

新機能 Open Server™ 12.5.1 と SDK 12.5.1 Microsoft Windows、Linux、および UNIX 版

ドキュメント ID : DC00071-01-1251-02

改訂 : 2006 年 4 月

このマニュアルでは、Open Server™ 12.5.1 と Software Developer's Kit (SDK) 12.5.1 (ESD の最新バージョン ESD#13) の新しい機能について説明します。

SDK は、次のコンポーネントで構成されています。

- Open Client™ (DB-Library™)
- Embedded SQL™/C (ESQL/C)
- Embedded SQL/COBOL (ESQL/COBOL)
- Sybase® Adaptive Server® Enterprise (ASE) ODBC ドライバ (Microsoft Windows および Linux)
- Sybase Adaptive Server Enterprise (ASE) OLE DB Provider (Microsoft Windows のみ)
- Adaptive Server Enterprise (ASE) ADO.NET Data Provider (Microsoft Windows のみ)
- Extended Architecture (XA)
- jConnect™ for JDBC™

後続の ESD で新機能が追加されたため、このマニュアルには次の内容が追加されます。

トピック	ページ
製品の互換性	3
ESD#13 の新機能	6
新しい jConnect 6.05 接続プロパティ	6
新しい jConnect 5.5 接続プロパティ	7
ESD#11 のマニュアル情報と変更点	9

トピック	ページ
ESD#12 の新機能	9
Linux AMD64 (Opteron)/EM64T、HP Itanium 32 ビット版、および 64 ビット版での SSL Plus のサポート	9
MIT Kerberos のサポート	10
DB-Library 版 MIT Kerberos	11
HP Itanium 32 ビット版での ESQ/COBOL のサポート	11
分散トランザクションでの ASE OLE DB プロバイダの使用	11
利用できるようになった OLE DB DSN マイグレーション・ツール	13
ASE OLE DB プロバイダの HA フェールオーバー	16
ESD#11 の新機能	19
Sun Solaris 9 x86 (32 ビット版) での MIT Kerberos のサポート	19
ディレクトリ・サービスの sql.ini とインタフェース・ファイルのサポート	19
ODBC DSN マイグレーション・ツール	22
ODBC と OLE DB に対するブックマーク・オペレーションとパルク・オペレーションのサポート	23
jConnect でのディレクトリ・サービスの sql.ini とインタフェース・ファイルのサポート	24
ESD#10 の新機能	25
HP-UX 11.11 (または HP-UX 11i v1.0) 32 ビット版と 64 ビット版プラットフォームのサポート	25
Linux AMD64 (Opteron)/EM64T での XA 64 ビット版のサポート	26
Microsoft Windows での MIT Kerberos のサポート	26
スクロール可能カーソル	26
ESD#9 の新機能	32
分散トランザクションの使用	32
ESD#8 の新機能	36
MIT Kerberos の拡張サポート	36
新しいプラットフォームとオペレーティング・システムのサポート	37
IBM AIX 64 ビット版の LDAP の機能	37
Windows での CyberSafe Kerberos クライアントのアップグレード	38
SSLPlus 3.1.x のセキュリティ修正	38
Windows 2000 での COBOL コンパイラのアップグレード	39
ESD#7 の新機能	39
XA 64 ビット版のサポート・プラットフォームの追加	39
ESD#6 の新機能	40
Sybase 製の新しい ASE OLE DB プロバイダ	40
ESD#5 の新機能	41
BCP 暗号化カラム	41
韓国語の文字に対するサポートの強化	42
SSLPlus 3.1.10H のセキュリティ修正	42

トピック	ページ
ODBC と OLE DB での MIT Kerberos のサポート	42
Windows での SDK 12.5.1 のインストール	43
Sybase 製 ASE ODBC ドライバのサポート・プラットフォームの拡張	43
ESD#4 の新機能	44
診断の機能強化	44
非同期通信のサポート	45
DB-Library の新しい設定変数	46
Adaptive Server Enterprise ODBC ドライバ Linux Intel 32 ビット版	48
新しい ASE ODBC ドライバ	48
ESD#3 の新機能	49
Adaptive Server Enterprise ADO.NET Data Provider バージョン 1.1 Microsoft Windows 版	49
ESD#2 の新機能	50
Solaris と Microsoft Windows での SSL Plus 3.1.5 + SBGSE のサポート	50
Solaris、Microsoft Windows XP、Microsoft Windows 2003 での IPV6 のサポート	51
Microsoft Windows 版の DB-Library LDAP の機能	52
ESD#1 の新機能	52
Sun Solaris 8 (SPARC 64 ビット版) での XA サポート	52

製品の互換性

表 1 に、Open Client と Open Server のプラットフォームの互換性を示します。

注意 Sun Solaris 10 x64 は ESD#13 以降でサポートされます。

表 1: Open Client と Open Server のプラットフォームの互換性

プラット フォーム	オペレーティ ング・システム・レ ベル	C および C++ コンパイラ	COBOL コン パイラ	Kerberos バージョン	LDAP (Light- weight Directory Access)	SSL (Secure Sockets Layer)
HP-UX 11.0 32 ビット版	HP-UX 11.0 パッチ・バン ドル 990P	HP C/ANSI 11.00.00 HP ANSI C++ B3910B A.03.10	Micro Focus Server Express 2.0.10	CyberSafe Trust Broker 2.1、MIT 1.3.6	Netscape LDAP 4.1	Certicom SSL Plus 3.1.15
HP-UX 11.0 64 ビット版	HP-UX 11.0 パッチ・バン ドル 990P	HP C 11.00.00 ANSI HP ANSI C++ B3910B A.03.10	使用不可	使用不可	Netscape LDAP 4.1	Certicom SSL Plus 3.1.15

製品の互換性

プラット フォーム	オペレーティ ング・システム・レ ベル	CおよびC++コンパイラ	COBOLコン パイラ	Kerberos バージョン	LDAP (Light- weight Directory Access)	SSL (Secure Sockets Layer)
HP-UX 11.11 (またはHP-UX 11i v1.0) 32ビット版	HP-UX 11i v1	HP C/ANSI C B.11.11.10 HP ANSI C++ B3910B A.03.10	Micro Focus Server Express 4.0	CyberSafe Trust Broker 2.1、MIT 1.3.6	Netscape LDAP 4.1	Certicom SSL Plus 3.1.15
HP-UX 11.11 (またはHP-UX 11i v1.0) 64ビット版	HP-UX 11i v1 パッチ・バン ドル 990P	HP C 11.11.10 ANSI HP ANSI C++ B3910B A.03.10	使用不可	MIT 1.4.3	Netscape LDAP 4.1	Certicom SSL Plus 3.1.15
HP Itanium 32 ビット版	HP-UX 11.23	HP C++/ANSI C B3910B A.0550	Micro Focus Server Express 4.0 SP2	使用不可	Netscape LDAP 4.1	Certicom SSL Plus 5.0.6f
HP Itanium 64 ビット版	HP-UX 11.23	HP C++/ANSI C B3910B A.0550	使用不可	使用不可	Netscape LDAP 4.1	Certicom SSL Plus 5.9.6h
HP Tru64	Digital UNIX TRU64 5.0a	C++ 6.0-010	DEC COBOL 2.7	CyberSafe Trust Broker 2.1	Netscape LDAP 4.1	Certicom SSL Plus 3.1.15
IBM AIX 32 ビット版	AIX 5.2	C++ 5.0.22	Micro Focus Server Express 2.0.10	CyberSafe Trust Broker 2.1	使用不可	Certicom SSL
IBM AIX 64 ビット版	AIX 5.2	C++ 5.0.22	使用不可	MIT 1.4.3	使用不可	Certicom SSL
Linux on POWER 32ビット	Red Hat Enterprise Linux AS 3.0	IBM XL C/C++ Advance Edition V7.0	使用不可	使用不可	使用不可	使用不可
Linux on POWER 64ビット版	Red Hat Enterprise Linux AS 3.0	IBM XL C/C++ Advance Edition V7.0	使用不可	使用不可	使用不可	使用不可
Linux AMD64 (Opteron)/ EM64T	Red Hat Enterprise AS 3.0	GCC 3.2.3 (Red Hat Linux 3.2.3 - 42)	使用不可	MIT 1.2.7	使用不可	Certicom SSL Plus 5.0.4m
Linux Intel 32 ビット版	Red Hat AS 2.1 または United Linux/ SuSe SLES 8.0/ UL1.0	オペレーティング・シ ステムでサポートされ るCコンパイラ	使用不可	MIT 1.3.1	Netscape LDAP 4.1	Certicom SSL
Linux Itanium 64ビット版	Red Hat Advance Server 2.1	GCC C 2.96	使用不可	使用不可	使用不可	使用不可

プラットフォーム	オペレーティング・システム・レベル	C および C++ コンパイラ	COBOL コンパイラ	Kerberos バージョン	LDAP (Light-weight Directory Access)	SSL (Secure Sockets Layer)
Linux PowerPC	Red Hat AS 3.0 または United Linux/ SuSe SLES 9.0 (SLES 9)	XL C/C++ Advance Edition V7.0	使用不可	使用不可	使用不可	使用不可
SGI 32 ビット版	IRIX 6.5.18	MIPSPro C7.3.x MIPSPro C++ 7.3.x	使用不可	使用不可	Netscape LDAP 4.0	使用不可
SGI 64 ビット版	IRIX 6.5.18	MIPSPro C7.3.x MIPSPro C++ 7.3.x	使用不可	使用不可	使用不可	使用不可
Sun Solaris 8 (SPARC 32 ビット版)	Solaris 8	Sun C/C++ 6.2	Micro Focus Server Express 2.0.10	CyberSafe Trust Broker 2.1、MIT 1.3.1	Netscape LDAP 4.0	Certicom SSL Plus 3.1.14
Sun Solaris 8 (SPARC 64 ビット版)	Solaris 8	Sun C/C++ 6.2	使用不可	MIT 1.3.1	Netscape LDAP 4.1	Certicom SSL Plus 3.1.14
Sun Solaris 9 x86 (32 ビット版)	Solaris 9	Sun C/C++ 6.2	使用不可	MIT 1.4.1	使用不可	使用不可
Sun Solaris 10 x64 (64 ビット版)	Solaris 10	Studio 10 (Sun C/C++ 5.7)	使用不可	MIT 1.4.1	オープン LDAP 2.2.26	Certicom SSL Plus 5.0.4
Windows NT 32 ビット版	NT 4.0 Service Pack 4	MS C 6.0 (Microsoft Developers Studio、未最適化、開発用)	Micro Focus Net Express 3.1	CyberSafe Trust Broker 2.1、MIT 1.3.6	Netscape LDAP 4.1	Certicom SSL Plus 3.1.15
Windows 2000	Service Pack 3	MS C 6.0 (Microsoft Developers Studio、未最適化、開発用)	Micro Focus Net Express 3.1	CyberSafe Trust Broker 2.1、MIT 1.3.6	Netscape LDAP 4.1	Certicom SSL Plus 3.1.15
Windows 2003	Service Pack 1	MS C 6.0 (Microsoft Developers Studio、未最適化、開発用)	Micro Focus Net Express 3.1	CyberSafe Trust Broker 2.1、MIT 1.3.6	Netscape LDAP 4.1	Certicom SSL Plus 3.1.15
Windows XP		MS C 6.0 (Microsoft Developers Studio、未最適化、開発用)	Micro Focus Net Express 3.1	CyberSafe Trust Broker 2.1、MIT 1.3.6	Netscape LDAP 4.1	Certicom SSL Plus 3.1.15

注意 プログラムを構築するために使用できるコンパイラはこれだけではありません。

ESD#13 の新機能

ESD#13 には、次の新機能があります。

新しい jConnect 6.05 接続プロパティ

表 2 に、jConnect 6.05 で使用できる新しい接続プロパティを示します。接続プロパティの一覧表については、『jConnect for JDBC 6.05 プログラマーズ・リファレンス』の「第 2 章 プログラミング情報」を参照してください。

表 2: jConnect 6.05 の接続プロパティ

プロパティ	説明	デフォルト値
CRC	このプロパティを <i>true</i> に設定すると、返される更新カウントは、実行される文が直接影響する更新と、実行される文の結果として呼び出されるトリガが直接影響する更新とを含む累積数になる。	<i>false</i>
DATABASE	接続情報が Sybase <i>interfaces</i> ファイルから取得される場合、このプロパティを使用して、接続に使用するデータベース名を指定する。 <i>interfaces</i> ファイルの URL にはデータベース名を指定できない。	<i>NULL</i>
DEFAULT_QUERY_TIMEOUT	この接続プロパティを設定すると、この接続で作成されるすべての文に対して、この接続プロパティがデフォルトのクエリ・タイムアウトとして使用される。	0 (タイムアウトなし)
IMPLICIT_CURSOR_FETCH_SIZE	データベースに送信されるすべての <i>select</i> クエリで読み込み専用カーソルをオープンするように jConnect に強制する場合、 <i>SELECT_OPENS_CURSOR</i> プロパティと共にこのプロパティを使用する。 <i>Statement.setFetchSize</i> メソッドによって上書きされない限り、このプロパティで設定された値のフェッチ・サイズがカーソルに適用される。	0
INTERNAL_QUERY_TIMEOUT	このプロパティを使用して、jConnect によって内部的に作成および実行される文に使用するクエリ・タイムアウトを設定する。クエリ・タイムアウトを設定することで、内部コマンドが適切な時間内に完了しなかった場合にアプリケーションがハングすることを防止できる場合がある。	0 (タイムアウトなし)
J2EE_TCK_COMPLIANT	このプロパティを <i>true</i> に設定すると、jConnect ドライバは、J2EE 1.4 TCK テスト・スイートに準拠した動作を有効にする。これによりパフォーマンスが若干低下することがある。このため、デフォルト値の <i>false</i> を使用することが推奨される。	<i>false</i>

新しい jConnect 5.5 接続プロパティ

表 3 に、jConnect 5.5 で使用できる新しい接続プロパティを示します。接続プロパティの一覧表については、『jConnect for JDBC 5.5 プログラマーズ・リファレンス』の「第 2 章 プログラミング情報」を参照してください。

表 3: jConnect 5.5 の接続プロパティ

プロパティ	説明	デフォルト値
CACHE_COLUMN_METADATA	DYNAMIC_PREPARE が <i>true</i> に設定されている場合、このプロパティを使用して、jConnect で連続する実行に対して <i>ResultSetMetadata</i> をキャッシュできるようにする。これにより、パフォーマンスが向上する。	False
CAPABILITY_TIME	このプロパティは、JCONNECT_VERSION が 6 以上である場合にかぎって使用する。jConnect が、TIME データ型をサポートするサーバに接続されている場合、 <i>java.sql.Time</i> 型のすべてのパラメータまたはエスケープ・リテラル {t...} は、TIME として処理される。以前のバージョンの jConnect では、このようなパラメータを DATETIME として扱い、 <i>java.sql.Time</i> パラメータの前に "1970-01-01" を付加する。基本となるデータ型が <i>datetime</i> または <i>smalldatetime</i> である場合は、日付もデータベースに格納される。新しいバージョンの jConnect で TIME が処理される場合、サーバは TIME を基本となるデータ型に変換し、サーバ独自の基底の年を前に付加する。これにより、古いデータと新しいデータの間非互換性が生じる可能性がある。 <i>java.sql.Time</i> の代わりに <i>datetime</i> または <i>smalldatetime</i> データ型を使用する場合、下位互換性を保つために、CAPABILITY_TIME を <i>false</i> にする必要がある。このプロパティを <i>false</i> に設定すると、サーバに TIME データ型を扱う機能があるかどうかにかかわらず、jConnect は <i>java.sql.Time</i> パラメータまたはエスケープ・リテラル {t...} を DATETIME として処理する。このプロパティを <i>true</i> に設定すると、jConnect は ASE 12.5.1 以降に接続した場合に <i>java.sql.Time</i> パラメータを TIME データ型として処理する。 <i>smalldatetime</i> または <i>datetime</i> カラムを使用して時刻値を格納する場合は、このプロパティを <i>false</i> にすることが推奨される。	False

プロパティ	説明	デフォルト値
CAPABILITY_WIDETABLE	カラム名などの JDBC ResultSetMetaData が不要な場合は、パフォーマンス向上のためにこのプロパティを <i>false</i> に設定できる。これにより、ネットワーク上のデータ交換が減少し、パフォーマンスが向上する。EAServer を使用しない場合はデフォルト設定を使用することが推奨される。『jConnect for JDBC プログラマーズ・リファレンス』の第2章にある「ASE 12.5 以降でのワイド・テーブル・サポートの使用」を参照。	True
CRC	このプロパティを <i>true</i> に設定すると、返される更新カウントは、実行される文が直接影響する更新と、実行される文の結果として呼び出されるトリガが直接影響する更新とを含む累積数になる。	False
DATABASE	接続情報が Sybase <i>interfaces</i> ファイルから取得される場合、このプロパティを使用して、接続に使用するデータベース名を指定する。 <i>interfaces</i> ファイルの URL にはデータベース名を指定できない。	Null
IMPLICIT_CURSOR_FETCH_SIZE	データベースに送信されるすべての <i>select</i> クエリで読み込み専用カーソルをオープンするように jConnect に強制する場合、 <i>SELECT_OPENS_CURSOR</i> プロパティと共にこのプロパティを使用する。 <i>Statement.setFetchSize</i> メソッドによって上書きされない限り、このプロパティで設定された値のフェッチ・サイズがカーソルに適用される。	0
QUERY_TIMEOUT_CANCEL_ALL	このプロパティを使用して、読み取りのタイムアウトが発生したときに、接続上のすべての文を強制的にキャンセルする。この動作は、クライアントが <i>execute()</i> を呼び出したときに、たとえば、別のトランザクションが更新中のテーブルからデータを読み取ろうとしたときのデッドロックが原因でタイムアウトが発生した場合に便利。今後の Sun との協議により、このプロパティは、 <i>BE_AS_JDBC_COMPLIANT_AS_POSSIBLE</i> プロパティの影響を受けるプロパティ値と組み合わせられる可能性がある。	False
SERVER_INITIATED_TRANSACTIONS	このプロパティは、サーバによるトランザクション制御を可能にする。デフォルトでは、このプロパティは <i>true</i> に設定されており、Transact-SQL の <i>set chained on</i> を使用して、サーバがトランザクションの開始と制御を行うことができる。このプロパティを <i>false</i> に設定すると、トランザクションは jConnect によって、 <i>transact sql begin tran</i> を使用して開始および制御される。サーバによるトランザクション制御は有効にしておくことが推奨される。	True

プロパティ	説明	デフォルト値
SERVERTYPE	OpenSwitch に接続する場合は、このプロパティを "OSW" に設定できる。これにより、jConnect から OpenSwitch に特定の命令を送信し、OpenSwitch が別のサーバ・インスタンスに移行された場合でも、独立性レベル、テキスト・サイズ、引用符付き識別子、オートコミットなどの初期接続設定を保持できる。	なし
TEXTSIZE	このプロパティを使用して、TEXTSIZE を設定する。ASE および ASA では、デフォルトで、image または text カラムから 32,627 バイトを読み込み可能。jConnect の MDA テーブルがインストールされている場合、jConnect は値を 2GB に変更する。ただし、OpenSwitch に接続する場合にこの値を設定すると、OpenSwitch が別のサーバ・インスタンスに移行するときに接続の設定を保持できる。	2GB

ESD#11 のマニュアル情報と変更点

ESD#11 でのディレクトリ・サービスの sql.ini とインタフェース・ファイルのサポート機能に関する記述が更新されました。詳細については、「[ディレクトリ・サービスの sql.ini とインタフェース・ファイルのサポート](#)」(19 ページ) を参照してください。

ESD#12 の新機能

ESD#12 には、次の新機能があります。

Linux AMD64 (Opteron)/EM64T、HP Itanium 32 ビット版、および 64 ビット版での SSL Plus のサポート

SSL Plus が次のプラットフォームでサポートされるようになりました。

- SSL5.0.4m – Linux AMD64 (Opteron)/EM64T
- SSL5.0.6f – HP Itanium 32 ビット版
- SSL5.9.6h – HP Itanium 64 ビット版

新しい SSL Plus のサポートは次の Sybase 品に適用されます。

- Software Developer's Kit のコンポーネント
 - Client-Library
 - ESQL/C

- Open Server

MIT Kerberos のサポート

HP-UX 11.11 (または HP-UX 11i v1.0) 64 ビット版と IBM AIX 64 ビット版の MIT Kerberos 5 は、バージョン 1.3.6 ベースからバージョン 1.4.3 ベースに変更されています。

HP-UX 11.11 (または HP-UX 11i v1.0) 32 ビット版と HP UX 11.0 32 ビット版では、MIT Kerberos 5 バージョン 1.3.6 がサポートされるようになりました。

表 7 に、Sybase がサポートする MIT Kerberos バージョン 5 のリリースとプラットフォームを示します。

表 4: MIT Kerberos バージョン 5 のリリースとサポートされているプラットフォーム

MIT Kerberos バージョン 5	プラットフォーム
リリース 1.4.3	<ul style="list-style-type: none"> • HP-UX 11.11 (または HP-UX 11i v1.0) 64 ビット版 • IBM AIX 64 ビット版
リリース 1.4.1	<ul style="list-style-type: none"> • Sun Solaris 9 x86 (32 ビット版)
リリース 1.3.6	<ul style="list-style-type: none"> • HP-UX 11.11 (または HP-UX 11i v1.0) 32 ビット版 • HP-UX 11.0 32 ビット版 • Microsoft Windows (Windows NT、Windows 2000、Windows 2003、Windows XP を含む)
リリース 1.3.1	<ul style="list-style-type: none"> • Linux Intel 32 ビット版 • Sun Solaris 8 (SPARC 32 ビット版と 64 ビット版)
リリース 1.2.7	<ul style="list-style-type: none"> • Linux AMD64 (Opteron)/EM64T

上記のプラットフォームでの MIT Kerberos セキュリティ・サービスの使用と設定については、次のマニュアルを参照してください。

- 『Open Client/Server 設定ガイド UNIX 版』の次の章と付録
 - 「第 6 章 セキュリティ・サービスの使い方」
 - 「付録 B 設定ファイル」
 - 「付録 E Kerberos セキュリティ・サービス」
- 『Open Client/Server プログラマーズ・ガイド補足 UNIX 版』の次の章と付録
 - 「第 1 章 Open Client Library/C」
 - 「付録 B 環境変数」
- 『Open Client/Server 設定ガイド Windows 版』の次の章と付録
 - 「第 6 章 セキュリティ・サービスの使い方」
 - 「付録 B 設定ファイル」

- 『Open Client/Server プログラマーズ・ガイド補足 Windows 版』の次の章
 - 「第1章 Open Client と Open Server のアプリケーションの構築」
 - 「第2章 Client Library/C のサンプル・プログラム」

DB-Library 版 MIT Kerberos

DB-Library で MIT Kerberos のセキュリティ・メカニズムがサポートされ、Kerberos のネットワーク・サービスと相互認証サービスを利用できるようになりました。この機能を使用すると、古い Sybase アプリケーションを Kerberos に対応させて Kerberos 認証サービスを使用できます。しかも、変更や再コンパイルの必要性は多くありません。

Kerberos をサポートするために、次の DB-Library マクロが追加されました。

- DBSETLNETWORKAUTH：ネットワーク・ベースの認証を有効または無効にする。
- DBSETLMUTUALAUTH：接続のセキュリティ・メカニズムの相互認証を有効または無効にする。
- DBSETLSERVERPRINCIPAL：必要に応じてサーバのプリンシパル名を設定する。

MIT Kerberos を DB-Library にインストールする方法については、Sybase SDK DB-Lib Kerberos Authentication Option 12.5.1 の『Installation and Release Bulletin』を参照してください。

注意 DB-Library でサポートされる Kerberos セキュリティ・メカニズムの機能は、ネットワーク認証サービスと相互認証サービスのみです。

HP Itanium 32 ビット版での ESQL/COBOL のサポート

HP Itanium 32 ビット版では、ESQL/COBOL がサポートされるようになりました。HP Itanium 32 ビット版の COBOL コンパイラは Micro Focus Server Express 4.0 SP2 です。

分散トランザクションでの ASE OLE DB プロバイダの使用

この機能は Windows でのみサポートされ、分散トランザクションを管理するトランザクション・コーディネータとして Microsoft 分散トランザクション・コーディネータ (MS DTC) が使用されている必要があります。

Sybase は次のプログラミング・モデルをサポートしています。

- MS DTC を直接使用するアプリケーション

- Microsoft Transaction Server (MTS) または (COM+) を使用するアプリケーション

MS DTC のプログラミング

❖ Microsoft 分散トランザクション・コーディネータ (MS DTC) を使用してプログラミングするには

- 1 DtcGetTransactionManager 関数を使用して MS DTC に接続します。MS DTC の詳細については、Microsoft 分散トランザクション・コーディネータのマニュアルを参照してください。
- 2 OLE DB の手順に従って、確立する Sybase ASE 接続ごとに IDBSession を取得します。
- 3 ITransactionDispenser::BeginTransaction 関数を呼び出して MS DTC トランザクションを開始し、そのトランザクションを表す OLE トランザクション・オブジェクトを取得します。
- 4 MS DTC トランザクションリストに登録する IDBSession (OLE DB 接続) ごとに ITransactionJoin を問い合わせ、渡された (手順3で取得した) パラメータ punkTransactionCoord をトランザクション・オブジェクトとして使用して、JoinTransaction を呼び出します。現時点では、Sybase は、分散トランザクションの独立性レベルとして ISOLATION_LEVEL_READCOMMITTED のみをサポートしています。ITransactionOptions はサポートしていません。
- 5 SQL Server を更新するには、IDBCommand を作成して実行するための OLE DB 手順に従います。
- 6 ITransaction::Commit 関数を呼び出して MS DTC トランザクションをコミットします。これでトランザクション・オブジェクトは無効になります。

MTS または COM+ に展開されるコンポーネントのプログラミング

次の手順は、MTS または COM+ で分散トランザクションに関与するコンポーネントの作成方法を説明しています。

❖ MTS または COM+ に展開されるコンポーネントをプログラミングするには

- 1 ASE 接続ごとに IDBSession を作成します。
- 2 実行する更新ごとに IDBCommand を作成して実行します。
- 3 コンポーネントを MTS または COM に展開し、必要に応じてトランザクション属性を設定します。

トランザクションの作成、トランザクションへの関与、トランザクションのコミットまたはロールバックは、COM+、OLE DB サービス、OLE DB プロバイダによって処理されます。

自動トランザクション登録を実行するには、OLE DB サービスが必要です。OLE DB サービスを有効にするには、ルールに従ってデータ・ソースを初期化する必要があります (MS OLE DB のマニュアルを参照)。自動トランザクション登録を有効にするには、`OLE_DB_SERVICES` レジストリと `DBPROP_INIT_OLEDBSERVICES` プロパティ値に `DBPROPVAL_OS_TXNENLISTMENT` ビットを設定するか、接続文字列で `OLE DB Services=2` を渡します。

分散トランザクションでの接続プロパティのサポート

ここでは、接続プロパティについて説明します。

- 分散トランザクション・プロトコル (DistributedTransactionProtocol) – 分散トランザクションをサポートするために使用されるプロトコルを指定するには、XA インタフェース標準または MS DTC OLE ネイティブ・プロトコルのいずれかを使用し、[OLE DB データ・ソース] ダイアログで [分散トランザクション・プロトコル] を選択し、OLE ネイティブ・プロトコルの接続文字列のプロバイダ文字列部分にプロパティ `DistributedTransactionProtocol=OLE` を設定するか、デフォルトのプロトコルである XA を使用します。
- 密結合トランザクション (TightlyCoupledTransaction) – 2 つのリソース・マネージャを使用する分散トランザクションで同一の ASE サーバが指定された場合、「密結合トランザクション」と呼ばれる状況が発生することがあります。この状況が発生した場合、このプロパティを 1 に設定していないと分散トランザクションが失敗することがあります。

つまり、同一の ASE サーバに対して 2 つのデータベース接続をオープンしてから、オープンした接続を同一の分散トランザクションに登録する場合は、`TightlyCoupledTransaction=1` を設定することをおすすめします。このプロパティを設定するには、[OLE DB データ・ソース] ダイアログ・ボックスで [密結合トランザクション] を選択するか、接続文字列のプロバイダ文字列部分でプロパティ `TightlyCoupledTransaction=1` を渡します。

利用できるようになった OLE DB DSN マイグレーション・ツール

OLE DB DSN マイグレーション・ツールを使用して、OLE DB ドライバ・キットから Sybase OLE DB プロバイダにデータ・ソース定義 (DSN: data source definitions) をマイグレートできます。DSN をマイグレートすると、OLE DB ドライバ・キットの代わりに新しい Sybase OLE DB プロバイダが DSN で使用されます。

Sybase ASE OLE DB プロバイダへのマイグレート

Sybase ASE OLE DB プロバイダを使用するように OLE DB アプリケーションをマイグレートするには、OLE DB クライアント・アプリケーションが使用する接続文字列を変更する必要があります。Sybase ASE OLE DB プロバイダの短い名前は、“ASEOLEDB” です。

OLE DB ドライバ・キットと Sybase ASE OLE DB プロバイダの動作で確認されている違いについては、使用しているプラットフォームに応じた、SDK 12.5.1 の『リリース・ノート』を参照してください。

注意 Sybase ASE OLE DB プロバイダの接続文字列の構文については、Sybase Adaptive Server Enterprise OLE DB プロバイダの Microsoft Windows 用の『ユーザーズ・ガイド』を参照してください。

この接続文字列の構文は、OLE DB ドライバ・キットの構文とは異なります。Sybase OLE DB プロバイダは OLE DB ドライバ・キットの構文にも対応していますが、可能な場合は接続文字列の構文を新しい構文にマイグレートすることをおすすめします。

Sybase ドライバへのデータ・ソース名のマイグレート

OLE DB ドライバ・キットから Sybase が作成したドライバにデータ・ソース名 (DSN: Data Source Names) をマイグレートするには、次の 2 つの方法があります。

- [Sybase ASE データ・ソース・アドミニストレータの使用](#)
- [DSN マイグレーション・ツールの使用](#)

Sybase ASE データ・ソース・アドミニストレータの使用

Sybase ASE データ・ソース・アドミニストレータは、既存の OLE DB ドライバ・キットのデータ・ソースのマイグレートと、ASE OLE DB プロバイダ用の新しいデータ・ソースの作成を可能にする GUI プロセスです。

❖ データ・ソース・アドミニストレータを使用してデータ・ソースをマイグレートするには

- 1 “Sybase Data Source Administrator” というタイトルのメイン・ウィンドウで、データ・ソースを選択します。
- 2 [マイグレート] をクリックします。

Sybase データ・ソース・アドミニストレータを使用して、OLE DB データ・ソースを追加、削除、設定、またはテストできます。

DSN マイグレーション・ツールの使用

DSN マイグレーション・ツールを使用して、OLE DB ドライバ・キットから Sybase OLE DB ドライバにデータ・ソースをマイグレートできます。dsnmigrate ツールでは、どの DSN をマイグレートするかを制御するスイッチを使用します。次のコマンドをコマンド・ラインから入力する必要があります。

```
dsnmigrate.exe [/?|/h|/help] [/oledb] [/l|/ul|/sl] [/a|/ua|/sa]
[/dsn|/udsn|/sdsn]=dsn] [/suffix=suffix]
```

OLE DB DSN から変換された新しい Sybase DSN では、変換前と同じ名前が使用されます。

変換スイッチ

次の表に、変換で使用されるスイッチを示し、各スイッチについて説明します。

表 5: 変換スイッチ

スイッチ	結果の説明
/?、/h、/help	このメッセージを表示します。このメッセージは、コマンド・ライン引数を指定せずに dsnmigrate を呼び出した場合にも表示されます。
/oledb	dsnmigrate を OLEDB モードにします。デフォルトでは、ODBC DSN がマイグレートされます。
/l	OLE DB ドライバ・キットのすべてのユーザ DSN とシステム DSN を一覧表示します。
/ul	OLE DB ドライバ・キットのすべてのユーザ DSN を一覧表示します。
/sl	OLE DB ドライバ・キットのすべてのシステム DSN を一覧表示します。
/a	OLE DB ドライバ・キットのすべてのユーザ DSN とシステム DSN を変換します。
/ua	OLE DB ドライバ・キットのすべてのユーザ DSN を変換します。
/sa	OLE DB ドライバ・キットのすべてのシステム DSN を変換します。
/dsn	OLE DB ドライバ・キットの特定のユーザ DSN またはシステム DSN を変換します。
/udsn	OLE DB ドライバ・キットの特定のユーザ DSN を変換します。
/sdsn	OLE DB ドライバ・キットの特定のシステム DSN を変換します。
dsn	変換される DSN の名前。
/suffix	DSN に名前を付ける方法を変更するオプション・スイッチ。このスイッチを使用すると、元の DSN が保持され、新しい DSN に “<dsn>-<suffix>” という名前が付けられます。
suffix	新しい DSN に名前を付けるために使用されるサフィックス。

ASE OLE DB プロバイダの HA フェールオーバ

表 6 に、ASE OLE DB プロバイダがサポートする高可用性 (HA) フェールオーバで使用される新しい接続パラメータを示します。

表 6: HA フェールオーバ接続パラメータ

プロパティ名	説明	必須	デフォルト値
HASession	高可用性を有効にするかどうかを指定する。0 は高可用性を無効にし、1 は高可用性を有効にする。	いいえ	0
SecondaryPort	アクティブ/アクティブ設定またはアクティブ/パッシブ設定でフェールオーバ・サーバとして機能する ASE サーバのポート番号。	はい (HASession が 1 に設定されている場合)	空
SecondaryServer	アクティブ/アクティブ設定またはアクティブ/パッシブ設定でフェールオーバ・サーバとして機能する ASE サーバの名前または IP アドレス。	はい (HASession が 1 に設定されている場合)	空

HA システムにおけるフェールオーバの使用

HA クラスタには、2 つ以上のマシンが含まれます。これらのマシンは、1 つのマシン (またはアプリケーション) が中断した場合にもう 1 つのマシンが両方のマシンの負荷を処理するように設定されています。各マシンは高可用性クラスタのノードです。HA クラスタはシステムが常に稼働していなければならないような環境で使用します。たとえば、クライアントが 1 年 365 日絶えず接続する銀行のシステムなどです。

マシンは、他のマシンのディスクを各マシンで読み込みできるように設定されています。ただし、同時読み込みはできません (フェールオーバで使用するためのディスクは共有ディスクに設定してください)。

たとえば、プライマリ・コンパニオン・サーバである Adaptive Server 1 がクラッシュした場合、セカンダリ・コンパニオンである Adaptive Server 2 は、Adaptive Server 1 を再起動できるようになるまでそのディスク (ディスク 1 ~ 4) を読み込んで、ディスク上のデータベースすべてを管理します。Adaptive Server 1 に接続していたクライアントは、自動的に Adaptive Server 2 に接続されます。

フェールオーバによって、Adaptive Server をアクティブ/アクティブ設定またはアクティブ/パッシブ設定の高可用性クラスタで運用できます。

フェールオーバーが発生すると、プライマリ・コンパニオンに接続していたクライアントは、フェールオーバー・プロパティを使用して、自動的にセカンダリ・コンパニオンへのネットワーク接続を再確立します。フェールオーバーを有効にするには、接続プロパティ **HASession** を“1”（デフォルト値は“0”）に設定します。このプロパティを設定しないと、サーバでフェールオーバーが設定されていても、セッションではフェールオーバーが行われません。**SecondaryServer**（セカンダリ ASE サーバの IP アドレスまたはマシン名）と **SecondaryPort**（セカンダリ ASE サーバのポート番号）のプロパティも設定する必要があります。使用するシステムの HA 設定の詳細については、ASE のマニュアル『高可用性システムにおける Sybase フェールオーバーの使用』を参照してください。

OLE DB プロバイダのドライバでプライマリ ASE サーバの接続エラーが検出されると、最初にプライマリ ASE サーバへの再接続が試行されます。再接続できない場合はフェールオーバーが行われたと見なされます。次に、**SecondaryServer** と **SecondaryPort** に設定された接続プロパティを使用して、セカンダリ ASE サーバへの接続が自動的に試行されます。

フェールオーバーの成功を確認する方法

セカンダリ ASE サーバへの接続が確立されると、ASE OLE DB ドライバは関数の戻り値 **HRESULT** に“E_FAIL”を返します。

フェールオーバーの成功を確認するには、**ERRORINFO** の **dwMinor** フィールド (**IErrorRecords::GetBasicErrorInfo** の戻り値) または **IErrorInfo::GetDescription** から返される説明を確認してください。HA フェールオーバーが成功していれば、**dwMinor** の値は“30130”になります。**IErrorInfo::GetDescription** では次の説明が返されます。文中の **ASEServerName** は、フェールオーバー先のサーバ名を表します。

```
"Sybase server is not available or has terminated your
connection, you have successfully connected to the next
available HA server ASEServerName. All transactions has been
rolled back."
```

注意 フェールオーバーが成功したかどうかの確認には、エラー説明ではなく、**dwMinor** により返されるコードを確認することをおすすめします。

クライアントは、新しい接続を使用して、失敗したトランザクションを再適用しなければなりません。トランザクションのオープン中にフェールオーバーが発生した場合、フェールオーバー前にデータベースにコミットされた変更のみが保持されます。

フェールオーバーの失敗の確認

セカンダリ ASE サーバへの接続が確立されなかった場合も、ASE OLE DB ドライバは関数の戻り値 HRESULT に “E_FAIL” を返します。ただし、ERRORINFO の dwMinor フィールド (IErrorRecords::GetBasicErrorInfo の戻り値) は “30131” になり、IErrorInfo::GetDescription から返される説明は次のようになります。

```
“Connection to Sybase server has been lost, connection to the
next available HA server also failed. All transactions have
been rolled back.”
```

フェールオーバーを確認するサンプル・コード

次のコードは、フェールオーバーのコーディング方法を示しています。

```
/* Declare required variables */
...
/* Open Database connection */
...
/* Perform a transaction */
...
/*Check HRESULT and dwMinor in
ERRORINFO, handle failover */
if (FAILED(hr))
{
    IErrorInfo* pErrorInfo;
    GetErrorInfo(0, &pErrorInfo);
    IErrorRecords * pErrorRecords;
    HRESULT hr1 = pErrorInfo->QueryInterface(
        IID_IErrorRecords,
        (void **)&pErrorRecords);

    if (SUCCEEDED(hr1))
    {
        ERRORINFO errorInfo;
        pErrorRecords->GetBasicErrorInfo(
            0, &errorInfo);
        pErrorRecords->Release();
        if (errorInfo.dwMinor == 30130)
        {
            //successful failover,
            //retry the transaction
        }
    }
}
}
```

ESD#11 の新機能

ESD#11 には、次の新機能があります。

Sun Solaris 9 x86 (32 ビット版) での MIT Kerberos のサポート

Sun Solaris 9 x86 (32 ビット版) で、MIT Kerberos 5 バージョン 1.4.1 がサポートされるようになりました。

表 7 に、Sybase がサポートする MIT Kerberos バージョン 5 のリリースとプラットフォームを示します。

表 7: MIT Kerberos バージョン 5 のリリースとサポートされているプラットフォーム

MIT Kerberos バージョン 5	プラットフォーム
リリース 1.4.1	<ul style="list-style-type: none"> Sun Solaris 9 x86 (32 ビット版)
リリース 1.3.6	<ul style="list-style-type: none"> IBM AIX 64 ビット版 HP-UX 11.11 (または HP-UX 11i v1.0) 64 ビット版 Microsoft Windows (Windows NT、Windows 2000、Windows 2003、Windows XP を含む)
リリース 1.3.1	<ul style="list-style-type: none"> Linux Intel 32 ビット版 Sun Solaris 8 (SPARC 32 ビット版と 64 ビット版)
リリース 1.2.7	<ul style="list-style-type: none"> Linux AMD64 (Opteron)/EM64T

ディレクトリ・サービスの *sql.ini* とインタフェース・ファイルのサポート

この新機能は、*sql.ini* ファイル (Windows 用) と *interfaces* ファイル (UNIX 用) を使用したサーバ情報の提供をサポートします。次のドライバとプロバイダがサポートされます。

- ADO.NET データ・プロバイダ
- ODBC ドライバ
- OLE DB プロバイダ

現時点では、ドライバまたはプロバイダに接続するには、サーバ名、IP アドレス、ASE サーバのポート番号などのさまざまなプロパティを指定する必要があります。*sql.ini* または *interfaces* ファイルを使用することで、エンタープライズ・ネットワークで使用可能なサービスに関する情報を ASE サーバ情報を含めてすべて集中管理できます。

接続文字列

sql.ini または *interfaces* ファイルを識別するには、次の情報を接続文字列に追加する必要があります。単一のディレクトリ・サービス URL (DSURL) または複数の DSURL に接続できます。

単一の DSURL 用の接続文字列

ADO.NET、ODBC、および OLE DB の各ドライバとプロバイダに対して単一の DSURL を提供するには、次のプロパティを次の書式で接続文字列に追加する必要があります。

```
DSURL=file://[path]<filename>[?][servicename]
```

各パラメータの意味は、次のとおりです。

- *path* (オプション) は、*interfaces* ファイルのパスです。指定しない場合は、デフォルト・パスとして Windows では `%SYBASE%¥ini` が使用され、Linux および Mac OS X では `$$SYBASE` が使用されます。
- *filename* は、*interfaces* ファイルの名前です。
 - Windows では、通常、*interfaces* ファイルの名前は `sql.ini` になります。
 - UNIX では、通常、*interfaces* ファイルの名前が `interfaces` になります。
- *servicename* (オプション) は、DSURL によって定義されるサービスの名前であり、サーバ名ともいいます。
 - DSURL に定義されていない場合は、接続文字列の中のサーバ・プロパティが使用されます。
 - サービス名とサーバの両方が存在する場合は、サービス名が使用されます。
 - 両方とも存在しない場合、ドライバまたはプロバイダは、サーバ情報を `sql.ini` または `interfaces` ファイルからロードします。この操作は、`sql.ini` または `interfaces` ファイルにエントリが 1 つだけある場合に実行されます。ファイルに複数のエントリが存在する場合は、エラーが返されます。

ODBC エラーの例：

```
[Sybase][ODBC Driver]Getting more than one servers with the connect string.
```

したがって、*servicename* またはサーバ・プロパティは常に接続 URL で指定する必要があります。通常、*interfaces* ファイルには複数のサーバ・エントリが含まれているからです。

Windows と UNIX の接続文字列の例：

Windows

デフォルト・パスの使用例：

```
DSURL=file://sql.ini?mangol
```

各パラメータの意味は、次のとおりです。

- この例では *path* を省略しているため、`%SYBASE%¥ini` がデフォルト・パスとして使用されます。ただし、`%SYBASE%` が `C:¥Sybase` と定義されている場合、使用されるパスは `C:¥Sybase¥ini` になります。
- *filename* は `sql.ini` です。
- *servicename* は `mangol` です。

明示的なパスの使用例：

```
DSURL=file://¥¥myServer¥myShare¥sql.ini?mango1
```

各パラメータの意味は、次のとおりです。

- path は ¥¥myServer¥myShare と指定されています。これはネットワーク共有ディレクトリを参照する UNC (Universal Naming Convention) パスです。
- filename は *sql.ini* です。
- servicename は *mango1* です。

Linux および Mac OS X

デフォルト・パスの使用例：

```
DSURL=file://interfaces?mango1
```

各パラメータの意味は、次のとおりです。

- この例では path を省略しているため、\$SYBASE がデフォルト・パスとして使用されます。ただし、\$SYBASE を */usr/sybase* と定義した場合、使用されるパスは */usr/sybase* になります。
- filename は *interfaces* です。
- servicename は *mango1* です。

明示的なパスの使用例：

```
DSURL=file:///remote/sybase/interfaces?mango1
```

各パラメータの意味は、次のとおりです。

- path は */remote/sybase* と指定されています。
- filename は *interfaces* です。
- servicename は *mango1* です。

複数の URL 用の接続文字列

DSURL では、複数の LDAP URL と *interfaces* ファイルをサポートできます。Sybase では、LDAP URL とファイル URL が混在した複数の URL もサポートされます。複数の URL を使用すると、ドライバは、*interfaces* ファイルを開くか LDAP サーバに接続するまで、URL を 1 つずつ処理します。

複数 URL の例

接続文字列で複数の URL を使用する例を次に示します。

```
DSURL={file:///mils1/sybase/sql.ini;file:///test/interface}
```

```
DSURL={ldap://SYBLDAP:389/dc=sybase,dc=com??one?sybaseServername=MANGO;file:///test/interface?MANGO}
```

Microsoft Windows、UNIX、SSL 用の interfaces ファイルのフォーマット

interfaces ファイルのフォーマットの詳細については、各プラットフォーム用の『Adaptive Server Enterprise 設定ガイド』を参照してください。

ODBC DSN マイグレーション・ツール

ODBC DSN マイグレーション・ツールを使用して、ODBC ドライバ・キットから Sybase 製 ODBC ドライバにデータ・ソースをマイグレートできます。DSN をマイグレートすると、DNS では、ODBC ドライバ・キットの代わりに新しい Sybase 製 ODBC ドライバが使用されます。

マイグレーション・ツールの使用

dsnigrate ツールでは、どの DSN をマイグレートするかを制御するスイッチを使用します。次のコマンドをコマンド・ラインから入力する必要があります。

```
dsnigrate.exe [/?|/help] [l|/ul|/sl] [/a|/ua|/sa]
[ [/dsn|/udsn|/sdsn]=dsn] [ /suffix=suffix]
```

マイグレートされるすべての DSN には、変換が完了する前に“<dsn>-backup”という名前が付けられます。新しい Sybase DSN が作成され、変換が完了すると、名前が“<dsn>”に変更されます。これにより、既存のアプリケーションを変更せずに継続して実行できます。

変換スイッチ

次の表に、変換で使用されるスイッチを示し、各スイッチについて説明します。

表 8: 変換スイッチ

スイッチ	結果の説明
/?、/h、/help	このメッセージを表示します。このメッセージは、コマンド・ライン引数を指定せずに dsnigrate を呼び出した場合にも表示されます。
/l	ODBC ドライバ・キットのすべてのユーザ DSN とシステム DSN を一覧表示します。
/ul	ODBC ドライバ・キットのすべてのユーザ DSN を一覧表示します。
/sl	ODBC ドライバ・キットのすべてのシステム DSN を一覧表示します。
/a	ODBC ドライバ・キットのすべてのユーザ DSN とシステム DSN を変換します。
/ua	ODBC ドライバ・キットのすべてのユーザ DSN を変換します。
/sa	ODBC ドライバ・キットのすべてのシステム DSN を変換します。

スイッチ	結果の説明
/dsn	ODBC ドライバ・キットの特定のユーザ DSN またはシステム DSN を変換します。
/udsn	ODBC ドライバ・キットの特定のユーザ DSN を変換します。
/sdsn	ODBC ドライバ・キットの特定のシステム DSN を変換します。
dsn	変換される DSN の名前を指定します。
/suffix	DSN に名前を付ける方法を変更するオプション・スイッチ。このスイッチを使用すると、元の DSN が保持され、新しい DSN に “<dsn>-<suffix>” という名前が付けられます。
suffix	新しい DSN に名前を付けるために使用されるサフィックスを指定します。

ODBC と OLE DB に対するブックマーク・オペレーションとバルク・オペレーションのサポート

Sybase では、ODBC ドライバと OLE DB プロバイダに対して、ブックマーク・オペレーションと SQL バルク・オペレーションをサポートします。

ODBC ドライバ向け

SQLBulkOperations を使用するバルク挿入では、SQL_ADD オプションが使用され、SQLSetPos (SQL_UPDATE、SQL_DELETE、SQL_POSITION) を使用してカーソルの位置付け更新と削除が行われます。SQL_ADD と SQLSetPos の使用方法については、Microsoft Developer Network ライブラリで ODBC の『プログラマーズ・リファレンス』を参照してください。

- SQLBulkOperations 関数 (<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/htm/odbcsqlbulkoperations.asp?frame=true>)
- SQLSetPos 関数 (<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/htm/odbcsqlsetpos.asp>)

OLE DB プロバイダ向け

IRowsetLocate インタフェースを使用するブックマーク・オペレーションでは、ブックマークの比較や、ブックマークに基づくローの取得を行うことができます。IRowsetLocate の使用方法については、MSDN の『OLE DB Programmer's Reference』を参照してください。

- IRowsetLocate (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/oledb/htm/oledbirowsetlocate__irowset.asp)

jConnect でのディレクトリ・サービスの `sql.ini` とインタフェース・ファイルのサポート

この新機能は、`sql.ini` ファイル (Windows 用) と `interfaces` ファイル (UNIX 用) を使用した jConnect for JDBC に対するサーバ情報の提供をサポートします。

現時点では、ドライバまたはプロバイダに接続するには、サーバ名、IP アドレス、ASE サーバのポート番号などのさまざまなプロパティを指定する必要があります。`sql.ini` または `interfaces` ファイルを使用することで、エンタープライズ・ネットワークで使用可能なサービスに関する情報を ASE サーバ情報を含めてすべて集中管理できます。

接続文字列

`sql.ini` または `interfaces` ファイルを識別するには、次の情報を接続文字列に追加する必要があります。jConnect for JDBC では、単一のディレクトリ・サービスの URL (DSURL) にのみ接続できます。

注意 jConnect では、複数の URL はサポートしません。

jConnect に対する単一の DSURL 用の接続文字列

jConnect に対して単一の DSURL を提供するには、次のプロパティを次の書式で接続文字列に追加する必要があります。

注意 `sql.ini` ファイルのパスとサーバ名を指定する必要があります。

```
String url = "jdbc:sybase:jndi:file://D:/syb1252/ini/sql.ini?mango1"
```

各パラメータの意味は、次のとおりです。

- `server name = mango1`
- `sql.ini file path = file://D:/syb1252/ini/sql.ini`

URL 内に `sql.ini` のパスまたはサーバ名が指定されていない場合、ドライバはエラーを返します。

SSL 用の `interfaces` ファイルのフォーマット

SSL 用の `sql.ini` ファイルのフォーマットを次に示します。

```
[SYBSRV2]
master=nlwmsck,mango1,4100,ssl
query=nlwmsck,mango1,4100,ssl
```



```
query=nlwnsck,mango1,5000,ssl
```

注意 jConnect では、*sql.ini* ファイル内の同じサーバ名の下に指定した複数のクエリ・エントリをサポートします。jConnect は、*sql.ini* ファイルに指定された順序に従って、クエリ・エントリから **host** または **port** の値に接続を試みます。クエリ・エントリ内に **SSL** が見つかった場合、jConnect は、アプリケーション固有のソケット・ファクトリを指定することで **SSL** 接続を処理するようにコーディングされたアプリケーションを必要とします。このアプリケーションがない場合、接続は失敗します。

ESD#10 の新機能

ESD#10 には、次の新機能があります。

HP-UX 11.11 (または HP-UX 11i v1.0) 32 ビット版と 64 ビット版プラットフォームのサポート

HP-UX 11.11 (または HP-UX 11i v1.0) 32 ビット版と 64 ビット版プラットフォームが、現在、表 9 に示すように、Sybase SDK 12.5.1 および Open Server 12.5.1 製品でサポートされるようになりました。

表 9: SDK と Open Server 12.5.1 の新しいプラットフォームとサポートされる機能

プラットフォーム	オペレーティング・システム・レベル	C および C++ コンパイラ	COBOL コンパイラ	Kerberos バージョン	LDAP (Light-weight Directory Access)	SSL (Secure Sockets Layer)
HP-UX 11.11 (または HP-UX 11i v1.0) 32 ビット版	HP-UX 11i v1	HP C/ANSI C B.11.11.10 HP ANSI C++ B3910B A.03.10	Micro Focus Server Express 4.0	CyberSafe Trust Broker 2.1, MIT 1.4.1	Netscape LDAP 4.1	Certicom SSL Plus 3.1.14
HP-UX 11.11 (または HP-UX 11i v1.0) 64 ビット版	HP-UX 11i v1 パッチ・バンドル 990P	HP C 11.11.10 ANSI HP ANSI C++ B3910B A.03.10	使用不可	MIT 1.4.1	Netscape LDAP 4.1	Certicom SSL Plus 3.1.14

Linux AMD64 (Opteron)/EM64T での XA 64 ビット版のサポート

XA 64 ビット版が、Linux AMD64 (Opteron)/EM64T でサポートされるようになりました。

Microsoft Windows での MIT Kerberos のサポート

MIT Kerberos 5 version 1.3.6 が、Microsoft Windows (Windows NT、Windows 2000、Windows 2003、Windows XP を含む) でサポートされるようになりました。

スクロール可能カーソル

ASE ODBC ドライバおよび ASE OLE DB プロバイダが、スクロール可能なカーソルをサポートするようになりました。

スクロール可能カーソルの使用

アプリケーションでスクロール可能カーソルを使用すると、適切なスクロール・オプションを指定することで、結果セットの任意の場所にカーソルの現在位置を設定できます。NEXT、PRIOR、FIRST、LAST、RELATIVE、または ABSOLUTE のいずれかのスクロール・オプションを使用して、自由に結果セットを移動できます。

スクロール可能カーソルが要求されると、サーバ側またはクライアント側のスクロール可能カーソルを呼び出すことができます。サーバ側スクロール可能カーソルを呼び出すには、Adaptive Server がスクロール可能カーソルをサポートしている必要があります。Adaptive Server がスクロール可能カーソルをサポートしていない場合、キャッシュされた結果セットを使用してスクロール可能カーソルの機能を真似ることができます。これをクライアント側スクロール可能カーソルと呼びます。

ASE ODBC ドライバおよび ASE OLE DB プロバイダでは、ASE バージョン 15.0 以降でサーバ側とクライアント側のスクロール可能カーソルをサポートできます。15.0 より古いすべてのバージョンの ASE では、クライアント側のみのスクロール可能カーソルがサポートされません。

目的のスクロール可能カーソルを取得するには、UseCursor プロパティを正しく設定する必要があります。

警告！ クライアント側スクロール可能カーソルでは、クライアント側のメモリ使用量が多くなり、ネットワーク・トラフィックが増える場合があります。

UseCursor 接続プロパティの設定

UseCursor 接続プロパティの設定では、クライアント側またはサーバ側のどちらのスクロール可能カーソルを使用するかを指定します。

- UseCursor 接続プロパティを 1 に設定すると、ASE バージョン 15.0 以降では、サーバ側のスクロール可能カーソルが使用されます。

注意 15.0 より古いすべてのバージョンの ASE では、サーバ側スクロール可能カーソルは使用できません。

- UseCursor 接続プロパティを 0 に設定すると、ASE のバージョンに関係なく、クライアント側のスクロール可能カーソル (キャッシュされた結果セット) が使用されます。

ASE ODBC ドライバ向け

次に、ASE ODBC ドライバ向けのスクロール可能カーソルについて説明します。

Static Insensitive スクロール可能カーソルのサポート

ASE ODBC ドライバは、Static Insensitive スクロール可能カーソルをサポートしています。ODBC SQLFetchScroll メソッドの実装により、ローの scroll と fetch を行います。SQLFetchScroll メソッドは MSDN ライブラリの『Microsoft Open Database Connectivity Software Development Kit Programmer's Reference, Volume 2』に定義されている標準 ODBC メソッドです。詳細については、Microsoft Web site (<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/hlm/odbcsqlfetchscroll.asp>) を参照してください。

ODBC ドライバは、次のようなスクロール・タイプをサポートします。

- SQL_FETCH_NEXT - 次のローセットが返される。
- SQL_FETCH_PRIOR - 前のローセットが返される。
- SQL_FETCH_RELATIVE - 現在のローセットの開始から n 番目のローセットが返される。
- SQL_FETCH_FIRST - 結果セットの最初のローセットが返される。
- SQL_FETCH_LAST - 結果セットの最後の完全なローセットが返される。
- SQL_FETCH_ABSOLUTE - ロー n から始まるローセットが返される。

スクロール可能カーソルの属性設定

スクロール可能カーソルを使用するには、次の属性を設定する必要があります。

- SQL_ATTR_CURSOR_SCROLLABLE - 使用しているスクロール可能カーソルのタイプ。設定値は SQL_SCROLLABLE の値。有効な値は、static、semi-sensitive、または insensitive。

- **SQL_ATTR_CURSOR_SENSITIVITY** – このスクロール可能カーソルの sensitivity の値。

注意 サポートされる値は **SQL_INSENSITIVE** だけです。

スクロール可能カーソルを使用する際のオプション属性を次に示します。

- **SQL_ATTR_ROW_ARRAY_SIZE** – **SQLFetchScroll()** メソッドの 1 回の呼び出しにつき返されるローの数。

注意 この値を設定しない場合、デフォルト値である 1 つのローが使用されます。

- **SQL_ATTR_CURSOR_TYPE** – 使用しているスクロール可能カーソルのタイプ。

注意 サポートされる値は、**SQL_CURSOR_FORWARD_ONLY** または **SQL_CURSOR_STATIC** のみです。

- **SQL_ATTR_ROWS_FETCHED_PTR** – フェッチされたローの数が格納されるアドレス。**SQL_ATTR_ROWS_FETCHED_PTR** は、データ型 **SQLUIINTEGER** の変数を示す。
- **SQL_ATTR_ROW_STATUS_PTR** – ロー・ステータスが格納されるアドレス。**SQL_ATTR_ROW_STATUS_PTR** は、データ型 **SQLUSMALLINT** の変数を示す。

スクロール可能カーソルの実行

❖ スクロール可能カーソルを実行するプログラムを設定するには

- 1 使用している環境に応じてスクロール可能カーソルの属性を設定します。

詳細については、「[スクロール可能カーソルの属性設定](#)」(27 ページ)を参照してください。

- 2 結果をバインドします。たとえば、プログラムに次の行を追加します。

```
res=SQLBindCol(m_StatementHandle, 2, SQL_C_DOUBLE, price, 0, NULL);
res=SQLBindCol(m_StatementHandle, 3, SQL_C_LONG, quantity, 0, NULL);
```

- 3 **SQLFetchScroll()** を使用して、スクロールとフェッチを行います。たとえば、プログラムに次の行を追加します。

```
res = SQLSetStmtAttr(m_StatementHandle, SQL_ATTR_CURSOR_SCROLLABLE ,
(SQLPOINTER)SQL_SCROLLABLE, SQL_IS_INTEGER);

res = SQLSetStmtAttr(m_StatementHandle, SQL_ATTR_CURSOR_SENSITIVITY ,
(SQLPOINTER)SQL_INSENSITIVE, SQL_IS_INTEGER);

res = SQLFetchScroll(m_StatementHandle, SQL_FETCH_NEXT, 0);
```

```

res = SQLFetchScroll(m_StatementHandle, SQL_FETCH_PRIOR, 0);
res = SQLFetchScroll(m_StatementHandle, SQL_FETCH_FIRST, 0);
res = SQLFetchScroll(m_StatementHandle, SQL_FETCH_LAST, 0);
res = SQLFetchScroll(m_StatementHandle, SQL_FETCH_ABSOLUTE, 2);
res = SQLFetchScroll(m_StatementHandle, SQL_FETCH_ABSOLUTE, -2);
res = SQLFetchScroll(m_StatementHandle, SQL_FETCH_RELATIVE, 1);

```

- 4 **Select** 文を実行します。たとえば、プログラムに次の行を追加します。

```

res = SQLExecDirect(m_StatementHandle, (SQLCHAR "select price, quantity from
book" SQL_NTS));

```

- 5 結果セットとカーソルをクローズします。たとえば、プログラムに次の行を追加します。

```

res = SQLFreeStmt(m_StatementHandle, SQL_CLOSE);

```

結果の参照

スクロール可能カーソルを実行すると、ローの総数を N 件、ローセットを m 件として結果が表示されます。この場合、 $N > m$ です。

結果	解釈
Absolute 0	ローは返されませんでした。エラーです。
Absolute 1	m 件のローが返されました。
Absolute N	1 件のローが返されました。
Absolute $N+1$	ローは返されませんでした。エラーです。
First	最初の (1.. m) 件のローが返されました。
Last	最後の ($N-m+1$.. N) 件のローが返されました。
Next	SQLFetch() と同じです。
Prior	現在のローセットの前のローセットが返されました。

現在のカーソルがロー k にあり、 $k-a > 0$, $k+m+a < N$, $a \geq 0$ である場合、結果は次のようになります。

結果	解釈
Relative $-a$	($k-a$, $k-a+m-1$) のローが返されます。
Relative a	($k+a$, $k+a+m-1$) のローが返されます。

スクロール可能カーソルの暗黙的な属性設定

アプリケーションで特定の属性を設定したときに暗黙的に設定される属性があります。ODBC がサポートする、暗黙的に設定されるスクロール可能カーソルの属性は次のとおりです。

アプリケーションで設定する属性	暗黙的に設定される他の属性
SQL_ATTR_CONCURRENCY = SQL_CONCUR_READ_ONLY	SQL_ATTR_CURSOR_SENSITIVITY = SQL_INSENSITIVE
SQL_ATTR_CONCURRENCY = SQL_CONCUR_LOCK	SQL_ATTR_CURSOR_SENSITIVITY = SQL_SENSITIVE
SQL_ATTR_CURSOR_SCROLLABLE = SQL_NONSCROLLABLE	SQL_ATTR_CURSOR_TYPE = SQL_CURSOR_FORWARD_ONLY
SQL_ATTR_CURSOR_SENSITIVITY = SQL_INSENSITIVE	SQL_ATTR_CONCURRENCY = SQL_CONCUR_READ_ONLY SQL_ATTR_CURSOR_TYPE = SQL_CURSOR_STATIC
SQL_ATTR_CURSOR_TYPE = SQL_CURSOR_FORWARD_ONLY	SQL_ATTR_CURSOR_SCROLLABLE = SQL_NONSCROLLABLE
SQL_ATTR_CURSOR_TYPE = SQL_CURSOR_STATIC	SQL_ATTR_CURSOR_SCROLLABLE = SQL_SCROLLABLE

ASE OLE DB プロバイダ向け

次に、ASE OLE DB プロバイダ向けのスクロール可能カーソルについて説明します。

スクロール可能カーソルの使用

OLE DB データ・プロバイダは、次のようなスクロール・タイプをサポートします。

- Next – 次のローが返される。
- Prior – 前のローが返される。
- Relative *n* rows – 現在のローセットから *n* 番目のローセットが返される。

スクロール可能カーソルの属性設定

スクロール可能カーソルを使用するには、次の属性を設定する必要があります。

- DBPROP_CANSCROLLBACKWARDS – VARIANT_TRUE に設定すると、GetNextRows の IRowsOffset パラメータに負の値を指定できます。
- DBPROP_CANFETCHBACKWARDS – VARIANT_TRUE に設定すると、GetNextRows の cRows パラメータに負の値を指定できます。

スクロール可能カーソルの実行

❖ スクロール可能カーソルを実行するプログラムを設定するには

1 ローセットのスクロール可能カーソルのプロパティを設定します。

```
DBPROP RowsetProperties [2];
for(int i = 0; i < 2; i++)
    VariantInit(&RowsetProperties[i].vValue);

RowsetProperties[0].dwPropertyID = DBPROP_CANFETCHBACKWARDS;
RowsetProperties[0].vValue.vt = VT_BOOL;
RowsetProperties[0].vValue.boolVal= VARIANT_TRUE;
RowsetProperties[0].dwOptions = DBPROPOPTIONS_REQUIRED;
RowsetProperties[0].colid = DB_NULLID;
RowsetProperties[1].dwPropertyID = DBPROP_CANSROLLBACKWARDS;
RowsetProperties[1].vValue.vt = VT_BOOL;
RowsetProperties[1].vValue.boolVal= VARIANT_TRUE;
RowsetProperties[1].dwOptions = DBPROPOPTIONS_REQUIRED;
RowsetProperties[1].colid = DB_NULLID;

DBPROPSET rgRowsetPropSet [1];
rgRowsetPropSet[0].guidPropertySet = DBPROPSET_ROWSET;
rgRowsetPropSet[0].cProperties = 2;
rgRowsetPropSet[0].rgProperties = RowsetProperties;
```

2 ローセットをオープンします。

```
IRowset* pIRowset = ds.OpenRowset("book", 1, rgRowsetPropSet);
```

3 ローを前方向にフェッチします。

```
DBCOUNITITEM cRowsReturned;
HROW hRow[3];
HROW* pRows = hRow;
hr = pIRowset->GetNextRows(NULL, 0, 3, &cRowsReturned, &pRows);
```

4 ローを解放します。

```
hr = pIRowset->ReleaseRows(cRowsReturned, pRows, NULL, NULL, NULL);
```

5 ローを後方向にフェッチします。

```
DBCOUNITITEM cRowsReturned;
HROW hRow[3];
HROW* pRows = hRow;
hr = pIRowset->GetNextRows(NULL, 0, -3, &cRowsReturned, &pRows);
```

6 ローを解放します。

```
hr = pIRowset->ReleaseRows(cRowsReturned, pRows, NULL, NULL, NULL);
```

7 ローセットを解放します。

```
pIRowset->Release();
```

結果の参照	結果および結果セットの解釈を確認するには、スクロール可能カーソルを実行後、Microsoft MSDN library (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/oledb/htm/oledbirowset_getnextrows.asp) を参照してください。
静的な非反映型のスクロール可能カーソル・プログラムの例	静的な非反映型のスクロール可能カーソル・プログラムの例については、前の項を参照してください。

ESD#9 の新機能

ESD#9 には、次の新機能があります。

分散トランザクションの使用

ここでは、ASE ODBC ドライバと ASE ADO.NET Data Provider を使用し、それを 2 フェーズ・コミット・トランザクションに含める方法について説明します。

分散トランザクションでの ASE ODBC ドライバの使用

この機能は Microsoft Windows でのみサポートされ、2 フェーズ・コミットを管理するトランザクション・コーディネータとして Microsoft 分散トランザクション・コーディネータ (MS DTC) が使用されている必要があります。

Sybase は次のすべてのプログラミング・モデルをサポートしています。

- MS DTC を直接使用するアプリケーション
- Sybase Enterprise Application Server (EAServer, 別名 Jaguar) を使用するアプリケーション
- Microsoft Transaction Server (MTS) または (COM+) を使用するアプリケーション

MS DTC のプログラミング

- ❖ **Microsoft 分散トランザクション・コーディネータ (MS DTC) を使用してプログラミングするには**
 - 1 DtcGetTransactionManager 関数を使用して MS DTC に接続します。MS DTC の詳細については、関連する Microsoft 分散トランザクション・コーディネータのマニュアルを参照してください。
 - 2 確立する Sybase ASE 接続ごとに 1 回、SQLDriverConnect または SQLConnect を呼び出します。

- 3 ITransactionDispenser::BeginTransaction 関数を呼び出して MS DTC トランザクションを開始し、そのトランザクションを表す OLE トランザクション・オブジェクトを取得します。
- 4 MS DTC トランザクションに登録する ODBC 接続ごとに 1 回または複数回、SQLSetConnectAttr を呼び出します。SQLSetConnectAttr の呼び出し時には、手順 3 で取得したトランザクション・オブジェクトの SQL_ATTR_ENLIST_IN_DTC および ValuePtr 属性を指定する必要があります。
- 5 insert または update SQL 文ごとに 1 回または複数回、SQLExecDirect を呼び出します。
- 6 ITransaction::Commit 関数を呼び出して MS DTC トランザクションをコミットします。これでトランザクション・オブジェクトは無効になります。

一連の MS DTC トランザクションを実行するには、手順 3～6 を繰り返します。トランザクション・オブジェクトへの参照を解放するには、ITransaction::Release 関数を呼び出します。

MS DTC トランザクションに使用している ODBC 接続をローカルの ASE Server トランザクションでも使用するには、SQL_DTC_DONE の ValuePtr を指定して SQLSetConnectAttr を呼び出し、トランザクションから接続の登録を解除します。

注意 手順 4 と 5 で示した呼び出し方法の代わりに、ASE Server ごとに別々に SQLSetConnectAttr と SQLExecDirect を呼び出すこともできます。

Sybase EAServer、MTS、または COM+ に展開されるコンポーネントのプログラミング

次の手順は、Sybase EAServer、MTS、または COM+ で分散トランザクションに関与するコンポーネントの作成方法を説明しています。

❖ Sybase EAServer、MTS、または COM+ に展開されるコンポーネントのプログラムを作成するには

- 1 確立する Sybase ASE 接続ごとに 1 回、SQLDriverConnect を呼び出します。
- 2 insert または update SQL 文ごとに 1 回または複数回、SQLExecDirect を呼び出します。
- 3 コンポーネントを MTS に展開し、必要に応じてトランザクション属性を設定します。

トランザクション・コーディネータは必要に応じて分散トランザクションを作成し、ASE ODBC ドライバを使用するコンポーネントがグローバル・トランザクションに自動登録されます。次に、分散トランザクションがコミットまたはロールバックされます。

分散トランザクションでの接続プロパティのサポート

次に、接続プロパティについて説明します。

- 分散トランザクション・プロトコル (DistributedTransactionProtocol) – 分散トランザクションをサポートするために使用されるプロトコルを指定するには、XA インタフェース標準または MS DTC OLE ネイティブ・プロトコルのいずれかを使用し、[ODBC データ・ソース] ダイアログで [分散トランザクション・プロトコル] を選択するか、接続文字列にプロパティ DistributedTransactionProtocol = OLE ネイティブ・プロトコルを渡します。デフォルトは XA です。
- 密結合トランザクション (TightlyCoupledTransaction) – 2つのリソース・マネージャを使用する分散トランザクションで同一の ASE サーバが指定されている場合、「密結合トランザクション」と呼ばれる状況が発生することがあります。この状況が発生した場合、このプロパティを 1 に設定していないと分散トランザクションは失敗します。

つまり、同一の ASE サーバに対して 2つのデータベース接続をオープンしてから、オープンした接続を同一の分散トランザクションに登録する場合は、TightlyCoupledTransaction=1 を設定する必要があります。

このプロパティを設定するには、[ODBC データ・ソース] ダイアログで [密結合トランザクション] を選択するか、接続文字列でプロパティ TightlyCoupledTransaction=1 を渡します。

警告！ SQLSetConnectAttr に SQL_AUTOCOMMIT_OFF を指定して実行するか、SQLExecDirect を使用して BEGIN TRANSACTION 文を明示的に実行することにより、その接続ですでにローカル・トランザクションを開始している場合、SQLSetConnectAttr を登録すると、SQL_ERROR が返されます。

分散トランザクションでの ASE ADO.NET Data Provider の使用

この機能では、分散トランザクションを管理する .NET Enterprise Services を使用する必要があります。

Enterprise Services を使用するプログラミング

COM+ サービス・インフラストラクチャは、マネージ・コードおよびアンマネージ・コードからアクセスできます。アンマネージ・コード内のサービスは、COM+ サービスと呼ばれています。.NET では、このサービスは Enterprise Services と呼ばれています。ADO.NET とともに Enterprise Services 内でトランザクションを使用するのは簡単です。

❖ Enterprise Services を使用してプログラムを作成するには

- 1 `System.EnterpriseService.ServicedComponent` からコンポーネントを抽出します。
- 2 必要なサービスとそのオプションを指定するための独自の属性 (`Transaction`、`AutoComplete` など) を指定します。属性の全リストについては、Enterprise Services のマニュアルを参照してください。

注意 .NET トランザクション属性の `Timeout Option` には、明示的に `-1` または非常に大きな数を設定する必要があります。.NET のマニュアルでは、ADO.NET トランザクションのデフォルトのタイムアウト値が `0` であることが記載されていますが、これはタイムアウトしないことを意味します。ただし、実際には即時にトランザクションがタイムアウトし、トランザクション全体がロールバックされます。

- 3 アセンブリを署名および構築します。
- 4 アセンブリを登録します。

分散トランザクションでの接続プロパティのサポート

次に、分散トランザクションのサポート時に使用する接続プロパティを示します。

- 分散トランザクション・プロトコル (`DistributedTransactionProtocol`) – 分散トランザクション、XA インタフェース標準、または MS DTC OLE ネイティブ・プロトコルをサポートするためのプロトコルを指定するには、接続文字列でプロパティ `DistributedTransactionProtocol=OLE` ネイティブ・プロトコルを設定します。デフォルトのプロトコルは `XA` です。
- 密結合トランザクション (`TightlyCoupledTransaction`) – 2つのリソース・マネージャを使用する分散トランザクションで同一の ASE サーバが指定されている場合、「密結合トランザクション」と呼ばれる状況が発生することがあります。この状況が発生した場合、このプロパティを `1` に設定していないと分散トランザクションは失敗します。

つまり、同一の ASE サーバに対して2つのデータベース接続をオープンしてから、オープンした接続を同一の分散トランザクションに登録する場合は、`TightlyCoupledTransaction=1` を設定する必要があります。

- 登録 – `AseConnection` オブジェクトは、トランザクションがアクティブであることを特定した場合に、既存の分散トランザクションに自動的に登録します。接続を開くか、接続プールから取得したときに、自動トランザクション登録が行われます。`AseConnection` の接続文字列パラメータとして `Enlist=0` を指定することにより、この自動登録機能を無効にできます。

自動登録が無効である場合、既存トランザクションへの参照である `ITransaction` パラメータとともに `AseConnection` の `EnlistDistributedTransaction` メソッドを呼び出すことにより、既存の分散トランザクションに登録できます。`EnlistDistributedTransaction` を呼び出した後、`AseConnection` のこのインスタンスを使用したすべての更新は、このグローバル・トランザクションの一部として適用されます。したがって、グローバル・トランザクションがコミットまたはロール・バックされると、それに伴ってこのトランザクションもコミットまたはロール・バックされます。

注意 `AseConnection` オブジェクトは、`EnlistDistributedTransaction` を呼び出す前に開いている必要があります。

ビジネス・オブジェクトをプールする場合、`EnlistDistributedTransaction` を使用できます。ビジネスオブジェクトが開いている接続とともにプールされている場合、その接続が開いているか、または接続プールから取得される場合に限り、自動トランザクション登録が行われます。プールされたビジネス・オブジェクトにより複数のトランザクションが実行される場合、そのオブジェクトの開いている接続は、新しく開始したトランザクションで自動的に登録されません。この場合は、`AseConnection` の自動トランザクション登録を無効にした後、`EnlistDistributedTransaction` を使用して `AseConnection` をトランザクションに登録できます。

警告! `AseConnection` が `BeginTransaction` を使用するか、または `AseCommand` とともに `BEGIN TRANSACTION` 文を明示的に実行することによりすでにトランザクションを開始している場合、`EnlistDistributedTransaction` は例外を返します。

ESD#8 の新機能

ESD#8 には、次の新機能があります。

MIT Kerberos の拡張サポート

MIT Kerberos 5 バージョン 1.3.6 が次のプラットフォームでサポートされるようになりました。

- IBM AIX 64 ビット版
- HP-UX 11.11 (または HP-UX 11i v1.0) 64 ビット版

Sybase がサポートする MIT Kerberos のバージョンとプラットフォームを次に示します。

- バージョン 1.3.1 – Linux Intel 32 ビット版と Sun Solaris 8 (SPARC 32 ビット版と 64 ビット版)
- バージョン 1.2.7 – Linux AMD64 (Opteron)/EM64T

新しいプラットフォームとオペレーティング・システムのサポート

表 10 に、Sybase SDK 12.5.1 製品と Open Server 12.5.1 製品をサポートする新しいプラットフォーム、オペレーティング・システム、機能を示します。

表 10: SDK と Open Server 12.5.1 の新しいプラットフォームとサポートされる機能

プラットフォーム	オペレーティング・システム・レベル	C および C++ コンパイラ	COBOL コンパイラ	Kerberos バージョン	LDAP (Light-weight Directory Access)	SSL (Secure Sockets Layer)
Linux on POWER 32 ビット版	Red Hat Enterprise Linux AS 3.0	IBM XL C/C++ Advance Edition V7.0	使用不可	使用不可	使用不可	使用不可
Linux on POWER 64 ビット版	Red Hat Enterprise Linux AS 3.0	IBM XL C/C++ Advance Edition V7.0	使用不可	使用不可	使用不可	使用不可
Linux AMD64 (Opteron) /EM64T	Red Hat Enterprise Linux AS 3.0	GCC 3.2.3 (Red Hat Linux 3.2.3 - 42)	使用不可	MIT 1.2.7	使用不可	使用不可
Sun x86 32 ビット版	Solaris 9	Sun C/C++ 6.2	使用不可	使用不可	使用不可	使用不可

IBM AIX 64 ビット版の LDAP の機能

IBM AIX 64 ビットで、LDAP (OpenLDAP) が使用できるようになりました。LDAP ディレクトリ・サービスを使用してネットワーク・エンティティから情報を作成、変更、検索できます。

OpenLDAP を有効にするには、`$$SYBASE/$SYBASE_OCS/config` ディレクトリにある `libtcl64.cfg` 設定ファイルを変更し、設定済みのディレクトリ・サービスに LDAP サーバを追加します。その方法を次に説明します。

❖ OpenLDAP ディレクトリ・サービスを有効にするには

- 1 使用するプラットフォームの環境変数を LDAP ライブラリに追加します。例：

```
setenv LIBPATH | ¥
${LIBPATH} : $$SYBASE/$SYBASE_OCS/lib3p64
```

- 2 `libtcl64.cfg` の [DIRECTORY] セクションで LDAP サーバを選択し、新しいキーワードと値のペアを追加します。

注意 LDAP URL は、1 行で記述してください。

```
ldap=libdldap64.so ldap://host.port/ditbase??scope??  
bindname=username.password
```

例：

```
[DIRECTORY]  
ldap=libdldap64.so ldap://huey:11389/dc=sybase,  
dc=com??one??bindname=cn=Manager, dc=sybase,  
dc=com?secret
```

- ❖ 設定した OpenLDAP ディレクトリ・サービスに LDAP サーバを追加するには
 - 1 `$$SYBASE/$$SYBASE_OCS/bin` ディレクトリの `dsedit` または `dscpc` ユーティリティを起動します。
 - 2 `server name` などのサーバ詳細と、`transport type`、`host name`、`port number` などのネットワーク・トランスポート詳細を入力します。

Windows での CyberSafe Kerberos クライアントのアップグレード

現在、CyberSafe ActiveTRUST Secure Client (Kerberos) 4.0 は、Microsoft Windows (Windows NT、Windows 2000、Windows 2003、Windows XP) でサポートされるようになりました。

CyberSafe Kerberos は、次の製品でサポートされるようになりました。

- Software Developer's Kit のコンポーネント
 - Open Client (Client-Library)
 - ESQ/C
 - ESQ/COBOL
- Open Server

SSLPlus 3.1.x のセキュリティ修正

次のバージョンの SSL (Secure Sockets Layer) では、SSLPlus のバグとメモリ・リークが修正されています。

- Sun Solaris 8 (SPARC 32 ビット版と 64 ビット版)、HP-UX 11.11 (32 ビット版と 64 ビット版)、HP Tru64、IBM AIX (32 ビット版と 64 ビット版)、Linux Intel 32 ビット版の SSL Plus 3.1.14c。このバージョンは、ESD#7 以降で使用できます。

- Microsoft Windows (Windows NT 32 ビット版、Windows 2000、Windows 2003、Windows XP) の SSL Plus 3.1.15b。このバージョンは、ESD#8 以降で使用できます。

上記のバージョンの SSLPlus 3.1.x は、Sybase SSL フィルタ *libfssl.** とリンクされます。

Windows 2000 での COBOL コンパイラのアップグレード

Microsoft Windows 2000 での COBOL コンパイラは、Micro Focus Net Express 4.0 にアップグレードされました。

ESD#7 の新機能

ESD#7 には、次の新機能があります。

XA 64 ビット版のサポート・プラットフォームの追加

現状の Solaris 64 ビットでの使用に加えて、次のプラットフォームでも Extended Architecture (XA) 64 ビット版を使用できるようになりました。

- IBM AIX 64 ビット版
- HP-UX 11.11 (または HP-UX 11i v1.0) 64 ビット版
- HP Itanium 64 ビット版

現在、次の 32 ビット・プラットフォームで XA 32 ビット版がサポートされるようになりました。

- IBM AIX 32 ビット版
- HP-UX 11.11 (または HP-UX 11i v1.0) 32 ビット版
- Sun Solaris 8 (SPARC 32 ビット版)
- Microsoft Windows (Windows NT、Windows 2000、Windows 2003、Windows XP)

XA インタフェースの概要

分散トランザクション・マネージャ (DTM) XA インタフェースは、XA インタフェース標準を Sybase が Adaptive Server に実装したものです。DTM XA インタフェースは、X/Open 分散トランザクション処理 (DTP) モデルの 1 つの要素であり、DTP アプリケーション開発における業界標準です。XA インタフェースは、Adaptive Server に格納されているデータに、CICS、Encina、または TUXEDO のトランザクション・マネージャ (TM) からアクセスします。

XA インタフェースのコンポーネント

Sybase XA インタフェースは次のコンポーネントで構成されています。

- Sybase DTM XA インタフェース。XA インタフェースを Sybase が Adaptive Server に実装したものです。これは、Adaptive Server の機能ですが、ライセンスが別途必要です。
- Adaptive Server と DTM 機能。ソフトウェアのインストールと機能ライセンスについては、使用しているプラットフォームに応じた Adaptive Server Enterprise の『インストール・ガイド』を参照してください。
- Sybase Open Client。Client-Library 呼び出しを、アプリケーションとリソース・マネージャ間のネイティブ・インタフェースの一部にすることができます。
- XA 設定ファイル。XA で使用するクライアント/サーバ接続の定義エントリが含まれています。
- Embedded SQL/C と Embedded SQL/COBOL。ESQL 呼び出しを、アプリケーションとリソース・マネージャ間のネイティブ・インタフェースの一部として作成します。
- XA 固有の dbcc コマンド・セット。システムの管理者はこれを使用してヒューリスティック・トランザクションを管理できます。
- グローバル・リカバリに使用できる TM 固有の設定ファイルとコマンド。

ネイティブの Adaptive Server DTM 機能の使用方法については、『XA インタフェース統合ガイド for CICS、Encina、TUXEDO』を参照してください。

ESD#6 の新機能

ESD#6 には、次の新機能があります。

Sybase 製の新しい ASE OLE DB プロバイダ

このリリースの SDK 12.5.1 では、「Sybase 製 Adaptive Server Enterprise OLE DB Provider」という Sybase ASE に対応した新しい OLE DB プロバイダが導入されています。OLE DB は、Microsoft が開発したデータ・アクセス・モデルです。OLE DB は、COM (Component Object Model) インタフェースを使用し、データ・ソースでの SQL クエリ・プロセッサの使用を前提としない点で、ODBC とは異なります。

サポートされるプラットフォーム

ASE OLE DB プロバイダは、OLE DB 2.5 以降で使用できます。サポートされるプラットフォームには、Windows NT 4.0、Windows 2000、Windows XP、Windows 2003 が含まれます。

サポートされる ASE 機能	<p>Adaptive Server Enterprise OLE DB プロバイダは、次の ASE 固有の機能をサポートします。</p> <ul style="list-style-type: none">• SSL• ディレクトリ・サービス (LDAP)• パスワードの暗号化
移行の準備	<p>将来のリリースで、既存の OLE DB プロバイダは、Sybase が開発したこの ASE OLE DB プロバイダに置き換えられます。現在では、両方のプロバイダが提供されており、同一マシン上で共存できます。新しいプロバイダへのアプリケーションの移行を開始し、テストを実行して互換性の問題を確認することをおすすめします。既存の OLE DB プロバイダ (プロバイダの短い名前は "Sybase.ASEOLEDBProvider") は、<code>%SYBASE%\OLEDB</code> ディレクトリにインストールされており、現在のバージョンは 2.70 です。新しい ASE OLE DB プロバイダ (プロバイダの短い名前は "ASEOLEDB") は、<code>%SYBASE%\DataAccess\OLEDB</code> ディレクトリにインストールされており、バージョンは 12.5.1.384 以降です。</p> <p>詳細については、Sybase 製 Adaptive Server Enterprise OLE DB プロバイダ用の『ユーザーズ・ガイド』を参照してください。</p>

ESD#5 の新機能

ESD#5 には、次の新機能があります。

BCP 暗号化カラム

現在、暗号化カラムがサポートされたテーブルに対して BCP を実行するとプレーン・テキスト形式でデータを取得し、ユーザにはカラムのデータを表示するパーミッションが付与されています。

新しい BCP コマンド・ライン・オプション (-C) を使用すると、権限を与えられたユーザは、暗号テキスト・データをバルク移動できます。この操作を行うには、-C オプションを使用することと、サーバにあるテーブルで暗号化カラムがサポートされていることが必要です。このオプションを使用すると、SQL コマンド `set ciphertext=on` がサーバに送信されてから、暗号テキストを生成するバルク・ライブラリ・ルーチンが開始されます。

注意 暗号化カラムがサポートされていないサーバで新しいオプション -C を指定してもエラーは発生しません。BCP では、テーブル `sysencryptkeys` の存在を確認した後、SQL コマンドが送信されます。このテーブルが存在するときに SQL コマンドが失敗した場合は、致命的なエラーとみなされます。

韓国語の文字に対するサポートの強化

Open Client と Open Server で、韓国語の文字セット cp949. がサポートされるようになりました。charsets ディレクトリに新しいディレクトリが追加され、utf8 charset ディレクトリ、utf8.cfg ファイル、locales.dat に文字セットのテーブル・ファイルが追加され、objectid.dat が変更されて文字セットが追加されました。この韓国語の文字セットは、他の文字セットと同じ方法で使用でき、cs_convert() による文字データの変換もできます。

SSLPlus 3.1.10H のセキュリティ修正

Secure Sockets Layer (SSL) SSLPlus 3.1.10H は、バージョン 3.1x の最新のリリースであり、ESD#5 以降で使用できます。このリリースでは、SSLPlus のバグとメモリ・リークが修正されています。Sun Solaris 8 (SPARC 32 ビット版と 64 ビット版)、Linux Intel 32 ビット版、IBM AIX 64 ビット版、Microsoft Windows NT で、SSLPlus 3.1.10H は Sybase SSL フィルタ libfssl.* とリンクされています。

ODBC と OLE DB での MIT Kerberos のサポート

ODBC 5.00.0034 と OLE DB 2.70.0041 から、Sybase ODBC ドライバと OLE DB プロバイダが、CyberSafe Kerberos のサポートに加えて、MIT Kerberos をサポートするようになりました。この機能の使用の詳細については、各製品に伴ってリリースされたヘルプ・ファイルを参照してください。

❖ Sybase ASE ODBC ドライバと Sybase ASE OLE DB プロバイダの Kerberos 設定情報を見つけるには

- 1 ODBC データ・ソース・アドミニストレータ・ツール (ODBCAD32.EXE) を起動します。
- 2 使用するデータ・ソースを選択し、[構成] をクリックします (OLE DB プロバイダだけを使用する場合は、[追加] をクリックして Sybase ASE ODBC ドライバを選択します)。
- 3 [Connection] パネルに移動し、[Help] をクリックします。
- 4 「Use Kerberos」節を参照してください。CyberSafe Kerberos または MIT Kerberos の資料が見つかります。

Windows での SDK 12.5.1 のインストール

SDK 12.5.1 にアップグレードし、以前の 12.5 ODBC ドライバのインストール・ディレクトリである SDK 12.5 Sybase ディレクトリ (*%SYBASE%\odbc*) にインストールした場合は、インストーラによって ODBC ドライバが更新され、すぐにアクセスできます。

注意 元の ODBC ドライバ・ディレクトリのロケーションを使用することをおすすめします。

以前の SDK 12.5 ODBC ドライバがインストールされているマシン上で (以前の ODBC ドライバ用の Sybase ディレクトリにインストールされている場合)、新しいディレクトリを選択して 12.5.1 をインストールすると、新しい ODBC ドライバにアクセスできません。インストール時に使用したデータ・ソースによって、*SYODASE.DLL* がどこに存在するかを示す絶対パスのハード・コピーが格納されます。Windows の ODBC ドライバの DSN レジストリを更新して、*SYODASE.DLL* の参照を最新の情報に更新するユーティリティはありません。ODBC ドライバのロケーションを変更する場合は、*SYODASE.DLL*、およびそれに関連した *DLL* とコンポーネントを以前の Sybase ディレクトリにコピーするか、すべての DSN を変更して新しいドライバのロケーションを使用してください。

Sybase 製 ASE ODBC ドライバのサポート・プラットフォームの拡張

ODBC インタフェースは、Microsoft がデータベース管理システムの標準インタフェースとして定義している呼び出しベースのアプリケーション・プログラミング・インタフェースです。Adaptive Server Enterprise ODBC 12.5.1 ドライバによって、ODBC 呼び出しインタフェースを使用して Adaptive Server 内のデータにアクセスするプログラムを作成できます。ODBC Level 2 機能のほかに、このドライバは次の機能をサポートしています。

- SSL
- HA フェールオーバー
- パスワードの暗号化

このリリースの SDK では、Sybase 製 ASE ODBC Driver をサポートするプラットフォームが追加されています。このドライバは、SDK 12.5.1 ESD#2 で Linux プラットフォームに導入され、このリリースではさらに、Windows と Mac OS X の各オペレーティング・システムのサポートが追加されました。

注意 将来のリリースで、既存の ODBC ドライバ・キットは、Sybase が開発したこのドライバに置き換えられます。現在では、両方のドライバが並行して出荷されており、同一マシン上で共存できます。新しいドライバへのアプリケーションの移行を開始し、テストを実行して互換性の問題を確認することをおすすめします。既存の ODBC ドライバ・キットは、ODBC ドライバ・マネージャに“Sybase ASE ODBC Driver”として登録され、現在のバージョンは 5.00.0034 です。Sybase 製 ASE ODBC ドライバは、“Adaptive Server Enterprise”として登録され、現在のバージョンは 12.5.1.376 です。

ESD#4 の新機能

ESD#4 には、次の新機能があります。

診断の機能強化

Client-Library の `ct_debug` ルーチンが変更され、Ribo (TDS トレース・ツール) のダウンロード、インストール、取得または使用のための設定をしないで、Tabular Data Stream™ (TDS) をトレースしてファイルにダンプできるようになりました。`ct_debug` ルーチンのパラメータへの変更点は次のとおりです。

- `CS_DBG_PROTOCOL` の有効化または `CS_DBG_PROTOCOL_FILE` の設定のために、アプリケーションで `devlib` (デバッグ) ライブラリを使用する必要がなくなりました。
- 接続に `CS_DBG_PROTOCOL_FILE` が設定されていないと、`mktemp` が呼び出されて、プロトコル・パケットをダンプするためのユニークな名前を持つファイルが生成されます。`mktemp` に渡されるプレフィクス文字列は“`captur`”であり、Ribo を使用して結果のプロトコル・ファイルをデコードできます。
- `ocs.cfg` ファイルを使用して、プロトコル・ダンプ機能を有効にすることもできます。その場合は、通常のリブラリと `devlib` ライブラリの両方にプロトコル・ダンプ・ファイルが作成されます。

非同期通信のサポート

非同期通信のサポートは、Sybase ASE ODBC ドライバ 5.00.0022 および Sybase ASE OLE DB プロバイダ 2.70.0038 の新機能です。現在、ODBC 3.5 仕様で指定された非同期の通信をサポートしています。

接続レベルの非同期サポート

TDS ODBC ドライバの **Cursor Mode (Select Method)** が **Cursor** に設定されている場合は、**SQLGetInfo** を通してドライバが接続レベルの非同期をサポートしていることを示します。情報タイプを **SQL_ASYNC_MODE** に設定して **SQLGetInfo** を呼び出すと、**SQL_AM_CONNECTION** が返されます。

文レベルの非同期サポート

TDS ODBC ドライバの **Cursor Mode** が **Direct** に設定されている場合は、**SQLGetInfo** を通してドライバが文レベルの非同期をサポートしていることを示します。情報タイプを **SQL_ASYNC_MODE** に設定して **SQLGetInfo** を呼び出すと、**SQL_AM_STATEMENT** が返されます。

非同期の同時文の最大数

TDS ODBC ドライバは **SQLGetInfo** を通して、非同期の同時文の数についての明示的な制限値が 1 であることを示します。情報タイプを **SQL_MAX_ASYNC_CONCURRENT_STATEMENTS** に設定して **SQLGetInfo** を呼び出すと、1 が返されます。

非同期通信の有効化

TDS ODBC ドライバの **cursor mode** が **Cursor** に設定されている場合は、**SQLSetConnectAttr** を呼び出して非同期通信サポートを有効にします。属性値 **SQL_ATTR_ASYNC_ENABLE** を **SQL_ASYNC_ENABLE_ON** に設定して **SQLSetConnectAttr** を呼び出すと、非同期サポートがオンになります。属性値 **SQL_ATTR_ASYNC_ENABLE** を **SQL_ASYNC_ENABLE_OFF** に設定して **SQLSetConnectAttr** を呼び出すと、非同期サポートがオフになります。

TDS ODBC ドライバの **cursor mode** が **Direct** に設定されている場合は、**SQLSetStmtAttr** を呼び出して非同期通信サポートを有効にします。属性値 **SQL_ATTR_ASYNC_ENABLE** を **SQL_ASYNC_ENABLE_ON** に設定して **SQLSetStmtAttr** を呼び出すと、非同期サポートがオンになります。属性値 **SQL_ATTR_ASYNC_ENABLE** を **SQL_ASYNC_ENABLE_OFF** に設定して **SQLSetStmtAttr** を呼び出すと、非同期サポートがオフになります。

非同期 ODBC 関数のサポート

次の ODBC 関数を非同期でサポートできます。

- SQLBulkOperations
- SQLColAttribute
- SQLColumnPrivileges
- SQLColumns
- SQLCopyDesc
- SQLDescribeCol
- SQLDescribeParam
- SQLExecDirect
- SQLExecute
- SQLFetch
- SQLFetchScroll

SQLForeignKeys
SQLGetData
SQLGetDescField
SQLGetDescRec
SQLGetDiagField
SQLGetDiagRec
SQLGetTypeInfo
SQLMoreResults
SQLNumParams
SQLNumResultCols
SQLParamData
SQLPrepare
SQLPrimaryKeys
SQLProcedureColumns
SQLProcedures
SQLPutData
SQLSetPos
SQLSpecialColumns
SQLStatistics
SQLTablePrivileges
SQLTables

上記の関数が非同期で実行中のときは、リターンコード `SQL_STILL_EXECUTING` が返されます。指定された関数が終了すると、同期実行の場合に返すコード、たとえば、`SQL_SUCCESS`、`SQL_ERROR`、`SQL_NEED_DATA` などが返されます。

処理の変更

非同期通信が有効になると、ODBC 3.5 仕様の詳細に従って、上記の関数が非同期で実行されます。非同期通信が無効になると、すべての関数が同期実行されます。他のすべての処理は、通常どおりに実行されます。

DB-Library の新しい設定変数

この新機能は、`dbsetversion` を呼び出すことによって、環境変数を使用して実行時に外部で DB-Library バージョン・レベルを設定してアプリケーション・コードを変更します。次のような機能があります。

- DB-Library の初期化段階において環境変数を取得する。
- 環境変数の値をバージョン・レベルとして格納する。

現在ソース・コードでは、DB-Library によって、デフォルトのバージョン・レベルが DBVERSION_46 に設定されています。このデフォルトのバージョン・レベルは、アプリケーションで `dbsetversion` を呼び出すことによるのみ変更できます。TDS プロトコル・バージョン 5.0 の機能（たとえば、`numeric` 型や `decimal` 型）を使用するには、ハード・コードされたデフォルトのバージョン・レベル (DBVERSION_46) を変更する必要があります。これまでは、呼び出し `dbsetversion(DBVERSION_100)` をコードに追加していました。この新機能を使用すると、アプリケーションから `dbsetversion` を呼び出すことなく DB-Library のバージョン・レベルを判定できます。

定義

次の定義を使用して、DB-Library の新しい設定変数を記述します。

- **DB-Library** – C プログラミング API ルーチンやマクロの集合であり、これらをアプリケーションで使用すると、データ・サーバや Open Server のアプリケーションと対話できるようになります。
- **CT-Library** – クライアント・アプリケーションを作成するときに使用する C プログラミング API ルーチンの集合です。
- **バージョン・レベル** – TDS プロトコルのバージョンを指定するときに使用する DB-Library のパラメータです。3 つの有効な値は次のとおりです。
 - **DBVERSION_46** – TDS バージョン 4.6 プロトコルを使用して DB-Library を実行します。
 - **DBVERSION_100** – TDS バージョン 5.0 プロトコルを使用して DB-Library を実行します。
 - **DBVERSION_UNKNOWN** – バージョン・レベルがまだ決定されていません。
- **dbsetversion** – クライアント・アプリケーションの DB-Library のバージョン・レベルを設定する DB-Library の API です。
- **dbinit** – DB-Library を初期化する API です。この API を最初に呼び出し、その後で他のすべての DB-Library ルーチンを呼び出します。

設定

この機能を実装するには、新しい環境変数 `SYBOCS_DBVERSION` を定義して、DB-Library のバージョン・レベルを外部で設定できるようにします。

この環境変数の有効な値は、次のとおりです。

- `DBVERSION_46`
- `DBVERSION_100`

上記以外の値を使用すると、DB-Library アプリケーションが失敗する原因になります。

処理の順序

以降の段落では、DB-Library アプリケーションでの環境変数と `dbsetversion` の処理順序について説明します。

- 環境変数が未定義であり、アプリケーションで `dbsetversion` を呼び出さなかった場合、バージョン・レベルは現在の DB-Library ソース・コードのデフォルト・バージョンである `DBVERSION_46` になります。

- 環境変数が未定義であり、DBVERSION_100 または DBVERSION_46 の値を使用して `dbsetversion` を呼び出した場合、バージョン・レベルはそれぞれ DBVERSION_100 または DBVERSION_46 です。
- 環境変数が DBVERSION_100 または DBVERSION_46 として定義され、アプリケーションで `dbsetversion` を呼び出さなかった場合、バージョン・レベルはそれぞれ DBVERSION_100 または DBVERSION_46 です。
- 環境変数を DBVERSION_100 として定義し、アプリケーションで `dbsetversion` を呼び出してバージョン・レベルを DBVERSION_46 として設定した場合は、`dbsetversion` によって、環境変数が上書きされ、最終バージョン・レベルが決定されるため、バージョン・レベルは DBVERSION_46 です。

Adaptive Server Enterprise ODBC ドライバ Linux Intel 32 ビット版

このリリースの SDK には、Open Database Connectivity (ODBC) ドライバが含まれています。ODBC インタフェースは、Microsoft がデータベース管理システムの標準インタフェースとして定義している呼び出しベースのアプリケーション・プログラミング・インタフェースです。Adaptive Server Enterprise ODBC 12.5.1 ドライバによって、ODBC 呼び出しインタフェースを使用して Adaptive Server 内のデータにアクセスするプログラムを作成できます。ODBC Level 2 機能のほかに、このドライバは次の機能をサポートしています。

- SSL
- HA フェールオーバー
- パスワードの暗号化

新しい ASE ODBC ドライバ

注意 ASE ODBC 5.0 ドライバのリリースは、ODBC 3.52 仕様にに基づいています。ASE ODBC 4.x ドライバで使用するために作成されたすべてのアプリケーションは、そのまま 5.0 リリースで使用できます。

ASE ODBC ドライバ 5.0 の新機能を簡単にまとめると、次のとおりです。

- `date` および `time` データ型のサポート
- パスワードの暗号化のサポート
- `BAS¥UTL¥FLT` 共有コンポーネントの削除

ESD#3の新機能

ESD#3には、次の新機能があります。

Adaptive Server Enterprise ADO.NET Data Provider バージョン 1.1 Microsoft Windows 版

このリリースには、Adaptive Server データベース用の ADO.NET プロバイダである ASE ADO.NET Data Provider 1.1 が含まれています。このプロバイダによって、C#、Visual Basic .NET、C++ マネージ拡張、J# など、.NET でサポートされる任意の言語を使用して Adaptive Server 内のデータにアクセスできます。新しい機能にアクセスするには、次の項の説明に従って、バージョン 1.1 にアップグレードしてください。

ASE ADO.NET Data Provider 1.1 へのアップグレード

前のバージョンの ASE ADO.NET Data Provider に対して構築されたアプリケーションの Common Language Runtime (CLR) バイトコードは、ASE ADO.NET Data Provider 1.1 とは互換性がありません。更新バージョンの ASE ADO.NET Data Provider を使用する場合は、使用しているアプリケーションを再構築してください。クライアント・マシンでは、ASE ADO.NET Data Provider の複数のバージョンが共存できます。そのため、古いアプリケーションでは引き続き前のバージョンの ASE ADO.NET Data Provider を使用できます。詳細については、『ASE ADO.NET Data Provider ユーザーズ・ガイド』を参照してください。

ASE ADO.NET Data Provider 1.1 の新機能

ASE ADO.NET Data Provider 1.1 の新機能は、次のとおりです。

- Secure Sockets Layer (SSL)
- ディレクトリ・サービス – Lightweight Directory Access Protocol (LDAP)
- 高可用性 (HA) フェールオーバー
- パスワードの暗号化
- ASECommandBuilder
- ASECommandExecutXmlReader

新機能の詳細については、『ASE ADO.NET Data Provider ユーザーズ・ガイド』を参照してください。

ESD#2 の新機能

ESD#2 には、次の新機能があります。

Solaris と Microsoft Windows での SSL Plus 3.1.5 + SBGSE のサポート

次の製品で、Solaris 32 ビットと 64 ビット、Microsoft Windows での Secure Sockets Layer (SSL) Plus 3.1.5 + SBGSE がサポートされるようになりました。

- Software Developer's Kit のコンポーネント
 - Client-Library
 - ESQ/C
 - ESQ/COBOL
- Open Server

合衆国連邦規定により、Sybase では FIPS 認定の暗号スイートを使用します。SDK 12.5.1 ESD#2 (Open Client を含む) と Open Server 12.5.1 ESD#2 以降にアップグレードしたら、Microsoft Windows では次の手順に従って Certicom ライブラリを初期化すると、SSL テストを実行できるようになります。Solaris の場合は、*lib3p* ディレクトリが動的ロード・ライブラリ・パス上にあることを確認します。

❖ Windows での Certicom ライブラリの初期化

- 1 *ctlib* ライブラリと *srvlib* ライブラリが、*libs.dll* の Certicom 暗号スイートを検索できるようにするには、動的ロード・ライブラリ・パスに次の内容を追加します。

```
%SYBASE%\%SYBASE_OCS%\lib3p
```

- 2 次のプログラムを実行します。

```
%SYBASE%\%SYBASE_OCS%\bin\setsslreg.exe
```

このプログラムは、*libs.dll* がセルフテストに使用する項目をシステム・レジストリに追加します。マシンのレジストリにその値がある場合は、なんらかの理由でその値が削除されないかぎり、*setsslreg.exe* プログラムを再実行する必要はありません。

- 3 次のプログラムを実行します。

```
%SYBASE%\%SYBASE_OCS%\bin\sbgtest.exe
```

このプログラムは、レジストリ・エントリと *libs.dll* が正しく設定されていることを確認します (*lib3p* が処理パス上にあることを確認してください)。 *sbgtest.exe* は、*libs.dll* がレジストリ・エントリにアクセスでき、セルフテストにパスすることを確認するために使用します。

次のような場合は、*sbgtest.exe* を2回以上使用する可能性があります。

- *setsslreg* を実行した後、レジストリ・エントリが追加されていることを確認する。
- SSL 使用時に全体的な障害が発生した後、*libsb.dll* がまだ機能しているかどうかを確認する。

注意 SSL を使用するアプリケーションを実行する場合、*lib3p* ディレクトリが、その動的ロード・ライブラリ・パス上にあることが非常に重要です。

Solaris、Microsoft Windows XP、Microsoft Windows 2003 での IPV6 のサポート

次の製品で、Sun Solaris 8 (SPARC 32 ビット版と 64 ビット版)、Microsoft Windows XP と Windows 2003 で、IPV6 がサポートされるようになりました。

- Software Developer's Kit のコンポーネント
 - Client-Library
 - ESQL/C
 - ESQL/COBOL
- Open Server

Open Client と Open Server はそれぞれ、*getaddrinfo()* を使用して次のようにホスト名をアドレスに変換します。

- Open Client は、*connect()* が成功するまで、*getaddrinfo()* によって返される各アドレスを使用してサーバへの接続を確立しようとします。
- Open Server は、*getaddrinfo()* によって IPV6 アドレスが返されるかどうかに応じて、次のような動作でサーバへの接続を確立しようとします。
 - *getaddrinfo()* によって IPV6 アドレスが返されない場合は、*getaddrinfo()* が返す各アドレスに1つのリスナを確立します。
 - *getaddrinfo()* によって IPV6 アドレスが返された場合は、受信ポートのアドレスのワイルドカードを作成します。Open Server が、ホストの有効な IP アドレスと IPV6 アドレスの一部でのみ受信する場合は、使用する実際のアドレスを *interfaces* ファイルの別々の行にそれぞれ入力します。

注意 IPV6 仕様の定義によって、Open Server は、*getaddrinfo()* がループバック・アドレスを返したことを特定すると、IPV6 ローカル・ホストと IP ローカル・ホストの両方にリスナを確立します。

Microsoft Windows 版の DB-Library LDAP の機能

この機能によって、`dbopen()` を呼び出す DB-Library (*dblib*) アプリケーションが、LDAP ディレクトリ・サービスを使用する Adaptive Server または Open Server に接続できます。このためには、Microsoft Windows 版の『Open Client/Server 設定ガイド』の説明に従って、*libctl.cfg* ファイルの [DIRECTORY] セクションを変更します。この機能を使用するためにアプリケーションを変更する必要はありません。

Open Client 製品と Open Server 製品の既存の LDAP にアクセスする機能と、一般的なディレクトリ・サービス・アクセス・コードは、*netlib* にあります。したがって、アプリケーションには *netlib* へのリンクだけが必要です。

注意 Sybase では、I/O に *netlib* を使用しているため、現在の DB-Library が生成するメッセージのかわりに *netlib* エラー・メッセージが生成されることがあります。

ESD#1 の新機能

ESD#1 には、次の新機能があります。

Sun Solaris 8 (SPARC 64 ビット版) での XA サポート

Sun Solaris 8 (SPARC 64 ビット版) で、XA 64 ビット版が使用できるようになりました。他の 64 ビット・プラットフォームでは XA 64 ビット版はサポートされていませんが、現在次の 32 ビット・プラットフォームで XA 32 ビット版がサポートされています。

- IBM AIX 32 ビット版
- HP-UX (または HP-UX 11i v1.0) 32 ビット版
- Sun Solaris 8 (SPARC 32 ビット版)
- Microsoft Windows

XA インタフェースの概要

DTM XA インタフェースは、XA インタフェース標準を Sybase が Adaptive Server に実装したものです。DTM XA インタフェースは、X/Open 分散トランザクション処理 (DTP) モデルの 1 つの要素であり、DTP アプリケーション開発における業界標準です。XA インタフェースは、Adaptive Server に格納されているデータに、CICS、Encina、または TUXEDO のトランザクション・マネージャ (TM) からアクセスします。

XA インタフェースのコンポーネント

Sybase XA インタフェースは次のコンポーネントで構成されています。

- Sybase DTM XA インタフェース。XA インタフェースを Sybase が Adaptive Server に実装したものです。これは、Adaptive Server の機能ですが、ライセンスが別途必要です。
 - Adaptive Server と DTM 機能。ソフトウェアのインストールと機能ライセンスについては、使用しているプラットフォームに応じた Adaptive Server Enterprise の『インストール・ガイド』を参照してください。
 - Sybase Open Client。Client-Library 呼び出しを、アプリケーションとリソース・マネージャ間のネイティブ・インタフェースの一部にすることができます。
 - XA 設定ファイル。XA で使用するクライアント/サーバ接続の定義エントリが含まれています。
 - Embedded SQL/C と Embedded SQL/COBOL。ESQL 呼び出しを、アプリケーションとリソース・マネージャ間のネイティブ・インタフェースの一部として作成します。
 - XA 固有の dbcc コマンド・セット。システムの管理者はこれを使用してヒューリスティック・トランザクションを管理できます。
 - グローバル・リカバリに使用できる TM 固有の設定ファイルとコマンド。
- ネイティブの Adaptive Server DTM 機能の使用方法については、『XA インタフェース統合ガイド for CICS、Encina、TUXEDO』を参照してください。

