

Nouvelles fonctionnalités Open Server™ 12.5.1 et SDK 12.5.1 pour Microsoft Windows, Linux et UNIX

DC00505-01-1251-01

Avril 2006

Ce document décrit les nouvelles fonctionnalités disponibles pour Open Server™ 12.5.1 et le Software Developer's Kit (SDK) 12.5.1, à partir de la dernière version ESD, ESD#13.

Le SDK contient les composants suivants :

- Open Client™ (DB-Library™, Client-Library™)
- Embedded SQL™/C (ESQL/C)
- Embedded SQL/COBOL (ESQL/COBOL)
- Adaptive Server® Enterprise (ASE) ODBC Driver by Sybase® (Microsoft Windows et Linux)
- Adaptive Server Enterprise (ASE) OLE DB Provider by Sybase (Microsoft Windows uniquement)
- Adaptive Server Enterprise (ASE) ADO.NET Data Provider (Microsoft Windows uniquement)
- Extended Architecture (XA)
- jConnect™ for JDBC™

Dans la mesure où de nouvelles fonctionnalités seront disponibles dans les prochaines versions ESD, ce document sera mis à jour afin de les inclure.

Sujet	Page
Compatibilités des produits	4
Nouvelles fonctionnalités d'ESD#13	7
Nouvelles propriétés de connexion de jConnect 6.05	7
Nouvelles propriétés de connexion de jConnect 5.5	8
Mises à jour et clarifications de la documentation pour ESD#11	12
Nouvelles fonctionnalités d'ESD#12	12
Prise en charge de SSL Plus sur Linux AMD64 (Opteron)/EM64T et HP Itanium 32 bits et 64 bits	12
Prise en charge de MIT Kerberos	12
MIT Kerberos sur DB-Library	14
Prise en charge d'ESQL/COBOL sur HP Itanium 32 bits	14
Participation d'ASE OLE DB Provider à des transactions distribuées	14
Outil OLE DB DSN Migration	18
Reprise haute disponibilité (HA) sur l'ASE OLE DB Provider	19
Nouvelles fonctionnalités d'ESD#11	22
Prise en charge de MIT Kerberos sur Sun Solaris 9 x86 (32 bits)	22
Prise en charge des fichiers sql.ini et interfaces des services de répertoire	22
Outil ODBC DSN Migration	25
Prise en charge des signets et de bulk pour ODBC et OLE DB	27
Prise en charge des fichiers sql.ini et interfaces des services de répertoire sur jConnect	27
Nouvelles fonctionnalités d'ESD#10	29
Prise en charge des plates-formes HP-UX 11.11 (ou HP-UX 11 liv1.0) 32 bits et 64 bits	29
Prise en charge de XA 64 bits sur Linux AMD64 (Opteron)/EM64T	29
Prise en charge de MIT Kerberos sur Microsoft Windows	29
Curseurs avec défilement	30
Nouvelles fonctionnalités d'ESD#9	37
Utilisation de transactions distribuées	37
Nouvelles fonctionnalités d'ESD#8	42
Prise en charge étendue de MIT Kerberos	42
Prise en charge pour de nouveaux systèmes d'exploitation et plates-formes	42

Sujet	Page
Fonctionnalité LDAP sur IBM AIX 64 bits	43
Mise à niveau de CyberSafe Kerberos Client sur Windows	44
Correctifs de sécurité SSLPlus 3.1.x	44
Mise à niveau du compilateur COBOL sur Windows 2000	45
Nouvelles fonctionnalités d'ESD#7	45
Prise en charge de XA 64 bits étendue à de nouvelles plates-formes	45
Nouvelles fonctionnalités d'ESD#6	47
Nouvel ASE OLE DB Provider by Sybase	47
Nouvelles fonctionnalités d'ESD#5	48
Colonnes cryptées BCP	48
Amélioration de la prise en charge des caractères coréens	48
Correctifs de sécurité SSLPlus 3.1.10H	49
Prise en charge de MIT Kerberos sur ODBC et OLE DB	49
Installation de SDK 12.5.1 sur Windows	49
Prise en charge étendue des plates-formes pour l'ASE ODBC Driver by Sybase	50
Nouvelles fonctionnalités d'ESD#4	51
Amélioration du diagnostic	51
Prise en charge des communications asynchrones	51
Nouvelle variable de configuration de DB-Library	53
Pilote Adaptive Server Enterprise ODBC pour Linux Intel 32 bits	55
Nouveau pilote ASE ODBC	56
Nouvelles fonctionnalités d'ESD#3	56
Adaptive Server Enterprise ADO.NET Data Provider 1.1 sur Microsoft Windows	56
Nouvelles fonctionnalités d'ESD#2	58
Prise en charge de SSL Plus 3.1.5 + SBGSE sur Solaris et Microsoft Windows	58
Prise en charge de IPV6 sur Solaris et Microsoft Windows XP et 2003	59
Fonctionnalité LDAP de DB-Library sur Microsoft Windows	60
Nouvelles fonctionnalités d'ESD#1	61
Prise en charge de XA sur Sun Solaris 8 (SPARC 64 bits)	61

Compatibilités des produits

Le [Tableau 1](#) présente la compatibilité plate-forme d'Open Client et Open Server :

Remarque Sun Solaris 10 x64 est pris en charge à partir de l'ESD#13.

Tableau 1 : Compatibilité plate-forme d'Open Client/Open Server

Plate-forme	Niveau du système d'exploitation	Compilateurs C et C++	Compilateurs COBOL	Version de Kerberos	Light-weight Directory Access (LDAP)	Secure Sockets Layer (SSL)
HP-UX 11.0 32 bits	HP-UX 11.0 avec patch 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 bits	HP-UX 11.0 avec patch 990P	HP C 11.00.00 ANSI HP ANSI C++ B3910B A.03.10	Non disponible	Non disponible	Netscape LDAP 4.1	Certicom SSL Plus 3.1.15
HP-UX 11.11 (ou HP-UX 11iv1.0) 32 bits	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 (ou HP-UX 11iv1.0) 64 bits	HP-UX 11i v1 avec patch 990P	HP C 11.11.10 ANSI HP ANSI C++ B3910B A.03.10	Non disponible	MIT 1.4.3	Netscape LDAP 4.1	Certicom SSL Plus 3.1.15
HP Itanium 32 bits	HP-UX 11.23	HP C++/ANSI C B3910B A.0550	Micro Focus Server Express 4.0 SP2	Non disponible	Netscape LDAP 4.1	Certicom SSL Plus 5.0.6f
HP Itanium 64 bits	HP-UX 11.23	HP C++/ANSI C B3910B A.0550	Non disponible	Non disponible	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 bits	AIX 5.2	C++ 5.0.22	Micro Focus Server Express 2.0.10	Cybersafe Trust Broker 2.1	Non disponible	Certicom SSL

Plate-forme	Niveau du système d'exploitation	Compilateurs C et C++	Compilateurs COBOL	Version de Kerberos	Light-weight Directory Access (LDAP)	Secure Sockets Layer (SSL)
IBM AIX 64 bits	AIX 5.2	C++ 5.0.22	Non disponible	MIT 1.4.3	Non disponible	Certicom SSL
Linux on POWER 32 bits	Red Hat Enterprise Linux AS 3.0	IBM XL C/C++ Advance Edition V7.0	Non disponible	Non disponible	Non disponible	Non disponible
Linux on POWER 64 bits	Red Hat Enterprise Linux AS 3.0	IBM XL C/C++ Advance Edition V7.0	Non disponible	Non disponible	Non disponible	Non disponible
Linux AMD64 (Opteron)/ EM64T	Red Hat Enterprise AS 3.0	GCC 3.2.3 (Red Hat Linux 3.2.3 - 42)	Non disponible	MIT 1.2.7	Non disponible	Certicom SSL Plus 5.0.4m
Linux Intel 32 bits	Red Hat AS 2.1 ou United Linux/SuSe SLES 8.0/UL1.0	Compilateur C pris en charge par le système d'exploitation	Non disponible	MIT 1.3.1	Netscape LDAP 4.1	Certicom SSL
Linux Itanium 64 bits	Red Hat Advance Server 2.1	GCC C 2.96	Non disponible	Non disponible	Non disponible	Non disponible
Linux PowerPC	Red Hat AS 3.0 ou United Linux/SuSe SLES 9.0 (SLES 9)	XL C/C ++ Advance Edition V7.0	Non disponible	Non disponible	Non disponible	Non disponible
SGI 32 bits	IRIX 6.5.18	MIPSPro C7.3.x MIPSPro C++ 7.3.x	Non disponible	Non disponible	Netscape LDAP 4.0	Non disponible
SGI 64 bits	IRIX 6.5.18	MIPSPro C7.3.x MIPSPro C++ 7.3.x	Non disponible	Non disponible	Non disponible	Non disponible
Sun Solaris 8 (SPARC 32 bits)	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 bits)	Solaris 8	Sun C/C++ 6.2	Non disponible	MIT 1.3.1	Netscape LDAP 4.1	Certicom SSL Plus 3.1.14

Compatibilités des produits

Plate-forme	Niveau du système d'exploitation	Compilateurs C et C++	Compilateurs COBOL	Version de Kerberos	Light-weight Directory Access (LDAP)	Secure Sockets Layer (SSL)
Sun Solaris 9 x 86 (32 bits)	Solaris 9	Sun C/C++ 6.2	Non disponible	MIT 1.4.1	Non disponible	Non disponible
Sun Solaris 10 x 64 (64 bits)	Solaris 10	Studio 10 (Sun C/C++ 5.7)	Non disponible	MIT 1.4.1	Open LDAP 2.2.26	Certicom SSL Plus 5.0.4
Windows NT 32 bits	NT 4.0 Service Pack 4	MS C 6.0 (Microsoft Developers Studio ; version non optimisée, uniquement destinée au développement)	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 ; version non optimisée, uniquement destinée au développement)	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 ; version non optimisée, uniquement destinée au développement)	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 ; version non optimisée, uniquement destinée au développement)	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

Remarque Vous pouvez toutefois utiliser d'autres compilateurs pour générer les programmes.

Nouvelles fonctionnalités d'ESD#13

ESD#13 propose les nouvelles fonctionnalités suivantes :

Nouvelles propriétés de connexion de jConnect 6.05

Le [Tableau 2](#) répertorie les nouvelles propriétés de connexion disponibles pour jConnect 6.05. Pour obtenir une liste complète des propriétés de connexion, reportez-vous au chapitre 2, « Programming Information », du guide *Programmer's Reference* de jConnect for JDBC 6.05.

Tableau 2 : Propriétés de connexion de jConnect 6.05

Propriété	Description	Valeur par défaut
CRC	Lorsque cette propriété est définie sur <i>true</i> , le nombre de mises à jour renvoyé correspond au nombre total et inclut les mises à jour directement affectées par l'instruction exécutée et les triggers invoqués dans le cadre de l'exécution de cette instruction.	<i>false</i>
DATABASE	Utilisez cette propriété pour spécifier le nom de la base de données pour une connexion lorsque les informations de connexion proviennent d'un fichier <i>interfaces</i> de Sybase. L'URL d'un fichier <i>interfaces</i> ne peut servir de nom de base de données.	<i>null</i>
DEFAULT_QUERY_TIMEOUT	Lorsque cette propriété de connexion est définie, elle est utilisée en tant que délai de requête par défaut pour toute instruction créée à l'aide de cette connexion.	<i>0</i> (pas de temporisation)
IMPLICIT_CURSOR_FETCH_SIZE	Utilisez cette propriété en combinaison avec la propriété <code>SELECT_OPENS_CURSOR</code> pour forcer jConnect à ouvrir un curseur en lecture seule sur chaque requête <code>select</code> envoyée à la base de données. La taille d'extraction du curseur correspondra à la valeur définie pour cette propriété, sauf si celle-ci est ignorée par la méthode <code>Statement.setFetchSize</code> .	<i>0</i>

Propriété	Description	Valeur par défaut
INTERNAL_QUERY_TIMEOUT	Utilisez cette propriété pour définir le délai de requête utilisé par les instructions créées en interne et exécutées par jConnect. Elle peut empêcher les interruptions de l'application lorsque les commandes internes ne s'exécutent pas dans un délai raisonnable.	0 (pas de temporisation)
J2EE_TCK_COMPLIANT	Lorsque cette propriété est définie sur <i>true</i> , le pilote jConnect adopte un comportement compatible avec la suite de test J2EE 1.4 TCK, ce qui provoque une diminution des performances. Sybase conseille dès lors d'utiliser la valeur par défaut <i>false</i> .	<i>false</i>

Nouvelles propriétés de connexion de jConnect 5.5

Le [Tableau 3](#) répertorie les nouvelles propriétés de connexion disponibles pour jConnect 5.5. Pour obtenir une liste complète des propriétés de connexion, reportez-vous au chapitre 2, « Programming Information », du guide *Programmer's Reference* de jConnect for JDBC 5.5.

Tableau 3 : Propriétés de connexion de jConnect 5.5

Propriété	Description	Valeur par défaut
CACHE_COLUMN_METADATA	Utilisez cette propriété pour permettre à jConnect de mettre en cache ResultSet Metadata en cas d'exécutions consécutives lorsque DYNAMIC_PREPARE est défini sur <i>true</i> . Les performances s'en trouvent améliorées.	Faux

Propriété	Description	Valeur par défaut
CAPABILITY_TIME	<p>Utilisez uniquement cette propriété lorsque JCONNECT_VERSION est supérieur ou égal à 6. Lorsque jConnect est connecté à un serveur qui prend en charge le type de données TIME, tous les paramètres de type java.sql.Time ou les littéraux d'échappement {t...} sont traités en tant que TIME. Les précédentes versions de jConnect traitent ces paramètres en tant que DATETIME et ajoutent le préfixe « 1970-01-01 » au paramètre java.sql.Time. La date est également stockée dans la base de données si le type de données sous-jacent est datetime ou smalldatetime. Dans les nouvelles versions de jConnect, lorsque TIME est traité, le serveur convertit TIME dans le type de données sous-jacent et ajoute sa propre année de référence en guise de préfixe. Cela peut provoquer des incompatibilités entre les anciennes et les nouvelles données. Si vous utilisez le type de données datetime ou smalldatetime pour java.sql.Time, vous devez laisser CAPABILITY_TIME sur <i>false</i> à des fins de compatibilité descendante. En laissant cette propriété sur <i>false</i>, vous obliger jConnect à traiter les paramètres java.sql.Time ou les littéraux d'échappement {t...} en tant que DATETIME quelle que soit la capacité de traitement du type de données TIME du serveur. Si vous définissez cette propriété sur <i>true</i>, jConnect traite les paramètres java.sql.Time en tant que type de données TIME lorsqu'il est connecté à ASE 12.5.1 ou à une version ultérieure. Sybase vous conseille de laisser cette propriété sur <i>false</i> si vous utilisez la colonne smalldatetime ou datetime pour stocker des valeurs temporelles.</p>	Faux

Propriété	Description	Valeur par défaut
CAPABILITY_WIDETABLE	Cette propriété peut être définie sur <i>false</i> pour améliorer les performances si vous n'avez pas besoin de JDBC ResultSetMetaData comme nom de colonne. L'échange de données sur le réseau est dès lors moins important, ce qui améliore les performances. Sybase vous conseille d'utiliser le paramètre par défaut, sauf si vous employez EAServer. Reportez-vous à la section « Using wide table support for ASE 12.5 and later » du chapitre 2 du guide <i>jConnect for JDBC Programmer's Reference</i> .	Vrai
CRC	Lorsque cette propriété est définie sur <i>true</i> , le nombre de mises à jour renvoyé correspond au nombre total et inclut les mises à jour directement affectées par l'instruction exécutée et les triggers invoqués dans le cadre de l'exécution de cette instruction.	Faux
DATABASE	Utilisez cette propriété pour spécifier le nom de la base de données pour une connexion lorsque les informations de connexion proviennent d'un fichier <i>interfaces</i> de Sybase. L'URL d'un fichier <i>interfaces</i> ne peut servir de nom de base de données.	Null
IMPLICIT_CURSOR_FETCH_SIZE	Utilisez cette propriété en combinaison avec la propriété <code>SELECT_OPENS_CURSOR</code> pour forcer jConnect à ouvrir un curseur en lecture seule sur chaque requête <code>select</code> envoyée à la base de données. La taille d'extraction du curseur correspondra à la valeur définie pour cette propriété, sauf si celle-ci est ignorée par la méthode <code>Statement.setFetchSize</code> .	0

Propriété	Description	Valeur par défaut
QUERY_TIMEOUT_CANCEL_ALL	Utilisez cette propriété pour contraindre jConnect à annuler toutes les instructions d'une connexion en cas de temporisation de la lecture. Ce comportement peut être utilisé lorsqu'un client appelle <code>execute()</code> et qu'une temporisation intervient en raison d'un interblocage (par exemple, en cas de tentative de lecture à partir d'une table en cours de mise à jour par une autre transaction). En fonction des discussions futures avec Sun, cette propriété pourrait être combinée avec les valeurs de propriété affectées par la propriété <code>BE_AS_JDBC_COMPLIANT_AS_POSSIBLE</code> .	Faux
SERVER_INITIATED_TRANSACTIONS	Cette propriété permet au serveur de contrôler des transactions. Par défaut, elle est configurée sur <code>true</code> et jConnect autorise le serveur à lancer et contrôler des transactions en utilisant <code>Transact-SQL set chained on</code> . Si la propriété est définie sur <code>false</code> , les transactions sont lancées et contrôles par jConnect à l'aide de <code>transact sql begin tran</code> . Sybase vous conseille d'autoriser le serveur à contrôler les transactions.	Vrai
SERVERTYPE	Cette propriété peut être définie sur « OSW » lorsque vous êtes connecté à OpenSwitch. Cela permet à jConnect d'envoyer certaines instructions à OpenSwitch, ce qui permet à OpenSwitch de garder en mémoire les paramètres de connexion initiaux, tels que le niveau d'isolement, <code>textsize</code> , <code>quoted identifier</code> et <code>autocommit</code> , même s'il est déplacé vers une autre instance de serveur.	Aucune
TEXTSIZE	Utilisez cette propriété pour définir <code>TEXTSIZE</code> . Par défaut, ASE et ASA permettent de lire 32 627 octets à partir d'une colonne <code>image</code> ou <code>text</code> . jConnect remplace la valeur par 2 Go si vous avez installé les tables jConnect <code>mda</code> . La configuration de cette valeur lors de la connexion à OpenSwitch permet toutefois à la connexion de se souvenir des paramètres lorsque OpenSwitch est déplacé vers une autre instance de serveur.	2GB

Mises à jour et clarifications de la documentation pour ESD#11

La documentation de la fonctionnalité « Prise en charge des fichiers sql.ini et interfaces des services de répertoire » de ESD#11 a été mise à jour. Pour plus d'informations, reportez-vous à la section « [Prise en charge des fichiers sql.ini et interfaces des services de répertoire sur jConnect](#) », page 27.

Nouvelles fonctionnalités d'ESD#12

ESD#12 propose les nouvelles fonctionnalités suivantes :

Prise en charge de SSL Plus sur Linux AMD64 (Opteron)/EM64T et HP Itanium 32 bits et 64 bits

Sybase prend désormais en charge SSL Plus sur les plates-formes suivantes :

- SSL5.0.4m sur Linux AMD64 (Opteron)/EM64T
- SSL5.0.6f sur HP Itanium 32 bits
- SSL5.9.6h sur HP Itanium 64 bits

La prise en charge de SSL Plus s'applique aux produits Sybase suivants :

- Composants du Software Developer's Kit :
 - Client-Library
 - ESQL/C
- Open Server

Prise en charge de MIT Kerberos

MIT Kerberos 5 sur HP-UX 11.11 (ou HP-UX 11iv1.0) 64 bits et IBM AIX 64 bits est passé de la version 1.3.6 à la 1.4.3.

MIT Kerberos 5 version 1.3.6 est désormais pris en charge sur HP-UX 11.11 (ou HP-UX 11iv1.0) 32 bits et HP-UX 11.0 32 bits.

Le [Tableau 4](#) dresse l'inventaire des versions de MIT Kerberos 5 sur les plates-formes actuellement prises en charge par Sybase.

Tableau 4 : Versions de MIT Kerberos 5 et plates-formes prises en charge

MIT Kerberos version 5	Plate-forme
Version 1.4.3	<ul style="list-style-type: none"> • HP-UX 11.11 (ou HP-UX 11iv1.0) 64 bits • IBM AIX 64 bits
Version 1.4.1	<ul style="list-style-type: none"> • Sun Solaris 9 x86 (32 bits)
Version 1.3.6	<ul style="list-style-type: none"> • HP-UX 11.11 (ou HP-UX 11iv1.0) 32 bits • HP-UX 11.0 32 bits • Microsoft Windows (y compris Windows NT, Windows 2000, Windows 2003, et Windows XP)
Version 1.3.1	<ul style="list-style-type: none"> • Linux Intel 32 bits • Sun Solaris 8 (SPARC 32 bits et 64 bits)
Version 1.2.7	<ul style="list-style-type: none"> • Linux AMD64 (Opteron)/EM64T

Pour utiliser et configurer les services de sécurité de MIT Kerberos sur les plates-formes ci-dessus, reportez-vous aux documents suivants :

- Chapitre et annexes suivants du guide *Open Client and Open Server Configuration Guide for UNIX* :
 - Chapitre 6, « Using Security Services »
 - Annexe B, « Fichiers de configuration »
 - Annexe E, « Kerberos Security Services »
- Chapitre et annexes suivants du guide *Open Client and Open Server Programmer's Supplement for UNIX* :
 - Chapitre 1, « Open Client Client-Library/C »
 - Annexe B, « Environment Variables »
- Chapitre et annexe suivants du guide *Open Client and Open Server Configuration Guide for Windows* :
 - Chapitre 6, « Using Security Services »
 - Annexe B, « Configuration Files »
- Chapitres suivants du guide *Open Client and Open Server Programmer's Supplement for Windows* :
 - Chapitre 1, « Building Open Client and Open Server Applications »
 - Chapitre 2, « Client Library/C Example Programs »

MIT Kerberos sur DB-Library

Le mécanisme de sécurité MIT Kerberos est désormais disponible sur DB-Library, où il propose des services d'authentification réseau et réciproque. Cette fonctionnalité permet à d'anciennes applications Sybase d'être « Kerberisées » afin d'utiliser les services d'authentification de Kerberos au terme d'un nombre moins important de modifications et de recompilations.

Les macros DB-Library suivantes ont été ajoutées pour assurer la prise en charge de Kerberos :

- DBSETLNETWORKAUTH : utilisée pour activer ou désactiver l'authentification réseau de base.
- DBSETLMUTUALAUTH : utilisée pour activer ou désactiver l'authentification réciproque du mécanisme de sécurité de la connexion.
- DBSETLSERVERPRINCIPAL : utilisée pour configurer le nom principal du serveur, le cas échéant.

Pour plus d'informations sur l'installation de MIT Kerberos sur DB-Library, reportez-vous au *Guide d'installation et Notes de mise à jour* de Sybase SDK DB-Lib Kerberos Authentication Option 12.5.1.

Remarque DB-Library prend uniquement en charge les services d'authentification réseau et réciproque du mécanisme de sécurité Kerberos.

Prise en charge d'ESQL/COBOL sur HP Itanium 32 bits

ESQL/COBOL est désormais pris en charge sur HP Itanium 32 bits. Le compilateur COBOL de HP Itanium 32 bits est Micro Focus Server Express 4.0 SP2.

Participation d'ASE OLE DB Provider à des transactions distribuées

Cette fonctionnalité est uniquement prise en charge sur Windows et exige que Microsoft Distributed Transaction Coordinator (MS DTC) soit le coordinateur de transaction assurant la gestion des transactions distribuées.

Sybase prend en charge les modèles de programmation suivants :

- applications utilisant directement MS DTC ;
- applications utilisant Microsoft Transaction Server (MTS) ou (COM+).

Programmation pour MS DTC

- ❖ **Pour programmer à l'aide de Microsoft Distributed Transaction Coordinator (MS DTC)**
 - 1 Connectez-vous à MS DTC à l'aide de la fonction `DtcGetTransactionManager`. Pour plus d'informations sur MS DTC, reportez-vous à la documentation relative à Microsoft Distributed Transaction Coordinator.
 - 2 Procurez-vous l'`IDBSession` pour chacune des connexions Sybase ASE que vous voulez établir en suivant les étapes OLE DB.
 - 3 Appelez la fonction `IDBTransactionDispenser::BeginTransaction` pour entamer une transaction MS DTC et obtenir un objet de transaction OLE représentant la transaction.
 - 4 Exécutez une requête `IDBTransactionJoin` à partir de chacune des `IDBSession` (connexion OLE DB) que vous voulez inclure dans la transaction MS DTC, puis appelez `JoinTransaction` avec le paramètre `punkTransactionCoord` transmis en tant qu'objet de la transaction (obtenu à l'étape 3). A l'heure actuelle, Sybase prend uniquement en charge le niveau d'isolement de `ISOLATION LEVEL_READCOMMITTED` pour la transaction distribuée et ne prend pas en charge `IDBTransactionOptions`.
 - 5 Pour mettre à jour un SQL Server, suivez la procédure de l'OLE DB pour créer et exécuter la commande `IDBCommand`.
 - 6 Appelez la fonction `IDBTransaction::Commit` afin de valider la transaction MS DTC. L'objet de la transaction n'est plus valide.

Programmation des composants déployés dans MTS ou COM+

La procédure suivante explique comment créer des composants participant à des transactions distribuées dans MTS ou COM+.

- ❖ **Pour programmer des composants déployés dans MTS ou COM+**
 - 1 Créez un `IDBSession` pour chaque connexion ASE.
 - 2 Créez et exécutez `IDBCommand` pour chacune des mises à jour que vous souhaitez effectuer.
 - 3 Déployez votre composant dans MTS ou COM et configurez les attributs de transaction selon vos besoins.

COM+, OLE DB Services et OLE DB Provider se chargent de créer la transaction, de participer à celle-ci et de la valider ou de l'annuler.

OLE DB Services est nécessaire pour l'enrôlement automatique des transactions. Pour activer OLE DB Services, vous devez respecter certaines règles pour initialiser la source de données (reportez-vous aux documents relatifs à MS OLE). Pour activer l'enrôlement automatique des transactions, vous pouvez définir le bit DBPROPVAL_OS_TXNENLISTMENT dans le registre *OLE_DB_SERVICES* et dans la valeur de propriété DBPROP_INIT_OLEDBSERVICES ou transmettre OLE DB Services = 2 dans la chaîne de connexion.

Propriétés de connexion pour la prise en charge des transactions distribuées

La section suivante décrit les propriétés de connexion :

- Protocole de transaction distribuée (DistributedTransactionProtocol) – Pour spécifier le protocole à utiliser pour la prise en charge de la transaction distribuée, utilisez soit la norme d'interface XA, soit le protocole natif MS DTC OLE, sélectionnez le protocole de transaction distribuée dans la boîte de dialogue OLE DB Data Source, configurez la propriété DistributedTransactionProtocol = *OLE* dans la partie chaîne du fournisseur de la chaîne de connexion pour le protocole natif OLE ou dans le protocole par défaut utilisé *XA*.
- Transaction étroitement liée (TightlyCoupledTransaction) – Lorsqu'une transaction distribuée utilisant deux gestionnaires de ressources pointe vers le même serveur ASE, vous vous trouvez face à une « transaction étroitement liée ». Dans ce cas-là, si vous ne configurez pas cette propriété sur ASE1, la transaction distribuée échoue.

En résumé, si vous ouvrez deux connexions de base de données vers le même serveur ASE, puis que vous enrôlez ces connexions dans la même transaction distribuée, Sybase vous recommande de définir TightlyCoupledTransaction=1. Pour définir cette propriété, sélectionnez la transaction étroitement liée dans la boîte de dialogue OLE DB Data Source ou transmettez la propriété TightlyCoupledTransaction=1 dans la partie chaîne du fournisseur de la chaîne de connexion.

Outil OLE DB DSN Migration

L'outil OLE DB DSN Migration vous aide à migrer vos définitions de sources de données (DSN) depuis l'OLE DB Driver Kit vers l'OLE DB Provider by Sybase. Lorsque vous migrez les DSN, ils utilisent le nouveau OLE DB Provider by Sybase à la place de l'OLE DB Driver Kit.

Migration vers ASE OLE DB Provider by Sybase

Pour migrer des applications OLE DB afin d'utiliser ASE OLE DB Provider by Sybase, vous devez modifier la chaîne de connexion utilisée par les applications clientes OLE DB. Le nom court du fournisseur de l'ASE OLE DB Provider by Sybase est « ASEOLEDB ».

Les différences de comportement connues entre l'OLE DB Driver Kit et l'ASE OLE DB Provider by Sybase sont présentées dans les *Notes de mise à jour* de SDK 12.5.1 pour votre plate-forme.

Remarque La syntaxe de la chaîne de connexion pour ASE OLE DB Provider by Sybase est décrite dans le *Guide de l'utilisateur* d'Adaptive Server Enterprise OLE DB Provider by Sybase pour Microsoft Windows.

La syntaxe de la chaîne de connexion diffère de celle de l'OLE DB Driver Kit. L'OLE DB Provider by Sybase applique la syntaxe de l'OLE DB Driver Kit, mais Sybase recommande de migrer la syntaxe de la chaîne de connexion vers la nouvelle chaque fois que possible.

Migrations des noms de sources de données vers les gestionnaires Sybase

Il existe deux méthodes pour migrer des noms de sources de données (DSN) depuis l'OLE DB Driver Kit vers les gestionnaires créés par Sybase :

- [Utilisation de l'administrateur de sources de données Sybase ASE](#)
- [Utilisation de l'outil DSN Migration](#)

Utilisation de l'administrateur de sources de données Sybase ASE

L'administrateur de sources de données Sybase ASE est un processus de l'interface utilisateur graphique qui vous permet de migrer des sources de données existantes de l'OLE DB Driver Kit et de créer de nouvelles sources de données pour l'ASE OLE DB Provider.

❖ **Pour migrer les sources de données à l'aide de l'administrateur de sources de données**

- 1 Dans la fenêtre principale intitulée « Sybase Data Source Administrator », choisissez la source de données.
- 2 Cliquez sur Migrate.

L'administrateur de sources de données Sybase vous permet d'ajouter, supprimer, configurer ou tester les sources de données de l'OLE DB.

Utilisation de l'outil DSN Migration

L'outil DSN Migration peut vous aider à migrer les sources de données depuis l'OLE DB Driver Kit vers l'OLE DB Driver by Sybase.

L'outil dsnmigrate utilise des commutateurs pour contrôler les DSN qui migrent. A partir de la ligne de commande, entrez :

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

Les nouveaux DSN Sybase portent le même nom que les DSN OLE DB convertis.

Commutateurs
de conversion

Le tableau suivant répertorie et décrit les commutateurs utilisés lors de la conversion.

Tableau 5 : Commutateurs de conversion

Commutateurs	Description des résultats
/?,/h,/help	Affiche ce message. Ce message apparaît également si l'outil dsnmigrate est appelé sans argument de ligne de commande.
/oledb	Fait passer dsnmigrate en mode OLEDB. Par défaut, la migration porte sur les DSN ODBC.
/l	Affiche une liste de tous les DSN utilisateur et système de l'OLE DB Driver Kit.
/ul	Affiche une liste de tous les DSN utilisateur de l'OLE DB Driver Kit.
/sl	Affiche une liste de tous les DSN système de l'OLE DB Driver Kit.
/a	Convertit tous les DSN utilisateur et système de l'OLE DB Driver Kit.
/ua	Convertit tous les DSN utilisateur de l'OLE DB Driver Kit.
/sa	Convertit tous les DSN système de l'OLE DB Driver Kit.
/dsn	Convertit des DSN utilisateur ou système spécifiques de l'OLE DB Driver Kit.
/udsn	Convertit des DSN utilisateur spécifiques de l'OLE DB Driver Kit.
/sdsn	Convertit des DSN système spécifiques de l'OLE DB Driver Kit.
dsn	Nom du DSN à convertir.
/suffix	Commutateur facultatif qui modifie la manière dont les DSN sont appelés. Lorsque ce commutateur est utilisé, le DSN initial est conservé et le nouveau est appelé « <dsn>-<suffix> ».
suffix	Suffixe utilisé pour attribuer un nom au nouveau DSN.

Reprise haute disponibilité (HA) sur l'ASE OLE DB Provider

Le [Tableau 6](#) répertorie les nouveaux paramètres de connexion utilisés pour la prise en charge de la reprise haute disponibilité (HA) sur l'ASE OLE DB Provider :

Tableau 6 : Paramètres de configuration de la reprise haute disponibilité (HA)

Noms de propriétés	Description	Exigé	Valeur par défaut
HASession	Indique si la haute disponibilité est activée. 0 indique que la haute disponibilité est désactivée et 1 qu'elle est activée.	Non	0
SecondaryPort	Nom du port du serveur ASE jouant le rôle de serveur de reprise dans une configuration active-active ou active-passive.	Oui, si HASession est configuré sur 1.	Vide
SecondaryServer	Nom ou adresse IP du serveur ASE jouant le rôle de serveur de reprise dans une configuration active-active ou active-passive.	Oui, si HASession est configuré sur 1.	Vide

Utilisation de la reprise dans des systèmes haute disponibilité (HA)

Un cluster haute disponibilité se compose de deux machines ou plus configurées de telle sorte que, si l'une d'elles (ou une application) tombe en panne, sa charge de travail est traitée par la seconde. Chacune de ces machines constitue un nœud du cluster haute disponibilité. Ce type de cluster est utilisé dans un environnement qui doit impérativement être toujours disponible, tel qu'un système bancaire auquel les clients doivent pouvoir se connecter en permanence, quel que soit le jour.

Les machines sont configurées pour lire chacune les disques de l'autre, mais pas en même temps (tous les disques repris doivent être des disques partagés).

Par exemple, si Adaptive Server 1 est le serveur compagnon primaire et tombe en panne, Adaptive Server 2, en tant que serveur compagnon secondaire, lit ses disques (1 à 4) et gère les bases de données stockées jusqu'à ce qu'Adaptive Server 1 puisse redémarrer. Tous les clients connectés à Adaptive Server 1 sont automatiquement reconnectés à Adaptive Server 2.

Le mode reprise sur le serveur secondaire permet à Adaptive Server de fonctionner dans un cluster haute disponibilité dans le cadre d'une configuration active-active ou active-passive.

Pendant la reprise sur le serveur secondaire, les clients connectés au compagnon primaire qui utilise la propriété de reprise sur le serveur secondaire rétablissent automatiquement leurs connexions réseau vers le compagnon secondaire. Le mode reprise peut être activé en configurant la propriété de connexion HASession sur « 1 » (la valeur par défaut est « 0 »). Si cette propriété n'est pas définie, la reprise de session n'a pas lieu et ce, même si le serveur est configuré pour la reprise. Vous devez également configurer les propriétés SecondaryServer (l'adresse IP ou le nom du serveur ASE secondaire) et SecondaryPort (le numéro de port du serveur ASE secondaire).

Reportez-vous au manuel *Utilisation de Sybase Failover en environnement haute disponibilité* pour plus d'informations sur la configuration de votre système pour la haute disponibilité.

Lorsque le gestionnaire d'OLE DB Provider détecte une défaillance de la connexion au serveur ASE primaire, il essaie d'abord de se reconnecter à celui-ci. S'il n'y parvient pas, il estime qu'une panne s'est produite. Il essaie alors automatiquement de se connecter au serveur ASE secondaire à l'aide des propriétés de connexion définies dans SecondaryServer et SecondaryPort.

Confirmation du succès de la reprise

Si la connexion au serveur ASE secondaire est établie, l'ASE OLE DB Driver renvoie « E_FAIL » pour la fonction HRESULT.

Pour confirmer le succès de la reprise, consultez le champ dwMinor dans ERRORINFO (renvoyé par IErrorRecords::GetBasicErrorInfo) ou la description renvoyée par IErrorInfo::GetDescription. La valeur de dwMinor doit être « 30130 » en cas de succès de la reprise haute disponibilité. La description de IErrorInfo::GetDescription doit se présenter comme suit, *NomServeurASE* correspondant au nom du serveur sur lequel la reprise a été effectuée :

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

Remarque Sybase vous conseille de vérifier le code renvoyé par dwMinor pour déterminer si la reprise a réussi plutôt que d'examiner la description de l'erreur.

Le client doit ensuite réappliquer la transaction ayant échoué à la nouvelle connexion. Si la reprise intervient alors qu'une transaction est ouverte, seules les modifications qui ont été validées dans la base de données avant la reprise sont conservées.

Vérification de l'échec d'une reprise

Si la connexion au serveur secondaire n'est pas établie, l'ASE OLE DB Driver renvoie également « E_FAIL » pour la fonction HRESULT. Par contre, le champ dwMinor dans ERRORINFO (renvoyé par IErrorRecords::GetBasicErrorInfo) affichera « 30131 » et la description renvoyée par IErrorInfo::GetDescription sera :

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

Exemple de code pour la vérification de la reprise

L'exemple de code suivant montre comment écrire un code pour la reprise :

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

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

Nouvelles fonctionnalités d'ESD#11

ESD#11 propose les nouvelles fonctionnalités suivantes :

Prise en charge de MIT Kerberos sur Sun Solaris 9 x86 (32 bits)

MIT Kerberos 5 version 1.4.1 est désormais pris en charge sur Sun Solaris 9 x86 (32 bits).

Le [Tableau 7](#) dresse l'inventaire des versions de MIT Kerberos 5 sur les plates-formes actuellement prises en charge par Sybase.

Tableau 7 : Versions de MIT Kerberos 5 et plates-formes prises en charge

MIT Kerberos version 5	Plate-forme
Version 1.4.1	<ul style="list-style-type: none">• Sun Solaris 9 x86 (32 bits)
Version 1.3.6	<ul style="list-style-type: none">• IBM AIX 64 bits• HP-UX 11.11 (ou HP-UX 11iv1.0) 64 bits• Microsoft Windows (y compris Windows NT, Windows 2000, Windows 2003, et Windows XP)
Version 1.3.1	<ul style="list-style-type: none">• Linux Intel 32 bits• Sun Solaris 8 (SPARC 32 bits et 64 bits)
Version 1.2.7	<ul style="list-style-type: none">• Linux AMD64 (Opteron)/EM64T

Prise en charge des fichiers *sql.ini* et interfaces des services de répertoire

Cette nouvelle fonctionnalité prend en charge l'utilisation du fichier *sql.ini* (pour Windows) et du fichier *interfaces* (pour UNIX) pour fournir des informations sur le serveur. Elle est prise en charge pour les gestionnaires et fournisseurs suivants :

- ADO.NET Data Provider
- ODBC Driver
- OLE DB Provider

A l'heure actuelle, lors de la connexion au gestionnaire ou au fournisseur, vous devez définir plusieurs propriétés, telles que le nom du serveur ou l'adresse IP et le numéro de port d'un serveur ASE. En utilisant le fichier *sql.ini* ou *interfaces*, les entreprises peuvent centraliser les informations sur les services disponibles dans les réseaux de l'entreprise, et notamment des informations sur le serveur ASE.

Chaîne de connexion

Vous devez ajouter les propriétés suivantes à la chaîne de connexion afin d'identifier le fichier *sql.ini* ou *interfaces*. Vous pouvez vous connecter à une URL de services de répertoire (DSURL) unique ou à plusieurs.

Chaîne de connexion dans le cas d'une DSURL unique

Dans le cas d'une DSURL unique pour les gestionnaires et fournisseurs ADO.NET, ODBC et OLE DB, vous devez ajouter les propriétés suivantes à la chaîne de connexion en respectant le format suivant :

```
DSURL=fichier://[chemin]<nomfichier>[?][nomservice]
```

où :

- *chemin* (facultatif) est le chemin d'accès conduisant au fichier interfaces. Si ce chemin n'est pas spécifié, *%SYBASE%ini* est utilisé en tant que chemin d'accès par défaut sur Windows et *\$\$SYBASE* sur Linux et Mac OS X.
- *nomfichier* est le nom du fichier interfaces.
 - Sous Windows, le fichier interfaces s'appelle généralement *sql.ini*.
 - Sur UNIX, le fichier interfaces s'appelle généralement *interfaces*.
- *nomservice* (facultatif) est le nom du service défini par la DSURL, également appelé nom du serveur.
 - S'il n'est pas défini dans la DSURL, la propriété du serveur de la chaîne de connexion est utilisée.
 - Si le nom du service et le serveur existent tous deux, le nom du service est utilisé.
 - Si aucun des deux n'existe, le gestionnaire ou fournisseur charge les informations sur le serveur depuis le fichier *sql.ini* ou *interfaces*, si ceux-ci ne possèdent qu'une seule entrée. Si le fichier contient plusieurs entrées, une erreur est renvoyée.

Exemple d'erreur ODBC :

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

Par conséquent, la propriété *servicename* ou *server* doit toujours être définie dans l'URL de connexion, dans la mesure où les fichiers interfaces contiennent généralement plusieurs entrées de serveur.

Exemples de chaîne de connexion pour Windows et UNIX :

Windows

Exemple utilisant un chemin d'accès par défaut :

```
DSURL=file://sql.ini?mango1
```

où :

- Le chemin d'accès est omis dans l'exemple ci-dessus. Dès lors, *%SYBASE%/ini* est utilisé comme chemin par défaut. Cependant, si *%SYBASE%* est défini sur *C:\Sybase*, le chemin d'accès utilisé sera *C:\Sybase\ini*.
- Le nom du fichier est *sql.ini*.
- Le nom du service est *mango1*.

Exemple utilisant un chemin d'accès explicite :

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

où :

- Le chemin d'accès est *\\myServer\myShare*, qui est un chemin d'accès Universal Naming Convention (UNC) qui renvoie à un répertoire partagé du réseau.
- Le nom du fichier est *sql.ini*.
- Le nom du service est *mango1*.

Linux et Mac OS X

Exemple utilisant un chemin d'accès par défaut :

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

où :

- Le chemin d'accès est omis dans l'exemple ci-dessus. Dès lors, *\$\$SYBASE* est utilisé comme chemin par défaut. Cependant, si *\$\$SYBASE* est défini sur */usr/sybase*, le chemin d'accès utilisé sera */usr/sybase*.
- Le nom du fichier est *interfaces*.
- Le nom du service est *mango1*.

Exemple utilisant un chemin d'accès explicite :

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

où :

- Le chemin d'accès spécifié est */remote/sybase*
- Le nom du fichier est *interfaces*.
- Le nom du service est *mango1*.

Chaîne de connexion dans le cas de plusieurs URL

La DSURL peut prendre en charge plusieurs URL LDAP et fichiers *interfaces*. Sybase prend également en charge une URL multiple avec des URL LDAP et des URL de fichier mixtes. Lorsque plusieurs URL sont utilisées, le gestionnaire les traite une par une jusqu'à ce qu'il ouvre un fichier *interfaces* ou se connecte à un serveur LDAP.

Exemples
d'URL multiples

Voici quelques exemples d'utilisation de plusieurs URL dans une chaîne de connexion :

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

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

Format du fichier interfaces pour Microsoft Windows, UNIX et SSL

Pour plus d'informations sur le format du fichier interfaces, reportez-vous au *Guide de configuration d'Adaptive Server Enterprise* pour votre plate-forme.

Outil ODBC DSN Migration

L'outil ODBC DSN Migration vous aide à migrer d'ODBC Driver Kit vers ODBC Driver by Sybase. Lorsque vous migrez vos DSN, ils utilisent le nouveau ODBC Driver by Sybase à la place de l'ODBC Driver Kit.

Utilisation de l'outil de migration

L'outil `dsnigrate` utilise des commutateurs pour contrôler les DSN qui migrent. A partir de la ligne de commande, entrez :

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

Tous les DSN migrés sont renommés « `<dsn>-backup` » tant que la conversion n'est pas terminée. Lorsque les nouveaux DSN Sybase sont créés et que la conversion est terminée, ils reprennent le nom « `<dsn>` », ce qui permet aux applications existantes de continuer à fonctionner sans aucune modification.

Commutateurs de conversion

Le tableau suivant répertorie et décrit les commutateurs utilisés lors de la conversion.

Tableau 8 : Commutateurs de conversion

Commutateurs	Description des résultats
/?,/h,/help	Affiche ce message. Ce message apparaît également si l'outil dsmigrate est appelé sans argument de ligne de commande.
/l	Affiche une liste de tous les DSN utilisateur et système de l'ODBC Driver Kit.
/ul	Affiche une liste de tous les DSN utilisateur de l'ODBC Driver Kit.
/sl	Affiche une liste de tous les DSN système de l'ODBC Driver Kit.
/a	Convertit tous les DSN utilisateur et système de l'ODBC Driver Kit.
/ua	Convertit tous les DSN utilisateur de l'ODBC Driver Kit.
/sa	Convertit tous les DSN système de l'ODBC Driver Kit.
/dsn	Convertit des DSN utilisateur ou système spécifiques de l'ODBC Driver Kit.
/udsn	Convertit des DSN utilisateur spécifiques de l'ODBC Driver Kit.
/sdsn	Convertit des DSN système spécifiques de l'ODBC Driver Kit.
dsn	Spécifie le nom du DSN à convertir.
/suffix	Commutateur facultatif qui modifie la manière dont les DSN sont appelés. Lorsque ce commutateur est utilisé, le DSN initial est conservé et le nouveau est appelé « <dsn>-<suffix> ».
suffix	Spécifie le suffixe appliqué au nouveau DSN.

Prise en charge des signets et de bulk pour ODBC et OLE DB

Sybase prend en charge et les opérations bulk SQL pour l'ODBC Driver et l'OLE DB Provider.

Pour l'ODBC Driver

Insertions bulk qui utilisent `SQLBulkOperations` avec l'option de `SQL_ADD` et des mises à jour et suppressions au niveau de la position du curseur à l'aide de `SQLSetPos` (`SQL_UPDATE`, `SQL_DELETE`, `SQL_POSITION`). Pour obtenir des instructions sur l'utilisation de `SQL_ADD` et `SQLSetPos`, reportez-vous à la section Microsoft Developer Network library, dans le guide *ODBC Programmer's Reference* :

- `SQLBulkOperations` Function à l'adresse
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/htm/odbcsqlbulkoperations.asp?frame=true>
- `SQLSetPos` Function à l'adresse
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/htm/odbcsqlsetpos.asp>

Pour l'OLE DB Provider

Opérations de signets utilisant l'interface `IRowsetLocate`, qui propose des méthodes pour comparer les signets et retrouver des lignes sur la base de signets. Pour obtenir des instructions sur l'utilisation de `IRowsetLocate`, reportez-vous au guide MSDN OLE DB *Programmer's Reference* :

- `IRowsetLocate` à l'adresse
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/oledb/htm/oledbirowsetlocate__irowset.asp

Prise en charge des fichiers `sql.ini` et interfaces des services de répertoire sur `jConnect`

Cette nouvelle fonctionnalité prend en charge l'utilisation du fichier `sql.ini` (pour Windows) et du fichier `interfaces` (pour UNIX) pour fournir des informations sur le serveur pour `jConnect for JDBC`.

À l'heure actuelle, lors de la connexion au gestionnaire ou au fournisseur, vous devez définir plusieurs propriétés, telles que le nom du serveur ou l'adresse IP et le numéro de port d'un serveur ASE. En utilisant le fichier `sql.ini` ou `interfaces`, les entreprises peuvent centraliser les informations sur les services disponibles dans les réseaux de l'entreprise, et notamment des informations sur le serveur ASE.

Chaîne de connexion

Vous devez ajouter les propriétés suivantes à la chaîne de connexion afin d'identifier le fichier *sql.ini* ou *interfaces*. Sur jConnect for JDBC, vous pouvez uniquement vous connecter à une URL des services de répertoire (DSURL) unique.

Remarque jConnect ne prend pas en charge les URL multiples.

Chaîne de connexion dans le cas d'une DSURL unique for jConnect

Dans le cas d'une DSURL unique pour jConnect, ajoutez les propriétés suivantes à la chaîne de connexion en respectant le format ci-dessous :

Remarque L'utilisateur doit spécifier le chemin d'accès au fichier *sql.ini* et le nom du serveur.

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

où :

- Nom du serveur = *mango1*.
- Chemin d'accès au fichier *sql.ini* = *file://D:/syb1252/ini/sql.ini*

Si le chemin d'accès au fichier *sql.ini* ou le nom du serveur n'est pas spécifié dans l'URL, le gestionnaire renvoie une erreur.

Format du fichier interfaces pour SSL

L'exemple suivant montre le format à utiliser pour le fichier *sql.ini* sous SSL :

```
[SYBSRV2]
master=nlwnsck,mango1,4100,ssl
query=nlwnsck,mango1,4100,ssl
query=nlwnsck,mango1,5000,ssl
```

Remarque jConnect prend en charge des entrées query multiples sous le même nom de serveur dans le fichier *sql.ini*. jConnect tente de se connecter à des valeurs pour host ou port à partir de l'entrée query de la séquence, comme dans le fichier *sql.ini*. Si jConnect trouve un SSL dans une entrée query, il exige que l'application soit cryptée pour gérer les connexions SSL en spécifiant un socket spécifique à l'application, sinon la connexion risque d'échouer.

Nouvelles fonctionnalités d'ESD#10

ESD#10 propose les nouvelles fonctionnalités suivantes :

Prise en charge des plates-formes HP-UX 11.11 (ou HP-UX 11iv1.0) 32 bits et 64 bits

Les plates-formes HP-UX 11.11 (ou HP-UX 11iv1.0) 32 bits et 64 bits sont désormais prises en charge sur les produits Sybase SDK 12.5.1 et Open Server 12.5.1, comme expliqué dans la [Tableau 9](#).

Tableau 9 : Nouvelles plates-formes de SDK et Open Server 12.5.1 et fonctionnalités prises en charge

Plate-forme	Niveau du système d'exploitation	Compilateurs C et C++	Compilateurs COBOL	Version de Kerberos	Light-weight Directory Access (LDAP)	Secure Sockets Layer (SSL)
HP-UX 11.11 (ou HP-UX 11iv1.0) 32 bits	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 (ou HP-UX 11iv1.0) 64 bits	HP-UX 11i v1 avec patch 99OP	HP C 11.11.10 ANSI HP ANSI C++ B3910B A.03.10	Non disponible	MIT 1.4.1	Netscape LDAP 4.1	Certicom SSL Plus 3.1.14

Prise en charge de XA 64 bits sur Linux AMD64 (Opteron)/EM64T

XA 64 bits est désormais pris en charge sur Linux AMD64 (Opteron)/EM64T.

Prise en charge de MIT Kerberos sur Microsoft Windows

MIT Kerberos 5 version 1.3.6 est désormais pris en charge sur Microsoft Windows (y compris Windows NT, Windows 2000, Windows 2003 et Windows XP).

Curseurs avec défilement

L'ASE ODBC Driver et l'ASE OLE DB Provider prennent désormais en charge les curseurs avec défilement.

Utilisation des curseurs avec défilement

Les curseurs avec défilement permettent à des applications de définir la position actuelle du curseur à n'importe quel endroit du jeu de résultats en spécifiant les options de défilement appropriées. Les applications peuvent utiliser les options de défilement NEXT, PRIOR, FIRST, LAST, RELATIVE ou ABSOLUTE pour parcourir le jeu de résultats à leur guise.

Lorsqu'un curseur avec défilement est requis, il est possible de l'invoquer côté serveur ou client. Pour qu'un curseur côté serveur puisse être invoqué, il faut qu'Adaptive Server prenne en charge les curseurs avec défilement. Si ce n'est pas le cas, la fonctionnalité souhaitée peut être imitée par le gestionnaire à l'aide du jeu de résultats mis en cache. On parle alors de curseurs avec défilement côté client.

L'ASE ODBC Driver et l'ASE OLE DB Provider prennent en charge des curseurs avec défilement côté client et serveur à partir de la version ASE version 15.0. Par contre, seuls les curseurs avec défilement côté client sont pris en charge sur les versions d'ASE antérieures à 15.0.

La propriété UseCursor doit être définie correctement en vue d'obtenir le curseur avec défilement souhaité.

Avertissement ! Les curseurs avec défilement côté client utilisent plus de mémoire côté client et peuvent générer davantage de trafic réseau.

Configuration de la propriété de connexion UseCursor

La manière dont vous configurez la propriété de connexion UseCursor détermine si des curseurs avec défilement côté client ou serveur sont utilisés :

- Lorsque vous configurez la propriété de connexion UseCursor sur 1 et que vous utilisez ASE version 15.0 ou supérieure, les curseurs avec défilement côté serveur sont utilisés.

Remarque Les curseurs avec défilement côté serveur ne sont pas disponibles pour les versions d'ASE antérieures à 15.0.

- Si vous configurez la propriété de connexion UseCursor sur 0, les curseurs avec défilement côté client (jeux de résultats mis en cache) sont utilisés, quelle que soit la version d'ASE.

Pour l'ASE ODBC Driver

La section suivante décrit l'utilisation des curseurs avec défilement dans le cas de l'ASE ODBC Driver.

Prise en charge du curseur avec défilement Static Inensitive

L'ASE ODBC Driver prend en charge le curseur avec défilement **Static Inensitive**. Il implémente la méthode ODBC `SQLFetchScroll` pour faire défiler (**scrow**) et extraire (**fetch**) des lignes. La méthode `SQLFetchScroll` est une méthode ODBC standard définie dans le guide *Microsoft Open Database Connectivity Software Development Kit Programmer's Reference, Volume 2*, qui fait partie de la bibliothèque MSDN. Consultez le Microsoft Web site à l'adresse <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/htm/odbcsqlfetchscroll.asp> pour plus d'informations.pour plus d'informations.

Le pilote ODBC prend en charge les types de défilement suivants :

- `SQL_FETCH_NEXT` – renvoie le jeu de lignes suivant.
- `SQL_FETCH_PRIOR` – renvoie le jeu de lignes précédent.
- `SQL_FETCH_RELATIVE` – renvoie le jeu de lignes *n* à partir du début du jeu de lignes actuel.
- `SQL_FETCH_FIRST` – renvoie le premier jeu de lignes du jeu de résultats.
- `SQL_FETCH_LAST` – renvoie le dernier jeu de lignes complet du jeu de résultats.
- `SQL_FETCH_ABSOLUTE` – renvoie le jeu de lignes à partir de la ligne *n*.

Configuration des attributs des curseurs avec défilement

Vous devez définir les attributs suivants pour utiliser les curseurs avec défilement :

- `SQL_ATTR_CURSOR_SCROLLABLE` – type de curseur avec défilement que vous utilisez. Il doit être configuré sur la valeur de `SQL_SCROLLABLE`. Les valeurs possibles sont `static`, `semi-sensitive`, `insensitive`.
- `SQL_ATTR_CURSOR_SENSITIVITY` – valeur de sensibilité du curseur avec défilement sélectionné.

Remarque La seule valeur prise en charge pour cet attribut est `SQL_INSENSITIVE`.

Les attributs suivants sont *facultatifs* lors de l'utilisation de curseurs avec défilement.

- `SQL_ATTR_ROW_ARRAY_SIZE` – nombre de lignes que vous souhaitez obtenir chaque fois que vous avez recours à la méthode `SQLFetchScroll()`.

Remarque Si vous ne définissez pas cette valeur, la valeur par défaut (une ligne) est utilisée.

- `SQL_ATTR_CURSOR_TYPE` – type de curseur avec défilement que vous utilisez.

Remarque Les seules valeurs prises en charge pour cet attribut sont `SQL_CURSOR_FORWARD_ONLY` et `SQL_CURSOR_STATIC`.

- `SQL_ATTR_ROWS_FETCHED_PTR` – adresse où les lignes extraites sont stockées. L'attribut `SQL_ATTR_ROWS_FETCHED_PTR` pointe sur une variable du type de données `SQLINTEGER`.
- `SQL_ATTR_ROW_STATUS_PTR` – adresse où est stocké l'état des lignes. L'attribut `SQL_ATTR_ROW_STATUS_PTR` pointe sur une variable du type de données `SQLUSMALLINT`.

Exécution des curseurs avec défilement

❖ Pour configurer un programme afin qu'il exécute un curseur avec défilement

- 1 Définissez les attributs du curseur avec défilement en fonction de votre environnement.

Pour plus d'informations, reportez-vous à la section « [Configuration des attributs des curseurs avec défilement](#) », page 32.

- 2 Liez les résultats. Par exemple, ajoutez l'instruction suivante à votre programme :

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

- 3 Faites défiler les lignes et extrayez-les à l'aide de SQLFetchScroll(). Par exemple, ajoutez l'instruction suivante à votre programme :

```
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 Exécutez l'instruction Select. Par exemple, ajoutez l'instruction suivante à votre programme :

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

- 5 Fermez le jeu de résultats et le curseur. Par exemple, ajoutez l'instruction suivante à votre programme :

```
res = SQLFreeStmt(m_StatementHandle, SQL_CLOSE);
```

Examen des résultats

Après l'exécution d'un curseur avec défilement, les résultats suivants s'affichent, en supposant qu'il y a un total de N lignes et un jeu de lignes de m où $N > m$:

Résultats	Interprétation
Absolute 0	Aucune ligne n'est renvoyée, erreur.
Absolute 1	La ligne m est renvoyée.
Absolute N	1 ligne est renvoyée.
Absolute $N+1$	Aucune ligne n'est renvoyée, erreur.
First	Les premières lignes (1.. m) sont renvoyées.
Last	Les dernières lignes ($N-m+1$.. N) sont renvoyées.
Next	Identique à SQLFetch().
Prior	Renvoie le jeu de lignes situé avant le jeu actuel.

Les résultats suivants devraient être obtenus si le curseur actuel pointe sur la ligne k et $k-a > 0$, $k + m + a < N$, $a \geq 0$:

Résultats	Interprétation
Relative $-a$	Les lignes ($k-a$, $k-a + m - 1$) sont renvoyées.
Relative a	Les lignes ($k + a$, $k+a + m - 1$) sont renvoyées.

Configuration implicite des attributs des curseurs avec défilement

Certains attributs sont définis implicitement lorsque votre application configure des attributs spécifiques. Les attributs des curseurs avec défilement ODBC pris en charge définis implicitement sont les suivants :

L'application configure l'attribut sur	Autres attributs définis implicitement
SQL_ATTR_CONCURRENCY sur SQL_CONCUR_READ_ONLY	SQL_ATTR_CURSOR_SENSITIVITY sur SQL_INSENSITIVE
SQL_ATTR_CONCURRENCY sur SQL_CONCUR_LOCK	SQL_ATTR_CURSOR_SENSITIVITY sur SQL_SENSITIVE
SQL_ATTR_CURSOR_SCROLLABLE sur SQL_NONSCROLLABLE	SQL_ATTR_CURSOR_TYPE sur SQL_CURSOR_FORWARD_ONLY
SQL_ATTR_CURSOR_SENSITIVITY sur SQL_INSENSITIVE	SQL_ATTR_CONCURRENCY sur SQL_CONCUR_READ_ONLY SQL_ATTR_CURSOR_TYPE sur SQL_CURSOR_STATIC
SQL_ATTR_CURSOR_TYPE sur SQL_CURSOR_FORWARD_ONLY	SQL_ATTR_CURSOR_SCROLLABLE sur SQL_NONSCROLLABLE
SQL_ATTR_CURSOR_TYPE sur SQL_CURSOR_STATIC	SQL_ATTR_CURSOR_SCROLLABLE sur SQL_SCROLLABLE

Pour l'ASE OLE DB Provider

La section suivante décrit l'utilisation des curseurs avec défilement dans le cas de l'ASE OLE DB Provider.

Utilisation des curseurs avec défilement

L'OLE DB Data Provider prend en charge les types de défilement suivants :

- Next – renvoie la ligne suivante.
- Prior – renvoie la ligne précédente.
- Relative *n* rows – renvoie la ligne, *n* lignes du jeu de lignes actuel.

Configuration des attributs des curseurs avec défilement

Vous devez définir les attributs suivants pour utiliser les curseurs avec défilement :

- DBPROP_CANSCROLLBACKWARDS – lorsque cet attribut est défini sur VARIANT_TRUE, le jeu de lignes permet au paramètre lRowsOffset de GetNextRows d'être négatif.
- DBPROP_CANFETCHBACKWARDS – lorsque cet attribut est défini sur VARIANT_TRUE, le jeu de lignes permet au paramètre cRows de GetNextRows d'être négatif.

Exécution des curseurs avec défilement

❖ Pour configurer un programme afin qu'il exécute un curseur avec défilement

- 1 Définissez les propriétés du curseur avec défilement pour le jeu de lignes :

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

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

DBPROPSET rgRowsetProperty[1];
```

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

2 Ouvrez le jeu de lignes :

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

3 Extrayez les lignes vers l'avant :

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

4 Libérez les lignes :

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

5 Extrayez les lignes vers l'arrière :

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

6 Libérez les lignes :

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

7 Libérez le jeu de lignes :

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

Examen des résultats

Pour identifier les résultats et l'interprétation du jeu de résultats après avoir exécuté un curseur avec déplacement, reportez-vous à la section Microsoft MSDN library à l'adresse http://msdn.microsoft.com/library/default.asp?url=/library/en-us/oledb/htm/oledbirowset__getnextrows.asp.

Exemple de programme de curseur avec défilement static insensitive

Pour obtenir un exemple de programme de curseur avec défilement static insensitive, reportez-vous à la section précédente.

Nouvelles fonctionnalités d'ESD#9

ESD#9 propose la nouvelle fonctionnalité suivante :

Utilisation de transactions distribuées

Cette section explique comment utiliser l'ASE ODBC Driver et l'ASE ADO.NET Data Provider pour participer à des transactions avec commit à deux phases.

Utilisation de l'ASE ODBC Driver pour participer à des transactions distribuées

Cette fonctionnalité est uniquement prise en charge sur Microsoft Windows et exige que Microsoft Distributed Transaction Coordinator (MS DTC) soit le coordinateur de transaction assurant la gestion du commit à deux phases.

Sybase prend en charge tous les modèles de programmation suivants :

- applications utilisant directement MS DTC ;
- applications utilisant Sybase Enterprise Application Server (EAServer, également connu sous le nom de Jaguar) ;
- applications utilisant Microsoft Transaction Server (MTS) ou (COM+).

Programmation pour MS DTC

❖ Pour programmer à l'aide de Microsoft Distributed Transaction Coordinator (MS DTC)

- 1 Connectez-vous à MS DTC à l'aide de la fonction `DtcGetTransactionManager`. Pour plus d'informations sur MS DTC, reportez-vous à la documentation correspondante sur Microsoft Distributed Transaction Coordinator.
- 2 Appelez `SQLDriverConnect` ou `SQLConnect` une fois pour chacune des connexions Sybase ASE que vous souhaitez établir.
- 3 Appelez la fonction `ITransactionDispenser::BeginTransaction` pour entamer une transaction MS DTC et obtenir un objet de transaction OLE représentant la transaction.

- 4 Appelez `SQLSetConnectAttr` une ou plusieurs fois pour chacune des connexions ODBC que vous souhaitez enrôler dans la transaction MS DTC. `SQLSetConnectAttr` doit être appelé avec un attribut de `SQL_ATTR_ENLIST_IN_DTC` et un `ValuePtr` de l'objet de la transaction (obtenu à l'étape 3).
- 5 Appelez `SQLExecDirect` une ou plusieurs fois pour chaque instruction SQL insert ou update.
- 6 Appelez la fonction `ITransaction::Commit` afin de valider la transaction MS DTC. L'objet de la transaction n'est plus valide.

Pour effectuer une série de transactions MS DTC, répétez les étapes 3 à 6.

Pour libérer la référence à l'objet de la transaction, appelez la fonction `ITransaction::Release`.

Pour utiliser une connexion ODBC avec une transaction MS DTC et employer ensuite la même connexion avec une transaction locale de l'ASE Server, appelez `SQLSetConnectAttr` avec `ValuePtr` de `SQL_DTC_DONE` pour désenrôler la connexion de la transaction.

Remarque Vous pouvez également appeler `SQLSetConnectAttr` et `SQLExecDirect` séparément pour chaque ASE Server au lieu de procéder comme décrit aux étapes 4 et 5.

Programmation des composants déployés dans Sybase EAServer, MTS ou COM+

La procédure suivante explique comment créer des composants participant à des transactions distribuées dans Sybase EAServer, MTS ou COM+.

- ❖ **Pour programmer des composants déployés dans Sybase EAServer, MTS ou COM+**
 - 1 Appelez `SQLDriverConnect` une fois pour chacune des connexions Sybase ASE que vous souhaitez établir.
 - 2 Appelez `SQLExecDirect` une ou plusieurs fois pour chaque instruction SQL insert ou update.
 - 3 Déployez votre composant dans MTS et configurez les attributs de transaction selon vos besoins.

Le coordinateur de transaction crée une transaction distribuée, si nécessaire, et le composant qui utilise l'ASE ODBC Driver est automatiquement enrôlé dans la transaction globale. Le coordinateur de transaction valide alors la transaction distribuée ou l'annule.

Propriétés de connexion pour la prise en charge des transactions distribuées

La section suivante décrit les propriétés de connexion :

- Protocole de transaction distribuée (DistributedTransactionProtocol) – Pour spécifier le protocole à utiliser pour la prise en charge de la transaction distribuée, la norme d'interface XA ou le protocole natif MS DTC OLE, sélectionnez le protocole de transaction distribuée dans la boîte de dialogue ODBC Data Source ou transmettez la propriété DistributedTransactionProtocol = *OLE* dans la chaîne de connexion. La valeur par défaut est *XA*.
- Transaction étroitement liée (TightlyCoupledTransaction) – Lorsque vous avez une transaction distribuée utilisant deux gestionnaires de ressources qui pointent vers le même serveur ASE, vous vous trouvez face à une « transaction étroitement liée ». Dans ce cas-là, si vous ne configurez pas cette propriété sur *1*, la transaction distribuée échoue.

En résumé, si vous ouvrez deux connexions de base de données au même serveur ASE et enrôlez ensuite ces connexions dans la même transaction distribuée, vous devez définir TightlyCoupledTransaction=*1*.

Pour configurer cette propriété, sélectionnez la transaction étroitement liée dans la boîte de dialogue ODBC Data Source ou transmettez la propriété TightlyCoupledTransaction=*1* dans la chaîne de connexion.

Avertissement ! L'enrôlement avec SQLSetConnectAttr renvoie SQL_ERROR si la connexion a déjà entamé une transaction locale que ce soit en utilisant SQLSetConnectAttr avec SQL_AUTOCOMMIT_OFF ou en exécutant l'instruction BEGIN TRANSACTION de manière explicite à l'aide de SQLExecDirect.

Utilisation de l'ASE ADO.NET Data Provider pour participer à des transactions distribuées

Cette fonctionnalité requiert l'utilisation de .NET Enterprise Services, qui gère les transactions distribuées.

Programmation à l'aide d'Enterprise Services

L'infrastructure de services COM+ est accessible à partir d'un code géré ou non géré. Les services proposés dans un code non géré sont connus sous le nom de services COM+. Dans .NET, ces services sont appelés Enterprise Services. L'utilisation de transactions dans Enterprise Services avec ADO.NET est très simple.

❖ Pour programmer à l'aide d'Enterprise Services

- 1 Appelez les composants depuis *System.EnterpriseService.ServicedComponent*.
- 2 Définissez les attributs personnalisés (tels que *Transaction*, *AutoComplete*, etc.) afin de configurer les services requis et leurs options. Pour obtenir une liste complète des attributs, reportez-vous à la documentation d'Enterprise Services.

Remarque L'option *Timeout* de l'attribut de transaction .NET doit être explicitement configurée sur *-1* ou un chiffre très grand. La documentation de .NET indique que la valeur par défaut de temporisation d'une transaction ADO.NET est *0*, ce qui signifie que la transaction n'est jamais temporisée. Cette valeur provoque toutefois une temporisation immédiate de la transaction, qui annule l'ensemble de la transaction.

- 3 Validez et construisez l'ensemble.
- 4 Enregistrez l'ensemble.

Propriétés de connexion pour la prise en charge des transactions distribuées

Les propriétés de connexion suivantes sont utilisées dans le cadre de la prise en charge des transactions distribuées.

- Protocole de transaction distribuée (*DistributedTransactionProtocol*) – Pour spécifier le protocole à utiliser pour la prise en charge de la transaction distribuée, la norme d'interface *XA* ou le protocole natif *MS DTC OLE*, en configurant la propriété *DistributedTransactionProtocol = OLE* dans la chaîne de connexion. Le protocole par défaut est *XA*.
- Transaction étroitement liée (*TightlyCoupledTransaction*) – Lorsque vous avez une transaction distribuée utilisant deux gestionnaires de ressources qui pointent vers le même serveur ASE, vous vous trouvez face à une « transaction étroitement liée ». Dans ce cas-là, si vous ne configurez pas cette propriété sur *1*, la transaction distribuée échoue.

En résumé, si vous ouvrez deux connexions de base de données au même serveur ASE et les enrôlez dans la même transaction distribuée, vous devez définir *TightlyCoupledTransaction=1*.

- Enrôlement – L'objet `AseConnection` est automatiquement enrôlé dans une transaction distribuée existante s'il trouve une transaction active. L'enrôlement automatique dans une transaction intervient lorsque la connexion est ouverte ou récupérée depuis le groupe de connexions. Vous pouvez désactiver cet enrôlement automatique en définissant `Enlist=0` en tant que paramètre de la chaîne de connexion pour `AseConnection`.

Si l'enrôlement automatique est désactivé, vous pouvez enrôler une connexion dans une transaction distribuée existante en appelant la méthode `EnlistDistributedTransaction` sur `AseConnection` avec un paramètre `ITransaction` transmis qui référence une transaction existante. Après avoir appelé `EnlistDistributedTransaction`, toutes les mises à jour effectuées avec cette instance de `AseConnection` sont intégrées dans cette transaction globale. De ce fait, elle est validée ou annulée en même temps que la transaction globale.

Remarque L'objet `AseConnection` doit être ouvert avant d'appeler `EnlistDistributedTransaction`.

Vous pouvez utiliser `EnlistDistributedTransaction` lorsque vous regroupez des objets commerciaux. Si un objet commercial est regroupé à l'aide d'une connexion ouverte, l'enrôlement automatique de la transaction intervient uniquement lorsque cette connexion est ouverte ou récupérée du groupe de connexions. Si plusieurs transactions sont effectuées à l'aide de l'objet commercial regroupé, la connexion ouverte pour cet objet n'enrôle *pas* automatiquement les nouvelles transactions. Dans ce cas, vous pouvez désactiver l'enrôlement automatique des transactions pour `AseConnection`, puis enrôler `AseConnection` dans des transactions à l'aide de `EnlistDistributedTransaction`.

Avertissement ! `EnlistDistributedTransaction` renvoie une exception si `AseConnection` a déjà entamé une transaction à l'aide de `BeginTransaction` ou en exécutant l'instruction `BEGIN TRANSACTION` de manière explicite avec `AseCommand`.

Nouvelles fonctionnalités d'ESD#8

ESD#8 propose les nouvelles fonctionnalités suivantes :

Prise en charge étendue de MIT Kerberos

MIT Kerberos 5 version 1.3.6 est désormais pris en charge sur les plates-formes suivantes :

- IBM AIX 64 bits
- HP-UX 11.11 (ou HP-UX 11iv1.0) 64 bits

Sybase prend actuellement en charge les versions suivantes de MIT Kerberos sur les plates-formes ci-dessous :

- Version 1.3.1 sur Linux Intel 32 bits et Sun Solaris 8 (SPARC 32 bits et 64 bits)
- Version 1.2.7 sur Linux AMD64 (Opteron)/EM64T

Prise en charge pour de nouveaux systèmes d'exploitation et plates-formes

Le [Tableau 10](#) répertorie les nouveaux systèmes d'exploitation, plates-formes et fonctionnalités pris en charge pour les produits Sybase SDK 12.5.1 et Open Server 12.5.1.

Tableau 10 : Nouvelles plates-formes de SDK et Open Server 12.5.1 et fonctionnalités prises en charge

Plate-forme	Niveau du système d'exploitation	Compilateurs C et C++	Compilateurs COBOL	Version de Kerberos	Light-weight Directory Access (LDAP)	Secure Sockets Layer (SSL)
Linux on POWER 32 bits	Red Hat Enterprise Linux AS 3.0	IBM XL C/C++ Advance Edition V7.0	Non disponible	Non disponible	Non disponible	Non disponible
Linux on POWER 64 bits	Red Hat Enterprise Linux AS 3.0	IBM XL C/C++ Advance Edition V7.0	Non disponible	Non disponible	Non disponible	Non disponible

Plate-forme	Niveau du système d'exploitation	Compilateurs C et C++	Compilateurs COBOL	Version de Kerberos	Light-weight Directory Access (LDAP)	Secure Sockets Layer (SSL)
Linux AMD64 (Opteron) /EM64T	Red Hat Enterprise Linux AS 3.0	GCC 3.2.3 (Red Hat Linux 3.2.3 - 42)	Non disponible	MIT 1.2.7	Non disponible	Non disponible
Sun x86 32 bits	Solaris 9	Sun C/C++ 6.2	Non disponible	Non disponible	Non disponible	Non disponible

Fonctionnalité LDAP sur IBM AIX 64 bits

LDAP (OpenLDAP) est désormais disponible sur IBM AIX 64 bits. Vous pouvez utiliser le service de répertoire LDAP pour créer, modifier et rechercher des informations depuis les entités du réseau.

Pour activer OpenLDAP, modifiez le fichier de configuration, *libtcl64.cfg*, disponible dans le répertoire *\$SYBASE/\$SYBASE_OCS/config*, puis ajoutez un serveur LDAP au service de répertoire configuré. Pour ce faire, suivez la procédure ci-dessous.

❖ Pour activer le service de répertoire OpenLDAP

- 1 Ajoutez la variable d'environnement de votre plate-forme à la bibliothèque LDAP. Par exemple :

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

- 2 Sélectionnez un serveur LDAP dans la section [DIRECTORY] de *libtcl64.cfg*, puis ajoutez une nouvelle paire de valeur de mot-clé :

Remarque L'URL LDAP doit tenir sur une seule ligne.

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

Par exemple :

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

❖ **Pour ajouter un serveur LDAP au service de répertoire OpenLDAP configuré**

- 1 Lancez l'utilitaire dsedit ou dscp dans le répertoire `$$SYBASE/$SYBASE_OCS/bin`.
- 2 Fournissez des détails sur le serveur, tels que server name, ainsi que sur le transport réseau (transport type, host name et port number, par exemple).

Mise à niveau de CyberSafe Kerberos Client sur Windows

Sybase prend désormais en charge CyberSafe ActiveTRUST Secure Client (Kerberos) 4.0 sur Microsoft Windows (Windows NT, Windows 2000, Windows 2003 et Windows XP).

CyberSafe Kerberos est pris en charge par les produits suivants :

- Composants du Software Developer's Kit :
 - Open Client (Client-Library)
 - ESQ/C
 - ESQ/COBOL
- Open Server

Correctifs de sécurité SSLPlus 3.1.x

Les versions suivantes de Secure Sockets Layer (SSL) ont corrigé certains bugs SSLPlus et fuites de mémoire :

- SSL Plus 3.1.14c sur Sun Solaris 8 (SPARC 32 bits et 64 bits), HP-UX 11.11 (32 bits et 64 bits), HP Tru64, IBM AIX (32 bits et 64 bits) et Linux Intel 32 bits. Cette version est disponible avec ESD#7 et les versions ultérieures.
- SSL Plus 3.1.15b sur Microsoft Windows (Windows NT 32-bit, Windows 2000, Windows 2003 et Windows XP). Cette version est disponible avec ESD#8 et les versions ultérieures.

Les versions ci-dessus de SSLPlus 3.1.x sont liées au filtre Sybase SSL, `libfssl.*`.

Mise à niveau du compilateur COBOL sur Windows 2000

Le compilateur COBOL a été mis à niveau vers Micro Focus Net Express 4.0 sur Microsoft Windows 2000.

Nouvelles fonctionnalités d'ESD#7

ESD#7 propose les nouvelles fonctionnalités suivantes :

Prise en charge de XA 64 bits étendue à de nouvelles plates-formes

L'interface Extended Architecture (XA) 64 bits, qui est déjà disponible sur Solaris 64 bits, a été étendue aux plates-formes suivantes :

- IBM AIX 64 bits
- HP-UX 11.11 (ou HP-UX 11iv1.0) 64 bits
- HP Itanium 64 bits

Sybase prend actuellement en charge XA 32 bits sur les plates-formes 32 bits suivantes :

- IBM AIX 32 bits
- HP-UX 11.11 (ou HP-UX 11iv1.0) 32 bits
- Sun Solaris 8 (SPARC 32 bits)
- Microsoft Windows (Windows NT, Windows 2000, Windows 2003 et Windows XP)

Présentation de l'interface XA

L'interface Distributed Transaction Manager (DTM) XA est l'implémentation Sybase de la norme d'interface XA pour Adaptive Server. L'interface DTM XA est l'un des éléments du modèle X/Open Distributed Transaction Processing (DTP), qui sert de norme industrielle pour le développement d'applications DTP. L'interface XA accède à des données stockées sur des Adaptive Servers depuis un gestionnaire de transactions CICS, Encina ou TUXEDO.

Composants de l'interface XA

L'interface XA de Sybase est constituée des composants suivants :

- L'interface Sybase DTM XA, qui est l'implémentation Sybase de l'interface XA pour Adaptive Server. Il s'agit d'une fonctionnalité Adaptive Server disponible sous licence séparée.
- Adaptive Server et la fonctionnalité DTM. L'installation du logiciel et les licences de fonctionnalités sont décrites dans le *Guide d'installation d'Adaptive Server Enterprise* pour votre plate-forme.
- Sybase Open Client, qui permet aux appels Client-Library d'être intégrés dans l'interface native entre votre application et le gestionnaire de ressources.
- Le fichier de configuration XA, qui contient des entrées définissant des connexions client/serveur conçues pour être utilisées avec XA.
- Embedded SQL/C et Embedded SQL/COBOL, qui créent des appels ESQL dans le cadre de l'interface native entre votre application et le gestionnaire de ressources.
- Un ensemble de commandes dbcc spécifiques à XA qui permettent aux administrateurs systèmes de gérer des transactions heuristiques.
- Des fichiers de configuration et des commandes spécifiques au gestionnaire de transactions que vous pouvez utiliser pour la reprise globale.

Pour plus d'informations sur l'utilisation des fonctionnalités Adaptive Server DTM natives, reportez-vous au guide *XA Interface Integration Guide* pour CICS, Encina et TUXEDO.

Nouvelles fonctionnalités d'ESD#6

ESD#6 propose la nouvelle fonctionnalité suivante :

Nouvel ASE OLE DB Provider by Sybase

Cette version de SDK 12.5.1 introduit un nouvel OLE DB Provider for Sybase ASE appelé « Adaptive Server Enterprise OLE DB Provider by Sybase ». OLE DB est un modèle d'accès aux données de Microsoft. Il utilise l'interface Component Object Model (COM) et, contrairement à l'ODBC, ne part pas du principe que la source de données utilise un programme de traitement des requêtes SQL.

Plates-formes supportées

L'ASE OLE DB Provider est conçu pour fonctionner avec OLE DB 2.5 et les versions ultérieures. Les plates-formes prises en charge sont notamment Windows NT 4.0, 2000, XP et 2003.

Fonctionnalités ASE prises en charge

L'Adaptive Server Enterprise OLE DB Provider prend en charge les fonctionnalités ASE suivantes :

- SSL
- Services de répertoire (LDAP)
- Cryptage du mot de passe

Préparation de la migration

Cet ASE OLE DB Provider, développé par Sybase, est destiné à remplacer l'OLE DB Provider existant dans une future version. A l'heure actuelle, les deux fournisseurs sont livrés et peuvent coexister sur la même machine. Il vous est conseillé de commencer à migrer et à tester vos applications avec le nouveau fournisseur afin d'identifier tous les problèmes de compatibilité. L'OLE DB Provider existant (nom court du fournisseur : « Sybase.ASEOLEDBProvider ») est installé dans le répertoire %SYBASE%\OLEDB et sa version actuelle est la 2.70. Le nouvel ASE OLE DB Provider (nom court du fournisseur : « ASEOLEDB ») est installé dans le répertoire %SYBASE%\DataAccess\OLEDB et la version disponible est la 12.5.1.384 ou une version supérieure.

Pour plus d'informations, reportez-vous au *Guide de l'utilisateur* pour Adaptive Server Enterprise OLE DB Provider by Sybase.

Nouvelles fonctionnalités d'ESD#5

ESD#5 propose les nouvelles fonctionnalités suivantes :

Colonnes cryptées BCP

A l'heure actuelle, les données sont récupérées en texte clair lorsque BCP est exécuté sur une table qui prend en charge des colonnes cryptées et l'utilisateur est autorisé à visualiser les données dans les colonnes.

La nouvelle option de ligne de commande BCP (-C) autorise le déplacement bulk de données cryptées par des utilisateurs autorisés. Pour cela, vous devez utiliser l'option -C et le tableau situé sur le serveur doit prendre en charge les colonnes cryptées. Vous obtenez ainsi une commande SQL, `set ciphertext=on`, qui doit être envoyée au serveur avant d'initier des routines de bibliothèque de `bulkcopy` pour produire des données cryptées.

Remarque Si la nouvelle option -C est définie mais que le serveur ne prend pas en charge les colonnes cryptées, aucune erreur n'est générée. A la place, BCP envoie la commande SQL sur la base de la présence d'une table, `sysencryptkeys`. Si une commande SQL échoue alors que cette table est présente, l'erreur est considérée comme fatale.

Amélioration de la prise en charge des caractères coréens

Open Client et Open Server prennent désormais en charge le jeu de caractères coréen `cp949`. Des répertoires ont été ajoutés au répertoire `charsets`, les fichiers de table de ces jeux de caractères ont été ajoutés au répertoire `utf8 charset` et les fichiers `utf8.cfg`, `locales.dat` et `objectid.dat` ont été modifiés pour inclure ces jeux de caractères. Ce jeu de caractères peut être utilisé de la même façon que tout autre jeu et voir ses données de caractères converties via la commande `cs_convert()`.

Correctifs de sécurité SSLPlus 3.1.10H

Secure Sockets Layer (SSL) SSLPlus 3.1.10H, la dernière mise à jour de la version 3.1x, est disponible avec l'ESD#5 et les versions ultérieures. Cette mise à jour corrige certains bugs SSLPlus et fuites de mémoire. SSLPlus 3.1.10H est lié au filtre Sybase SSL *libfssl.**, sur Sun Solaris 8 (SPARC 32 bits et 64 bits), Linux Intel 32 bits, IBM AIX 64 bits et Microsoft Windows NT.

Prise en charge de MIT Kerberos sur ODBC et OLE DB

Depuis ODBC 5.00.0034 et OLE DB 2.70.0041, le Sybase ODBC Driver et l'OLE DB Provider prennent en charge MIT Kerberos, en plus de CyberSafe Kerberos. Reportez-vous aux fichiers d'aide fournis avec les produits pour plus d'informations sur l'utilisation de cette fonctionnalité.

❖ **Pour localiser les informations de configuration de Kerberos pour Sybase ASE ODBC Driver et Sybase ASE OLE DB Provider**

- 1 Lancez l'outil ODBC Data Source Administrator (*ODBCAD32.EXE*).
- 2 Sélectionnez votre source de données et cliquez sur Configurer. (Si vous utilisez uniquement OLE DB Provider, cliquez sur Add, puis sélectionnez Sybase ASE ODBC Driver.)
- 3 Allez dans Connection Panel et cliquez sur Help.
- 4 Lisez la section « Use Kerberos ». Elle propose des informations tant pour CyberSafe Kerberos que pour MIT Kerberos.

Installation de SDK 12.5.1 sur Windows

Si vous mettez à niveau votre système et installez SDK 12.5.1 dans un ancien répertoire SDK 12.5 Sybase dans lequel le pilote ODBC 12.5 était installé (*%SYBASE%\odbc*), le programme d'installation met à jour le pilote ODBC, de sorte que vous pouvez y avoir directement accès.

Remarque Sybase recommande d'utiliser le répertoire du pilote ODBC d'origine.

Si vous choisissez d'installer la version 12.5.1 dans un nouveau répertoire sur une machine qui contient l'ancien pilote SDK 12.5 ODBC (dans l'ancien répertoire Sybase pour ODBC), vous n'aurez pas accès au nouveau pilote ODBC. Les sources de données utilisées lors de l'installation stockent une copie de sauvegarde du chemin d'accès absolu conduisant au répertoire dans lequel se trouve *SYODASE.DLL*. Il n'existe pas d'utilitaire de mise à jour du registre Windows afin de permettre aux DSN du pilote ODBC de mettre à jour la référence au fichier *SYODASE.DLL*. Si vous changez l'emplacement du pilote ODBC, vous devez soit copier *SYODASE.DLL* et ses *DLL* associés, de même que les composants, dans l'ancien répertoire Sybase, soit modifier tous vos DSN pour qu'ils utilisent le nouvel emplacement du pilote.

Prise en charge étendue des plates-formes pour l'ASE ODBC Driver by Sybase

L'interface ODBC est une interface de programmation d'applications basée sur les appels définie par Microsoft Corporation en tant qu'interface standard des systèmes de gestion des bases de données. Le pilote Adaptive Server Enterprise ODBC 12.5.1 vous permet d'écrire des programmes capables d'accéder à des données dans Adaptive Server à l'aide de l'interface d'appel ODBC. Outre les fonctionnalités ODBC de niveau 2, le pilote prend en charge les fonctionnalités suivantes :

- SSL,
- reprise haute disponibilité,
- cryptage du mot de passe.

Cette version de SDK inclut la prise en charge de plates-formes supplémentaires pour l'ASE ODBC Driver by Sybase. Le pilote a été introduit dans l'ESD#2 de DK 12.5.1 pour la plate-forme Linux et cette version ajoute la prise en charge des systèmes d'exploitation Windows et Mac OS X.

Remarque Ce pilote, développé par Sybase, est destiné à remplacer l'ODBC Driver Kit existant dans une future version. A l'heure actuelle, les deux pilotes sont fournis en parallèle et peuvent coexister sur la même machine. Il vous est conseillé de commencer à migrer et à tester vos applications avec le nouveau pilote afin d'identifier tous les problèmes de compatibilité. L'ODBC Driver Kit existant est enregistré avec l'ODBC Driver Manager sous « Sybase ASE ODBC Driver » et est actuellement à la version 5.00.0034. L'ASE ODBC Driver by Sybase est enregistré sous « Adaptive Server Enterprise » et en est à la version 12.5.1.376.

Nouvelles fonctionnalités d'ESD#4

ESD#4 propose les nouvelles fonctionnalités suivantes :

Amélioration du diagnostic

La routine `ct_debug` de Client-Library a été révisée et vous permet de tracer et de sauvegarder le Tabular Data Stream™ (TDS) dans un fichier sans qu'il soit nécessaire de télécharger, installer et configurer la capture ou d'utiliser Ribo (outil de trace TDS). Les modifications suivantes ont été apportées aux paramètres de la routine `ct_debug` :

- Pour activer `CS_DBG_PROTOCOL` ou pour configurer `CS_DBG_PROTOCOL_FILE`, une application n'a plus besoin d'utiliser les bibliothèques *devlib* (débogage).
- Si vous ne définissez pas `CS_DBG_PROTOCOL_FILE` pour une connexion, `mktemp` est appelé afin de générer un nom de fichier unique destiné à contenir la sauvegarde des paquets de protocole. La chaîne de préfixe transmise à `mktemp` est « `captur` » et le fichier de protocole ainsi obtenu peut être décodé par Ribo.
- Il est possible d'activer la fonctionnalité de sauvegarde du protocole à l'aide du fichier *ocs.cfg*. Dans ce cas, les bibliothèques normale et *devlib* créent toutes deux des fichiers de sauvegarde du protocole.

Prise en charge des communications asynchrones

Il s'agit d'une nouvelle fonctionnalité de Sybase ASE ODBC Driver 5.00.0022 et de Sybase ASE OLE DB Provider 2.70.0038. Ces produits prennent désormais en charge les communications asynchrones telles que définies dans la spécification ODBC 3.5.

Prise en charge asynchrone du niveau de connexion

Si la commande Cursor Mode (Select Method) du pilote ODBC TDS est définie sur Cursor, le pilote indique via `SQLGetInfo` qu'il prend en charge le support asynchrone du niveau de connexion. L'appel de `SQLGetInfo` avec le type d'information `SQL_ASYNC_MODE` renvoie `SQL_AM_CONNECTION`.

Prise en charge asynchrone de niveau instruction

Si la commande Cursor Mode du pilote ODBC TDS est définie sur Direct, le pilote indique via `SQLGetInfo` qu'il prend en charge le support asynchrone de niveau instruction. L'appel de `SQLGetInfo` avec le type d'information `SQL_ASYNC_MODE` renvoie `SQL_AM_STATEMENT`.

Nombre maximum d'instructions asynchrones simultanées	Le pilote ODBC TDS indique via SQLGetInfo une limite spécifique de un (1) au nombre d'instructions asynchrones simultanées. L'appel de SQLGetInfo avec le type d'information SQL_MAX_ASYNC_CONCURRENT_STATEMENTS renvoie un (1).
Activation de la prise en charge des communications synchrones	<p>Si la commande cursor mode du pilote ODBC TDS est définie sur Cursor, le pilote TDS active la prise en charge des communications asynchrones en appelant SQLSetConnectAttr. L'appel de SQLSetConnectAttr avec la valeur d'attribut SQL_ATTR_ASYNC_ENABLE définie sur SQL_ASYNC_ENABLE_ON active la prise en charge asynchrone. L'appel de SQLSetConnectAttr avec la valeur d'attribut SQL_ATTR_ASYNC_ENABLE définie sur SQL_ASYNC_ENABLE_OFF désactive la prise en charge asynchrone.</p> <p>Si la commande cursor mode du pilote TDS est définie sur Direct, le pilote TDS active la prise en charge des communications asynchrones en appelant SQLSetStmtAttr. L'appel de SQLSetStmtAttr avec la valeur d'attribut SQL_ATTR_ASYNC_ENABLE définie sur SQL_ASYNC_ENABLE_ON active la prise en charge asynchrone. L'appel de SQLSetStmtAttr avec la valeur d'attribut SQL_ATTR_ASYNC_ENABLE définie sur SQL_ASYNC_ENABLE_OFF désactive la prise en charge asynchrone.</p>
Prise en charge de la fonction ODBC asynchrone	<p>Les fonctions ODBC suivantes peuvent être prises en charge de manière asynchrone :</p> <ul style="list-style-type: none">SQLBulkOperationsSQLColAttributeSQLColumnPrivilegesSQLColumnsSQLCopyDescSQLDescribeColSQLDescribeParamSQLExecDirectSQLExecuteSQLFetchSQLFetchScrollSQLForeignKeysSQLGetDataSQLGetDescFieldSQLGetDescRecSQLGetDiagFieldSQLGetDiagRec

SQLGetTypeInfo
 SQLMoreResults
 SQLNumParams
 SQLNumResultCols
 SQLParamData
 SQLPrepare
 SQLPrimaryKeys
 SQLProcedureColumns
 SQLProcedures
 SQLPutData
 SQLSetPos
 SQLSpecialColumns
 SQLStatistics
 SQLTablePrivileges
 SQLTables

Lorsqu'une de ces fonctions sont exécutées en mode asynchrone, elle renvoie un code de retour `SQL_STILL_EXECUTING`. Une fois que la fonction en question se termine, elle renvoie le code qu'elle aurait renvoyé si elle avait exécutée en mode synchrone (par exemple, `SQL_SUCCESS`, `SQL_ERROR` ou `SQL_NEED_DATA`).

Changements au
niveau du traitement

Une fois que la communication asynchrone a été activée, le pilote TDS exécute les fonctions spécifiées ci-dessus de manière asynchrone, comme décrit dans la spécification ODBC 3.5. Une fois que la communication asynchrone est désactivée, toutes les fonctions sont exécutées en mode synchrone. Tous les autres traitements se déroulent normalement.

Nouvelle variable de configuration de DB-Library

Cette nouvelle fonctionnalité utilise une variable d'environnement pour configurer au niveau externe le niveau de la version de DB-Library au moment de l'exécution afin de modifier le code de l'application en appelant `dbsetversion`. Elle permet de :

- retrouver la variable d'environnement lors de la phase d'initialisation de DB-Library ;
- stocker la valeur de la variable d'environnement en tant que niveau de la version.

A l'heure actuelle, DB-Library définit dans le code source le niveau de la version par défaut sur `DBVERSION_46`. La seule façon de changer ce niveau est d'appeler `dbsetversion` dans une application. Si un utilisateur souhaite utiliser les fonctionnalités proposées par le protocole TDS version 5.0 (par exemple, les types `numeric` et `decimal`), le niveau de version par défaut codé en dur (`DBVERSION_46`) doit être changé. Auparavant, les utilisateurs devaient ajouter l'appel `dbsetversion(DBVERSION_100)` à leur code. Cette fonctionnalité vous permet de choisir le niveau de version de DB-Library sans appeler `dbsetversion` depuis une application.

Définitions

Les définitions suivantes sont utilisées lors de la description de la nouvelle variable de configuration DB-Library :

- DB-Library – collection de routines et de macros API de programmation C qui permet à une application d'interagir avec le serveur de données et des applications Open Server.
- CT-Library – collection de routines API de programmation C utilisée lors de l'écriture d'applications clientes.
- Niveau de version – paramètre de DB-Library utilisé pour spécifier la version du protocole TDS. Les trois valeurs possibles sont les suivantes :
 - `DBVERSION_46` – DB-Library est exécuté avec le protocole TDS version 4.6.
 - `DBVERSION_100` – DB-Library est exécuté avec le protocole TDS version 5.0.
 - `DBVERSION_UNKNOWN` – le niveau de la version n'a pas encore été décidé.
- `dbsetversion` – API de DB-Library qui définit le niveau de version de DB-Library dans l'application cliente.
- `dbinit` – API qui initialise DB-Library. Elle doit être appelée avant tout autre routine DB-Library.

Configuration

Pour pouvoir implémenter cette fonctionnalité, vous devez définir une nouvelle variable d'environnement `SYBOCS_DBVERSION` afin de permettre la configuration externe du niveau de version de DB-Library.

Les valeurs possibles de cette variable d'environnement sont les suivantes :

- `DBVERSION_46`
- `DBVERSION_100`

Toute autre valeur provoque l'échec de l'application DB-Library.

Ordre de traitement

Les paragraphes suivants expliquent l'ordre de traitement de la variable d'environnement et de `dbsetversion` dans une application DB-Library :

- Si aucune variable d'environnement n'est définie et que `dbsetversion` n'est pas appelé dans une application, le niveau de la version est `DBVERSION_46`, qui est actuellement la version par défaut dans le code source de DB_Library.
- Si aucune variable d'environnement n'est définie et que `dbsetversion` est appelé avec la valeur `DBVERSION_100` ou `DBVERSION_46`, le niveau de la version sera `DBVERSION_100` ou `DBVERSION_46`, respectivement.
- Si une variable d'environnement est définie en tant que `DBVERSION_100` ou `DBVERSION_46` et que `dbsetversion` n'est pas appelé dans une application, le niveau de la version sera `DBVERSION_100` ou `DBVERSION_46`, respectivement.
- Si une variable d'environnement est définie en tant que `DBVERSION_100` et que `dbsetversion` est appelé pour définir le niveau de version sur `DBVERSION_46` dans l'application, le niveau de version sera `DBVERSION_46`, dans la mesure où `dbsetversion` annule la variable d'environnement et décide du niveau final de la version.

Pilote Adaptive Server Enterprise ODBC pour Linux Intel 32 bits

Cette version de SDK inclut le pilote Open Database Connectivity (ODBC). L'interface ODBC est une interface de programmation d'applications basée sur les appels définie par Microsoft Corporation en tant qu'interface standard des systèmes de gestion des bases de données. Le pilote Adaptive Server Enterprise ODBC 12.5.1 vous permet d'écrire des programmes capables d'accéder à des données dans Adaptive Server à l'aide de l'interface d'appel ODBC. Outre les fonctionnalités ODBC de niveau 2, le pilote prend en charge les fonctionnalités suivantes :

- SSL
- Reprise haute disponibilité (HA)
- Cryptage du mot de passe

Nouveau pilote ASE ODBC

Remarque La version du pilote ASE ODBC 5.0 repose sur la spécification ODBC 3.52. Toute application écrite pour utiliser le pilote ASE ODBC 4.x fonctionnera avec la version 5.0 sans qu'aucune modification ne soit nécessaire.

La section suivante présente un résumé des nouvelles fonctionnalités proposées par le pilote ASE ODBC 5.0 :

- prise en charge des types de données date et time ;
- prise en charge du cryptage de mot de passe ;
- suppression des composants partagés BAS\UTL\FLT.

Nouvelles fonctionnalités d'ESD#3

ESD#3 propose la nouvelle fonctionnalité suivante :

Adaptive Server Enterprise ADO.NET Data Provider 1.1 sur Microsoft Windows

Cette version inclut ASE ADO.NET Data Provider 1.1, qui est un fournisseur ADO.NET pour la base de données Adaptive Server. Il vous permet d'accéder à des données d'Adaptive Server à l'aide de n'importe quel langage pris en charge par .NET, dont C#, Visual Basic .NET, C++ avec extensions gérées et J#. Pour accéder aux nouvelles fonctionnalités, vous devez passer à la version 1.1 comme décrit dans la section suivante

Mise à niveau vers ASE ADO.NET Data Provider 1.1

Le code octet Common Language Runtime (CLR) des applications reposant sur d'anciennes versions d'ASE ADO.NET Data Provider n'est pas compatible avec ASE ADO.NET Data Provider 1.1. Pour pouvoir utiliser la version mise à jour d'ASE ADO.NET Data Provider, vous devez reconstruire vos applications. Plusieurs versions d'ASE ADO.NET Data Provider peuvent coexister sur la machine client, de sorte que votre ancienne application peut continuer à utiliser la précédente version d'ASE ADO.NET Data Provider. Reportez-vous au *Guide de l'utilisateur* d'Adaptive Server Enterprise ADO.NET Data Provider pour plus d'informations.

Nouvelles fonctionnalités d'ASE ADO.NET Data Provider 1.1 sont :

Les nouvelles fonctionnalités d'ASE ADO.NET Data Provider 1.1 sont :

- Secure Sockets Layer (SSL),
- services de répertoire – Lightweight Directory Access Protocol (LDAP),
- reprise haute disponibilité (HA),
- cryptage du mot de passe,
- ASECommandBuilder,
- ASEcommandExecutXmlReader().

Pour plus d'informations sur ces nouvelles fonctionnalités, reportez-vous au *Guide de l'utilisateur* d'Adaptive Server Enterprise ADO.NET Data Provider.

Nouvelles fonctionnalités d'ESD#2

ESD#2 propose les nouvelles fonctionnalités suivantes :

Prise en charge de SSL Plus 3.1.5 + SBGSE sur Solaris et Microsoft Windows

Sybase prend désormais en charge Secure Sockets Layer (SSL) Plus 3.1.5 + SBGSE pour Solaris 32 bits et 64 bits et Microsoft Windows sur les produits suivants :

- Composants du Software Developer's Kit :
 - Client-Library
 - ESQL/C
 - ESQL/COBOL
- Open Server

En vertu des réglementations fédérale, Sybase est tenu d'utiliser les suites de chiffrement certifiées FIPS. Après être passé à SDK 12.5.1 ESD#2 (qui inclut Open Client) et Open Server 12.5.1 ESD#2 ou une version ultérieure, pour Microsoft Windows, vous devez initialiser les bibliothèques Certicom comme décrit dans la procédure ci-dessous avant de pouvoir effectuer des test SSL. Dans le cas de Solaris, vous devez vérifier que le répertoire *lib3p* se trouve dans le chemin d'accès à la bibliothèque de chargement dynamique.

❖ Initialisation des bibliothèques Certicom pour Windows

- 1 Pour permettre aux bibliothèques *ctlib* et *srvlib* de trouver les suites de chiffrement Certicom dans *libsb.dll*, ajoutez ce qui suit au chemin d'accès à la bibliothèque de chargement dynamique :

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

- 2 Exécutez le programme suivant :

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

Ce programme ajoute un élément au registre système utilisé par *libsb.dll* pour effectuer un auto-test. Une fois que la valeur figure dans le registre d'une machine donnée, vous ne devez plus réexécuter le programme *setsslreg.exe*, à moins que la valeur ne soit supprimée pour une raison ou une autre.

3 Exécutez le programme suivant :

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

Ce programme vérifie que l'entrée de registre et *libsb.dll* sont correctement configurés (vérifiez que *lib3p* se trouve dans le chemin d'accès au processus). *sbgtest.exe* est utilisé pour confirmer que *libsb.dll* peut accéder à l'entrée de registre et effectuer son auto-test.

Il est possible que vous deviez utiliser *sbgtest.exe* plusieurs fois dans les situations suivantes :

- après avoir exécuté *setsslreg*, pour confirmer l'existence de l'entrée de registre ;
- après avoir connu une panne totale à l'aide de SSL, pour vérifier que *libsb.dll* fonctionne toujours.

Remarque Lorsque vous exécutez une application utilisant SSL, il est très important que le répertoire *lib3p* se trouve dans le chemin d'accès à la bibliothèque de chargement dynamique.

Prise en charge de IPV6 sur Solaris et Microsoft Windows XP et 2003

Sybase prend désormais en charge IPV6 pour Solaris 32 bits et 64 bits et Microsoft Windows XP et Windows 2003 sur les produits suivants :

- Composants du Software Developer's Kit :
 - Client-Library
 - ESQL/C
 - ESQL/COBOL
- Open Server

Open Client et Open Server utilisent tous deux `getaddrinfo()` pour convertir le nom d'hôte en adresses comme suit :

- Open Client tente d'établir une connexion à un serveur en utilisant chacune des adresses renvoyées par `getaddrinfo()` jusqu'à ce que `connect()` réussisse.
- Open Server tente d'établir une connexion à un serveur selon que les adresses IPV6 sont renvoyées par `getaddrinfo()` et se comporte comme suit :

- Si les adresses IPV6 ne sont *pas* renvoyées par `getaddrinfo()`, Open Server établit un port d'écoute pour chaque adresse renvoyée par `getaddrinfo()`.
- Si les adresses IPV6 sont renvoyées par `getaddrinfo()`, Open Server crée un caractère joker pour l'adresse du port d'écoute. Si Open Server écoute uniquement un sous-ensemble des adresses IP et IPV6 valides d'un hôte, les adresses réelles à utiliser doivent être entrées dans des lignes distinctes du fichier *interfaces*.

Remarque Du fait de la définition dans la spécification IPV6, Open Server établit un port d'écoute tant pour l'hôte local IPV6 que pour l'hôte local IP s'il détermine que `getaddrinfo()` a renvoyé l'adresse de bouclage.

Fonctionnalité LDAP de DB-Library sur Microsoft Windows

Cette fonctionnalité permet aux applications DB-Library (*dblib*) invoquant `dbopen` de se connecter à un Adaptive Server ou un Open Server en utilisant les services de répertoire LDAP. Pour ce faire, vous devez modifier le fichier *libctl.cfg* dans la section [DIRECTORY], comme décrit dans le *Guide de configuration* d'Open Client et d'Open Server pour Microsoft Windows. Aucune modification n'est requise au niveau de l'application pour utiliser cette fonctionnalité.

La possibilité d'accéder au LDAP existe déjà dans les produits Open Client et Open Server et le code d'accès aux services de répertoire génériques se trouve dans *netlib*. Par conséquent, les applications ont uniquement besoin de se connecter à *netlib*.

Remarque Dans la mesure où Sybase utilise *netlib* pour l'E/S, il est possible que des messages d'erreur *netlib* soient générés à la place des messages actuels produits par DB-Library.

Nouvelles fonctionnalités d'ESD#1

ESD#1 propose la nouvelle fonctionnalité suivante :

Prise en charge de XA sur Sun Solaris 8 (SPARC 64 bits)

XA 64 bits est désormais disponible sur Sun Solaris 8 (SPARC 64 bits). Bien que XA 64 bits ne soit pas pris en charge sur d'autres plates-formes 64 bits, Sybase supporte actuellement XA 32 bits sur les plates-formes 32 bits suivantes :

- IBM AIX 32 bits
- HP-UX (ou HP-UX 11iv1.0) 32 bits
- Sun Solaris 8 (SPARC 32 bits)
- Microsoft Windows

Présentation de l'interface XA

L'interface DTM XA est l'implémentation Sybase de la norme d'interface XA pour Adaptive Server. L'interface DTM XA est l'un des éléments du modèle X/Open Distributed Transaction Processing (DTP), qui sert de norme industrielle pour le développement d'applications DTP. L'interface XA accède à des données stockées sur des Adaptive Servers depuis un gestionnaire de transactions CICS, Encina ou TUXEDO.

Composants de l'interface XA

L'interface XA de Sybase est constituée des composants suivants :

- L'interface Sybase DTM XA, qui est l'implémentation Sybase de l'interface XA pour Adaptive Server. Il s'agit d'une fonctionnalité Adaptive Server disponible sous licence séparée.
- Adaptive Server et la fonctionnalité DTM. L'installation du logiciel et les licences de fonctionnalités sont décrites dans le *Guide d'installation d'Adaptive Server Enterprise* pour votre plate-forme.
- Sybase Open Client, qui permet aux appels Client-Library d'être intégrés dans l'interface native entre votre application et le gestionnaire de ressources.

- Le fichier de configuration XA, qui contient des entrées définissant des connexions client/serveur conçues pour être utilisées avec XA.
- Embedded SQL/C et Embedded SQL/COBOL, qui créent des appels ESQL dans le cadre de l'interface native entre votre application et le gestionnaire de ressources.
- Un ensemble de commandes dbcc spécifiques à XA qui permettent aux administrateurs systèmes de gérer des transactions heuristiques.
- Des fichiers de configuration et des commandes spécifiques au gestionnaire de transactions que vous pouvez utiliser pour la reprise globale.

Pour plus d'informations sur l'utilisation des fonctionnalités Adaptive Server DTM natives, reportez-vous au guide *XA Interface Integration Guide* pour CICS, Encina et TUXEDO.