定」『SQL Anywhere サーバ プログラミング』を参照してください。

OEM による Interactive SQL での結果の表示方法の指定が可能　OEM.ini ファイルで、次のオプションを使用して結果の表示方法を指定します。使用するオプションは、defaultShowResultsForAllStatements と defaultShowMultipleResultSets です。「管理ツールの設定」『SQL Anywhere サーバ プログラミング』を参照してください。

数値をテキスト結果セット内で右揃えで表示　結果セットは、コンソールアプリケーションとして実行した場合はテキストとして表示され、**[結果をテキストとして表示]** オプションを選択した場合はウィンドウアプリケーションとして表示されます。従来はすべての値が左揃えで表示されていました。現在、数値は右揃えで表示されるようになりました。そのため、値の比較が容易になっています。

エラーウィンドウでの行番号の表示　文の実行でエラーが発生した場合に、開かれるエラーウィンドウにエラーの行番号が表示されるようになりました。

テキスト補完機能でのパラメータリストの表示　プロシージャ、ファンクション、空間メソッドの入力時に、Interactive SQL の **[オプション]** ウィンドウの **[ツールチップを表示]** オプションによりパラメータリストが表示されるようになりました。パラメータリストを閉じるには、[Esc] キーを押します。

テキスト補完機能により閉じカッコ、アポストロフィ、引用符を自動挿入　開きカッコ、角カッコ、大カッコ、アポストロフィ、または引用符を入力すると、テキスト補完機能により閉じの句読表記が挿入されます。閉じの句読表記の挿入を制御するには、**[ツール]**、**[オプション]**、**[エディタ]**、**[テキスト保管]** を順にクリックします。

インポートウィザードによりインポート済みテキストファイルからのカラム名の読み込みが可能　インポートウィザードでは、ファイルの最初の行からカラム名のリストを読み込むことができます。このオプションが使用できるのは、区切りの入ったテキストファイルをインポートする場合のみです。

Interactive SQL の動作の変更と廃止予定機能

次に、バージョン 16.0 で導入された Interactive SQL の変更を示します。

START DATABASE 文の ON 句が廃止予定

START DATABASE 文は、現在のサーバ上でのみデータベースを起動できます。START DATABASE 文の ON 句は廃止される予定です。従来は、Interactive SQL でこの句によって、使用するデータベースサーバを指定できました。「START DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

SQL Anywhere モニタの新機能

次に、バージョン 16.0 で導入された SQL Anywhere モニタの新機能を示します。

- **JRE 7 に基づく**　SQL Anywhere モニタで JRE 7 (Java Runtime Environment 1.7.0) が使用されるようになり、これが必要となっています。
- **警告通知設定**　管理者が **[管理]** ウィンドウの **[電子メール警告通知設定]** オプションを使用して、ユーザとリソースを関連付けることができるようになりました。

SQL Anywhere モニタの動作の変更

次に、バージョン 16.0 で導入された SQL Anywhere モニタの変更点を示します。

モニタから削除するメトリックの変更

使用されるディスク要領を制限するため、モニタからメトリックを定期的に削除する必要があります。デフォルトでは、メンテナンスプランの一部として、作成されてから 5 日を超えたメトリックは毎日削除されます。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

従来はメトリックを時間制限なしで保存し、メトリックが占める容量を制御することができました。現在は、メトリックを保存できる最長期間は 2 週間で、メトリックが占める容量の指定はできなくなっています。また、メトリックデータを新しいバージョンの SQL Anywhere モニタに移行することはできません。移行ユーティリティの -c オプションは削除されています。「モニタのバックアップ」『SQL Anywhere サーバ データベース管理』を参照してください。

データベースユーザ ID およびパスワードクレデンシャルの変更

モニタにデータベースリソースを追加する場合、データベースのユーザ ID およびパスワードクレデンシャルを入力するよう要求されます。バージョン 16 のデータベースを追加する場合、SYS_SAMONITOR_ADMIN_ROLE ロールを持つユーザのクレデンシャルを入力する必要があります。同様に、バージョン 16 のデータベースリソースを修復する場合、同じロールを持つデータベースユーザのクレデンシャルを入力する必要があります。

データベースへの接続を終了するためにユーザ ID およびパスワードクレデンシャルを入力するよう要求された場合、DROP CONNECTION システム権限を持つユーザのクレデンシャルを入力する必要があります。

バージョン 16.0 より前のデータベースのリソースの場合、DBA 権限を持つデータベースユーザのクレデンシャルを引き続き入力することができます。「データベースリソースの追加」『SQL Anywhere サーバ データベース管理』を参照してください。

SQL Anywhere コンソールの新機能

次に、バージョン 16.0 で導入された SQL Anywhere コンソールの新機能を示します。

- **JRE 7 に基づく**　SQL Anywhere コンソールユーティリティ (dbconsole) で JRE 7 (Java Runtime Environment 1.7.0) が使用されるようになり、これが必要となっています。
- **[要求のロギングを可能にする] オプション**　**[オプション]** ウィンドウの **[コンソール]** タブにある **[要求のロギングを可能にする]** オプションで以下のオプションがサポートされるようになりました。

オプション	説明
ブロック	別の接続で接続がブロックされ、ブロック解除されたタイミングを示す詳細のロギングを有効にします。
ホスト変数	ホスト変数値のロギングを有効にします。**SQL** 用にリストされる情報もロギングされます。
プラン	実行プランのロギングを有効にします (短期)。**[プロシージャ]** が有効になっている場合、プロシージャの実行プランも記録されます。

オプション	説明
プロシージャ	プロシージャとユーザ定義関数内から実行される文のロギングを有効にします。
SQL	START DATABASE 文、STOP DATABASE 文、STOP SERVER 文、EXECUTE IMMEDIATE 文、COMMIT 文、ROLLBACK 文、DROP STATEMENT 文のほか、文の準備と実行、オプション設定、PREPARE TO COMMIT 操作、接続と切断、トランザクションの開始、文、カーソルの説明、エラー、カーソルのオープン／クローズ／再開のロギングも有効にします。
トリガ	トリガ内から実行される文のロギングを有効にします。
その他	FETCH や PREFETCH など、**SQL** に含まれない追加の要求タイプのロギングを有効にします。**[その他]** を指定すると、自動的に **[SQL]** が有効になります。**[その他]** を指定すると、要求ログファイルが急速に拡大し、サーバのパフォーマンス低下につながることがあります。
ロギングが開始されたらログファイルを置き換える	ロギングの開始時に、既存の要求ログが同じ名前の新しい (空の) ログに置き換わります。これを指定しない場合、既存の要求ログが開かれ、新しいエントリはファイルの最後に追加されます。

バージョン 12.0.1 の新機能

バージョン 10 以前の SQL Anywhere の新機能と動作変更については、ftp://ftp2.ianywhere.jp/public/tech/dbwnja10.pdf を参照してください。

12.0.1 での追加事項：新機能と動作の変更

「SQL Anywhere 12.0.1 のマニュアルの更新：」このリリースの SQL Anywhere 12.0.1 マニュアルは、2011 年 2 月 15 日の SQL Anywhere 12.0.1 の初回のマニュアルリリース時のバージョンよりも更新されています。このマニュアルには、EBF 3554 (Windows) までの新機能と動作変更が含まれています。特に、SQL Anywhere OnDemand Edition のクラウドの新機能をサポートするために必要な変更も含んでいます。

SQL Anywhere OnDemand Edition に対するサポートの追加

SQL Anywhere OnDemand Edition とは、独立系ソフトウェアベンダー (ISV) がビジネスアプリケーションをクラウドに移行し、ソフトウェアを Software as a Service (SaaS) として提供できるようにしたものです。ISV は、大規模なクラウドアプリケーションを構築、配備および管理できるようになります。

SQL Anywhere OnDemand Edition をサポートするよう、SQL Anywhere にはいくつかの拡張および動作変更が行われています。次に主要な変更点を示します。これ以外の変更の詳細は、http://dcx.sybase.com/cloud100 の SQL Anywhere OnDemand Edition のマニュアルを参照してください。

- **クラウド内のデータベース用の新しいログインモード**　login_mode データベースオプションに、CloudAdmin という新しい値が追加されました。このログインモードは、クラウドの内部で使用します。「login_mode オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Sybase Central と Interactive SQL の [接続] ウィンドウの拡張**　Sybase Central と Interactive SQL の **[接続]** ウィンドウに、クラウド内のテナントデータベースに接続するために、**[クラウドで稼動しているデータベースに接続]** オプションが用意されました。

- **[SQL Anywhere の ODBC 設定] ウィンドウの拡張**　ODBC データソースアドミニストレータの **[SQL Anywhere の ODBC 設定]** ウィンドウに、クラウド内のテナントデータベースに接続するために、**[クラウドで稼動しているデータベースに接続]** オプションが用意されました。

- **NodeType (NODE) 接続パラメータの強化**　NodeType 接続値で、MIRROR 値と READONLY 値を新規にサポートするようになりました。

  MIRROR　テナントのミラーデータベースにアプリケーションをリダイレクトします。

  READONLY　アプリケーションが読み込み専用データベースをコピーしないように制限します。コピーノードまたはミラーコピーのいずれかです。読み込み専用コピーノードがない場合は、READONLY は MIRROR 設定と同様になります。読み込み専用コピーノードのみある場合は、READONLY は COPY 設定と同様になります。

  「NodeType (NODE) 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **新しい NodeType (NODE) 接続パラメータの動作変更**　クラウド内のデータベースに接続する際の NodeType 接続パラメータのデフォルト値は PRIMARY です。これでは、テナントのプライマリデータベースへのリダイレクトが自動的に実行されます。NodeType を DIRECT に設定した場合、リダイレクトはされず、データベースが指定されたホストで実行されている場合のみ接続が成功します。

  「NodeType (NODE) 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **クラウドデータベースへの接続時に必要な DatabaseName 接続パラメータ**　クラウド内のデータベースに接続するときは、DatabaseName 接続パラメータを指定する必要があります。クラウド内のデータベースにはデフォルトのデータベース名はありません。「DatabaseName (DBN) 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **ServerName 接続パラメータのクラウド内での制限**　クラウド内のデータベースに接続する際、NodeType 接続パラメータが DIRECT に設定されているときは、ServerName 接続パラメータしか使用できません。多くの場合、クラウドサーバへの接続時には、ServerName または NodeType まはた DIRECT は指定しないようにする必要があります。「ServerName (Server) 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **SQL 文の強化**　クラウド内のテナントデータベースに対して使用しやすくするために、次の文が強化されています。

  - 「ALTER SYNCHRONIZATION SUBSCRIPTION 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』
  - 「ALTER SYNCHRONIZATION USER 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』
  - 「START SYNCHRONIZATION SCHEMA CHANGE 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』

  クラウド内でデータベースが実行されている場合、一部の SQL Anywhere 文を使用できなかったり、制限がある場合があります。この情報については、それぞれのファンクションとシステムプロシージャの説明の中で示しています。

- **システムプロシージャとファンクションをクラウドのデータベースで実行した場合の結果の制限**　データベースがクラウド内で実行されている場合、SQL Anywhere の一部のファンクションとシステムプロシージャでは、現在のテナントデータベースに関する情報しか返されません。同じクラウドサーバ上で実行される別のテナントデータベースに関する情報は示されません。この情報については、それぞれのファンクションとシステムプロシージャの説明の中で示しています。

- **クラウドで使用できない一部のデータベースオプション、サーバオプション、接続パラメータ**　一部のデータベースオプション、サーバオプション、接続パラメータは、クラウドでは使用できません。詳細については、それぞれのオプションまたはパラメータの説明の中で示しています。

SQL Anywhere のその他の新機能と動作変更

- **IPv6 アドレスの指定**　ポート番号を含む IPv6 アドレスは、角カッコまたはカッコで囲む必要があります。「SQL Anywhere での IPv6 サポート」『SQL Anywhere サーバ データベース管理』を参照してください。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

- **テーブルのロードの強化**　以前のリリースでは、読み込み専用データベースのテンポラリテーブルのテーブルのロード操作は実行できませんでした。この制限は取り除かれました。

  以前、プライマリ、ミラーまたはルートノードサーバ上で実行されるデータベースに対する LOAD TABLE のデフォルトは、WITH FILE NAME LOGGING でした。現在、これらのデータベースタイプに対する LOAD TABLE ログのデフォルトは WITH ROW LOGGING になりました。

- **データベーススキーマの比較と同一化に対するサポートの追加**　Sybase Central を使用して、2 つのデータベースを比較できます。この比較により SQL 文を生成して、2 つのデータベース間の違いを判別できます。その SQL 文を実行し、1 つのデータベースを別のデータベースと同じにできます。「データベーススキーマの比較」『SQL Anywhere サーバ データベース管理』と「もう一方と一致させるためのデータベーススキーマの変換」『SQL Anywhere サーバ データベース管理』を参照してください。

- **DBA ユーザのアカウントのロックアウト**　不正なパスワードの入力が、ログインポリシーで定義した接続試行の失敗回数を超過した場合、DBA ユーザのアカウントは 1 分間ロックされます。データベースから無期限にロックアウトされることはありません。

- ***OEM.ini* ファイルの新しいディレクトリオプション**　OEM.ini ファイルの **[preferences]** セクションの **directory** オプションでは、管理ツールによって使用される、ユーザ専用の設定ファイルを保存するディレクトリを指定します。これらのファイルには管理ツールの設定や履歴に関する情報が格納されます。ディレクトリ名は、デリミタで終了しない完全修飾されたものを指定する必要があります。「管理ツールの設定」『SQL Anywhere サーバ プログラミング』を参照してください。

- **Timeout、SendBufferSize、ReceiveBufferSize TCP プロトコルオプションの新しい制限**　TCP/IP を使用する際、次のプロトコルオプションに上限が追加されました。

  - **ReceiveBufferSize (RCVBUFSZ) プロトコルオプション**　TCP/IP プロトコルスタックで使用されるバッファに指定できるサイズの上限は 1 MB です。

  - **SendBufferSize (SNDBUFSZ) プロトコルオプション**　TCP/IP プロトコルスタックで使用されるバッファに指定できるサイズの上限は 1 MB です。

  - **Timeout (TO) プロトコルオプション**　通信の確立 (TCP/IP) 時の応答の待機に指定できる最大時間は 3600 秒です。

  この上限を超える値を指定すると、接続エラーが返されます。「接続エラー : %1」『エラーメッセージ』を参照してください。

- **データベースミラーリングの強化**　-xa サーバオプションで、次の値がサポートされるようになりました。

  - **DBN=*** の指定は、データベースがサーバを監視サーバとして使用できることを意味します。

  - 認証文字列を省略すると、ミラーサーバが提供する認証文字列を検証しないようにできます。

○認証文字列を 1 つのみ指定した場合、すべてのデータベースでその認証文字列が使用されることを意味します。

「-xa データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

高可用性の読み込み専用スケールアウトシステムでは、コピーノードを監視サーバとして使用できます。この機能を使用するには、データベースミラリングシステム内のすべてのサーバをアップグレードする必要があります。「監視サーバとしてコピーノードを使用する」『SQL Anywhere サーバ データベース管理』を参照してください。

- **文のパフォーマンスの向上**　ストアドプロシージャまたはユーザ定義関数を使用する文のパフォーマンスが向上しました。

- **URL の不明な HTTP メソッドを示す LogFormat (LF) プロトコルオプション**　HTTP 要求メソッドがサポートされていなかったり、URI のフォーマットが不正であるか必要なデータベース名がないために Web 要求が失敗した場合は、HTTP メソッド (@M) と HTTP バージョン (@V) では、文字列 **???** を返し、URI (@U) が **>>>** に続いて指定された要求を返します。「LogFormat (LF) プロトコルオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **新しいプロパティ**　次の接続プロパティが追加されました。

  ○ LastCommitRedoPos

  次のデータベースプロパティが追加されました。

  ○ LastCommitRedoPos
  ○ LastWrittenRedoPos
  ○ LastSyncedRedoPos

  次のデータベースサーバプロパティが追加されました。

  ○ ProcessID

- **ストアドプロシージャまたはユーザ定義関数を使用する文のパフォーマンスの向上**　データベースサーバがユーザ定義プロシージャおよび関数を処理する方法が向上し、パフォーマンスが向上しました。このパフォーマンスの向上は、プロシージャの数とユーザ定義関数の呼び出し数に依存します。

- **ALTER SERVER 文の動作変更**　以前のリリースでは、ALTER SERVER 文のすべての句でオートコミットが実行されていました。現在では、CONNECTION CLOSE 句では、オートコミットは実行されません。「ALTER SERVER 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **OPENSTRING 句の強化**　FROM 句の OPENSTRING 句で [NOT | AUTO] COMPRESSED オプションと ENCRYPTED オプションがサポートされるようになりました。ENCRYPTED オプションを使用すると、ENCRYPTED 句を指定したときに UNLOAD 文によって作成されたファイルを読み込むことができます。「FROM 句」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **証明書作成ユーティリティ (createcert) の強化**　証明書作成ユーティリティに多数のオプションが追加され、簡単に証明書が作成できるようになりました。「証明書作成ユーティリティ (createcert)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **クエリプラン文字列の強化**　プランには、文の既知の値が含まれるようになりました。長いプランには、さらに文のキャッシュプランの情報も含まれるようになりました。

  アプリケーションプロファイリングを使用する場合、長いプランには、クエリの最適化の方法と、部分インデックススキャンで使用される述部に関する情報が追加されています。

- **クエリ最適化**　検索引数可能な IN 述部の推定 (AND 述部に変換できない OR 述部からの部分インデックススキャンに使用可能) に最適化が追加されました。「クエリ処理中に実行される最適化」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **ADO.NET Entity Framework 4.2 に追加されたサポート**　SQL Anywhere で、ADO.NET Entity Framework 4.2 Code First 機能がサポートされるようになりました。「Entity Framework のサポート」『SQL Anywhere サーバ プログラミング』を参照してください。

- **SetupVSPackage .NET アセンブリインストーラの新しい salocation オプション**　SetupVSPackage アプリケーションで salocation オプションを使用し、SQL Anywhere のインストール場所を指定できるようになりました。「.NET クライアントの配備」『SQL Anywhere サーバ プログラミング』を参照してください。

- **JDBC 4.0 ドライバの OSGi バンドルに対するサポートの追加**　SQL Anywhere の JDBC 4.0 ドライバ (sajdbc4.jar) に適切なマニフェスト情報が含まれ、OSGi (Open Services Gateway initiative) バンドルとしてロードできるようになりました。「JDBC サポート」『SQL Anywhere サーバ プログラミング』を参照してください。

SQL Remote の動作の変更

SQL Remote では、リモートデータベースの SQL Remote メッセージのメッセージシステムとして HTTP/HTTPS がサポートされています。データベースを設定するには、SET REMOTE OPTION コマンドを使用します。

次の項を参照してください。

- 「HTTP メッセージシステム」『SQL Remote』
- 「SET REMOTE OPTION 文 [SQL Remote]」『SQL Remote』
- 「チュートリアル：HTTP メッセージシステムを使用したレプリケーションシステムの設定」『SQL Remote』
- 「チュートリアル：メッセージサーバとして HTTP メッセージシステムと統合データベースを使用するレプリケーションシステムの設定」『SQL Remote』
- 「チュートリアル：HTTP メッセージシステムを使用し、Relay Server を経由して統合データベースをメッセージサーバとしたレプリケーションシステムの設定」『SQL Remote』

Ultra Light の動作の変更

- **ALTER TABLE...ADD CONSTRAINT でのユニークな制約の追加時の NULL 入力不可カラムの指定**　ALTER TABLE 文の向上により、新しいユニークな制約を追加するとき、すべての制約カラムを NULL 入力不可にしなければならなくなりました。

「ALTER TABLE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』を参照してください。

- **Android スマートフォンでの Mobile Link ファイル転送の新規サポート**　DatabaseManager.createFileTransfer メソッドを使用して、Android クライアントと Mobile Link サーバ間でファイルを転送できるようになりました。

  次の項を参照してください。

  - DatabaseManager.createFileTransfer メソッド [Ultra Light J]『Ultra Light Java プログラミング』
  - FileTransfer インタフェース [Ultra Light J]『Ultra Light Java プログラミング』
  - FileTransferProgressListener インタフェース [Ultra Light J]『Ultra Light Java プログラミング』
  - FileTransferProgressData インタフェース [Ultra Light J]『Ultra Light Java プログラミング』

- **Android スマートフォンでの ZLIB 圧縮の新規サポート**　HTTPS プロトコルを介して Ultra Light J API で Android スマートフォンと Mobile Link サーバの同期を取る際、ZLIB データ圧縮がサポートされるようになりました。

  Android スマートフォンで ZLIB 圧縮を許可するには、次のメソッドを利用します。

  - StreamHTTPParms.setZlibCompression
  - StreamHTTPParms.setZlibDownloadWindowSize
  - StreamHTTPParms.setZlibUploadWindowSize
  - StreamHTTPParms.zlibCompressionEnabled

  次の項を参照してください。

  - 「Mobile Link クライアントネットワークプロトコルオプション」『Mobile Link クライアント管理』
  - StreamHTTPParms インタフェース [Ultra Light J]『Ultra Light Java プログラミング』

- **Android スマートフォンでの End-to-end 暗号化 (E2EE) の新規サポート**　HTTPS プロトコルを介して Ultra Light J API で Android スマートフォンと Mobile Link サーバの同期を取る際、End-to-end 暗号化がサポートされるようになりました。

  Android スマートフォンで E2EE をサポートするには、次のメソッドを利用します。

  - StreamHTTPParms.getE2eePublicKey
  - StreamHTTPParms.setE2eePublicKey

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

次の項を参照してください。

○「Mobile Link クライアントネットワークプロトコルオプション」『Mobile Link クライアント管理』
○ StreamHTTPParms インタフェース [Ultra Light J]『Ultra Light Java プログラミング』

● **Android スマートフォンでのデフォルトの信用できる証明書ストアの信頼できる証明書への HTTPS プロトコルを介しての同期の新規サポート**　証明書は、次の優先度の規則に従って使用されます。

1. StreamHTTPSParms.setTrustedCertificates メソッドが呼び出された場合、指定したファイルの証明書が使用されます。

2. StreamHTTPSParms.setTrustedCertificates メソッドが呼び出されず、ulinit または ulload ユーティリティによって証明書がデータベースが設定されている場合、それらの証明書が使用されます。

3. StreamHTTPSParms.setTrustedCertificates メソッド、ulinit または ulload ユーティリティのいずれによっても証明書が指定されず、Android を使用している場合は、証明書はオペレーティングシステムの信頼できる証明書ストアから読み込まれます。この証明書ストアは、Web サーバを安全に管理するために HTTPS を介して接続するときに、Web ブラウザで使用されます。

次の項を参照してください。

○ StreamHTTPSParms.setTrustedCertificates メソッド [Ultra Light J]『Ultra Light Java プログラミング』

● **Android スマートフォンでのカーソルの移動方法の新規サポート**　Ultra Light J API で、他の Ultra Light API で利用可能な SQL 結果セットのナビゲーションメソッドがすべてサポートされるようになりました。

Android スマートフォンで、次のナビゲーションメソッドが利用できるようになりました。

○ResultSet.afterLast

○ResultSet.beforeFirst

○ResultSet.first

○ResultSet.getRowCount

○ResultSet.last

○ResultSet.relative

次の項を参照してください。

○「ローデータの検索」『Ultra Light Java プログラミング』
○ ResultSet インタフェース [Ultra Light J]『Ultra Light Java プログラミング』

- **Android スマートフォンでの BlackBerry 専用のいくつかの ResultSet メソッドのサポート**　BlackBerry スマートフォンでサポートされる文字列パラメータを必要とする get メソッドが、Android スマートフォンで利用できるようになりました。

  Android で新規にサポートされるようになったメソッドは、次のとおりです。

  ○ResultSet.getBlobInputStream(String name)

  ○ResultSet.getBoolean(String name)

  ○ResultSet.getClobReader(String name)

  ○ResultSet.getBytes(String name)

  ○ResultSet.getDate(String name)

  ○ResultSet.getDecimalNumber(String name)

  ○ResultSet.getDouble(String name)

  ○ResultSet.getFloat(String name)

  ○ResultSet.getLong(String name)

  ○ResultSet.getSize(String name)

  ○ResultSet.getString(String name)

  ○ResultSet.getUUIDValue(String name)

  ○ResultSet.isNull(String name)

  ResultSet インタフェース [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

- **Android スマートフォンでの ResultSetMetaData メソッドの新規サポート**　Android スマートフォンで、次のメソッドと Blackberry スマートフォンに返される戻り値と非常に似た戻り値がサポートされるようになりました。

  ○ResultSetMetaData.getAliasName(int column_no)

  ○ResultSetMetaData.getDomainName(int column_no)

  ○ResultSetMetaData.getCorrelationName(int column_no)

  ○ResultSetMetaData.getQualifiedName(int column_no)

  ○ResultSetMetaData.getTableColumnName(int column_no)

  ○ResultSetMetaData.getTableName(int column_no)

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

○ResultSetMetaData.getWrittenName(int column_no)

ResultSetMetadata インタフェース [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

- **Android スマートフォンでの DatabaseInfo.getPageSize メソッドの新規サポート**　DatabaseInfo.getPageSize メソッドによって Ultra Light データベースに対しクエリが実行され、ページサイズが取得されるようになりました。

  DatabaseInfo.getPageSize メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

- **Android スマートフォンでの DatabaseInfo.getNumberRowsToUpload メソッドの新規サポート**　DatabaseInfo.getNumberRowsToUpload メソッドで、アップロードを待機しているローの数が返されるようになりました。

  DatabaseInfo.getNumberRowsToUpload メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

- **Android スマートフォンに返される DatabaseInfo.getRelease 値へのビルド番号の新規追加**　DatabaseInfo.getRelease メソッドで、ソフトウェアの完全なリリース番号が返されます。

  DatabaseInfo.getRelease メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

- **Android スマートフォンでの Extra Mobile Link クライアントネットワークプロトコルオプション設定の新規サポート**　Mobile Link クライアントネットワークプロトコルオプションのセミコロンで区切られたリストを指定および取得する StreamHTTPParms.getExtraParameters メソッドと StreamHTTPParms.setExtraParameters メソッドが追加されました。Android スマートフォンで以前にサポートされていなかった他のオプションもサポートするようになりました。

  次の項を参照してください。

  ○ StreamHTTPParms.getExtraParameters メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』
  ○ StreamHTTPParms.setExtraParameters メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』

- **DatabaseManager クラス Ultra Light J API でのエラーレポートの向上**　リリースメソッドの呼び出し後、connect または createDatabase メソッドが呼び出された場合に SQLE_NOT_CONNECTED エラーコードで ULjException error エラーがスローされるようになりました。

  DatabaseManager.release メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

- **Ultra Light J API で利用可能になった再開可能な HTTP**　再開可能な HTTP が有効になっている場合、Ultra Light J では信頼性の低いネットワークでネットワークが頻繁に失敗しないよう、ネットワークの割り込みが許容されます。

次の項を参照してください。

- StreamHTTPParms.setRestartable メソッド [Ultra Light J]『Ultra Light Java プログラミング』
- StreamHTTPParms.isRestartable メソッド [Ultra Light J]『Ultra Light Java プログラミング』

- **Ultra Light J API での NULL 結果セットの処理の向上**　ResultSet.getString メソッドで取得される NULL 値が、空の文字列ではなく NULL として返されるようになりました。

  ResultSet.getString メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

- **ロー制限により BlackBerry スマートフォンで常にインデックスの遅延ロードが有効**　ロー制限が有効な状態でアクセスされる BlackBerry スマートフォン上のデータベースで、常にインデックスが遅延ロードされるようになりました。

  次の項を参照してください。

  - ConfigPersistent.setRowScoreFlushSize メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』
  - ConfigPersistent.setRowScoreMaximum メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』

- **BlackBerry Enterprise Server を介した BlackBerry スマートフォンの同期の安全性の向上**　BlackBerry Enterprise Server (BES) と一部の BlackBerry スマートフォンの HTTPS プロトコルを介しての同期の失敗を避けるには、URL サフィックスに ";EndToEndRequired" の語を追加する必要がある場合があります。

  StreamHTTPParms.setURLSuffix メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

- **Ultra Light Java Edition データベースロードユーティリティとその -z オプションでのデータベースページサイズの調整**　Ultra Light Java Edition のデータベースのページサイズを、新しい -z オプションを使用してバイト単位で指定できるようになりました。

  「Ultra Light Java Edition データベースロードユーティリティ (uljload)」『Ultra Light データベース管理とリファレンス』を参照してください。

- **Ultra Light for M-Business Anywhere での Connection.getNewUUID と UUID.toString メソッドのサポート**　Connection.getNewUUID メソッドと UUID.toString メソッドが再度使用できるようになりました。

- **Ultra Light C++ API でテーブルカラムの修飾された名前が取得可能**　ul_column_name_type 列挙の ul_name_type_qualified 識別子を ULResultSetSchema.GetColumnName メソッドと使用した場合、ResultSet オブジェクトの関連付けられているカラムの修飾された名前を取得できます。

  次の項を参照してください。

  - ul_column_name_type 列挙体 [Ultra Light C++]『Ultra Light C/C++ プログラミング』
  - ULResultSetSchema.GetColumnName メソッド [Ultra Light C++]『Ultra Light C/C++ プログラミング』

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

- **Ultra Light C++ API での Pocket PC 2003 プラットフォームの新規サポート**　Pocket PC 2003 で Ultra Light がサポートされるようになりました。TLS と HTTPS 間の同期での暗号化はサポートされていません。

- **Xcode 4.2 の CustDB と Names サンプルへのサポート**　Xcode 3.2 と 4.2 で CustDB サンプルと Names サンプルがサポートされるようになりました。

Mobile Link の新機能

- **Mobile Link サーバ API への NetworkData クラスの追加**　要求で使用される HTTP ヘッダとクライアント側証明書にアクセスできるよう、Java と .NET スクリプト API に NetworkData クラスが追加されました。

  次の項を参照してください。

  - NetworkData インタフェース [Mobile Link サーバ Java]『Mobile Link サーバ管理』
  - NetworkData インタフェース [Mobile Link サーバ .NET]『Mobile Link サーバ管理』

- **mlsrv12 -x オプションへの collect_network_data プロトコルの新規追加**　各同期時に同期スクリプトでネットワーク情報を読み込むには、collect_network_data プロトコルオプションで mlsrv12 -x オプションを指定して使用します。「-x mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

Mobile Link の動作の変更

- **同期されないカラムに対する更新**　同期されないカラムに対する更新 (同期モデルで定義したタイムスタンプカラムまたは論理削除カラムは除く) で、同期モデルでトリガが使用された場合、タイムスタンプを維持するためにタイムスタンプカラムが更新されなくなりました。

- **テーブルマッピングの方向メニュー**　マッピングの方向を "Not synchronized" に変更した場合、テーブルマッピングの方向メニューは有効のままとなります。「テーブルとカラムのマッピング」『Mobile Link クイックスタート』を参照してください。

- **mlsis と dbmlsync の -qi オプション**　mlsis と dbmlsync の使用法に -qi オプションが追加されました。「-qi dblsn オプション」『Mobile Link サーバ起動同期』と「-qi dbmlsync オプション」『Mobile Link クライアント管理』を参照してください。

- **mlreplay の新しいエラーメッセージ**　Mobile Link リプレイユーティリティに次のエラーメッセージが追加されました。

  - **MLGENREPLAYAPI_FAILED_TO_DETERMINE_CLIENT_TYPE (-5062)**　記録されたプロトコルが dbmlsync クライアントからのものであるか判断できません。

  - **MLREPLAY_INCOMPATIBLE_RECORDED_PROTOCOL_VERSION (-5075)**　記録された %1 は、このバージョンの mlreplay には対応しないプロトコルバージョンです。

  - **MLGENREPLAYAPI_INCOMPATIBLE_RECORDED_PROTOCOL_VERSION (-5076)**　記録された %1 は、このバージョンの Replay API generator には対応しないプロトコルバージョンです。

- **dbmlsync のロギングの強化**　Dbmlsync で、実行しているオペレーティングシステムとマシンのアーキテクチャに関する情報をログに出力できるようになりました。また、実行プログ

ラムがコンパイルされたプラットフォームも出力されます。「SQL Anywhere クライアントのログ」『Mobile Link クライアント管理』を参照してください。

- **SHA2 証明書に対するサポート**　SHA2 を使用して署名された証明書が、Mobile Link サーバと Mobile Link クライアントで新規にサポートされるようになりました。「trusted_certificates」『Mobile Link クライアント管理』と「-x mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

Relay Server の新機能

- **Windows IIS6、IIS7、Apache への Relay Server のクイック配備**　Relay Server には、Windows IIS 6.0、Windows IIS 7.0 または 7.5 と Apache の配備を段階的に処理する一連のユーティリティが用意されています。詳細については、次の項を参照してください。

  - 「Windows Server 2003 上の Microsoft IIS 6.0 への Relay Server コンポーネントの配備」『Relay Server』
  - 「Microsoft IIS 7.0、7.5、または 8.0 への Relay　Server の配備」『Relay Server』
  - 「Linux 上の Apache への Relay Server コンポーネントの配備」『Relay Server』

- **Relay Server での ID 転送のサポート**　SAP Gateway には、信頼できる仲介者を通じて転送される X.509 証明書など、クライアントを認証するいくつかの手段があります。Relay Server では、リモートクライアントからの ID 情報の転送と、HTTP ヘッダを使用したの ID 情報の SAP NetWeaver Gateway または Web Dispatcher への転送が可能です。「バックエンドファームセクションのプロパティ」『Relay Server』を参照してください。

SQL Anywhere の新機能

次に、SQL Anywhere バージョン 12.0.1 の新機能を示します。

サポートされるプラットフォームのリストに加えられた変更については、http://www.sybase.com/detail?id=1061806 を参照してください。

- **空間データのサポートの強化**　SQL Anywhere の空間データサポートに以下の強化が行われました。

  パフォーマンスの改善　次の空間操作のパフォーマンスが大幅に改善されました。

  - シェイプファイルのロード時間
  - 多角形と複数多角形および曲面の空間参照系に属する多角形を含むコレクションのロード
  - 複雑な多角形と複数多角形のロード (たとえば、多数のリングまたは多数の点を含むリングで定義される多角形)
  - 複雑なジオメトリに適用される ST_Union および ST_Intersection などの空間セット操作
  - ST_Contains and ST_Intersects などの空間述部
  - ジオメトリの 1 つが点である空間述部

●インデックス付きの曲面ジオメトリカラムに ST_WithinDistance または ST_Distance を適用する空間述部

Interactive SQL：インポートウィザードでシェイプファイルをサポート　インポートウィザードに ESRI シェイプファイルをインポートするオプションが含まれるようになりました。「インポートウィザードを使用したデータのインポート」『SQL Anywhere サーバ SQL の使用法』を参照してください。

Interactive SQL：INPUT 文の新しい FORMAT SHAPEFILE および SRID 句　INPUT 文で FORMAT SHAPEFIL 句を使用した ESRI シェイプファイルのロードをサポートするようになりました。INPUT 文にシェイプファイルをロードするときに使用する SRID を指定する SRID 句も追加されました。「INPUT 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

新しい st_geometry_load_shapefile システムプロシージャ　st_geometry_load_shapefile システムプロシージャを使用すれば、ファイルの名前、データをロードするために使用する SRID およびデータを作成してロードするテーブルの名前を提供することによって ESRI シェイプファイルをロードできます。テーブルのカラムは、シェイプファイルで指定されたカラム名から取得されます。「st_geometry_load_shapefile システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

注意
この新しいストアドプロシージャにアクセスするには、データベースをアップグレードする必要があります。

ST_CircularString 補間許容値を指定する新しいデータベースオプション　ST_CircularString ジオメトリの補間を制御するために st_geometry_interpolation オプションが追加されました。「st_geometry_interpolation オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

ST_WithinDistanceFilter の強化　曲面の空間参照系のジオメトリで空間述部 ST_WithinDistanceFilter がサポートされるようになりました。ST_Geometry タイプの ST_WithinDistanceFilter メソッド『SQL Anywhere サーバ 空間データサポート』を参照してください。

- **java_class_path オプションの強化**　java_class_path database オプションを使用して、システムクラスローダが Java VM を起動する前にクラスパスに追加するクラスと JAR ファイルを指定できるようになりました。このオプションは、アプリケーションでディレクトリまたは JAR ファイルを提供する場合に便利です。「java_class_path option」『SQL Anywhere サーバ データベース管理』を参照してください。

- **新しい MATVIEW ODBC 接続パラメータ**　MATVIEW 接続パラメータを使用すれば、ODBC 関数 SQLTables を実行するときに実体化ビューのテーブルタイプに返す文字列を指定できます。デフォルトでは、SQLTables 関数によって返される値は VIEW です。「MatView (MATVIEW) 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **新しい -kp データベースサーバオプション**　-kp オプションを使用して、標準形式の *server-name/hostname@REALM* でサーバプリンシパルを指定できるようになりました。-kp オプ

ションは、データベースサーバに対する Kerberos 認証された接続を有効にします。「-kp データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **PartnerState プロパティの強化**　PartnerState プロパティが DB_PROPERTY 関数と共に使用した場合に次のいずれかの値を返すようになりました。

  - **connected**　現在のサーバから指定されたサーバへの接続および指定されたサーバから現在のサーバへの接続があります。
  - **incoming only**　指定されたサーバからこのサーバへの接続があります。
  - **outgoing only**　このサーバから指定されたサーバへの接続があります。
  - **disconnected**　このサーバと指定されたサーバの間に接続がありません。
  - **NULL**　データベースはミラーリングされていません。

  PartnerState データベースプロパティ『SQL Anywhere サーバ データベース管理』を参照してください。

- **MirrorServerState プロパティの強化**　MirrorServerState プロパティが、DB_EXTENDED_PROPERTY 関数と共に使用した場合に次のいずれかの値を返すようになりました。

  - **connected**　現在のサーバから指定されたサーバへの接続および指定されたサーバから現在のサーバへの接続があります。
  - **incoming only**　指定されたサーバからこのサーバへの接続があります。
  - **outgoing only**　このサーバから指定されたサーバへの接続があります。
  - **disconnected**　このサーバと指定されたサーバとの間に接続はありません。
  - **NULL**　データベースはミラーリングされていません。

  MirrorServerState データベースプロパティ『SQL Anywhere サーバ データベース管理』、「DB_PROPERTY 関数 [システム]」『SQL Anywhere サーバ SQL リファレンス』、「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい sp_forward_to_remote_server プロシージャ**　sp_forward_to_remote_server ストアドプロシージャを使用して、アプリケーションがリモートサーバ上で SQL 文を実行し、この文によって生成される結果セットを取得できます。「sp_forward_to_remote_server システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

> **注意**
> このストアドプロシージャにアクセスするには、データベースをアップグレードする必要があります。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

- **新しい sa_user_defined_counter_add システムプロシージャ**　sa_user_defined_counter_add システムプロシージャを使用して、ユーザ定義のプロパティの値を変更できます。sa_user_defined_counter_add システムプロシージャを参照してください。「sa_user_defined_counter_add システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  注意
  このストアドプロシージャにアクセスするには、データベースをアップグレードする必要があります。

- **新しい sa_user_defined_counter_set システムプロシージャ**　sa_user_defined_counter_set システムプロシージャを使用して、ユーザ定義のプロパティの値を設定できます。「sa_user_defined_counter_set システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  注意
  このストアドプロシージャにアクセスするには、データベースをアップグレードする必要があります。

- **SQLANYSAMP12 環境変数**　Unix および Mac OS X インストーラで *sa_config* および *sample_config* スクリプトに SQLANYSAMP12 環境変数が設定されるようになりました。Unix では、*sample_config* スクリプトを使用して、ユーザ別のサンプルのコピーを作成できます。これはマルチユーザインストールに便利です。シングルユーザインストールの場合、*sa_config* スクリプトで *$SQLANY12/samples* に SQLANYSAMP12 を設定します。「SQLANYSAMP16 環境変数」『SQL Anywhere サーバ データベース管理』と「sa_config.sh と sample_config32.sh/sample_config64.sh のソース指定 [UNIX および Mac OS X] 」『SQL Anywhere サーバ データベース管理』を参照してください。

- **データ型 LONG VARCHAR、LONG BINARY、および LONG NVARCHAR の IN パラメータをリモートプロシージャコールで使用可能**　リモートプロシージャコールにデータ型 LONG VARCHAR、LONG BINARY、および LONG NVARCHAR の IN パラメータを含むことができるようになりました。さらに、データ型 VARCHAR、NVARCHAR、および BINARY のパラメータが 255 バイトに制限されなくなりました。「リモートプロシージャの作成 (Sybase Central の場合)」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **CREATE SERVER 文の USING 句に変数を含めることが可能**　CREATE SERVER 文の USING 句に変数を含めることができるようになりました。この機能を使用すれば、ユーザは動的なリモートデータアクセスサーバを作成できます。「CREATE SERVER 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **リモートデータアクセスで SQL Anywhere の ODBC ドライバを直接ロード可能**　Windows と UNIX の両プラットフォームで ODBC ドライバマネージャをバイパスして、リモートデータアクセスで SQL Anywhere ドライバが直接ロードされるようにリモートサーバを定義できます。リモートサーバを指定する際、次の構文を使用します。この後には、接続文字列の残りの部分を続けます。

```
CREATE SERVER remote-server CLASS 'SAODBC' USING 'DRIVER=SQL Anywhere Native;...';
```

'DRIVER=SQL Anywhere Native' を使用しないで定義されている SQL Anywhere リモートサーバが複数ある場合、リモートデータアクセスは他のリモートサーバのドライバマネージャを引き続き使用します。

- **CREATE EXISTING TABLE 文および CREATE PROCEDURE 文の AT 句に変数を含めることが可能**　CREATE EXISTING TABLE および CREATE PROCEDURE 文の AT 句に変数を含めることができるようになりました。この機能を使用すれば、ユーザはプロキシテーブルまたはプロキシプロシージャを複数のリモートテーブルまたはプロシージャにマッピングできます。「CREATE EXISTING TABLE 文」『SQL Anywhere サーバ SQL リファレンス』と「CREATE PROCEDURE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **MERGE 文の RAISERROR 句および RAISERROR 文への強化**　RAISERROR 文を使用することによって、SQL Anywhere データベースサーバはアプリケーションでカスタマイズしたエラーを発生させることができます。データベースサーバはまた組み込みグローバル変数 SQLCODE を提供します。この変数の値を調べて、現在の接続の最後の文の実行中に発生した特定のエラーを識別できます。データベースサーバは固定 -631 エラーメッセージの代わりに SQLCODE のユーザ指定のエラー番号をレポートするようになりました。「MERGE 文」『SQL Anywhere サーバ SQL リファレンス』と「RAISERROR 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **VALIDATE TEXT INDEX 文**　VALIDATE TEXT INDEX 文を使用して、テキストインデックス内の単語の位置情報が正常であることを確認できます。位置情報が壊れている場合、エラーが生成されます。「VALIDATE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **TOP および LIMIT 句の拡張構文**　TOP { ALL | limit } START AT startat および LIMIT limit [OFFSET offset] 句が *limit*, *offset*、および *startat* 引数の簡単な計算式をサポートするようになりました。TOP は ALL limit をサポートし、指定された *startat* 値の後にすべてのローが返されることを示します。(*limit* + *offset*) と (*limit* + *startat* -1) の最大値が 9223372036854775807 = 2^64-1 に増えました。

  次の項を参照してください。

  - 「SELECT、UPDATE、DELETE クエリブロック内のロー制限句」『SQL Anywhere サーバ SQL の使用法』
  - 「SELECT 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DELETE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「UPDATE 文」『SQL Anywhere サーバ SQL リファレンス』

- **SQL Anywhere OLE DB プロバイダが DBTYPE_DBTIMESTAMPOFFSET データ型をサポート**　SQL Anywhere OLE DB プロバイダが DBTYPE_DBTIMESTAMPOFFSET データ型をサポートするようになりました。DBTYPE_DBTIMESTAMPOFFSET (146) は TIMESTAMP WITH TIME ZONE (または DATETIMEOFFSET) データ型をサポートする OLE DB 型です。このデータ型をサポートすることにより、SQL Anywhere データベースと他のデータベース管理システム (SQL Anywhere を含む) 間でのデータテーブルの転送が容易になります。

- **Web サービスで HTTP リダイレクト操作の改善された制御をサポート**　CREATE PROCEDURE および CREATE FUNCTION 文の新しい SET REDIR 句は許容される最大数のリダイレクションの制御を可能にし、自動的にリダイレクトする HTTP ステータスを指定します。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

303 ステータスを受信する POST HTTP メソッドを指定する Web サービスプロシージャが GET HTTP メソッドを使用してリダイレクト要求を発行します。

HTTP クライアントプロシージャで相対パスのリダイレクションを処理するようになりました。以前は、サーバで絶対 URL が提供された場合にのみリダイレクションを実行できました。

リダイレクトを受け取る GET メソッドのみが、リダイレクト応答のロケーションヘッダ URL によって指定されたクエリコンポーネントを提供します。リダイレクトを受け取る POST メソッドがリダイレクト応答のロケーションヘッダによって指定されたパスとクエリコンポーネントを含む要求の URL を発行します。この要求の本文には、プロシージャによって生成されたクエリコンポーネントが含まれます。

URL 句内および (自動的に生成されて) プロシージャに渡されたパラメータからの両方でクエリパラメータを指定できるようになりました。これは、GET HTTP メソッドを指定するプロシージャにのみ該当します。

- **Windows Mobile でセキュア Web サービスをサポート**　Windows Mobile で HTTPS and HTTPS_FIPS を使用する Web サービスプロシージャがサポートされるようになりました。

- **SQL Anywhere JDBC ドライバが PreparedStatement.setClob() をサポート**　SQL Anywhere JDBC ドライバが PreparedStatement.setClob() をサポートするようになりました。

  以前のリリースでは、SQL Anywhere JDBC ドライバは PreparedStatement.setBlob、ResultSet.getBlob、ResultSet.getClob をサポートしていました。SQL Anywhere JDBC ドライバは 3 つの PreparedStatement.setClob メソッドのうちの 2 つをサポートするようになりました。これらのメソッドについては、以下に説明します。

  ```
  PreparedStatement.setClob( int parameterIndex, Clob x )
  PreparedStatement.setClob( int parameterIndex, Reader reader, long length )
  ```

  以下のバリエーションは、SQL Anywhere JDBC ドライバではサポートされていません。

  ```
  PreparedStatement.setClob( int parameterIndex, Reader reader )
  ```

  PreparedStatement.setClob(int parameterIndex, Clob x) オーバロードを使用する場合、ユーザ提供の Clob 実装は Clob.length および Clob.getCharacterStream メソッドのみサポートする必要があります。また、大きい文字列および大きい文字ストリームの場合、SQL Anywhere JDBC ドライバ内の新しい PreparedStatement.setClob メソッドが PreparedStatement.setString および PreparedStatement.setCharacterStream メソッドよりもパフォーマンスとメモリ使用率の両方で推奨されます。「JDBC サポート」『SQL Anywhere サーバ プログラミング』を参照してください。

- **SQL Anywhere Java VM ClassLoader がシャットダウンフックをサポート**　データベースサポートでの Java の提供に使用される SQL Anywhere Java VM ClassLoader を使用して、アプリケーションでシャットダウンフックをインストールできます。「Java VM でのシャットダウンフック」『SQL Anywhere サーバ プログラミング』を参照してください。

- **SQL Anywhere .NET SetupVSPackage インストーラ**　SetupVSPackage アプリケーションで Global Assembly Cache や Windows Microsoft.NET *machine.config* ファイルの更新などのいくつかのインストーラ機能を実行できるようになりました。SQL Server 2008 以降がシステムに

インストールされている場合、SetupVSPackage で *MSSqlToSA.xml* および *SAToMSSql10.xml* と呼ばれる 2 つのマッピングファイルを SQL Server の *DTS¥MappingFiles* フォルダにインストールすることもできます。「.NET クライアントの配備」『SQL Anywhere サーバ プログラミング』を参照してください。

パフォーマンスの強化

次に、パフォーマンスの向上以外にユーザに見える変更がないバージョン 12.0.1 でのパフォーマンスの強化を示します。

- **即時テキストインデックス**　即時テキストインデックスメンテナンスのパフォーマンスが強化されました。目に見えるパフォーマンスの強化は負荷および文書とインデックスの内容によって変わります。この強化によって最も大きな恩恵をこうむるのは、長い文書にインデックスを付けるテキストインデックスです。

- **物理的な削除をクリーナーに遅延処理**　トランザクションのパフォーマンスを改善するために、ローはコミット時間に物理的に削除されません。代わりに、それらのローは論理的に削除されたとマークされ、実際の削除は遅れて、クリーナーで行われます。データベースが過度に大きくなるのを防ぐために、クリーニングスケジュールがより積極的に行われるようになりました。「sa_clean_database システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **バイパス可能なクエリで FROM 句のテーブルに計算されたカラムを定義可能**　オプティマイザをバイパスするために検討しなければならないクエリの条件の 1 つは、FROM 句で参照されるベーステーブルに計算されたカラムが定義されていないことでした。SA 12.0.1 からこの制限が取り除かれました。計算されたカラムを使用する共通の部分式をなくすプロセスが SQL Anywhere オプティマイザを使用するクエリと同様のバイパスされたクエリで実行されます。

- **ベーステーブルに計算されたカラムが含まれる場合にバイパスクエリが可能**　以前は、FROM 句で参照されるベーステーブルに計算されたカラムがない場合、クエリをバイパスクエリとして修飾できませんでした。この制限が取り除かれました。「クエリ処理のフェーズをスキップするための条件」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **オプティマイザでのインデックス使用の強化**　SQL Anywhere のインデックス機能に部分インデックススキャンや同じインデックスを使用する複数インデックススキャンを使用するクエリを最適化する改善などのいくつかの強化が行われました。

- **IS NOT NULL 探索条件**　オプティマイザで IS NOT NULL 探索条件を評価するインデックスを使用できるようになりました。「クエリの述部」『SQL Anywhere サーバ SQL の使用法』を参照してください。

プロパティとパフォーマンスモニタの統計値

次に、SQL Anywhere バージョン 12.0.1 でのプロパティとパフォーマンスモニタの統計の強化を示します。

- **接続プロパティ**　このリリースには、次の接続プロパティが追加されています。

  - ConnectedTime
  - UserDefinedCounterRate01
  - UserDefinedCounterRate02
  - UserDefinedCounterRate03
  - UserDefinedCounterRate04
  - UserDefinedCounterRate05
  - UserDefinedCounterRaw01
  - UserDefinedCounterRaw02
  - UserDefinedCounterRaw03
  - UserDefinedCounterRaw04
  - UserDefinedCounterRaw05

  これらのプロパティの詳細については、「接続プロパティの一覧」『SQL Anywhere サーバ データベース管理』を参照してください。

- **データベースプロパティ**　このリリースには、次のデータベースプロパティが追加されています。

  - ApproximateCPUTime
  - BytesReceived
  - BytesReceivedUncomp
  - BytesSent
  - BytesSentUncomp
  - CarverHeapPages
  - ClientStmtCacheHits
  - ClientStmtCacheMisses
  - ConnectedTime
  - HeapsCarver
  - HeapsLocked
  - HeapsQuery
  - HeapsRelocatable
  - PacketsReceived
  - PacketsReceivedUncomp
  - PacketsSent
  - PacketsSentUncomp
  - PrepStmt
  - QueryHeapPages
  - QueryMemActiveCurr
  - QueryMemGrantFailed
  - QueryMemGrantGranted
  - QueryMemGrantRequested
  - QueryMemGrantWaited
  - QueryMemGrantWaiting
  - ReqCountActive
  - ReqCountBlockContention
  - ReqCountBlockIO
  - ReqCountBlockLock
  - ReqCountUnscheduled
  - ReqTimeActive
  - ReqTimeBlockContention
  - ReqTimeBlockIO
  - ReqTimeBlockLock
  - ReqTimeUnscheduled
  - RequestsReceived
  - Rlbk
  - UserDefinedCounterRate01
  - UserDefinedCounterRate02
  - UserDefinedCounterRate03
  - UserDefinedCounterRate04
  - UserDefinedCounterRate05
  - UserDefinedCounterRaw01
  - UserDefinedCounterRaw02
  - UserDefinedCounterRaw03

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

- UserDefinedCounterRaw04
- UserDefinedCounterRaw05

これらのプロパティの詳細については、「データベースプロパティの一覧」『SQL Anywhere サーバ データベース管理』を参照してください。

- **データベースサーバプロパティ**　このリリースには、次のデータベースサーバプロパティが追加されています。

  - ApproximateCPUTime
  - Commit
  - ConnectedTime
  - Cursor
  - CursorOpen
  - PrepStmt
  - Rlbk
  - UserDefinedCounterRate01
  - UserDefinedCounterRate02
  - UserDefinedCounterRate03
  - UserDefinedCounterRate04
  - UserDefinedCounterRate05
  - UserDefinedCounterRaw01
  - UserDefinedCounterRaw02
  - UserDefinedCounterRaw03
  - UserDefinedCounterRaw04
  - UserDefinedCounterRaw05

  これらのプロパティの詳細については、「データベースサーバプロパティの一覧」『SQL Anywhere サーバ データベース管理』を参照してください。

- **パフォーマンスモニタの統計値**　このリリースには、次のパフォーマンスモニタの統計値が追加されています。

  - ユーザ定義変化率：カウンタ 1
  - ユーザ定義変化率：カウンタ 2
  - ユーザ定義変化率：カウンタ 3
  - ユーザ定義変化率：カウンタ 4
  - ユーザ定義変化率：カウンタ 5
  - ユーザ定義未加工データ：カウンタ 1
  - ユーザ定義未加工データ：カウンタ 2
  - ユーザ定義未加工データ：カウンタ 3
  - ユーザ定義未加工データ：カウンタ 4
  - ユーザ定義未加工データ：カウンタ 5

SQL Anywhere の動作の変更

次に、バージョン 12.0.1 で導入された SQL Anywhere の動作の変更を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

- **LOAD および UNLOAD TABLE 文で使用されるエスケープ文字**　以前のリリースでは、これらの文のエスケープ文字は 1 バイトより長くすることはできませんでした。現在は、エスケープ文字に指定する文字列は 255 バイトを超えないことが推奨されていますが、文字列は 1 文字以下にすることが推奨されています。

- **最大キャッシュサイズに応じて最小および初期キャッシュサイズが増える**　初期キャッシュサイズまたは最小キャッシュサイズを最大キャッシュサイズの 8 分の 1 より小さい値に設定すると、最大キャッシュサイズに関連して、初期および最小キャッシュサイズが自動的に増えます。その結果、最小キャッシュサイズおよび初期キャッシュサイズが以前のバージョンよりも大きくなる場合がよくあります。

- **reserved_keywords オプションの範囲の変更**　以前は、reserved_keywords オプションを個別のユーザに指定したり、設定が一時的な範囲であることを指定できました。しかし、特定の DDL 文を実行するときに一時的な設定やユーザレベルの設定が対応する PUBLIC 設定と異なる場合、データベースの受信または再構築時に問題が発生する場合があります。

  reserved_keywords オプションへの次の動作変更は、新しい 12.0.1 バージョンのデータベースに適用されます。

  ○一時的な設定および非 PUBLIC 設定は許可され**ません**。

  reserved_keywords オプションへの次の動作変更は、既存のバージョン 12 のデータベースに適用されます。

  ○既存の非 PUBLIC 設定は実行中に無視されます。

  ○既存の非 PUBLIC 設定の削除が許可されます。

  ○Dbunload は、既存の非 PUBLIC 設定を無視します。

- **マルチプログラミングレベルのオプション**　以前は、ネットワークデータベースサーバ (dbsrv12) に厳密に適用されるマルチプログラミングレベルのデータベースサーバオプションはパーソナルデータベースサーバ (dbeng12) によって、エラーを生成することなく無視されました。現在は、以下のデータベースサーバオプションは、パーソナルデータベースサーバで使用される場合、警告を生成します。

  ○「-gna」　dbsrv12 サーバオプション
  ○「-gnh」　dbsrv12 サーバオプション
  ○「-gnl」　dbsrv12 サーバオプション
  ○「-gns」　dbsrv12 サーバオプション

- **-zt データベースサーバオプションに対する変更**　-zt データベースサーバオプションを設定することによって、ReqCountBlockIO プロパティと ReqTimeBlockIO プロパティが利用可能にな

ることはなくなりました。これらの 2 つのプロパティは、RequestTiming プロパティがオンにされているかどうかにかかわらず現在では常に利用できます。

- **SQLTables 関数の ODBC 接続パラメータへの変更**　ODBC 関数 SQLTables を実行すると、TABLE_TYPE カラムはデフォルトで実体化ビューに対して値 VIEW を返すようになりました。以前は、この関数は値 MATERIALIZED VIEW を返しました。MATVIEW 接続パラメータを使用してこのパラメータを変更できます。「MatView (MATVIEW) 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **CONVERT 関数**　フォーマット化された文字列の解析が強化されて、文字列の時間部分がフォーマット hh:nn:ss.sssssssAA に一致すると受け入れられるようになりました。時間文字列は時間の桁を指定する必要がありますが、その他の時間部分はすべてオプションです。AM/PM インジケータは時間部分が省略されるかどうかにかかわらず、常に受け入れられます。これにより、秒の後のマイクロ秒を表す 6 桁までが許容されます。

  この変更は文字列の TIME および TIMESTAMP への変換にも影響を与えます。「CONVERT 関数 [データ型変換]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  以前は、*format-style* を指定して CONVERT 関数を使用して文字列を時間に変換すると、SQL Anywhere 10 以降では以前のバージョンで許可された変換が拒否されました。たとえば、次の文はバージョン 9 では受け入れられましたが、バージョン 10 以降では拒否されました。

  ```
  SELECT CONVERT( TIME, '11:45am', 14 ) tm_conv
  ```

  文字列を TIME に変更する動作が SQL Anywhere のバージョン 9 からバージョン 10 以降で変更され、バージョン 10 以降では、文字列を TIMESTAMP に変換するために使用される規則と同じ規則が適用されます。たとえば、文字列 11:45am はフォーマットスタイル 14 (hh:nn:ss.sss) とは正確には一致しません。なぜなら、この文字列にはスタイルに存在しない am インジケータが含まれるからです。

  次の文はバージョン 9 では受け入れられましたが、バージョン 10 以降では拒否されました。文字列がスタイルフォーマット 101 (mm/dd/yyyy) と一致しないからです。

  ```
  SELECT CONVERT( TIME, '1991-02-03 11:45', 101 )
  ```

- **サーバライセンスユーティリティ (dblic) によるライセンスされたユーザ名と会社名の変更**　ユーザカウントとライセンスタイプを指定しなくてもライセンスされたユーザと会社を変更できるようになりました。「サーバライセンス取得ユーティリティ (dblic)」『SQL Anywhere サーバ データベース管理』を参照してください。

SQL Anywhere の廃止予定機能とサポート終了機能

> **注意**
> 廃止予定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、変更される可能性があります。

バージョン 12.0.1 で廃止された機能のリストを次に示します。

- **-kr オプション**　以前の Kerberos サーバプリンシパルは、*server-name@default-realm* の形式にする必要があります。

  領域を含めた Kerberos サーバプリンシパルは -kp サーバオプションで指定できます。-kp で指定されるサーバプリンシパルは、データベースサーバを実行するコンピュータで Kerberos keytab ファイルに抽出されている必要があります。-kp オプションが指定されている場合、-kr オプションは指定できません。「-kp データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **SQL Anywhere JDBC 3.0 ドライバ**　SQL Anywhere JDBC 3.0 ドライバはこのソフトウェアリリースで廃止されています。*sajdbc.jar* を使用するアプリケーションを *sajdbc4.jar* に切り替えることをお勧めします。「JDBC サポート」『SQL Anywhere サーバ プログラミング』を参照してください。

Mobile Link の新機能

次に、バージョン 12.0.1 で導入された Mobile Link の追加機能を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

統合データベース

次に、SQL Anywhere バージョン 12.0.1 での Mobile Link 用の統合データベースサポートの強化を示します。

- **SAP Sybase IQ のサポート**　Mobile Link サーバは SAP Sybase IQ を統合データベースとしてサポートするようになりました。「SAP Sybase IQ の統合データベース」『Mobile Link サーバ管理』を参照してください。 推奨ドライバの詳細については、http://www.sybase.com/detail?id=1011880 を参照してください。

- **ASE 15.5 のサポートを追加**　Windows および Linux 上の Mobile Link サーバで ASE 15.5 サーバ上で動作する統合データベースをサポートするようになりました。推奨ドライバの詳細については、http://www.sybase.com/detail?id=1011880 を参照してください。

Mobile Link サーバ

次に、SQL Anywhere バージョン 12.0.1 での Mobile Link サーバの強化を示します。

- **データベースワーカスレッドの自動調整**　Mobile Link サーバはスループットを最大化するために、データベースワーカスレッドの数を自動的に調整できるようになりました。「-wm mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **新しい名前付きシステムパラメータ**　次の名前付きシステムパラメータが Mobile Link に追加されて、スクリプトで新しいリモート ID またはユーザ名を識別できるようになりました。

○ **new_remote_id**　この新しいシステムパラメータは新しいリモート ID を示しており、authenticate_user and begin_synchronization 接続スクリプト、および begin_synchronization テーブルスクリプトで使用できます。

○ **new_username**　この新しいシステムパラメータは新しいユーザ名を示しており、authenticate_user and begin_synchronization 接続スクリプト、および begin_synchronization テーブルスクリプトで使用できます。

詳細については、次の項を参照してください。

○「authenticate_user 接続イベント」『Mobile Link サーバ管理』
○「begin_synchronization 接続イベント」『Mobile Link サーバ管理』
○「begin_synchronization テーブルイベント」『Mobile Link サーバ管理』

● **ASE 15.5 をサポートするために bigtime および bigdatetime データ型を追加**　これらのデータ型は、SQL Anywhere および Ultra Light の TIME および TIMESTAMP にそれぞれマッピングされます。「Adaptive Server Enterprise データのマッピング」『Mobile Link サーバ管理』を参照してください。

● **認証パラメータの最大長が拡張された**　各 Mobile Link 認証パラメータは 4000 バイトまでの長さで使用できます。以前の制限は 128 バイトでした。「認証パラメータ」『Mobile Link サーバ管理』を参照してください。

mslrv12 の新機能

● **データベースワーカスレッドの自動調整を可能にする新しい -wm オプション**　-wm mslrv12 オプションが追加されました。-wm オプションを使用してデータベースワーカスレッドの最大数を設定し、Mobile Link サーバで最適なデータベースワーカスレッド数を自動的に選択できるようにします。「-wm mslrv16 オプション」『Mobile Link サーバ管理』を参照してください。

● **-cmax、-cmin および -cinit オプションがサイズパラメータのパーセンテージの使用をサポート**　-cmax、-cmin、および -cinit サーバオプションのサイズパラメータをパーセンテージとして指定できるようになりました。「-cmax mslrv16 オプション」『Mobile Link サーバ管理』、「-cmin mslrv16 オプション」『Mobile Link サーバ管理』、「-cinit mslrv16 オプション」『Mobile Link サーバ管理』を参照してください。

● **-x mlsrv12 オプションの新しい log_bad_request パラメータ**　-x mlsrv12 オプションで HTTP、HTTPS、および OE の log_bad_request オプションをサポートするようになりました。yes に設定すると、Mobile Link サーバは不完全または予期しない HTTP 要求を受信した場合にエラーを出力します。「-x mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

● **エンドツーエンド暗号化で DER でコード化されたキーをサポート**　-x mlsrv12 オプションの e2ee_private_key オプションで PEM と DER の両方でコード化されたキーをサポートするようになりました。

新しい Mobile Link によるリモートデータベース集中管理機能

SQL Anywhere バージョン 12.0.1 で導入された Mobile Link によるリモートデータベースの集中管理機能の強化の一覧を以下に示します。

- **タスク条件の新しい変数**　リモートデータベースの集中管理用 Mobile Link エージェントでタスク条件の 2 つの新しい変数をサポートするようになりました。{is_online} と {network_conn_name} の 2 つです。変数 {is_online} は、クライアントデバイスがエージェントの Mobile Link サーバにアクセスできるネットワークに接続されているかどうかを確認するために使用できます。変数 {network_conn_name} は、Mobile Link サーバと通信を行うエージェントによって使用されるネットワーク接続の名前を評価します。「パラメータの変数」『Mobile Link サーバ管理』を参照してください。

- **タスク実行のランダムな遅延の新規サポート**　リモートタスクに秒単位で設定されるランダムな遅延間隔を設定できるようになりました。各エージェントはこの間隔を使用して、各タスクの実行を遅らせるランダムな秒数を生成します。「リモートタスク」『Mobile Link サーバ管理』を参照してください。

- **リモートデータベースの管理用 Mobile Link エージェントを対話形式で設定できるようになりました。**　Mobile Link エージェントをリモートデバイスで一連の設定ウィンドウを使用して、対話形式で設定できるようになりました。

サーバユーティリティ

次に、SQL Anywhere バージョン 12.0.1 での Mobile Link サーバユーティリティの強化を示します。

Mobile Link リプレイユーティリティ (mlreplay)

- **mlreplay ユーティリティは複数のシミュレートされたクライアントを実行し、コマンドラインのみを使用して記録されたプロトコルをリプレイできます。**　-n mlreplay オプションを使用して、多数のシミュレートされたクライアントを管理します。「Mobile Link Replay ユーティリティ (mlreplay)」『Mobile Link サーバ管理』を参照してください。

- **新しい -rep mlreplay オプションの追加**　-rep mlreplay オプションで、シミュレートしたクライアントを使用して、記録されたプロトコルをリプレイする回数を指定します。「Mobile Link Replay ユーティリティ (mlreplay)」『Mobile Link サーバ管理』を参照してください。

- **新しい -rnt mlreplay オプションの追加**　-rnt mlreplay オプションは、シミュレートしたクライアントに指定したランタイムに達するまでプロトコルリプレイの新たな繰り返しを開始するように指示します。「Mobile Link Replay ユーティリティ (mlreplay)」『Mobile Link サーバ管理』を参照してください。

- **設定ファイルをサポートする新しい @data mlreplay および mlgenreplayapi オプションの追加**　*@data* mlreplay オプションを使用すれば、mlreplay と mlgenreplayapi を設定ファイルに格納されたコマンドラインの指定で実行できます。「Mobile Link Replay ユーティリティ (mlreplay)」『Mobile Link サーバ管理』、「生成された Mobile Link リプレイ API ユーティリティ (mlgenreplayapi)」『Mobile Link サーバ管理』、「設定ファイル」『SQL Anywhere サーバ データベース管理』を参照してください。

- **新しい -ls mlreplay オプションの追加**　-ls mlreplay オプションは、シミュレートしたクライアントごとの統計情報をログに記録します。「Mobile Link Replay ユーティリティ (mlreplay)」『Mobile Link サーバ管理』を参照してください。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

- **新しい -os mlreplay および mlgenreplayapi オプションの追加**　-os オプションはログファイルの最大サイズを制限します。「Mobile Link Replay ユーティリティ (mlreplay)」『Mobile Link サーバ管理』と「生成された Mobile Link リプレイ API ユーティリティ (mlgenreplayapi)」『Mobile Link サーバ管理』を参照してください。

- **コマンドラインで複数のシミュレートしたクライアントを実行可能**　-u、-p、および -r パラメータでアスタリスク文字を使用して、コマンドラインで複数のユーザ名、パスワード、およびリモート ID をそれぞれ指定できるようになりました。アスタリスク文字はシミュレートしたクライアント呼び出しごとにシミュレートしたクライアント数に置き換えられます。

  新しい -rp mlreplay オプションを使用すれば、アスタリスクのワイルドカード記号を別の記号に変更できます。次に新しい記号を使用して、複数のユーザ名、パスワード、およびリモート ID を指定できます。「Mobile Link Replay ユーティリティ (mlreplay)」『Mobile Link サーバ管理』を参照してください。

Sybase Central の Mobile Link プラグイン

次に、SQL Anywhere バージョン 12.0.1 での Sybase Central 用 Mobile Link プラグインの強化を示します。

- **同期モデルの展開で同期プロファイルを作成できるようになりました**　同期モデルをリモートデータベースに展開するときに、自動的に同期プロファイルが作成されるようになりました。

- **同期モデルが SAP Sybase IQ をサポート**　SAP Sybase IQ 統合データベースで同期モデルを作成できるようになりました。「SAP Sybase IQ の統合データベース」『Mobile Link サーバ管理』を参照してください。

- **トリガの代わりにカラムデフォルトを使用するタイムスタンプダウンロード**　SQL Anywhere、SAP Sybase IQ、IBM DB2 LUW、および MySQL 統合データベースを使用する同期モデルでトリガを使用して TIMESTAMP カラムを維持することなく、タイムスタンプベースのダウンロードを使用できるようになりました。ローが挿入または更新されるときに、トリガの代わりにカラムデフォルト値を使用して、ロー内の TIMESTAMP カラムを自動的に更新できます。これを簡単にするために、テーブルマッピングエディタの**同期モデル作成ウィザード**および **[ダウンロードタイプ]** タブの **[タイムスタンプダウンロードのオプション]** ページに新しい **[トリガの代わりにカラムデフォルトを使用する]** オプションが追加されました。

- **統合データベースからリモートスキーマ名をインポート可能**　**プロジェクト作成ウィザード**または**統合データベース追加ウィザード**のいずれを使用する場合も、プロジェクトに統合データベースを追加するときは、ウィザードによって、プロジェクトに存在しないリモートスキーマ名が統合データベースに定義されているかどうかが自動的にチェックされます。定義されている場合は、それをインポートするかどうか尋ねられます。

- **ファイルからグループを作成可能**　**グループ作成ウィザード**を使用すれば、エージェントの名前を含むテキストファイルを指定して、グループのエージェントを選択できます。統合データベースで定義されたエージェントを参照して、メンバーを追加することもできます。

- **リモートタスクコマンドの新しいヘルプボタン**　**[コマンドタイプ]** ドロップダウンリストの右にリモートタスクの作成に使用するヘルプを表示できる疑問符マークのアイコンが付きました。

- **エージェントの新しい [説明] フィールド**　**[エージェントのプロパティ]** ウィンドウおよび **Mobile Link エージェント作成ウィザード**にエージェントの説明を入力できる **[説明]** フィールドができました。

- **新しい CustDB サンプルプロジェクトファイルを追加**　CustDB サンプルプロジェクトファイル、*%SQLANY12%¥MobiLink¥CustDB¥project.mlp* を使用できるようになり、CustDB プロジェクトを簡単に操作して、データベーススクリプトを表示できるようになりました。詳細については、「Mobile Link の CustDB サンプル」『Mobile Link クイックスタート』と「チュートリアル：Ultra Light の CustDB サンプルアプリケーションの構築」『Ultra Light データベース管理とリファレンス』を参照してください。

Mobile Link クライアント

次に、SQL Anywhere バージョン 12.0.1 での Mobile Link クライアントの強化を示します。

- **HTTP セッションの自動再開**　HTTP 接続が失われると、Mobile Link クライアントが自動的に Mobile Link サーバに接続して同期を続けようとします。数回試行した後にクライアントが Mobile Link サーバに再接続できないと、同期は失敗します。

- **エンドツーエンド暗号化で DER でコード化されたキーをサポート**　e2ee_public_key クライアントプロトコルオプションで PEM でコード化されたキーと DER でコード化されたキーをサポートするようになりました。

- **ECC 曲線サポートを 15 曲線をサポートするように拡張**　サポートされる ECC 曲線は、sect163k1、sect163r2、sect233k1、sect233r1、sect283k1、sect283r1、sect409k1、sect409r1、sect571k1、sect571r1、secp192r1、secp224r1、secp256r1、secp384r1、および secp521r1 です。

- **x.509 証明書からのパブリックキーのサポート**　エンドツーエンド暗号化で x.509 証明書からのパブリックキーをサポートするようになりました。「e2ee_public_key」『Mobile Link クライアント管理』を参照してください。

Relay Server

このリリースでは、次の Relay Server 機能が追加されています。

- **Relay Server Outbound Enabler (RSOE) が Web サーバおよび HTTP プロキシに対して HTTP 認証 (基本認証とダイジェスト認証) をサポートするようになりました。**　Outbound Enabler に次の新しい (オプションの) ネットワーク接続オプションが追加されました。

  - **http_userid**　認証のユーザ ID。
  - **http_password**　認証のパスワード。
  - **http_proxy_userid**　プロキシ認証のユーザ ID。
  - **http_proxy_password**　プロキシ認証のパスワード。
  - **proxy_host**　プロキシサーバのホスト名または IP アドレスを指定します。

○ **proxy_port**　プロキシサーバのポート番号を指定します。

詳細については、次の項を参照してください。

○ 「Outbound Enabler」『Relay Server』
○ 「Mobile Link クライアントネットワークプロトコルオプション」『Mobile Link クライアント管理』

Mobile Link の動作の変更

次に、バージョン 12.0.1 で導入された Mobile Link の動作の変更を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

Mobile Link サーバの変更

● **64 ビットのオペレーティングシステムでの 32 ビットの Mobile Link サーバのサポート廃止**　64 ビットのオペレーティングシステムでは、Mobile Link サーバは 64 ビットのアプリケーションとしてのみインストールされ、64 ビットのアプリケーションとして動作します。旧バージョンの 12 のインストールからシステムに残っている 32 ビットの Mobile Link サーバは 64 ビットシステムでは動作できません。

> **注意**
> 12.0.1 より以前のバージョンの Mobile Link サーバの場合、Mobile Link サーバが 64 ビットのオペレーティングシステム上で 32 ビットのアプリケーションとして実行された場合、エラーは発生**しません**。

12.0.1 以降のバージョンでは、Mobile Link サーバが 64 ビットのオペレーティングシステム上で 32 ビットのアプリケーションとして実行された場合、エラー -10381 が返されるようになりました。「Mobile Link サーバは、64 ビットオペレーティングシステムで、64 ビットアプリケーションとして実行する必要があります」『エラーメッセージ』を参照してください。

● **-w mlsrv12 オプションの新しい動作**　-wm オプションを使用してデータベースワーカスレッドの数を自動的に調整した場合に、mlsrv12 の -w オプションでデータベースワーカスレッドの初期数が設定されるようになりました。以前は、このオプションでデータベースワーカスレッドの数を設定しました。「-w mlsrv16 オプション」『Mobile Link サーバ管理』と「-wm mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

● **-cm オプションが -cmax オプションのエイリアスになる**　-cm オプションが -cmax オプションのエイリアスになりました。「-cmax mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

Mobile Link サーバユーティリティの変更

次に、バージョン 12.0.1 で導入された Mobile Link サーバユーティリティの動作の変更を示します。

Mobile Link リプレイユーティリティ (mlreplay)

- **-o、-ot mlreplay および mlgenreplayapi オプションの新しい動作**　mlreplay および mlgenreplayapi の -o および -ot オプションでコマンドラインオプションをログに記録するようになりました。「Mobile Link Replay ユーティリティ (mlreplay)」『Mobile Link サーバ管理』と「生成された Mobile Link リプレイ API ユーティリティ (mlgenreplayapi)」『Mobile Link サーバ管理』を参照してください。

- **一定の条件下で -n オプションと -sci mlreplay オプションを同時に使用可能**　-n オプションと -sci mlreplay オプションは、-n オプションで指定されシミュレートされたクライアントの数がシミュレートされたクライアント情報ファイルのシミュレートされたクライアント数と同じか少ないときには、一緒に使用できます。同時に使用するときは、-n では実行するシミュレートされたクライアントの数を指定します。

  これらのオプションを使用することにより、1 つのシミュレートされたクライアント情報ファイルにシミュレートされたクライアント数 *x* を指定することによって、1 ～*x* の範囲の数のシミュレートされたクライアントのプロトコルをリプレイできます。以前の動作は、正確に *x* 数のシミュレートされたクライアントで記録されたプロトコルをリプレイするためにだけ使用できます。「Mobile Link Replay ユーティリティ (mlreplay)」『Mobile Link サーバ管理』を参照してください。

- **IdentifySimulatedClient および FiniIdentifySimulatedClient コールバックでコードテンプレートを使用可能**　リプレイセッションに関する一般ユーザ名、パスワード、およびリモート ID を使用するときに実装できるように、これらのコールバックには生成時にコメントアウトされたコードが記述されています。「IdentifySimulatedClient コールバック」『Mobile Link サーバ管理』と「FiniIdentifySimulatedClient コールバック」『Mobile Link サーバ管理』を参照してください。

Mobile Link クライアントの変更

次に、バージョン 12.0.1 で導入された Mobile Link クライアントの動作の変更を示します。

- **[SQL Anywhere の DBMLSync 設定] ウィンドウで次の変更が行われました**

  ○**[リモートプログレスに対してリトライする]**、**[リモートのオフセットが後]**、**[リモートのオフセットが前]**、**[競合する接続の削除]**、**[サイトスクリプト]**、**[コマンドラインヘルプ]** オプションは削除されました。

  ○**[パブリケーション]** オプションは **[サブスクリプション]** オプションに置き換えられました。

  ○Mobile Link パスワードを変更しようとすると、ソフトウェアはウィンドウを閉じる前に、空以外のパスワードを入力したこと、新しいパスワードと確認用のパスワードが一致することを確認します。

  ○不正な Mobile Link パスワードが入力されたために、同期の試行後にウィンドウが開くと、問題を示すメッセージが表示されます。以前は、エラーメッセージは dbmlsync ウィンドウに表示されました。

- **永続的な HTTP および HTTPS の同期の新しいデフォルト**　デフォルトで、HTTP または HTTPS を使用して同期する Mobile Link クライアント (Ultra Light J を除く) が永続的な同期を使用するようになりました。以前の動作を返すには、クライアントはクライアントネットワークプロトコルオプションで persistent=no を指定する必要があります。

Sybase Central の Mobile Link プラグインの変更

次に、バージョン 12.0.1 で導入された Sybase Central の Mobile Link プラグインの動作の変更を示します。

- **同期モデル展開ウィザードの [Mobile Link ユーザおよびサブスクリプション] ページが変わりました**　同期モデル展開ウィザードの **[Mobile Link ユーザおよびサブスクリプション]** ページの名前が **[Mobile Link ユーザおよび同期プロファイル]** ページに変わりました。このページに同期プロファイル名を指定するオプションができました。同期モデルが展開されると、同期プロファイルが作成されるようになりました。

- **リモートタスクに適したデフォルトを選択する同期モデル展開オプション**　同期モデル展開ウィザードの最初のページにリモートデータベースの集中管理用のリモートタスクを作成するために適切な設定でウィザードを初期化するようにリモートデータベースに展開する場合のオプションが追加されました。このオプションはデフォルトで選択されています。

Mobile Link の廃止予定機能とサポート終了機能

> **注意**
> 廃止予定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、変更される可能性があります。

- **IBM DB2 メインフレームのサポート終了**　IBM DB2 メインフレームは統合データベースとしてサポートされなくなりました。ただし、DB2 LUW (Linux、UNIX、Windows) は従来どおり、Mobile Link で統合データベースとしてサポートされます。

SQL Remote の新機能

次に、バージョン 12.0.1 で導入された SQL Remote の追加機能を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

- **SET REMOTE OPTION 文の新しい制御パラメータ**　SQL Remote が継続モードで実行されていて、メッセージシステムへのアクセス時にエラーが発生すると、次の項目を制御できるようになりました。

  ○シャットダウンまでに SQL Remote で送受信を再試行する回数。max_retries 共通オプションについては、「SET REMOTE OPTION 文 [SQL Remote]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

○再試行間で待機する時間。pause_after_failure 共通オプションについては、「SET REMOTE OPTION 文 [SQL Remote]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

SQL Remote の動作の変更

次に、バージョン 12.0.1 で導入された SQL Remote の変更点を示します。

- **見つからないメッセージを処理するパフォーマンスの向上**　SQL Remote バージョン 16.0 以前では、マルチパートのメッセージのすべてのメッセージが受信される前に統合データベースが終了すると、リモートデータベースはすべての受信メッセージを削除して、統合データベースに再送信要求を送信します。統合データベースは再送信要求を満たすために該当するトランザクションログをスキャンします。SQL Remote バージョン 16.0 では、起動時に SQL Remote が見つからないメッセージに関する警告メッセージを発行し、既存のメッセージを削除して、コミットを含み、マルチパートのメッセージの最後のパートであるメッセージまたは前のマルチパートのメッセージ後の次のメッセージとなるメッセージのみを再送信するように求めます。

Ultra Light の新機能

次に、バージョン 12.0.1 で導入された Ultra Light の追加機能を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

一般的な機能

- **Android のスマートフォンで Ultra Light がサポートされるようになりました**　Ultra Light が Android のスマートフォンをサポートするようになりました。Ultra Light J API の Android 実装を使用して、Ultra Light データベースを使用するアプリケーションを作成できます。この実装の使用方法は、BlackBerry スマートフォンに固有の Ultra Light データベースをサポートする BlackBerry の実装とは異なります。詳細については、「Windows Mobile 用の Ultra Light API の利点」『Ultra Light データベース管理とリファレンス』と「Ultra Light J アプリケーションの開発」『Ultra Light Java プログラミング』を参照してください。

  新しいチュートリアルでは、Android のスマートフォンまたは Eclipse 環境のシミュレータで Ultra Light アプリケーションを作成する方法を示します。このチュートリアルは、*%SQLANY12%¥UltraLiteJ¥Android¥CustDB* ディレクトリにある新しいコードサンプルに基づいています。「チュートリアル：Android アプリケーションの構築」『Ultra Light Java プログラミング』を参照してください。

- **BlackBerry アプリケーション作成用のチュートリアルで Eclipse 環境がサポートされるようになりました**　BlackBerry チュートリアルが Eclipse 環境を使用するように更新されました。「チュートリアル：BlackBerry アプリケーションの構築」『Ultra Light Java プログラミング』を参照してください。

- **動的キャッシュサイズ決定**　Ultra Light で、データベース操作への応答で保証され、利用可能なメモリで許容される場合、データベースファイルキャッシュが増加するようになりました。アプリケーションで明示的にキャッシュサイズを決定できるようにもなりました (通常、アプリケーションがメモリ使用率を減らすように要求される場合)。次の項を参照してください。

  ○「Ultra Light CACHE_SIZE 接続パラメータ」『Ultra Light データベース管理とリファレンス』
  ○「Ultra Light CACHE_MIN_SIZE 接続パラメータ」『Ultra Light データベース管理とリファレンス』
  ○「UltraLite CACHE_MAX_SIZE 接続パラメータ」『Ultra Light データベース管理とリファレンス』
  ○「Ultra Light cache_allocation オプション」『Ultra Light データベース管理とリファレンス』

- **Ultra Light でダウンロード専用テーブルをサポート**

  _download_only サフィックス　(Ultra Light Java Edition データベースではない) Ultra Light データベースでは、統合データベース上のテーブルへの変更が同期中にダウンロードされますが、ローカルの変更は Mobile Link に送信されません。サフィックス **_download_only** 付きのテーブルは、ダウンロード専用とマークされます。CREATE TABLE and ALTER TABLE SQL 文の synchronization constraint 句で **SYNCHRONIZE DOWNLOAD** を指定して、テーブルをダウンロード専用に設定することもできます。「Ultra Light ダウンロード専用テーブル」『Ultra Light データベース管理とリファレンス』、「CREATE TABLE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』、「ALTER TABLE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』を参照してください。

  新しい TABLE_IS_DOWNLOAD_ONLY フラグ　テーブルを同期化するときに、コミットされていないクライアント側の変更が原因でダウンロードの衝突が生じる場合があります。Ultra Light および Ultra Light Java Edition では、systable システムテーブルの table_flags カラムに TABLE_IS_DOWNLOAD_ONLY フラグを含むことができるようになりました。

  テーブルの sync=download 属性　アンロードされた Ultra Light データベースの XML フォーマットでテーブルに sync="download" 属性も含まれるようになりました。「systable システムテーブル」『Ultra Light データベース管理とリファレンス』を参照してください。

- **TLS ID をデータベース内に格納できる**　(Ultra Light Java Edition を含まない) Ultra Light データベースでは、X.509 証明書、プライベートキー、およびオプションでクライアントの証明書に署名した一連の証明書認証機関の証明書で構成される TLS ID を作成時に Ultra Light データベースに格納できるようになりました。ulinit および ulload ユーティリティに新しいオプションが追加されました。「Ultra Light データベース初期化ユーティリティ (ulinit)」『Ultra Light データベース管理とリファレンス』、「Ultra Light データベースへの XML のロードユーティリティ (ulload)」『Ultra Light データベース管理とリファレンス』を参照してください。

- **Ultra Light J コードサンプルがデバイスおよびプラットフォームサポート別に整理されました**　コードサンプルは、次のディレクトリに置かれるようになりました。

  ○BlackBerry サンプルの場合：*%SQLANY12%¥UltraLiteJ¥BlackBerry*

  ○Java J2ME のサンプルの場合：*%SQLANY12%¥UltraLiteJ¥J2ME*

○Java J2SE のサンプルの場合：*%SQLANY12%¥UltraLiteJ¥J2SE*

○Android サンプルの場合：*%SQLANY12%¥UltraLiteJ¥Android*

「サンプルコード」『Ultra Light Java プログラミング』を参照してください。

プラットフォームとデバイス

- **新しいデバイス**　Ultra Light J プログラミングインタフェースで Android スマートフォンがサポートされるようになりました。Ultra Light J プログラミングインタフェースは BlackBerry プラットフォームと Android プラットフォームで共通で、アプリケーションは各プラットフォーム用に Java で開発する必要があります。BlackBerry スマートフォンでは、基礎になる DBMS は Ultra Light の Java 実装ですが、Android では、DBMS は iPhone や iPad、Windows Mobile、および Windows でも提供される C++ バージョンです。

  次の項を参照してください。

  ○「Ultra Light 概要」『Ultra Light データベース管理とリファレンス』
  ○「Ultra Light J アプリケーションの開発」『Ultra Light Java プログラミング』
  ○「チュートリアル：Android アプリケーションの構築」『Ultra Light Java プログラミング』
  ○「Ultra Light J API リファレンス」『Ultra Light Java プログラミング』

プログラミングインタフェース

Ultra Light.NET

- **非同期の同期**　非同期の同期をサポートするために次のメンバーが追加されました。

  ○ **public delegate void ULSyncProgressedDlg(IAsyncResult result, ULSyncProgressData data)**　このメソッドは、同期化中に進捗情報を提供するために呼び出されます。ULSyncProgressedDlg デリゲート [Ultra Light.NET]『Ultra Light .NET プログラミング』を参照してください。

  ○ **public IAsyncResult ULConnection.BeginSynchronize()**　このメソッドは、同期化を実行する新しいスレッドを作成して、すぐに返します。ULConnection.BeginSynchronize メソッド [Ultra Light.NET]『Ultra Light .NET プログラミング』を参照してください。

  ○ **public IAsyncResult ULConnection.BeginSynchronize(Control control, ULSyncProgressedDlg dlg, object state)**　このメソッドは、同期化を実行する新しいスレッドを作成して、すぐに返します。ULConnection.BeginSynchronize メソッド [Ultra Light.NET]『Ultra Light .NET プログラミング』を参照してください。

  ○ **public void ULConnection.CancelSynchronize(IAsyncResult asyncResult)**　このメソッドは、同期化スレッドに終了を伝えて、すぐに返します。ULConnection.CancelSynchronize メソッド [Ultra Light.NET]『Ultra Light .NET プログラミング』を参照してください。

  ○ **public void ULConnection.EndSynchronize(IAsyncResult asyncResult)**　このメソッドは非同期で起動された同期が終了するまでブロックします。同期が失敗すると、

ULException がスローされます。ULConnection.EndSynchronize メソッド [Ultra Light.NET]『Ultra Light .NET プログラミング』を参照してください。

○ **public bool ULSyncProgressData.IsFinalSyncProgress**　これが同期の最終同期進捗メッセージである場合、このプロパティは true です。ULSyncProgressData.IsFinalSyncProgress プロパティ [Ultra Light.NET]『Ultra Light .NET プログラミング』を参照してください。

Ultra Light for M-Business Anywhere

この API は、Ultra Light バージョン 12.0.0 および 12.0.1 で廃止されています。

Ultra Light J

● **Android デバイス用のデータベース設定サポート**　新しい DatabaseManager.CreateConfigurationFileAndroid メソッドは、Android デバイス上のファイルに保存される永続的なデータベースの ConfigFileAndroid オブジェクトを確立します。ConfigFileAndroid インタフェース [Android] [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

● **ConfigPersistent インタフェースへの Andoroid 関連の更新**　次のメンバーは、Ultra Light J の Android サポートの影響を受けます。

メンバー名	ステータス
enableAesDBEncryption メソッド	Android 専用の新しいメソッドです。データベースの AES 暗号化を可能にします。ConfigPersistent.enableAesDBEncryption メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。
getAutoCheckpoint メソッド	BlackBerry では廃止され、Android では有効です。
getConnectionString メソッド	Android 専用の新しいメソッドです。SetConnectionString で登録される接続文字列を取得します。ConfigPersistent.getConnectionString メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。
getCreationString メソッド	Android 専用の新しいメソッドです。SetCreationString で登録される作成文字列を取得します。ConfigPersistent.getCreationString メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

メンバー名	ステータス
getDatabaseKey メソッド	Android 専用の新しいメソッドです。setEncryptionKey で登録されるデータベース暗号化キーを取得します。ConfigPersistent.getEncryptionKey メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。
getUserName メソッド	Android 専用の新しいメソッドです。setUserName で設定されるユーザの名前を取得します。ConfigPersistent.getUserName メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。
setAutocheckpoint メソッド	BlackBerry では廃止され、Android では有効です。
setConnectionString メソッド	Android 専用の新しいメソッドです。データベースの作成または接続に使用される接続文字列を設定します。ConfigPersistent.setConnectionString メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。
setCreationString メソッド	Android 専用の新しいメソッドです。データベースを作成するために使用される作成文字列を設定します。ConfigPersistent.setCreationString メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。
setDatabaseKey メソッド	Android 専用の新しいメソッドです。暗号化キーを設定します。ConfigPersistent.setEncryptionKey メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。
setEncryption メソッド	BlackBerry 専用です。
setUserName メソッド	Android 専用の新しいメソッドです。ユーザの名前を設定します。ConfigPersistent.setUserName メソッド [Android] [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

詳細については、ConfigPersistent インタフェース [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

- **Connection インタフェースへの Andoroid 関連の更新**　次のメンバーは、Ultra Light J の Android サポートの影響を受けます。

メンバー名	ステータス
checkpoint メソッド	BlackBerry では廃止され、Android では有効です。
emergencyShutdown メソッド	Android では使用できません。
getDatabaseId メソッド	Android では使用できません。
getDatabaseProperty メソッド	Android では使用できません。
getState メソッド	Android では使用できません。
isSynchronizationDeleteDisabled メソッド	Android では使用できません。
OPTION_BLOB_FILE_BASE_DIR 変数	BlackBerry および Java SE 専用です。

詳細については、Connection インタフェース [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

- **DatabaseManager クラスへの Andoroid 関連の更新**　次のメンバーは Android では使用できません。

○createConfigurationNonPersistent メソッド

○createFileTransfer メソッド

詳細については、DatabaseManager クラス [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

- **ResultSet インタフェースへの Andoroid 関連の更新**　次のメンバーは Android では使用できません。

○getBoolean メソッド

○getBytes メソッド

○getClobReader メソッド

○getDate メソッド

○getDecimalNumber メソッド

○getDouble メソッド

○getFloat メソッド

○getLong メソッド

○getSize メソッド

○getString メソッド

○getUUIDValue メソッド

○isNull メソッド

詳細については、ResultSet インタフェース [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

Ultra Light の動作の変更と廃止予定機能

次に、バージョン 12.0.1 で導入された Ultra Light の廃止予定機能と動作の変更を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

注意
廃止予定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、変更される可能性があります。

動作の変更

- **SET OPTION 動作への変更**　Ultra Light は、次の永続的なオプションが設定されているとコミット操作を実行するようになりました。「SET OPTION 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』を参照してください。

- **永続的なネットワークプロトコルオプションのデフォルト設定が true になりました**　Mobile Link クライアント (Ultra Light J を除く) の永続的なネットワークプロトコルオプションのデフォルト値が "true" になりました。永続的な接続を使用する同期は、特に HTTPS の場合、非永続的な接続よりも高速です。ただし、仲介サーバが非永続的な接続を要求したり、クライアントで仲介サーバが永続的な接続に対応していないことが検出されたりすると、残りの同期化セッションで接続が自動的に非永続的な接続にダウングレードされます。「persistent」『Mobile Link クライアント管理』を参照してください。

- **SQLE_COMMUNICATIONS_ERROR (-85) エラーメッセージが SQLE_MOBILINK_COMMUNICATIONS_ERROR (-1305) に変わりました**　通信の問題に関するすべての情報がこのエラーに含まれます。別々のストリームエラーオブジェクトからこれらの値にアクセスする必要はなくなりましたが、これらの値を参照する既存のコードは引き続き機能します。このエラーのパラメータはストリームエラーコード、パラメータおよびシステムコードです。「通信エラーが発生しました。」『エラーメッセージ』を参照してください。

- **SQLE_UNKNOWN_PROPERTY エラーメッセージ**　不明なプロパティ名に対して SQLE_UNKNOWN_PROPERTY が通知されるようになりました。「'%1' は認識できないプロパティです。」『エラーメッセージ』を参照してください。

- **SQLE_MOBILINK_AUTHENTICATION_FAILED エラーメッセージ**　同期化中に SQLE_INVALID_LOGON ではなく SQLE_MOBILINK_AUTHENTICATION_FAILED が通知され、認証ステータスと値が含まれます。「Mobile Link が値 '%2' の認証ステータス '%1' を返したため、同期に失敗しました。」『エラーメッセージ』を参照してください。

管理ツールの新機能

次に、バージョン 12.0.1 で導入された管理ツールの追加機能を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

- **OEM での DBISQL、Sybase Central、DBConsole および ML Monitor で使用する優先ディレクトリの指定**　管理ツールとともに *OEM.ini* ファイルを配備する場合、ファイルに管理ツールの優先ディレクトリを指定する次の行が含まれるようにします。

  ```
  [preferences]
  directory=preferences_file_directory
  ```

  「管理ツールの設定」『SQL Anywhere サーバ プログラミング』を参照してください。

Sybase Central プラグインの新機能

次に、バージョン 12.0.1 で導入された Sybase Central のプラグインの追加機能を示します。

SQL Anywhere プラグインの新機能

- **新しいアプリケーションプロファイリングの推奨内容**　**[アプリケーションプロファイリング] ウィザード**を使用してデータベースのスキーマに基づいてデータベースの全体的なパフォーマンスを確認する場合、このウィザードによって、テーブル内のカラムがチェックされます。幅の広いカラムの大半が幅の狭いカラムよりも前にある場合、ウィザードにより推奨内容が作成されます。幅の広いカラムが幅の狭いカラムよりも前にあるように並べると、応答時間に悪影響を及ぼす場合があります。幅の狭いカラムは、アクセス頻度が低い場合以外は、テーブル宣言で幅の広いカラムよりも前に定義します。

  幅の広いカラムとは、15 バイトより大きなサイズのカラム、LONG 型データのカラム (LONG VARCHAR など)、または XML として定義されたカラムを指します。「ヒント：テーブル内のカラムの順序の確認」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **ユーザのパーミッションの管理および表示方法の強化**　プラグインでユーザのパーミッションを表示する方法に多数の強化が行われました。データベースオブジェクトに明示的に設定されたパーミッションの表示と変更がより簡単になりました。データベースオブジェクトを選択すると、パーミッション情報が右ウィンドウ枠のタブに表示されるようになりました。カラムのパーミッションは対応するテーブルパーミッションと共に表示されます。「オブジェクトレベル権限の付与 (Sybase Central の場合)」『SQL Anywhere サーバ データベース管理』を参照してください。

Sybase Central プラグイン動作の変更

次に、バージョン 12.0.1 で導入された Sybase Central のプラグインの動作変更を示します。

SQL Anywhere プラグインの新機能

- **Sybase Central は、SQL Anywhere バージョン 10x 以降のデータベースのみサポート**　バージョン 9 のデータベースサーバと、バージョン 9 のソフトウェアで作成されたデータベースのサポートが SQL Anywhere プラグインから削除されました。データベースをアンロードして、再ロードファイル、新しいデータベース、または既存のデータベースに再ロードする場合、バージョン 9 以降のデータベースサーバで稼働している、バージョン 5、6、7、8 または 9 のソフトウェアで作成したデータベースには引き続き接続できます。「SQL Anywhere サーバのアップグレード」289 ページを参照してください。

Interactive SQL の新機能

次に、バージョン 12.0.1 で導入された Interactive SQL の追加機能を示します。

- **[接続] ウィンドウが SAP Sybase IQ をサポート**　Interactive SQL を使用して、SAP Sybase IQ データベースに接続できます。**[接続]** ウィンドウで、**[データベースタイプを変更]** をクリックし、次に **[SAP Sybase IQ]** をクリックします。

- **テキスト補完**　次のリストは、SQL と Sybase Central のテキスト補完機能に関する変更を示しています。

  ○**[SQL 文]** ウィンドウ枠に入力すると、デフォルトでテキスト補完ウィンドウが自動的に表示されるようになりました。テキスト補完ウィンドウは、**[編集]** » **[テキスト補完を開く]** をクリックするか、[Ctrl+Space] キーを押しても開くことができます。

  ○デフォルトでは、SQL キーワードおよび補完されたデータベースオブジェクト名は二重引用符で囲まれます。

  ○SQL 文とキーワードが候補のリストに含まれるようになりました。

  ○テキスト補完ウィンドウで文字列は大文字と小文字が区別されるが、ID は区別されないデータベースを取り扱うことができるようになりました。

  ○次のキーボードショートカットが変更されました。

以前のショートカット	新しいショートカット	説明
[Tab]	[Ctrl + A]	内容に関係なく一致するデータのリストを表示します。
+	[Ctrl + プラス記号 (+)]	項目をパラメータリスト付きで **[SQL 文]** ウィンドウ枠に追加します。

以前のショートカット	新しいショートカット	説明
*	[Ctrl + アスタリスク]	項目をパラメータおよびタイプリスト付きで **[SQL 文]** ウィンドウ枠に追加します。
'	[Ctrl + 二重引用符 (")]	項目を引用符で囲んで **[SQL 文]** ウィンドウ枠に追加します。
なし	[Tab]	選択内容を受け入れ、テキスト補完ウィンドウを閉じます。

ショートカットの完全なリストについては、「テキスト補完キーボードショートカット」『SQL Anywhere サーバ データベース管理』を参照してください。

詳細については、「テキスト補完」『SQL Anywhere サーバ データベース管理』を参照してください。

- **新しい [インポート／エクスポート] オプション**　**[オプション] » [インポート／エクスポート]** をクリックすると、NULL 値を **[NULL 値のエクスポート方法]** フィールドにエクスポートする方法を指定できます。「output_nulls オプション [Interactive SQL]」『SQL Anywhere サーバ データベース管理』を参照してください。

Interactive SQL の動作の変更

次に、バージョン 12.0.1 で導入された Interactive SQL の変更点を示します。

- **READ 文エンコードのアルゴリズムの変更**　Interactive SQL で READ 文を実行するとき、データの読み込みに使用されるエンコードが次の順序で決定されるようになりました。

  1. ENCODING 句で指定されたエンコード (この句が指定されている場合)。
  2. ファイル内でバイト順マークで指定されたエンコード (バイト順マークが指定されている場合)。
  3. デフォルトのテキスト形式のエンコード (この形式のエンコードが指定されている場合)。
  4. 実行しているプラットフォームのデフォルトのエンコード。英語版 Windows コンピュータでは、デフォルトのエンコードは 1252 です。

  「READ 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

SQL Anywhere モニタの新機能

次に、バージョン 12.0.1 で導入された SQL Anywhere モニタの追加機能を示します。

- **モニタで SNTP サーバを通して、電子メールで警告を送信できるようになりました**　SQL Anywhere モニタが TLS 接続を必要とする SMTP サーバを使用して、電子メールによる警告を送信できるようになりました (GMail など)。「モニタによる警告電子メールの送信の有効化」『SQL Anywhere サーバ データベース管理』を参照してください。

- **読み込み専用のスケールアウトおよびデータベースミラーリングシステムのモニタデータベース**　モニタは読み込み専用のスケールアウトシステムのルートノードのデータベースサーバおよびミラーリングシステムのプライマリデータベースサーバのモニタリングに対応します。ミラーリングシステムで、モニタはフェイルオーバが発生しても常にプライマリデータベースサーバをモニタリングするように設定できます。新しいウィジェットである、**[SQL Anywhere スケールアウトトポロジ]** ウィジェットは、SQL Anywhere のミラーリングおよびスケールアウトシステムのトポロジ情報を表示します。

  読み込み専用のスケールアウトシステムのルートノードおよびミラーリングシステムのプライマリデータベースのリソースを作成するには、**[リソースの追加]** ウィンドウを使用し、**[SQL Anywhere サーバ]** リソースを作成します。「データベースリソースの追加」『SQL Anywhere サーバ データベース管理』を参照してください。

  読み込み専用のスケールアウトシステムとその他のデータベースサーバのルートノードのリソース設定に違いはありません。リソースが設定または移行されると、モニタはそのリソースが読み込み専用のスケールアウトシステムのルートノードであることを検出します。

  ミラーリングシステムのプライマリデータベースのリソースには、特別な設定が必要です。フェイルオーバが発生してもモニタが常にプライマリデータベースサーバをモニタリングするようにするには、次のオプションを使用してリソースを設定する必要があります。

  ○**[ホスト]** フィールドでプライマリサーバとミラーサーバのホスト名とポート番号をカンマ区切りのリストで指定します。(例：my-primary-server:2638,my-mirror-server:49152)。

  ○**[ポート]** フィールドが空であることを確認します。

  ○**[サーバ]** フィールドにプライマリサーバの代替サーバ名を入力します。つまり、データベースミラーリングシステムのプライマリサーバとして動作するデータベースサーバに接続するためにクライアントで使用する名前を指定します。

  [SQL Anywhere スケールアウトトポロジ] をダッシュボードに追加する必要があります。これは新しいダッシュボードの作成時に表示されるデフォルトのウィジェットの 1 つではないからです。「ウィジェット」『SQL Anywhere サーバ データベース管理』を参照してください。

  「レッスン 9：(省略可) SQL Anywhere モニタからの読み込み専用スケールアウトシステムのモニタリング」『SQL Anywhere サーバ データベース管理』と「レッスン 8：(省略可) データベースミラーリングシステムのモニタリング」『SQL Anywhere サーバ データベース管理』を参照してください。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

さらに、新しい警告、**[切断されたスケールアウトノードの数が指定したスレッショルドを越えた場合に警告します:]** が追加されました。「警告スレッショルドの指定」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Mobile Link サーバの新しいモニタのメトリックと警告**　Mobile Link サーバリソースのモニタのメトリックに **[統合を使用できますか]** というメトリックが含まれるようになりました。このメトリックは、Mobile Link サーバが統合データベースに接続できない場合に **[True]** を示します。統合データベースを使用できなくなると、**[統合データベースを使用できません]** 警告が発効されます。

- **Mobile Link リソースのメトリックの新しい名前**　次の Mobile Link サーバリソースの SQL Anywhere モニタのメトリックが変わりました。

以前の名前	新しい名前
バイト読み取り率	TCP バイト読み取り率
バイト書き込み率	TCP バイト書き込み率

- **[警告の通知] ウィンドウ**　**[警告の通知]** ウィンドウに日付および送信された電子メールの種類別の電子メールセットの数を制限する 2 つのオプションが含まれるようになりました。

  優先度の高い警告の場合のみ電子メール通知を送る　優先度の高い警告が発行された場合のみ、電子メール通知を受け取る場合、このオプションを設定します。

  1 日に指定した数以上の電子メールを送信しない　1 日当たりで各ユーザに送信される通知メールの数を制限する場合にこのオプションを選択します。

  「モニタによる警告電子メールの送信の有効化」『SQL Anywhere サーバ データベース管理』を参照してください。

- **[警告の削除] ウィンドウ**　ユーザが警告を選択して、**[削除]** ボタンをクリックすると、**[警告の削除]** ウィンドウが表示され、ユーザに以下のオプションのいずれかを選択するように求めます。

  選択した警告を削除する　**[警告リスト]** ウィジェットで選択した警告を削除する場合のみ、このオプションを選択します。

  次より前に受信した警告を削除する　指定した時間より前に受信したすべての警告を削除するには、このオプションを選択します。

  次のリソースの警告を削除する　選択したリソースのすべての警告を削除するには、このオプションを選択します。

  「警告の削除」『SQL Anywhere サーバ データベース管理』を参照してください。

SQL Anywhere モニタの動作の変更

次に、バージョン 12.0.1 で導入された SQL Anywhere モニタの変更点を示します。

- **エクスポートできるデータの最大量**　ファイルにエクスポートできるデータの最大量は 25 メトリックまたは 100 万ポイントです。「メトリックのエクスポート」『SQL Anywhere サーバデータベース管理』を参照してください。

マニュアルの強化

次に、バージョン 12.0.1 で SQL Anywhere マニュアルに加えられた変更を示します。

- **SQL Anywhere チュートリアルのリスト**　このマニュアルに、SQL Anywhere のすべてのチュートリアル一覧が掲載されるようになりました。「SQL Anywhere チュートリアルのリスト」『SQL Anywhere 16 紹介』を参照してください。

バージョン 12.0.0 の新機能

バージョン 10 以前の SQL Anywhere の新機能と動作変更については、ftp://ftp2.ianywhere.jp/public/tech/dbwnja10.pdf を参照してください。

製品全体の新機能

次に、バージョン 12.0.0 で導入された製品全体の追加機能を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

- **設定ファイルでサポートされているエスケープ文字**　設定ファイルの解析が強化され、¥ のエスケープシーケンスとして ¥¥、" のエスケープシーケンスとして ¥" がサポートされるようになりました。「設定ファイルのエスケープ文字」『SQL Anywhere サーバ データベース管理』を参照してください。

- **展開ウィザードでの 64 ビットの展開のサポート**　以前のバージョンの展開ウィザードは 32 ビットの展開をサポートしていました。新バージョンでは、64 ビットの展開もサポートされるようになりました。「Deployment ウィザード」『SQL Anywhere サーバ プログラミング』を参照してください。

製品全体の動作の変更

次に、バージョン 12.0.0 で導入された製品全体の動作の変更を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

- **Sybase Replication Server 用の SQL Anywhere Replication Agent がサポートされない**　Sybase Replication Server 用の SQL Anywhere Replication Agent は、バージョン 12 ではサポートされません。Mobile Link や SQL Remote などの代替のレプリケーションまたは同期テクノロジを使用してください。「Mobile Link テクノロジ」『Mobile Link クイックスタート』と「SQL Remote システム」『SQL Remote』を参照してください。

  この変更により、次の変更がソフトウェアに追加されました。

  - **a_change_log DBTools 構造体**　ignore_ltm_trunc メンバーは、サポートされなくなりました。
  - **LTMGeneration データベースプロパティ**　このプロパティはシステムで使用するために予約されています。
  - **LTMTrunc データベースプロパティ**　このプロパティはシステムで使用するために予約されています。
  - **Log Transfer Manager ユーティリティ (dbltm)**　このユーティリティは削除されました。

- **ログ変換ユーティリティ (dbtran)**　-is オプションで、値 RepServer がサポートされなくなりました。

  -rsu オプションは削除されました。

- **サービスユーティリティ (dbsvc)**　Replication Agent 用にサービスを作成できなくなりました。SQLANYLTM サービスグループはサポートされなくなりました。

  -w オプションと -t オプションで、値 **dbltm** がサポートされなくなりました。

- **サポートユーティリティ (dbsupport)**　このユーティリティから、SQL Anywhere Replication Agent (dbltm) の情報が返されなくなりました。

- **トランザクションログユーティリティ (dblog)**　-g オプションと -il オプションは、サポートされなくなりました。「トランザクションログユーティリティ (dblog)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **replicate_all データベースオプション**　このオプションは削除されました。

- **delete_old_logs データベースオプション**　Replication Agent では、このオプションはサポートされません。

- **ALTER PROCEDURE 文**　次の構文は、サポートされなくなりました。

  ALTER PROCEDURE [*owner*.]*procedure-name*
  REPLICATE { **ON** | **OFF** }

  「ALTER PROCEDURE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ALTER TABLE 文**　**REPLICATE** { **ON** | **OFF** } 句はサポートされなくなりました。「ALTER TABLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

● **実行プログラムをデーモンとして実行する場合にユーザの umask 設定が考慮される**　以前のリリースでは、UNIX で実行プログラムをデーモンとして実行 (-ud オプションを指定して起動) した場合、ユーザの umask 設定は無視され、グループパーミッションおよび他の読み込みと書き込みのパーミッションを使用して新しいファイルを作成する、umask(0) が呼び出されていました。SQL Anywhere 12 の実行プログラムをデーモンとして起動すると、umask(0) は呼び出されず、ユーザの umask 設定が考慮されます。現在のユーザの umask 設定によって実行プログラムのパーミッションが制御されるため、実行プログラムを起動する前に、ユーザの umask 値が目的のレベルに設定されていることを確認してください。

この動作の変更は、次の実行プログラムに適用されます。

- dbeng12
- dbsrv12
- dbltm
- dbmlsync
- dbns12
- dbremote
- mlsrv12
- uleng12

SQL Anywhere の新機能

次に、SQL Anywhere バージョン 12.0.0 の新機能を示します。

サポートされるプラットフォームのリストに加えられた変更については、http://www.sybase.com/detail?id=1061806 を参照してください。

主な機能

次に、SQL Anywhere バージョン 12.0.0 の主な機能を示します。

- **新しい空間データサポート**　SQL Anywhere 12.0.0 の新しい空間データ機能をサポートするために、次の機能が追加されました。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。

> **注意**
> 32 ビット Windows と 32 ビット Linux の空間データサポートには、SSE2 命令をサポートする CPU が必要となります。これは、Intel Pentium 4 (2001 年リリース) 以降と AMD Opteron (2003 年リリース) 以降でサポートされます。

SQL 文　空間機能をサポートするために、次のような SQL 文の強化が行われました。

- **SHAPEFILE 句**　FROM 句の OPENSTRING サブ句で、新しい SHAPEFILE フォーマットオプションを使用できます。「FROM 句」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  また、LOAD TABLE 文の FORMAT 句で新しい SHAPEFILE フォーマットオプションを使用できます。「LOAD TABLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **CREATE SPATIAL REFERENCE SYSTEM 文**　空間参照系を作成するか、置き換えます。「CREATE SPATIAL REFERENCE SYSTEM 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ALTER SPATIAL REFERENCE SYSTEM 文**　既存の空間参照系の設定を変更します。空間参照系を変更する前に、注意事項について「備考」の項を参照してください。「ALTER

SPATIAL REFERENCE SYSTEM 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **DROP SPATIAL REFERENCE SYSTEM 文**　空間参照系を削除します。「DROP SPATIAL REFERENCE SYSTEM 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **CREATE SPATIAL UNIT OF MEASURE 文**　空間測定単位を作成するか、置き換えます。「CREATE SPATIAL UNIT OF MEASURE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **DROP SPATIAL UNIT OF MEASURE 文**　空間測定単位を削除します。「DROP SPATIAL UNIT OF MEASURE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

Interactive SQL の変更　Interactive SQL に新しいビューアツールの **[空間ビューア]** が追加され、空間ジオメトリを表示できるようになりました。ビューアの上部で空間データを問い合わせると、ビューアの下部に結果がイメージとして表示されます。「空間データのイメージとしての表示 (Interactive SQL の場合)」『SQL Anywhere サーバ 空間データサポート』を参照してください。

また、Interactive SQL で結果ローを確認する場合、新しい **[空間プレビュー]** タブを使用して、ジオメトリを SVG (Scalable Vector Graphic) としてプレビューできるようになりました。「空間データのイメージとしての表示 (Interactive SQL の場合)」『SQL Anywhere サーバ 空間データサポート』を参照してください。

新しいデータ型、メソッド、コンストラクタ　新しい型、メソッド、コンストラクタが追加され、空間データにアクセスし、そのデータをモデル化および分析できるようになりました。「空間データへのアクセスとそのデータの操作」『SQL Anywhere サーバ 空間データサポート』を参照してください。

また、空間データにアクセスし、そのデータを操作するときに通常の SQL 関数と似た処理を行う、数多くの空間互換関数を使用できるようになりました。これらの関数は、他の製品との互換性を確保するために用意されました。これらの関数では SQL Anywhere の空間メソッドとコンストラクタを利用します。「空間互換関数」『SQL Anywhere サーバ 空間データサポート』を参照してください。

新しい関数とシステムプロシージャ　データベースの空間データをサポートするために、次の関数とシステムプロシージャが追加されました。

- **TREAT 関数**　ジオメトリ式の宣言されたタイプをサブタイプに変更できます。「TREAT 関数 [データ型変換]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_describe_shapefile システムプロシージャ**　ESRI シェイプファイルに含まれるカラムの名前と型を記述します。このシステム機能は、空間機能とともに使用します。「sa_describe_shapefile システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

 Copyright © 2013, SAP AG or an SAP affiliate company. - SAP Sybase SQL Anywhere 16.0

- **sa_install_feature システムプロシージャ**　SQL Anywhere のインストール時にデータベースに存在していなかった追加の機能をインストールします。「sa_install_feature システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **st_geometry_dump システムプロシージャ**　入力に含まれるジオメトリオブジェクトの 1 つを表す各ローとともに、ジオメトリオブジェクトを結果セットに展開します。「st_geometry_dump システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

ウィザード　Sybase Central では、空間データ機能をサポートするために、次のウィザードが追加されました。

- **空間参照系の作成ウィザード**　**空間参照系の作成ウィザード**を使用すると、新しい空間参照系を作成できます。「空間参照系の作成」『SQL Anywhere サーバ 空間データサポート』を参照してください。

- **測定単位の作成ウィザード**　**測定単位の作成ウィザード**を使用すると、空間データで使用する新しい測定単位を作成できます。「測定単位の作成」『SQL Anywhere サーバ 空間データサポート』を参照してください。

カタログの変更　新しい空間データサポートの一環として、カタログは次のように変更されました。

- **SYSSPATIALREFERENCESYSTEM システムビュー**　SYSSPATIALREFERENCESYSTEM システムビューの各ローは、データベースに定義されている空間参照系に関する記述です。「SYSSPATIALREFERENCESYSTEM システムビュー」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **SYSUNITOFMEASURE システムビュー**　SYSUNITOFMEASURE システムビューの各ローは、データベースに定義されている測定単位に関する記述です。「SYSUNITOFMEASURE システムビュー」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ST_GEOMETRY_COLUMNS 統合ビュー**　ST_GEOMETRY_COLUMNS システムビューの各ローは、データベースに定義されている空間カラムに関する記述です。「ST_GEOMETRY_COLUMNS 統合ビュー」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ST_SPATIAL_REFERENCE_SYSTEMS 統合ビュー**　ST_SPATIAL_REFERENCE_SYSTEMS システムビューの各ローは、データベースに定義されている空間参照系に関する記述です。「ST_SPATIAL_REFERENCE_SYSTEMS 統合ビュー」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ST_UNITS_OF_MEASURE 統合ビュー**　ST_UNITS_OF_MEASURE システムビューの各ローは、データベースに定義されている測定単位に関する記述です。「ST_UNITS_OF_MEASURE 統合ビュー」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

データベースオプションとプロパティ　空間データ機能をサポートするために、次のデータベースオプションとプロパティが追加されました。

- **st_geometry_asbinary_format オプション**　ジオメトリからバイナリへの空間データ値の変換方法を制御します。「st_geometry_asbinary_format オプション」『SQL Anywhere サーバ データベース管理』を参照してください。
- **st_geometry_astext_format オプション**　ジオメトリからテキストへの空間データ値の変換方法を制御します。「st_geometry_astext_format オプション」『SQL Anywhere サーバ データベース管理』を参照してください。
- **st_geometry_asxml_format オプション**　ジオメトリから XML への空間データ値の変換方法を制御します。「st_geometry_asxml_format オプション」『SQL Anywhere サーバ データベース管理』を参照してください。
- **st_geometry_describe_type オプション**　空間データの記述方法を制御します。「st_geometry_describe_type オプション」『SQL Anywhere サーバ データベース管理』を参照してください。
- **st_geometry_on_invalid オプション**　ジオメトリが基本的な検証に失敗した場合の動作を制御します。「st_geometry_on_invalid オプション」『SQL Anywhere サーバ データベース管理』を参照してください。
- **st_geometry_asbinary_format 接続プロパティ**　空間値がジオメトリからバイナリに変換される方法を示す値を返します。st_geometry_asbinary_format 接続プロパティ『SQL Anywhere サーバ データベース管理』を参照してください。
- **st_geometry_astext_format 接続プロパティ**　空間値がジオメトリからテキストに変換される方法を示す値を返します。st_geometry_astext_format 接続プロパティ『SQL Anywhere サーバ データベース管理』を参照してください。
- **st_geometry_asxml_format 接続プロパティ**　空間値がジオメトリから xml に変換される方法を示す値を返します。st_geometry_asxml_format 接続プロパティ『SQL Anywhere サーバ データベース管理』を参照してください。
- **st_geometry_describe_type 接続プロパティ**　空間データ値をクライアントに対してどのような方法で記述するかを示す値を返します。st_geometry_describe_type 接続プロパティ『SQL Anywhere サーバ データベース管理』を参照してください。
- **st_geometry_on_invalid 接続プロパティ**　ジオメトリが基本的な検証に失敗した場合の動作を示す値を返します。st_geometry_on_invalid 接続プロパティ『SQL Anywhere サーバ データベース管理』を参照してください。

SYS_SPATIAL_ADMIN_ROLE グループ　このグループのメンバーシップが与えられたユーザは、空間参照系と計測単位を作成、変更、または削除できます。

SQL Anywhere の空間サポートの詳細については、「空間データ」『SQL Anywhere サーバ 空間データサポート』を参照してください。

- **読み込み専用のスケールアウト**　読み込み専用のスケールアウトシステムで SQL Anywhere を使用できるようになりました。この構成では、1 つのデータベースサーバ (ルートノード)

でデータベースの読み込み／書き込みコピーが実行される一方、他のデータベースサーバでデータベース (コピーノード) の読み込み専用コピーが実行されます。読み込み専用コピーを使用すると、データベースへの読み込みアクセスを必要とするレポートなどの操作の負荷を軽減できます。読み込み専用のスケールアウトは、そのままで、またはデータベースミラーリングと共に使用できます。この機能を使用するには、既存のデータベースをアップグレードまたは再構築する必要があります。

%SQLANYSAMP12%¥SQLAnywhere¥DBMirror に、データベースミラーリングシステムとスケールアウトシステムを組み合わせて使用するサンプルが追加されました。

次の項を参照してください。

- 「SQL Anywhere の読み込み専用スケールアウト」『SQL Anywhere サーバ データベース管理』
- 「NodeType (NODE) 接続パラメータ」『SQL Anywhere サーバ データベース管理』
- 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバ SQL リファレンス』
- 「ALTER MIRROR SERVER 文」『SQL Anywhere サーバ SQL リファレンス』
- 「SET MIRROR OPTION 文」『SQL Anywhere サーバ SQL リファレンス』
- 「DROP MIRROR SERVER 文」『SQL Anywhere サーバ SQL リファレンス』
- MIRROR SERVER 句： 「COMMENT 文」『SQL Anywhere サーバ SQL リファレンス』
- 「SYSMIRROROPTION システムビュー」『SQL Anywhere サーバ SQL リファレンス』
- 「SYSMIRRORSERVER システムビュー」『SQL Anywhere サーバ SQL リファレンス』
- 「SYSMIRRORSERVEROPTION システムビュー」『SQL Anywhere サーバ SQL リファレンス』
- MirrorRole データベースプロパティ『SQL Anywhere サーバ データベース管理』
- MirrorServerState プロパティと MirrorState プロパティ： 「sa_mirror_server_status システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』

- **データベースミラーリングの強化**　データベースサーバのコマンドラインでミラーリングの設定を指定する代わりに、SQL 文を使用してデータベースミラーリングシステムを設定できるようになりました。この機能を使用するには、既存のデータベースをアップグレードまたは再構築する必要があります。

次の項を参照してください。

- 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバ SQL リファレンス』
- 「ALTER MIRROR SERVER 文」『SQL Anywhere サーバ SQL リファレンス』
- 「SET MIRROR OPTION 文」『SQL Anywhere サーバ SQL リファレンス』
- 「DROP MIRROR SERVER 文」『SQL Anywhere サーバ SQL リファレンス』
- 「COMMENT 文」『SQL Anywhere サーバ SQL リファレンス』に追加された MIRROR SERVER 句
- 「SYSMIRROROPTION システムビュー」『SQL Anywhere サーバ SQL リファレンス』
- 「SYSMIRRORSERVER システムビュー」『SQL Anywhere サーバ SQL リファレンス』
- 「SYSMIRRORSERVEROPTION システムビュー」『SQL Anywhere サーバ SQL リファレンス』
- MirrorRole データベースプロパティ『SQL Anywhere サーバ データベース管理』
- 拡張データベースプロパティ：MirrorServerState、MirrorState
- 「sa_mirror_server_status システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
- データベースミラーリングの動作の変更と廃止予定機能149 ページ

- **Host 接続パラメータ**　新しい Host 接続パラメータでは、ホスト名 (または IP アドレス) とオプションのポート番号を使用して、データベースサーバがある場所をクライアントに知らせます。クライアントと異なるコンピュータで実行されているデータベースサーバに接続する場合は、この接続パラメータを使用することをおすすめします。「Host 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **自動統計管理の強化**　SQL Anywhere 12 には、データベースカラムに関する統計の自動保守を向上させる統計ガバナーが備えられています。データベース内の各統計の正常性と有用性が自動的に評価され、統計を自己モニタして自己回復できるように、必要な保守が実行されます。統計の保守はバックグラウンドで実行され、データベースサーバのパフォーマンスに対する大幅な負荷は発生しません。「統計ガバナーが統計を保守する方法」『SQL Anywhere サーバ SQL の使用法』を参照してください。

  sa_server_option システムプロシージャで、統計の管理に役立つオプション DropBadStatistics、DropUnusedStatistics、StatisticsCleaner がサポートされるようになりました。「sa_server_option システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **シーケンス**　SQL Anywhere でシーケンスの生成がサポートされるようになりました。シーケンスは、アプリケーションによってユニークキーの値を生成する場合に使用されます。シーケンス値を使用すると、アプリケーションは同時実行性とパフォーマンスの問題を回避できます。

  Sybase Central の SQL Anywhere プラグインを使用して、シーケンスを作成、編集、管理することもできます。たとえば、**シーケンスジェネレータの作成ウィザード**を使用して、データベースで新しいシーケンスを作成します。

参照：

- 「ユニークな値を生成するためのシーケンスの使用」『SQL Anywhere サーバ SQL の使用法』
- 「CREATE SEQUENCE 文」『SQL Anywhere サーバ SQL リファレンス』
- 「ALTER SEQUENCE 文」『SQL Anywhere サーバ SQL リファレンス』
- 「DROP SEQUENCE 文」『SQL Anywhere サーバ SQL リファレンス』
- 「SYSSEQUENCE システムビュー」『SQL Anywhere サーバ SQL リファレンス』
- 「SYSSEQUENCEPERM システムビュー」『SQL Anywhere サーバ SQL リファレンス』
- SEQUENCE 句、「COMMENT 文」『SQL Anywhere サーバ SQL リファレンス』
- GRANT USAGE ON SEQUENCE 構文、「GRANT 文」『SQL Anywhere サーバ SQL リファレンス』
- REVOKE USAGE ON SEQUENCE 構文、「REVOKE 文」『SQL Anywhere サーバ SQL リファレンス』

シーケンスを使用するには、既存のデータベースをアップグレードまたは再構築する必要があります。

- **マルチプログラミングレベルの強化**　ネットワークデータベースサーバ (dbsrv12) では、デフォルトでマルチプログラミングレベルが自動的に制御されるようになりました。この動作により、データベースサーバではスループットが向上し、DBA の介入なしで負荷の変更に適応できます。

  データベースサーバは、起動時にサービス要求に使用されるワーカーのプールを作成します。ワーカーの数は、サーバの現在のマルチプログラミングレベルです。プールには最小値と最大値があり、現在のマルチプログラミングレベルは常にこの範囲内にあります。DBA は、起動時にデータベースサーバオプションを使用するか、データベースサーバの実行中に sa_server_option システムプロシージャを使用して、最小値と最大値を変更できます。

  データベースサーバのマルチプログラミングレベルを制御できるように、次のオプションが追加されました。

データベースサーバオプション	sa_server_option 値	説明
「-gn データベースサーバオプション」『SQL Anywhere サーバ データベース管理』	CurrentMultiProgrammingLevel	データベースサーバの初期マルチプログラミングレベルを設定します。
「-gna データベースサーバオプション」『SQL Anywhere サーバ データベース管理』	AutoMultiProgrammingLevel	データベースサーバのマルチプログラミングレベルの動的チューニングをオンまたはオフにします。
「-gnh データベースサーバオプション」『SQL Anywhere サーバ データベース管理』	MaxMultiprogrammingLevel	データベースサーバが同時に実行できるタスクの最大数を設定します。

データベースサーバオプション	sa_server_option 値	説明
「-gnl データベースサーバオプション」『SQL Anywhere サーバ データベース管理』	MinMultiProgrammingLevel	データベースサーバが同時に実行できるタスクの最小数を設定します。
「-gns データベースサーバオプション」『SQL Anywhere サーバ データベース管理』	AutoMultiProgrammingLevelStatistics	マルチプログラミングレベルの自動変更に関する統計をデータベースサーバメッセージログに出力するかどうかを制御します。

SQL Anywhere のマルチプログラミングレベルの詳細については、「マルチプログラミングレベルのデータベースサーバ設定」『SQL Anywhere サーバ データベース管理』を参照してください。

- **即時マテリアライズドビューの外部ジョインのサポート**　定義内に OUTER JOIN があるマテリアライズドビューを即時宣言できるようになりました。「マテリアライズドビューの制限」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **DML 文からの選択**　SELECT 文の FROM 句で DML 文を指定できるようになりました。この機能を使用すると、UPDATE、INSERT、DELETE、または MERGE 文によって変更されたローが移植された派生テーブルに対して SQL クエリを作成し、これらの更新されたローからアプリケーションへ値を返すことができます。

  この機能の最も一般的な用途は、アプリケーションによって変更されたローの値の確認や検査です。以前のバージョンでは、このことは、トリガおよび複数の SQL 文を使用することによって実行できました。「FROM 句」『SQL Anywhere サーバ SQL リファレンス』と「DML 文に対する SELECT」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **全文検索機能による外部事前フィルタライブラリと外部単語区切りライブラリのサポート**　新しい API が追加されたことによって、全文インデックスを作成および更新するときに、外部事前フィルタライブラリと外部単語区切りライブラリに接続できるようになりました。このことは、XML、PDF、Word のようなドキュメントフォーマットを使用し、内容にインデックスを付ける前に、不要なタグとメタデータを削除できることを意味します。サンプルの単語区切りライブラリを使用して、言語またはアプリケーション固有の単語区切りを実行できます。サンプルの事前フィルタライブラリと単語区切りライブラリを利用して独自のライブラリを作成したり、サードパーティのライブラリを使用したりできます。「高度 : 外部単語区切りライブラリと事前フィルタライブラリ」『SQL Anywhere サーバ SQL の使用法』を参照してください。

  データベースサーバを実行しているシステムに Microsoft Office がインストールされている場合は、Word や Excel などの Office ドキュメント用の IFilter を使用できます。サーバに Acrobat Reader がインストールされている場合は、PDF IFilter を使用できます。

  ALTER TEXT CONFIGURATION 文に PREFILTER EXTERNAL NAME 句と TERM BREAKER EXTERNAL NAME 句が追加され、外部ライブラリの名前とロケーションを指定できるようになりました。「ALTER TEXT CONFIGURATION 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

ISYSTEXTCONFIG システムテーブルは、トークン化か事前フィルタリングまたはその両方に使用されるエントリポイントと外部ライブラリに関する情報を格納するように変更されました。具体的には、prefilter カラムのデータ型が LONG VARCHAR になり、外部事前フィルタライブラリのエントリポイントとライブラリ名を格納するように変更されました。外部単語区切りライブラリのエントリポイントとライブラリ名を格納するために、新しい LONG VARCHAR カラムの external_term_breaker が追加されました。「SYSTEXTCONFIG システムビュー」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

外部事前フィルタライブラリと外部単語区切りライブラリを使用するには、データベースをアップグレードする必要があります。

- **チェックサムの強化**　データベースページの書き込みチェックサム (ページがディスクに書き込まれるときにのみ作成されるチェックサム) を作成するかどうかを、データベースバージョンに基づいて、データベースサーバが決定するようになりました。デフォルトでは、バージョン 10 と 11 のデータベースでグローバルチェックサムは無効ですが、バージョン 12 のデータベースでグローバルチェックサムは有効です。バージョン 12 のデータベースサーバで以前のバージョンのデータベースを起動すると、データベースサーバのデフォルトの動作で書き込みチェックサムは有効です。バージョン 12 のデータベースの場合、データベースサーバのデフォルトの動作で書き込みチェックサムは無効です。これは、バージョン 12 のデータベースでは、デフォルトでグローバルチェックサムが有効になっているためです。新しいデータベースでチェックサムがデフォルトで有効146 ページを参照してください。

  データベースまたはデータベースサーバの起動時に START DATABASE 文の CHECKSUM 句または -wc オプションを使用して、データベースサーバの書き込みチェックサムの動作を変更できます。「-wc データベースオプション」『SQL Anywhere サーバ データベース管理』、「-wc データベースサーバオプション」『SQL Anywhere サーバ データベース管理』、「START DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  ALTER DATABASE 文の CHECKSUM 句を使用して、データベースのチェックサムを無効にできます。「ALTER DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

データベース接続

次に、SQL Anywhere バージョン 12.0.0 でのデータベース接続の強化を示します。

- **テンポラリ接続の指定**　テンポラリ接続を使用して、バックアップの実行やデータベースの初期化などの操作が行われます。sa_conn_info システムプロシージャ、sa_conn_list システムプロシージャ、Name 接続プロパティ、ParentConnection 接続プロパティを使用して、テンポラリ接続に関する情報を取得できます。次の項を参照してください。

  - 「テンポラリ接続」『SQL Anywhere サーバ データベース管理』
  - Name 接続プロパティ『SQL Anywhere サーバ データベース管理』
  - ParentConnection 接続プロパティ『SQL Anywhere サーバ データベース管理』
  - 「sa_conn_info システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
  - 「sa_conn_list システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』

- **接続プーリング**　ConnectionPool 接続パラメータは、クライアントの接続プールの動作を制御します。「ConnectionPool (CPOOL) 接続パラメータ」『SQL Anywhere サーバ データベース管理』と「接続プーリング」『SQL Anywhere サーバ データベース管理』を参照してください。

- **ODBC データソースの Escape 接続パラメータのサポート**　デフォルトでは、ODBC ドライバでチルダ (~) がエスケープ文字として使用されますが、一部のアプリケーションではエスケープ文字に円記号 (¥) が使用されることもあります。Escape 接続パラメータを使用して、アプリケーションのエスケープ文字を指定できます。Escape 接続パラメータ (ODBC)『SQL Anywhere サーバ データベース管理』を参照してください。

  また、**[接続]** ウィンドウで Escape 接続パラメータを指定することもできます。

バックアップとリカバリ

次に、SQL Anywhere バージョン 12.0.0 でのバックアップとリカバリの強化を示します。

- **バージョン 12 で作成されたアーカイブバックアップのリストア**　バージョン 12 では、バージョン 11 以前のデータベースサーバで作成したアーカイブバックアップをリストアできません。

- **アーカイブバックアップでの空きページの除去**　アーカイブバックアップでは、デフォルトで空きページがスキップされるため、バックアップが小規模で高速になる可能性があります。トランザクションログファイルには空きページが含まれないため、空きページを除去してもトランザクションログファイルのバックアップには影響がありません。したがって、大規模なトランザクションログファイルがあるデータベースで空きページを除去しても、トランザクションログファイルの規模が小さいデータベースほど利点はありません。BACKUP 文の FREE PAGE ELIMINATION 句を使用するか、**データベースバックアップウィザード**を使用して、この動作を制御できます。「BACKUP 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  空きページの除去をオンにしてバックアップされた、強力に暗号化されているデータベースをリストアする場合は、データベースの暗号化キーを指定してください。RESTORE DATABASE 文の KEY 句を使用するか、**データベースリストアウィザード**を使用して、暗号化キーを指定できます。

セキュリティ

次に、SQL Anywhere バージョン 12.0.0 でのセキュリティの強化を示します。

- **FIPS 認定のアルゴリズムが 64 ビット Windows と 32 ビットおよび 64 ビットの Linux オペレーティングシステムで新規利用可能**　64 ビット Windows と 32 ビットおよび 64 ビットの Linux オペレーティングシステムで FIPS 認定のアルゴリズムを使用できるようになりました。

  FIPS 認定のアルゴリズムをサポートするプラットフォームのリストについては、「サポートされるプラットフォーム」『SQL Anywhere 16 紹介』を参照してください。

FIPS 認定のアルゴリズムの使用方法の詳細については、「FIPS 認定の暗号化テクノロジ」『SQL Anywhere サーバ データベース管理』を参照してください。

データベースセキュリティ

これらの変更点を使用するには、データベースをアップグレードまたは再構築する必要があります。「SQL Anywhere サーバのアップグレード」289 ページを参照してください。

- **SYS_SPATIAL_ADMIN_ROLE グループ**　このグループのメンバーシップが与えられたユーザは、空間参照系と計測単位を作成、変更、または削除できます。

データベースユーティリティ

次に、SQL Anywhere バージョン 12.0.0 でのデータベースユーティリティの強化を示します。

- **データベースのページサイズをキロバイト単位で指定できる**　抽出ユーティリティ (dbxtract)、初期化ユーティリティ (dbinit)、アンロードユーティリティ (dbunload) を使用して、データベースのページサイズをバイトまたはキロバイト単位で指定できるようになりました。以前のバージョンでは、ページサイズをバイト単位でのみ指定できました。次の項を参照してください。

  - 「抽出ユーティリティ (dbxtract)」『SQL Remote』
  - 「初期化ユーティリティ (dbinit)」『SQL Anywhere サーバ データベース管理』
  - 「アンロードユーティリティ (dbunload)」『SQL Anywhere サーバ データベース管理』

- **ファイル非表示ユーティリティ (dbfhide) の強化**　Windows では、ファイル非表示ユーティリティ (dbfhide) で、Windows 暗号化 API を使用した設定ファイルの難読化がサポートされるようになりました。このユーティリティでは、クワイエットモードで実行できるように -q オプションもサポートされています。「ファイル非表示ユーティリティ (dbfhide)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **サーバライセンス取得ユーティリティ (dblic) の強化**　以前のリリースでは、SQL Anywhere のエディションを変更する場合、ソフトウェアをアンインストールしてから、新しいライセンスキーを使用して再インストールする必要がありました。このリリースでは、Sybase iAnywhere から新しいライセンスキーを受け取ったときに、サーバライセンス取得ユーティリティ (dblic) の -k オプションを使用して、SQL Anywhere のエディションを変更できるようになりました。「サーバライセンス取得ユーティリティ (dblic)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **サービスユーティリティ (dbsvc) の強化**　Windows で dbsvc ユーティリティの -rs オプションを使用してサービスの依存性を指定する場合、表示名または実際のサービス名を指定できるようになりました。「Windows 用サービスユーティリティ (dbsvc)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **サポートユーティリティ (dbsupport) の強化**　-ac オプションを使用すると、エラーレポートに含められるコメントを追加できます。-af オプションを使用すると、エラーレポート送信にファイルを含めることができます。「サポートユーティリティ (dbsupport)」『SQL Anywhere サーバ データベース管理』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**　dbunload ユーティリティで次のオプションがサポートされるようになりました。

  - -kd オプションを指定すると、データベースが 1 つの DB 領域にアンロードされます。「アンロードユーティリティ (dbunload)」『SQL Anywhere サーバ データベース管理』を参照してください。
  - -qr オプションを指定すると、テーブルのロードやインデックスの作成が実行されるときに、進行メッセージの作成と表示が行われなくなります。「アンロードユーティリティ (dbunload)」『SQL Anywhere サーバ データベース管理』を参照してください。

データベースオプション

次に、SQL Anywhere バージョン 12.0.0 でのデータベースオプションの強化を示します。

- **uuid_has_hypens オプション**　uuid_has_hyphens では、ユニークな識別子の値を文字列に変換するときのフォーマットを設定します。この機能はバージョン 11 で削除されましたが、また使用できるようになりました。「uuid_has_hyphens オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **blocking_others_timeout オプション**　このオプションでは、現在の接続がロールバックされる前に、現在の接続のローとテーブルのロックを他の接続からブロックできる時間を指定します。このオプションを使用すると、優先度の低いタスクが指定時間を超えて他の接続をブロックすることを防止できます。「blocking_others_timeout オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **http_connection_pool_basesize オプション**　このオプションでは、HTTP によって使用される接続プールのベースラインサイズを指定します。「http_connection_pool_basesize オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **http_connection_pool_timeout オプション**　このオプションでは、未使用の接続を HTTP 接続プールに保持できる時間の上限を指定します。「http_connection_pool_timeout オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **progress_messages オプション**　このオプションは、データベースサーバからクライアントに進行メッセージを送信するかどうかを制御します。ユーティリティデータベースのテンポラリオプションとしてこのオプションを設定できます。「progress_messages オプション」『SQL Anywhere サーバ データベース管理』と「ユーティリティデータベースに使用できる文」『SQL Anywhere サーバ データベース管理』を参照してください。

  Interactive SQL では、デフォルトで進行メッセージが **[メッセージ]** ウィンドウ枠に表示されます。また、**[ツール]** » **[オプション]** » **[SQL Anywhere]** » **[コマンド]** をクリックし、**[進行メッセージを表示]** をクリックして、SQL Anywhere データベースオプションの progress_messages を設定することもできます。このオプションを選択すると、データベースの progress_messages オプションは Formatted に設定されます。クリアすると、データベースの progress_messages オプションは Off に設定されます。デフォルトでは、**[進行メッセージを表示]** オプションは選択されています。

- **timestamp_with_time_zone_format オプション**　このオプションは、TIMESTAMP WITH TIME ZONE 値が文字列に変換される方法を制御します。「timestamp_with_time_zone_format オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **reserved_keywords オプション**　このオプションは、個々のキーワードをオンにします。「reserved_keywords オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **st_geometry_asbinary_format オプション**　ジオメトリからバイナリへの空間データ値の変換方法を制御します。「st_geometry_asbinary_format オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **st_geometry_astext_format オプション**　ジオメトリからテキストへの空間データ値の変換方法を制御します。「st_geometry_astext_format オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **st_geometry_asxml_format オプション**　ジオメトリから XML への空間データ値の変換方法を制御します。「st_geometry_asxml_format オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **st_geometry_describe_type オプション**　空間データの記述方法を制御します。「st_geometry_describe_type オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **st_geometry_on_invalid オプション**　ジオメトリが基本的な検証に失敗した場合の動作を制御します。「st_geometry_on_invalid オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

データベースサーバオプション

次に、SQL Anywhere バージョン 12.0.0 でのデータベースサーバオプションの強化を示します。

- **-xm オプション**　-xm データベースサーバオプションでは、データベースサーバが新しい IP アドレスをチェックする頻度を制御します。コンピュータが新しいネットワークに接続しているか、既存のネットワークから切断しているときに -xm オプションが指定された場合、変更が検出されると、データベースサーバは新しいネットワークで受信を開始します。「-xm データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

プロパティとパフォーマンスモニタの統計値

次に、SQL Anywhere バージョン 12.0.0 でのプロパティとパフォーマンスモニタの統計の強化を示します。

- **接続プロパティ**　このリリースには、次の接続プロパティが追加されています。

  - blocking_others_timeout
  - http_connection_pool_basesize
  - http_connection_pool_timeout
  - ParentConnection
  - Progress
  - progress_messages
  - QueryRowsFetched
  - reserved_keywords
  - st_geometry_asbinary_format
  - st_geometry_astext_format
  - st_geometry_asxml_format
  - st_geometry_describe_type
  - st_geometry_on_invalid
  - timestamp_with_time_zone_format

  これらのプロパティの詳細については、「接続プロパティの一覧」『SQL Anywhere サーバ データベース管理』を参照してください。

- **データベースプロパティ**　このリリースには、次のデータベースプロパティが追加されています。

  - ConnPoolCachedCount
  - ConnPoolHits
  - ConnPoolMisses
  - DriveBus
  - DriveModel
  - HasTornWriteFix
  - HttpConnPoolCachedCount
  - HttpConnPoolHits
  - HttpConnPoolMisses
  - HttpConnPoolSteals
  - LastCheckpointTime
  - MirrorRole
  - QueryRowsFetched
  - SynchronizationSchemaChangeActive
  - WriteChecksums

  これらのプロパティの詳細については、「データベースプロパティの一覧」『SQL Anywhere サーバ データベース管理』を参照してください。

- **データベースサーバプロパティ**　このリリースには、次のデータベースサーバプロパティが追加されています。

  - AutoMultiProgrammingLevel
  - AutoMultiProgrammingLevelStatistics
  - CurrentMultiProgrammingLevel
  - IPAddressMonitorPeriod
  - IsPortableDevice
  - MaxMultiProgrammingLevel
  - MinMultiProgrammingLevel
  - ObjectType
  - ThreadDeadlocksAvoided
  - ThreadDeadlocksReported

  これらのプロパティの詳細については、「データベースサーバプロパティの一覧」『SQL Anywhere サーバ データベース管理』を参照してください。

システムプロシージャとファンクション

次に、SQL Anywhere バージョン 12.0.0 で追加されたシステムプロシージャとファンクションの強化を示します。

- **新しい sa_text_index_vocab_nchar システムプロシージャ**　この新しいシステムプロシージャは、NCHAR テキストインデックスで使用され、sa_text_index_vocab と同じです。「sa_text_index_vocab_nchar システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい sa_copy_cursor_to_temp_table システムプロシージャ**　カーソルの内容をテンポラリテーブルにコピーします。「sa_copy_cursor_to_temp_table システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい sa_describe_cursor システムプロシージャ**　カーソル内のカラムの名前とデータ型を返します。この情報は、さまざまなクライアントプログラミングインタフェースから取得できますが、以前はストアドプロシージャ内でアクセスできませんでした。「sa_describe_cursor システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい sa_install_feature システムプロシージャ**　SQL Anywhere のインストール時にデータベースに存在していなかった追加の機能をインストールします。「sa_install_feature システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい sa_list_cursors システムプロシージャ**　現在の接続についてデータベースサーバが保持している各カーソルの 1 行を含む結果セットを返します。結果セットは、カーソル名、カーソルが現在開いているかどうかを示す値、その他のメタ情報を示します。「sa_list_cursors システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい sa_mirror_server_status システムプロシージャ**　このプロシージャは、プロシージャが実行されているデータベースサーバの下にあるツリー内のコピーノードのステータスをレポートします。「sa_mirror_server_status システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい sa_reserved_words システムプロシージャ**　このプロシージャは、SQL Anywhere の予約語のリストを返します。「sa_reserved_words システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_server_option システムプロシージャの強化**　sa_server_option システムプロシージャに次のオプションが追加されました。

  - AutoMultiProgrammingLevel
  - AutoMultiProgrammingLevelStatistics
  - CurrentMultiProgrammingLevel
  - DropBadStatistics
  - DropUnusedStatistics
  - IPAddressMonitorPeriod
  - MaxMultiProgrammingLevel
  - MinMultiProgrammingLevel
  - StatisticsCleaner

  「sa_server_option システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_table_page_usage システムプロシージャの強化**　progress_messages データベースオプションが Raw または Formatted に設定されている場合、このシステムプロシージャの実行中にデータベースサーバからクライアントに進行状況メッセージが送信されます。「sa_table_page_usage システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **xp_read_file システムプロシージャの強化**　xp_read_file システムプロシージャに、遅延読み込みを指定するための任意のパラメータが含まれるようになりました。この任意のパラメータを指定し、その値が 0 以外の場合、ファイルは読み込まれ、直ちにロックが解除されます。「xp_read_file システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **xp_startsmtp システムプロシージャの強化**　xp_startsmtp システムプロシージャで、4 つの新しいパラメータ trusted_certificates、certificate_company、certificate_unit、certificate_name がサポートされます。これらのパラメータにより、セキュア SMTP を使用してメールを送信できます。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。「xp_startsmtp システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **OPENXML 演算子の強化**　OPENXML 文字列演算子で、USING FILE 句と USING VALUE 句がサポートされるようになりました。これらの句を使用すると、CHAR、NCHAR、BINARY、または LONG BINARY 型のファイルや式、または BLOB 文字列からデータをロードできます。「OPENXML 演算子」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい MEDIAN 関数**　ローのセットの数値式の中央値を計算します。「MEDIAN 関数 [集合]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **HASH 関数の強化**　HASH 関数で CRC32 アルゴリズム型を使用できるようになりました。「HASH 関数 [文字列]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **BIT_OR、BIT_AND、BIT_XOR 関数の強化**　BIT_OR、BIT_AND、BIT_XOR 関数で、整数値とバイナリ値がサポートされるようになりました。また、並列実行プランで BIT_OR、BIT_AND、BIT_XOR 関数を使用できるようになりました。

  次の項を参照してください。

  ○「BIT_OR 関数 [集合]」『SQL Anywhere サーバ SQL リファレンス』
  ○「BIT_AND 関数 [集合]」『SQL Anywhere サーバ SQL リファレンス』
  ○「BIT_XOR 関数 [集合]」『SQL Anywhere サーバ SQL リファレンス』

- **DATEADD、DATEDIFF、DATEPART、DATENAME 関数の強化**　DATEADD、DATEDIFF、DATEPART、DATENAME 関数で、マイクロ秒日付単位と TIMESTAMP WITH TIME ZONE データ型がサポートされるようになりました。次の項を参照してください。

  ○「DATEADD 関数 [日付と時刻]」『SQL Anywhere サーバ SQL リファレンス』
  ○「DATEDIFF 関数 [日付と時刻]」『SQL Anywhere サーバ SQL リファレンス』
  ○「DATEPART 関数 [日付と時刻]」『SQL Anywhere サーバ SQL リファレンス』
  ○「DATENAME 関数 [日付と時刻]」『SQL Anywhere サーバ SQL リファレンス』

- **DB_EXTENDED_PROPERTY 関数の強化**　DB_EXTENDED_PROPERTY 関数を MirrorServerState プロパティおよび MirrorState プロパティとともに使用して、ミラーサーバの同期ステータスと接続ステータスを確認できるようになりました。「DB_EXTENDED_PROPERTY 関数 [システム]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい HTTP_RESPONSE_HEADER 関数**　HTTP 応答ヘッダの値を返します。「HTTP_RESPONSE_HEADER 関数 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **HTTP_VARIABLE 関数の強化**　@BINARY 属性を使用して、x-www-form-urlencoded バイナリデータ値を返せるようになりました。「HTTP_VARIABLE 関数 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい ISENCRYPTED 関数**　文字列が暗号化されているかどうかを確認します。「ISENCRYPTED 関数 [システム]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい NEXT_HTTP_RESPONSE_HEADER 関数**　次の HTTP 応答ヘッダの名前を取得します。「NEXT_HTTP_RESPONSE_HEADER 関数 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい SWITCHOFFSET 関数**　元のタイムゾーンオフセットから指定のタイムゾーンオフセットに変換される TIMESTAMP WITH TIME ZONE 値を返します。「SWITCHOFFSET 関数 [日付と時刻]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい SYSDATETIMEOFFSET 関数**　データベースサーバの現在の日付、時刻、タイムゾーンオフセットを返します。「SYSDATETIMEOFFSET 関数 [日付と時刻]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい TODATETIMEOFFSET 関数**　指定のタイムゾーンオフセットを使用して、TIMESTAMP 値を TIME STAMP WITH TIME ZONE 値に変換します。

「TODATETIMEOFFSET 関数 [日付と時刻]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい COUNT_BIG 関数**　指定されたパラメータに従って、グループのロー数をカウントします。「COUNT_BIG 関数 [集合]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

SQL 文

次に、SQL Anywhere バージョン 12.0.0 での SQL の強化を示します。

- **名前付きインデックスヒントの修飾の改善**　1 つのテーブルの PRIMARY KEY インデックスと FOREIGN KEY インデックスは名前が同じである場合があります。このことが当てはまる場合は、名前付きインデックスヒントを明確に指定できません。名前付きインデックスヒントの指定が拡張され、ヒントが指定されたインデックスを PRIMARY KEY インデックスまたは FOREIGN KEY インデックスとして修飾できるようになりました。INDEX 句、FROM 句『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい IS DISTINCT FROM および IS NOT DISTINCT FROM 探索条件**　IS DISTINCT FROM および IS NOT DISTINCT FROM 探索条件を使用すると、NULL の場合の等価性の評価を制御できます。「IS DISTINCT FROM および IS NOT DISTINCT FROM 探索条件」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **CREATE SYNCHRONIZATION PROFILE 文**　この文は、SQL Anywhere の同期プロファイルを作成します。同期プロファイルによって、SQL Anywhere データベースが Mobile Link サーバと同期する方法を定義します。「CREATE SYNCHRONIZATION PROFILE 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい CREATE INDEX 文の WITH NULL NOT DISTINCT 句**　CREATE INDEX 文に、UNIQUE インデックスの作成時に使用する新しい句 WITH NULLS NOT DISTINCT が追加されました。この句を使用すると、インデックスキー内の NULL がユニークでないことを指定できます。この機能を使用するには、既存のデータベースをアップグレードまたは再構築する必要があります。「CREATE INDEX 文」『SQL Anywhere サーバ SQL リファレンス』の UNIQUE 句の説明を参照してください。

- **逆引用符識別子デリミタ**　逆引用符を識別子デリミタとして使用できるようになりました。「識別子」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ALTER TEXT CONFIGURATION 文の新しい SAVE OPTION VALUES**　この新しい句を使用して、テキスト設定オブジェクトに保存された date_format、time_format、timestamp_format、timestamp_with_time_zone_format オプションの値を変更します。「ALTER TEXT CONFIGURATION 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ALTER SERVER 文と CREATE SERVER 文**　新しい IQODBC サーバクラスと IQJDBC サーバクラスを使用すると、SAP Sybase IQ サーバをリモート接続として指定できます。「ALTER SERVER 文」『SQL Anywhere サーバ SQL リファレンス』と「CREATE SERVER 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい LOCK FEATURE 文**　この文は、他の同時接続でデータベースサーバの機能が使用されないようにします。「LOCK FEATURE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **BEGIN、CREATE VARIABLE、DECLARE 文の強化**　変数宣言に変数の初期開始値を含めることができるようになりました。次の項を参照してください。

  - 「BEGIN 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE VARIABLE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DECLARE 文」『SQL Anywhere サーバ SQL リファレンス』

- **LOAD TABLE 文の強化**　LOAD TABLE 文で load-option 句がサポートされるようになりました。この句を使用すると、データのロード方法を制御できます。また、FORMAT 句の値 XML もサポートされます。「LOAD TABLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい IF NOT EXISTS 句**　新しい IF NOT EXISTS 句を指定し、指定したオブジェクトがすでに存在する場合、変更は行われず、エラーは返されません。次の項を参照してください。

  - 「CREATE INDEX 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE PUBLICATION 文 [Mobile Link] [SQL Remote]」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE SPATIAL REFERENCE SYSTEM 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE TABLE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE TEXT INDEX 文」『SQL Anywhere サーバ SQL リファレンス』

- **新しい IF EXISTS 句**　新しい IF EXISTS 句により、存在しないデータベースオブジェクトの削除を DROP 文で試みてもエラーを返さないように指定できます。次の項を参照してください。

  - 「DROP EVENT 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP FUNCTION 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP INDEX 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP MATERIALIZED VIEW 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP PROCEDURE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP PUBLICATION 文 [Mobile Link] [SQL Remote]」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP SPATIAL REFERENCE SYSTEM 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP SPATIAL UNIT OF MEASURE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP SYNCHRONIZATION PROFILE 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP TABLE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP TRIGGER 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP VARIABLE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP VIEW 文」『SQL Anywhere サーバ SQL リファレンス』

- **CASE 文の強化**　Transact-SQL のプロシージャとバッチで CASE 文がサポートされるようになりました。

- **新しい OR REPLACE 句**　新しい OR REPLACE 句を使用すると、プロファイルまたは変数を作成したり、同じ名前のオブジェクトが存在する場合は、置換したりできます。次の項を参照してください。

  - 「CREATE FUNCTION 文 [外部呼び出し]」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE FUNCTION 文 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE FUNCTION 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE PROCEDURE 文 [外部呼び出し]」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE PROCEDURE 文 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE PROCEDURE 文 [T-SQL]」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE PROCEDURE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE SEQUENCE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE SPATIAL REFERENCE SYSTEM 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE SYNCHRONIZATION PROFILE 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE TRIGGER 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE VARIABLE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE VIEW 文」『SQL Anywhere サーバ SQL リファレンス』

- **SELECT 文の新しい LIMIT 句のサポート**　SELECT 文で新しい LIMIT 句を使用してローカウントとオフセットを指定できるようになりました。「SELECT 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **SET OPTION 文の強化**　SET OPTION 文の構文 1 で、変数の内容を使用したオプションの設定がサポートされるようになりました。「SET OPTION 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい IF [NOT] OF 探索条件**　IF [NOT] OF *type-expression* 探索条件が追加されました。「探索条件」『SQL Anywhere サーバ SQL リファレンス』と「空間データ型の構文」『SQL Anywhere サーバ 空間データサポート』を参照してください。

- **INSERT 文の強化**　INSERT 文には、次のような強化が行われています。「INSERT 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **複数の値のリストのサポート**　INSERT 文に複数の値のリストを含めることが可能になったことによって、一度に複数のローを挿入できるようになりました。次に例を示します。

    ```
    INSERT INTO T (c1,c2,c3)
    VALUES (1,10,100), (2,20,200), (3,30,300);
    ```

  - **すべてのデフォルト値を含むローの挿入のサポート**　SQL Anywhere では、テーブル内のカラムのサブセットに対して指定された値を VALUES 句に含めることができます。指定されていないすべてのカラムには、CREATE TABLE 文の DEFAULT 句、NULL 句、COMPUTE 句によってカラムごとに指定されたデフォルト値が適用されます。以前のバージョンでは、データベースサーバには、テーブル内の少なくとも 1 つのカラムの入力値を指定する必要がありました。

    今回のバージョンでは、次のいずれかの構文拡張によって、すべてのカラムにデフォルトが適用されるようになりました。

INSERT [INTO] *table-name options* **DEFAULT VALUES** ...

INSERT [INTO] *table-name* **( )** *options* **VALUES ( ) [, ( ) ...]**

DEFAULT VALUES または VALUES を指定することは、セマンティック上、次の構文を使用することと同じです。デフォルトエントリの数は、テーブル内のカラムの数と同じになります。

INSERT [INTO] *table-name* **VALUES(default, default, ..., default)**

DEFAULT VALUES 句は SQL/2008 標準の一部ですが、VALUES 句はベンダー拡張です。

- **新しい START SERVER 文**　START SERVER 文が追加されました。この文は、廃止される START ENGINE 文の代わりに使用してください。「START SERVER 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい STOP SERVER 文**　STOP SERVER 文が追加されました。この文は、廃止される STOP ENGINE 文の代わりに使用してください。「STOP SERVER 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ALTER DATABASE 文の新しい CHECKSUM 句**　CHECKSUM OFF 句を使用すると、データベースのグローバルチェックサムを無効にすることができます。新しいバージョン 12 のデータベースでは、デフォルトでグローバルチェックサムが有効になっています。「ALTER DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **DELETE 文の強化**　DELETE 文で相関名がサポートされるようになりました。「DELETE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

データ型

次に、SQL Anywhere バージョン 12.0.0 でのデータ型の強化を示します。

- **TIMESTAMP WITH TIME ZONE データ型**　タイムゾーンオフセット付きの時刻を格納します。「TIMESTAMP WITH TIME ZONE データ型」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  エイリアス DATETIMEOFFSET もサポートされます。「DATETIMEOFFSET データ型」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **CHAR、NCHAR、NVARCHAR、VARCHAR データ型の 32767 文字長のサポート**　NCHAR および NVARCHAR データ型は、最大 32767 文字まで宣言できるようになりました。CHAR および VARCHAR データ型は、文字長セマンティックを使用して最大 32767 文字まで宣言できるようになりました。クライアントの文字セットで、NCHAR、NVARCHAR、CHAR、または VARCHAR データ型の、文字長セマンティックを使用して記述されたバイト長が 32767 を超える場合、これらのデータ型は、LONG NVARCHAR または LONG VARCHAR データ型として記述されます。CHAR および VARCHAR データ型のバイト長が、バイト長セマンティックを使用して記述されている場合は、値をクライアントの文字セットに変換するときに、最大可能文字セット拡張が行われます。最大拡張を含む結果長が 32767 を超える場合、そのデータ型は LONG VARCHAR として記述されます。クライアントでシングルバイト文字

セットが使用されているか、クライアントの文字セットとデータベースの文字セットのエンコードが同じ場合、拡張は行われません。次の項を参照してください。

○「CHAR データ型」『SQL Anywhere サーバ SQL リファレンス』
○「NCHAR データ型」『SQL Anywhere サーバ SQL リファレンス』
○「NVARCHAR データ型」『SQL Anywhere サーバ SQL リファレンス』
○「VARCHAR データ型」『SQL Anywhere サーバ SQL リファレンス』

- **空間データのデータ型**　空間データをサポートするために、複数の新しいデータ型が追加されました。「空間データへのアクセスとそのデータの操作」『SQL Anywhere サーバ 空間データサポート』を参照してください。

プログラミングインタフェース

次に、SQL Anywhere バージョン 12.0.0 でのプログラミングインタフェースの強化を示します。

- **Web サービスパフォーマンスの強化**　プランのキャッシュの効果と潜在的なパフォーマンスの向上による利点を最大化するために、HTTP サービスで自動接続プールがサポートされるようになりました。「HTTP Web サーバで Web サービスアプリケーションを開発する方法」『SQL Anywhere サーバ プログラミング』を参照してください。

- **カスタマイズされた ODBC ドライバ名のサポート**　SQL Anywhere ODBC ドライバの複数の独立したコピーをクライアントシステムに容易にインストールおよび登録するため、ODBC ドライバにカスタマイズされた名前を割り当てることができるようになりました。「ODBC ドライバの設定」『SQL Anywhere サーバ プログラミング』を参照してください。

- **UNIX の ANSI 専用 ODBC ドライバ**　SQLWCHAR を 32 ビット (UTF-32) 数として定義するバージョンの UNIX ODBC ドライバマネージャは、ワイド呼び出しをサポートする SQL Anywhere ODBC ドライバに使用できません。これは、このドライバは 16 ビット SQLWCHAR 用に作成されているためです。このような場合に備えて ANSI 専用バージョンの SQL Anywhere ODBC ドライバが用意されています。このバージョンの ODBC ドライバでは、ワイド呼び出しインタフェース (SQLConnectW など) はサポートされていません。「UNIX 用 UTF-32ODBC ドライバマネージャ」『SQL Anywhere サーバ プログラミング』を参照してください。

- **DBTools : no_reload_status ビットフィールドの an_unload_db 構造への追加**　an_unload_db 構造に新しいビットフィールド no_reload_status が追加されました。no_reload_status を使用して、テーブルとインデックスの進行メッセージの再ロードを抑制できます。an_unload_db 構造体 [データベースツール]『SQL Anywhere サーバ プログラミング』を参照してください。

- **新しい SQL Anywhere TYPE-2 JDBC ドライバ**　SQL Anywhere への接続時に、新しい TYPE-2 JDBC ドライバを JDBC アプリケーションで使用できるようになりました。ODBC の最上部に位置し、ODBC 経由でさまざまなサーバに接続するために使用できる iAnywhere JDBC ドライバとは異なり、SQL Anywhere JDBC ドライバは SQL Anywhere のみに接続し、SQL Anywhere ODBC ドライバのインストールや登録を必要としません。

  SQL Anywhere JDBC ドライバは、JDBC 3.0 ドライバと JDBC 4.0 ドライバに付属しています。

> **注意**
> 現在 iAnywhere JDBC ドライバを使用している場合は、新しい SQL Anywhere JDBC ドライバに変更することを強くおすすめします。

JDBC 4.0 ドライバは自動的にロードおよび登録されます。

バージョン 3.0 の SQL Anywhere JDBC ドライバを使用するには、java.sql.Driver インタフェースを実装する sybase.jdbc.sqlanywhere.IDriver クラスをロードし、JDBC ドライバマネージャを使用して SQL Anywhere JDBC ドライバを登録する必要があります。ロードすると、URL jdbc:sqlanywhere:*connection-string-parameters* を使用して、SQL Anywhere JDBC ドライバを使用する接続を確立できます。*connection-string-parameters* は、SQL Anywhere に接続するときに必要な標準の接続パラメータです。SQL Anywhere JDBC ドライバを使用する場合、アプリケーションで *connection-string-parameters* に DRIVER= または DSN= を指定する必要はなくなりました。「JDBC サポート」『SQL Anywhere サーバ プログラミング』を参照してください。

- **JDBC statement クラスメソッドの新しいサポート**　以前のバージョンでは、PreparedStatement クラスの addBatch メソッドと executeBatch メソッドのみが JDBC ドライバによってサポートされていました。今回のバージョンでは、Statement クラスの addBatch メソッド、clearBatch メソッド、および executeBatch メソッドも、JDBC ドライバによってサポートされるようになりました。Statement クラスの executeBatch メソッドの動作に関する JDBC 仕様は明確でないため、このメソッドを SQL Anywhere JDBC ドライバで使用する場合は、次のことに注意する必要があります。

  1. SQL 例外または結果セットが発生すると、バッチの処理はすぐに停止します。バッチの処理が停止すると、executeBatch メソッドによって BatchUpdateException がスローされます。BatchUpdateException で getUpdateCounts メソッドが呼び出されると、ローカウントの整数配列が返されます。この配列では、バッチエラーが発生する前のカウントセットには有効な負でない更新カウントが含まれますが、バッチエラーが発生した時点以降のすべてのカウントには -1 の値が含まれます。BatchUpdateCount を SQLException にキャストすると、バッチ処理が停止した理由に関する追加の詳細が提供されます。

  2. バッチは、clearBatch メソッドが明示的に呼び出された場合にのみクリアされます。その結果、executeBatch メソッドを繰り返し呼び出すと、バッチが繰り返し再実行されます。また、execute(sql_query) または executeQuery(sql_query) を正しく呼び出すと、指定された SQL クエリが実行されますが、基本となるバッチはクリアされません。したがって、executeBatch メソッド、execute(sql_query)、executeBatch メソッドの順に呼び出すと、バッチ文のセットが実行され、次に、指定された SQL クエリが実行され、次に、再びバッチ文のセットが実行されます。

  「JDBC サポート」『SQL Anywhere サーバ プログラミング』を参照してください。

- **RESUME 文はローカウントまたはロー推定値を返す**　RESUME 文は、プロシージャコールの結果セット内のローの数、または次の結果セットのロー推定値を返すようになりました。以前のバージョンでは、このローカウントは RESUME 文の後に使用できませんでした。RESUME ローカウントを取得するには、クライアントアプリケーションとデバイスサーバの両方が SQL Anywhere 12 である必要があります。

  この変更は、次のクライアント API に影響を及ぼします。

API	影響を受ける関数呼び出しまたは文	ローカウントを返す
Embedded SQL	RESUME 文	SQLCOUNT フィールド
ODBC	SQLMoreResults 関数	SQLRowCount 関数
OLE DB	IMultipleResults::GetResult メソッド	pcRowsAffected 出力パラメータ
PHP	sasql_next_result 関数	sasql_num_rows 関数
PHP	sasql_stmt_next_result 関数	sasql_stmt_num_rows 関数
SQL Anywhere C API	sqlany_get_next_result 関数	sqlany_num_rows 関数
SQL Anywhere Ruby API	sqlany_get_next_result 関数	sqlany_num_rows 関数
SQL Anywhere Python API	nextset メソッド	rowcount 属性

- **CGI/1.1 に必要なすべての変数が PHP $_SERVER 変数に含まれるようになった**　HTTP 要求を実行するときに CGI/1.1 に必要なすべての変数が PHP の $_SERVER 変数に含まれるようになりました。「PHP 外部環境」『SQL Anywhere サーバ プログラミング』を参照してください。

  CGI/1.1 に必要な変数の詳細については、http://www.ietf.org/rfc/rfc3875 を参照してください (RFC3875 の Section 4)。

- **sqlanydb への Python データ型マッピングに対する詳細な制御の追加**　変換コールバックを登録すると、データベースサーバから結果がフェッチされるときに、Python オブジェクトへデータベースの型がマッピングされる方法を制御できます。「データベースタイプの変換」『SQL Anywhere サーバ プログラミング』を参照してください。

- **新しい DBLib fill_sqlda_ex 関数**　fill_sqlda_ex 関数には、fill_sqlda に対する追加機能があります。特に、記述子を埋めるときに、DT_LONGVARCHAR、DT_LONGNVARCHAR、DT_LONGBINARY 型を保持するためにフラグを提供できます。「fill_sqlda_ex 関数」『SQL Anywhere サーバ プログラミング』を参照してください。

- **サイレントインストール機能を選択できるようになった**　Windows では、コマンドラインから SQL Anywhere のコンポーネントや機能をインストール用に選択または省略できます。たとえば、32 ビットの管理ツールを選択するには、AT32=1 を指定します。「SQL Anywhere インストーラを使用したサイレントインストール」『SQL Anywhere サーバ プログラミング』を参照してください。

- **TDS による時刻および日時データの 6 桁精度のサポート**　jConnect 7 以降または Open Client 15.5 以降を使用して SQL Anywhere に接続するアプリケーションでは、時刻および日時データを問い合わせるときに 6 桁の精度を取得できるようになりました。

カタログの変更

次に、SQL Anywhere バージョン 12.0.0 でのカタログの変更を示します。

- **新しい ISYSSEQUENCE システムテーブルと SYSSEQUENCE システムビュー**　ISYSSEQUENCE システムテーブルには、各ユーザ定義シーケンスごとに 1 つのローがあります。「SYSSEQUENCE システムビュー」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい ISYSSEQUENCEPERM システムテーブルと SYSSEQUENCEPERM システムビュー**　ISYSSEQUENCEPERM システムテーブルには、ユーザまたはグループがシーケンスに対して保持する権限が記録されます。「SYSSEQUENCEPERM システムビュー」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ISYSTEXTCONFIG システムテーブル**　ISYSTEXTCONFIG システムテーブルは、トークン化か事前フィルタリングまたはその両方に使用されるエントリポイントと外部ライブラリに関する情報を格納するように変更されました。具体的には、prefilter カラムのデータ型が LONG VARCHAR になり、外部事前フィルタライブラリのエントリポイントとライブラリ名を格納するように変更されました。外部単語区切りライブラリのエントリポイントとライブラリ名を格納するために、新しい LONG VARCHAR カラムの external_term_breaker が追加されました。「SYSTEXTCONFIG システムビュー」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ISYSTABCOL システムテーブル : new base_type_str カラム**　このカラムには、カラムの物理的な型を表す注釈付きの型文字列が格納されます。

- **ISYSPROCPARM システムテーブル : new base_type_str カラム**　このカラムには、パラメータの物理的な型を表す注釈付きの型文字列が格納されます。

- **ISYSUSERTYPE システムテーブル : new base_type_str カラム**　このカラムには、ユーザタイプの物理的な型を表す注釈付きの型文字列が格納されます。

- **ISYSOBJECT システムテーブル : new object_type_str カラム**　このカラムには、SYSOBJECT.object_type 内の値の単語説明 (MAT VIEW など) が格納されます。

  以前のバージョンでは、SYSOBJECT ビューには TINYINT として表されるオブジェクトタイプを示す object_type のみがありました。つまり、整数を単語説明に変換するには、マニュアルまたはビュー定義のいずれかにアクセスする必要がありました。このバージョンでは、object_type_str カラムを問い合わせて、オブジェクトの単語説明を確認できます。

Windows Mobile の強化

次に、SQL Anywhere バージョン 12.0.0 での Windows Mobile に関する強化を示します。

- **Windows Mobile の内部実行スレッドのデフォルトスタックサイズの変更**　Windows Mobile のデフォルトのスタックサイズは 96 KB、最小スタックサイズは 64 KB、最大スタックサイズは 512 KB になりました。「-gss データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

UNIX/Linux の強化

次に、SQL Anywhere バージョン 12.0.0 での UNIX と Linux に関する強化を示します。

- **パフォーマンス統計ユーティリティ (dbstats)**　dbstats ユーティリティは、UNIX コンピュータに関するパフォーマンス統計の値を返します。このユーティリティの動作は、Windows パフォーマンスモニタの動作と似ています。「パフォーマンス統計値ユーティリティ (dbstats) (UNIX)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Linux サービス用に作成される PID ファイル**　SQL Anywhere サービスが Linux で実行されている場合、PID ファイルは */var/run* ディレクトリで作成されます。「Linux 用サービスユーティリティ (dbsvc)」『SQL Anywhere サーバ データベース管理』を参照してください。

パフォーマンスの強化

次に、パフォーマンスの向上以外にユーザに見える変更がないバージョン 12.0.0 でのパフォーマンスの強化を示します。

- **プライマリローの更新時のロックの向上**　プライマリローの非キーカラムの更新と、そのローを参照する外部ローの変更は、相互に干渉しなくなりました。たとえば、サンプルデータベースで、一方の処理が他方を待機する必要なしに、対応する顧客アドレスの更新中に sales_order ローを追加できます。「sa_locks システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』と「ロックの種類」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **独立性レベル 2 および 3 でのロックの減少**　テーブルが共有モード (LOCK TABLE...IN SHARE MODE 文) でロックされている場合、個々のローに対する読み込みロックは取得されなくなりました。これにより、独立性レベル 2 および 3 でのロックのオーバヘッドが減少する可能性があります。

- **インデックスのパフォーマンス向上**　SQL Anywhere 12 では、多数のクラスタ化された連続値をインデックスから削除する場合のパフォーマンスを改善するために、アルゴリズムが強化され、新しいディスク上レイアウトが用意されました。

- **検証パフォーマンスの改善**　SQL Anywhere 12 では、大規模なデータベースの検証を改善するために、大幅な強化が加えられました。

- **要求の優先順位付けの向上**　SQL Anywhere 12 は、I/O バウンド要求の優先順位を上げるように強化されました。これにより、ハードウェアリソースの使用率が向上します。

- **リモートデータアクセスパフォーマンスの向上**　SQL Anywhere 12 では、プロキシテーブルのパフォーマンスの向上を含め、リモートデータアクセスパフォーマンスを向上させるための大幅な強化が行われています。

- **新しいコストモデル**　SQL Anywhere 12 には、最新のハードウェアのクエリ実行コストをより正確に推定する CPU コストモデルが用意されています。この動作により、一部のアクセスプランが変更される可能性があります。

- **ユーザ定義関数に埋め込まれたクエリの強化**　ユーザ定義関数に埋め込まれた SQL クエリをクエリオプティマイザによってインライン化できるようになりました。これにより、呼び出

しごとのプロシージャを介した切り替えが回避され、オプティマイザが文を最適化するときの自由度が大きくなります。

- **式の異なるデータ型への変換の改善**　式を異なるデータ型に変換するときにデータベースサーバが使用する評価ルールが改善されました。新しい評価ルールにより、変換がより効率的に実行されるようになります。

その他

次に、SQL Anywhere バージョン 12.0.0 でのその他の強化を示します。

- **サンプルデータベースのコピーを作成するための新しいスクリプト**　SQL Anywhere インストールディレクトリの *bin32* または *bin64* ディレクトリに *newdemo.bat* ファイルと *newdemo.sh* ファイルが配置されました。これらのファイルを使用して、サンプルデータベースのすべてのデータを含むサンプルデータベースのコピーを作成できます。このスクリプトを使用すると、サンプルデータベースを再作成したり、異なる名前でサンプルデータベースの新しいコピーを作成したりできます。「サンプルデータベースの再作成 (demo.db)」『SQL Anywhere 16 紹介』を参照してください。

- **クエリオプティマイザの新しい選択性推定ソースタイプ**　新しい選択性推定ソースタイプ JOIN が追加されました。この新しいソースタイプは、クエリオプティマイザによって、フォーム T.X = R.X のアトミックな述部の選択性推定に使用されます。「ESTIMATE_SOURCE 関数 [その他]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **オーバーフローエラーの処理の改善**　算術演算 (+, -, *, /, SUM, AVG) は、演算の結果をそのデータ型で表現できないことによってオーバーフローする場合があります。以前のバージョンでは、データ型が INT の式でこのオーバーフローが発生した場合にエラーが返されましたが、その他のすべてのデータ型ではオーバーフローが発生した場合に未定義の値が返されました。今回のバージョンでは、すべてのデータ型に対するすべての算術計算でオーバーフローが検出され、結果をそのデータ型で表現できない場合に、エラーが返されます。

- **ALTER EXTERNAL ENVIRONMENT 文を使用した dbmlsync のロケーションの設定**　同期中に、データベースサーバが使用している PATH 環境変数を使用して dbmlsync 実行プログラムを検出できない場合、ALTER EXTERNAL ENVIRONMENT 文を使用してデータベースサーバに dbmlsync 実行プログラムのロケーションを通知できます。「ALTER EXTERNAL ENVIRONMENT 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **逆引用符のデリミタとしてのサポート**　逆引用符 (`) を使用して、SQL Anywhere の識別子を区切ることができるようになりました。「識別子」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **日本語の Unicode 照合アルゴリズム (UCA) 照合の適合化オプション**　新しい日本語の UCA 照合の適合化オプションを使用できるようになりました。このオプションを使用して、すべてのひらがなとカタカナの文字間におけるレベル 1 の違いを定義できます。この新しい適合化オプションを使用すると、大文字と小文字が区別されない照合において、ひらがなとカタカナの文字を正確に等号比較できます。「照合の適合化オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **サーバメッセージウィンドウと Windows システムトレイアイコンの変更**　データベースサーバメッセージウィンドウのタイトルバーで、パーソナルサーバを実行しているか、ネットワークサーバを実行しているかを表示できるようになりました。Windows システムトレイアイコンのツールチップにもデータベースサーバのタイプが表示されます。また、データベースサーバの **[バージョン情報]** ウィンドウには、実行中の SQL Anywhere のエディションが表示されます。

- **情報ユーティリティ (dbinfo) の強化**　dbinfo ユーティリティで、データベースの CHAR 照合指定、CHAR エンコード、NCHAR 照合指定、または NCHAR エンコードに関する情報が返されるようになりました。「情報ユーティリティ (dbinfo)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **キャッシュ以外の使用に確保されるアドレス領域のサイズの制御**　-ch オプションによって、より多くのアドレス領域がキャッシュ以外の用途に残されるようになり、32 ビットオペレーティングシステムの最大非 AWE キャッシュサイズは縮小されました。「-ch データベースサーバオプション」『SQL Anywhere サーバ データベース管理』と「-chx データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **CPU バウンド要求と比較される I/O バウンドの優先度管理の強化**　ディスクスループットおよびハードウェアリソースの使用を改善するため、データベースサーバによって I/O バウンドの要求が動的に検出され、CPU バウンドタスクに対する優先度が上げられるようになりました。

- **Windows での停電に対する堅牢性の向上**　SQL Anywhere を配備するとき、Windows レジストリエントリを設定して、特定の Intel ストレージドライバを使用するシステムの停電に対する堅牢性を高めることができます。このパラメータを設定しないと、停電発生時にデータが失われたり、データベースが破損したりする可能性があります。配備の一部としてこれらのエントリが必要かどうかを確認するには、「Intel ストレージドライバの堅牢性の向上」『SQL Anywhere サーバ プログラミング』を参照してください。

- **SQL FLAGGER の強化**　SQL FLAGGER で SQL/2008 標準がサポートされるようになりました。「SQL Flagger を使用した SQL 準拠のテスト」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **進行メッセージ**　一部の SQL 文で、データベースサーバからクライアントへの進行メッセージの送信がサポートされるようになりました。新しい progress_messages オプション 130 ページを参照してください。

SQL Anywhere の動作の変更

次に、バージョン 12.0.0 で導入された SQL Anywhere の動作の変更を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

- **新しいデータベースでチェックサムがデフォルトで有効**　新しいデータベースを作成すると、グローバルチェックサムはデフォルトで有効になっています。グローバルチェックサムは、データベースページの読み込みまたは書き込みが行われるたびに計算、検証され、データベースページがディスク上で変更されているかどうかを確認するために使用されます。初期化

ユーティリティの -s[+ | -] オプションを使用するか、CREATE DATABASE 文の CHECKSUM 句を使用して、データベースでグローバルチェックサムを有効にするかどうかを制御できます。次の項を参照してください。

○ チェックサムの強化 127 ページ
○「初期化ユーティリティ (dbinit)」『SQL Anywhere サーバ データベース管理』
○「CREATE DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』
○「START DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』

● **チェックポイントログの変更**　以前のリリースでは、データベースを停止すると、SQL Anywhere はチェックポイントのログを完全にトランケートしました。バージョン 12 では、チェックポイントログの使用履歴がデータベースに格納され、次のセッションのチェックポイントログに適切なサイズを決定するために使用されます。

セッションにわたってチェックポイントログを格納すると、次回データベースを起動するときにチェックポイント割り当てのオーバヘッドを防ぎ、ファイルの拡張時に発生する可能性があるファイルの断片化を回避できます。今回のリリースでは、データベース停止時のデータベースファイルは以前のリリースよりも大きくなりますが、次回データベースを再起動するときにチェックポイントログに追加のスペースが再利用されます。「チェックポイントログ」『SQL Anywhere サーバ データベース管理』を参照してください。

● **ネットワークデータベースサーバがワーカーの最大数を割り当てるようになった**　以前のリリースでは、ネットワークデータベースサーバが起動するとき、データベースサーバのマルチプログラミングレベルに対応する数のワーカーが割り当てられました。バージョン 12 のネットワークサーバでは、サーバの最大マルチプログラミングレベルに対応する数のワーカーが割り当てられます。このようにワーカーの数が増加することによって、ネットワークデータベースサーバでワーカースタック用に必要となるアドレス領域が増加し、データベースサーバが割り当てることができるデータベースキャッシュの量に影響を及ぼすことがあります。

たとえば、32 ビットの Windows プラットフォームでは、デフォルトで、各ワーカーにはスタック用に 1 MB のアドレス領域が必要です。Windows で起動するバージョン 11 のネットワークサーバでは、デフォルトのマルチプログラミングレベルが 20 であるため、ワーカースタック用に 20 MB のアドレス領域が必要となります。ただし、Windows で起動するバージョン 12 のネットワークサーバでは、デフォルトのワーカーの最大数が 80 であるため、80 MB のアドレス領域が必要となります。この変更は、パーソナルサーバや Windows Mobile には影響を及ぼしません。「SQL Anywhere のスレッド化」『SQL Anywhere サーバ データベース管理』を参照してください。

● **古い統計はデータベースの再構築時にロードされない**　バージョン 11 以前のデータベースを再構築する場合、LOAD STATISTICS 文は古い文字列統計の新しいデータベースへのロードを通知しないで省略しますが、文字列統計のバージョンはアップグレードされます。「LOAD STATISTICS 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

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しても、文字列統計のバージョンはアップグレードされません。

● **位置付け DELETE 文と UPDATE 文**　以前のバージョンでは、位置付け UPDATE 文または DELETE 文で TOP 句または FIRST 句を指定できましたが、これらの句は無視されました。

今回のバージョンでは、位置付け UPDATE 文または DELETE 文で TOP または FIRST を指定すると、構文エラーが返されるようになりました。「UPDATE (位置付け) 文 [ESQL] [SP]」『SQL Anywhere サーバ SQL リファレンス』と「DELETE 文 (位置付け) [ESQL] [SP]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **JDBC ドライバは CHAR だけでなく CHAR と VARCHAR をレポートするようになった**　以前のバージョンでは、iAnywhere JDBC ドライバを使用してアプリケーションを接続し、CHAR カラムを含むテーブルまたは結果セットのメタデータを記述しようとすると、メタデータではカラムのタイプ名が CHAR と返されましたが、SQL タイプでは Types.VARCHAR として返されました。JDBC アプリケーションで JDBC ドライバが CHAR カラムの SQL タイプを Types.CHAR と返すようにするには、アプリケーションで odbc_distinguish_char_and_varchar データベースオプションを設定する必要がありました。今回のバージョンでは、新しい SQL Anywhere JDBC ドライバと非推奨の iAnywhere JDBC ドライバで、データベースオプションの設定に関係なく、タイプ CHAR のテーブルと結果セットのカラムに対してタイプ名 CHAR および SQL タイプ Types.CHAR が返され、タイプ VARCHAR のカラムに対してタイプ名 VARCHAR および SQL タイプ Types.VARCHAR が返されるようになりました。

- **CURRENT UTC TIMESTAMP および UTC TIMESTAMP 空間値の変更**　CURRENT UTC TIMESTAMP 空間値、およびデフォルト値 UTC TIMESTAMP 空間値の基本となるデータ型は、TIMESTAMP WITH TIME ZONE になりました。これらの値が TIMESTAMP として定義されたカラムで使用された場合、タイムゾーンオフセットはトランケートされ、動作に目立った違いはありません。ただし、これらの値が CHAR または VARCHAR カラムで使用された場合、オフセットは以前とは異なる値が生成されます。「CURRENT UTC TIMESTAMP 特別値」『SQL Anywhere サーバ SQL リファレンス』と「UTC TIMESTAMP 特別値」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **パーソナルサーバはデフォルトで TCP/IP を起動しない**　デフォルトでは、パーソナルデータベースサーバは共有メモリプロトコルのみを起動します。TCP/IP プロトコルを使用する場合は、パーソナルデータベースサーバを起動するとき、-x サーバオプションを使用して TCP/IP プロトコルを指定する必要があります。「-x データベースサーバオプション」『SQL Anywhere サーバ データベース管理』と「通信プロトコルオプションの考慮事項」『SQL Anywhere サーバ データベース管理』を参照してください。

- **TCP/IP 接続**　TCP/IP を介して接続する場合、HOST 接続パラメータでホスト名が指定されていれば、データベースサーバ名 (ServerName (SERVER) 接続パラメータで指定) は必須ではなくなりました。「Host 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Host (IP) プロトコルオプション**　以前のリリースでは、Host プロトコルオプションによって、データベースサーバを実行する可能性がある 1 つ以上のホストが示されましたが、このことは、クライアントライブラリへのヒントと見なされました。これらのホストでデータベースサーバが見つからない場合、データベースサーバを検出するためにネットワークブロードキャストが実行されました。今回のリリースでは、Host オプションを指定すると、データベースサーバについて指定したホストのみが検索され、デフォルトでは、クライアントはデータベースサーバを検出するためにブロードキャストを実行しません。

  この動作は、DoBroadcast プロトコルオプションを Direct に設定した場合と同じです。HOST プロトコルオプションで指定したコンピュータ以外のコンピュータでデータベースサーバが実行されている場合、データベースサーバは検出されません。前回のリリースと同じように

動作させる場合は、接続文字列で DoBroadcast=All を指定します。「Host (IP) プロトコルオプション (クライアント側のみ)」『SQL Anywhere サーバ データベース管理』と「DoBroadcast (DOBROAD) プロトコルオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **ServerPort (PORT) プロトコルオプション**　以前のリリースでは、PORT プロトコルオプションによって、データベースサーバが受信する可能性がある 1 つ以上のポート番号が示されましたが、このことは、クライアントライブラリへのヒントと見なされました。クライアントライブラリはブロードキャストを送信する場合、PORT プロトコルオプションで指定されたポート番号とデフォルトのポート番号 2638 を使用しました。今回のリリースでは、PORT オプションを指定すると、データベースサーバを検出するために、指定したポートのみが使用されます。「ServerPort (PORT) プロトコルオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **共有メモリ接続での Idle 接続パラメータの尊重**　Idle 接続パラメータでは、接続のアイドルタイムアウト時間を指定します。共有メモリ接続でこの接続パラメータが尊重されるようになりました。共有メモリ接続のデフォルトのアイドルタイムアウトは 0 (アイドルタイムアウトなし) です。「Idle 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **データベースサーバ名は ServerName (Server) 接続パラメータで指定した名前と同じでない場合がある**　以前のリリースでは、Name データベースプロパティの値は ServerName (Server) 接続パラメータで指定した値と常に一致していました。ただし、クライアントがデータベースに接続して新しい NodeType (NODE) 接続パラメータを使用し、異なるデータベースサーバに接続するようにリダイレクトされた場合、名前は一致しません。「NodeType (NODE) 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **代替サーバ名を使用した接続時に一部の操作がサポートされない**　以前のリリースでは、クライアントが代替サーバ名を使用してデータベースに接続した場合、クライアントは同じデータベースサーバ上の他のデータベースを作成、停止、削除できました。代替サーバ名を使用して接続した場合、これらの操作はサポートされなくなりました。

- **データベースミラーリングの動作の変更と廃止予定機能**　-xp オプションによる監視サーバの名前、認証文字列、同期実行モードなどのデータベースミラーリングオプションの定義は廃止される予定です。ただし、ミラーリングシステムでデータベースサーバを使用する場合は、引き続き **-xp on** を指定する必要があります。「-xp データベースオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

  次の SQL 文を使用してデータベースミラーリング設定を定義できるようになりました。

  - 「CREATE MIRROR SERVER 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「ALTER MIRROR SERVER 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「SET MIRROR OPTION 文」『SQL Anywhere サーバ SQL リファレンス』

今回リリースでは、データベースミラーリングについて、次の動作の変更が導入されています。

○ 以前のリリースでは、ミラーサーバのステータス情報ファイルの名前は、デフォルトでサーバ名に基づいた名前でした。今回のリリースでは、ステータス情報ファイルの名前を指定する必要があります。
○ 以前のリリースでは、ミラーサーバに送信された Web サービス要求はプライマリサーバにリダイレクトされました。今回のリリースでは、Web サービス要求は、受け取ったサーバによって処理されます。

今回のリリースでのデータベースミラーリングの強化については、データベースミラーリングの強化 123 ページを参照してください。

- **Embedded SQL カーソル動作の変更**　Embedded SQL カーソルは、READ ONLY のデフォルトになりました。明示的な FOR READ ONLY 句または FOR UPDATE 句は、DECLARE 文ではなく、PREPARE 文で指定することが必要になりました。次の項を参照してください。

  ○「DECLARE CURSOR 文 [ESQL] [SP]」『SQL Anywhere サーバ SQL リファレンス』
  ○「PREPARE 文 [ESQL]」『SQL Anywhere サーバ SQL リファレンス』

- **CREATE OR REPLACE PROCEDURE 文のカーソルは閉じられる**　CREATE OR REPLACE PROCEDURE 文を実行する場合、接続が開いた他のカーソルはすべて閉じられます。「CREATE PROCEDURE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **逆引用符識別子デリミタ**　逆引用符を識別子デリミタとして使用できるようになりました。「識別子」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ロックの動作の変更**　同時実行性を最大化するために、ローのキーとキーではない部分を別々にロックできるようになりました。ローを参照する外部ローの挿入と削除を妨げずに、そのローのキーではないカラムを更新できます。「ロックの仕組み」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **ODBC ドライバの動作の変更**　ODBC 関数 SQLTables を使用すると、SQL_ALL_SCHEMAS 引数で関数を呼び出して、すべてのスキーマ (ユーザ) のリストを取得できます。以前のバージョンでは、この関数によって返されるユーザのリストには、テーブルを所有するユーザのみが含まれていました。新しく初期化したデータベースでは、一部のユーザ、特に DBA ユーザが除外されました。今回のリリースでは、テーブルを所有しないユーザを含む、スキーマの完全なリストが返されます。

- **divide_by_zero_error オプション**　divide_by_zero_error オプションは、バージョン 11 ではサポートされていませんでした。このオプションはバージョン 12 でサポートされます。マテリアライズドビューを使用している場合は、このオプションを On (デフォルト) に設定して、新しいマテリアライズドビューを作成してください。「divide_by_zero_error オプション」『SQL Anywhere サーバ データベース管理』と「マテリアライズドビューの制限」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **PrefetchBuffer (PBUF) 接続パラメータ**　以前のリリースでは、PrefetchBuffer (PBUF) 接続パラメータで指定したローをバッファするためのメモリ量は、すべての接続間で共有されました。今回のリリースでは、この接続パラメータで指定したメモリ量は、各接続で使用できま

す。「PrefetchBuffer (PBUF) 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **削除されたユーザの外部ログインは自動的に削除される**　データベースからユーザを削除すると、そのユーザのすべての外部ログインは自動的に削除されるようになりました。以前のリリースでは、外部ログインを個別に削除する必要がありました。「ユーザの削除 (Sybase Central の場合)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **データベースクリーナーとデータベース検証は同時に実行されなくなった**　以前のリリースでは、データベース検証とデータベースクリーナーを同時に実行し、データベースページへの同時アクセスによるエラーをレポートできました。データベース検証とデータベースクリーナーは、同じデータベース上で同時に実行されなくなりました。検証はデータベースクリーナーの終了を待機し、データベースクリーナーが sa_clean_database を呼び出して開始された場合は、データベースクリーナーが検証の終了を待機します。データベースクリーナーは、テーブル検証またはインデックス検証と同時に実行できます。

- **sa_index_density システムプロシージャ内のカラムに対するデータ型の変更**　密度カラムとスキューカラムのデータ型は、numeric(8,6) から double に変更されました。「sa_index_density システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **DATEDIFF 関数**　以前のリリースでは、DATEDIFF 関数は時間とそれ以下の日付の単位に INTEGER を返しました。今回のリリースでは、DATEDIFF はこれらの日付の単位に BIGINT を返すようになりました。「DATEDIFF 関数 [日付と時刻]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **OPENXML 演算子**　OPENXML 文字列演算子は、dbo データベースユーザによって所有されなくなりました。OPENXML 演算子を使用して **dbo.** で名前を修飾するクエリは **dbo.** を削除するように変更する必要があります。次のような文を実行すると、エラーが返されます。

  ```
  SELECT ... FROM dbo.OPENXML(...)
  ```

  以前のリリースでは、OPENXML 演算子は NCHAR データを CHAR エンコードに変換し、CHAR フォーマットでデータを解析しました。今回のリリースでは、出力に NCHAR カラムがある場合、NCHAR データは NCHAR エンコードで解析されます。

  以前のリリースでは、WITH 句内の xpath 引数には、リテラル文字列のみを指定できました。今回のリリースでは、リテラル文字列と変数を指定できるようになりました。「OPENXML 演算子」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  バージョン 11 以前からアップグレードされたデータベースには、SYS.SYSPROCEDURE システムビューに OPENXML のローが含まれますが、その定義はバージョン 12 のデータベースでは使用できません。演算子を次のように使用すると、構文エラーが返されます。

  ```
  SELECT * FROM dbo."OPENXML"(...)
  ```

- **sa_text_index_vocab システムプロシージャ**　NCHAR テキストインデックスで sa_text_index_vocab を呼び出そうとするとエラーが返されるようになりました。代わりに新しい sa_text_index_vocab_nchar システムプロシージャを使用します。「sa_text_index_vocab_nchar システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

さらに、次の点が変更されました。

○tab_owner パラメータはオプションになりました。

○CALL 文で sa_text_index_vocab を使用できるようになりました。

○プロシージャ内の文で sa_text_index_vocab を使用できるようになりました。

○パラメータ値をホスト変数または式にできるようになりました。

- **Mac OS X のデフォルト照合の変更**　以前のリリースでは、Mac OS X に環境変数がない場合、初期化ユーティリティ (dbinit) はデフォルトの CHAR 文字セットに ISO_8859-1:1987 を使用し、CHAR 照合に ISO1LATIN1 を使用しました (Linux と同じ動作)。今回のリリースでは、UTF-8 と UTF8BIN 照合が選択されます。「推奨文字セットと照合」『SQL Anywhere サーバ データベース管理』を参照してください。

- **予約語**　次に、SQL Anywhere バージョン 12.0.0 でデータベースに追加された予約語を示します。

  ○datetimeoffset

  ○inner

  ○openxml

  ○spatial

  ○treat

  次に、SQL Anywhere バージョン 12.0.0 でデータベースから削除された予約語を示します。

  ○index_lparen

  ○lock

  ○with_cube

  ○with_lparen

  ○syntax_error

  ○with_rollup

- **WITH (*index-hint*) 句の動作変更**　以前は、テーブルのプライマリキーインデックス、外部キーインデックス、または標準インデックスの名前が同じであった場合、オプティマイザによってプライマリーまたは外部キーインデックスの名前に解決されていました。現在では、オプティマイザによって標準のインデックスの名前に解決されるようになりました。それが存在しない場合、オプティマイザはまずプライマリーキーの名前に解決し、次いで外部キーに解決します。WITH table-hint 句、FROM 句『SQL Anywhere サーバ SQL リファレンス』を参照してください。

SQL Anywhere の廃止予定機能とサポート終了機能

注意
廃止予定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、変更される可能性があります。

- **Address Windowing Extensions (AWE) の廃止**　32 ビット Windows の Address Windowing Extensions は使用されなくなりました。サイズの大きいキャッシュが必要な場合は、64 ビット版のオペレーティングシステムで 64 ビット版の SQL Anywhere データベースサーバを使用することをおすすめします。

- **CALL 文**　この文を使用した関数の呼び出しは廃止される予定です。呼び出す関数がある場合は、代入文を使用して関数を呼び出し、その結果を変数に代入することを検討してください。次に例を示します。

  ```
  DECLARE varname INT; SET varname=test( );
  ```

  「CALL 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **STOP ENGINE 文**　STOP ENGINE 文は使用されなくなりました。代わりに STOP SERVER 文を使用します。「STOP SERVER 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **Windows 2000 サポートの削除**　SQL Anywhere は、バージョン 12.0.0 からは Windows 2000 でサポートされなくなりました。

- **再構築ユーティリティの削除**　このリリースでは、SQL Anywhere データベースを再構築する場合に再構築ユーティリティはサポートされなくなりました。アンロードユーティリティ (dbunload) を使用してデータベースを再構築できます。「アンロードユーティリティ (dbunload)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **サポート対象外のデータベースプロパティ**　このリリースでは、次のプロパティが削除されています。

  - CheckpointLogBitmapPagesWritten データベースプロパティ
  - CheckpointLogBitmapSize データベースプロパティ
  - java_main_userid 接続プロパティ
  - QueryRowsBufferFetch 接続プロパティ
  - QueryRowsBufferFetch データベースプロパティ

- **JDBC ベースのサーバクラスの廃止**　次の JDBC ベースのサーバクラスのサポートは廃止されました。

  - ASEJDBC
  - IQJDBC
  - SAJDBC

  ODBC ベースのサーバクラスを使用するようにアプリケーションを更新してください。「ODBC 外部サーバ定義」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **SQL Anywhere Explorer のサポート終了**　Visual Studio 用の SQL Anywhere Explorer と SQL Anywhere ツールバーはサポートされなくなりました。代わりに Microsoft のサーバエクスプローラーを使用します。

- **short int embedded SQL インジケータ変数の廃止**　32 ビット長および 64 ビット長とインジケータを今後使用できるように、Embedded SQL インジケータ変数での short int の使用は推奨されなくなりました。a_sql_len を代わりに使用します。「インジケータ変数」『SQL Anywhere サーバ プログラミング』を参照してください。

- **EngineName (ENG) 接続パラメータの廃止**　EngineName (ENG) 接続パラメータは使用されなくなりました。代わりに、Use the ServerName (Server) 接続パラメータを使用します。ServerName 接続パラメータの短縮形は ENG から Server に変更されました。「ServerName (Server) 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **iAnywhere JDBC ドライバの廃止**　Type 1 iAnywhere JDBC ドライバは使用されなくされました。代わりに Type 2 SQL Anywhere JDBC ドライバを使用します。新しい SQL Anywhere TYPE-2 JDBC ドライバ 140 ページを参照してください。

- **-gu all データベースサーバオプションの廃止**　-gu データベースサーバオプションの値 **all** は使用されなくなりました。「-gu データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **-sm dbsrv12 データベースオプション (旧式)**　-sm データベースオプションは廃止されます。代わりに CREATE MIRROR SERVER 文を使用します。「CREATE MIRROR SERVER 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **SET OPTION 文**　識別子を SET OPTION 文で文字列リテラルではなくオプション値として指定する機能は、使用されなくなりました。

- **サービスユーティリティ (dbsvc ユーティリティ)**　-t オプションの値 Standalone は使用されなくなりました。代わりに -t Personal を使用して、パーソナルデータベースサーバのサービスを作成します。「Linux 用サービスユーティリティ (dbsvc)」『SQL Anywhere サーバ データベース管理』と「Windows 用サービスユーティリティ (dbsvc)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **サーバからの Host (IP) プロトコルオプション**　サーバコマンドでの Host (IP) プロトコルオプションのサポートが削除されました。コマンド「dbeng12 -x tcpip(host=host-name) "%SQLANY12%¥demo.db"」などはサポートされなくなり、エラーが返されます。ただし、クライアント側から CommLinks(LINKS) 接続パラメータで Host (IP) プロトコルを指定することはできます。「Host (IP) プロトコルオプション (クライアント側のみ)」『SQL Anywhere サーバ データベース管理』を参照してください。

注意
Host プロトコルオプションは、Host 接続パラメータとは別のものです。Host プロトコルオプションは、CommLinks 接続パラメータによって使用されます。CommLinks (LINKS) 接続パラメータは、Host および ServerPort (PORT) 以外の TCP/IP オプションを指定する必要がある場合にのみ使用してください。接続文字列に CommLinks と Host の両方は指定できません。「CommLinks (LINKS) 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

ほとんどの場合、HOST 接続パラメータを使用してください。「Host 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

Mobile Link の新機能

次に、バージョン 12.0.0 で導入された Mobile Link の追加機能を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

- **リモートデータベースの集中管理**　リモートデータベースの集中管理を使用して、次のことができます。

  ○リモートデータベースが Mobile Link と同期するときに集中管理します。以前は、dbmlsync オプションを使用してクライアントで設定するか、アプリケーションにハードコードしていました。

  ○リモートデータベースにスキーマの変更を適用します。

  ○特定のリモートデータベースや一般的な同期システムに関する問題を診断します。

  リモートデータベースの集中管理については、「リモートデータベースの集中管理」『Mobile Link サーバ管理』を参照してください。

- **新しい Mobile Link Replay ユーティリティ (mlreplay)**　Mobile Link Replay ユーティリティは、Mobile Link サーバによって記録された Mobile Link プロトコルのリプレイに使用されるツールです。「Mobile Link Replay ユーティリティ (mlreplay)」『Mobile Link サーバ管理』を参照してください。

- **Sybase Central の Mobile Link 12 プラグインの再設計**　Mobile Link プラグインはバージョン 12 で再設計され、リモートデータベースの集中管理がサポートされるようになりました。新しいプラグインには、モデルと管理の 2 つの Mobile Link モードが組み込まれました。今回のリリースでは、Mobile Link プラグインを使用して、統合データベース、グループ、同期モデル、リモートタスクを含む同期プロジェクトを作成できるようになりました。同期プロジェクトに古い同期モデルをインポートできます。「リモートデータベースの集中管理」『Mobile Link サーバ管理』を参照してください。

- **特別値への統合カラムのマッピング**　特別値 (ML ユーザなど) にリモートカラムの代わりに統合カラムをマッピングできるようになりました。

- **ダウンロード削除サブセットのサポート**　ダウンロード削除サブセットを指定できるようになりました。ダウンロード削除サブセットは、デフォルトでダウンロードサブセットと同じです。

- **新しいツリービューを使用したカラムとテーブルの選択**　統合データベースを作成したり更新したりするとき、または既存のリモートデータベースで同期対象のテーブルを選択するときに、カラムとテーブルを選択できるようになりました。この選択は、テーブルとカラムのツリービューで、同期するテーブルやカラムの横にチェックマークを付けることによって簡単に実行できます。

- **空間データ型のサポート**　同期モデルで空間データ型および TIMESTAMP WITH TIME ZONE データ型がサポートされるようになりました。

- **シャドウテーブルにインデックスを用意**　同期モデルによって作成されるシャドウテーブルにインデックスが作成されるようになりました。これにより、シャドウテーブルを使用する download_delete_cursor スクリプトと download_cursor スクリプトを高速化できます。

- **SQL Anywhere モニタで Mobile Link サーバファームと Relay Server ファームをサポート**　今回のリリースでは、SQL Anywhere モニタを使用して、SQL Anywhere データベースと Mobile Link サーバだけでなく Mobile Link サーバファームと Relay Server ファームもモニタリングできるようになりました。「SQL Anywhere モニタ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **サーバファーム用の新しい Mobile Link 監視サーバユーティリティ**　サーバ起動同期について、Mobile Link 監視サーバは、サーバファーム内の 1 つの Mobile Link サーバだけがプライマリサーバとして動作するようにします。「アーキテクチャ」『Mobile Link クイックスタート』を参照してください。

- **サーバファームの強化サポート**　新しいリモート ID ロックロジックを使用すると、Mobile Link 高可用性の同じリモート ID からの冗長な同期を防止できます。-ss オプションを設定する必要がなくなりました。サーバ起動同期を Mobile Link サーバファームと使用する場合は、監視サーバが必要です。

> **注意**
> サーバファームでの Mobile Link サーバの実行は、Mobile Link の高可用性オプションの機能であり、別途ライセンスが必要です。「別途ライセンスが必要なコンポーネント」『SQL Anywhere 16 紹介』を参照してください。

- **TLS 暗号パッケージプログラムのサポートの変更**　Mobile Link サーバおよびクライアントで、RSA と ECC の両方の 256 ビット AES 暗号パッケージプログラムがサポートされるようになりました。また、ECC 暗号パッケージプログラムの RFC 4492 バージョンのサポートも追加されました。

- **動的メモリキャッシュ**　動的メモリキャッシュでは、メモリキャッシュの使用量が増えるため、スワップを防ぐためにより大きいキャッシュサイズが必要になる場合があります。全体的なメモリの使用量はほぼ同じです。

- **新しい統合化 Outbound Enabler**　mlsrv12 の -x オプションの新しい OE プロトコルを使用して、rsoe コマンドで呼び出されるスタンドアロンの Outbound Enabler の代わりに統合化

Outbound Enabler を使用します。「-x mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **Sybase Central 用の新しい Relay Server プラグイン**　Sybase Central 用の新しい Relay Server プラグインを使用すると、バックエンドファームとバックエンドサーバを設定できます。Relay Server プラグインは、Windows と Linux のみでサポートされています。「Sybase Central 用の Relay Server プラグイン」『Relay Server』を参照してください。

- **アップロードおよびダウンロードデータスクリプトの要件**　ml_add_missing_dnld_scripts ストアドプロシージャを使用して、欠落している download_cursor スクリプトか download_delete_cursor スクリプトまたはその両方を修正します。「ml_add_missing_dnld_scripts システムプロシージャ」『Mobile Link サーバ管理』を参照してください。

SQL 文

次に、SQL Anywhere バージョン 12.0.0 での Mobile Link 用の SQL の強化を示します。

- **新しい SYNCHRONIZE 文 [Mobile Link]**　この新しい文を使用すると、SQL Anywhere リモートデータベースを Mobile Link サーバと同期できます。「SYNCHRONIZE 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい START SYNCHRONIZATION SCHEMA CHANGE 文 [Mobile Link]**　この新しい文を使用すると、SQL Anywhere リモートデータベースで Mobile Link の同期スキーマ変更を開始できます。「START SYNCHRONIZATION SCHEMA CHANGE 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい STOP SYNCHRONIZATION SCHEMA CHANGE 文 [Mobile Link]**　この新しい文を使用すると、SQL Anywhere リモートデータベースで Mobile Link の同期スキーマ変更を停止できます。「STOP SYNCHRONIZATION SCHEMA CHANGE 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **CREATE SYNCHRONIZATION SUBSCRIPTION 文 [Mobile Link] の強化**　スクリプトバージョンとスクリプト名を指定できるようになりました。「CREATE SYNCHRONIZATION SUBSCRIPTION 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ALTER SYNCHRONIZATION SUBSCRIPTION 文 [Mobile Link] の強化**　新しい SET SCRIPT VERSION 句を使用して、同期中に使用するスクリプトバージョンを指定できます。RENAME 句を使用してサブスクリプションの名前を変更することもできます。「ALTER SYNCHRONIZATION SUBSCRIPTION 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

統合データベース

次に、SQL Anywhere バージョン 12.0.0 での Mobile Link 用の統合データベースサポートの強化を示します。

- **新しくサポートされる統合データベース**　Mobile Link バージョン 12.0.0 では、次の新しい統合データベースがサポートされます。
  - IBM DB2 LUW 9.7
  - SQL Anywhere 12
- **空間データの同期**　空間データ型の同期が統合データベース SQL Anywhere、Oracle、Microsoft SQL Server、IBM DB2、MySQL に対してサポートされるようになりました。「空間データの同期」『Mobile Link サーバ管理』を参照してください。
- **より長い CHAR カラムのサポート**　Mobile Link サーバで、バイト長が 32767 を超える CHAR、VARCHAR、NCHAR、NVARCHAR リモートデータベースカラムの同期がサポートされるようになりました。
- **TIMESTAMP WITH TIME ZONE データ型を持つカラムの同期のサポート**　Mobile Link サーバで、TIMESTAMP WITH TIME ZONE データ型を持つリモートデータベースカラムの同期がサポートされるようになりました。

iAnywhere Solutions 12 - Oracle ODBC ドライバ

次に、SQL Anywhere バージョン 12.0.0 に導入された iAnywhere Solutions 12 - Oracle ODBC ドライバの強化点を示します。

- **パッケージ化されたプロシージャでの VARRAY のサポート**　iAnywhere Solutions 12 - Oracle ODBC ドライバで、パッケージ化されたプロシージャでの VARRAY の使用がサポートされるようになりました。

Mobile Link サーバ

次に、SQL Anywhere バージョン 12.0.0 での Mobile Link サーバの強化を示します。

mlsrv12 の新機能

- **サーバ名を mlsrv12 に変更**　Mobile Link サーバは mlsrv11 から mlsrv12 に変更されました。
- **mlsrv12 の新しい -cmax オプション**　新しいオプションは、動的キャッシュサイズ決定機能の一部です。サーバのメモリキャッシュの最大サイズを設定します。「-cmax mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。
- **mlsrv12 の新しい -cinit オプション**　新しいオプションは、動的キャッシュサイズ決定機能の一部です。サーバのメモリキャッシュの初期サイズを設定します。「-cinit mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。
- **mlsrv12 の新しい -cmin オプション**　新しいオプションは、動的キャッシュサイズ決定機能の一部です。サーバのメモリキャッシュの最小サイズを設定します。「-cmin mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。
- **mlsrv12 の -x オプションの新しいプロトコルオプション**　mlsrv12 の -x オプションで、新しい oe プロトコルオプションがサポートされるようになりました。このプロトコルオプションを使用すると、Relay Server の使用時に新しい統合化 Outbound Enabler を使用できます。「-x mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **mlsrv12 の -sl java オプションの新しいデフォルト**　デフォルトで、Mobile Link サーバでは、サーバ Java VM が存在する場合、クライアント VM ではなくサーバ Java VM のロードを試みるようになりました。この新しいデフォルトを上書きするには、-sl java オプションに -client を追加して、クライアント VM を明示的に要求します。「-sl java mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **mlsrv12 の新しい -ca オプション**　複数のサーバを使用し、ライトウェイトポーリングなしでサーバ起動同期を行う場合は、Mobile Link 監視サーバを実行するホストの名前を入力します。「-ca mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **mlsrv12 の新しい -ftru オプション**　このオプションは、Mobile Link サーバでファイルアップロードサポートを有効にするために追加されました。このオプションを使用すると、mlfiletransfer ユーティリティによって、アップロードするファイルのルートディレクトリを設定できます。「-ftru mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **mlsrv12 の -ppv オプションの新しいメトリック**　mlsrv12 の -ppv オプションに多数の新しいメトリックが追加されました。「-ppv mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **mlsrv12 の新しい -rrp オプション**　mlsrv12 の新しい -rrp オプションが Mobile Link Replay ユーティリティ (mlreplay) と組み合わせて追加されました。このオプションを指定すると、Mobile Link サーバの起動時に mlreplay ユーティリティが実行され、指定したディレクトリ内の記録されたセッションがすべてリプレイされます。「-rrp mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **mlsrv12 の新しい -rp オプション**　mlsrv12 の新しい -rp オプションが Mobile Link Replay ユーティリティ (mlreplay) と組み合わせて追加されました。このオプションは、mlreplay ユーティリティよるプレイバック用の同期を記録するディレクトリを指定するために使用します。「-rp mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **mlsrv12 の新しい -vR オプション**　mlsrv12 の新しい -vR オプションを使用すると、同期の各ログメッセージ内のリモート ID が表示されます。「-v mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **mlsrv12 の新しい -vU オプション**　mlsrv12 の新しい -vU オプションを使用すると、同期の各ログメッセージ内の Mobile Link ユーザ名が表示されます。「-v mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **mlsrv12 の新しい -vk オプション**　新しいオプションは、動的キャッシュサイズ決定機能の一部です。このオプションでは、キャッシュサイズが増大または縮小すると、ログに 1 行が追加されます。「-v mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **Mobile Link サーバでのメモリ不足パフォーマンスの改善**　メモリ不足状態における動作と堅牢性が改善されました。

- **2 つの新しい競合検出方法**　upload_fetch、upload_fetch_column_conflict、upload_new_row_insert、upload_old_row_insert の各スクリプトを使用して、競合を検出できます。「競合検出」『Mobile Link サーバ管理』を参照してください。

- **新しい ml_add_missing_dnld_scripts 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して、欠落した download_cursor スクリプトと download_delete_cursor スクリプトを追加できます。「ml_add_missing_dnld_scripts システムプロシージャ」『Mobile Link サーバ管理』を参照してください。

Mobile Link サーバの新しい API 機能

- **新しい TimestampWithTimeZone クラスの追加 (Java)**　このクラスは、Timestamp 値のタイムゾーンオフセットを取得および指定できるメソッドを提供します。TimestampWithTimeZone クラス [Mobile Link サーバ Java]『Mobile Link サーバ管理』を参照してください。

- **新しい DateTimeWithTimeZone クラスの追加 (.NET)**　このクラスは、DateTime 値のタイムゾーンオフセットを取得および指定できるメソッドを提供します。DateTimeWithTimeZone クラス [Mobile Link サーバ .NET]『Mobile Link サーバ管理』を参照してください。

- **新しい SpatialUtilities クラスの追加 (.NET および Java)**　このクラスは、空間データを操作する場合に役立つメソッドを提供します。SpatialUtilities クラス [Mobile Link サーバ .NET]『Mobile Link サーバ管理』 (.NET) と SpatialUtilities クラス [Mobile Link サーバ Java]『Mobile Link サーバ管理』 (Java) を参照してください。

Mobile Link の新しいスクリプト機能

- **upload_fetch スクリプトと upload_fetch_column_conflict スクリプトに使用できる新しいシステムパラメータ**　Mobile Link サーバのシステムパラメータ「remote_id」と「username」を upload_fetch スクリプトと upload_fetch_column_conflict スクリプトに使用できるようになりました。

- **authenticate_file_transfer スクリプトに使用できる新しいパラメータ**　authenticate_file_transfer スクリプトに新しいパラメータを使用できます。「authenticate_file_transfer 接続イベント」『Mobile Link サーバ管理』を参照してください。

- **新しい authenticate_file_upload 接続イベント**　ファイルアップロードのサポートに新しい接続イベントを使用できます。「authenticate_file_upload 接続イベント」『Mobile Link サーバ管理』を参照してください。

- **新しい generate_next_last_download_timestamp スクリプト**　次の最終ダウンロード時間を決定する Mobile Link のデフォルトアルゴリズムを抑制する場合は、modify_next_last_download_timestamp の代わりにこのスクリプトを使用します。「generate_next_last_download_timestamp イベント」『Mobile Link サーバ管理』を参照してください。

サーバユーティリティ

次に、SQL Anywhere バージョン 12.0.0 での Mobile Link サーバユーティリティの強化を示します。

- **新しい mlarbiter ユーティリティ**　新しい mlarbiter ユーティリティを使用して、Mobile Link 監視サーバを起動します。サーバ起動同期を Mobile Link サーバファームと使用する場合は、

監視サーバが必要です。「Windows 用 Mobile Link 監視サーバユーティリティ (mlarbiter)」『Mobile Link サーバ管理』を参照してください。

Mobile Link モニタ

次に、SQL Anywhere バージョン 12.0.0 での Mobile Link モニタの強化を示します。

- **グラフウィンドウ枠の新しいエントリ**　Mobile Link モニタのグラフウィンドウ枠に次のエントリが追加されました。OE ワークキュー、Notifier ワークキュー、動的キャッシュワークキュー。「[使用率グラフ]の動作方法」『Mobile Link サーバ管理』を参照してください。

- **新しい [信頼できる証明書ファイル] オプション**　新しい **[信頼できる証明書ファイル]** オプションと **[参照]** ボタンが **[Mobile Link サーバへの接続]** ウィンドウに追加されました。

Mobile Link クライアント

次に、SQL Anywhere バージョン 12.0.0 での Mobile Link クライアントの強化を示します。

- **dbmlsync によるバックグラウンド同期のサポートの強化**　dbmlsync がロックしたデータベースリソースへのアクセスを別の接続が待機している場合、データベースエンジンでリモートデータベースへの dbmlsync 接続を削除 (および dbmlsync のコミットされていない操作をロールバック) できるようになりました。これにより、他の接続は同期の完了を待機しないで先に進むことができます。

  バックグラウンド同期に次のオプションを使用します。

  - 「-bk dbmlsync オプション」『Mobile Link クライアント管理』
  - 「-bkr dbmlsync オプション」『Mobile Link クライアント管理』
  - 「Mobile Link 同期プロファイル」『Mobile Link クライアント管理』の Background オプションと BackgroundRetry オプション

- **dbmlsync での名前付きサブスクリプションのサポート**　dbmlsync の新しい -s オプションを使用すると、同期させるサブスクリプションの名前を指定できます。「-s dbmlsync オプション」『Mobile Link クライアント管理』を参照してください。

- **HTTP 応答バッファリング**　新しいネットワークプロトコルオプション http_buffer_responses を使用すると、Mobile Link からの HTTP パケットを、ネットワークから読み込むと同時に処理するのではなく、処理前に、中間バッファに完全にストリーミングすることができます。「http_buffer_responses」『Mobile Link クライアント管理』を参照してください。

- **クライアント側の証明書を使用したサードパーティのサーバとプロキシへの Mobile Link クライアントの認証**　この機能をサポートするために、identity 同期パラメータと identity_password 同期パラメータが追加されました。

- **リモートデータベースでのスキーマ変更の実装に同期が不要になった (SQL Anywhere クライアントのみ)**　START SYNCHRONIZATION SCHEMA CHANGE 文と STOP SYNCHRONIZATION SCHEMA CHANGE 文を使用してスキーマの変更を区切ることによ

り、リモートデータベースでのスキーマの変更を簡略化できます。次の項を参照してください。

○「START SYNCHRONIZATION SCHEMA CHANGE 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』
○「STOP SYNCHRONIZATION SCHEMA CHANGE 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』

CREATE SYNCHRONIZATION SUBSCRIPTION 文または ALTER SYNCHRONIZATION SUBSCRIPTION 文を使用して、各同期サブスクリプションのスクリプトバージョンを指定します。次の項を参照してください。

○「CREATE SYNCHRONIZATION SUBSCRIPTION 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』
○「ALTER SYNCHRONIZATION SUBSCRIPTION 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』

- **Ultra Light の新しい mlfileupload メソッド**　Ultra Light クライアントの MLFileUpload メソッドが追加されました。

- **dbmlsync オプションウィンドウの変更**　dbmlsync オプションウィンドウは、次のように変更されました。

  ○ **[リモートプログレスに対してリトライする]**、**[リモートのオフセットが後]**、**[リモートのオフセットが前]**、**[競合する接続の削除]**、**[サイトスクリプト]**、**[コマンドラインヘルプ]** は削除されました。
  ○ **[パブリケーション]** オプションは **[サブスクリプション]** オプションに置き換えられました。

新しい dbmlsync オプション

- **dbmlsync の新しい -s オプション**　dbmlsync の新しい -s オプションを使用して、同期させるサブスクリプションの名前を指定します。「-s dbmlsync オプション」『Mobile Link クライアント管理』を参照してください。

- **dbmlsync の新しい -bk オプション**　このオプションは、バックグラウンド同期を有効にします。「-bk dbmlsync オプション」『Mobile Link クライアント管理』を参照してください。

- **dbmlsync の新しい -bkr オプション**　このオプションは、バックグラウンド同期が中断された後の dbmlsync の動作を制御します。「-bkr dbmlsync オプション」『Mobile Link クライアント管理』を参照してください。

- **dbmlsync の新しい -ci オプション**　このオプションは、新しい動的キャッシュサイズ決定機能の一部です。このオプションを使用して、データの同期時に dbmlsync で使用するキャッシュの初期サイズを設定します。「-ci dbmlsync オプション」『Mobile Link クライアント管理』を参照してください。

- **dbmlsync の新しい -cl オプション**　このオプションは、新しい動的キャッシュサイズ決定機能の一部です。このオプションを使用して、dbmlsync キャッシュファイルを縮小する最小サイズを設定します。「-cl dbmlsync オプション」『Mobile Link クライアント管理』を参照してください。

- **dbmlsync の新しい -cm オプション**　このオプションは、新しい動的キャッシュサイズ決定機能の一部です。このオプションを使用して、dbmlsync キャッシュファイルの最大サイズ制限を設定します。「-cm dbmlsync オプション」『Mobile Link クライアント管理』を参照してください。

- **dbmlsync の新しい BufferDownload 拡張オプション**　このオプションは、新しい動的キャッシュサイズ決定機能の一部です。このオプションでは、Mobile Link サーバからのダウンロードを、リモートデータベースに適用する前にすべてキャッシュに読み込むかどうかを指定します。「BufferDownload (bd) 拡張オプション」『Mobile Link クライアント管理』を参照してください。

新しい dbmlsync C++ API オブジェクト

- **CancelSync メソッド**　この方法を使用すると、Mobile Link クライアントで同期をキャンセルできます。DbmlsyncClient.CancelSync メソッド [Dbmlsync C++]『Mobile Link クライアント管理』を参照してください。

- **DBSC_CancelRet 列挙**　この列挙は、同期のキャンセル試行の結果を示します。DBSC_CancelRet 列挙 [Dbmlsync C++]『Mobile Link クライアント管理』を参照してください。

- **DBSC_ERR_ACTIVE_SYNC_NOT_CANCELED メンバー**　このメンバーは、同期がアクティブであるために、サーバが同期要求をキャンセルできなかったことを示します。DBSC_ErrorType 列挙 [Dbmlsync C++]『Mobile Link クライアント管理』を参照してください。

- **DBSC_ERR_DEAD_SERVER メンバー**　このメンバーは、起動時に dbmlsync サーバにエラーが発生し、シャットダウンされたことを示します。DBSC_ErrorType 列挙 [Dbmlsync C++]『Mobile Link クライアント管理』を参照してください。

- **enable status プロパティ**　このプロパティが有効な場合、3 つの新しいイベント (DBSC_EVENTTYPE_ML_CONNECT、DBSC_EVENTTYPE_UPLOAD_COMMITTED、DBSC_EVENTTYPE_DOWNLOAD_COMMITTED) が Mobile Link クライアントに送信されます。DBSC_EventType 列挙 [Dbmlsync C++]『Mobile Link クライアント管理』、DbmlsyncClient.SetProperty メソッド [Dbmlsync C++]『Mobile Link クライアント管理』、DbmlsyncClient.GetProperty メソッド [Dbmlsync C++]『Mobile Link クライアント管理』を参照してください。

新しい dbmlsync .NET API オブジェクト

- **CancelSync メソッド**　この方法を使用すると、Mobile Link クライアントで同期をキャンセルできます。DbmlsyncClient.CancelSync メソッド [Dbmlsync .NET]『Mobile Link クライアント管理』を参照してください。

- **DBSC_CancelRet 列挙**　この列挙は、同期のキャンセル試行の結果を示します。DBSC_CancelRet 列挙 [Dbmlsync .NET]『Mobile Link クライアント管理』を参照してください。

- **DBSC_ERR_ACTIVE_SYNC_NOT_CANCELED メンバー**　このメンバーは、同期がアクティブである場合、サーバが同期要求をキャンセルできなかったかどうかを示します。DBSC_ErrorType 列挙 [Dbmlsync .NET]『Mobile Link クライアント管理』を参照してください。

- **DBSC_ERR_DEAD_SERVER メンバー**　このメンバーは、起動時に dbmlsync サーバにエラーが発生し、シャットダウンされたことを示します。DBSC_ErrorType 列挙 [Dbmlsync .NET]『Mobile Link クライアント管理』を参照してください。

- **"enable status" プロパティ**　このプロパティが有効な場合、3 つの新しいイベント (DBSC_EVENTTYPE_ML_CONNECT、DBSC_EVENTTYPE_UPLOAD_COMMITTED、DBSC_EVENTTYPE_DOWNLOAD_COMMITTED) が Mobile Link クライアントに送信されます。DBSC_EventType 列挙 [Dbmlsync .NET]『Mobile Link クライアント管理』、DbmlsyncClient.SetProperty メソッド [Dbmlsync .NET]『Mobile Link クライアント管理』、DbmlsyncClient.GetProperty メソッド [Dbmlsync .NET]『Mobile Link クライアント管理』を参照してください。

新しいクライアントユーティリティ機能

- **mlfiletransfer を使用したファイルのアップロード**　mlfiletransfer ユーティリティを使用してファイルをアップロードできるようになりました。「Mobile Link ファイル転送ユーティリティ (mlfiletransfer)」『Mobile Link クライアント管理』を参照してください。

- **mlfiletransfer ユーティリティの新しい -i オプション**　再開機能を無効にするために、mlfiletransfer に -i オプションが追加されました。「Mobile Link ファイル転送ユーティリティ (mlfiletransfer)」『Mobile Link クライアント管理』を参照してください。

Relay Server

このリリースでは、次の Relay Server 機能が追加されています。

- **SQL Anywhere モニタのサポート**　SQL Anywhere モニタを使用して、Relay Server リソースをモニタリングできるようになりました。「SQL Anywhere モニタ」『SQL Anywhere サーバデータベース管理』を参照してください。

- **集中管理のサポート**　Relay Server は、Sybase Central を使用して集中管理をサポートします。

- **Mac OS での RSOE サポート**　Mac OS で Relay Server Outbound Enabler がサポートされるようになりました。サポートされるプラットフォームの詳細については、http://www.sybase.com/detail?id=1061806 を参照してください。

- **Windows IIS 7 と 7.5 のサポート**　Relay Server は、Windows Server 2008 上の IIS 7 と Windows Server 2008 R2 上の IIS 7.5 でサポートされるようになりました。サポートされるプラットフォームの詳細については、http://www.sybase.com/detail?id=1061806 を参照してください。

- **新しい active_cookie オプション**　Relay Server 設定のバックエンドファームセクションに active_cookie オプションが追加されました。「バックエンドファームセクションのプロパティ」『Relay Server』を参照してください。

- **新しい active_header オプション**　Relay Server 設定のバックエンドファームセクションに active_header オプションが追加されました。「バックエンドファームセクションのプロパティ」『Relay Server』を参照してください。

- **Relay Server トラブルシューティングの向上**　トラブルシューティングを向上させるため、Relay Server に、ローカライズされたエラーメッセージを含む標準化された Relay Server エラーコード、システムエラーコードを含む選択的エラーメッセージ、埋め込みのシステムエラーメッセージが用意されました。「Relay Server のエラーメッセージ」『エラーメッセージ』を参照してください。

- **バックエンド HTTP サーバとしての SQL Anywhere データベースのサポート**　Relay Server で、SQL Anywhere フェイルオーバと読み込み専用のスケールアウトがサポートされるようになりました。「SQL Anywhere Web サービスの高可用性とスケールアウトソリューション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Outbound Enabler 動的応答バッファサイズ決定**　動的応答バッファサイズ決定によって、Outbound Enabler のメモリオーバヘッドが大幅に減少します。

- **Outbound Enabler ユーザインタフェースの改善**　ウィンドウタイトル内のインスタンス識別子、systray ヒントテキストと systray メニューがすべて Outbound Enabler ユーザインタフェースに追加され、Outbound Enabler ステータスが systray ヒントテキストに追加されました。

- **共有メモリのより効率的な使用**　Relay Server では共有メモリ (shared_mem 設定オプションによって設定) がより効率的に使用されます。比較的読み込みが遅いクライアントが、サイズの大きい応答を伴う HTTP 要求を実行する配備では、Relay Server を非常に少ない共有メモリで実行できます。

- **Relay Server Outbound Enabler による HTTP を使用した定期的なバックエンドサーバステータス要求の新規サポート**　RSOE が強化され、HTTP を使用した定期的なバックエンドサーバステータス要求がサポートされるようになりました。バックエンドサーバステータス要求は活性 ping の代替であり、バックエンドサーバがクライアント要求を受け入れることができるかどうかを確認する場合に使用できます。rsoe -cs オプションの一部として指定する新しい status_url パラメータを使用すると、定期的なバックエンドサーバステータス要求を有効にできます。

Mobile Link の動作の変更

次に、バージョン 12.0.0 で導入された Mobile Link の動作の変更を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

Mobile Link サーバの変更

- **Mobile Link サーバで *log4j.jar* が不要になった**　*log4j.jar* ファイルは Mobile Link サーバで不要になり、Mobile Link サーバに配備されなくなりました。*log4j.jar* が必要な場合は、jar の独自のバージョンをインストールし、クラスパスに格納する必要があります。

- **-cn オプションの新しい動作**　mlsrv12 の -cn オプションでは、データベースワーカスレッドに使用されるデータベース接続の最大数が設定されます。SQL Anywhere 12 より以前のバージョンでは、mlsrv12 の -cn オプションでデータベース接続の最大数が設定されました。「-cn mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **-sl dnet オプションの新しい動作**　以前のバージョンでは、Mobile Link サーバは、.NET スクリプトの使用時にデフォルトでワークステーション CLR をロードしました。このリリースでは、サーバ CLR をロードするようになりました。以前の動作をリストアするには、mlsrv12 の -sl dnet オプションに **-clrFlavor=wks** を追加します。「-sl dnet mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **Mobile Link サーバで V_$TRANSACTION の代わりに GV_$TRANSACTION Oracle システムビューを使用**　Mobile Link サーバで使用される Oracle アカウントに、V_$TRANSACTION システムビューではなく GV_$TRANSACTION Oracle システムビューのパーミッションが必要になりました。「Oracle 統合データベース」『Mobile Link サーバ管理』を参照してください。

- **バージョン 6.0.x のアップグレードスクリプトの削除**　6.0.x の Mobile Link アップグレードスクリプトは削除されました。このアップグレードが必要な場合は、テクニカルサポート (http://www.sybase.com/support) に連絡してください。

- **ml_add_column システムプロシージャは不要**　バージョン 12 以降のクライアントでは、名前付きパラメータ内のカラム名を使用する場合、ml_add_column システムプロシージャは不要になりました。デフォルトで、追加設定しなくてもカラム名を直接参照できるようになりました。「ml_add_column システムプロシージャ (廃止予定)」『Mobile Link サーバ管理』を参照してください。

- **Java と .NET サーバ API の新しい BIGINT データ型サポート**　SQL データ型の BIGINT は Java データ型と .NET データ型の LONG にマッピングされるようになりました。「SQL データ型と Java データ型」『Mobile Link サーバ管理』と「SQL データ型と .NET データ型」『Mobile Link サーバ管理』を参照してください。

- **データスクリプトが必要**　誤ってデータスクリプトが作成されないことによってデータが消失するリスクを減らすため、Mobile Link サーバでは、次のイベントについて、無視されたスクリプトか有効なスクリプトのいずれかが必要になりました。「無視されたスクリプト」『Mobile Link サーバ管理』を参照してください。

  - **upload_insert**　挿入されたローがリモートからアップロードされ、かつ、handle_UploadData 接続スクリプトが定義されていない場合。

  - **upload_update**　更新されたローがリモートからアップロードされ、かつ、handle_UploadData 接続スクリプトが定義されていない場合。

  - **upload_delete**　削除されたローがリモートからアップロードされ、かつ、handle_UploadData 接続スクリプトが定義されていない場合。

  - **download_cursor と download_delete_cursor**　handle_DownloadData 接続スクリプトが定義されておらず、かつ、同期がアップロード専用でない場合。

  バージョン 12 にアップグレードする場合は、ml_add_missing_dnld_scripts ストアドプロシージャを使用して無視されたスクリプトを追加すると、ダウンロードスクリプトの欠如によるエラーを回避できます。「ml_add_missing_dnld_scripts システムプロシージャ」『Mobile Link サーバ管理』を参照してください。

Mobile Link クライアントの変更

次に、バージョン 12.0.0 で導入された Mobile Link クライアントの動作の変更を示します。

- **バージョン 12 の Mobile Link 同期クライアントはデフォルトでカラム名を送信**　バージョン 12 では、Mobile Link 同期クライアントはすべてデフォルトで Mobile Link サーバにカラム名を送信します。つまり、ほとんどの場合、Mobile Link スクリプトの名前付きパラメータで使用するカラム名と順序付けを定義するときに ml_add_column システムストアドプロシージャを使用する必要がなくなりました。「ml_add_column システムプロシージャ (廃止予定)」『Mobile Link サーバ管理』を参照してください。

- **強化された TLS セッション再ネゴシエーションのサポート**　Mobile Link クライアントで、脆弱なサードパーティのサーバを保護する新しい TLS 拡張がサポートされるようになりました。

- **MLFileTransfer メソッドの名前変更**　Ultra Light クライアントの MLFileTransfer メソッドは、MLFileUpload メソッドと MLFileDownload メソッドに分割されました。

- **-df の -lf への名前変更**　mlfiletransfer ユーティリティの -df オプションは -lf に名前変更され、宛先ファイルではなくローカルファイルを参照するようになりました。

- **-dp の -lp への名前変更**　mlfiletransfer ユーティリティの -dp オプションは -lp に名前変更され、宛先パスではなくローカルパスを参照するようになりました。

- **UPLD_ERR_USERID_ALREADY_IN_USE の変更**　Dbmlsync は sp_hook_dbmlsync_upload_end イベントフックの障害の原因として UPLD_ERR_USERID_ALREADY_IN_USE を返さなくなりました。代わりに、値 UPLD_ERR_REMOTE_ID_ALREADY_IN_USE が返されます。

- **Ultra Light クライアントでの Palm のサポート終了**　Palm OS のサポートは終了しました。Palm を使用する場合は、引き続き SQL Anywhere 11 を使用してください。

Sybase Central の Mobile Link プラグインの変更

次に、バージョン 12.0.0 で導入された Sybase Central の Mobile Link プラグインの動作の変更を示します。

- **同期モデルのマッピングエディタの変更**

  ○同期モデルのテーブルマッピングエディタとカラムマッピングエディタでは、左側に統合データベース、右側にリモートデータベースが表示されるようになりました。「テーブルとカラムのマッピング」『Mobile Link クイックスタート』を参照してください。

  ○テーブルマッピングエディタには、同期される統合データベースのみが表示されます。テーブルマッピングの方向を **[同期しない]** に変更することは、テーブルマッピングを削除し、モデルの保存時に、データベーススキーマを変更しないでそのローをエディタから削除することと同じです。

  ○同期されていない統合テーブルのテーブルマッピングを追加し、必要に応じてリモートスキーマにテーブルを追加するには、**[テーブルマッピング新規作成]** コマンドを使用します。

[スキーマの更新] コマンドを使用して、モデルの統合スキーマまたはリモートスキーマを変更することもできます。

○テーブルマッピングとカラムマッピングの右側の選択に、ドロップダウンリストの代わりにポップアップメニューが使用されるようになりました。メニュー内のすべての選択肢が適合しない場合、オプションのウィンドウ (省略記号ボタンでアクセス) が表示されます。テーブルマッピングでは、同期されていないリモートテーブルのみが表示されます。カラムマッピングでは、リモートテーブル内の同期されていないカラムが、値マッピングのオプションおよびカラムを同期しないオプションとともに表示されます。

○テーブルマッピングの同期オプションを変更する場合、下ウィンドウ枠が、対応するサブオプションのタブに自動的に切り替わります。

- **VARBIT カラムと LONG VARBIT カラム**　リモートスキーマを使用する同期モデルを新しい Ultra Light リモートデータベースに展開する場合、VARBIT カラムと LONG VARBIT カラムの代わりに、VARCHAR カラムと LONG VARCHAR カラムがそれぞれ使用されます。

- **文デリミタとしての GO の使用**　SQL Anywhere データベースと UltraLight データベースでは、同期モデルで文のデリミタとしてセミコロンの代わりに GO が使用されるようになりました。これにより、同期モデルで生成された SQL をリモートタスクの集中管理で使用できるようになりました。

Mobile Link の廃止予定機能とサポート終了機能

注意
廃止予定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、変更される可能性があります。

- **IBM DB2 メインフレームはバージョン 12.0.0 以降でサポート終了**　バージョン 12.0.0 では、IBM DB2 メインフレームは統合データベースとしてサポートされていません。ただし、Mobile Link では、DB2 LUW (Linux、UNIX、Windows) が統合データベースとして引き続きサポートされています。

- **Adaptive Server Enterprise 12.5.x のサポート終了**　Adaptive Server Enterprise 12.5.x は、Mobile Link バージョン 12.0.0 ではサポートされなくなりました。

- **IBM DB2 LUW 8.2 のサポート終了**　IBM DB2 LUW 8.2 は、Mobile Link バージョン 12.0.0 ではサポートされなくなりました。

- **mlsrv12 の -xo オプションの削除**　バージョン 10 より前のクライアントはサポートされなくなりました。

- **mlsrv12 の -f オプションの削除**　-zf mlsrv12 オプションを使用すると、各同期の始めに Mobile Link サーバがスクリプトの変更をチェックするように指定できます。「-zf mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **mlsrv12 の -nba オプションの削除**　ブロッキングダウンロード確認はサポートされなくなりました。リモートデータベースがダウンロード確認を要求する場合、Mobile Link サーバは常に非ブロッキング確認を使用します。

- **mlsrv12 の -fr オプションの削除**　mlsrv12 の -fr オプションはサポートされなくなりました。(データ消失の原因となる可能性がある) スクリプトを無視する場合は、スクリプトを --{ml_ignore} として定義します。

- **SQL を返す Java データスクリプトと .NET データスクリプトは廃止予定**　Java および .NET スクリプトロジックの、Mobile Link サーバによって SQL スクリプトとして解釈される文字列を返す機能は、データスクリプトで廃止される予定です。統合データベースでスクリプトによって変更を加える場合は、Java または .NET から直接変更を加えます。

  次の項を参照してください。

  ○「SQL を返す Java と .NET のデータスクリプト (削除)」『Mobile Link サーバ管理』
  ○「ダイレクトローハンドリング」『Mobile Link サーバ管理』
  ○「データスクリプト」『Mobile Link サーバ管理』

- **ダウンロードエラーフックの削除**　次の dbmlsync フックはバージョン 12 で削除されました。sp_hook_dbmlsync_download_com_error、sp_hook_dbmlsync_download_fatal_sql_error、sp_hook_dbmlsync_download_sql_error。

- **Mobile Link ファイル転送ユーティリティの -f オプションの削除**　mlfiletransfer ユーティリティの -f オプションはサポートされなくなりました。

- **Mobile Link ファイル転送ユーティリティの -r オプションの削除**　mlfiletransfer ユーティリティの -r オプションはサポートされなくなりました。

- **dbmlsync の Memory (mem) 拡張オプションと DownloadBufferSize (dbs) 拡張オプションの削除**　dbmlsync の Memory (mem) 拡張オプションと DownloadBufferSize (dbs) 拡張オプションはサポートされなくなりました。代わりに、CacheMin オプション、CacheInit オプション、CacheMax オプションを使用します。

- **SQL パススルーの dbmlsync サポートの削除**　SQL パススルー機能は、Mobile Link クライアントでサポートされなくなりました。リモートデータベース機能の集中管理に置き換えられました。「リモートデータベースの集中管理」『Mobile Link サーバ管理』を参照してください。

- **mlsrv12 の -ss オプションは不要**　12 より以前のバージョンでは、mlsrv12 の -ss オプションは、Mobile Link サーバをサーバファーム環境で実行できるようにするために使用されました。冗長な同期を防止する新しいリモート ID ロックロジックによって、-ss オプションは、Mobile Link サーバをサーバファームで実行する場合に必要でなくなったため、削除されました。サーバ起動同期を Mobile Link サーバファームと使用する場合は、監視サーバが必要です。

  注意
  サーバファームでの Mobile Link サーバの実行は、Mobile Link の高可用性オプションの機能であり、別途ライセンスが必要です。「別途ライセンスが必要なコンポーネント」『SQL Anywhere 16 紹介』を参照してください。

- **Mobile Link リダイレクタの削除**　リダイレクタは使用できなくなりました。代わりに Relay Server を使用します。「Relay Server の概要」『Relay Server』を参照してください。

- **スクリプトバージョンの使用の推奨事項**　ScriptVersion 拡張オプションを使用しないことを強くおすすめします。代わりに、CREATE SYNCHRONIZATION SUBSCRIPTION 文と ALTER SYNCHRONIZATION SUBSCRIPTION 文で SCRIPT VERSION 句を使用してください。「CREATE SYNCHRONIZATION SUBSCRIPTION 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』と「ALTER SYNCHRONIZATION SUBSCRIPTION 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **dbmlsync の -n オプションは廃止予定**　このオプションは廃止される予定です。代わりに -s dbmlsync オプションを使用することをおすすめします。「-s dbmlsync オプション」『Mobile Link クライアント管理』を参照してください。

- **dbmlsync の -u オプションは廃止予定**　このオプションは廃止される予定です。代わりに -s dbmlsync オプションを使用することをおすすめします。「-s dbmlsync オプション」『Mobile Link クライアント管理』を参照してください。

- **Publication 同期プロファイルオプションは廃止予定**　このオプションは廃止される予定です。代わりに -s dbmlsync オプションを使用することをおすすめします。「-s dbmlsync オプション」『Mobile Link クライアント管理』を参照してください。

- **MLUser 同期プロファイルオプションは廃止予定**　このオプションは廃止される予定です。代わりに -s dbmlsync オプションを使用することをおすすめします。「-s dbmlsync オプション」『Mobile Link クライアント管理』を参照してください。

- **dbmlsync 統合コンポーネントの削除**　dbmlsync 統合コンポーネントは削除されました。代わりに、dbmlsync プログラミングインタフェースを使用してください。「Dbmlsync API」『Mobile Link クライアント管理』を参照してください。

- **強制的な競合モードは廃止予定**　upload_insert、upload_update、upload_delete スクリプトがすべて未定義の場合、Mobile Link サーバは強制的な競合解決を使用します。この機能は廃止される予定です。

- **upload_update による競合の検出は廃止予定**　upload_update スクリプトでアップロードされた更新の競合を検出して解決するか、upload_fetch または upload_fetch_column_conflict スクリプトを使用して競合を検出する必要があります。Mobile Link サーバに依存して、upload_update スクリプトによって影響を受けるローをカウントして競合を検出し、その後競合解決スクリプトを呼び出す方法は、廃止される予定です。「upload_update スクリプトによる競合の検出」『Mobile Link サーバ管理』を参照してください。

- **SQL スクリプトでの疑問符の使用**　Mobile Link SQL スクリプトでの単純な疑問符の使用は廃止される予定です。代わりに名前付きパラメータを使用します。「名前付きスクリプトパラメータ」『Mobile Link サーバ管理』を参照してください。

SQL Remote の新機能

次に、バージョン 12.0.0 で導入された SQL Remote の追加機能を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

- **SQL Remote で空間値のレプリケーションのサポート**　SQL Remote で空間データ型のレプリケーションがサポートされるようになりました。また、新しい TIMESTAMP WITH TIMEZONE データ型のレプリケーションもサポートされるようになりました。「TIMESTAMP WITH TIME ZONE データ型」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **抽出ユーティリティ (dbxtract) への -g オプションの追加**　MANUAL REFRESH と定義されているマテリアライズドビューは、デフォルトで再ロード時に初期化されません。dbxtract で -g オプションを使用して、このようなマテリアライズドビューを再ロードの一部として初期化できるようになりました。「抽出ユーティリティ (dbxtract)」『SQL Remote』を参照してください。

- **パブリッシャが未定義の場合の、SQL Remote によるエラーメッセージ出力**　リモートユーザまたは統合ユーザが定義されており、パブリッシャが定義されていないデータベースに SQL Remote が接続すると、データベースにパブリッシャが定義されていないことを示すエラーメッセージが SQL Remote から返されます。

Ultra Light の新機能

次に、バージョン 12.0.0 で導入された Ultra Light の追加機能を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

一般的な機能

- **空間データのサポート**　Ultra Light の新しい空間データ機能をサポートするため、次の機能が追加されました。

  - **ST_Geometry データ型**　ST_Geometry データ型では、空間値に適用できる関数がサポートされています。「ST_GEOMETRY データ型」『Ultra Light データベース管理とリファレンス』を参照してください。

  - **空間データ関数**　Ultra Light には、空間データ処理をサポートするための関数がいくつか備えられています。「ST_GEOMETRY データ型」『Ultra Light データベース管理とリファレンス』を参照してください。

- **新しい暗号化の例**　BlackBerry スマートフォンでの Ultra Light J 暗号化を説明するため、新しいコードサンプルが追加されました。詳細については、*%SQLANYSAMP12%¥UltraLiteJ* ディレクトリを参照してください。

プラットフォームとデバイス

iPhone と Mac OS X のサポート

Ultra Light C/C++ API. を使用して、Mac OS X と iPhone を対象にした Ultra Light アプリケーションを Mac OS X で開発できるようになりました。詳細については、「Ultra Light C++ アプリケー

ション開発」『Ultra Light C/C++ プログラミング』を参照してください。 また、iPhone アプリケーションの開発を支援する詳しいチュートリアルもあります。

64 ビット Linux へのインストール

SQL Anywhere を 64 ビットの Linux コンピュータにインストールする場合、32 ビットのサブシステムも別個にインストールする必要があります。これは、一部の 64 ビットの Linux オペレーティングシステムに、32 ビットの互換性ライブラリが事前にインストールされていないためです。32 ビットのクライアントソフトウェアを使用するには、Linux ディストリビューション用に 32 ビットの互換性ライブラリをインストールする必要があります。たとえば、Ubuntu では、次のコマンドの実行が必要となることがあります。

```
sudo apt-get install ia32-libs
```

セキュリティ

- **Ultra Light で FIPS 140-2 認定の暗号化をサポート**　64 ビットの Windows では、FIPS 140-2 認定の暗号化が Ultra Light 用にサポートされるようになりました。

- **Ultra Light データベース暗号化で 256 ビットの AES を使用**　Ultra Light データベース暗号化が、128 ビットの AES ではなく、256 ビットの AES を使用して実行されるようになりました。

- **パスワードのハッシングにソルトを使用**　新しいユーザが作成されるか、または既存のユーザが各自のパスワードを変更すると、ランダムな 4 バイトのソルト値が生成されるようになりました。「Ultra Light PWD 接続パラメータ」『Ultra Light データベース管理とリファレンス』を参照してください。

ユーティリティ

- **Ultra Light ユーティリティの改善**　ulinit ユーティリティは、SQL Anywhere データベースから抽出されたスキーマに Ultra Light でサポートされない要素 (カラムのデータ型やインスタンスのデフォルト値など) が含まれていても、SQL Anywhere データベースの情報に基づいて Ultra Light データベースを作成できるように改善されました。これは、デフォルトの動作になりました (-f オプションを使用してオフにすることが可能)。「Ultra Light データベース初期化ユーティリティ (ulinit)」『Ultra Light データベース管理とリファレンス』を参照してください。 ulcreate ユーティリティは、サポートされなくなりました。

  ulinit、ulerase、ulinfo、ulload、ulsync の各ユーティリティにも変更が加えられました。

次の項を参照してください。

- 「Ultra Light データベース初期化ユーティリティ (ulinit)」『Ultra Light データベース管理とリファレンス』
- 「Ultra Light ユーティリティ消去 (ulerase)」『Ultra Light データベース管理とリファレンス』
- 「Ultra Light 情報ユーティリティ (ulinfo)」『Ultra Light データベース管理とリファレンス』
- 「Ultra Light データベースへの XML のロードユーティリティ (ulload)」『Ultra Light データベース管理とリファレンス』
- 「Ultra Light 同期ユーティリティ (ulsync)」『Ultra Light データベース管理とリファレンス』

SQL

- **CREATE／DROP／ALTER USER の追加**　次の項を参照してください。

  - 「CREATE USER 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  - 「DROP USER 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  - 「ALTER USER 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』

- **CREATE TABLE と ALTER TABLE の新しいテーブル制約**　CREATE TABLE 文または ALTER TABLE 文で、追加のテーブル制約 (SYNCHRONIZE ON | OFF | ALL) を指定できます。次の項を参照してください。

  - 「CREATE TABLE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  - 「ALTER TABLE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』

- **DROP 文への IF EXISTS 句の追加**　DROP INDEX、DROP PUBLICATION、DROP SYNCHRONIZATION PROFILE、DROP TABLE の各文で、新しい IF EXISTS 句をオプションで指定できるようになりました。 次の項を参照してください。

  - 「DROP INDEX 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  - 「DROP PUBLICATION 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  - 「DROP SYNCHRONIZATION PROFILE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  - 「DROP TABLE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』

- **CREATE 文への IF NOT EXISTS 句の追加**　CREATE TABLE、CREATE INDEX、CREATE PUBLICATION、CREATE SYNCHRONIZATION PROFILE の各文で、新しい IF NOT EXISTS 句をオプションで指定できるようになりました。次の項を参照してください。

  - 「CREATE TABLE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  - 「CREATE INDEX 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  - 「CREATE PUBLICATION 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  - 「CREATE SYNCHRONIZATION PROFILE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』

- **CREATE SYNCHRONIZATION PROFILE では、OR REPLACE 句もサポートされるようになりました。**　CREATE SYNCHRONIZATION PROFILE 文では、テーブルを置き換えるためのオプションもサポートされるようになりました。「CREATE SYNCHRONIZATION PROFILE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』を参照してください。

- **COUNT_UPLOAD_ROWS 関数の追加**　COUNT_UPLOAD_ROWS 関数では、次回の同期でアップロードされるロー数を問い合わせできます。「COUNT_UPLOAD_ROWS 関数 [集合]」『Ultra Light データベース管理とリファレンス』を参照してください。

データ型

- **ST_Geometry**　ST_Geometry データ型では、空間値に適用できる関数がサポートされています。「ST_GEOMETRY データ型」『Ultra Light データベース管理とリファレンス』を参照してください。

- **TIMESTAMP WITH TIMEZONE データ型**　TIMESTAMP WITH TIMEZONE データ型を使用すると、日付値と時刻値をタイムゾーンオフセットと一緒に格納できます。タイムゾーンオフセットは、UTC の前または後の分数です。「Ultra Light データベース初期化ユーティリティ (ulinit)」『Ultra Light データベース管理とリファレンス』と「Ultra Light timestamp_with_time_zone_format 作成パラメータ」『Ultra Light データベース管理とリファレンス』を参照してください。

プログラミングインタフェース

Ultra Light C/C++

- **新しい Ultra Light C++ API の追加**　Ultra Light C++ API の新バージョンが、このリリースに追加されました。この API は、*ulcpp.h* ヘッダファイルで宣言されます。「Ultra Light C/C++ API リファレンス」『Ultra Light C/C++ プログラミング』を参照してください。

  このバージョンの Ultra Light C++ API で、次のオブジェクトの名前が変更されました。

  - ul_synch_info は ul_sync_info に名前が変更されました。
  - ul_synch_result は ul_sync_result に名前が変更されました。
  - ul_synch_status は ul_sync_status に名前が変更されました。
  - ul_synch_stats は ul_sync_stats に名前が変更されました。
  - ul_synch_observer_fn は ul_sync_observer_fn に名前が変更されました。
  - ul_synch_state は ul_sync_state に名前が変更されました。
  - ULInitSynchInfo は ULInitSyncInfo に名前が変更されました。
  - ULSetSynchInfo は ULSetSyncInfo に名前が変更されました。
  - ULGetSynchResult は ULGetSyncResult に名前が変更されました。
  - All UL_SYNCH_STATE オブジェクトは UL_SYNC_STATE オブジェクトに名前が変更されました。
  - UL_SYNCH_STATUS_FLAG_IS_BLOCKING は UL_SYNC_STATUS_FLAG_IS_BLOCKING に名前が変更されました。
  - ULRegisterErrorCallback は ULSetErrorCallback に名前が変更されました。
  - ULRegisterSynchronizationCallback は ULSetSynchronizationCallback に名前が変更されました。

  メソッド GetDatabaseID、SetDatabaseID、IsCaseSensitive、GetCollationName が C++ API から削除されました。この機能は、GetDatabaseProperty と SetDatabaseOption で処理されるようになりました。ULConnection.GetDatabaseProperty メソッド [Ultra Light C++]『Ultra Light C/C++

プログラミング』と ULConnection.SetDatabaseOption メソッド [Ultra Light C++]『Ultra Light C/C++ プログラミング』を参照してください。

GetDatabasePropertyInt メソッドが追加されました。ULConnection.GetDatabasePropertyInt メソッド [Ultra Light C++]『Ultra Light C/C++ プログラミング』を参照してください。

Ultra Light.NET

- **新しい ULConnection.ValidateDatabase(ULDBValid) メソッドの追加**　tableName に NULL を渡して ULConnection.ValidateDatabase(ULDBValid, String) を呼び出すのと同じ機能として、このメソッドが追加されました。ULConnection.ValidateDatabase(ULDBValid) メソッド [Ultra Light.NET]『Ultra Light .NET プログラミング』を参照してください。

Ultra Light for M-Business Anywhere

- **API 機能**　このリリースには、次のオブジェクトが追加されています。

  - SyncParms クラスの setPublications メソッド。このメソッドは、同期させるパブリケーションを設定します。
  - SyncParms クラスの getPublications メソッド。このメソッドは、同期させるパブリケーションを返します。
  - CreationParms クラスの timestampWithTimeZoneFormat メンバー。このメンバーは、データベースから取り出したタイムゾーンでタイムスタンプのフォーマットを設定します。
  - PreparedStatement クラスの setTimestampWithTimeZoneParameter メソッドと ULTable クラスの setTimestampWithTimeZone メソッド。これらのメソッドは、指定された SQLType.TIMESTAMP_WITH_TIME_ZONE 型のパラメータの値を、Date オブジェクトを使用して設定します。
  - ResultSet と ULTable クラスの getTimestampWithTimeZone メソッド。これらのメソッドは、指定されたカラムの値を Date オブジェクトとして返します。

Ultra Light の動作の変更と廃止予定機能

> **注意**
> 廃止予定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、変更される可能性があります。

次に、バージョン 12.0.0 で導入された Ultra Light の廃止予定機能と動作の変更を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

廃止予定機能

- **ulcreate ユーティリティのサポート終了**　ulcreate ユーティリティは使用できなくなりました。すべての機能は ulinit で処理されるようになりました。「Ultra Light データベース初期化ユーティリティ (ulinit)」『Ultra Light データベース管理とリファレンス』を参照してください。

- **SQL パススルーのサポート終了**　Ultra Light 11 でリリースされた SQL パススルーは、サポートされなくなりました。この機能は「リモートデータベースの集中管理」『Mobile Link サーバ管理』 で処理されるようになりました。

- **ALTER SYNCHRONIZATION PROFILE での OR REPLACE 句の使用のサポート終了**　**OR REPLACE** 句は、**ALTER SYNCHRONIZATION PROFILE** 文から削除されました。

- **Ultra Light ODBC API のサポート終了**　Ultra Light ODBC API がサポートされなくなりました。代わりに、Ultra Light C/C++ API を使用します。Ultra Light C/C++ プログラミングを参照してください。

廃止予定のプラットフォーム

- **Ultra Light for M-Business Anywhere のサポート終了**　M-Business Anywhere の Ultra Light サポートは、Ultra Light 12 で廃止されます。

- **Palm オペレーティングシステムのサポート終了**　Palm オペレーティングシステムは、Ultra Light 12 ではサポートされません。

削除、廃止、変更された API

- **Ultra Light C/C++ API の置き換え**　*uliface.h* ヘッダファイルで定義される Ultra Light C/C++ API は、*ulcpp.h* ヘッダファイルで定義される新バージョンに置き換えられました。以前のバージョンの API は、引き続き使用できます。廃止された Ultra Light C/C++ API の詳細については、http://dcx.sybase.com/1101ja/ulc_ja11/c-common-apiref.html を参照してください。

  Ultra Light C/C++ API の古い実装を使用するには、Ultra Light アプリケーションプロジェクトに *%SQLANY11%¥SDK¥C¥ulcpp11.cpp* ファイルを追加します。SQLANY11 は、SQL Anywhere のインストールディレクトリをポイントする環境変数です。

- **Ultra Light C/C++ API オブジェクトの変更**　次のオブジェクトは前回のリリースから変更され、新しい Ultra Light C/C++ API に適用されます。

  ○SQL パススルーは、API によってサポートされなくなりました。次のオブジェクトが削除されました。

  - ul_sql_passthrough_state
  - ul_sql_passthrough_status
  - ul_sql_passthrough_observer_fn

- **Ultra Light C/C++ 共通 API オブジェクトの変更**　前回のリリース以降、次のオブジェクトが変更されています。

  ○MLFileTransfer 関数は、MLFileDownload に名前が変更されました。MLFileDownload メソッド [Ultra Light Embedded SQL]『Ultra Light C/C++ プログラミング』を参照してください。

  ○force_transfer field of ml_file_transfer_info 構造体は削除されました。

  ○enable_resume field of ml_file_transfer_info のデフォルトは、false ではなく true になりました。MLFileDownload メソッド [Ultra Light Embedded SQL]『Ultra Light C/C++ プログラミング』を参照してください。

○ファイル転送の制御を強化するため、Mobile Link サーバに渡される新しい remote_key field of ml_file_transfer_info が MLFileDownload でサポートされています。MLFileDownload メソッド [Ultra Light Embedded SQL]『Ultra Light C/C++ プログラミング』を参照してください。

- **Ultra Light 11.0 での埋め込み Visual C++ のサポート終了**　Visual Studio 2003 のサポートは、Ultra Light 11.0 で終了しました。したがって、埋め込み Visual C++ のサポートは、Visual Studio 2005 に移動されました。

- **埋め込み SQL API オブジェクトの変更**　前回のリリース以降、次のオブジェクトが変更されています。

  ○SQL パススルーは、API によってサポートされなくなりました。次のオブジェクトが削除されました。

  - ULGetSQLPassthroughScriptCount
  - ULExecuteNextSQLPassthroughScript
  - ULExecuteSQLPassthroughScripts
  - ULSetSQLPassthroughCallback

- **Ultra Light for M-Business Anywhere オブジェクトの変更**　前回のリリース以降、次のオブジェクトが変更されています。

  ○Connection クラスの GetSQLPassthroughScriptCount、ExecuteNextSQLPassthroughScript、ExecuteSQLPassthroughScripts メソッドが削除されました。

  ○DatabaseManager クラスの CreateDatabase メソッドの構文が変更されました。

- **Ultra Light .NET API オブジェクトの変更**　前回のリリース以降、次のオブジェクトが変更されています。

  ○ULConnection クラスの DatabaseManager プロパティが削除され、不要になりました。ULDatabaseManager は、シングルトンクラスではなく、静的になりました。ULDatabaseManager クラス [Ultra Light.NET]『Ultra Light .NET プログラミング』を参照してください。

  ○ULConnectionParms クラスの DatabaseOnCE プロパティの名前が DatabaseOnDevice に変更されました。ULConnectionParms.DatabaseOnDevice プロパティ [Ultra Light.NET]『Ultra Light .NET プログラミング』を参照してください。

  ○ULTableSchema クラスの GetOptimalIndex メソッドで、最適なインデックスの名前が返されるようになりました。ULTableSchema.GetOptimalIndex メソッド [Ultra Light.NET]『Ultra Light .NET プログラミング』を参照してください。

  ○ULConnection クラスの CountUploadRows(String, UInt32) メソッドが削除されました。CountUploadRows(String, Int64) を代わりに使用します。ULConnection.CountUploadRows メソッド [Ultra Light.NET]『Ultra Light .NET プログラミング』を参照してください。

○SQL パススルーは、API によってサポートされなくなりました。次のオブジェクトが削除されました。

- ULConnection.GetSQLPassthroughScriptCount
- ULConnection.ExecuteNextSQLPassthroughScript
- ULConnection.ExecuteSQLPassthroughScripts
- ULSqlPassthroughProgressListener
- ULSqlProgressData
- ULSqlProgressState

○ULDatabaseSchema クラスの GetPublicationSchema メソッドと一緒に、ULPublicationSchema クラスとそのメソッドが削除されました。SYNC_ALL_DB フィールドと SYNC_ALL_PUBS フィールドが、ULConnection クラスに移動されました。ULConnection.SYNC_ALL_DB フィールド [Ultra Light.NET]『Ultra Light .NET プログラミング』と ULConnection.SYNC_ALL_PUBS フィールド [Ultra Light.NET]『Ultra Light .NET プログラミング』を参照してください。

その他

- **ユーザパブリケーション制限の増大**　ユーザパブリケーションの最大数は 63 に増加されました。

- **Ultra Light データベースのデフォルトのエンコードは UTF-8 エンコード**　Ultra Light データベースは、UTF-8 エンコードがデフォルトになりました。「Ultra Light utf8_encoding 作成パラメータ」『Ultra Light データベース管理とリファレンス』を参照してください。

Ultra Light J の新機能

次に、バージョン 12.0.0 で導入された Ultra Light J の追加機能を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

- **外部 BLOB ファイルのサポート**　Ultra Light J で BlackBerry OS または Java SE を使用している場合、データベースファイルの分割がサポートされ、データベース内の特定のカラムを使用して参照されるファイルとともに、外部ファイルを使用して大きな BLOB 値を格納できるようになります。これは、CREATE TABLE SQL 関数の一部として実装されます。

  BlackBerry アプリケーションでのサンプルの使用シナリオとしては、小規模で高速の永続ストアを使用して Ultra Light J データベースを格納し、大規模で低速のフラッシュドライブまたは SD カードを使用して写真などの大規模な BLOB 値を格納します。写真をキャプチャしてデータベースに格納するアプリケーションで、バッテリー電力と写真をデータベースにコピーする時間が無駄にならないという利点があります。

  「CREATE TABLE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』を参照してください。

- **BlackBerry の内部フラッシュと SD カードのサポート**　Ultra Light J では、Ultra Light J データベースと内部フラッシュメモリまたは SD カードとの間で読み込み／書き込みを行うことができます。

- **複数バージョンの Ultra Light J のサポート**　Ultra Light J の複数のバージョンが BlackBerry と Java ME 環境で共存できるようにするため、Ultra Light J の JAR ファイル、COD ファイル、Java パッケージの名前にバージョン番号が次のように含まれるようになりました。

  - JAR ファイルは *UltraLiteJ12.jar* と呼ばれます。
  - COD ファイルは *UltraLiteJ12.cod* と呼ばれます。
  - すべてのパブリッククラスは、パッケージ com.ianywhere.ultralitej12 に含まれます。
  - Unsigned64 クラスは、パッケージ com.ianywhere.ultralitej12 に移動されました。

- **Ultra Light J アプリケーションの暗号化とセキュリティ保護**　BlackBerry スマートフォン用にセキュリティ保護が非常に強化された Ultra Light J アプリケーションの作成方法を示す新しいサンプルが追加されました。詳細については、*Samples¥UltraLite¥UltraLiteJ¥BlackBerryEncryption¥ReadMe.html* を参照してください。BlackBerry セキュリティの詳細については、http://www.sybase.com/detail_list?id=9814 にあるホワイトペーパー**『Ultra Light J Security on BlackBerry Devices』**を参照してください。

- **新しい ST_Geometry データ型**　ST_Geometry データ型では、空間値に適用できる関数がサポートされています。「ST_GEOMETRY データ型」『Ultra Light データベース管理とリファレンス』を参照してください。

- **TIMESTAMP WITH TIMEZONE データ型**　TIMESTAMP WITH TIMEZONE データ型を使用すると、日付値と時刻値をタイムゾーンオフセットと一緒に格納できます。タイムゾーンオフセットは、UTC の前または後の分数です。「Ultra Light データベース初期化ユーティリティ (ulinit)」『Ultra Light データベース管理とリファレンス』と「Ultra Light timestamp_with_time_zone_format 作成パラメータ」『Ultra Light データベース管理とリファレンス』を参照してください。

- **RAND SQL 関数のサポート**　Ultra Light J では、RAND SQL 関数をサポートしています。「RAND 関数 [数値]」『Ultra Light データベース管理とリファレンス』を参照してください。

- **CREATE SYNCHRONIZATION PROFILE、ALTER SYNCHRONIZATION PROFILE、DROP SYNCHRONIZATION PROFILE、SYNCHRONIZE のサポート**　これらの文は、SQL を使用した同期パラメータの編成と同期の起動を行うための代替手段を提供するためのものです。既存の Connection.createSyncParm()、Connection.synchronize(SyncParm)、および関連 API は、引き続き動作します。次の項を参照してください。

  - 「CREATE SYNCHRONIZATION PROFILE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  - 「ALTER SYNCHRONIZATION PROFILE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  - 「DROP SYNCHRONIZATION PROFILE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  - 「SYNCHRONIZE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』

- **CREATE TABLE と ALTER TABLE の新しいテーブル制約**　CREATE TABLE 文または ALTER TABLE 文で、追加のテーブル制約 (SYNCHRONIZE ON | OFF | ALL) を指定できます。次の項を参照してください。

  ○「CREATE TABLE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  ○「ALTER TABLE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』

- **IF EXISTS 句の追加**　DROP INDEX、DROP PUBLICATION、DROP SYNCHRONIZATION PROFILE、DROP TABLE の各文で、新しい IF EXISTS 句をオプションで指定できるようになりました。次の項を参照してください。

  ○「DROP INDEX 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  ○「DROP PUBLICATION 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  ○「DROP SYNCHRONIZATION PROFILE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  ○「DROP TABLE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』

- **IF NOT EXISTS 句の追加**　CREATE TABLE、CREATE INDEX、CREATE PUBLICATION、CREATE SYNCHRONIZATION PROFILE の各文で、新しい IF NOT EXISTS 句をオプションで指定できるようになりました。次の項を参照してください。

  ○「CREATE TABLE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  ○「CREATE INDEX 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  ○「CREATE PUBLICATION 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
  ○「CREATE SYNCHRONIZATION PROFILE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』

- **CREATE SYNCHRONIZATION PROFILE では、OR REPLACE 句もサポートされるようになりました。**　CREATE SYNCHRONIZATION PROFILE 文では、テーブルを置き換えるためのオプションもサポートされるようになりました。「CREATE SYNCHRONIZATION PROFILE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』を参照してください。

- **Mobile Link 経由のファイル転送**　Ultra Light J では、Mobile Link サーバ経由で、リモートデータベースとの間でファイルのアップロードとダウンロードを行うことができます。

  Ultra Light J のデスクトップバージョンでは、Mobile Link からローカルファイルシステムにどのタイプのファイルでもダウンロードでき、ローカルファイルシステムから Mobile Link にどのタイプのファイルでもアップロードできます。

  Ultra Light J の BlackBerry OS バージョンでは、有効であり暗号化および難読化されていないデータベースファイルを Mobile Link からダウンロードしてオブジェクトストアに格納したり、これらのタイプのデータベースを Mobile Link にアップロードしたりできます。Mobile Link からメディアカードまたはフラッシュストレージにどのタイプのファイルでもダウンロードでき、フラッシュカードやメディアカードから Mobile Link にどのタイプのファイルでもアップロードできます。

  Ultra Light J のファイル転送の詳細については、FileTransfer インタフェース [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

- **Ultra Light J データベース転送ユーティリティで、BlackBerry のファイルシステムでデータベースを開いたり転送したりできる**　Ultra Light J では、データベース名に基づいてデータベースの場所 (オブジェクトストアまたはファイルシステム) が自動的に判断されます。名前の先頭が *file:*// (大文字と小文字が区別される) である場合はデータベースがファイルシステムで検索され、それ以外の場合はデータベースがオブジェクトストアで検索されます。

- **BlackBerry インストールディレクトリ名の変更**　BlackBerry スマートフォンファイルのインストールディレクトリは、BlackBerry の最小 OS バージョンを使用するように名前が変更されました。*UltraLite¥UltraLiteJ¥Java MERIM11* ディレクトリは *UltraLite¥UltraLiteJ ¥BlackBerry4.2* になり、BlackBerry OS 4.2 以降と互換性のあるファイルであることを示します。

- **ULjLoad と ULjUnload での blobfile タイプのサポート**　Ultra Light J のロードユーティリティとアンロードユーティリティでは、blobfile タイプのカスタム実装をサポートしています。blobfile タイプの Ultra Light J データベースのサンプル実装については、「Ultra Light Java Edition データベースアンロードユーティリティ (uljunload)」『Ultra Light データベース管理とリファレンス』と「Ultra Light Java Edition データベースロードユーティリティ (uljload)」『Ultra Light データベース管理とリファレンス』を参照してください。

- **ロー制限アルゴリズムの改善**　カラムが多いテーブルのローが、カラムが少ないテーブルのローより多くのリソースを使用できるようにロー制限アルゴリズムが改善されました。

- **システムテーブルの変更**　Ultra Light J システムテーブルに、次の変更が加えられました。

  - **syscolumn システムテーブルの column_default_value カラムで、DEFAULT AUTOFILENAME デフォルト句がサポートされる**　このカラムでは、カラムデフォルト値 DEFAULT AUTOFILENAME で指定された VARCHAR 型を処理できます。「syscolumn システムテーブル」『Ultra Light データベース管理とリファレンス』を参照してください。

  - **syscolumn システムテーブルへの filename_colid カラムの追加**　スキーマ定義の file_name カラムが参照される場合は、このカラムに参照先カラムのカラム ID が格納され、そうでない場合は、このカラムが NULL になります。「syscolumn システムテーブル」『Ultra Light データベース管理とリファレンス』を参照してください。

  - **systable システムテーブルへの table_partition_size カラムの追加**　このカラムには、定義された分割サイズの値が格納されます。「systable システムテーブル」『Ultra Light データベース管理とリファレンス』を参照してください。

- **暗号化のパフォーマンスの変更**　低速の CPU 環境でパフォーマンスを向上するため、EncryptionControl インタフェースが次のように改善されました。

  - Ultra Light J で、データとシステムにとって重要なページのみが暗号化されるようになりました。

  - 復号化メソッドで、復号化する必要のあるバイト数を指定するため、追加パラメータを指定できるようになりました。

Ultra Light J API の新機能

- **API 機能**　このリリースには、次のオブジェクトが追加されています。

○ConfigFileME インタフェースが追加され、Java ME デバイスファイルシステム上のファイルに保存される永続データベースを設定できるようになりました。

○ConfigPersistent インタフェースの setRowScoreFlushSize メソッドと setRowScoreMaximum メソッドが追加され、setRowMinimumThreshold メソッドと setRowMaximumThreshold メソッドに置き換えられました。これらのメソッドは、新しいロー制限アルゴリズムに対応しています。ロー制限アルゴリズムによって、カラムが多いテーブルのリソース管理が改善されます。ConfigPersistent.setRowScoreFlushSize メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』と ConfigPersistent.setRowScoreMaximum メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

ユニークな識別子を記述するため、UUIDValue インタフェースが追加されました。UUIDValue インタフェース [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

○現在の接続で DEFAULT AUTOINCREMENT または DEFAULT GLOBAL AUTOINCREMENT カラムに挿入された最新の値を取り出すため、Connection インタフェースに getLastIdentity メソッドが追加されました。Connection.getLastIdentity メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

○接続用に現在登録されている SyncObserver オブジェクトを返すため、Connection インタフェースに getSyncObserver メソッドが追加されました。Connection.getSyncObserver メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

○接続で最後に実行された SYNCHRONIZE SQL 文の結果を返すため、Connection インタフェースに getSyncResult メソッドが追加されました。Connection.getSyncResult メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

○接続で同期の進行状況をモニタする SyncObserver オブジェクトを設定するため、Connection インタフェースに setSyncObserver メソッドが追加されました。Connection.setSyncObserver メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

○データベースファイルの転送に使用できる FileTransfer オブジェクトを作成するため、DatabaseManager インタフェースに createFileTransfer メソッドと createObjectStoreTransfer メソッドが追加されました。DatabaseManager.createFileTransfer メソッド [Ultra Light J]『Ultra Light Java プログラミング』と DatabaseManager.createObjectStoreTransfer メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

○ファイル転送オプションを指定するメソッドを提供するため、FileTransfer インタフェースが追加されました。FileTransfer インタフェース [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

○observer コールバックにデータを渡すことができるようにするため、FileTransferProgressData インタフェースが追加されました。FileTransferProgressData インタフェース [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

○ファイル転送 observer として FileTransferProgressListener インタフェースが追加されました。 FileTransferProgressListener インタフェース [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

○クエリプランをコンピュータのモニタに表示したり印刷したりする場合の読みやすさを向上させるために、PreparedStatement インタフェースに getPlanTree メソッドが追加されました。PreparedStatement.getPlanTree メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

○シャドウページングがオンになったことを検出するために、ConfigPersistent インタフェースに hasShadowPaging メソッドが追加されました。ConfigPersistent.hasShadowPaging メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

Ultra Light J の動作の変更と廃止予定機能

次に、バージョン 12.0.0 で導入された Ultra Light J の廃止予定機能と動作の変更を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

削除、廃止予定、変更されたユーティリティオプション

- **ULjInfo、ULjLoad、ULjUnload の -p オプションは省略可能**　**-p** が指定されない場合は、**dba** がデフォルトのパスワードとして使用されます。

廃止予定のプラットフォーム

- **Ultra Light J での BlackBerry OS 4.1 のサポート終了**　Ultra Light J で、BlackBerry OS 4.2 以降をサポートするようになりました。

削除、廃止、変更された API

- **Ultra Light J API オブジェクトの変更**　前回のリリース以降、次のオブジェクトが変更されています。

  ○ianywhere.ultralitej パッケージの名前が変更されました。Ultra Light J の複数のバージョンが BlackBerry と Java ME 環境で共存できるようにするため、Ultra Light J の JAR ファイル、COD ファイルのパッケージ名にバージョン番号が含まれるようになりました。「Ultra Light J API リファレンス」『Ultra Light Java プログラミング』を参照してください。

  ○Ultra Light J の syscolumn システムテーブルから table_autoinc カラムが削除されました。「syscolumn システムテーブル」『Ultra Light データベース管理とリファレンス』を参照してください。

  ○CollectionOfValueReaders インタフェースが削除されました。すべての CollectionOfValueReader メソッドが ResultSet インタフェースに移動されました。ResultSet インタフェース [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

  ○CollectionOfValueWriters インタフェースが削除されました。すべての CollectionOfValueWriter メソッドが PreparedStatement インタフェースに移動されました。PreparedStatement インタフェース [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

  ○ConfigPersistent methods setRowMaximumThreshold と setRowMinimumThreshold が、setRowScoreFlushSize と setRowScoreMaximum に置き換えられました。

ConfigPersistent.setRowScoreFlushSize メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』と ConfigPersistent.setRowScoreMaximum メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

○スキーマ関連のすべてのメソッドとインタフェースが削除されました。ForeignKeySchema インタフェースと、ColumnSchema、IndexSchema、TableSchema インタフェースのすべてメソッドが削除されました。次のメソッドが Connection インタフェースから削除されました。

- createPublication
- createTable
- dropForeignKey
- dropPublication
- dropTable
- enableSynchronization
- renameTable
- schemaCreateBegin
- schemaCreateComplete
- setNeverSynchronized
- startSynchronizationDelete
- stopSynchronizationDelete
- truncateTable

スキーマ操作を実行するには、Ultra Light J でサポートされている SQL 文 (CREATE TABLE や CREATE PUBLICATION など) を使用します。「Ultra Light SQL 文」『Ultra Light データベース管理とリファレンス』を参照してください。

○Connection インタフェースの setDatabaseId と getDatabasePartitionSize メソッドが削除されました。デフォルトの分割サイズは、指定できなくなりました。デフォルトの分割サイズを上書きするには、DEFAULT GLOBAL AUTOINCREMENT 文を使用します。「Ultra Light の GLOBAL AUTOINCREMENT カラム」『Ultra Light データベース管理とリファレンス』を参照してください。

○Connection インタフェースの getDatabaseID メソッドは、Connection.getOption(OPTION_DATABASE_ID) の呼び出しと同じ効果になりました。Connection.getOption メソッド [BlackBerry] [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

○Value、ValueReader、ValueWriter クラスは削除されました。

その他

- **ユーザパブリケーション制限の増大**　ユーザパブリケーションの最大数は 63 に増加されました。

管理ツールの新機能

次に、バージョン 12.0.0 で導入された管理ツールの追加機能を示します。サポートされているプラットフォームとバージョンについては、http://www.sybase.com/detail?id=1061806 を参照してください。

- **[接続] ウィンドウ**　バージョン 12.0.0 では、**[接続]** ウィンドウのレイアウトが簡易化され、**[接続アシスタント]** は削除されました。以前は、目的の接続タイプに必要なオプションを把握しておく必要がありました。現在は、選択した接続タイプに応じたオプションが **[接続]** ウィンドウに表示されます。

警告

○**[アクション]** ドロップダウンリストから、次のいずれかの接続タイプを選択してください。

[このコンピュータで稼働しているデータベースに接続]　このコンピュータ上ですでに稼働しているデータベースに接続します。

[ODBC データソースを使用した接続]　ODBC データソースを使用してデータベースに接続します。

[別のコンピュータで稼働しているデータベースに接続]　ネットワーク内の別のコンピュータ上ですでに稼働しているデータベースに接続します。

[このコンピュータのデータベースを起動して接続]　このコンピュータでデータベースを開始して、接続します。

[別のコンピュータのデータベースを起動して接続]　ネットワークを経由して別のコンピュータ上でデータベースを開始し、接続します。

[接続文字列を使用して接続]　接続文字列を使用してデータベースに接続します。

[アクション] ドロップダウンリストの下に表示されるオプションは、選択内容によって異なります。

○Interactive SQL では、**[SQL Anywhere データベースに接続]** 見出しの下にある **[データベースタイプを変更]** をクリックして、接続先となるデータベースのタイプを変更します。

たとえば、ODBC データソースがあるデータベースに接続している場合、**[接続]** ウィンドウには、**[ODBC データソース名]** と **[ODBC データソースファイル]** の 2 つのオプションのみが表示されます。

必要に応じて **[詳細]** をクリックして、TCP/IP や暗号化オプションと、その他の詳細オプションを指定できます。「[接続] ウィンドウを開く (Sybase Central の場合)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **ソフトウェア更新をチェックする新しいデフォルト**　Interactive SQL、Sybase Central、SQL Anywhere コンソールユーティリティ、Mobile Link モニタで、デフォルトにより更新が毎日チェックされるようになりました。ソフトウェアの以前のバージョンのデフォルトでは、

チェックされませんでした。「ソフトウェアの更新」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Interactive SQL と Sybase Central でのイメージデータ、HTML データ、XML データの表示**　結果セットで、イメージデータ、HTML データ、XML データをプレビューできます。「Interactive SQL でイメージと SVG を表示する」『SQL Anywhere サーバ データベース管理』と「Interactive SQL での HTML データと XML データの表示」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Sybase Central と Interactive SQL 用の新しいシステムトレイアイコン**　Sybase Central または Interactive SQL の高速ランチャオプションを有効にすると、システムトレイに新しいアイコンが表示されるようになりました。アプリケーションを開くには、アイコンを右クリックして **[開く]** をクリックし、アプリケーションを閉じて (実行中の場合) 高速ランチャプロセスを終了するには、**[終了]** をクリックします。「高速ランチャオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **アクセシビリティ有効化オプション**　アクセシビリティ有効化オプションが、デフォルトでインストールされるようになりました。以前は、このオプションを別個にインストールする必要がありました。アクセシビリティ有効化オプションを使用すると、Interactive SQL、Sybase Central、SQL Anywhere コンソールユーティリティ、Mobile Link モニタを画面リーダーなどのアクセシビリティ補助で操作することができます。「Java Access Bridge を使用したアクセシビリティ」『SQL Anywhere 16 紹介』を参照してください。

Sybase Central プラグインの新機能

次に、バージョン 12.0.0 で導入された Sybase Central のプラグインの追加機能を示します。

SQL Anywhere プラグインの新機能

- **ウィザードによって作成された SQL 文とユーティリティコマンドの表示**　データベースオブジェクトを作成したりデータベースユーティリティを実行したりするほとんどの SQL Anywhere プラグインウィザードの最後には、新しいページが表示されます。このページには、**[完了]** をクリックすると実行される SQL 文とユーティリティコマンドが表示されます。**[完了]** をクリックすると、SQL 文かユーティリティコマンドまたはその両方が実行されます。

  または、SQL 文をクリップボードにコピーし、**[キャンセル]** をクリックしてウィザードを終了した後、Interactive SQL を使用して SQL 文を実行することもできます。この機能を使用すると、データベースを変更することなくウィザードを使用して SQL スクリプトを生成できます。「ウィザードによって生成された SQL 文とユーティリティコマンドの表示」『SQL Anywhere サーバ データベース管理』を参照してください。

- **空間データのサポート**　空間データに関連する、ウィザードを含めたプラグインの新機能の詳細については、空間データのサポート 119 ページを参照してください。

- **新しいデータベース[断片化] タブ**　ベーステーブルとインデックスの断片化をグラフィックで表示できます。**[断片化]** タブには、ベーステーブルで sa_table_fragmentation システムプロシージャを実行した結果がグラフィックで表示されます。「[断片化] タブ (SQL Anywhere プラグインの場合)」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **データベース変更のロギング**　SQL Anywhere プラグインによって実行された、データベースを変更するすべての SQL 文を記録できます。ログは、*.sql* ファイルに保存されます。「SQL 文のロギング」『SQL Anywhere サーバ データベース管理』を参照してください。

- **WITH NULLS NOT DISTINCT 句、CREATE INDEX 文のサポート**　SQL Anywhere プラグインを使用して、WITH NULLS NOT DISTINCT 句を使用するインデックスを作成、表示、変更することができます。[インデックス] フォルダには、**[NULL 排他]** カラムがあります。インデックスがプライマリキーインデックス、外部キーインデックス、一意性制約インデックス、または非ユニークインデックスの場合、カラム値は空白になります。

  「CREATE INDEX 文」『SQL Anywhere サーバ SQL リファレンス』の UNIQUE 句の説明を参照してください。

- **単語区切り**　SQL Anywhere プラグインでは、テキスト設定オブジェクトの外部単語区切りと事前フィルタがサポートされています。「ALTER TEXT CONFIGURATION 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **データベースドキュメントの改善**　**データベースドキュメントウィザード**を使用すると、生成されたドキュメントにテーブルに関する情報が含まれます。テーブル情報には、所有者名、テーブルを変更するプロシージャ、カラム情報、コメントなどがあります。「データベースのドキュメント化」『SQL Anywhere サーバ データベース管理』を参照してください。

- **[タイプフィルタの設定] ウィンドウの改善**　**[タイプフィルタの設定]** ウィンドウでは、Sybase Central の左ウィンドウ枠にあるフォルダリストに表示するデータベースオブジェクトを指定できます。独自の指定をデフォルトのタイプフィルタとして設定できます。Sybase Central では、タイプフィルタが指定されていないデータベースに接続するたびに、デフォルトのタイプフィルタが使用されます。「Sybase Central のナビゲーション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **メンテナンスプラン作成ウィザードの改善**　メンテナンスプランには、ユーザ定義の操作が含まれています。**メンテナンスプラン作成ウィザード**で、ユーザ定義の操作を、検証前またはバックアップ後に実行する SQL 文として追加できます。

  また、**[メンテナンスプラン]** フォルダに **[ステータス]** カラムが含まれるようになりました。「メンテナンスプラン」『SQL Anywhere サーバ データベース管理』を参照してください。

- **ファンクション作成ウィザードの改善**　以前は、**ファンクション作成ウィザード**で戻り値のタイプが組み込みタイプに制限されていましたが、組み込みタイプまたはドメインのいずれかを選択できるようになりました。「ユーザ定義関数の作成」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **データベースバックアップウィザードの改善**　**データベースバックアップウィザード**で、空きページの削除がデフォルトで有効になるようになりました。「アーカイブバックアップの作成 (Sybase Central の場合)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **[概要] タブの改善**　データベースの **[概要]** タブには、コピーサーバの正常性と統計情報がデータベースミラーリング情報と一緒に表示されます。「データベースの正常性と統計情報」『SQL Anywhere サーバ データベース管理』を参照してください。

- **[同期サブスクリプションのプロパティ] ウィンドウの改善**　SQL Anywhere 12 以降で作成されたデータベースでは、同期サブスクリプションの名前が、**[同期サブスクリプションのプロパティ]** ウィンドウと **[同期サブスクリプション]** タブの両方に表示されます。以前のバージョンの SQL Anywhere で作成されたデータベースでは、名前が **(unnamed)** として表示されます。「同期サブスクリプションの作成」『Mobile Link クライアント管理』を参照してください。

- **新しい [同期プロファイルを使用して同期] ウィンドウ**　同期プロファイルを右クリックし、**[同期]** をクリックしてから **[OK]** をクリックすると、SQL Anywhere データベースを同期できます。「Mobile Link 同期プロファイル」『Mobile Link クライアント管理』を参照してください。

Sybase Central の動作の変更と廃止予定機能

次に、バージョン 12.0.0 で導入された Sybase Central の変更点を示します。

- **Sybase Central 6.1.0 へのバージョン変更**　SQL Anywhere 12 には、バージョン 6.1.0 の Sybase Central が含まれています。

- **Sybase Central は、SQL Anywhere バージョン 10x 以降のデータベースのみサポート**　バージョン 9 のデータベースサーバと、バージョン 9 のソフトウェアで作成されたデータベースのサポートが SQL Anywhere プラグインから削除されました。データベースをアンロードして、再ロードファイル、新しいデータベース、または既存のデータベースに再ロードする場合、バージョン 9 以降のデータベースサーバで稼働している、バージョン 5、6、7、8 または 9 のソフトウェアで作成したデータベースには引き続き接続できます。

  バージョン 6、7、8、9 データベースサーバで動作するバージョン 5、6、7、8、9 ソフトウェアで作成されたデータベースの場合、一時的に SQL Anywhere プラグインからデータベースに接続して、次のいずれかのタスクを実行できます。

  ○データベースを再ロードファイルにアンロードします。

  ○データベースをアンロードして、新しいバージョン 12 データベースに再ロードします。

  ○データベースをアンロードして、既存のバージョン 12 データベースに再ロードします。

  アンロードされるデータベースのファイルはローカルのコンピュータに置いておく必要があります。

  バージョン 5 以前のサーバで動作するバージョン 4 以前のソフトウェアによって作成されたデータベースの SQL Anywhere プラグインはサポートされません。

  「SQL Anywhere サーバのアップグレード」289 ページを参照してください。

- **32 ビットと 64 ビットのコンピュータがサポートされています。**　Sybase Central の設定ファイルの名前が変更されました。

  ○32 ビットのコンピュータでは、*.scRepository600* ファイルの名前が *.scRepository610_32* に変更されました。

  ○64 ビットのコンピュータでは、インストール環境に応じて、*.scRepository600* ファイルの名前を *.scRepository610_32* または *.scRepository610_64* に変更できます。

- **Sybase Central を配備するときに、プラグイン用の JPR ファイルを作成する必要がなくなった**　以前は、Sybase Central と SQL Anywhere を配備するときに、プラグインごとに JPR ファイルを作成する必要がありましたが、Sybase Central で環境変数 *%SQLANY12%* を使用してプラグインの検出と登録が行われるようになりました。

SQL Anywhere プラグインの変更機能

- **メンテナンスプランの強化**　メンテナンスプランが次のように強化されています。

  - 検証前に実行される SQL 文を含めることができる。
  - バックアップ後に実行される SQL 文を含めることができる。
  - メンテナンスプランの実行中にステータスを表示できる。
  - 一度に実行できるメンテナンスプランは 1 つのみに限定される。

  詳細については、「メンテナンスプラン」『SQL Anywhere サーバ データベース管理』を参照してください。

- **動的オブジェクトとプロパティの自動再表示**　イベントの動的プロパティ、メンテナンスプラン、Windows サービスに加え、接続とロックのリストがデフォルトにより 10 秒ごとに自動的に再表示されます。「再表示頻度の設定」『SQL Anywhere サーバ データベース管理』を参照してください。

- **データベース作成ウィザードの新しいデフォルト**　データベース作成ウィザードを使用して新しいバージョン 12 のデータベースを作成する場合、グローバルチェックサムがデフォルトで有効になりました。ウィザードを使用してバージョン 11 以降のデータベースを作成する場合、グローバルチェックサムはデフォルトで無効になります。Windows Mobile 用のデータベースの作成では、グローバルチェックサムはデフォルトで常に有効になります。新しいデータベースでチェックサムがデフォルトで有効 146 ページを参照してください。

Interactive SQL の新機能

次に、バージョン 12.0.0 で導入された Interactive SQL の追加機能を示します。

- **[空間プレビュー] と [空間ビューア] を使用した空間データの表示**　「空間データのイメージとしての表示 (Interactive SQL の場合)」『SQL Anywhere サーバ 空間データサポート』を参照してください。

- **COMMIT 文と ROLLBACK 文の新しい実行方法**　Interactive SQL で、COMMIT 文を実行するには **[SQL]** » **[コミット]** をクリックし、ROLLBACK 文を実行するには **[SQL]** » **[ロールバック]** をクリックします。また、キーボードショートカットも使用できます。COMMIT 文を実行するには [Ctrl+Shift+C] キーを使用し、ROLLBACK 文を実行するには [Ctrl+Shift+R] キーを使用します。

  [SQL] メニューまたはキーボードショートカットを使用して COMMIT または ROLLBACK を実行した場合、**[SQL 文]** ウィンドウ枠の内容は変更されません。ただし、**[結果]** ウィンドウ枠の **[結果]** タブの内容はクリアされます。「Interactive SQL のキーボードショートカット」『SQL Anywhere サーバ データベース管理』を参照してください。

- **結果セットからのカラム、ロー、セルの選択とコピー方法に加えられた変更**　Interactive SQL の **[結果]** ウィンドウ枠で、結果セットから複数のカラム、ロー、セルを選択してコピーできま

す。たとえば、複数のカラムを選択するには、[Ctrl] キーを押しながらコピーするカラム内のセルをクリックし、右クリックして **[データのコピー]** » **[カラム]** を選択します。「Interactive SQL 結果セットからロー、カラム、セルをコピーする」『SQL Anywhere サーバ データベース管理』を参照してください。

- **OEM ユーザがパスワードをお気に入りに保存できなくする**　OEM 配備で、Interactive SQL のお気に入りの接続にユーザがパスワードを保存することを禁止できるようになりました。「管理ツールの設定」『SQL Anywhere サーバ プログラミング』の allowPasswordsInFavorites オプションを参照してください。

- **お気に入りの編集、インポート、エクスポート**　Interactive SQL のお気に入りを編集、エクスポート、インポートできるようになりました。「SQL スクリプトファイル、SQL 文、および接続をお気に入りに追加する」『SQL Anywhere サーバ データベース管理』と「お気に入りの共有」『SQL Anywhere サーバ データベース管理』を参照してください。

- **実行時間**　Interactive SQL のステータスバーに、現在の SQL 文の実行時間が表示されます。

- **[結果] ウィンドウ枠の変更**

  - **結果がテキストとして、またはスクロールテーブルに表示されます。**　以前は、**[オプション]** ウィンドウの設定を変更して、**[結果]** ウィンドウ枠の結果セットの表示のみが設定可能でした。**[データ]** » **[結果をスクロール可能なテーブルとして表示]** をクリックして、結果セットをスクロール可能なテーブルに表示できるようになりました。また、**[データ]** » **[結果をテキストとして表示]** をクリックすることによって、結果セットをテキストとして表示することもできます。変更内容を有効にするには、文を実行する必要があります。「Interactive SQL のカスタマイズ」『SQL Anywhere サーバ データベース管理』を参照してください。

  - **Interactive SQL でのカラムのサイズ設定**　結果セットを右クリックし、ウィンドウまたはデータに合わせてカラムのサイズを設定するかどうかを選択できます。「Interactive SQL のカスタマイズ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **SQL 関数のヘルプ**　**[SQL 文]** ウィンドウ枠で、SQL 関数を右クリックし、**[*function-name* のヘルプ]** をクリックすると、関数の説明が表示されます。

- **Interactive SQL で警告メッセージを非表示にする**　Interactive SQL に表示される警告メッセージの一部を無効にすることができます。たとえば、**[SQL 文]** ウィンドウ枠でテキストを保存しない場合に表示される警告を非表示にすることができます。

  [ツール] » **[オプション]** » **[メッセージ]** をクリックし、**[オプションのメッセージ]** リストのチェックボックスをオフにすることで、警告メッセージを無効にすることができます。「Interactive SQL のカスタマイズ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Interactive SQL でのファイルのリカバリ**　Interactive SQL が予期せず終了した場合は、*.sql* ファイルに保存されていない変更内容をリカバリしようとします。ファイルを編集すると、最後の変更から 30 秒が経過し、文の実行前に、Interactive SQL でバックアップコピーが作成されます。

Interactive SQL の動作の変更と廃止予定機能

次に、バージョン 12.0.0 で導入された Interactive SQL の変更点を示します。

- **32 ビットと 64 ビットのコンピュータがサポートされています。**

  ○Interactive SQL の 32 ビットバージョンと 64 ビットバージョンの両方を同じコンピュータにインストールできるようになりました。

- **Windows のデフォルトエンコードの変更**　次の変更は、たとえば、英語版の Windows XP コンピュータで ANSI エンコードと OEM エンコードが異なる場合に、Windows コンピュータでコンソールアプリケーションとして稼働している Interactive SQL (ウィンドウベースのユーザインタフェースは使用しない) に適用されます。

  1. 以前は、Interactive SQL をコンソールアプリケーションとして実行する場合、ENCODING 句が明示的に指定されていないと、INPUT 文と READ 文では OEM エンコード (英語版の Windows XP コンピュータの cp437) を使用してファイルがエンコードされていると見なされました。同様に、OUTPUT 文では OEM エンコードを使用してファイルが出力されました。

     Interactive SQL をコンソールアプリケーションとして実行する場合、INPUT 文と READ 文では、ANSI エンコード (英語版の Windows XP コンピュータの cp1252) を使用してファイルがエンコードされていると見なされるようになりました。同様に、OUTPUT 文では ANSI エンコードを使用してファイルが出力されます。

     OEM エンコードを使用するファイルをコマンドプロンプトから処理するには、エンコードを明示的に指定する必要があります。次に例を示します。

     ```
     dbisql READ ENCODING 'cp437' myfile.sql
     ```

  2. 以前は、Interactive SQL をコンソールアプリケーションとして実行した場合、ANSI エンコード (英語版の Windows XP コンピュータの cp1252) を使用するコマンドプロンプトとの間で結果の書き込み／読み込みが行われていたため、拡張文字が正しく表示されないことがありました。

     Interactive SQL をコンソールアプリケーションとして実行した場合、OEM エンコード (英語版の Windows XP コンピュータの cp437) を使用するコマンドプロンプトとの間で結果の書き込み／読み込みが行われるようになりました。

  「default_isql_encoding オプション [Interactive SQL]」『SQL Anywhere サーバ データベース管理』を参照してください。

- **CLEAR 文、[クリア] メニュー項目、[Esc] キーの変更**　CLEAR 文を実行すると、開いているすべての結果セットが閉じ、**[SQL 文]** ウィンドウ枠の内容はそのままとなります。「CLEAR 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  また、**[編集]** » **[結果を閉じる]** メニュー項目を使用することは、CLEAR 文を実行することと同じであり、開いているすべての結果セットが閉じ、SQL 文のウィンドウ枠の内容はそのままとなります。

  SQL 文のウィンドウ枠の内容をクリアするために使用する **[編集]** » **[クリア]** メニュー項目は削除されました。その結果、**[クリア]** メニュー項目のキーボードショートカットである [Esc]

キーも削除されました。デフォルトでは、[Esc] キーを押しても、何も起こらなくなりました。

ただし、**[SQL 文]** ウィンドウ枠の内容をクリアし、開いている結果セットを閉じるように [Esc] キーを設定できます。**[ツール]** » **[オプション]** » **[互換性]** をクリックし、**[Esc キーを押すと、SQL 文がクリアされ、結果セットが閉じます。]** をクリックします。

- **-codepage オプションが削除された**　Interactive SQL に特定のコードページを含むファイルを読み込ませる場合、INPUT 文、OUTPUT 文、または READ 文の ENCODING 句を使用してください。-codepage オプションはソフトウェアから削除されました。次の項を参照してください。

  - 「Interactive SQL ユーティリティ (dbisql)」『SQL Anywhere サーバ データベース管理』
  - 「Ultra Light 用 Interactive SQL ユーティリティ (dbisql)」『Ultra Light データベース管理とリファレンス』
  - 「INPUT 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』
  - 「OUTPUT 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』
  - 「READ 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』

- **SET OPTION 文 [Interactive SQL] の変更**　以前は、オプションを設定するための SET OPTION 文で値を指定しなかった場合、オプションは Off に設定されました。オプション値を省略した場合に、指定したオプションがデフォルト値に設定されるようになりました。この変更は、auto_commit、auto_refetch、bell、commit_on_exit、echo の各オプションに適用されます。「SET OPTION 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **OUTPUT 文の変更**　結果を TEXT ファイルに出力する場合、WITH COLUMN NAMES 句を使用して、カラム名をファイルの先頭に挿入できます。「OUTPUT 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **INPUT 文の変更**　INPUT 文で TEXT ファイルから行を挿入する場合、SKIP 句を使用して、ファイルの先頭から指定の行数を省略できるようになりました。「INPUT 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **READ キーワードが不要になった**　コマンドプロンプトから Interactive SQL を実行する場合、実行する *.sql* ファイルを指定するときに READ キーワードがオプションになりました。*.sql* ファイルにパラメータが必要な場合は、ファイル名の後にパラメータを指定します。

  たとえば、次の 2 つのコマンドは同じです。

READ キーワードを指定する	READ キーワードを指定しない
READ file.sql parm1	file.sql parm1

  「Interactive SQL ユーティリティ (dbisql)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Microsoft Excel ODBC ドライバのサポートの改善**　次に、Microsoft Excel ODBC ドライバを使用してデータを SQL Anywhere から Excel ファイルにエクスポートする場合の変更点を示します。

○以前は、CHAR、LONG VARCHAR、NCHAR、NVARCHAR、LONG NVARCHAR データ型として格納されているデータはエクスポートできませんでした。

Microsoft Excel ODBC ドライバを使用して、CHAR、LONG VARCHAR、NCHAR、NVARCHAR、または LONG NVARCHAR データ型として格納されているデータを SQL Anywhere データベースからエクスポートする場合、データは VARCHAR (Excel ドライバによってサポートされる最も近い型) として格納されます。

Microsoft Excel ODBC ドライバがサポートするテキストカラムの幅は、255 文字までです。

○REAL、FLOAT、BIGINT データ型として格納されているデータは、エクスポートできます。

○MONEY データ型と SMALLMONEY データ型として格納されているデータは、CURRENCY データ型にエクスポートされます。それ以外の場合、数値データは数値としてエクスポートされます。

○テーブルは、**エクスポートウィザード**を使用してエクスポートできます。

「データエクスポート」『SQL Anywhere サーバ SQL の使用法』を参照してください。

SQL Anywhere モニタの新機能

次に、バージョン 12.0.0 で導入された SQL Anywhere モニタの追加機能を示します。

- **新しいユーザインタフェース**　モニタのインタフェースは、ダッシュボードベースです。ダッシュボードには、メトリック、警告、リソース情報を表示するためのウィジェットがあります。ダッシュボードは、各ユーザに固有です。どのユーザも各自のダッシュボードを追加、編集、または削除できます。「ダッシュボード」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Mobile Link サーバファームと Relay Server ファームのモニタリング**　モニタを使用して、SQL Anywhere データベースと Mobile Link サーバだけでなく、Mobile Link サーバファームと Relay Server ファームもモニタリングできます。「SQL Anywhere モニタ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **モニタ内からの接続の切断**　モニタ内から、リソースデータベースへの接続を切断できます。「モニタを使用した接続の切断」『SQL Anywhere サーバ データベース管理』を参照してください。

- **モニタリングする SQL Anywhere リソースのインポート**　リソースのリストを含む CSV ファイルを作成して、このリストをモニタにインポートできます。以前は、モニタに一度に追加できるリソースは 1 つのみでした。「CSV ファイルによる複数のリソースの追加」『SQL Anywhere サーバ データベース管理』を参照してください。

- **オンデマンドでのメンテナンスの実行**　管理者は、スケジュールされていないメンテナンスをモニタで実行できます。「モニタのバックアップ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **SQL Anywhere リソースの新しいバックアップ警告**　管理者は、SQL Anywhere リソースが指定日数の間に正常にバックアップされていない場合、警告を発行するようにモニタを設定できます。デフォルトでは、リソースが正常にバックアップされてから 14 日が経過した後で警告が発行されます。「警告スレッショルドの指定」『SQL Anywhere サーバ データベース管理』を参照してください。

- **ブラウザのローカル時刻によるレポート**　モニタでレポートされる時刻は、常に、使用しているブラウザのローカル時刻です。ブラウザとリソース間の時刻の差異を確認するには、「時刻の表示方法の概要」『SQL Anywhere サーバ データベース管理』を参照してください。

- **メトリックのエクスポート**　グラフや表を関連付けられたメトリックを XML ファイルにエクスポートできます。たとえば、**[主要なパフォーマンスのメトリック]** ウィジェットのほとんどのメトリックをエクスポートできます。「メトリックのエクスポート」『SQL Anywhere サーバ データベース管理』を参照してください。

- **トラブルシューティング機能**　モニタをトラブルシューティングする必要がある場合、管理者は **[メッセージログ]** 機能と **[例外レポート]** 機能を使用できます。「メッセージログの表示。」『SQL Anywhere サーバ データベース管理』と「例外レポートの表示」『SQL Anywhere サーバ データベース管理』を参照してください。

SQL Anywhere モニタの動作の変更

次に、バージョン 12.0.0 で導入された SQL Anywhere モニタの変更点を示します。

- **メトリック収集の変更**　以前は、モニタで収集するメトリックのタイプや警告スレッショルドを設定できましたが、警告スレッショルドのみが設定できるようになりました。「SQL Anywhere モニタ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **リソースの削除**　リソースを停止せずに、モニタリング済みのリソースをモニタから削除できます。以前は、削除する前にリソースを停止する必要がありました。「リソースの削除」『SQL Anywhere サーバ データベース管理』を参照してください。

- **リソースの状態がなくなった**　リソースにはステータスのみが存在するようになりました。「ダッシュボード」『SQL Anywhere サーバ データベース管理』を参照してください。

- **警告ステータスの変更**　警告をトリガする条件が存在しない場合、警告ステータスは **[非アクティブ]** に変更されます。ステータスが **[非アクティブ]** の警告は、その警告をトリガした条件は存在しませんが、モニタのユーザが手動で解決していないことを示します。また、モニタで発行された各警告には重大度が割り当てられます。「ダッシュボード」『SQL Anywhere サーバ データベース管理』を参照してください。

- **警告スレッショルドのデフォルトの変更**　以前は、CPU メモリの使用率が 2 つの連続する収集間隔で最大キャッシュサイズの 85 パーセントに達したときに、モニタによって警告が発行されました。デフォルトは 90 パーセントになり、時間スレッショルドは 30 秒になりました。「アラートの定義とスレッショルド」『SQL Anywhere サーバ データベース管理』を参照してください。

- **すべてのユーザがモニタにログインする必要がある**　以前は、モニタへの読み込み専用アクセスにログインは不要でしたが、すべてのユーザがモニタにログインすることが必要になりました。「モニタのユーザ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **ユーザの言語設定の変更**　ユーザがログインすると、そのユーザの言語設定を使用して、モニタに表示される言語と警告に使用される言語が設定されます。以前は、モニタではブラウザで設定された言語が使用されていました。「モニタユーザの編集」『SQL Anywhere サーバ データベース管理』を参照してください。

マニュアルの強化

次に、バージョン 12.0.0 で SQL Anywhere マニュアルに加えられた変更を示します。

- **よくある質問 (FAQ)**　よくある質問の包括的なリストを使用できるようになりました。FAQ は、主に初級レベルと中級レベルのユーザを対象としており、フォーラムのディスカッションで頻繁に発生する、数多くの一般的な質問に対する回答が含まれています。「よくある質問 - SQL Anywhere」『SQL Anywhere 16 紹介』を参照してください。

- **データベースサーバのパフォーマンス警告**　データベースサーバメッセージウィンドウに表示されるパフォーマンス警告に関して、その説明と、パフォーマンス問題の対応策を示す情報のリンクを使用できるようになりました。「データベースサーバパフォーマンス警告のトラブルシューティング」『SQL Anywhere サーバ データベース管理』を参照してください。

- **DocCommentXchange はデフォルトのマニュアル**　DocCommentXchange は、SQL Anywhere 12 のデフォルトのマニュアルセットです。DocCommentXchange は、Web で SQL Anywhere マニュアルにアクセスしたり、マニュアルについて議論するためのオンラインコミュニティです。

- **Sybase.com の新しい開発者センター**　SQL Anywhere マニュアルを補足するには、http://www.sybase.com/developer/library/sql-anywhere-techcorner にある SQL Anywhere 開発者センターにアクセスしてください。技術ホワイトペーパー、FAQ、テクニカルノート、ダウンロード、テックキャストなどを参照し、質問に対する回答を見つけることができます。

バージョン 11.0.1 の新機能

> **注意**
> 廃止予定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、変更される可能性があります。

バージョン 10 以前の SQL Anywhere の新機能と動作変更については、ftp://ftp2.ianywhere.jp/public/tech/dbwnja10.pdf を参照してください。

SQL Anywhere の新機能

次に、バージョン 11.0.1 で導入された SQL Anywhere データベースとデータベースサーバの追加機能を示します。

接続プロパティ

SQL Anywhere バージョン 11.0.1 で追加された接続プロパティは次のとおりです。

- Authenticated
- IsDebugger
- QueryBypassedCosted
- QueryBypassedHeuristic
- QueryBypassedOptimized
- QueryOpened
- QueryDescribedBypass
- QueryDescribedOptimizer
- StatementDescribes
- StatementPostAnnotates
- StatementPostAnnotatesSimple
- StatementPostAnnotatesSkipped
- WaitStartTime
- WaitType

これらのプロパティの説明については、「接続プロパティの一覧」『SQL Anywhere サーバ データベース管理』を参照してください。

データベースプロパティ

SQL Anywhere バージョン 11.0.1 で追加されたデータベースプロパティは次のとおりです。

- Authenticated
- Prepares
- QueryBypassedCosted
- QueryBypassedHeuristic
- QueryBypassedOptimized
- QueryOpened
- QueryDescribedBypass
- QueryDescribedOptimizer
- StatementDescribes
- StatementPostAnnotates
- StatementPostAnnotatesSimple
- StatementPostAnnotatesSkipped

これらのプロパティの説明については、「データベースプロパティの一覧」『SQL Anywhere サーバ データベース管理』を参照してください。

データベースサーバプロパティ

次に、SQL Anywhere バージョン 11.0.1 でのデータベースサーバプロパティの強化を示します。

- ServerEdition

これらのプロパティの説明については、「データベースサーバプロパティの一覧」『SQL Anywhere サーバ データベース管理』を参照してください。

データベースユーティリティ

次に、SQL Anywhere バージョン 11.0.1 でのデータベースユーティリティの強化を示します。

- **サービスユーティリティ (dbsvc) の強化**　Windows で、サービスユーティリティを使用して、Mobile Link Relay Server (rshost)、Relay Server Outbound Enabler (RSOE)、および SQL Anywhere ボリュームシャドウコピーサービス (dbvss11) のサービスを作成できるようになりました。「Windows 用サービスユーティリティ (dbsvc)」『SQL Anywhere サーバ データベース管理』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**　dbunload ユーティリティで次のオプションがサポートされるようになりました。

  - **-cm オプション**　アンロードされるデータベースの dbinit コマンドまたは CREATE DATABASE 文を表示します。

  - **-l オプション**　再構築されたデータベースの AUTOINCREMENT カラムに、次に使用可能な値を保持します。

  「アンロードユーティリティ (dbunload)」『SQL Anywhere サーバ データベース管理』を参照してください。

システムプロシージャとファンクション

次に、SQL Anywhere バージョン 11.0.1 で追加されたシステムプロシージャとファンクションの強化を示します。

- **sa_get_table_definition システムプロシージャ**　新しい sa_get_table_definition システムプロシージャは、指定されたテーブルおよびそのインデックス、外部キー、トリガ、付与されたパーミッションの作成に必要な SQL 文を含む LONG VARCHAR 文字列を返します。「sa_get_table_definition システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **FIRST_VALUE 関数 [集合]**　FIRST_VALUE 関数 [集合] に RESPECT NULLS 句が含まれるようになりました。「FIRST_VALUE 関数 [集合]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **LAST_VALUE 関数 [集合]**　LAST_VALUE 関数 [集合] に RESPECT NULLS 句が含まれるようになりました。「LAST_VALUE 関数 [集合]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_set_http_option システムプロシージャ**　AcceptCharset オプションにより、文字セットをより柔軟に選択できるようになりました。「sa_set_http_option システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

SQL 文

次に、SQL Anywhere バージョン 11.0.1 での SQL の強化を示します。

- **新しい DEFAULT VALUES 句、INSERT 文**　INSERT 文に新しく追加された DEFAULT VALUES 句により、すべてのカラムにデフォルト値を挿入できます。「INSERT 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **CREATE ENCRYPTED DATABASE 文**　この文は、既存のデータベースの暗号化されたコピー (すべてのトランザクションログ、ミラーログ、DB 領域を含む) を作成します。この文を使用して、テーブル暗号化が有効になっているデータベースのコピーも作成できます。「CREATE ENCRYPTED DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  リカバリが必要なデータベースを暗号化したい場合 (テクニカルサポートに送る場合など) は、今までどおり CREATE ENCRYPTED FILE 文を使用する必要があります。「CREATE ENCRYPTED FILE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **CREATE DECRYPTED DATABASE 文**　この文は、既存のデータベースの復号化されたコピー (すべてのトランザクションログ、ミラーログ、DB 領域を含む) を作成します。「CREATE DECRYPTED DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  リカバリが必要なデータベースを復号化したい場合 (テクニカルサポートに送る場合など) は、今までどおり CREATE DECRYPTED FILE 文を使用する必要があります。「CREATE DECRYPTED FILE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ALTER DATABASE 文の強化**　ミラーリング中のデータベースサーバに ALTER DATABASE UPGRADE 文を実行すると、エラーになります。「ALTER DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **MESSAGE 文の強化**　IMMEDIATE 句を指定すると、接続がアイドルであるか要求を行っているかに関係なく、クライアントのメッセージコールバックルーチンによってメッセージが短時間に受信されます。「MESSAGE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **関数、プロシージャ、トリガ、ビューの新規作成または同じ名前のオブジェクトの置換**　新しい OR REPLACE 句により、関数、プロシージャ、トリガ、ビューの新規作成や、同じ名前のオブジェクトが存在する場合は置換することができます。次の項を参照してください。

  - 「CREATE FUNCTION 文 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE PROCEDURE 文 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE TRIGGER 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE VIEW 文」『SQL Anywhere サーバ SQL リファレンス』

- **存在しないデータベースオブジェクトの削除を文で試みてもエラーを表示しない**　新しい IF EXISTS 句により、存在しないデータベースオブジェクトの削除を DROP 文で試みてもエラーを返さないように指定できます。次の項を参照してください。

  - 「DROP EVENT 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP FUNCTION 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP MATERIALIZED VIEW 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP PROCEDURE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP TABLE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP TRIGGER 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP VIEW 文」『SQL Anywhere サーバ SQL リファレンス』

- **新しい INTO LOCAL TEMPORARY TABLE 句、SELECT 文**　SELECT 文に新しく追加された INTO LOCAL TEMPORARY TABLE 句により、ローカルテンポラリテーブルを作成し、SELECT 文の結果セットを設定できます。これまでは、テンポラリテーブル名が # から始まる場合に INTO 句を使用する方法しかありませんでした。「SELECT 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい IF NOT EXISTS 句、CREATE TABLE 文**　CREATE TABLE 文に新しく追加された IF NOT EXISTS 句により、永久テーブル、グローバルテンポラリテーブル、ローカルテンポラリテーブルが既存しない場合に作成できます。「CREATE TABLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **テンポラリプロシージャやテンポラリ関数の作成時に所有者を指定**　CREATE FUNCTION 文および CREATE PROCEDURE 文により、テンポラリプロシージャまたはテンポラリ関数に対して任意で所有者を指定できるようになりました。次の項を参照してください。

  - 「CREATE FUNCTION 文 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE PROCEDURE 文 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』

プログラミングインタフェース

次に、SQL Anywhere バージョン 11.0.1 でのプログラミングインタフェースの強化を示します。

- **新しい ASP.NET プロバイダ**　次の新しい ASP.NET プロバイダは、標準の ASP.NET プロバイダの機能を模倣しますが、データを SQL Server データベースではなく SQL Anywhere データベースに格納します。

  - **メンバーシップ**　ログイン、ログアウト、ユーザおよびパスワードの管理を実現します。
  - **ロール**　ユーザのグループへの割り当て、および単純で簡単なパーミッション管理を実現します。
  - **プロファイル**　ユーザ変数を格納します。
  - **Web パーツパーソナル化**　Web パーツデータの格納を管理し、ユーザによるビューのパーソナル化を実現します。
  - **Web イベント**　ヘルスモニタリング機能と連携し、フラッシュされた Web イベント情報をデータベースに格納します。

  「SQL Anywhere ASP.NET プロバイダ」『SQL Anywhere サーバ プログラミング』を参照してください。

- **Ruby のサポート**　SQL Anywhere は、Ruby オープンソースプログラミング言語をサポートするようになりました。「SQL Anywhere の Ruby API サポート」『SQL Anywhere サーバ プログラミング』を参照してください。

- **OLE DB での CATALOGS と SCHEMATA の各ローセットのサポート**　OLE DB の CATALOGS および SCHEMATA の各ローセットがサポートされるようになりました。SQL Anywhere はカタログの概念をサポートしていないため、SQL Anywhere OLE DB プロバイダは、CATALOGS の代わりに現在開始されているすべてのデータベースとそのロケーションを含む結果セットを返します。同様に SCHEMATA では、結果セット内でカタログとしてデータベース名が使用されます。

- **ADO.NET Entity Framework に追加されたサポート**　SQL Anywhere で、ADO.NET Entity Framework モデルがサポートされるようになりました。「SQL Anywhere .NET データプロバイダの機能」『SQL Anywhere サーバ プログラミング』を参照してください。

その他

次に、SQL Anywhere バージョン 11.0.1 でのその他の強化を示します。

- **単純で低コストな文の実行に関するパフォーマンスの向上**　実行時間が低コストの文のための実行プランを生成するのに要するコストは、文を実行するためのコストを上回る場合があります。次の強化により、SQL Anywhere で認識される、実行時間が低コストの単純な文のクラスが拡張され、このような文がオプティマイザをバイパスできるようになりました。

  詳細については、「クエリ処理のフェーズをスキップするための条件」『SQL Anywhere サーバ SQL の使用法』を参照してください。

○ **新しい FORCE NO OPTIMIZATION 句**　以前までは、コストベースの最適化が必要な文はすべてオプティマイザによって処理されていました。今回、FORCE NO OPTIMIZATION 句が追加されたことで、文がオプティマイザをバイパスするよう指定できるようになりました。文が複雑すぎて (データベースオプションの設定またはスキーマやクエリの特徴などにより) このような処理が実行できない場合、文は失敗し、データベースサーバはエラーを返します。

次の文は、新しい FORCE NO OPTIMIZATION 句をサポートしています。

- 「DELETE 文」『SQL Anywhere サーバ SQL リファレンス』
- 「INSERT 文」『SQL Anywhere サーバ SQL リファレンス』
- 「SELECT 文」『SQL Anywhere サーバ SQL リファレンス』
- 「UPDATE 文」『SQL Anywhere サーバ SQL リファレンス』

○ **新しい接続プロパティ**

- QueryBypassedCosted
- QueryBypassedHeuristic
- QueryBypassedOptimized
- QueryOpened
- QueryDescribedBypass
- QueryDescribedOptimizer
- StatementDescribes
- StatementPostAnnotates
- StatementPostAnnotatesSimple
- StatementPostAnnotatesSkipped

○ **新しいデータベースプロパティ**

- Prepares
- QueryBypassedCosted
- QueryBypassedHeuristic
- QueryBypassedOptimized
- QueryOpened
- QueryDescribedBypass
- QueryDescribedOptimizer
- StatementDescribes
- StatementPostAnnotates
- StatementPostAnnotatesSimple
- StatementPostAnnotatesSkipped

○ **新しい [最適化方法] フィールド**　グラフィカルなプランの [オプティマイザ統計] セクションに、[最適化方法] フィールドが追加されました。このフィールドは、クエリオプティマイザによって選択された実行方式を返します。「実行プランのコンポーネント」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **データベースサーバがデフォルトデータベースサーバにならないようにする**　-xd サーバオプションにより、データベースサーバがデフォルト TCP ポートで受信するのを防止し、データベースサーバがデフォルトデータベースサーバにならないようにします。「-xd データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **並列アーカイブバックアップのサポート**　SQL Anywhere データベースサーバで、サーバ側のアーカイブバックアップに対する並列バックアップがサポートされるようになりました。並列データベースバックアップでは、物理 I/O を利用して、直列ではなく並列で読み書き操作を行うことで、パフォーマンスが向上します。

  BACKUP DATABASE 文に、並列アーカイブバックアップをサポートするための新しい句が 2 つ追加されました。

  - **WITH CHECKPOINT LOG [NO] COPY**
  - **MAX WRITE** { *n* | **AUTO** }

  バージョン 11.0.0 以前のデータベースサーバでは、バージョン 11.0.1 のデータベースサーバで生成されたアーカイブバックアップをリストアできません。バージョン 11.0.1 のデータベースサーバでは、旧バージョンのデータベースサーバで生成されたバックアップをリストアできます。

  「BACKUP 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **Mac OS X での Developer Edition と Evaluation Edition のデータベースサーバの実行**　Mac OS X で、Developer Edition と Evaluation Edition のデータベースサーバを管理ツールから自動的に起動できるようになりました。

SQL Anywhere の動作の変更と廃止予定機能

次に、バージョン 11.0.1 で導入された SQL Anywhere データベースとデータベースサーバに加えられた変更を、カテゴリごとに示します。

動作の変更

- **全文検索**　全文検索をサポートするよう、次の動作変更が行われました。

  - **演算子の優先度が適用される**　以前までは、クエリ文字列に演算子の優先度は適用されていませんでした。このリリースでは、次の演算子の優先度が適用されるようになりました。

    - NEAR、FUZZY 演算子
    - AND NOT 演算子
    - AND 演算子
    - OR 演算子

    「CONTAINS 探索条件における演算子の優先度」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **NEAR 句の引数は単語またはプレフィクス単語にする**　近接検索を実行する場合、NEAR 句の引数は単語またはプレフィクス単語である必要があります。「CONTAINS 探索条件」『SQL Anywhere サーバ SQL リファレンス』と「近接検索」『SQL Anywhere サーバ SQL の使用法』を参照してください。

  - **ハイフンおよび AND NOT 句の使用**　フレーズ内では、ハイフンは特殊文字ではなく単語区切りとして処理されます。フレーズの外にあるハイフンの処理は、ハイフンを囲む構文

によって異なります。「ハイフン (-) を使用できる構文」『SQL Anywhere サーバ SQL リファレンス』と「全文検索での AND NOT 演算子の使用」『SQL Anywhere サーバ SQL の使用法』を参照してください。

○ **アスタリスクおよびプレフィクス検索の使用**　プレフィクス検索を実行する場合、単語にアスタリスクを付加し、クエリ文字列の末尾とするか、またはそのすぐ後にスペースか許可された特殊文字を 1 つ挿入する必要があります。「アスタリスク (*) を使用できる構文」『SQL Anywhere サーバ SQL リファレンス』と「プレフィクス検索」『SQL Anywhere サーバ SQL の使用法』を参照してください。

○ **重複するテキストインデックスを作成するとエラーが返される**　重複するテキストインデックスを作成できなくなりました。次の設定が既存のテキストインデックスと同じ場合、そのテキストインデックスは重複するものと見なされます。

●参照されるベーステーブル

●インデックス対象カラム (順序は問わない)

●使用される設定オブジェクトの設定内容 (TERM BREAKER、MINIMUM TERM LENGTH、MAXIMUM TERM LENGTH、STOPLIST、照合情報)

SQL Anywhere 11.0.1 よりも以前のバージョンで作成された重複するテキストインデックスは、データベース内に残すことができ、11.0.1 のデータベースサーバで開始してもエラーは発生しません。ただし、重複するテキストインデックスを含むデータベースをバージョン 11.0.1 以降で再ロードすると、エラーが返されます。

既存のデータベース内にある重複するテキストインデックスを識別するには、次のクエリを実行します。

```
SELECT LIST( i.index_name )
FROM SYS.SYSIDX i
   JOIN SYS.SYSTEXTIDX t ON i.object_id = t.index_id AND t.sequence = 1
   JOIN SYS.SYSTEXTCONFIG F ON F.object_id = t.text_config
   JOIN (
   SELECT table_id, index_id, LIST( column_id, ', ' ORDER BY column_id ) col_id
   FROM SYS.SYSIDXCOL
   GROUP BY table_id, index_id) x
   ON x.table_id = i.table_id AND x.index_id = i.index_id
WHERE i.index_category=4
GROUP BY i.table_id, f.term_breaker, f.min_term_length, f.max_term_length,
   f.collation, ISNULL( f.char_stoplist, '-' ),
   ISNULL( f.nchar_stoplist, '-' ), x.col_id
HAVING count(*) > 1
```

このクエリは、ストップリストを表す文字列がまったく同じであるか、NO STOPLIST が指定されている場合のみ有効です。たとえば、ストップリスト 'a b c' とストップリスト 'a - b c' は、このクエリでは同じものとは見なされませんが、テキストインデックスを作成する際の重複のチェックでは同じものと見なされます。

● **正規表現**　SIMILAR TO、REGEXP の各探索条件、および REGEXP_SUBSTR 関数の動作に変更が加えられました。変更の主な目的は、SIMILAR TO の ANSI/SQL 標準との整合性を保ちながら、REGEXP と REGEXP_SUBSTR の動作を Perl と整合させることにあります。

○ **データベースの照合および一致**　以前までは、REGEXP と REGEXP_SUBSTR は照合等価やソート順を使用して、パターン内のリテラルクラスや文字クラスの範囲が文字列と一致するかを判断していました。このリリースでは、REGEXP と REGEXP_SUBSTR はコードポイント値のバイナリ比較を使用して、範囲の一致や評価を行っています。この変更は、Perl 5.0 と動作の整合性が保たれるようにするため行われています。

SIMILAR TO は依然としてデータベース照合を使用して一致や範囲の評価を行っています。「LIKE 探索条件、REGEXP 探索条件、SIMILAR TO 探索条件」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

○ **データベースの大文字と小文字の区別および [[:upper:]] と [[:lower:]] の部分文字クラス**　大文字と小文字を区別しないデータベースでは、SIMILAR TO と REGEXP の [[:upper:]] と [[:lower:]] の部分文字クラスは区別されていませんでした。データベースでの大文字と小文字の区別にかかわらず、[[:upper:]] は大文字とだけ一致し、[[:lower:]] は小文字とだけ一致するように、これは変更されました。

○ **メタ文字としての脱字記号 (^)、アンダースコア (_)、パーセント記号 (%) の処理**　次の表に、メタ文字としての各文字がどのように処理されるかを、以前の動作と新しい動作に分けて示します。

文字	以前の動作	新しい動作
_ (アンダースコア)	SIMILAR TO、REGEXP、および REGEXP_SUBSTR では、アンダースコアはメタ文字として処理され、任意の 1 文字と一致していました。	SIMILAR TO では、アンダースコアはメタ文字として処理され、任意の 1 文字と一致します。 REGEXP と REGEXP_SUBSTR では、アンダースコアはメタ文字として処理されません。REGEXP と REGEXP_SUBSTR で任意の 1 文字と一致させるためには、代わりにピリオド (.) を使用します。

文字	以前の動作	新しい動作
%	SIMILAR TO、REGEXP、および REGEXP_SUBSTR では、パーセント記号はメタ文字として処理され、0 個以上の任意の文字列と一致していました。	SIMILAR TO では、パーセント記号はメタ文字として処理され、0 個以上の任意の文字列と一致します。 REGEXP と REGEXP_SUBSTR では、パーセント記号はメタ文字として処理されません。REGEXP と REGEXP_SUBSTR で任意の数の任意の文字と一致させるためには、代わりにピリオドとアスタリスク (.*) を使用します。
^	SIMILAR TO、REGEXP、および REGEXP_SUBSTR では、脱字記号は右にある文字の否定や減算として処理され、文字クラスの否定として解釈されていました。	SIMILAR TO では、脱字記号は右にある文字の否定や減算として処理されます。たとえば、SIMILAR TO [a-d^c] は a、b、d とは一致しますが、c とは一致しません。 REGEXP と REGEXP_SUBSTR では、脱字記号は文字クラス内で先頭に位置する場合のみメタ文字として処理され、文字クラスの否定として解釈されます。たとえば、REGEXP [^abc] は a、b、c 以外の任意の 1 文字と一致し、REGEXP [a-d^c] は a、b、c、d、および ^ と一致します。

- **アップグレードユーティリティ (dbupgrad) の動作の変更**　データベースミラーリングに参加しているデータベースをアップグレードするために、アップグレードユーティリティを使用するとエラーになります。「アップグレードユーティリティ (dbupgrad)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Mac OS X で dbmodenv が不要になった**　以前のリリースでは、Mac OS X でグラフィカルな管理ツールを使用するためには、ユーザの *$HOME/.MacOSX/environment.plist* ファイルの PATH と DYLD_LIBRARY_PATH に、SQL Anywhere のバイナリとライブラリのロケーショ

ンを追加する必要がありました。そのために dbmodenv ツールを使用していました。SQL Anywhere は、*$HOME/.MacOSX/environment.plist* の設定に依存しなくなったため、dbmodenv を実行したり、SQL Anywhere をインストールした後にログアウトし、ログインし直す必要がなくなりました。

- **dbisqlc の OUTPUT 文から返される NULL 値のデフォルトが変更**　以前のリリースでは、dbisqlc の OUTPUT 文を使用すると、文は NULL 値として **(NULL)** 値を返していました。このリリースでは、文で NULL 値のデフォルトとして空の文字列が返されるようになりました。output_nulls オプションを設定することで、NULL 値のエクスポート方法を変更できます。「output_nulls オプション [Interactive SQL]」『SQL Anywhere サーバ データベース管理』を参照してください。

- **エンディアンのサポート**　アップグレード後、ビッグエンディアンコンピュータで作成された 11.0.1 よりも前のテキストインデックスは、トランケートおよびリフレッシュ (MANUAL REFRESH および AUTO REFRESH のテキストインデックスの場合)、または再作成 (IMMEDIATE REFRESH のインデックスの場合) が必要になります。

- **Mac OS X の管理ツール**　Mac OS X の SQL Anywhere 管理ツールで、64 ビットの JDK 1.6 が使用されるようになりました。管理ツールは、Apple JDK 1.6 (Mac OS X 10.5.2 以降) でサポートされている、64 ビットのプロセッサを搭載した Intel ベースの Macintosh コンピュータだけで動作します。Mac OS X 用の管理ツールを配備する場合、ネイティブライブラリは *$SQLANY11/System/lib64* にあります。「Linux、Solaris、Mac OS X における管理ツールの配備」『SQL Anywhere サーバ プログラミング』を参照してください。

- **HTTP クライアントのチャンクされた転送コーディングの新しいデフォルトサイズ**　これまでは、HTTP クライアントから 2048 バイトを超えるデータが送信された場合、デフォルトで (またはユーザが CREATE PROCEDURE...SET 'HTTP(CH=auto)' を指定した場合)、チャンクされた転送コーディングが試行されました。デフォルトのサイズが 2048 バイトから 8196 バイトに変更されました。また、チャンクされた転送コーディングを使用しないで要求を再発行する基準に、新しいステータス 411 **Length Required** が追加されました。「CREATE PROCEDURE 文 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ansi_substring オプションのサポート**　ansi_substring オプションはバージョン 11.0.0 で非推奨になりましたが、バージョン 11.0.1 でまたサポートされるようになりました。「ansi_substring オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

廃止予定機能とサポート終了機能

- **COMMENT ON EXTERNAL ENVIRONMENT OBJECT *object-name***　この構文は **COMMENT ON EXTERNAL OBJECT** *object-name* に変更されました。以前の構文は現在でも受け入れられますが、今後のリリースではサポートされない可能性があります。「COMMENT 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

Mobile Link の新機能

次に、バージョン 11.0.1 で導入された Mobile Link の追加機能を示します。

- **リモートデータベーススキーマのスキーマキャッシュ**　新しいスキーマキャッシュ機能により、小規模な同期のオーバヘッドが縮小されます。リモートスキーマは、最初の同期で Mobile Link サーバによってキャッシュされます。以降の同期では、スキーマがキャッシュされていない場合にのみリモートスキーマ情報が Mobile Link サーバに送信されます。

- **mlsrv11 の -vi オプション**　アップロードされた各ローのカラム値を表示します。「-v mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **mlsrv11 の -vq オプション**　ダウンロードされた各ローのカラム値を表示します。「-v mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **mlsrv11 の -vm オプション**　同期が完了するたびに、各同期と各同期フェーズの継続時間をログに出力します。「-v mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **mlsrv11 の -ppv オプション**　指定された期間に従って、Mobile Link が新しい定期モニタリングの値を出力するようにします。「-ppv mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **ml_ignore プレフィクス**　Mobile Link サーバでは、--{ml_ignore} で始まる SQL スクリプトが、意図的に無視するスクリプトとして認識されます。「無視されたスクリプト」『Mobile Link サーバ管理』を参照してください。

- **dblsn の -sv オプション**　データベースを認証するために Mobile Link Listener が使用するスクリプトバージョンを指定します。「-sv dblsn オプション」『Mobile Link サーバ起動同期』を参照してください。

- **Oracle VArray のサポート**　iAnywhere Solutions 11 - Oracle ODBC ドライバで、ストアドプロシージャでの Oracle VArray の使用がサポートされるようになりました。「Oracle VARRAY」『Mobile Link サーバ管理』を参照してください。

- **ライトウェイトポーリングのリスナキーワード変数**　ライトウェイトポーリングをサポートする、poll_connect、poll_notifier、poll_key、poll_every の各リスナキーワードが追加されました。

- **ライトウェイトポーリングのアクション変数**　ライトウェイトポーリングをサポートする、$poll_connect、$poll_notifier、$poll_key、$poll_every の各アクション変数が追加されました。

- **Common Access Card を使用したクライアント認証**　Mobile Link クライアントで、Common Access Card (CAC) のクライアント ID を使用した認証がサポートされるようになりました。「identity_name」『Mobile Link クライアント管理』を参照してください。

> **注意**
> この機能は CAC 認証アドオンの一部であり、別途ライセンスが必要です。「別途ライセンスが必要なコンポーネント」『SQL Anywhere 16 紹介』を参照してください。

- **Microsoft SQL Server 2008 のサポート**　Mobile Link 同期サーバで Microsoft SQL Server 2008 上で実行される統合データベースがサポートされるようになりました。Microsoft の新しい DATE、TIME、DATETIME2 データ型のマッピングの詳細については、「Microsoft SQL Server データのマッピング」『Mobile Link サーバ管理』を参照してください。

- **.NET DownloadTableData インタフェースの新しいメソッド**　getLastDownloadTime メソッドによってテーブルの最終ダウンロード時刻が返されます。DownloadTableData.GetLastDownloadTime メソッド [Mobile Link サーバ .NET]『Mobile Link サーバ管理』を参照してください。

- **対象 Mobile Link ユーザとリモート ID に対するログの冗長性**　対象とする Mobile Link ユーザまたはリモート ID のログに、別の冗長性を設定できるようになりました。「対象 Mobile Link ユーザとリモート ID に対するログの冗長性」『Mobile Link サーバ管理』を参照してください。

- **モデルモードでの MySQL のサポート**　Mobile Link プラグインで MySQL 統合データベースがサポートされるようになりました。

- **mlmon の -c オプション**　Mobile Link モニタ用に、mlmon コマンドに -c オプションが追加されました。-c オプションは、接続の最後で Mobile Link モニタを閉じ、指定したデータベースにセッションを保存します。

Mobile Link の動作の変更と廃止予定機能

次に、バージョン 11.0.1 で導入された Mobile Link の変更点を示します。

- **mlsrv11 の -sm オプション**　mlsrv11 の -sm オプションが改善され、非永続的な HTTP/HTTPS に使用したときに -nc オプションに似た機能を提供するようになりました。「-sm mlsrv16 オプション」『Mobile Link サーバ管理』と「-nc mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **Microsoft SQL Server データ型**　SQL Server 2005 で、TEXT、NTEXT、IMAGE の各データ型が推奨されなくなりました。代わりに VARCHAR(*max*)、NVARCHAR(*max*)、VARBINARY(*max*) を使用してください。「Microsoft SQL Server データのマッピング」『Mobile Link サーバ管理』を参照してください。

Ultra Light の新機能

次に、バージョン 11.0.1 で導入された Ultra Light の追加機能を示します。

- **ミラーファイル**　Ultra Light には、潜在的に信頼性が低いストレージシステムのフォールトトレランスを改善する基本的なデータベースファイルミラーリング機能があります。ミラーリングには、ミラーファイルが使用されます。データベースへのすべての書き込みは、メインデータベースファイルに対して発行されると同時にミラーファイルにも発行されます。「Ultra Light MIRROR_FILE 接続パラメータ」『Ultra Light データベース管理とリファレンス』を参照してください。

- **ml_remote_id の設定解除**　SET OPTION を使用して **ml_remote_id** の設定を解除できるようになりました。「SET OPTION 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』と「ML_GET_SERVER_NOTIFICATION 関数 [システム]」『Ultra Light データベース管理とリファレンス』を参照してください。

- **ML_GET_SERVER_NOTIFICATION**　この関数を使用すると、Ultra Light ユーザはライトウェイトポーリングを使用して、サーバ起動同期要求について Mobile Link サーバ上の Notifier に問い合わせできます。「ML_GET_SERVER_NOTIFICATION 関数 [システム]」『Ultra Light データベース管理とリファレンス』を参照してください。

- **SYNC_PROFILE_OPTION_VALUE 関数**　この関数は、同期プロファイル内の指定されたオプションの値を返します。「SYNC_PROFILE_OPTION_VALUE 関数 [システム]」『Ultra Light データベース管理とリファレンス』を参照してください。

- **Ultra Light.NET API の StreamErrorParameters プロパティ**　**StreamErrorParameters** が **SyncResult** クラスに追加されました。このメンバーには、StreamErrorCode でレポートされるストリームエラーコードのエラーパラメータをカンマで区切ったリストが含まれます。ULSyncResult.StreamErrorParameters プロパティ [Ultra Light.NET]『Ultra Light .NET プログラミング』を参照してください。

- **Ultra Light for M-Business API の getStreamErrorParameters メソッド**　このメソッドは、同期ストリーム処理によってレポートされるエラーパラメータをカンマで区切ったリストを返します。

Ultra Light の動作の変更と廃止予定機能

次に、バージョン 11.0.1 で導入された Ultra Light の変更点を示します。

- **Palm のサスペンド機能**　推奨されなくなりました。

Ultra Light J の新機能

次に、バージョン 11.0.1 で導入された Ultra Light J の追加機能を示します。

- **追加 SQL サポート**　Ultra Light J には、次の追加 SQL サポートがあります。

  ○ALTER TABLE

  ○CREATE TABLE

  ○CREATE INDEX

  ○DROP INDEX

  ○DROP TABLE

  ○TRUNCATE TABLE

  また、次の制約がなくなりました。

  ○HAVING がサポートされます。

  ○集合関数内の DISTINCT がサポートされます。

○CURRENT TIME、CURRENT TIMESTAMP、CURRENT DATE がサポートされます。

「Ultra Light SQL 文」『Ultra Light データベース管理とリファレンス』を参照してください。

- **DatabaseInfo の新しいメソッド**　**getPageReads()** と **getPageWrites** の 2 つの新しいメソッドが **DatabaseInfo** インタフェースに追加されました。これらのメソッドは、DatabaseInfo オブジェクトの作成時点のページ読み込みとページ書き込みの数を返します。DatabaseInfo.getPageReads メソッド [Ultra Light J]『Ultra Light Java プログラミング』と DatabaseInfo.getPageWrites メソッド [Ultra Light J]『Ultra Light Java プログラミング』を参照してください。

- **Ultra Light J データベース転送ユーティリティの更新**　Ultra Light J データベース転送ユーティリティで、クライアントからのデータベースの転送に加えて、データベースの削除、データベース情報の表示、データベース転送ログの表示または電子メール送信が可能になりました。「Ultra Light Java Edition データベース転送ユーティリティ」『Ultra Light データベース管理とリファレンス』を参照してください。

Ultra Light J の動作の変更と廃止予定機能

次に、バージョン 11.0.1 で導入された Ultra Light J の変更点を示します。

- **@@identity グローバル変数**　**@@identity** グローバル変数は Ultra Light J でサポートされていません。

管理ツールの新機能

次に、バージョン 11.0.1 で導入された Sybase Central と Interactive SQL の追加機能を示します。

Sybase Central の新機能

次に、バージョン 11.0.1 で導入された Sybase Central のプラグインの追加機能を示します。

すべてのプラグイン

- **サービス作成ウィザードの強化**　サービス作成ウィザードを使用して、Mobile Link Relay Server (rshost)、Relay Server Outbound Enabler (RSOE)、SQL Anywhere ボリュームシャドウコピーサービス (dbvss11)、Mobile Link Listener ユーティリティ (dblsn)、Broadcast Repeater ユーティリティ (dbns11) のサービスを作成できるようになりました。「サービスまたはデーモンとしてデータベースサーバを実行する方法」『SQL Anywhere サーバ データベース管理』を参照してください。

SQL Anywhere プラグイン

- **Sybase Central 内でイベントのステータスを表示**　Sybase Central で、イベントの現在の実行ステータスを確認できるようになりました。**[イベント]** フォルダの新しい **[実行中]** カラムに、イベントの現在の実行ステータスが表示されます。左ウィンドウ枠でフォルダを選択す

ると、フォルダの内容が更新されます。また、**[イベントのプロパティ]** ウィンドウに **[実行中]** プロパティが表示されます。

この値は、イベントが実行中の場合は **[はい]**、イベントが実行中ではない場合は **[いいえ]**、イベントが SQL Anywhere 9.0.2 以前のデータベースにある場合は **[不明]** になります。

- **Sybase Central 内でのメンテナンスプランのステータスの表示**　Sybase Central で、メンテナンスプランの現在の実行ステータスを確認できるようになりました。**[メンテナンスプラン]** フォルダの新しい **[実行中]** カラムに、プランの現在の実行ステータスが表示されます。左ウィンドウ枠でフォルダを選択すると、フォルダの内容が更新されます。

  この値は、メンテナンスプランが実行中の場合は **[はい]**、プランが実行中ではない場合は **[いいえ]**、プランが SQL Anywhere 9.0.2 以前のデータベースにある場合は **[不明]** になります。

- **メンテナンスプランの強化**　メンテナンスプラン作成ウィザードで、メンテナンスプランレポートを電子メールで送信する設定を行う際に、テスト電子メールを送信できるようになりました。「メンテナンスプラン」『SQL Anywhere サーバ データベース管理』を参照してください。

- **プロシージャ作成ウィザードとファンクション作成ウィザードの強化**　ファンクション作成ウィザードとプロシージャ作成ウィザードで、プロシージャまたはファンクションを記述する SQL ダイアレクトまたは言語を、Watcom-SQL、Transact-SQL、外部 C/C++、または外部環境から選択できるようになりました。外部環境を選択した場合、C_ESQL32、C_ESQL64、C_ODBC32、C_ODBC64、CLR、JAVA、PERL、PHP のいずれかの言語を選択できます。これまでは、Watcom-SQL または Transact-SQL だけを選択できました。C/C++ または Java を選択すると、EXTERNAL NAME 句を含む関数またはプロシージャのコードスケルトンが生成されます。「CREATE FUNCTION 文 [外部呼び出し]」『SQL Anywhere サーバ SQL リファレンス』と「CREATE PROCEDURE 文 [外部呼び出し]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

Mobile Link プラグイン

- **Oracle データベース用の Mobile Link モデルの強化**　同期モデル作成ウィザードを使用して Oracle 統合データベースを使用する同期モデルを作成するときに、スキーマ全体をロードするか、同期に必要なテーブルの所有者のサブセットを選択するかを選べるようになりました。所有者のサブセットを選択すると、スキーマのロード時間を短縮できます。

- **Mobile Link モデルモードでの MySQL のサポート**　Mobile Link モデルモードで、MySQL 統合データベースの使用がサポートされるようになりました。

管理ツールの動作の変更と廃止予定機能

次に、バージョン 11.0.1 で導入された Sybase Central と Interactive SQL の変更点を示します。

Sybase Central の動作の変更と廃止予定機能

次に、バージョン 11.0.1 で導入された Sybase Central の変更点を示します。

SQL Anywhere プラグイン

- **読み込み専用データベースの強化**　読み込み専用データベースに接続するときに警告が表示されるようになりました。この警告が表示されないようにするには、**[ツール]** » **[SQL Anywhere 11]** » **[ユーザ設定]** をクリックして設定します。

Interactive SQL の動作の変更と廃止予定機能

次に、バージョン 11.0.1 で導入された Interactive SQL の変更点を示します。

- **データベースロックの自動解放の設定**　結果セットが表示されるときに作成されるデータベーススキーマロックを解放するように Interactive SQL を設定できるようになりました。このように設定するには、Interactive SQL で **[ツール]** » **[オプション]** » **[SQL Anywhere]** » **[データベースロックの自動解放]** をクリックします。

  このオプションを選択すると、結果セットを返す文を実行した後、データベースにコミットされていない変更が接続にあるかどうかが Interactive SQL によって確認されます。コミットされていない変更がなかった場合は Interactive SQL によってスキーマロックが解放されます。それ以外の場合、スキーマロックは解放されません。つまり、データベースにコミットされていない変更がある場合、スキーマロックは解放されません。

製品全体の新機能

次に、バージョン 11.0.1 で導入された製品全体の追加機能を示します。

- **新しいオンラインマニュアルフォーラム、DocCommentXchange (DCX)**　DocCommentXchange と呼ばれる新しいオンラインフォーラムが作成されました。DocCommentXchange は、SQL Anywhere マニュアルにアクセスし、マニュアルについて議論するためのコミュニティです。

  DocCommentXchange は次の目的に使用できます (現在のところ、日本語はサポートされておりません)。

  ○マニュアルを表示する

  ○マニュアルの項目について明確化するために、ユーザによって追加された内容を確認する

  ○すべてのユーザのために、今後のリリースでマニュアルを改善するための提案や修正を行う

  http://dcx.sybase.com を参照してください。

- **SQL Anywhere モニタ**　SQL Anywhere モニタは、SQL Anywhere データベースや Mobile Link サーバの正常性や可用性に関する情報を示す、ブラウザベースの管理ツールです。

  Mobile Link では、この機能は既存の Mobile Link モニタの機能に代わるものではなく、また Mobile Link モニタの機能と重複しません。

  「SQL Anywhere モニタ」『SQL Anywhere サーバ データベース管理』を参照してください。

マニュアルの強化

- **SQL Anywhere のバックアップとリカバリのマニュアルの強化**　SQL Anywhere に付属のバックアップツールおよびリカバリツールを使用するためのマニュアルが、今回のリリースに合わせて書き直されました。「バックアップとデータリカバリ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **全文検索のマニュアルの強化**　全文検索機能のマニュアルの構成が変更され、例やチュートリアルが追加されました。「全文検索」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **Mobile Link サーバ起動同期のマニュアル**　Mobile Link サーバ起動同期のマニュアルが、バージョン 11.0.0 から改善されました。これには、目次の構成の改善、サーバ起動同期の説明の詳細化、リファレンス情報のフォーマットの改善などが含まれます。Mobile Link サーバ起動同期を参照してください。

- **SQL Remote のマニュアル**　SQL Remote のマニュアルは、構成の変更と改訂が行われ、使いやすくなりました。SQL Remote を参照してください。

バージョン 11.0.0 の新機能

バージョン 10 以前の SQL Anywhere の新機能と動作変更については、ftp://ftp2.ianywhere.jp/public/tech/dbwnja10.pdf を参照してください。

> **注意**
> 廃止予定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、変更される可能性があります。

SQL Anywhere

- 「SQL Anywhere の新機能」216 ページ
- 「SQL Anywhere の動作の変更」244 ページ
- 「SQL Anywhere の廃止予定機能とサポート終了機能」253 ページ

Mobile Link

- 「Mobile Link の新機能」257 ページ
- 「Mobile Link の動作の変更と廃止予定機能」261 ページ

SQL Remote

- 「SQL Remote の新機能」263 ページ
- 「SQL Remote の動作の変更と廃止予定機能」263 ページ

Ultra Light

- 「Ultra Light の新機能」263 ページ
- 「Ultra Light の動作の変更と廃止予定機能」269 ページ

Sybase Central と Interactive SQL

- 「Sybase Central と Interactive SQL の新機能」270 ページ
- 「Sybase Central と Interactive SQL の動作の変更と廃止予定機能」274 ページ

マニュアルの強化

- 「マニュアルの強化」279 ページ

製品全体の機能

- 「製品全体の新機能」280 ページ
- 「製品全体の動作の変更」281 ページ

SQL Anywhere

次の項では、SQL Anywhere バージョン 11.0.0 の新機能、動作の変更、廃止予定機能について説明します。

SQL Anywhere の新機能

次に、バージョン 11.0.0 で導入された SQL Anywhere データベースとデータベースサーバの追加機能を示します。

主な機能

次に、SQL Anywhere バージョン 11.0.0 の主な機能を示します。

- **テーブルのマージのサポート**　SQL Anywhere では、テーブル、ビュー、システムプロシージャの結果をテーブルまたはビューにマージできるようになりました。「MERGE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ログインポリシーのサポート**　SQL Anywhere では、ログインポリシーがサポートされるようになりました。ログインポリシーは、ユーザに対してデータベース接続が確立されるときに適用するルールを定義する一連のオプションです。SQL Anywhere には、すべてのログインポリシーのデフォルト値を含むルートポリシーがあります。ルートログインポリシーに優先する設定を含む個別のログインポリシーを作成できます。ログインポリシーは各ユーザに個別に割り当てます。各ユーザが所属するグループからログインポリシーを継承することはありません。特定のユーザへのポリシーの割り当ては、必要に応じて変更できます。「ログインポリシー」『SQL Anywhere サーバ データベース管理』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。「SQL Anywhere サーバのアップグレード」289 ページを参照してください。

- **全文検索のサポート**　SQL Anywhere で全文検索がサポートされるようになりました。全文検索を行うと、データベース内で特定の単語が出現するすべての箇所を簡単に検索できます。全文検索は、単語単位であるという点、またテーブルのローをスキャンするのではなくテキストインデックスを使用する点で、LIKE、REGEXP、SIMILAR TO などの述部を使用した検索とは異なります。「全文検索」『SQL Anywhere サーバ SQL の使用法』を参照してください。

  全文検索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。「SQL Anywhere サーバのアップグレード」289 ページを参照してください。

- **正規表現のサポート**　SQL Anywhere では、REGEXP と SIMILAR TO の 2 つの探索条件を使用した正規表現がサポートされるようになりました。「REGEXP 探索条件」『SQL Anywhere サーバ SQL リファレンス』と「SIMILAR TO 探索条件」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  「正規表現の概要」『SQL Anywhere サーバ SQL リファレンス』と「LIKE 探索条件、REGEXP 探索条件、SIMILAR TO 探索条件」『SQL Anywhere サーバ SQL リファレンス』も参照してください。

- **データベースオプション設定のトランザクションログへの記録**　LOAD 操作時に有効なデータベースオプション設定がトランザクションログに記録されるようになりました。このため、最初の LOAD 操作と、リカバリ時にトランザクションログを適用して実行される最後の LOAD 操作で、date_order や nearest_century などのオプションが違っても、データの矛盾が生じません。

- **オプティマイザでのインデックス使用の強化**　SQL Anywhere のインデックス機能がいくつかの点で強化されました。これらの新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。「SQL Anywhere サーバのアップグレード」289 ページを参照してください。

  - **複合インデックススキャンのサポート**　オプティマイザが強化され、複数のインデックス (最大 4 つまで) を考慮して、ベーステーブルの複数の述部に基づいてテーブルからデータが取り出されるようになりました。これまでは、クエリのインデックスヒントとしてインデックスを 1 つだけ指定できました。SELECT 文の WITH 句の新しいインデックスヒントを使用すると、複合インデックススキャンを使用可能なことを指定できます。「FROM 句」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

    複合インデックススキャンという新しいテーブルアクセスアルゴリズムが追加されました。

  - **インデックス専用取得のサポート**　オプティマイザが強化され、インデックス専用取得がサポートされるようになりました。インデックス専用取得では、テーブル内の対応するローにアクセスする必要がなく、インデックスのデータを使用してクエリが処理されます。オプティマイザは、可能な場合は常にインデックス専用取得を実行します。INDEX ONLY {ON | OFF} ヒントを使用して、インデックス専用取得を実行するかどうかを制御できます。「FROM 句」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **データのロードとアンロードの強化**　データのロードとアンロードが次のように強化されています。

  - **クライアントコンピュータにあるファイルからのデータのロードとファイルへのデータのアンロード**　SQL 文と関数は、データベースサーバに存在するデータの読み込みと書き込みに使用されます。今回実装された新しい機能では、この機能をクライアントコンピュータにあるファイルにも適用できるようになったので、クライアントのファイルをデータベースサーバにコピーする必要がなくなりました。このデータの転送は効率的に行われる一方、クライアントコンピュータ上のデータに対するセキュリティとアクセス制御が実現します。

    クライアントコンピュータにあるファイルの実際の読み込みはクライアントライブラリによって透過的に行われます。したがって、既存のクライアントアプリケーションで新しい SQL 言語のサポートを使用することで、この新機能をすぐに利用できます。

    この新機能を利用するには、クライアントとデータベースサーバの両方が SQL Anywhere バージョン 11.0.0 であり、またクライアントで Command Sequence 通信プロトコル (CmdSeq) を使用する必要があります。

    「クライアントコンピュータ上のデータへのアクセス」『SQL Anywhere サーバ SQL の使用法』を参照してください。

  - **変数へのデータのアンロード**　UNLOAD 文が強化されて INTO VARIABLE 句が追加され、変数にデータをアンロードできるようになりました。「UNLOAD 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **別のテーブルのカラムからのデータのロード**　LOAD TABLE 文が強化されて USING COLUMN 句が追加され、別のテーブルのカラムからデータをロードできるようになりま

した。「LOAD TABLE 文」『SQL Anywhere サーバ SQL リファレンス』と「LOAD TABLE 文を使用したデータのインポート」『SQL Anywhere サーバ SQL の使用法』を参照してください。

この新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。「SQL Anywhere サーバのアップグレード」289 ページを参照してください。

○ **値 (BLOB) からのデータのロード**　LOAD TABLE 文が強化されて USING VALUE 句が追加され、関数やシステムプロシージャの結果などの値式からデータをロードできるようになりました。「LOAD TABLE 文」『SQL Anywhere サーバ SQL リファレンス』と「LOAD TABLE 文を使用したデータのインポート」『SQL Anywhere サーバ SQL の使用法』を参照してください。

この新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。「SQL Anywhere サーバのアップグレード」289 ページを参照してください。

○ **LOAD TABLE 文のリカバリとミラーリングの強化**　これまでは、ミラーリングされたデータベース構成で LOAD TABLE 文を使用してファイルからデータをロードできませんでした。これは、LOAD TABLE 文だけがトランザクションログに記録され、ロード対象のデータは記録されなかったからです。また、データベースをリカバリするときに、LOAD TABLE 文を使用してロードされたデータは、リカバリ時に元のロードファイルがないかぎり回復不能でした。

LOAD TABLE 文が強化され、WITH CONTENT LOGGING、WITH ROW LOGGING、WITH FILE NAME LOGGING の 3 つの新しいログオプション句が追加されました。これらの句を使用すると、ロードするデータをトランザクションログに記録するかどうかを制御できます。データベースのミラーリングシステムでは、このデータを使用してミラーデータベースをロードできます。また、リカバリ時にロードファイルがなくてもリカバリが可能になります。「LOAD TABLE 文」『SQL Anywhere サーバ SQL リファレンス』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。「SQL Anywhere サーバのアップグレード」289 ページを参照してください。

● **マテリアライズドビューの強化**　マテリアライズドビューのサポートが次のように強化されました。

○ **即時マテリアライズドビューのサポート**　基本となるテーブル内のデータの変更がマテリアライズドビュー内のデータに影響する場合に、マテリアライズドビューをすぐにリフレッシュするように設定できます。このようにすぐにリフレッシュするビューを「即時ビュー」と呼び、すぐにリフレッシュされないビューは今後「手動ビュー」と呼びます。このリリースの前に作成されたマテリアライズドビューは手動ビューと見なされます。また、新規に作成するマテリアライズドビューはデフォルトで手動ビューになります。

手動ビューと即時ビューの詳細については、「マテリアライズドビュー」『SQL Anywhere サーバ SQL の使用法』と「再表示タイプの手動または即時への設定」『SQL Anywhere サーバ SQL の使用法』を参照してください。

この機能をサポートするシステムプロシージャを使用するには、データベースをアップグレードする必要があります。「SQL Anywhere サーバのアップグレード」289 ページを参照してください。

○ **複数のマテリアライズドビューの同時リフレッシュ**　これまでは、マテリアライズドビューは一度に 1 つずつリフレッシュする必要がありました。各リフレッシュ操作の間で基本となるデータが変更された場合、マテリアライズドビューの間で矛盾が生じる可能性がありました。このリリースでは、同じデータを使用するマテリアライズドビューをリフレッシュするときに、REFRESH MATERIALIZED VIEW 文にマテリアライズドビューのリストを指定できるようになりました。「REFRESH MATERIALIZED VIEW 文」『SQL Anywhere サーバ SQL リファレンス』と「マテリアライズドビューの手動再表示」『SQL Anywhere サーバ SQL の使用法』を参照してください。

○ **REFRESH MATERIALIZED VIEW 文の新しい WITH SHARE MODE 句**　REFRESH MATERIALIZED VIEW 文に、新しい句 WITH SHARE MODE が追加されました。このモードでは、リフレッシュ操作中に、基本となるテーブルを他のトランザクションで読み込むことができます。この句を指定すると、基本となるすべてのテーブルの共有テーブルロックが取得されてから、リフレッシュ操作が実行されます。マテリアライズドビューが IMMEDIATE REFRESH と定義されているか、データベースのスナップショットアイソレーションが有効になっていないかぎり、デフォルトのモードは WITH SHARE MODE になります。リフレッシュのデフォルト動作の詳細については、「REFRESH MATERIALIZED VIEW 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

● **ファイルまたは BLOB 文字列の内容の問い合わせのサポート**　FROM 句の新しい OPENSTRING 句を使用して、ファイルや BLOB 文字列のデータを問い合わせることができるようになりました。OPENSTRING 句では、問い合わせるオブジェクト、またデータのスキーマやその他の解析情報を指定できます。「FROM 句」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

OPENSTRING 操作が実行されると、新しいプラン項目 OpenString が実行プランに表示されます。

● **圧縮インデックスのサポートの向上**　圧縮インデックスのサポートが向上された結果、データベースをアンロードしてから再ロードして再構築すると、再構築されたデータベースが元のデータベースよりも小さくなる場合があります。このデータベースのサイズの縮小は、問題や、データが失われたことを示すものでは**ありません**。

● **ミラーサーバで実行されているデータベースへの読み込み専用アクセス**　データベースミラーリングを使用している場合に、ミラーサーバで実行されているデータベースに接続できるようになりました。したがって、プライマリサーバの可用性を維持しながら、リソースを大量に使用する可能性があるレポート操作の負荷をミラーサーバに割り当てることができます。ミラーデータベースに接続するには、読み込み専用のミラーデータベースへのアクセスに使用できる -sm サーバオプションでデータベースサーバ名を指定します。「-sm データベースオプション (廃止予定)」『SQL Anywhere サーバ データベース管理』を参照してください。

データベース接続

次に、SQL Anywhere バージョン 11.0.0 でのデータベース接続の強化を示します。

- **AppInfo 接続パラメータの強化**　AppInfo 接続パラメータで OSUSER キーがサポートされるようになりました。このキーは、クライアントプロセスに関連付けられたオペレーティングシステムユーザ名を返します。Linux と Solaris で EXE キーがサポートされるようになりました。「AppInfo (APP) 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Elevate 接続パラメータ**　Elevate 接続パラメータは、Windows Vista で自動的に起動した SQL Anywhere データベースサーバを昇格します。「Elevate 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **NewPassword 接続パラメータ [NEWPWD]**　NewPassword 接続パラメータを使用すると、DBA に依頼しなくてもユーザが各自でパスワードを変更できます。パスワードの有効期限が切れていてもかまいません。「NewPassword (NEWPWD) 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **プリフェッチの強化**　PrefetchBuffer (PBUF) 接続パラメータのデフォルト値が変更されました。デフォルト値は、Windows Mobile では 64 KB、その他のプラットフォームでは 512 KB になりました。この接続パラメータには、64 KB ～ 8 MB の値を指定できます。「PrefetchBuffer (PBUF) 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

  これまでのバージョンでは、プリフェッチされる最大ロー数は、プリフェッチできる最大データ量に基づいていました。このリリースでは、プリフェッチされる最大ロー数には、プリフェッチされる実際のデータ量に加えて、PrefetchBuffer 接続パラメータで指定されたデータ制限が考慮されるようになりました。その結果、カラム内のデータ量が、ホスト変数長と記述長の両方よりも大幅に少ない場合に、パフォーマンスが大きく向上する可能性があります。

  また、パフォーマンスが向上する可能性が高い場合に、プリフェッチするロー数が動的に増えます。「プリフェッチ」『SQL Anywhere サーバ プログラミング』を参照してください。

バックアップとリカバリ

次に、SQL Anywhere バージョン 11.0.0 でのバックアップとリカバリの強化を示します。

- **Microsoft ボリュームシャドウコピーサービス (VSS) のサポート**　SQL Anywhere は、Microsoft ボリュームシャドウコピーサービス (VSS) との互換性があります。VSS を使用するには、既存のデータベースをすべて再構築する必要があります。「SQL Anywhere ボリュームシャドウコピーサービス (VSS)」『SQL Anywhere サーバ データベース管理』を参照してください。

セキュリティ

次に、SQL Anywhere バージョン 11.0.0 でのセキュリティの強化を示します。

- **テーブルの暗号化が有効になっているときの ISYSUSER と ISYSEXTERNLOGIN の各システムテーブルの暗号化**　これまでは、データベースを暗号化するときや、テーブルの暗号化を有効にしてデータベースを作成するときに、ISYSCOLSTAT システムテーブルが自動的に暗号化されました。このリリースでは、さらにセキュリティを強化するため、ISYSUSER と ISYSEXTERNLOGIN の各システムテーブルも暗号化されるようになりました。

- **監査の強化**　このリリースでは、Sybase Central を使用して監査を制御できるようになりました。**[データベースのプロパティ]** ウィンドウから、監査を有効または無効にしたり、監査する情報を指定したりできます。監査情報は、Sybase Central の右ウィンドウ枠の **[監査]** タブで確認できます。「監査の設定 (Sybase Central の場合)」『SQL Anywhere サーバ データベース管理』と「監査情報の取得 (Sybase Central の場合)」『SQL Anywhere サーバ データベース管理』を参照してください。

  監査が有効になっているときは、接続失敗のエラーが、失敗の理由とともにログに記録されます。

- **256 ビットの AES 暗号化のサポート**　SQL Anywhere で、データベース、テーブル、ファイル、データに対して 256 ビットの AES 暗号化がサポートされるようになりました。この強化は、次に示す複数の領域に影響します。

  - **データベースとテーブルの暗号化**　CREATE DATABASE 文の ENCRYPTION 句に、AES256 と AES256_FIPS を指定できるようになりました。「CREATE DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

    初期化ユーティリティ (dbinit) とアンロードユーティリティ (dbunload) の -ea オプションに、AES256 と AES256_FIPS を指定することもできます。「初期化ユーティリティ (dbinit)」『SQL Anywhere サーバ データベース管理』と「アンロードユーティリティ (dbunload)」『SQL Anywhere サーバ データベース管理』を参照してください。

  - **FIPS 認定のアルゴリズム**　256 ビットの FIPS 認定の AES アルゴリズムを使用できるようになりました。「-fips データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

  - **データの暗号化と復号化**　ENCRYPT 関数と DECRYPT 関数を使用してデータを暗号化するときに、AES256 と AES256_FIPS を指定できるようになりました。「ENCRYPT 関数 [文字列]」『SQL Anywhere サーバ SQL リファレンス』と「DECRYPT 関数 [文字列]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **データベース、トランザクションログ、DB 領域の暗号化されたコピーの作成**　CREATE ENCRYPTED FILE 文を使用して、暗号化されているか、暗号化されてないデータベース、トランザクションログ、または DB 領域の暗号化されたコピーを作成するとき、256 ビットの AES アルゴリズム (AES256 または AES256_FIPS) を指定できるようになりました。「CREATE ENCRYPTED FILE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **DBTools での 256 ビットの AES 暗号化のサポート**　a_create_db と an_unload_db の各構造体が拡張され、encryption_algorithm メンバーの値として AES256 と AES256_FIPS がサポートされるようになりました。a_create_db 構造体 [データベースツール]『SQL Anywhere サーバ プログラミング』と an_unload_db 構造体 [データベースツール]『SQL Anywhere サーバ プログラミング』を参照してください。

  参照：

  - 「データベースの暗号化と復号化」『SQL Anywhere サーバ データベース管理』
  - 「データベースプロパティの一覧」『SQL Anywhere サーバ データベース管理』

- **jConnect と Open Client のパスワード暗号化のサポート**　jConnect 接続と Open Client 接続でパスワードの暗号化がサポートされるようになりました。次の項を参照してください。

  - 「jConnect JDBC ドライバ」『SQL Anywhere サーバ プログラミング』
  - 「JDBC クライアントの配備」『SQL Anywhere サーバ プログラミング』
  - 「SQL Anywhere における Open Client の既知の制限」『SQL Anywhere サーバ プログラミング』

セキュリティの変更

これらの変更点を使用するには、データベースをアップグレードする必要があります。「SQL Anywhere サーバのアップグレード」289 ページを参照してください。

- **継承のサポート**　SQL Anywhere により、PROFILE、READCLIENTFILE、READFILE、WRITECLIENTFILE 権限の継承がサポートされるようになりました。

- **新しい権限：**

  - PROFILE
  - READFILE
  - READCLIENTFILE
  - WRITECLIENTFILE

- **DB 領域の CREATE 権限のサポート**　SQL Anywhere で、指定の DB 領域にデータベースオブジェクトを作成できる CREATE ON パーミッションがサポートされるようになりました。CREATE 権限は、ユーザに割り当てるか、グループから継承できます。

データベースユーティリティ

次に、SQL Anywhere バージョン 11.0.0 でのデータベースユーティリティの強化を示します。

- **設定ファイルの強化**　設定ファイル内でアンパサンド (&) を使用して、前のトークンが次の行に続いていることを指定できるようになりました。「設定ファイル」『SQL Anywhere サーバ データベース管理』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**　dbunload が次のように強化されています。

  - 新しいオプション -cp が追加され、dbunload でデータ出力ファイルを圧縮できるようになりました。

  - これまでは、再ロードオプション -an または -ar を指定しないで暗号化オプション -ek、-ep、または -ea を指定すると、エラーが返されました。このリリースでは、dbunload で暗号化オプションが受け入れられ、作成される出力ファイルに適用されるようになりました。

  - -g オプションによって、MANUAL REFRESH と定義されているテキストインデックスが再表示されるようになりました。「テキストインデックスの概念と参照」『SQL Anywhere サーバ SQL の使用法』を参照してください。

○MANUAL REFRESH と定義されているテキストインデックスは、デフォルトで再ロード時に初期化されません。これらのテキストインデックスを初期化したい場合は、dbunload -g オプションを指定できます。

○-no オプションを使用して、オブジェクト定義をオブジェクトタイプ別にアルファベット順でアンロードできます。これはデータベースの *reload.sql* ファイルを比較する場合に便利です。

「アンロードユーティリティ (dbunload)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **検証ユーティリティ (dbvalid) の強化**　これまで、dbvalid ユーティリティでは、デフォルトでテーブルとマテリアライズドビューがすべて検証されていました。このリリースでは、dbvalid によって VALIDATE DATABASE 文も実行されます。

  検証ユーティリティを実行してデータベースを自動的に起動すると、データベースは読み込み専用モードで起動します。このようにすることで、バックアップとリカバリのプランの一環としてデータベースが検証されている場合にデータベースを変更できないようになっています。

  「検証ユーティリティ (dbvalid)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **ログ変換ユーティリティ (dbtran) で破損が検出された場合の生成された *.sql* ファイルの保守**　ログ変換ユーティリティの -k オプションを使用すると、トランザクションログファイルが破損しているためログファイルの変換に失敗した場合に不完全な *.sql* ファイルを削除しないことを指定できます。「ログ変換ユーティリティ (dbtran)」『SQL Anywhere サーバ データベース管理』を参照してください。

データベースオプション

次に、SQL Anywhere バージョン 11.0.0 でのデータベースオプションの強化を示します。

- **allow_read_client_file オプション**　このオプションは、クライアントコンピュータ上のファイルの読み込みを許可するかどうかを制御します。「allow_read_client_file オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **allow_write_client_file オプション**　このオプションは、クライアントコンピュータへのファイルの書き込みを許可するかどうかを制御します。「allow_write_client_file オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **login_procedure オプション**　パスワードの有効期限が切れていることを示すエラーメッセージを通知できるようになりました。「login_procedure オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **max_priority オプション**　このオプションは、データベース接続の最高優先度レベルを制御します。「max_priority オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **priority オプション**　このオプションは、接続からの要求を実行する優先度レベルを制御します。「priority オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **query_mem_timeout オプション**　このオプションは、要求にメモリが付与されるまで待つ時間を制御します。「query_mem_timeout オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

データベースサーバオプション

次に、SQL Anywhere バージョン 11.0.0 でのデータベースサーバオプションの強化を示します。

- **-es サーバオプション**　SQL Anywhere のこれまでのバージョンでは、-ec オプションを指定してデータベースサーバが起動され (トランスポートレイヤセキュリティのサポートのため)、許可する暗号化プロトコルのリストに NONE または SIMPLE が含まれなかった場合、共有メモリポートが起動しませんでした。これは、このポートでトランスポートレイヤセキュリティがサポートされていないからです。このため、データベースサーバへの接続は、強力な暗号化を使用して TCP/IP で行う必要がありました。

  -es サーバオプションは、共有メモリを経由した、暗号化されていない接続を許可するようにデータベースサーバに指定します。「-es データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **-gb サーバオプション**　サーバプロセスの優先度クラスを制御する -gb サーバオプションが、Windows に加えて UNIX でサポートされるようになりました。「-gb データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **-im サーバオプション**　アプリケーションで、データベースの操作内容がすべて失われてもかまわない場合は、データベースを完全にメモリ内で実行できます。この機能は、データが頻繁に挿入される、高速のテンポラリデータストアとして SQL Anywhere を使用する場合を対象としています。「-im データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **クライアントコンピュータにあるファイルの読み込みと書き込み**　-sf サーバオプションを使用して、クライアントコンピュータにあるファイルの読み込みと書き込みの機能を制御できるようになりました。「-sf データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **-um サーバオプション**　-um オプションを使用して、*DBLauncher.app* インスタンスが実行中の場合にこのインスタンスに接続し、*DBLauncher.app* 内の新しいウィンドウにデータベースサーバメッセージを表示することができます。このオプションは Mac OS X だけに適用されます。「-um データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Windows パフォーマンスモニタのオプション**　Windows パフォーマンスモニタをさらに詳細に設定するため、次のサーバオプションが追加されました。

  - **-ks オプション**　パフォーマンスモニタで、データベースサーバからカウンタ値を収集するために使用される共有メモリの作成を無効にします。「-ks データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **-ksc オプション**　パフォーマンスモニタでモニタできる接続の最大数を指定します。「-ksc データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **-ksd オプション**　パフォーマンスモニタでモニタできるデータベースの最大数を指定します。「-ksd データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

プロパティとパフォーマンスモニタの統計値

次に、SQL Anywhere バージョン 11.0.0 でのプロパティとパフォーマンスモニタの統計の強化を示します。

- **新しい接続プロパティ**　このリリースには、次の接続プロパティが追加されています。

  - allow_read_client_file
  - allow_write_client_file
  - AuthType
  - CacheReadWorkTable
  - ClientNodeAddress
  - DiskReadWorkTable
  - DiskSyncRead
  - DiskSyncWrite
  - DiskWaitRead
  - DiskWaitWrite
  - DiskWriteHint
  - DiskWriteHintPages
  - LockIndexID
  - LockRowID
  - max_priority
  - OSUser
  - priority
  - query_mem_timeout
  - QueryMemActiveCurr
  - QueryMemExtraAvail
  - QueryMemGrantFailed
  - QueryMemGrantGranted
  - QueryMemGrantWaiting
  - QueryMemGrantRequested
  - QueryMemWaited
  - ServerNodeAddress
  - ReadHint
  - ReadHintScatter

  これらのプロパティの説明については、「接続プロパティの一覧」『SQL Anywhere サーバ データベース管理』を参照してください。

- **新しいデータベースサーバプロパティ**　このリリースには、次のデータベースサーバプロパティが追加されています。

  - DiskRetryRead
  - DiskRetryReadScatter
  - DiskRetryWrite
  - EventTypeDesc
  - EventTypeName
  - HttpAddresses
  - HttpsAddresses
  - HttpNumActiveReq
  - HttpNumConnections
  - HttpNumSessions
  - HttpsNumActiveReq
  - HttpsNumConnections
  - MaxEventType
  - MaxRemoteCapability
  - MessageCategoryLimit
  - OptionWatchAction
  - OptionWatchList
  - QueryMemActiveCurr
  - QueryMemActiveEst
  - QueryMemActiveMax
  - QueryMemExtraAvail
  - QueryMemGrantBase
  - QueryMemGrantBaseMI
  - QueryMemGrantExtra
  - QueryMemGrantFailed
  - QueryMemGrantGranted
  - QueryMemGrantWaiting
  - QueryMemGrantRequested
  - QueryMemPages
  - QueryMemPercentOfCache
  - QueryMemWaited
  - ReadHintScatterLimit
  - RemoteCapability
  - StreamsUsed
  - TcpIpAddresses
  - WebClientLogFile
  - WebClientLogging

  これらのプロパティの説明については、「データベースサーバプロパティの一覧」『SQL Anywhere サーバ データベース管理』を参照してください。

- **新しいデータベースプロパティ**　このリリースには、次のデータベースプロパティが追加されています。

  - AlternateMirrorServerName
  - CacheReadWorkTable
  - DiskReadWorkTable
  - DiskRetryReadScatter
  - DiskSyncRead
  - DiskSyncWrite
  - DiskWaitRead
  - DiskWaitWrite
  - DiskWriteHint
  - DiskWriteHintPages
  - HasEndianSwapFix
  - MirrorMode
  - ReadHint
  - ReadHintScatter

  これらのプロパティの説明については、「データベースプロパティの一覧」『SQL Anywhere サーバ データベース管理』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**　このリリースには、次のパフォーマンスモニタの統計値が追加されています。

  - キャッシュ読み込み：ワークテーブル
  - ディスク読み込み：ワークテーブル

システムプロシージャとファンクション

次に、SQL Anywhere バージョン 11.0.0 で追加されたシステムプロシージャとファンクションの強化を示します。

- **sa_get_dtt_groupreads システムプロシージャ**　新しい sa_get_dtt_groupreads システムプロシージャを使用して、データベースサーバに対してグループ読み込みを発行するコストを推定できます。「sa_get_dtt_groupreads システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **PROPERTY_NAME ファンクションの強化**　指定した接続レベルで、指定したプロパティ ID を持つプロパティの名前を返すようになりました。「PROPERTY_NAME 関数 [システム]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **READ_CLIENT_FILE ファンクション**　新しい READ_CLIENT_FILE ファンクションは、クライアントコンピュータにある、指定したファイルからデータを読み込みます。「READ_CLIENT_FILE 関数 [文字列]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **WRITE_CLIENT_FILE ファンクション**　新しい WRITE_CLIENT_FILE ファンクションは、クライアントコンピュータにある、指定したファイルにデータを書き込みます。

「WRITE_CLIENT_FILE 関数 [文字列]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **REGEXP_SUBSTR ファンクション**　新しい REGEXP_SUBSTR ファンクションを使用すると、文字列内で部分文字列を検索できます。この新しいファンクションでは、引数として正規表現を指定できます。「REGEXP_SUBSTR 関数 [文字列]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_char_terms システムプロシージャ**　新しい sa_char_terms システムプロシージャは、CHAR 文字列を単語に分解し、各単語とその位置を返します。「sa_char_terms システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_nchar_terms システムプロシージャ**　新しい sa_nchar_terms システムプロシージャは、NCHAR 文字列を単語に分解し、各単語とその位置を返します。「sa_nchar_terms システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_refresh_text_indexes システムプロシージャ**　新しい sa_refresh_text_indexes システムプロシージャは、MANUAL REFRESH または AUTO REFRESH として定義されているすべてのテキストインデックスを再表示します。「sa_refresh_text_indexes システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_text_index_stats システムプロシージャ**　新しい sa_text_index_stats システムプロシージャは、データベース内のすべてのテキストインデックスに関する統計情報を返します。これには、最終再表示時刻と保留中の変更のサイズが含まれます。「sa_text_index_stats システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_text_index_vocab システムプロシージャ**　新しい sa_text_index_vocab システムプロシージャは、テキストインデックスに含まれるすべての単語と、各単語が含まれるインデックス化された値の合計数のリストを返します。「sa_text_index_vocab システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  2 つの新しいシステムプロシージャ sa_internal_text_index_vocab と sa_internal_text_index_postings も追加されていますが、これらは sa_text_index_vocab システムプロシージャでのみ使用されます。

- **sa_text_index_postings システムプロシージャ**　この新しいシステムプロシージャは内部でのみ使用されます。

- **sa_text_index_handles システムプロシージャ**　この新しいシステムプロシージャは内部でのみ使用されます。

- **sa_get_user_status システムプロシージャ**　新しい sa_get_user_status システムプロシージャを使用すると、ユーザの現在のログインステータスを確認できます。「sa_get_user_status システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **インボーカーとしてのプロシージャやファンクションの実行**　プロシージャまたはファンクションを作成するときに、そのプロシージャまたはファンクションを、呼び出したユーザ (インボーカ) のパーミッションによって実行するか、所有者 (デファイナ) のパーミッションによって実行するかを指定できます。これを指定するには、CREATE PROCEDURE 文または CREATE FUNCTION 文の SQL SECURITY 句を使用します。「CREATE FUNCTION 文」『SQL

Anywhere サーバ SQL リファレンス』と「CREATE PROCEDURE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

この変更は、外部プロシージャと関数にも適用されます。

- **sa_disk_free_space システムプロシージャ**　sa_disk_free_space システムプロシージャで、total_space という新しいカラムが返されるようになりました。このカラムは、DB 領域がある ドライブで使用可能な合計ディスク領域を示します。バージョン 11.0.0 より以前の SQL Anywhere で作成されたデータベースでは、データベースをアップグレードするまで total_space カラムは返されません。「sa_disk_free_space システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_external_library_unload システムプロシージャ**　新しいシステムプロシージャ sa_external_library_unload が追加され、使用していない外部ライブラリをアンロードできるようになりました。「sa_external_library_unload システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_index_density システムプロシージャがスキュー量を返す**　sa_index_density システムプロシージャが強化され、インデックス内にあるスキュー量を返すようになりました。スキューが多いと、バランスがとれているインデックスと比べてパフォーマンスが低くなる可能性があります。「インデックスの断片化とスキューの削減」『SQL Anywhere サーバ SQL の使用法』と「sa_index_density システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_materialized_view_info システムプロシージャの強化**　sa_materialized_view_info で返される Status カラムの情報が、Status と DataStatus の 2 つのカラムに分割されました。Status には、ビューが有効であるか、無効であるかに関する情報が返されます。新しい DataStatus カラムには、ビュー内にデータがあるかどうか、またデータが最新かどうかに関する情報が返されます。また、ビューが手動ビューであるか、即時ビューであるかを示す新しいカラム RefreshType が追加されています。「sa_materialized_view_info システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_materialized_view_can_be_immediate システムプロシージャ**　新規に作成するマテリアライズドビューはデフォルトで手動ビューになりますが、即時ビューの制約に違反していなければ、即時ビューに変更できます。新しい sa_materialized_view_can_be_immediate システムプロシージャを使用すると、手動ビューを即時ビューに変更できるかどうかをテストできます。「sa_materialized_view_can_be_immediate システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』と「マテリアライズドビューを手動から即時に変更する際の制限」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **sa_post_login_procedure システムプロシージャ**　ユーザのパスワードの有効期限が近いときに警告するかどうかを指定できる新しいシステムプロシージャが追加されました。「sa_post_login_procedure システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **EVENT_PARAMETER ファンクションの強化**　EVENT_PARAMETER ファンクションで、DisconnectReason として abnormal がサポートされるようになりました。この新しい理由は、データベースへの接続を切断する前にクライアントアプリケーションが異常終了したか、クライアントコンピュータとサーバコンピュータの間で通信エラーが発生した結果、切断が発

生したことを示します。「EVENT_PARAMETER 関数 [システム]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_server_option システムプロシージャの強化**　sa_server_option システムプロシージャに OptionWatchList と OptionWatchAction の 2 つのプロパティが追加されました。これらのプロパティを使用して、データベースオプションの設定が変更されようとしていないかどうかをモニタし、そのときの処置を指定できます。「オプション設定のモニタリング」『SQL Anywhere サーバ データベース管理』と「sa_server_option システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_db_properties システムプロシージャの強化**　sa_db_properties システムプロシージャが、NULL 値を持つ有効なプロパティを返すようになりました。「sa_db_properties システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **sa_conn_properties システムプロシージャの強化**　sa_conn_properties システムプロシージャが、NULL 値を持つ有効なプロパティを返すようになりました。「sa_conn_properties システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

SQL 文

次に、SQL Anywhere バージョン 11.0.0 での SQL の強化を示します。

- **新しい CALIBRATE GROUP READ 句、ALTER DATABASE 文**　ALTER DATABASE 文の新しい CALIBRATE GROUP READ 句を使用して、テンポラリ DB 領域に対してグループ読み込みの調整を実行できます。「ALTER DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい CHECK 句、CREATE MATERIALIZED VIEW 文**　CREATE MATERIALIZED VIEW 文の新しい CHECK 句を使用して、ビューの作成前に文を検証できます。「CREATE MATERIALIZED VIEW 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい RECOMPILE 句、ALTER FUNCTION 文**　ALTER FUNCTION 文に新しい句 RECOMPILE が追加され、ユーザ定義関数を再コンパイルできるようになりました。「ALTER FUNCTION 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい RECOMPILE 句、ALTER PROCEDURE 文**　ALTER PROCEDURE 文に新しい句 RECOMPILE が追加され、ストアドプロシージャを再コンパイルできるようになりました。「ALTER PROCEDURE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **新しい REFRESH 句、ALTER MATERIALIZED VIEW 文**　ALTER MATERIALIZED VIEW 文に新しい句 REFRESH が追加され、マテリアライズドビューのリフレッシュ方法を指定できるようになりました。「ALTER MATERIALIZED VIEW 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **リカバリとミラーリングをサポートするための LOAD TABLE 文の強化**　リカバリとミラーリングをサポートするため、次の句が LOAD TABLE 文に追加されました。

  - **WITH CONTENT LOGGING 句**　WITH CONTENT LOGGING 句は、データソースの内容をトランザクションログに記録することをデータベースサーバに指定します。データは、入

力が LOAD TABLE によって処理されるときに小さいチャンクで記録されます。これらのチャンクは、ミラーリングデータベースで、またトランザクションログを使用したリカバリ時に、ローに再構成できます。WITH CONTENT LOGGING 句は、後でリカバリするために元のデータファイルを維持しないほうがいい場合に便利です。「LOAD TABLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

○ **WITH ROW LOGGING 句**　WITH ROW LOGGING 句は、ロードされているすべてのローを、一連の INSERT 文として記録することをデータベースサーバに指定します。このレベルは、同期するデータベースや、ロード先のテーブルに、計算カラムや CURRENT TIMESTAMP のデフォルトなど、非決定的な値が含まれる場合に最適です。「LOAD TABLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

○ **WITH FILE NAME LOGGING 句**　WITH FILE NAME LOGGING 句は、LOAD TABLE 文だけを記録するようにデータベースサーバに指定します。これがデフォルトの動作であり、SQL Anywhere のこれまでのバージョンでのロギングの動作と同じです。「LOAD TABLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

● **クライアントファイルのロードとアンロードのための新しい句**　クライアントファイルの新しいロード／アンロード機能をサポートするため、LOAD TABLE 文と UNLOAD TABLE 文が強化されました。

○ **LOAD TABLE 文の新しい USING CLIENT FILE 句**　クライアントコンピュータにあるファイル内のデータを使用してテーブルをロードできます。「LOAD TABLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

○ **UNLOAD TABLE 文の新しい INTO CLIENT FILE 句**　データのアンロード先としてクライアントコンピュータにあるファイルを指定できます。「UNLOAD 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

● **新しいログインポリシー文**　新しいログインポリシー機能をサポートするために、次の文が追加されました。

○ CREATE LOGIN POLICY 文：「CREATE LOGIN POLICY 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
○ ALTER LOGIN POLICY 文：「ALTER LOGIN POLICY 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
○ DROP LOGIN POLICY 文：「DROP LOGIN POLICY 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
○ CREATE USER 文：「CREATE USER 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
○ ALTER USER 文：「ALTER USER 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
○ DROP USER 文：「DROP USER 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

● **全文検索用の新しい文と句**　新しい全文検索機能をサポートするために、次の文が追加されました。

○ **新しい CONTAINS 探索条件**　CONTAINS 探索条件を使用して、指定するカラムのリスト内に、指定する単語やフレーズがあるかどうかを確認できます。CONTAINS 探索条件は、

TRUE または FALSE を返します。複数の単語やフレーズを検索する場合は、さまざまなブール演算子で結合できます。「CONTAINS 探索条件」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **SELECT 文の FROM 句内の新しい CONTAINS 句**　CONTAINS 句は、SELECT 文の FROM 句内で指定します。この句は CONTAINS 探索条件に似ていますが、一致するカラムごとのスコアと、一致する各ローの合計スコアも返します。「FROM 句」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **CREATE TEXT CONFIGURATION 文**　この文は、テキスト設定オブジェクトを作成します。テキスト設定オブジェクトは、テキストインデックスの特性を制御する一連の設定です。「CREATE TEXT CONFIGURATION 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ALTER TEXT CONFIGURATION 文**　この文は、テキスト設定オブジェクトを変更します。「ALTER TEXT CONFIGURATION 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **DROP TEXT CONFIGURATION 文**　この文は、テキスト設定オブジェクトを削除します。「DROP TEXT CONFIGURATION 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **CREATE TEXT INDEX 文**　この文は、テキストインデックスを作成します。テキストインデックスには、インデックス化されたすべてのカラム内の全単語の出現箇所の詳細な位置情報が含まれます。「CREATE TEXT INDEX 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ALTER TEXT INDEX 文**　この文は、テキストインデックスを変更します。「ALTER TEXT INDEX 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **DROP TEXT INDEX 文**　この文は、データベースからテキストインデックスを削除します。「DROP TEXT INDEX 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **REFRESH TEXT INDEX 文**　この文は、テキストインデックスを再表示します。「REFRESH TEXT INDEX 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **TRUNCATE TEXT INDEX 文**　この文は、テキストインデックスからデータをトランケートします。「TRUNCATE TEXT INDEX 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

● **ALTER EVENT 文の強化**　ALTER EVENT...SET HIDDEN 文を使用して、イベントハンドラの定義を隠すことができるようになりました。この文を使用すると、ISYSEVENT システムテーブルの action カラムに格納されているイベントハンドラの定義が難読化されます。「ALTER EVENT 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

● **BEGIN SNAPSHOT 文**　BEGIN SNAPSHOT 文を使用すると、スナップショットアイソレーションのためスナップショットをいつ開始するかを制御できます。「BEGIN SNAPSHOT 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **CASE 文と CASE 式の強化**　互換性の向上のため、CASE 文と CASE 式を END または END CASE で終了できるようになりました。「CASE 文」『SQL Anywhere サーバ SQL リファレンス』と「CASE 式」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **COMMENT 文の強化**　ログインポリシーテーブルと DB 領域にコメントを追加できるようになりました。次の項を参照してください。

  - 「COMMENT 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「ログインポリシー」『SQL Anywhere サーバ データベース管理』
  - 「追加の DB 領域の考慮事項」『SQL Anywhere サーバ データベース管理』

- **CREATE MATERIALIZED VIEW 文の強化**　CREATE MATERIALIZED VIEW 文の新しい IMMEDIATE REFRESH 句を使用して、基本となるデータが変更されたときにリフレッシュされるマテリアライズドビューを作成できるようになりました。「CREATE MATERIALIZED VIEW 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **DESCRIBE 文の強化**　Interactive SQL の DESCRIBE 文を使用して、Interactive SQL が接続しているデータベースまたはデータベースサーバに関する情報を取得できるようになりました。「DESCRIBE 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **IF 文と IF 式の強化**　互換性の向上のため、IF 文と IF 式を ENDIF または END IF で終了できるようになりました。「IF 文」『SQL Anywhere サーバ SQL リファレンス』と「IF 式」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **LOAD TABLE 文の強化**　LOAD TABLE 文の新しい COMPRESSED 句または ENCRYPTED 句を使用して、入力ファイルのデータが圧縮されていることや、暗号化されていることを指定できるようになりました。「LOAD TABLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **SELECT 文の強化**

  - **INDEX 句の強化**　INDEX 句を使用してインデックスヒントを指定するときに、データベースサーバで使用する必要があるインデックスを 4 つまで指定できるようになりました。「FROM 句」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **新しい INDEX ONLY 句**　INDEX 句を使用してインデックスヒントを指定するときに、必要に応じて INDEX ONLY 句を指定して、データベースサーバでインデックス専用取得 (インデックスのデータのみを使用してクエリを処理する) を行うかどうかを制御できます。「FROM 句」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **新しい CROSS APPLY 句と OUTER APPLY 句**　SELECT 文が拡張され、FROM 句で適用式 (CROSS APPLY 句と OUTER APPLY 句) がサポートされるようになりました。適用式は、右側が左側に依存するジョインを簡単に指定できる方法です。たとえば、適用式を使用して、テーブル式内のローごとに 1 回ずつプロシージャまたは派生テーブルを評価できます。「適用式から生成されるジョイン」『SQL Anywhere サーバ SQL の使用法』と「FROM 句」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

○ **新しい OPENSTRING 句**　新しい OPENSTRING 句を使用すると、SELECT 文を使用してファイル内のデータを問い合わせることができます。「FROM 句」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

● **イベントの作成時、変更時、削除時、コメント時の所有者の指定**　CREATE EVENT、ALTER EVENT、DROP EVENT、COMMENT ON EVENT の各文で、必要に応じて所有者を指定できるようになりました。次の項を参照してください。

○「CREATE EVENT 文」『SQL Anywhere サーバ SQL リファレンス』
○「ALTER EVENT 文」『SQL Anywhere サーバ SQL リファレンス』
○「DROP EVENT 文」『SQL Anywhere サーバ SQL リファレンス』
○「COMMENT 文」『SQL Anywhere サーバ SQL リファレンス』

● **UNLOAD 文の強化**　UNLOAD 文の COMPRESSED 句または ENCRYPTED 句を使用して、アンロード対象のデータの圧縮や暗号化を行うかどうかを指定できるようになりました。「UNLOAD 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

これらの句を使用して圧縮または暗号化したファイルは、LOAD TABLE などを使用して、SQL Anywhere 11.0.0 のデータベースサーバでのみロードできます。その他のツールを使用して圧縮または暗号化したファイルは SQL Anywhere で使用できません。

● **UPDATE 文の強化**　検索と位置付け更新のときに、SET 句を使用してカラム値をデフォルト値に設定できるようになりました。「UPDATE 文」『SQL Anywhere サーバ SQL リファレンス』と「UPDATE (位置付け) 文 [ESQL] [SP]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

● **OPTION 句の拡張**　INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT、INTERSECT の各文の OPTION 句で、user_estimates データベースオプションの設定を上書きできるようになりました。次の項を参照してください。

○「INSERT 文」『SQL Anywhere サーバ SQL リファレンス』
○「UPDATE 文」『SQL Anywhere サーバ SQL リファレンス』
○「DELETE 文」『SQL Anywhere サーバ SQL リファレンス』
○「SELECT 文」『SQL Anywhere サーバ SQL リファレンス』
○「UNION 文」『SQL Anywhere サーバ SQL リファレンス』
○「EXCEPT 文」『SQL Anywhere サーバ SQL リファレンス』
○「INTERSECT 文」『SQL Anywhere サーバ SQL リファレンス』

プログラミングインタフェース

次に、SQL Anywhere バージョン 11.0.0 でのプログラミングインタフェースの強化を示します。

● **新しい SQL Anywhere C API**　SQL Anywhere C アプリケーションプログラミングインタフェース (API) により、PHP、Perl、Python、Ruby など、複数のインタプリター型プログラミング言語での C や C++ ラッパードライバの作成が簡単になります。SQL Anywhere C API は DBLIB ライブラリの上層に位置し、Embedded SQL で実装されています。

DBLIB に代わるものではありませんが、この API は、C や C++ によるアプリケーションの作成を簡単にします。SQL Anywhere C API を使用するのに、Embedded SQL に関する高度な知

識は必要ありません。「SQL Anywhere C API のサポート」『SQL Anywhere サーバ プログラミング』を参照してください。

- **新しい Python データベース API (sqlanydb)**　新しい Python データベース API (sqlanydb) を使用すると、Python で作成されたスクリプトから SQL Anywhere データベースにアクセスできるようになります。sqlanydb モジュールは、Python データベース API 仕様バージョン 2.0 を拡張して実装しています。「Python サポート」『SQL Anywhere サーバ プログラミング』を参照してください。

- **外部環境**　SQL Anywhere で、Java、Perl、PHP、CLR、Embedded SQL、ODBC の 6 つの外部ランタイム環境がサポートされるようになりました。これまで SQL Anywhere では、C または C++ で記述されたコンパイル済みネイティブ関数を呼び出すことができました。ただし、これらのプロシージャがサーバで実行されるとき、ダイナミックリンクライブラリまたは共有オブジェクトが常にデータベースサーバによってロードされ、ネイティブ関数への呼び出しがデータベースサーバによって行われていました。この方法には、ネイティブ関数が原因で障害が発生した場合、データベースサーバがクラッシュするというリスクがあります。データベースサーバの外部環境でコンパイル済みネイティブ関数を実行できると、サーバへのこれらのリスクをなくすことができます。「SQL Anywhere 外部環境のサポート」『SQL Anywhere サーバ プログラミング』を参照してください。

  この新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。「SQL Anywhere サーバのアップグレード」289 ページを参照してください。

- **PHP 外部環境のサポート**　SQL Anywhere 11.0.0 には、5.1.1 ～ 5.1.6 と 5.2.0 ～ 5.2.6 を含むさまざまな PHP バージョン用の構築済みバイナリが含まれます。このいずれかのバージョンをサーバコンピュータにインストールしてある場合は、PHP 外部環境を構築しないで SQL Anywhere の構築済みバイナリを使用してください。Linux と Solaris 用には、32 ビットと 64 ビットの両方のバージョンのバイナリが用意されています。Windows とその他のシステム用には、32 ビットのバージョンだけがあります。

  上記とは異なる PHP バージョンをインストールしてある場合は、ソフトウェアを構築するか、PHP バージョンを、SQL Anywhere の構築済みバージョンと同じバージョンに切り替える必要があります。SQL Anywhere PHP モジュールの構築方法については、「SQL Anywhere PHP 拡張」『SQL Anywhere サーバ プログラミング』を参照してください。

- **Perl 外部環境のサポート**　Perl 外部環境を使用する前に、SQL Anywhere Perl DBD ドライバのバージョンを更新することが非常に重要です。Perl DBD ドライバを更新しなかった場合、サーバ側 Perl は機能しません。

  また、PHP とは異なり、SQL Anywhere には Perl のさまざまなバージョン用の構築済みバイナリは含まれません。SQL Anywhere Perl DBD ドライバのソースコードは *%SQLANY11%¥SDK¥perl* にあります。SQL Anywhere Perl DBD ドライバの構築方法については、「Perl DBI サポート」『SQL Anywhere サーバ プログラミング』を参照してください。

- **Web サーバでの UTF-8 URL のサポート**　これまでは、要求 URL (または要求の本文内の application/x-www-form-urlencoded データ) 内のパーセント記号 (%) でコード化されたデータが、Web サーバによってデータベースの文字セットにデコードされていました。このリリースでは、パーセント記号 (%) でコード化されたデータの内容で、UTF-8 シーケンスの有無がテストされ、最大エクステントに基づいてデータベースの文字セットに変換されるようにな

りました。UTF-8 以外でコード化されたデータは、デコードされ、すでにデータベースの文字セットになっているかのように処理されます。

クライアント HTTP アプリケーションではパーセント記号 (%) でコード化された UTF-8 データを排他的に送信する必要があります。ASCII は UTF-8 でそのまま表されます。たとえば、スペースは %20 とコード化されます。

- **新しいクライアントコールバック API**　クライアント側でデータのロードとアンロードを実行する新しい機能のサポートのために、新しいクライアントコールバック API が追加されました。Embedded SQL については、「db_register_a_callback 関数」『SQL Anywhere サーバ プログラミング』の「DB_CALLBACK_VALIDATE_FILE_TRANSFER」を参照してください。ODBC については、「SQLSetConnectAttr 拡張接続属性」『SQL Anywhere サーバ プログラミング』の SA_REGISTER_VALIDATE_FILE_TRANSFER_CALLBACK を参照してください。

- **SQL_ATTR_CONNECTION_DEAD による切断された接続の検出の高速化**　ODBC の SQLGetConnectAttr 呼び出しを使用して SQL_ATTR_CONNECTION_DEAD 属性を取得すると、接続が切断されていた場合、切断後にサーバに要求が送信されていなくても、値 SQL_CD_TRUE が取得されるようになりました。接続が切断したかどうかの確認は、サーバに要求を送信しないで行われ、切断された接続は数秒以内に検出されます。接続が切断されるのには、アイドルタイムアウトなどの複数の理由があります。この変更の前は、SQL_ATTR_CONNECTION_DEAD で値 SQL_CD_TRUE が取得されたのは、接続が明示的に切断されたか、接続が切断された後に、SQLExecDirect を呼び出すなどして ODBC ドライバからサーバに要求が送信された場合だけでした。「接続属性を取得する方法」『SQL Anywhere サーバ プログラミング』を参照してください。

- **JDBC ドライバでの ResultSet.getBlob().getBinaryStream() のサポート**　iAnywhere JDBC ドライバでは、現在 ResultSet.getBlob() メソッドがサポートされています。このメソッドは、JDBC 仕様ではオプションです。オプションの ResultSet.getBlob().getBinaryStream() メソッドのサポートが追加されました。「JDBC 4.0 API のサポート」『SQL Anywhere サーバ プログラミング』を参照してください。

- **iAnywhere JDBC ドライバで URL ヘッダとして jdbc:odbc に加えて jdbc:ianywhere を許可**　これまでは、URL ヘッダ jdbc:odbc を使用するアプリケーションでは、この URL への接続時に JDBC ドライバマネージャで iAnywhere JDBC ドライバが使用されると見なすことができました。しかし、Java VM の最近のバージョンでは JDBC ドライバとして Sun JDBC-ODBC ブリッジが登録されるようになり、Sun JDBC-ODBC ブリッジでは jdbc:odbc で始まる URL も許可されるので、アプリケーションに iAnywhere JDBC ドライバではなく Sun JDBC-ODBC ブリッジが使用される可能性が高くなりました。JDBC ドライバマネージャで、Sun JDBC-ODBC ブリッジではなく iAnywhere JDBC ドライバが確実に使用されるようにするには、アプリケーションで URL ヘッダ jdbc:ianywhere を使用してください。「JDBC クライアントアプリケーションからの接続」『SQL Anywhere サーバ プログラミング』を参照してください。

- **ODBC ドライバマネージャでの driver=iAnywhere Solutions 11 - Oracle の許可**　UNIX ODBC ドライバマネージャで driver=iAnywhere Solutions 11 - Oracle が許可されるようになりました。また、アプリケーションがスレッド化されている場合は、スレッド化された Oracle 用 iAnywhere ODBC ドライバがロードされます。アプリケーションがスレッド化されていない場合、ドライバはロードされません。スレッド化されていない Oracle 用 iAnywhere ODBC ドライバはサポートされていません。「SQL Anywhere 16 - Oracle ODBC ドライバ」『Mobile Link サーバ管理』を参照してください。

- **ODBC ドライバマネージャでの driver=UltraLite 11 の許可**　UNIX ODBC ドライバマネージャでは、driver=SQL Anywhere 10 が許可され、SQL Anywhere ODBC ドライバがロードされます (アプリケーションに応じて、スレッドバージョンまたは非スレッドバージョン)。UNIX ODBC ドライバマネージャでは driver=SQL Anywhere 11 と driver=UltraLite 11 も許可されるようになりました。Ultra Light ドライバの場合、ドライバマネージャでは Ultra Light ODBC ドライバのスレッドバージョンだけがロードされます。このドライバはスレッドバージョンだけが存在します。

- **TDS 接続の強化**　SQL Anywhere データベースサーバで、Open Client のログインサーバ名がデフォルトデータベースの名前と一致しなくてもデフォルトデータベースへの TDS 接続が可能になりました。ただし、接続文字列でデータベースを起動していないこと (DBF=... がないこと)、またデータベースサーバでデータベースが 1 つだけ実行されていること、という条件を満たしている必要があります。

- **管理ツールのランチャの再配備の簡素化**　データベースツール (Sybase Central、DBISQL、DBConsole、ML Monitor) のランチャ実行プログラムの再配備が簡単になりました。JAR ファイルの場所に関するレジストリのエントリやセットディレクトリ構造が不要になりました。各実行プログラムには、ツールのロード方法を詳細に示す (実行ファイルと同じ名前の) *.ini* ファイルが、実行ファイルと同じディレクトリに必要です。「管理ツールの配備」『SQL Anywhere サーバ プログラミング』を参照してください。

- **SQL Anywhere の .NET データプロバイダでの分散トランザクションのエンリストのサポート**　.NET 2.0 フレームワークで、トランザクションアプリケーションを記述するためのクラスが含まれる新しいネームスペース System.Transactions が導入されました。クライアントアプリケーションで 1 つまたは複数の参加者が存在する分散トランザクションを作成し、そのトランザクションに参加できます。クライアントアプリケーションでは、TransactionScope クラスを使用して、暗黙的にトランザクションを作成できます。接続オブジェクトでは、TransactionScope によって作成されたアンビエントトランザクションの存在を検出し、自動的にエンリストできます。クライアントアプリケーションでは、CommittableTransaction を作成し、EnlistTransaction メソッドを呼び出してエンリストすることもできます。

  この機能は SQL Anywhere .NET データプロバイダでサポートされています。分散トランザクションには、大きなパフォーマンスのオーバヘッドがあります。非分散トランザクションにデータベーストランザクションを使用することをおすすめします。「Transaction 処理」『SQL Anywhere サーバ プログラミング』を参照してください。

- **SQL Anywhere .NET データプロバイダでの名前付きパラメータのサポート**　SQL Anywhere プロバイダで、SACommand の名前付きパラメータがサポートされるようになりました。ユーザがすべてのパラメータ名を指定すると、コマンドの実行時にプロバイダによってパラメータ値がマッピングされます。名前付きパラメータを使用するとき、パラメータの順序が、ホスト変数の順序と一致している必要はありません。

```
SACommand cmd = new SACommand(
   "UPDATE MyTable SET name = :name WHERE id = :id", conn );

SAParameter p1 = new SAParameter(
   "id", SADbType.Integer );
p1.Direction = ParameterDirection.Input;
p1.Value = 1;
cmd.Parameters.Add( p1 );
```

```
SAParameter p2 = new SAParameter(
   "name", SADbType.Char, 40 );
p2.Direction = ParameterDirection.Input;
p2.Value = "asdasd";
cmd.Parameters.Add( p2 );

cmd.ExecuteNonQuery();
```

- **Web サービスの強化**　このリリースでは、次の Web サービスが強化されています。

  - **HTTP:POST タイプの Web クライアントサービスプロシージャの拡張によるユーザ定義の本文の許可**　CREATE PROCEDURE 文と CREATE FUNCTION 文の TYPE 句が拡張され、MIME タイプを指定できるようになりました。「CREATE FUNCTION 文 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』または「CREATE PROCEDURE 文 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **Web サービスクライアントプロシージャの拡張による PUT、DELETE、HEAD HTTP の各メソッドのサポート**　Web サービスクライアントのプロシージャと関数で PUT、DELETE、HEAD HTTP の各メソッドがサポートされるようになりました。CREATE PROCEDURE 文と CREATE FUNCTION 文の TYPE 句が拡張され、これらのメソッドがサポートされるようになりました。POST メソッドと同様に、PUT では TYPE 句内に Content-Type の拡張を必要とし、1 つの (非代入) パラメータだけが許可されています。「CREATE SERVICE 文 [HTTP Web サービス]」『SQL Anywhere サーバ SQL リファレンス』、「CREATE SERVICE 文 [SOAP Web サービス]」『SQL Anywhere サーバ SQL リファレンス』、「CREATE FUNCTION 文 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』、「CREATE PROCEDURE 文 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **sa_http_php_page システムプロシージャと sa_http_php_page_interpreted システムプロシージャ**　新しい Web サービスシステムプロシージャ sa_http_php_page と sa_http_php_page_interpreted は、PHP インタプリターを経由して渡された PHP スクリプトの結果を返します。「sa_http_php_page システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』と「sa_http_php_page_interpreted システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **HTTP_BODY システム関数**　Web サービスの新しい関数が追加されました。HTTP_BODY 関数は、HTTP 要求の本文をバイナリ形式で返します。「HTTP_BODY 関数 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

  - **WSDLC での Web サービスクライアント SOAP プロシージャ生成のサポート**　WSDLC では、SQL Anywhere 用の SQL SOAP (Web サービス) クライアントプロシージャの生成がサポートされるようになりました。WSDLC では、WSDL1.1 準拠の URL またはファイルが読み込まれ、WSDL 内に記述された対応する各 SOAP 操作にマッピングする、適切なパラメータと句を持つプロシージャ (または関数) が生成されます。生成された SQL 文は SQL ファイルに書き込まれます。

  - **FORMAT 'CONCRETE' 句を指定して定義した HTTP SOAP サービスに EXPLICIT OFF または ON を追加可能**　HTTP SOAP サービスを作成するとき、FORMAT 句のデフォルトは EXPLICIT ON です。このため、DISH サービスによって生成される WSDL では、結果セットで返されるカラムごとに明示的な名前とデータ型が指定されます。したがって、結果セットを表すクライアント側のオブジェクトとインタフェースを SOAP クライアント

ツールキットで自動的に生成でき、カラム値へのネイティブアクセスが可能です。この機能が追加される前は、カラム値は、抽象 XML データ要素としてのみアクセスできました。EXPLICIT OFF を指定すると、この以前の動作を有効にできます。

EXPLICIT 応答オブジェクトまたは汎用の SimpleDataset の定義方法の詳細については、「CREATE SERVICE 文 [SOAP Web サービス]」『SQL Anywhere サーバ SQL リファレンス』と「チュートリアル : JAX-WS を使用した SOAP/DISH Web サービスへのアクセス」『SQL Anywhere サーバ プログラミング』を参照してください。

○ **JSON Web サービスのサポート**　SQL Anywhere で、JSON 形式の応答を返す Web サービスがサポートされるようになりました。「CREATE SERVICE 文 [HTTP Web サービス]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

● **Web サービスのクライアントのロギング**　データベースサーバで、Web サービスクライアントの接続を出力ファイルにロギングできるようになりました。-zoc サーバオプションを指定するか、WebClientLogFile プロパティと WebClientLogging プロパティを sa_server_option システムプロシージャとともに使用して、ログを制御し、Web サービスクライアントのログファイルの場所を指定できます。-sf サーバオプションを使用してこの機能の使用を無効にすることもできます。次の項を参照してください。

○「-zoc データベースサーバオプション」『SQL Anywhere サーバ データベース管理』
○「-sf データベースサーバオプション」『SQL Anywhere サーバ データベース管理』
○「sa_server_option システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』

Windows Mobile の強化

次に、SQL Anywhere バージョン 11.0.0 での Windows Mobile に関する強化を示します。

● **-gss サーバオプションのサポート**　Windows CE 4 (Pocket PC 2003) 以降で、-gss サーバオプションを使用して、内部実行スレッドのデフォルトのスタックサイズを指定できます。「-gss データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

● **チェックサムがデフォルトで有効**　データベースが Windows Mobile で実行されているとき、データベースに対してチェックサムが有効になっているかどうかに関係なく、データベースサーバによってチェックサムが自動的に有効にされます。この機能を使用するには、既存のデータベースをアップグレードするか、データベースを新規に作成する必要があります。「チェックサムを使用した破損の検出」『SQL Anywhere サーバ データベース管理』を参照してください。

UNIX/Linux の強化

次に、SQL Anywhere バージョン 11.0.0 での UNIX と Linux に関する強化を示します。

● **テンポラリファイルのパーミッションの制御**　これまでのリリースでは、データベースサーバとクライアントによって作成されたテンポラリファイルには、読み込み、書き込み、実行のグローバルパーミッションがありました。SATMP 環境変数を、目的のパーミッションを持つ

ディレクトリに設定することで、テンポラリファイルのパーミッションを制御できます。「SATMP 環境変数」『SQL Anywhere サーバ データベース管理』を参照してください。

- **SELinux のサポート**　SELinux のポリシーを使用して、アプリケーションのシステムリソースへのアクセスを制御します。Red Hat Enterprise Linux 5 で SQL Anywhere を使用するときにデフォルトのポリシーを使用できますが、この方法で実行した場合、SQL Anywhere は保護されません。このリリースでは、Red Hat Enterprise Linux 5 で SQL Anywhere を保護するポリシーが含まれるようになりました。ポリシーを使用するには、コンパイルしてインストールする必要があります。ポリシーのソースコードが SQL Anywhere のインストールに含まれます。

  SQL Anywhere の SELinux ポリシーのコンパイルとインストールについては、*$SQLANY11/selinux/readme* を参照してください。

- **Linux の [アプリケーション] メニュー項目**　SQL Anywhere 11 を Linux にインストールするときは、**[アプリケーション]** メニュー項目を作成することを選択できます。

Mac OS X の強化

- **Mac OS X での暗号化のサポート**　Mac OS のデータベースサーバとクライアントの両方で通信の RSA 暗号化がサポートされるようになりました。強力な暗号化の使用の詳細については、「トランスポートレイヤセキュリティ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Mac OS X での HTTPS のサポート**　Mac OS X のデータベースサーバで HTTPS 通信がサポートされるようになりました。HTTPS の使用の詳細については、「-xs データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

その他

次に、SQL Anywhere バージョン 11.0.0 でのその他の強化を示します。

- **無効なビューの変更**　これまでは、INVALID ステータスの通常のビューは変更できず、変更が必要な場合はビューを削除してから再作成する必要がありました。このリリースでは、無効なビューの定義を変更して、有効になるようにすることができます。

- **LENGTH 関数への同意語の追加**　LENGHT 関数の実行時、LENGTH の同義語として LEN を使用できるようになりました。「LENGTH 関数 [文字列]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **ビッグエンディアンとリトルエンディアンの UTF-16 エンコードのサポート**　SQL Anywhere では、プラットフォームのエンディアンに関係なく、すべてのプラットフォームでビッグエンディアンとリトルエンディアンの両方の UTF-16 エンコードがサポートされるようになりました。LOAD TABLE 文と UNLOAD 文、また CSCONVERT 関数で UTF-16 エンコードを使用できます。ただし、接続またはデータベースのエンコードとして UTF-16 エンコードを使用することはできません。「LOAD TABLE 文」『SQL Anywhere サーバ SQL リファレンス』と「UNLOAD 文」『SQL Anywhere サーバ SQL リファレンス』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。「SQL Anywhere サーバのアップグレード」289 ページを参照してください。

- **インデックスのパフォーマンスの向上**　特にキャッシュが満杯になったときのインデックスのパフォーマンスが向上しました。インデックスのパフォーマンスを向上するには、インデックスを再構築する必要があります。最も簡単な方法は、データベースを再構築することです。再構築したら、データベースファイルが大幅に小さくなっている可能性があります。これは正常な現象であり、問題ではありません。

- **圧縮されたカラムでの INLINE と PREFIX の設定の尊重**　これまでは、カラムに対して指定された INLINE と PREFIX の設定は、カラムが圧縮されていた場合は無視され、0 として処理されていました。このリリースでは、カラムが圧縮されていても、カラムに対するこれらの設定が尊重されるようになりました。「BLOB の考慮事項」『SQL Anywhere サーバ データベース管理』と「CREATE TABLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **バッチでのホスト変数の使用許可**　いくつかの制約付きで、バッチ内でのホスト変数の参照が許可されるようになりました。「バッチ」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **IN リストアルゴリズムの強化**　これまでは、IN リスト内のすべての要素が定数値であるか、最適化時に評価すると定数値になる場合にのみ、オプティマイザで IN リストアルゴリズムが使用されました。このリリースでは、IN リスト述部に、開くときだけに評価される値 (CURRENT DATE、CURRENT TIMESTAMP、非決定的システム関数、ユーザ定義関数) や、クエリブロック 1 回の実行中に定数である値 (外部参照) を含めることができるようになりました。

- **単純な DML 文でのプランのキャッシュ**　プランのキャッシュが拡張され、クエリを省略できる SELECT 文 (単純な文) が含まれるようになりました。「プランのキャッシュ」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **新しいデータベースのサイズ縮小**　システムテーブルの次のカラムが圧縮され、新しい (空の) データベースのサイズが約 200 KB 縮小されました。これは、Windows Mobile で使用するデータベースを作成するときに役立ちます。

  - ISYSEVENT.action
  - ISYSJARCOMPONENT.contents
  - ISYSPROCEDURE.proc_defn
  - ISYSSOURCE.source
  - ISYSTEXTCONFIG.char_stoplist
  - ISYSTEXTCONFIG.nchar_stoplist
  - ISYSTRIGGER.trigger_defn
  - ISYSVIEW.view_def

- **デフォルトパケットサイズと最小パケットサイズの拡大**　Windows Mobile を除くすべてのオペレーティングシステムで、デフォルトのパケットサイズが 7300 バイトに拡大されました。Windows Mobile では、デフォルトは 1460 バイトのままです。最小パケットサイズは 500 バイトに拡大されました。「CommBufferSize (CBSIZE) 接続パラメータ」『SQL Anywhere サーバ データベース管理』と「-p データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **リモートデータアクセスのための新しい ODBC クラスのサポート**　次の ODBC クラスのサポートが追加されました。

  - MSACCESSODBC
  - MYSQLODBC
  - ULODBC
  - ADSODBC

  詳細については、「リモートデータアクセスのサーバクラス」『SQL Anywhere サーバ SQL の使用法』を参照してください。

  注意
  これまで SQL Anywhere for MS Access 移行ユーティリティ (upsize ツール) を使用して Microsoft Access データベースを SQL Anywhere に移行していた場合、今後は MSACCESSODBC クラスを使用できます。

- **データベースサーバのメッセージの強化**　データベースサーバからのメッセージにカテゴリと重要度が割り当てられるようになりました。この情報には sa_server_messages システムプロシージャを使用してアクセスできます。また、MessageCategoryLimit プロパティを使用して、維持するメッセージ数を設定できます。「sa_server_messages システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **a_validate_type 列挙体の新しい VALIDATE_COMPLETE パラメータ**　a_validate_type 列挙体に、データベースに対して可能な検証をすべて実行できる新しいパラメータ VALIDATE_COMPLETE が追加されました。a_validate_db 構造体 [データベースツール]『SQL Anywhere サーバ プログラミング』を参照してください。

- **外部アンロードの強化**　データベースの外部アンロードを行うとき、生成される *reload.sql* の先頭に、コメント付き CREATE DATABASE 文が含まれるようになりました。この文を使用して、アンロードしたデータベースと同じデータベースを作成できます。

  アンロードしたデータベースがバージョン 9 以前の SQL Anywhere で作成され、カスタム照合があった場合は、COLLATION 句は次のようになります。

  COLLATION collation-label DEFINITION collation-definition

  ここで *collation-definition* はカスタム照合を指定する文字列です。

  アンロードしたデータベースが強力な暗号化を使用して作成されていた場合は、CREATE DATABASE 文の KEY 句の値が疑問符 3 つ (???) として表示されます。

  詳細については、「内部アンロードと外部アンロード、内部再ロードと外部再ロード」『SQL Anywhere サーバ データベース管理』を参照してください。

- **新しい SQL Anywhere Extension Agent OID**　このリリースには、次の OID が追加されています。

  - saAgent.saRestart
  - saAgent.saInifile

詳細については、「SQL Anywhere MIB リファレンス」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Deadlock システムイベント**　Deadlock システムイベントは、デッドロックが発生したときに起動します。イベントハンドラでは、sa_report_deadlocks プロシージャを使用して、デッドロックが発生するに至った状況に関する情報を取得できます。Deadlock システムイベントを使用するには、既存のデータベースをアップグレードする必要があります。「システムイベント」『SQL Anywhere サーバ データベース管理』を参照してください。

- **データベースの上限の増加**　SQL Anywhere データベースの制限値がいくつか増加しました。「SQL Anywhere のサイズと数の制限」『SQL Anywhere サーバ データベース管理』を参照してください。

- **実行プランの変更**　オプティマイザによって生成される長いプランに、プラン全体に関連する次のエントリが表示されるようになりました。

  - **見積り済み最良プラン**　オプティマイザが検出した最良アクセスプランの数。

  - **見積り済みプラン**　オプティマイザが算出したアクセスプランの数。

  - **最適化時間**　クエリの最適化に要した時間。

  「実行プランのコンポーネント」『SQL Anywhere サーバ SQL の使用法』を参照してください。

  グラフィカルなプランには次のエントリが表示されるようになりました。

  - **見積り済み最良プラン**　このエントリはルートノードの **[オプティマイザ統計]** セクションにあり、オプティマイザが検出した最良アクセスプランの数を示します。

  - **見積り済みプラン**　このエントリはルートノードの **[オプティマイザ統計]** セクションにあり、オプティマイザが算出したアクセスプランの数を示します。

  - **最適化時間**　このエントリはルートノードの **[オプティマイザ統計]** セクションにあり、クエリの最適化に要した時間を示します。

  - **FirstRowRunTime**　このエントリは **[ノード統計]** セクションにあり、最初のローのフェッチ時間を示します。

  - **算出されたジョイン**　このエントリはジョイン演算子の **[高度な詳細]** セクションにあり、ジョイン演算子の右側のサブツリーの最適化処理中にオプティマイザが算出したジョイン演算子のリストを示します。

  - **事前フィルタ述部**　このエントリは **[詳細]** ウィンドウ枠の新しいスキャンノードのセクションにあり、スキャンの開始前に評価される述部のリストを示します。

  - **スキャン述部**　このエントリは **[詳細]** ウィンドウ枠のスキャンノードのセクションにあり、ローからフェッチされるカラムとして評価される述部のリストを示します。スキャン述部でローが拒否された場合、それ以上カラムは読み込まれません。スキャン述部は T.x <= 3 や T.x IS NULL などのシンプルな 1 つのカラムの述部です。

- **ポストスキャン述部**　このエントリは **[詳細]** ウィンドウ枠の新しいスキャンノードのセクションにあり、ローがテーブルページから読み込まれた直後に評価される述部のリストを示します。ポストスキャン述部では、複数のカラムを参照でき、また関数や算術演算を使用できます。

- **未確定述部**　このエントリは **[詳細]** ウィンドウ枠の新しいスキャンノードのセクションにあり、ローのセットがメモリにフェッチされた後に評価される述部のリストを示します。未確定述部は、通常はサブクエリやユーザ定義関数などの複雑な操作が含まれ、スキャン述部やポストスキャン述部として評価できません。

- **算出されたインデックス**　**[高度な詳細]** ウィンドウ枠にあるこのエントリは、このスキャン演算子によって参照されているテーブルの最適化処理中にオプティマイザが算出したインデックススキャンまたはテーブルスキャンのリストを示します。リスト内の各項目のフォーマットは、**[詳細]** ウィンドウ枠内のアクセスプランで使用されているスキャン演算子の詳細と似ています。

- **Primary Key Table**　このエントリはインデックススキャン演算子の **[インデックス]** セクションにあり、プライマリキーテーブル名を示します。

- **Primary Key Table Estimated Rows**　このエントリはインデックススキャン演算子の **[インデックス]** セクションにあり、プライマリキーテーブル内のロー数を示します。

- **Primary Key Column**　このエントリはインデックススキャン演算子の **[インデックス]** セクションにあり、プライマリキーカラムの名前を示します。

- **連続変換**　このエントリはインデックススキャン演算子の **[インデックス]** セクションにあり、各物理インデックスについて、インデックスのクラスタ化の程度を示す統計を示します。

- **ランダム変換**　このエントリはインデックススキャン演算子の **[インデックス]** セクションにあり、各物理インデックスについて、インデックスのクラスタ化の程度を示す統計を示します。

- **キー値**　このエントリはインデックススキャン演算子の **[インデックス]** セクションにあり、インデックス内のユニークなエントリの数を示します。

SQL Anywhere の動作の変更

次に、SQL Anywhere バージョン 11.0.0 での動作の変更を、カテゴリごとに示します。

- **カタログの変更**　次の表は、バージョン 11.0.0 でのカタログの変更点を示しています。

  これらの変更点を使用するには、データベースをアップグレードする必要があります。「SQL Anywhere サーバのアップグレード」289 ページを参照してください。

テーブル名とビュー名	変更点の説明
ISYSTAB/SYSTAB	○新しいカラム dbspace_id が追加されました。既存の file_id カラムは最終的にこのカラムよって置き換えられます。 ○file_id カラムは推奨されなくなりました。今後は dbspace_id を使用してください。グローバルテンポラリテーブルでは、SYSTAB.file_id はシステム DB 領域ではなく、テンポラリ DB 領域を指すようになりました。 ○テーブルを変更したトランザクションのシーケンス番号を格納する last_modified_tsn という新しいカラムが追加されました。
ISYSIDX/SYSIDX	○新しいカラム dbspace_id が追加されました。既存の file_id カラムは最終的にこのカラムよって置き換えられます。 ○file_id カラムは推奨されなくなりました。今後は dbspace_id を使用してください。
ISYSFILE	このシステムテーブルは推奨されなくなりました。lob_map を除くすべてのカラムは新しい ISYSDBSPACE システムテーブルにあります。lob_map カラムは新しい ISYSDBFILE システムテーブルにあります。
ISYSDBFILE/SYSDBFILE	DB 領域に関する情報を格納するための新しいテーブルです。
ISYSDBSPACE/SYSDBSPACE	DB 領域に関する情報を格納するための新しいテーブルです。
SYSDBSPACEPERM/ISYSDBSPACEPERM	DB 領域のパーミッションを格納するための新しいテーブルです。
ISYSOBJECT/SYSOBJECT	file_id カラムの名前が dbspace_id に変更されました。また、object_type カラムに 17 (テキスト設定) と 18 (DB 領域) の 2 つの新しい値が格納されるようになりました。

テーブル名とビュー名	変更点の説明
SYSINDEXES	外部キーとプライマリキーのインデックスを他のインデックスと区別するために、indextype フィールドでそれぞれ「外部キー」および「プライマリキー」と識別されるようになりました。
ISYSCAPABILITYNAME	このテーブルはカタログに含まれなくなりました。対応する SYSCAPABILITYNAME システムビューはまだ使用できますが、サーバプロパティを使用して生成されます。
ISYSEVENTTYPE	このテーブルはカタログに含まれなくなりました。対応する SYSEVENTTYPE システムビューはまだ使用できますが、サーバプロパティを使用して生成されます。
ISYSVIEW	マテリアライズドビューをリフレッシュしたトランザクションのシーケンス番号を格納する mv_last_refreshed_tsn という新しいカラムが追加されました。
ISYSLOGINMAP/SYSLOGINMAP	ログインポリシーに関する情報を格納するための新しいテーブルです。
ISYSLOGINPOLICY/SYSLOGINPOLICY	ログインポリシーに関する情報を格納するための新しいテーブルです。
ISYSLOGINPOLICYOPTION/ SYSLOGINPOLICYOPTION	ログインポリシーに関する情報を格納するための新しいテーブルです。
ISYSTEXTCONFIG/SYSTEXTCONFIG	テキスト設定オブジェクトに関する情報を格納するための新しいテーブルです。
ISYSTEXTIDX/SYSTEXTIDX	テキストインデックスに関する情報を格納するための新しいテーブルです。
ISYSTEXTIDXTAB/SYSTEXTIDXTAB	テキストインデックスに関する情報を格納するための新しいテーブルです。

- **PHP 関数名の変更**　PHP 関数名のプレフィクスは、すべて sqlanywhere_ から sasql_ に変更されました。sqlanywhere_ プレフィクスも関数を呼び出すときにまだ使用できますが、廃止される予定です。新しいプレフィクスを使用するようにアプリケーションを変更してください。

- **INSERT...ON EXISTING UPDATE 文でのトリガの起動**　これまでは、INSERT...ON EXISTING UPDATE 文を実行したとき、データが更新された場合はトリガは起動しませんでした。このリリースでは、更新に対して文レベルの AFTER トリガが起動します。

- **REFRESH MATERIALIZED VIEW 文**　リフレッシュの独立性レベルとして STATEMENT SNAPSHOT と READONLY STATEMENT SNAPSHOT を指定できなくなりました。独立性レベルに SNAPSHOT を指定するとこれらのオプションと同じ効果が得られます。「REFRESH MATERIALIZED VIEW 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **REORGANIZE TABLE 文**　同じテーブルに対して複数の REORGANIZE TABLE 文を同時に実行しようとするとエラーが発生します。

- **sa_validate システムプロシージャ**　sa_validate の check_type、エクスプレス、チェックサムの引数が廃止され、指定しても効果はありません。このリリースでは、デフォルトでチェックサム検証が実行されます。また、引数を指定しないで sa_validate システムプロシージャを呼び出すと、データベースサーバでは、すべてのテーブル、マテリアライズドビュー、インデックスの検証に加えて、チェックサムを含むデータベース自体が検証されます。「sa_validate システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **-gss サーバオプション**　Windows XP 以降で -gss サーバオプションがサポートされるようになりました。これまでのリリースでは、このオプションは Windows オペレーティングシステムではサポートされていませんでした。「-gss データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **-gx サーバオプションのサポート終了**　このリリースでは、-gx サーバオプションがサポートされなくなりました。SQL Anywhere データベースサーバの起動時に -gx オプションを指定すると、エラーが発生します。

- **LazyClose 接続パラメータのデフォルト設定が AUTO に変更**　これまでのバージョンでは、アプリケーションでカーソルを閉じるとき、LazyClose 接続パラメータが NO に設定されていなかった場合はデータベースサーバへの往復が必要でした。このリリースでは、カーソルを閉じる要求はデフォルトで複数のカーソル分がキューイングされるようになったので、往復がなくなり、パフォーマンスが向上します。LazyClose 接続パラメータには、YES、NO、AUTO (デフォルト) の 3 つの値を指定できるようになりました。これまでのリリースでは、YES がデフォルト設定でした。「LazyClose (LCLOSE) 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Embedded SQL のインポートライブラリの変更**　DBLIB のインポートライブラリの Watcom と Borland のバージョンが含まれなくなりました。これらはそれぞれ *dblibtw.lib* と *dblibtb.lib* です。これらのインポートライブラリの代わりにインポート定義ファイル (*%SQLANY11%¥SDK¥Lib¥Def¥dblib.def* ファイル) が用意されています。

- **データベースツールのインポートライブラリの変更**　データベースツールのインポートライブラリの Watcom と Borland のバージョンが含まれなくなりました。これらはそれぞれ *dbtlstw.lib* と *dbtlstb.lib* です。これらのインポートライブラリの代わりにインポート定義ファイル (*%SQLANY11%¥SDK¥Lib¥Def¥dbtool.def* ファイル) が用意されています。

- **ローを受信しなかったときの DBLIB インジケータの動作の定義**　フェッチ時または実行時にデータベースサーバからローを受信しなかった場合 (エラーが発生したか、結果セットの末尾

に到達した場合)、インジケータの値が変更されなくなりました。「インジケータ変数」『SQL Anywhere サーバ プログラミング』を参照してください。

- **ODBC SQLGetConnectAttr**　ODBC の SQLGetConnectAttr 呼び出しを使用して SQL_ATTR_CONNECTION_DEAD 属性を取得すると、接続が切断されていた場合、切断後にサーバに要求が送信されていなくても、値 SQL_CD_TRUE が取得されるようになりました。接続が切断したかどうかの確認は、サーバに要求を送信しないで行われ、切断された接続は数秒以内に検出されます。接続が切断されるのには、アイドルタイムアウトなどの複数の理由があります。

  これまでのリリースでは、SQL_ATTR_CONNECTION_DEAD で値 SQL_CD_TRUE が取得されたのは、接続が明示的に切断されたか、接続が切断された後に、SQLExecDirect を呼び出すなどして ODBC ドライバからサーバに要求が送信された場合だけでした。

- **utility_db という名前のデータベースを作成または起動できない**　utility_db という名前は SQL Anywhere サーバユーティリティデータベースのために予約されています。*utility_db.db* という名前のデータベースを新規に作成するか、この名前の既存のデータベースを起動しようとすると、エラーが返されます。utility_db という名前の既存のデータベースがある場合は、別の名前で起動できます。「ユーティリティデータベース」『SQL Anywhere サーバ データベース管理』を参照してください。

- **計算カラムの依存性**　これまでは、更新操作または挿入操作をエラーなしで続行するには、アプリケーションでトリガを使用して、NOT NULL と宣言されているカラムに NULL 以外の値を割り当てることができました。これは、このカラムに依存する計算カラムに影響し、意図した計算が計算値に反映されない場合がありました。このリリースでは、計算カラムが依存する NOT NULL カラムに NULL 値を設定しようとするとエラーメッセージが表示され、トリガは起動しません。「計算カラムへの挿入と更新」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **ピリオドを含む DB 領域名のエラー**　これまでのリリースでは、引用符で囲んでいない DB 領域名にピリオドが含まれていた場合、DB 領域名のピリオドの前の部分は、サーバで通知されることなく無視されていました。このリリースでは、このような名前があった場合、データベースサーバでエラーが生成されます。

- **SQL Anywhere Web サーバでの SSL バージョン 2.0 のサポート終了**　SQL Anywhere Web サーバを使用するとき、SSL バージョン 3.0 と TLS バージョン 1.0 の接続だけがサポートされます。SSL バージョン 2.0 の接続はサポートされません。

- **CREATE SERVICE オプションの DATATYPE のデフォルト値の変更**　DATATYPE 句のデフォルト値が OFF から ON に変更されました。以前の動作が必要な場合は、CREATE SERVICE の定義に DATATYPE OFF を明示的に含めてください。「CREATE SERVICE 文 [SOAP Web サービス]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **チェックサムの動作の変更**　バージョン 11 で作成したデータベースまたはバージョン 11 にアップグレードしたデータベースを、ネットワークドライブやリムーバブルドライブなどのメディアで実行した場合、データベースサーバによってチェックサムが自動的に有効にされます。データベースがこのようなデバイスにあるかぎり、チェックサムは有効なままです。「チェックサムを使用した破損の検出」『SQL Anywhere サーバ データベース管理』を参照してください。

- **HTTP 接続でデータベースが自動停止されない**　これまでのリリースでは、データベースを自動的に停止するように設定すると、HTTP 接続が切断され、データベースへの接続が他になかった場合にデータベースが停止していました。このリリースでは、最後の Command Sequence 接続または TDS 接続が切断したときにのみデータベースが自動的に停止します。

  データベースへの唯一の接続が HTTP 接続で、自動的に停止するようにデータベースが設定されている場合、HTTP 接続が切断したときにデータベースは自動的に停止しません。また、自動的に停止するように設定されているデータベースに HTTP 接続と Command Sequence 接続または TDS 接続がある場合は、最後の Command Sequence 接続または TDS 接続が切断したときにデータベースが停止します。このときに HTTP 接続がまだあった場合は切断されます。「-ga データベースサーバオプション」『SQL Anywhere サーバ データベース管理』と「AutoStop (ASTOP) 接続パラメータ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **データベースミラーリングの動作の変更**　これまでのリリースでは、プライマリサーバまたはミラーサーバの -xp オプションで指定した接続パラメータが無効であった場合、データベースサーバで接続が繰り返し試行されましたが、接続できませんでした。このリリースでは、-xp オプションで指定したパラメータが無効であり、サーバで複数のデータベースが実行されている場合、ミラーデータベースの起動に失敗し、再接続は試行されません。ミラーデータベースが、データベースサーバで実行されている唯一のデータベースである場合は、データベースサーバが起動しません。

- **マテリアライズドビューのデフォルトのリフレッシュ動作**　これまでは、マテリアライズドビューのデフォルトのリフレッシュ動作は WITH EXCLUSIVE MODE でした。このリリースでは、デフォルトのリフレッシュ動作は、マテリアライズドビューが IMMEDIATE REFRESH と定義されているかどうか、またデータベースでスナップショットアイソレーションレベルが有効になっているかどうかによって異なります。「REFRESH MATERIALIZED VIEW 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **post_login_procedure データベースオプションの動作の変更**　post_login_procedure データベースオプションのデフォルト設定が sa_post_login_procedure システムプロシージャになりました。「post_login_procedure オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **non_keywords データベースオプション**　これまでのリリースでは、個々のキーワードの指定に加え、次に示すキーワードリストの中から特別値を使用して、特定のリリース以降のすべてのキーワードをオフにできました。

  keywords_4_0_d, keywords_4_0_c, keywords_4_0_b, keywords_4_0_a, keywords_4_0, keywords_5_0_01, keywords_5_0

  これらの特別値はサポートされなくなりました。個々のキーワードをオフにすることはできます。「non_keywords オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **cooperative_commit_timeout データベースオプション**　コミット動作が自動的にチューニングされるため、現在ではこのオプション設定は無視されるようになりました。

- **cooperative_commits データベースオプション**　コミット動作が自動的にチューニングされるため、現在ではこのオプション設定は無視されるようになりました。

- **リモートデータアクセスでの quoted_identifier データベースオプション設定の尊重**　リモートデータアクセスを使用している場合、quoted_identifier オプションのローカル設定を使用して、Adaptive Server Enterprise と Microsoft SQL Server に対する引用符付き識別子の使用を制御できるようになりました。たとえば、quoted_identifier オプションをローカルで Off に設定すると、Adaptive Server Enterprise に対して引用符付き識別子がオフになります。次の項を参照してください。

  - 「サーバクラス ASEODBC」『SQL Anywhere サーバ SQL の使用法』
  - 「サーバクラス MSSODBC」『SQL Anywhere サーバ SQL の使用法』

- **precision と scale の各データベースオプションのスコープの変更**　これまでのリリースでは、個々のユーザに precision と scale の各データベースオプションを設定したり、設定のスコープがテンポラリであることを指定したりできました。ただし、これらの設定はデータベースのリカバリ性に影響する可能性があります。テーブルやドメインを作成または変更する DDL 文の実行時に、テンポラリ設定またはユーザレベルの設定が、対応する PUBLIC 設定と異なる場合、データベースの再構築時に問題が生じる可能性があります。このリリースでは、precision と scale の各データベースオプションの動作は次のようになっています。

データベースサーバのバージョン	バージョン 10 以前のデータベース	バージョン 11 のデータベース	バージョン 11 にアップグレードしたデータベース	バージョン 10 以前のデータベースのアンロード
11	PUBLIC 設定は可能 ユーザ設定は可能 テンポラリ設定は**不可**	PUBLIC 設定は可能 ユーザ設定は**不可** テンポラリ設定は**不可**	PUBLIC 設定は可能 ユーザ設定は**不可** テンポラリ設定は**不可**	PUBLIC 設定はアンロードされる ユーザ設定はアンロード時に破棄される
10 以前	PUBLIC 設定は可能 ユーザ設定は可能 テンポラリ設定は可能	なし	なし	PUBLIC 設定はアンロードされる ユーザ設定はアンロードされる

バージョン 10 以前のデータベースサーバでは、引き続き scale と precision の各オプションをテンポラリに、また個々のユーザに設定できます。

警告
precision と scale の各データベースオプションのユーザレベル設定またはテンポラリ設定を頼りにしないことをおすすめします。データベースの再構築時に問題が生じる可能性が含まれており、データベースサーバが予測できない動作をする可能性もあります。

次の項を参照してください。

○「precision オプション」『SQL Anywhere サーバ データベース管理』
○「scale オプション」『SQL Anywhere サーバ データベース管理』

● **OPTION 句の動作の変更**　INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT、INTERSECT の各文の OPTION 句でサポートされていないデータベースオプションを指定すると、エラーが返されるようになりました。次の項を参照してください。

○「INSERT 文」『SQL Anywhere サーバ SQL リファレンス』
○「UPDATE 文」『SQL Anywhere サーバ SQL リファレンス』
○「DELETE 文」『SQL Anywhere サーバ SQL リファレンス』
○「SELECT 文」『SQL Anywhere サーバ SQL リファレンス』
○「UNION 文」『SQL Anywhere サーバ SQL リファレンス』
○「EXCEPT 文」『SQL Anywhere サーバ SQL リファレンス』
○「INTERSECT 文」『SQL Anywhere サーバ SQL リファレンス』

● **読み込み専用データベースのロールバックログの動作の変更**　これまでのリリースでは、トランザクション指向テンポラリオブジェクトを含んだ読み込み専用データベースでの操作は、ロールバックログ情報が保持されないため、トランザクションとして扱われませんでした。このリリースでは、読み込み専用データベース内のトランザクション指向テンポラリオブジェクトに完全にトランザクション指向のセマンティックがあります。これらのオブジェクトは、コミット、ロールバック、セーブポイントまでのロールバックの対象となります。

● **Itanium 64 ビットのサポート対象プラットフォームの変更**　これまでのバージョンでは、Itanium II チップ搭載の Windows Server 2003 を対象としたソフトウェアの 64 ビット通常版と、64 ビットの Linux と HP-UX の各オペレーティングシステムを対象としたクライアント／サーバ限定版がありました。

このリリースでは、64 ビット HP-UX を対象としたクライアント／サーバ限定版だけを使用できます。

● **アンロードユーティリティ (dbunload) の動作の変更**　これまでのリリースでは、dbunload の -ea、-ek、-ep の各オプションを、-an オプションまたは -ar オプションと同時に指定して、新しいデータベースの暗号化を制御する必要がありました。このリリースでは、データベースの全体またはその一部をアンロードした後に再ロードしなかった場合、-ea、-ek、-ep の各オプションによって、作成されるテーブルデータファイルの暗号化が制御されます。これらのファイルを使用して Interactive SQL からデータベースを再ロードするとき、READ 文のパラメータとして暗号化キーを指定する必要があります。「アンロードユーティリティ (dbunload)」『SQL Anywhere サーバ データベース管理』を参照してください。

また、これまでのリリースでは、データベースの抽出に使用する dbunload のバージョンが、データベースを実行しているデータベースサーバと同じバージョンである必要はありませんでした。このリリースでは、バージョン 10.0.0 以降のデータベースで dbunload を使用する場合、使用する dbunload のバージョンが、データベースへのアクセスに使用するデータベースサーバのバージョンと一致する必要があります。dbunload のバージョンがデータベースサーバのバージョンより古いまたは新しい場合は、エラーがレポートされます。

● **抽出ユーティリティ (dbxtract) の動作の変更**　これまでのリリースでは、データベースの抽出に使用する dbxtract のバージョンが、データベースを実行しているデータベースサーバと同

じバージョンである必要はありませんでした。このリリースでは、バージョン 10.0.0 以降のデータベースで dbxtract を使用する場合、使用する dbxtract のバージョンが、データベースへのアクセスに使用するデータベースサーバのバージョンと一致する必要があります。dbxtract のバージョンがデータベースサーバのバージョンより古いまたは新しい場合は、エラーがレポートされます。

- **ロックの動作の変更**　これまでのリリースでは、独立性レベル 0 で実行されている UPDATE 文または DELETE 文が、文の処理対象ではないローのローロックをブロックする場合がありました。このリリースでは、UPDATE 文または DELETE 文が、文の処理対象ではないローに意図的ロックまたは排他ロックをかける可能性は低くなっています。アプリケーションを開発する際は、UPDATE 文と DELETE 文に独立性レベル 0 または 1 を使用する場合は注意し、動作がアプリケーションで許容可能であることを確認する必要があります。「更新時のロック」『SQL Anywhere サーバ SQL の使用法』と「削除時のロック」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **プロパティ名の変更**　このリリースでは、次のプロパティの名前が変更されています。

以前の名前	新しい名前
CacheHitsEng	CacheHits
CacheReadEng	CacheRead
DiskReadEng	DiskRead
ReadHint	DiskReadHint
ReadHintScatter	DiskReadHintPages
ReadHintScatterLimit	DiskReadHintScatterLimit

詳細については、「接続、データベース、データベースサーバのプロパティ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **言語選択ユーティリティ (dblang)**　これまでのリリースでは、このユーティリティは、インストール時に多言語リソース開発キット (IRDK) を選択した場合にのみインストールされていました。このリリースでは、すべての多言語リソースと言語選択ユーティリティ (dblang) が常にインストールされます。

- **テンポラリテーブルとインデックスのデフォルトの DB 領域**　テンポラリテーブルは、TEMPORARY DB 領域だけに作成できます。CREATE TABLE 文の IN 句で SYSTEM DB 領域を指定すると、IN 句は無視され、テンポラリテーブルはテンポラリ DB 領域に作成されます。CREATE TABLE 文の IN 句でユーザ定義の DB 領域を指定すると、エラーが返されます。また、テンポラリオブジェクト作成時は default_dbspace オプションが無視されます。

- **テンポラリテーブルへのデータのロード**　テンポラリテーブルにデータをロードするとき、ON COMMIT DELETE と指定されたローカルテンポラリテーブルをロードできなくなりました。これまでのリリースでは、ON COMMIT DELETE ROWS と指定して定義されたローカルテンポラリテーブルにデータをロードできました。

このリリースでは、LOAD TABLE 文を実行すると、オートコミットが自動的に実行されます。これまでのリリースではオートコミットは実行されない場合もありました。

- **データベースサーバオプション**　サーバオプション -uc と -ui が Mac OS X でサポートされるようになりました。これまでは、Linux だけでサポートされていました。Linux では、-ui サーバオプションを使用すると、**[サーバ起動オプション]** ウィンドウが開き、データベースサーバメッセージウィンドウが表示され、X-Window Server が起動するかどうかにかかわらず、データベースサーバが起動します。Mac OS X では、-ui を使用すると、データベースサーバメッセージが新しいウィンドウに表示され、使用可能な表示がない場合はデータベースサーバがシェルモードで起動します。-uc サーバオプションでは、データベースサーバがシェルモードで起動します。「-uc データベースサーバオプション」『SQL Anywhere サーバ データベース管理』と「-um データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Unicode 呼び出しをサポートしない ODBC ドライバでリモートデータアクセスが機能しない**　Unicode 呼び出しをサポートしない ODBC ドライバでリモートデータアクセスが機能しなくなりました。このため、Unicode 以外の ODBC ドライバの場合、リモートデータアクセスでは ODBC ドライバからのデータに対して文字セットの変換が行われません。

- **SYSFILE システムビュー**　SYSFILE 互換ビューにテンポラリファイル用のローが含まれるようになりました。

SQL Anywhere の廃止予定機能とサポート終了機能

- **LOAD TABLE、UNLOAD TABLE、INPUT、OUTPUT の各文の FORMAT ASCII 句は廃止予定**　LOAD TABLE、UNLOAD TABLE、INPUT、OUTPUT の各文の FORMAT ASCII 句が推奨されなくなり、FORMAT TEXT に置き換えられました。dbunload などのユーティリティでは、FORMAT ASCII ではなく FORMAT TEXT を含む再ロードスクリプトが生成されるようになりました。

  OUTPUT 文では、FORMAT TEXT 句によって、旧バージョンでの FORMAT ASCII と同じファイル形式でデータが書き込まれます。これまで FORMAT TEXT によって作成されていた出力は使用できなくなりました。

- **データベースプロパティ**　次のデータベースプロパティはサポートされなくなりました。

  - MapPages
  - PreserveSource
  - UniqueIdentifier

- **サーバプロパティ**　このリリースでは、次のサーバプロパティが推奨されなくなっています。

  - MaxMessage
  - Message
  - MessageTime
  - MessageText
  - MessageWindowSize

- **SPX プロトコルのサポート終了**　このリリースでは、SPX プロトコルがサポートされなくなりました。その結果、次のプロトコルオプションがサポートされなくなりました。

  - ExtendedName プロトコルオプション [ENAME]
  - RegisterBindery プロトコルオプション [REGBIN]
  - SearchBindery プロトコルオプション [BINSEARCH]

  SQL Anywhere .NET データプロバイダの次の機能がサポートされなくなりました。

  - SACommLinksOptionsBuilder クラス：SpxOptionsBuilder プロパティ
  - SACommLinksOptionsBuilder クラス：SpxOptionsString プロパティ
  - SASpxOptionsBuilder クラス

- **dbinit -e オプションのサポート終了**　データベースの作成時に単純暗号化を指定するための dbinit -e オプションはサポートされなくなりました。単純暗号化を指定するには -ea simple オプションを使用してください。「初期化ユーティリティ (dbinit)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **サポート終了したデータベースオプション**　このリリースでは、次のデータベースオプションと、対応するデータベースプロパティがサポートされなくなりました。

オプション	このリリースでの動作
ansi_integer_overflow	オーバーフローが発生すると必ず SQLSTATE = 22003 - overflow エラーが発生するようになりました。 マテリアライズドビューがある古いデータベースにアンロードまたは接続するときは、このオプションの設定は無視されます。
ansi_substring	SUBSTRING 関数の動作が ANSI/ISO SQL/2003 の動作と一致するようになりました。「SUBSTRING 関数 [文字列]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
automatic_timestamp	TIMESTAMP データ型の新規カラムに明示的なデフォルト値が定義されていない場合、デフォルト値として Transact-SQL の timestamp が適用されなくなりました。
divide_by_zero_error	ゼロによる除算があると、SQLSTATE 22012 のエラーが発生します。 マテリアライズドビューがある古いデータベースにアンロードまたは接続するときは、このオプションの設定は無視されます。

オプション	このリリースでの動作
float_as_double	SQL Anywhere で、精度が指定されていない場合、FLOAT キーワードが Adaptive Server Enterprise の FLOAT キーワードと同じように動作しません。SQL Anywhere では、FLOAT 値は DOUBLE 値と同じように処理されません。 Open Client 接続と jConnect 接続では、この動作は、これまでのリリースのデフォルトの動作と異なります。 マテリアライズドビューがある古いデータベースにアンロードまたは接続するときは、このオプションの設定は無視されます。
optimistic_wait_for_commit	このオプションはサポートされなくなりました。
query_plan_on_open	カーソルに対して OPEN を実行したときにプランが返されなくなりました。EXPLAIN 文か PLAN 機能を使用すると詳細な説明が得られます。「EXPLAIN 文 [ESQL]」『SQL Anywhere サーバ SQL リファレンス』と「PLAN 関数 [その他]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
ri_trigger_time	UPDATE または DELETE の後に参照整合性アクションが実行されるようになりました。
truncate_with_auto_commit	TRUNCATE TABLE 文の実行前と実行後の両方に COMMIT が実行されるようになりました。
tsql_hex_constant	16 進定数がバイナリ文字定数として処理されるようになりました。
uuid_has_hyphens	UUID 文字列に 4 つのハイフンが含まれるようになりました。 マテリアライズドビューがある古いデータベースにアンロードまたは接続するときは、このオプションの設定は無視されます。

オプション	このリリースでの動作
percent_as_comment	これまでのリリースでは、percent_as_comment データベースオプションの設定によっては、パーセント記号 (%) をコメントマーカとして使用できました。このリリースでは、% 記号は SQL Anywhere でモジュロ演算子として処理されるようになりました。「MOD 関数 [数値]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **SQLANY10 環境変数のサポート終了**　これまでのリリースでは、SQL Anywhere ソフトウェアの一部が共有ディレクトリにインストールされていました。この場所は SQLANYSH10 環境変数で指定できました。このリリースでは、インストールプロセスによってソフトウェアが共有ディレクトリにインストールされなくなり、SQLANYSH10 環境変数は使用されなくなりました。

  サイレントインストールを作成するときに SHARED_DIR の場所を設定する必要がなくなりました。「SQL Anywhere インストーラを使用したサイレントインストール」『SQL Anywhere サーバ プログラミング』を参照してください。

- **sa_get_server_messages システムプロシージャのサポート終了**　これまでのリリースでは、sa_get_server_messages システムプロシージャを使用して、データベースサーバメッセージウィンドウから結果セットとして定数を返すことができました。このリリースでは、sa_server_messages システムプロシージャを使用して同じ情報を取得できるようになりました。「sa_server_messages システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **background_priority オプションは廃止予定**　background_priority オプションは推奨されなくなりました。今後は priority オプションを使用してください。「priority オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **encrypt_aes_random_iv オプションのサポート終了**　このリリースでは、encrypt_aes_random_iv データベースオプションがサポートされなくなりました。ランダム IV (初期化ベクトル) が常に使用されるようになりました。

- **DLL プロトコルオプションのサポート終了**　DLL プロトコルオプションがサポートされなくなりました。Windows データベースサーバとクライアントでは Winsock 2.2 が使用されます。Windows Mobile クライアントでは Winsock 1.1 が使用されます。「TCP/IP プロトコル」『SQL Anywhere サーバ データベース管理』を参照してください。

- **SQL Anywhere Broadcast Repeater ユーティリティの名前変更**　バージョン 10 では、SQL Anywhere Broadcast Repeater ユーティリティを実行するコマンドは dbns10 でした。このリリースでは、dbns11 です。「Broadcast Repeater ユーティリティ (dbns16)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **SQLANY10 と SQLANYSAMP10 の各環境変数の名前変更**　SQLANY10 と SQLANYSAMP10 の各環境変数は、それぞれ SQLANY11 と SQLANYSAMP11 に名前が変更されました。次の項を参照してください。

  ○「SQLANY16 環境変数」『SQL Anywhere サーバ データベース管理』
  ○「SQLANYSAMP16 環境変数」『SQL Anywhere サーバ データベース管理』

- **一部のセキュリティ機能の名前変更**　このリリースでは、次のセキュリティ機能の名前が変更されています。

以前の名前	新しい名前
xp_read_file	read_file
xp_write_file	write_file
unload_table	write_file
load_table	read_file

  詳細については、「-sf データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

Mobile Link

次の項では、Mobile Link バージョン 11.0.0 の新機能、動作の変更、廃止予定機能について説明します。

Mobile Link の新機能

次に、バージョン 11.0.0 で導入された Mobile Link の追加機能を示します。

統合データベース

- **DB2 メインフレームを統合データベースとしてサポート**　Mobile Link では、これまで DB2 LUW (Linux、UNIX、Windows) が統合データベースとしてサポートされていました。このリリースでは、DB2 メインフレームもサポートされるようになりました。

- **MySQL を統合データベースとしてサポート**　Mobile Link で、統合データベースとして MySQL がサポートされるようになりました。

  「MySQL 統合データベース」『Mobile Link サーバ管理』を参照してください。

- **Mobile Link システムデータベース (MLSD)**　個別データベースに Mobile Link システムデータ (MLSD - Mobile Link システムデータベース) を保持できるようになりました。この機能は Microsoft DTC (分散トランザクションコーディネーター) とともに使用してください。「-cs mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

新しいシステムオブジェクト

- **新しい Mobile Link サーバシステムテーブルとスキーマ**　Mobile Link システムテーブルは、次のように変更されています。

  ○複数の新しい Mobile Link システムテーブルが追加されました。

  - ml_qa_delivery_archive
  - ml_qa_repository_archive
  - ml_qa_repository_props_archive
  - ml_qa_status_history_archive
  - ml_server
  - ml_active_remote_id
  - ml_passthrough
  - ml_passthrough_repair
  - ml_passthrough_script
  - ml_passthrough_status

  ○複数の新しい Mobile Link システムプロシージャが追加されました。次の項を参照してください。

  - 「ml_add_passthrough システムプロシージャ」『Mobile Link サーバ管理』
  - 「ml_add_passthrough_repair システムプロシージャ」『Mobile Link サーバ管理』
  - 「ml_add_passthrough_script システムプロシージャ」『Mobile Link サーバ管理』
  - 「ml_delete_passthrough システムプロシージャ」『Mobile Link サーバ管理』
  - 「ml_delete_passthrough_repair システムプロシージャ」『Mobile Link サーバ管理』
  - 「ml_delete_passthrough_script システムプロシージャ」『Mobile Link サーバ管理』

iAnywhere Solutions Oracle ドライバ

- **Oracle DSN が暗号化されたパスワードを格納**　Windows ODBC アドミニストレータで Oracle ODBC データソースを作成するときに、ODBC データソースに格納されたパスワードを暗号化するよう選択できるようになりました。「SQL Anywhere 16 - Oracle ODBC ドライバ」『Mobile Link サーバ管理』を参照してください。

- **Oracle ODBC ドライバが Microsoft 分散トランザクションをサポート**　Oracle ODBC ドライバで Microsoft 分散トランザクションがサポートされるようになりました。Windows ODBC アドミニストレータで、[Microsoft 分散トランザクションを有効にする] を選択し、Oracle クライアントとともに適切な DLL がインストールされていることを確認してください。「SQL Anywhere 16 - Oracle ODBC ドライバ」『Mobile Link サーバ管理』を参照してください。

Mobile Link サーバ

- **リレーサーバ**　リレーサーバは、Mobile Link サーバ、Afaria サーバ、および OneBridge サーバと、モバイルデバイスとの間の、安全で負荷分散された Web サーバ経由の通信を実現する、Web 拡張機能セットです。「Relay Server の概要」『Relay Server』を参照してください。

- **Sybase リレーサーバのホスティングサービス**　Sybase リレーサーバのホスティングサービスは、Sybase をホストとするリレーサーバのファームです。Mobile Link データ同期を使用するモバイルアプリケーションの開発を容易にすること、また特に公共無線ネットワークを使用

してデータを送信する場合に開発者による評価プロセスを簡素化することを目的としています。「Sybase ホストのリレーサービス」『Relay Server』を参照してください。

- **Mobile Link サーバファーム**　Mobile Link サーバを同一のサーバファームに明示的にグループ化できるようになりました。

  同一のリモート ID による冗長な同時同期をファーム全体から自動的に検出できるようになりました。これにより、負荷分散によって同一のリモート ID を Mobile Link サーバに送信し続ける必要がなくなりました。

  各 Mobile Link サーバに対して同一の設定ができるようになります。

  Notifier に対するフェイルオーバが自動的にサポートされています。ファームにより、自動的に Mobile Link サーバが指定されて Notifier と QAnywhere コネクタが実行されます。最初に指定されたコンピュータで障害が発生した場合は、新しいサーバが選択されて Notifier と QAnywhere コネクタが実行されます。

- **64 ビットのプラットフォーム**　いくつかの 64 ビットプラットフォームで Mobile Link サーバが完全に 64 ビット対応になりました。サポートされる 64 ビットプラットフォームのリストについては、http://www.sybase.com/detail?id=1002288 を参照してください。

- **Java DownloadTableData インタフェースの新しいメンバー**　getLastDownloadTime メソッドによってテーブルの最終ダウンロード時刻が返されます。
  DownloadTableData.getLastDownloadTime メソッド [Mobile Link サーバ Java]『Mobile Link サーバ管理』を参照してください。

- **SQL パススルー**　SQL パススルー機能によって、統合データベースから SQL Anywhere または Ultra Light のクライアントに SQL 文のスクリプトをダウンロードし、クライアント上で適切な時間に SQL 文を実行させることができます。

- **情報メッセージの待機**　Java と .NET API では、プレフィクス "I" が付けられた情報の行がログに出力されたときに通知を受け取るよう登録できるようになりました。

  次の項を参照してください。

  - LogMessage.INFO 変数 [Mobile Link サーバ Java]『Mobile Link サーバ管理』
  - ServerContext.addInfoListener メソッド [Mobile Link サーバ Java]『Mobile Link サーバ管理』
  - ServerContext.removeInfoListener メソッド [Mobile Link サーバ Java]『Mobile Link サーバ管理』
  - LogMessage.MessageType 列挙体 [Mobile Link サーバ .NET]『Mobile Link サーバ管理』
  - ServerContext.InfoListener イベント [Mobile Link サーバ .NET]『Mobile Link サーバ管理』

mlsrv11 の新機能

- **-cs オプション**　システムテーブル、プロシージャ、トリガ、ビューなどの Mobile Link サーバシステムオブジェクトを、統合データベース以外のデータベースに格納できるようになりました。Mobile Link システム オブジェクトを格納するデータベースは、Mobile Link システムデータベース (MLSD) と呼ばれます。

  MLSD の接続パラメータを指定するには、-cs を使用します。「-cs mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **-lsc オプション**　ローカルサーバの接続情報を指定します。この情報は、サーバファームのその他のサーバに渡されます。「-lsc mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **-ss オプション**　Mobile Link サーバをサーバファームで実行できます。

> **注意**
> -ss オプションは、Mobile Link の高可用性オプションの機能であり、別途ライセンスが必要です。「別途ライセンスが必要なコンポーネント」『SQL Anywhere 16 紹介』を参照してください。

- **-tc オプション**　SQL スクリプト実行のカウントダウンタイマーを設定します。「-tc mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **-tf オプション**　-tc を使用して指定されたカウントダウンタイマーの期限が切れたら、SQL スクリプトを実行できないようにします (Oracle は対象外)。「-tf mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

Mobile Link の新しいスクリプト機能

- **非ブロッキングダウンロード ACK スクリプト**　「nonblocking_download_ack 接続イベント」『Mobile Link サーバ管理』と「publication_nonblocking_download_ack 接続イベント」『Mobile Link サーバ管理』を参照してください。

Mobile Link リダイレクタの強化

- **リダイレクタは廃止予定**　リダイレクタは推奨されなくなり、リレーサーバに置き換えられました。「Relay Server の概要」『Relay Server』を参照してください。

- **新しいリレーサーバ**　リレーサーバは、Mobile Link サーバ、Afaria サーバ、および OneBridge サーバと、モバイルデバイスとの間の、安全で負荷分散された Web サーバ経由の通信を実現する、Web 拡張機能セットです。「Relay Server の概要」『Relay Server』を参照してください。

Mobile Link クライアント

SQL Anywhere クライアント

- **C++ および .NET 用の dbmlsync API**　dbmlsync API は、C++ または .NET で記述された Mobile Link クライアントアプリケーションが同期を起動し、要求した同期の進行状況に関するフィードバックを受け取れるようにするプログラミングインタフェースです。「Dbmlsync API」『Mobile Link クライアント管理』を参照してください。

- **同期プロファイル**　dbmlsync コマンドラインを同期プロファイルとしてデータベースに格納できるようになりました。dbmlsync の -sp オプションを使用すると、同期プロファイルからコマンドライン同期オプションにオプションを追加できます。「-sp dbmlsync オプション」『Mobile Link クライアント管理』を参照してください。

- **挿入操作でロードされたローをロギングする LOAD TABLE のオプション機能**　新しい WITH ROW LOGGING 句によって、SQL Anywhere リモートデータベースで LOAD TABLE 文を使用してデータをロードできるようになりました。これまでは、LOAD TABLE ではローのログ

が取られずに dbmlsync から無視されていたため、LOAD TABLE は同期環境において必ずしも有益ではありませんでした。「LOAD TABLE 文を使用したデータのインポート」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **dbmlsync ログスキャンの最適化**　重複のないパブリケーション用に最適化されています。

セキュリティ

このリリースでは、次のセキュリティ機能が追加されています。

- **エンドツーエンド暗号化**　Mobile Link 同期ストリームおよびクライアントで、プロトコルレベルでエンドツーエンド暗号化がサポートされるようになりました。「エンドツーエンド暗号化」『SQL Anywhere サーバ データベース管理』を参照してください。

- **キーペアジェネレータユーティリティ (createkey)**　このユーティリティでは、Mobile Link エンドツーエンド暗号化で使用する RSA と ECC のキーペアを作成します。「キーペアジェネレータユーティリティ (createkey)」『SQL Anywhere サーバ データベース管理』を参照してください。

サーバ起動同期

- **Mobile Link サーバファームのサポート**　SIS が強化され、Mobile Link サーバファーム環境での動作が改善されました。ファーム内のすべての Mobile Link サーバで Notifier を実行できるようになりました。これらの Notifier によって、同じ Listener への冗長な通知がなくなります。「-lsc mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

Listener の強化

- **新しい dblsn オプション**　Windows 用の Mobile Link Listener に次のオプションが追加されています。

  - **-ni**　IP 追跡を無効にします。

  - **-pc-**　永続的な接続を無効にします。

  - **-nu**　デフォルトの UDP 受信を無効にします。

Mobile Link の動作の変更と廃止予定機能

次に、バージョン 11.0.0 で導入された Mobile Link の変更点を示します。

Mobile Link サーバの変更

- **デフォルトのダウンロード確認の変更**　-nba オプションのデフォルト値が -nba+ になり、デフォルトで非ブロッキングダウンロード確認が実行されるようになりました。既存の配備環

境で、ダウンロード確認をブロックするダウンロードを使用する場合、次のいずれかの操作を行ってください。

○ -nba- オプションを使用する。
○ 同期スクリプトを確認して nonblocking_download_ack スクリプトか publication_nonblocking_download_ack スクリプトまたはその両方を使用する (推奨)。

- **certificate と certificate_password の各プロトコルオプションの名前変更** TLS および HTTPS の certificate と certificate_password の各プロトコルオプションは、それぞれ identity と identity_password に名前が変更されました。「-x mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

Mobile Link クライアントの変更

- **dbmlsync 統合コンポーネントは廃止予定** dbmlsync 統合コンポーネントは推奨されなくなり、dbmlsync API に置き換えられました。

  「Dbmlsync API」『Mobile Link クライアント管理』を参照してください。

- **dbmlsync StreamCompression 拡張オプションのサポート終了** このオプションはサポートされなくなりました。

- **-lt 拡張オプションのデフォルトは OFF に変更** dbmlsync の LockTables (-lt) 拡張オプションのデフォルトは以前は ON でした。このリリースでは、デフォルトが OFF になり、デフォルトでは dbmlsync によって同期テーブルがロックされません。「LockTables (lt) 拡張オプション」『Mobile Link クライアント管理』を参照してください。

その他の Mobile Link の動作の変更

サーバ起動同期

- **Sierra Wireless Aircards のサポート終了** Sierra Wireless Aircards の SMS 受信ライブラリは、サポートされなくなりました。

- **-g オプションは廃止予定** -g Listener オプションは -ni オプションに置き換えられました。

- **-ga オプションは廃止予定** -ga Listener オプションは推奨されなくなりました。非同期の IP 追跡は、暗黙的に行われます。

- **-gi デフォルトの変更** デフォルトのポーリング間隔が、10 秒から 60 秒に変更されました。

SQL Remote

次の項では、SQL Remote バージョン 11.0.0 の新機能、動作の変更、廃止予定機能について説明します。

SQL Remote の新機能

- **抽出ユーティリティ (dbxtract) の -ea オプションの強化**　dbxtract の -ea オプションでは、暗号化タイプとして、なし (none) と単純 (simple) の両方を使用できるようになりました。none を指定すると、暗号化は行われません。simple を指定すると、単純暗号化が行われます。また、-ea とともに -ek、-et、-ep のどのオプションが指定されたかに応じて、デフォルトの暗号化タイプが変更されました。「抽出ユーティリティ (dbxtract)」『SQL Remote』を参照してください。

- **抽出ユーティリティ (dbxtract) への -ap、-er、-et、-nl オプションの追加**　dbxtract で -ap、-er、-et、-nl の各オプションを使用できるようになりました。「抽出ユーティリティ (dbxtract)」『SQL Remote』を参照してください。

SQL Remote の動作の変更と廃止予定機能

次に、バージョン 11.0.0 で導入された SQL Remote の変更点を示します。

- **挿入操作でロードされたローをロギングする LOAD TABLE のオプション機能**　新しい WITH ROW LOGGING 句によって、SQL Anywhere リモートデータベースで LOAD TABLE 文を使用してデータをロードできるようになりました。これまでは、LOAD TABLE ではローのログが取られずに dbremote または dbltm から無視されていたため、LOAD TABLE は同期環境やレプリケート環境において必ずしも有益ではありませんでした。「LOAD TABLE 文を使用したデータのインポート」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **VIM と MAPI の各メッセージタイプのサポート終了**　このリリースでは、VIM と MAPI の各メッセージシステムがサポートされなくなりました。VIM または MAPI を使用するデータベースを SQL Anywhere バージョン 11.0.0 にアップグレードする場合は、メッセージタイプを File、FTP、または SMTP に変更してください。メッセージタイプが MAPI または VIM の場合、*dbremote.exe* は起動しません。

  最も簡単な変更方法は、SMTP/POP に切り替えることです。メッセージタイプを変更すると、SMTP/POP をサポートするためにメールサーバの変更が必要になる場合があります。

  SQL Remote メッセージタイプの選択については、「SQL Remote メッセージシステム」『SQL Remote』を参照してください。

Ultra Light

次の項では、Ultra Light バージョン 11.0.0 の新機能、動作の変更、廃止予定機能について説明します。

Ultra Light の新機能

次に、バージョン 11.0.0 で導入された Ultra Light の追加機能を示します。

主な機能

- **同期プロファイルのサポート**　Ultra Light 11.0.0 と 11.0.1 で同期プロファイルがサポートされます。「ALTER SYNCHRONIZATION PROFILE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』、「CREATE SYNCHRONIZATION PROFILE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』、「DROP SYNCHRONIZATION PROFILE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』を参照してください。

- **Ultra Light SELECT 文**　Ultra Light の SELECT 文に明示的な FOR 句がない場合、デフォルトで FOR READ ONLY になります。この変更によって、Ultra Light では、更新が許可されていないクエリにより適したプランを選択できるようになります。「SELECT 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』を参照してください。

- **Ultra Light SYNCHRONIZE 文**　Ultra Light 同期プロファイルまたは特定の同期オプションを同期するための新しい文です。「SYNCHRONIZE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』を参照してください。

- **Ultra Light CREATE SYNCHRONIZATION PROFILE 文**　Ultra Light 同期プロファイルを作成するための新しい文です。同期プロファイルによって、Ultra Light データベースが Mobile Link サーバと同期する方法を定義します。「CREATE SYNCHRONIZATION PROFILE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』を参照してください。

- **Ultra Light ALTER SYNCHRONIZATION PROFILE 文**　Ultra Light 同期プロファイルを変更するための新しい文です。「ALTER SYNCHRONIZATION PROFILE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』を参照してください。

- **Ultra Light DROP SYNCHRONIZATION PROFILE 文**　Ultra Light 同期プロファイルを削除するための新しい文です。「DROP SYNCHRONIZATION PROFILE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』を参照してください。

- **SQL Anywhere パススルースクリプトのサポート**　この機能は Ultra Light 12 ではサポートされていません。

  Ultra Light ユーティリティで、SQL Anywhere パススルースクリプトがサポートされるようになりました。この変更点は次のユーティリティに適用されます。

  - ulcond11
  - ulunload
  - ulload
  - ulinfo
  - ulsync

  次の項を参照してください。

  - 「Ultra Light データベースのアンロードユーティリティ (ulunload)」『Ultra Light データベース管理とリファレンス』
  - 「Ultra Light データベースへの XML のロードユーティリティ (ulload)」『Ultra Light データベース管理とリファレンス』
  - 「Ultra Light 情報ユーティリティ (ulinfo)」『Ultra Light データベース管理とリファレンス』
  - 「Ultra Light 同期ユーティリティ (ulsync)」『Ultra Light データベース管理とリファレンス』

- **Ultra Light データベース検証**　ulvalid ユーティリティまたは ValidateDatabase API を使用して Ultra Light データベースを検証できるようになりました。検証によって、データベースファイル内に特定タイプの破損がないかテストされます。コマンドラインパラメータを使用して結果を絞り込むことができます。「Ultra Light データベース検証ユーティリティ (ulvalid)」『Ultra Light データベース管理とリファレンス』と「Ultra Light データベースの検証」『Ultra Light データベース管理とリファレンス』を参照してください。

  Ultra Light .NET で ValidateDatabase 関数がサポートされるようになりました。接続の有無にかかわらず、データベースや特定のテーブルを検証できるようになりました。ULDatabaseManager クラス [Ultra Light.NET]『Ultra Light .NET プログラミング』と ULConnection クラス [Ultra Light.NET]『Ultra Light .NET プログラミング』を参照してください。

  Sybase Central の**データベース検証ウィザード**を使用して、Ultra Light データベースを検証できるようになりました。**[データベースの検証]** オプションは、**[ツール]** メニューから選択できます。

- **イベントと通知のサポート**　Ultra Light でイベントと通知がサポートされるようになりました。イベントの発生時に、通知メッセージが登録されたキューまたは接続に送信されます。アプリケーションによるユーザイベントの定義やトリガも可能です。イベントと通知の API は、サポートされている各言語で提供されます。また、API 機能にアクセスするための SQL 関数も提供されています。

- **Ultra Light における独立性レベルのサポート**　接続が複数ある場合、デフォルトで互いに独立するようになりました。他の接続によるコミットされていない変更およびダウンロードは、コミットされるまで参照できません。

  独立性レベルを READ_COMMITTED または READ_UNCOMMITTED に設定できるようになりました。「独立性レベル」『Ultra Light データベース管理とリファレンス』と「Ultra Light でのトランザクション処理」『Ultra Light データベース管理とリファレンス』を参照してください。

  Ultra Light .NET で ReadUncommitted 独立性レベルがサポートされるようになりました。オートコミットモードでの接続の独立性レベルは、デフォルトで ReadCommitted になります。「Ultra Light でのトランザクション処理」『Ultra Light データベース管理とリファレンス』と「独立性レベル」『Ultra Light データベース管理とリファレンス』を参照してください。

- **Ultra Light ALTER DATABASE SCHEMA FROM FILE 文**　ALTER DATABASE SCHEMA FROM FILE 文を使用して Ultra Light スキーマを変更できるようになりました。ALTER DATABASE SCHEMA FROM FILE 文は、現在は削除されている UpgradeSchemaFromFile メソッドまたは ApplyFile メソッドによって実装されたバージョン 9.0.2 におけるスキーマアップグレード機能を置き換えるものとなります。ulinit ユーティリティまたは ulunload ユーティリティのいずれかを使用して、必要な DDL 文が構文的に正しいことを確認してください。

次の項を参照してください。

- 「ALTER DATABASE SCHEMA FROM FILE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』
- 「Ultra Light データベーススキーマのアップグレードの配備」『Ultra Light データベース管理とリファレンス』
- 「Ultra Light データベース初期化ユーティリティ (ulinit)」『Ultra Light データベース管理とリファレンス』
- 「Ultra Light データベースのアンロードユーティリティ (ulunload)」『Ultra Light データベース管理とリファレンス』

- **データベース抽出ウィザードの動作の変更**　抽出プロセスからテーブルを除外できるようになりました。**データベース抽出ウィザード**では、使用可能なパブリケーションのリストから重複する名前の付いたパブリケーションが省略されます。「Ultra Light のアップグレード」331 ページを参照してください。

- **Ultra Light クライアントバージョンとビルド番号の Mobile Link ログファイルへの追加**　同期中に、Ultra Light クライアントがバージョンとビルド番号を Mobile Link サーバログに追加するようになりました。次の項を参照してください。

  - 「Mobile Link サーバログの表示」『Mobile Link サーバ管理』
  - 「Ultra Light の同期パラメータ」『Ultra Light データベース管理とリファレンス』

- **Ultra Light LOAD TABLE 文**　LOAD TABLE 文をデスクトップコンピュータ上で実行できるようになりました。Ultra Light LOAD TABLE 文を参照してください。「LOAD TABLE 文 [Ultra Light]」『Ultra Light データベース管理とリファレンス』を参照してください。

- **バックグラウンド同期のサポート**　アプリケーションのどの時点からでも別のスレッドの同期を開始できるようになりました。Ultra Light では、アップロードの開始時にコミットされたローのみがアップロードされます。アップロード中にデータベースを変更し、アップロードに影響を及ぼさずにその変更をコミットできるようになりました。アップロード中にコミットされたローは、アップロードからは無視されます。「Ultra Light の同時実行性」『Ultra Light データベース管理とリファレンス』を参照してください。

- **強化された GUID 識別子のサポート**　Ultra Light のこれまでのバージョンでは、16 バイトのバイナリまたは文字列で表した UUID (ユニバーサルユニーク識別子) または GUID (グローバルユニーク識別子) 識別子の入出力がランタイムにより許可されていました。これらの識別子はエンディアン変換によって GUID 構造体と互換にされていました。Ultra Light 11 では、GUID 構造体は、ランタイムから明示的に入出力できるため、エンディアン変換が不要になりました。

- **ul_stream_error 構造体**　Ultra Light 11 では、stream_id フィールド、stream_context フィールド、および error_string_length フィールドが削除されました。また、error_string フィールドが、ユーザ定義の char * から静的 char 配列に変更されました。

プラットフォームとデバイス

- **64 ビット Windows プラットフォーム (64 ビットの XP 以降) へのネイティブ amd64/x64 ESQL および C++ アプリケーション配備のサポート**　Ultra Light で、64 ビット Windows プラット

フォーム (64 ビットの XP 以降) に対するネイティブ amd64/x64 ESQL および C++ 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。ただし、64 ビットのコンピュータで Ultra Light アプリケーションを開発するには、Ultra Light のユーティリティの 32 ビットバージョンを使用する必要があります。また、64 ビットのコンピュータで Ultra Light に接続する場合は、DBISQL と Sybase Central の 32 ビットバージョンを使用する必要があります。

- **Linux (32 ビット) にポートされた Ultra Light ユーティリティ** 「ユーティリティ」267 ページを参照してください。

セキュリティ

- **エンドツーエンド暗号化** Ultra Light で、プロトコルレベルでエンドツーエンド暗号化がサポートされるようになりました。データは、暗号ブロック連鎖 (CBC) モードで 128 ビット AES を使用して暗号化され、キー交換は RSA または ECC によって処理されます。

ユーティリティ

- **Ultra Light データベースの XML へのアンロードユーティリティ (ulunload)** -s オプションを使用してスキーマをアンロードし、SQL Anywhere 互換のフォーマットでデータを保存できるようになりました。「Ultra Light データベースのアンロードユーティリティ (ulunload)」『Ultra Light データベース管理とリファレンス』を参照してください。

- **Ultra Light データベース初期化ユーティリティ (ulinit)** -d オプションを使用して SQL Anywhere データベースから新しい Ultra Light データベースにデータをコピーできるようになりました。「Ultra Light データベース初期化ユーティリティ (ulinit)」『Ultra Light データベース管理とリファレンス』を参照してください。

- **ulerase** Ultra Light データベース (データベースに関連するテンポラリファイルを含む) を消去するための新しいユーティリティプログラムです。このユーティリティには、データベースへのアクセスを確認するためのユーザ ID とパスワードが必要です。「Ultra Light ユーティリティ消去 (ulerase)」『Ultra Light データベース管理とリファレンス』を参照してください。

- **ulvalid** Ultra Light データベースを検証するための新しいユーティリティプログラムです。検証によって、データベースファイル内に特定タイプの破損がないかテストされます。コマンドラインパラメータに従って検証を設定できます。「Ultra Light データベース検証ユーティリティ (ulvalid)」『Ultra Light データベース管理とリファレンス』を参照してください。

- **Linux にポートされた Ultra Light ユーティリティ (32 ビットバージョンのみで使用可能)**
  - **ulcreate** 新しい空の Ultra Light データベースを作成します
  - **ulerase** Ultra Light データベースと関連するチェックポイントファイルを永久的に消去します
  - **ulinfo** 既存の Ultra Light データベースに関する情報を表示します
  - **ulinit** SQL Anywhere リファレンスデータベースで使用可能なスキーマから新しい Ultra Light データベースを作成します

○ **ulload**　ulunload によって保存された XML データから Ultra Light データベースを作成し、ロードします

○ **ulsync**　Mobile Link を転送エージェントとして使用して、Ultra Light データベースを統合データベースと同期します

○ **ulunload**　Ultra Light データベースを XML にアンロードします

○ **ulvalid**　Ultra Light データベースに妥当性検査を実行します

ulunloadold ユーティリティは Linux では使用できません。

SQL

● **IF 文、CASE 文、CASE 式の強化**　互換性の向上のため、IF 式を ENDIF または END IF で終了できるようになりました。また、CASE 式を END または END CASE で終了できるようになりました。「IF 式」『Ultra Light データベース管理とリファレンス』と「CASE 式」『Ultra Light データベース管理とリファレンス』を参照してください。

プログラミングインタフェース

一般的な向上点

パブリケーションマスクは、パブリケーションリストに置き換えられました。キーワード Publications は、パブリケーション名をカンマで区切ったリストで指定します。

Ultra Light C/C++

新しいメソッド、UltraLite_Table* OpenTableEx() は、UltraLite_Connection オブジェクトの一部になりました。このメソッドによって、SQL 以外のアプリケーションからテーブルを開いたり、ローのダイレクトスキャンを実行する際の汎用性が向上します。「ダイレクトページスキャン」『Ultra Light データベース管理とリファレンス』を参照してください。

このメソッドを使用すると、次のいずれかの方法を指定してテーブルを開ことができます。

●プライマリキーの順序でローを返すには、ul_table_open_primary_key を使用します。

●任意の順序でローを返すには、ul_table_open_no_index を使用します。

●インデックスによって指定された順序でローを返すには、ul_table_open_with_index を使用します。

Ultra Light Embedded SQL

●エラー解釈に関連する 2 つの関数のマニュアルについては、ULGetErrorParameter メソッド [Ultra Light Embedded SQL]『Ultra Light C/C++ プログラミング』と ULGetErrorParameterCount メソッド [Ultra Light Embedded SQL]『Ultra Light C/C++ プログラミング』を参照してください。

- パブリケーションマスクからパブリケーションリストへの変更を反映するため、ULGetLastDownloadTime、ULResetLastDownloadTime、および ULCountUploadRows の各関数の構文が変更されました。

- 関数 ULGetPublicationMask は使用できなくなりました。

Ultra Light.NET

- ULDataReader クラスに、指定された最大ロー数までローカウントを取得できる新しいメソッド GetRowCount(threshold) が追加されました。

- ULDataReader クラスで IListSource インタフェースが実装されるようになりました。

Ultra Light for M-Business Anywhere

- Connection クラスにイベント処理と通知用の新しいメソッドとして次が追加されました。cancelGetNotification、createNotificationQueue、declareEvent、destroyNotificationQueue、getNotification、getNotificationParameter、registerForEvent、sendNotification、triggerEvent。

Ultra Light J

Ultra Light J は Java による Ultra Light の実装で、Java SE 環境や BlackBerry スマートフォンを含む Java ME 環境をサポートします。「システムの要件とサポート対象プラットフォーム」『Ultra Light Java プログラミング』を参照してください。

Ultra Light の動作の変更と廃止予定機能

次に、バージョン 11.0.0 で導入された Ultra Light の変更点を示します。

廃止予定のプラットフォーム

- Ultra Light C++ インタフェースで Symbian OS がサポートされなくなりました。Symbian 向けの Ultra Light アプリケーションの開発者は、Ultra Light J を使用してください。

- Ultra Light.NET インタフェースでは、.NET 1.0 コンポーネントがサポートされなくなりました。.NET 1.0 API リファレンスは、マニュアルから削除されました。

バージョン 11 では、Ultra Light for AppForge はサポートされていません。

データベースプロパティ

このリリースでは、次のデータベースプロパティが推奨されなくなっています。

- CollationName

接続パラメータ

このリリースでは、Ultra Light の次の接続パラメータが推奨されなくなりました。

- ORDERED_TABLE_SCAN

削除されたユーティリティ

C++ API 移行ウィザードは Sybase Central で使用できなくなりました。

削除、廃止予定、変更された関数

- ESQL インタフェースの ULGetPublicationMask 関数は削除されました (パブリケーションマスクはパブリケーションリストに置き換えられました)。
- パブリケーションマスクからパブリケーションリストへの変更を反映するため、ESQL インタフェースの ULGetLastDownloadTime、ULResetLastDownloadTime、および ULCountUploadRows の各関数の構文が変更されました。
- M-Business Anywhere API で、次の関数は推奨されなくなりました。DatabaseSchema.getTableCountInPublications、DatabaseSchema.SYNC_ALL_DB、DatabaseSchema.SYNC_ALL_PUBS、PublicationSchema.getMask、SyncParms.getPublicationMask、SyncParms.setPublicationMask。
- M-Business Anywhere および .NET API では、次のメソッドはパラメータとしてパブリケーションマスクを取っていました。このリリースでは、パラメータがパブリケーションリスト (文字列) に変更されました。影響を受けるメソッドは、Connection クラスの countUploadRows、getLastDownloadTime、および resetLastDownloadTime です。
- ul_sync_info 構造体が変更され、**disable_concurrency**、**checkpoint_store**、**table_order** の各フィールドが削除されました。これらのオプションは、**additional_parms** という新しいフィールドに、キーワードと値の組み合わせをセミコロンで区切って指定します。

その他

- **CustDB の 2 番目のインスタンスの実行**　これまでのバージョンでは、CustDB の 2 番目のインスタンスを実行すると、エラーが発生しました。このバージョンでは、CustDB の 2 番目のインスタンスを起動すると、最初のインスタンスがフォアグランドに移され、2 番目のインスタンスは終了します。

Sybase Central と Interactive SQL

次の項では、バージョン 11.0.0 の Sybase Central と Interactive SQL での新機能、動作の変更、廃止予定機能について説明します。

Sybase Central と Interactive SQL の新機能

次に、バージョン 11.0.0 で導入された Sybase Central と Interactive SQL の追加機能を示します。

- **新しい高速ランチャ方式**　これまで、Interactive SQL および Sybase Central の高速ランチャは、ユーザのログイン時に起動していました。このバージョンでは、高速ランチャは Interactive SQL または Sybase Central の起動時にのみ起動します。起動されると、高速ランチャはアプリケーションのシャットダウン後 30 分間動作するよう設定されています。この新しい方式によって、30 分以内に実行される後続の Interactive SQL および Sybase Central の高速ランチャをより短時間で起動できるようになります。「高速ランチャオプション」『SQL Anywhere サーバ データベース管理』を参照してください。
- **[接続] ウィンドウの強化と変更**　Sybase Central および Interactive SQL の **[接続]** ウィンドウが、次のように強化されています。

- **新しい [ODBC データソースとして保存] ツール**　**[ODBC データソースとして保存]** ツールを使用すると、**[接続]** ウィンドウで指定した接続パラメータを使用して ODBC データソースを生成できます。このツールを使用するには、**[接続]** ウィンドウの **[ツール]** ボタンをクリックして、**[ODBC データソースとして保存]** をクリックし、画面の指示に従います。「ODBC データソースの作成 (管理ツール)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **新しい [接続文字列をクリップボードにコピー] ツール**　**[接続文字列をクリップボードにコピー]** ツールを使用すると、接続文字列をクリップボードにコピーして、データベースサーバに渡されている値を確認できます。このツールを使用するには、**[接続]** ウィンドウの **[ツール]** ボタンをクリックして、**[接続文字列をクリップボードにコピー]** をクリックします。接続文字列をテキストエディタに貼り付けて表示します。

- **接続アシスタント**　**[接続アシスタント]** ツールが、**[接続]** ウィンドウの右半分に表示されるようになりました。**[接続アシスタント]** は、データベースへの接続を支援するように設計された、ウィザードのような機能です。

- **[詳細] タブの強化**　**[詳細]** タブに、プロパティの表とプロパティの簡単な説明が表示されるようになりました。

- **新しい [ネットワーク] タブ**　**[ネットワーク]** タブを使用して、サポートされているプロトコルの共有メモリと TCP/IP のオプションを指定できます。この新しいタブが、これまで **[データベース]** タブの **[ネットワーク上でデータベースサーバを検索]** オプションによって提供されていた機能を置き換えます。

- **キーボードショートカット**　Sybase Central のコードエディタと、Interactive SQL の **[SQL 文]** ウィンドウ枠で、キーボードショートカットを使用してコードとコメントのインデントを増減できるようになりました。また、キーボードショートカットを使用して、二重ハイフンコメントと二重スラッシュコメントの両方のインジケータを追加および削除できます。「Sybase Central キーボードショートカット」『SQL Anywhere サーバ データベース管理』を参照してください。

Sybase Central の新機能

次に、バージョン 11.0.0 で導入された Sybase Central のプラグインの追加機能を示します。

SQL Anywhere プラグインの新機能

- **データベースの [概要] タブ**　データベースサーバとその機能の正常性と統計情報の概要を確認できるようになりました。「データベースの正常性と統計情報」『SQL Anywhere サーバ データベース管理』を参照してください。

- **データベースドキュメント**　**データベースドキュメントウィザード**を使用してデータベースのドキュメントを生成できるようになりました。「データベースのドキュメント化」『SQL Anywhere サーバ データベース管理』を参照してください。

- **新しいウィザード**　SQL Anywhere プラグインに、次の新しいウィザードが含まれるようになりました。

  - ログインポリシー作成ウィザード
  - データベースドキュメントウィザード
  - テキストインデックス作成ウィザード
  - テキスト設定オブジェクト作成ウィザード
  - データベース作成ウィザード

- **新しいプロパティウィンドウ**　SQL Anywhere プラグインに、次の機能のプロパティウィンドウが含まれるようになりました。

  - テキストインデックス
  - テキスト設定オブジェクト
  - 外部環境
  - マテリアライズドビュー
  - データベース監査

- **Sybase Central 内の監査ユーザ設定**　Sybase Central を使用して、データベース監査のユーザ設定を設定し、監査情報を表示できるようになりました。「監査の設定 (Sybase Central の場合)」『SQL Anywhere サーバ データベース管理』と「監査情報の取得 (Sybase Central の場合)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **照合の適合化設定のサポート**　**[データベースのプロパティ]** ウィンドウの **[設定]** タブを使用して、照合の適合化オプションを指定できるようになりました。**データベース作成ウィザード**を使用して、新しいデータベースを作成するときに照合の適合化オプションを指定することもできます。

Ultra Light プラグインの新機能

- **新しいウィザード**　Ultra Light プラグインに、次の新しいウィザードが含まれるようになりました。

  - データベース検証ウィザード
  - 同期プロファイル作成ウィザード

- **新しいプロパティウィンドウ**　Ultra Light プラグインに、次の機能のプロパティウィンドウが含まれるようになりました。

  - 同期プロファイル

Mobile Link プラグインの新機能

- **同期プロファイルのサポート**　Sybase Central で同期プロファイルを作成し、管理できます。

- **新しいウィザード**　Mobile Link プラグインに、次のウィザードが含まれるようになりました。

  - パススルースクリプト作成ウィザード
  - パススルーダウンロード作成ウィザード

- **新しいプロパティウィンドウ**　Mobile Link プラグインに、次の機能のプロパティウィンドウが含まれるようになりました。

  - パススルースクリプト
  - パススルーダウンロード

Interactive SQL の新機能

次に、バージョン 11.0.0 で導入された Interactive SQL の追加機能を示します。

- **データベースロックの自動解放**　Interactive SQL でテーブルを表示すると、テーブルを変更しなくても、テーブルに対するスキーマロックがデータベースサーバによって作成されます。

  このリリースでは、結果セットが表示されるときに作成されるデータベーススキーマロックを Interactive SQL が解放しようとします。

  結果セットを返す文を実行した後、データベースにコミットされていない変更が接続にあるかどうかが Interactive SQL によって確認されます。コミットされていない変更がなかった場合は Interactive SQL によってスキーマロックが解放されます。それ以外の場合、スキーマロックは解放されません。つまり、データベースにコミットされていない変更がある場合、スキーマロックは解放されません。

- **読み込み専用結果セットのサポート**　Interactive SQL で結果セットを読み込み専用にできるようになりました。これを行うには、**[オプション]** » **[結果]** をクリックして、**[編集の無効化]** オプションをクリックします。この設定は、それ以降にフェッチされた結果に適用されます。「Interactive SQL の結果セットでローを編集する」『SQL Anywhere サーバ データベース管理』を参照してください。

  Interactive SQL を展開するときは、*OEM.ini* ファイル内の lockedPreferences のエントリに disableResultSetEditing を追加することで、ユーザがこの設定を変更できないようにすることができます。「管理ツールの設定」『SQL Anywhere サーバ プログラミング』を参照してください。

- **SQL 文を 1 文ずつ実行する**　これまで、SQL 文を 1 文ずつ実行する場合、SQL 文を選択してから **[選択の実行]** を実行する操作を繰り返す必要がありました。このリリースでは、**[シングルステップ]** をクリックすると、選択した文が実行され、次に実行される文が選択されます。**[選択の実行]** と同様に、**[シングルステップ]** は Interactive SQL の **[SQL]** メニューから選択できます。「Interactive SQL の SQL 文」『SQL Anywhere サーバ データベース管理』を参照してください。

- **お気に入りの SQL ファイルとデータベース接続**　よく使用するデータベース接続のリストと SQL スクリプトファイルのリストを、Interactive SQL の **[お気に入り]** メニューで作成および管理できるようになりました。「SQL スクリプトファイル、SQL 文、および接続をお気に入りに追加する」『SQL Anywhere サーバ データベース管理』を参照してください。

- **結果セットのテーブル編集を無効にする**　Interactive SQL を展開するとき、Interactive SQL からの SQL Anywhere と Ultra Light の結果セットのテーブル編集を無効にできるようになりました。「管理ツールの設定」『SQL Anywhere サーバ プログラミング』を参照してください。

- **新しいキーボードショートカット**　Interactive SQL に新しいキーボードショートカットが追加されました。「Interactive SQL のキーボードショートカット」『SQL Anywhere サーバ データベース管理』を参照してください。

- **クライアントアプリケーションでのオプション変更を防止するためのサポート**　ユーザが Interactive SQL のオプション設定を変更しないよう、*OEM.ini* ファイルで一部の設定をロックできるようになりました。「管理ツールの設定」『SQL Anywhere サーバ プログラミング』を参照してください。

- **dbisql の新しい -version オプション**　コマンドプロンプトで dbisql -version と入力すると、Interactive SQL のバージョン番号が表示されます。「Interactive SQL ユーティリティ (dbisql)」『SQL Anywhere サーバ データベース管理』と「Ultra Light 用 Interactive SQL ユーティリティ (dbisql)」『Ultra Light データベース管理とリファレンス』を参照してください。

Sybase Central と Interactive SQL の動作の変更と廃止予定機能

次に、バージョン 11.0.0 で導入された Sybase Central と Interactive SQL の変更点を示します。

- **データベースツールのランチャの再配備の簡素化**　Sybase Central、Interactive SQL、データベースコンソールユーティリティ、Mobile Link モニタの各ランチャ実行プログラムの再配備が簡単になりました。JAR ファイルの場所に関するレジストリのエントリやセットディレクトリ構造が不要になりました。各実行プログラムには、(*filename.exe* ファイルと同じ名前の) 対応する *filename.INI* ファイルが、exe ファイルと同じディレクトリに必要です。*.INI* ファイルには、ツールをロードする方法の詳細が含まれます。「管理ツールの配備」『SQL Anywhere サーバ プログラミング』を参照してください。

- ***OEM.ini* [help] セクションのサポート終了**　*OEM.ini* ファイルの [help] セクションはサポートされなくなりました。詳細については、「管理ツールの設定」『SQL Anywhere サーバ プログラミング』を参照してください。

Sybase Central の動作の変更と廃止予定機能

次に、バージョン 11.0.0 で導入された Sybase Central の変更点を示します。

- **Sybase Central の設定ファイルの名前変更**　*.screpository* ファイルの名前が *.screpository600* に変更されました。

SQL Anywhere プラグインの変更機能

- **プロパティウィンドウの強化**　次のプロパティページが更新されました。

  - **Web サービスのプロパティ** ウィンドウ
  - **ユーザのプロパティ** ウィンドウ
  - **ビューのプロパティ** ウィンドウ
  - **マテリアライズドビューのプロパティ** ウィンドウ

- **特定ユーザのデバッグ**　SQL Anywhere プラグインで **[デバッグ]** モードを選択すると、デバッグの対象となるユーザを指定する必要があります。「チュートリアル：デバッガの使用開始」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- **更新されたウィザード**　次のウィザードが更新されました。

  - ユーザ作成ウィザード
  - **Web** サービス作成ウィザード
  - **Deployment** ウィザード

Mobile Link プラグインの変更機能

- **更新されたウィザード**　次のウィザードが更新されました。

  - 同期モデル展開ウィザード

SQL Anywhere プラグインの廃止予定機能

- **削除されたプロパティタブ**　次のプロパティウィンドウが更新されました。

  - **[テーブルのプロパティ]** ウィンドウ
  - **[Ultra Light プロジェクトのプロパティ]** ウィンドウ
  - **[Ultra Light 文のプロパティ]** ウィンドウ

Interactive SQL の動作の変更と廃止予定機能

次に、バージョン 11.0.0 で導入された Interactive SQL の変更点を示します。

- **プランビューアでグラフィカルプランを表示可能**　Interactive SQL で、プランビューアと呼ばれるサイズ変更可能な別ウィンドウで SQL Anywhere データベースのグラフィカルなプランを表示できるようになりました。この変更によって、[プランビューア] ウィンドウを一度に複数開くことができるので、プランの表示と比較が簡単になります。プランビューアにアクセスするには、**[ツール]** » **[プランビューアを開く]** をクリックします。Ultra Light データベースのテキストプランもプランビューアに表示されます。「詳細とノードの統計を含むグラフィカルなプランの作成」『SQL Anywhere サーバ データベース管理』を参照してください。

  また、Interactive SQL の isql_plan オプションはサポートされていません。

- **長いプランと短いプランの表示はサポート終了**　SQL Anywhere データベースのテキストプランを Interactive SQL で表示できなくなりました。ただし、EXPLANATION 関数と PLAN 関数を使用してテキストプランを取得することはできます。Ultra Light データベースのテキストプランは、Interactive SQL でプランビューアを使用して表示できます。

- **実行プランと結果セットの印刷**　[Ctrl + P] を押すか、**[ファイル]** » **[印刷]** をクリックして、**[SQL 文]** ウィンドウ枠の内容と結果セットを印刷できるようになりました。これまでは、**[SQL 文]** ウィンドウ枠の内容しか印刷できませんでした。プランビューアで印刷するには、**[印刷]** ボタンをクリックします。

- **[SQL 文] ウィンドウ枠への行番号の追加**　**[SQL 文]** ウィンドウ枠の左側に行番号が表示されるようになりました。この行番号によって、構文エラーの場所を特定できるようになります。

- **[SQL 文の実行] ツールバーボタンの強化**　これまでは、[Interactive SQL] ツールバーで、**[SQL 文の実行]** ボタンをクリックするとすべての SQL 文が実行されていました。このリリースでは、ボタンのクリック時に、すべての文を実行するか選択されている文のみを実行するかを指定できます。

  [SQL 文の実行] ボタンの動作を設定するには、**[ツール]** » **[オプション]** » **[ツールバー]** をクリックします。「Interactive SQL の SQL 文」『SQL Anywhere サーバ データベース管理』を参照してください。

- **バッチ文の実行の強化**

  ○Interactive SQL で文のバッチが実行されるときのフィードバックが改善されました。**[SQL 文]** ウィンドウ枠から SQL 文を実行する場合、実行されている文が選択され、全体が見えるようにスクロールされます。**[ファイル]** » **[スクリプトの実行]** をクリックしてスクリプトファイルを実行する場合、スクリプトの進行状況を示すステータスウィンドウが表示されます。「SQL 文の実行 (Interactive SQL)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **[結果] ウィンドウ枠の強化**

  ○**[結果]** ウィンドウ枠で、[Ctrl + A] を押すと、すべての結果を選択できるようになりました。現在フェッチされている結果だけでなく、結果セット全体も選択できます。**[結果]** ウィンドウ枠に結果セット全体が含まれていない場合、残りの結果をフェッチするよう要求されます。残りをフェッチしないと、現在フェッチされている結果のみが選択されます。

  ○**[結果]** タブからセルをコピーすると、コピーされたデータが isql_field_separator、isql_quote、isql_escape_character の各 Interactive SQL オプションに基づいてフォーマットされるようになりました。また、結果セットから選択した値、ロー、およびカラムをクリップボードにコピーすることもできます。「Interactive SQL 結果セットからロー、カラム、セルをコピーする」『SQL Anywhere サーバ データベース管理』を参照してください。

  ○**[結果]** タブのカラムヘッダをクリックすると、そのカラムを基準にして結果がソートされます。**[結果]** ウィンドウ枠に結果セット全体が含まれていない場合、残りの結果をフェッチするよう要求されます。残りをフェッチしないと、現在フェッチされている結果のみがソートされます。

  ○結果セットの選択されたローに基づいて INSERT 文、DELETE 文、UPDATE 文を生成し、クリップボードにコピーできるようになりました。「Interactive SQL の結果セットから SQL 文を生成する」『SQL Anywhere サーバ データベース管理』を参照してください。

  ○Interactive SQL の **[結果]** ウィンドウ枠が強化され、次の機能が追加されました。これらの機能は右クリックメニューから選択できます。

  - **[コピー]** » **[セルのコピー]**　選択されたセルの内容をコピーします。
  - **[コピー]** » **[カラムのコピー]**　選択されたセルのカラム値をコピーします。
  - **[生成]** » **[INSERT 文]**　選択されている各ローに対して INSERT 文を生成し、クリップボードにコピーします。

- **[生成] » [DELETE 文]**　選択されている各ローに対して DELETE 文を生成し、クリップボードにコピーします。
- **[生成] » [UPDATE 文]**　選択されている各ローに対して UPDATE 文を生成し、クリップボードにコピーします。生成された文によって、カラムの値が現在の値に設定されます。したがって、文を実行しても、実際にはカラムの値は変更されません。この機能は、実行に先立って、編集可能なテンプレートの UPDATE 文を提供する際に便利です。

「Interactive SQL 結果セットからロー、カラム、セルをコピーする」『SQL Anywhere サーバ データベース管理』と「Interactive SQL の結果セットから SQL 文を生成する」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Interactive SQL 文の強化**
  - **DESCRIBE 文の強化**　DESCRIBE 文によって、Interactive SQL に接続されているデータベースまたはデータベースサーバに関する情報を返せるようになりました。「DESCRIBE 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
  - **INPUT 文と READ 文の強化**　INPUT 文と READ 文で相対パスの解決を試行する際に、2 つの方法が取られるようになりました。「INPUT 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』と「READ 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
  - **INPUT 文と OUTPUT 文の強化**
    - **ODBC ソースからのインポートとエクスポートの新規サポート**　INPUT 文と OUTPUT 文を使用してデータベースからインポートしたりデータベースにエクスポートしたりするときに、ODBC データソースを指定できるようになりました。これを行うには、新しい USING 句を使用します。「INPUT 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』と「OUTPUT 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

      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も ODBC データソースを指定できます。「インポートウィザードを使用したデータのインポート」『SQL Anywhere サーバ SQL の使用法』と「エクスポートウィザードを使用したデータのエクスポート」『SQL Anywhere サーバ SQL の使用法』を参照してください。
    - **バイト順マーク (BOM) の新規サポート**　データ中のバイト順マーク (BOM) を処理するかどうかを制御できるようになりました。これを行うには、新しい BYTE ORDER MARK 句を使用します。「INPUT 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』と「OUTPUT 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
    - **INPUT 文のサポートフォーマットの変更**　INPUT 文では、dBase、Lotus、Excel および FoxPro の各ファイルフォーマットがサポートされなくなりました。TEXT と FIXED は引き続きサポートされています。サポートされなくなったファイルフォーマットを引き続き使用したい場合は、ODBC ドライバを使用する必要があります。「INPUT 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。
    - **OUTPUT 文のサポートフォーマットの変更**　OUTPUT 文では、dBase、Lotus、Excel および FoxPro の各ファイルフォーマットがサポートされなくなりました。TEXT、

FIXED、HTML、SQL、および XML は引き続きサポートされています。「OUTPUT 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- **dbisqlc ユーティリティの INPUT 文と OUTPUT 文で ASCII フォーマットの名前を TEXT に変更**　dbisqlc ユーティリティと一緒に INPUT 文と OUTPUT 文を使用する場合、このリリースでは次のフォーマットの名前が変更されています。

以前の SQL Anywhere リリースでのフォーマットの名前	新しいフォーマットの名前
ASCII	TEXT
TEXT	COLUMNS

- **インポートウィザードとエクスポートウィザードの変更**　インポートウィザードまたはエクスポートウィザードが終了するとき、ウィザードによって生成された SQL 文がコマンド履歴に保存されます。生成された SQL 文を表示するには、**[SQL]** » **[履歴]** をクリックします。

- **Interactive SQL オプション**

  - **isql_allow_read_client_file と isql_allow_write_client_file**　これらの 2 つのオプションは、クライアント側のファイルの読み込み／書き込み要求に対する Interactive SQL の応答方法を記述します。「isql_allow_read_client_file オプション [Interactive SQL]」『SQL Anywhere サーバ データベース管理』と「isql_allow_write_client_file オプション [Interactive SQL]」『SQL Anywhere サーバ データベース管理』を参照してください。

  - **-codepage オプションは廃止予定**　Interactive SQL に特定のコードページを含むファイルを読み込ませる場合、INPUT 文、OUTPUT 文、または READ 文の ENCODING 句を使用してください。次の項を参照してください。

    - 「Interactive SQL ユーティリティ (dbisql)」『SQL Anywhere サーバ データベース管理』
    - 「Ultra Light 用 Interactive SQL ユーティリティ (dbisql)」『Ultra Light データベース管理とリファレンス』
    - 「INPUT 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』
    - 「OUTPUT 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』
    - 「READ 文 [Interactive SQL]」『SQL Anywhere サーバ SQL リファレンス』

  - **isql_plan オプションのサポート終了**　Interactive SQL の isql_plan オプションはサポートされなくなりました。下位互換のため、このオプションを設定しようとすると通知されずに無視されます。「詳細とノードの統計を含むグラフィカルなプランの作成」『SQL Anywhere サーバ データベース管理』を参照してください。

  - **SET OPTION 文の PUBLIC キーワードの削除**　SET OPTION 文を使用して Interactive SQL のオプションを設定するための PUBLIC キーワードはサポートされなくなりました。「Interactive SQL オプション」『SQL Anywhere サーバ データベース管理』を参照してください。

- **Interactive SQL ランチャの変更**　Windows バージョンの Interactive SQL ランチャの実行プログラムは、*dbisqlg.exe* から *dbisql.exe* に変更されました。

コマンドラインバージョンの Interactive SQL ランチャの実行プログラムは、*dbisql.exe* から *dbisql.com* に変更されました。バッチスクリプトでは、*dbisql.exe* ではなく、*dbisql* または *dbisql.com* を呼び出す必要があります。

SQL Anywhere コンソールユーティリティの動作の変更

次に、バージョン 11.0.0 で導入された SQL Anywhere コンソールユーティリティの変更点を示します。

- **新しいコンソールオプション**　日付と時刻を指定して、**[オプション]** ウィンドウからデータベースサーバを停止できるようになりました。**[ファイル]** メニューで **[オプション]** » **[コンソール]** をクリックします。

Mobile Link モニタの動作の変更

次に、バージョン 11.0.0 で導入された Mobile Link モニタの変更点を示します。

Mobile Link モニタでは、バージョン 9 以前の Mobile Link サーバで作成されたモニタファイルを読み込むことができません。Mobile Link モニタは、同じメジャーバージョンの Mobile Link サーバだけと使用してください。また、ワーカーカラムが削除され、Mobile Link モニタの次のプロパティの名前が変更されました。

古いプロパティ名	新しいプロパティ名
preload_upload	sync_request
verify_upload	authenticate_user

マニュアルの強化

- **マニュアルディレクトリの強化**　以前は、マニュアルは *%SQLANY11%¥docs* にありました。現在、マニュアルは *%SQLANY11%¥documentation* にあります。個々の HTML ヘルプファイルと PDF ファイルのファイル名も更新されました。

HTML ヘルプの新しいファイル名は次のとおりです。

- *dbadmin_ja11.chm*
- *dbprogramming_ja11.chm*
- *dbreference_ja11.chm*
- *dbusage_ja11.chm*
- *mlclient_ja11.chm*
- *mlserver_ja11.chm*
- *mlsisync_ja11.chm*
- *mlstart_ja11.chm*
- *sachanges_ja11.chm*
- *saerrors_ja11.chm*
- *saintro_ja11.chm*
- *scplugin_ja11.chm*
- *sqlanywhere_ja11.chm*
- *sqlremote_ja11.chm*
- *uladmin_ja11.chm*
- *ulc_ja11.chm*
- *uldotnet_ja11.chm*
- *ulj_ja11.chm*
- *ulmbus_ja11.chm*

- **サポートされているプラットフォームのページに Web サイトからアクセス可能**　サポートされているプラットフォームに関する情報は、iAnywhere の Web サイト http://www.sybase.com/detail?id=1002288 にあります。

- **[Locate] ボタン**　HTML ヘルプブラウザで、**[Locate]** ボタンを使用して、現在のヘルプページが目次のどこに含まれているかを確認できるようになりました。

製品全体の機能

次の項では、SQL Anywhere バージョン 11.0.0 のすべてのコンポーネントに影響する新機能、動作の変更、廃止予定機能について説明します。

製品全体の新機能

次に、バージョン 11.0.0 で導入された製品全体の追加機能を示します。

- **エラーレポートの強化**　Windows、Windows Mobile、Linux では、エラーレポートが生成されるとウィンドウが表示されます。iAnywhere に送信するか選択する前に、このウィンドウでエラーレポートの内容を確認できます。「トラブルシューティング：SQL Anywhere でのエラーのレポート」『SQL Anywhere サーバ データベース管理』を参照してください。

  -ce オプションを使用してサポートユーティリティ (dbsupport) を設定できるようになりました。このオプションを使用すると、dbsupport でアプリケーションを監視しているときにアプリケーションがクラッシュすると、電子メールが送信されます。「サポートユーティリティ (dbsupport)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **追加の Windows Mobile プラットフォームのサポート**　SQL Anywhere で、スマートフォン用 Windows Mobile 5 と Windows Mobile 6 Standard エディションがサポートされるようになりました。Windows Mobile での SQL Anywhere サーバの実行については、「SQL Anywhere for Windows Mobile」『SQL Anywhere サーバ データベース管理』と「インストール時の考慮事項：Windows Mobile 5 for Smartphone および Windows Mobile 6 for Smartphone の制限」『SQL Anywhere サーバ データベース管理』を参照してください。

- **SQL Anywhere サンプルデータベース (*demo.db*) の新しいテーブル**　SQL Anywhere サンプルデータベースに新しいテーブル MarketingInformation が追加されました。このテーブルの各ローには、Products テーブル内の製品を説明する HTML ページが格納されています。このテーブルが追加されたことで、機能をテストしたり試したりするときにより複雑な文字データに対してクエリを実行できます。「SQL Anywhere サンプルデータベース」『SQL Anywhere 16 紹介』を参照してください。

製品全体の動作の変更

次に、バージョン 11.0.0 で導入された製品全体の変更点を示します。

- **Windows CE から Windows Mobile への変更**　Windows CE の名前を使用した方が正確である場合を除き、マニュアルとソフトウェアで Windows CE という名前が Windows Mobile に変更されました。

- **readcert、gencert、reqtool の削除**　ユーティリティの readcert、gencert、および reqtool は削除されました。これらは、これまで廃止予定になっていました。これらに代わり、createcert と viewcert を使用できます。「証明書ユーティリティ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **createcert と viewcert の各ユーティリティの Mac OS X でのサポート**　証明書ユーティリティの createcert と viewcert が、Mac OS X でサポートされるようになりました。「証明書ユーティリティ」『SQL Anywhere サーバ データベース管理』を参照してください。

- **certificate と certificate_password の各プロトコルオプションの名前変更**　TLS および HTTPS の certificate と certificate_password の各プロトコルオプションは、それぞれ identity と identity_password に名前が変更されました。次の項を参照してください。

  - SQL Anywhere データベースサーバ：「-ec データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。
  - SQL Anywhere Web サーバ：「-xs データベースサーバオプション」『SQL Anywhere サーバ データベース管理』を参照してください。
  - SQL Anywhere プロトコルオプション：「Identity プロトコルオプション」『SQL Anywhere サーバ データベース管理』と「Identity_Password プロトコルオプション」『SQL Anywhere サーバ データベース管理』を参照してください。
  - Mobile Link サーバ：「-x mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

- **サンプル ID ファイルの変更**　サンプル証明書を含む ID ファイルと、対応する TLS のプライベートキーを含む ID ファイルの名前が、このリリースでそれぞれ変更されています。ファイル *rsaserver.crt* の名前は *rsaserver.id* に、ファイル *sample.crt* の名前は *eccserver.id* に変更され

ました。これらの ID ファイルのパスワードはともに、**tJ1#m6+W** から **test** に変更されました。

- **インストールディレクトリの変更**　このリリースでは、32 ビットのソフトウェアのインストール先が *win32* ディレクトリから *bin32* ディレクトリに、64 ビットのソフトウェアのインストール先が *X64* ディレクトリから *bin64* ディレクトリに変更されました。たとえば、これまでのバージョンで *C:¥Program Files¥SQL Anywhere 11¥win32* にインストールされたソフトウェアは、このリリースでは *C:¥Program Files¥SQL Anywhere 11¥bin32* にインストールされます。

- **サンプルデータベースの ODBC データソースの変更**　これまでのリリースでは、ソフトウェアとともにインストールされたサンプルデータベースの ODBC データソースはユーザデータソースでした。このリリースでは、SQL Anywhere 11 Demo、SQL Anywhere 11 CustDB の各データソースがシステムデータソースになりました。

- **.NET 1.0 のサポート終了**　SQL Anywhere 11 では Microsoft Visual Studio .NET 2002 または Visual Studio .NET 2003 はサポートされません。Microsoft Visual Studio 2005 (.NET 2.0) と Visual Studio 2008 (.NET 3.x) はサポートされています。

SQL Anywhere 16 へのアップグレード方法

メジャーリリース、マイナーリリース、サポートパッケージの違い

- **サポートパッケージ**　サポートパッケージは、1 つまたは複数のバグフィックスを含むソフトウェアのサブセットです。これらのバグフィックスは、更新のリリースノートにリストされます。バグフィックス更新を適用できるのは、同じバージョン番号を持つインストール済みのソフトウェアに対してだけです。ソフトウェアでテストを行っている場合は、そのソフトウェアでアプリケーションを徹底的にテストしないかぎり、アプリケーションとともにこれらのファイルを配布しないでください。

- **マイナーリリース**　マイナーリリースとは、インストールされたソフトウェアを古いバージョンから同じ主要バージョン番号 (バージョン番号形式は、*主要.マイナー.ビルド*) に更新するソフトウェアの完全セットです。バグフィックスとその他の変更については、アップグレードのリリースノートにリストされます。

- **メジャーリリース**　メジャーリリースとは、独自のバージョン番号を持つソフトウェアの完全セットです。

既存のデータベース、サーバ、管理ツールとの互換性

以下についての詳細は、Components by Platforms のページを参照してください。

- データベースとデータベースへのサーバ接続、以前のバージョン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、サーバのサポート

- 以前のバージョン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、サーバ、データベースに接続するための管理ツールのサポート

参照

- 「アンロードユーティリティ (dbunload)」『SQL Anywhere サーバ データベース管理』
- 「複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法」292 ページ

データベースの再構築 (アンロード／再ロード)、データベースのアップグレード、クライアントライブラリの更新を必要とする機能

最新バージョンのデータベースサーバで古いデータベースを実行しても多くの機能を使用できますが、一部の機能にアクセスするには、データベースの再構築 (アンロードと再ロード)、データベースファイルでのアップグレードの実行、クライアントライブラリの更新を行う必要があります。

> **注意**
> 以下のカテゴリに示されない機能には、SQL Anywhere 16.0 データベースサーバのみが必要であり、既存のデータベースのアップグレードやクライアントライブラリの更新は必要ありません。SQL Anywhere 16 の新機能の完全なリストについては、「バージョン 16.0 の新機能」1 ページを参照してください。

再構築 (アンロード／再ロード) されたデータベースとアップグレードされたデータベースの違い

- **アップグレードされたデータベース**　アップグレードされたデータベースとは、機能の一部が更新されたデータベースのことです。たとえば、アップグレードではシステムテーブルとビューが更新され、新しいデータベースオプションが追加され、すべてのシステムストアドプロシージャが再作成され、jConnect サポートがインストールされ、データベースの Java のサポートが変更されます。データベースをアップグレードするには、アップグレードユーティリティ (dbupgrad)、または Sybase Central の**データベースアップグレードウィザード**を使用してください。

- **再構築されたデータベース**　再構築されたデータベースとは、古いデータベースのデータが保存された新しいデータベースのことです。データベースを再構築するには、UNLOAD 文を使用するか、アンロードユーティリティ (dbunload) に -an オプションを指定して実行してください。

データベースの再構築 (アンロード／再ロード) を必要とする機能

- Ultra Light バージョン 16 では、以前のバージョンの Ultra Light で作成したデータベースに接続できません。したがって、アンロード／再ロードが必要です。ただし、Ultra Light バージョン 11 以降のクライアントは、Mobile Link バージョン 16 と同期できます。

- ALTER TABLE 文の ADD COLUMN 句 (バージョン 10 以前のデータベース)「ALTER TABLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- CESU-8 文字セットのサポート。サポートされている照合の詳細については、「代替照合」『SQL Anywhere サーバ データベース管理』を参照してください。

アップグレードのみを必要とする機能 (dbupgrad ユーティリティまたは UPGRADE DATABASE 文)

データベースのアップグレードのみを必要とする機能は、データベースをアンロード／再ロードすれば動作します。

- SMTP と MAPI の新しいエラーコード、変更されたリターンコード。「MAPI と SMTP 用のシステムプロシージャのリターンコード」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- ALTER TABLE 文の ADD COLUMN 句 (バージョン 11 以降のデータベース)「ALTER TABLE 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- CREATE INDEX 文。「CREATE INDEX 文」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- ロールと権限 (新しいロールベースのセキュリティモデルをサポートするための Sybase Central の変更を含む)。「新しいセキュリティモデル：ロールベースのアクセス制御 (RBAC)」1 ページを参照してください。

- 一部のシステムプロシージャはデファイナ (所有者) ではなく、インボーカ (呼び出し元) の権限で実行されるようになりました。「プロシージャとファンクションを設定し所有者または呼び出し側の権限で実行する」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- LDAP ユーザ認証サポート。「LDAP ユーザ認証」『SQL Anywhere サーバ SQL の使用法』を参照してください。

- ROW データ型と ARRAY データ型のサポート。「複合データ型」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

- OData のサポート。「OData のサポート」『SQL Anywhere サーバ プログラミング』を参照してください。

- 予約語：

  - array
  - json
  - row
  - rowtype
  - unnest
  - varray

- ログインポリシーオプション：

  - ldap_primary_server
  - ldap_secondary_server
  - ldap_auto_failback_period
  - ldap_failover_to_std
  - ldap_refresh_dn
  - change_password_dual_control

  「ルートログインポリシー」『SQL Anywhere サーバ データベース管理』を参照してください。

- データベースオプション：

  - 「auto_commit_on_create_local_temp_index オプション」『SQL Anywhere サーバ データベース管理』
  - 「disk_sandbox オプション」『SQL Anywhere サーバ データベース管理』
  - 「-sbx データベースオプション」『SQL Anywhere サーバ データベース管理』
  - 「min_role_admins オプション」『SQL Anywhere サーバ データベース管理』
  - 「db_publisher オプション」『SQL Anywhere サーバ データベース管理』
  - 「login_mode オプション」『SQL Anywhere サーバ データベース管理』
  - 「trusted_certificates_file オプション」『SQL Anywhere サーバ データベース管理』
  - 「-al データベースオプション」『SQL Anywhere サーバ データベース管理』

●データベースサーバオプション：

○「-gu データベースサーバオプション」『SQL Anywhere サーバ データベース管理』
○「-gk データベースサーバオプション」『SQL Anywhere サーバ データベース管理』
○「-gd データベースサーバオプション」『SQL Anywhere サーバ データベース管理』
○「-gl データベースサーバオプション」『SQL Anywhere サーバ データベース管理』
○「-al データベースサーバオプション」『SQL Anywhere サーバ データベース管理』

●接続プロパティ：

○ auto_commit_on_create_local_temp_index 接続プロパティ『SQL Anywhere サーバ データベース管理』
○ extern_login_credentials 接続プロパティ『SQL Anywhere サーバ データベース管理』

●システムビュー：

○「SYSGROUP 互換ビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSGROUPS 互換ビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSLDAPSERVER システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSROLEGRANT システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSROLEGRANTS 統合ビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSROLEGRANTEXT システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSUSERAUTHORITY 互換ビュー (旧式)」『SQL Anywhere サーバ SQL リファレンス』
○「SYSTABLEPERM システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSTABAUTH 統合ビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSCERTIFICATE システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSTABCOL システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSUSER システムビュー」『SQL Anywhere サーバ SQL リファレンス』

●システムテーブル：

○「ISYSCERTIFICATE システムテーブル」『SQL Anywhere サーバ SQL リファレンス』
○「ISYSTABCOL システムテーブル」『SQL Anywhere サーバ SQL リファレンス』
○「ISYSTAB システムテーブル」『SQL Anywhere サーバ SQL リファレンス』
○「ISYSUSER システムテーブル」『SQL Anywhere サーバ SQL リファレンス』

●現在、タイムスタンプはそれに対応する UTC (協定世界時) ベースのタイムスタンプとしてシステムテーブルに格納されるようになりましたが、システムビューは UTC タイムスタンプと

ローカルタイムスタンプの両方を公開します。以下のシステムビューとその基本となるテーブルのカラム数に依存するアプリケーションは更新する必要があります。

○「SYSTAB システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSUSER システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSVIEW システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSOBJECT システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSTEXTIDX システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSEXTERNENVOBJECT システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSJAR システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSJAVACLASS システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSHISTORY システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSREMOTEUSER システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSREMOTEUSERS 統合ビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSCOLSTAT システムビュー」『SQL Anywhere サーバ SQL リファレンス』
○「SYSCOLSTATS 統合ビュー」『SQL Anywhere サーバ SQL リファレンス』

●関数：

○「ARRAY コンストラクタ [複合]」『SQL Anywhere サーバ SQL リファレンス』
○「ROW コンストラクタ [複合]」『SQL Anywhere サーバ SQL リファレンス』
○「ENCRYPT 関数 [文字列]」『SQL Anywhere サーバ SQL リファレンス』
○「DECRYPT 関数 [文字列]」『SQL Anywhere サーバ SQL リファレンス』

●演算子：

○「UNNEST 配列演算子」『SQL Anywhere サーバ SQL リファレンス』

●システムプロシージャ：

○「sa_certificate_info システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sa_cpu_topology システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sa_db_option システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sa_get_ldapserver_status システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sa_get_user_status システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sa_parse_json システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sa_server_option システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_copy_directory システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_copy_file システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_create_directory システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_delete_directory システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_delete_file システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_move_directory システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_move_file システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_create_secure_feature_key システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_alter_secure_feature_key システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_drop_secure_feature_key システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_list_secure_feature_key システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_use_secure_feature_key システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「xp_get_mail_error_code システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「xp_get_mail_error_text システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「xp_getenv システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「xp_startmail システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「xp_startsmtp システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「xp_sendmail システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「xp_stopmail システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「xp_stopsmtp システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_objectpermission システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_displayroles システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_has_role システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_proc_priv システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_auth_sys_role_info システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sp_sys_priv_role_info システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sa_get_ldapserver_status システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』
○「sa_get_user_status システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』

- 新規または強化された SQL 文
  - 「GRANT 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「REVOKE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE ROLE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「ALTER ROLE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP ROLE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「GRANT ROLE SYS_REPLICATION_ADMIN_ROLE 文 [Mobile Link] [SQL Remote]」『SQL Anywhere サーバ SQL リファレンス』
  - 「GRANT ROLE SYS_RUN_REPLICATION_ROLE 文 [Mobile Link] [SQL Remote]」『SQL Anywhere サーバ SQL リファレンス』
  - 「GRANT PUBLISH 文 [SQL Remote]」『SQL Anywhere サーバ SQL リファレンス』
  - 「REVOKE PUBLISH 文 [SQL Remote]」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE LDAP SERVER 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「ALTER LDAP SERVER 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP LDAP SERVER 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「VALIDATE LDAP SERVER 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「ALTER USER 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE LOGIN POLICY 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「ALTER LOGIN POLICY 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「COMMENT 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「SET MIRROR OPTION 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE CERTIFICATE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP CERTIFICATE 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE FUNCTION 文 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE PROCEDURE 文 [Web サービス]」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE TEXT INDEX 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「CREATE TEMPORARY TRACE EVENT SESSION 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「ALTER TRACE EVENT SESSION 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP TRACE EVENT 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「DROP TRACE EVENT SESSION 文」『SQL Anywhere サーバ SQL リファレンス』
  - 「NOTIFY TRACE EVENT 文」『SQL Anywhere サーバ SQL リファレンス』

参照

- 「バージョン 16.0 の新機能」1 ページ

SQL Anywhere サーバのアップグレード

このバージョンのソフトウェアで既存のアプリケーションを使用する前に、動作の変更のリストを確認して、アプリケーションに影響がないかどうかを確認してください。SQL Anywhere 16 変更点とアップグレード 1 ページを参照してください。

バージョン 10 以降のデータベースのアップグレード

バージョン 10 以降からアップグレードするには、データベースをアップグレードするか再構築 (またはその両方) します。バージョン 16 のソフトウェアでバージョン 10 以降のデータベース

を使用できるので、アップグレードや再構築を実行するかどうかは任意です。ただし、バージョン 16 の新機能をすべて利用するには、データベースを再構築する必要があります。「バージョン 10 以降のデータベースのアップグレード手順」304 ページと「バージョン 10 以降のデータベースの再構築手順」294 ページを参照してください。

注意
データベースサーバをアップグレードした後、またはアップグレード後のデータベースサーバで使用できるようにデータベースを再構築またはアップグレードした後は、データベース内のマテリアライズドビューを再表示することをおすすめします。「マテリアライズドビューの手動再表示」『SQL Anywhere サーバ SQL の使用法』を参照してください。

バージョン 9 以前のデータベースのアップグレード

バージョン 9 以前からバージョン 16 にアップグレードするには、データベースを再構築します。再構築するには、古いデータベースをアンロードしてから、バージョン 16 の新しいデータベースに再ロードします。バージョン 9 以前のデータベースを開始しようとすると、データベースの開始時にエラーが発生します。既存のデータベースを再構築するには、いくつかの方法があります。

- -an (データベースの新規作成) オプションまたは -ar (古いデータベースの置き換え) オプションを指定して、バージョン 16 のアンロードユーティリティ (dbunload) を実行します。アップロードユーティリティを使用したバージョン 9 以前のデータベースの再構築 (コマンドライン)303 ページを参照してください。

  注意
  アンロード ユーティリティ (dbunload) には、SQL Anywhere の全バージョンで同じファイル名が使用されています。正しいバージョンを使用していることを確認してください。使用しているアンロードユーティリティのバージョンを確認するには、dbunload -? コマンドを実行します。「複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法」292 ページを参照してください。

- バージョン 16 のアンロードユーティリティを使用してデータベースをアンロードしてから、バージョン 16 のデータベースサーバにある *reload.sql* ファイルを使用してデータベースを再ロードします。

  スキーマを変更する必要がある場合は、この方法を使用してアップグレードすることをおすすめします。スキーマの変更後は、データベースを作成して、再ロードスクリプトを適用できます。

- Sybase Central の**データベースアンロードウィザード**を使用します。新しいデータベースを作成するか、既存のデータベースを新しいデータベースに置き換えるか、データベースをファイルにアンロードします。バージョン 9 以前のデータベースの再構築 (Sybase Central)301 ページを参照してください。

- 旧バージョンの dbunload を使用してデータベースをアンロードし、*reload.sql* ファイルとバージョン 16 のデータベースサーバを使用してデータベースを再ロードします。廃止予定またはサポート対象外のデータベースオプション設定、オブジェクト、または SQL 構文が *reload.sql* ファイルにアンロードされる可能性があるため、この方法は他の方法が失敗する場合にのみ使

用してください。再ロード中に問題が発生したら、ファイルを手動で編集してください。バージョン 16 では、内部再ロード機能により、このような問題の多くが解決されています。

注意
Mac OS X 用の SQL Anywhere 9.0.2 は PPC でサポートされ、SQL Anywhere 10 以降は Intel でサポートされていました。Mac OS X に 9.0.2 以前のバージョンのデータベースがある場合は、次の 2 つの方法でデータベースをアンロードできます。

- バージョン 9.0.2 のソフトウェアを使用してデータベースをアンロードします。
- SQL Anywhere 16 がインストールされている別のプラットフォームにデータベースをコピーしてから、バージョン 16 のソフトウェアを使用してデータベースをアンロードします。

データベースをアンロードした後は、Mac OS X でバージョン 16 のソフトウェアを使用してデータベースを再ロードできます。

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。「データベースの再構築」『SQL Anywhere サーバ SQL の使用法』を参照してください。

アップロードと再ロードの注意事項

SQL Anywhere をアップグレードする前に、次の作業を行ってください。

- **動作の変更のチェック**　記述されている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ください。影響する場合は、既存のアプリケーションを更新してください。「バージョン 16.0 の新機能」1 ページを参照してください。

- **アプリケーションのテスト**　事前に SQL Anywhere 16 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。

- **正しいバージョンのユーティリティの使用**　新しいデータベースには、正しいバージョンのデータベースユーティリティを使用してください。「複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法」292 ページを参照してください。

- **データベースの検証とバックアップ**　アップグレードを開始する前に、データベースを検証し、ソフトウェアとデータベースをバックアップします。後でリカバリできるように、アップグレードの終了時にデータベースをバックアップします。

- **アップグレードの前に同期**　Ultra Light データベースや、Mobile Link インストール環境の SQL 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の場合は、同期を正しく行ってからアップグレードしてください。

- **アップグレード手順のテスト**　アップグレード手順は、十分にテストしてから運用システムで実行してください。

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があるわけではありません。

複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法

SAP Sybase SQL Anywhere の複数のバージョンが同じ Windows コンピュータ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。インストールでは、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が実行されている場合があります。

たとえば、パス内でバージョン 8 の実行プログラムのディレクトリがバージョン 16 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 8 のユーティリティを使用することになり、その結果、バージョン 8 のデータベースが作成されてしまいます。

バージョン 16 のユーティリティが確実に使用されるようにするには、いくつかの方法があります。

- バージョン 16 の実行プログラムディレクトリが、旧バージョン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。
- 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 16 の実行プログラムディレクトリに変更します。
- 実行するユーティリティの正確な場所を示すユーティリティ名への完全修飾パスを指定します。
- 正しいバージョンのユーティリティが使用されるように、環境を変更するスクリプトを作成します。
- 旧バージョンのソフトウェアをアンインストールします。

データベースの再構築 (アンロード／再ロード)

旧バージョンのユーザは、次の手順に従って、データベースをバージョン 16 に再構築できます。

前提条件

バージョン 16 のデータベースでは、次のシステム権限が必要です。

- BACKUP DATABASE

●VALIDATE ANY OBJECT

●SERVER OPERATOR

●SELECT ANY TABLE

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。「アップロードと再ロードの注意事項」291 ページを参照してください。

データベースをバックアップします。次に例を示します。

```
dbbackup -c "DBF=mydb.db;UID=DBA;PWD=sql" old-db-backup-dir
```

♦ データベースの再構築 (コマンドライン)

1. 断片化されているドライブでは、データベースのパフォーマンスが低下するので、可能な場合は、新しいデータベースを保存するドライブの断片化を解除してください。
2. バージョン 16 の dbunload ユーティリティは、旧バージョンのデータベースサーバで実行されているデータベースに対して使用できないので、すべての SQL Anywhere データベースサーバを停止します。次に例を示します。

   ```
   dbstop -c "DBF=mydb.db;UID=DBA;PWD=sql"
   ```

3. 既存のデータベースをアンロードして、新しいバージョン 16 のデータベースに再ロード (再構築) します。次に例を示します。

   ```
   dbunload -c "DBF=mydb.db;UID=DBA;PWD=sql" -an mydb16.db
   ```

4. 新しいデータベースは、バックアップしてから使用します。次に例を示します。

   ```
   dbbackup -c "DBF=mydb16.db;UID=DBA;PWD=sql" new-db-backup-dir
   ```

5. 新しいデータベースは、検証してから使用します。次に例を示します。

   ```
   dbvalid -c "DBF=mydb16.db;UID=DBA;PWD=sql"
   ```

結果

データベースが最新バージョンに再構築されます。デフォルトでは、データベースが停止されてから再起動されます。

次の手順

アプリケーションを使用して、再構築されたデータベースをテストできます。

参照

- アップロードユーティリティを使用したバージョン 9 以前のデータベースの再構築 (コマンドライン)303 ページ
- バージョン 9 以前のデータベースの再構築 (Sybase Central)301 ページ
- 「認証データベースのアップグレード」『SQL Anywhere サーバ データベース管理』

バージョン 10 以降のデータベースの再構築手順

データベースを再構築するには、データベースをアンロードしてから再ロードすることでファイルフォーマットをアップグレードします。ファイルフォーマットをアップグレードすると、ディスク上でデータの保管とアクセスに使用されるフォーマットが変更され、ソフトウェアの最新バージョンの新機能とパフォーマンス向上をすべて利用できます。

警告
大規模なデータベースのアンロードや再ロードには時間がかかり、大容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 倍のディスク領域を要する場合があります。

SQL Anywhere のインデックスが変更された結果、データベースをアンロードしてから再ロードして再構築すると、再構築されたデータベースが元のデータベースよりも小さくなる場合があります。このデータベースのサイズの縮小は、問題や、データが失われたことを示すものではありません。

注意
データベースは、再構築の前にバックアップすることをおすすめします。

AUTOINCREMENT カラムのあるテーブルの再ロード

dbunload の -l オプションを指定すると、再構築されたデータベースの AUTOINCREMENT カラムに次に使用可能な値を保持できます。このオプションを指定すると、AUTOINCREMENT 値が含まれるテーブルごとに、生成される *reload.sql* スクリプトに sa_reset_identity システムプロシージャへの呼び出しが追加され、SYSTABCOL.max_identity の現在の値が維持されます。

データベースの再構築 (Sybase Central)

データベースアンロードウィザードを使用して、バージョン 10 以降の SQL Anywhere データベースを最新バージョンにアップグレードできます。

前提条件

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。「アップロードと再ロードの注意事項」291 ページを参照してください。

SELECT ANY TABLE および SERVER OPERATOR システム権限が必要です。

データベースのアンロードと再ロードを行う前に、データベースをバックアップしてください。次に例を示します。

```
dbbackup -c "DBF=mydb.db;UID=DBA;PWD=sql" old-db-backup-dir
```

♦ データベースの再構築 (Sybase Central)

1. **[スタート]** » **[プログラム]** » **[SQL Anywhere 16]** » **[管理ツール]** » **[Sybase Central]** をクリックします。

2. アップグレードするデータベースを実行するためにバージョン 16 のデータベースサーバを起動してから、Sybase Central からデータベースに接続します。

3. **[ツール] » [SQL Anywhere 16] » [データベースのアンロード]** をクリックします。

4. **データベースアンロードウィザード**の最初のページの説明を読んで、**[次へ]** をクリックします。

5. **[現在のバージョンのサーバで実行中のデータベースをアンロードする]** をクリックし、リストからデータベースを選択します。**[次へ]** をクリックします。

6. [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。

7. データベースの新しいファイル名を指定します。

8. 新しいデータベースのページサイズを指定することもできますが、指定するページサイズは、データベースサーバのページサイズ以下である必要があります。デフォルトのページサイズは 4096 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。強力な暗号化を選択した場合は、データベースを起動するたびに暗号化キーが必要です。**[次へ]** をクリックします。

9. **[構造とデータをアンロード]** をクリックします。必要に応じて、他のオプションも選択します。**[次へ]** をクリックします。

10. **[すべてのデータベースオブジェクトをアンロード]** をクリックします。**[次へ]** をクリックします。

11. 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。

12. **[完了]** をクリックすると、処理が開始します。

結果

データベースが最新バージョンにアップグレードされます。デフォルトでは、データベースが停止されてから再起動されます。

次の手順

新しいデータベースを検査し、再構築が正常に完了したことを確認すると同時に、アップグレードされたデータベースをアプリケーションでテストします。

参照

- 「データベースのバックアップ」『SQL Anywhere サーバ データベース管理』
- 「データベースアンロードウィザードを使用したデータのエクスポートに関するヒント」『SQL Anywhere サーバ SQL の使用法』
- 「データベースの再構築」『SQL Anywhere サーバ SQL の使用法』
- 「データベースの暗号化と復号化」『SQL Anywhere サーバ データベース管理』

データベースの再構築 (コマンドライン)

dbunload ユーティリティを使用して、バージョン 10 以降の SQL Anywhere データベースを最新バージョンにアップグレードできます。

前提条件

connection-string で指定されたデータベースユーザは、SELECT ANY TABLE および SERVER OPERATOR システム権限を持っている必要があります。

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。「アップロードと再ロードの注意事項」291 ページを参照してください。

アップグレードするデータベースに排他的にアクセスできること、またシステムパスで、バージョン 16 のユーティリティのパスがその他のユーティリティのパスより前に指定されていることを確認します。「複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法」292 ページを参照してください。

データベースのアンロードと再ロードを行う前に、データベースをバックアップしてください。次に例を示します。

```
dbbackup -c "DBF=mydb.db;UID=DBA;PWD=sql" old-db-backup-dir
```

内容と備考

バージョン 10 以降のデータベースで dbunload を使用する場合は、使用する dbunload のバージョンが、データベースへのアクセスに使用するデータベースサーバのバージョンと一致する必要があります。dbunload のバージョンがデータベースサーバのバージョンより古いまたは新しい場合は、エラーが返されます。

dbunload ユーティリティを使用して、Mobile Link インストール環境のリモートデータベース、または SQL Remote レプリケーションに使用するデータベースを再構築する場合は、-ar オプションまたは -an オプションを必ず使用してください。これらのオプションによって、新しいデータベースのトランザクションログのオフセットが、古いデータベースと同じになるように設定されます。

♦ データベースの再構築 (コマンドライン)

1. アンロードユーティリティ (dbunload) を実行し、-an オプションを使用してデータベースを作成します。

   ```
   dbunload -c "connection-string" -an new-db-file
   ```

   このコマンドにより、データベースが作成されます。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換えるには、-an オプションの代わりに -ar オプションを指定します。-ar オプションを使用するには、パーソナルデータベースサーバに接続するか、アンロードユーティリティ (dbunload) と同じコンピュータ上にあるネットワークデータベースサーバに接続します。

   アンロードユーティリティ (dbunload) のその他のオプションについては、「アンロードユーティリティ (dbunload)」『SQL Anywhere サーバ データベース管理』を参照してください。

2. 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。「データベースの再構築」『SQL Anywhere サーバ SQL の使用法』を参照してください。

結果

データベースが最新バージョンにアップグレードされます。デフォルトでは、データベースが停止されてから再起動されます。

次の手順

新しいデータベースを検査し、再構築が正常に完了したことを確認すると同時に、アップグレードされたデータベースをアプリケーションでテストします。

参照

- 「データベースのバックアップ」『SQL Anywhere サーバ データベース管理』
- 「アンロードユーティリティ (dbunload)」『SQL Anywhere サーバ データベース管理』
- 「データベースの再構築」『SQL Anywhere サーバ SQL の使用法』

バージョン 9 以前のデータベースの再構築手順

再構築の前に、データベースをバックアップすることをおすすめします。

> **注意**
> データベースのページサイズには、2048、4096、8192、16384、32768 バイトのいずれかを指定できます。デフォルトは元のデータベースのページサイズです。

Windows Mobile データベースのアップグレードについては、「Windows Mobile でのデータベースの再構築」『SQL Anywhere サーバ データベース管理』を参照してください。

> **注意**
> Mac OS X 用の SQL Anywhere 9.0.2 は PPC でサポートされ、SQL Anywhere 10.0.0 以降は Intel でサポートされていました。Mac OS X に 9.0.2 以前のバージョンのデータベースがある場合は、次の 2 つの方法でデータベースをアンロードできます。
>
> ●バージョン 9.0.2 のソフトウェアを使用してデータベースをアンロードします。
>
> ●SQL Anywhere 16 がインストールされている別のプラットフォームにデータベースをコピーしてから、バージョン 16 のソフトウェアを使用してデータベースをアンロードします。
>
> データベースをアンロードした後は、Mac OS X でバージョン 16 のソフトウェアを使用してデータベースを再ロードできます。

警告
大規模なデータベースのアンロードや再ロードには時間がかかり、大容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 2 倍のディスク領域にアクセスする必要があります。

アップグレードの制限

バージョン 16 のツールを使用してバージョン 9 以前のデータベースを再構築するときには、次の制限事項があります。

- 旧バージョンの全データベースサーバのデータベースを切断して、コンピュータで実行している旧バージョンのデータベースサーバを停止する必要があります。また、コンピュータで実行しているバージョン 16 のデータベースサーバも停止する必要があります。dbunload によってこれらの状態が検出され、処理を続行できない場合は、エラーが発生し、再構築に失敗します。

- 再構築前のデータベースの dbunload 接続文字列に、ENG、START、または LINKS 接続パラメータを含めないでください (-c オプションで指定)。これらの接続パラメータを指定すると、無視され、警告が表示されます。Sybase Central の **[接続]** ウィンドウの **[サーバ名]** フィールドまたは **[開始行]** フィールドには、値を入力しないでください。

- 既存のデータベースへダイレクトにファイルシステムへのアクセスを使用して、コンピュータで dbunload を実行します (dbunload が、共有メモリを使用してデータベースに接続できる状態である必要があります)。

- 再構築を実行するコンピュータで、dbunload_support_engine という名前のデータベースサーバを実行しないでください。

- バージョン 9 以前のデータベースは、データベースファイルのリカバリが必要となった場合に、アンロードしないでください。リカバリが必要なデータベースファイルでアンロードユーティリティ (dbunload) を実行すると、データベースを起動できないことを示すメッセージが返されます。dbunload を再試行する前に、バージョン 9 のデータベースサーバを使用して、データベースを起動してから停止します。バージョン 9 以降のデータベースをアンロードするには、データベースを読み込み専用モードで開始できるようにしておく必要があります。「-r データベースオプション」『SQL Anywhere サーバ データベース管理』を参照してください。

特別な注意事項

- **パスワードの大文字と小文字の区別**　新しく作成された SQL Anywhere 16 データベースでは、データベースでの設定にかかわらず、すべてのパスワードは大文字と小文字が区別されます。新しいデータベースのデフォルトの DBA パスワードは、**sql** です。

  既存のデータベースを再構築する場合、SQL Anywhere でのパスワードの大文字と小文字の区別は、次のように決まります。

  - パスワードを最初に入力したのが大文字と小文字を区別しないデータベースだった場合、そのパスワードの大文字と小文字は区別されません。

  - パスワードを最初に入力したのが大文字と小文字を区別するデータベースだった場合、大文字のパスワードと、大文字と小文字が混在したパスワードでは、大文字と小文字が区別

されます。ただし、パスワードをすべて小文字で入力した場合、パスワードの大文字と小文字は区別されません。

○既存のパスワードと新しいパスワードの両方に加えられた変更は、大文字と小文字が区別されます。

- **ページサイズ**　SQL Anywhere 16 データベースのデフォルトのデータベースページサイズは 4096 バイトです。バージョン 16 でサポートされるページサイズは、2048、4096、8192、16384、32768 バイトです。既存のデータベースで、サポートされていないページサイズが使用されている場合は、新しいデータベースのページサイズはデフォルトで 4096 バイトに設定されます。dbinit の -p オプションまたは dbunload の -ap オプションを使用して、別のページサイズを指定できます。「初期化ユーティリティ (dbinit)」『SQL Anywhere サーバ データベース管理』と「アンロードユーティリティ (dbunload)」『SQL Anywhere サーバ データベース管理』を参照してください。

- **照合**　バージョン 9 以前の SQL Anywhere では、CHAR データ型で使用される 1 つの照合しかサポートしていませんでした。この照合は、SQL Anywhere 照合アルゴリズム (SACA) を使用していました。バージョン 10 以降の SQL Anywhere では、照合アルゴリズムとして、SACA と UCA (Unicode 照合アルゴリズム) の 2 つをサポートしています。再構築されたデータベース用に新しい照合や別の照合を指定していない場合は、既存のデータベースの SACA 照合がアンロードされ、再構築されたデータベースで再利用されます。

  カスタム照合を使用してデータベースを再構築する場合に 1 ステップで再構築すると、その照合は保持されます (内部アンロード)。データベースをアンロードしてから、作成したデータベースにスキーマとデータをロードする場合は、提供されるいずれかの照合を使用する必要があります。「代替照合」『SQL Anywhere サーバ データベース管理』を参照してください。

- **データベースのファイルサイズ**　SQL Anywhere のインデックスが変更された結果、データベースをアンロードしてから再ロードして再構築すると、再構築されたデータベースが元のデータベースよりも小さくなる場合があります。このデータベースのサイズの縮小は、問題や、データが失われたことを示すものではありません。

既知の問題

dbunload または**データベースアンロードウィザード**を実行したときに再構築処理に失敗した場合は、次の手順に従って失敗の原因を診断できます。

1. 新しい、バージョン 16 の空のデータベースを作成します。

   ```
   dbinit -dba DBA,sql test.db
   ```

2. *reload.sql* ファイルを空のデータベースに適用します。

   ```
   dbisql -c "DBF=test.db;UID=DBA;pwd=sql" reload.sql
   ```

3. *reload.sql* ファイルを新しいデータベースに適用したときに受信したメッセージに基づいて、元のデータベースまたは *reload.sql* ファイルを変更します。

次の表に、再構築の失敗につながる既知の問題とその解決策を示します。

既知の問題	解決策
テーブル名のプレフィクスに所有者名が付いている場合は、プロシージャまたはトリガ内の DECLARE LOCAL TEMPORARY TABLE 文により、構文エラーが発生します。	所有者名を削除します。
CREATE TRIGGER 文で、トリガが定義されたテーブルの所有者名が指定されておらず、ユーザが *reload.sql* ファイルを実行して参照するときに、テーブルが所有者名で修飾されている必要がある場合、CREATE TRIGGER 文は失敗し、テーブルが見つからなかったことを示すエラーが返されます。	テーブル名のプレフィクスに所有者名を付けます。

オブジェクト名 (テーブル、カラム、変数、パラメータ名など) が SQL Anywhere の以後のバージョンで導入された予約語 (NCHAR など) に対応している場合は、再ロードに失敗します。次に例を示します。
 	
CREATE PROCEDURE p( )
BEGIN
DECLARE NCHAR INT;
SET NCHAR = 1;
END;
	予約語へのすべての参照を、別の名前を使用するように変更します。変数名の場合は、名前の競合を回避する @ を名前のプレフィクスとして使用するのが、一般的な規則です。
	
	予約語の完全なリストについては、「予約語」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

データベースをバージョン 9 以前の dbunload でアンロードする場合、*reload.sql* ファイルに ml_add_property システムプロシージャの呼び出しが含まれることがありますが、新しいバージョン 16 データベースにはこのプロシージャは存在しません。	バージョン 16 の dbunload ユーティリティを使用してデータベースをアンロードしてください。 適切なバージョンのデータベースユーティリティを使用していることを確認する方法については、「複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法」292 ページを参照してください。
バージョン 9 以前の dbunload を使用してデータベースをアンロードする場合、Transact-SQL 外部ジョイン (*= または =* で指定) を使用するビューを再ロードすると、それらのビューが適切に作成されないことがあります。	次のセクション行を再ロードスクリプトに追加します。 SET TEMPORARY OPTION tsql_outer_joins='on'; Transact-SQL 外部ジョインを使用しているすべてのビューを、後で書き直してください。

既知の問題	解決策
CREATE PROCEDURE 文と ALTER PROCEDURE 文で [NOT] DETERMINISTIC 句がサポートされていません。この句を指定すると、再ロードに失敗し構文エラーが返されます。	[NOT] DETERMINISTIC 句が指定されたユーザ定義プロシージャを含むデータベースをアップグレードする場合、句を削除してからデータベースのアンロードと再ロードを行う必要があります。

バージョン 9 以前のデータベースの再構築 (Sybase Central)

データベースアンロードウィザードでは、再ロードファイルとデータファイルにアンロードするか、アンロードと新しいデータベースへの再ロードを行うか、またはアンロードと既存のデータベースへの再ロードを行うことにより、バージョン 9 以前のデータベースを再構築します。再構築されたデータベースは、最新ソフトウェアバージョンでのすべての新機能とパフォーマンス拡張をサポートします。

前提条件

SELECT ANY TABLE および SERVER OPERATOR システム権限が必要です。

再構築の前に、データベースをバックアップすることをおすすめします。

データベースファイルは、SQL Anywhere 16 がインストールされているコンピュータと同じコンピュータに保存されている必要があります。

データベースからはテーブルのサブセットはアンロードできません。dbunload ユーティリティを使用してアンロードしてください。

データベースアンロードウィザードで、データベースファイルがすでに実行されていると認識された場合は、アンロードを続行する前にデータベースが停止します。

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。「アップロードと再ロードの注意事項」291 ページを参照してください。

断片化されているドライブでは、データベースのパフォーマンスが低下するので、可能な場合は、新しいデータベースを保存するドライブの断片化を解除してください。

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。

♦ バージョン 9 以前のデータベースの再構築 (Sybase Central)

1. **[スタート]** » **[プログラム]** » **[SQL Anywhere 16]** » **[管理ツール]** » **[Sybase Central]** をクリックします。
2. **[ツール]** » **[SQL Anywhere 16]** » **[データベースのアンロード]** をクリックします。
3. **データベースアンロードウィザード**の概要ページを読んで、**[次へ]** をクリックします。

4. **[旧バージョンのサーバで実行中のデータベース、または実行中ではないデータベースをアンロードする]** をクリックします。データベースの接続情報を指定します。**[次へ]** をクリックします。

5. **[新しいデータベースへのアンロードと再ロード]** をクリックします。**[次へ]** をクリックします。

6. 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。バージョン 16 では、デフォルトのページサイズ (推奨値) は 4096 バイトです。

   データベースファイルを暗号化できます。強力な暗号化を選択した場合は、データベースを起動するたびに暗号化キーが必要です。「データベースの暗号化と復号化」『SQL Anywhere サーバ データベース管理』を参照してください。

7. [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。

8. 再構築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。

9. **[完了]** をクリックします。新しいデータベースを調べて、再構築が正常に完了したことを確認します。

結果

データベースが最新バージョンにアップグレードされます。デフォルトでは、データベースが停止されてから再起動されます。

次の手順

新しいデータベースを検査し、再構築が正常に完了したことを確認すると同時に、再構築されたデータベースをアプリケーションでテストします。

参照

- 「データベースのバックアップ」『SQL Anywhere サーバ データベース管理』
- 「アンロードユーティリティ (dbunload)」『SQL Anywhere サーバ データベース管理』
- 「データベースの再構築」『SQL Anywhere サーバ SQL の使用法』

アップロードユーティリティを使用したバージョン 9 以前のデータベースの再構築 (コマンドライン)

アンロードユーティリティ (dbunload) の -an または -ar オプションを使用して、バージョン 9 以前のデータベースを再構築できます。再構築されたデータベースは、最新ソフトウェアバージョンでのすべての新機能とパフォーマンス拡張をサポートします。

前提条件

connection-string で指定されたデータベースユーザは、SELECT ANY TABLE および SERVER OPERATOR システム権限を持っている必要があります。

再構築の前に、データベースをバックアップすることをおすすめします。

データベースファイルは、SQL Anywhere 16 がインストールされているコンピュータと同じコンピュータに保存されている必要があります。

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。「アップロードと再ロードの注意事項」291 ページを参照してください。

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。

システムパスで、他のユーティリティより前にバージョン 16 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。「複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法」292 ページを参照してください。

内容と備考

-an オプションを使用すると、元のデータベースをそのまま残して新規データベースが作成されるので、このオプションを使用することをおすすめします。-ar オプションを使用すると、既存のデータベースが新しいバージョン 16 のデータベースに置き換わります。

♦ アップロードユーティリティを使用したバージョン 9 以前のデータベースの再構築 (コマンドライン)

1. バージョン 16 の dbunload ユーティリティは、旧バージョンのデータベースサーバで実行されているデータベースに対して使用できないので、すべての SQL Anywhere と Adaptive Server Anywhere のデータベースサーバを停止します。次に例を示します。

   ```
   dbstop -c "DBF=mydb.db;UID=DBA;PWD=sql"
   ```

2. 断片化されているドライブでは、データベースのパフォーマンスが低下するので、可能な場合は、新しいデータベースを保存するドライブの断片化を解除してください。

3. データベースをバックアップします。次に例を示します。

   ```
   dbbackup -c "DBF=mydb.db;UID=DBA;PWD=sql" old-db-backup-dir
   ```

   BACKUP DATABASE システム権限が必要です。「データベースのバックアップ」『SQL Anywhere サーバ データベース管理』を参照してください。

4. -an または -ar オプションを指定してアンロードユーティリティ (dbunload) を実行して、新しいデータベースを作成します。

   ```
   dbunload -c "connection-string" -an database-filename
   ```

   次に例を示します。

   ```
   dbunload -c "DBF=mydb.db;UID=DBA;PWD=sql" -an mydb16.db
   ```

   このコマンドを使用すると、データベースが作成されます (-an を指定した場合)。-ar オプションを指定すると、既存のデータベースは再構築後のデータベースに置き換わります。-ar オプションを使用するには、パーソナルデータベースサーバに接続するか、アンロードユー

ティリティ (dbunload) と同じコンピュータ上にあるネットワークデータベースサーバに接続します。

結果

データベースが最新バージョンにアップグレードされます。デフォルトでは、データベースが停止されてから再起動されます。

次の手順

新しいデータベースを検査し、再構築が正常に完了したことを確認すると同時に、再構築されたデータベースをアプリケーションでテストします。

参照

- 「アンロードユーティリティ (dbunload)」『SQL Anywhere サーバ データベース管理』

バージョン 10 以降のデータベースのアップグレード手順

データベースをアップグレードすると、バージョン 16 の機能を有効にするために、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。ディスク上でデータが保管、アクセスされるファイルフォーマットは変更されないので、ソフトウェアの最新バージョンの新機能とパフォーマンス向上をすべて利用できるわけではありません。

データベースのファイルフォーマットのアップグレードの詳細については、「バージョン 10 以降のデータベースの再構築手順」294 ページを参照してください。

バージョン 10 以降のデータベースのアップグレード (Sybase Central)

データベースアップグレードウィザードを使用してシステムテーブル、システムプロシージャ、データベースオプションを追加、変更することにより、バージョン 10 以降のデータベースをアップグレードし、バージョン 16 の機能を有効にすることができます。

前提条件

次のシステム権限が必要です。

- ALTER DATABASE ...
- BACKUP DATABASE
- SERVER OPERATOR

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。「アップロードと再ロードの注意事項」291 ページを参照してください。

警告
データベースファイルをバックアップしてからアップグレードしてください。既存のファイルにアップグレードを適用した場合、アップグレードに失敗すると、これらのファイルは使用できなくなります。データベースをバックアップするには、BACKUP DATABASE システム権限が必要です。データベースのバックアップの詳細については、「バックアップとデータリカバリ」『SQL Anywhere サーバ データベース管理』を参照してください。

内容と備考

データベースアップグレードウィザードでは、バージョン 9.0.2 以前のデータベースをバージョン 16 にアップグレードすることはできません。バージョン 9.0.2 以前の既存のデータベースをバージョン 16 にアップグレードするには、dbunload または**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をアンロードし、再ロードする必要があります。 「バージョン 9 以前のデータベースのアップグレード」290 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 (Sybase Central)

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。「アップロードと再ロードの注意事項」291 ページを参照してください。

2. **[スタート]** » **[プログラム]** » **[SQL Anywhere 16]** » **[管理ツール]** » **[Sybase Central]** をクリックします。

3. **[SQL Anywhere 16]** のプラグインから、アップグレードするデータベースに接続します。データベースはバージョン 16 のデータベースサーバで実行されている必要があります。

4. **[ツール]** » **[SQL Anywhere 16]** » **[データベースのアップグレード]** をクリックします。

5. **データベースアップグレードウィザード**の指示に従います。デフォルトでは、データベースが停止されてから再起動されます。

6. (省略可) 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コピーしてアーカイブすることをウィザードで選択しなかった場合は、その操作を行います。

ヒント
データベースアップグレードウィザードは、次の方法でもアクセスできます。

●データベースを右クリックし、**[データベースのアップグレード]** をクリックします。
●データベースを選択し、**[ファイル]** » **[データベースのアップグレード]** をクリックします。

結果

データベースがアップグレードされ、新しい、および修正されたシステムテーブル、システムプロシージャ、データベースオプションを含むようになります。

次の手順

新しいデータベースを検査し、アップグレードが正常に完了したことを確認すると同時に、アップグレードされたデータベースをアプリケーションでテストします。

参照

- 「アップグレードユーティリティ (dbupgrad)」『SQL Anywhere サーバ データベース管理』

バージョン 10 以降のデータベースのアップグレード (コマンドライン)

dbupgrad ユーティリティを使用してシステムテーブル、システムプロシージャ、データベースオプションを追加、変更することにより、バージョン 10 以降のデータベースをアップグレードし、バージョン 16 の機能を有効にすることができます。

前提条件

connection-string に指定するデータベースユーザには ALTER DATABASE システム権限が必要です。また、このデータベースに他の接続がないことが必要です。

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。「アップロードと再ロードの注意事項」291 ページを参照してください。

アップグレードするデータベースに排他的にアクセスできること、またシステムパスで、バージョン 16 のユーティリティがその他のユーティリティより前に指定されていることを確認します。「複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法」292 ページを参照してください。

警告
データベースファイルをバックアップしてからアップグレードしてください。既存のファイルにアップグレードを適用した場合、アップグレードに失敗すると、これらのファイルは使用できなくなります。データベースのバックアップの詳細については、「バックアップとデータリカバリ」『SQL Anywhere サーバ データベース管理』を参照してください。

♦ データベースのアップグレード (コマンドライン)

1. データベースに対してアップグレードユーティリティ (dbupgrad) を実行します。

   ```
   dbupgrad -c "connection-string"
   ```

   詳細については、「アップグレードユーティリティ (dbupgrad)」『SQL Anywhere サーバ データベース管理』を参照してください。

2. 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圧縮します。

結果

データベースがアップグレードされ、新しい、および修正されたシステムテーブル、システムプロシージャ、データベースオプションを含むようになります。

次の手順

新しいデータベースを検査し、アップグレードが正常に完了したことを確認すると同時に、アップグレードされたデータベースをアプリケーションでテストします。

参照

- 「アップグレードユーティリティ (dbupgrad)」『SQL Anywhere サーバ データベース管理』

バージョン 10 以降のデータベースのアップグレード (SQL)

ALTER DATABASE 文を使用してシステムテーブル、システムプロシージャ、データベースオプションを追加、変更することにより、バージョン 10 以降のデータベースをアップグレードし、バージョン 16 の機能を有効にすることができます。

前提条件

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。「アップロードと再ロードの注意事項」291 ページを参照してください。

データベースの所有者であるか、ALTER DATABASE システム権限を付与されている必要があります。また、このデータベースに他の接続がないことが必要です。

警告
データベースファイルをバックアップしてからアップグレードしてください。既存のファイルにアップグレードを適用した場合、アップグレードに失敗すると、これらのファイルは使用できなくなります。

♦ データベースのアップグレード (SQL)

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。他のどの接続も、データベースを同時に使用できません。

2. ALTER DATABASE 文を実行します。

   たとえば、次の文はデータベースをアップグレードします。

   ```
   ALTER DATABASE UPGRADE;
   ```

3. 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圧縮します。

結果

データベースがアップグレードされ、新しい、および修正されたシステムテーブル、システムプロシージャ、データベースオプションを含むようになります。

次の手順

新しいデータベースを検査し、アップグレードが正常に完了したことを確認すると同時に、アップグレードされたデータベースをアプリケーションでテストします。

参照

- 「ALTER DATABASE 文」『SQL Anywhere サーバ SQL リファレンス』
- 「アップロードと再ロードの注意事項」291 ページ
- 「バックアップとデータリカバリ」『SQL Anywhere サーバ データベース管理』

データベースミラーリングシステムでのアップグレードと再構築

データベースミラーリングシステム内のすべてのデータベースサーバで、同じマイナーリリースを使用する必要があります。

データベースの再構築を行わずにデータベースサーバのソフトウェアをミラーリングシステム用のメジャーバージョンにアップグレード

ミラーリングシステムのデータベースサーバのソフトウェアを更新するには、ソフトウェアをインストールし、各データベースサーバを停止し、新しいソフトウェアが動作するサーバ上でデータベースを起動します。データベース自体はアップグレードも再構築も行われません。

前提条件

BACKUP DATABASE システム権限が必要です。データベースの所有者であるか、VALIDATE ANY OBJECT システム権限を持っている必要があります。

デフォルトでは、ネットワークデータベースサーバの停止には SERVER OPERATOR システム権限が必要です。

内容と備考

運用環境で手順を実行する前に、アプリケーションを使用して非運用環境で次の手順をテストします。

データベースミラーリングシステム内のすべてのデータベースサーバで、SQL Anywhere の同じマイナーリリースを使用する必要があります。そのため、更新中、ミラーリングシステムは一時的に停止します。

♦ SQL Anywhere のメジャーバージョンのデータベースミラーリングシステムへの適用

1. プライマリデータベースのバックアップを作成し、そのバックアップをコピーして検証します。

   たとえば、次のコマンドを実行して、*mydb.db* という名前のデータベースをバックアップします。

   ```
   dbbackup -c "DBN=mydb;ENG=myserver;UID=DBA;PWD=sql" backup-dir
   ```

   バックアップのコピーを作成してトランザクションログをバックアップのコピーに適用するには、次のコマンドを実行します。

   ```
   xcopy backup-dir¥mydb.db validatebackup-dir
   xcopy backup-dir¥mydb.log validatebackup-dir
   ```

   バックアップのコピーを検証します。

   ```
   dbvalid -c "DBF=validatebackup-dir¥backupmydb.db;UID=DBA;PWD=sql"
   ```

   バックアップのコピーの検証が失敗した場合は、先に進む前に検証失敗の原因となった問題を修正します。修正を行わないと、データを失うリスクがあります。

2. 新しいソフトウェアをプライマリ、ミラー、および監視サーバにインストールします。

3. ミラーリングシステムが読み込み専用のスケールアウトに含まれる場合、コピーノード上にソフトウェアをインストールします。

4. 次の順番にサーバを停止します。

   a. コピーノード
   b. ミラーサーバ
   c. プライマリサーバ
   d. 監視サーバ

   たとえば、Stop ユーティリティ (dbstop) を実行します。

   ```
   dbstop -y -c "UID=DBA;PWD=sql;Server=myserver"
   ```

5. (省略可) データベースをアップグレードまたは再構築します。

6. 次の順序で新しいサーバ上でデータベースを起動します。

   a. 監視サーバ、プライマリサーバ、ミラーサーバ
   b. コピーノード

結果

ミラーリングシステムのデータベースは、ソフトウェアの新しいバージョンで実行されます。

次の手順

データベースミラーリングシステムを検査し、更新が正常に完了したことを確認すると同時に、データベースミラーリングシステムをアプリケーションでテストします。

参照

- 「ミラーリングシステムのデータベースサーバの停止」『SQL Anywhere サーバ データベース管理』
- 「ユーザが開始したロールの切り替え (フェイルオーバ)」『SQL Anywhere サーバ データベース管理』
- 「サーバ停止ユーティリティ (dbstop)」『SQL Anywhere サーバ データベース管理』

データベースの再構築を行わずにデータベースサーバのソフトウェアをミラーリングシステム用の別のマイナーリリースにアップグレード

ミラーリングシステムのデータベースサーバのソフトウェアを更新するには、ソフトウェアをインストールし、各サーバを停止し、新しいサーバ上でデータベースを起動します。データベースミラーリングシステム内のすべてのデータベースサーバで、同じマイナーリリースを使用する必要があります。

前提条件

BACKUP DATABASE システム権限が必要です。データベースの所有者であるか、VALIDATE ANY OBJECT システム権限を持っている必要があります。

デフォルトでは、ネットワークデータベースサーバの停止には SERVER OPERATOR システム権限が必要です。

運用環境で手順を実行する前に、アプリケーションを使用して非運用環境で次の手順をテストします。

内容と備考

更新中、ミラーリングシステムは一時的に停止します。

♦ SQL Anywhere のメジャーバージョンまたはメンテナンスバージョンのデータベースミラーリングシステムへの適用

1. プライマリデータベースのバックアップを作成し、そのバックアップをコピーして検証します。

   たとえば、次のコマンドを実行して、*mydb.db* という名前のデータベースをバックアップします。

   ```
   dbbackup -c "DBN=mydb;ENG=myserver;UID=DBA;PWD=sql" backup-dir
   ```

   バックアップのコピーを作成します。

```
xcopy backup-dir¥*.db validatebackup-dir¥backupmydb.db
```

バックアップのコピーを検証します。

```
dbvalid -c "DBF=validatebackup-dir¥backupmydb.db;UID=DBA;PWD=sql"
```

バックアップのコピーの検証が失敗した場合は、先に進む前に検証失敗の原因となった問題を修正します。修正を行わないと、データを失うリスクがあります。

2. 次の順番にサーバを停止します。
   a. コピーノード
   b. ミラーサーバ
   c. プライマリサーバ
   d. 監視サーバ

   たとえば、Stop ユーティリティ (dbstop) を実行します。

   ```
   dbstop -y -c "UID=DBA;PWD=sql;Server=myserver"
   ```

3. 新しいソフトウェアをプライマリ、ミラー、および監視サーバにインストールします。
4. 次の順番にサーバを起動します。
   a. 監視サーバ
   b. プライマリサーバ
   c. ミラーサーバ
5. ミラーリングシステムが読み込み専用のスケールアウトに含まれる場合、コピーノード上にソフトウェアをインストールしてから、コピーノードを再起動します。

結果

ミラーリングシステムのデータベースは、ソフトウェアの新しいバージョンで実行されます。

次の手順

データベースミラーリングシステムを検査し、アップグレードが正常に完了したことを確認すると同時に、データベースミラーリングシステムをアプリケーションでテストします。

参照

- 「ミラーリングシステムのデータベースサーバの停止」『SQL Anywhere サーバ データベース管理』
- 「ユーザが開始したロールの切り替え (フェイルオーバ)」『SQL Anywhere サーバ データベース管理』
- 「サーバ停止ユーティリティ (dbstop)」『SQL Anywhere サーバ データベース管理』

データベースの再構築を行わずにデータベースサーバのソフトウェアをデータベースミラーリングシステム用のサポートパッケージにアップグレード

ミラーリングシステムのデータベースサーバのソフトウェアを更新するには、各サーバを停止し、ソフトウェアをインストールし、新しいサーバ上でデータベースを起動します。

前提条件

BACKUP DATABASE システム権限が必要です。データベースの所有者であるか、VALIDATE ANY OBJECT システム権限を持っている必要があります。

デフォルトでは、ネットワークデータベースサーバの停止には SERVER OPERATOR システム権限が必要です。

内容と備考

システムのサーバは 1 台ずつしか停止できないため、サーバソフトウェアのサポートパッケージの更新中、ミラーリングシステムは動作を続行できます。更新プロセス中、プライマリサーバからミラーサーバへのフェイルオーバが少なくとも 1 回発生します。フェイルオーバ中、プライマリサーバとミラーサーバへの接続は切断されます。

運用環境で手順を実行する前に、アプリケーションを使用して非運用環境で次の手順をテストします。

♦ データベースミラーリングシステムにサポートパッケージを適用します。

1. プライマリデータベースのバックアップを作成し、そのバックアップをコピーして検証します。

   たとえば、次のコマンドを実行して、*mydb.db* という名前のデータベースをバックアップします。

   ```
   dbbackup -c "DBN=mydb;ENG=myserver;UID=DBA;PWD=sql" backup-dir
   ```

   バックアップのコピーを作成します。

   ```
   xcopy backup-dir¥*.db validatebackup-dir¥backupmydb.db
   ```

   バックアップのコピーを検証します。

   ```
   dbvalid -c "DBF=validatebackup-dir¥backupmydb.db;UID=DBA;PWD=sql"
   ```

   バックアップのコピーの検証が失敗した場合は、先に進む前に検証失敗の原因となった問題を修正します。修正を行わないと、データを失うリスクがあります。

2. ミラーリングシステムが読み込み専用のスケールアウトシステムの一部である場合、各コピーノードで以下を行います。

   a. コピーノードサーバを停止します。

b. ソフトウェアをインストールします。

c. コピーノードを起動します。

3. ミラーサーバを停止します。

4. ミラーサーバにソフトウェアをインストールします。

5. ミラーサーバを起動し、同期状態になっていることを確認します。

6. 監視サーバを停止します。

7. 監視サーバにソフトウェアをインストールします。

8. 監視サーバを起動します。

9. プライマリデータベースに接続し、次の文を実行して、フェイルオーバを開始します。

```
ALTER DATABASE SET PARTNER FAILOVER;
```

フェイルオーバ中、プライマリサーバとミラーサーバへの接続は切断されます。現在のプライマリサーバがミラーサーバになります。

10. ミラーサーバを停止します。

11. ミラーサーバにソフトウェアをインストールします。

12. ミラーサーバを起動し、同期状態になっていることを確認します。

結果

ミラーリングシステムのデータベースは、ソフトウェアの新しいバージョンで実行されます。

次の手順

データベースミラーリングシステムを検査し、アップグレードが正常に完了したことを確認すると同時に、データベースミラーリングシステムをアプリケーションでテストします。

参照

- 「ミラーリングシステムのデータベースサーバの停止」『SQL Anywhere サーバ データベース管理』
- 「ユーザが開始したロールの切り替え (フェイルオーバ)」『SQL Anywhere サーバ データベース管理』
- 「サーバ停止ユーティリティ (dbstop)」『SQL Anywhere サーバ データベース管理』

データベースミラーリングシステムでのデータベースのアップグレードまたは再構築 (アンロード／再ロード)

プライマリデータベースをアップグレードし、アップグレードしたデータベースとトランザクションログをミラーデータベースにコピーします。ミラーリングシステムが一時的に停止しま

す。通常、データベースのアップグレードは、ソフトウェアのメジャーバージョンまたはマイナーリリースを更新する場合に実行されます。

前提条件

BACKUP DATABASE システム権限が必要です。データベースの所有者であるか、VALIDATE ANY OBJECT システム権限を持っている必要があります。

デフォルトでは、ネットワークデータベースサーバの起動と停止には SERVER OPERATOR システム権限が必要です。

データベースをアップグレードするには、ALTER DATABASE システム権限が必要です。また、このデータベースに他の接続がないことが必要です。

アンロードユーティリティ (dbunload) を使用してデータベースを再構築 (アンロード／再ロード) するには、SELECT ANY TABLE システム権限が必要です。再ロードを伴うアンロードの場合は、SERVER OPERATOR システム権限も必要です。

運用環境で手順を実行する前に、アプリケーションを使用して非運用環境で次の手順をテストします。

♦ SQL Anywhere のマイナーリリースのデータベースミラーリングシステムへの適用

1. 新しいバージョンのソフトウェアをシステムの各コンピュータにインストールします。データベースミラーリングシステム内のすべてのデータベースサーバで、SQL Anywhere の同じメンテナンスバージョンを使用する必要があります。

2. プライマリデータベースのバックアップを作成し、そのバックアップをコピーして検証します。

   たとえば、次のコマンドを実行して、*mydb.db* という名前のデータベースをバックアップします。

   ```
   dbbackup -c "DBN=mydb;ENG=myserver;UID=DBA;PWD=sql" backup-dir
   ```

   バックアップのコピーを作成します。

   ```
   xcopy backup-dir¥*.db validatebackup-dir¥backupmydb.db
   ```

   バックアップのコピーを検証します。

   ```
   dbvalid -c "DBF=validatebackup-dir¥backupmydb.db;UID=DBA;PWD=sql"
   ```

   バックアップのコピーの検証が失敗した場合は、先に進む前に検証失敗の原因となった問題を修正します。修正を行わないと、データを失うリスクがあります。

3. 次の順番にサーバを停止します。
   a. コピーノード
   b. ミラー
   c. プライマリ

4.

オプション	アクション
データベースのアップグレード	プライマリデータベースに対してアップグレードユーティリティ (dbupgrad) を実行します。次に例を示します。 `dbupgrad -c "UID=DBA;PWD=sql;DBF=C:¥primary-database.db"` データベースがアップグレードされ、新しいトランザクションログが作成され、データベースが停止します。古いトランザクションログは削除できます。
データベースの再構築 (アンロード／再ロード)	`dbunload -c "UID=DBA;PWD=sql;DBF=C:¥primary-database.db"`

5. アップグレードまたは再構築されたデータベースとその新しいトランザクションログをミラーサーバとスケールアウトコピーノードにコピーします。

6. 次の順番にサーバを起動します。
   a. プライマリ
   b. ミラー
   c. コピーノード

結果

ミラーリングシステムのデータベースがアップグレードまたは再構築され、ミラーリングシステムが動作中になります。

次の手順

データベースミラーリングシステムを検査し、アップグレードが正常に完了したことを確認すると同時に、データベースミラーリングシステムをアプリケーションでテストします。

参照

- 「ミラーリングシステムのデータベースサーバの停止」『SQL Anywhere サーバ データベース管理』
- 「ユーザが開始したロールの切り替え (フェイルオーバ)」『SQL Anywhere サーバ データベース管理』
- 「サーバ停止ユーティリティ (dbstop)」『SQL Anywhere サーバ データベース管理』

トラブルシューティング：データベースのアップグレード

この項では、データベースのアップグレード時によく発生する問題について説明します。

トラブルシューティング：サポートパッケージの適用時は JDBC アプリケーションが動作していないことを確認してください

サポートパッケージの適用後、JDBC アプリケーションが動作を停止し、次のようなメッセージを受信する場合があります。

The sajdbc4.jar build does not match the shared object build.

このメッセージは、サポートパッケージの適用時、Interactive SQL または Sybase Central の高速ランチャや独自の JDBC アプリケーションが実行されていると返されることがあります。この場合、Java VM は、ロードしたすべての DLL または共有オブジェクトはロックしていますが JAR ファイルはロックしていません。その結果、サポートパッケージを適用することによって、対応する dbjdbc16 および dbjodbc16 DLL または共有オブジェクトはアップグレードせずに sajdbc4 と jodbc4 の JAR ファイルはアップデートされてしまうことが頻繁に発生します。JDBC アプリケーションが再起動されたとき、JDBC JAR ファイルと対応する DLL または共有オブジェクトのビルドが一致せず、上記のメッセージが返されます。

まず、JDBC ベースのアプリケーションをすべてシャットダウンし、サポートパッケージを再適用してください。サポートパッケージの再適用で効果がない場合、次の方法で問題を解決してください。

- **サポートパッケージのインストーラにより DLL と共有オブジェクトが正しく更新されたことを確認**

  ○Windows の場合、サポートパッケージの適用時に *dbjdbc16.dll* ファイルと *dbjodbc16.dll* ファイルが正しく更新されたことを確認してください。

  ○Linux/Unix の場合、サポートパッケージの適用時に *libdbjdbc16.so.1* および *libdbjodbc16.so.1* 共有オブジェクトが正しく更新されたことを確認してください。

  ○Mac OS X の場合、サポートパッケージの適用時に *libdbjdbc16.dylib* および *libdbjodbc16.dylib* 共有オブジェクトが正しく更新されたことを確認してください。

  注意
  システムには 32 ビットバージョンと 64 ビットバージョンの両方の DLL または共有オブジェクトがインストールできます。DLL と共有オブジェクトの確認の際には、SQL Anywhere サーバのビット設定ではなく、JAVA VM のビット設定と一致するものを確認する必要があります。

- **システムに DLL と共有オブジェクトのコピーが複数ないことを確認**　DLL または共有オブジェクトが正常にアップデートされている場合、同じビット設定の DLL または共有オブジェ

クトのコピーが複数ないことを確認する必要があります。DLL の複数のコピーは、複数のクラスローダに DLL や共有オブジェクトがロードされないようにする Java の制限をバイパスするために DLL または共有オブジェクトを Java VM の拡張フォルダにコピーした場合に発生することがあります。

- **JAR ファイルが正常にアップデートされていることを確認**　DLL または共有オブジェクトが正常にアップデートされ、システムにコピーが重複していなければ、さまざまな JAR ファイルが正常にアップデートされたことを確認します。各 JAR ファイルを確認するには、次のコマンドを実行し、JAR ファイルによってレポートされた SQL Anywhere のバージョンとビルド番号が、インストールしたサポートパッケージの SQL Anywhere のバージョンとビルド番号と一致することを確認します。

  sajdbc4.jar を確認するには、次のコマンドを実行します (*path* を JAR ファイルのパスに置き換えてください)。

  ```
  java -cp path¥sajdbc4.jar sybase.jdbc4.sqlanywhere.IBuildNum
  ```

  jodbc4.jar を確認するには、次のコマンドを実行します (*path* を JAR ファイルのパスに置き換えてください)。

  ```
  java -cp path¥jodbc4.jar ianywhere.ml.jdbcodbc.jdbc4.IBuildNum
  ```

  どの JAR ファイル、DLL、または共有オブジェクトがサポートパッケージのビルド番号と一致しないか判断できたら、アプリケーションでファイルがロックされていないことを確認し、サポートパッケージを再適用します。

トラブルシューティング：集合関数と外部参照

SQL Anywhere は、サブクエリでの集合関数の使用法を明確に規定した SQL/2008 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。以前は有効だ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。

集合関数がサブクエリに使用され、集合関数の参照先カラムが外部参照の場合は、集合関数全体が外部参照として扱われます。集合関数はサブクエリ内ではなく外部ブロック内で計算され、サブクエリ内の定数となります。

サブクエリに含まれる外部参照の集合関数の使用には、次の制限が適用されます。

- 外部参照の集合関数を指定できるのは、SELECT リストまたは HAVING 句にあるサブクエリの中だけです。また、これらの句は、外部ブロックの隣になくてはなりません。
- 外部参照の集合関数に指定できるのは、1 つの外部カラム参照だけです。
- ローカルカラム参照と外部カラム参照を、同じ集合関数に混在させることはできません。

新しい標準に関連する問題は、ローカル参照しか含まないように集合関数を書き換えると回避することができます。たとえば、サブクエリ (SELECT MAX(S.y + R.y) FROM S) がローカルカラム参照 (S.y) と外部カラム参照 (R.y) の両方で構成されていると、不正になります。このサブクエリは (SELECT MAX(S.y) + R.y FROM S) に書き換えることができます。書き換えると、集合関

数はローカルカラム参照だけを持つことになります。外部参照の集合関数が SELECT や HAVING 以外の句に指定されている場合にも、同様の書き換えを使用できます。

例 1

次のクエリの場合、SQL Anywhere 7 以前のバージョンでは有効な結果が生成されました。

```
SELECT Name,
    ( SELECT SUM( p.Quantity )
     FROM SalesOrderItems )
FROM Products p WHERE p.ID = 300;
```

Name	SUM(p.Quantity)
Tee shirt	30,716

SQL Anywhere 8 以降のバージョンでは、同じクエリを実行したときに、関数またはカラムの参照が GROUP BY 句の中になければならないというエラーメッセージが生成されます。SUM(p.Quantity) は外部参照として扱われ、外部クエリブロック内で計算されるため、この文は有効でなくなっています。上記のクエリは次のクエリと同等です。

```
SELECT Name,
    SUM( p.Quantity ) AS Z,
    ( SELECT Z
        FROM SalesOrderItems )
  FROM Products p WHERE p.ID = 300;
```

ただし、集合関数は外部クエリブロック内で計算されるようになったため、外部クエリブロックはグループクエリとして扱われ、カラム名を SELECT リストの GROUP BY 句に記述する必要があります。その意味でもこのクエリは無効で、同じエラーメッセージが生成されます。SQL Anywhere 8 以降のバージョンで SQL Anywhere 7 以前のバージョンの最初のクエリと同じ結果セットを返すには、次のクエリを使用してください。

```
SELECT Name,
p.Quantity * ( SELECT COUNT( * ) FROM SalesOrderItems )
FROM Products p WHERE p.ID = 300;
```

例 2

SQL Anywhere 7 以前のバージョンでは、SUM(p.Quantity) はネストされた SELECT クエリブロックの内部で計算され、p.Quantity のみが外部参照として扱われたため、次のクエリでは 30,716 という結果が生成されました。

```
SELECT ( SELECT FIRST SUM( p.Quantity ) FROM SalesOrderItems ) AS ss
FROM Products p WHERE p.ID = 300;
```

SQL Anywhere 8 以降のバージョンでは、SUM(p.Quantity) は外部参照として扱われ、外部クエリブロックの内部で計算されるため、同じクエリは 28 という結果を返します。つまり、上記のクエリは次のクエリと同等です。

```
SELECT DT.ss
FROM ( SELECT SUM( p.Quantity ) AS asum,
          ( SELECT FIRST asum FROM SalesOrderItems ) AS ss
    FROM Products p WHERE p.ID = 300 ) AS DT;
```

SQL Anywhere 8 以降のバージョンで SQL Anywhere 7 以前のバージョンの最初のクエリと同じ結果を返すには、次のクエリを使用してください。

```
SELECT p.Quantity * ( SELECT COUNT( * ) FROM SalesOrderItems ) AS ss
FROM Products p WHERE p.ID = 300;
```

参照

- 「HAVING 句：データグループの選択」『SQL Anywhere サーバ SQL の使用法』

Mobile Link のアップグレード

既存のソフトウェアとの互換性

- バージョン 16 の Mobile Link クライアントは、バージョン 16.0 より以前の Mobile Link サーバとは互換性がありません。

- バージョン 16 の Mobile Link サーバは、バージョン 11 以降のクライアントで使用できます。これより以前のクライアントをサポートする必要がある場合は、そのバージョンのクライアントをサポートしている古いバージョンの Mobile Link サーバを使用してください。

- 記述されている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ください。影響する場合は、既存のアプリケーションを更新してください。SQL Anywhere 16 変更点とアップグレード 1 ページを参照してください。

アップグレードの順序

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。

   「統合データベースのアップグレード」320 ページを参照してください。

3. Mobile Link サーバをアップグレードします。

   「Mobile Link サーバのアップグレード」328 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。

   バージョン 16 の Mobile Link サーバは、バージョン 11 以降のクライアントでのみ使用できます。バージョン 16 の Mobile Link サーバと一緒に使用するには、バージョン 11 より以前の Mobile Link クライアントはアップグレードする必要があります。

   SQL Anywhere リモートデータベースの詳細については、「SQL Anywhere Mobile Link クライアントのアップグレード」328 ページを参照してください。

6. Ultra Light アプリケーションの詳細については、「Ultra Light のアップグレード」331 ページを参照してください。

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。

参照

- 「Mobile Link の動作の変更」49 ページ
- 「アップロードと再ロードの注意事項」291 ページ

統合データベースのアップグレード

新しい Mobile Link サーバと既存の統合データベースを使用できるようにするには、新しいシステムオブジェクトをインストールするアップグレードスクリプトを実行する必要があります。アップグレードスクリプトは、現在インストールされている Mobile Link システムテーブルの所有者が実行する必要があります。Mobile Link システム設定を更新するには、次の方法を使用することもできます。

- Sybase Central の Mobile Link プラグインで、**[Mobile Link 16]** » **[プロジェクト]** » **[統合データベース]** をクリックし、データベース名を右クリックして、**[Mobile Link システム設定のチェック]** をクリックします。データベースの設定またはアップグレードが必要な場合は、続行のプロンプトが表示されます。

- **同期モデル展開ウィザード**を使用する場合は、統合データベースに接続するとシステム設定がチェックされます。データベースの設定またはアップグレードが必要な場合は、続行のプロンプトが表示されます。「同期モデル」『Mobile Link クイックスタート』を参照してください。

6.0.x の Mobile Link アップグレードスクリプトは削除されました。このアップグレードが必要な場合は、テクニカルサポート (http://www.sybase.com/support) に連絡してください。

説明

- ml_add_missing_dnld_scripts ストアドプロシージャを使用して、欠落している download_cursor スクリプトか download_delete_cursor スクリプトまたはその両方を修正します。スクリプトバージョン名を使用してこのプロシージャを呼び出すと、指定したスクリプトバージョンで使用されるすべての同期テーブルに対して、欠落している download_cursor スクリプトと download_delete_cursor スクリプトが無視されるスクリプトとして定義されます。

- 10.0.0 より以前のバージョンで作成された authenticate_user_hashed スクリプトを使用する場合は、お使いの RDBMS でバイナリに相当する型を使用して、VARBINARY(20) ではなく VARBINARY(32) を受け入れるようにスクリプトを変更する必要があります。

統合データベースのアップグレード (SQL Anywhere 10.0.0 以降)

新しい Mobile Link サーバとバージョン 10.0.0 以降の既存の SQL Anywhere 統合データベースを使用できるようにするには、アップグレードスクリプトを実行し、新しいシステムオブジェクトをインストールしてください。

前提条件

設定スクリプトを実行するには、現在インストールしている Mobile Link システムテーブルの所有者になっている必要があります。

ロック/ブロック検出ロジックを呼び出すには、MONITOR システム権限が必要です。

内容と備考

Mobile Link システム設定を更新するには、次の方法を使用することもできます。

- Sybase Central の Mobile Link プラグインで、**[Mobile Link 16]** » **[プロジェクト]** » **[統合データベース]** をクリックし、データベース名を右クリックして、**[Mobile Link システム設定のチェック]** をクリックします。データベースの設定またはアップグレードが必要な場合は、続行のプロンプトが表示されます。

- **同期モデル展開ウィザード**を使用する場合は、統合データベースに接続するとシステム設定がチェックされます。データベースの設定またはアップグレードが必要な場合は、続行のプロンプトが表示されます。「同期モデル」『Mobile Link クイックスタート』を参照してください。

♦ タスク

1. SQL Anywhere ソフトウェアをアップグレードします。

   「バージョン 10 以降のデータベースのアップグレード」289 ページを参照してください。

2. アップグレード元のバージョン用の適切なアップグレードスクリプトを実行して、Mobile Link システム設定をアップグレードします。

   アップグレードスクリプトは、*upgrade_sa.sql* という名前です。これは、SQL Anywhere インストール環境の *MobiLink¥upgrade¥version* にあります。*version* は、アップグレード元の SQL Anywhere バージョンを示します。

   たとえば、Interactive SQL からデータベースに接続し、次の文を実行します。

   ```
   READ "C:¥Program Files¥SQL Anywhere 16¥MobiLink¥upgrade¥10.0.x¥upgrade_sa.sql"
   ```

結果

統合データベースは、新しい Mobile Link サーバとともに使用できるようになりました。

統合データベースのアップグレード (Adaptive Server Enterprise、SAP Sybase IQ、Oracle、MySQL、または Microsoft SQL Server)

新しい Mobile Link サーバと既存の統合データベースを使用できるようにするには、アップグレードスクリプトを実行し、新しいシステムオブジェクトをインストールしてください。

前提条件

設定スクリプトを実行するには、現在インストールしている Mobile Link システムテーブルの所有者になっている必要があります。

ASE の場合、Mobile Link サーバのログイン ID には、master..systransactions に対する SELECT 権限が必要です。

SAP Sybase IQ の場合、ダウンロード用のスナップショットアイソレーションを使用するには、Mobile Link サーバで SP_IQTRANSACTION に対する EXECUTE 権限が必要です。

Oracle の場合、Mobile Link サーバを統合データベースに接続するために使用される RDBMS ユーザは、Mobile Link システムテーブル、プロシージャなどを修飾子なしで使用できる必要があります (SELECT * from ml_user など)。また、RDBMS ユーザの場合は、GV$TRANSACTION、GV$SESSION、GV$LOCK、DBA_OBJECTS での SELECT 権限、および DBMS_UTILITY での EXECUTE 権限も必要です。同義語である GV$TRANSACTION、GV$SESSION、GV$LOCK には直接パーミッションを付与することができません。そのため、同義語の派生元である GV_$TRANSACTION、GV_$SESSION、および GV_$LOCK の動的パフォーマンスビューに対してパーミッションを付与する必要があります。このアクセス権を付与するには、SYS として接続する必要があります。

Microsoft SQL Server の場合、統合データベースへの接続のために Mobile Link サーバが使用する RDBMS ユーザは、VIEW SERVER STATE のパーミッション、SYS.DATABASES からの SELECT のパーミッション、SYS.DM_TRAN_LOCKS、SYS.PARTITIONS、SYS.SYSPROCESSES からの SELECT のパーミッションが必要です。

内容と備考

Adaptive Server Enterprise、Oracle、MySQL、または Microsoft SQL Server で構築された統合データベース内の Mobile Link システムオブジェクトをアップグレードする必要があるのは、16.0 より以前のバージョンの Mobile Link サーバを使用している場合だけです。

Mobile Link システム設定を更新するには、次の方法を使用することもできます。

- Sybase Central の Mobile Link プラグインで、**[Mobile Link 16]** » **[プロジェクト]** » **[統合データベース]** をクリックし、データベース名を右クリックして、**[Mobile Link システム設定のチェック]** をクリックします。データベースの設定またはアップグレードが必要な場合は、続行のプロンプトが表示されます。

- **同期モデル展開ウィザード**を使用する場合は、統合データベースに接続するとシステム設定がチェックされます。データベースの設定またはアップグレードが必要な場合は、続行のプロンプトが表示されます。「同期モデル」『Mobile Link クイックスタート』を参照してください。

♦ タスク

1. Adaptive Server Enterprise の場合は、SELECT INTO データベースオプションを設定します。Sybase Interactive SQL で次の文を実行します。

```
USE MASTER
go
```

```
sp_dboption your-database-name, "SELECT INTO", true
go
USE your-database-name
go
checkpoint
go
```

2. アップグレードするデータベースのバージョンに対応したアップグレードスクリプトを実行します。

   アップグレードスクリプトは *upgrade_XXX.sql* という名前です。*XXX* は、統合データベースの RDBMS を示します。これは、SQL Anywhere インストール環境の *MobiLink¥upgrade¥version* にあります。*version* は、アップグレード元の Mobile Link バージョンを示します。

   たとえば、バージョン 9.0.2 の Mobile Link システムテーブルが適用された Microsoft SQL Server データベースをアップグレードするには、次のコマンドを実行します。

```
osql -S server_name -U user_name -P password -i
"C:¥Program Files¥SQL Anywhere 16¥MobiLink¥upgrade¥9.0.2¥upgrade_mss.sql"
```

結果

統合データベースは、新しい Mobile Link サーバとともに使用できるようになりました。

IBM DB2 LUW 統合データベースのアップグレード

新しい Mobile Link サーバと既存の IBM DB2 LUW 統合データベースを使用できるようにするには、アップグレードスクリプトを実行し、新しいシステムオブジェクトをインストールしてください。

前提条件

設定スクリプトを実行するには、現在インストールしている Mobile Link システムテーブルの所有者になっている必要があります。

DB2 LUW 10.1 の場合、ロック／ブロック検出ロジックを呼び出すには、Mobile Link サーバで SYSIBMADM.MON_LOCKWAITS、SNAPSHOT_APPL_INFO からの SELECT のパーミッションが必要です。

DB2 LUW 9 の場合、ロック／ブロック検出ロジックを呼び出すには、Mobile Link サーバで SYSIBMADM.LOCKWAITS、SNAPSHOT_APPL_INFO からの SELECT のパーミッションが必要です。

内容と備考

IBM DB2 LUW の統合データベースをアップグレードする必要があるのは、以前のバージョンの Mobile Link サーバが 16.0 より前のバージョンである場合だけです。

IBM DB2 LUW 設定スクリプトを実行する方法については、「IBM DB2 LUW 統合データベース」『Mobile Link サーバ管理』を参照してください。

Mobile Link システム設定を更新するには、次の方法を使用することもできます。

- Sybase Central の Mobile Link プラグインで、**[Mobile Link 16]** » **[プロジェクト]** » **[統合データベース]** をクリックし、データベース名を右クリックして、**[Mobile Link システム設定のチェック]** をクリックします。データベースの設定またはアップグレードが必要な場合は、続行のプロンプトが表示されます。

- **同期モデル展開ウィザード**を使用する場合は、統合データベースに接続するとシステム設定がチェックされます。データベースの設定またはアップグレードが必要な場合は、続行のプロンプトが表示されます。「同期モデル」『Mobile Link クイックスタート』を参照してください。

♦ タスク

1. IBM DB2 LUW 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_db2.sql* という名前で、SQL Anywhere インストール環境の *MobiLink/upgrade/version* サブフォルダにあります。*version* ディレクトリは、アップグレード元の Mobile Link のバージョンを示します。

2. *upgrade_db2.sql* をコピーし、このコピーを変更します。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。

結果

統合データベースは、新しい Mobile Link サーバとともに使用できるようになりました。

統合データベースのアップグレード (10.0.0 より以前の SQL Anywhere)

新しい Mobile Link サーバとバージョン 10.0.0 より以前の既存の SQL Anywhere 統合データベースを使用できるようにするには、アップグレードスクリプトを実行し、新しいシステムオブジェクトをインストールしてください。

前提条件

SQL Anywhere 統合データベースは設定済みで、同期はまだ一度も実行していない場合は、アップグレードスクリプトではなく、設定スクリプトを実行する必要があります。この手順は、SQL Anywhere 統合データベースだけに適用されます。「SQL Anywhere 統合データベース」『Mobile Link サーバ管理』を参照してください。

内容と備考

- SQL Anywhere 10.0.0 より以前のバージョンでは、Mobile Link システムテーブルの所有者は dbo でした。SQL Anywhere データベースの設定スクリプトを実行するには、Mobile Link システムテーブルの所有者として統合データベースにログインする必要があります。テーブルの変更パーミッションを持つユーザとしてこのスクリプトを実行しても十分ではありません。アップグレードスクリプトを実行するには、SQL 文 SETUSER を使用して dbo を同一化する方法もあります。次に例を示します。

```
SETUSER "dbo";
```

Sybase Central で統合データベースをアップグレードするには、GRANT CONNECT 文を使用して dbo のパスワードを作成してから、dbo として接続する必要があります。次に例を示します。

```
GRANT CONNECT TO dbo IDENTIFIED BY password;
```

この場合、アップグレード後に、ALTER USER を使用して dbo のパスワードを削除する必要があります。次に例を示します。

```
ALTER USER TO dbo IDENTIFIED BY "";
```

♦ タスク

1. バージョン 10.0.0 より以前の SQL Anywhere の統合データベースをアップグレードする場合は、先にデータベースをバージョン 16 にアップグレードする必要があります。
   a. データベースサーバを停止します。
   b. データベースをバージョン 16 にアップグレードします。

      手順については、次の項を参照してください。

      ● 「バージョン 9 以前のデータベースの再構築手順」297 ページ
   c. DBA としてログインした状態で、データベースサーバを起動します。

      アップグレードするために DBA としてログインします。
2. アップグレードするデータベースのバージョンに対応したアップグレードスクリプトを実行します。

   アップグレードスクリプトは、*upgrade_asa.sql* という名前です。これは、SQL Anywhere インストール環境の *MobiLink¥upgrade¥version* にあります。*version* は、アップグレード元の SQL Anywhere バージョンを示します。

   アップグレードスクリプトを実行するには、SETUSER SQL 文を使用して dbo ユーザに同一化します。

   たとえば、SQL Anywhere バージョン 9.0.2 の統合データベースをアップグレードする場合は、Interactive SQL でデータベースに接続し、次の文を実行します。

   ```
   SETUSER "dbo";
   READ 'C:¥Program Files¥SQL Anywhere 16¥MobiLink¥upgrade¥9.0.2¥upgrade_asa.sql'
   ```
3. dbo のパスワードを削除します。次に例を示します。

   ```
   GRANT CONNECT TO "dbo";
   ```
4. DBA 以外のユーザとして Mobile Link サーバを実行している場合は、新しい Mobile Link システムオブジェクトの EXECUTE パーミッションをこのユーザに付与してください。新しいシステムオブジェクトは、アップグレード対象のバージョンによって異なります。次のコードで、すべての Mobile Link システムオブジェクトへの必要なパーミッションが付与されます。このコードを実行する前に、ユーザ名 my_user を、Mobile Link サーバを実行しているユーザの名前に変更してください。

   ```
   GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_column to my_user;
   GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_connection_script to my_user;
   ```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_database to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_device to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_device_address to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_listening to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_passthrough to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_passthrough_repair to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_passthrough_script to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_passthrough_status to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_primary_server to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_property to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_delivery to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_delivery_archive to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_global_props to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_notifications to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_repository to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_repository_archive to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_repository_props to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_repository_props_archive to
my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_repository_staging to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_status_history to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_status_history_archive to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_qa_status_staging to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_agent to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_agent_property to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_agent_staging to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_deployed_task to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_event to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_event_staging to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_managed_remote to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_notify to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_remote_db_class to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_task to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_task_command to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_task_command_property to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_ra_task_property to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_script to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_script_version to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_scripts_modified to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_sis_sync_state to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_subscription to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_table to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_table_script to my_user;
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.ml_user to my_user;
GRANT EXECUTE ON dbo.ml_add_column to my_user;
GRANT EXECUTE ON dbo.ml_add_connection_script to my_user;
GRANT EXECUTE ON dbo.ml_add_dnet_connection_script to my_user;
GRANT EXECUTE ON dbo.ml_add_dnet_table_script to my_user;
GRANT EXECUTE ON dbo.ml_add_java_connection_script to my_user;
GRANT EXECUTE ON dbo.ml_add_java_table_script to my_user;
GRANT EXECUTE ON dbo.ml_add_lang_conn_script_chk to my_user;
GRANT EXECUTE ON dbo.ml_add_lang_connection_script to my_user;
GRANT EXECUTE ON dbo.ml_add_lang_table_script to my_user;
GRANT EXECUTE ON dbo.ml_add_lang_table_script_chk to my_user;
GRANT EXECUTE ON dbo.ml_add_missing_dnld_scripts;
GRANT EXECUTE ON dbo.ml_add_passthrough to my_user;
GRANT EXECUTE ON dbo.ml_add_passthrough_repair to my_user;
GRANT EXECUTE ON dbo.ml_add_passthrough_script to my_user;
GRANT EXECUTE ON dbo.ml_add_property to my_user;
GRANT EXECUTE ON dbo.ml_add_table_script to my_user;
GRANT EXECUTE ON dbo.ml_add_user to my_user;
GRANT EXECUTE ON dbo.ml_delete_device to my_user;
GRANT EXECUTE ON dbo.ml_delete_device_address to my_user;
GRANT EXECUTE ON dbo.ml_delete_listening to my_user;
```

```
GRANT EXECUTE ON dbo.ml_delete_passthrough to my_user;
GRANT EXECUTE ON dbo.ml_delete_passthrough_repair to my_user;
GRANT EXECUTE ON dbo.ml_delete_passthrough_script to my_user;
GRANT EXECUTE ON dbo.ml_delete_remote_id to my_user;
GRANT EXECUTE ON dbo.ml_delete_sync_state to my_user;
GRANT EXECUTE ON dbo.ml_delete_sync_state_before to my_user;
GRANT EXECUTE ON dbo.ml_delete_user to my_user;
GRANT EXECUTE ON dbo.ml_delete_user_state to my_user;
GRANT EXECUTE ON dbo.ml_lock_rid to my_user;
GRANT EXECUTE ON dbo.ml_qa_add_delivery to my_user;
GRANT EXECUTE ON dbo.ml_qa_add_message to my_user;
GRANT EXECUTE ON dbo.ml_qa_handle_error to my_user;
GRANT EXECUTE ON dbo.ml_qa_stage_status_from_client to my_user;
GRANT EXECUTE ON dbo.ml_qa_staged_status_for_client to my_user;
GRANT EXECUTE ON dbo.ml_qa_upsert_global_prop to my_user;
GRANT EXECUTE ON dbo.ml_ra_add_agent_id to my_user;
GRANT EXECUTE ON dbo.ml_ra_assign_task to my_user;
GRANT EXECUTE ON dbo.ml_ra_cancel_notification to my_user;
GRANT EXECUTE ON dbo.ml_ra_cancel_task_instance to my_user;
GRANT EXECUTE ON dbo.ml_ra_clone_agent_properties to my_user;
GRANT EXECUTE ON dbo.ml_ra_delete_agent_id to my_user;
GRANT EXECUTE ON dbo.ml_ra_delete_events_before to my_user;
GRANT EXECUTE ON dbo.ml_ra_delete_task to my_user;
GRANT EXECUTE ON dbo.ml_ra_get_agent_events to my_user;
GRANT EXECUTE ON dbo.ml_ra_get_agent_ids to my_user;
GRANT EXECUTE ON dbo.ml_ra_get_agent_properties to my_user;
GRANT EXECUTE ON dbo.ml_ra_get_latest_event_id to my_user;
GRANT EXECUTE ON dbo.ml_ra_get_orphan_taskdbs to my_user;
GRANT EXECUTE ON dbo.ml_ra_get_remote_ids to my_user;
GRANT EXECUTE ON dbo.ml_ra_get_task_results to my_user;
GRANT EXECUTE ON dbo.ml_ra_get_task_status to my_user;
GRANT EXECUTE ON dbo.ml_ra_int_cancel_notification to my_user;
GRANT EXECUTE ON dbo.ml_ra_int_move_events to my_user;
GRANT EXECUTE ON dbo.ml_ra_manage_remote_db;
GRANT EXECUTE ON dbo.ml_ra_notify_agent_sync to my_user;
GRANT EXECUTE ON dbo.ml_ra_notify_task to my_user;
GRANT EXECUTE ON dbo.ml_ra_reassign_taskdb to my_user;
GRANT EXECUTE ON dbo.ml_ra_set_agent_property to my_user;
GRANT EXECUTE ON dbo.ml_ra_ss_agent_auth_file_xfer to my_user;
GRANT EXECUTE ON dbo.ml_ra_ss_download_ack to my_user;
GRANT EXECUTE ON dbo.ml_ra_ss_download_prop to my_user;
GRANT EXECUTE ON dbo.ml_ra_ss_download_remote_dbs to my_user;
GRANT EXECUTE ON dbo.ml_ra_ss_download_task to my_user;
GRANT EXECUTE ON dbo.ml_ra_ss_download_task2;
GRANT EXECUTE ON dbo.ml_ra_ss_download_task_cmd to my_user;
GRANT EXECUTE ON dbo.ml_ra_ss_end_upload to my_user;
GRANT EXECUTE ON dbo.ml_ra_ss_upload_prop to my_user;
GRANT EXECUTE ON dbo.ml_ra_unmanage_remote_id to my_user;
GRANT EXECUTE ON dbo.ml_reset_sync_state to my_user;
GRANT EXECUTE ON dbo.ml_set_device to my_user;
GRANT EXECUTE ON dbo.ml_set_device_address to my_user;
GRANT EXECUTE ON dbo.ml_set_listening to my_user;
GRANT EXECUTE ON dbo.ml_set_sis_sync_state to my_user;
GRANT EXECUTE ON dbo.ml_upload_update_device_address to my_user;
GRANT EXECUTE ON dbo.ml_upload_update_listening to my_user;
```

結果

統合データベースは、新しい Mobile Link サーバとともに使用できるようになりました。

Mobile Link サーバのアップグレード

バージョン 16 のリモートデータベースを同期する場合、Mobile Link サーバをバージョン 16 にアップグレードしてください。

バージョン 16 の Mobile Link 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。SQL Anywhere 16 変更点とアップグレード 1 ページを参照してください。

Mobile Link バージョン 16 サーバでサポートされるのは、バージョン 10、11、および 12 の SQL Anywhere クライアントと Ultra Light クライアントだけです。これより前のクライアントをサポートする必要がある場合は、そのバージョンのクライアントをサポートしている古いバージョンの Mobile Link サーバを使用してください。

SQL 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 サーバの両方をアップグレードしてから、SQL Anywhere リモートデータベースをアップグレードします。

バージョン 10.0.0 で、Adaptive Server Anywhere は SQL Anywhere に名前が変更されました。

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

ソフトウェアのアップグレード

dbmlsync と SQL Anywhere データベースサーバは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 16 の Mobile Link クライアントはバージョン 16 データベースサーバで動作するバージョン 16 データベースのみ同期できます。

バージョン 16 の Mobile Link クライアントでは、同期を実行するためにバージョン 16 以降の Mobile Link サーバが必要です。バージョン 16 の Mobile Link クライアントは、バージョン 16 より以前の Mobile Link 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」319 ページを参照してください。

リモートの SQL Anywhere のアップグレード

Mobile Link SQL Anywhere リモートデータベースは、SQL Anywhere データベースのアップグレードと同じ手順でアップグレードできます。手順については、「SQL Anywhere サーバのアップグレード」289 ページを参照してください。

スキーマの変更やデータベースでのその他の重要な変更などがある場合は、手動でアンロードと再ロードを行う必要があります。「リモートの SQL Anywhere データベースの手動でのアンロードおよび再ロード」329 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スクリプトバージョン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。両方のスクリプトバージョンを同時に使用することができるため、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簡単にできます。

バージョン 9.0.0 以降では、Mobile Link サーバの -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。また、クライアントをアップグレードして名前付きパラメータの使用を開始することで、スクリプトパラメータを任意の順序で配置できます。

リモートの SQL Anywhere データベースの手動でのアンロードおよび再ロード

スキーマの変更やデータベースでのその他の重要な変更などがある場合は、手動でアンロードと再ロードを行う必要があります。

前提条件

バージョン 16 のデータベースでは、次のシステム権限が必要です。

- BACKUP DATABASE
- VALIDATE ANY OBJECT
- SERVER OPERATOR
- SELECT ANY TABLE

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。「アップロードと再ロードの注意事項」291 ページを参照してください。

♦ タスク

1. すべてのデータベースアクティビティを停止します。
2. 同期が正常に完了してから、リモートデータベースを検証し、バックアップします。
3. dbtran ユーティリティを実行してデータベースのトランザクションログの開始オフセットと終了オフセットを表示します。終了オフセットをメモしてください。

   「ログ変換ユーティリティ (dbtran)」『SQL Anywhere サーバ データベース管理』を参照してください。

4. アンロードプロセス中に変更されないように、トランザクションログの名前を変更しておきます。名前変更したログファイルを、オフラインディレクトリなどの安全なロケーションに移動します。

5. データベースをアンロードします。データベースを別のデータベースに自動的に再ロードする dbunload -a スイッチは使用しないでください。

   「アンロードユーティリティ (dbunload)」『SQL Anywhere サーバ データベース管理』を参照してください。

6. 新しいデータベースを初期化します。

   「初期化ユーティリティ (dbinit)」『SQL Anywhere サーバ データベース管理』を参照してください。

7. dbisql を使用してデータを新しいデータベースに再ロードし、dbunload によって生成された **reload.sql** ファイルを読み込みます。

   「Interactive SQL ユーティリティ (dbisql)」『SQL Anywhere サーバ データベース管理』を参照してください。

8. 新しいデータベースを停止します。

9. 新しいデータベースのトランザクションログを消去します。

10. 次のオプションを使用して、新しいデータベースで dblog を実行します。
    - -z を使用して、メモした終了オフセットを指定します。
    - -x を使用して、相対オフセットをゼロに設定します。

    次に例を示します。

    ```
    dblog -x 0 -z 137829 database-name.db
    ```

    「トランザクションログユーティリティ (dblog)」『SQL Anywhere サーバ データベース管理』を参照してください。

11. dbmlsync を起動し、移動した元のログファイルの場所を指定します。

    「dbmlsync 構文」『Mobile Link クライアント管理』を参照してください。

12. 古いログファイルが不要な場合は、データベースオプション delete_old_logs を設定します。

    「delete_old_logs オプション [SQL Remote]」『SQL Anywhere サーバ データベース管理』を参照してください。

結果

リモート SQL Anywhere データベースがアンロードされ、再ロードされます。

Ultra Light のアップグレード

Ultra Light バージョン 16 データベースへのアップグレードの前に、次の点に留意してください。

- Ultra Light バージョン 16 は、以前のバージョンのソフトウェアを使用して作成された Ultra Light データベースを読み込むことができません。
- デバイス上でデータベースをアップグレードすることはできません。
- データベースがアップグレードされたら、以前のバージョンのアプリケーション、ユーティリティ、ソフトウェアに接続できなくなります。
- SQL Anywhere の複数のバージョンが同じコンピュータにインストールされている場合は、正しいユーティリティを使用するため、システムのパスに注意してください。Ultra Light ユーティリティの名前はバージョン間で共通で、ディレクトリのパス名のみが異なります。

このバージョンのソフトウェアで既存のアプリケーションを使用する前に、新機能と動作の変更のリストを確認して、アプリケーションに影響がないかどうかを確認してください。

バージョン 12 以前のデータベースのアップグレード

バージョン 12 以前の Ultra Light のデータベースは、データベースの作成に使用したバージョンのソフトウェアを使用して、SQL または XML ファイルにアンロードする必要があります。ソフトウェアに付属している ulunload ユーティリティを使用してください。

古いバージョンのソフトウェアがインストールされていなくて、Windows デスクトップオペレーティングシステムを使用している場合、SQL Anywhere バージョン 16 に組み込まれているいずれかの古い ulunload ユーティリティを使用できます。使用している Ultra Light データベースのバージョンに合った ulunload ユーティリティを選択してください。

データベースのアンロードが完了したら、バージョン 16 の ulload ユーティリティを使用して XML ファイルをバージョン 16 のデータベースにロードします。データベースを SQL ファイルにアンロードした場合は、dbisql ユーティリティを使用します。

バージョン 12.0.1 Java Edition データベースのアップグレード

新しい Ultra Light Java Edition データベースを作成するためのバージョン 16 の uljload ユーティリティで使用可能な XML ファイルを生成するには、Ultra Light Java Edition バージョン 12.0.1 の ULjUnload ユーティリティを使用します。

既存のソフトウェアとの互換性

- Ultra Light 16 のデータベースファイルは、バージョン 16 のクライアントアプリケーション、またはバージョン 16 の Ultra Light エンジンからの接続のみをサポートします。
- Ultra Light バージョン 16 ランタイムと Ultra Light バージョン 16 エンジンは、バージョン 9 以前の Ultra Light で作成したデータベースファイルおよびアプリケーションコードでは使用できません。

参照

- 「Ultra Light の新機能」263 ページ
- 「Ultra Light の動作の変更と廃止予定機能」269 ページ
- 「複数のバージョンのユーティリティがインストールされている場合に正しいバージョンを実行していることを確認する方法」292 ページ

Ultra Light データベースのアップグレード (Windows)

旧バージョンのユーザは、次の手順に従って、Ultra Light データベースをバージョン 16 に再構築できます。

前提条件

- Ultra Light バージョン 9 以前のデータベースからアップグレードする場合は、そのバージョンの SQL Anywhere がインストールされていなければなりません。
- 既存の Ultra Light データベースのバックアップコピーを作成してください。
- 同期されていない変更が含まれている可能性がある運用データベースの場合は、データベースを同期します。

内容と備考

Ultra Light バージョン 16 は、以前のバージョンのソフトウェアを使用して作成された Ultra Light データベースを読み込むことができません。

♦ タスク

1. データベースの内容で XML ファイル (複数可) を作成します。

   コマンドプロンプトを開き、以前のバージョンの SQL Anywhere がインストールされたディレクトリに移動し、データベースに対して ulunload ユーティリティを実行してください。

   Ultra Light データベースがバージョン 10、11、12 の SQL Anyuwhere で作成されている場合、コマンドプロンプトを開き、*%SQLANY16%¥UltraLite¥Unload¥* ディレクトリに移動し、使用しているバージョンの Ultra Light データベースに適用されるディレクトリ (*V10*、*V11*、*V12* など) を選択し、そのディレクトリから ulunload ユーティリティを実行するという方法もあります。

2. Ultra Light バージョン 16 ulload ユーティリティを使用するか、Sybase Central の **Ultra Light 16** プラグインで**データベースロードウィザード**を使用して、スキーマとデータを新しいバージョン 16 のデータベースにロードします。

   Ultra Light データベースのデフォルトは UTF8 エンコーディングです。このエンコードがニーズに適合しない場合は、utf8_encoding パラメータを明示的にオフに設定する必要があります。

3. 生成された XML ファイルをチェックして、UTF-8 エンコーディングの設定を確認します。

結果

Ultra Light データベースが最新バージョンにアップグレードされます。

参照

- 「Ultra Light utf8_encoding 作成パラメータ」『Ultra Light データベース管理とリファレンス』
- 「Ultra Light データベースのアンロードユーティリティ (ulunload)」『Ultra Light データベース管理とリファレンス』
- 「Ultra Light データベースへの XML のロードユーティリティ (ulload)」『Ultra Light データベース管理とリファレンス』

Ultra Light データベースのアップグレード (Linux)

旧バージョンのユーザは、次の手順に従って、データベースをバージョン 16 に再構築できます。

前提条件

- データベースを作成したバージョンの SQL Anywhere をインストールしてください。
- 既存の Ultra Light データベースのバックアップコピーを作成してください。
- 同期されていない変更が含まれている可能性がある運用データベースの場合は、データベースを同期します。

内容と備考

Ultra Light バージョン 16 は、以前のバージョンのソフトウェアを使用して作成された Ultra Light データベースを読み込むことができません。

♦ タスク

1. データベースの内容で XML ファイル (複数可) を作成します。

   コマンドプロンプトで *%SQLANY%/bin32* (**%SQLANY%** は前の SQL Anywhere がインストールされている場所) に移動し、データベースに対して ulunload ユーティリティを実行します。

2. Ultra Light バージョン 16 ulload ユーティリティを使用するか、Sybase Central の **Ultra Light 16** プラグインで**データベースロードウィザード**を使用して、スキーマとデータを新しいバージョン 16 のデータベースにロードします。

   Ultra Light データベースのデフォルトは UTF8 エンコーディングです。このエンコードがニーズに適合しない場合は、utf8_encoding パラメータを明示的にオフに設定する必要があります。

3. 生成された XML ファイルをチェックして、UTF-8 エンコーディングの設定を確認します。

結果

Ultra Light データベースが最新バージョンにアップグレードされます。

参照

- 「Ultra Light utf8_encoding 作成パラメータ」『Ultra Light データベース管理とリファレンス』
- 「Ultra Light データベースのアンロードユーティリティ (ulunload)」『Ultra Light データベース管理とリファレンス』
- 「Ultra Light データベースへの XML のロードユーティリティ (ulload)」『Ultra Light データベース管理とリファレンス』

SQL Remote 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、SQL Remote Message Agent があります。

それぞれのサイトでは、SQL Remote 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。

- **ソフトウェアアップグレードは一度に 1 サイトずつ実行できる**　旧バージョンの Message Agent (dbremote) は、バージョン 16 の Message Agent とメッセージを交換することができます。バージョン 5 の SQL Remote では、compression データベースオプションの値が -1 (マイナス 1) に設定されている場合、バージョン 5 の Message Agent とバージョン 16 の Message Agent で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。「compression オプション [SQL Remote]」『SQL Anywhere サーバ データベース管理』を参照してください。

- **データベースのアップグレード**　SQL Anywhere バージョン 9 以前を使用していたリモートデータベースまたは統合データベースをアップグレードする場合は、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。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、「バージョン 9 以前のデータベースの再構築手順」297 ページを参照してください。

- **Adaptive Server Enterprise 統合データベースのアップグレード**　SQL Remote では、Adaptive Server Enterprise 統合データベースはサポートされなくなりました。Adaptive Server Enterprise データベースを同期するには、Mobile Link にアップグレードする必要があります。

  SQL Remote から Mobile Link への移行については、http://www.sybase.com/detail?id=1034174 を参照してください。

- **SQL Remote のアップグレード**　バージョン 11.0.0 では、SQL Remote の VIM と MAPI の各メッセージシステムがサポートされなくなりました。VIM または MAPI を使用するデータベースを SQL Anywhere バージョン 16 にアップグレードする場合は、メッセージタイプを File、FTP、または SMTP に変更してください。メッセージタイプが MAPI または VIM の場合、*dbremote.exe* は起動しません。

例

以下に、バージョン 5 の SQL Remote の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 SQL Remote Message Agent をアップグレードしてから、統合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ファイル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 の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。 統合データベースのアンロードと再ロードの方法については、「同期やレプリケーションに関連するデータベースの再構築 (コマンドラインの場合)」『SQL Anywhere サーバ SQL の使用法』を参照してください。

2. リモートデータベースサーバと Message Agent を 1 つずつアップグレードしてから、リモート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ファイルフォーマット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 以外の値に設定すると、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コード化機能を利用できます。リモートデータベースのアンロードと再ロードの方法については、「同期やレプリケーションに関連するデータベースの再構築 (コマンドラインの場合)」『SQL Anywhere サーバ SQL の使用法』を参照してください。

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。

SQL Anywhere モニタのアップグレードおよびリソースの移行

1 つのモニタから、新しくインストールしたモニタへリソースを移行するには、移行ユーティリティを使用します。

前提条件

> **警告**
> モニタをアンインストールすると、アプリケーションだけでなく、リソース、収集したメトリックも削除されます。現在のモニタのリソースを保持する場合は、次の操作を実行します。
>
> 1. モニタの新しいバージョンをインストールします。
> 2. リソースを移行します。
> 3. モニタの古いバージョンをアンインストールします。

不要な警告を回避するため、サーバを停止する前に、サーバの SQL Anywhere モニタでブラックアウト期間をスケジュールしてください。

♦ タスク

1. 既存のモニタのデータベースファイル *samonitor.db* のバックアップコピーを作成します。

   バージョン 11.0.1 モニタおよびバージョン 16.0 モニタのデータベースファイルのデフォルトのロケーションを次の表に示します。

オペレーティングシステム	モニタタイプ	バージョン 11.0.1 と 12 のディレクトリ	バージョン 16.0 のディレクトリ
Windows XP	モニタ Developer Edition	C:¥Documents and Settings¥All Users ¥Documents¥SQL Anywhere 11¥Monitor ¥samonitor.db C:¥Documents and Settings¥All Users ¥Documents¥SQL Anywhere 12¥Monitor ¥samonitor.db	C:¥Documents and Settings¥All Users ¥Documents¥SQL Anywhere 16¥Monitor ¥samonitor.db
Windows XP	モニタ Production Edition	C:¥Documents and Settings¥All Users ¥Documents¥SQL Anywhere 11 Monitor ¥samonitor.db C:¥Documents and Settings¥All Users ¥Documents¥SQL Anywhere 12 Monitor ¥samonitor.db	C:¥Documents and Settings¥All Users ¥Documents¥SQL Anywhere 16 Monitor ¥samonitor.db
Windows Vista 以降のバージョンの Windows	モニタ Developer Edition	C:¥Users¥Public ¥Documents¥SQL Anywhere 11¥Monitor ¥samonitor.db C:¥Users¥Public ¥Documents¥SQL Anywhere 12¥Monitor ¥samonitor.db	C:¥Users¥Public ¥Documents¥SQL Anywhere 16¥Monitor ¥samonitor.db
Windows Vista 以降のバージョンの Windows	モニタ Production Edition	C:¥Users¥Public ¥Documents¥SQL Anywhere 11 Monitor ¥samonitor.db C:¥Users¥Public ¥Documents¥SQL Anywhere 12 Monitor ¥samonitor.db	C:¥Users¥Public ¥Documents¥SQL Anywhere 16 Monitor ¥samonitor.db
Linux	モニタ Developer Edition	/opt/sqlanywhere11/samonitor.db /opt/sqlanywhere12/samonitor.db	/opt/sqlanywhere16/samonitor.db
Linux	モニタ Production Edition	/opt/samonitor11/samonitor.db /opt/samonitor12/samonitor.db	/opt/sqlanywhere16/samonitor.db

2. 新しいモニタをインストールします。インストールメディアの *Monitor* ディレクトリにある *setup.exe* ファイルを実行し、表示される指示に従います。インストールが完了したら、新しいモニタを停止します (稼働している場合)。

   コンピュータ上で一度に実行できるモニタのバージョンは 1 つだけです。

   古いバージョンのモニタが稼働しているコンピュータにバージョン 16.0 のモニタをインストールすると、古いバージョンのモニタが停止します。

3. コマンドプロンプトで移行ユーティリティを実行します。

   次のオプションを指定して移行ユーティリティを実行します。

   - ***-t temporary-directory*** テンポラリファイルのディレクトリを指定します。デフォルトでは、テンポラリファイルは、run_migrator ファイルと同じディレクトリに作成されます。

     注意
     モニタ移行で作成されるテンポラリファイルは、移行プロセスの最後で削除されます。-t オプションを使用して、これらのテンポラリファイルのディレクトリを指定します。テンポラリファイルには、古いバージョンのモニタデータベースファイルと同等の領域が必要です。指定のディレクトリに十分な領域があることを確認してください。

   - ***source-filename*** 古いモニタのデータベースファイルにパスとファイル名を指定します。たとえば、バージョン 11.0.1 の *samonitor.db* ファイルのパスを指定します。
   - ***destination-filename*** リソースと設定がロードされる新しいモニタのファイルにパスとファイル名を指定します。たとえば、バージョン 16.0 の *samonitor.db* ファイルのパスを指定します。

   例：

   ```
   C:¥Program Files¥SQL Anywhere 16¥run_migrator.cmd -t c:¥monitorbackup c:¥Program Files
   ¥SQL Anywhere 11¥Monitor¥samonitor11.db C:¥Program Files¥SQL Anywhere 16¥Monitor
   ¥samonitor16.db
   ```

   次の表を使用して、移行ユーティリティのロケーションを確認します。

オペレーティングシステム	モニタタイプ	移行ユーティリティのデフォルトロケーション
Windows	モニタ Developer Edition	*C:¥Program Files¥SQL Anywhere 16¥Bin32¥run_migrator.cmd*
	モニタ Production Edition	*C:¥Program Files¥SQL Anywhere 16¥Bin32¥run_migrator.cmd*
Linux	モニタ Developer Edition	*/opt/sqlanywhere16/bin32/run_migrator.sh*

オペレーティングシステム	モニタタイプ	移行ユーティリティのデフォルトロケーション
	モニタ Production Edition	*/opt/sqlanywhere16/bin32/run_migrator.sh*

結果

モニタがアップグレードされます。

次の手順

モニタを起動します。

参照

- 「モニタの起動」『SQL Anywhere サーバ データベース管理』

索引

記号

.dbisqlg
バージョン 11.0.0 で名前が変更, 278
.NET
バージョン 11.0.0 の動作変更, 282
.NET および C++ 用の dbmlsync API
バージョン 11.0.0 の新機能, 260
.scRepository
バージョン 11.0.0 で名前が変更, 274
@data オプション
バージョン 11.0.0 の強化, 222
バージョン 12.0.1 の新機能、Mobile Link (mlgenreplayapi), 98
バージョン 12.0.1 の新機能、Mobile Link (mlreplay), 98
#hook_dict テーブル
バージョン 16.0 の動作変更、Mobile Link, 51
% 演算子
バージョン 11.0.0 の動作変更, 254
1 回のアプリケーションアクセス
バージョン 16.0 の新機能、Ultra Light, 59
256 ビットの AES 暗号化
バージョン 11.0.0 の新機能, 221
バージョン 16.0 の新機能、Ultra Light, 59
32 ビット Mobile Link サーバ
バージョン 16.0 の動作変更, 50
32 ビット Relay Server
バージョン 16.0 のサポート終了 (Linux), 56
バージョン 16.0 のサポート終了 (Windows), 56
64 ビット
バージョン 11.0.0 の新機能、Mobile Link サポート, 259
バージョン 11.0.0 の動作変更, 251
-AcceptCharset オプション
バージョン 11.0.1 の新機能, 199
-ac オプション
バージョン 12.0.0 の強化、dbsupport ユーティリティ, 129
-al オプション
バージョン 16.0 の新機能, 20
-aw オプション
バージョン 16.0 の新機能, 13
-bc オプション
バージョン 16.0 の新機能, 13
-bkr オプション
バージョン 12.0.0 の新機能、Mobile Link, 162
-bk オプション
バージョン 12.0.0 の新機能、Mobile Link, 162
-ca オプション
バージョン 12.0.0 の新機能、Mobile Link, 159
-cc オプション
バージョン 16.0 の動作変更, 33
-chx オプション
バージョン 12.0.0 の新機能, 146
-ch オプション
バージョン 16.0 で dbsupport のサポート終了, 33
バージョン 16.0 の強化, 31
-cinit オプション
バージョン 12.0.0 の新機能、Mobile Link, 158
バージョン 12.0.1 の強化、Mobile Link, 97
-ci オプション
バージョン 12.0.0 の新機能、Mobile Link, 162
-cl オプション
バージョン 12.0.0 の新機能、Mobile Link, 162
バージョン 12.0.1 の動作変更, 94
-cmax オプション
バージョン 12.0.0 の新機能、Mobile Link, 158
バージョン 12.0.1 の強化、Mobile Link, 97
-cmin オプション
バージョン 12.0.0 の新機能、Mobile Link, 158
バージョン 12.0.1 の強化、Mobile Link, 97
-cm オプション
バージョン 11.0.1 の新機能、dbunload, 198
バージョン 12.0.0 の新機能、Mobile Link, 163
バージョン 12.0.1 の動作変更、Mobile Link (mlsrv12), 101
バージョン 16.0 でサポート終了, 40
-cn オプション
バージョン 12.0.0 の動作変更、Mobile Link, 165
-codepage オプション
バージョン 11.0.0 で廃止予定, 278
バージョン 12.0.0 で廃止, 192
-cp オプション
バージョン 11.0.0 の強化、dbunload, 222
バージョン 16.0 の動作変更, 33
-cs オプション、Mobile Link (mlsrv11)
バージョン 11.0.0 の新機能, 259
-cw オプション
バージョン 16.0 でサポート終了, 40
-c オプション

バージョン 11.0.1 の強化、Mobile Link (mlmon), 209
バージョン 12.0.1 の動作変更, 94
-df オプション
バージョン 12.0.0 の強化、Mobile Link, 167
バージョン 12.0.0 の動作変更、Mobile Link, 167
-dp オプション
バージョン 12.0.0 の強化、Mobile Link, 167
バージョン 12.0.0 の動作変更、Mobile Link, 167
-ec データベースサーバオプション
バージョン 16.0 で廃止, 7
-es オプション
バージョン 11.0.0 の新機能, 224
-e オプション
バージョン 11.0.0 でサポート終了、初期化ユーティリティ (dbinit) のオプション, 254
-fr オプション
バージョン 12.0.0 のサポート終了、Mobile Link, 168
-ftru オプション
バージョン 12.0.0 の新機能、Mobile Link, 159
-f オプション
バージョン 12.0.0 のサポート終了、Mobile Link, 168
-ga オプション
バージョン 11.0.0 で廃止予定、Mobile Link Listener ユーティリティ (dblsn) のオプション, 262
-gb オプション
バージョン 11.0.0 の強化, 224
-gd オプション
バージョン 16.0 の動作変更, 3
-gi オプション
バージョン 11.0.0 で廃止予定、Mobile Link Listener ユーティリティ (dblsn) のオプション, 262
-gk オプション
バージョン 16.0 の動作変更, 3
-gl オプション
バージョン 16.0 の動作変更, 3
-gna オプション
バージョン 12.0.0 の強化, 125
バージョン 12.0.1 の動作変更, 94
-gnh オプション
バージョン 12.0.0 の強化, 125
バージョン 12.0.1 の動作変更, 94
-gnl オプション
バージョン 12.0.0 の強化, 125
バージョン 12.0.1 の動作変更, 94
-gns オプション
バージョン 12.0.0 の新機能, 125
バージョン 12.0.1 の動作変更, 94
-gn オプション
バージョン 12.0.0 の強化, 125
-gss オプション
バージョン 11.0.0 の強化, 239
バージョン 11.0.0 の動作変更, 247
-gta オプション
バージョン 16.0 の新機能, 20
-gu オプション
バージョン 12.0.0 の動作変更, 154
バージョン 16.0 の動作変更, 3
-gx オプション
バージョン 11.0.0 でサポート終了, 247
-g オプション
バージョン 11.0.0 で廃止予定、Mobile Link Listener ユーティリティ (dblsn) のオプション, 262
バージョン 11.0.0 の強化、dbunload, 222
バージョン 12.0.0 で削除、トランザクションログユーティリティ (dblog) のオプション, 118
-h オプション
バージョン 16.0 の新機能, 13
-il オプション
バージョン 12.0.0 で削除、トランザクションログユーティリティ (dblog) のオプション, 118
-im オプション
バージョン 11.0.0 の新機能, 224
-is オプション
バージョン 12.0.0 の動作変更、ログ変換ユーティリティ (dbtran) のオプション, 118
-i オプション
バージョン 12.0.0 の新機能、Mobile Link, 164
-kd オプション
バージョン 12.0.0 の強化、dbunload ユーティリティ, 130
-kp オプション
バージョン 12.0.1 の新機能, 85
-kr オプション
バージョン 12.0.1 で廃止, 96
-ksc オプション
バージョン 11.0.0 の新機能, 225
-ksd オプション
バージョン 11.0.0 の新機能, 225

-ks オプション
バージョン 11.0.0 の新機能, 224
-k オプション
バージョン 12.0.0 の強化、dblic ユーティリティ, 129
-lsc オプション、Mobile Link (mlsrv11)
バージョン 11.0.0 の新機能, 260
-ls オプション
バージョン 12.0.1 の新機能、Mobile Link (mlreplay), 98
バージョン 16.0 の新機能、Mobile Link, 51
-lu オプション
バージョン 16.0 の新機能、Mobile Link, 51
-l オプション
バージョン 11.0.1 の新機能、dbunload, 198
-nba オプション
バージョン 12.0.0 のサポート終了、Mobile Link, 168
-ni オプション
バージョン 11.0.0 の新機能、dblsn, 261
-no オプション
バージョン 11.0.0 の強化、dbunload, 222
-ns オプション
バージョン 16.0 の動作変更, 50
-nu オプション
バージョン 11.0.0 の新機能、dblsn, 261
バージョン 16.0 の動作変更, 51
-n オプション
バージョン 12.0.0 で廃止、dbmlsync, 170
バージョン 12.0.1 の新機能、Mobile Link (mlreplay), 98
バージョン 12.0.1 の動作変更、Mobile Link (mlreplay), 102
-os オプション
バージョン 12.0.1 の新機能、Mobile Link (mlgenreplayapi), 99
バージョン 12.0.1 の新機能、Mobile Link (mlreplay), 99
-ot オプション
バージョン 12.0.1 の動作変更、Mobile Link (mlgenreplayapi), 102
バージョン 12.0.1 の動作変更、Mobile Link (mlreplay), 102
-o オプション
バージョン 12.0.1 の動作変更、Mobile Link (mlgenreplayapi), 102
バージョン 12.0.1 の動作変更、Mobile Link (mlreplay), 102
-pc オプション
バージョン 11.0.0 の新機能、dblsn, 261
-pd オプション
dbinit、バージョン 16.0 の新機能, 29
dbupgrad、バージョン 16.0 の新機能, 30
-ppv オプション
バージョン 11.0.1 の強化、Mobile Link (mlsrv11), 208
バージョン 12.0.0 の強化、Mobile Link, 159
-p オプション
バージョン 16.0 の強化, 31
-qi オプション
バージョン 12.0.1 の動作変更、Mobile Link (mlsrv16), 83
-qr オプション
バージョン 12.0.0 の強化、dbunload ユーティリティ, 130
-rep オプション
バージョン 12.0.1 の新機能、Mobile Link (mlreplay), 98
-rnt オプション
バージョン 12.0.1 の新機能、Mobile Link (mlreplay), 98
-rp オプション
バージョン 12.0.0 の新機能、Mobile Link, 159
バージョン 12.0.1 の新機能、Mobile Link (mlreplay), 99
-rrp オプション
バージョン 12.0.0 の新機能、Mobile Link, 159
-rsu オプション
バージョン 12.0.0 で削除、ログ変換ユーティリティ (dbtran) のオプション, 118
-rs オプション
バージョン 12.0.0 の強化、dbsvc ユーティリティ, 129
-r オプション
バージョン 12.0.0 のサポート終了、Mobile Link, 168
-sbx オプション
バージョン 16.0 の新機能, 16
-sci オプション
バージョン 12.0.1 の動作変更、Mobile Link (mlreplay), 102
バージョン 16.0 の動作変更、Mobile Link (mlreplay), 44
-sf オプション
バージョン 11.0.0 の強化, 224, 239
バージョン 16.0 の強化, 14

-sk オプション
バージョン 16.0 の強化, 14
-sl dnet
バージョン 12.0.0 の動作変更、Mobile Link, 166
-sl java オプション
バージョン 12.0.0 の動作変更、Mobile Link, 159
-sm オプション
バージョン 11.0.0 の新機能, 219
バージョン 11.0.1 の動作変更、Mobile Link, 209
バージョン 12.0.0 の動作変更, 154
-sp オプション
バージョン 11.0.0 の新機能、dbmlsync, 260
-ss オプション
バージョン 12.0.0 の動作変更、Mobile Link, 169
-sv オプション
バージョン 11.0.1 の強化、Mobile Link (dblsn), 208
バージョン 16.0 の新機能、Mobile Link (mlreplay), 44
-s オプション
バージョン 12.0.0 の強化、初期化ユーティリティ (dbinit), 146
バージョン 12.0.0 の新機能、Mobile Link, 161, 162
-tc オプション、Mobile Link (mlsrv11)
バージョン 11.0.0 の新機能, 260
-tdsl オプション
バージョン 16.0 の新機能, 14
-tf オプション、Mobile Link (mlsrv11)
バージョン 11.0.0 の新機能, 260
-ts オプション
バージョン 16.0 の新機能、Mobile Link dblsn オプション, 42
バージョン 16.0 の新機能、Mobile Link dbmlsync オプション, 42
バージョン 16.0 の新機能、Mobile Link mlsrv16 オプション, 42
-t オプション
バージョン 12.0.0 の動作変更、サービスユーティリティ (dbsvc) のオプション, 118
-uc サーバオプション
バージョン 11.0.0 の動作変更, 253
-ud オプション
バージョン 12.0.0 の動作変更, 118
-ufd オプション
バージョン 16.0 の新機能, 20
-uf オプション
バージョン 16.0 の強化, 20
-ui サーバオプション
バージョン 11.0.0 の動作変更, 253
-um オプション
バージョン 11.0.0 の新機能, 224
-u オプション
バージョン 12.0.0 で廃止、dbmlsync, 170
-version
バージョン 11.0.0 の新機能、dbisql, 274
-vi オプション
バージョン 11.0.1 の強化、Mobile Link (mlsrv11), 208
-vk オプション
バージョン 12.0.0 の新機能、Mobile Link, 159
-vm オプション
バージョン 11.0.1 の強化、Mobile Link (mlsrv11), 208
-vq オプション
バージョン 11.0.1 の強化、Mobile Link (mlsrv11), 208
-vR オプション
バージョン 12.0.0 の新機能、Mobile Link, 159
-vU オプション
バージョン 12.0.0 の新機能、Mobile Link, 159
-wa オプション
バージョン 16.0 の新機能, 13
-wb オプション
バージョン 16.0 の新機能, 13
-wc オプション
バージョン 12.0.0 の新機能、データベースサーバ, 127
-wm オプション
バージョン 12.0.1 の新機能、Mobile Link, 97
-wn mlsrv16 オプション
バージョン 16.0 の新機能、Mobile Link, 43
-w オプション
バージョン 12.0.0 の動作変更、サービスユーティリティ (dbsvc) のオプション, 118
バージョン 12.0.1 の動作変更、Mobile Link (mlsrv12), 101
-xa オプション
バージョン 12.0.1 の強化, 75
-xd オプション
バージョン 11.0.1 の新機能、データベースサーバ, 202

-xm オプション
バージョン 12.0.0 の新機能、データベースサーバ, 131
-xo オプション
バージョン 12.0.0 のサポート終了、Mobile Link, 168
-xp オプション
バージョン 11.0.0 の動作変更, 249
バージョン 12.0.0 の動作変更, 149
-xs オプション
バージョン 16.0 の強化, 20
-x オプション
バージョン 12.0.0 の強化、Mobile Link, 158
バージョン 12.0.1 の強化、Mobile Link, 97
-zoc オプション
バージョン 11.0.0 の新機能, 239
-zt オプション
バージョン 12.0.1 の動作変更、SQL Anywhere, 94
-z オプション
バージョン 12.0.1 の新機能, 82

A

a_backup_db 構造体
バージョン 16.0 の動作変更, 36
a_change_log 構造体
バージョン 12.0.0 の動作変更, 117
a_sync_db 構造体
バージョン 16.0 の動作変更, 36
バージョン 16.0 の変更, 52
a_validate_type 列挙
バージョン 11.0.0 の強化, 242
active_cookie オプション
バージョン 12.0.0 の新機能、Relay Server, 164
active_header
バージョン 12.0.0 の新機能、Relay Server, 164
Adaptive Server Enterprise 12.5.x
バージョン 12.0.0 のサポート終了、Mobile Link, 168
Adaptive Server Enterprise、Oracle、MySQL、または Microsoft SQL Server のアップグレード
Mobile Link 統合データベース, 321
Adaptive Server Enterprise サポート
バージョン 12.0.1 の新機能、Mobile Link, 96, 97
addBatch
バージョン 12.0.0 の強化, 141
Address Windowing Extensions
バージョン 12.0.0 で廃止, 153
バージョン 16.0 でサポート終了, 40
ADO.NET
バージョン 11.0.1 の強化, 201
バージョン 12.0.1 の強化, 77, 89
バージョン 16.0 の新機能, 25
ADSODBC サーバクラス
バージョン 11.0.0 の新機能, 242
AES256_FIPS 暗号化アルゴリズム
バージョン 11.0.0 の新機能, 221
AES256 暗号化アルゴリズム
バージョン 11.0.0 の新機能, 221
AES 暗号化アルゴリズム
バージョン 16.0 の強化, 28
allow_read_client_file オプション
バージョン 11.0.0 の新機能, 223
allow_read_client_file プロパティ
バージョン 11.0.0 の新機能, 225
allow_write_client_file オプション
バージョン 11.0.0 の新機能, 223
allow_write_client_file プロパティ
バージョン 11.0.0 の新機能, 225
allowPasswordsInFavorites
バージョン 12.0.0 の新機能, 190
ALLOW 句
バージョン 16.0 の新機能, 35
ALTER DATABASE 文
バージョン 10 以降のデータベースのアップグレード, 307
バージョン 11.0.0 の強化, 230
バージョン 11.0.1 の強化, 200
バージョン 12.0.0 の強化, 139
バージョン 16.0 の強化, 12, 25, 30
バージョン 16.0 の動作変更, 37
ALTER EVENT 文
バージョン 11.0.0 の強化, 232
バージョン 16.0 の強化, 23
バージョン 16.0 の新機能, 11
ALTER EXTERNAL ENVIRONMENT 文
バージョン 12.0.0 の強化, 145
ALTER FUNCTION 文
バージョン 11.0.0 の強化, 230
ALTER LDAP SERVER 文
バージョン 16.0 の新機能, 6
ALTER LOGIN POLICY 文
バージョン 11.0.0 の新機能, 231
バージョン 16.0 の強化, 2

ALTER MATERIALIZED VIEW 文
バージョン 11.0.0 の強化, 230
ALTER MIRROR SERVER 文
バージョン 12.0.0 の新機能, 123
AlternateMirrorServerName プロパティ
バージョン 11.0.0 の新機能, 227
ALTER PROCEDURE 文
バージョン 11.0.0 の強化, 230
バージョン 12.0.0 の動作変更, 118
ALTER ROLE 文
バージョン 16.0 の新機能, 3
ALTER SEQUENCE 文
バージョン 12.0.0 の新機能, 124
ALTER SERVER 文
バージョン 12.0.0 の強化, 136
バージョン 12.0.1 の動作変更, 76
バージョン 16.0 の動作変更, 37
ALTER SPATIAL REFERENCE SYSTEM 文
バージョン 12.0.0 の新機能, 119
ALTER SYNCHRONIZATION SUBSCRIPTION 文
バージョン 12.0.0 の強化, 157
バージョン 12.0.0 の強化、Mobile Link, 161
バージョン 12.0.1 の強化、OnDemand, 74
バージョン 16.0 の強化, 47
ALTER SYNCHRONIZATION USER 文
バージョン 12.0.1 の強化、OnDemand, 74
バージョン 16.0 の強化, 47
ALTER TABLE 文
バージョン 12.0.0 の動作変更, 118
バージョン 12.0.1 の強化, 77
バージョン 16.0 の強化, 23, 28
ALTER TEXT CONFIGURATION 文
バージョン 11.0.0 の新機能, 232
バージョン 12.0.0 の強化, 136
ALTER TEXT INDEX 文
バージョン 11.0.0 の新機能, 232
ALTER TRACE EVENT SESSION 文
バージョン 16.0 の新機能, 24
ALTER USER 文
バージョン 11.0.0 の新機能, 231
バージョン 16.0 の強化, 6
an_unload_db 構造
バージョン 12.0.0 の強化, 140
Android
バージョン 12.0.1 での新規サポート、Ultra Light, 104
バージョン 12.0.1 の強化、Ultra Light, 107
バージョン 12.0.1 の新機能、Android での Ultra Light サポート, 106
Android 上でのアップロードローのカウント
バージョン 16.0 の強化、Ultra Light J, 63
Android 上での暗号化キーの変更
バージョン 16.0 の強化、Ultra Light J, 63
Android 上での再起動可能なダウンロード
バージョン 16.0 の強化、Ultra Light J, 64
Android 上でのデータベース検証
バージョン 16.0 の強化、Ultra Light J, 63
Android 上のシステムイベント
バージョン 16.0 の強化、Ultra Light J, 64
ansi_integer_overflow オプション
バージョン 11.0.0 でサポート終了, 254
ansi_substring オプション
バージョン 11.0.0 でサポート終了, 254
バージョン 11.0.1 で非推奨の取り消し, 207
Apache
バージョン 12.0.1 クイック配備、Relay Server, 84
バージョン 16.0、32 ビット Relay Server のサポート終了, 56
バージョン 16.0 の新機能、Relay Server のクイック設定, 54
AppInfo 接続パラメータ
バージョン 11.0.0 の強化, 220
ApproximateCPUTime プロパティ
バージョン 12.0.1 の新機能, 92
APP 接続パラメータ
バージョン 11.0.0 の強化, 220
ARRAY_AGG 関数
バージョン 16.0 の新機能, 9
ARRAY_MAX_CARDINALITY 関数
バージョン 16.0 の新機能, 9
ARRAY コンストラクタ
バージョン 16.0 の新機能, 9
ASE 15.7 のサポート
バージョン 16.0 の新機能、Mobile Link, 43
ASEJDBC サーバクラス
バージョン 12.0.0 で廃止, 153
バージョン 16.0 でサポート終了, 40
ASP.NET
バージョン 11.0.1 の新機能, 201
authenticate_file_upload 接続イベント
バージョン 12.0.0 の新機能、Mobile Link, 160
Authenticated プロパティ
バージョン 11.0.1 の新機能、接続プロパティ, 197

バージョン 11.0.1 の新機能、データベースプロパティ, 198
AuthType プロパティ
バージョン 11.0.0 の新機能, 225
auto_commit_on_create_local_temp_index オプション
バージョン 16.0 の新機能, 19
バージョン 16.0 の動作変更, 31
auto_commit_on_create_local_temp_index プロパティ
バージョン 16.0 の新機能, 20
auto_commit オプション
バージョン 12.0.0 の動作変更, 192
auto_refetch
バージョン 12.0.0 の動作変更, 192
auto_unlock_time オプション
バージョン 16.0 の新機能, 2
AUTOINCREMENT
reset.sql スクリプトを使用して次に使用可能な値を保持, 294
再構築されたデータベースで次に使用可能な値を保持, 294
automatic_timestamp オプション
バージョン 11.0.0 でサポート終了, 254
AutoMultiProgrammingLevelStatistics プロパティ
バージョン 12.0.0 の新機能, 133
AutoMultiProgrammingLevel プロパティ
バージョン 12.0.0 の新機能, 133
AWE
バージョン 12.0.0 で廃止, 153
バージョン 16.0 でサポート終了, 40

B

background_priority オプション
バージョン 11.0.0 で廃止予定, 256
BACKUP 権限
バージョン 16.0 で廃止, 38
BACKUP 文
バージョン 11.0.1 の強化, 203
バージョン 12.0.0 の強化, 128
BatchUpdateCount
バージョン 12.0.0 の強化, 141
BatchUpdateException
バージョン 12.0.0 の強化, 141
BEGIN SNAPSHOT 文
バージョン 11.0.0 の新機能, 232
BEGIN 文
バージョン 12.0.0 の強化, 137
bell オプション
バージョン 12.0.0 の動作変更, 192
BIGINT UNSIGNED データ型
バージョン 16.0 の新機能, 25
BIGINT データ型
バージョン 12.0.0 の動作変更、Mobile Link, 166
bin32 ディレクトリ
バージョン 11.0.0 の動作変更, 282
bin64 ディレクトリ
バージョン 11.0.0 の動作変更, 282
BINSEARCH プロトコルオプション
バージョン 11.0.0 でサポート終了, 254
BINTOHEX 関数
バージョン 16.0 の新機能, 23
BIT_AND 関数
バージョン 12.0.0 の強化, 135
BIT_OR 関数
バージョン 12.0.0 の強化, 135
BIT_XOR 関数
バージョン 12.0.0 の強化, 135
BlackBerry
バージョン 11.0.0 の新機能、BlackBerry での Ultra Light サポート, 269
BlackBerry Enterprise Server
バージョン 12.0.1 の強化, 82
BLOB
バージョン 11.0.0 の新機能、問い合わせ, 219
blocking
バージョン 12.0.0 の強化, 130
blocking_others_timeout オプション
バージョン 12.0.0 の新機能, 130
Broadcast Repeater ユーティリティ (dbns12)
バージョン 12.0.0 の動作変更, 118
BufferDownload 拡張オプション
バージョン 12.0.0 の新機能、Mobile Link, 163
BytesReceivedUncomp プロパティ
バージョン 12.0.1 の新機能, 92
BytesReceived プロパティ
バージョン 12.0.1 の新機能, 92
BytesSentUncomp プロパティ
バージョン 12.0.1 の新機能, 92
BytesSent プロパティ
バージョン 12.0.1 の新機能, 92

C

C++ API 移行ウィザード
バージョン 11.0.0 でサポート終了, 269
CacheHitsEng プロパティ
バージョン 11.0.0 で名前が変更, 252
CacheHits プロパティ
バージョン 11.0.0 の新機能, 252
CacheReadEng プロパティ
バージョン 11.0.0 で名前が変更, 252
CacheReadWorkTable プロパティ
バージョン 11.0.0 の新機能, 225, 227
CacheRead プロパティ
バージョン 11.0.0 の新機能, 252
CALL 文
バージョン 12.0.0 で廃止、関数の呼び出し, 153
CancelSync メソッド
バージョン 12.0.0 の新機能、Mobile Link, 163
CARDINALITY 関数
バージョン 16.0 の新機能, 9
CarverHeapPages プロパティ
バージョン 12.0.1 の新機能, 92
CASE 式
バージョン 11.0.0 の強化, 233
バージョン 11.0.0 の強化、Ultra Light, 268
CASE 文
バージョン 11.0.0 の強化, 233
バージョン 11.0.0 の強化、Ultra Light, 268
バージョン 12.0.0 の強化, 137
CATALOGS ローセット、OLE DB
バージョン 11.0.1 の新機能, 201
certificate_password プロトコルオプション
バージョン 11.0.0 でサポート終了, 281
バージョン 11.0.0 の動作変更, 262
certificate プロトコルオプション
バージョン 11.0.0 でサポート終了, 281
バージョン 11.0.0 の動作変更, 262
CESU8BIN 照合
バージョン 16.0 の強化, 28
CESU-8 文字セット
バージョン 16.0 の強化, 28
change_password_dual_control オプション
バージョン 16.0 の新機能, 13
CHAR データ型
バージョン 12.0.0 の強化, 139
CheckpointLogBitmapPagesWritten プロパティ
バージョン 12.0.0 でサポート対象外, 153
CheckpointLogBitmapSize プロパティ
バージョン 12.0.0 でサポート対象外, 153
checkpoint メソッド
バージョン 16.0 で削除された API、Ultra Light, 66
CHECKSUM 句
バージョン 12.0.0 の強化, 139
バージョン 12.0.0 の強化、ALTER DATABASE 文, 127
バージョン 12.0.0 の強化、CREATE DATABASE 文, 146
バージョン 12.0.0 の強化、START DATABASE 文, 127
ClassLoader
バージョン 12.0.1 の強化, 89
clearBatch
バージョン 12.0.0 の強化, 141
CLEAR 文
バージョン 12.0.0 の動作変更, 191
ClientNodeAddress プロパティ
バージョン 11.0.0 の新機能, 225
ClientStmtCacheHits プロパティ
バージョン 12.0.1 の新機能, 92
ClientStmtCacheMisses プロパティ
バージョン 12.0.1 の新機能, 92
collect_network_data プロトコルオプション
バージョン 12.0.1 の新機能、Mobile Link, 83
CommBufferSize 接続パラメータ
バージョン 16.0 の強化, 31
COMMENT 文
バージョン 11.0.0 の強化, 233
バージョン 11.0.1 で廃止予定の句, 207
バージョン 12.0.0 の強化, 123, 124
バージョン 16.0 の強化, 6
commit_on_exit オプション
バージョン 12.0.0 の動作変更, 192
Commit プロパティ
バージョン 12.0.1 の新機能, 93
COMMIT 文
バージョン 12.0.0 の強化, 189
Common Access Card
バージョン 11.0.1 の新機能, 208
Common Access Card を使用したクライアント認証
バージョン 11.0.1 の新機能, 208
Compact Framework 3.5 のサポート
バージョン 16.0 の新機能, 26

ConfigPersistent.setRowScoreFlushSize メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 82
ConfigPersistent.setRowScoreMaximum メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 82
conflicted_deletes プロパティ
バージョン 16.0 で削除、Mobile Link, 53
conflicted_inserts プロパティ
バージョン 16.0 で削除、Mobile Link, 53
ConnectedTime プロパティ
バージョン 12.0.1 の新機能, 91
CONNECTION_EXTENDED_PROPERTY 関数
バージョン 16.0 の動作変更, 33
CONNECTION_PROPERTY 関数
バージョン 16.0 の動作変更, 33
Connection.getNewUUID メソッド [Ultra Light for M-Business Anywhere]
バージョン 12.0.1 の新機能, 82
CONNECTION CLOSE 句
バージョン 12.0.1 の動作変更, 76
バージョン 16.0 でサポート終了, 37
ConnectionPool 接続パラメータ
バージョン 12.0.0 の新機能, 128
Connect メソッド
バージョン 16.0 の新機能、Dbmlsync .NET API, 46
バージョン 16.0 の新機能、Dbmlsync C++ API, 46
ConnPoolCachedCount プロパティ
バージョン 12.0.0 の新機能, 132
ConnPoolHits プロパティ
バージョン 12.0.0 の新機能, 132
ConnPoolMisses プロパティ
バージョン 12.0.0 の新機能, 132
CONTAINS 探索条件
バージョン 11.0.0 の新機能, 231
バージョン 16.0 の強化, 28
CONVERT 関数
バージョン 12.0.1 の動作変更, 95
cookie
Relay Server, 54
cooperative_commit_timeout オプション
バージョン 11.0.0 の動作変更, 249
cooperative_commits オプション
バージョン 11.0.0 の動作変更, 249
CopyNodeParent
バージョン 16.0 の新機能, 12
COUNT_BIG 関数
バージョン 12.0.0 の新機能, 136
Counter1 統計情報
バージョン 12.0.1 の新機能, 93
Counter2 統計情報
バージョン 12.0.1 の新機能, 93
Counter3 統計情報
バージョン 12.0.1 の新機能, 93
Counter4 統計情報
バージョン 12.0.1 の新機能, 93
Counter5 統計情報
バージョン 12.0.1 の新機能, 93
CPOOL 接続パラメータ
バージョン 12.0.0 の新機能, 128
CPU
バージョン 16.0 の動作変更, 32
CRC32 アルゴリズム
バージョン 12.0.0 の強化, 134
CREATE CERTIFICATE 文
バージョン 16.0 の新機能, 23
createcert ユーティリティ
バージョン 16.0 の強化, 19
CREATE DATABASE 文
バージョン 16.0 の強化, 25, 28, 29
バージョン 16.0 の動作変更, 32
CREATE DECRYPTED DATABASE 文
バージョン 11.0.1 の新機能, 199
バージョン 16.0 の強化, 25
バージョン 16.0 の動作変更, 32
CREATE DECRYPTED FILE 文
バージョン 16.0 の強化, 25
バージョン 16.0 の動作変更, 32
CREATE ENCRYPTED DATABASE 文
バージョン 11.0.1 の新機能, 199
バージョン 16.0 の強化, 25
バージョン 16.0 の動作変更, 32
CREATE ENCRYPTED FILE 文
バージョン 16.0 の強化, 25
バージョン 16.0 の動作変更, 32
CREATE ENCRYPTED TABLE 文
バージョン 16.0 の動作変更, 32
CREATE EVENT 文
バージョン 11.0.0 の強化, 234
バージョン 16.0 の強化, 23
バージョン 16.0 の新機能, 11
CREATE EXISTING TABLE 文
バージョン 12.0.1 の強化, 88
CREATE EXTERNLOGIN 文

バージョン 16.0 の動作変更, 32
CREATE FUNCTION 文
バージョン 11.0.1 の強化, 200
バージョン 12.0.0 の強化, 138
バージョン 16.0 の強化, 23
CREATE INDEX 文
バージョン 12.0.0 の強化, 136, 187
バージョン 16.0 の強化, 23
createkey ユーティリティ
バージョン 11.0.0 の新機能, 261
CREATE LDAP SERVER 文
バージョン 16.0 の新機能, 6
CREATE LOGIN POLICY 文
バージョン 11.0.0 の新機能, 231
バージョン 16.0 の強化, 2, 6
CREATE MATERIALIZED VIEW 文
バージョン 11.0.0 の強化, 233
CREATE MIRROR SERVER 文
バージョン 12.0.0 の新機能, 123
CREATE PROCEDURE 文
バージョン 11.0.1 の強化, 200
バージョン 12.0.0 の強化, 138
バージョン 12.0.0 の動作変更, 150
バージョン 12.0.1 の強化, 88
バージョン 16.0 の強化, 23
CREATE PUBLICATION 文
バージョン 12.0.0 の強化, 137
CREATE ROLE 文
バージョン 16.0 の新機能, 3
CREATE SEQUENCE 文
バージョン 12.0.0 の新機能, 124
CREATE SERVER 文
バージョン 12.0.0 の強化, 136
バージョン 12.0.1 の強化, 87
バージョン 16.0 の強化, 23
CREATE SPATIAL REFERENCE SYSTEM 文
バージョン 12.0.0 の新機能, 119
CREATE SPATIAL UNIT OF MEASURE 文
バージョン 12.0.0 の新機能, 119
CREATE SYNCHRONIZATION PROFILE 文
バージョン 12.0.0 の新機能, 136
CREATE SYNCHRONIZATION SUBSCRIPTION 文
バージョン 12.0.0 の強化, 157
バージョン 12.0.0 の強化、Mobile Link, 161
CREATE TABLE 文
バージョン 11.0.1 の強化, 200
バージョン 12.0.0 の強化, 137
CREATE TEMPORARY TRACE EVENT SESSION 文
バージョン 16.0 の新機能, 24
CREATE TEMPORARY TRACE EVENT 文
バージョン 16.0 の新機能, 24
CREATE TEXT CONFIGURATION 文
バージョン 11.0.0 の新機能, 232
CREATE TEXT INDEX 文
バージョン 11.0.0 の新機能, 232
バージョン 12.0.0 の強化, 137
バージョン 16.0 の強化, 23
CREATE TRIGGER 文
バージョン 11.0.1 の強化, 200
バージョン 12.0.0 の強化, 138
バージョン 16 へのアップグレードのトラブルシューティング, 300
CREATE USER 文
バージョン 11.0.0 の新機能, 231
CREATE VARIABLE 文
バージョン 12.0.0 の強化, 137, 138
CREATE VIEW 文
バージョン 11.0.1 の強化, 200
バージョン 12.0.0 の強化, 138
バージョン 16.0 の動作変更, 30, 37
CREATE 文
バージョン 16.0 の強化, 3
CROSS APPLY 句
バージョン 11.0.0 の新機能, 233
current_download_row_count
バージョン 16.0 の強化、Ultra Light.NET, 61
バージョン 16.0 の強化、Ultra Light C/C++, 61
CurrentMultiProgrammingLevel プロパティ
バージョン 12.0.0 の新機能, 133
CURRENT UTC TIMESTAMP 空間値
バージョン 12.0.0 の動作変更, 148
CursorOpen プロパティ
バージョン 12.0.1 の新機能, 93
Cursor プロパティ
バージョン 12.0.1 の新機能, 93
CustDB
バージョン 11.0.0 の動作変更、Ultra Light, 270
CustDB Ultra Light サンプル
バージョン 11.0.0 の動作変更, 270
CustDB サンプル
バージョン 16.0 で削除された機能、Ultra Light, 65
バージョン 16.0 の強化、Ultra Light J, 64

D

DatabaseInfo.getNumberRowsToUpload メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 81
DatabaseInfo.getPageSize メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 81
DatabaseInfo.getRelease メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 81
DatabaseManager.createFileTransfer メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 78
DatabaseMetaData
バージョン 16.0 の動作変更, 36
DatabaseName 接続パラメータ
バージョン 12.0.1 の動作変更、OnDemand, 74
DATEADD 関数
バージョン 12.0.0 の強化, 135
DATEDIFF 関数
バージョン 12.0.0 の強化, 135
バージョン 12.0.0 の動作変更, 151
DATENAME 関数
バージョン 12.0.0 の強化, 135
DATEPART 関数
バージョン 12.0.0 の強化, 135
DATETIMEOFFSET データ型
バージョン 12.0.0 の新機能, 139
DateTimeWithTimeZone クラス
バージョン 12.0.0 の新機能、Mobile Link, 160
DB_EXTENDED_PROPERTY 関数
バージョン 12.0.0 の強化, 135
バージョン 12.0.1 の強化, 86
バージョン 16.0 の動作変更, 34
DB_NAME 関数
バージョン 16.0 の動作変更, 34
DB_PROPERTY 関数
バージョン 16.0 の動作変更, 34
db_publisher プロパティ
バージョン 16.0 の新機能, 3
DBA
バージョン 12.0.1 の強化, 75
dBase フォーマット、INPUT 文
バージョン 11.0.0 で削除、サポート, 277
dBase フォーマット、OUTPUT 文
バージョン 11.0.0 で削除、サポート, 277
DBA 権限
バージョン 16.0 で廃止, 38
dbbackup ユーティリティ
バージョン 16.0 の強化, 13
dbcapi.dll
バージョン 11.0.0 の新機能, 234
dbconsole ユーティリティ
バージョン 11.0.0 の強化, 279
dbeng12
バージョン 12.0.0 の動作変更, 118, 148
バージョン 12.0.1 の動作変更, 94
dbfhide ユーティリティ
バージョン 12.0.0 の強化, 129
dbinfo ユーティリティ
バージョン 12.0.0 の強化, 146
dbinit ユーティリティ
バージョン 12.0.0 の動作変更, 152
バージョン 16.0 の強化, 28, 29
dbisql.exe
バージョン 11.0.0 の動作変更, 278
dbisqlc ユーティリティ
バージョン 11.0.0 の動作変更, 278
バージョン 11.0.1 の動作変更, 207
dbisql ユーティリティ
バージョン 11.0.0 の動作変更, 253
バージョン 16.0 の新機能, 69
dblgen12.dll レジストリエントリ
バージョン 12.0.0 の強化, 146
DBLIB
バージョン 11.0.0 の新機能, 247
dblibtb.dll
バージョン 11.0.0 の新機能, 247
dblibtw.dll
バージョン 11.0.0 の新機能, 247
dblic ユーティリティ
バージョン 12.0.0 の強化, 129
バージョン 12.0.1 の強化, 95
dblog ユーティリティ
バージョン 12.0.0 の動作変更, 118
dbltm ユーティリティ
バージョン 12.0.0 でサポートされない, 117
バージョン 12.0.0 の動作変更, 118
dbmlsync -sp オプション
バージョン 11.0.0 の新機能, 260
dbmlsync StreamCompression 拡張オプション
バージョン 11.0.0 でサポート終了, 262
dbmlsync 統合コンポーネント
バージョン 11.0.0 で廃止予定, 262
バージョン 12.0.0 での削除, 170
dbmlsync メッセージログ
バージョン 12.0.1 の強化、Mobile Link, 83, 84

dbmlsync メッセージログスキャン
バージョン 11.0.0 の強化, 261
dbmlsync ユーティリティ
バージョン 12.0.0 の動作変更, 118
dbmodenv
バージョン 11.0.1 の動作変更, 206
dbns12 ユーティリティ
バージョン 12.0.0 の動作変更, 118
dbremote ユーティリティ
バージョン 12.0.0 の動作変更, 118
DBSC_CancelRet 列挙
バージョン 12.0.0 の新機能、Mobile Link, 163
DBSC_ERR_ACTIVE_SYNC_NOT_CANCELED メンバー
バージョン 12.0.0 の新機能、Mobile Link, 163
DBSC_ERR_DEAD_SERVER メンバー
バージョン 12.0.0 の新機能、Mobile Link, 163
DBSC_EVENTTYPE_DOWNLOAD_COMMITTED
バージョン 16.0 の新機能、Dbmlsync .NET API, 46
バージョン 16.0 の新機能、Dbmlsync C++ API, 46
DBSC_EVENTTYPE_DOWNLOAD_START イベント
バージョン 16.0 の新機能、Dbmlsync .NET API, 47
バージョン 16.0 の新機能、Dbmlsync C++ API, 46
DBSC_EVENTTYPE_UPLOAD_SENT イベント
バージョン 16.0 の新機能、Dbmlsync .NET API, 47
バージョン 16.0 の新機能、Dbmlsync C++ API, 46
DBSC_EVENTTYPE_UPLOAD_START イベント
バージョン 16.0 の新機能、Dbmlsync .NET API, 47
バージョン 16.0 の新機能、Dbmlsync C++ API, 46
DBSC_Event 構造体
バージョン 16.0 の新機能、Dbmlsync .NET API, 46
バージョン 16.0 の新機能、Dbmlsync C++ API, 46
dbspawn ユーティリティ
バージョン 16.0 の強化, 19
dbsrv12
バージョン 12.0.0 の動作変更, 118
dbstats ユーティリティ
バージョン 12.0.0 の新機能, 144
dbsupport ユーティリティ
バージョン 11.0.0 の強化, 280
バージョン 12.0.0 の強化, 129
バージョン 12.0.0 の動作変更, 118
バージョン 16.0 の動作変更, 33
dbsvc ユーティリティ
バージョン 11.0.1 の強化, 198
バージョン 12.0.0 の強化, 129
バージョン 12.0.0 の新機能, 144
バージョン 12.0.0 の動作変更, 118, 154
バージョン 16.0 の動作変更, 10
dbtlstb.dll
バージョン 11.0.0 の新機能, 247
dbtlstw.dll
バージョン 11.0.0 の新機能, 247
dbtran ユーティリティ
バージョン 11.0.0 の強化, 223
バージョン 12.0.0 の動作変更, 118
DBTYPE_DBTIME
Borland Delphi, 40
DBTYPE_DBTIME2
バージョン 16.0 の強化, 26
DBTYPE_DBTIMESTAMPOFFSET
Borland Delphi, 40
, 88
dbunload ユーティリティ
バージョン 10 以降のデータベースの再構築, 296
バージョン 9 以前のデータベースの再構築, 302
バージョン 11.0.0 の強化, 222
バージョン 11.0.0 の動作変更, 251
バージョン 11.0.1 の強化, 198
バージョン 16.0 の強化, 19
バージョン 16 へのアップグレードのトラブルシューティング, 300
dbupgrad ユーティリティ
バージョン 10 以降のデータベースのアップグレード, 306
バージョン 11.0.1 の動作変更, 206
バージョン 16.0 の強化, 12, 30
dbvalid ユーティリティ
バージョン 11.0.0 の強化, 223
dbxtract ユーティリティ
バージョン 11.0.0 の強化, 263
バージョン 11.0.0 の動作変更, 251
バージョン 12.0.0 の強化, 129, 170

DB 領域
バージョン 11.0.0 の動作変更, 248, 252
DCX
バージョン 11.0.1 の新機能, 213
バージョン 12.0.0 の強化, 195
バージョン 12.0.1 の強化, 116
Deadlock システムイベント
バージョン 11.0.0 の新機能, 243
DECLARE CURSOR 文
バージョン 12.0.0 の動作変更, 150
DECLARE LOCAL TEMPORARY TABLE 文
バージョン 16 へのアップグレードのトラブルシューティング, 300
DECLARE 文
バージョン 12.0.0 の強化, 137
DECRYPT 関数
バージョン 11.0.0 の強化, 221
バージョン 16.0 の強化, 14
バージョン 16.0 の動作変更, 32
defaultShowMultipleResultSets
バージョン 16.0 の新機能, 69
defaultShowResultsForAllStatements
バージョン 16.0 の新機能, 69
DEFAULT VALUES 句
バージョン 11.0.1 の新機能, 199
delete_old_logs オプション
バージョン 12.0.0 の動作変更, 118
DELETE 文
バージョン 11.0.0 の強化, 234
バージョン 11.0.0 の動作変更, 251
バージョン 11.0.1 の強化, 202
バージョン 12.0.0 の強化, 139
バージョン 12.0.0 の動作変更, 147
バージョン 12.0.1 の強化, 88
Delphi (Borland)
バージョン 16.0 でサポート終了, 40
demo.db
バージョン 11.0.0 の強化, 281
Deployment ウィザード
バージョン 11.0.0 の強化, 275
DER コード化されたキー
バージョン 12.0.1 の新機能、Mobile Link, 97
DESCRIBE 文
バージョン 11.0.0 の強化, 277
バージョン 16.0 の動作変更, 32
disk_sandbox オプション
バージョン 16.0 の新機能, 16
disk_sandbox プロパティ
バージョン 16.0 の新機能, 18
DiskReadEng プロパティ
バージョン 11.0.0 で名前が変更, 252
DiskReadHintPages プロパティ
バージョン 11.0.0 の新機能, 252
DiskReadHintScatterLimit プロパティ
バージョン 11.0.0 の新機能, 252
DiskReadHint プロパティ
バージョン 11.0.0 の新機能, 252
DiskReadWorkTable プロパティ
バージョン 11.0.0 の新機能, 225
DiskRead プロパティ
バージョン 11.0.0 の新機能, 252
DiskRetryReadScatter
バージョン 11.0.0 の新機能, 226
DiskRetryReadScatter プロパティ
バージョン 11.0.0 の新機能, 227
DiskRetryRead プロパティ
バージョン 11.0.0 の新機能, 226
DiskRetryWrite プロパティ
バージョン 11.0.0 の新機能, 226
DiskSandbox プロパティ
バージョン 16.0 の新機能, 18
DiskSyncRead プロパティ
バージョン 11.0.0 の新機能, 225
DiskSyncWrite プロパティ
バージョン 11.0.0 の新機能, 225
DiskWaitRead プロパティ
バージョン 11.0.0 の新機能, 225
DiskWaitWrite プロパティ
バージョン 11.0.0 の新機能, 225
DiskWriteHintPages プロパティ
バージョン 11.0.0 の新機能, 225
DiskWriteHint プロパティ
バージョン 11.0.0 の新機能, 225
divide_by_zero_error オプション
バージョン 11.0.0 でサポート終了, 254
バージョン 12.0.0 でサポート, 150
DLL プロトコルオプション
バージョン 11.0.0 でサポート終了, 256
DML
バージョン 12.0.0 の新機能、選択, 126
DML から選択
バージョン 12.0.0 の新機能, 126
DocCommentXchange (DCX)
バージョン 11.0.1 の新機能, 213
DriveBus プロパティ
バージョン 12.0.0 の新機能, 132

DriveModel プロパティ
バージョン 12.0.0 の新機能, 132
Driver 接続パラメータ
バージョン 11.0.0 の強化, 237
DropBadStatistics プロパティ
バージョン 12.0.0 の新機能, 134
DROP CERTIFICATE 文
バージョン 16.0 の新機能, 23
DROP DATABASE 文
バージョン 16.0 の強化, 25
バージョン 16.0 の動作変更, 32
DROP EVENT 文
バージョン 11.0.0 の強化, 234
バージョン 11.0.1 の強化, 200
バージョン 12.0.0 の強化, 137
DROP FUNCTION 文
バージョン 11.0.1 の強化, 200
バージョン 12.0.0 の強化, 137
DROP INDEX 文
バージョン 12.0.0 の強化, 137
DROP LDAP SERVER 文
バージョン 16.0 の新機能, 6
DROP LOGIN POLICY 文
バージョン 11.0.0 の新機能, 231
DROP MATERIALIZED VIEW 文
バージョン 11.0.1 の強化, 200
バージョン 12.0.0 の強化, 137
DROP MIRROR SERVER 文
バージョン 12.0.0 の新機能, 123
DROP PROCEDURE 文
バージョン 11.0.1 の強化, 200
バージョン 12.0.0 の強化, 137
DROP PUBLICATION 文
バージョン 12.0.0 の強化, 137
DROP REMOTE CONNECTION 文
バージョン 16.0 の新機能, 24
DROP ROLE 文
バージョン 16.0 の新機能, 4
DROP SEQUENCE 文
バージョン 12.0.0 の新機能, 124
DROP SPATIAL REFERENCE SYSTEM 文
バージョン 12.0.0 の新機能, 119
DROP SPATIAL UNIT OF MEASURE 文
バージョン 12.0.0 の新機能, 119
DROP TABLE 文
バージョン 11.0.1 の強化, 200
バージョン 12.0.0 の強化, 137
DROP TEXT CONFIGURATION 文
バージョン 11.0.0 の新機能, 232
DROP TEXT INDEX 文
バージョン 11.0.0 の新機能, 232
DROP TRACE EVENT SESSION 文
バージョン 16.0 の新機能, 24
DROP TRACE EVENT 文
バージョン 16.0 の新機能, 24
DROP TRIGGER 文
バージョン 11.0.1 の強化, 200
バージョン 12.0.0 の強化, 137
DropUnusedStatistics プロパティ
バージョン 12.0.0 の新機能, 134
DROP USER 文
バージョン 11.0.0 の新機能, 231
DROP VARIABLE 文
バージョン 11.0.1 の強化, 200
バージョン 12.0.0 の強化, 137
DROP VIEW 文
バージョン 11.0.1 の強化, 200
バージョン 12.0.0 の強化, 137

E

e2ee_type プロトコルオプション
バージョン 16.0 で廃止、Mobile Link, 53
ECC オプション
バージョン 16.0 で廃止, 7
ECC サポート
バージョン 16.0 で削除された機能、Relay Server, 56
バージョン 16.0 で削除された機能、Ultra Light, 65
ECC 証明書
バージョン 16.0 で廃止, 7
echo オプション
バージョン 12.0.0 の動作変更, 192
Elevate 接続パラメータ
バージョン 11.0.0 の新機能, 220
embedded SQL
バージョン 12.0.0 の動作変更, 150
Embedded SQL のインポートライブラリ
バージョン 11.0.0 の新機能, 247
EnableFlush レジストリエントリ
バージョン 12.0.0 の新機能, 146
enable status プロパティ
バージョン 12.0.0 の新機能、Mobile Link, 163
ENAME プロトコルオプション
バージョン 11.0.0 でサポート終了, 254

encrypt_aes_random_iv オプション
バージョン 11.0.0 でサポート終了, 256
ENCRYPT_PASSWORD 接続プロパティ
バージョン 11.0.0 の新機能, 222
ENCRYPTION 接続パラメータ
バージョン 16.0 で廃止, 7
ENCRYPT 関数
バージョン 11.0.0 の強化, 221
バージョン 16.0 の強化, 14
バージョン 16.0 の動作変更, 32
end_synchronization スクリプト
バージョン 16.0 の動作変更, 49
EngineName 接続パラメータ
バージョン 12.0.0 で廃止, 154
ENG 接続パラメータ
バージョン 12.0.0 で廃止, 154
Entity Framework 5.0 のサポート
バージョン 16.0 の新機能, 26
Entity Framework サポート
バージョン 11.0.1 の新機能, 201
バージョン 12.0.1 の新機能, 77
environment.plist
バージョン 11.0.1 の動作変更, 206
error_handler
バージョン 16.0 の動作変更, 50
ER 図
バージョン 16.0 の強化, 68
Escape 接続パラメータ
バージョン 12.0.0 の新機能, 128
[Esc] キー
バージョン 12.0.0 の動作変更, 191
ESTIMATE_SOURCE 関数
バージョン 12.0.0 の強化, 145
ETD ファイル
バージョン 16.0 の新機能, 10
EVENT_PARAMETER 関数
バージョン 11.0.0 の強化, 229
EventTypeDesc
バージョン 11.0.0 の新機能, 226
EventTypeName
バージョン 11.0.0 の新機能, 226
EXCEL フォーマット、INPUT 文
バージョン 11.0.0 で削除、サポート, 277
EXCEL フォーマット、OUTPUT 文
バージョン 11.0.0 で削除、サポート, 277
EXCEPT 句
バージョン 11.0.0 の強化, 234
バージョン 11.0.0 の動作変更, 251
executeBatch
バージョン 12.0.0 の強化, 141
ExtendedName プロトコルオプション
バージョン 11.0.0 でサポート終了, 254
extern_login_credentials オプション
バージョン 16.0 の新機能, 19
extern_login_credentials プロパティ
バージョン 16.0 の新機能, 20

F

FAQ
バージョン 12.0.0 の新機能, 195
FileTransferProgressData インタフェース [Ultra Light J]
バージョン 12.0.1 の強化, 78
FileTransferProgressListener インタフェース [Ultra Light J]
バージョン 12.0.1 の強化, 78
FileTransfer インタフェース [Ultra Light J]
バージョン 12.0.1 の強化, 78
fill_sqlda_ex 関数
バージョン 12.0.0 の新機能, 142
FiniIdentifySimulatedClient コールバック
バージョン 12.0.1 の動作変更、Mobile Link (mlgenreplayapi), 102
FIPS
バージョン 11.0.0 の強化, 221
バージョン 12.0.0 の強化, 128
FIRST_VALUE 関数
バージョン 11.0.1 の強化, 199
float_as_double オプション
バージョン 11.0.0 でサポート終了, 254
FORCE NO OPTIMIZATION 句
バージョン 11.0.1 の新機能, 202
FORMAT 句
バージョン 11.0.0 で廃止、FORMAT ASCII, 253
FOXPRO フォーマット、INPUT 文
バージョン 11.0.0 で削除、サポート, 277
FOXPRO フォーマット、OUTPUT 文
バージョン 11.0.0 で削除、サポート, 277
Framework 4.5 のサポート
バージョン 16.0 の新機能, 26
FROM 句
バージョン 11.0.0 の強化, 216
バージョン 12.0.0 の強化, 126
バージョン 12.0.1 の強化, 76

G

gencert ユーティリティ
バージョン 11.0.0 で削除, 281
generate_next_last_download_timestamp スクリプト
バージョン 12.0.0 の新機能、Mobile Link, 160
getAutoCheckpoint メソッド
バージョン 16.0 で削除された API、Ultra Light, 66
getDatabaseKey メソッド
バージョン 16.0 の動作変更、Ultra Light, 66
getLastDownloadTime
バージョン 11.0.1 の強化, 209
getPageReads() メソッド
バージョン 11.0.1 の強化、Ultra Light J, 211
getPageWrites メソッド
バージョン 11.0.1 の強化、Ultra Light J, 211
getParameterCount メソッド
バージョン 16.0 の強化、Ultra Light J, 62
getParameterType メソッド
バージョン 16.0 の強化、Ultra Light J, 62
getParameter メソッド
バージョン 16.0 の強化、Ultra Light J, 62
getSendColumnNames メソッド
バージョン 16.0 の動作変更、Ultra Light, 8
getStreamErrorParameters メソッド
バージョン 11.0.1 の強化、Ultra Light, 210
getTables
バージョン 16.0 の動作変更, 36
getUpdateCounts
バージョン 12.0.0 の強化, 141
GRANT CONNECT 文
バージョン 16.0 の動作変更, 32
GRANT PUBLISH 文
バージョン 16.0 の動作変更, 4
GRANT ROLE SYS_REPLICATION_ADMIN_ROLE 文
バージョン 16.0 の新機能, 4
GRANT ROLE SYS_RUN_REPLICATION_ROLE 文
バージョン 16.0 の新機能, 4
GRANT 文
バージョン 12.0.0 の強化, 124
GUID 識別子
バージョン 11.0.0 の強化、Ultra Light, 266
GV_$TRANSACTION Oracle システムビュー
バージョン 12.0.0 の動作変更、Mobile Link, 166

H

HANAODBC サーバクラス
バージョン 16.0 の新機能, 11
HANA リモートデータアクセス
バージョン 16.0 の新機能, 11
HasEndianSwapFix プロパティ
バージョン 11.0.0 の新機能, 227
HASH 関数
バージョン 12.0.0 の強化, 134
HasTornWriteFix プロパティ
バージョン 12.0.0 の新機能, 132
header_limit オプション
バージョン 16.0 の新機能、Mobile Link, 43
HeapsCarver プロパティ
バージョン 12.0.1 の新機能, 92
HeapsLocked プロパティ
バージョン 12.0.1 の新機能, 92
HeapsQuery プロパティ
バージョン 12.0.1 の新機能, 92
HeapsRelocatable プロパティ
バージョン 12.0.1 の新機能, 92
HEXTOBIN 関数
バージョン 16.0 の新機能, 23
Host 接続パラメータ
バージョン 12.0.0 の新機能, 124
host プロトコルオプション
バージョン 12.0.0 の動作変更, 148
HTTP
バージョン 11.0.0 の動作変更, 249
http_buffer_responses プロトコルオプション
バージョン 12.0.0 の新機能、Mobile Link, 161
http_connection_pool_basesize オプション
バージョン 12.0.0 の新機能, 130
http_connection_pool_timeout オプション
バージョン 12.0.0 の新機能, 130
http_password オプション
バージョン 12.0.1 の新機能、Relay Server, 100
http_proxy_password オプション
バージョン 12.0.1 の新機能、Relay Server, 100
http_proxy_userid オプション
バージョン 12.0.1 の新機能、Relay Server, 100
HTTP_RESPONSE_HEADER 関数
バージョン 12.0.0 の新機能, 135
http_userid オプション
バージョン 12.0.1 の新機能、Relay Server, 100
HTTP_VARIABLE 関数
バージョン 12.0.0 の新機能, 135

HttpAddresses プロパティ
バージョン 11.0.0 の新機能, 226
HttpConnPoolCachedCount プロパティ
バージョン 12.0.0 の新機能, 132
HttpConnPoolHits プロパティ
バージョン 12.0.0 の新機能, 132
HttpConnPoolMisses プロパティ
バージョン 12.0.0 の新機能, 132
HttpConnPoolSteals プロパティ
バージョン 12.0.0 の新機能, 132
HttpNumActiveReq プロパティ
バージョン 11.0.0 の新機能, 226
HttpNumConnections プロパティ
バージョン 11.0.0 の新機能, 226
HttpNumSessions プロパティ
バージョン 11.0.0 の新機能, 226
HTTPS
バージョン 11.0.0 の強化, 240
HttpsAddresses プロパティ
バージョン 11.0.0 の新機能, 226
HttpsNumActiveReq プロパティ
バージョン 11.0.0 の新機能, 226
HttpsNumConnections プロパティ
バージョン 11.0.0 の新機能, 226
HTTP クライアントのチャンクモードの転送コーディング
バージョン 11.0.1 の動作変更, 207
HTTP サーバ
バージョン 16.0 の動作変更, 31
HTTP サービス
バージョン 12.0.0 の強化, 140
HTTP 認証
バージョン 16.0 の新機能、Ultra Light, 58
HTTP のパフォーマンスの向上
バージョン 16.0 の新機能、Mobile Link, 46
HTTP メッセージングシステム
バージョン 12.0.1 の動作変更, 77

I

iAnywhere JDBC ドライバ
バージョン 12.0.0 で廃止, 154
iAnywhere Solutions 12 - Oracle ODBC ドライバ
バージョン 12.0.0 の強化, 158
iAnywhere Solutions Oracle ドライバ
バージョン 11.0.0 の強化, 258
バージョン 16.0 の強化, 7
iastorv レジストリエントリ
バージョン 12.0.0 の新機能, 146
iastor レジストリエントリ
バージョン 12.0.0 の新機能, 146
IBM DB2 LUW
バージョン 16.0 の新機能、Mobile Link, 43
IBM DB2 LUW 8.2
バージョン 12.0.0 のサポート終了、Mobile Link, 168
IBM DB2 LUW 9.5 のサポート
バージョン 16.0 でサポート終了、Mobile Link, 52
IBM DB2 LUW のアップグレード
Mobile Link 統合データベース, 323
IBM DB2 メインフレーム
バージョン 12.0.0 のサポート終了、Mobile Link, 168
IdentifySimulatedClient コールバック
バージョン 12.0.1 の動作変更、Mobile Link (mlgenreplayapi), 102
identity_password 同期パラメータ
バージョン 12.0.0 の新機能、Mobile Link, 161
identity 同期パラメータ
バージョン 12.0.0 の新機能、Mobile Link, 161
Idle 接続パラメータ
バージョン 12.0.0 の動作変更, 149
ID 転送
バージョン 12.0.1、Relay Server, 84
ID ファイル
バージョン 11.0.0 の動作変更, 281
IF EXISTS 句
バージョン 11.0.1 の新機能, 200
バージョン 12.0.0 の新機能, 137
バージョン 12.0.0 の新機能、Ultra Light J, 173
IF NOT EXISTS 句
バージョン 11.0.1 の新機能, 200
バージョン 12.0.0 の新機能, 137
バージョン 12.0.0 の新機能、Ultra Light J, 173
IF 式
バージョン 11.0.0 の強化, 233
IF 文
バージョン 11.0.0 の強化, 233
バージョン 11.0.0 の強化、Ultra Light, 268
IIS
バージョン 16.0、Windows 版 32 ビット Relay Server のサポート終了, 56
IIS 6
バージョン 12.0.1 クイック配備, 84
IIS 7

バージョン 12.0.1 クイック配備、Relay Server, 84
IIS 7、7.5、8.0 のクイック設定
バージョン 16.0 の新機能、Relay Server, 53
IIS 8
バージョン 16.0 の新機能、Relay Server でのサポート, 53
IMMEDIATE 句
バージョン 11.0.1 の強化、MESSAGE 文, 200
INDEX ONLY 句
バージョン 11.0.0 の新機能, 233
INPUT 文
バージョン 11.0.0 の強化, 277
バージョン 11.0.0 の動作変更, 277, 278
バージョン 12.0.0 の強化, 192
バージョン 12.0.1 の強化, 85
INPUT INTO 文
バージョン 11.0.0 の動作変更, 277
INSERT 文
バージョン 11.0.0 の強化, 234, 241
バージョン 11.0.0 の動作変更, 251
バージョン 11.0.1 の強化, 199
バージョン 12.0.0 の強化, 138
複数のローの強化, 138
INTEGER UNSIGNED データ型
バージョン 16.0 の新機能, 25
Intel ストレージドライバ
バージョン 12.0.0 の強化, 146
Interactive SQL
オートコンプリート, 112
テキスト補完, 112
バージョン 11.0.0 の強化, 275
バージョン 11.0.0 の新機能, 273
バージョン 12.0.0 の強化, 192
バージョン 12.0.0 の新機能, 189
バージョン 12.0.1 の強化, 85, 113
バージョン 12.0.1 の新機能, 112
バージョン 16.0 の新機能, 69
バージョン 16.0 の動作変更, 70
Interactive SQL エラーウィンドウ
バージョン 16.0 の強化, 70
Interactive SQL の新機能
バージョン 11.0.0, 270
バージョン 11.0.1, 211
バージョン 12.0.0, 185
バージョン 12.0.1, 111
バージョン 16.0, 69
Interactive SQL の動作の変更
バージョン 11.0.0, 274
バージョン 11.0.1, 212
バージョン 12.0.0, 191
バージョン 12.0.1, 113
INTERSECT 句
バージョン 11.0.0 の強化, 234
バージョン 11.0.0 の動作変更, 251
INTO LOCAL TEMPORARY TABLE 句
バージョン 11.0.1 の新機能, 200
IN リストアルゴリズム (IN)
バージョン 11.0.0 の強化, 241
IPAddressMonitorPeriod プロパティ
バージョン 12.0.0 の新機能, 134
IPv6
バージョン 12.0.1 の動作変更, 74
IP アドレス
バージョン 12.0.0 の強化, 131
IP プロトコルオプション
バージョン 12.0.0 の動作変更, 148
IQJDBC サーバクラス
バージョン 12.0.0 で廃止, 153
バージョン 16.0 でサポート終了, 40
IQODBC サーバクラス
バージョン 12.0.0 の新機能, 136
IRDK
バージョン 11.0.0 の動作変更, 252
is_online 変数
バージョン 12.0.1 の新機能、Mobile Link, 98
IsAesniAvailable プロパティ
バージョン 16.0 の新機能, 21
IsDebugger プロパティ
バージョン 11.0.1 の新機能、接続プロパティ, 197
IS DISTINCT FROM 探索条件
バージョン 12.0.0 の新機能, 136
IsEccAvailable プロパティ
サーバプロパティの説明, 7
ISENCRYPTED 関数
バージョン 12.0.0 の新機能, 135
IS NOT DISTINCT FROM 探索条件
バージョン 12.0.0 の新機能, 136
IS NOT NULL 探索条件
バージョン 12.0.1 の強化, 90
IsPortableDevice プロパティ
バージョン 12.0.0 の新機能, 133
isql_allow_client_file_read
バージョン 11.0.0 で名前が変更, 278
isql_allow_client_file_write

バージョン 11.0.0 で名前が変更, 278
isql_allow_read_client_file オプション
バージョン 11.0.0 で名前が変更, 278
isql_allow_write_client_file オプション
バージョン 11.0.0 で名前が変更, 278
isql_command_timing オプション
バージョン 12.0.0 の強化, 190
isql_plan オプション
バージョン 11.0.0 で削除, 278
ISYSCAPABILITYNAME
バージョン 11.0.0 の動作変更, 244
ISYSCERTIFICATE
バージョン 16.0 の新機能, 26
ISYSCOLSTAT
バージョン 16.0 の強化, 26
ISYSDBFILE
バージョン 11.0.0 の動作変更, 244
ISYSDBSPACE
バージョン 11.0.0 の動作変更, 244
ISYSDBSPACEPERM
バージョン 11.0.0 の動作変更, 244
ISYSEVENTTYPE
バージョン 11.0.0 の動作変更, 244
ISYSEXTERNENVOBJECT
バージョン 16.0 の強化, 26
ISYSFILE
バージョン 11.0.0 の動作変更, 244
ISYSGROLEGRANT
バージョン 16.0 の新機能, 4
ISYSGROLEGRANTEXT
バージョン 16.0 の新機能, 4
ISYSHISTORY
バージョン 16.0 の強化, 26
ISYSIDX
バージョン 11.0.0 の動作変更, 244
ISYSJAR
バージョン 16.0 の強化, 26
ISYSJAVACLASS
バージョン 16.0 の強化, 26
ISYSLOGINMAP
バージョン 11.0.0 の新機能, 244
ISYSLOGINPOLICY
バージョン 11.0.0 の新機能, 244
ISYSLOGINPOLICYOPTION
バージョン 11.0.0 の新機能, 244
ISYSMIRROROPTION
バージョン 12.0.0 の新機能, 123
ISYSMIRRORSERVER
バージョン 12.0.0 の新機能, 123
ISYSMIRRORSERVEROPTION
バージョン 12.0.0 の新機能, 123
ISYSOBJECT
バージョン 11.0.0 の動作変更, 244
バージョン 12.0.0 の強化, 143
バージョン 16.0 の強化, 26
ISYSPROCPARM
バージョン 12.0.0 の強化, 143
ISYSREMOTE USER
バージョン 16.0 の強化, 26
ISYSSEQUENCE
バージョン 12.0.0 の新機能, 143
ISYSSEQUENCEPERM
バージョン 12.0.0 の新機能, 143
ISYSSPATIALREFERENCESYSTEM
バージョン 12.0.0 の新機能, 121
ISYSTAB
バージョン 11.0.0 の動作変更, 244
バージョン 16.0 の強化, 26
ISYSTABCOL
バージョン 12.0.0 の強化, 143
バージョン 16.0 の強化, 26
ISYSTEXTCONFIG
バージョン 11.0.0 の新機能, 244
バージョン 12.0.0 の強化, 143
ISYSTEXTIDX
バージョン 11.0.0 の新機能, 244
バージョン 16.0 の強化, 26
ISYSTEXTIDXTAB
バージョン 11.0.0 の新機能, 244
ISYSUNITOFMEASURE
バージョン 12.0.0 の新機能, 121
ISYSUSER
バージョン 16.0 の強化, 26
ISYSUSERTYPE
バージョン 12.0.0 の強化, 143
ISYSVIEW
バージョン 11.0.0 の動作変更, 244
バージョン 16.0 の強化, 26
Itanium
バージョン 11.0.0 の動作変更, 251

J

java_class_path オプション
バージョン 12.0.1 の新機能, 85
java_location

バージョン 16.0 で廃止, 38
java_main_userid
バージョン 16.0 で廃止, 38
java_main_userid プロパティ
バージョン 12.0.0 でサポート対象外, 153
Java DownloadTableData インタフェース
バージョン 11.0.0 の強化, 259
Java ME
バージョン 11.0.0 の新機能、Java ME での Ultra Light サポート, 269
バージョン 16.0 で削除された機能、Java ME での Ultra Light サポート, 66
Java SE
バージョン 11.0.0 の新機能、Java SE での Ultra Light サポート, 269
jConnect
バージョン 11.0.0 の強化, 222
JDBC
トラブルシューティング, 316
バージョン 12.0.1 の強化, 89
バージョン 16.0 の動作変更, 36
JDBC 3.0
バージョン 12.0.0 の強化, 140
バージョン 15 で廃止, 96
JDBC 3.0 ドライバ
バージョン 16.0 でサポート終了, 40
JDBC 4.0
バージョン 12.0.0 の強化, 140
バージョン 12.0.1 の強化, 77
JDBC ドライバ
バージョン 12.0.0 の強化, 140
バージョン 12.0.0 の動作の変更, 148
JDBC ベースのサーバクラス
バージョン 12.0.0 で廃止, 153
バージョン 16.0 でサポート終了, 40
JSON サービス
バージョン 16.0 の動作変更, 37

L

LAST_VALUE 関数
バージョン 11.0.1 の強化, 199
LastCheckpointTime プロパティ
バージョン 12.0.0 の新機能, 132
LastCommitRedoPos プロパティ
バージョン 12.0.1 の新機能, 76
バージョン 16.0 の新機能, 21
LastStatement プロパティ
バージョン 16.0 の動作変更, 32
LastSyncedRedoPos プロパティ
バージョン 12.0.1 の新機能, 76
バージョン 16.0 の新機能, 21
LastWrittenRedoPos プロパティ
バージョン 12.0.1 の新機能, 76
バージョン 16.0 の新機能, 21
LazyClose 接続パラメータ
バージョン 11.0.0 の動作変更, 247
LCLOSE 接続パラメータ
バージョン 11.0.0 の動作変更, 247
LDAP
バージョン 16.0 の新機能, 5
バージョン 16.0 の新機能、Mobile Link, 41
バージョン 16.0 の新機能、Mobile Link プラグイン, 49
ldap_auto_failback_period オプション
バージョン 16.0 の新機能, 5
ldap_failover_to_std オプション
バージョン 16.0 の新機能, 5
ldap_primary_server オプション
バージョン 16.0 の新機能, 5
ldap_refresh_dn オプション
バージョン 16.0 の新機能, 5
ldap_secondary_server オプション
バージョン 16.0 の新機能, 5
LDAP ユーザ認証
バージョン 16.0 の強化, 5
バージョン 16.0 の新機能, 5
LENGTH 関数
バージョン 11.0.0 の強化, 240
LEN 関数
バージョン 11.0.0 の強化, 240
LIMIT 句
バージョン 12.0.0 の新機能, 138
バージョン 12.0.1 の強化, 88
Linux
バージョン 11.0.0 の強化, 240
バージョン 11.0.0 の新機能, 239
バージョン 11.0.0 の動作変更, 251
バージョン 12.0.0 の強化, 144
バージョン 16.0、32 ビット Relay Server のサポート終了, 56
バージョン 16.0 の強化, 27
Linux 上の Apache のクイック設定
バージョン 16.0 の新機能、Relay Server, 54
LOAD STATISTICS 統計
バージョン 12.0.0 の動作変更, 147

LOAD TABLE 文
バージョン 11.0.0 の強化, 230, 233, 263
バージョン 12.0.0 の強化, 137
バージョン 12.0.1 の強化, 75
バージョン 12.0.1 の動作変更, 94
バージョン 16.0 の強化, 24
バージョン 16.0 の動作変更, 37
LOAD 権限
バージョン 16.0 の新機能, 2
LOCK FEATURE 文
バージョン 12.0.0 の新機能, 137
LockIndexID プロパティ
バージョン 11.0.0 の新機能, 225
LockRowID プロパティ
バージョン 11.0.0 の新機能, 225
LOCK TABLE 文
バージョン 12.0.0 の強化, 144
log4j.jar
バージョン 12.0.0 の動作変更、Mobile Link, 165
LogFormat プロトコルオプション
バージョン 12.0.1 の動作変更, 76
login_mode オプション (*参照* バージョン 12.0.1 の新機能、OnDemand)
バージョン 16.0 の強化, 6
login_procedure オプション
バージョン 11.0.0 の強化, 223
Log Transfer Manager ユーティリティ
バージョン 12.0.0 でサポートされない, 117
Log Transfer Manager ユーティリティ (dbltm)
バージョン 12.0.0 の動作変更, 118
LOTUS フォーマット、INPUT 文
バージョン 11.0.0 で削除、サポート, 277
LOTUS フォーマット、OUTPUT 文
バージョン 11.0.0 で削除、サポート, 277
LTMGeneration プロパティ
バージョン 12.0.0 の動作変更, 117
LTM (Log Transfer Manager ユーティリティ)
バージョン 12.0.0 でサポートされない, 117
LTMTrunc プロパティ
バージョン 12.0.0 の動作変更, 117

M

Mac OS X
SQL Anywhere 16 用にデータベースを再構築, 291
バージョン 12.0.0 の強化, 144
バージョン 12.0.0 の動作変更, 152
バージョン 16.0 の強化, 27
MAPI
バージョン 16.0 の動作変更, 34
MAPI および SMTP システムプロシージャ
バージョン 16.0 の動作変更, 34
MAPI メッセージタイプ
バージョン 11.0.0 でサポート終了, 263
MapPages プロパティ
バージョン 11.0.0 でサポート終了, 253
MatView 接続パラメータ
バージョン 12.0.1 の新機能, 85
バージョン 12.0.1 の動作変更, 95
max_priority オプション
バージョン 11.0.0 の新機能, 223
max_priority プロパティ
バージョン 11.0.0 の新機能, 225
max_retries 制御パラメータ
バージョン 12.0.1 の強化, 103
MaxEventType
バージョン 11.0.0 の新機能, 226
MaxMessage プロパティ
バージョン 11.0.0 で廃止, 253
MaxMultiProgrammingLevel プロパティ
バージョン 12.0.0 の新機能, 133
MaxRemoteCapability プロパティ
バージョン 11.0.0 の新機能, 226
MaxRequestVars プロトコルオプション
バージョン 16.0 の新機能, 13
MAXVARS プロトコルオプション
バージョン 16.0 の新機能, 13
MEDIAN 関数
バージョン 12.0.0 の新機能, 134
Memory (mem) 拡張オプション
バージョン 12.0.0 のサポート終了、Mobile Link, 168
MERGE 文
バージョン 11.0.0 の新機能, 216
MessageCategoryLimit プロパティ
バージョン 11.0.0 の新機能, 226
MESSAGE LOG 句
バージョン 16.0 の新機能, 35
MessageText プロパティ
バージョン 11.0.0 で廃止, 253
MessageTime プロパティ
バージョン 11.0.0 で廃止, 253
MessageWindowSize プロパティ
バージョン 11.0.0 で廃止, 253

Message プロパティ
バージョン 11.0.0 で廃止, 253
MESSAGE 文
バージョン 11.0.1 の強化, 200
Microsoft Access
バージョン 11.0.0 の強化、リモートデータアクセス, 242
Microsoft Excel
バージョン 11.0.0 で削除、INPUT 文のサポート, 277
バージョン 11.0.0 で削除､OUTPUT 文のサポート, 277
Microsoft IIS
バージョン 16.0、Windows 版 32 ビット Relay Server のサポート終了, 56
Microsoft IIS 8
バージョン 16.0 の新機能、Relay Server でのサポート, 53
Microsoft SQL Server 2005 のサポート
バージョン 16.0 でサポート終了、Mobile Link, 52
Microsoft SQL Server 2008 のサポート
バージョン 11.0.1 の強化, 208
Microsoft SQL Server 2012 のサポート
バージョン 16.0 の新機能、Mobile Link, 42
Microsoft 分散トランザクションコーディネーター
バージョン 11.0.0 の新機能, 258
バージョン 12.0.0 の新機能, 158
Microsoft ボリュームシャドウコピーサービス (VSS)
バージョン 11.0.0 の新機能, 220
min_password_length オプション
バージョン 16.0 の動作変更, 30
min_role_admins オプション
バージョン 16.0 の新機能, 2, 3
min_role_admins プロパティ
バージョン 16.0 の新機能, 2
MinMultiProgrammingLevel プロパティ
バージョン 12.0.0 の新機能, 133
MIRROR_FILE 接続パラメータ
バージョン 11.0.1 の強化、Ultra Light, 209
MirrorMode プロパティ
バージョン 11.0.0 の新機能, 227
MirrorRole プロパティ
バージョン 12.0.0 の新機能, 132
MirrorServerState プロパティ
バージョン 12.0.0 の新機能, 135
バージョン 12.0.1 の強化, 86
MirrorState プロパティ
バージョン 12.0.0 の新機能, 135
MIRROR サーバクラス
バージョン 16.0 の新機能, 11
ml_add_certificates_file システムプロシージャ
バージョン 16.0 の新機能、Mobile Link, 41
ml_add_column システムプロシージャ
バージョン 12.0.0 の動作変更、Mobile Link, 166
バージョン 16.0 で廃止、Mobile Link, 53
ml_add_ldap_server システムプロシージャ
バージョン 16.0 の新機能、Mobile Link, 41
ml_add_missing_dnld_scripts システムプロシージャ
バージョン 12.0.0 の新機能、Mobile Link, 160
ml_add_user_auth_policy システムプロシージャ
バージョン 16.0 の新機能、Mobile Link, 41
ml_column システムテーブル
バージョン 16.0 で廃止、Mobile Link, 53
ML_GET_SERVER_NOTIFICATION 関数
バージョン 11.0.1 の強化、Ultra Light, 210
ml_ignore
バージョン 11.0.1 の新機能、Mobile Link (mlsrv11), 208
ml_model_check_all_schema システムプロシージャ
バージョン 16.0 の新機能、Mobile Link, 41
ml_model_check_version_schema システムプロシージャ
バージョン 16.0 の新機能、Mobile Link, 41
ml_model_drop システムプロシージャ
バージョン 16.0 の新機能、Mobile Link, 41
ml_remote_id オプション
バージョン 11.0.1 の強化、Ultra Light, 209
mlarbiter ユーティリティ
バージョン 12.0.0 の新機能、Mobile Link, 160
mlarbstop ユーティリティ
バージョン 16.0 の新機能, 44
mlfiletransfer
バージョン 12.0.0 の強化、Mobile Link, 164
MLFileTransfer メソッド
バージョン 12.0.0 の動作変更、Mobile Link, 167
MLFileUpload メソッド
バージョン 12.0.0 の新機能、Mobile Link, 162
mlreplay 活性タイムアウト

バージョン 16.0 の強化、Mobile Link (mlreplay), 44
mlreplay ユーティリティ
バージョン 16.0 のコールバックの削除, 52
mlreplay ユーティリティ
バージョン 12.0.0 の新機能、Mobile Link, 155
mlreplay ユーティリティのエラーメッセージ
バージョン 12.0.1 の強化、Mobile Link, 83
mlsrv12
バージョン 12.0.0 の新機能、Mobile Link, 158
バージョン 12.0.0 の動作変更, 118
mlstop -f オプション
バージョン 16.0 で廃止、Mobile Link, 52
MLUser 同期プロファイル
バージョン 12.0.0 で廃止、Mobile Link, 170
mluser ユーティリティ
バージョン 16.0 の新機能, 44
Mobile Link
アップグレード, 319
Mobile Link Relay Server
バージョン 11.0.0 の新機能, 258
Mobile Link エージェント
バージョン 12.0.1 の新機能、MobiLink, 98
Mobile Link クライアント
バージョン 12.0.1 の新機能, 100
バージョン 12.0.1 の動作の変更, 102
Mobile Link サーバ
32 ビットサポート、バージョン 16.0 の動作変更, 50
アップグレード, 328
Mobile Link サーバ 32 ビットサポート
バージョン 12.0.1 の動作の変更、Mobile Link, 101
Mobile Link サーバの互換性
バージョン 16.0 の新機能、Mobile Link, 41
[Mobile Link サーバのコマンドラインのプロパティ] ウィンドウ
バージョン 16.0 の新機能、Mobile Link プラグイン, 48
Mobile Link サーバファーム
バージョン 11.0.0 の新機能, 259
Mobile Link システムオブジェクト
アップグレード, 320
バージョン 11.0.0 の強化, 258
Mobile Link システムデータベース
バージョン 11.0.0 の新機能, 257
Mobile Link システムプロシージャ
バージョン 11.0.0 の強化, 258
Mobile Link でサポートされなくなった機能
バージョン 12.0.1, 103
Mobile Link で廃止された機能
バージョン 12.0.1, 103
Mobile Link の新機能
バージョン 11.0.0, 257
バージョン 11.0.1, 207
バージョン 12.0.0, 155
バージョン 12.0.1, 96
バージョン 16.0, 40
Mobile Link の接続
バージョン 16.0 の動作変更, 50
Mobile Link の動作の変更
バージョン 11.0.0, 261
バージョン 11.0.1, 209
バージョン 12.0.0, 165
バージョン 12.0.1, 101
バージョン 16.0, 49
Mobile Link プラグイン
バージョン 12.0.1 の強化、Mobile Link, 99
バージョン 12.0.1 の動作の変更, 103
Mobile Link プロファイラ
バージョン 16.0 の新機能, 45
バージョン 16.0 の新機能、Mobile Link, 40
Mobile Link モデルモードでの MySQL のサポート
バージョン 11.0.1 の強化, 209
Mobile Link モニタ
バージョン 11.0.0 の動作変更, 279
バージョン 16.0 で削除、Mobile Link, 40, 45
Mobile Link ログファイル
バージョン 11.0.0 の強化、Ultra Light, 266
MSACCESSODBC サーバクラス
バージョン 11.0.0 の新機能, 242
MYSQLODBC サーバクラス
バージョン 11.0.0 の新機能, 242
MySQL server 5.5.X のサポート
バージョン 16.0 の新機能、Mobile Link, 43
MySQL 統合データベース
バージョン 11.0.0 の新機能, 257

N

Name プロパティ
バージョン 12.0.0 の動作変更, 149
NCHAR データ型
バージョン 12.0.0 の強化, 139
network_conn_name 変数

バージョン 12.0.1 の新機能、Mobile Link, 98
NetworkData クラス
バージョン 12.0.1 の新機能、Mobile Link, 83
new_remote_id システムパラメータ
バージョン 12.0.1 の新機能、Mobile Link, 97
new_username システムパラメータ
バージョン 12.0.1 の新機能、Mobile Link, 97
newdemo
バージョン 12.0.0 の新機能, 145
NewPassword 接続パラメータ
バージョン 11.0.0 の新機能, 220
NEWPWD 接続パラメータ
バージョン 11.0.0 の新機能, 220
NEXT_CONNECTION 関数
バージョン 16.0 の動作変更, 33
NEXT_DATABASE 関数
バージョン 16.0 の動作変更, 34
NEXT_HTTP_RESPONSE_HEADER 関数
バージョン 12.0.0 の新機能, 135
no_reload_status
バージョン 12.0.0 の新機能, 140
NodeType 接続パラメータ
バージョン 12.0.0 の新機能, 122
バージョン 12.0.1 の強化, 73
バージョン 12.0.1 の動作変更、OnDemand, 74
NODE 接続パラメータ
バージョン 12.0.0 の新機能, 122
non_keywords オプション
バージョン 11.0.0 の動作変更, 249
NOTIFY TRACE EVENT 文
バージョン 16.0 の新機能, 24
NumLocalTempTables プロパティ
バージョン 16.0 の新機能, 20
NVARCHAR データ型
バージョン 12.0.0 の強化, 139

O

ObjectType プロパティ
バージョン 12.0.0 の新機能, 133
OData サーバ
バージョン 16.0 の新機能, 10
OData プロデューサ
バージョン 16.0 の新機能, 10
ODBC
バージョン 16.0 の動作変更, 36
ODBC サーバクラス
バージョン 11.0.0 の強化, 242
ODBC 接続パラメータ
バージョン 12.0.0 の強化, 128
ODBC データソース
バージョン 12.0.0 の強化, 128
ODBC データソースアドミニストレータ
バージョン 12.0.1 の強化、OnDemand, 73
ODBC データソースとして保存
バージョン 11.0.0 の新機能, 271
ODBC ドライバ
バージョン 12.0.0 の強化, 140
ODBC ドライバマネージャ
バージョン 11.0.0 の強化, 237
OEM.ini
バージョン 11.0.0 の強化, 274
バージョン 12.0.0 の強化, 190
バージョン 12.0.1 の強化, 75
バージョン 12.0.1 の新機能, 111
バージョン 16.0 の強化, 67, 69
OFFSET 句
バージョン 12.0.1 の強化, 88
OLE DB
バージョン 11.0.1 の強化, 201
バージョン 12.0.1 の強化, 88
バージョン 16.0 の強化, 26
OnDemand Edition のサポート
バージョン 12.0.1 の新機能, 73
ON 句
バージョン 16.0 でサポート終了, 37
Open Client
バージョン 11.0.0 の強化, 222
OpenString アルゴリズム
バージョン 11.0.0 の新機能, 219
OPENSTRING 句
バージョン 11.0.0 の新機能, 219
バージョン 12.0.1 の強化, 76
バージョン 16.0 の強化, 25
OpenTableEx 関数 [UL C++]
バージョン 11.0.0 の強化、Ultra Light, 268
OPENXML 演算子
バージョン 12.0.0 の強化, 134
バージョン 12.0.0 の動作変更, 151
optimistic_wait_for_commit オプション
バージョン 11.0.0 でサポート終了, 254
OptionWatchAction プロパティ
バージョン 11.0.0 の新機能, 226
OptionWatchList プロパティ
バージョン 11.0.0 の新機能, 226
OPTION 句

バージョン 11.0.0 の強化, 234
バージョン 11.0.0 の動作変更, 251
Oracle
バージョン 11.0.0 の強化, 258
バージョン 12.0.0 の強化, 158
バージョン 16.0 の新機能、Mobile Link, 43
Oracle varray
バージョン 11.0.1 の新機能, 208
Oracle ドライバ
バージョン 11.0.0 の強化, 258
バージョン 12.0.0 の強化, 158
OR REPLACE 句
バージョン 11.0.1 の強化, 200
バージョン 12.0.0 の新機能, 138
OSGi 配備バンドル
バージョン 12.0.1 の強化, 77
OSUser プロパティ
バージョン 11.0.0 の新機能, 225
Outbound Enabler
バージョン 12.0.0 の強化、Relay Server, 165
OUTER APPLY 句
バージョン 11.0.0 の新機能, 233
OUTPUT 文
バージョン 11.0.0 の強化, 277
バージョン 11.0.0 の動作変更, 277, 278
バージョン 11.0.0 の動作変更、dbisql, 253
バージョン 11.0.1 の動作変更, 207
バージョン 12.0.0 の強化, 192

P

PacketsReceivedUncomp プロパティ
バージョン 12.0.1 の新機能, 92
PacketsReceived プロパティ
バージョン 12.0.1 の新機能, 92
PacketsSentUncomp プロパティ
バージョン 12.0.1 の新機能, 92
PacketsSent プロパティ
バージョン 12.0.1 の新機能, 92
Palm OS
バージョン 12 の廃止予定機能、Ultra Light, 269
Palm デバイス用の Mobile Link Listener C API
バージョン 12.0.0 の削除, 167
Palm デバイス用の Mobile Link Listener ユーティリティ
バージョン 12.0.0 の削除, 167
ParentConnection プロパティ
バージョン 12.0.0 の新機能, 132
PartnerState プロパティ
バージョン 12.0.1 の強化, 86
バージョン 16.0 の動作変更, 12
pause_after_failure 制御パラメータ
バージョン 12.0.1 の強化, 103
PBUF 接続パラメータ
バージョン 11.0.0 の強化, 220
バージョン 12.0.0 の動作変更, 150
percent_as_comment オプション
バージョン 11.0.0 の動作変更, 254
PHP
バージョン 12.0.0 の強化, 142
バージョン 16.0 でサポートされない関数, 38
バージョン 16.0 の強化, 26
PHP モジュール
バージョン 11.0.0 の動作変更、関数名の変更, 246
PID ファイル
バージョン 12.0.0 の新機能, 144
Pocket PC 2003
バージョン 12.0.1 の新機能, 83
バージョン 16.0 の強化、Ultra Light, 60
Pocket PC 2003 での RSA TLS と HTTPS
バージョン 16.0 の強化、Ultra Light, 60
port プロトコルオプション
バージョン 12.0.0 の動作変更, 149
post_login_procedure オプション
バージョン 11.0.0 の動作変更, 249
precision オプション
バージョン 11.0.0 の動作変更, 250
PrefetchBuffer (PBUF) 接続パラメータ
バージョン 12.0.0 の動作変更, 150
PrefetchBuffer 接続パラメータ
バージョン 11.0.0 の強化, 220
PREFILTER 句
バージョン 12.0.0 の新機能, 126
Prepares プロパティ
バージョン 11.0.1 の新機能, 198
PREPARE 文
バージョン 12.0.0 の動作変更, 150
PrepStmt プロパティ
バージョン 12.0.1 の新機能, 92
PreserveSource プロパティ
バージョン 11.0.0 でサポート終了, 253
priority オプション
バージョン 11.0.0 の新機能, 224
priority プロパティ

バージョン 11.0.0 の新機能, 225
ProcessID プロパティ
バージョン 12.0.1 の新機能, 76
ProcessorAffinity プロパティ
バージョン 16.0 の新機能, 21
PROFILE 権限
バージョン 16.0 で廃止, 38
progress_messages オプション
バージョン 12.0.0 の新機能, 130
progress_messages プロパティ
バージョン 12.0.0 の新機能, 132
Progress プロパティ
バージョン 12.0.0 の新機能, 132
promotion_time オプション
バージョン 16.0 の新機能, 12
PROPERTY_NAME 関数
バージョン 11.0.0 の強化, 227
proxy_host オプション
バージョン 12.0.1 の新機能、Relay Server, 100
proxy_port オプション
バージョン 12.0.1 の新機能、Relay Server, 101
Publication 同期プロファイル
バージョン 12.0.0 で廃止、Mobile Link, 170
Python
バージョン 11.0.0 の新機能、Python データベース API, 235
バージョン 12.0.0 の新機能, 142

Q

QAnywhere
バージョン 16.0 でサポート終了, 53
query_mem_timeout オプション
バージョン 11.0.0 の新機能, 224
query_mem_timeout プロパティ
バージョン 11.0.0 の新機能, 225
query_plan_on_open オプション
バージョン 11.0.0 でサポート終了, 254
QueryBypassedCosted プロパティ
バージョン 11.0.1 の新機能、接続プロパティ, 197
バージョン 11.0.1 の新機能、データベースプロパティ, 198
QueryBypassedHeuristic プロパティ
バージョン 11.0.1 の新機能、接続プロパティ, 197
バージョン 11.0.1 の新機能、データベースプロパティ, 198
QueryBypassedOptimized プロパティ
バージョン 11.0.1 の新機能、接続プロパティ, 197
バージョン 11.0.1 の新機能、データベースプロパティ, 198
QueryDescribedBypass プロパティ
バージョン 11.0.1 の新機能、接続プロパティ, 197
バージョン 11.0.1 の新機能、データベースプロパティ, 198
QueryDescribedOptimizer プロパティ
バージョン 11.0.1 の新機能、接続プロパティ, 197
バージョン 11.0.1 の新機能、データベースプロパティ, 198
QueryHeapPages プロパティ
バージョン 12.0.1 の新機能, 92
QueryMemActiveCurr プロパティ
バージョン 11.0.0 の新機能, 225
バージョン 12.0.1 の新機能, 92
QueryMemActiveEst プロパティ
バージョン 11.0.0 の新機能, 226
QueryMemActiveMax プロパティ
バージョン 11.0.0 の新機能, 226
QueryMemExtraAvail プロパティ
バージョン 11.0.0 の新機能, 225
QueryMemGrantBaseMI プロパティ
バージョン 11.0.0 の新機能, 226
QueryMemGrantBase プロパティ
バージョン 11.0.0 の新機能, 226
QueryMemGrantExtra プロパティ
バージョン 11.0.0 の新機能, 226
QueryMemGrantFailed プロパティ
バージョン 11.0.0 の新機能, 225
バージョン 12.0.1 の新機能, 92
QueryMemGrantGranted プロパティ
バージョン 11.0.0 の新機能, 225
バージョン 12.0.1 の新機能, 92
QueryMemGrantRequested プロパティ
バージョン 11.0.0 の新機能, 225
バージョン 12.0.1 の新機能, 92
QueryMemGrantWaited プロパティ
バージョン 12.0.1 の新機能, 92
QueryMemGrantWaiting プロパティ
バージョン 11.0.0 の新機能, 225
QueryMemPages プロパティ
バージョン 11.0.0 の新機能, 226
QueryMemPercentOfCache プロパティ

バージョン 11.0.0 の新機能, 226
QueryMemWaited プロパティ
バージョン 11.0.0 の新機能, 225
QueryOpened プロパティ
バージョン 11.0.1 の新機能、接続プロパティ, 197
バージョン 11.0.1 の新機能、データベースプロパティ, 198
QueryReused プロパティ
バージョン 11.0.0 の新機能, 227
QueryRowsBufferFetch プロパティ
バージョン 12.0.0 でサポート対象外, 153
QueryRowsFetched プロパティ
バージョン 12.0.0 の新機能, 132
quoted_identifier オプション
バージョン 11.0.0 の動作変更, 250

R

RAISERROR 句
バージョン 12.0.1 の強化, 88
RAISERROR 文
バージョン 12.0.1 の強化, 88
READ_CLIENT_FILE 関数
バージョン 11.0.0 の新機能, 227
readcert ユーティリティ
バージョン 11.0.0 で削除, 281
READCLIENTFILE 権限
バージョン 16.0 で廃止, 38
READFILE 権限
バージョン 16.0 で廃止, 38
ReadHintScatterLimit プロパティ
バージョン 11.0.0 で名前が変更, 252
バージョン 11.0.0 の新機能, 226
ReadHintScatter プロパティ
バージョン 11.0.0 で名前が変更, 252
バージョン 11.0.0 の新機能, 225
ReadHint プロパティ
バージョン 11.0.0 で名前が変更, 252
バージョン 11.0.0 の新機能, 225
ReadPK ロック
バージョン 12.0.0 の新機能, 144
READ 文
バージョン 12.0.0 の強化, 113
ReceiveBufferSize プロトコルオプション
バージョン 12.0.1 の強化, 75
REFRESH MATERIALIZED VIEW 文
バージョン 11.0.0 の強化, 218
バージョン 11.0.0 の動作変更, 247
REFRESH TEXT INDEX 文
バージョン 11.0.0 の新機能, 232
REGBIN プロトコルオプション
バージョン 11.0.0 でサポート終了, 254
REGEXP_SUBSTR 関数
バージョン 11.0.0 の新機能, 216, 228
バージョン 11.0.1 の動作変更, 204
REGEXP 検索条件
バージョン 11.0.1 の動作変更, 204
REGEXP 探索条件
バージョン 11.0.0 の新機能, 216
RegisterBindery プロトコルオプション
バージョン 11.0.0 でサポート終了, 254
Relay Server
バージョン 12.0.1 の新機能, 100
バージョン 16.0、32 ビットバージョンのサポート終了 (Linux), 56
バージョン 16.0、32 ビットバージョンのサポート終了 (Windows), 56
Relay Server [ステータス] ページの改善
バージョン 16.0 の動作変更, 55
Relay Server の新機能
ID 転送, 84
バージョン 16.0, 53
Relay Server の動作変更
IIS 7、7.5、8.0 のクイック設定, 53
Linux 上の Apache のクイック設定, 54
up_pad_size, 54
仮想メモリの管理、向上, 54
重複 cookie の更新, 54
パススルーを使用したプロキシサポート, 54
バージョン 16.0, 55
メモリの管理、向上, 55
Relay Server のホスト
バージョン 11.0.0 の新機能, 258
Relay Server プラグインの動作変更
[詳細] タブ、[バックエンドサーバファームのプロパティ], 56
[ネットワーク] タブ, 55
Relay Server プラグインの変更
[ネットワーク] タブ、[Relay Server ファームのプロパティ], 56
reload.sql
バージョン 11.0.0 の強化, 242
remote_id システムパラメータ
バージョン 12.0.0 の新機能、Mobile Link, 160
RemoteCapability プロパティ

バージョン 11.0.0 の新機能, 226
REMOTE DBA 権限
バージョン 16.0 で廃止, 38
REORGANIZE TABLE 文
バージョン 11.0.0 の動作変更, 247
Relay Server プラグイン
バージョン 12.0.0 の新機能、Mobile Link, 157
replicate_all オプション
バージョン 12.0.0 でサポートされない, 118
Replication Server
バージョン 12.0.0 でサポートされない, 117
ReqCountActive プロパティ
バージョン 12.0.1 の新機能, 92
ReqCountBlockContention プロパティ
バージョン 12.0.1 の新機能, 92
ReqCountBlockIO プロパティ
バージョン 12.0.1 の新機能, 92
ReqCountBlockLock プロパティ
バージョン 12.0.1 の新機能, 92
ReqCountUnscheduled プロパティ
バージョン 12.0.1 の新機能, 92
ReqTimeActive プロパティ
バージョン 12.0.1 の新機能, 92
ReqTimeBlockContention プロパティ
バージョン 12.0.1 の新機能, 92
ReqTimeBlockIO プロパティ
バージョン 12.0.1 の新機能, 92
ReqTimeBlockLock プロパティ
バージョン 12.0.1 の新機能, 92
ReqTimeUnscheduled プロパティ
バージョン 12.0.1 の新機能, 92
reqtool ユーティリティ
バージョン 11.0.0 で削除, 281
RequestsReceived プロパティ
バージョン 12.0.1 の新機能, 92
reserved_keywords オプション
バージョン 12.0.0 の新機能, 131
バージョン 12.0.1 の動作変更, 94
reset.sql
オートインクリメントカラムのあるテーブルの再ロード, 294
RESOURCE 権限
バージョン 16.0 で廃止, 38
RESTART 句
バージョン 16.0 の新機能, 12
RESTORE DATABASE 文
バージョン 12.0.0 の強化, 128
バージョン 16.0 の強化, 25
ResultSet.afterLast メソッド [Ultra Light J]
バージョン 12.0.1 の新機能, 79
ResultSet.beforeFirst メソッド [Ultra Light J]
バージョン 12.0.1 の新機能, 79
ResultSet.first メソッド [Ultra Light J]
バージョン 12.0.1 の新機能, 79
ResultSet.getBlobInputStream メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSet.getBoolean メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSet.getBytes メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSet.getClobReader メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSet.getDate メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSet.getDecimalNumber メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSet.getDouble メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSet.getFloat メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSet.getLong メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSet.getRowCount メソッド [Ultra Light J]
バージョン 12.0.1 の新機能, 79
ResultSet.getSize メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSet.getString メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80, 82
ResultSet.getUUIDValue メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSet.isNull メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSet.last メソッド [Ultra Light J]
バージョン 12.0.1 の新機能, 79
ResultSet.relative メソッド [Ultra Light J]
バージョン 12.0.1 の新機能, 79
ResultSetMetaData.getAliasName メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSetMetaData.getCorrelationName メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSetMetaData.getDomainName メソッド [Ultra Light J]

バージョン 12.0.1 の強化, 80
ResultSetMetaData.getQualifiedName メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSetMetaData.getTableColumnName メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSetMetaData.getTableName メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
ResultSetMetaData.getWrittenName メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 80
RESUME 文
バージョン 12.0.0 の強化, 141
REVOKE PUBLISH 文
バージョン 16.0 の動作変更, 4
REVOKE ROLE 文
バージョン 16.0 の新機能, 3
REVOKE 文
バージョン 12.0.0 の強化, 124
バージョン 16.0 の強化, 3
REWRITE 関数
バージョン 16.0 の動作変更, 32
ri_trigger_time オプション
バージョン 11.0.0 でサポート終了, 254
RIM
バージョン 11.0.0 の新機能、BlackBerry での Ultra Light サポート, 269
Rlbk プロパティ
バージョン 12.0.1 の新機能, 92
ROLLBACK 文
バージョン 12.0.0 の強化, 189
root_auto_lock_time オプション
バージョン 16.0 の新機能, 2
ROW LOG 句
バージョン 16.0 の新機能, 35
ROW コンストラクタ
バージョン 16.0 の新機能, 9
rsoe
バージョン 12.0.0 の強化、Relay Server, 164
Ruby
バージョン 11.0.1 の新機能, 201

S

sa_audit_string システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_certificate_info システムプロシージャ
バージョン 16.0 の新機能, 21
sa_char_terms システムプロシージャ
バージョン 11.0.0 の新機能, 228
sa_clean_database システムプロシージャ
バージョン 12.0.1 の動作変更, 90
バージョン 16.0 の動作変更, 29
sa_column_stats システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_config.csh
バージョン 12.0.1 の強化、Unix, 87
sa_config.sh
バージョン 12.0.1 の強化、Unix, 87
sa_conn_activity システムプロシージャ
バージョン 16.0 の動作変更, 29, 33
sa_conn_compression_info システムプロシージャ
バージョン 16.0 の動作変更, 29, 33
sa_conn_info システムプロシージャ
バージョン 12.0.0 の強化, 127
バージョン 16.0 の動作変更, 29, 33
sa_conn_list システムプロシージャ
バージョン 16.0 の動作変更, 29, 33
sa_conn_options システムプロシージャ
バージョン 16.0 の動作変更, 29, 33
sa_conn_properties システムプロシージャ
バージョン 11.0.0 の強化, 230
バージョン 16.0 の動作変更, 29, 33
sa_copy_cursor_to_temp_table システムプロシージャ
バージョン 12.0.0 の新機能, 133
sa_cpu_topology システムプロシージャ
バージョン 16.0 の新機能, 21
sa_db_info system システムプロシージャ
バージョン 16.0 の動作変更, 29, 34
sa_db_list システムプロシージャ
バージョン 16.0 の動作変更, 29, 34
sa_db_option システムプロシージャ
バージョン 16.0 の新機能, 18, 21
sa_db_properties システムプロシージャ
バージョン 11.0.0 の強化, 230
バージョン 16.0 の動作変更, 29, 34
sa_describe_cursor システムプロシージャ
バージョン 12.0.0 の新機能, 133
sa_describe_shapefile システムプロシージャ
バージョン 12.0.0 の新機能, 120
sa_disable_auditing_type システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_disk_free_space システムプロシージャ

バージョン 11.0.0 の強化, 229
バージョン 16.0 の動作変更, 29
sa_enable_auditing_type システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_external_library_unload システムプロシージャ
バージョン 11.0.0 の新機能, 229
バージョン 16.0 の動作変更, 29
sa_flush_cache システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_flush_statistics システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_get_dtt_groupreads システムプロシージャ
バージョン 11.0.0 の新機能, 227
sa_get_histogram システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_get_ldapserver_status システムプロシージャ
バージョン 16.0 の新機能, 6
sa_get_request_profile システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_get_request_times システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_get_server_messages システムプロシージャ
バージョン 11.0.0 で廃止予定, 256
sa_get_table_definition システムプロシージャ
バージョン 11.0.1 の新機能、dbunload, 199
バージョン 16.0 の動作変更, 29
sa_get_user_status システムプロシージャ
バージョン 11.0.0 の新機能, 228
バージョン 16.0 の強化, 6
sa_index_density システムプロシージャ
バージョン 11.0.0 の強化, 229
バージョン 12.0.0 の動作変更, 151
バージョン 16.0 の動作変更, 29
sa_index_levels システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_install_feature システムプロシージャ
バージョン 12.0.0 の新機能, 133
バージョン 16.0 の動作変更, 29
sa_internal_text_index_postings システムプロシージャ
バージョン 11.0.0 の新機能、内部でのみ使用, 228
sa_java_loaded_classes システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_list_cursors システムプロシージャ
バージョン 12.0.0 の新機能, 133
sa_load_cost_model システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_locks システムプロシージャ
バージョン 12.0.0 の強化, 144
sa_make_object システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_materialized_view_can_be_immediate システムプロシージャ
バージョン 11.0.0 の新機能, 229
バージョン 16.0 の動作変更, 29
sa_materialized_view_info システムプロシージャ
バージョン 11.0.0 の強化, 229
sa_mirror_server_status システムプロシージャ
バージョン 12.0.0 の新機能, 133
sa_nchar_terms システムプロシージャ
バージョン 11.0.0 の新機能, 228
sa_parse_json システムプロシージャ
バージョン 16.0 の新機能, 21
sa_post_login_procedure システムプロシージャ
バージョン 11.0.0 の新機能, 229
sa_procedure_profile_summary システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_procedure_profile システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_recompile_views システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_refresh_materialized_views システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_refresh_text_indexes システムプロシージャ
バージョン 11.0.0 の新機能, 228
バージョン 16.0 の動作変更, 29
sa_remove_tracing_data システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_reserved_words システムプロシージャ
バージョン 12.0.0 の新機能, 134
sa_reset_identity システムプロシージャ
オートインクリメントカラムのあるテーブルの再ロード, 294
バージョン 16.0 の動作変更, 29
sa_save_trace_data システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_send_udp システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_server_messages システムプロシージャ
バージョン 11.0.0 の新機能, 256
sa_server_option システムプロシージャ
バージョン 11.0.0 の強化, 230, 239
バージョン 12.0.0 の強化, 124, 134

バージョン 16.0 の強化, 18, 21
バージョン 16.0 の動作変更, 29
sa_set_http_option システムプロシージャ
バージョン 11.0.1 の強化, 199
sa_set_tracing_level システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_table_fragmentation システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_table_page_usage システムプロシージャ
バージョン 12.0.0 の強化, 134
バージョン 16.0 の動作変更, 29
sa_table_stats システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_text_index_handles システムプロシージャ
バージョン 11.0.0 の新機能、内部でのみ使用, 228
sa_text_index_postings システムプロシージャ
バージョン 11.0.0 の新機能、内部でのみ使用, 228
sa_text_index_stats システムプロシージャ
バージョン 11.0.0 の新機能, 228
sa_text_index_vocab_nchar システムプロシージャ
バージョン 12.0.0 の新機能, 133
バージョン 16.0 の動作変更, 29
sa_text_index_vocab システムプロシージャ
バージョン 11.0.0 の新機能, 228
バージョン 12.0.0 の動作変更, 151
sa_unload_cost_model システムプロシージャ
バージョン 16.0 の動作変更, 29
sa_user_defined_counter_add システムプロシージャ
バージョン 12.0.1 の新機能, 87
バージョン 16.0 の動作変更, 29
sa_user_defined_counter_set システムプロシージャ
バージョン 12.0.1 の新機能, 87
バージョン 16.0 の動作変更, 29
sa_validate システムプロシージャ
バージョン 11.0.0 の動作変更, 247
バージョン 16.0 の動作変更, 29
sa_verify_password システムプロシージャ
バージョン 16.0 の動作変更, 29, 32
sajdbc.jar
バージョン 12.0.1 でサポートされない, 96
バージョン 16.0 でサポート終了, 40
sajdbc4.jar
バージョン 12.0.1 の動作変更, 96
SAJDBC サーバクラス
バージョン 12.0.0 で廃止, 153
バージョン 16.0 でサポート終了, 40
sample_config.csh
バージョン 12.0.1 の強化、Unix, 87
sample_config.sh
バージョン 12.0.1 の強化、Unix, 87
SAP HANA
バージョン 16.0 の新機能, 11
バージョン 16.0 の新機能、Mobile Link, 42
SAP Sybase IQ
バージョン 12.0.0 の新機能, 112
SAP Sybase IQ 15.2 のサポート
バージョン 16.0 でサポート終了、Mobile Link, 52
SAP Sybase IQ 15.4 のサポート
バージョン 16.0 の新機能、Mobile Link, 42
SASpxOptionsBuilder クラス [SA .NET 2.0]
バージョン 11.0.0 でサポート終了, 254
SATMP 環境変数
バージョン 11.0.0 の強化, 239
SAVE OPTION VALUES 句
バージョン 12.0.0 の新機能, 136
scale オプション
バージョン 11.0.0 の動作変更, 250
SCHEMATA ローセット、OLE DB
バージョン 11.0.1 の新機能, 201
script_full_path フィールド
バージョン 16.0 で削除、Mobile Link, 52
scvwen1600.jar
バージョン 16.0 で廃止, 68
SearchBindery プロトコルオプション
バージョン 11.0.0 でサポート終了, 254
secure_feature_key オプション
バージョン 16.0 で廃止, 38
SELECT 文
バージョン 11.0.0 の強化, 216, 233
バージョン 11.0.1 の強化, 200
バージョン 12.0.0 の強化, 126, 138
バージョン 12.0.1 の強化, 88
SELinux のポリシー
バージョン 11.0.0 の新機能, 240
send_column_names 同期パラメータ
バージョン 16.0 の動作変更、Ultra Light, 8
SendBufferSize プロトコルオプション
バージョン 12.0.1 の強化, 75
SendColumnNames 拡張オプション
バージョン 16.0 で廃止、Mobile Link, 50
バージョン 16.0 の動作変更、Mobile Link, 8
ServerEdition プロパティ

バージョン 11.0.1 の新機能, 198
ServerName 接続パラメータ
バージョン 12.0.0 の動作変更, 149, 154
バージョン 12.0.1 の動作変更、OnDemand, 74
ServerNodeAddress プロパティ
バージョン 11.0.0 の新機能, 225
ServerPort プロトコルオプション
バージョン 12.0.0 の動作変更, 149
Server 接続パラメータ
バージョン 12.0.0 の動作変更, 149
setAutoCheckpoint メソッド
バージョン 16.0 で削除された API、Ultra Light, 66
setClob
PreparedStatement, 89
setDatabaseKey メソッド
バージョン 16.0 の動作変更、Ultra Light, 66
SET MIRROR OPTION 文
バージョン 12.0.0 の新機能, 123
バージョン 16.0 の強化, 12
SET OPTION 文
バージョン 11.0.0 の動作変更、Interactive SQL, 278
バージョン 12.0.0 の強化, 138
バージョン 12.0.0 の動作変更, 154
バージョン 12.0.0 の動作変更、Interactive SQL, 192
バージョン 12.0.1 の動作変更、Ultra Light, 110
SET REMOTE OPTION
バージョン 12.0.1 の動作変更, 77
SET REMOTE OPTION 文
バージョン 12.0.1 の強化, 103
setSendColumnNames メソッド
バージョン 16.0 の動作変更、Ultra Light, 8
SET TEMPORARY OPTION 文
バージョン 16.0 の動作変更, 32
SetupVSPackage
バージョン 12.0.1 の強化, 77
SHAPEFILE 句
バージョン 12.0.0 の新機能、FROM 句, 119
バージョン 12.0.0 の新機能、LOAD TABLE 文の句, 119
バージョン 12.0.1 の新機能、INPUT 文, 85
SHARED_DIR 環境変数
バージョン 11.0.0 でサポート終了, 256
showPerformanceDataUI
バージョン 16.0 の新機能, 67
Sierra Wireless Aircards
バージョン 11.0.0 でサポート終了, 262
SIMILAR TO 検索条件
バージョン 11.0.1 の動作変更, 204
SIMILAR TO 探索条件
バージョン 11.0.0 の新機能, 216
SMALLINT UNSIGNED データ型
バージョン 16.0 の新機能, 25
SMS リスナ
バージョン 16.0 の動作変更, 50
SMTP
バージョン 12.0.0 の強化, 134
バージョン 16.0 の動作変更, 34
SNMP
バージョン 11.0.0 の強化, 242
SP
データベースミラーリング, 312
sp_alter_secure_feature_key システムプロシージャ
バージョン 16.0 の新機能, 15
sp_auth_sys_role_info システムプロシージャ
バージョン 16.0 の新機能, 2
sp_copy_directory システムプロシージャ
バージョン 16.0 の新機能, 22
バージョン 16.0 の動作変更, 29
sp_copy_file システムプロシージャ
バージョン 16.0 の新機能, 22
バージョン 16.0 の動作変更, 29
sp_create_directory システムプロシージャ
バージョン 16.0 の新機能, 22
バージョン 16.0 の動作変更, 29
sp_create_secure_feature_key システムプロシージャ
バージョン 16.0 の新機能, 14
バージョン 16.0 の動作変更, 32
sp_delete_directory システムプロシージャ
バージョン 16.0 の新機能, 22
バージョン 16.0 の動作変更, 29
sp_delete_file システムプロシージャ
バージョン 16.0 の新機能, 22
バージョン 16.0 の動作変更, 29
sp_displayroles システムプロシージャ
バージョン 16.0 の新機能, 2
sp_drop_secure_feature_key システムプロシージャ
バージョン 16.0 の新機能, 15
sp_forward_to_remote_server システムプロシージャ
バージョン 12.0.1 の新機能, 86
バージョン 16.0 の動作変更, 29

sp_get_last_synchronize_result システムプロシージャ
バージョン 16.0 の動作変更, 29
sp_has_role システムプロシージャ
バージョン 16.0 の新機能, 2
sp_list_directory システムプロシージャ
バージョン 16.0 の動作変更, 29
sp_list_secure_feature_keys システムプロシージャ
バージョン 16.0 の新機能, 15
sp_move_directory システムプロシージャ
バージョン 16.0 の新機能, 22
バージョン 16.0 の動作変更, 29
sp_move_file システムプロシージャ
バージョン 16.0 の新機能, 22
バージョン 16.0 の動作変更, 29
sp_objectpermission システムプロシージャ
バージョン 16.0 の新機能, 2
sp_password システムプロシージャ
バージョン 16.0 の動作変更, 32
sp_proc_priv システムプロシージャ
バージョン 16.0 の新機能, 2
sp_remote_columns システムプロシージャ
バージョン 16.0 の動作変更, 29
sp_remote_exported_keys システムプロシージャ
バージョン 16.0 の動作変更, 29
sp_remote_imported_keys システムプロシージャ
バージョン 16.0 の動作変更, 29
sp_remote_primary_keys システムプロシージャ
バージョン 16.0 の動作変更, 29
sp_remote_procedures システムプロシージャ
バージョン 16.0 の動作変更, 29
sp_remote_tables システムプロシージャ
バージョン 16.0 の動作変更, 29
sp_sys_priv_role_info システムプロシージャ
バージョン 16.0 の新機能, 2
sp_trace_event_fields システムプロシージャ
バージョン 16.0 の新機能, 10
sp_trace_event_session_events システムプロシージャ
バージョン 16.0 の新機能, 10
sp_trace_event_session_target_options システムプロシージャ
バージョン 16.0 の新機能, 10
sp_trace_event_session_targets システムプロシージャ
バージョン 16.0 の新機能, 10
sp_trace_event_sessions システムプロシージャ
バージョン 16.0 の新機能, 10
sp_trace_events システムプロシージャ
バージョン 16.0 の新機能, 10
sp_use_secure_feature_key システムプロシージャ
バージョン 16.0 の新機能, 15
バージョン 16.0 の動作変更, 32
SpatialUtilities クラス
バージョン 12.0.0 の新機能、Mobile Link, 160
SPX
バージョン 11.0.0 でサポート終了, 254
SpxOptionsBuilder プロパティ [SA .NET 2.0]
バージョン 11.0.0 でサポート終了, 254
SpxOptionsString プロパティ [SA .NET 2.0]
バージョン 11.0.0 でサポート終了, 254
sqlanydb
バージョン 11.0.0 の新機能, 235
バージョン 12.0.0 の新機能, 142
SQLANYSAMP12 環境変数
バージョン 12.0.1 の強化、Unix, 87
SQL Anywhere
バージョン 11.0.1 の新機能, 211
sqlanywhere_commit 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_connect 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_data_seek 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_disconnect 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_errorcode 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_error 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_execute 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_fetch_array 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_fetch_field 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_fetch_object 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_fetch_row 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_free_result 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_identity 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_num_fields 関数 (旧式)
バージョン 11.0.0 で廃止, 246

sqlanywhere_num_rows 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_pconnect 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_query 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_result_all 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_rollback 関数 (旧式)
バージョン 11.0.0 で廃止, 246
sqlanywhere_set_option 関数 (旧式)
バージョン 11.0.0 で廃止, 246
SQL Anywhere 10.0.0 以降のアップグレード
Mobile Link 統合データベース, 320
SQL Anywhere 11.0.1 のサポート
バージョン 16.0 でサポート終了、Mobile Link, 52
SQL Anywhere 11 CustDB データソース
バージョン 11.0.0 の動作変更, 282
SQL Anywhere 11 Demo データソース
バージョン 11.0.0 の動作変更, 282
SQL Anywhere 16
アップグレード, 283
バージョン 16.0 の新機能、Mobile Link, 42
SQL Anywhere C API
バージョン 11.0.0 の新機能, 234
SQL Anywhere Explorer
バージョン 12.0.0 でサポート対象外, 154
SQL Anywhere for MS Access 移行ユーティリティ
バージョン 11.0.0 でサポート終了, 242
SQL Anywhere JDBC ドライバ
バージョン 12.0.0 の新機能, 140, 141
SQL Anywhere MIB
バージョン 11.0.0 の強化, 242
SQL Anywhere ODBC ドライバ
バージョン 12.0.0 の強化, 140
バージョン 12.0.0 の動作変更, 150
SQL Anywhere OnDemand のサポート
バージョン 12.0.1 の新機能, 73
SQL Anywhere SNMP Extension Agent
バージョン 11.0.0 の強化, 242
SQL Anywhere のアップグレード
説明, 283
SQL Anywhere のサポートされなくなった機能
バージョン 12.0.0, 153
バージョン 12.0.1, 95
SQL Anywhere のサポート終了機能
バージョン 11.0.0, 253
バージョン 11.0.1, 207
バージョン 16.0, 38
SQL Anywhere の新機能
バージョン 11.0.0, 216
バージョン 12.0.0, 119
バージョン 12.0.1, 84
バージョン 16.0, 9
SQL Anywhere の動作の変更
バージョン 11.0.0, 244
バージョン 11.0.1, 203
バージョン 12.0.0, 146
バージョン 12.0.1, 94
バージョン 16.0, 29
SQL Anywhere の廃止予定機能
バージョン 11.0.0, 253
バージョン 11.0.1, 207
バージョン 12.0.0, 153
バージョン 12.0.1, 95
バージョン 16.0, 38
SQL Anywhere のローカライズ版
バージョン 16.0 の強化, 27
SQL Anywhere パススルースクリプト
バージョン 11.0.0 の強化、Ultra Light, 264
SQL Anywhere プラグイン
サポートされているバージョン, 283
SQL Anywhere プラグインでの照合の適合化サポート
バージョン 11.0.0 の新機能, 272
SQL Anywhere モニタ
バージョン 12.0.0 の強化、Relay Server, 164
バージョン 16.0 の動作変更, 70
SQL Anywhere モニタの動作の変更
バージョン 12.0.0, 194
バージョン 12.0.1, 115
SQL FLAGGER
バージョン 12.0.0 の強化, 146
SQLGetConnectAttr
バージョン 11.0.0 の動作変更, 248
SQL Remote
アップグレード, 334
バージョン 12.0.1 の動作変更, 77
見つからないメッセージ、処理, 104
SQL Remote Message Agent (dbremote)
バージョン 12.0.0 の動作変更, 118
SQL Remote の新機能
バージョン 11.0.0, 263
バージョン 12.0.0, 170
バージョン 12.0.1, 103

SQL Remote の動作の変更
バージョン 11.0.0, 263
バージョン 12.0.1, 104
SQL Remote メッセージ
バージョン 12.0.1 の強化, 104
SQL SECURITY 句
バージョン 11.0.0 の新機能, 228
SQLTables
バージョン 12.0.0 の動作変更, 150
バージョン 12.0.1 の動作変更, 95
バージョン 16.0 の動作変更, 36
SQL キーワード
バージョン 12.0.0 の強化, 134
SQL サポート
バージョン 11.0.1 の強化、Ultra Light, 210
SQL パススルー
バージョン 11.0.0 の新機能, 259
バージョン 12.0.0 のサポート終了、Mobile Link, 168
SQL 文の実行
バージョン 11.0.0 の強化、Interactive SQL, 276
SQL 文字列の区切り
バージョン 12.0.0 の強化, 145
SQL を返す Java スクリプトと .NET スクリプト
バージョン 12.0.0 のサポート終了、Mobile Link, 168
SRID 句
バージョン 12.0.1 の新機能、INPUT 文, 85
SSL
バージョン 11.0.0 の動作変更, 248
st_geometry_asbinary_format オプション
バージョン 12.0.0 の新機能, 122
st_geometry_asbinary_format プロパティ
バージョン 12.0.0 の新機能, 122
st_geometry_astext_format オプション
バージョン 12.0.0 の新機能, 122
st_geometry_astext_format プロパティ
バージョン 12.0.0 の新機能, 122
st_geometry_asxml_format オプション
バージョン 12.0.0 の新機能, 122, 131
st_geometry_asxml_format プロパティ
バージョン 12.0.0 の新機能, 122
ST_GEOMETRY_COLUMNS
バージョン 12.0.0 の新機能、統合ビュー, 121
st_geometry_describe_type オプション
バージョン 12.0.0 の新機能, 122
st_geometry_describe_type プロパティ
バージョン 12.0.0 の新機能, 122
st_geometry_dump システムプロシージャ
バージョン 12.0.0 の新機能, 120
st_geometry_interpolation オプション
バージョン 12.0.1 の新機能, 85
st_geometry_load_shapefile システムプロシージャ
バージョン 12.0.1 の新機能, 85
st_geometry_on_invalid オプション
バージョン 12.0.0 の新機能, 122
st_geometry_on_invalid プロパティ
バージョン 12.0.0 の新機能, 122
st_geometry_predefined_srs システムプロシージャ
バージョン 12.0.0 の新機能、内部でのみ使用, 120
バージョン 16.0 の動作変更, 29
st_geometry_predefined_uom システムプロシージャ
バージョン 12.0.0 の新機能、内部でのみ使用, 120
バージョン 16.0 の動作変更, 29
ST_SPATIAL_REFERENCE_SYSTEMS
バージョン 12.0.0 の新機能、統合ビュー, 121
ST_UNITS_OF_MEASURE
バージョン 12.0.0 の新機能、統合ビュー, 121
ST_WithinDistanceFilter
メソッド、バージョン 12.0.1 の強化, 85
START AT 句
バージョン 12.0.1 の強化, 88
START DATABASE 文
バージョン 12.0.0 の強化, 127
バージョン 16.0 の強化, 12, 16, 25
バージョン 16.0 の動作変更, 32, 37, 70
START LOGGING 文
バージョン 12.0.0 の強化, 190
START SERVER 文
バージョン 12.0.0 の新機能, 139
START SYNCHRONIZATION SCHEMA CHANGE 文
バージョン 12.0.0 の新機能, 157
バージョン 12.0.0 の新機能、Mobile Link, 161
バージョン 12.0.1 の強化、OnDemand, 74
バージョン 16.0 の新機能, 47
StatementDescribes プロパティ
バージョン 11.0.1 の新機能、接続プロパティ, 197
バージョン 11.0.1 の新機能、データベースプロパティ, 198
StatementPostAnnotatesSimple プロパティ

バージョン 11.0.1 の新機能、接続プロパティ, 197
バージョン 11.0.1 の新機能、データベースプロパティ, 198
StatementPostAnnotatesSkipped プロパティ
バージョン 11.0.1 の新機能、接続プロパティ, 197
バージョン 11.0.1 の新機能、データベースプロパティ, 198
StatementPostAnnotates プロパティ
バージョン 11.0.1 の新機能、接続プロパティ, 197
バージョン 11.0.1 の新機能、データベースプロパティ, 198
StatisticsCleaner プロパティ
バージョン 12.0.0 の新機能, 134
STOP ENGINE 文
バージョン 12.0.0 で廃止, 153
STOP SERVER 文
バージョン 12.0.0 の新機能, 139
STOP SYNCHRONIZATION SCHEMA CHANGE 文
バージョン 12.0.0 の新機能, 157
バージョン 12.0.0 の新機能、Mobile Link, 161
StreamErrorParameters プロパティ
バージョン 11.0.1 の強化、Ultra Light, 210
StreamHTTPParms.getE2eePublicKey メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 78
StreamHTTPParms.getExtraParameters メソッド [Ultra Light J]
バージョン 12.0.1 の新機能, 81
StreamHTTPParms.isRestartable メソッド [Ultra Light J]
バージョン 12.0.1 の新機能, 81
StreamHTTPParms.setE2eePublicKey メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 78
StreamHTTPParms.setExtraParameters メソッド [Ultra Light J]
バージョン 12.0.1 の新機能, 81
StreamHTTPParms.setRestartable メソッド [Ultra Light J]
バージョン 12.0.1 の新機能, 81
StreamHTTPParms.setURLSuffix メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 82
StreamHTTPParms.setZlibCompression メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 78
StreamHTTPParms.setZlibDownloadWindowSize メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 78
StreamHTTPParms.setZlibUploadWindowSize メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 78
StreamHTTPParms.zlibCompressionEnabled メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 78
StreamHTTPSParms.setTrustedCertificates メソッド [Ultra Light J]
バージョン 12.0.1 の強化, 79
StreamsUsed プロパティ
バージョン 11.0.0 の新機能, 226
STRIP 句
バージョン 16.0 でサポート終了, 37
SuppressInfoForDataTypes ODBC 接続パラメータ
バージョン 16.0 の新機能, 13
SUSER_NAME 関数
バージョン 16.0 の動作変更, 35
SWITCHOFFSET 関数
バージョン 12.0.0 の新機能, 135
Sybase Adaptive Server Enterprise 15.0 のサポート
バージョン 16.0 でサポート終了、Mobile Link, 52
Sybase Central
データベースのアップグレード, 304
バージョン 10 以降のデータベースの再構築, 294
バージョン 12.0.0 の動作変更, 188
バージョン 16.0 の新機能, 68
バージョン 16.0 の動作変更, 68
バージョン 9 以前のデータベースの再構築, 301
バージョンのサポート, 283
古いデータベースの管理, 289
Sybase Central の Mobile Link 12 プラグイン
バージョン 12.0.0 で再設計, 155
Sybase Central の新機能
バージョン 11.0.0, 270
バージョン 11.0.1, 211
バージョン 12.0.0, 185
バージョン 12.0.1, 111
バージョン 16.0, 68
Sybase Central の動作の変更

バージョン 11.0.0, 274
バージョン 11.0.1, 212
バージョン 12.0.0, 188
Sybase Central の動作変更
バージョン 16.0, 68
Sybase Central プラグイン
バージョン 16.0 の新機能, 68
SAP Sybase IQ サポート
バージョン 12.0.1 の新機能、Mobile Link, 96
Sybase Relay Server のホスティングサービス
バージョン 11.0.0 の新機能, 258
Sybase Replication Server
バージョン 12.0.0 でサポートされない, 117
SYNC_PROFILE_OPTION_VALUE 関数
バージョン 11.0.1 の強化、Ultra Light, 210
SynchronizationSchemaChangeActive プロパティ
バージョン 12.0.0 の新機能, 132
SYNCHRONIZE 文
バージョン 12.0.0 の新機能, 157
SyncResult インタフェース
バージョン 16.0 の強化、Ultra Light J, 61
SYS_REPLICATION_ADMIN_ROLE システムロール
バージョン 16.0 の新機能、Mobile Link クライアント, 45
SYS_RUN_REPLICATION_ROLE システムロール
バージョン 16.0 の新機能、Mobile Link クライアント, 45
SYS_SPATIAL_ADMIN_ROLE グループ
バージョン 12.0.0 の新機能, 129
SYSCERTIFICATE
バージョン 16.0 の新機能, 26
SYSCOLSTAT
バージョン 16.0 の強化, 26
SYSDATETIMEOFFSET 関数
バージョン 12.0.0 の新機能, 135
SYSEXTERNENVOBJECT
バージョン 16.0 の強化, 26
SYSGROLEGRANT
バージョン 16.0 の新機能, 4
SYSGROLEGRANTEXT
バージョン 16.0 の新機能, 4
SYSGROLEGRANTS
バージョン 16.0 の新機能, 4
SYSGROUP
バージョン 16.0 の動作変更, 4
SYSGROUPS
バージョン 16.0 で廃止, 4
SYSHISTORY
バージョン 16.0 の強化, 26
SYSINDEXES
バージョン 11.0.0 の動作変更, 244
SYSJAR
バージョン 16.0 の強化, 26
SYSJAVACLASS
バージョン 16.0 の強化, 26
SYSLDAPSERVER
バージョン 16.0 の新機能, 6
SYSLOGINMAP
バージョン 11.0.0 の新機能, 244
SYSLOGINPOLICY
バージョン 11.0.0 の新機能, 244
SYSLOGINPOLICYOPTION
バージョン 11.0.0 の新機能, 244
SYSMIRROROPTION システムビュー
バージョン 12.0.0 の新機能, 123
SYSMIRRORSERVER
バージョン 12.0.0 の新機能、システムビュー, 123
SYSMIRRORSERVEROPTION
バージョン 12.0.0 の新機能、システムビュー, 123
SYSOBJECT
バージョン 11.0.0 の動作変更, 244
バージョン 12.0.0 の強化, 143
バージョン 16.0 の強化, 26
SYSPROCPARM
バージョン 12.0.0 の強化、システムビュー, 143
SYSREMOTE USER
バージョン 16.0 の強化, 26
SYSSEQUENCE
バージョン 12.0.0 の新機能, 143
SYSSEQUENCEPERM
バージョン 12.0.0 の新機能, 143
SYSSPATIALREFERENCESYSTEM
バージョン 12.0.0 の新機能、システムビュー, 121
syssyncresult システムテーブル
バージョン 16.0 の強化、Ultra Light, 59
SYSTAB
バージョン 16.0 の強化, 26
SYSTABAUTH
バージョン 16.0 の強化, 4
SYSTABCOL

バージョン 12.0.0 の強化、システムビュー, 143
SYSTABLEPERM
バージョン 16.0 の強化, 4
SYSTEM PROCEDURE AS DEFINER 句
ALTER DATABASE、バージョン 16.0 の新機能, 30
CREATE DATABASE、バージョン 16.0 の新機能, 29
バージョン 16.0 の新機能、CREATE DATABASE 文, 29
SYSTEXTCONFIG
バージョン 11.0.0 の新機能, 244
SYSTEXTIDX
バージョン 11.0.0 の新機能, 244
バージョン 16.0 の強化, 26
SYSTEXTIDXTAB
バージョン 11.0.0 の新機能, 244
SYSUNITOFMEASURE
バージョン 12.0.0 の新機能、システムビュー, 121
SYSUSER
システムビュー、バージョン 16.0 の動作変更, 37
バージョン 16.0 の強化, 4, 6, 26
SYSUSERAUTHORITY
バージョン 16.0 の動作変更, 4
SYSUSERTYPE
バージョン 12.0.0 の強化, 143
SYSVIEW
バージョン 16.0 の強化, 26

T

TCP/IP
バージョン 12.0.0 の動作変更, 148
バージョン 12.0.1 の動作変更, 74
TcpIpAddresses プロパティ
バージョン 11.0.0 の新機能, 226
TDS 通信プロトコル
バージョン 12.0.0 の強化, 142
TERM BREAKER 句
バージョン 12.0.0 の新機能, 126
ThreadDeadlocksAvoided プロパティ
バージョン 12.0.0 の新機能, 133
ThreadDeadlocksReported プロパティ
バージョン 12.0.0 の新機能, 133
Timeout プロトコルオプション
バージョン 12.0.1 の強化, 75
timestamp_with_time_zone_format オプション
バージョン 12.0.0 の新機能, 131
TimestampWithTimeZone クラス
バージョン 12.0.0 の新機能、Mobile Link, 160
TIMESTAMP WITH TIME ZONE データ型
バージョン 12.0.0 の新機能, 139
バージョン 12.0.0 の新機能、Mobile Link, 158
TINYINT UNSIGNED データ型
バージョン 16.0 の新機能, 25
TLS
バージョン 11.0.0 の動作変更, 248
バージョン 12.0.0 の強化、Mobile Link, 156
バージョン 16.0 の強化、Mobile Link, 42
tls_type 引数
ENCRYPTION 接続パラメータ, 7
tls_type プロトコルオプション
-ec データベースサーバオプション, 7
バージョン 16.0 で廃止、Mobile Link, 53
TLS のサポート
バージョン 16.0 の動作変更, 7
TODATETIMEOFFSET 関数
バージョン 12.0.0 の新機能, 135
TOP 句
バージョン 12.0.1 の強化, 88
total_download_row_count
バージョン 16.0 の強化、Ultra Light.NET, 61
バージョン 16.0 の強化、Ultra Light C/C++, 61
TRACEBACK 関数
バージョン 16.0 の動作変更, 34
Transact-SQL 外部ジョイン
バージョン 16 へのアップグレードのトラブルシューティング, 300
TREAT 関数
バージョン 12.0.0 の新機能, 120
TRIM_ARRAY 関数
バージョン 16.0 の新機能, 9
truncate_with_auto_commit オプション
バージョン 11.0.0 でサポート終了, 254
TRUNCATE TEXT INDEX 文
バージョン 11.0.0 の新機能, 232
TRUNCATE 権限
バージョン 16.0 の新機能, 1
trusted_certificates_file オプション
バージョン 16.0 の新機能, 6, 46
trusted_certificates_file プロパティ
バージョン 16.0 の新機能, 20
trusted_certificates オプション

バージョン 16.0 の新機能、Mobile Link, 43
TRY 文
バージョン 16.0 の新機能, 24
tsql_hex_constant オプション
バージョン 11.0.0 でサポート終了, 254

U

UCA 照合
バージョン 12.0.0 の強化, 145
UDP リスナ
バージョン 16.0 の動作変更, 51
ul_column_name_type 列挙 [Ultra Light C++]
バージョン 12.0.1 の新機能, 82
ul_stream_error 構造体
バージョン 11.0.0 の強化、Ultra Light, 266
ul_sync_info
バージョン 11.0.0 の動作変更、Ultra Light, 270
ul_sync_stats_download
バージョン 16.0 の強化、Ultra Light C/C++, 60
ul_sync_stats_upload
バージョン 16.0 の強化、Ultra Light C/C++, 60
ulcpp11.cpp
バージョン 16.0 で削除された API、Ultra Light, 67
uleng12 ユーティリティ
バージョン 12.0.0 の動作変更, 118
uliface.h
バージョン 16.0 で削除された API、Ultra Light, 67
ULjDbT
バージョン 11.0.1 の Ultra Light J ユーティリティ, 211
ULjException クラス [Ultra Light J]
バージョン 12.0.1 の強化, 81
ULODBC サーバクラス
バージョン 11.0.0 の新機能, 242
ULResultSetSchema.GetColumnName メソッド [Ultra Light C++]
バージョン 12.0.1 の強化, 82
Ultra Light
データベースとアプリケーションのアップグレード, 331
Ultra Light.NET
バージョン 11.0.0 の強化、Ultra Light, 269
バージョン 12.0.1 の強化、Ultra Light, 106
バージョン 16.0 の強化、Ultra Light, 61
非同期の同期, 106
Ultra Light ALTER DATABASE SCHEMA FROM FILE 文
バージョン 11.0.0 の新機能、Ultra Light, 265
Ultra Light ALTER SYNCHRONIZATION PROFILE 文
バージョン 11.0.0 の新機能、Ultra Light, 264
Ultra Light C/C++
バージョン 16.0 の強化、Ultra Light, 60
Ultra Light C/C++ API
バージョン 11.0.0 の強化、Ultra Light, 268
Ultra Light CREATE SYNCHRONIZATION PROFILE 文
バージョン 11.0.0 の強化、Ultra Light, 264
Ultra Light DROP SYNCHRONIZATION PROFILE 文
バージョン 11.0.0 の強化、Ultra Light, 264
Ultra Light ESQL
バージョン 11.0.0 の強化、Ultra Light, 268
Ultra Light for M-Business Anywhere
廃止される, 107
バージョン 11.0.0 の強化、Ultra Light, 269
バージョン 12.0.0 の強化、Ultra Light, 175
バージョン 16.0 で削除された API、Ultra Light, 66
Ultra Light isolation_level オプション
バージョン 11.0.0 の新機能、Ultra Light, 265
Ultra Light J
コードサンプル ロケーション, 105
バージョン 12.0.1 の強化、Ultra Light, 107
バージョン 16.0 の強化, 61
Ultra Light Java Edition データベース
バージョン 16.0 の新機能, 56
Ultra Light J コードサンプル ロケーション
バージョン 12.0.1 の新機能、Ultra Light, 105
Ultra Light J データベース
バージョン 12.0.0 で削除または廃止予定の機能, 183
バージョン 12.0.0 の強化, 183
バージョン 12.0.1 の新機能, 104
Ultra Light J データベース転送ユーティリティ BlackBerry, 211
Ultra Light J の新機能
バージョン 11.0.1, 210
Ultra Light J の動作の変更
バージョン 11.0.1, 210, 211
Ultra Light LOAD TABLE 文
バージョン 11.0.0 の強化、Ultra Light, 266
Ultra Light SELECT 文

バージョン 11.0.0 の強化、Ultra Light, 264
Ultra Light SYNCHRONIZE 文
バージョン 11.0.0 の強化、Ultra Light, 264
Ultra Light アプリケーション
バージョン 11.0.0 の新機能、Ultra Light ロートラバーサル, 268
Ultra Light クライアント
バージョン 11.0.0 の強化、Ultra Light, 266
Ultra Light データベース
旧バージョンのアップグレード, 331
バージョン 11.0.0 の強化, 264
バージョン 11.0.0 の新機能, 263
バージョン 12.0.0 の新機能, 171
バージョン 12.0.1 の新機能, 104
バージョン 16.0 の新機能, 56
Ultra Light データベース検証ユーティリティ (ulvalid)
バージョン 11.0.0 の新機能、Ultra Light, 265
Ultra Light データベース初期化ユーティリティ (ulinit)
バージョン 11.0.0 の強化、Ultra Light, 267
Ultra Light データベースの XML へのアンロードユーティリティ (ulunload)
バージョン 11.0.0 の強化、Ultra Light, 267
Ultra Light テーブル
バージョン 11.0.0 の新機能、Ultra Light ロートラバーサル, 268
Ultra Light テーブル API
バージョン 11.0.0 の強化、Ultra Light, 268
Ultra Light 動作の変更
バージョン 12.0.1, 110
Ultra Light の新機能
バージョン 11.0.0, 263
バージョン 11.0.1, 209
バージョン 12.0.0, 171
バージョン 12.0.1, 104
バージョン 16.0, 56
Ultra Light の動作の変更
バージョン 11.0.0, 269
バージョン 12.0.0, 175
バージョン 16.0, 65
Ultra Lite
エラーメッセージ、新規, 110
UltraLite_Connection オブジェクト
バージョン 11.0.0 の新機能、Ultra Light ロートラバーサル, 268
ulvalid ユーティリティ
バージョン 11.0.0 の新機能、Ultra Light, 265
UNION 文
バージョン 11.0.0 の強化, 234
バージョン 11.0.0 の動作変更, 251
UniqueIdentifier プロパティ
バージョン 11.0.0 でサポート終了, 253
Unix
バージョン 16.0 の強化, 27
UNIX
バージョン 11.0.0 の新機能, 239
バージョン 12.0.0 の強化, 144
バージョン 12.0.0 の動作変更, 118
UNLOAD TABLE 文
バージョン 11.0.0 の強化, 234
バージョン 12.0.1 の動作変更, 94
UNLOAD 文
バージョン 11.0.0 の強化, 234
バージョン 16.0 の強化, 25
UNNEST 演算子
バージョン 16.0 の新機能, 9
up_pad_size
Relay Server, 54
UPDATE 文
バージョン 11.0.0 の強化, 234, 241
バージョン 11.0.0 の動作変更, 251
バージョン 11.0.1 の強化, 202
バージョン 12.0.0 の動作変更, 147
バージョン 12.0.1 の強化, 88
upload_update
競合の検出、バージョン 12.0.0 で廃止, 170
user_estimates オプション
バージョン 11.0.0 の強化, 234
USER_NAME 関数
バージョン 16.0 の動作変更, 35
UserDefinedCounterRate01 プロパティ
バージョン 12.0.1 の新機能, 91
UserDefinedCounterRate02 プロパティ
バージョン 12.0.1 の新機能, 91
UserDefinedCounterRate03 プロパティ
バージョン 12.0.1 の新機能, 91
UserDefinedCounterRate04 プロパティ
バージョン 12.0.1 の新機能, 91
UserDefinedCounterRate05 プロパティ
バージョン 12.0.1 の新機能, 91
UserDefinedCounterRaw01 プロパティ
バージョン 12.0.1 の新機能, 91
UserDefinedCounterRaw02 プロパティ
バージョン 12.0.1 の新機能, 91
UserDefinedCounterRaw03 プロパティ

バージョン 12.0.1 の新機能, 91
UserDefinedCounterRaw04 プロパティ
バージョン 12.0.1 の新機能, 91
UserDefinedCounterRaw05 プロパティ
バージョン 12.0.1 の新機能, 91
username システムパラメータ
バージョン 12.0.0 の新機能、Mobile Link, 160
UTCTimestampCatalog プロパティ
バージョン 16.0 の新機能, 21
UTC TIMESTAMP 空間値
バージョン 12.0.0 の動作変更, 148
UTF-16 エンコード
バージョン 11.0.0 の強化、CSCONVERT 関数, 240
バージョン 11.0.0 の強化、LOAD TABLE 文, 240
バージョン 11.0.0 の強化、UNLOAD 文, 240
UTF-8
バージョン 11.0.0 の動作変更、URL のサポート, 235
uuid_has_hyphens オプション
バージョン 11.0.0 でサポート終了, 254
バージョン 12.0.0 で回復した機能, 130
uuid_has_hyphens プロパティ
バージョン 12.0.0 で回復した機能, 130
UUID.toString メソッド [Ultra Light for M-Business Anywhere]
バージョン 12.0.1 の新機能, 82

V

V_$TRANSACTION Oracle システムビュー
バージョン 12.0.0 の動作変更、Mobile Link, 166
VALIDATE DATABASE 文
バージョン 12.0.0 の動作変更, 151
VALIDATE LDAP SERVER 文
バージョン 16.0 の新機能, 6
VALIDATE TEXT INDEX 文
バージョン 12.0.1 の新機能バージョン 12.0.1 の強化, 88
VALIDATE 権限
バージョン 16.0 で廃止, 38
VALIDATE 文
バージョン 12.0.1 の強化, 88
VARCHAR データ型
バージョン 12.0.0 の強化, 139
varray (Oracle)
バージョン 11.0.1 の新機能, 208
varray のサポート
バージョン 12.0.0 の強化, 158
verify_password_function オプション
バージョン 16.0 の強化, 13
viewcert ユーティリティ
バージョン 16.0 の強化, 19
VIM メッセージタイプ
バージョン 11.0.0 でサポート終了, 263
Vista
バージョン 11.0.0 の強化, 220
Visual Studio
バージョン 11.0.0 の動作変更, 282
Visual Studio .NET
バージョン 11.0.0 の動作変更, 282
Visual Studio 2005 プロジェクトファイル
バージョン 16.0 で削除された機能、Ultra Light, 65
Visual Studio 2012 のサポート
バージョン 16.0 の新機能, 25
VSS
バージョン 11.0.0 の新機能, 220

W

WaitStartTime プロパティ
バージョン 11.0.1 の新機能、接続プロパティ, 197
WaitType プロパティ
バージョン 11.0.1 の新機能、接続プロパティ, 197
WebClientLogFile プロパティ
バージョン 11.0.0 の新機能, 239
WebClientLogging プロパティ
バージョン 11.0.0 の新機能, 239
webservice_sessionid_name オプション
バージョン 16.0 の新機能, 19
webservice_sessionid_name プロパティ
バージョン 16.0 の新機能, 20
Web サーバ
バージョン 11.0.0 の動作変更, 248
Web サービス
バージョン 11.0.0 の動作変更, 248
バージョン 12.0.0 の強化, 140
バージョン 12.0.1 の強化, 89
Web サービスクライアント
バージョン 11.0.1 の動作変更, 207
win32 ディレクトリ

バージョン 11.0.0 の動作変更, 282
Windows
バージョン 11.0.0 の動作変更, 251
Windows 2000
バージョン 12.0.0 でサポート対象外, 153
Windows CE から Windows Mobile への変更
バージョン 11.0.0 の動作変更, 281
Windows IIS のサポート
バージョン 12.0.0 の強化、Relay Server, 164
Windows Mobile
バージョン 11.0.0 の強化, 239
バージョン 12.0.0 の強化, 143
Windows Vista
バージョン 11.0.0 の強化, 220
Windows サービス
バージョン 12.0.0 の動作変更, 118
Windows パフォーマンスモニタ
バージョン 11.0.0 の強化, 224
WITH CONTENT LOGGING 句
バージョン 11.0.0 の新機能, 230
WITH FILE NAME LOGGING 句
バージョン 11.0.0 の新機能, 231
WITH NULLS NOT DISTINCT 句
バージョン 12.0.0 の新機能, 136, 187
WITH ROW LOGGING 句
バージョン 11.0.0 の新機能, 231
WITH SHARE MODE 句、REFRESH MATERIALIZED VIEW 文
バージョン 11.0.0 の新機能, 219
Word ドキュメント、全文インデックス
バージョン 12.0.0 の新機能, 126
WRITE_CLIENT_FILE 関数
バージョン 11.0.0 の新機能, 227
WriteChecksums プロパティ
バージョン 12.0.0 の新機能, 132
WRITECLIENTFILE 権限
バージョン 16.0 で廃止, 38
WriteNoPK ロック
バージョン 12.0.0 の新機能, 144

X

X64 ディレクトリ
バージョン 11.0.0 の動作変更, 282
Xcode 4.2 サンプル
バージョン 12.0.1 の新機能, 83
XMLTYPE データ型
バージョン 16.0 の新機能、Mobile Link, 41
バージョン 16.0 の新機能、Mobile Link プラグイン, 49
xp_cmdshell システムプロシージャ
バージョン 16.0 の動作変更, 29
xp_get_mail_error_code システムプロシージャ
バージョン 16.0 の新機能, 22
xp_get_mail_error_text システムプロシージャ
バージョン 16.0 の新機能, 22
xp_getenv システムプロシージャ
バージョン 16.0 の新機能, 22
バージョン 16.0 の動作変更, 29
xp_read_file システムプロシージャ
バージョン 12.0.0 の強化, 134
バージョン 16.0 の動作変更, 29
xp_sendmail システムプロシージャ
バージョン 16.0 の動作変更, 29, 34
xp_startmail システムプロシージャ
バージョン 16.0 の動作変更, 29, 32, 34
xp_startsmtp システムプロシージャ
バージョン 12.0.0 の強化, 134
バージョン 16.0 の強化, 22
バージョン 16.0 の動作変更, 29, 32, 34
xp_stopmail システムプロシージャ
バージョン 16.0 の動作変更, 29, 34
xp_stopsmtp システムプロシージャ
バージョン 16.0 の動作変更, 29, 34
xp_write_file システムプロシージャ
バージョン 16.0 の動作変更, 29

Z

zlib 圧縮
バージョン 16.0 の新機能、Ultra Light, 57

あ

アクセシビリティ
バージョン 12.0.0 の強化, 186
アクセスプラン
バージョン 12.0.1 の強化, 77
アサーション
バージョン 16.0 の動作変更, 32
アスタリスク
バージョン 11.0.1 の動作変更、全文検索, 204
圧縮されたカラム
バージョン 11.0.0 の強化, 241
アップグレード
ALTER DATABASE を使用したバージョン 10 以降のデータベース, 307

dbupgrad を使用したバージョン 10 以降のデータベース, 306
Mobile Link, 319
Mobile Link サーバ, 328
Mobile Link システムオブジェクト, 320
Mobile Link 統合データベース, 320
Mobile Link リモートデータベース, 328
SQL Anywhere, 289
SQL Remote, 334
Sybase Central を使用したバージョン 10 以降のデータベース, 304
Ultra Light のデータベースとアプリケーションの概要, 331
注意事項, 291
データベースミラーリング, 308
同期スクリプト, 329
トラブルシューティング, 316
バージョン 16.0 の強化, 12
バージョン 16 のデータベースの再構築, 297, 304
バージョン 16 へのアップグレードについて, 283
マテリアライズドビューが含まれるデータベース, 290
モニタ, 335
アップグレードユーティリティ (dbupgrad)
バージョン 11.0.1 の動作変更, 206
バージョン 16.0 の強化, 12, 30
バージョン 16.0 の動作変更, 33
アドレス領域ト
バージョン 12.0.0 の動作変更, 147
暗号化
バージョン 11.0.0 の強化, 221, 240
バージョン 12.0.0 の強化, 135
バージョン 16.0 の強化, 28
バージョン 16.0 の新機能、Ultra Light, 59
暗号化制御
バージョン 16.0 で削除された API、Ultra Light, 65
安全な同期
バージョン 12.0.1 の強化, 82
アンロード
バージョン 11.0.0 の新機能、変数へ, 217
バージョン 12.0.0 の強化, 130
アンロードユーティリティ (dbunload)
バージョン 10 以降のデータベースの再構築, 296
バージョン 9 以前のデータベースの再構築, 302
バージョン 11.0.0 の強化, 222
バージョン 11.0.0 の動作変更, 251
バージョン 11.0.1 の強化, 198
バージョン 12.0.0 の強化, 129, 130
バージョン 16.0 の強化, 19
バージョン 16 へのアップグレードのトラブルシューティング, 300
アーカイブのバックアップ
バージョン 11.0.1 の強化, 203
アーカイブバックアップ
バージョン 12.0.0 の強化, 128

い

移行
モニタ, 335
イベント
バージョン 11.0.1 の強化, 211
イベントトレースデータ (ETD) ファイル管理ユーティリティ (dbmanageetd)
バージョン 16.0 の新機能, 10
イベントトレースデータファイル
バージョン 16.0 の新機能, 10
イベントのトレース
バージョン 16.0 の新機能, 10
バージョン 16.0 の新機能、Mobile Link, 42
印刷
バージョン 11.0.0 の強化、Interactive SQL, 275
インジケータ変数
バージョン 12.0.0 でサポート対象外、short int, 154
インストールディレクトリ
バージョン 11.0.0 の動作変更, 282
インデックス
バージョン 11.0.0 の強化, 217
バージョン 11.0.0 の強化、パフォーマンス, 241
バージョン 11.0.0 の動作変更, 252
バージョン 12.0.0 の強化, 144
バージョン 16.0 の強化, 19
バージョン 16.0 の動作変更, 31
インデックスのパフォーマンスの向上
バージョン 16.0 の新機能、Ultra Light, 59
インデックスヒント
バージョン 12.0.0 の強化, 136
バージョン 12.0.0 の動作変更, 152

インポートウィザード
バージョン 11.0.0 の強化, 278
バージョン 12.0.1 の強化, 85
バージョン 16.0 の強化, 70
引用符付き識別子
バージョン 11.0.0 の動作変更, 250

う

ウィザード
バージョン 11.0.0 の強化, 275
バージョン 11.0.0 の新機能、ウィザード, 272
ウォッチリスト
バージョン 11.0.0 の新機能, 230

え

エイリアス
バージョン 16.0 の強化, 29
エクスポートウィザード
バージョン 11.0.0 の強化, 278
エスケープ文字
バージョン 12.0.0 の強化、ユーティリティ, 117
バージョン 12.0.1 の動作変更, 94
エラーコード
バージョン 12.0.0 の新機能、Relay Server, 165
エラーメッセージ
バージョン 12.0.0 の新機能、Relay Server, 165
バージョン 12.0.1 の新機能、Ultra Light, 110
エラーレポート
バージョン 11.0.0 の強化, 280
バージョン 12.0.0 の強化, 129
演算子の優先度、全文検索
バージョン 11.0.1 の動作変更, 203
エンディアン
バージョン 11.0.0 の強化, 240
バージョン 11.0.1 の動作変更, 207
エンドツーエンド暗号化
バージョン 11.0.0 の強化、Ultra Light, 267
バージョン 11.0.0 の新機能, 261

お

お気に入りリスト
バージョン 11.0.0 の新機能, 273
バージョン 12.0.0 の強化, 190
オプションウィンドウ
バージョン 16.0 の強化, 70
オプションのウォッチリスト
バージョン 11.0.0 の新機能, 230
オプティマイザ
バージョン 11.0.0 の強化, 217
バージョン 11.0.1 の強化, 201
バージョン 12.0.1 の強化, 90
バージョン 16.0 の強化, 9
オートコミット
バージョン 11.0.0 の動作変更, 252
オーバーフローエラー
バージョン 12.0.0 の強化, 145

か

外部アンロード
バージョン 11.0.0 の強化, 242
外部参照
集合関数, 317
外部ジョイン
バージョン 12.0.0 の強化, 126
バージョン 16 へのアップグレードのトラブルシューティング, 300
外部ログイン
バージョン 12.0.0 の動作変更, 151
[概要] ウィンドウ枠
バージョン 11.0.0 の新機能, 271
[概要] タブ
バージョン 12.0.0 の強化, 187
書き込みチェックサム
バージョン 12.0.0 の新機能, 127
カスタマイズセキュリティ機能キー
バージョン 16.0 の動作変更, 32
カスタム照合
内部アンロードでのみ保持される, 299
カスタム認証メッセージ
バージョン 16.0 の強化、Ultra Light, 59
カタログ
バージョン 11.0.0 の動作変更, 244
活性タイムアウト
バージョン 16.0 の強化、Mobile Link (mlreplay), 44
カラム
バージョン 16.0 の強化, 28
カラム属性
再構築されたデータベースで次に使用可能な値を保持, 294
カラムマッピング
バージョン 16.0 の動作変更, 49
カラム名

バージョン 16.0 の動作変更, 8
バージョン 16.0 の動作変更、Ultra Light, 66
カラム名の送信
バージョン 16.0 の動作変更, 8
バージョン 16.0 の動作変更、Mobile Link, 50
バージョン 16.0 の動作変更、Ultra Light, 66
監査
バージョン 11.0.0 の強化, 221
バージョン 11.0.0 の新機能, 272
監視サーバ
バージョン 12.0.1 の強化, 75
関数
集合, 317
バージョン 11.0.1 の強化, 200
管理ツール
バージョン 11.0.1 の強化, 203
バージョン 11.0.1 の動作変更, 207
バージョン 16.0 の強化, 67
カーソル
バージョン 12.0.0 の強化, 133

き

疑問符
SQL スクリプトでの使用、バージョン 12.0.0 で廃止, 170
逆引用符
バージョン 12.0.0 の新機能, 145
キャッシュ
バージョン 12.0.0 の強化, 146
バージョン 12.0.1 の動作変更, 94, 96
バージョン 16.0 の強化、Ultra Light, 60
キャッシュ読み込み：ワークテーブル統計値
バージョン 11.0.0 の新機能, 227
競合の検出
バージョン 16.0 で削除、Mobile Link, 53
競合の処理
バージョン 16.0 の強化、Mobile Link プラグイン, 48
強制的な競合モード
バージョン 12.0.0 で廃止、Mobile Link, 170
バージョン 16.0 で削除、Mobile Link, 53
行番号
バージョン 11.0.0 の強化、Interactive SQL, 275
共有メモリ
バージョン 12.0.0 の強化、Relay Server, 165
バージョン 12.0.0 の動作変更, 149
近接検索
バージョン 11.0.1 の動作変更, 203
バージョン 16.0 の強化, 28
キーペアジェネレータユーティリティ (createkey)
バージョン 11.0.0 の新機能, 261
キーボードショートカット
バージョン 11.0.0 の新機能, 271
バージョン 12.0.0 の動作変更, 191

く

クイックスタート
バージョン 16 へのデータベースのアップグレード, 292
空間参照系の作成ウィザード
バージョン 12.0.0 の新機能, 121
空間データ
バージョン 12.0.0 の新機能, 119
バージョン 12.0.1、ロード機能の強化, 84
空間データ型
バージョン 12.0.0 の新機能、Mobile Link, 156
空間データの同期
バージョン 12.0.0 の新機能、Mobile Link, 158
[空間ビューア]
バージョン 12.0.0 の新機能, 120
[空間プレビュー] タブ
バージョン 12.0.0 の新機能, 120
クエリ
バージョン 16.0 の強化, 9
クエリ最適化
バージョン 11.0.1 の強化, 201
バージョン 12.0.1 の強化, 77
クエリパフォーマンス
バージョン 11.0.1 の強化, 201
クライアントファイル、書き込み
バージョン 11.0.0 の新機能, 217
クライアントファイル、読み込み
バージョン 11.0.0 の新機能, 217
クライアントライブラリ
バージョン 16.0 の動作変更, 36
クラウドのサポート
バージョン 12.0.1 の新機能, 73
グラフィカルなプラン
バージョン 16.0 の強化, 28
グローバルチェックサム
バージョン 12.0.0 の強化, 127, 139
バージョン 12.0.0 の動作変更, 146

け

警告
モニタバージョン 12.0.1 の新機能, 114, 115
警告メッセージを Android に表示
バージョン 16.0 の強化、Ultra Light J, 63
計算カラム
バージョン 11.0.0 の動作変更, 248
結果
バージョン 11.0.0 の新機能、読み込み専用に指定, 273
[結果] ウィンドウ枠
バージョン 11.0.0 の強化, 276
バージョン 12.0.0 の強化, 190
結果セット
バージョン 11.0.0 の新機能、読み込み専用に指定, 273
バージョン 16.0 の強化, 69
結果セットからの SQL 文の生成
バージョン 11.0.0 の強化, 276
結果セットの編集の無効化
バージョン 11.0.0 の新機能, 273
権限
バージョン 16.0 の新機能, 1
バージョン 16.0 の動作変更、ロールによる置き換え, 38
検証
バージョン 11.0.0 の動作変更, 223
バージョン 12.0.0 の強化, 144
バージョン 12.0.0 の動作変更, 151
検証ユーティリティ (dbvalid)
バージョン 11.0.0 の強化, 223
バージョン 12.0.0 の動作変更, 151

こ

更新のチェック
バージョン 12.0.0 の強化, 185
高速ランチャ
バージョン 11.0.0 の強化, 270
バージョン 12.0.0 の強化, 186
バージョン 16.0 の強化, 69
互換性
Ultra Light ソフトウェアのアップグレード, 331
Ultra Light のアップグレード, 331
クライアント／サーバ, 289
データベースとデータベースサーバ, 289
問題, 289
コピーノード
バージョン 12.0.0 の新機能, 122
コマンドラインユーティリティ
Ultra Light でのアップグレード, 331
Ultra Light の複数バージョン, 331
アップグレード, 292
複数バージョン, 292
コンソールユーティリティ (dbconsole)
バージョン 16.0 の新機能, 71
コードサンプル ロケーション
Ultra Light J, 105

さ

再開可能な HTTP
バージョン 12.0.1 の新機能, 81
再構築
dbunload を使用したバージョン 9 以前のデータベース, 302
アップグレードを行う前の注意事項, 291
制限, 298
トラブルシューティング, 299
バージョン 16 のデータベース, 292, 304
バージョン 9 以前のデータベース, 301
再構築の失敗
バージョン 16 へのデータベースのアップグレード, 299
再構築ユーティリティ
バージョン 12.0.0 でサポート対象外, 153
最小キャッシュサイズ
バージョン 12.0.1 の動作変更, 94
再表示
バージョン 16.0 の強化、Mobile Link プラグイン, 49
サイレントインストール
バージョン 11.0.0 の動作変更, 256
バージョン 12.0.0 の新機能, 142
サポートされるプラットフォーム
バージョン 11.0.0 の動作変更, 251
サポートユーティリティ
バージョン 11.0.0 の強化, 280
バージョン 12.0.0 の動作変更, 118
サポートユーティリティ (dbsupport)
バージョン 16.0 の動作変更, 33
サポートユーティリティ (dbsupport)
バージョン 12.0.0 の強化, 129
サンプル証明書
バージョン 16.0 の動作変更, 7, 51

サンプルデータベース
バージョン 11.0.0 の強化、SQL Anywhere, 281
バージョン 12.0.0 の強化, 145
サーバ側のバックアップ
バージョン 11.0.1 の強化, 203
サーバ起動同期の Listener
バージョン 11.0.0 の強化, 261
サーババックグラウンド起動ユーティリティ (dbspawn)
バージョン 16.0 の強化, 19
サーバファーム
バージョン 12.0.0 の強化、Mobile Link, 156
サーバファームでのサーバ起動同期
バージョン 11.0.0 の強化, 261
サーバファーム用の mlreplay ユーティリティ
バージョン 12.0.0 の新機能、Mobile Link, 156
サーバプロパティ
バージョン 11.0.0 の新機能, 226
サーバメッセージ
バージョン 11.0.0 の強化, 242
サーバメッセージウィンドウ
バージョン 12.0.0 の強化, 146
サーバライセンス取得ユーティリティ (dblic)
バージョン 12.0.0 の強化, 129
バージョン 12.0.1 の強化, 95
サービス
バージョン 11.0.1 の強化, 198
バージョン 12.0.0 の強化, 129, 144
バージョン 12.0.0 の動作変更, 118
サービス作成ウィザード
バージョン 11.0.1 の強化, 211
サービスユーティリティ
バージョン 12.0.0 の動作変更, 118
サービスユーティリティ (dbsvc)
バージョン 11.0.1 の強化, 198
バージョン 12.0.0 の強化, 129
バージョン 12.0.0 の動作変更, 154

し

シェイプファイル
バージョン 12.0.1 の強化, 85
バージョン 12.0.1、ロード機能の強化, 84
式
バージョン 12.0.0 の強化, 145
識別子
バージョン 12.0.0 の強化, 136, 145, 150
バージョン 16.0 の強化, 29
システムオブジェクト
バージョン 16.0 の強化、Mobile Link プラグイン, 49
システムセキュリティ機能キー
バージョン 16.0 の動作変更, 32
システムテーブル
バージョン 11.0.0 の動作変更, 244
バージョン 16.0 の強化, 26
システムパス
Ultra Light のユーティリティ, 331
ユーティリティ, 292
システムビュー
バージョン 11.0.0 の動作変更, 244
バージョン 16.0 の強化, 26
システムプロシージャ
バージョン 12.0.1 の動作変更、OnDemand, 74
バージョン 16.0 の動作変更, 29
実行プラン
バージョン 11.0.0 の強化, 243
バージョン 11.0.1 の強化, 201
マテリアライズドビュー
バージョン 12.0.0、抽出されたデータベースでの SQL Remote の初期化, 170
自動起動オプション
Relay Server、バージョン 16 でサポート終了, 56
自動キャッシュサイズ決定
バージョン 12.0.1 の新機能、Ultra Light, 105
自動接続プール
バージョン 12.0.0 の強化, 140
シャドウテーブルインデックス
バージョン 12.0.0 の新機能、Mobile Link, 156
シャドウページング
バージョン 16.0 で削除された機能、Ultra Light, 66
集合関数
外部参照, 317
終了時書き込み型の永続ストア
バージョン 16.0 で削除された機能、Ultra Light, 66
照合
バージョン 12.0.0 の動作変更, 152
状態
バージョン 12.0.0 で廃止, 194
情報ユーティリティ (dbinfo)
バージョン 12.0.0 の強化, 146
証明書
バージョン 11.0.0 の動作変更, 281

バージョン 16.0 の動作変更, 7
証明書作成ユーティリティ (createcert)
バージョン 16.0 の強化, 19
証明書ビューアユーティリティ (viewcert)
バージョン 16.0 の強化, 19
初期化ユーティリティ (dbinit)
バージョン 12.0.0 の強化, 129, 146
バージョン 12.0.0 の動作変更, 152
バージョン 16.0 の強化, 28, 29
初期キャッシュサイズ
バージョン 12.0.1 の動作変更, 94
ショートカット
バージョン 11.0.0 の新機能, 274
新機能
バージョン 11.0.0, 215
バージョン 11.0.1, 197
バージョン 12.0.0, 117
バージョン 12.0.1, 73
バージョン 16.0, 1
バージョン 9.0.2 以前, vii
シングルステップ
バージョン 11.0.0 の新機能, 273
進行メッセージ
バージョン 12.0.0 の強化, 134
診断トレーシング
バージョン 16.0 の動作変更, 32
信用できる証明書
バージョン 12.0.1 の強化, 79
シーケンス
バージョン 12.0.0 の新機能, 124
シーケンスジェネレータの作成ウィザード
バージョン 12.0.0 の新機能, 124

す

スクリプト
アップグレード, 329
スクリプト化されたアップロード
バージョン 10.0.0 の動作変更、Mobile Link, 51
スクリプトの無視
バージョン 11.0.1 の新機能、Mobile Link (mlsrv11), 208
スケールアウト
バージョン 12.0.0 の新機能, 122
スタックサイズ
バージョン 12.0.0 の強化、Windows Mobile, 143
ステータス情報ファイル
バージョン 12.0.0 の動作変更, 149
ストアドプロシージャ
バージョン 12.0.1 の強化, 76
スマートフォン
バージョン 12.0.1 の強化、Ultra Light, 106
スマートフォン
バージョン 11.0.0 の強化, 281
バージョン 11.0.0 の新機能、Ultra Light, 269
スレッド
バージョン 12.0.0 の動作変更, 147

せ

正規表現
バージョン 11.0.0 の新機能, 216
バージョン 11.0.1 の動作変更, 204
制限
バージョン 11.0.0 の増加, 243
正常性と統計情報
バージョン 11.0.0 の新機能, 271
セキュリティ
バージョン 11.0.0 の強化、Ultra Light, 267
セキュリティ機能
バージョン 11.0.0 の動作変更, 257
バージョン 16.0 の強化, 14, 15
バージョン 16.0 の動作変更, 32, 68
接続
バージョン 11.0.0 の動作変更, 249
接続アシスタント
バージョン 11.0.0 の新機能, 271
バージョン 12.0.0 で削除, 185
[接続] ウィンドウ
バージョン 11.0.0 の強化, 270
バージョン 11.0.0 の新機能, 271
バージョン 12.0.0 の強化, 185
バージョン 12.0.0 の新機能, 112
バージョン 12.0.1 の強化、OnDemand, 73
バージョン 16.0 の動作変更, 67
接続状態を Android に表示
バージョン 16.0 の強化、Ultra Light J, 63
接続プロパティ
バージョン 11.0.0 の新機能, 225
バージョン 12.0.0 の新機能, 132
バージョン 12.0.1 の強化, 76
バージョン 12.0.1 の新機能, 91
バージョン 16.0 の強化, 20
接続プール
バージョン 12.0.0 の強化, 140

バージョン 12.0.0 の新機能, 128
接続文字列をクリップボードにコピー
バージョン 11.0.0 の新機能, 271
設定ファイル
バージョン 11.0.0 の強化, 222
バージョン 12.0.0 の強化, 117, 129
選択性推定
バージョン 12.0.0 の強化, 145
全文検索
バージョン 11.0.0 の新機能, 216
バージョン 11.0.1 の動作変更, 203
バージョン 16.0 の強化, 28

そ

相関名
バージョン 12.0.0 の強化, 139
相対パス
バージョン 16.0 の動作変更, 30
即時ビュー
バージョン 11.0.0 の新機能, 218
バージョン 16.0 の強化, 23
測定単位の作成ウィザード
バージョン 12.0.0 の新機能, 121

た

対象ユーザに対するログの冗長性
バージョン 11.0.1 の強化, 209
代替サーバ名
バージョン 12.0.0 の動作変更, 149
タイムスタンプ
バージョン 12.0.0 の強化, 131
ダイレクトページスキャン
バージョン 11.0.0 の新機能、Ultra Light テーブル API, 268
ダウンロード確認
バージョン 11.0.0 の動作変更, 261
ダウンロード削除サブセット
バージョン 12.0.0 の新機能、Mobile Link, 156
ダウンロード専用テーブル
バージョン 12.0.1 の新機能、Ultra Light, 105
楕円曲線暗号化の証明書
バージョン 16.0 で廃止, 7
多言語リソース開発キット
バージョン 11.0.0 の動作変更, 252
探索条件
バージョン 12.0.0 の強化, 138
バージョン 12.0.0 の新機能, 136
[断片化] タブ
バージョン 12.0.0 の新機能, 186

ち

チェックサム
バージョン 11.0.0 の強化, 239
バージョン 11.0.0 の動作変更, 248
バージョン 12.0.0 の強化, 127
バージョン 12.0.0 の動作変更, 146
チェックポイント
オートインクリメントカラムのあるテーブル, 294
オートインクリメントカラムのあるテーブルのデータ, 294
バージョン 12.0.0 の動作変更, 147
チェックポイントログ
バージョン 12.0.0 の動作変更, 147
遅延時間
Relay Server、改善, 54
注意事項
アップグレード, 291
抽出ユーティリティ (dbxtract)
バージョン 11.0.0 の強化, 263
バージョン 11.0.0 の動作変更, 252
バージョン 12.0.0 の強化, 170
重複 cookie の更新
Relay Server, 54
重複するテキストインデックス
バージョン 11.0.1 の動作変更, 204

つ

通知サポート
バージョン 16.0 でサポート終了、Mobile Link プラグイン, 52
ツールバー
バージョン 11.0.0 の強化、Interactive SQL, 276

て

ディスクサンドボックス
バージョン 16.0 の新機能, 15
ディスク読み込み：ワークテーブル統計値
バージョン 11.0.0 の新機能, 227
停電
バージョン 12.0.0 の強化, 146
テキストインデックス
バージョン 11.0.0 の新機能, 216
バージョン 11.0.1 の動作変更, 203

バージョン 12.0.1 の強化, 90
バージョン 16.0 の強化, 23
テキスト設定オブジェクト
バージョン 11.0.0 の新機能, 216
テキスト設定オブジェクト作成ウィザード
バージョン 12.0.0 の強化, 187
テキストプラン
バージョン 11.0.0 で廃止予定、Interactive SQL 機能, 275
テキスト補完
バージョン 12.0.1 の強化, 112
バージョン 16.0 の強化, 70
適用式
バージョン 11.0.0 の新機能, 233
デバイス
バージョン 11.0.0 の強化、Ultra Light, 266
バージョン 12.0.0 の強化、Ultra Light, 171
バージョン 12.0.1 の強化、Ultra Light, 106
バージョン 16.0 の強化、Ultra Light, 60
デバッグモード
バージョン 11.0.0 の強化, 275
デフォルトデータベースサーバ
バージョン 11.0.1 の強化, 202
展開ウィザード
バージョン 12.0.0 の強化, 117
電子メール
モニタのバージョン 16.0 の強化, 70
モニタバージョン 12.0.1 の新機能, 114, 115
テンポラリ関数
バージョン 11.0.1 の強化, 200
テンポラリ接続
バージョン 12.0.0 の強化, 127
テンポラリテーブル
バージョン 11.0.0 の動作変更, 252
バージョン 12.0.0 の強化, 133
テンポラリファイル
バージョン 11.0.0 の強化、UNIX, 239
テンポラリプロシージャ
バージョン 11.0.1 の強化, 200
データ型
バージョン 12.0.0 の新機能, 139
バージョン 16.0 の強化, 25
データ型変換
バージョン 12.0.0 の強化, 145
データスクリプト
バージョン 12.0.0 の動作変更、Mobile Link, 166
バージョン 16.0 の新機能、Mobile Link, 41
データソース
バージョン 12.0.0 の強化, 128
データのアンロード
バージョン 12.0.0 の強化, 130
データベース
ALTER DATABASE を使用したバージョン 10 以降のアップグレード, 307
dbunload を使用した再構築, 296
dbunload を使用したバージョン 9 以前の再構築, 302
dbupgrad を使用したバージョン 10 以降のアップグレード, 306
Sybase Central を使用したバージョン 10 以降のデータベースのアップグレード, 304
アップグレード, 289
バージョン 10.0.0 以降からのファイルフォーマットのアップグレード, 304
バージョン 9 以前からのファイルフォーマットのアップグレード, 297
バージョン 9 以前の再構築, 301
比較、バージョン 12.0.1 の新機能, 75
ミラーリングシステムでのデータベースのアップグレード, 313
データベース
Sybase Central の再構築, 294
データベースアップグレードウィザード
使用, 304
バージョン 10 以降のデータベースのアップグレード, 304
データベース暗号化
バージョン 11.0.0 の強化, 221
データベースアンロードウィザード
バージョン 10 以降のデータベースの再構築, 294
バージョン 9 以前のデータベースの再構築, 301
データベースオプション
バージョン 11.0.0 の強化, 216
バージョン 12.0.1 の動作変更、OnDemand, 74
バージョン 16.0 の強化, 19
データベースクリーナー
バージョン 12.0.0 の動作変更, 151
バージョン 12.0.1 の動作変更, 90
データベース検証ウィザード
バージョン 11.0.0 の新機能, 272
データベースサイズ
バージョン 11.0.0 の強化, 241
データベース作成ウィザード

バージョン 11.0.0 の強化, 275
バージョン 12.0.0 の強化, 189
データベースサーバプロパティ
バージョン 12.0.0 の新機能, 133
バージョン 12.0.1 の強化, 76
バージョン 12.0.1 の新機能, 93
バージョン 16.0 の強化, 21
データベースサーバメッセージウィンドウ
バージョン 12.0.0 の強化, 195
データベース抽出ウィザード
バージョン 11.0.0 の強化、Ultra Light, 266
データベースツールのインポートライブラリ
バージョン 11.0.0 の新機能, 247
データベースドキュメントウィザード
バージョン 11.0.0 の新機能, 271
バージョン 12.0.0 の強化, 187
データベースドキュメントの生成
バージョン 11.0.0 の新機能, 271
データベース内の Java
バージョン 12.0.1 の強化, 89
データベースに格納された TLS ID
バージョン 12.0.1 の新機能、Ultra Light, 105
データベースのアップグレード
Sybase Central, 304
制限, 298
データベースの再構築
再構築の失敗, 299
バージョン 11.0.0 の強化, 219
データベースの停止
バージョン 11.0.0 の動作変更, 249
バージョン 12.0.0 の動作変更, 147
データベースのドキュメント化
バージョン 11.0.0 の新機能, 271
バージョン 12.0.0 の強化, 187
[データベースの比較] ウィンドウ
バージョン 12.0.1 の新機能, 75
データベースのミラーリング
マイナーリリースへの (データベースサーバの) アップグレード, 310
データベースバックアップウィザード
バージョン 12.0.0 の強化, 187
データベースファイルフォーマット
アップグレード, 297, 304
データベースプロパティ
バージョン 11.0.0 の新機能, 227
バージョン 12.0.0 の新機能, 92, 132
バージョン 12.0.1 の強化, 76
バージョン 16.0 の強化, 21
データベース変更のロギングの開始
バージョン 12.0.0 の新機能, 187
データベースミラーリング
アップグレード, 308
サポートパッケージの適用, 312
データベースのアップグレード, 313
バージョン 11.0.0 の強化, 219
バージョン 11.0.0 の動作変更, 249
バージョン 11.0.1 の動作変更, 206
バージョン 12.0.0 の強化, 123
バージョン 12.0.0 の動作変更, 149
バージョン 12.0.1 の強化, 75
データベースワーカスレッド
バージョン 12.0.1 の新機能、Mobile Link, 96
テーブル API
バージョン 11.0.0 の強化、Ultra Light, 268
テーブル暗号化
バージョン 11.0.0 の強化, 220, 221
テーブルの編集の無効化
バージョン 11.0.0 の新機能, 273
テーブルマッピングエディタ
バージョン 16.0 の動作変更, 48
テーブルマッピングの方向
バージョン 12.0.1 の動作変更、Mobile Link (mlsrv16), 83
テーブル、ダウンロード専用
バージョン 12.0.1 の新機能、Ultra Light, 105
デーモン
バージョン 12.0.0 の動作変更, 118

と

問い合わせ
バージョン 11.0.0 の新機能、BLOB, 219
バージョン 11.0.0 の新機能、ファイル, 219
同期されないカラムに対する更新
バージョン 12.0.1 の動作変更、Mobile Link, 83
同期ストリームのセキュリティ
バージョン 11.0.0 の新機能, 261
同期パラメータ
バージョン 16.0 の動作変更、Ultra Light, 8
同期プロファイル
バージョン 11.0.0 の強化、Ultra Light, 264
バージョン 11.0.0 の新機能, 260
同期プロファイルオプション
バージョン 16.0 の動作変更、Ultra Light, 8
同期プロファイル作成ウィザード
バージョン 11.0.0 の新機能, 272

同期モデル
バージョン 12.0.0 の動作変更、Mobile Link, 167
バージョン 12.0.1 の新機能、Mobile Link, 99
同期モデル作成ウィザード
バージョン 16.0 の新機能、Mobile Link プラグイン, 49
同期モデル展開ウィザード
バージョン 11.0.0 の強化, 275
バージョン 16.0 の動作変更, 48
同期モデルの削除
バージョン 16.0 の新機能、Mobile Link プラグイン, 48
同期モデルのテスト
バージョン 16.0 の新機能、Mobile Link プラグイン, 47
統計
バージョン 12.0.0 の強化, 124
統計ガバナー
バージョン 12.0.0 の新機能, 124
統計クリーナー
バージョン 12.0.0 の新機能, 124
統合化 Outbound Enabler
バージョン 12.0.0 の新機能、Mobile Link, 156
統合データベース
Adaptive Server Enterprise、SAP Sybase IQ、Oracle、MySQL、または Microsoft SQL Server のアップグレード, 321
IBM DB2 LUW, 323
SQL Anywhere 10.0.0 以降のアップグレード, 320
アップグレード, 320
バージョン 10.0.0 より以前の SQL Anywhere バージョンのアップグレード, 324
統合データベースのサポート
バージョン 12.0.0 の新機能、Mobile Link, 158
動作の変更
バージョン 11.0.0, 215
バージョン 11.0.1, 197
バージョン 12.0.0, 117
バージョン 12.0.1, 73
バージョン 16.0, 1
バージョン 9.0.2 以前, vii
動的なキャッシュサイズの変更
バージョン 12.0.1 の動作変更, 94
独立性レベル
バージョン 11.0.0 の強化、Ultra Light ReadUncommitted, 265
独立性レベル 2
バージョン 12.0.0 の強化, 144
独立性レベル 3
バージョン 12.0.0 の強化, 144
トラブルシューティング
JDBC, 316
Ultra Light のコマンドラインユーティリティ, 331
コマンドラインユーティリティ, 292
バージョン 12.0.0 の強化, 195
バージョン 16 へのデータベースのアップグレード, 299
トランケート削除
バージョン 16.0 の新機能、Ultra Light, 57
トランザクションログ
バージョン 11.0.0 の強化, 216
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, 297
トランザクションログユーティリティ
バージョン 12.0.0 の動作変更, 118
トレース
バージョン 16.0 の新機能、Mobile Link, 42

な

内部アンロード
カスタム照合の保持, 299
長いテキストプラン
バージョン 16.0 の強化, 28
名前付きサブスクリプション
バージョン 12.0.0 の新機能、Mobile Link, 161
名前付きシステムパラメータ
バージョン 12.0.1 の新機能、Mobile Link, 96
名前付きパラメータ
バージョン 16.0 の新機能、Mobile Link, 43

に

二重制御パスワード
バージョン 16.0 の新機能, 13
認証同期スクリプト
バージョン 16.0 の強化、Ultra Light, 59
認証パラメータ
バージョン 12.0.1 の新機能、Mobile Link, 97

ね

ネットワークサーバ
バージョン 12.0.0 の強化, 125
バージョン 12.0.0 の動作変更, 147

[ネットワーク] タブ
バージョン 11.0.0 の新機能, 271

は

廃止された機能
バージョン 12.0.0, 117
バージョン 12.0.1, 73
バージョン 9.0.2 以前, vii
廃止予定機能
バージョン 16.0, 1
廃止予定の機能
バージョン 11.0.0, 215
バージョン 11.0.1, 197
バイト順マーク
バージョン 11.0.0 の強化, 277
バイパスクエリ
バージョン 12.0.1 の強化, 90
ハイフン
バージョン 11.0.1 の動作変更、全文検索, 203
配列演算子
バージョン 16.0 の新機能, 9
パケットサイズ
バージョン 11.0.0 のデフォルト変更, 241
パススルースクリプト作成ウィザード
バージョン 11.0.0 の新機能, 272
パススルーダウンロード作成ウィザード
バージョン 11.0.0 の新機能, 272
パスワード
バージョン 11.0.0 の強化, 220
バージョン 12.0.1 の強化, 75
バージョン 16.0 の強化, 13
バージョン 16.0 の動作変更, 30
破損したデータベース
バージョン 12.0.0 の強化, 146
バックアップ
バージョン 11.0.1 の強化, 203
バージョン 12.0.0 の強化, 128
バックアップとリカバリ
バージョン 11.0.0 の強化, 220
バージョン 12.0.0 の強化, 128
バージョン 16.0 の強化, 13
バックアップユーティリティ (dbbackup)
バージョン 16.0 の強化, 13
バックグラウンド同期
バージョン 11.0.0 の強化、Ultra Light, 266
バージョン 12.0.0 の強化、Mobile Link, 161
ハッシュサイズ
バージョン 16.0 の新機能、Ultra Light, 58
バッチ
バージョン 11.0.0 の強化, 241
パフォーマンス
データベースファイルのアップグレード, 297
バージョン 12.0.0 の強化, 144
バージョン 12.0.1 の強化, 90
バージョン 16.0 の強化, 28
パフォーマンス警告
バージョン 12.0.0 の強化, 195
パフォーマンスデータ
バージョン 16.0 の強化, 67
パフォーマンス統計ユーティリティ (dbstats)
バージョン 12.0.0 の新機能, 144
パフォーマンスモニタ
バージョン 11.0.0 の強化, 224
バージョン
同期スクリプトのアップグレード, 329
バージョン 10.0.0 より以前の SQL Anywhere バージョンのアップグレード
Mobile Link 統合データベース, 324
バージョン 11.0.0
新機能, 215
動作の変更, 215
廃止予定機能, 215
バージョン 11.0.0 の強化
LENGTH 関数, 240
LEN 関数, 240
バージョン 11.0.0 の新機能
概要, 215
バージョン 11.0.1
新機能, 197
動作の変更, 197
廃止予定機能, 197
バージョン 11.0.1 の新機能
概要, 197
バージョン 12.0.0
Interactive SQL の新機能, 185
Mobile Link の新機能, 155
SQL Anywhere の新機能, 119
Sybase Central の新機能, 185
Ultra Light の新機能, 171
新機能, 117
動作の変更, 117
廃止された機能, 117
バージョン 12.0.0 の新機能
概要, 117
バージョン 12.0.0 の動作変更

インデックスヒント, 152
バージョン 12.0.1
Interactive SQL の新機能, 111
Mobile Link の新機能, 96
SQL Anywhere の新機能, 84
Sybase Central の新機能, 111
Ultra Light の新機能, 104
強化、Ultra Light, 106
サポートされなくなった機能、Mobile Link, 103
新機能, 73
動作の変更, 73
動作の変更、Mobile Link, 101
動作の変更、Ultra Light, 110
廃止された機能, 73
廃止された機能、Mobile Link, 103
バージョン 12.0.1 の強化
Ultra Light J, 107
バージョン 12.0.1 の新機能
Mobile Link クライアント, 100
Relay Server, 100
概要, 73
バージョン 12.0.1 の動作の変更
Mobile Link クライアント, 102
Mobile Link プラグイン, 103
バージョン 16.0
Interactive SQL の新機能, 69
Mobile Link の新機能, 40
Relay Server の新機能, 53
Relay Server の動作変更, 55
SQL Anywhere の新機能, 9
Sybase Central の新機能, 68
Ultra Light の新機能, 56
アップグレード, 283
アップグレードの既知の問題, 300
削除された API、Ultra Light, 66
削除された機能、Ultra Light, 65
削除されたプラットフォーム、Ultra Light, 66
新機能, 1
動作の変更, 1
動作の変更、Ultra Light, 65
廃止予定機能, 1
バージョン 16.0 の強化
Ultra Light.NET, 61
Ultra Light C/C++, 60
Ultra Light J, 61
バージョン 16.0 の新機能
概要, 1
バージョン 5
SQL Remote インストール環境のアップグレード, 334
バージョン 9 以前のデータベースの再構築
説明, 297
バージョンごとの変更箇所
11.0.0, 215
11.0.1, 197
12.0.0, 117
12.0.1, 73
16.0, 1
バージョンの変更
バージョン 9.0.2 以前, vii
パーソナルサーバ
バージョン 12.0.0 の動作変更, 148
パーミッション
バージョン 16.0 の動作変更、権限による置き換え, 1

ひ

ビッグエンディアン UTF-16 エンコード
バージョン 11.0.0 の新機能, 240
日付
バージョン 12.0.0 の強化, 135
非同期の同期
Ultra Light.NET, 106
非ブロッキングダウンロード確認スクリプト
バージョン 11.0.0 の強化, 260
バージョン 12.0.0 の強化, 160
ビュー
バージョン 11.0.0 の強化, 240
バージョン 16.0 の動作変更, 30, 37
標準アップグレードの注意事項
説明, 291

ふ

ファイバー
バージョン 12.0.0 の動作変更, 147
ファイル
バージョン 11.0.0 の新機能、問い合わせ, 219
ファイル非表示ユーティリティ (dbfhide)
バージョン 12.0.0 の強化, 129
ファイルフォーマット
アップグレード, 297
ファンクション
バージョン 12.0.1 の動作変更、OnDemand, 74
ファンクション作成ウィザード

バージョン 12.0.0 の強化, 187
フェイルオーバ
バージョン 11.0.0 の新機能、Mobile Link サーバファーム, 259
負荷分散
バージョン 11.0.0 の新機能、Mobile Link サーバファーム, 259
複合データ型
バージョン 16.0 の新機能, 9
複数バージョン
Adaptive Server Anywhere, 292
Ultra Light, 331
フック
バージョン 16.0 の動作変更、Mobile Link, 51
部分インデックススキャン
バージョン 12.0.1 の強化, 90
プラグイン
サポートされているバージョン, 283
プラットフォーム
バージョン 11.0.0 の強化、Ultra Light, 266
バージョン 12.0.0 の強化、Ultra Light, 171
バージョン 12.0.1 の強化、Ultra Light, 106
バージョン 16.0 の強化、Ultra Light, 60
プラン
バージョン 12.0.1 の強化, 77
プランのキャッシュ
バージョン 11.0.0 の強化, 241
バージョン 16.0 の動作変更, 31
プランビューア
バージョン 11.0.0 で廃止予定、機能, 275
バージョン 11.0.0 の強化, 275
プリフェッチ
バージョン 11.0.0 の強化, 220
プロキシテーブル
バージョン 12.0.0 の強化, 144
プロジェクト作成ウィザード
バージョン 16.0 の新機能、Mobile Link プラグイン, 48
ブロック
バージョン 11.0.0 の動作変更, 252
バージョン 16.0 の強化、Mobile Link, 42
プロパティウィンドウ
バージョン 11.0.0 で廃止予定、Ultra Light 機能, 275
文
バージョン 12.0.1 の強化, 76

へ

並列バックアップ
バージョン 11.0.1 の強化, 203
ページサイズ
バージョン 12.0.0 の強化, 129

ま

マイナーリリース
ミラーリング (データベースサーバ), 310
マテリアライズドビュー
アップグレードの考慮事項, 290
バージョン 11.0.0 の強化, 218
バージョン 12.0.0 の強化, 126
バージョン 16.0 の強化, 23
マニュアルの強化
バージョン 11.0.0, 279
バージョン 11.0.1, 214
バージョン 12.0.0, 195
バージョン 12.0.1, 116
マルチスレッドネットワーク処理
バージョン 16.0 の新機能、Mobile Link, 43
マルチプログラミングレベル
バージョン 12.0.0 の強化, 125
バージョン 12.0.0 の動作変更, 147
バージョン 12.0.1 の動作変更, 94

み

未加工の暗号化
バージョン 16.0 の新機能, 14
ミラーデータベース
バージョン 11.0.0 の強化, 219
ミラーリング
アップグレード, 308
バージョン 11.0.0 の強化, 218
バージョン 16.0 の新機能, 11
バージョン 16.0 の動作変更, 31
メジャーバージョンへの (データベースサーバの) アップグレード, 308

む

無視された更新
バージョン 16.0 の新機能、Ultra Light, 57
無視された削除
バージョン 16.0 の新機能、Ultra Light, 57

め

メッセージ制御パラメータ

バージョン 12.0.1 の強化, 103
メモリ
バージョン 12.0.0 の強化、Mobile Link, 156
メモリ内モード
バージョン 11.0.0 の新機能, 224
メモリ不足
バージョン 16.0 の動作変更, 36
メンテナンスプラン
バージョン 12.0.0 の強化, 189
メンテナンスプラン作成ウィザード
バージョン 11.0.1 の強化, 212
バージョン 12.0.0 の強化, 187

も

文字データ型
バージョン 12.0.0 の強化, 139
モニタ
アップグレード, 335
バージョン 11.0.1 の新機能, 213
バージョン 12.0.0 の新機能, 193
バージョン 12.0.0 の動作変更, 194
バージョン 12.0.1 の新機能, 114
バージョン 12.0.1 の動作変更, 115
バージョン 16.0 の新機能, 70
バージョン 16.0 の動作変更, 70
リソースの移行, 335
モニタの動作の変更
バージョン 12.0.0, 194
バージョン 12.0.1, 115

ゆ

ユーザ作成ウィザード
バージョン 11.0.0 の強化, 275
ユーザ定義関数
バージョン 12.0.0 の強化, 144
バージョン 12.0.1 の強化, 76
ユーティリティ
バージョン 12.0.0 の強化, 117
バージョン 12.0.0 の動作変更, 118
バージョン 16.0 の動作変更、Ultra Light, 66
ユーティリティデータベース
バージョン 11.0.0 の動作変更, 248
バージョン 12.0.0 の強化, 130

よ

要求のロギング
バージョン 16.0 の強化, 71
要求ログ
バージョン 16.0 の動作変更, 32
読み込み専用
バージョン 11.0.0 の新機能、ミラーデータベースのアクセス, 219
読み込み専用データベース
バージョン 11.0.1 の動作変更, 213
バージョン 12.0.1 の強化, 75
読み込み専用のスケールアウト
バージョン 12.0.0 の新機能, 122
読み込みロック
バージョン 12.0.0 の強化, 144
予約語
バージョン 12.0.0 の強化, 134
バージョン 12.0.0 の動作変更, 152
バージョン 16.0 の強化, 29
バージョン 16 へのアップグレードのトラブルシューティング, 300
予約語オプション
バージョン 12.0.0 の新機能, 131

ら

ライセンス
バージョン 12.0.0 の強化, 129
バージョン 16.0 の動作変更, 32
ライトウェイトポーリングのアクション変数
バージョン 11.0.1 の強化, 208
ライトウェイトポーリングのリスナキーワード
バージョン 11.0.1 の強化, 208

り

リダイレクタ
バージョン 11.0.0 で廃止予定, 260
バージョン 12.0.0 の削除, 169
リターンコード
バージョン 16.0 の動作変更, 34
リトルエンディアン UTF-16 エンコード
バージョン 11.0.0 の新機能, 240
リフレッシュ
バージョン 11.0.0 の動作変更、マテリアライズドビュー, 249
リモートスキーマ名
バージョン 12.0.1 の新機能、Mobile Link, 99
リモートタスク
バージョン 12.0.1 の新機能、Mobile Link, 98
バージョン 16.0 の強化、Mobile Link プラグイン, 49

リモートデータアクセス
バージョン 11.0.0 の動作変更, 253
バージョン 12.0.0 の強化, 144
バージョン 12.0.1 の強化, 87
リモートデータアクセスクラス
バージョン 16.0 の強化, 11
リモートデータベース
アップグレード, 328
バージョン 12.0.0 の強化, 136
リモートデータベースの集中管理
バージョン 12.0.0 の強化、Relay Server, 164
バージョン 12.0.0 の新機能、Mobile Link, 155
バージョン 12.0.1 の強化、Mobile Link, 97
バージョン 12.0.1 の新機能、Mobile Link, 99
リモートテーブルの所有者
バージョン 16.0 の変更、Mobile Link プラグイン, 52
リモートプロシージャ
バージョン 12.0.1 の新機能, 87

る

ルートログインポリシー
バージョン 16.0 の強化, 5

ろ

ログインポリシー
バージョン 11.0.0 の新機能, 216
バージョン 12.0.1 の強化, 75
バージョン 16.0 の強化, 5, 13, 68
ログ変換ユーティリティ
バージョン 12.0.0 の動作変更, 118
ログ変換ユーティリティ (dbtran)
バージョン 11.0.0 の強化, 223
ロック
バージョン 11.0.0 の新機能、Interactive SQL, 273
バージョン 11.0.0 の動作変更, 252
バージョン 11.0.1 の動作変更、Interactive SQL, 213
バージョン 12.0.0 の強化, 144
バージョン 12.0.0 の動作変更, 150
バージョン 16.0 の強化、Mobile Link, 42
ローカウント
バージョン 12.0.0 の強化, 141
ローカルテンポラリテーブル
バージョン 16.0 の強化, 19
ロー制限
バージョン 12.0.1 の強化, 82
ロード
バージョン 11.0.0 の新機能、値から, 218
バージョン 11.0.0 の新機能、カラムから, 217
バージョン 11.0.0 の新機能、トランザクションログから, 218
ロートラバーサル
バージョン 11.0.0 の強化、Ultra Light, 268
ローのキャッシュ
バージョン 16.0 の強化、Ultra Light, 60
ローのレポート
バージョン 16.0 の強化、Ultra Light J, 61
ロール
バージョン 16.0 の新機能, 1
バージョン 16.0 の新機能、Mobile Link クライアント, 45
ロールバックログ
バージョン 11.0.0 の動作変更, 251

わ

ワイド文字のデータストリーミング
バージョン 16.0 の強化、Ultra Light C/C++, 61
ワーカー
バージョン 12.0.0 の動作変更, 147