New Features
Open Server™ 15.5 and SDK 15.5
for Windows, Linux, UNIX, and Mac OS X

Document ID: DC20155-01-1550-09

Last revised: October 2011

This document describes the new features in Sybase® Open Server™ 15.5 and the Software Developer's Kit (SDK) 15.5.

Product platforms and compatibilities

Table 1 lists the platforms and the year Open Server and SDK were first built and released on these platforms:

Table 1: Platforms that support Open Server and SDK

Platform	Release date
Apple Mac OS X Intel (SDK only)	November 2009
HP HP-UX Itanium 32-bit and 64-bit	November 2009
HP HP-UX PA-RISC 32-bit and 64-bit	November 2009
IBM AIX POWER 32-bit and 64-bit	November 2009
Linux x86 32-bit	November 2009
Linux x86-64 64-bit	November 2009
Linux POWER 32-bit and 64-bit	November 2009
Microsoft Windows x86 32-bit	November 2009
Microsoft Windows x86-64 64-bit	November 2009
Sun Solaris x86-64 32-bit and 64-bit	November 2009
Sun Solaris SPARC 32-bit and 64-bit	November 2009

Note Not all Open Server and SDK components are available on the platforms listed above. See “Product components” on page 5 for the complete list of components available on each platform.

Table 2 lists the platforms, compilers, and third-party products Open Server and SDK products are built and tested on:

Table 2: Open Client and Open Server platform compatibility matrix

Platform	Operating system level	C and C++ compilers	COBOL compiler	Kerberos version	Lightweight Directory Access (LDAP)	Secure Sockets Layer (SSL)
Apple Mac OS X Intel (SDK only)	Mac OS X 10.5.4	gcc 4.0.1	None planned	None planned	None planned	CSI-Crypto 2.6
HP HP-UX Itanium 32-bit	HP 11.23	HP ANSI C A.06.17	MF SE 5.0	MIT 1.4.1	OpenLDAP 2.4.16 with OpenSSL 0.9.8k	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6

Platform	Operating system level	C and C++ compilers	COBOL compiler	Kerberos version	Lightweight Directory Access (LDAP)	Secure Sockets Layer (SSL)
HP HP-UX Itanium 64-bit	HP 11.23	HP ANSI C A.06.17	MF SE 5.1	MIT 1.4.1	OpenLDAP 2.4.16 with OpenSSL 0.9.8k	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
HP HP-UX PA-RISC 32-bit	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 with OpenSSL 0.9.8k	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
HP HP-UX PA-RISC 64-bit	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 with OpenSSL 0.9.8k	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
IBM AIX POWER 32-bit	AIX 5.3	XL C 8.0	MF SE 5.1	Cybersafe Trustbroker 2.1, MIT 2.6.5	OpenLDAP 2.4.16 with OpenSSL 0.9.8k	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
IBM AIX POWER 64-bit	AIX 5.3	XL C 8.0	MF SE 5.1	MIT 1.4.1	OpenLDAP 2.4.16 with OpenSSL 0.9.8k	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
Linux x86 32-bit	Red Hat Enterprise Linux 4.5	gcc 3.4.6 20060404	MF SE 5.1	MIT 1.4.1	OpenLDAP 2.4.16 with OpenSSL 0.9.8k	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
Linux x86-64 64-bit	Red Hat Enterprise Linux 4.5 (Nahant Update 4)	gcc 3.4.6 20060404	MF SE 5.1	MIT 1.4.1	OpenLDAP 2.4.16 with OpenSSL 0.9.8k	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
Linux POWER 32-bit	Red Hat Enterprise Linux 4.5	XL C 8.0	None planned	MIT 1.4.1	OpenLDAP 2.4.16 with OpenSSL 0.9.8k	CSI-OpenSSL (SBGSE 2.2) CSI-Crypto 2.6
Linux POWER 64-bit	Red Hat Enterprise Linux 4.5	XL C 8.0	MF SE 5.1	MIT 1.4.1	OpenLDAP 2.4.16 with OpenSSL 0.9.8k	CSI-OpenSSL (SBGSE 2.2) CSI-Crypto 2.6
Microsoft Windows x86-64 64-bit	Windows 2003 Service Pack 1	Microsoft Visual C++ 2005	MF Net Express 5.1	Cybersafe Trustbroker 4.3.1.	OpenLDAP 2.4.16 with OpenSSL 0.9.8k	CSI-OpenSSL CSI-Crypto 2.6

Platform	Operating system level	C and C++ compilers	COBOL compiler	Kerberos version	Lightweight Directory Access (LDAP)	Secure Sockets Layer (SSL)
Microsoft Windows x86 32-bit	Windows 2003 Service Pack 2	Microsoft Visual C++ 2005	MF Net Express 5.1	Cybersafe Trustbroker 2.1, MIT 2.6.5	OpenLDAP 2.4.16 with OpenSSL 0.9.8k	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
Sun Solaris SPARC 32-bit	Solaris 8	Sun Studio 10	MF SE 5.1	Cybersafe Trustbroker 2.1, MIT 1.4.1	OpenLDAP 2.4.16 with OpenSSL 0.9.8k	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
Sun Solaris SPARC 64-bit	Solaris 8	Sun Studio 10	MF SE 5.1	Cybersafe Trustbroker 2.1, MIT 1.4.1	OpenLDAP 2.4.16 with OpenSSL 0.9.8k	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
Sun Solaris x86 32-bit	Solaris 10	Sun Studio 10	MF SE 5.1	MIT 1.4.1	OpenLDAP 2.4.16 with OpenSSL 0.9.8k	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6
Sun Solaris x86-64 64-bit	Solaris 10	Sun Studio 10	MF SE 5.1	MIT 1.4.1	OpenLDAP 2.4.16 with OpenSSL 0.9.8k	Certicom SSL Plus 5.2.2 (SBGSE 2.2) CSI-Crypto 2.6

Note For the most current Open Server and SDK certifications support, see the Sybase platform certifications page at http://certification.sybase.com/ucr/search.do

Product components

The following sections summarize the components of SDK and Open Server products, and the platforms these components are supported on.

Open Server

Open Server is a set of APIs and supporting tools that you can use to create custom servers to respond to client requests submitted through Open Client™ or jConnect™ for JDBC™ routines. Table 3 lists the Open Server components and the platforms these components are supported on.

Table 3: Open Server components and supported platforms

Open Server components	Platforms
Open Server Server-Library	All platforms except Apple Mac OS X Intel
Open Server Client-Library	All platforms
Language modules	All platforms

Software Developer’s Kit

The Software Developer’s Kit (SDK) is a set of libraries and utilities you can use to develop client applications. Table 4 lists the SDK components and the platforms these components are supported on.

Table 4: SDK components and supported platforms

SDK components	Platforms
Open Client Client-Library	All platforms
Open Client DB-Library™	All platforms
Embedded SQL™/C (ESQL/C)	All platforms except on Apple Mac OS X Intel
Embedded SQL/COBOL (ESQL/COBOL)	• HP HP-UX Itanium 32-bit and 64-bit • HP HP-UX PA-RISC 32-bit and 64-bit • IBM AIX 32-bit and 64-bit • Linux POWER 64-bit • Linux x86 32-bit • Linux x86-64 64-bit • Microsoft Windows x86 32-bit • Microsoft Windows x86-64 64-bit • Sun Solaris SPARC 32-bit and 64-bit • Sun Solaris x86 32-bit • Sun Solaris x86-64 64-bit **Note** Except for Linux POWER 64-bit, all 64-bit platforms support both 32-bit and 64-bit ESQL/COBOL.

SDK components	Platforms
Extended Architecture (XA)	• HP HP-UX Itanium 32-bit and 64-bit • HP HP-UX PA-RISC 32-bit and 64-bit • IBM AIX 32-bit and 64-bit • Linux x86-64 64-bit • Microsoft Windows x86 32-bit • Microsoft Windows x86-64 64-bit • Sun Solaris SPARC 32-bit and 64-bit • Sun Solaris x86-64 64-bit
jConnect for JDBC	All platforms
Adaptive Server® Enterprise ODBC Driver by Sybase	• Apple Mac OS X Intel • HP HP-UX Itanium 64-bit • IBM AIX POWER 64-bit • Linux x86 32-bit • Linux x86-64 64-bit • Linux POWER 64-bit • Microsoft Windows x86 32-bit • Microsoft Windows x86-64 64-bit • Sun Solaris SPARC 64-bit • Sun Solaris x86-64 64-bit
Adaptive Server Enterprise OLE DB Provider by Sybase	• Microsoft Windows x86 32-bit • Microsoft Windows x86-64 64-bit
Adaptive Server Enterprise ADO.NET Data Provider	• Microsoft Windows x86 32-bit • Microsoft Windows x86-64 64-bit
Language modules	All platforms

SDK DB-Library Kerberos Authentication Option

The Sybase SDK DB-Library Kerberos Authentication Option allows the MIT Kerberos security mechanism to be used on DB-Library and is available on:

- Linux x86 32-bit
- Microsoft Windows x86 32-bit
- Sun Solaris SPARC 32-bit and 64-bit

IPv6 support

Open Server and SDK support IPv6 on all the platforms on which these products are released. Below is an *interfaces* file entry where the last two server addresses specified are in IPv6 format:

```
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
```

Character set support

Open Server and SDK support the character sets supported by Adaptive Server.

Upgrading to the new version

To upgrade Open Server applications (srvlib):

- For statically linked applications, perform a complete rebuild of the applications with the new version of software. Recompile and relink the applications with the new header files and libraries.
- For dynamically linked applications, recompile and relink the SDK libraries that have changed to include “syb” in library names.

Note If you have made any changes to the application files, you must recompile them.

Ensure that the runtime libraries are for the same major release as the version used to build the application.

New Features for ESD #11

ESD #11 includes the enhancement on Adaptive Server ADO.NET Data Provider support for connection pooling when a command time-out occurs.

Enhanced Adaptive Server ADO.NET Data Provider support for connection pooling

During a command time-out, the command is cancelled when a response is not received within the time-out. After the cancel is sent to the server, the driver may possibly lose the connection's state. To fill in this gap, the Adaptive Server ADO.NET Data Provider support for connection pooling is enhanced.

The Adaptive Server ADO.NET Data Provider now closes the real connection whenever a command time-out occurs and marks the client's connection as closed. If the application wants to continue using the connection, it reopens the connection, goes to the pool, and if necessary establishes a new connection to the server.

New Features for ESD #10

New features in ESD #10 include Adaptive Server ADO.NET Data Provider support for SQL Server Integration Services (SSIS) and a Kerberos driver library for Microsoft Windows x86-64 64-bit.

Appendix A, “Open Server Error Messages,” in *Open Server Server-Library/C Reference Manual* has also been added.

Adaptive Server ADO.NET Data Provider support for SSIS

Adaptive Server ADO.NET Data Provider can be integrated into SQL Server Integration Services (SSIS), allowing for native access to ADO.NET Data Provider functions.

With the integration, you can use Adaptive Server as an:

- ADO.NET Connection Manager
- ADO.NET Source data flow component

- ADO.NET Destination data flow component

The enhanced Adaptive Server ADO.NET Data Provider supports SSIS 2008, 32-bit. The SSIS support relies on the DDEX Provider for Adaptive Server which must be installed before SSIS can be used.

❖ Setting up the Adaptive Server connection

Before connecting to the Adaptive Server:

- Install the DDEX Provider for Adaptive Server.
- Add the driver to the global assembly cache (GAC) if you have not yet installed SDK:

  ```
  AseGacUtility -i Sybase.AdoNet2.AseClient.dll
  ```

  Afterwards, run *AdoNetRegistrar*.

1 On the Data Flow tab, right click on the ADO NET Source/Destination component you want to configure and select Edit.

2 Click the New button located next to Connection Manager.

3 In the Configure ADO.NET Connection Manager window, click New.

4 Select Sybase Adaptive Server Enterprise Data Provider from the listed providers.

5 Enter the appropriate connection properties.
To use Adaptive Server ADO.NET with SSIS, set QuotedIdentifier to 1.

6 Click OK.

Note By default, an ADO.NET Destination component batches the insert commands it performs. Currently, performing simple insert commands are faster than performing batch uploads to Adaptive Server through SSIS. To set the Destination component to perform simple insert commands, set the BatchSize property to 1.

Cybersafe Kerberos driver on 64-bit Microsoft Windows

Open Client and Open Server includes *libsybskrb64.dll*, which is a 64-bit Cybersafe Trustbroker Kerberos driver library for Microsoft Windows x86-64 64-bit. *libsybskrb64.dll* is located in *%SYBASE%\%SYBASE_OCS%\dll*; its behavior is similar to the 32-bit CyberSafe TrustBroker Kerberos driver library *libsybskrb.dll*.

New Features for ESD #9

New features in ESD #9 include Open Client and Open Server 64-bit components, a LAN Manager driver for 64-bit Windows, a new jConnect connection property, and ADO.NET Data Provider support for .NET Framework 4 and Bulk-Library.

In addition, Open Client and Open Server support for Netscape LDAP and the OpenLDAP library *libldap_r-2.2.so7* has been discontinued.

isql64 and *bcp64* for 64-bit products on UNIX and Windows platforms

isql64 and bcp64—64-bit versions of isql and bcp—are available on UNIX and Windows platforms that Open Client and Open Server supports.

Prior to Open Server and SDK 15.5 ESD #9, only 64-bit isql.exe and bcp.exe are available on 64-bit Windows. If you have a script that references isql.exe or bcp.exe, and you intend to use the 64-bit version, you must change the reference in the script to isql64.exe or bcp64.exe.

New platform for ESQL/COBOL 64-bit

ESQL/COBOL 64-bit is now available on Microsoft Windows x86-64 64-bit.

LAN Manager driver for Microsoft Windows 64-bit

Open Client and Open Server includes *libsybsmssp64.dll*, which is a 64-bit LAN Manager driver for Microsoft Windows x86-64 64-bit. *libsybsmssp64.dll* is located in *%SYBASE%\%SYBASE_OCS%\dll*; its behavior is similar to the 32-bit driver *libsybsmss.dll*.

JAVA_CHARSET_MAPPING connection property

Use the jConnect connection property JAVA_CHARSET_MAPPING to supersede the default Adaptive Server character set mapping.

Example 1 Maps the server character set cp949 to ms949:

```
props.put("CHARSET", "cp949"); /* Server character set */
props.put("JAVA_CHARSET_MAPPING", "ms949"); /* Java character set mapping */
```

Adaptive Server character set mapping reference

Most of the Adaptive Server character sets have the same name as the Java character sets that they are mapped to. Those that are mapped to a Java character set with a different name are listed below.

Server Character Set	Equivalent Java Character Set
cp932	MS932
cp936	GBK
deckanji	EUC_JP
euccns	EUC_TW
eucgb	EUC_CN
eucjis	EUC_JP
eucksc	EUC_KR
iso15	ISO8859_15_FDIS
iso88592	ISO8859_2
iso88595	ISO8859_5
iso88596	ISO8859_6
iso88597	ISO8859_7
iso88598	ISO8859_8
iso88599	ISO8859_9
iso_1	ISO8859_1
koi8	KOI8_R
mac	MacRoman

Server Character Set	Equivalent Java Character Set
mac_cyr	MacCyrillic
mac_ee	MacCentralEurope
roman8	hp-roman8
sjis	MS932
tis620	MS874

.NET Framework 4 support

Adaptive Server ADO.NET Data Provider has been enhanced to support the expanded Entity Data Model (EDM) canonical functions defined in Entity Framework 4.0, expose Adaptive Server functions in Visual Studio Language-Integrated Query (LINQ), and allow model-first development in Entity Designer.

If you have not installed SDK, register or reference *Sybase.AdoNet4.AseClient.dll* using AseGacUtility4 before connecting to an Adaptive Server.

Note The .NET Framework 4 support has been tested and certified on Adaptive Server 15.5 ESD #2 and later.

For complete information about .NET Framework 4 features, see Microsoft Developer Network at http://msdn.microsoft.com.

Support for EDM canonical functions

Adaptive Server ADO.NET Data Provider supports the new EDM canonical functions defined in Entity Framework 4.0. However, AddNanoseconds(), CreateDateTimeOffset(), DiffNanoseconds(), and GetTotalOffsetMinutes() are not available in Adaptive Server.

Exposed Adaptive Server functions

Use the AseFunction class methods to call Adaptive Server-specific functions in LINQ.

Table 5: Adaptive Server Functions Available in LINQ

Adaptive Server Function	AseFunction Class Equivalent
acos	Acos
ascii	Ascii
asin	Asin
atan	Atan
atn2	Atn2
biginttohex	BigIntToHex
bintostr	BinaryToString
char	Char
char_length	CharLength
cos	Cos
cot	Cot
current_bigdatetime	CurrentBigDatetime
current_bigtime	CurrentBigTime
current_date	CurrentDate
current_time	CurrentTime
datalength	DataLength
datename	DateName
degrees	Degrees
difference	Difference
exp	Exp
hextobigint	HexToBigInt
hextoint	HexToInt
inttohex	IntToHex
isdate	IsDate
isnumeric	IsNumeric
log	Log
log10	Log10
pi	Pi
radians	Radians
rand	Rand
rand2	Rand2
replicate	Replicate
sign	Sign
sin	Sin
soundex	SoundCode
space	Space

Adaptive Server Function	AseFunction Class Equivalent
sqrt	SquareRoot
square	Square
str	StringConvert
strtobin	StringToBinary
stuff	Stuff
tan	Tan
to_unichar	ToUnicodeChar
uhighsurr	UnicodeHighSurrogate
ulowsurr	UnicodeLowSurrogate
uscalar	UnicodeScalar
user_name	UserName

Model-first development

Model-first development lets you generate DDLs from user-defined EDMs.

❖ **Generating DDLs from EDMs**

1. Create a project in Visual Studio.
2. Create an EDM, and edit it to define tables and associations.
3. From the Properties view, select SSDLToASE155.tt(VS) as the DDL generation template.
4. Right-click the Entity Designer window and select Generate Database from Model.
5. In the Generate Database wizard, choose a connection and click Next.

   A DDL script is generated.
6. Click Finish.

   The DDL script is added to the project and the DDL is created.

Bulk-load support in Adaptive Server ADO.NET Data Provider

Adaptive Server ADO.NET Data Provider supports bulk-load interface for fast insertions of large sets of rows to Adaptive Server. Setting the ENABLEBULKLOAD connection property allows ASEBulkCopy to invoke the bulk-load interface. Two types of bulk loading are supported:

- Array Inserts – use this type of bulk-loading within a single or multistatement transaction.
- Bulk Copy – this is supported only in single-statement transactions, and you must ensure that the select into/bulkcopy option on Adaptive Server is turned on.

  If the target table meets the criteria for high-speed version of bulk copy, Adaptive Server inserts the rows using this version of bulk copy.

Note Using the bulk-copy mode with the select into/bulkcopy option enabled affects the recoverability of the database. After the bulk copy operation is complete, the system administrator must dump the database to ensure its future recoverability.

Table 6: Bulk-Load Option Usage

Use Case	Additional Consideration	Bulk-Load Option to Use	Note
Insertion of single or small number of rows.		None	
Insertion of large batch of rows.	The batch is part of a multistatement transaction.	Array Inserts	Rows are inserted faster than when bulk load is disabled.
	You cannot enable the Adaptive Server select into or bulkcopy option because of recoverability considerations.	Array Inserts	Rows are inserted faster than when bulk load is disabled.
	The batch is a single transaction and the Adaptive Server select into/bulkcopy option is enabled.	Bulk Copy	Adaptive Server can use high-speed bulk copy, which is faster than array inserts. The performance of Bulk Copy is still slightly faster than Array Inserts even if high-speed bulk copy is not used.

See the *Adaptive Server Enterprise Utility Guide* for information about the implications of enabling select into/bulkcopy and the conditions under which high-speed or logged bulk copy is used.

ENABLEBULKLOAD connection property

Enable or disable bulk-load support using the ENABLEBULKLOAD connection property:

- 0 – off mode, the default value.

- 1 – enables bulk-load using array insert.
- 2 – enables bulk-load using the bulk copy interface.

❖ Enabling bulk load using the ADO.NET connection string

1. Use SQLDriverConnect to specify a connection string.
2. Set the ENABLEBULKLOAD connection string property to 0, 1, or 2, as appropriate. For example:

```
Data Source=server1;port=port1;UID=sa;PWD=;
Driver=AdaptiveServerEnterprise;
ENABLEBULKLOAD=1;
```

Performance considerations

Although this feature does not require special configuration on the server, a larger page size and network packet size significantly improves performance. Depending on the client memory, using larger batches also improves performance.

Supported *ASEBulkCopy* options

ASEBulkCopy Options	Supported Bulk-load Mode
Default	Array Inserts, Bulk Copy, Off
KeepIdentity	Array Inserts, Bulk Copy, Off
KeepNulls	Array Inserts, Bulk Copy, Off
UseInternalTransaction	Array Inserts, Bulk Copy, Off
CheckConstraints	Off
FireTriggers	Off
TableLock	Not supported

Limitations

- Computed and encrypted columns are not supported. Also, triggers are ignored on tables selected for bulk-loading.
- The CheckConstraints, FireTriggers, and TableLock AseBulkCopy options are supported only as default values; these values are not supported when bulk-loading is disabled.

New Features for ESD #8

New features in ESD #8 include an enhanced DDEX Provider for Adaptive Server and ESQL/COBOL support for four UNIX platforms.

New ESQL/COBOL platforms

For ESQL/COBOL 32-bit:

- Sun Solaris x86 32-bit
- Sun Solaris x86-64 64-bit
- Linux x86-64 64-bit

For ESQL/COBOL 64-bit:

- Sun Solaris x86-64 64-bit
- Linux x86-64 64-bit
- Linux POWER 64-bit

Enhanced DDEX Provider for Adaptive Server

You can now use Entity Framework to create data models with Adaptive Server databases. This enhancement is supported in Visual Studio 2008 SP1 and Visual Studio 2010.

❖ Creating an Adaptive Server Entity Data Model using Entity Framework

Create data access classes using the Microsoft Entity Framework.

You need a valid connection to an active Adaptive Server to perform this task.

1. Create a new application project.
2. In the Solution Explorer window, right-click the project and select Add | New Item.
3. Select Data as the category and ADO.NET Entity Data Model as the template.
4. Enter a name for the entity data model and click Add.
   The Data Model wizard launches.
5. Choose Generate from database. Click Next.
6. Select an existing Adaptive Server connection or click New Connection.
   If you chose New Connection, enter the name of the connection settings. Click Next.
7. Select an Adaptive Server object. Click Finish.

Use the Entity Designer to modify the model. Data access classes are automatically generated by the Entity Framework when you save the model.

New features for ESD #7

ESD #7 includes the Data Designer Extensibility (DDEX) Provider for Adaptive Server and a new property for Adaptive Server ODBC Driver.

Appendix A, “Precompiler Warning and Error Messages,” in both the *Open Client Embedded SQL/C Programmers Guide* and *Open Client Embedded SQL/COBOL Programmers Guide* has also been updated.

DDEX Provider for Adaptive Server

The DDEX Provider for Adaptive Server enables Visual Studio components, such as Server Explorer, to interact with Adaptive Server and its objects through the Adaptive Server ADO.NET Data Provider. With the DDEX Provider for Adaptive Server, you can:

- Connect and log in to Adaptive Server from Visual Studio.
- List Adaptive Server objects as hierarchy nodes in Visual Studio Server Explorer.
- Drag and drop Adaptive Server tables and views from Server Explorer onto data designers.

The DDEX Provider for Adaptive Server is compatible with Visual Studio 2005, 2008, and 2010.

Note The DDEX Provider for Adaptive Server has been tested and certified on Adaptive Server 15.5 ESD #2 and later.

❖ Connecting to Adaptive Server

Add a database connection to Adaptive Server using the Visual Studio Server Explorer view.

Before connecting to the Adaptive Server:

- Add the driver to the global assembly cache (GAC) if you have not yet installed SDK:
  - For Visual Studio 2005 and 2008, enter:

    ```
    AseGacUtility -i Sybase.AdoNet2.AseClient.dll
    ```

  - For Visual Studio 2010, enter:

    ```
    AseGacUtility4 -i Sybase.AdoNet2.AseClient.dll
    ```

- If your application references a version of the Adaptive Server ADO.NET Data Provider that is earlier than 15.5 ESD #7, run:
  - For Visual Studio 2005 and 2008:

    ```
    AseGacUtility -i
       policy.2.155.Sybase.AdoNet2.AseClient.dll
    ```

  - For Visual Studio 2010:

    ```
    AseGacUtility4 -i
       policy.2.155.Sybase.AdoNet2.AseClient.dll
    ```

- Run *AdoNetRegistrar* for Visual Studio 2005 and 2008, or *AdoNetRegistrar4* for Visual Studio 2010.

1. In the Server Explorer view, right-click Data Connections.
2. Select Add Connection.
3. Select Sybase ASE Database as the data source and .NET Framework Data Provider for Sybase ASE as the data provider.
4. Enter the Adaptive Server connection properties.

❖ Viewing Adaptive Server objects

View Adaptive Server objects and their related information in Visual Studio Server Explorer.

You need a valid connection to an active Adaptive Server to perform this task.

1. Connect to Adaptive Server.
2. Expand the Adaptive Server object to explore.
3. Click an Adaptive Server object to view its property information.

Supported Adaptive Server objects

The DDEX Provider for Adaptive Server exposes Adaptive Server objects, which can be viewed and accessed in the Visual Studio Server Explorer.

Table 7: Supported Adaptive Server Objects

Database Object	Properties
Check Constraint	Name, Catalog, Schema, Table, Constraint Column, Constraint Type
Column of Web Service as Table	Name, Catalog, Schema, Table, Default Value, Is Nullable, Ordinal, Length, Precision, Scale, System Type
Database	Name, Create Date, Dump Date
Datatype	Name
Default	Name, Catalog, Schema
Foreign Key	Name, Catalog, Schema, Table, Referenced Table, Referenced Table Catalog, Referenced Table Column, Referenced Table Schema
Index	Name, Catalog, Schema, Table, Referred Column
Instead-of Trigger	Name, Catalog, Schema, View
Primary Key	Name, Catalog, Schema, Table, Referred Column
Proxy Table	Name, Catalog, Schema
Proxy Table Column	Name, Catalog, Schema, Table, Default Value, Is Nullable, Ordinal, Length, Precision, Scale, System Type
Rule	Name, Catalog, Schema
Schema	Name
Stored Procedure	Name, Catalog, Schema
Stored Procedure Parameter	Name, Catalog, Schema, Stored Procedure, Is Output, Ordinal, Length, Precision, Scale, System Type
Table	Name, Catalog, Schema, Type
Table Column	Name, Catalog, Schema, Table, Default Value, Is Nullable, Ordinal, Length, Precision, Scale, System Type
Trigger	Name, Catalog, Schema, Table
Unique Constraint	Name, Catalog, Schema, Table, Referred Column
User-defined Function	Name, Catalog, Schema
User-defined Function Parameter	Name, Catalog, Schema, User-defined Function, Is Output, Ordinal, Length, Precision, Scale, System Type
User-defined Type	Name, Catalog, Schema
View	Name, Catalog, Schema
View Column	Name, Catalog, Schema, View, Default Value, Is Nullable, Ordinal, Length, Precision, Scale, System Type
Web Services as Table	Name, Catalog, Schema, Method, Timeout, WSDL URI

New CHANGEBIGINTDEFAULT property

CHANGEBIGINTDEFAULT specifies the default ODBC mapping used for Adaptive Server bigint columns.

When you use a column of type bigint as an identifier in an Adaptive Server table (for example, as an identity or primary key), and applications such as Microsoft Access access the table through Adaptive Server ODBC Driver, the values of such column may appear as "#deleted", and prevent further operations on the table. As a workaround, set CHANGEBIGINTDEFAULT to 1.

CHANGEBIGINTDEFAULT values:

- 0 – the default value, binds SQL_C_DEFAULT to SQL_C_BIGINT.
- 1 – binds SQL_C_DEFAULT to SQL_C_CHAR. Use this setting when you want to access Adaptive Server tables with bigint identifiers from applications such as Microsoft Access or Microsoft Excel.
- 2 – binds SQL_C_DEFAULT to SQL_C_WCHAR.

New features for ESD #5

New features in ESD #5 include Adaptive Server ODBC Driver support for five UNIX platforms, and a new connection property for Adaptive Server ADO.NET Data Provider.

Extended Adaptive Server ODBC Driver support

The Adaptive Server ODBC Driver now supports:

- HP HP-UX Itanium 64-bit
- IBM AIX POWER 64-bit
- Linux POWER 64-bit
- Sun Solaris SPARC 64-bit
- Sun Solaris x86-64 64-bit

However, these ODBC features are not available on the new platforms:

- Kerberos authentication
- LDAP as a directory service
- Asynchronous execution for ODBC

For information about configuring and using Adaptive Server ODBC Driver on the new platforms, see the *Adaptive Server Enterprise ODBC Driver Users Guide*.

New SECURECONNECTIONSTRING property

SECURECONNECTIONSTRING is an Adaptive Server ADO.NET Data Provider connection property that removes the password property from the connection string. This ensures that the password is not exposed when the connection string is accessed using AseConnection.ConnectionString.

Values:

- 0 – the default value; Adaptive Server ADO.NET Data Provider keeps the password in the connection string.
- 1 – Adaptive Server ADO.NET Data Provider removes the password from the connection string.

New features for ESD #4

New features in ESD #4 include bulk-load support in Adaptive Server ODBC Driver, management of Adaptive Server ADO.NET Data Provider tracing, and a new jConnect property. This section also clarifies a bulk-load limitation described in the *jConnect for JDBC Programmers Reference*.

Bulk-load support in Adaptive Server ODBC Driver

The Adaptive Server ODBC Driver now supports bulk-load interface for fast insertions of large sets of rows to Adaptive Server. This interface is invoked when SQLBulkOperations is used with the SQL_ADD option and the ENABLEBULKLOAD connection property is set. Two types of bulk loading are supported:

- Array Inserts – you can use this type of bulk-loading within a single or multistatement transaction; the database connection can be set to autocommit off.

- Bulk Copy – this is supported only in single statement transactions, and you must ensure that:
  - The database connection is set to autocommit on.
  - The select into/bulkcopy option on Adaptive Server is turned on.

  If the target table meets the criteria for high-speed version of bulk copy, Adaptive Server inserts the rows using this version of bulk copy.

Note Using the bulk-copy mode with the select into/bulkcopy option enabled affects the recoverability of the database. After the bulk copy operation is complete, the system administrator must dump the database to ensure its future recoverability.

The following table guides you on what bulk-load option to use.

Table 8: Bulk-load option usage

Use case	Additional consideration	Bulk-load option to use	Note
Insertion of single or small number of rows.		None	
Insertion of large batch of rows.	The batch is part of a multistatement transaction.	Array Inserts	Rows are inserted faster than when bulk load is disabled.
	You cannot enable the Adaptive Server select into or bulkcopy option because of recoverability considerations.	Array Inserts	Rows are inserted faster than when bulk load is disabled.
	The batch is a single transaction and the Adaptive Server select into/bulkcopy option is enabled.	Bulk Copy	Adaptive Server can use high-speed bulk copy, which is faster than array inserts. The performance of Bulk Copy is still slightly faster than Array Inserts even if high-speed bulk copy is not used.

See the *Adaptive Server Enterprise Utility Guide* for information about the implications of enabling select into/bulkcopy and the conditions under which high-speed or logged bulk copy is used.

ENABLEBULKLOAD connection property

Enable or disable bulk-load support using the ENABLEBULKLOAD connection property:

- 0 – the default value, which disables bulk load.

- 1 – enables bulk-load using array insert.
- 2 – enables bulk-load using the bulk copy interface.

Performance considerations

Although this feature does not require special configuration on the server, a larger page size and network packet size significantly improves performance. Depending on the client memory, using larger batches also improves performance.

Limitations

Computed and encrypted columns are not supported. Also, triggers are ignored on tables selected for bulk loading.

Enabling bulk load

❖ Enabling bulk load using the ODBC Data Source Administrator user interface

1. Open the Data Source Name (DSN) Configure window from the ODBC Data Source Administrator.
2. Select the Advanced tab.
3. Select the appropriate option under "Enable Bulk Load."

   The default value of ENABLEBULKLOAD is 0, which means insert commands are used.

❖ Enabling bulk load using the ODBC connection string

1. Use SQLDriverConnect to specify a connection string.
2. Set the ENABLEBULKLOAD connection string property to 0, 1, or 2, as appropriate. For example:

```
Driver=AdaptiveServerEnterprise;server=server1;
port=port1;UID=sa;PWD=;ENABLEBULKLOAD=1;
```

Bulk-load support in jConnect for JDBC

You can bulk load unsigned, bigint, and unitext datatypes in jConnect for JDBC versions 6.05 and 7.0; the *jConnect for JDBC Programmers Reference* incorrectly states that you cannot.

Managing Adaptive Server ADO.NET Data Provider tracing

The ENABLETRACING connection property lets you enable or disable tracing of Adaptive Server ADO.NET Data Provider activities, which has always been enabled in versions earlier than ESD #4. ENABLETRACING is disabled by default to allow for better performance during normal execution where tracing is not needed. When this property is disabled, the TraceEnter and TraceExit events are not triggered, and tracing events are not executed.

ENABLETRACING is supported in Adaptive Server ADO.NET Data Provider versions 1.155 and 2.155. You can configure ENABLETRACING in the connection string using these values:

- True – triggers the TraceEnter and TraceExit events.
- False – the default value; Adaptive Server ADO.NET Data Provider ignores the TraceEnter and TraceExit events.

For information about the TraceEnter and TraceExit events, see the *Adaptive Server Enterprise ADO.NET Data Provider Users Guide*.

New jConnect property DELETE_WARNINGS_FROM_EXCEPTION_CHAIN

DELETE_WARNINGS_FROM_EXCEPTION_CHAIN is a jConnect connection property that specifies whether or not to remove SQLWarning from the SQLException chain.

Values:

- True – the default value for jConnect 7.0 and later; jConnect removes SQLWarning objects from the SQLException chain.
- False – the default value for jConnect 6.05; jConnect keeps the SQLWarning objects in the SQLException chain.

New feature for ESD #3

These connection properties have been added or modified to improve jConnect performance:

- DYNAMIC_PREPARE – when set to true, DYNAMIC_PREPARE enables you to precompile dynamic SQL statements. The default value of this property has been changed to true, because, in most cases, this setting improves jConnect performance.

  However, you cannot use the prepare command to precompile all SQL statements. For such SQL statements, Sybase recommends that you do not use PreparedStatement, and to set DYNAMIC_PREPARE to false to avoid unnecessary precompilation attempts. For more information about DYNAMIC_PREPARE, and the advantages and disadvantages of using prepared statements, see the *jConnect for JDBC Programmers Reference*.

- IGNORE_WARNINGS – specifies whether or not to check for and generate warning messages. Currently, this property checks only for warning regarding the loss of precision when storing timestamp values into Adaptive Server date and time datatypes, which have lower precision than the Java timestamp.

  Valid values:

  - True – jConnect does not check for and generate warning messages, thus improving performance.
  - False – the default value, which directs jConnect to check and generate warning messages.

- OPTIMIZE_FOR_PERFORMANCE – specifies whether or not to enable jConnect performance enhancing properties. Currently, this property controls only the IGNORE_WARNINGS property.

  Valid values:

  - True – jConnect runs in enhanced performance mode.
  - False – the default value, which means that jConnect runs in normal mode.

Note Before setting IGNORE_WARNINGS or OPTIMIZE_FOR_PERFORMANCE to true, thoroughly test the impact of such a configuration on your application.

Open Client 15.5 and Open Server 15.5 features

This section describes the new features in Open Client 15.5 and Open Server 15.5.

Microsecond granularity for time data

The Open Client and Open Server CS_BIGDATETIME and CS_BIGTIME datatypes provide microsecond-level precision for time data. These datatypes are intended to hold 8-byte binary values.

These datatypes function similarly to the respective CS_DATETIME and CS_TIME datatypes: CS_BIGDATETIME can be used anywhere that CS_DATETIME can be used, and CS_BIGTIME can be used anywhere that CS_TIME can be used. All Open Client and Open Server routines that can be applied to the CS_DATETIME and CS_TIME datatypes can also be applied to the CS_BIGDATETIME and CS_BIGTIME datatypes.

- CS_BIGDATETIME corresponds to the Adaptive Server bigdatetime datatype, and contains the number of microseconds that have passed since January 1, 0000 00:00:00.000000. The range of legal CS_BIGDATETIME values is from January 1, 0001 00:00:00.000000 to December 31, 9999 23:59:59.999999.

  Note January 1, 0000 00:00:00.000000 is the base starting value from which microseconds are counted. Any value earlier than January 1, 0001 00:00:00.000000 is invalid.

  The definition of CS_BIGDATETIME can be found in *cstypes.h*:

  ```
  typedef CS_UBIGINT CS_BIGDATETIME;
  ```

- CS_BIGTIME corresponds to the Adaptive Server bigtime datatype and indicates the number of microseconds that have passed since the beginning of the day. The range of legal CS_BIGTIME values is from 00:00:00.000000 to 23:59:59.999999. The definition of CS_BIGTIME can be found in *cstypes.h*:

  ```
  typedef CS_UBIGINT CS_BIGTIME;
  ```

- CS_BIGDATETIME and CS_BIGTIME data is presented to the client in the native-byte order (endianness) of the underlying client platform. Any necessary byte-swapping is performed at the server before the data is sent to the client, or after the data is received from the client.

For information about Open Client and Open Server datatypes, see the *Open Client Client-Library/C Reference Manual* and *Open Server Sever-Library/C Reference Manual*, respectively.

Capabilities

The ct_capability routine uses these parameters to support CS_BIGDATETIME and CS_BIGTIME:

- CS_DATA_BIGDATETIME and CS_DATA_BIGTIME are request capabilities sent to the server to determine whether CS_BIGDATETIME and CS_BIGTIME are supported.
- CS_DATA_NOBIGDATETIME and CS_DATA_NOBIGTIME are response capabilities sent from the server indicating that CS_BIGDATETIME and CS_BIGTIME should not be used.

For information about request and response capabilities, see the *Open Client Client-Library/C Reference Manual* and *Open Server Sever-Library/C Reference Manual*, respectively.

Constants

The following constants are provided for the use of CS_BIGDATETIME and CS_BIGTIME:

- CS_BIGDATETIME_TYPE
- CS_BIGTIME_TYPE

For information about Open Client and Open Server constants, see the *Open Client Client-Library/C Reference Manual* and *Open Server Sever-Library/C Reference Manual*, respectively.

Mapping

The Adaptive Server bigdatetime and bigtime datatypes are respectively mapped to the client CS_BIGDATETIME and CS_BIGTIME datatypes with their formats described in a client-side CS_DATAFMT structure. The *datatype* setting for the CS_DATAFMT structure must be set to CS_BIGDATETIME_TYPE or CS_BIGTIME_TYPE.

Datatype conversions

Table 9 indicates the datatype conversions that are supported for CS_BIGDATETIME and CS_BIGTIME. The source datatypes are listed in the leftmost column, and the destination datatypes are listed in the top row of the chart. "•" indicates that the conversion is supported. A blank space indicates that the conversion is not supported.

Table 9: Datatype conversions

Open Client datatypes	CS_BINARY	CS_LNGBINARY	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	•	•	•	•	•	•	•				•	•	•	•	•	•	•	•	•			•	•				•	
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	•	•	•	•	•	•	•	•	•	•																	•	•

Open Client datatypes	CS_BINARY	CS_LNGBINARY	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_DATE			•		•	•	•	•	•	•												•	•	•	•			
CS_TIME			•		•	•	•	•	•	•												•	•	•		•		•
CS_BIGINT	•	•	•	•	•	•	•				•	•	•	•	•	•	•	•	•			•	•	•				
CS_BIGTIME			•		•	•	•	•	•	•												•		•		•		•

Conversion between CS_BIGDATETIME and CS_BIGTIME and the following datatypes is not supported:

- CS_UBIGINT
- CS_UINT
- CS_USMALLINT
- CS_XML

Conversion between CS_BIGDATETIME and CS_BIGTIME and the CS_UNITEXT datatype is supported.

All conversions to and from CS_BIGDATETIME and CS_BIGTIME are handled in the same way as existing datetime and time conversions.

Conversion to CS_CHAR

The cs_dt_info routine sets or retrieves language-specific date, time, or datetime information. The date conversion format, which you can set or retrieve by calling cs_dt_info with the *type* parameter value of CS_DT_CONVFMT, describes the format of the result when a CS_BIGDATETIME or CS_BIGTIME value is converted to a character-based datatype.

*Table 10: Values for *buffer when type is CS_DT_CONVFMT (cs_dt_info)*

Symbolic value	CS_CHAR converted from CS_BIGDATETIME, for example: Aug 24 2009 5:36PM	CS_CHAR converted from CS_DATE, for example: Aug 24 2009	CS_CHAR converted from CS_BIGTIME, for example: 5:36PM
CS_DATES_HMSUSA, or 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, or 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`

Updates to *cs_dt_crack*

In addition to the *datetype* parameter values used for indicating CS_DATE, CS_TIME, CS_DATETIME, and CS_DATETIME4 datatypes, you can use these values to indicate CS_BIGDATETIME and CS_BIGTIME datatypes when using the cs_dt_crack CS-Library routine:

Value of datetype	Indicates
CS_BIGDATETIME_TYPE	CS_BIGDATETIME **dateval*
CS_BIGTIME_TYPE	CS_BIGTIME **timeval*

Two fields in the CS_DATEREC structure accommodate microsecond precision:

- *datesecfrac* is a CS_INT field holding the number of second fractions.
- *datesecprec* is a CS_INT field holding the precision. For CS_BIGDATETIME and CS_BIGTIME, this field is always 10^6.

These fields are used only with datetime datatypes having a level of precision greater than milliseconds.

A CS_DATEREC is defined as:

```
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     datesecond;     /* second              */
     CS_INT     datemsecond;    /* millisecond         */
     CS_INT     datetzone;      /* timezone            */
     CS_INT     datesecfrac;    /* second fractions    */
     CS_INT     datesecprec;    /* precision           */

 } CS_DATEREC;
```

For information about cs_dt_crack, see the *Open Client and Open Server Common Libraries Reference Manual*.

bcp file storage types

If bcp is invoked and no value is supplied for the -c, -f, or -n parameters, a bcp prompt requests the file storage type. The file storage type can be any valid Adaptive Server datatype. Storage types for the bigdatetime and bigtime Adaptive Server datatypes are specified as:

Storage type	Table datatype
A	bigdatetime
B	bigtime

For information about bcp, see the *Open Client and Open Server Programmers Supplement* for your platform.

bcp **host file datatype storage format**

The following host file datatypes can be specified for a bcp format file using the bigdatetime or bigtime datatypes.

Table 11: Host file datatype storage formats

Storage format	Adaptive Server datatype
SYBBIGDATETIME	bigdatetime
SYBBIGTIME	bigtime

ct_send_data extension

An Open Client application normally sends text and image data to an Open Server application using a writetext stream. To improve performance, an Open Client application may instead send text and image data directly to the bulk handler of the Open Server application.

For more information about the ct_send_data command, see the *Open Client Client-Library/C Reference Manual*.

Processing text and image data

To update a text or image column, a client application typically calls the ct_command routine to initiate a send-data command. The client then calls the ct_data_info command to retrieve CS_IODESC and determine the appropriate SQL command to generate (update or writetext) in a subsequent call to the ct_send_data routine. One or more additional calls to ct_send_data follow for the purpose of sending the text or image data. The client concludes with calls to the ct_send and ct_results routines. A language event occurs at the server, which processes the SQL command. A bulk event follows, causing the server to read the specified amount of text or image data.

To simplify this process and potentially improve performance, a client can suppress the generation of the SQL command (update or writetext) and send data directly to the server bulk handler. The client must initiate the send-data command by calling the ct_command routine with the *type* parameter set to CS_SEND_DATA_NOCMD. The client application can then use send-data commands to send only text or image data to the server bulk handler. When a bulk event occurs at the server, a 4-byte field is sent indicating the total number of bytes to be sent, followed by the text or image data. The bulk handler reads the total number of bytes expected using srv_text_info and the data using srv_get_data.

sp_mda

The server must define a stored procedure, sp_mda, to indicate whether or not it supports the ct_send_data routine sending only text or image data without a SQL command. sp_mda is called only if the CS_SENDDATA_NOCMD connection property is set prior to logging in to the server.

If the server can receive text or image data without a SQL command, sp_mda returns:

Parameter	Value
mdinfo	"SENDDATA_NOCMD"
querytype	2
query	senddata no cmd

Note Adaptive Server cannot receive image or text data without a SQL command.

Open Server dynamic listeners

Open Server dynamic listeners enable Open Server applications to start a new listener even after the srv_run() call. This allows Open Server applications to start with minimal entries in its interface file and to restart listeners. The functionality documented in this section supports dynamic listeners.

Events

Two Open Server events, SRV_LISTEN_PREBIND and SRV_LISTEN_POSTBIND, support dynamic listeners.

Table 12: Open Server events

Event	Description	Argument to handler	Default event handler
SRV_LISTEN_PREBIND	Allows additional configuration of the listener identified by the specified SRV_PROC control structure. For example, you can specify an alternate SSL certificate. This configuration is accomplished using thread properties.	SRV_PROC*	The default handler takes no action.
SRV_LISTEN_POSTBIND	Allows the final configuration of the listener identified by the specified SRV_PROC control structure to be determined. For example, the address to which the listener was bound can be determined. Configuration is determined using thread properties.	SRV_PROC*	The default handler takes no action.

Server properties

Two Open Server properties, SRV_S_NUMLISTENERS and SRV_S_MAXLISTENERS, support dynamic listeners.

Table 13: Server properties

Property name	Definition	Notes
SRV_S_MAXLISTENERS	Limits the maximum number of listener threads.	The default value is CS_MAX_NOMAX. This value is used only when trying to create a new listener thread. Setting this property to a value smaller than the current value of SRV_S_NUMLISTENERS does not cause any listeners to fail.

Property name	Definition	Notes
SRV_S_NUMLISTENERS	Returns the number of SRV_PROC control structures involved in listening for client connections.	This is a retrieve-only property.

Thread type

Listener threads will have a thread type of SRV_TLISTENER.

Thread properties

Use the SRV_T_LISTENADDR, SRV_T_LOCALID, and SRV_T_REMOTEADDR thread properties to configure threads to support dynamic listeners.

Table 14: Thread properties

Property name	Definition	Notes
SRV_T_LISTENADDR	Returns the address of the listener identified by the specified SRV_PROC control structure. If SRV_PROC is a listener, this property returns the address on which the listener accepts connections.	This property requires a pointer to a CS_TRANADDR structure as an argument. This is a retrieve-only property.
SRV_T_LOCALID	Specifies the SSL certificate to use for a listener.	This property allows a listener to use an SSL certificate other than the global server-level SSL certificate. This is a set-only property.
SRV_T_REMOTEADDR	Returns the address of a SRV_PROC peer. This property value is valid only for a client SRV_PROC.	This property requires a pointer to a CS_TRANADDR structure as an argument. This is a retrieve-only property.

Set the SRV_T_CIPHER_SUITE and SRV_T_SSL_VERSION thread properties from within the SRV_LISTEN_PREBIND Open Server event.

Starting a dynamic listener

You can start a dynamic listener by calling the srv_spawn routine using these parameter values:

- sppp – the pointer to the thread structure pointer should be null.

- *stacksize* – specify the stack size as CS_UNUSED.
- *funcp* – the entry-point function pointer should be SRV_C_START_LISTENER.
- *argp* – point this parameter to a CS_TRANADDR structure.
- *priority* – specify as CS_UNUSED.

For information about Open Server events, properties, and threads, see the *Open Server Server-Library/C Reference Manual*.

Performance enhancements

The following performance enhancements have been implemented:

- Improved network I/O operations for Client-Library and Bulk-Library routines, especially ct_fetch(), blk_rowxfer(), and blk_rowxfer_mult().
- Improved compiler optimization settings in Sun Solaris and Linux.

CS_RES_NOXNLMETADATA response capability

The CS_RES_NOXNLMETADATA response capability improves server and client application performance by optimizing the information type and structure that server applications send to client applications. You can use CS_RES_NOXNLMETADATA to inform your server not to send metadata such as Column Label, Catalog Name, Schema Name, and Table Name when this information is not needed.

By default, CS_RES_NOXNLMETADATA is on for applications that are set to version CS_VERSION_125, CS_VERSION_150, or CS_VERSION_155. For information about response capabilities, see the *Open Client Client-Library/C Reference Manual* or *Open Server Server-Library/C Reference Manual*.

FIPS-140-2-compliant password encryption

Encryption of login and remote passwords in Open Client and Open Server is accomplished with the Sybase Common Security Infrastructure (CSI). CSI 2.6 complies with the Federal Information Processing Standard (FIPS) 140-2.

To support FIPS encryption, a Certicom Security Builder shared library named *libsbgse2.so* (UNIX and Linux platforms) or *libsbgse2.dll* (Microsoft Windows platforms) is installed on platforms that do not already use the Certicom Security Builder. Also, the *sybcsi* subdirectory found in *$SYBASE/$SYBASE_OCS/lib3p* or *$SYBASE/$SYBASE_OCS/lib3p64* has been removed.

For information about Secure Sockets Layer (SSL) in Open Client and Open Server, see the *Open Client and Open Server Configuration Guide* for your platform.

SQL Server references

References to “SQL Server” in Open Client and Open Server error messages and header files have been replaced with “Server.”

SDK 15.5 features for jConnect and Adaptive Server driver and providers

This section describes the new features for these SDK 15.5 components:

- jConnect 7.0 for JDBC
- Adaptive Server ODBC Driver by Sybase
- Adaptive Server OLE DB Provider by Sybase
- Adaptive Server ADO.NET Data Provider 2.155 by Sybase

For information about jConnect or the Adaptive Server driver and providers, see the *jConnect for JDBC Programmers Reference*; *Adaptive Server Enterprise ODBC Driver by Sybase for Microsoft Windows, Linux, and Mac OS X*; *Adaptive Server Enterprise ADO.NET Data Provider for Microsoft Windows*; or *Adaptive Server Enterprise OLE DB Provider by Sybase for Microsoft Windows*.

Microsecond granularity for time data

jConnect 7.0 and the Adaptive Server ODBC Driver, OLE DB Provider, and ADO.NET Data Provider 2.155 provide microsecond-level precision for time data by supporting the SQL datatypes bigdatetime and bigtime.

bigdatetime and bigtime function similarly to and have the same data mappings as the SQL datetime and time datatypes:

- bigdatetime corresponds to the Adaptive Server bigdatetime datatype and indicates the number of microseconds that have passed since January 1, 0000 0:00:00.000000. The range of legal bigdatetime values is from January 1, 0001 00:00:00.000000 to December 31, 9999 23:59:59.999999.
- bigtime corresponds to the Adaptive Server bigtime datatype and indicates the number of microseconds that have passed since the beginning of the day. The range of legal bigtime values is from 00:00:00.000000 to 23:59:59.999999.

Usage

- When connecting to Adaptive Server 15.5, jConnect 7.0 for JDBC and the Adaptive Server ODBC Driver, OLE DB Provider, and ADO.NET Data Provider 2.155 transfer data using the bigdatetime and bigtime datatypes even if the receiving Adaptive Server columns are defined as datetime and time.

  This means that Adaptive Server may silently truncate the values from the drivers and data providers to fit Adaptive Server columns. For example, a bigtime value of 23:59:59.999999 is saved as 23:59:59.996 in an Adaptive Server column with datatype time.
- When connecting to Adaptive Server 15.0.x and earlier, jConnect 7.0 and the Adaptive Server ODBC Driver, OLE DB Provider, and ADO.NET Data Provider 2.155 transfer data using the datetime and time datatypes.
- jConnect 6.05 and Adaptive Server ODBC Driver 15.0, OLE DB Provider 15.0, and ADO.NET Data Provider 1.15 use datetime and time datatypes when connecting to Adaptive Server 15.5.

FIPS-140-2-compliant password encryption

You can encrypt login and remote passwords in Adaptive Server ODBC Driver, OLE DB Provider, and ADO.NET Data Provider 2.155 using the Sybase Common Security Infrastructure (CSI). CSI 2.6 complies with the Federal Information Processing Standard (FIPS) 140-2.

Distributed transaction management support

Adaptive Server ODBC Driver, OLE DB Provider, and ADO.NET Data Provider 2.155 support distributed transactions, which is also an Adaptive Server Cluster Edition feature. For information about distributed transactions, see the *Adaptive Server Enterprise Using Adaptive Server Distributed Transaction Management Features*.

Simplified character set conversion

In earlier releases of Adaptive Server ODBC Driver and OLE DB Data Provider, client applications and Adaptive Server communicate using the character set specified in the driver's CharSet connection property. The drivers assume that both the client applications and Adaptive Server support the specified character set. To handle cases where this is not true, additional connection properties such as NomalizeUnicodeString have been created.

To simplify character set conversion in Adaptive Server ODBC Driver and OLE DB Data Provider version 15.5, CharSet has been redefined to only specify the character set that is used to communicate with Adaptive Server. The ClientCharSet connection property has been created to specify the character set that the drivers use to communicate with client applications.

Set ClientCharset to the same value as CharSet to reproduce the behavior of Adaptive Server ODBC Driver and OLE DB Data Provider 15.0 and earlier.

JDBC 4.0 support

jConnect 7.0 supports these JDBC 4.0 specifications:

- Connection management
- Automatic SQL driver loading
- National character set conversion
- Database metadata
- Wrapper pattern
- Scalar functions CHAR_LENGTH, CHARACTER_LENGTH, CURRENT_DATE, CURRENT_TIME, CURRENT_TIMESTAMP, EXTRACT, and OCTET_LENGTH, POSITION

See the Sun Developer Network at http://developers.sun.com/ for information about JDBC 4.0 specifications.

Microsoft ADO.NET Entity Framework and LINQ support

The Adaptive Server ADO.NET Data Provider supports the Visual Studio Language-Integrated Query (LINQ) and the Microsoft ADO.NET Entity Framework including its LINQ-to-SQL component. The following, however, are not supported due to Microsoft ADO.NET Entity Framework limitations:

- Use of the LINQ Contains extension method. Contains maps to the SQL IN clause.
- Creation of LINQ extension methods.
- Creation of associations between entity classes.

One of the advantages of ADO.NET Entity Framework and LINQ is that these allow you to work with a conceptual model of a relational storage schema, thus decreasing development and maintenance efforts for data-oriented applications. To use Microsoft ADO.NET Entity Framework and LINQ, reference *Sybase.AdoNet2.AseClient.dll*.

For more information about the ADO.NET Entity Framework and LINQ, see the Microsoft Developer Network at http://msdn.microsoft.com.

Microsoft Enterprise Library Database Access Application Block for Adaptive Server

The Adaptive Server ADO.NET Data Provider 2.155 extends the Microsoft Enterprise Library 4.1 Data Access Application Block (DAAB) to support Adaptive Server. DAAB is a collection of classes that simplifies common database functions such as creating database instances and updating database records. DAAB also encapsulates database-specific features, which allows for a database-independent application design.

DAAB for Adaptive Server classes are supported in Microsoft .NET Framework 3.5 and Microsoft Visual Studio 2008. To use DAAB for Adaptive Server, you need to update your copy of Microsoft Enterprise Library 4.1. See *Adaptive Server Enterprise ADO.NET Data Provider 2.155 Users Guide* for details.

For information about the Enterprise Library Data Access Application Block, see the Microsoft Developer Network at http://msdn.microsoft.com.

Accessibility features

Section 508 requires that U.S. Federal agencies' electronic and information technology is accessible to people with disabilities. Sybase strongly supports Section 508 and has made a range of Sybase products Section 508-compliant, including Open Client and Open Server version 15.5.

Documents in the 15.5 release are available in HTML specialized for accessibility. You can navigate the HTML with an adaptive technology such as a screen reader, or view it with a screen enlarger. Open Client and Open Server documentation has been tested for compliance with U.S. government Section 508 Accessibility requirements. Documents that comply with Section 508 generally also meet non-U.S. accessibility guidelines, such as the World Wide Web Consortium (W3C) guidelines for Web sites.

You might need to configure your accessibility tool for optimal use. Some screen readers pronounce text based on its case; for example, they pronounce ALL UPPERCASE TEXT as initials, and MixedCase Text as words. You might find it helpful to configure your tool to announce syntax conventions. Consult the documentation for your tool.

For information about how Sybase supports accessibility, see Sybase Accessibility at http://www.sybase.com/accessibility. The Sybase Accessibility site includes links to information on Section 508 and W3C standards.