

SYBASE®

新機能ガイド

Adaptive Server® Enterprise

15.5

ドキュメント ID：DC01373-01-1550-02

改訂：2010年6月

Copyright © 2010 by Sybase, Inc. All rights reserved.

このマニュアルは Sybase ソフトウェアの付属マニュアルであり、新しいマニュアルまたはテクニカル・ノートで特に示されないかぎり、後続のリリースにも付属します。このマニュアルの内容は予告なしに変更されることがあります。このマニュアルに記載されているソフトウェアはライセンス契約に基づいて提供されるものであり、無断で使用することはできません。

このマニュアルの内容を弊社の書面による事前許可を得ずに、電子的、機械的、手作業、光学的、またはその他のいかなる手段によっても、複製、転載、翻訳することを禁じます。

Sybase の商標は、Sybase の商標リスト (<http://www.sybase.com/detail?id=1011207>) で確認できます。Sybase およびこのリストに掲載されている商標は、米国法人 Sybase, Inc. の商標です。® は、米国における登録商標であることを示します。

Java および Java 関連の商標は、米国およびその他の国における Sun Microsystems, Inc. の商標または登録商標です。

Unicode と Unicode のロゴは、Unicode, Inc. の登録商標です。

IBM および Tivoli は、International Business Machines Corporation の米国およびその他の国における登録商標です。

このマニュアルに記載されている上記以外の社名および製品名は、当該各社の商標または登録商標の場合があります。

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

目次

Adaptive Server バージョン 15.5 Cluster Edition の新機能	1
Adaptive Server 15.5 Cluster Edition の機能とプラットフォームの一覧.....	1
複数の同時発生フェールオーバー.....	3
アーカイブ・データベースへの領域の追加.....	3
共有ディスク・クラスタでの分散トランザクション管理.....	3
Adaptive Server 15.5 Cluster Edition でのシステムの変更点	5
変更されたコマンド.....	5
モニタリング・テーブル.....	5
設定パラメータ.....	7
関数.....	7
Adaptive Server® バージョン 15.5 の新機能	9
Adaptive Server 15.5 の機能とプラットフォームの一覧.....	9
インメモリ・データベースおよびリラックス持続性データベース.....	11
バックアップの高速圧縮.....	13
IBM® Tivoli® Storage Manager 用の Backup Server サポート.....	13
ユーザ定義のストアド・プロシージャ用の遅延名前解決.....	14
FIPS 140-2 ログイン・パスワードの暗号化.....	14
増分データ転送.....	15
bigdatetime および bigtime データ型.....	16
tempdb グループの作成および管理.....	16
Adaptive Server 15.5 でのシステムの変更点	17

データ型	17
関数	18
システム・ストアド・プロシージャ	19
コマンド	21
設定パラメータ	24
モニタリング・テーブル	24
システム・テーブル	25
ユーティリティ	26
監査	26
Adaptive Server 15.0.3 の新機能	29
SQL 文の複写	29
セキュリティの機能強化	29
LDAPS ユーザ認証の強化	29
自動的な LDAP ユーザ認証とフェールバック	30
外部認証のログイン・マッピング	30
SSL を使用した共通名の指定	30
Kerberos による同時認証	31
仮想ハッシュ・テーブル	31
ヒュージ・ページ	32
高可用性設定時のアップグレード	33
システム・ストアド・プロシージャの再インストール	33
分散トランザクション管理 (DTM)	33
Adaptive Server プラグインの更新	34
Java インタフェース	35
Adaptive Server 15.0.3 でのシステムの変更点	37
関数	37
システム・ストアド・プロシージャ	37
コマンド	38
設定パラメータ	39
モニタリング・テーブル	39
システム・テーブル	40

Adaptive Server 15.0.2 の新機能	43
暗号化カラム	43
アーカイブ・データベースへのアクセス	44
長時間実行されているクエリの検出	45
遅延コンパイル	45
中国語と日本語文字セットの大文字と小文字を区別 しないソート順	46
統計集合関数	47
標準偏差と分散	47
積極的集約と消極的集約	48
ベクトル集合およびスカラ集合	48
データ挿入のパフォーマンスの向上	49
ページ分割時の非同期書き込みの使用	50
tempdb トランザクションのスループットの向 上	50
コミット後の最適化	51
クエリ・プロセッサの変更点	52
遅延コンパイル	52
バイナリ以外の文字セットのヒストグラム補間	52
選択性の見積もりのヒストグラム式	53
現在のオプティマイザ設定の表示	53
新しいセキュリティ機能	53
AIX における 64 ビット版 Adaptive Server の PAM サポート	53
グローバル・ログイン・トリガの自動設定	54
SSL サポート	54
パスワードのセキュリティの改善	55
監査機能の強化	55
高可用性に関する考慮事項	56
モニタリング・テーブルのインストールおよび編集 ..	56

ステートメント・キャッシュ用のモニタリング・テーブル	57
システム・テーブルのロー・レベルのロック	58
xmltable() 関数	59
再配置ジョイン	59
ユーザ定義の SQL 関数	60
instead of トリガ	60
Adaptive Server 15.0.2 でのシステムの変更点	63
トレース・フラグ	63
コマンド	63
set コマンドの変更	66
ユーティリティ	66
システム・ストアド・プロシージャ	67
システム・テーブル	71
設定パラメータ	75
関数	77
グローバル変数	77
Adaptive Server 15.0.1 の新機能	79
抽象プランの変更	79
新しいクエリ・レベル設定	79
抽象プランおよびオプティマイザ基準でのオペレータ名の配置	80
オプティマイザ基準 set 構文の拡張	81
リテラルのパラメータ化	81
Adaptive Server 15.0.1 でのシステムの変更点	83
関数	83
設定パラメータ	83
コマンド	85
モニタリング・テーブル	88
Adaptive Server 15.0 の新機能	89
パーティションのサポート	89
ロー・ロックのシステム・カタログ	90

クエリ・プロセッサ	90
長い識別子	91
計算カラム	92
計算カラムと関数ベース・インデックスの違い	92
実体化された計算カラムと実体化されていない 計算カラムの違い	92
スクロール可能カーソル	93
unitext データ型のサポート	93
big int データ型のサポート	94
符号なし整数値データ型のサポート	94
整数 identity	95
XML サービスの機能強化	95
Adaptive Server プラグインの機能強化	96
Interactive SQL	97
ユーザ定義 Web サービス	97
非常に大きな記憶域のサポート	98
update statistics の自動実行	98
SySAM ライセンス管理	99
クエリ処理測定基準 (QP 測定基準)	100
抽象プランの更新	101
showplan の変更	101
SSL での FIPS 140-2 の使用	102
Adaptive Server 15.0 でのシステムの変更点	105
ユーティリティ	105
予約語	106
グローバル変数	106
設定パラメータ	107
関数	109
コマンド	110
システム・ストアド・プロシージャ	112
システム・テーブル	113

モニタリング・テーブル	118
Adaptive Server 12.5.4 の新機能	119
Kerberos の機能強化	119
LDAP ユーザ認証の強化	119
複雑なパスワードの強化	119
アーカイブ・データベースへのアクセスのサポート	120
共有ディレクトリの変更	120
Sybase によるドライバのサポート	121
動的にロードされる TIBCO ライブラリ	121
JRE のサポート	122
Adaptive Server プラグインの変更	122
システム・カタログの更新	122
モニタリング・テーブルの変更	123
syscomments の変更	123
Windows Terminal Server 環境での共有メモリ・サポ ート	123
グローバル・ログイン・トリガ	123
ログイン・トリガからの set オプションのエクスポ ート	124
数値データ型のモジュロ算術	124
Adaptive Server 12.5.4 でのシステムの変更点	125
コマンド	125
設定パラメータ	125
関数	126
Transact-SQL コマンド	126
追加の説明や情報の入手	129
索引	131

Adaptive Server バージョン 15.5 Cluster Edition の新機能

Adaptive Server® 15.5 Cluster Edition では、複数の同時発生フェールオーバ、分散トランザクション管理、**mount** コマンドと **unmount** コマンド、**alter database** を使用してアーカイブ・データベースに領域を追加する機能が導入されています。

注意： 現在 Cluster Edition では、インメモリ・データベース、リラックス持続性データベース、テンプレート・データベース、または最低限のログを取る DML はサポートされていません。

Adaptive Server 15.5 Cluster Edition の機能とプラットフォームの一覧

次の機能とプラットフォームの一覧は、Adaptive Server 15.5 Cluster Edition でサポートされる各オペレーティング・システムで利用できる機能を示します。"Y" は、機能がそのプラットフォームでサポートされていることを示します。

Adaptive Server Cluster Edition のオプション	HP-UX Itanium 64 ビット版	IBM AIX 64 ビット版	Linux Opteron 64 ビット版	Solaris 64 ビット版
Security and Directory Services	Y	Y	Y	Y
Cybersafe Kerberos				Y
プラグ可能認証モジュール	Y	Y	Y	Y
詳細なアクセス制御	Y	Y	Y	Y
LDAP サーバ・ディレクトリ	Y	Y	Y	Y
LDAP ユーザ認証	Y	Y	Y	Y

Adaptive Server Cluster Edition のオプション	HP-UX Itanium 64 ビット版	IBM AIX 64 ビット版	Linux Opteron 64 ビット版	Solaris 64 ビット版
プラットフォーム・ネイティブ Kerberos				Y
SSL (Secure Sockets Layer)	Y	Y	Y	Y
MIT Kerberos	Y	Y	Y	Y
暗号化カラム (詳細なアクセス制御 (FGAC) を含む)	Y	Y	Y	Y
高可用性				
パーティション	Y	Y	Y	Y
インメモリ・データベース				
Backup Server 対応 Tivoli Storage Manager	Y	Y	Y	Y
Active Messaging	Y	Y	Y	Y
拡張型全文検索 (EFTS)				
<i>基本の Adaptive Server に組み込まれている機能</i>				
プラットフォーム間でのダンプとロード	Y	Y	Y	Y
Job Scheduler	Y	Y	Y	Y
ネイティブ XML	Y	Y	Y	Y
IPv6	Y	Y	Y	Y
Java オプション	Y	Y	Y	Y
Web Services	Y	Y	Y	Y
分散トランザクション管理	Y	Y	Y	Y
コンテンツ管理 (外部ファイルのサポート)	Y	Y	Y	Y
アーカイブ・データベースへのアクセス	Y	Y	Y	Y

複数の同時発生フェールオーバー

Adaptive Server Cluster Edition バージョン 15.5 以降では、複数の同時発生インスタンス・エラーがサポートされています。

注意： Adaptive Server バージョン 15.5 Cluster Edition では、Veritas VCS 上での複数の同時発生エラーはサポートされていません。

複数の同時発生エラーのサポートは、単一のクラスタ・ビュー内で複数のインスタンスがエラーになったときに提供され、その状況でもクラスタはオンラインのまま維持され、単一のインスタンスがエラーになったときと同じようにフェールオーバー・リカバリが行われます。

エラーの数は **cluster redundancy level** 設定パラメータの値を超えることはできません。データベース管理者は、このパラメータを使用してクラスタでのリカバリ可能な同時発生インスタンス・エラーの最大数を設定できます。

アーカイブ・データベースへの領域の追加

Adaptive Server Cluster Edition バージョン 15.5 以降では、アーカイブ・データベースがサポートされています。一般に、アーカイブ・データベースへのアクセスは、クラスタ Adaptive Server と非クラスタ Adaptive Server で同じです。どちらの環境でも、アーカイブ・データベースの領域が不足したら、**alter database** コマンドを使用してアーカイブ・データベースに領域を追加します。

クラスタ Adaptive Server では、アーカイブ・データベースを更新するノードから **alter database** を実行します。別のノードから **alter database** を実行した場合は、実際にアーカイブ・データベースを更新しているノードの番号とエラー・メッセージが出力されます。

共有ディスク・クラスタでの分散トランザクション管理

Adaptive Server バージョン 15.5 以降では、クラスタ・アーキテクチャでの分散トランザクション管理 (DTM) がサポートされています。

クラスタ Adaptive Server の特徴は次のとおりです。

- XA-Server のような追加のサービスなしにリソース・マネージャ (RM) として機能する場合は、X/Open XA プロトコルに完全準拠している。

- Adaptive Server のデータをリモート・プロシージャ・コール (RPC) とコンポーネント統合サービス (CIS) を介して更新するトランザクションに、一貫したコミットとロールバックを保証する。
- Adaptive Server Transaction Coordination (ASTC) メカニズムを使用している他の Adaptive Server インストールでコーディネートされた分散トランザクションの一部になることができる。
- ASTC メカニズムを使用して、複数の Adaptive Server インストール間の分散トランザクションをコーディネートできる。

注意： Cluster Edition は Microsoft 分散トランザクション・コーディネータ (MSDTC) 独自のプロトコルはサポートしていません。

一般に、分散トランザクションのユーザ・インタフェースは、Adaptive Server クラスタ環境と非クラスタ環境で同じです。非クラスタ Adaptive Server で DTM を使用しているアプリケーションは、クラスタ Adaptive Server で同じアプリケーションを使用できます。『Adaptive Server 分散トランザクション管理機能の使用』を参照してください。

Cluster Edition での分散トランザクションのユーザ・インタフェースは非クラスタ Adaptive Server のユーザ・インタフェースと同じですが、クラスタでの分散トランザクションのサポートでは、『Cluster ユーザーズ・ガイド』に記載のクラスタ固有の問題を考慮する必要があります。次に例を示します。

- クラスタをリソース・マネージャ (RM) として使用
- 非所有者インスタンスに対する要求
- インスタンス・エラーの処理
- ASTC でのトランザクション・コーディネーションの使用
- 接続マイグレーションの影響
- システム設定

Adaptive Server 15.5 Cluster Edition でのシステムの変更点

Adaptive Server 15.5 Cluster Edition では、コマンド、モニタリング・テーブル、設定パラメータが変更されています。

変更されたコマンド

Adaptive Server Cluster Edition バージョン 15.5 以降では、共有ディスク・クラスタで **mount database** および **unmount database** を使用できます。

mount database または **unmount database** が進行中にインスタンスでエラーが発生した場合は、コマンドがアボートされる可能性があります。この場合、インスタンスのフェールオーバー・リカバリが完了したら、**mount database** または **unmount database** を再発行する必要があります。

モニタリング・テーブル

Adaptive Server バージョン 15.5 では、モニタリング・テーブルが追加されています。

Adaptive Server Cluster Edition バージョン 15.5 で追加されているモニタリング・テーブルは次のとおりです。

- **monTableTransfer** — Adaptive Server のアクティブなメモリに存在するテーブルの転送履歴情報を提供する。
- **monInmemoryStorage** — インメモリ・データベース用として使用される。内部目的でのみ使用。

Cluster Edition バージョン 15.0.1 以降では、モニタリング・テーブルを使用してテーブル統計情報が収集されます。バージョン 15.5 に用意されているモニタリング・テーブルは次のとおりです。

- **monCIPC** — 現在のインスタンス、またはすべてのインスタンスを基準にした、クラスタ内でのメッセージング全体を要約した数字を示します。
- **monCIPCendpoints** — クラスタ・インスタンス内にある各サブシステムのトラフィック・データを含む詳細な一覧を示します。
- **monCIPClinks** — クラスタ内のインスタンスの間のリンクの状態をモニタします。

- **monCIPCMesh** — インスタンスごとにその接続網、つまり現在のインスタンスから同じクラスタ内にある他のすべてのインスタンスへの接続の合計の数字を示します。
- **monCLMObjectActivity** — クラスタ・ロック情報を収集します。
- **monClusterCacheManager** — 各インスタンスで実行されているクラスタ・キャッシュ・マネージャ・デーモンに関する診断情報を格納します。
monClusterCacheManager は、クラスタ全体にわたる情報をインスタンス単位でレポートします。
- **monCMSFailover** — クラスタ・メンバシップ・サービスによるエラーの検出、新しいクラスタ・ビューの取得、ハートビートの再同期、エラー・イベントの通知、エラー・イベントの完了が行われる時刻を追跡します。インスタンスごとに1つのローがあります。
- **monDBRecovery** — クラスタ内のすべてのインスタンスからのロー、およびリカバリに必要な各データベースのローが含まれます。
- **monDBRecoveryLRTypes** — リカバリ中に参照されたログ・レコードを追跡します。最低でも1つのログ・レコードがリカバリによって参照された、各ログ・レコード・タイプ用のローが含まれます。
- **monFailoverRecovery** — クラスタ・ロック・マネージャ (CLM)、データベース・リカバリ、CMS の各モジュールについて集計されたフェールオーバー・リカバリ診断情報が含まれます。
- **monLogicalCluster** — 現在システム上に設定されている論理クラスタに関する情報を表示します。
- **monLogicalClusterAction** — ローカル・クラスタに対するすべての管理アクションを示します (起動からこれらのアクションの解放まで)。
- **monLogicalClusterInstance** — インスタンスと論理クラスタ間の多対多の関係に関する情報を表示します。
- **monLogicalClusterRoute** — 設定されているルートに関する情報 (アプリケーション、ログイン、エイリアス・バインディング) を表示します。このモニタリング・テーブルに対するクエリを実行するのに **mon_role** の役割は不要です。
- **monPCM** — クラスタ内のピア・コーディネーション・モジュール (PCM: peer coordination module) クライアント・アクティビティ (送受信されたフラグメント数など) を追跡します。また、このテーブルには、各 PCM クライアントのローが含まれます。
- **monProcessMigration** — 現在マイグレートしている接続に関する情報を表示します。
- **monSysLoad** — エンジン単位の統計情報の傾向を提供します。このモニタリング・テーブルに対するクエリを実行するのに **mon_role** の役割は不要です。

- **monTempdbActivity** — インスタンスが **tempdb** 設定モードで開始された場合に、グローバル・システム **tempdb** も含めてオープンしているすべてのローカル・テンポラリ・データベースの統計情報を提供します。
- **monWorkload** — インスタンスごとに各論理クラスタの負荷スコアをそのロード・プロファイルに従って表示します。
- **monWorkloadPreview** — プロファイルを有効にせずにロード・プロファイルが負荷スコアにどれだけ影響を及ぼすかの見積もりを提供します。**monWorkload** には、論理クラスタが実行されている論理クラスタおよびインスタンスごとに 1 つのローが含まれます。
- **monWorkloadProfile** — 現在設定されている負荷プロファイルを表示します。このモニタリング・テーブルに対するクエリを実行するのに **mon_role** の役割は不要です。
- **monWorkloadRaw** — インスタンスごとにロー負荷統計情報を提供します。このモニタリング・テーブルに対するクエリを実行するのに **mon_role** は不要です。

設定パラメータ

Adaptive Server Cluster Edition バージョン 15.5 以降では、設定パラメータが追加されています。

cluster redundancy level — 同時に障害が発生しても許容されるインスタンスの最大数であり、この数値内であれば、リカバリして他のアクティビティを同時に続行できます。障害が発生したインスタンス数が最大値を超えた場合、クラスタは停止されます。

関数

Adaptive Server Cluster Edition バージョン 15.5 以降では、関数が追加されています。

- **xact_owner_instance** — 外部トランザクションが実行されているインスタンス、または 0 を返します。
- **xact_conmigrate_check** — 接続で外部トランザクションを処理できるかどうかを判断します。

Adaptive Server® バージョン 15.5 の新機能

Adaptive Server 15.5 では、インメモリ・データベースとリラックス持続性データベース、IBM Tivoli Storage Manager 用の Backup Server サポート、バックアップの高速圧縮、ストアド・プロシージャの遅延名前解決、増分データ転送、FIPS 140-2 パスワード暗号化のサポート、新しいデータ型が導入されています。

Adaptive Server 15.5 の機能とプラットフォームの一覧

次の機能とプラットフォームの一覧は、Adaptive Server 15.5 でサポートされる各オペレーティング・システムで利用できる機能を示します。"Y" は、機能がそのプラットフォームでサポートされていることを示します。

Adaptive Server のオプション	HP-UX Itanium 64 ビット版	HP-UX PA Risc 64 ビット版	IBM AIX 64 ビット版	Linux on Power 64 ビット版	Linux Opteron 64 ビット版	Linux x86 32 ビット版	Solaris 32 ビット版	Solaris 64 ビット版	Solaris Opteron 64 ビット版	Windows Opteron X64	Windows x86 32 ビット版
Security and Directory Services	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Cybersafe Kerberos							Y	Y			Y
プラグ可能認証モジュール	Y		Y	Y	Y	Y	Y	Y	Y		
詳細なアクセス制御	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
LDAP サーバ・ディレクトリ	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
LDAP ユーザ認証	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
プラットフォーム・ネイティブ Kerberos							Y	Y			

Adaptive Server のオプション	HP-UX Itanium 64 ビット版	HP-UX PA Risc 64 ビット版	IBM AIX 64 ビット版	Linux on Power 64 ビット版	Linux Opteron 64 ビット版	Linux x86 32 ビット版	Solaris 32 ビット版	Solaris 64 ビット版	Solaris Opteron 64 ビット版	Windows Opteron X64	Windows x86 32 ビット版
SSL (Secure Sockets Layer)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
MIT Kerberos	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y
暗号化カラム (詳細なアクセス制御 (FGAC) を含む)	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y
高可用性	Y	Y	Y		Y	Y	Y	Y			Y
パーティション	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
インメモリ・データベース	Y	Y	Y	Y	Y			Y	Y	Y	
Backup Server 対応 Tivoli Storage Manager	Y		Y		Y			Y	Y	Y	
Active Messaging	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y
拡張型全文検索 (EFTS)		Y	Y		Y	Y	Y	Y			Y
<i>基本の Adaptive Server に組み込まれている機能</i>											
プラットフォーム間でのダンプとロード	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Job Scheduler	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ネイティブ XML	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
IPv6	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Java オプション	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Web Services	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
分散トランザクション管理	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Adaptive Server のオプション	HP-UX Itanium 64 ビット版	HP-UX PA Risc 64 ビット版	IBM AIX 64 ビット版	Linux on Power 64 ビット版	Linux Opteron 64 ビット版	Linux x86 32 ビット版	Solaris 32 ビット版	Solaris 64 ビット版	Solaris Opteron 64 ビット版	Windows Opteron X64	Windows x86 32 ビット版
コンテンツ管理 (外部ファイルのサポート)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
アーカイブ・データベースへのアクセス	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

インメモリ・データベースおよびリラックス持続性データベース

インメモリ・データベースおよびリラックス持続性データベースにより、パフォーマンスが強化されます。

インメモリ・データベースは名前付きキャッシュ (つまり Adaptive Server メモリ領域) 内ですべて実行され、データやログの保存にディスク記憶領域を使用しません。インメモリ・データベースでは I/O が不要であるため、従来のディスク常駐型データベースよりもより優れたパフォーマンスを実現できます。ただし、インメモリ・データベースはリカバリ用には設計されていません。インメモリ・データベースのトランザクション・ログはディスクではなくキャッシュに書き込まれるため、サーバを再起動するとデータの変更内容がすべて失われます。インメモリ・データベースは、実行時ロールバックのため、およびトリガの起動、遅延モードの更新、複製などのその他の操作のためのトランザクション・ロギングを実行します。

ディスク常駐型データベースは、ディスクへの書き込みを実行することで、原子性、一貫性、整合性、持続性 (ACID プロパティとも呼ばれます) のトランザクション・プロパティを保証します。持続性とは、トランザクションがコミットされた後の永続性を意味します。従来の Adaptive Server データベースは、トランザクションがコミットされるとトランザクション・ログをディスクに書き込むことにより、完全な持続性で動作します。この動作に加え、データ・ページのディスクへの定期的な書き込みも行うことで、コミットされたすべてのトランザクションの持続性を保証します。

インメモリ・データベースはデータやログをディスクに書き込まず、トランザクションの持続性と引き換えにパフォーマンスを向上させます。データベースに障害が発生した場合、インメモリ・データベースは回復できません。サーバの障害や通常の停止にともなうデータのリカバリ性が必要なアプリケーションでは、従来の Adaptive Server データベースの使用を考慮してください。

リラックス持続性をサポートすることで、Sybase はインメモリ・データベースから得られるパフォーマンスのメリットをディスク常駐型データベースにまで拡張します。従来のディスク常駐型データベースは、サーバの障害からのトランザクションのリカバリを保証します。リラックス持続性データベースは、コミットされたトランザクションの完全な持続性と引き換えに、トランザクションの負荷に対する実行時のパフォーマンスを向上させます。

インメモリ・データベースおよびリラックス持続性データベースから得られるパフォーマンスのメリットは次のとおりです。

- I/O の待機なし (インメモリ・データベースは完全にメモリ内で動作するため)。
- バッファおよびユーザ・ログのキャッシュ管理の向上。これにより、Adaptive Server が同じデータに対する更新を同時に実行する場合、ユーザ・ログ・キャッシュの書き込みおよびバッファ管理のオーバーヘッドを生み出す必要がない。
- トランザクションのコミットまたはアボート時に、ユーザ・ログ・キャッシュに登録されたタスクの変更がトランザクション・ログにフラッシュされないようにするランタイム方法。これにより、メモリ内ログ・ページの競合が少なくなる。
- メモリ内ロギング手法を使用して最低限のログを取る DML オペレーションのサポート。これにより、大量の DML オペレーションのパフォーマンスが向上する。

Adaptive Server バージョン 15.5 では、次のタイプのデータベースを作成できます。

- 持続性が full に設定されているディスク常駐型データベース (デフォルトの Adaptive Server データベース)
- ユーザ定義のディスク常駐型テンポラリー・データベース
- 持続性が no_recovery に設定されているメモリ内ユーザ・データベース
- 持続性が no_recovery に設定されているユーザ定義のインメモリ・テンポラリー・データベース
- 持続性が no_recovery または at_shutdown に設定されているディスク常駐型リラックス持続性データベース

Adaptive Server では、インメモリ・データベースとリラックス持続性データベースのどちらに対しても **dump database** および **load database** がサポートされています。また、インメモリ・データベースをダンプして、それをディスク常駐型データベースにロードすることもできます (その逆も可能)。Adaptive Server では、サー

バの再起動時にリカバリ不可能なデータベースの内容を再初期化する方法として
テンプレート・データベースがサポートされています。

『インメモリ・データベース・ユーザズ・ガイド』を参照してください。

バックアップの高速圧縮

データベースとトランザクションのダンプに使用する圧縮レベルが追加され、
CPU 集中が少なく、より高速かつ完全な圧縮が可能になっています。

2つの新しい圧縮レベルが **dump database...compression=** および **dump transaction...compression=** コマンドに追加されています。追加されたレベルは 100 と 101 です。圧縮レベル 100 ではより高速な圧縮が行われ、レベル 101 ではより最適な圧縮が行われます。新しい圧縮レベルはどちらもレベル 0～9 よりも CPU の集中使用が緩和されています。

『システム管理ガイド』および『リファレンス・マニュアル：コマンド』を参照してください。

IBM® Tivoli® Storage Manager 用の Backup Server サポート

Adaptive Server 15.5 では、記憶領域管理サービスを提供する IBM Tivoli Storage Manager (TSM) と Backup Server の連携がサポートされています。TSM のサポートは、ライセンスが必要な Adaptive Server 機能です。

TSM は、ライセンスされたユーザに記憶領域管理サービスを提供するサード・パーティのクライアント/サーバ・プログラムです。Adaptive Server Backup Server は、バックアップ・メディアとしてテープ・ドライブとディスク・ファイルをサポートしています。TSM は Backup Server と連動して、より多くのバックアップ・メディアを使用できるようにします。TSM がサポートするすべてのバックアップ・メディアに Adaptive Server のバックアップを保管できます。

Adaptive Server は TSM との間でデータベースおよびトランザクションのダンプとロードを処理します。TSM は記憶領域と、記憶メディアからの取得を処理します。TSM は記憶領域管理サービスを提供しますが、ユーザは Adaptive Server のすべてのバックアップ操作とリストア操作を Backup Server から管理できます。この操作には、バックアップ・オブジェクトに対するクエリやバックアップ・オブジェクトの削除が含まれます。

dump コマンドまたは **load** コマンドを実行するとき、Backup Server は TSM API との Sybase インタフェースを呼び出し、これによって TSM と通信できるようになり

まず、**dump** コマンドを使用するときは、バックアップ・オブジェクトとユニークに関連付けられたオブジェクト名を指定します。このオブジェクト名は TSM オブジェクト名と同じであり、後でロード操作を実行するときに同じデータベースまたはトランザクション・ダンプを指定するために使用します。TSM が設定されていない場合は、通常、TSM での **dump** コマンドと **load** コマンドには、Backup Server で使用するときと同じオプションを使用できます。

Adaptive Server では、TSM からのバックアップ・オブジェクトのクエリと削除をサポートするために、次のストアド・プロシージャが導入されています。

- **sp_deletesmobj** – TSM から現在のサーバの一部またはすべてのバックアップ・オブジェクトを削除します。
- **sp_querysmobj** – サーバのバックアップ・オブジェクトの一覧を取得します。

「IBM Tivoli Storage Manager と Backup Server の使用」を参照してください。

ユーザ定義のストアド・プロシージャ用の遅延名前解決

遅延名前解決を使用すると、参照先のオブジェクトがまだ存在していなくてもストアド・プロシージャを作成できます。

15.5 よりも前のバージョンの Adaptive Server では、オブジェクトを参照するストアド・プロシージャを作成するには、参照先のオブジェクトがすでに存在している必要がありました。遅延名前解決機能により、オブジェクト (ユーザ定義のデータ型オブジェクトを除く) をストアド・プロシージャの初回実行時に解決できるようになりました。

この機能では、サーバ・レベルで動作する **deferred name resolution** 設定パラメータ、または接続レベルで動作する新しい **set** コマンドである **set deferred_name_resolution** が使用されます。

『Transact-SQL ユーザーズ・ガイド』、『システム管理ガイド 第 1 巻』、『リファレンス・マニュアル：コマンド』を参照してください。

FIPS 140-2 ログイン・パスワードの暗号化

Adaptive Server 15.5 では、FIPS 140-2 検証済み暗号化モジュールがサポートされています。

FIPS 140-2 認定の Certicom Security Builder GSE は、**FIPS login password encryption** 設定パラメータを使用した場合に、メモリやディスク上で転送されるログイン・パケットのログイン・パスワードを暗号化します。

```
sp_configure 'FIPS login password encryption', 1
```

注意： このパラメータを有効にするには、Security and Directory Services ライセンスが必要です。このパラメータが有効でない場合、OpenSSL セキュリティ・プロバイダによってログイン・パスワードの暗号化が実行されます。

『暗号化カラム・ユーザズ・ガイド』を参照してください。

増分データ転送

増分データ転送を使用すると、データを Adaptive Server または他の製品に転送できます。

transfer table コマンドを使用すると、データを増分転送できます。必要に応じて、異なる製品に増分転送することもできます。15.5 よりも前のバージョンの Adaptive Server では、1 台の Adaptive Server からもう 1 台の Adaptive Server にテーブル全体しか転送できませんでした。

注意： データ転送機能は、インメモリ・データベース・ライセンスを購入してインストールおよび登録した時点で、Adaptive Server によって有効化されます。

増分データ転送では、次のことが可能です。

- 増分転送用のマークが付けられた Adaptive Server のテーブルから、前回の転送後に変更されたデータだけをエクスポートできます。
- 通常のロックを取得しない、ローの取得順序を指定しない、読み取り中または更新中の他のデータを妨害しないで、テーブル・データを読み取ることができます。
- 選択したローは、定義されている受信側用にフォーマットした出力ファイルや名前付きパイプ (IQ (Sybase IQ)、ASE (Adaptive Server Enterprise)、バルク・コピー (**bcp**)、文字コード化された出力に書き込むことができます。選択したすべてのローは暗号化せずに転送されます。また、暗号化されたカラムがローに含まれている場合は、デフォルトで復号化されてから転送されます。書き込むファイルは、Adaptive Server が稼働しているマシンで認識される必要があります (ファイルは、Adaptive Server がローカル・ファイルとして開くことのできる NFS ファイルにすることもできます)。
- 増分転送用テーブルの転送履歴が保持され、不要になったテーブルの転送履歴をユーザが削除できます。一定の制限に従って、増分転送不適格として宣言されたテーブルからデータをエクスポートします。
- 指定されたテーブルから、ロー単位でデータを転送します。現時点では、特定のカラムの選択、テーブル内の一部分の選択、SQL クエリ結果の転送は実行できません。

『Transact-SQL ユーザズ・ガイド』の「第 8 章 データの追加、変更、転送、削除」を参照してください。

bigdatetime および bigtime データ型

bigdatetime および *bigtime* には、マイクロ秒の精度があります。

2つの新しいデータ型である *bigdatetime* および *bigtime* は、年、月、日、時、分、秒、秒の小数位 6 桁で構成されるマイクロ秒の精度のタイムスタンプを提供します。*bigdatetime* 値には 8 バイトの記憶領域が必要です。*bigtime* 値は、時、分、秒、秒の小数位で構成される時間です。秒の小数点以下は 6 桁まで格納されます。*bigtime* 値には 8 バイトの記憶領域が必要です。

『ASE Transact-SQL ユーザーズ・ガイド』、『リファレンス・マニュアル：ビルディング・ブロック』、『Adaptive Server Enterprise における Java』を参照してください。

tempdb グループの作成および管理

バージョン 15.5 では、**tempdb** グループが更新されています。

Adaptive Server では、デフォルトの **tempdb** グループを管理できるだけでなく、ユーザ作成の **tempdb** グループも作成および管理できます。ユーザ作成の **tempdb** グループに、他のユーザ作成テンポラリー・データベースを含め、アプリケーションおよびログインのバインドをサポートできます。

システム **tempdb** は、デフォルトのテンポラリー・データベース・グループから削除できません。システム **tempdb** は、他のユーザ作成の **tempdb** グループに追加できません。

ユーザ作成の **tempdb** グループを指定し、ディスク常駐型データベースまたはインメモリ・テンポラリー・データベースのみを含めるように管理できます。サーバで明示的にこのような制限が課せられることはありませんが、メンバシップを制御して、ディスク専用またはメモリ内専用の **tempdb** グループを特定のログインやアプリケーションに割り当てることができます。

Adaptive Server 15.5 でのシステムの変更点

Adaptive Server 15.5 では、データ型、関数、システム・プロシージャ、コマンド、設定パラメータ、モニタリング・テーブル、システム・テーブル、ユーティリティが追加および変更されています。また、新しい監査オプションも追加されています。

データ型

新しい *bigtime* および *bigdatetime* データ型は、正確なタイムスタンプ情報を提供します。

データ型	説明
<i>bigtime</i>	<i>bigtime</i> 値は、時、分、秒、秒の小数位で構成されます。秒の小数点以下は 6 桁まで格納されます。
<i>bigdatetime</i>	<i>bigdatetime</i> 値は、年、月、日、時、分、秒、秒の小数位 6 桁で構成されます。

次の 2 つの新しい関数は、*bigtime* 値および *bigdatetime* 値を返します。

- **current_bigtime**
- **current_bigdatetime**

bigtime および *bigdatetime* は、次の既存の関数で使用できます。

- **datepart**
- **datename**
- **datediff**
- **dateadd**

『リファレンス・マニュアル：ビルディング・ブロック』を参照してください。

関数

Adaptive Server 15.5 では、関数が追加および変更されています。

表 1：新しい関数

関数	説明
<code>db_attr</code>	指定したデータベースの <code>durability</code> 、 <code>dml_logging</code> 、 <code>template</code> の設定を返します。
<code>object_attr</code>	セッション固有の設定、テーブル全体の設定、データベース全体の設定に応じて、テーブルの現在のロギング・モードをレポートします。
<code>cache_usage</code>	指定したオブジェクトがバインドされているキャッシュのキャッシュ使用率を、そのキャッシュにバインドされているすべてのオブジェクトで現在使用されているキャッシュのパーセンテージとして返します。
<code>current_bigdatetime</code>	現在の日時を表すマイクロ秒の精度の <code>bigdatetime</code> 値を返します。
<code>current_bigtime</code>	現在の時刻を表すマイクロ秒の精度の <code>bigtime</code> 値を返します。

表 2：変更された関数

関数	説明
<code>datepart</code>	指定した日付の指定した <code>datepart</code> 引数を整数として生成します。
<code>datename</code>	指定した <code>datepart</code> を文字列として生成します。
<code>datediff</code>	指定した 2 つの日付間または時刻間で日付要素の数を計算します。
<code>dateadd</code>	指定した日付または時刻に間隔を追加します。

『リファレンス・マニュアル：コマンド』を参照してください。

システム・ストアド・プロシージャ

追加および変更されたシステム・ストアド・プロシージャでは、Tivoli Storage Manager、インメモリ・データベース、リラックス持続性データベースがサポートされています。

表 3：新しいシステム・ストアド・プロシージャ

システム・ストアド・プロシージャ	説明
sp_deletesmobj	TSM からバックアップ・オブジェクトを削除します。
sp_querysmobj	TSM からバックアップ・オブジェクトの一覧を取得します。

表 4：変更されたシステム・ストアド・プロシージャ

システム・ストアド・プロシージャ	説明
sp_addsegment	インメモリ・データベースの領域を管理するために更新されました。
sp_addthreshold	インメモリ・データベースの領域を管理するために更新されました。
sp_bindcache	オブジェクトやデータベースはメモリ内記憶域キャッシュにバインドできません。また、インメモリ・データベースやインメモリ・データベースのオブジェクトはキャッシュにバインドできません。
sp_cacheconfig	メモリ内記憶域キャッシュの作成、サイズの拡大、または削除を行います。
sp_cachestrategy	prefetch パラメータと MRU パラメータは、インメモリ・データベースのテーブルおよびインデックスに適用されません。
sp_dbextend	自動データベース拡張は、現在インメモリ・データベースでサポートされていません。
sp_deviceattr	directio 属性と dsync device 属性は、メモリ内デバイスに適用されません。

システム・ストアド・プロシージャ	説明
sp_downgrade	インメモリ・データベースやリラックス持続性データベース、またはテンプレートや最低限のロギングを使用するデータベースが含まれている Adaptive Server のダウングレードをサポートします。
sp_diskdefault	sp_diskdefault を使用して、メモリ内デバイスをデフォルトのデバイスとして指定することはできません。
sp_dropdevice	作成されたメモリ内デバイスをメモリ内記憶域キャッシュから削除します。
sp_dropsegment	インメモリ・データベースの領域を管理するために更新されました。
sp_dropthreshold	インメモリ・データベースの領域を管理するために更新されました。
sp_extendsegment	インメモリ・データベースの領域を管理するために更新されました。
sp_help	最低限のロギング属性など、テーブルのプロパティをレポートします。
sp_helpcache	メモリ内記憶域キャッシュのプロパティ、このキャッシュに作成されたインメモリ・データベースのプロパティ、このキャッシュの空き容量の詳細を表示します。
sp_helpdb	持続性、DML ロギング・レベル、インメモリ・データベースかどうか、テンプレート・データベースがある場合はテンプレート・データベースの使用、テンプレート・データベースとしての使用など、データベースのプロパティをレポートします。
sp_helpdevice	メモリ内記憶域キャッシュで作成されたメモリ内デバイスのプロパティをレポートします。
sp_modifythreshold	インメモリ・データベースの領域を管理するために更新されました。
sp_plan_dbccdb	インメモリ・データベースで <code>checkstorage</code> を実行するための <code>dbccdb</code> を設定します。

システム・ストアド・プロシージャ	説明
<code>sp_poolconfig</code>	インメモリ・データベースでは、サイズの大きな I/O バッファ・プールはサポートされていません。
<code>sp_post_xpload</code>	インメモリ・データベースに関するプラットフォームを問わない操作をサポートしません。
<code>sp_tempdb</code>	ユーザ作成のテンポラリ・データベース・グループをサポートし、テンポラリ・データベース・グループおよびインメモリ・データベースに対するログインまたはアプリケーションのバインドをサポートしません。
<code>sp_unbindcache, sp_unbindcache_all</code>	インメモリ・データベース自体のオブジェクトのバインドをホストのメモリ内記憶域キャッシュから解除できません。

『リファレンス・マニュアル：プロシージャ』を参照してください。

コマンド

Adaptive Server 15.5 では、コマンドが追加および変更されています。

表 5：新しいコマンド

コマンド	説明
<code>transfer table</code>	<p>テーブルの増分転送を開始します。</p> <p>新しい <code>grant with grant option</code> では、transfer table がサポートされています。このオプションでは、指定したユーザに指定したテーブルを転送するパーミッションを付与します。</p> <pre>grant transfer table on table_name to user with grant option</pre>

表 6 : 変更されたコマンド

コマンド	変更点
alter database	データベースの持続性、DML ロギングのレベル、データベースのテンプレートの変更をサポートするための構文が追加されました。
alter table	insert、update、delete (DML) オペレーションに対するテーブルのロギング・モードの変更をサポートするための構文が追加されました。 transfer table をサポートするための構文が追加されました。 <code>set transfer table [on off]</code>
create database	持続性が full、no_recovery、または at_shutdown に設定されたインメモリ・データベースおよびラックス持続性データベースを作成するための構文が追加されました。 DML ロギング・レベルおよびテンプレート・データベース (存在する場合) を指定するための構文が追加されました。
create table	インメモリ・データベースのテーブルに対する DML ロギング・レベルを指定するための構文が追加されました。 transfer table をサポートするための構文が追加されました。 <code>with transfer table [on off]</code>
disk init	インメモリ・データベースのメモリ内データ・デバイスを作成するための構文が追加されました。
dump database	Tivoli Storage Manager (TSM) をサポートするための構文が追加されました。キーワード syb_tsm は TSM API (libsyb_tsm) との Sybase インタフェースを呼び出します。 <code>database_name to "syb_tsm::object_name"</code>
dump database ... compression=	高速で CPU 集中の少ない圧縮レベル 100 および 101 をサポートするための構文が追加されました。
dump transaction	TSM をサポートするための構文が追加されました。キーワード syb_tsm は TSM API (libsyb_tsm) との Sybase インタフェースを呼び出します。 <code>database_name to "syb_tsm::object_name"</code>

コマンド	変更点
dump transaction ... compression=	高速で CPU 集中の少ない圧縮レベル 100 および 101 をサポートするための構文が追加されました。
load database	TSM をサポートするための構文が追加されました。キーワード syb_tsm は TSM API (libsyb_tsm) との Sybase インタフェースを呼び出します。 <pre> database_name from "syb_tsm:: [[-S source_server_name] [-D source_database_name]::]object_name"</pre>
load transaction	TSM をサポートするための構文が追加されました。キーワード syb_tsm は TSM API (libsyb_tsm) との Sybase インタフェースを呼び出します。 <pre> database_name from "syb_tsm:: [[-S source_server_name] [-D source_database_name]::]object_name"</pre>
select into	インメモリ・データベースまたはリラックス持続性データベースへの挿入対象として選択することで作成されるテーブルに対する DML ロギング・レベルを指定するための構文が追加されました。
set	次の項目が追加されています。 <ul style="list-style-type: none"> • dml_logging パラメータ (セッションに対するロギングの量を指定) • deferred_name_resolution (接続レベルで遅延名前解決をアクティブ化) • builtin_date_strings number 値は次のとおりです。 <ul style="list-style-type: none"> • 0 – 文字列が日付順のシステム関数への引数として指定された場合、サーバは示された精度にかかわらず、その文字列を <i>datetime</i> 値として解釈します。これがデフォルト値です。 • 1 – サーバは引数文字列を <i>bigdatetime</i> として解釈します。これは、日付順のシステム関数の結果に影響します。

『リファレンス・マニュアル：コマンド』を参照してください。

設定パラメータ

Adaptive Server 15.5 では、**deferred name resolution** 設定パラメータが導入されています。

設定パラメータ	説明
deferred name resolution	<p>遅延名前解決を使用してプロシージャを作成できます。値は次のとおりです。</p> <ul style="list-style-type: none"> • 0 – 遅延名前解決を無効にします。これがデフォルト値です。 • 1 – 遅延名前解決を有効にします。
builtin date strings	<p>値は次のとおりです。</p> <ul style="list-style-type: none"> • 0 – 日付順の組み込み関数に引数として指定した文字列リテラルが <code>datetime</code> 型として解釈されます。 • 1 – 日付順の組み込み関数に引数として指定した文字列リテラルが <code>bigdatetime</code> 型として解釈されます。

『システム管理ガイド 第1巻』および『Transact-SQL ユーザーズ・ガイド』を参照してください。

モニタリング・テーブル

Adaptive Server 15.5 では、*monTableTransfer* モニタリング・テーブルが導入されています。

モニタリング・テーブル	説明
<i>monTableTransfer</i>	Adaptive Server のアクティブなメモリに存在するテーブルの転送履歴情報を提供します。

『リファレンス・マニュアル：テーブル』を参照してください。

システム・テーブル

Adaptive Server 15.5 では、システム・テーブルが追加および変更されています。

表 7: 新しいシステム・テーブル

システム・テーブル	説明
<i>spt_TableTransfer</i>	テーブル転送の結果を保管します。

表 8: 変更されたシステム・テーブル

システム・テーブル	変更点
<i>sysdevices</i>	<i>name</i> および <i>phyname</i> カラムのメモリ内記憶域キャッシュをリストします。メモリ内デバイスには、ディスク・デバイスのフル・パスは格納されず、代わりにメモリ内デバイスが作成されたキャッシュの名前が格納されます。
<i>sysdatabases</i>	データベースの持続性レベルを示す <i>durability</i> カラムを追加します。 <i>durability</i> カラムでは、 <i>int</i> データ型が使用されます。値は次のとおりです。 <ul style="list-style-type: none"> • 1 – full • 5 – at_shutdown • 6 – no_recovery

『リファレンス・マニュアル：テーブル』を参照してください。

ユーティリティ

Adaptive Server 15.5 では、ユーティリティが追加および変更されています。

表 9 : 新しいユーティリティ

ユーティリティ	説明
openssl	certreq、certauth、certpk12 で実装されたすべての証明書管理タスクを実行します。Sybase では便宜上このバイナリを組み込んでいますが、バイナリを使用して発生した問題についてはいっさい責任を負いません。詳細については、 www.openssl.org を参照してください。

表 10 : 変更されたユーティリティ

ユーティリティ	変更点
backupserver	構文が変更され、-V パラメータの別の詳細度レベル (V4) がサポートされています。V4 では、接続イベントごとに出力される "Connection from Server" メッセージを除くすべての -V0 メッセージが表示されます。

『ユーティリティ・ガイド』を参照してください。

監査

インメモリ・データベースとリラックス持続性データベース、増分データ転送、遅延名前解決をサポートする監査オプションが追加されています。

表 11 : 監査機能の強化

監査オプション	監査されるコマンドまたはアクセス	イベント	extrainfo の情報
all、create	transfer table	136	キーワードまたはオプション： transfer_table

監査オプション	監査されるコマンドまたはアクセス	イベント	extrainfo の情報
all、create	alter table	3	<p>alter table に set transfer table on が含まれている場合、Adaptive Server は次の情報を extrainfo に出力：SET TRANSFER TABLE ON</p> <p>alter table に set transfer table off が含まれている場合、Adaptive Server は次の情報を extrainfo に出力：SET TRANSFER TABLE OFF</p>
all、create	create table	12	<p>create table に with transfer table on が含まれている場合、Adaptive Server は次の情報を extrainfo に出力：WITH TRANSFER TABLE ON</p> <p>create table に with transfer table off が含まれている場合、Adaptive Server は次の情報を extrainfo に出力：WITH TRANSFER TABLE OFF</p>
all、create	create database	9	キーワードまたはオプション： inmemory
all、create	alter database	2	キーワードまたはオプション： inmemory
all、create	create procedure	11	キーワードまたはオプション： deferred_name_resolution

Adaptive Server 15.0.3 の新機能

Adaptive Server 15.0.3 では、分散トランザクション管理、Java インタフェースの機能強化、仮想ハッシュ・テーブル、ヒュージ・ページ、Adaptive Server プラグインの更新、高可用性設定時のアップグレード方法、SQL 文の複写のサポートが導入されています。

SQL 文の複写

Replication Server 15.2 では、Adaptive Server データベースに対する SQL 文の複写がサポートされています。

Adaptive Server Enterprise 15.0.3 では、SQL 文の複写が導入されています。これは、Replication Server 15.2 以降でサポートされています。

Replication Server のマニュアルを参照してください。

セキュリティの機能強化

Adaptive Server バージョン 15.0.3 では、いくつかのセキュリティ機能が強化されています。

LDAPS ユーザ認証の強化

CA によって信頼されたルート・ファイルを変更したときに、サーバの再起動が不要になりました。

以前のバージョンの Adaptive Server では、CA (認証局) によって信頼されたルート・ファイルに変更を加えた場合に、Adaptive Server を再起動して変更を有効にする必要があります。Adaptive Server バージョン 15.0.3 以降では、信頼されたルート・ファイルへの変更がサポートされているため、サーバを再起動する必要がありません。新しく追加されたサブコマンド **reinit_descriptors** は、LDAP サーバ記述子のバインドを解除して、ユーザ認証サブシステムを再初期化します。

自動的な LDAP ユーザ認証とフェールバック

Adaptive Server のハウスキーピング・ユーティリティでは、障害の発生した LDAP サーバを自動的にアクティブにできます。

Adaptive Server 15.0.3 では、セカンダリ LDAP サーバがサポートされています。以前のバージョンでは、障害の発生したプライマリ LDAP サーバをオンライン状態にしたら、新しい LDAP ログインを認証して、プライマリ LDAP サーバに移動するために、LDAP サーバを手動でアクティブにする必要がありました。

バージョン 15.0.3 以降では、LDAP サーバを自動的にアクティブにするための、新しい処理 '**set_failback_interval**' が Adaptive Server のハウスキーピング・ユーティリティに追加されています。

フェールバック間隔がユーザによって設定されると、ハウスキーピング・タスクは、処理を一掃するたびに、障害の発生した LDAP サーバの有無を調べます。障害の発生した LDAP サーバが見つかった場合は、フェールバック時間間隔で指定した時間が経過すると、LDAP サーバのアクティブ化が試みられます。

外部認証のログイン・マッピング

Adaptive Server では、内部 Adaptive Server ログインに対する外部ユーザのユニークなマッピングを 1 つ使用します。

外部認証メカニズムを設定したときに、内部 Adaptive Server ログインに対する外部ユーザのマッピングが 1 つだけあり、認証が成功した場合、Adaptive Server は外部ユーザのパスワードと一致するように内部ログインのパスワードを更新します。

たとえば、次のような状況が考えられます。

1. USER1 の Adaptive Server ログイン名は "user_ase"、パスワードは "user_password" です。
2. 別のユーザの LDAP ログイン名は "user_ldap"、パスワードは "user_ldappasswd" です。
3. Adaptive Server では、"user_ldap" と "user_ase" が 1 対 1 でマッピングされています。
4. ユーザ "user_ldap" は、パスワード "user_ldappasswd" を使用して Adaptive Server にログインします。Adaptive Server は、パスワード "user_ase" を "user_ldappasswd" に更新します。

SSL を使用した共通名の指定

SSL 証明書の共通名の完全修飾ドメイン名を使用します。

ディレクトリ・サービス・エントリで指定したサーバ名は、SSL ハンドシェイクを実行する際に SSL サーバ証明書が使用する共通名とは異なる可能性があります。

これにより、SSL 証明書の共通名の完全修飾ドメイン名 (たとえば、server1.bigcompany.com) を使用できます。

interfaces ファイルに共通名を追加するには、次のコマンドを使用します。

```
asel
  master tcp ether host_name
           port_number ssl="CN='common_name'"
  query tcp ether host_name
           port_number ssl="CN='common_name'"
```

SSL を使用する Adaptive Server にクライアントが SSL を使用して接続する場合は、interfaces ファイルのポート番号の後に SSL フィルタが配置されます。ディレクトリ・サービスには、**dsedit** またはテキスト・エディタを使用して追加できる共通名が含まれます。

sp_listener にはパラメータ **CN=common_name** が含まれており、SSL 証明書の共通名を指定できます。

Kerberos による同時認証

Adaptive Server では、複数の Kerberos 認証セッションを確立できます。

以前のバージョンでは、Kerberos による認証時にロック・メカニズムを使用することによって内部データ構造を保護していましたが、Adaptive Server バージョン 15.0.3 では、Kerberos による同時認証がサポートされるようになりました。

Kerberos 認証を使用した同時ログインがある場合は、Adaptive Server によって複数の Kerberos 認証セッションが確立されます。

バージョン 15.0.3 では、Kerberos による認証時に同時ログイン・セッションがブロックされる問題も解決されています。同時実行性に関連したこの問題は、以前のバージョンの Adaptive Server を、MIT バージョン 1.3.x および 1.4.x の Kerberos GSSAPI ライブラリとともに使用する場合に発生します。

仮想ハッシュ・テーブル

仮想ハッシュ・テーブルを作成して、テーブルを効率的に整理できます。

注意： 仮想ハッシュ・テーブルは、Linux pSeries でのみ使用できます。

ハッシュベース・インデックス・スキャンは、ノンクラスタード・インデックスまたはデータオンリーロック・テーブルのクラスタード・インデックスで実行できます。このスキャン時、各ワーカー・プロセスは、高いレベルのインデックスを操作し、インデックスのリーフレベル・ページを読み込みます。次に、各ワーカー・プロセスが別個のハッシュ・テーブル内のデータ・ページ ID またはキー値

に基づいてハッシュを行い、どのデータ・ページまたはデータ・ローを処理するか決定します。

仮想ハッシュ・テーブルは、別個のハッシュ・テーブルを必要としないため、テーブルを効率的に整理できる方法です。代わりに、ハッシュ・キーを使用してクエリ プロセッサがロー ID (ローの序数に基づく) およびデータの位置を判断できるように、ローを格納します。別個のハッシュ・テーブルを使用して情報を保持しないため、「仮想」ハッシュ・テーブルと呼ばれます。

中央処理装置 (CPU: central-processing unit) を効率的に使用する必要があるシステムには、仮想ハッシュ・テーブルの使用が適しています。

仮想ハッシュ・テーブルを作成するには、**create table** コマンドを使用してハッシュ領域の最大値を指定します。

ヒュージ・ページ

ヒュージ・ページを有効にして、物理アドレス空間をカバーするために使用するページを少なくします。

注意： この機能は、Linux pSeries でのみ使用できます。

CPU キャッシュ TLB (Translation Lookaside Buffer) には、仮想ページ・アドレスから物理ページ・アドレスへの変換に関する情報が保管され、物理メモリへのバイト・アクセスごとに変換 (キャッシュ・ミス) が必要となります。これらのキャッシュ・ミスは非常にコストがかかりますが、「ヒュージ・ページ」を有効にすることによって、TLB ヒットを向上させることができます。

ヒュージ・ページを有効にすると、物理アドレス空間をカバーするために使用するページが少なくなるので、「トランケーション・ポイントの書き換え」(仮想アドレスから物理アドレスへのマッピング) のサイズが減ります。そのため、TLB 内で必要なエントリが減り、システムのパフォーマンスが向上します。

Adaptive Server バージョン 15.0.3 以降では、デフォルトでヒュージ・ページを使用して共有メモリを割り付けます。ただし、システムに十分なヒュージ・ページがない場合、またはヒュージ・ページを使用するように設定されていない場合には、Adaptive Server は通常のページを使用します。

ヒュージ・ページを有効にするには、トレース・フラグ 1653 を指定して Adaptive Server を起動します。Adaptive Server は、共有メモリを 256MB の最も近い倍数に調整します。

高可用性設定時のアップグレード

高可用性 (HA) を有効にしたら、使用しているプラットフォームのインストール・ガイドのアップグレード手順に従ってください。

この項に示す手順は、『インストール・ガイド』の手順を補足するものです。

システム・ストアド・プロシージャの再インストール

高可用性を有効にした後、システム・ストアド・プロシージャを再インストールします。

1. プライマリ・サーバで HA を無効にします。

```
sp_companion secondary-server-name, 'drop'
sp_configure 'enable HA', 0
```

2. セカンダリ・サーバで HA を無効にします。

```
sp_configure 'enable HA', 0
```

3. 両方のサーバを再起動します。

4. 両方のサーバで installmaster スクリプトを実行します。

5. 両方のサーバで HA プロパティを有効にします。

```
sp_configure 'enable HA', 1
```

6. 両方のサーバを再起動します。

7. 両方のサーバで installhasvss スクリプトを実行します。このスクリプトは \$SYBASE/\$SYBASE_ASE/scripts にあります。

8. コンパニオン関係を再確立します。

```
sp_companion [companion_server_name], configure [,
with_proxydb]
```

分散トランザクション管理 (DTM)

Adaptive Server は、分散トランザクション内で実行されるべき SQL コマンドが、分散トランザクションの外側で実行されないように自動的に制御します。

分散 (外部) トランザクションは、XA トランザクション・マネージャ (TM) などの外部トランザクション・コーディネータによって管理されます。

15.0.3 より前のバージョンの Adaptive Server では、DML コマンドの実行時に外部トランザクションがロールバックされたかどうかの判断はユーザ・アプリケーションが行っていました。Adaptive Server が外部トランザクションを暗黙的にサポートしたことをアプリケーションが認識しない場合、通常はこのトランザク

ション内で実行されるはずの DML コマンドが、Adaptive Server によって開始された暗黙的なトランザクション内で実行される可能性があります。この動作により、ビジネス・データに矛盾が生じることがあります。

バージョン 15.0.3 以降では、トランザクション・マネージャからの分離要求がないかぎり、Adaptive Server は外部トランザクションに関連付けられている接続での DML コマンドの実行を許可しません。分離要求は、コマンドのバッチの終わりで外部トランザクションを実行することを意味します。

15.0.3 以降では、Adaptive Server は、分散トランザクション内で実行されるべき SQL コマンドが、分散トランザクションの外側で実行されないように自動的に制御します。つまり、すべてのコマンドを発行する前に、ユーザ・アプリケーションでグローバル変数を確認する必要がなくなりました。トランザクションが暗黙的にアボートされると、「外部トランザクションがロールバックされたため、コマンドを実行できません」というエラー・メッセージ (3953) が表示されます。

detach transaction コマンドが発行されると、このメッセージの表示が消えます。

Adaptive Server プラグインの更新

Adaptive Server プラグインが Sybase Central 6.00 上で実行されるようになりました。また、いくつかの新機能が追加されています。

Adaptive Server Plug-in for Sybase Central は、さまざまな Adaptive Server Enterprise 製品を管理します。15.0.3 より前のバージョンでは、Adaptive Server プラグインは Sybase Central 4.3 上で実行されていました。15.0.3 では、Adaptive Server プラグインは Sybase Central 6.00 上で実行されます。15.0.3、Sybase Central 6.00 の新機能を次に示します。

- 検索ツールは、プラグインによって表示されるオブジェクトを見つけるのに役立つ。
- [接続プロファイル]、[インポート]、[エクスポート] オプションを使用すると、プロファイル接続にテキストの説明を追加できる。
- Windows Vista に対するサポートが向上している。

バージョン 15.0.3 Adaptive Server プラグインの新機能を次に示します。

- コンテキストで区別されるツールバーから [追加] アイコンを選択してオブジェクトを作成できる。
- ストアド・プロシージャと SQLJ プロシージャは、[プロシージャ] フォルダに格納される。
- スカラ関数またはユーザ定義関数がサポートされるようになった。
- コンテキストで区別されるツールバーのメニューからユーティリティ項目にアクセスできる。

以前は Adaptive Server プラグインに付属していた DBISQL11 は、別個の製品 (バージョン 11.0) となりました。次の機能があります。

- 複数の結果セットの数が 10 に制限されなくなった。
- Adaptive Server のログイン・ダイアログは保持され、最後の 5 回の接続サーバ名を表示する。
- DBISQL11 または対話型 SQL で接続プロファイルと同様の機能、接続のお気に入り機能がサポートされるようになる。
- [SQL 文] ウィンドウ枠内に行番号が含まれるようになった。
- [結果] ウィンドウ枠内には、選択したローから、**select all, insert/update/delete** SQL 文の使用、並べ替え、生成を行った結果が表示されるようになった。

Java インタフェース

Adaptive Server での Java でサード・パーティの JRE と JVM コンポーネント (J2SE など) がサポートされるようになりました。

Adaptive Server バージョン 15.0.3 では、既成の Java Runtime Environment (JRE) と JVM コンポーネント (J2SE など) を Adaptive Server にプラグインできます。この Adaptive Server Java フレームワークは、プラグ可能コンポーネント・インタフェース (PCI: Pluggable Component Interface) と呼ばれます。このインタフェースには、プラグ可能コンポーネント・アダプタ (PCA: Pluggable Component Adaptor) が備わっています。Adaptive Server 用に設定された JVM はすべて「プラグイン」と呼ばれます。

Adaptive Server Java フレームワークでは、Adaptive Server バージョン 15.0.2 およびそれ以前の Java ソリューションに基づく構築が可能であり、それにより重要な機能が損なわれることはありません。15.0.3 より前のバージョンの Adaptive Server で開発した Java アプリケーションは、15.0.3 以降のバージョンのフレームワークで作成する Java アプリケーションとともにシームレスに実行できます。

PCI に対応するように Adaptive Server を設定したら、Java 6 以降をサポートする任意の標準 JVM を含めることができます。これにより、Java アプリケーションと Adaptive Server が分離され、Adaptive Server から独立して Java アプリケーションを変更またはアップグレードしたり、新しい Java 機能が利用可能になった時点でそれを活用したりできるようになります。

新しい Adaptive Server Java インタフェースの詳細については、『Adaptive Server Enterprise における Java』を参照してください。

Adaptive Server 15.0.3 でのシステムの変更点

Adaptive Server 15.0.3 では、関数、システム・ストアド・プロシージャ、コマンド、設定パラメータ、システム・テーブルが追加および変更されています。

関数

Adaptive Server 15.0.3 では、**password_random** および **pssinfo** 関数が導入されています。

関数	説明
password_random	Adaptive Server で定義されているグローバル・パスワード複雑性チェックの条件を満たす疑似乱数パスワードを生成します。
pssinfo	プロセス・ステータス構造体からの情報を返します (任意の spid のトランザクション独立性レベルを取得するオプションを指定)。

『リファレンス・マニュアル：コマンド』を参照してください。

システム・ストアド・プロシージャ

Adaptive Server 15.0.3 では、システム・ストアド・プロシージャが追加および変更されています。

表 12: 新しいシステム・ストアド・プロシージャ

ストアド・プロシージャ	説明
sp_tabsuspectptn	文字ベースのパーティション・キーを使用する範囲分割されたテーブルは、ソート順を変更すると suspect になる可能性があり、ハッシュ分割されたテーブルは、プラットフォーム間のダンプとロードを行うと suspect になる可能性があります。
sp_jreconfig	これは Java ストアド・プロシージャです。

表 13 : 変更されたシステム・ストアド・プロシージャ

プロシージャ	変更点
sp_ldapadmin	新しいパラメータ set_failback_interval および reinit_descriptors がサポートされています。
sp_addserver	リモート・プロシージャ・コール用のリモート・サーバを追加するための filter パラメータがサポートされています。
sp_passwordpolicy	validate password options パラメータがサポートされています。
sp_pciconfig	データベース内での Java がサポートされています。
sp_sysmon	追加のカウンタがサポートされています。

『リファレンス・マニュアル：プロシージャ』を参照してください。

コマンド

Adaptive Server 15.0.3 では、**create table** および **update statistics** コマンドが変更されています。

コマンド	変更点
create table	このオプションを指定して作成したテーブルは、 BCP IN および 'alter table unpartition' オペレーションでのみ使用できます。
update statistics	グローバル・ノンクラスタード・インデックスのデータ変更カウンタをリセットします。

『リファレンス・マニュアル：コマンド』を参照してください。

設定パラメータ

Adaptive Server 15.0.3 では、設定パラメータが追加されています。

プロシージャ	変更点
<code>enable pci</code>	これは Java 設定パラメータです。
<code>maximum nesting level</code>	最大ネスト・レベルが 100 に引き上げられています。
<code>mnc_full_index_filter</code>	次の項目がある場合に、サーバ・レベルでの制限探索指数を持たないカバーされていないインデックスを Adaptive Server が考慮しないように指定します。 <ul style="list-style-type: none"> インデックス内のカラム ヒストグラムがない述語
<code>pci memory size</code>	これは Java 設定パラメータです。

『システム管理ガイド 第 1 巻』を参照してください。

モニタリング・テーブル

Adaptive Server 15.0.3 では、モニタリング・テーブルが追加および変更されています。

表 14 : 新しいモニタリング・テーブル

テーブル	説明
<code>monSQLRepActivity</code>	SQL 文の複写を使用して複写された DML 文においてオープンしている全オブジェクトの統計情報を提供します。
<code>monSQLRepMisses</code>	SQL 文の複写が使用されなかった複写オペレーションの統計情報を提供します。 <i>threshold</i> 、 <i>querylimitation</i> 、 <i>configuration</i> カラムは、これらの要因のいずれかによってオブジェクトに対する SQL 文の複写が防止された回数を示します。

表 15 : 変更されたモニタリング・テーブル

モニタリング・テーブル	変更点
<i>monSysStatement</i> 、 <i>monSysPlanText</i> 、 <i>monSysSQLText</i>	<i>BatchID</i> 、 <i>ContextID</i> 、 <i>ProcedureID</i> 、 <i>PlanID</i> の各カラムの値が変更されています。
<i>monSysStatement</i>	2つの新しいカラム <i>ProcNestLevel</i> および <i>StatementNumber</i> がサポートされています。

『リファレンス・マニュアル：テーブル』を参照してください。

システム・テーブル

Adaptive Server 15.0.3 では、いくつかのシステム・テーブルに新しいカラムが追加されています。

テーブル	変更点
<i>sysqueryplans</i>	<p>新しいカラム：</p> <ul style="list-style-type: none"> • <i>dbid, int null</i> • <i>qptime, datetime null</i> • <i>sprocid, int null</i> • <i>hashkey2, int null</i> • <i>key1, int null</i> • <i>key2, int null</i> • <i>key3, int null</i> • <i>key4, int null</i> <p>注意：これらのカラムは、今後の使用に備えて予約されています。</p>
<i>sysprocedures</i>	<p>新しいカラム：<i>qp_setting varbinary(6) null</i></p> <p>注意：このカラムは、今後の使用に備えて予約されています。</p>

テーブル	変更点
<i>sysprocesses</i>	新しいカラム： <i>clientport</i> <ul style="list-style-type: none">クライアント・プロセスのクライアント・ポート番号を表示します。システム・プロセスの場合は0を表示します。データ型は符号なし <i>smallint</i> です。
<i>syservers</i>	カラム <i>srvnetname</i> が <i>varchar(32)</i> から <i>varchar(255)</i> に変更されています。

『リファレンス・マニュアル：テーブル』を参照してください。

Adaptive Server 15.0.2 の新機能

Adaptive Server 15.0.2 では、多数の新機能と機能強化が導入されています。セキュリティ、暗号化カラム、パフォーマンスなどの機能が強化されています。新機能には、アーカイブ・データベースへのアクセス、遅延コンパイル、積極的集約処理と消極的集約処理、ユーザ定義の SQL 関数などがあります。

暗号化カラム

Adaptive Server Enterprise 15.0.2 では、暗号化カラムの機能が強化されています。

新機能は次のとおりです。

- 管理者からデータを保護します。キーおよび暗号化カラムを独自のパスワードで保護して、DBO またはシステム管理者の権限からデータのプライバシーを保護できます。
- ログイン・パスワードで保護されたキー・コピーを使用して、アプリケーションの透過性を維持します。したがって、キー・コピーを作成して、個々のユーザに割り当てることができます。ユーザは、ログイン・パスワードを使用して自分のキー・コピーを暗号化できます。キー・コピーをログイン・パスワードに関連付けた後は、キーで暗号化されたデータにアクセスするとき、ユーザはキー暗号化パスワードを入力する必要はありません。
- キー・リカバリを実現します。パスワードを忘れてもキーへのアクセスをリカバリできます。キー所有者がリカバリ・キー・コピーを設定します。このリカバリ・キー・コピーは、パスワードを忘れたときにキーの再暗号化を行うために使用できます。
- `decrypt` パーミッションを持たないユーザにデフォルト値を返します。テーブルを作成または変更して、`decrypt` パーミッションを持たないユーザに対して、**select** 文を使用して、指定したデフォルト値を返すことができます。この機能により、プライベートなデータを不正なユーザから保護しながら、パーミッション・エラーを発生させることなく既存のアプリケーションおよびレポートを実行できます。暗号化データは、不正なユーザによって生成されたレポートには表示されません。
- 自動 `decrypt` パーミッションを制限します。**restricted decrypt permission** 設定パラメータが有効な場合、システム・セキュリティ担当者は、`decrypt` パーミッションを明示的に付与して、データへのアクセスを制限します。**restricted decrypt permission** が有効な場合、次のようになります。
 - テーブル所有者には `decrypt` パーミッションが暗黙的に付与されません。スキーマ所有者は、キーにアクセスするためにシステム暗号化パスワードに

依存するシステムであっても、ユーザ・データに自動的または暗黙的にアクセスできません。

- `sso_role` のあるユーザだけが `decrypt` パーミッションを付与できます。
`decrypt` パーミッションでは、**with grant** オプションがサポートされます。
- ビューとテーブルまたはプロシージャとテーブルの間の所有権の連鎖による暗黙的なアクセスが制限されます。
- データ型が追加されています。次の追加のデータ型を暗号化できます。*date*, *time*, *datetime*, *smalldatetime*, *money*, *smallmoney*, *big int*, *unsigned big int*, *bit*, *unichar*, *univarchar*

アーカイブ・データベースへのアクセス

データベース・ダンプを従来の読み取り専用データベース (アーカイブ・データベース) と同様に扱うことにより、データベース・ダンプ (アーカイブ) のデータ検証やデータの選択的なりカバリを行います。

従来のデータベースとは異なり、アーカイブ・データベースは実際のデータベース・ダンプを主要なディスク記憶領域デバイスとして使用し、従来の記憶領域を最小限にすることによって、データベース・ダンプのリカバリの結果生成される新しいページまたは変更済みのページを表します。データベース・ダンプには、すでに多くの (ほとんどではない場合) データベース・ページのイメージが含まれているため、Backup Server を使用してページをアーカイブから従来のデータベース記憶領域に転送しなくてもアーカイブ・データベースをロードできます。その結果、従来のデータベースよりもロードが大幅に速くなります。

アーカイブ・データベースにアクセスすると、さまざまな操作をデータベース・ダンプに対して直接実行できます。

アーカイブ・データベースでは、元のデータベース全体をコピーする必要はありません。`sp_dumpoptimize` を使用してデータベースをダンプするときに行われる最適化に応じて、データがアーカイブ・データベースに完全に格納される (データベースのすべてのページがデータベース・ダンプに格納される) か、部分的に格納されます (割り付けられたページだけがデータベース・ダンプに格納される)。

データベース・ダンプは、読み取り専用のデータベースとして存在するため、データベース管理者は次のような使いやすいつールと手法を使用してクエリを実行できます。

- 運用データベースからダンプされた最新のコピーに対する、データベース一貫性検査。これらの検査を別のサーバで実行することにより、運用環境におけるリソースの競合を避けることができる。リソースを考慮する必要のない場合は、アーカイブが作成されたのと同じサーバでアーカイブを直接検査できる。アーカイブでの検証により、リストア操作の実行前に必要な保証を得られる。

- データベース・ダンプの整合性が問題となる場合、アーカイブ・データベースにデータベース・ダンプをロードすると問題ないかどうかを簡単にテストできるため、従来のデータベースのリストアに使用される適切なデータベース・ダンプを特定するのに良いツールとなる。
- データベース・ダンプからのオブジェクト・レベルのリストア。失われたデータは、**select into** を使用してアーカイブ・データベース内のテーブルからリストア対象のローをコピーすることにより、リカバリされる。**select into** 操作は、アーカイブ・データベースを持つサーバで直接実行するか、オブジェクトのリストアが必要なサーバとは別のサーバでアーカイブ・データベースが使用可能な場合はコンポーネント統合サービス・プロキシ・テーブルを使用することにより実行できる。

また、トランザクション・ログをアーカイブ・データベースにロードでき、それによってリストア・オペレーションを実行する際に同じロード・シーケンスを適用できます。

長時間実行されているクエリの検出

Adaptive Server 15.0.2 では、長時間実行されているクエリに関する情報を収集する新しい **set** コマンドが導入されています。

set コマンドで次のパラメータを使用すると、**showplan** または他の検査用パラメータを事前に有効にしなくても、適切に実行されていないクエリに関する診断情報を収集できます。

- **tracefile** – トレース・ファイルに診断情報を保存する。
- **show_sqltext** – SQL テキストを表示する。
- **export_options** – セッション設定を保持する。

遅延コンパイル

Adaptive Server 15.0.2 では、遅延コンパイルが導入されています。オプティマイザで遅延コンパイルを使用すると、実際のランタイム値を参照するストアド・プロシージャ文をコンパイルできます。

現在、オプティマイザでは、クエリがマジック・ナンバーではなく実際のランタイム値で最適化されるように、ローカル変数およびテンポラリー・テーブルを参照するプロシージャ文のランタイム・コンパイルを実行できます。

- Adaptive Server では、検索句でローカル変数およびパラメータを参照するクエリ、同じプロシージャで作成されたテンポラリー・テーブルでジョインが使用されているクエリ、サブクエリでテンポラリー・テーブルを参照するクエリに対して遅延コンパイルが使用されます。

- 遅延コンパイルの対象となる文は、ストアド・プロシージャの初回実行時にコンパイルされます。ストアド・プロシージャの初回呼び出し時に実行されない文(たとえば、IF 句によって省略された文)は、ストアド・プロシージャの後続の実行で実際にこれらの文が実行されるまでコンパイルされません。
- 条件文がコンパイルされると、プロシージャが初回実行であるか、後続実行であるかにかかわらず、クエリ・プランに統合され、再コンパイルは行われません。
- プロシージャ・パラメータを参照するクエリは、以前は、これらのパラメータの値をストアド・プロシージャに入力したときにのみコンパイルおよび最適化されていました。バージョン 15.0.2 以降では、これらの文は、クエリの初回実行時に含まれているパラメータ値で最適化されます。ストアド・プロシージャの実行中にパラメータ値が変更された場合、現在の値が最適化に使用されます。
- グローバル・スイッチ -T7730 を使用してサーバを起動することによって、遅延コンパイルをオフにできます。

中国語と日本語文字セットの大文字と小文字を区別しない ソート順

この項では、中国語と日本語文字セットの大文字と小文字を区別しないソート順について説明します。

- EUC-GB
- GB-18030
- CP-936
- EUC-JIS
- SJIS
- DECKANJI

表 16 : 中国語 (簡体字) と日本語で使用できるソート順

言語またはスクリプト	文字セット	ソート順
中国語 (簡体字)	EUC-GB、GB-18030、CP936	汎用 (大文字と小文字を区別しない、辞書順)
日本語	EUCJIS、SJIS、DECKANJI	汎用 (大文字と小文字を区別しない、辞書順)

統計集合関数

Adaptive Server 15.0.2 では、分散と標準偏差を計算する統計集合関数が導入されています。

集合関数は、データベースに含まれるローのグループのデータを要約します。**select** 文の **group by** 句を使用してグループを作成します。

単純な集合関数 (**sum**、**avg**、**max**、**min**、**count_big**、**count** など) は、**select** リスト、および **select** 文の **having** 句、**order by** 句、**compute** 句の中だけで使用できます。これらの関数は、データベースに含まれるローのグループのデータを要約します。

Adaptive Server Enterprise は、数値データの統計的分析を行うための統計集合関数をサポートするようになりました。これらの関数としては、**stddev**、**stddev_samp**、**stddev_pop**、**variance**、**var_samp**、**var_pop** があります。

stddev と **variance** を含むこれらの関数は、クエリの **group by** 句の指定に従ってローのグループの値を計算できる集合関数です。**max** や **min** などのその他の基本的な集合関数と同様に、これらの計算は入力データ内の **null** 値を無視します。また、分析される式のドメインに関係なく、分散と標準偏差の計算では必ず IEEE の倍精度浮動小数点数が使用されます。

分散関数または標準偏差関数への入力が空のデータ・セットである場合、これらの関数は結果として **null** 値を返します。分散関数または標準偏差関数への入力が単一の値である場合、これらの関数は結果として **0** を返します。

標準偏差と分散

新しい統計集合関数とそのエイリアスについて説明します。

- **stddev_pop** (同様に **stdevp**) – 母標準偏差。グループの各ロー (**distinct** が指定されている場合、重複が削除された後に残る各ロー) に対して評価される、指定された値式の母標準偏差を計算します。これは、母分散の平方根として定義されます。
- **stddev_samp** (同様に **stdev** と **stddev**) – 標本標準偏差。グループの各ロー (**distinct** が指定されている場合、重複が削除された後に残る各ロー) に対して評価される、指定された値式の標本標準偏差を計算します。これは、標本分散の平方根として定義されます。
- **var_pop** (同様に **varp**) – 母分散。グループの各ロー (**distinct** が指定されている場合、重複が削除された後に残る各ロー) に対して評価される、指定された値式の母分散を計算します。これは、値式と値式の平均の差の 2 乗和を、グループ内のローの数で割った値として定義されます。

- **var_samp** (同様に **var** と **variance**) – 標本分散。グループの各ロー (**distinct** が指定されている場合は、重複が削除された後に残る各ロー) に対して評価される値式の標本分散を計算します。これは、値式と値式の平均の差の 2 乗和を、グループ内のローの数より 1 少ない数で割った値として定義されます。

積極的集約と消極的集約

集約処理では、大量のデータを集約値によって要約します。

集約処理は、DBMS 環境における最も有用なオペレーションの 1 つです。次の値が要約されます。

- 指定されたローのセット内のカラムの値の最小値、最大値、合計値、または平均値
- 条件に一致するロー数
- 他の統計関数

SQL では、集約処理は **min()**、**max()**、**count()**、**sum()**、**avg()** 集約関数と、**group by** および **having** 句を使用して実行されます。SQL 言語には、ベクトル集合およびスカラ集合の 2 種類の集約処理が実装されています。 **select-project-join** (SPJ) クエリで、これらの 2 種類の集約処理を示します。

```
select r1, s1
from r, s
where r2 = s2
```

ベクトル集合およびスカラ集合

Adaptive Server 15.0.2 では、ベクトル集合およびスカラ集合がサポートされています。

ベクトル集合の場合、SPJ の結果セットが **group by** 句の式によってグループ化され、**select** 句の集合関数が各グループに適用されます。クエリは、1 つのグループに 1 つの結果ローを生成します。

```
select r1, sum (s1)
from r, s
where r2 = s2
group by r1
```

スカラ集合の場合、**group by** 句はありません。SPJ の結果セット全体が同じ **select** 句の集合関数によって 1 つのグループとして集約されます。クエリは、1 つの結果ローを生成します。

```
select sum (s1)
from r, s
where r2 = s2
```

データ挿入のパフォーマンスの向上

Adaptive Server 15.0.2 では、データ挿入のパフォーマンスが最適化されています。

次の 15.0.2 機能により、データ挿入のパフォーマンスが向上します。

- 高速 **bcp** では、ノンクラスタード・インデックスまたはトリガを持つテーブルにデータをコピーして、大量のデータを挿入するときの Adaptive Server のパフォーマンスを向上させることができます。
- Adaptive Server バージョン 15.0.2 にはセッションのテンポラリ・データベース用に個別のユーザ・ログ・キャッシュ (ULC) があるので、シングル・ユーザ・データベースを持つマルチデータベース・トランザクションおよびセッションのテンポラリ・データベースでは、データベース間でユーザが切り替わった場合や次の条件をすべて満たす場合、ULC のフラッシュは必要ありません。
 - Adaptive Server が現在トランザクションをコミットしている。
 - すべてのログ・レコードが ULC にある。
 - コミット後のログ・レコードがない。

session tempdb log cache size 設定オプションを使用すると、ULC のサイズを設定して、必要なフラッシュの回数を決定できます。

- Adaptive Server がインデックスまたはデータ・ページを分割するときには、ローの一部が元のページから新しく作成されたページに移動されます。ローの移動操作はログに記録されません。Adaptive Server バージョン 15.0.2 ではディスクへの非同期書き込みを使用し、書き込みの完了を待つため、サーバをブロックする必要はありません。Adaptive Server バージョン 15.0.2 では、ユーザが何もしなくてもこれらの非同期書き込みを自動的に使用します。
- tempdb トランザクションのスループットが向上します。

- Adaptive Server バージョン 15.0.2 では、コミット後の最適化が実行されます。サーバでは、ログに対して 2 つのスキャンが実行されます。最初のスキャンではデータ・ページの割り付け解除と予約されていないページを探し、2 番目のスキャンではログ・ページの割り付け解除を探します。これらのスキャンは内部の最適化なのでユーザからは見えません。スキャンは自動的に実行され、スキャンのオン・オフを切り替えることはできません。

コミット後の最適化では、Adaptive Server はこれらのログ・レコードが含まれている後方の「次の」ログ・ページを記憶します。コミット後の段階で、Adaptive Server は、ページのレコードの処理後に、コミット後の作業が必要な「次の」ページに移動します。多数のユーザが同時にトランザクションのログを *syslogs* に記録する並列処理環境では、コミット後の最適化により不要なログ・ページの読み込みやスキャンが回避され、コミット後の処理のパフォーマンスが向上します。

ページ分割時の非同期書き込みの使用

Adaptive Server がインデックスまたはデータ・ページを分割するときには、ローの一部が元のページから新しく作成されたページに移動されます。ローの移動操作はログに記録されません。Adaptive Server 15.0.2 では、非同期書き込みを使用して一貫性を保ちます。

一貫性と持続性の両方を保つには、Adaptive Server が次の条件を満たす必要があります。

- Adaptive Server は、新しいページをディスクに書き込んでから変更された (ローが削除された) ページをディスクに書き込む。これによって、トランザクションが取り消された場合に Adaptive Server はページの前のバージョンをリストアできる。Adaptive Server は、古いページのローが失われ、ローの内容が記録されていないにもかかわらず、新しいページでこれらのローを見つけて古いページに戻すことができる。
- 新しいページがディスクに記録されてからトランザクションがコミットされるので、Adaptive Server がコミットされたデータを失うことがない。ローの移動はログに記録されていないので、新しいページのトランザクションを取り消すのは不可能であるが、トランザクションがコミットされていれば、Adaptive Server は新しいページのトランザクションを取り消す必要がない。取り消しを行う場合は、新しいページの割り付けは取り消され、リストアする更新前イメージのページは存在しない。

以前のバージョンの Adaptive Server では、同期をとりながら新しいページをディスクに書き込むことによって、これらの 2 つの条件を満たしていました。しかし、同期書き込みの復帰までサーバがブロックされることがあるので、パフォーマンス低下の原因になっていました。

Adaptive Server バージョン 15.0.2 では上記の条件を満足するディスクへの非同期書き込みを使用し、書き込みの完了を待つためにサーバをブロックする必要がありません。

Adaptive Server バージョン 15.0.2 では、ユーザが何もしなくてもこれらの非同期書き込みを自動的に使用します。

tempdb トランザクションのスループットの向上

以前のバージョンの Adaptive Server では、*tempdb* やリカバリが不要なデータベースについてはクラッシュのリカバリをサポートしていなかったため、データ・ページおよび 1 つのログ・レコード (SLR) をフラッシュしていました。

SLR は、レコードをログに記録した直後に *syslogs* へのユーザ・ログ・キャッシュ (ULC) のフラッシュを強制するログ・レコードです。SLR は、OAM 変更のために

作成され、Adaptive Server は混合ログおよび SLR としてのデータ・データベース内のアロケーション・ページに影響するログ・レコードを作成します。

- 通常のデータベースでは、SLR を含んでいる ULC はすぐにフラッシュされ、バッファの「位置決め」のときに生じる隠れたデッドロックが回避される。SLR の ULC フラッシュを回避することによって、ログ・セマフォの競合が減り、パフォーマンスが向上する。

ULC フラッシュによって、バッファの位置決めによって生じるデッドロックが回避される。Adaptive Server では、リカバリが不要なデータベースのバッファを位置決めしないので、このデッドロックが回避され、SLR の ULC をフラッシュする必要がない。

- リカバリが必要なデータベースでは、Adaptive Server はチェックポイントのときにダーティ・ページをディスクにフラッシュする。これによって、Adaptive Server がクラッシュした場合、コミットされたデータはすべてディスクに保存される。ただし、リカバリが不要なデータベースについては、Adaptive Server はクラッシュのリカバリをサポートしないで実行時のロールバックをサポートする。このためにチェックポイントでのダーティ・データ・ページのフラッシュを回避でき、パフォーマンスが向上する。
- Adaptive Server では、リカバリが不要なデータベースの先書きログをサポートしていない。先書きログによって、コミットされたトランザクションのデータは、ログの「再実行」(ログにリストされたトランザクションの再実行) およびアポートまたはロールバックされたトランザクションによって行われた変更の「取り消し」によってリカバリでき、データベースの一貫性が保たれる。先書きログは、「バッファの位置決め」メカニズムによって実装されている。Adaptive Server は、リカバリを必要としないデータベースに必ず先書きログを行うわけではないので、これらのデータベースに対してはバッファを位置決めしない。したがって、トランザクションをコミットしたときはログのフラッシュを省略できる。

コミット後の最適化

Adaptive Server バージョン 15.0.2 では、ログに対して 2 つのスキャンが実行されます。最初のスキャンではデータ・ページの割り付け解除と予約されていないページを探し、2 番目のスキャンではログ・ページの割り付け解除を探します。これらのスキャンは内部の最適化なのでユーザからは見えません。スキャンは自動的に実行され、スキャンのオン・オフを切り替えることはできません。

以前のバージョンの Adaptive Server の動作は次のとおりです。

- ページの割り付け解除、ページの予約解除、ログの割り付け解除のそれぞれに対して 1 つずつ、コミットされたトランザクションの後にログ・レコードの 3 つのスキャンを使用していた。

- ページのリンクを使用して、ログ・ページの後方スキャンを実行していた。一部のページは、コミット後の作業が必要なログ・レコードを持っていない。

コミット後の最適化では、Adaptive Server はこれらのログ・レコードが含まれている後方の「次の」ログ・ページを記憶します。コミット後の段階で、Adaptive Server は、ページのレコードの処理後に、コミット後の作業が必要な「次の」ページに移動します。多数のユーザが同時にトランザクションのログを *syslogs* に記録する並列処理環境では、コミット後の最適化により不要なログ・ページの読み込みやスキャンが回避され、コミット後の処理のパフォーマンスが向上します。最適化は、診断情報に表示されません。

クエリ・プロセッサの変更点

クエリ・プロセッサの変更により動作が変わり、クエリ・プランの選択に影響が生じる可能性があります。

遅延コンパイル

Adaptive Server バージョン 15.0.2 のクエリ・プロセッサは、ストアド・プロシージャ内の文の最適化を、その文の実行時まで遅延します。

文の最適化を遅延させると、ローカル変数の値がそれぞれの文の最適化に使用可能になるため、クエリ・プロセッサには有益です。

Adaptive Server の以前のバージョンでは、ローカル変数を使用する述部の選択性的見積りにデフォルトの推測を使用していました。

バイナリ以外の文字セットのヒストグラム補間

Adaptive Server バージョン 15.0.2 では、バイナリ文字セットと同様の正確性を選択性的見積もりで実現できます。

以前のバージョンの Adaptive Server では、デフォルトのバイナリ文字セットにおいてのみ、範囲述部の選択性的見積りに使用されるヒストグラム補間が役立っていました。その他すべての文字セットでは、ヒストグラム・セルについて 50% の選択性が見積もられました。そのため、通常 Adaptive Server は、この見積もりに関連するエラーを削減するために、文字カラム・ヒストグラムに多数のヒストグラム・セルを使用する必要がありました。

Adaptive Server バージョン 15.0.2 を使用すると、過剰な数のヒストグラム・ステップを必要としないで、バイナリ文字セットと同様の正確性を、選択性的見積もりで実現できます。このことは、範囲述部のある次のようなクエリに役立ちます。

```
select * from t1 where charcolumn > "LMC0021" and  
charcolumn <= "LMC0029"
```

指定した範囲が同じヒストグラム・セル内に収まる場合、Adaptive Serverはこの選択性をさらに正確に見積もることができます。

選択性的見積もりのヒストグラム式

Adaptive Server バージョン 15.0.2 は、単一のカラムにヒストグラムが存在する場合は、ヒストグラム見積もりをそのカラム述部に適用します。

以前のバージョンの Adaptive Server では、選択性的見積もりにデフォルトの "guesses" を使用していました。

Adaptive Server バージョン 15.0.2 は、単一のカラムにヒストグラムが存在する場合は、ヒストグラム見積もりをそのカラム述部に適用します。これにより、さらに正確なローの見積もりが可能になり、クエリ・プランのジョイン順が改善されます。

この例では、式が非常に選択的な場合、テーブル *t1* をジョイン順の冒頭に配置することが効果的です。

```
select * from t1,t2 where substring(t1.charcol, 1, 3)
= "LMC" and t1.a1 = t2.b
```

現在のオプティマイザ設定の表示

新しいフェイクテーブル *sysoptions* と新しいストア・プロシージャ *sp_options* を使用して、オプティマイザ設定を表示できます。

オプティマイザ設定を簡単に表示できるように、Adaptive Server バージョン 15.0.2 では次のものが導入されています。

- *sysoptions*— 各 **set** オプション、そのカテゴリ、現在の設定とデフォルト設定に関する情報を保管する新しいフェイクテーブル (*sysoptions* には、オプションに関する詳細情報を含むビットマップも格納されます)
- *sp_options*— オプションの値を示す新しいストア・プロシージャ

新しいセキュリティ機能

Adaptive Server 15.0.2 では、セキュリティ機能が追加されています。

AIX における 64 ビット版 Adaptive Server の PAM サポート

Adaptive Server 15.0.1 では、AIX 5.2 上での PAMUA がサポートされています。

AIX 5.2 64 ビット版の Adaptive Server バージョン 15.0.2 は、プラグ可能認証モジュールに基づくユーザ認証 (PAMUA) をサポートしています。サポート担当者に

問い合わせ、使用している IBM ホストに対応する PAM の最新パッチを入手してください。

AIX 5.2 64 ビット版の Adaptive Server 15.0.2 で PAMUA を使用するには PAM モジュールを `/etc/pam.conf` ファイル内に指定する必要があります。次に例を示します。

```
ase auth required /usr/lib/security/pam_aix
```

グローバル・ログイン・トリガの自動設定

Adaptive Server バージョン 15.0.2 以降では、ログイン・トリガでエクスポート可能なオプションを設定または解除すると、サーバが起動する時点のログイン・プロセスで反映されます。

Adaptive Server バージョン 15.0.1、12.5.4、およびそれ以前では、ログイン・トリガのオプションを有効にするには、トレース・フラグ 4073 を有効にして Adaptive Server を起動する必要があります。

現在は、ログイン・トリガでエクスポート可能なオプションを設定または解除すると、サーバが起動する時点のログイン・プロセスで反映されます。

新しい動作を無効にするには、ログイン・トリガ内で **set export_options off** を実行します。

SSL サポート

Adaptive Server 15.0.2 では、Windows 2003 x64 Enterprise Edition での SSL がサポートされています。

Adaptive Server バージョン 15.0.2 では、Windows 2003 X64 Enterprise Edition での Secure Sockets Layer (SSL) がサポートされています。Windows 2003 X64 では、OpenSSL ライブラリを使用して SSL 機能を実装します。

SSL 機能は他のプラットフォーム上での Adaptive Server と同じです。『システム管理ガイド』に説明されているように、同じ暗号スイートと事前に定義されたリストがサポートされています。Windows 2003 x64 用の Adaptive Server で使用する OpenSSL ライブラリは、SSLPlus で使用される暗号スイート名をそのまま使用して SSLPlus クライアントやサーバと情報のやり取りをします。暗号スイート名は Transport Layer Security (TLS) に準拠しています。TLS は SSL 3.0 を拡張したものであり、SSL バージョン 3.0 暗号スイートと互換性があります。

パスワードのセキュリティの改善

Adaptive Server 15.0.2 では、パスワードのセキュリティが改善されています。

Adaptive Server 15.0.2 では、既存のパスワード保護メカニズムに次のような高レベルのセキュリティが追加されています。

- ネットワークを介して送信されるパスワードの暗号化の強化
- *syslogins* (ディスク上) およびメモリに保管されるパスワードの暗号化の強化
- 新しいタイムスタンプ・データと追加されたアカウント・レポート機能によるログインとパスワードの使用の管理

監査機能の強化

Adaptive Server バージョン 15.0.2 では、セキュリティ監査機能が2つ追加されています。

システム・ストアド・プロシージャとコマンドのパスワード・パラメータを隠す

システム・ストアド・プロシージャとコマンドのパスワードは、監査レコード内でアスタリスクで置き換えることができます。

監査が設定されて有効になっているとき、**sp_audit** にオプションの **'cmdtext'** が設定されていると、監査ログ内の監査レコードではシステム・ストアド・プロシージャとコマンドのパスワード・パラメータが固定長のアスタリスク文字列で置き換えられます。

たとえば、次のコマンドを実行します。

```
sp_password 'oldpassword', 'newpassword'
```

監査が有効になっていて **sp_audit cmdtext** が設定されている場合の出力結果は次のようになります。

```
sp_password '*****', '*****'
```

これで監査ログにアクセスできる他のユーザにパスワードを見られる心配がなくなります。

失敗したログイン試行のモニタリング

login_locked 監査オプションを使用して、失敗したログイン試行をモニタします。

ログイン試行の失敗回数が所定の限度を超えたためにログイン・アカウントがロックされると、新しい監査オプションの **login_locked** と **Locked Login (value 112)** イベントが記録されます。このイベントは監査オプションの **login_locked** が設定されると有効になります。**login_locked** を設定するには、次のように入力します。

```
sp_audit "login_locked", "all", "all", "ON"
```

監査テーブルが満杯でイベントを記録できない場合は、その情報がエラー・ログに記録されます。

ホスト名とネットワークの IP アドレスが監査レコードに記録されます。監査ログを使用して **Locked Login** イベント (112) をモニタリングすると、ログイン・アカウントに対する攻撃の識別に役立ちます。

高可用性に関する考慮事項

Adaptive Server 15.0.2 でのパスワードのセキュリティの変更は、高可用性 (HA) に影響します。

HA の設定

HA を設定する前に、プライマリ・サーバとコンパニオン・サーバの **"allow password downgrade"** の値が同じになっている必要があります。 **"allow password downgrade"** の値がプライマリ・サーバとコンパニオン・サーバの両方で同じになっているかどうかをチェックするために、新しい定属性として **"allow password downgrade"** が追加されました。HA アドバイザリ・チェックは定属性の値が同じであれば成功し、異なっていれば失敗します。

HA によって変更される syslogins でのパスワードの動作

Adaptive Server 15.0.2 にアップグレードし、HA を正しく設定した後、プライマリ・サーバへの初回の接続ではユーザ・ログインのパスワードがプライマリ・サーバとコンパニオン・サーバの両方で更新されます。これでプライマリ・サーバとコンパニオン・サーバのログイン・パスワードが同じオンディスク暗号化フォーマットで同期化されます。こうしておくことで、**sp_passwordpolicy** での指定に従って **"allow password downgrade"** 期間が終了し、**sp_downgrade** によってパスワードが以前の Adaptive Server 15.0 バージョンにダウングレードされたときに、パスワードのリセットやロックを避けることができます。パスワード暗号化フォーマットを同期することによって、**sp_passwordpolicy** や **sp_downgrade** によるログイン・パスワードのリセットやロックを避けて、使用を継続できます。

モニタリング・テーブルのインストールおよび編集

Adaptive Server バージョン 15.0.2 では、モニタリング・テーブルのインストールおよび編集機能が新しく追加されています。

- モニタリング・テーブルのインストール – モニタリング・テーブルをインストールするために `installmontables` スクリプトを実行する必要がなくなりました。

Adaptive Server の 15.0.2 以前のバージョンでは、モニタリング・テーブルをインストールするために `installmontables` スクリプトを実行する必要がありました。Adaptive Server バージョン 15.0.2 では、`installmaster` スクリプトにモニタリング・テーブルのインストール機能が含まれています。

- モニタリング・テーブルへのリモート・アクセスと編集 – Sybase では、モニタリング・テーブルにリモートでアクセスする方法を示すサンプル・スクリプトとして `installmontables` を提供しています。`installmontables` を実行すると、編集手順が表示されます。

ステートメント・キャッシュ用のモニタリング・テーブル

2つの新しいモニタリング・テーブルを使用すると、ステートメント・キャッシュの内容を分析できます。

Adaptive Server のステートメント・キャッシュには、**update**、**delete**、**select** の各アドホック文に加え、再使用される可能性がある他の文の SQL テキストが格納されます。ステートメント・キャッシュが有効になっている場合、これらの文はライトウェイト・プロシージャに変換され、プランは後で再使用できるように保存されます。新しい文が発行されると、再使用するプランが Adaptive Server によってステートメント・キャッシュで検索されます。再使用するプランが見つかった場合は、文の再コンパイルが回避され、パフォーマンスの向上につながります。

Adaptive Server version 15.0.1 では、リテラルのパラメータ化が導入されているので、リテラル値が異なる以外は同一のクエリを認識して、ステートメント・キャッシュを使用するときの再コンパイル・コストを節減できます。リテラルのパラメータ化は、パフォーマンスを高めるだけでなく、測定基準や文をキャッシュに格納するときの領域を大幅に縮小するというメリットももたらします。

Adaptive Server バージョン 15.0.2 では、2つの新しいモニタリング・テーブルが追加されています。これらのテーブルを使用すると、ステートメント・キャッシュの内容を容易に分析できます。

- `monStatementCache` には、ステートメント・キャッシュの概要スナップショットが示される。
- `monCachedStatement` には、キャッシュされた各文に関する詳細情報が示される。

各テーブルのカラムには、"counter" (カラムにカウンタ値がある場合) と "reset" (`sp_sysmon` などのメカニズムを使用してカラムをリセットできる場合) という 2つの属性を指定できます。

システム・テーブルのロー・レベルのロック

Adaptive Server バージョン 15.0.2 では、パフォーマンスを向上させるために、システム・テーブルに対してロー・レベルのロックが使用されます。

Adaptive Server の 15.0.2 以前のバージョンでは、データ定義言語 (DDL) とユーティリティ・コマンドの実行時にシステム・テーブルで排他テーブル・ロックを使用していました。ロックされるシステム・テーブル Adaptive Server のセットは、実行する DDL 動作の種類によって異なりました。同時に実行されている別の DDL が同じシステム・テーブルに対して矛盾する排他テーブル・ロックを試みた場合、この DDL は、いずれかのシステム・カタログでのロックを獲得するために待機する必要がありました。これらの DDL の動作は連続して実行されました。

この方法ではテンポラリー・データベースのパフォーマンスが阻害され、DDL アクティビティもかなり高まり、その結果、カタログの競合が非常に高くなります。このため、テンポラリー・テーブルを使用するアプリケーションでは Adaptive Server のスループットが制限されていました。

Adaptive Server バージョン 15.0.2 では、次の問題を解決するためにロー・レベルのロックが使用されます。

- システム・テーブルの競合。多くの DDL とユーティリティのボトルネックの原因になります。
- *tempdb* の競合。システム・テーブルがロー・レベルでロックされるため、Adaptive Server 15.0.2 は *tempdb* の競合を削除します。
- DDL およびユーティリティ実行時の共有または排他的なテーブル・レベルのロック。旧バージョンでは大部分のシステム・テーブルをデータオンリー・ロック (DOL) に変換していましたが、DDL やユーティリティの実行時に共有または排他的なテーブル・レベルのロックが作成されていました。システム・テーブルでロー・レベルのロックを使用すると、この競合が削除されます。Adaptive Server は意図的ロックをカタログにのみ設定し、潜在的な競合を削除します (意図的ロックは、ページ・レベルまたはロー・レベルのロックがテーブルで現在保持されていることを示します)。
- DDL とユーティリティによる相互ブロッキング。Adaptive Server 15.0.2 では、DDL とユーティリティを同時に実行できます。

以前のバージョンの Adaptive Server では、テーブル・ロックを使用してシステム・カタログの同期を実現していました。Adaptive Server 15.0.2 ではテーブル・レベルの同期に意図的ロックを、ロー・レベルの同期にはロー・ロックを使用します。Adaptive Server の旧リリースではオブジェクトへの操作を実行している間にシステム・カタログ全体がロックされていたため、単一のロック要求が行われていました。しかし、Adaptive Server バージョン 15.0.2 では、システム・カタログ内の

オブジェクトに複数のローが対応している場合は、そのオブジェクトの操作の実行中に該当するすべてのローに対してロックが求められます。

この変更は、Adaptive Server 15.0.2 では同じ操作を実行するために旧リリースよりも多くのロックが必要となり、システムが必要とするロック・リソース数が増えることを意味しています。そのため、Adaptive Server をアップグレードしたら、必要に応じて **number of locks** 設定オプションを変更してください。

xmltable() 関数

xmltable() 関数は、XML テーブル内の要素から SQL テーブルを作成します。

xmltable() は、XML ドキュメントから複数の値を持つ要素のシーケンスを抽出し、これらの要素の SQL テーブルを構築します。**xmltable()** を 1 回呼び出すことは、繰り返しのたびに **xmlextract()** を複数回呼び出す Transact-SQL ループに代わるものです。**xmltable()** は抽出テーブル (別の SQL クエリの **from** 句で指定されたカッコで囲んだサブクエリ) として呼び出されます。**xmltable()** の呼び出しは、**xmltable()** によって生成されたテーブルの各ローに対して **xmlextract()** 式を 1 回実行するのと同様です。

xmltable() は **xmlextract()** を一般化したものです。両方の関数とも関数の引数である XML ドキュメントから抽出したデータを返します。これらの違いは次のとおりです。

- **xmlextract()** は、1 つの XPath クエリによって識別されたデータを返す。
- **xmltable()** は、XPath クエリによって識別されたデータのシーケンス、またはロー・パターンを抽出し、そのシーケンスの各要素から他の XPath クエリのリストによって識別されたデータ、またはカラム・パターンを抽出する。SQL テーブルのすべてのデータを返す。

再配置ジョイン

ローカル・テーブルとリモート・テーブル間のジョインをリモート・サーバに配置できる再配置ジョインが導入されています。

Adaptive Server バージョン 15.0.2 では、ローカル・テーブルとリモート・テーブル間のジョインをリモート・サーバに再配置できる再配置ジョインが導入されています。リモート・システムでの再配置ジョインは、ローカル・テーブルを参照する、動的に作成されたプロシキ・テーブルで実行されます。リモート・システムでこのジョインが実行されると、大量のネットワーク・トラフィックが回避されます。

ユーザ定義の SQL 関数

Adaptive Server 15.0.2 では、ユーザ定義の SQL 関数が導入されています。

create function を使用してユーザ定義関数を作成し、**drop function** を使用してユーザ定義関数を削除します。

スカラ関数に次の要素を含めることができます。

- 関数に対してローカルであるデータ変数およびカーソルを定義する **declare** 文
- 関数に対してローカルなオブジェクトに割り当てられた値 (たとえば、**select** コマンドまたは **set** コマンドを含むテーブルに対してローカルなスカラおよび変数に割り当てられた値)
- 関数で宣言、オープン、クローズ、割り付け解除されたローカル・カーソルを参照するカーソル操作
- フロー制御文
- **set** オプション (関数のスコープでのみ有効)

Adaptive Server は、データをクライアントに返すスカラ関数に **fetch** 文を許可しません。次のものを含めることはできません。

- データをクライアントに返す **select** または **fetch** 文
- **insert**、**update**、または **delete** 文
- **dbcc**、**dump**、**load** コマンドなどのユーティリティ・コマンド
- **print** 文
- **rand**、**rand2**、**getdate**、または **newid** を参照する文

ローカル変数にのみ値を割り当てる **select** または **fetch** 文を含めることができます。

instead of トリガ

instead of トリガを使用して、デフォルトのトリガ・アクションを上書きします。

ビューは、通常、論理データベース・スキーマを物理スキーマから分離するために使用されます。**instead of** トリガをビューで定義して、**update**、**insert**、または **delete** 文の標準的なアクションを置き換えることができます。**instead of** トリガを使用すると、更新不可のビューを含め、すべてのビューを更新できるようになります。

instead of トリガは、トリガ文 (**insert**、**update**、**delete**) のデフォルトのアクションを上書きして、ユーザ定義のアクションを実行する特殊なストアド・プロシージャです。

instead of トリガは、**for** トリガと同様に、特定のビューでデータ修正文が実行されるたびに実行します。for トリガはテーブルの **insert/update/delete** 文の後で起動するので、**after** トリガと呼ばれることもあります。1つの **instead of** トリガは1つの特定のトリガ・アクションに適用できます。

```
instead of update
```

また、複数のアクションに適用することもできます。その場合は、同じトリガがリスト上のすべてのアクションを実行します。

```
instead of insert,update,delete
```

for トリガと同様に、**instead of** トリガは、トリガがアクティブな間に変更されたレコードを論理的な **inserted** テーブルと **deleted** テーブルを使用して格納します。これらのテーブルの各カラムは、トリガで参照されるベース・ビューのカラムに直接マップします。たとえば、V1 という名前のビューに C1、C2、C3、C4 のカラムがあると、トリガによってカラム C1 と C3 だけが変更される場合でも、**inserted** テーブルと **deleted** テーブルには、4つのカラムすべての値が含まれます。Adaptive Server は **inserted** テーブルと **deleted** テーブルをメモリに存在するオブジェクトとして自動的に作成して管理します。

instead of トリガを使用すると、ビューが更新をサポートできるようになり、バッチの一部を拒否してその他を成功させるコード・ロジックを実装できます。

instead of トリガはデータ修正文1つにつき1回だけ起動します。**while** ループを含む複雑なクエリは、**update** 文または **insert** 文を何度も繰り返し、そのたびに **instead of** トリガを起動できます。

Adaptive Server 15.0.2 でのシステムの変更点

Adaptive Server 15.0.2 では、コマンド、関数、ユーティリティ、システム・プロシージャ、システム・テーブル、設定パラメータ、グローバル変数に変更されています。

トレース・フラグ

Adaptive Server 15.0.2 では、トレース・フラグが追加されています。

- 15340 は、advanced_aggregation の設定に関係なくサーバ・ワイドに有効化します。
- 15341 は、advanced_aggregation の設定に関係なくサーバ・ワイドに無効化します。

コマンド

Adaptive Server 15.0.2 では、コマンドが変更されています。

表 17: 変更されたコマンド

コマンド	変更点
disk init, disk reinit	<p>ブロック・デバイスを使用しないように推奨されているプラットフォーム上でブロック・デバイスを作成しようとする、警告メッセージが表示されます。</p> <p>HP-UX、Windows、Linux プラットフォームでのみデータベース・デバイスとしてブロック・デバイスを使用するようにおすすめします。</p>

コマンド	変更点
<p>create proxy table, create table at remote server, alter table.</p>	<p>SQL UDF をサポートしていません。</p>

create proxy table, create table at remote

コマンド	変更点
disk init, disk resize	<i>skip_alloc</i> を true に設定すると、ページは 0 初期化されません。 <i>skip_alloc</i> のデフォルト設定は false です。このパラメータは、Windows 以外のファイル・システムおよび Windows ロー・システムで作成されたデバイスでサポートされています。
dump transaction	dump transaction には、 string および char_variable パラメータを含めることができます。ただし、 string および char_variable オプションを指定して、 begin transaction 、 commit 、 connect to 、 declare cursor 、 rollback 、 dbcc 、 use 、またはネストされた execute コマンドを実行することはできません。

s
e
r
v
e
r
,
a
l
t
e
r

t
a
b
l
e
.

『リファレンス・マニュアル：コマンド』を参照してください。

set コマンドの変更

Adaptive Server 15.0.2 では、**set** コマンドにいくつかの変更が加えられています。

set コマンドの変更点
set advanced_aggregation では、セッション・レベルで高度集約を有効化または無効化します。
set switch では、トレース・フラグおよびスイッチ名をローカルに設定することも、サーバ・ワイドに設定することもできます。
set パラメータを使用してストアド・プロシージャを作成したり、Transact-SQL バッチでこのパラメータを実行したりするときの一部の抽象プランの set パラメータのコンパイル時の動作が変更されました。
set コマンドには、 string パラメータおよび char_variable パラメータを含めることができます。

ユーティリティ

Adaptive Server 15.0.2 では、**ddlgen** ユーティリティの機能が強化されています。

変更点	説明
ddlgen でパスワードの入力が要求される	15.0.2 より前のバージョンでは、 -P パスワード・パラメータを指定しないと、エラーが返されていました。 ddlgen ではパスワードの入力が要求されるようになりました。
ddlgen では 15.0.2 より前の暗号化と 15.0.2 の暗号化の両方がサポートされる	構文の例を次に示します。 ddlgen -Usa -P -Sserver -TEK -NSampleKeysDB.dbo.ssn_key

変更点	説明
-XOD フラグ指定なしの ddlgen	次の2つのことが発生する可能性があります。 <ul style="list-style-type: none"> 暗号化キーの作成時にパスワードを指定しないと、ddlgen によりパスワードなしの DDL が生成されます。 暗号化キーの作成時にパスワードを指定すると、ddlgen によりデフォルトのパスワードが生成されます。
-XOD フラグを指定した ddlgen	システム暗号化パスワードが生成されます。
キー・コピーのサポート	ddlgen では、キー・コピー用とベース・キー用の DDL を生成できます。

『ユーティリティ・ガイド』を参照してください。

システム・ストアド・プロシージャ

Adaptive Server 15.0.2 では、システム・プロシージャが追加および変更されています。

表 18 : 新しいシステム・ストアド・プロシージャ

システム・ストアド・プロシージャ	説明
sp_downgrade	以前の 15.0.x リリースへのダウングレードの準備ができているかどうかを検証します。Adaptive Server 15.0.2 によって変更されたシステム・カタログの変更もダウングレードされます。

システム・ストア ド・プロシージャ	説明
sp_spaceusage	テーブル、インデックス、トランザクション・ログの領域使用率に関するレポートを作成したり、データベースのテーブルおよびインデックスの断片化量の推定値を計算したりします。推定値は、データ・ローとインデックス・ローの平均の長さとしてテーブル内のローの数を使用して計算されます。領域使用率および断片化データをアーカイブしておき、将来、レポートや傾向分析に使用できます。 sp_spaceusage プロシージャでは、 help 、 display 、 archive 、 report など、Adaptive Server の現在の領域の使用率のレポートを作成するための多数のアクションがサポートされています。

表 19 : 変更されたシステム・ストアド・プロシージャ

システム・ストア ド・プロシージャ	変更点
sp_autoformat	<i>int</i> (<i>smallint</i> 、 <i>bigint</i> 、 <i>tinyint</i> 、 <i>unsigned int</i>)、 <i>numeric</i> 、 <i>money</i> 、 <i>date/time</i> 、 <i>float</i> 、 <i>real</i> 、 <i>double</i> 精度のデータ型のカラムが受け入れられるようになりました。
sp_changedbowner	データベースの所有者を変更します。今回、 <i>sa_role</i> 権限または <i>sso_role</i> 権限を使用してデータベースの所有者を変更できるようになりました。そのデータベースのスレッシュホルドの所有者も、指定されたユーザーに変更されます。
sp_checksourc	ユーザー定義関数のテキストを暗号化します。
sp_configure	デフォルト値以外の設定を表示するようになりました。
sp_depends	ユーザー定義関数のオブジェクトの依存性の有無をチェックします。
sp_deviceattr	ファイル・システム上のデータベース・デバイスに対して dsync オプションが無効になっている場合、警告メッセージを表示します。

システム・ストア ド・プロシージャ	変更点
sp_displaylogin	<p>変更点は次のとおりです。</p> <ul style="list-style-type: none"> ワイルドカード式とサーバ・ユーザ ID の両方をサポートし、一致するログイン情報を表示します。 <pre>sp_displaylogin ['user_id' '[loginame wildcard]'</pre> <ul style="list-style-type: none"> user_id – 表示するログイン情報の所有者の ID (<i>suid</i>) wildcard – 検索に使用するワイルドカード文字 <p>56 という <i>suid</i> を持つユーザのログイン・アカウントを表示します。 <pre>sp_displaylogin '56'</pre> <p>"st" で始まるすべてのユーザのログイン・アカウントの情報を表示します。 <pre>sp_displaylogin 'st%'</pre> </p></p>
sp_droplogin	<p>sp_droplogin が、任意のデータベース内の <i>sysusers</i> テーブルでログイン <i>suid</i> を参照しているユーザが存在しているためにログインを削除できない場合は、参照が検出されたデータベースの名前がエラー・メッセージに表示されます。</p>
sp_help	<p>ユーザ定義関数に関する情報を表示します。</p>
sp_helpdevice	<p>sp_helpdevice の <i>description</i> カラムには、デバイス・タイプに関する情報が表示されます。デバイス・タイプは、raw device、block device、または file system device のいずれかです。</p>
sp_helprotect	<p>sp_helprotect の新しいオプション permission_name で、所定のデータベースについて許可されている特定のパーミッションに関する情報 (付与者の名前、付与されたユーザの名前、テーブル/カラム名、付与可能性) を表示できます。</p>
sp_hidetext	<p>ユーザ定義関数のテキストを暗号化します。</p>
sp_locklogin	<p>sp_locklogin の lock オプションを number of inactive days の値を設定して使用すると、指定された期間に認証を受けなかった非アクティブなアカウントをロックできます。次の例では、過去 60 日間に認証を受けなかったすべてのログイン・アカウントをロックします。</p>

システム・ストア・プロシージャ	変更点
sp_modifylogin account	'max failed_logins' オプションの新しい値は、認証に失敗するたびに <i>syslogins</i> カラムの logincount のログイン失敗回数は更新されても、アカウントはロックされないことを示します。
sp_modifystats	システム管理者や、このプロシージャを実行してターゲット・テーブルの統計を更新するパーミッションを付与されたすべてのユーザは、 <i>sysstatistics</i> のカラムの密度値を変更できません。
sp_monitorconfig	result_tabl_name のテーブル名を渡すときにこのテーブルがまだ作成されていない場合は、結果セットを格納するテーブルを作成します。
sp_passwordpolicy	<ul style="list-style-type: none"> sp_passwordpolicy の set コマンドと clear コマンドが監査イベント 115 「パスワード管理」を通して監査されるようになりました。 追加の構文は次のとおりです。 <pre> sp_passwordpolicy "enable last login updates", "allow password downgrade", "regenerate keypair", "expire login passwords", "[login_name wildcard]" "expire role passwords", "[role_name wildcard]" "expire stale login passwords", "datetime" "expire stale role passwords", "datetime" "maximum failed logins", -1 </pre>
sp_fixindex	1つのインデックスではなく、インデックスのセットに対して作用するようになりました。 sp_fixindex は、ターゲット・テーブルに配置インデックスまたはクラスタード・インデックスが含まれている場合、データ・レイヤを再構築します (システム・テーブルの配置インデックスまたはクラスタード・インデックスを処理している間、未使用領域を再利用します)。
sp_sendmsg	以前のリリースでは、このシステム・プロシージャを使用して送信できるメッセージの最大長は 255 文字でした。Adaptive Server リリース 15.0.2 では、 sp_sendmsg で送信可能なメッセージの最大長は 4096 文字となっています。

システム・ストア ド・プロシージャ	変更点
sp_who	新しいカラム <i>tempdbname</i> には、すべてのアクティブ・セッションのテンポラリ・データベース名が表示されます。
sp_helptext	ユーザ定義関数のテキストをレポートします。numlines および printops パラメータが導入されています。
sp_ldapadmin	新しいパラメータ set_max_ldapua_desc 、 set_num_retries 、 set_log_interval が導入されています。
sp_monitor	event および help パラメータの機能が強化されています。
sp_tempdb	show および who パラメータが導入されています。

『リファレンス・マニュアル：プロシージャ』を参照してください。

システム・テーブル

Adaptive Server 15.0.2 では、システム・テーブルが追加および変更されています。

表 20 : 新しいシステム・テーブル

システム・テーブル	説明
<i>sysoptions</i>	<i>sysoptions</i> は、 sp_options によるクエリの対象となるフェイクテーブルです。カラム名には大文字と小文字の区別があります。

表 21 : 変更されたシステム・テーブル

システム・テーブル	変更点
<i>sysquerymetrics</i>	Adaptive Server バージョン 15.0.2 では、ユーザ ID 間で共有される測定基準の数が増えました。その結果、 <i>sysquerymetrics</i> (<i>sysqueryplans</i> テーブルのビュー) のエントリ数が減り、異なるユーザ ID 間の同一クエリの測定基準が自動的に統合されるようになりました。

Adaptive Server 15.0.2 でのシステムの変更点

システム・テーブル	変更点
<i>syscolumns</i>	<p>暗号化カラムに <i>decrypt_default</i> が付加されていることを示すステータス・ビットを <i>status2</i> カラムに追加します。</p> <p>0x00001000 (4096) – カラムに復号化のデフォルト値が含まれています。</p>
<i>sysobjects</i>	<p><i>sysobjects</i> テーブルの <i>type</i> カラムには、復号化のデフォルト値を持っている各オブジェクトの "DD" というエントリが含まれています。</p>
<i>sysaudits</i>	<p>変更点は次のとおりです。</p> <ul style="list-style-type: none"> • <i>Alter Encryption Key</i> 監査イベント名が <i>AEK As/Not Default</i> に変更されました。 • Adaptive Server リリース 15.0.2 では、次の監査イベントおよび番号がサポートされています。 <ul style="list-style-type: none"> • 118 – AEK Modify Encryption • 119 – AEK Add Encryption • 120 – AEK Drop Encryption • 121 – AEK Modify Owner • 122 – AEK Key Recovery

システム・テーブル	変更点
<i>sysattributes</i>	<p><i>sysattributes</i> の変更点は次のとおりです。</p> <ul style="list-style-type: none"> • 次の 2 つのクラスが追加されています。 <ul style="list-style-type: none"> • クラス 31 <i>allow password downgrade</i> – 1 に設定すると、<i>allow password downgrade</i> は、Adaptive Server リリース 15.0 およびそれより以前のリリースとの互換性を確保するためのログイン・パスワードの特殊な処理を有効にします。 • クラス 32 <i>enable last login updates</i> – 1 に設定すると、<i>enable last login updates</i> は、前回のログイン日を格納するようにシステム・テーブルを有効にします。 • <i>sysattributes</i> には、デフォルトの復号化方式に関する情報が含まれます。カラムに関する変更点は次のとおりです。 <ul style="list-style-type: none"> • <i>attribute</i> – タイプが <i>EC</i> でクラスが 25 であるオブジェクトについて、値が 1 (DECRYPT-DEFAULT_ID) の暗号化カラムに対するデフォルトの復号化方式を指定します。 • <i>object</i> – 復号化デフォルト ID を格納します。 • <i>object_info_1</i> – 復号化デフォルト値を定義している暗号化カラムを持つテーブルのテーブル ID を格納します。 • <i>object_info2</i> – 復号化デフォルト値を格納する暗号化カラムの colid を指定します。

システム・テーブル	変更点
<i>sysencryptkeys</i>	<p><i>sysencryptkeys</i> の変更点は次のとおりです。</p> <ul style="list-style-type: none"> 新しい型を次に示します。 <ul style="list-style-type: none"> <i>EK_KEYCOPY</i> – 0x0010 <i>EK_KEYBASE</i> – 0x0020 <i>EK_RECOVERY</i> – 0x0040 新しいステータス・ビットを次に示します。 <ul style="list-style-type: none"> <i>EK_KEYRECOVERY</i>(0x00000004) – 紛失したパスワードを保護するために暗号化されたキー <i>EK_LOGINACCESS</i>(0x00000008) – ログイン・アクセスのために暗号化されたキー <i>EK_LOGINPASS</i>(0x00000010) – ログイン・パスワードを使用して暗号化されたキー <i>EK_USERPWD</i>(0x00000100) – ユーザ暗号化パスワードを使用して暗号化されたキー <i>uid</i> カラムの説明が変更されました。ユーザ・アクセスまたはキー・リカバリ・ローが該当します。<i>uid</i> には、現在のローに関連付けられているデータベース・ユーザを識別するユーザ ID (<i>uid</i>) が含まれます。以前のバージョンの Adaptive Server では、<i>uid</i> カラムは使用されませんでした。
<i>syslogins</i>	<p>以前のリリースでは、<i>password</i> カラムの最大長は 30 バイトでした。Adaptive Server リリース 15.0.2 では、<i>password</i> カラムの最大長は 128 バイトとなっています。</p> <p>新しいカラムは <i>lastlogindate</i>、<i>crdate</i>、<i>locksuid</i>、<i>lockreason</i>、<i>lockdate</i> です。</p>
<i>sysssvroles</i>	<p>以前のリリースでは、<i>password</i> カラムの最大長は 30 バイトでした。Adaptive Server リリース 15.0.2 では、<i>password</i> カラムの最大長は 128 バイトとなっています。</p>

『リファレンス・マニュアル：テーブル』を参照してください。

設定パラメータ

Adaptive Server 15.0.2 では、設定パラメータが追加および変更されています。

表 22 : 新しい設定パラメータ

設定パラメータ	説明
enable merge join	サーバ・レベルのマージ・ジョインを有効または無効にします。
cost of a logical io	1 つの論理 I/O のコストを指定します。
cost of a physical io	1 つの物理 I/O のコストを指定します。
cost of a cpu unit	1 つの CPU 処理のコストを指定します。
enable encrypted columns	暗号化カラム機能を有効または無効にします (暗号化カラムのライセンスを保有している場合)。
max online q engines	MQ Series が必要です。 max online q engines では、オンラインで実行できる Q エンジンの最大数を指定します。
metrics elap max	このクエリの経過時間が metrics elap max の値より小さい場合、このクエリに関連付けられている測定基準はシステム・テーブルに書き込まれません。 enable metrics capture がオンである必要があります。
metrics exec max	このクエリの実行時間が metrics exec max の値より小さい場合、このクエリに関連付けられている測定基準はシステム・テーブルに書き込まれません。 enable metrics capture がオンである必要があります。
metrics lio max	このクエリの論理 I/O 時間が metrics lio max の値より小さい場合、このクエリに関連付けられている測定基準はシステム・テーブルに書き込まれません。 enable metrics capture がオンである必要があります。
metrics pio max	このクエリの物理 I/O 時間が metrics pio max の値より小さい場合、このクエリに関連付けられている測定基準はシステム・テーブルに書き込まれません。 enable metrics capture がオンである必要があります。

設定パラメータ	説明
min pages for parallel scan	大量のクエリを並列でスキャンできます。
not password encryption read	Adaptive Server 認証が使用されているすべての受信ログイン認証要求で、ネットワークで転送されるログイン・パスワードが暗号化されるようにします。
number of Q engines at startup	MQ Series に必要です。オンラインで実行できる Q エンジンの最大数を指定します。
prod-consumer overlap factor	最適化に影響します。Adaptive Server では、group by アルゴリズムが変更されています。並列プランに set statistics IO を使用することはできません。
send doneinproc tokens	dbcc tune オプション doneinproc に代わるものです。

表 23 : 変更された設定パラメータ

設定パラメータ	変更点
max async i/os per engine	デフォルト値が 2147483647 から 1024 に変更されました。
maximum failed logins	新しい値 "-1" は、認証に失敗するたびに <i>syslogins</i> カラムの <i>logincount</i> のログイン失敗回数が更新されても、アカウントはロックされないことを示します。
print deadlock information	新しいパラメータ値 "2" が追加されました。この値では、デッドロック情報の概要をエラー・ログに出力できます。

『リファレンス・マニュアル：テーブル』を参照してください。

関数

Adaptive Server 15.0.2 では、関数が追加および変更されています。

表 24 : 新しい関数

関数	説明
<code>authmec()</code>	ログイン・サーバ・プロセスIDセッションで使用された認証方法を返します。
<code>index_name()</code>	インデックス ID、データベース ID、およびインデックスが定義されているオブジェクトを指定すると、インデックスの名前を返します。
<code>hashbytes()</code>	固定長のハッシュ値式を生成します。

表 25 : 変更された関数

関数	変更点
<code>used_pages()</code>	クラスタード・インデックスを持つ全ページ・ロック・テーブルでは、値が <code>indid = 0</code> の場合、 <code>used_pages()</code> にはデータ・レイヤの使用ページ数のみが渡されます。 <code>indid = 1</code> を渡した場合は、データ・レイヤとクラスタード・インデックス・レイヤの使用ページ数が返されます。
関数が作成されると、Adaptive Server は、この関数が SQL UDF か SQLJ UDF かをチェックします。SQLJ UDF である場合、Adaptive Server は "sa" パーミッションの有無をチェックします。SQL 関数である場合、Adaptive Server は <code>create function</code> 権限の有無をチェックします。	

グローバル変数

Adaptive Server 15.0.2 では、`@@lastlogindate` グローバル変数が導入されています。

表 26 : 新しいグローバル変数

グローバル変数	説明
<code>@@lastlogindate</code>	<p>グローバル T-SQL 変数の <code>@@lastlogindate</code> は各ユーザ・ログイン・セッションで使用できます。 <code>datetime</code> データ型です。その値は、現在のセッションが確立される前のログイン・アカウントの <code>lastlogindate</code> カラム内の値です。この変数は各ログイン・セッションに特有のもので、そのアカウントへの前回のログイン日時を知るために各セッションで使用できます。以前に使用されたことのないアカウントの場合、または <code>"sp_passwordpolicy 'set', enable last login updates"</code> が 0 に設定されている場合、<code>@@lastlogindate</code> の値は NULL となります。</p>

表 27 : 変更されたグローバル変数

グローバル変数	変更点
<code>@@opttimeoutlimit</code>	<p>以前のバージョンの Adaptive Server ユーザ・ドキュメントでは、クエリ最適化のための現在の最適化タイムアウト制限を表示するサーバ・グローバル変数として <code>@@opttimeout</code> を取り上げていました。</p> <p>これは誤りです。クエリ最適化のための現在の最適化タイムアウト制限を表示するグローバル変数の正しい名前は <code>@@opttimeoutlimit</code> です。</p>

Adaptive Server 15.0.1 の新機能

Adaptive Server 15.0.1 では、抽象プランに対するいくつかの機能強化が導入され、SQL クエリ内のリテラル値が自動的にパラメータ記述に変換されるようになっていきます。

抽象プランの変更

Adaptive Server 15.0.1 では、抽象プランの機能が強化されています。

機能強化は次のとおりです。

- 抽象プラン構文が拡張され、以前はセッション・レベルでのみ使用可能だったものがいくつかのクエリ・レベル設定で使用できるようになりました。
- Adaptive Server は、拡張された抽象プラン構文の中で `h_join` キーワードと `hash_join` キーワードの両方を受け付けます。
- `set` コマンドでは、現在の最適化目標設定のオンとオフを切り替える `opt criteria` パラメータがサポートされています。

新しいクエリ・レベル設定

オプティマイザが選択したクエリ・プランを強制的に実行するために Adaptive Server Enterprise で使用される抽象プラン構文が拡張され、以前はセッション・レベルでのみ使用可能だったいくつかのクエリ・レベル設定が使用できるようになりました。

最適化基準は、セッション・レベルでは次の `set` 文により処理されます。

```
set
    nl_join|merge_join|hash_join|..
    0|1
```

`use ...` 抽象プラン構文が拡張され、抽象プランの抽出テーブルの前にいくつでも `use` 形式を使用できるようになりました。以前は、抽出テーブルのある抽象プランでは `optgoal` と `opttimeout` をともに使用することはできませんでした。たとえば、次の文は、1つのクエリ内では `optgoal` 文と分離する必要がありました。

```
select ...
    plan
    "(use opttimeoutlimit 10) (i_scan r)"
```

ただし、Adaptive Server 15.0.1 では、次の2つの方法で同一の抽象プラン内に複数の文を共存させることができます。

- 次の例のように複数の **use** 文を使用する。

```
select ...
      plan
      "(use optgoal allrows_dss)
       (use nl_join off) (...)"
```

- 次の例のように 1 つの **use** 形式内に複数の項目を含める。

```
select ...
      plan
      "(use (optgoal allrows_dss) (nl_join off)) (...)"
```

クエリ・レベルでは、**use ...** 抽象プラン構文で最適化目標 (**opt_goal**) または最適化タイムアウト (**opttimeout**) 設定を使用します。セッション・レベルでは、**set plan ...** 構文で次の設定を使用します。

- 最適化目標
- 最適化タイムアウト

抽象プランおよびオプティマイザ基準でのオペレータ名の配置

Adaptive Server は、抽象プランの中で **h_join** キーワードと **hash_join** キーワードの両方を受け付けます。

アルゴリズム名は、抽象プランおよび **set** コマンドでの使用方法によって異なります。たとえば、ハッシュ・ジョインは、抽象プランでは **h_join** ですが、**set** コマンドでは **hash_join** になります。拡張された抽象プラン構文ではどちらのキーワードも受け付けられます。次に例を示します。

```
select ...
      plan
      "(h_join (t_scan r) (t_scan s))"
```

上記の構文は次の構文と同等です。

```
select ...
      plan
      "(hash_join (t_scan r) (t_scan s))"
```

および

```
select ...
      plan
      "(use h_join on)"
```

および

```
select ...
      plan
      "(use hash_join on)"
```

テーブル抽象プランが存在する場合、次のようにそのプランが優先されます。

```
select
from r, s, t
...
      plan
```

```
"(use hash_join off)(h_join (t_scan r)
(t_scan s))"
```

このクエリでは **r** および **s** スキャンに対して `hash_join` が使用されていますが、**t** のジョインの場合は、テーブル抽象プランで指定されていなかったため、`use` 抽象プラン形式で指定されているように `hash_join` は使用されていません。

オプティマイザ基準 `set` 構文の拡張

`set` コマンドでは、現在の最適化目標設定のオンとオフを切り替える `opt criteria` パラメータがサポートされています。

0 または 1 構文を使用する `set <opt criteria>` 文では `on/off/default` を指定できます。ここで、`default` はこの最適化基準に対して現在の最適化目標設定を使用していることを示します。

『リファレンス・マニュアル：コマンド』を参照してください。

リテラルのパラメータ化

Adaptive Server バージョン 15.0.1 では、SQL クエリのリテラル値が自動的にパラメータ記述に変換されます (変数と同様の使用)。

Adaptive Server のこれまでのバージョンでは、1 つ以上のリテラル値のみが異なる 2 つのクエリは、ステートメント・キャッシュに 2 つの別個のクエリ・プランを格納するか、`sysqueryplans` に 2 つのローを追加していました。たとえば、次のクエリに対するクエリ・プランは、ほぼ同一であっても別個に格納されていました。

```
select count(*) from titles where total_sales > 100
select count(*) from titles where total_sales > 200
```

Adaptive Server バージョン 15.0.1 では、SQL クエリのリテラル値が自動的にパラメータ記述に変換されるようになりました (変数と同様の使用)。`sp_configure` の新しいオプションではこの機能がサポートされています。このオプションは `enable literal autoparam` です。

`enable literal autoparam` をサーバ・ワイドに有効または無効にするには、次のように指定します。

```
sp_configure "enable literal autoparam", [0 | 1]
```

ここで、1 を指定するとリテラルがパラメータに自動的に変換され、0 を指定すると変換機能が無効になります。デフォルトは 1 です。

セッション・レベルでは、リテラルのパラメータ化は次の `set` コマンドで設定できます。

```
set literal_autoparam [off | on]
```


Adaptive Server 15.0.1 でのシステムの変更点

関数

Adaptive Server 15.0.1 では、3つの新しい関数が導入されています。

関数	説明
<code>isdate</code>	入力式が有効な <i>datetime</i> 値かどうかを確認します。
<code>isnumeric</code>	式が有効な <i>numeric</i> データ型かどうかを確認します。
<code>partition_object_id</code>	指定されたパーティションIDとデータベースIDのオブジェクトIDを表示します。

『リファレンス・マニュアル：コマンド』を参照してください。

設定パラメータ

Adaptive Server 15.0.1 では、設定パラメータが追加および変更されています。

表 28 : 新しい設定パラメータ

設定パラメータ	説明
<code>startup delay</code>	サーバの起動時のどの段階で RepAgent を起動するかを制御します。デフォルトでは、RepAgent は Adaptive Server と同時に起動します。Adaptive Server は、待機時間を示すメッセージをエラー・ログに書き込みます。
<code>enable literal autoparam</code>	リテラルのパラメータ化をサーバ・ワイドに有効または無効にします。

設定パラメータ	説明
cis idle connection timeout.	<p>指定された秒数よりも長い時間未使用になっているリモート・サーバへの CIS 接続がないか確認するように Adaptive Server を設定します。Adaptive Server は未使用の接続を削除し、それらのリソースを再割り付けします。</p> <p>指定する数値は秒単位ですが、ハウスキーピング・タスクは 1 分間に 1 回ウェイクアップするため、アイドル接続は設定された値よりもかなり長くアイドル状態になる場合があります。トランザクションが接続でアクティブな場合、アイドル接続は削除されず、ユーザが接続にアクセスするためにコマンドを実行すると自動的に接続が再確立されます。</p>
sproc optimize timeout limit	Adaptive Server がシステム・プロシージャを最適化するために費やすことができる時間を、予想実行時間に対する割合として指定します。

表 29 : 変更された設定パラメータ

設定パラメータ	変更点
optimization timeout limit	optimization timeout limit に指定できる値の範囲が変更されました。バージョン 15.0.1 では、0 ~ 1000 です。値 0 は、最適化のタイムアウトがないことを示します。
max parallel degree	max parallel degree を 1 (有効) に設定すると、Adaptive Server は逐次クエリ実行を強制し、オプティマイザはこの設定が無効にされている場合よりも並列度の高いプランを選択することがあります。

設定パラメータ	変更点
number of worker processes	number of worker processes をワーカー・スレッド・プールの十分な数のスレッド数に設定していない場合、Adaptive Server はより少ない数のワーカー・スレッドを使用するように実行時にクエリ・プランを調整します。Adaptive Server が実行時にクエリを調整できない場合、クエリは逐次モードで再コンパイルされます。ただし、十分な数のワーカー・スレッドがない場合、 alter table および execute immediate コマンドはアボートされます。

『システム管理ガイド 第1巻』を参照してください。

コマンド

Adaptive Server 15.0.1 では、**alter table**、**create index**、**create existing table**、**update statistics**、**set** コマンドに構文と他のいくつかの変更が加えられています。

表 30 : 変更されたコマンド

テーブル	変更点
alter table	alter table コマンドは、1つ以上のリスト分割または範囲分割を削除するために使用します。 alter table を使用してハッシュ分割またはラウンドロビン分割を削除することはできません。構文は次のとおりです。 <pre>alter table table_name drop partition partition_name [, partition_name]...</pre>
create index	範囲分割テーブル、リスト分割テーブル、ハッシュ分割テーブルでユニークなローカル・インデックスを作成すると、インデックス・キー・リストは、分割キー・リストのスーパーセットになります。

テーブル	変更点
create existing table	<p>RPC が現在の接続と別の接続のどちらを使用するかを決定する構文が追加されています。</p> <pre>create existing table (<column_list>) EXTERNAL [non_transactional transactional] PROCEDURE at 'location'</pre> <ul style="list-style-type: none"> • non_transactional – RPC の実行には別の接続を使用します。 • transactional – RPC の実行には既存の接続を使用します。 <p>デフォルトは transactional です。</p>
update statistics	<p>Adaptive Server 15.0.1 では、グローバル・インデックスに対して update statistics を実行する機能が追加されています。</p> <pre>update table statistics table_name [partition data_partition_name] [index_name [partition index_partition_name]]</pre> <p>update table statistics を実行すると update statistics 実行の I/O コストが発生するため、カラム統計とテーブル統計の両方を生成するには update statistics を使用します。</p> <p>グローバル統計は、グローバル・インデックスを作成してから削除することで生成できます。</p>

表 31 : set コマンドの新しいオプション

set コマンドの新しいオプション	説明
set literal_autoparam on off	リテラルのパラメータ化をセッション・レベルで有効または無効にします。
set opttimeoutlimit	opttimeoutlimit の値の範囲が 0 ~ 4000 (0 は最適化の制限がない) に変更されました。

set コマンドの新しいオプション	説明
<p><code>set index_union on off</code></p>	<p><code>set index_union</code> を有効にすると、<code>or</code> 句によってテーブルのスキャンが制限されません。</p> <p>インデックス和集合(または <code>or</code> 方式)は、<code>or</code> 句を含むクエリに使用されます。次に例を示します。</p> <pre>select * from titleauthor where au_id = "409-56-7008" or title_id = "PC8888"</pre> <p><code>index_union</code> を有効にした場合、この例では <code>au_id</code> のインデックスを使用して <code>au_id = "409-56-7008"</code> を使用するすべての <code>titleauthor</code> タプルのロー ID (RID) が検索され、<code>title_id</code> のインデックスを使用して <code>title_id = "PC8888"</code> を使用するすべての <code>titleauthor</code> タプルの RID が検索されます。次に、Adaptive Server はすべての RID に対して和集合を行って重複を削除します。結果の RID は RidJoin とジョインされてデータ・タプルにアクセスします。</p> <p><code>index_union</code> を無効にすると、Adaptive Server はテーブル・スキャンを制限するためにクエリでインデックス和集合方式を使用しません。代わりに、テーブルのその他のアクセス・パスを使用し(前述の例では、テーブル <code>titleauthor</code> にテーブル・スキャンを使用する)、<code>or</code> 句を <code>scan</code> 演算子のフィルタとして適用します。</p>

『リファレンス・マニュアル：コマンド』を参照してください。

モニタリング・テーブル

Adaptive Server 15.0.1 では、*monProcedureCacheMemoryUsage* および *monProcedureCacheModuleUsage*. という 2 つの新しいモニタリング・テーブルが導入されています。

モニタリング・テーブル	説明
<i>monProcedureCacheMemoryUsage</i>	プロシージャ・キャッシュ・アロケータごとに 1 つのローがあります。アロケータはアロケータ ID で識別されます。このアロケータ ID は、Adaptive Server の内部 ID です。
<i>monProcedureCacheModuleUsage</i> .	プロシージャ・キャッシュからメモリを割り付けるモジュールごとに 1 つのローがあります。モジュールは、モジュール ID で識別され、Adaptive Server プロシージャ・キャッシュ管理の内部の機能領域分類です。

『パフォーマンス&チューニング・ガイド：モニタリングと分析』および『リファレンス・マニュアル：テーブル』を参照してください。

Adaptive Server 15.0 の新機能

Adaptive Server 15.0 では、データ・パーティションのサポート、ロー・ロックのシステム・カタログ、クエリ・プロセッサの機能強化、長い識別子、計算カラム、スクロール可能カーソル、新しいデータ型のサポート、XML の機能強化、対話型 SQL のサポート、SySAM と Adaptive Server プラグインの機能強化、ユーザ定義 Web サービスのサポート、セキュリティの機能強化が導入されています。

パーティションのサポート

パーティション分割が便利なのは、大きいテーブルやインデックスを管理するときです。テーブルやインデックスを小さく分割することによって、管理しやすくなります。パーティションを使用すると、大規模なインデックスを使用する場合と同様に、データへのアクセスが高速化され、かつアクセスしやすくなります。

パーティションはデータベース・オブジェクトであり、独立して管理できます。パーティション・レベルでは、データのロードなどを実行できますが、**create index** は不可能です。ただし、パーティションの存在をエンド・ユーザが意識することはなく、テーブルが分割されている場合もされていない場合も同じコマンドを使用してデータの選択、挿入、削除を行うことができます。

Adaptive Server 15.0 では水平分割がサポートされています。これは、テーブル・ロー全体のうちの一部を複数のディスク・デバイスに分散できる機能です。テーブルまたはインデックスのローはそれぞれ、セマンティックまたはラウンドロビンのパーティション方式に従って 1 つのパーティションに割り当てられます。

セマンティック方式では、各ローの指定されたキー・カラムのデータ値を使用して、そのローのパーティション割り当てが決定されます。ラウンドロビン方式では、データ値を参照せずにランダムにローが割り当てられます。

次のパーティション分割方式があります。

- ハッシュ分割(セマンティック) – システムによって生成されたハッシュ関数によって、各ローのパーティション割り当てを決定します。
- リスト分割(セマンティック) – キー・カラムの値が、パーティションごとに固有のユーザ指定の値のセットと比較されます。厳密な一致でパーティションの割り当てが決まります。
- 範囲分割(セマンティック) – キー・カラムの値が、各パーティションに関連付けられたユーザ指定の上限および下限のセットと比較されます。キー・カラムの値がどの範囲に属するかによって、パーティションの割り当てが決まります。

- ラウンドロビン分割 — 各パーティションにほぼ同じ数のローが含まれるように、ラウンドロビン方式でランダムにローがパーティションに割り当てられます。これがデフォルトの方式です。

次の処理を実行できます。

- **create table** および **create index** コマンドを使用してテーブルまたはインデックスを作成するときに、パーティションを作成する。
- **alter table** コマンドを使用して、テーブルのパーティション分割方式を変更する。
- **add partition** を使用して、既存のテーブルにパーティションを追加する。
- テーブルを最終的には非分割テーブルとして使用する場合であっても、分割を利用することで、大量のテーブル・データのロードを短時間で実行できます。

ロー・ロックのシステム・カタログ

多くのシステム・カタログでデータロー・ロック・スキームを使用できるようになりました。

Adaptive Server バージョン 15.0 では、ほとんどのシステム・カタログがデータロー・ロック・スキームに変換されます。次のシステム・カタログは、全ページ・ロック・スキームを引き続き使用します。

- *syslocks* や *sysprocesses* などの実体化されたテーブル。これらのテーブルは実行時に生成され、そのロック・スキームは同時実行性とは関係ない。
- *sysmessages* と *sysusermessages* (読み取り専用のテーブル)。
- *sybsecurity* 内の監査テーブル。書き込みは 1 回行われ、読み込みは何回も行われる。

Adaptive Server の内部アップグレード・プロセスによって、インストール、アップグレード、またはロード・アップグレードの実行時にシステム・テーブルのロック・スキームが変換されます。

Adaptive Server リリース 15.0 の DDL は、12.5.x 以前のバージョンと同様にテーブル・レベル・ロックを使用するので、DDL を実行するときの同時実行性は向上しません。

クエリ・プロセッサ

Adaptive Server 15.0 では、クエリ・プロセッサの機能が強化されています。

Adaptive Server バージョン 15.0 のクエリ・プロセッサはセルフチューニング機能を備えているので、以前のバージョンに比べて介入の回数が少なくなります。このバージョンの Adaptive Server では、エンジンがステップ間のデータ・フローを

サポートするので、ステップ間の実体化のために使用されるワークテーブルの数が減少します。ただし、ハッシュ操作とマージ操作が有効であると Adaptive Server が判断した場合は、使用されるワークテーブルが多くなります。

新機能では次のことがサポートされています。

- クエリ処理のための垂直並列処理と水平並列処理の両方
- インデックス選択機能の向上 (特に **OR** 句によるジョインや、データ型が一致していないが互換性のあるジョインと探索指数 (SARG) の場合)
- より効率的なアルゴリズム
- ジョイン・カラムのデータに偏りがあるジョインにジョイン・ヒストグラムを使用することによる、コスト計算の改善
- クエリ・プラン選択機能の向上によるパフォーマンス向上
 - 複数のインデックスに対する **and/or** 述部を持つクエリのための新しいインデックス和集合方式およびインデックス交差方式
 - **group by** 句と **order by** 句を持つクエリに対する、メモリ内ソートとハッシュを使用した実行中のグループ化と順序付け
 - 大規模な多方向ジョイン、およびスター/スノーフレイク・スキーマ・ジョインに対して順列検索方式を使用する、コスト・ベースの排除とタイムアウトのメカニズム
- 問題の診断および解決機能の向上
 - 検索可能な XML 形式のトレース出力
 - 新しい **set** コマンドからの診断出力
- 多数のテーブルが関係するジョイン
- データとインデックスのパーティション分割 (非常に大きなデータ・セットに対して特に効果が高い)

パーティション分割は、並列処理の基本的な構成要素です。

Adaptive Server リリース 15.0 は、ラウンドロビン方式の分割機能を備えています。ラウンドロビン分割は、12.5 スタイルの分割と同じ機能です。Adaptive Server リリース 15.0 へのアップグレード時に、既存のテーブルの分割がすべて解除され、1 方向ラウンドロビン分割テーブルに自動的に変換されます。

長い識別子

Adaptive Server 15.0 では、区切り識別子の制限が拡張されています。

オブジェクト名や識別子の長さに対する制限が変更されています。通常の識別子の場合は 255 バイト、区切り識別子の場合は 253 バイトです。この新しい制限は、テーブル名、カラム名、インデックス名などのほとんどのユーザ定義識別子に適用されます。制限が拡張されたため、一部のシステム・テーブル (カタログ) および組み込み関数も拡張されました。

変数の "@" が 1 バイトとしてカウントされるため、変数名は最大で 254 バイトです。

計算カラム

計算カラムと関数ベース・インデックスにより、データの操作が容易になり、データへのアクセスが迅速化されます。

計算カラム、計算カラム・インデックス、関数ベース・インデックスを作成して、データの操作を容易にし、データへのアクセスを迅速化できます。

- 計算カラム – 同じローの通常のカラム、関数、算術演算子、またはパス名を使用した式によって定義されます。
- 計算カラムに対するインデックス (計算カラム・インデックス) – 1 つ以上の計算カラムがインデックス・キーとして含まれるインデックスです。
- 関数ベース・インデックス – 1 つ以上の式がインデックス・キーとして含まれるインデックスです。
- 確定プロパティ – 式に対する入力の設定が同じならば、その式を何度実行しても常に同じ結果が返されることを保証するプロパティです。

計算カラムと関数ベース・インデックスは、どちらも 1 つの式または関数を基礎としてより複雑な関数を構築できる点が似ています。

計算カラムと関数ベース・インデックスの違い

計算カラムと関数ベース・インデックスは、いくつかの点で違いがあります。

- 計算カラムを使用すると、式の簡略形を作成するとともにインデックスを作成できますが、関数ベース・インデックスでは簡略形は作成されず、式に直接インデックスが作成されます。
- 計算カラムは deterministic と nondeterministic のどちらでもかまいません。一方、関数ベース・インデックスは deterministic でなければなりません。"deterministic" とは、式の入力値が同じ場合は、戻り値も同じでなければならないということです。

実体化された計算カラムと実体化されていない計算カラムの違い

計算カラムは、実体化することも、実体化しないこともできます。

- 実体化されたカラムは、ベース・カラムが挿入または更新されたときに事前評価されてテーブルに格納されます。関連する値は、データ・ローとインデックス・ローの両方に格納されます。実体化されたカラムに対するそれ以降のアクセスでは、再評価は必要なく、事前評価された結果がアクセスされます。カラムが実体化されると、そのカラムにアクセスするたびに同じ値が返されます。

- 実体化されていないカラムは、仮想カラムとも呼ばれます。仮想カラムは、アクセスされた時点で実体化されます。仮想カラムつまり非実体化カラムの場合は、カラムがアクセスされるたびに、結果の値を評価する必要があります。つまり、仮想計算カラムが nondeterministic 式に基づく式である場合、または nondeterministic 式を呼び出す場合は、アクセスするたびに異なる値が返される可能性があります。また、仮想計算カラムにアクセスすると、ドメイン・エラーのような実行時例外が発生する場合があります。

スクロール可能カーソル

Adaptive Server Enterprise 15.0 では、スクロール可能カーソルと非スクロール可能カーソルの両方がサポートされています。

「スクロール可能」とは、一度に 1 つのローではなく、任意つまり多数のローをフェッチすることでカーソル結果セット内をスクロールできることを意味します。また、結果セットを繰り返しスキャンすることもできます。スクロール可能カーソルを宣言するには Transact-SQL または JDBC を使用する必要があります、Adaptive Server 15.0 以降のクエリ・エンジンが必要です。スクロール可能カーソルを使用すると、カーソルがオープンしている状態のときには、**fetch** 文で **first**、**last**、**absolute**、**next**、**prior**、または **relative** オプションを指定することで、カーソル結果セットの任意の場所にカーソルの位置を設定できます。

結果セットの最後のローをフェッチするには、次のように入力します。

```
fetch last [from] <cursor_name>
```

結果セットの特定のロー (この場合は 500 番目のロー) を選択するには、次のように入力します。

```
fetch absolute 500 [from] <cursor_name>
```

「非反映型」と「半反映型」は、カーソルの外部からのデータ変更がどの程度までカーソルに反映されるかを表します。半反映型であるけれども非スクロール可能のカーソルを作成することもできます。

すべてのスクロール可能カーソルは読み込み専用です。すべての **update** カーソルは非スクロール可能です。

unitext データ型のサポート

Adaptive Server 15.0 では、可変長の *unitext* データ型がサポートされています。

可変長の *unitext* データ型は、Unicode 文字で最大 1,073,741,823 文字 (2,147,483,646 バイト) まで保持できます。*unitext* は、*text* データ型を使用する任意の場所で同じ

セマンティックで使用できます。*unitext* カラムは、Adaptive Server で使用されているデフォルトの文字セットにかかわらず、UTF-16 エンコーディングで保管されま

す。
unitext の利点は次のとおりです。

- 大きい Unicode 文字データ *unichar* データ型および *univarchar* データ型と併せて、Adaptive Server では Unicode データ型が完全にサポートされるため、多言語アプリケーションをインクリメンタル開発する場合に最適である。
- *unitext* は UTF-16 でデータを格納。これは、Windows 環境と Java 環境で使用するネイティブなコード化データ型である。
『システム管理ガイド』を参照してください。

big int データ型のサポート

Adaptive Server バージョン 15.0 では、真数値データ型 *bigint* がサポートされています。

次に、*bigint* データ型で使用できる値の範囲を示します。

データ型	符号付きデータ型の範囲
<i>bigint</i>	$-2^{63} \sim 2^{63} - 1$ (-9,223,372,036,854,775,808 ~ +9,223,372,036,854,775,807) の間の整数

Adaptive Server での *bigint* のサポートとともに、**hextobigint**、**biginttohex**、**count_big** の各関数も追加されています。

『リファレンス・マニュアル：ビルディング・ブロック』を参照してください。

符号なし整数値データ型のサポート

Adaptive Server 15.0 では、符号なし整数値データ型がサポートされています。

これらの符号なし整数値データ型を使用すると、必要な記憶域のサイズを増やすことなく、既存の整数値データ型の正の値の範囲を拡大できます。つまり、これらのデータ型の符号付きバージョンの範囲は、正の方向と負の方向の両方に広がります (たとえば、-32 ~ +32)。一方、符号なしバージョンの範囲は、正の方向のみに広がります。次に、符号付きデータ型と符号なしデータ型の範囲を示します。

データ型	符号付きデータ型の範囲	符号なしデータ型の範囲
<i>bigint</i>	$-2^{63} \sim 2^{63} - 1$ (-9,223,372,036,854,775,808 ~ +9,223,372,036,854,775,807) の間の整数値	0 ~ 18,446,744,073,709,551,615 の間の整数値
<i>int</i>	$-2^{31} \sim 2^{31} - 1$ (-2,147,483,648 ~ 2,147,483,647) の間の整数値	0 ~ 4,294,967,295 の間の整数値
<i>smallint</i>	$-2^{15} \sim 2^{15} - 1$ (-32,768 ~ 32,767) の間の整数値	0 ~ 65535 の間の整数値

整数 identity

Adaptive Server 15.0 では、特定のデータ型を identity 値として使用できます。

次のデータ型を identity 値として使用します。

- *bigint*
- *int*
- *numeric*
- *smallint*
- *tinyint*
- *unsigned bigint*
- *unsigned int*
- *unsigned smallint*

XML サービスの機能強化

バージョン 15.0 での XML 関連の機能強化には、XML スキーマのサポート、**for xml** 句の拡張、Unicode (I18N) のサポートがあります。

XML スキーマのサポート

DTD または XML スキーマに対して XML ドキュメントを検証できます。DTD またはスキーマは、**xmlvalidate** コマンドで指定するか、ドキュメントの中で指定します。

XML スキーマ宣言を含む XML ドキュメントの解析、格納、問い合わせを行うことができます。

for xml の拡張

Transact-SQL では、式サブクエリはカッコで囲まれたサブクエリです。式サブクエリは単一のカラム (値は式サブクエリの結果) を持ち、単一のローを返す必要があります。式サブクエリは、式を使用できるほとんどすべての場所で使用できません。

サブクエリの詳細については、『Transact-SQL ユーザーズ・ガイド』を参照してください。 **for xml** サブクエリ機能では、**for xml** 句を式サブクエリとして含む任意のサブクエリを使用できます。**for xml** サブクエリの構文については、『Adaptive Server Enterprise における XML サービス』を参照してください。

Unicode の国際化 (I18N) のサポート

I18N 拡張機能は次の 3 つに分類されます。

- **for xml** 句での I18N のサポート XML にマップする結果セットのカラムに、非 ASCII データを格納できる。生成される SQLX XML ドキュメントでは、このようなデータをプレーンな文字として、または数値文字表現 (NCR: Numeric Character Representation) として表すことができる。
- **xmlparse** および **xmlvalidate** における I18N。非 ASCII データが含まれるドキュメントを格納および検証する。
- **xmlextract** および **xmltest** における I18N。非 ASCII データが含まれる XML ドキュメントおよびクエリを処理する。

Adaptive Server プラグインの機能強化

Adaptive Server 15.0 では、Adaptive Server プラグインの機能が強化されています。

Adaptive Server プラグインの機能が強化され、効率と使いやすさが向上しています。

- Server Discovery (システムで使用可能なサーバを検索できます) と自動サーバ・ステータスを含むエンタープライズ・ビュー
- サーバの更新、リモート・サーバの管理、サーバ・ログの管理
- SQL プレビューと Job Scheduler の統合
- グラフィカルなクエリ・プラン・ビューア
- 外部ツールの統合

Interactive SQL

Interactive SQL を使用して、SQL 文の実行、スクリプトの作成、サーバへのデータベース・データの表示ができます。

Interactive SQL は、単独で実行することも、Adaptive Server プラグインから実行することもできます。Adaptive Server プラグインには、標準クエリ・ツールとして統合されています。Interactive SQL を使用すると、次の処理を実行できます。

- データベース内の情報を一覧表示する。
- アプリケーションで使用する予定の SQL 文をテストする。
- データをデータベースにロードし、管理作業を実行する。

さらに、Interactive SQL では、コマンド・ファイルやスクリプト・ファイルを実行できます。たとえば、データベースに対して実行する繰り返し可能なスクリプトを作成し、Interactive SQL を使用してそれらのスクリプトをバッチで実行できます。

ユーザ定義 Web サービス

Web Services を使用すると、Web サービスを作成し、Adaptive Server で SQL を実行できます。

Adaptive Server Web Services Engine で提供されている Web メソッドに加えて、Web Services を利用すると、Web サービスを作成でき、Web ブラウザまたは SOAP クライアントを使用して、Adaptive Server Enterprise で SQL コマンドを実行できます。このようなユーザ定義 Web サービスは、Adaptive Server Enterprise に固有の既存のセキュリティおよび監査制御機能を使用します。

ユーザ定義 Web サービスは、**create service** コマンドを使用して作成できます。実行する SQL を指定したり、**grant** コマンドでパーミッションを制御できる優れたオブジェクトを作成したり、Web ブラウザまたは SOAP クライアントでサービスを呼び出せるかどうかを制御したりできます。ASE Web Services Engine によって、ユーザ定義 Web サービス用の WSDL が自動的に生成されます。

Adaptive Server Enterprise の『Web Services ユーザーズ・ガイド』を参照してください。

非常に大きな記憶域のサポート

Adaptive Server 15.0 では、使用できるディスク・デバイスの数および各デバイスで使用できる 2K ブロックの数が拡張されています。

バージョン 15.0 より前の Adaptive Server では、内部的に 1 つの仮想ページが 32 ビットの整数で記述されています。最初のバイトはデバイス番号 (**vdevno**) を保持し、それ以降の 3 バイトでは、デバイス内でのページ・オフセットが 2K バイトの単位で記述されています (仮想ページ番号)。このようなアーキテクチャのため、デバイスの数は 256 まで、各デバイスのサイズは 32 ギガバイトまでに制限され、サーバ全体での記憶域は最大 8 テラバイトまでとなっています。

Adaptive Server 15.0 では、デバイス番号とページ・オフセットがそれぞれ別の 32 ビット整数に格納されます。この新しいアーキテクチャにより、作成可能なディスク・デバイス数は最大 2,147,483,647 となり、各デバイスの最大サイズは 2K ブロック 2,147,483,648 個分、つまり 4 テラバイトとなります。

注意： システム・テーブル *sysdevices* と *sysusages* のスキーマが変更されたため、これらのテーブルにアクセスするスクリプトとストアド・プロシージャの変更が必要な場合があります。デバイス識別子は、*sysdevices* と *sysusages* の *vdevno* カラムから取得する必要があります。これらのテーブルの *high*、*low*、*vstart* の各カラムには、デバイスと仮想ページ番号の両方が格納されるのではなく、仮想ページ番号だけが格納されるようになりました。

update statistics の自動実行

update statistics コマンドは、サイトで都合のよい時間帯に自動的に実行します。

特定の時刻に手動で **update statistics** を実行する代わりに、サイトで都合のよい時間帯に **update statistics** が自動実行されるように設定して、このコマンドの実行がシステムに支障を来すことのないようにすることができます。**update statistics** を実行するのに最適な時間帯は、**datachange** 関数からのフィードバックに基づいて決定します。**datachange** は、**update statistics** の不要な実行を防止するのにも役立ちます。テンプレートを使用して、オブジェクト、スケジュール、優先度、**update statistics** をトリガする **datachange** スレッシュホールドを指定することにより、クエリ・プロセッサによってより効率的なプランが生成されたときのみ重要なリソースが使用されるようにすることができます。

update statistics の実行には多量のリソースが必要なので、実行するかどうかは特定の条件に基づいて決定します。**update statistics** を実行する適切なタイミングを決定するのに役立つ主なパラメータとしては、次のものがあります。

- 前回 **update statistics** を実行してからデータ特性がどの程度変化したか。これは、"datachange" パラメータでわかる。
- **update statistics** を実行するのに十分なリソースが使用可能であるか。これには、アイドル CPU サイクルの数などのリソースが含まれる。また、**update statistics** の実行中に重要なオンライン・アクティビティが発生しないことも確認する必要がある。

datachange は、前回 **update statistics** を実行してから変更されたデータの量を測定するのに役立つ重要な測定基準であり、**datachange** 関数によって観測できます。この測定基準とリソース使用可能性の条件を使用することで、**update statistics** を実行する処理を自動化できます。Job Scheduler は、**update statistics** を自動的に実行するためのメカニズムを備えています。Job Scheduler には、**update statistics** をいつ実行するかを決定する、カスタマイズ可能なテンプレートのセットが含まれています。これらの入力には、**update statistics** へのすべてのパラメータ、**datachange** スレッシュホールド値、**update statistics** を実行する時刻などがあります。Job Scheduler は **update statistics** を低い優先度で実行するので、同時に実行される重要なジョブに影響を与えることはありません。

SySAM ライセンス管理

Sybase Software Asset Management (SySAM) の実装が変更されています。

変更点は次のとおりです。

- SySAM バージョン 2.0 では、資産管理ツールおよび資産レポート・ツールが追加されている。これらのツールを使用すると、ライセンスの使用状況と順守状況を監視できます。
- 単一のインストール方法で、Adaptive Server のすべてのエディションがサポートされます。
- SySAM の設定はオプションではなくなりました。
- 柔軟な SySAM 設定オプションが用意されています。
- SySAM ライセンスは、発注された製品と一緒に発送されなくなりました。ライセンス証明書を Sybase 製品ダウンロード センタ (SPDC) から入手する必要があります。
- SySAM ライセンス・キーには、購入されたサポート・プランに関する情報が格納されています。サポート・プランを更新するたびに、これらのライセンスを更新する必要があります。

- ライセンス・ポリシーは厳密かつ一貫して適用されます。
- Adaptive Server は、ライセンスを取得できない場合も、猶予期間が過ぎるまでは機能します。猶予期間は、ライセンスを取得できない原因となっている問題に対処できるように設けられています。猶予期間中、Adaptive Server は通常どおり機能し続けます。ライセンスの問題が解決されない場合は、猶予期間が終了すると Adaptive Server の機能またはサーバ自体が停止します。
- ライセンスに関するイベントの通知をリアルタイムに電子メールで受け取ることができます。
- SPDC から発行されるライセンスには、そのライセンスの配備先となるホスト・マシンに関する情報が含まれています。SPDC から再発行を受けない限り、ライセンスを別のマシンで使用することはできません。

以上の変更により、Adaptive Server のインストールと設定の処理が影響を受けます。SySAM の設定と配備のオプションの詳細については、『Adaptive Server Enterprise 設定ガイド』の、SySAM の設定に関する章を参照してください。インストール前の計画および SySAM のインストールの情報については、使用しているプラットフォームに対応した『ASE インストール・ガイド』を参照してください。

SySAM の配備計画は、Adaptive Server をインストールする前に立ててください。

警告！ SySAM では、ライセンスの問題が発生した場合に備えて猶予期間が設けられています。猶予期間に入ると、Adaptive Server のエラー・ログがこの情報によって更新されます。このようなイベントが電子メールで通知されるように設定することもできます。管理者は、Adaptive Server が猶予状態に入る原因となった問題を解決する必要があります。猶予期間中でも Adaptive Server は通常どおり機能しますが、ライセンス障害の原因となった問題が猶予期間内に解決されない場合は、Adaptive Server が停止する、またはライセンス対象の機能が使用不可能になる可能性があります。

クエリ処理測定基準 (QP 測定基準)

クエリ処理 (QP) 測定基準は、クエリの実行において経験的な測定基準値を識別して比較します。クエリが実行されると、この測定基準は、QP 測定基準での比較の基礎となる定義済みの測定基準のセットと関連付けられます。

次のような測定基準が取得されます。

- CPU 実行時間 — クエリを実行するのにかかった時間 (ミリ秒単位)
- 経過時間 — コマンドが開始した時刻と現在の時刻との差 (ミリ秒) (時刻は、オペレーティング・システムのクロックから取得します)
- 論理 I/O (LIO) 読み込み — 論理 I/O 読み込みの回数
- 物理 I/O (PIO) 読み込み — 物理 I/O 読み込みの回数

- カウント — クエリが実行された回数
- アボート・カウント — リソース制限を超えたために Resource Governor によってクエリがアボートされた回数

各測定基準には、最小、最大、平均の3つの値があります。ただし、カウントとアボート・カウントは除きます。

抽象プランの更新

Adaptive Server 15.0 では、クエリ・プロセッサの機能が強化されています。

抽象プラン関連の新機能と変更点については、『クエリ処理と抽象プラン』を参照してください。

showplan の変更

Adaptive Server バージョン 15.0 の **showplan** では、クエリ・プロセッサによって実行されるステップの表現が改善されています。

クエリ・プランの形状をより適切に表すように **showplan** メッセージのフォーマットが変更されています。以前のバージョンでは、**showplan** のメッセージが次のように縦長のフォーマットで表示されていました。

```
delete
from authors
where au_lname = "Willis"
and au_fname = "Max"

QUERY PLAN FOR STATEMENT 1 (at line 1).

The type of query is DELETE.
  The update mode is direct.

  FROM TABLE
    authors
  Nested iteration.
  Using Clustered Index.
  Index : au_names_ix
  Forward scan.
  Positioning by key.
  Keys are:
    au_lname ASC
    au_fname ASC
  Using I/O Size 2 Kbytes for index leaf pages.
  With LRU Buffer Replacement Strategy for index leaf pages.
  Using I/O Size 2 Kbytes for data pages.
  With LRU Buffer Replacement Strategy for data pages.
  TO TABLE
    authors
```

Adaptive Server バージョン 15.0 の **showplan** では、演算子によって実行された各ステップを区別できるように、連続する「パイプ」("|" 記号)が表示されます。

次に示すクエリには 3 つの演算子 EMIT、DELETE、SCAN があるので、この構成を表示するためにクエリには 3 セットのパイプが含まれます。

```
delete
from authors
where au_lname = "Willis"
and au_fname = "Max"

QUERY PLAN FOR STATEMENT 1 (at line 1).
2 operator(s) under root

The type of query is DELETE.
ROOT:EMIT Operator
|DELETE Operator
|  The update mode is direct.
|
| |SCAN Operator
| |  FROM TABLE
| |  authors
| |  Index : aumind
| |  Forward Scan.
| |  Positioning by key.
| |  Keys are:
| |    au_lname ASC
| |    au_fname ASC
| |  Using I/O Size 8 Kbytes for index leaf pages.
| |  Using LRU Buffer Replacement Strategy for index leaf pages
| |  Using I/O Size 8 Kbytes for data pages.
| |  With LRU Buffer Replacement Strategy for data pages.
|
| TO TABLE
|  authors
|  Using I/O Size 8 Kbytes for data pages.
```

注意：このバージョンの Adaptive Server には、**showplan** のメッセージを XML 形式で表示する機能もあります。

SSL での FIPS 140-2 の使用

Adaptive Server 15.0 では、SSL で FIPS 140-2 レベル 1 検証済みの暗号化モジュールが使用されます。

SSL (Secure Socket Layer) は、クレジットカード番号、株式売買、銀行取引などの機密情報を、インターネット上で安全に転送するための規格です。パブリック・キー/シークレット・キー暗号法に依存します。

Adaptive Server リリース 15.0 で使用される SSL は、FIPS 140-2 レベル 1 検証済みの暗号化モジュールを使用します。この暗号化モジュールは、Windows、Solaris、AIX、HPUX の各オペレーティング・システム上で稼動する Adaptive Server 製品向けの Certicom Security Builder GSE です。詳細については、NIST Web サイト (<http://csrc.nist.gov/cryptval/140-1/1401val.htm>) で 2005 年 6 月 2 日付けの検証証明書 #542 を参照してください。

Adaptive Server 15.0 でのシステムの変更点

Adaptive Server 15.0 では、ユーティリティ、コマンド、システム・テーブル、関数、ストアド・プロシージャ、予約語、モニタリング・テーブルが変更されています。

ユーティリティ

Adaptive Server 15.0 では、**bcp**、**dataserver**、**sqlsrvr**、**ddlgen**、**preupgrade** ユーティリティが変更されています。

表 32 : ユーティリティ・プログラムの変更

ユーティリティ	変更点
bcp	<p>計算カラムと関数インデックスをサポートするための新しいパラメータ --sho-fi と --hide-vcc が追加されています。</p> <p>分割されたテーブルへの並列ロードをサポートするための新しいパラメータ --maxconn が追加されています。</p> <p>特定のパーティションとの間で bcp in および bcp out を実行できるように bcp のインタフェースが変更されています。</p>
dataserver	-b master_database_size パラメータをテラバイト単位で指定できます。
sqlsrvr	-b master_database_size パラメータをテラバイト単位で指定できます。
ddlgen	ユーザ定義 Web サービスをサポートするために、 -T object_type パラメータで指定可能なオブジェクト・タイプに WS が追加されています。
preupgrade	さまざまなアップグレード・チェックに対して増分チェックを実行するためのオプションが追加されています。また、 load database を使用してアップグレード中の単一のデータベースに対して preupgrade を実行できるように機能が拡張されています。

『ユーティリティ・ガイド』を参照してください。

予約語

Adaptive Server 15.0 では、スクロール可能カーソルと XML サービスをサポートする新しい予約語が追加されています。

- **insensitive** – スクロール可能カーソルをサポート
- **xmlextract** – XML サービスをサポート
- **xmlparse** – XML サービスをサポート
- **xmltest** – XML サービスをサポート

以前のリリースのサーバをアップグレードする前に、新しい予約語と一致するすべてのデータベース名を変更する必要があります。テーブル、ビュー、カラム名を変更することや、区切り識別子を使用できます。バージョン 15.0 にアップグレードした後は、プロシージャ、SQL スクリプト、アプリケーションを変更しないかぎり、新しい予約語と同じ名前のデータベース・オブジェクトは使用できません。

『リファレンス・マニュアル：ビルディング・ブロック』を参照してください。

グローバル変数

Adaptive Server 15.0 では、新しいグローバル変数が導入されています。

表 33 : 新しいグローバル変数

変数	表示対象	値
@@rowcount	<p>スクロール可能カーソルまたは非スクロール可能カーソルが移動したローの数を表示するように拡張されました。</p> <p>非スクロール可能カーソルでは、基本となるテーブルからクライアントへフェッチされたローの数を表します。</p> <p>スクロール可能カーソルでは、基本となるテーブルからではなく、現在の結果セットからフェッチされたローの数を表します。</p>	<p>グローバル変数 @@rowcount の値は、指定されたカーソル・タイプの影響を受けます。</p> <p>デフォルトの非スクロール可能カーソルは、1 ローずつ前方向に移動します。最大値は、結果セット内のローの数です。</p> <p>スクロール可能カーソルの場合は、fetch コマンドの方向に関係なく、値は増加し続けます。最大値はありません。</p>

変数	表示対象	値
<code>@@fetch_status</code>	スクロール可能カーソルに使用される <code>fetch</code> コマンドのステータスを表示します。	<p>0 – <code>fetch</code> 文の実行が成功しました。</p> <p>-1 – <code>fetch</code> 文の実行に失敗しました。または、要求されたローが結果セットの範囲外にあります。</p> <p>-2 – 値は予約済みです。</p>
<code>@@cursor_rows</code>	カーソル結果セット内のローの総数を表示します。	<p>0 – オープンしているカーソルが1つもありません。または、最後にオープンされたカーソルの条件を満たすローがありません。</p> <p>-1 – カーソルは半反映型でスクロール可能であるが、スクロールのワークテーブルにまだ値が格納されていません。カーソルの条件を満たすローの数は不明です。</p> <p><i>n</i> – 最後にオープンされた、またはフェッチされたカーソル結果セットの値がすべて格納されています。返される値は、結果セット内のローの総数です。</p>

設定パラメータ

Adaptive Server 15.0 では、設定パラメータが追加および変更されています。

表 34 : 新しい設定パラメータ

関数	説明
<code>enable metrics capture</code>	Adaptive Server がサーバ・レベルで測定基準を取得できるようにします。
<code>enable semantic partitioning</code>	ライセンスのあるサイトでのテーブルとインデックスのセマンティック (ハッシュ、リスト、範囲) 分割を可能にします。
<code>enable web services</code>	Web サービスを有効にします。
<code>enable xml</code>	XML サービスを有効にします。

関数	説明
max native threads per engine	サーバが1つのエンジンについて生成するネイティブ・スレッドの最大数を定義します。
max partition degree	Adaptive Server が必要とするパーティションの数を動的に設定します。これにより、Adaptive Server で水平並列処理を実行できます。
max resource granularity	システムのリソースの何パーセントまでをクエリで使用できるかを設定します。
number of devices	Adaptive Server が使用できるデータベース・デバイスの数を指定します。
number of dump threads	Adaptive Server がメモリ・ダンプを実行するために生成するスレッド数を制御します。
number of open partitions	Adaptive Server が一度にアクセスできるパーティションの数を指定します。
optimization goal	3つの最適化目標を、サーバ・レベル、セッション・レベル、クエリ・レベルの3つの層で指定できます。
optimization timeout limit	Adaptive Server がクエリを最適化するために費やすことができる時間を、クエリの処理に費やす時間全体に対する割合として指定します。
rtm thread idle wait period	Adaptive Server が使用するネイティブ・スレッドが無処理のときの待ち時間を定義します。
sysstatistics flush interval	<i>sysstatistics</i> をフラッシュする間隔を分単位で指定します。
statement cache size	プロシージャ・キャッシュ・メモリのサーバ割り付けを増やし、プロシージャ・キャッシュ・プールのうち文のキャッシュに使用されるメモリ量を制限します。ステートメント・キャッシュ機能はサーバ・ワイドに有効化されます。

表 35 : 変更された設定パラメータ

パラメータ	変更点
default network packet size	以前のバージョンの Adaptive Server では、 default network packet size として 512 が使用されていました。Adaptive Server バージョン 15.0 では、 default network packet size は 2048 です。

『システム管理ガイド 第1巻』を参照してください。

関数

Adaptive Server 15.0 では、関数が追加および変更されています。

表 36 : 新しい関数

関数	説明
biginttohex	指定した整数値に相当する、プラットフォームの影響を受けない 16 進文字列を返します。
count_big	null でない値の数 (重複する値を除く) または選択されたローの数を <i>bigint</i> の値で返します。
datachange	update statistics 実行後のデータ分布の変化量を測定します。
data_pages	指定されたテーブル、インデックス、または特定のパーティションによって使用されているページの数を返します。
hextobigint	16 進文字列に相当する <i>bigint</i> 値を返します。
is_quiesced	データベースがクワイース状態の場合は 1、クワイース状態でない場合は 0 を返します。
partition_id	指定されたデータまたはインデックス・パーティション名のパーティション ID を返します。
partition_name	新しいパーティションの明示的な名前を返します。
reserved_pages	テーブル、インデックス、または特定のパーティションのために予約されているページ数をレポートします。
row_count	指定したテーブル内のローの予測数を返します。
showplan_in_xml	XML 形式の実行プランを返します。
ssel_message	メッセージ ID を指定したときにメッセージ・テキストを返します。
tran_dumpable_status	dump transaction を使用できるかどうかを示す true/false の値を返します。
used_pages	テーブル、インデックス、または特定のパーティションで使用されているページ数をレポートします。

関数	説明
<code>xmlvalidate</code>	XML ドキュメントを検証します。非 ASCII 文字を使用しているドキュメントも対象です (I18N)。『Adaptive Server Enterprise における XML サービス』を参照してください。

いくつかの関数名が、よりわかりやすい名前に変更されています。

表 37 : 古い関数名と新しい関数名

古い関数名	新しい関数名
<code>data_pgs</code>	<code>data_pages</code>
<code>used_pgs</code>	<code>used_pages</code>
<code>reserved_pgs</code>	<code>reserved_pages</code>
<code>rowcnt</code>	<code>row_count</code>
<code>ptn_data_pgs</code>	<code>data_pages</code>

コマンド

Adaptive Server 15.0 では、コマンドが追加および変更されています。

表 38 : 新しいコマンド

コマンド	機能
<code>create service</code>	ユーザ定義 Web サービスの作成用です。
<code>drop service</code>	ユーザ定義 Web サービスの削除用です。
<code>update table statistics</code>	テーブルまたはパーティションに対する <i>systabstats</i> 統計値を更新します。

表 39 : `set` コマンドの新しいオプション

set のオプション	説明
<code>set delayed_commit</code>	ログ・レコードをいつディスクに書き込むかを指定できます。 <code>delayed_commit</code> パラメータを <code>true</code> に設定すると、ログ・レコードはディスクに非同期に書き込まれ、I/O の完了を待たずに、制御はクライアントに戻ります。

set のオプション	説明
set plan optgoal	最適化目標をセッション・レベルで設定します。
set plan opttimeoutlimit	実行時間の長い複雑なクエリに要する時間の制限をセッション・レベルで設定します。
set metrics_capture on/off	QP 測定基準をセッション・レベルでアクティブにします。

表 40 : 変更されたコマンド

コマンド	変更点
alter table	実体化または非実体化計算カラムをサポートするための構文が追加されました。パーティションに対するサポートが追加されました。
create index	通常のカラムと同じように計算カラムをインデックス・キーとして使用したり、関数ベース・インデックスを作成したりできるように拡張されました。パーティションに対するサポートが追加されました。
create table	実体化または非実体化計算カラムをサポートするための構文が追加されました。パーティションに対するサポートが追加されました。
dbcc	パーティションに対するサポートが追加されました。
declare cursor	スクロール可能カーソルの構文が追加されました。 semi_sensitive 、 insensitive 、 scrollable カーソルをサポートするための構文が追加されました。
delete statistics	パーティションに対するサポートが追加されました。
disk init	size パラメータをテラバイト単位で指定できます。 directio パラメータが追加されました。このパラメータでは、オペレーティング・システムのバッファ・キャッシュを回避して、データをディスクに直接転送できます。
disk reinit	directio パラメータが追加されました。このパラメータでは、オペレーティング・システムのバッファ・キャッシュを回避して、データをディスクに直接転送できます。

コマンド	変更点
fetch	スクロール可能カーソルをサポートするための fetch_orientation のオプション (next 、 prior 、 first 、 last 、 absolute 、 relative) が追加されました。
reorg	パーティションに対するサポートが追加されました。
select	XML サービスをサポートするための for xml 句が追加されました。パーティションに対するサポートが追加されました。
truncate table	パーティションに対するサポートが追加されました。
update all statistics	パーティションに対するサポートが追加されました。
update statistics	パーティションに対するサポートが追加されました。
update partition statistics	旧式になりました。

システム・ストアド・プロシージャ

Adaptive Server 15.0 では、システム・ストアド・プロシージャが追加および変更されています。

表 41：新しいストアド・プロシージャ

ストアド・プロシージャ	機能
sp_helpcomputedcolumn	指定されたテーブルのすべての計算カラムに関する情報をレポートします。
sp_version	最後に実行されたインストール・スクリプト (installmaster 、 installdbccdb など) のバージョン情報と、そのスクリプトが正常に実行されたかどうかを返します。

表 42：変更されたシステム・ストアド・プロシージャ

ストアド・プロシージャ	変更点
sp_checksourc	計算カラムのソース・テキストが存在することを確認します。
sp_help	計算カラム、関数ベース・インデックス、パーティションに関する情報をレポートします。

ストアド・プロシージャ	変更点
sp_helppartition	パーティションの詳細な情報を出力に追加します。
sp_helpindex	計算カラム・インデックス、関数ベース・インデックス、パーティションに関する情報をレポートします。
sp_helptext	計算カラム、関数ベース・インデックスの定義、パーティションのソース・テキストを表示します。
sp_hidetext	計算カラム、関数ベース・インデックスのキー、パーティション条件のテキストを隠します。
sp_modifylogin	オプション " enable logins during recovery " が追加されています。
sp_webservices	ユーザ定義 Web サービスをサポートするための addalias 、 deploy 、 dropalias 、 listudws 、 listalias 、 undeploy オプションが追加されています。
sp_monitorconfig	number of open partitions 設定パラメータがサポートされています。
sp_countmetadata	number of open partitions 設定パラメータがサポートされています。
sp_helpsegment	オブジェクトとパーティションに対するセグメントのバインドを出力します。
sp_objectsegment	特定のオブジェクトに対するすべてのパーティションのセグメント情報を表示します。
sp_placeobject	パーティションに対する以降の領域割り当てが新しいセグメントから行われるようにします。
sp_dbcc_faultreport	特定の OPID またはフォールト・タイプに対するレポートを作成します。
sp_sysmon	オープン・パーティションに関連する情報を設定ファイルの Metadata Cache Management セクションにレポートします。

『リファレンス・マニュアル：プロシージャ』を参照してください。

システム・テーブル

Adaptive Server 15.0 では、システム・テーブルが追加および変更されています。

Adaptive Server バージョン 15.0 は、複数ユーザによる同時 DDL (データ定義言語) 操作の機能強化をサポートするために必要なロー・ロック・カタログ・インフラストラクチャを備えています。ただし、このリリースでは、DDL 操作に対するカタログ・ロックの動作は変更されていません。負荷の大きい、複数ユーザによる DDL 操作 (*tempdb* でのテーブルの作成や削除など) を実行するアプリケーションについても、このリリースではカタログ・ブロックに関する動作が特に変わることはなく、DDL の同時実行性が向上することはありません。

表 43 : 新しいシステム・テーブル

テーブル	説明
<i>syspartitions</i>	<i>syspartitions</i> は、15.0 より前のバージョンのテーブルから完全に変更されています。すべてのカラムが新しくなっています。 <i>syspartitions</i> では、テーブルとインデックスのセマンティック分割とラウンドロビン分割の両方がサポートされています。
<i>syspartitionkeys</i>	ハッシュ分割、範囲分割、リスト分割の各テーブルのパーティション・キーのカラムごとに 1 つのローが格納されます。

表 44 : 変更されたシステム・テーブル

テーブル	変更点
<i>syscolumns</i>	<p>新しいフィールド：</p> <ul style="list-style-type: none"> • computedcol • status3 <p>新しいカラム：</p> <ul style="list-style-type: none"> • <i>enctype</i> – 暗号化のタイプ • <i>enclen</i> – 暗号化カラムの長さ • <i>enrkeydid</i> – 暗号化キー ID • <i>enrkeydb</i> – 暗号化キーが格納されているデータベースの名前 • <i>enrdate</i> – カラムを暗号化した日付 <p><i>status2</i> フィールドの新しいビット：</p> <ul style="list-style-type: none"> • 16 進：0x00000010、10 進：16 – カラムは計算カラムです。 • 16 進：0x00000020、10 進：32 – カラムは実体化された計算カラムです。 • 16 進：0x00000040、10 進：64 – カラムはビュー内の計算カラムです。

テーブル	変更点
<i>sysconstraints</i>	status フィールドの新しい内部ビット：16 進：0x0100、10 進：265 – 計算カラム・オブジェクトを示します。
<i>sysdevices</i>	新しいカラム： <ul style="list-style-type: none"> • <i>vdevno</i> – デバイス ID 番号 • <i>crdate</i> – デバイスが作成された日 • <i>resizedate</i> – デバイスのサイズが変更された日 • <i>status2</i> – 追加の status2 ビット
<i>sysusages</i>	新しいカラム： <i>vdevno</i> – デバイス ID 番号
<i>sysstatistics</i>	新しいカラム： <ul style="list-style-type: none"> • <i>indid</i> – データ・パーティションのインデックス ID、常に 0 • <i>partitionid</i> – データ・パーティションの ID • <i>ststatus</i> – 内部ステータス・ビット <p><i>id</i>、<i>indid</i>、<i>partitionid</i>、<i>statid</i>、<i>colidarry</i>、<i>formatid</i>、<i>sequence</i> に対するユニーク配置インデックス</p>
<i>systabstats</i>	新しいカラム： <ul style="list-style-type: none"> • <i>partitionid</i> – データまたはインデックス・パーティションの ID • <i>statmoddate</i> – ディスク上で統計値が最後に変更された日 • <i>unusedpgcnt</i> – 未使用ページの数 • <i>oampagecnt</i> – 各パーティションに対する OAM ページの数
<i>syspartitions</i>	<i>syspartitions</i> は、15.0 より前のバージョンのテーブルから完全に変更されています。すべてのカラムが新しくなっています。 <i>syspartitions</i> では、テーブルとインデックスのセマンティック分割とラウンドロビン分割の両方がサポートされています。
<i>syscomments</i>	新しいカラム： <i>partitionid</i> – データまたはインデックス・パーティションの ID 計算カラムまたは関数ベース・インデックス・キー式のテキストを格納するようにテーブルが拡張されています。

テーブル	変更点
<p><i>sysindexes</i></p>	<p>新しいカラム：</p> <ul style="list-style-type: none"> • <i>partitiontype</i> – パーティション分割方式：1 – 範囲、2 – ハッシュ、3 – ラウンドロビン、4 – リスト • <i>conditionid</i> – パーティション条件の ID <p>新しいロー：関数ベース・インデックスまたは計算カラムに作成されたインデックスごとに 1 つのローが格納されます。</p> <p><i>status2</i> フィールドに 1 つの新しい内部ステータス・ビットが追加されました：16 進：0x8000、10 進：32768 – インデックスは関数ベース・インデックスです。</p>
<p><i>syslocks</i></p>	<p>新しいカラム：</p> <ul style="list-style-type: none"> • <i>nodeid</i> – 今後のために予約済み • <i>partitionid</i> – データまたはインデックス・パーティションの ID、今後のために予約済み、常に 0
<p><i>sysobjects</i></p>	<p><i>type</i> カラムの新しいオブジェクト：<i>N</i> – パーティション条件</p> <p>新しいカラム：</p> <ul style="list-style-type: none"> • <i>identburnmax</i> – <i>identity</i> カラムの場合、消去済み <i>identity</i> の最大値 • <i>spacestates</i> – 追跡されているスペース状態の数(DOL テーブルに対してのみ適用) • <i>erlchgt</i>s – 予測ロー長が最後に変更されたときのタイムスタンプ (DOL テーブルに対してのみ適用) <p>新しいロー：計算カラムおよび関数ベース・インデックス・キーのオブジェクトごとに 1 つのロー</p> <ul style="list-style-type: none"> • <i>type</i> フィールド：オブジェクトが計算カラムの場合のタイプ "C" が追加されています。 • <i>status2</i> フィールド：テーブルに 1 つ以上の関数ベース・インデックスが含まれることを示す新しいビットが追加されています。
<p><i>sysprocedures</i></p>	<p>計算カラムまたは関数ベース・インデックス定義に対するシーケンス・ツリーがバイナリ形式で格納されます。</p>

表 45 : システム・テーブルのカラムのデータ型の変更

システム・テーブル	変更されたカラム	データ型の変更	識別子名
<i>sysattributes</i>	<i>object_cinfo</i> <i>char_info</i>	<i>varchar(30) null ~ varchar(255) null</i> <i>varchar(255) ~ varchar(768)</i>	オブジェクトの識別子
<i>sysaudits01 - sysaudits08</i>	<i>objname</i>	<i>varchar(30) not null ~ varchar(255) not null</i>	オブジェクト名
<i>syscolumns</i>	<i>name</i>	<i>varchar(30) not null ~ varchar(255) not null</i>	カラム名
	<i>remote_name</i>	<i>varchar(30) null ~ varchar(255) null</i>	リモート名にローカル名をマップする
<i>sysconfigures</i>	<i>name</i>	<i>varchar(80) null ~ varchar(255) null</i>	
<i>sysindexes</i>	<i>name</i>	<i>varchar(30) null ~ varchar(255) null</i>	テーブル名に対するインデックス
<i>sysjars</i>	<i>jname</i>	<i>varchar(30) null ~ varchar(255) null</i>	JAR 名
<i>sysobjects</i>	<i>name</i>	<i>varchar(30) not null ~ varchar(255) not null</i>	オブジェクト名
<i>sysprocesses</i>	<i>hostname</i>	<i>char(10) not null ~ varchar(30) null</i>	ホスト・コンピュータ名
	<i>program_name</i>	<i>char(16) not null ~ varchar(30) null</i>	<i>front_end</i> モジュールの名前
	<i>hostprocess</i>	<i>char(8) not null ~ varchar(30) null</i>	ホスト・プロセスの ID 番号
	<i>cmd</i>	<i>char(16) not null ~ varchar(30) null</i>	現在実行中のコマンドまたはプロセス(if または while ループなどの条件文の評価が返される)
<i>systimeranges</i>	<i>name</i>	<i>varchar(30) not null ~ varchar(255) not null</i>	時間範囲のユニークな名前
<i>systypes</i>	<i>name</i>	<i>varchar(30) ~ varchar(255)</i>	データ型名

システム・テーブル	変更されたカラム	データ型の変更	識別子名
<i>sysdatabases</i>	<i>def_remote_loc</i>	<i>varchar(255) null ~ varchar(349) null.</i>	

『リファレンス・マニュアル：テーブル』を参照してください。

モニタリング・テーブル

Adaptive Server 15.0 では、モニタリング・テーブルが追加および強化されています。

表 46：新しいモニタリング・テーブル

モニタリング・テーブル	説明
<i>monOpenPartitionActivity</i>	パーティションに対するモニタリング情報が保持されます。

表 47：変更されたモニタリング・テーブル

モニタリング・テーブル	変更点
<i>monEngine</i>	ハウスキーピング GC タスク用の新しいカラム
<i>monCachedObject</i>	パーティション用の新しいカラム
<i>monProcessObject</i>	パーティション用の新しいカラム

『リファレンス・マニュアル：テーブル』を参照してください。

Adaptive Server 12.5.4 の新機能

Adaptive Server 12.5.4 では、セキュリティの機能強化、アーカイブ・データベースへのアクセスのサポート、共有ディレクトリの変更、新しいデータ・プロバイダのサポート、動的にロードされる TIBCO ライブラリ、JRE サポート、数値データ型のモジュロ算術、Windows のターミナル・サーバ共有メモリ要件のサポート、ログイン・トリガの機能強化が導入されています。

Kerberos の機能強化

Adaptive Server 12.5.4 では、Kerberos の機能が強化されています。

- Adaptive Server 名とは異なる Kerberos プリンシパル名を指定する新しいオプション
- Kerberos に対するプラットフォーム・サポートの拡張
- **sp_modifylogin** とオプション **authenticate with** の使用 (個別のログインで Kerberos 認証を要求)
- **sybmapname** (外部のユーザ・プリンシパル名を Adaptive Server ログインのネーム・スペースに変換するカスタマイズ可能なユーティリティ)

LDAP ユーザ認証の強化

Adaptive Server 12.5.4 では、LDAP ユーザ認証機能が強化されています。

- プライマリ LDAP サーバに対する認証からセカンダリ LDAP サーバへの自動フェールオーバーを行う。
- Adaptive Server は、LDAP サーバとの通信時に発生したエラーからリカバリする。
- LDAP サーバから Adaptive Server クライアントに渡されるパスワード有効期限に関するメッセージの通信が強化されている。

複雑なパスワードの強化

Adaptive Server バージョン 12.5.4 では、新しいログインまたはリセットされたパスワードについてのルールを設定できます。

Adaptive Server 12.5.4 では、複雑なパスワードの属性を幅広く指定できます。

- パスワードのアルファベット文字の最小文字数を設定する。
- パスワードにログイン名を含めないことを指定する。
- パスワードで使用する特殊文字の最小文字数を設定する。
- パスワードの大文字の最小文字数を設定する。
- パスワードの小文字の最小文字数を設定する。
- 初回ログイン時にパスワードをリセットする必要があることを指定する。

ASE プラグインでこれらの新しいオプションをそれぞれ設定するか、新しいストア・プロシージャ **sp_passwordpolicy** を使用できます。

アーカイブ・データベースへのアクセスのサポート

アーカイブ・データベースにアクセスすると、さまざまな操作をデータベース・ダンプに対して直接実行できます。

データベース・ダンプを従来の読み取り専用データベースと同様に扱えることにより、データベース管理者はアーカイブ・データベースにアクセスして、データベース・ダンプのデータ検証やデータの選択的なりカバリを行うことができます。

共有ディレクトリの変更

Adaptive Server バージョン 12.5.4 では、共有ディレクトリ構造が変更されています。

表 48 : UNIX および Linux プラットフォームの共有ディレクトリ構造の変更

コンポーネント	以前のロケーション	新しいロケーション
共有ディレクトリ	\$SYBASE/shared-1_0	\$SYBASE/shared
Sybase Central	\$SYBASE/sybcent41	\$SYBASE/shared/sybcentral43
JRE	\$SYBASE/shared-1_0/JRE-1_3	\$SYBASE/shared/jre142
共有 jar ファイル	\$SYBASE/shared-1_0/lib	\$SYBASE/shared/lib

表 49 : Microsoft Windows プラットフォームの共有ディレクトリ構造の変更

コンポーネント	以前のロケーション	新しいロケーション
共有ディレクトリ	%SYBASE%\shared-1_0	%SYBASE%\Shared

コンポーネント	以前のロケーション	新しいロケーション
Sybase Central	%SYBASE%\sybcent41	%SYBASE%\Shared\Sybase Central 4.3
JRE	%SYBASE%\shared-1_0\JRE-1_3	%SYBASE%\Shared\Sun\jre142
共有 jar ファイル	%SYBASE%\shared-1_0\lib	%SYBASE%\Shared\lib

Sybase によるドライバのサポート

Adaptive Server 12.5.4 では、新しい Adaptive Server データ・プロバイダがサポートされています。

新しいドライバは次のとおりです。

- ASE ADO.NET Data Provider 1.1
- Sybase 12.5.1 で提供される ASE OLE DB プロバイダ
- Sybase 12.5.1 で提供される ASE ODBC ドライバ

動的にロードされる TIBCO ライブラリ

TIBCO JMS は、Adaptive Server の実行プログラムに静的にリンクされるのではなく、動的にロードされるようになりました。

Adaptive Server バージョン 12.5.4 では、TIBCO EMS および IBM MQ メッセージ・バスと対話するために必要なメッセージング・ライブラリを動的にロードします。

Adaptive Server メッセージング・ライブラリにはメッセージ・ロジックが含まれ、これは、TIBCO や IBM などのベンダが提供するメッセージング・ライブラリの上に存在するラップとして機能します。これらの Adaptive Server メッセージング・ライブラリは、RTDS 3.5 およびそれ以降のバージョンを購入すると使用できます。Adaptive Server をインストールした後に、Adaptive Server メッセージング DLL をインストールするには、RTDS 3.5 をインストールする必要があります。これらのライブラリは、\$SYBASE/ASE-12_5/lib ディレクトリにインストールします。

Adaptive Server メッセージング・ライブラリに加え、ベンダ (TIBCO および IBM) が提供するメッセージング・ライブラリおよび DLL も必要です。

JRE のサポート

Adaptive Server 12.5.4 には、JRE 1.4 が含まれます。

JRE 1.4 は、完全インストールおよび標準インストールではデフォルトでインストールされます。また、カスタム・インストールでは、JRE の実行が必要なコンポーネントが選択されるとインストールされます。

Adaptive Server プラグインの変更

Adaptive Server プラグインでは、アーカイブ・データベースへのアクセスと複雑なパスワードがサポートされています。

Sybase Central Adaptive Server プラグインでは、Adaptive Server バージョン 12.5.4 でのアーカイブ・データベースへのアクセスと複雑なパスワード・オプションの設定がサポートされています。

Sybase Central Adaptive Server プラグインを使用して、アーカイブ・データベースを管理します。データベース・フォルダの下には、アーカイブ・データベースと呼ばれる新しいフォルダがあります。

Sybase Central Adaptive Server プラグインを使用して、サーバ・プロパティ・シートから Adaptive Server に複雑なパスワードを設定します。Adaptive Server バージョン 12.5.4 には、サーバ・プロパティ・シートに [ログイン・パスワードの設定] という新しいタブがあります。

『Adaptive Server Enterprise の管理』を参照してください。

システム・カタログの更新

サーバワイドのコマンド **allow updates to system catalogs** は、**allow updates** のストアド・プロシージャ設定よりも優先されます。

モニタリング・テーブルの変更

モニタリング・テーブル *monSysStatement* および *monProcessStatement* が変更されています。

表 50 : モニタリング・テーブル内の新しいカラム

カラム	モニタリング・テーブル	説明
<i>RowsAffected</i>	<i>monSysStatement</i> および <i>monProcessStatement</i>	現在の文からの影響を受けるローの数を示します。効率の低いクエリ・プランを使用するクエリを検索するときに便利です。このようなクエリを使用すると、返される各ローに対して多数の論理 I/O が生じることがあります。

『リファレンス・マニュアル：テーブル』を参照してください。

syscomments の変更

ストアド・プロシージャまたはトリガのソース・テキストがシステム・テーブル *syscomments* に格納される場合、**select *** を使用するクエリは、**select *** で参照されるカラム・リストを拡張する *syscomments* に格納されます。

Windows Terminal Server 環境での共有メモリ・サポート

Windows Terminal Server の共有メモリ要件に対応するために、Adaptive Server 12.5.4 では新しい環境変数 SYBASE_TS_MODE が導入されています。

グローバル・ログイン・トリガ

Adaptive Server バージョン 12.5.4 では、各ユーザ・ログインで実行される新しいグローバル・ログイン・トリガを設定できます。

ログイン・トリガからの **set** オプションのエクスポート

Adaptive Server バージョン 12.5.4 では、ログイン・トリガ内の **set** オプションをユーザ・セッション全体で有効にできます。

数値データ型のモジュロ算術

Adaptive Server バージョン 12.5.4 では、実数、浮動小数点、小数、数値にも整数と同様のモジュロ算術を実行できます。

Adaptive Server 12.5.4 でのシステムの変更点

Adaptive Server 12.5.4 では、コマンド、設定パラメータ、関数が追加および変更されています。

コマンド

Adaptive Server 12.5.4 では、**shutdown**、**dump database**、**load database** コマンドが変更されています。新しいコマンドはありません。

表 51 : コマンドの変更

コマンド	変更点
shutdown	新しい with wait および with nowait パラメータがサポートされています。これらのパラメータを使用すると、サーバを即座に停止するか、指定の時間待ってから停止できます。
dump database	新しい with verify パラメータが導入されています。このパラメータを使用すると、データベースをダンプするときに検証を行うことができます。
load database	新しい with verify パラメータが導入されています。このパラメータを使用すると、データベースをロードするときに検証を行うことができます。

『リファレンス・マニュアル：コマンド』を参照してください。

設定パラメータ

Adaptive Server 12.5.4 では、設定パラメータが追加されています。

表 52 : 新しい設定パラメータの説明

設定パラメータ	説明
allow updates to system catalogs	allow updates のストアド・プロシージャ設定よりも優先されるシステムワイドのパラメータです。有効でない場合、サーバワイドのストアド・プロシージャ設定によりシステム・カタログを変更できるかどうかが決まります。

『システム管理ガイド』を参照してください。

関数

Adaptive Server 12.5.4 では、関数が追加されています。

表 53 : 新しい関数の説明

関数	説明
pssinfo	pss からの情報を返します。IPv6 アーキテクチャでは、IP アドレスの長さは 64 バイトです。sysprocesses のサイズは大きくできないため、passinfo では完全な IP アドレスが取得されます。
xa_bqual	ASCII XA トランザクション ID の bqual コンポーネントのバイナリ・バージョンを返します。
xa_gtrid	ASCII XA トランザクション ID の gtrid コンポーネントのバイナリ・バージョンを返します。

『リファレンス・マニュアル：ビルディング・ブロック』を参照してください。

Transact-SQL コマンド

Adaptive Server 12.5.4 では、Transact-SQL コマンドが変更されています。

表 54 : Transact-SQL コマンドの変更

Transact-SQL コマンド	変更点
select *	<p>12.5.4 では、識別子 (テーブル名、カラム名など) が識別子のルールに準じているかどうかを確認するよう、カラム・リストの拡張が強化されています。</p> <p><i>select *</i> を拡張したテキストで使用されるすべての識別子で、識別子が識別子のルールに従わない場合は角カッコが追加されます。</p> <p>最新リリースにアップグレードしても Adaptive Server が SQL テキストを使用できるように、識別子の前後に角カッコを追加します。</p>

『Transact-SQL ユーザーズ・ガイド』を参照してください。

追加の説明や情報の入手

Sybase Getting Started CD、SyBooks▲TM▲Online、オンライン・ヘルプを利用すると、この製品リリースについて詳しく知ることができます。

- Getting Started CD (またはダウンロード) には、PDF 形式のリリース・ノートとインストール・ガイド、SyBooks CD に含まれていない他のマニュアルや更新情報が収録されています。
- SyBooks Online (<http://sybooks.sybase.com/>) – オンライン・バージョンの SyBooks であり、標準の Web ブラウザを使用してアクセスできます。ドキュメントをオンラインで閲覧したり、PDF としてダウンロードしたりできます。また、Web サイトには、製品マニュアルのほか、EBFs/Maintenance、Technical Documents、Case Management、Solved Cases、ニュース・グループなどのリソースへのリンクもあります。
- 製品のオンライン・ヘルプ (利用可能な場合)

PDF 形式のドキュメントを表示または印刷するには、Adobe の Web サイトから無償でダウンロードできる Adobe Acrobat Reader が必要です。

注意：製品のリリース後に追加された製品およびマニュアルに関する重要な情報を含んだ最新のリリース・ノートは、SyBooks Online から入手できます。

追加の説明や情報の入手

索引

記号

@@lastlogindate グローバル変数 77

A

Adaptive Server での Java 35
 Adaptive Server での並列処理 90
 Adaptive Server プラグイン 96, 122
 Interactive SQL 96
 Adaptive Server プラグインの更新 34
 allow updates to system catalogs 設定パラメータ 125
 authmec() 関数 77

B

Backup Server 13
 backupserver ユーティリティ 26
 bcp ユーティリティ 105
 big int データ型 94
 bigdatetime データ型 17
 biginttohex 関数 109
 bigtime データ型 17
 builtin date strings 設定パラメータ 24

C

cache_usage 関数 18
 cis idle connection timeout 設定パラメータ 83
 Cluster Edition
 の機能とプラットフォームの一覧 1
 Cluster Edition、アーカイブ・データベース 3
 cost of a cpu unit 設定パラメータ 75
 cost of a logical io 設定パラメータ 75
 cost of a physical io 設定パラメータ 75
 count_big 関数 109
 create service コマンド 110
 create table コマンド 31

D

data_pages 関数 109

datachange 関数 109
 dataserver ユーティリティ 105
 db_attr 関数 18
 ddlgen ユーティリティ 66, 105
 deferred name resolution 設定パラメータ 24
 detach transaction コマンド 33
 drop service コマンド 110
 dsedit ユーティリティ 30
 DTM
 トランザクション・マネージャ 33
 外部ロールバック 33
 dump database コマンド 125

E

enable encrypted columns 設定パラメータ 75
 enable literal autoparm 設定パラメータ 83
 enable merge join 設定パラメータ 75
 enable metrics capture 設定パラメータ 107
 enable semantic partitioning 107
 enable web services 107
 enable xml 107

F

FIPS 140-2 14
 FIPS 140-2 と SSL 102

H

hashbytes() 関数 77
 hextobigint 関数 109

I

IBM Tivoli Storage Manager 13
 index_name() 関数 77
 Interactive SQL 97
 is_quiesced 関数 109
 isdate 関数 83
 isnumeric 関数 83

K

Kerberos 119
Kerberos による認証 31

L

LDAP 119
LDAP サーバ認証 30
LDAPS 29
Linux pSeries の機能 31, 32
load database コマンド 125

M

max native threads per engine 107
max online q engines 設定パラメータ 75
max partition degree 設定パラメータ 107
max resource granularity 設定パラメータ 107
maximum nesting level 設定パラメータ 39
metrics elap max 設定パラメータ 75
metrics exec max 設定パラメータ 75
metrics lio max 設定パラメータ 75
metrics pio max 設定パラメータ 75
min pages for parallel scan 設定パラメータ 75
mnc_full_index_filter 設定パラメータ 39
monOpenPartitionActivity モニタリング・テーブル 118
monProcedureCacheMemoryUsage モニタリング・テーブル 88
monProcedureCacheModuleUsage モニタリング・テーブル 88
monProcessStatement 123
monSQLRepActivity モニタリング・テーブル 39
monSQLRepMisses モニタリング・テーブル 39
monSysStatement 123
monTableTransfer モニタリング・テーブル 24

N

number of devices 設定パラメータ 107
number of dump threads 設定パラメータ 107
number of open partitions 設定パラメータ 107

O

object_attr 関数 18

openssl ユーティリティ 26
optimization goal 設定パラメータ 107
optimization timeout limit 設定パラメータ 107

P

partition_id 関数 109
partition_name 関数 109
partition_object_id 関数 83
password_random 関数 37
pci memory size 設定パラメータ 39
preupgrade ユーティリティ 105
pssinfo 関数 37, 126

R

reinit_descriptors サブコマンド 29
Replication Server 29
reserved_pages 関数 109
row_count 関数 109
rtm thread idle wait period 設定パラメータ 107

S

Secure Sockets Layer (SSL) 54
select * Transact-SQL コマンド 126
send doneinprov tokens 設定パラメータ 75
set コマンド 45, 66
set...opt criteria コマンド 81
showplan の変更 101
showplan_in_xml 関数 109
shutdown コマンド 125
sp_deletesmobj ストアド・プロシージャ 19
sp_downgrade システム・プロシージャ 67
sp_helpcomputedcolumn ストアド・プロシージャ 112
sp_jreconfig システム・プロシージャ 37
sp_listener ストアド・プロシージャ 30
sp_querysmobj ストアド・プロシージャ 19
sp_spaceusage システム・プロシージャ 67
sp_tabsuspectptn システム・プロシージャ 37
sp_version ストアド・プロシージャ 112
spoc optimize timeout limit 設定パラメータ 83
spt_TableTransfer システム・テーブル 25
SQL 文の複写 29

sqlsrvr ユーティリティ 105
 ssl_message 関数 109
 SSL (Secure Socket Layer) と FIPS 140-2 102
 SSL 証明書の共通名 30
 SSL のサポート 54
 startup delay 設定パラメータ 83
 statement cache size 設定パラメータ 107
 Sybase Central 96
 SySAM の変更 99
 syscomments、変更 123
 sysdatabases システム・テーブル 25
 sysdevices システム・テーブル 25
 sysoptions システム・テーブル 71
 syspartitionkeys システム・テーブル 113
 syspartitions システム・テーブル 113
 sysstatistics flush interval 設定パラメータ 107

T

tempdb グループ 16
 TIBCO ライブラリ 121
 Tivoli Storage Manager 13
 Tivoli Storage Manager、システム・プロセス
 ャ 19
 tran_dumpable_status 関数 109
 transfer table コマンド 21

U

Unicode の機能強化 95
 unitext のサポート 93
 update statistics コマンド 98
 update table statistics コマンド 110
 used_pages 関数 109

W

Web サービス 97

X

xa_bqual 関数 126
 xa_gtrid 関数 126
 XML

スキーマのサポート 95
 国際化のサポート 95
 XML 関連の機能強化 95
 xmltable() 関数 59
 xmlvalidate 関数 109

あ

アーカイブ・データベース 3
 アーカイブ・データベースへのアクセス 44,
 120
 圧縮レベル 13
 暗号化カラム、機能強化 43

い

インメモリ・データベース 11
 インメモリ・データベース、システム・プロ
 シージャ 19
 インメモリ・テンポラリ・データベース 11

お

オブティマイザ、遅延コンパイル 45
 オブティマイザ設定、表示 53

か

カーソル、スクロール可能 93
 仮想ハッシュ・テーブル 31
 監査、隠されたパスワード 55
 監査、失敗したログイン 55
 監査機能の変更 26
 関数ベース・インデックス 92
 計算カラムとの違い 92

き

キーワード 106
 機能とプラットフォームの一覧 1, 9
 共有ディレクトリの変更 120
 共有メモリ 32

く

クエリ、長時間実行されている 45
 クエリ処理測定基準 100
 クエリ・プラン 52, 90
 クエリ・プランのフォーマット 101
 クエリ・プロセッサ 90
 グループ、tempdb 16
 グローバル変数 106

け

計算カラム 92
 関数ベース・インデックスとの違い 92
 実体化と非実体化 92
 計算カラム、実体化 92
 計算カラム、非実体化 92

こ

高可用性 33
 高可用性とパスワード 56

し

システム・ストアド・プロシージャのインス
 トール 33
 システム・テーブル、新規 114
 システム・テーブル、変更 114
 システム・テーブル、ロー・ロック 90
 システム・テーブルの変更 113
 集約処理 48
 ジョイン、再配置 59

す

スカラ集合 48
 スクロール可能カーソル 93
 ストアド・プロシージャ、遅延コンパイル 45

せ

整数 identity、identity 値としてのデータ型 95
 セキュリティ、LDAP 119
 セキュリティ、暗号化カラム 43

セキュリティ、複雑なパスワード 119
 セキュリティの機能強化、Kerberos 119

そ

増分データ転送 15
 ソート順 46

ち

遅延コンパイル 45
 遅延コンパイル、ストアド・プロシージャ 52
 遅延名前解決 14
 抽象プラン 80
 抽象プラン、機能強化 79
 抽象プラン、クエリ・レベル設定 79
 抽象プラン、更新 101
 長時間実行されているクエリ 45

て

ディスク・デバイス、非常に大きな記憶域の
 サポート 98
 データ型
 unitext 93
 データの転送 15
 データ・プロバイダ、新規 121
 データベース・ダンプ 44

と

統計集合関数 47
 統計情報、収集 98
 動的リンク、メッセージング・ライブラリ
 121
 ドライバ、新規 121
 トリガ、instead of 60

な

長い識別子 91

は

パーティションのサポート 89

パーティション分割
クエリ・プロセッサ 90
パーティション分割、ハッシュ 89
パーティション分割、範囲 89
パーティション分割、ラウンドロビン 89
ハウスキーピング・タスク 30
パスワード 30
パスワードのセキュリティ 55
バックアップ、データベース 13
バックアップ、トランザクション 13
ハッシュ・テーブル 31

ひ

非常に大きな記憶域のサポート (VLSS) 98
ヒストグラムによる選択性の見積もり 53
ヒストグラム補間 52
ヒュージ・ページ 32
標準偏差 47

ふ

複雑なパスワードの属性 119
符号なし整数値データ型 94
プラグ可能認証モジュール (PAM) 53
分散 47
分散トランザクション管理 (DTM) 33

へ

ベクトル集合 48

め

メッセージング・ライブラリ 121

も

文字セット、中国語 (簡体字) と日本語 46
モジュロ算術 124
モニタリング・テーブル 5, 56
モニタリング・テーブル、ステートメント・
キャッシュ 57

ゆ

ユーザ定義関数 60

よ

予約語 106

り

リテラルのパラメータ化 81
利用できる機能、サポートされるプラットフォーム 1, 9
リラックス持続性データベース 11
リラックス持続性データベース、システム・
プロシージャ 19

ろ

ロー・レベルのロック、システム・テーブル 58
ロー・ロックのシステム・カタログ 90
ログイン・トリガ 54
ログイン・トリガ、グローバル 123
ログイン・マッピング 30

