

新機能 Open Server™ 15.5 および SDK 15.5 Windows、Linux、UNIX、Mac OS X 版

ドキュメント ID : DC00071-01-1550-01

改訂 : 2010 年 3 月

このマニュアルでは、Sybase® Open Server™ 15.5 と Software Developer's Kit (SDK) 15.5 の新しい機能について説明します。

トピック	ページ
製品のプラットフォームと互換性	2
製品のコンポーネント	4
Open Server	4
Software Developer's Kit	5
SDK DB-Library Kerberos Authentication Option	6
IPv6 のサポート	6
文字セットのサポート	6
新しいバージョンへのアップグレード	7
ESD #3 の新機能	7
Open Client 15.5 と Open Server 15.5 の機能	8
マイクロ秒の精度の time データ	8
ct_send_data 拡張機能	14
Open Server の動的なリスナ	15
パフォーマンスの強化	18
CS_RES_NOXNLMETADATA 応答機能	18
FIPS-140-2 準拠のパスワード暗号化	18
SQL サーバの参照	18
jConnect および Adaptive Server のドライバおよびプロバイダに対応する SDK 15.5 機能	19
マイクロ秒の精度の time データ	19
FIPS-140-2 準拠のパスワード暗号化	20
分散トランザクション管理のサポート	20
簡素化された文字セット変換	20
JDBC 4.0 のサポート	21
Microsoft ADO.NET Entity Framework と LINQ のサポート	21
Adaptive Server 用の Microsoft Enterprise Library DAAB	22
アクセシビリティ機能	22

製品のプラットフォームと互換性

表 1 に、Open Server および SDK が構築されたプラットフォームとそれらの最初のリリース年月を示します。

表 1: Open Server および SDK をサポートするプラットフォーム

プラットフォーム	リリース日
Apple Mac OS X Intel (SDK のみ)	2009 年 11 月
HP HP-UX Itanium 32 ビット版および 64 ビット版	2009 年 11 月
HP HP-UX PA-RISC 32 ビット版および 64 ビット版	2009 年 11 月
IBM AIX POWER 32 ビット版および 64 ビット版	2009 年 11 月
Linux x86 32 ビット版	2009 年 11 月
Linux x86-64 64 ビット版	2009 年 11 月
Linux POWER 32 ビット版および 64 ビット版	2009 年 11 月
Microsoft Windows x86 32 ビット版	2009 年 11 月
Microsoft Windows x86-64 64 ビット版	2009 年 11 月
Sun Solaris x86-64 32 ビット版および 64 ビット版	2009 年 11 月
Sun Solaris SPARC 32 ビット版および 64 ビット版	2009 年 11 月

注意 上記のプラットフォームで Open Server と SDK のすべてのコンポーネントが使用できるとは限りません。各プラットフォームで使用できるコンポーネントについては、「製品のコンポーネント」(4 ページ)を参照してください。

表 2 に、Open Server および SDK 製品が構築およびテストされたプラットフォーム、コンパイラ、およびサードパーティ製品を示します。

表 2: Open Client と Open Server のプラットフォームの互換性の一覧

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)	Secure Sockets Layer (SSL)
Apple Mac OS X Intel (SDK のみ)	Mac OS X 10.5.4	gcc 4.0.1	サポート 予定なし	サポート 予定なし	サポート 予定なし	CSI-Crypto 2.6
HP HP-UX Itanium 32 ビット版	HP 11.23	HP ANSI C A.06.17	MF SE 5.0	MIT 1.4.1	OpenLDAP 2.4.16 (OpenSSL 0.9.8k を含む)	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
HP HP-UX Itanium 64 ビット版	HP 11.23	HP ANSI C A.06.17	MF SE 5.1	MIT 1.4.1	OpenLDAP 2.4.16 (OpenSSL 0.9.8k を含む)	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6

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)	Secure Sockets Layer (SSL)
HP HP-UX PA-RISC 32 ビット版	HP-UX 11.11	HP ANSI C++ A.03.064 / HP C 11.00	MF SE 5.1	Cybersafe Trustbroker 2.1、MIT 1.4.1	OpenLDAP 2.4.16 (OpenSSL 0.9.8k を含む)	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
HP HP-UX PA-RISC 64 ビット版	HP-UX 11.11	HP ANSI C++ A.03.064 / HP C 11.00	MF SE 5.1	MIT 1.4.1	OpenLDAP 2.4.16 (OpenSSL 0.9.8k を含む)	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
IBM AIX POWER 32 ビット版	AIX 5.3	XL C 8.0	MF SE 5.1	Cybersafe Trustbroker 2.1、MIT 2.6.5	OpenLDAP 2.4.16 (OpenSSL 0.9.8k を含む)	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
IBM AIX POWER 64 ビット版	AIX 5.3	XL C 8.0	MF SE 5.1	MIT 1.4.1	OpenLDAP 2.4.16 (OpenSSL 0.9.8k を含む)	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
Linux x86 32 ビット版	Red Hat Enterprise Linux 4.5	gcc 3.4.6 20060404	MF SE 5.1	MIT 1.4.1	OpenLDAP 2.4.16 (OpenSSL 0.9.8k を含む)	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
Linux x86-64 64 ビット版	Red Hat Enterprise Linux 4.5 (Nahant Update 4)	gcc 3.4.6 20060404	サポート 予定なし	MIT 1.4.1	OpenLDAP 2.4.16 (OpenSSL 0.9.8k を含む)	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
Linux POWER 32 ビット版	Red Hat Enterprise Linux 4.5	XL C 8.0	サポート 予定なし	MIT 1.4.1	OpenLDAP 2.4.16 (OpenSSL 0.9.8k を含む)	CSI-OpenSSL (SBGSE 2.2) CSI-Crypto 2.6
Linux POWER 64 ビット版	Red Hat Enterprise Linux 4.5	XL C 8.0	サポート 予定なし	MIT 1.4.1	OpenLDAP 2.4.16 (OpenSSL 0.9.8k を含む)	CSI-OpenSSL (SBGSE 2.2) CSI-Crypto 2.6
Microsoft Windows x86-64 64 ビット版	Windows 2003 Service Pack 1	Microsoft Visual C++ 2005	サポート 予定なし	サポート予定 なし	OpenLDAP 2.4.16 (OpenSSL 0.9.8k を含む)	CSI-OpenSSL CSI-Crypto 2.6
Microsoft Windows x86 32 ビット版	Windows 2003 Service Pack 2	Microsoft Visual C++ 2005	MF SE 5.1	Cybersafe Trustbroker 2.1、MIT 2.6.5	OpenLDAP 2.4.16 (OpenSSL 0.9.8k を含む)	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
Sun Solaris SPARC 32 ビット版	Solaris 8	Sun Studio 10	MF SE 5.1	Cybersafe Trustbroker 2.1、MIT 1.4.1	OpenLDAP 2.4.16 (OpenSSL 0.9.8k を含む)	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6

製品のコンポーネント

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)	Secure Sockets Layer (SSL)
Sun Solaris SPARC 64 ビット版	Solaris 8	Sun Studio 10	MF SE 5.1	Cybersafe Trustbroker 2.1、MIT 1.4.1	OpenLDAP 2.4.16 (OpenSSL 0.9.8k を含む)	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
Sun Solaris x86-64 32 ビット版	Solaris 10	Sun Studio 10	サポート 予定なし	MIT 1.4.1	OpenLDAP 2.4.16 (OpenSSL 0.9.8k を含む)	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
Sun Solaris x86-64 64 ビット版	Solaris 10	Sun Studio 10	サポート 予定なし	MIT 1.4.1	OpenLDAP 2.4.16 (OpenSSL 0.9.8k を含む)	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6

注意 最新の Open Server および SDK の動作確認サポートについては、Sybase platform certifications ページ (<http://certification.sybase.com/ucr/search.do>) を参照してください。

製品のコンポーネント

以下の項では、SDK および Open Server 製品のコンポーネントとそのコンポーネントがサポートされるプラットフォームの概要を説明します。

Open Server

Open Server は、Open Client™ または jConnect™ for JDBC™ ルーチンを通じて送信されたクライアント要求に応答するカスタム・サーバを作成するために使用できる API とサポート・ツールのセットです。表 3 に、Open Server のコンポーネントとそのコンポーネントがサポートされるプラットフォームを示します。

表 3: Open Server のコンポーネントとサポートされるプラットフォーム

Open Server のコンポーネント	プラットフォーム
Open Server Server-Library	Apple Mac OS X Intel を除くすべてのプラットフォーム
Open Server Client-Library	すべてのプラットフォーム
言語モジュール	すべてのプラットフォーム

Software Developer's Kit

Software Developer's Kit (SDK) は、クライアント・アプリケーションの開発に使用できるライブラリとユーティリティのセットです。表 4 に、SDK のコンポーネントとそのコンポーネントがサポートされるプラットフォームを示します。

表 4: SDK のコンポーネントとサポートされるプラットフォーム

SDK のコンポーネント	プラットフォーム
Open Client Client-Library	すべてのプラットフォーム
Open Client DB-Library™	すべてのプラットフォーム
Embedded SQL™/C (ESQL/C)	Apple Mac OS X Intel を除くすべてのプラットフォーム
Embedded SQL/COBOL (ESQL/COBOL)	<ul style="list-style-type: none"> • HP HP-UX Itanium 32 ビット版および 64 ビット版 • HP HP-UX PA-RISC 32 ビット版および 64 ビット版 • IBM AIX 32 ビット版および 64 ビット版 • Linux x86 32 ビット版 • Microsoft Windows x86 32 ビット版 • Sun Solaris SPARC 32 ビット版および 64 ビット版
Extended Architecture (XA)	<ul style="list-style-type: none"> • HP HP-UX Itanium 32 ビット版および 64 ビット版 • HP HP-UX PA-RISC 32 ビット版および 64 ビット版 • IBM AIX 32 ビット版および 64 ビット版 • Linux x86-64 64 ビット版 • Microsoft Windows x86 32 ビット版 • Microsoft Windows x86-64 64 ビット版 • Sun Solaris SPARC 32 ビット版および 64 ビット版 • Sun Solaris x86-64 64 ビット版
jConnect for JDBC	すべてのプラットフォーム
Sybase 製 Adaptive Server® Enterprise ODBC ドライバ	<ul style="list-style-type: none"> • Apple Mac OS X Intel • Linux x86 32 ビット版 • Linux x86-64 64 ビット版 • Microsoft Windows x86 32 ビット版 • Microsoft Windows x86-64 64 ビット版
Sybase 製 Adaptive Server Enterprise OLE DB プロバイダ	<ul style="list-style-type: none"> • Microsoft Windows x86 32 ビット版 • Microsoft Windows x86-64 64 ビット版

IPv6 のサポート

SDK のコンポーネント	プラットフォーム
Adaptive Server Enterprise ADO.NET Data Provider	<ul style="list-style-type: none">• Microsoft Windows x86 32 ビット版• Microsoft Windows x86-64 64 ビット版
言語モジュール	すべてのプラットフォーム

SDK DB-Library Kerberos Authentication Option

Sybase SDK DB-Library Kerberos Authentication Option では、MIT Kerberos セキュリティ・メカニズムを DB-Library で使用できます。これは、次のプラットフォームで使用できます。

- Linux x86 32 ビット版
- Microsoft Windows x86 32 ビット版
- Sun Solaris SPARC 32 ビット版および 64 ビット版

注意 SDK DB-Library Kerberos Authentication Option は別途提供されます。

IPv6 のサポート

Open Server と SDK は、これらの製品がリリースされているすべてのプラットフォームで IPv6 をサポートします。以下に示す *interfaces* ファイルのエントリでは、指定された最後の 2 つのサーバ・アドレスが IPv6 フォーマットになっています。

```
BARNARD_OS
master tcp ether barnards.sybase.com 18200
query tcp ether barnards.sybase.com 18200
master tcp ether barnards.v6.sybase.com 18200
query tcp ether barnards.v6.sybase.com 18200
master tcp ether
    fd77:55d:59d9:165:203:baff:fe68:aa12 18200
query tcp ether
    fd77:55d:59d9:165:203:baff:fe68:aa12 18200
```

文字セットのサポート

Open Server および SDK は、Adaptive Server でサポートされる文字セットをサポートしています。

新しいバージョンへのアップグレード

Open Server アプリケーション (srvlib) をアップグレードするには、次の手順を実行します。

- 静的にリンクしているアプリケーションでは、新しいバージョンのソフトウェアを使用してアプリケーションを完全に再構築します。新しいヘッダ・ファイルとライブラリを使用してアプリケーションを再コンパイルおよび再リンクします。
- 動的にリンクしているアプリケーションでは、ライブラリ名に“syb”が追加された SDK ライブラリを再コンパイルおよび再リンクします。

注意 アプリケーション・ファイルを変更した場合は、それらのファイルを再コンパイルする必要があります。

アプリケーションの構築に使用するバージョンと同じメジャー・リリースのランタイム・ライブラリを使用してください。

ESD #3 の新機能

jConnect のパフォーマンスを向上させるために、以下の接続プロパティが追加または修正されています。

- `DYNAMIC_PREPARE` — `true` に設定すると、動的 SQL 文をプリコンパイルできるようになります。ほとんどの場合、この設定によって jConnect のパフォーマンスが向上するため、このプロパティのデフォルト値が `true` に変更されました。

ただし、`prepare` コマンドを使用して、すべての SQL 文をプリコンパイルできるわけではありません。プリコンパイルできない SQL 文については、`PreparedStatement` を使用しないようにし、`DYNAMIC_PREPARE` を `false` に設定して、不要なプリコンパイルが試行されないようにすることをおすすめします。`DYNAMIC_PREPARE`、および `prepared` 文を使用する利点と欠点の詳細については、『jConnect for JDBC プログラマーズ・リファレンス』を参照してください。

- `IGNORE_WARNINGS` — 警告メッセージのチェックと生成を行うかどうかを指定します。タイムスタンプ値を `Adaptive Server` の `date` データ型と `time` データ型に格納するときに、Java タイムスタンプよりも精度が下がります。現在、このプロパティでは、この格納時の精度の損失に関する警告だけをチェックします。

有効な値は次のとおりです。

- `True` — `jConnect` は警告メッセージのチェックと生成を行わないため、パフォーマンスが向上する。
- `False` — デフォルト値。この値に設定すると、`jConnect` は警告メッセージのチェックと生成を行う。
- `OPTIMIZE_FOR_PERFORMANCE` — `jConnect` のパフォーマンスを強化するプロパティを有効にするかどうかを指定します。現在、このプロパティが制御するのは、`IGNORE_WARNINGS` プロパティだけです。

有効な値は次のとおりです。

- `True` — `jConnect` はパフォーマンス強化モードで実行される。
- `False` — デフォルト値。この場合、`jConnect` は通常モードで実行される。

注意 `IGNORE_WARNINGS` または `OPTIMIZE_FOR_PERFORMANCE` を `true` に設定する場合は、これらの設定がアプリケーションに及ぼす影響を十分にテストしてから設定するようにしてください。

Open Client 15.5 と Open Server 15.5 の機能

この項では、Open Client 15.5 および Open Server 15.5 の新機能について説明します。

マイクロ秒の精度の `time` データ

Open Client/Server の `CS_BIGDATETIME` データ型および `CS_BIGTIME` データ型は、マイクロ秒の精度の `time` データを提供します。これらのデータ型は、8 バイトの `binary` 値を保持します。

これらのデータ型はそれぞれ、`CS_DATETIME` データ型および `CS_TIME` データ型に似ています。`CS_BIGDATETIME` は、`CS_DATETIME` を使用する場所ならどこでも使用可能です。`CS_BIGTIME` は、`CS_TIME` を使用する場所ならどこでも使用可能です。`CS_DATETIME` データ型と `CS_TIME` データ型に適用できるすべての Open Client/Open Server ルーチンは、`CS_BIGDATETIME` データ型と `CS_BIGTIME` データ型にも適用できます。

- CS_BIGDATETIME は、Adaptive Server のデータ型 `bigdatetime` に対応し、0000 年 1 月 1 日の 00:00:00.000000 から経過したマイクロ秒数を示します。有効な CS_BIGDATETIME 値の範囲は、0001 年 1 月 1 日 00:00:00.000000 から 9999 年 12 月 31 日 23:59:59.999999 までです。

注意 0000 年 1 月 1 日の 00:00:00.000000 は、マイクロ秒数のカウントが開始される基本の値です。0001 年 1 月 1 日の 00:00:00.000000 より前の値は無効です。

次に示す CS_BIGDATETIME の定義は、`cstypes.h` にあります。

```
typedef CS_UBIGINT CS_BIGDATETIME;
```

- CS_BIGTIME は、Adaptive Server の `bigtime` データ型に対応し、当日の午前 0 時ちょうどから経過したマイクロ秒数を示します。有効な CS_BIGTIME 値の範囲は、00:00:00.000000 から 23:59:59.999999 までです。次に示す CS_BIGTIME の定義は、`cstypes.h` にあります。

```
typedef CS_UBIGINT CS_BIGTIME;
```

- CS_BIGDATETIME データと CS_BIGTIME データは、基本となるクライアント・プラットフォームのネイティブのバイト順序 (エンディアン) のクライアントに示されます。必要であればサーバで行われるバイト・スワッピングは、クライアントにデータを送る前、またはクライアントからのデータを受け取った後に行われます。

Open Client/Server のデータ型の詳細については、『Open Client Client-Library/C リファレンス・マニュアル』および『Open Server Server-Library/C リファレンス・マニュアル』をそれぞれ参照してください。

機能

`ct_capability` ルーチンでは、次のパラメータを使用して CS_BIGDATETIME および CS_BIGTIME をサポートします。

- CS_DATA_BIGDATETIME と CS_DATA_BIGTIME は、CS_BIGDATETIME と CS_BIGTIME がサポートされるかどうかを確認するためにサーバに送信される要求機能です。
- CS_DATA_NOBIGDATETIME と CS_DATA_NOBIGTIME は、CS_BIGDATETIME と CS_BIGTIME を使用すべきではないことを示す、サーバから送信される応答機能です。

要求機能と応答機能の詳細については、『Open Client Client-Library/C リファレンス・マニュアル』および『Open Server Server-Library/C リファレンス・マニュアル』を参照してください。

定数

次の定数は、CS_BIGDATETIME および CS_BIGTIME で使用するために提供されています。

- CS_BIGDATETIME_TYPE
- CS_BIGTIME_TYPE

Open Client/Server の定数の詳細については、『Open Client Client-Library/C リファレンス・マニュアル』および『Open Server Server-Library/C リファレンス・マニュアル』をそれぞれ参照してください。

マッピング

Adaptive Server の bigdatetime データ型および bigtime データ型はそれぞれクライアントの CS_BIGDATETIME データ型と CS_BIGTIME データ型に、クライアント・サイドの CS_DATAFMT 構造体に記述されたフォーマットでマップされます。CS_DATAFMT 構造体の datatype 設定は、CS_BIGDATETIME_TYPE または CS_BIGTIME_TYPE に設定する必要があります。

データ型の変換

表 5 に、CS_BIGDATETIME および CS_BIGTIME でサポートされるデータ型変換を示します。変換元のデータ型は一番左のカラムにあり、変換先データ型は表の上部にあります。“.” は変換がサポートされていることを示します。空白は変換がサポートされていないことを示します。

表 5: データ型の変換

Open Client のデータ型	CS_BINARY	CS_LNBINARY	CS_VARBINARY	CS_BIT	CS_CHAR	CS_LONGCHAR	CS_VARCHAR	CS_DATETIME	CS_DATETIME4	CS_BIGDATETIME	CS_TINYINT	CS_SMALLINT	CS_INT	CS_DECIMAL	CS_NUMERIC	CS_FLOAT	CS_REAL	CS_MONEY	CS_MONEY4	CS_BOUNDARY	CS_SENSITIVITY	CS_TEXT	CS_IMAGE	CS_UNICHAR	CS_DATE	CS_TIME	CS_BIGINT	CS_BIGTIME
CS_BINARY
CS_LONGBINARY
CS_VARBINARY
CS_BIT
CS_CHAR
CS_LONGCHAR
CS_VARCHAR
CS_DATETIME
CS_DATETIME4
CS_BIGDATETIME
CS_TINYINT

Open Client のデータ型	CS_BINARY	CS_LNBINARY	CS_VARBINARY	CS_BIT	CS_CHAR	CS_LONGCHAR	CS_VARCHAR	CS_DATETIME	CS_DATETIME4	CS_BIGDATETIME	CS_TINYINT	CS_SMALLINT	CS_INT	CS_DECIMAL	CS_NUMERIC	CS_FLOAT	CS_REAL	CS_MONEY	CS_MONEY4	CS_BOUNDARY	CS_SENSITIVITY	CS_TEXT	CS_IMAGE	CS_UNICHAR	CS_DATE	CS_TIME	CS_BIGINT	CS_BIGTIME
CS_SMALLINT	•	•	•	•	•	•	•				•	•	•	•	•	•	•	•	•			•	•			•		
CS_INT	•	•	•	•	•	•	•				•	•	•	•	•	•	•	•	•			•	•				•	
CS_DECIMAL	•	•	•	•	•	•	•				•	•	•	•	•	•	•	•	•			•	•				•	
CS_NUMERIC	•	•	•	•	•	•	•				•	•	•	•	•	•	•	•	•			•	•				•	
CS_FLOAT	•	•	•	•	•	•	•				•	•	•	•	•	•	•	•	•			•	•				•	
CS_REAL	•	•	•	•	•	•	•				•	•	•	•	•	•	•	•	•			•	•				•	
CS_MONEY	•	•	•	•	•	•	•				•	•	•	•	•	•	•	•	•			•	•				•	
CS_MONEY4	•	•	•	•	•	•	•				•	•	•	•	•	•	•	•	•			•	•				•	
CS_BOUNDARY					•	•	•													•		•						
CS_SENSITIVITY					•	•	•														•	•						
CS_TEXT	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			•	•				•	•
CS_IMAGE	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			•	•				•	
CS_UNICHAR	•	•	•	•	•	•	•	•	•	•																	•	•
CS_DATE			•		•	•	•	•	•	•												•	•	•	•			
CS_TIME			•		•	•	•	•	•	•												•	•	•	•	•		•
CS_BIGINT	•	•	•	•	•	•	•				•	•	•	•	•	•	•	•	•			•	•	•	•			
CS_BIGTIME			•		•	•	•	•	•	•												•	•			•	•	•

CS_BIGDATETIME および CS_BIGTIME と、以下のデータ型との間の変換はサポートされていません。

- CS_UBIGINT
- CS_UINT
- CS_USMALLINT
- CS_XML

CS_BIGDATETIME および CS_BIGTIME と CS_UNITEXT データ型との間の変換がサポートされます。

CS_BIGDATETIME と CS_BIGTIME 間の変換は、すべて既存の日時変換および時間変換と同様に処理されます。

CS_CHAR への変換

cs_dt_info ルーチンは、言語特有の日付、時刻、または日時情報を設定または取得します。日付変換フォーマットは、CS_BIGDATETIME または CS_BIGTIME の値が文字ベースのデータ型に変換される場合の結果のフォーマットを記述します。日付変換フォーマットを設定または取得するには、type パラメータの値に CS_DT_CONVFMT を設定して cs_dt_info を呼び出します。

表 6: type が CS_DT_CONVFMT (cs_dt_info) の場合の *buffer の値

記号値	CS_BIGDATETIME から変換された CS_CHAR (例: Aug 24 2009 5:36PM)	CS_DATE から変換された CS_CHAR (例: Aug 24 2009)	CS_BIGTIME から変換された CS_CHAR (例: 5:36PM)
CS_DATES_ HMSUSA、または CS_DATES_ HMSUSA_YYYY	hh:mm:ss.zzzzzz[AM PM] 5:36:00.000000PM	hh:mm:ss.zzzz[AM PM] 12:00:00:000000AM	hh:mm:ss.zzzz[AM PM] 5:36:00.000000PM
CS_DATES_ HMSUS、または CS_DATES_ HMSUS_YYYY	hh:mm:ss.zzzzzz 17:36:00.000000	hh:mm:ss.zzzzzz 00:00:00.000000	hh:mm:ss.zzzzzz 17:36:00.000000
CS_DATES_ LONGUSA	mon dd yy hh:mm:ss.zzzzzz[AM PM] Aug 24 09 5:36:00.000000PM	mon dd yy hh:mm:ss.zzzzzz[AM PM] Aug 24 09 12:00:00.000000AM	mon dd yy hh:mm:ss.zzzzzz[AM PM] Jan 1 01 5:36:00.000000PM
CS_DATES_ LONGUSA_YYYY	mon dd yyyy hh:mm:ss.zzzzzz[AM PM] Aug 24 2009 5:36:00.000000PM	mon dd yyyy hh:mm:ss.zzzzzz[AM PM] Aug 24 2009 12:00:00.000000AM	mon dd yyyy hh:mm:ss.zzzzzz[AM PM] Jan 1 0001 5:36:00.000000PM
CS_DATES_ LONGUS	mon dd yy hh:mm:ss.zzzzzz Aug 24 09 17:36:00.000000	mon dd yy hh:mm:ss.zzzzzz Aug 24 09 00:00:00.000000	mon dd yy hh:mm:ss.zzzzzz Jan 1 01 17:36:00.000000
CS_DATES_ LONGUS_YYYY	mon dd yyyy hh:mm:ss.zzzzzz Aug 24 2009 17:36:00.000000	mon dd yy hh:mm:ss.zzzzzz Aug 24 2009 00:00:00.000000	mon dd yy hh:mm:ss.zzzzzz Jan 1 0001 17:36:00.000000
CS_DATES_ YMDHMSUS_YYYY	yyyy-mm-dd hh:mm:ss.zzzzzz 2009-08-24 17:36:00.000000	yyyy-mm-dd hh:mm:ss.zzzzzz 2009-08-24 00:00:00.000000	yyyy-mm-dd hh:mm:ss.zzzzzz 0001-01-01 17:36:00.000000

cs_dt_crack の更新

cs_dt_crack CS-Library ルーチンを使用する場合、CS_DATE、CS_TIME、CS_DATETIME、CS_DATETIME4 の各データ型を示すために使用される *datatype* パラメータ値に加え、次の値を使用して、CS_BIGDATETIME データ型と CS_BIGTIME データ型を示すことができます。

datatype の値	意味
CS_BIGDATETIME_TYPE	CS_BIGDATETIME *dateval
CS_BIGTIME_TYPE	CS_BIGTIME *timeval

CS_DATEREC 構造体の次の 2 つのフィールドは、マイクロ秒の精度に対応します。

- *datesecfrac* は、秒以下の数を格納する CS_INT フィールドです。
- *datesecprec* は、精度を格納する CS_INT フィールドです。CS_BIGDATETIME と CS_BIGTIME の場合、このフィールドは常に 10^6 になります。

これらのフィールドは、ミリ秒より精度の高い *datetime* データ型でのみ使用されます。

CS_DATEREC は次のように定義されています。

```
typedef struct cs_daterec {
    CS_INT    dateyear;        /* year          */
    CS_INT    datemonth;      /* month         */
    CS_INT    datedmonth;     /* day of month  */
    CS_INT    datedyear;     /* day of year   */
    CS_INT    datedweek;     /* day of week   */
    CS_INT    datehour;      /* hour          */
    CS_INT    dateminute;    /* minute        */
    CS_INT    datesecsecond; /* second        */
    CS_INT    datemsecond;   /* millisecond   */
    CS_INT    datetzone;     /* timezone      */
    CS_INT    datesecfrac;  /* second fractions */
    CS_INT    datesecprec; /* precision       */
} CS_DATEREC;
```

cs_dt_crack の詳細については、『Open Client/Server Common Libraries リファレンス・マニュアル』を参照してください。

bcp ファイル記憶タイプ

bcp が呼び出されたときに、`-c`、`-f`、または `-n` パラメータに値が指定されていない場合は、bcp プロンプトがファイル記憶タイプを要求します。ファイル記憶タイプは Adaptive Server で有効な任意のデータ型です。bigdatetime および bigtime Adaptive Server の記憶タイプは次のように指定されます。

記憶タイプ	テーブルでのデータ型
A	bigdatetime
B	bigtime

bcp の詳細については、使用しているプラットフォームの『Open Client/Server プログラマーズ・ガイド補足』を参照してください。

bcp ホスト・ファイルのデータ型の記憶フォーマット

次のホスト・ファイルのデータ型は、bigdatetime データ型または bigtime データ型を使用する bcp フォーマット ファイルに指定できます。

表 7: ホスト・ファイルのデータ型の記憶フォーマット

記憶フォーマット	Adaptive Server データ型
SYBBIGDATETIME	bigdatetime
SYBBIGTIME	bigtime

ct_send_data 拡張機能

Open Client アプリケーションは通常、writetext ストリームを使用して text データおよび image データを Open Server アプリケーションに送信します。パフォーマンスを向上させるためには、この方法ではなく、Open Client アプリケーションで text データおよび image データを Open Server アプリケーションのバルク・ハンドラに直接送信できます。

ct_send_data コマンドの詳細については、『Open Client Client-Library/C リファレンス・マニュアル』を参照してください。

text データと image データの処理

text または image カラムを更新するには、通常、クライアント・アプリケーションでは ct_command ルーチン呼び出し、データ送信コマンドを開始します。その後、クライアントは ct_data_info コマンド呼び出し、CS_IODESC を取得し、後続の ct_send_data ルーチン呼び出しで生成する適切な SQL コマンド (update または writetext) を判断します。text データまたは image データを送信するために、ct_send_data が追加で 1 回以上呼び出されます。クライアントは終了時に、ct_send および ct_results ルーチン呼び出しします。サーバで言語イベントが発生し、SQL コマンドが処理されます。バルク・イベントが続くと、サーバは指定された量の text または image データを読み込みます。

この処理を単純化しパフォーマンスを向上させるために、クライアントは SQL コマンド (`update` または `writetext`) の生成を抑制し、サーバのバルク・ハンドラに直接データを送信することもできます。その場合、クライアントは `type` パラメータを `CS_SEND_DATA_NOCMD` に設定し、`ct_command` ルーチン呼び出すことによってデータ送信コマンドを開始しなければなりません。その後、クライアント・アプリケーションは、データ送信コマンドを使用してサーバのバルク・ハンドラに `text` データのみまたは `image` データのみを送信できます。サーバでバルク・イベントが発生すると、送信する合計バイト数を示す 4 バイトのフィールドに続き、`text` または `image` データが送信されます。バルク・ハンドラは `srv_text_info` を使用して予想される合計バイト数を読み取り、`srv_get_data` を使用してデータを読み取ります。

sp_mda

サーバは、ストアド・プロシージャ `sp_mda` を定義して、SQL コマンドを使用せずに `text` データまたは `image` データのみを送信する `ct_send_data` ルーチンがサーバでサポートされているかどうかを示す必要があります。`sp_mda` は、サーバにログインする前に `CS_SENDDATA_NOCMD` 接続プロパティが設定されている場合のみ呼び出されます。

サーバが SQL コマンドを使用せずに `text` データまたは `image` データを受信できる場合、`sp_mda` は次の結果を返します。

パラメータ	値
<code>mdinfo</code>	“SENDDATA_NOCMD”
<code>querytype</code>	2
<code>query</code>	<code>senddata no cmd</code>

注意 Adaptive Server では、SQL コマンドを使用せずに `image` データまたは `text` データを受信することはできません。

Open Server の動的なリスナ

Open Server の動的なリスナでは、`srv_run()` を呼び出した後でも、Open Server アプリケーションで新しいリスナを起動できます。これにより、Open Server アプリケーションは `interface` ファイル内の最小のエントリで開始し、リスナを再起動できます。この項で説明されている機能は、動的なリスナをサポートします。

イベント

SRV_LISTEN_PREBIND と SRV_LISTEN_POSTBIND の 2 つの Open Server イベントは、動的なリスナをサポートします。

表 8: Open Server イベント

イベント	説明	ハンドラに対する引数	デフォルト・イベント・ハンドラ
SRV_LISTEN_PREBIND	指定された SRV_PROC 制御構造体で識別されたリスナの追加設定を許可する。たとえば、代替の SSL 証明書を指定できる。この設定は、スレッド・プロパティを使用して実行される。	SRV_PROC*	デフォルト・ハンドラはアクションを起こさない。
SRV_LISTEN_POSTBIND	指定された SRV_PROC 制御構造体で識別されたリスナの最終的な設定を判断できる。たとえば、リスナがバインドされているアドレスを判断できる。 設定は、スレッド・プロパティを使用して判断される。	SRV_PROC*	デフォルト・ハンドラはアクションを起こさない。

サーバ・プロパティ

SRV_S_NUMLISTENERS と SRV_S_MAXLISTENERS の 2 つの Open Server プロパティは、動的なリスナをサポートします。

表 9: サーバ・プロパティ

プロパティ名	定義	注意
SRV_S_MAXLISTENERS	リスナ・スレッドの最大値を制限する。	デフォルト値は CS_MAX_NOMAX。この値は、新しいリスナ・スレッドを作成するときのみ使用される。このプロパティを SRV_S_NUMLISTENERS の現在の値よりも小さい値に設定すると、リスナは失敗しない。
SRV_S_NUMLISTENERS	クライアント接続の受信に関連する SRV_PROC 制御構造体の数を返す。	これは、取得だけが可能なプロパティである。

スレッド・タイプ

リスナ・スレッドのスレッド・タイプは、SRV_TLISTENER になります。

スレッド・プロパティ

SRV_T_LISTENADDR、SRV_T_LOCALID、および SRV_T_REMOTEADDR のスレッド・プロパティを使用して、動的なリスナをサポートするようにスレッドを設定します。

表 10: スレッド・プロパティ

プロパティ名	定義	注意
SRV_T_LISTENADDR	指定された SRV_PROC 制御構造体で識別されたリスナのアドレスを返す。 SRV_PROC がリスナの場合、このプロパティは、リスナが接続を受け入れるアドレスを返す。	このプロパティには、引数として CS_TRANADDR 構造体へのポインタが必要。 これは、取得だけが可能なプロパティである。
SRV_T_LOCALID	リスナで使用する SSL 証明書を指定する。	このプロパティを使用すると、リスナはグローバル・サーバ・レベルの SSL 証明書以外の SSL 証明書を使用できる。 これは、設定専用プロパティである。
SRV_T_REMOTEADDR	SRV_PROC ピアのアドレスを返す。このプロパティは、クライアントの SRV_PROC に対してのみ有効である。	このプロパティには、引数として CS_TRANADDR 構造体へのポインタが必要。 これは、取得だけが可能なプロパティである。

SRV_LISTEN_PREBIND Open Server イベント内から SRV_T_CIPHER_SUITE および SRV_T_SSL_VERSION スレッド・プロパティを設定します。

動的なリスナの起動

動的なリスナを起動するには、次のパラメータ値を使用する `srv_spawn` ルーチン呼び出します。

- `sppp` – スレッド構造体ポインタへのポインタは NULL であることが必要です。
- `stacksize` – スタック・サイズを CS_UNUSED として指定します。
- `funcp` – エントリ・ポイント関数ポインタは SRV_C_START_LISTENER であることが必要です。
- `argp` – このパラメータが CS_TRANADDR 構造体を指し示すようにします。
- `priority` – CS_UNUSED として指定します。

Open Server イベント、プロパティ、スレッドの詳細については、『Open Server Server-Library/C リファレンス・マニュアル』を参照してください。

パフォーマンスの強化

以下のパフォーマンスの強化が実装されています。

- Client-Library ルーチンおよび Bulk-Library ルーチン (特に `ct_fetch()`、`blk_rowxfer()`、`blk_rowxfer_mult()`) のネットワーク I/O 処理の改善。
- Sun Solaris および Linux でのコンパイラ最適化設定の改善。

CS_RES_NOXNLMETADATA 応答機能

CS_RES_NOXNLMETADATA 応答機能は、サーバ・アプリケーションがクライアント・アプリケーションに送信する情報のタイプと構造を最適化することによって、サーバ・アプリケーションとクライアント・アプリケーションのパフォーマンスを向上させます。この情報が不要である場合には、CS_RES_NOXNLMETADATA を使用してメタデータ (カラム・ラベル、カタログ名、スキーマ名、テーブル名など) を送信しないようサーバに伝えることができます。

デフォルトでは、バージョンが CS_VERSION_125、CS_VERSION_150、CS_VERSION_155 に設定されているアプリケーションでは CS_RES_NOXNLMETADATA は有効です。応答機能の詳細については、『Open Client Client-Library/C リファレンス・マニュアル』または『Open Server Server-Library/C リファレンス・マニュアル』を参照してください。

FIPS-140-2 準拠のパスワード暗号化

Open Client と Open Server のログイン・パスワードとリモート・パスワードの暗号化は、Sybase CSI (Common Security Infrastructure) によって実現されます。CSI 2.6 は、連邦情報処理標準 (FIPS: Federal Information Processing Standard) 140-2 に準拠しています。

FIPS 暗号化をサポートするために、Certicom Security Builder をまだ使用していないプラットフォームには、`libsbgse2.so` (UNIX と Linux の各プラットフォーム) または `libsbgse2.dll` (Microsoft Windows プラットフォーム) という名前の Certicom Security Builder 共有ライブラリがインストールされます。また、`$$SYBASE/$SYBASE_OCS/lib3p` または `$$SYBASE/$SYBASE_OCS/lib3p64` にある `sybsci` サブディレクトリは、削除されました。

Open Client/Server における SSL (Secure Sockets Layer) の詳細については、プラットフォームの『Open Client/Server 設定ガイド』を参照してください。

SQL サーバの参照

Open Client/Open Server エラー メッセージおよびヘッダ・ファイルでの「SQL Server」に対する参照は、「Server」に置き換えられました。

jConnect および Adaptive Server のドライバおよびプロバイダに対応する SDK 15.5 機能

ここでは、次の SDK 15.5 コンポーネントの新機能について説明します。

- jConnect 7.0 for JDBC
- Sybase 製 Adaptive Server ODBC ドライバ
- Sybase 製 Adaptive Server OLE DB プロバイダ
- Sybase 製 Adaptive Server ADO.NET Data Provider 2.155

jConnect または Adaptive Server のドライバおよびプロバイダの詳細については、『jConnect for JDBC プログラマーズ・リファレンス』、『Adaptive Server Enterprise ODBC ドライバ・ユーザーズ・ガイド for Microsoft Windows、Linux、Mac OS X』、『Adaptive Server Enterprise ADO.NET Data Provider ユーザーズ・ガイド for Microsoft Windows』、または『Adaptive Server Enterprise OLE DB プロバイダ・ユーザーズ・ガイド for Microsoft Windows』を参照してください。

マイクロ秒の精度の time データ

jConnect 7.0 および Adaptive Server ODBC ドライバ、OLE DB プロバイダ、および ADO.NET Data Provider 2.155 は、SQL データ型の **bigdatetime** と **bigtime** をサポートすることで、マイクロ秒レベルの精度の time データを提供します。

bigdatetime と **bigtime** は同様に機能し、SQL データ型の **datetime** および **time** とデータ・マッピングが同じです。

- **bigdatetime** は、Adaptive Server のデータ型 **bigdatetime** に対応し、0000 年 1 月 1 日の 0:00:00.000000 から経過したマイクロ秒数を示します。有効な **bigdatetime** 値の範囲は、0001 年 1 月 1 日の 0:00:00.000000 から 9999 年 12 月 31 日の 23:59:59.999999 までです。
- **bigtime** は、Adaptive Server の **bigtime** データ型に対応し、当日の午前 0 時ちょうどから経過したマイクロ秒数を示します。有効な **bigtime** 値の範囲は、00:00:00.000000 から 23:59:59.999999 までです。
- Adaptive Server 15.5 への接続時に、jConnect 7.0 for JDBC と Adaptive Server ODBC ドライバ、OLE DB プロバイダ、および ADO.NET Data Provider 2.155 は、**bigdatetime** データ型と **bigtime** データ型を使用してデータを転送します。受信した Adaptive Server カラムが **datetime** および **time** として定義されている場合でも同様です。

使用法

これは、Adaptive Server は、Adaptive Server カラムに合わせるために、ドライバやデータ・プロバイダから取得した値を暗黙的にトランケートする可能性があることを意味します。たとえば、**bigtime** の値 23:59:59.999999 は、**time** データ型の Adaptive Server カラムに 23:59:59.996 として保存されます。

- Adaptive Server 15.0.x 以前のバージョンへの接続時には、jConnect 7.0 および Adaptive Server ODBC ドライバ、OLE DB プロバイダ、および ADO.NET Data Provider 2.155 は、`datetime` データ型と `time` データ型を使用してデータを転送します。
- Adaptive Server 15.5 への接続時には、jConnect 6.05 と Adaptive Server ODBC ドライバ 15.0、OLE DB プロバイダ 15.0、ADO.NET Data Provider 1.15 は、`datetime` データ型と `time` データ型を使用します。

FIPS-140-2 準拠のパスワード暗号化

Adaptive Server ODBC ドライバ、OLE DB プロバイダ、ADO.NET Data Provider 2.155 では、Sybase CSI (Common Security Infrastructure) を使用してログインおよびリモート・パスワードを暗号化できます。CSI 2.6 は、連邦情報処理標準 (FIPS: Federal Information Processing Standard) 140-2 に準拠しています。

分散トランザクション管理のサポート

Adaptive Server ODBC ドライバ、OLE DB プロバイダ、および ADO.NET Data Provider 2.155 は、Adaptive Server クラスター・エディションの機能でもある分散トランザクションをサポートしています。分散トランザクションの詳細については、『Adaptive Server Enterprise 分散トランザクション管理機能の使用』を参照してください。

簡素化された文字セット変換

以前のリリースの Adaptive Server ODBC ドライバと OLE DB データ・ドライバでは、クライアント・アプリケーションと Adaptive Server がドライバの CharSet 接続プロパティで指定された文字セットを使用して通信します。ドライバは、クライアント・アプリケーションと Adaptive Server の両方が指定された文字セットをサポートしているものと見なします。これに該当しない場合に対処するために、`NormalizeUnicodeString` などの追加の接続プロパティが作成されました。

Adaptive Server ODBC ドライバと OLE DB データ・プロバイダ・バージョン 15.5 での文字セット変換を簡素化するために、Adaptive Server との通信に使用される文字セットだけを指定するよう CharSet が再定義されています。クライアント・アプリケーションとの通信にドライバが使用する文字セットを指定するための `ClientCharset` 接続プロパティが作成されました。

`ClientCharset` を `Charset` と同じ値に設定すると、Adaptive Server ODBC ドライバおよび OLE DB データ・プロバイダ 15.0 以前の動作を再現できます。

JDBC 4.0 のサポート

jConnect 7.0 は、JDBC 4.0 の次の仕様をサポートしています。

- 接続管理
- SQL ドライバの自動ロード
- 各国文字セット変換
- データベース・メタデータ
- ラッパ・パターン
- スカラ関数 CHAR_LENGTH, CHARACTER_LENGTH, CURRENT_DATE, CURRENT_TIME, CURRENT_TIMESTAMP, EXTRACT, OCTET_LENGTH, POSITION

JDBC 4.0 の仕様の詳細については、Sun Developer Network を参照してください。
(<http://developers.sun.com/>)

Microsoft ADO.NET Entity Framework と LINQ のサポート

Adaptive Server ADO.NET Data Provider は、Visual Studio LINQ (Language-Integrated Query) と、LINQ-to-SQL コンポーネントを含む、Microsoft ADO.NET Entity Framework をサポートしています。ただし、Microsoft ADO.NET Entity Framework の制限により、次の処理はサポートされていません。

- LINQ Contains 拡張メソッドの使用。Contains は SQL の IN 句に対応します。
- LINQ 拡張メソッドの作成。
- エンティティ・クラス間の関連付けの作成。

ADO.NET Entity Framework と LINQ の利点としては、リレーショナル・ストレージ・スキーマの概念モデルを操作できることが挙げられます。これにより、データ指向アプリケーションの開発および保守に伴う作業を簡素化できます。Microsoft ADO.NET Entity Framework と LINQ を使用するには、*Sybase.AdoNet2.AseClient.dll* への参照を追加します。

ADO.NET Entity Framework と LINQ の詳細については、Microsoft Developer Network を参照してください。(<http://msdn.microsoft.com>)

Adaptive Server 用の Microsoft Enterprise Library DAAB

Adaptive Server ADO.NET Data Provider 2.155 は、Microsoft Enterprise Library 4.1 DAAB (Data Access Application Block) の機能を拡張して Adaptive Server をサポートします。DAAB はクラスのコレクションであり、データベース・インスタンスの作成やデータベース・レコードの更新など、一般的なデータベース機能を簡素化します。また、DAAB は、データベース固有の機能のカプセル化も行います。これにより、データベースに依存しないアプリケーション設計が可能になります。

Adaptive Server クラス用の DAAB は、Microsoft .NET Framework 3.5 と Microsoft Visual Studio 2008 でサポートされています。Adaptive Server 用の DAAB を使用するには、Microsoft Enterprise Library 4.1 のコピーを更新する必要があります。詳細については、『Adaptive Server Enterprise ADO.NET Data Provider 2.155 ユーザーズ・ガイド』を参照してください。

Enterprise Library DAAB (Data Access Application Block) の詳細については、Microsoft Developer Network (<http://msdn.microsoft.com>) を参照してください。

アクセシビリティ機能

第 508 条では、米国連邦機関の電子および情報技術は身体障害者が利用できるものと規定されています。Sybase は、第 508 条を強力にサポートし、Open Client および Open Server バージョン 15.5 を含む広範な製品を、第 508 条に準拠させています。

リリース 15.5 のマニュアルには、アクセシビリティを重視した HTML 版もあります。この HTML 版マニュアルは、スクリーン・リーダーで読み上げる、または画面を拡大表示する方法により、その内容を理解できるよう配慮されています。Open Client および Open Server のマニュアルは、連邦リハビリテーション法第 508 条のアクセシビリティ規定に準拠していることがテストにより確認されています。第 508 条に準拠しているマニュアルは通常、World Wide Web Consortium (W3C) の Web サイト用ガイドラインなど、米国以外のアクセシビリティ・ガイドラインにも準拠しています。

アクセシビリティ・ツールを効率的に使用するには、設定が必要な場合もあります。一部のスクリーン・リーダーは、テキストの大文字と小文字を区別して発音します。たとえば、すべて大文字のテキスト (ALL UPPER CASE TEXT など) はイニシャルで発音し、大文字と小文字の混在したテキスト (Mixed Case Text など) は単語として発音します。構文規則を発音するようにツールを設定すると便利かもしれません。詳細については、ツールのマニュアルを参照してください。

Sybase のアクセシビリティに対する取り組みについては、[Sybase Accessibility \(http://www.sybase.com/accessibility\)](http://www.sybase.com/accessibility) を参照してください。Sybase Accessibility サイトには、第 508 条と W3C 標準に関する情報へのリンクもあります。