



新機能の概要

---

**Adaptive Server<sup>®</sup> Enterprise**

**16.0**

ドキュメント ID：DC01373-01-1600-01

改訂：2014年3月

Copyright © 2014 by SAP AG or an SAP affiliate company. All rights reserved.

このマニュアルの内容を SAP AG の明示的許可を得ずに、いかなる手段によっても、複製、転載することを禁じます。ここに記載された情報は事前の通知なしに変更されることがあります。

SAP AG およびディストリビュータが販売しているソフトウェア製品には、他のソフトウェアベンダー独自のソフトウェアコンポーネントが含まれているものがあります。国内製品の仕様は変わることがあります。

これらの資料は SAP AG および関連会社 (SAP グループ) が情報のみを目的として提供するものであり、いかなる種類の表明または保証も行わないものではなく、SAP グループはこの資料に関する誤りまたは脱落について責任を負わないものとします。SAP グループの製品およびサービスに関する保証は、かかる製品およびサービスに付属している明確な保証文書がある場合、そこで明記されている保証に限定されます。ここに記載されているいかなる内容も、追加保証を構成するものとして解釈されるものではありません。

ここに記載された SAP および他の SAP 製品とサービス、ならびに対応するロゴは、ドイツおよび他の国における SAP AG の商標または登録商標です。その他の商標に関する情報および通知については、<http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> を参照してください。

# 目次

<b>バージョン 16.0</b> .....	<b>1</b>
パーティションロックによるデータ可用性の向上 .....	1
HANA サーバ対応の CIS サポート .....	1
クエリ制限の緩和 .....	1
スタージョインによるクエリプランの最適化 .....	2
クエリのパフォーマンス強化 .....	2
全文監査 .....	2
ストアドプロシージャ内の権限チェックの監査 .....	3
オブジェクト定義の置換 .....	3
HTML 形式のクエリプランと実行統計 .....	3
インデックス圧縮 .....	3
SAP JVM サポート .....	4
データベースの完全暗号化 .....	4
スケーラビリティの拡張と機能 .....	4
スレッシュホールドベースイベントの監視 .....	4
複数のトリガ .....	4
残存データの削除 .....	4
設定履歴の追跡 .....	5
dump database 用巡回冗長検査 .....	5
トランザクションログの増大率の計算 .....	5
SAP ASE バージョン 16.0 でのシステムの変更点 .....	5
<b>バージョン 15.7 SP100</b> .....	<b>13</b>
Adaptive Server 15.7 SP100 の機能とプラットフォーム	
フォームの一覧 .....	13
データベースの縮小 .....	15
バックアップとリストアの機能強化 .....	15
構文が類似しているクエリから測定基準を集計 .....	16
事前計算済み結果セットの更新 .....	16

データロードパフォーマンスの向上 .....	16
インクリメンタル再編成 .....	17
データアクセスをブロックしないインデックス作成 .....	17
HTML 形式のクエリプランと実行統計 .....	17
create index によるハッシュベースの統計の収集 .....	17
ブルームフィルタによるクエリプラン最適化 .....	18
カラム値による複写のサポート .....	18
複数のスキヤナスレッドのサポート .....	18
Sybase Central Adaptive Server プラグイン .....	18
VMware vSphere 上での Adaptive Server の再実行 .....	18
OData のサポート .....	19
Windows 64 ビット版での NTLM および MIT	
Kerberos サポート .....	19
応答ファイルに追加されたプロパティ .....	19
自動物理データベース再割り当て .....	19
Adaptive Server バージョン 15.7 SP100 でのシステ	
ムの変更点 .....	19
コマンド .....	20
関数 .....	21
システムストアドプロシージャ .....	22
設定パラメータ .....	23
システムテーブル .....	24
ユーティリティ .....	24
モニタリングテーブルの変更 .....	25
コマンドおよび関数のパーミッション変更 .....	27
<b>バージョン 15.7 ESD #4 .....</b>	<b>29</b>
15.7 ESD #4.2 .....	29
ロールバックレポートの向上 .....	29
15.7 ESD #4 .....	29
リモート Backup Server 名最大長 .....	29
クエリプランの位置決めサポート .....	29
AF_UNIX サポート .....	29

sp_addthreshold および役割への変更 .....	30
Adaptive Server バージョン 15.7 ESD #4 にお けるシステム変更 .....	30
<b>バージョン 15.7 ESD #3 .....</b>	<b>31</b>
sybrestore ユーティリティ .....	31
Adaptive Server バージョン 15.7 ESD #3 におけるシ ステム変更 .....	31
コマンド .....	31
関数 .....	31
システムストアプロシージャ .....	32
モニタリングテーブルの変更 .....	32
<b>バージョン 15.7 ESD #2 .....</b>	<b>33</b>
細密なパーミッション .....	33
述部付きの権限 .....	33
遅延テーブルの作成 .....	34
オンラインユーティリティ .....	35
パーティションのマージおよび分割 .....	35
ステートメントキャッシュに保存されるクエリの最 大サイズ .....	36
show_cached_plan_in_xml の機能強化 .....	36
高速にログを取るバルクコピー .....	36
事前計算済み結果セット .....	37
dump database コマンドと dump transaction コマン ドの同時実行 .....	37
ハッシュベースの update statistics .....	37
dump および load の機能強化 .....	38
データコピーなしの alter table drop column .....	39
最大データベースサイズの拡張 .....	39
ユーザ定義の最適化目標 .....	39
共有クエリプラン .....	40
データベースの非同期的な初期化 .....	40
ロー内ラージオブジェクトの圧縮 .....	41

共有メモリダンプの設定 .....	41
Adaptive Server バージョン 15.7 ESD #2 におけるシ ステム変更 .....	41
コマンド .....	41
関数 .....	44
システムストアプロセス .....	44
設定パラメータ .....	45
システムテーブル .....	46
ユーティリティ .....	48
モニタリングテーブルの変更 .....	48
<b>バージョン 15.7 ESD #1 .....</b>	<b>51</b>
プロキシテーブルにおけるロー内 LOB カラムのサ ポート .....	51
複写マスタデータベースでのシステムプロセス のサポート .....	51
パラメータ数の増加 .....	51
Windows での複数リスナ .....	51
Windows での Adaptive Server 高可用性のサポート ...	52
追加および変更された設定パラメータ .....	52
<b>バージョン 15.7 .....</b>	<b>53</b>
アプリケーション機能設定グループ .....	53
新しい Adaptive Server カーネル .....	53
Adaptive Server のデータの圧縮 .....	54
新しいセキュリティ機能 .....	55
エンドツーエンドの CIS Kerberos 認証 .....	55
暗号化キーのデュアルコントロールと無人起 動 .....	55
ログイン、役割、パスワードの保護のための 管理拡張機能 .....	56
ログインプロファイル .....	56
従業員のライフサイクル管理 .....	57
外部パスワードと隠しテキスト .....	57

キャッシュされた文の抽象プラン .....	57
ログ領域の縮小 .....	58
sysoptions による現在設定されているスイッチの表 示 .....	58
ラージオブジェクトの変更 .....	58
ロー内/ロー外の LOB .....	59
ストアードプロシージャにおけるラージオブ ジェクトの text、unitext、image データ型の 使用 .....	59
Transact-SQL 文における LOB ロケータの使用 .....	59
ラージオブジェクトに対する where 句の拡張 機能 .....	60
XML でのキャッシュされたプランの表示 .....	60
str を使用した文字フィールドの埋め込み .....	60
select for update の変更 .....	61
マテリアライズされていない非 Null カラムの作成 .....	61
インラインデフォルトの共有 .....	62
モニタリングデータの保持 .....	62
動的パラメータの分析 .....	63
ロックタイムアウトのモニタ .....	63
後続の 0 のトランケート .....	64
完全にリカバリ可能な DDL .....	64
merge によるソーステーブルからターゲットテーブ ルへのローの転送 .....	64
sp_showoptstats による統計とヒストグラムの表示 .....	65
カーソルの変更 .....	65
カーソルクローズ時のカーソルロックの解放 .....	65
カーソルの拡張トランザクションサポート .....	65
カーソル文のモニタ .....	66
ネストされた select 文の機能強化 .....	66

連鎖トランザクションのコマンドとシステムプロ シージャの変更 .....	67
可変長のローの拡張 .....	68
like パターン一致の変更 .....	68
引用符付き識別子の変更 .....	69
Unicode 非文字の許可 .....	69
クエリ処理遅延時間の短縮 .....	70
sybdiag ユーティリティ .....	70
オプティマイザの診断ユーティリティ .....	71
Adaptive Server バージョン 15.7 でのシステムの変 更点 .....	72
コマンド .....	72
関数 .....	75
システムストアプロシージャ .....	77
設定パラメータ .....	80
システムテーブル .....	82
ユーティリティ .....	84
モニタリングテーブルの変更 .....	84
グローバル変数 .....	91
<b>バージョン 15.5 Cluster Edition .....</b>	<b>93</b>
Adaptive Server 15.5 Cluster Edition の機能とプラッ トフォームの一覧 .....	93
複数の同時発生フェールオーバ .....	95
アーカイブデータベースへの領域の追加 .....	95
共有ディスククラスタでの分散トランザクション管 理 .....	95
Adaptive Server バージョン 15.5 Cluster Edition にお けるシステム変更 .....	96
変更されたコマンド .....	96
モニタリングテーブル .....	97
設定パラメータ .....	98
関数 .....	99



<b>バージョン 15.5 .....</b>	<b>101</b>
Adaptive Server 15.5 の機能とプラットフォームの一 覧 .....	101
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 .....	103
バックアップの高速圧縮 .....	105
IBM® Tivoli® Storage Manager 用の Backup Server サ ポート .....	105
ユーザ定義のストアドプロシージャ用の遅延名前解 決 .....	106
FIPS 140-2 ログインパスワードの暗号化 .....	106
増分データ転送 .....	107
bigdatetime および bigtime データ型 .....	108
tempdb グループの作成および管理 .....	108
Adaptive Server バージョン 15.5 でのシステムの変 更点 .....	109
データ型 .....	109
関数 .....	109
システムストアドプロシージャ .....	110
コマンド .....	112
設定パラメータ .....	115
モニタリングテーブル .....	115
システムテーブル .....	115
ユーティリティ .....	116
監査 .....	117
<b>バージョン 15.0.3 .....</b>	<b>119</b>
SQL 文の複写 .....	119
セキュリティの機能強化 .....	119
LDAPS ユーザ認証の強化 .....	119
自動的な LDAP ユーザ認証とフェールバック ..	120
外部認証のログインマッピング .....	120
SSL を使用した共通名の指定 .....	120

Kerberos による同時認証 .....	121
仮想ハッシュテーブル .....	121
ヒュージページ .....	122
高可用性設定時のアップグレード .....	123
システムストアプロシージャの再インス トール .....	123
分散トランザクション管理 (DTM) .....	123
Adaptive Server プラグインの更新 .....	124
Java インタフェース .....	125
Adaptive Server バージョン 15.0.3 でのシステムの変 更点 .....	126
関数 .....	126
システムストアプロシージャ .....	126
コマンド .....	127
設定パラメータ .....	127
モニタリングテーブル .....	128
システムテーブル .....	128
<b>バージョン 15.0.2 .....</b>	<b>131</b>
暗号化カラム .....	131
アーカイブデータベースへのアクセス .....	132
長時間実行されているクエリの検出 .....	133
遅延コンパイル .....	133
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 .....	134
統計集合関数 .....	135
標準偏差と分散 .....	135
積極的集約と消極的集約 .....	136
ベクトル集合およびスカラ集合 .....	136
データ挿入のパフォーマンスの向上 .....	137
ページ分割時の非同期書き込みの使用 .....	138
tempdb トランザクションのスループットの向 上 .....	138

コミット後の最適化 .....	139
クエリプロセッサの変更点 .....	140
遅延コンパイル .....	140
バイナリ以外の文字セットのヒストグラム補 間 .....	140
選択性の見積もりのヒストグラム式 .....	141
現在のオプティマイザ設定の表示 .....	141
新しいセキュリティ機能 .....	141
AIX における 64 ビット版 Adaptive Server の PAM サポート .....	141
グローバルログイントリガの自動設定 .....	142
SSL サポート .....	142
パスワードのセキュリティの改善 .....	143
監査機能の強化 .....	143
高可用性に関する考慮事項 .....	144
モニタリングテーブルのインストールおよび編集 .....	144
ステートメントキャッシュ用のモニタリングテーブ ル .....	145
システムテーブルのローレベルのロック .....	146
xmltable() 関数 .....	147
再配置ジョイン .....	147
ユーザ定義の SQL 関数 .....	148
instead of トリガ .....	148
Adaptive Server バージョン 15.0.2 でのシステムの変 更点 .....	149
トレースフラグ .....	149
コマンド .....	150
set コマンドの変更 .....	150
ユーティリティ .....	151
システムストアプロシージャ .....	152
システムテーブル .....	155
設定パラメータ .....	157

関数 .....	159
グローバル変数 .....	159
<b>バージョン 15.0.1 .....</b>	<b>161</b>
抽象プランの変更 .....	161
新しいクエリレベル設定 .....	161
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 .....	162
オプティマイザ基準 set 構文の拡張 .....	163
リテラルのパラメータ化 .....	163
Adaptive Server バージョン 15.0.1 でのシステムの変 更点 .....	164
関数 .....	164
設定パラメータ .....	164
コマンド .....	165
モニタリングテーブル .....	167
<b>バージョン 15.0 .....</b>	<b>169</b>
パーティションのサポート .....	169
ローロックのシステムカタログ .....	170
クエリプロセッサ .....	170
長い識別子 .....	171
計算カラム .....	172
計算カラムと関数ベースインデックスの違い ...	172
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 .....	172
スクロール可能カーソル .....	173
unitext データ型のサポート .....	173
big int データ型のサポート .....	174
符号なし整数値データ型のサポート .....	174
整数 identity .....	175
XML サービスの機能強化 .....	175
Adaptive Server プラグインの機能強化 .....	176
Interactive SQL .....	177

ユーザ定義 Web サービス .....	177
非常に大きな記憶域のサポート .....	178
update statistics の自動実行 .....	178
SySAM ライセンス管理 .....	179
クエリ処理測定基準 (QP 測定基準) .....	180
抽象プランの更新 .....	181
showplan の変更 .....	181
SSL での FIPS 140-2 の使用 .....	182
Adaptive Server バージョン 15.0 でのシステムの変 更点 .....	183
ユーティリティ .....	183
予約語 .....	184
グローバル変数 .....	184
設定パラメータ .....	185
関数 .....	187
コマンド .....	188
システムストアドプロシージャ .....	190
システムテーブル .....	191
モニタリングテーブル .....	195
ヘルプと追加情報の取得 .....	<b>197</b>
サポートセンタ .....	197
製品更新版のダウンロード .....	197
製品およびコンポーネントの動作確認 .....	198
アクセシビリティ機能 .....	198

# 目次

# バージョン 16.0

SAP® Adaptive Server® Enterprise バージョン 16.0 では、多数の新機能と機能強化が導入されています。

## パーティションロックによるデータ可用性の向上

---

新しいパーティションレベルのロック機能を使用して、ロックの細分性をより細かくして同時実行の DDL 文と DML 文の他のパーティションへのアクセスを可能にすることで、データ可用性が向上します。

- 分割されたテーブルの属性としてのパーティションロック
- スキーマロック。これにより、同時オペレーションからの独立性を達成することにより、テーブルスキーマまたはメタデータを更新するパーティションレベルオペレーションを向上することができます。
- ロックの共存性とロックの十分性。ロックと同時実行の問題をサポートします。
- パーティションレベルのロック。テーブルのパーティションのサブセットへのオペレーション (パーティションレベルオペレーション) がアクティブな場合にテーブルデータにアクセスできます。

## HANA サーバ対応の CIS サポート

---

SAP ASE バージョン 16.0 から HANA サーバに直接接続できるようにコンポーネント統合サービス (CIS) にネイティブ ODBC インタフェースを追加しています。

## クエリ制限の緩和

---

SAP ASE 16.0 では、カラムとテーブルの制限が引き上げられています。

## スタージョインによるクエリプランの最適化

---

SAP ASE 16.0 では、**use fact\_table** 抽象プランヒントを導入しています。これは、スタージョインクエリで中央ファクトテーブルを指定し、スタージョインクエリの特別なクエリプラン最適化方式をトリガします。

**use fact\_table** 抽象プランヒントにより、クエリプロセッサはスタージョインクエリに並列ハッシュジョインプランを選択できます。並列プランにより、クエリプロセッサはブルームフィルタプローブ (ディメンションテーブルとファクトテーブル間のより高速なジョインを実現する) を **Xchg** 演算子より低くすることができるので、ファクトテーブルから該当するローをさらに減らします。

## クエリのパフォーマンス強化

---

SAP ASE バージョン 16.0 では、クエリパフォーマンスが向上しています。

- 静的ではなく、動的スレッド割り当てを使用します。動的スレッド割り当てにより、SAP ASE は並列クエリプランをより高速に、かつより少ないリソースで実行できます。SAP ASE は、動的スレッド割り当てを、**select** クエリに生成される並列 Lava クエリプランに適用します。
- 代替リストからの探索回数にのみ必要なカラムの削除などを含め、圧縮データでテーブルで実行されるクエリのパフォーマンスを向上しています。
- SORT 演算子を含む一部の並列クエリのパフォーマンスを向上しています。
- HASH JOIN 演算子を含む特定の並列クエリプランのパフォーマンスの向上とリソース使用量の削減を実現しています。

## 全文監査

---

SAP ASE バージョン 16.0 以降は、全文 DML 監査をサポートしています。DML 監査オプション (*table\_access* と *view\_access* を含む) を有効にすると、機密パラメータはマスクされた状態で DML のパラメータ名とその値が出力されます。

SAP ASE は以下のコマンドの全文監査情報を記録します。

- **select**
- **insert**
- **delete**
- **update**
- **select into**



## ストアドプロシージャ内の権限チェックの監査

---

監査オプション `sproc_auth` を指定すると、システムストアドプロシージャ内で実行される権限チェックの監査が有効になります。

## オブジェクト定義の置換

---

既存のオブジェクト定義を新しい定義に置き換えました。元の名前、オブジェクト ID、監査オプション、パーミッションはそのまま保持されます。

**create or replace** 機能は、新しいオブジェクトの作成、または同じ名前の既存オブジェクトの置換を行います。**or replace** 句により、SAP ASE はデータベース内の同じ名前とタイプの既存オブジェクトを暗黙的に削除して再作成することができます。**or replace** 句は、オブジェクトの定義を変更しますが、パーミッションや監査オプションなどの既存セキュリティ属性を保持するので、付与したパーミッションを保存してオブジェクトを削除して再作成した後、再付与する手間が省けます。

## HTML 形式のクエリプランと実行統計

---

新しい動的スレッド割り当てモデルが導入され、ワーカースレッドの使用が拡張されています。

## インデックス圧縮

---

リレーショナルデータベースの圧縮により、データのより効率的な格納、メモリ消費量の削減、I/O 要求の緩和によるパフォーマンス向上が実現します。

インデックス圧縮は以下をサポートします。

- インデックスリーフページ圧縮
- DOL および APL インデックスリーフページ形式
- テーブル、インデックス、およびローカルインデックスパーティションレベルの圧縮

## SAP JVM サポート

---

SAP ASE バージョン 16.0 以降では、SAP JRE を使用して Java アプリケーションをサポートします。

## データベースの完全暗号化

---

SAP ASE バージョン 16.0 では、データベース全体の暗号化機能を導入し、既存アプリケーションに影響を与えることなくデータベース全体を保護します。

データベースを暗号化すると、データ、インデックス、トランザクションログのすべてが暗号化されます。この暗号化は透過的であるため、ユーザはその違いに気づくことなく、テーブル、インデックスなどの操作を実行できます。

以前のバージョンの SAP ASE では、カラム暗号化のみをサポートしていました。

## スケーラビリティの拡張と機能

---

SAP ASE バージョン 16.0 では、実行時のロギングとロック、メタデータ、およびラッチ管理の強化により、スケーラビリティ機能が向上しています。

## スレッシュホールドベースイベントの監視

---

SAP ASE バージョン 16.0 以降では、スレッシュホールドイベントを設定、記録、リストする機能が含まれています。

## 複数のトリガ

---

複数のトリガを作成し、文実行後トリガを起動する順序を指定します。

## 残存データの削除

---

残存データを削除してデータベースデータのセキュリティを強化します。

スペースを削除するデータベース操作の中には、データが常に物理的に消去されるとは限らないものがあります。この残存データは **dbcc** ユーティリティを使用し

てユーザが見ることができるため、セキュリティ脅威がもたらされる可能性があります。これを防ぐため、SAP ASE バージョン 16.0 では、このようなデータベース操作を実行したときに追加作業なしで残存データを削除する機能が導入されています。

データを機密としてマークし、削除操作または更新操作を実行した後に残存データを消去するように SAP ASE を設定することができます。

## 設定履歴の追跡

SAP ASE バージョン 16.0 では、**sp\_confighistory** システムプロシージャが導入されました。これにより、サーバ設定の変更を追跡できます。

**sp\_confighistory** は、変更された設定オプション、変更前の値および変更後の値、変更したユーザ、および変更した時刻など、SAP ASE 設定の変更を記録します。変更は、sybsecurity の ch\_events ビューに格納され、各設定変更イベントに 1 つのローが含まれています。

## dump database 用巡回冗長検査

SAP ASE では、圧縮で作成されたデータベースダンプまたはトランザクションダンプのローデータに対する偶発的な変更をチェックし、この圧縮ブロックを正しく読み込んで、解凍できるかを検証するための巡回冗長検査が追加されています。

## トランザクションログの増大率の計算

SAP ASE バージョン 16.0 では、指定期間におけるトランザクションログの増大率を計算する、sp\_logging\_rate システムプロシージャが追加されています。

## SAP ASE バージョン 16.0 でのシステムの変更点

SAP ASE 16.0 では、次のシステム変更が導入されています。

グローバル変数  
新しいグローバル変数:

変数	説明
@@trigger_name	実行中のトリガ名を返す。

変数	説明
@@tranrollback	ロールバックが発生している場合、ロールバックのタイプを返す。

設定パラメータ  
新しいパラメータ:

パラメータ	説明
enable utility lvl 0 scan wait	Adaptive Server が独立性レベル 0 スキャンを実行中に、 <b>alter table ... add   drop partition</b> コマンドを実行できる。
large allocation auto tune	クエリ実行用に大量メモリを事前割り付けするよう SAP ASE を設定する。これによりプロシージャキャッシュ競合が減少する。
engine local cache percent	プロシージャキャッシュのパーセンテージとしてエンジンローカルキャッシュを変更できる。
enable large chunk elc	エンジンローカルキャッシュの大量の割り付けを有効にする。
aggressive task stealing	SAP ASE スケジューラタスクスチールポリシーを設定する。
max network peek depth	SAP ASE が保留中のキャンセルのために調査する、接続オペレーティングシステム受信バッファの深さのレベル数を指定する。
user log cache queue size	ロギングのためにキューイング方式が使用されているかどうかを判断する。
threshold event monitoring	スレッシュホールドイベント記録を有効または無効にする。
threshold event max messages	monThresholdEvent テーブルに格納されているイベント数を特定する。

コマンド  
変更されたコマンド:

コマンド	説明
alter data-base	既存データベースを完全暗号化する。
alter index	<b>index_compression</b> 句を使用して、今後のインデックスの挿入または更新の圧縮状態を変更する。

コマンド	説明
<b>alter table</b>	<p>状況に応じて次のようになる。</p> <ul style="list-style-type: none"> <li>インデックス圧縮 - <b>index_compression</b> 句を使用して、今後のインデックスの挿入または更新の圧縮状態を変更する。</li> <li>複数トリガ-テーブル所有者はテーブルで定義された複数トリガのいずれかまたはすべてを無効にできる。</li> <li>残存データ削除 - 削除操作の後、自動的に残余データを削除する。</li> </ul>
<b>create archive database</b>	完全暗号化アーカイブデータベースを作成する。
<b>create database</b>	完全暗号化データベースを作成する。
<b>create default</b>	<b>or replace</b> 句を使用してオブジェクトの定義を置換する。
<b>create encryption key</b>	データベース暗号化キーを作成する。
<b>create function</b>	<b>or replace</b> 句を使用してオブジェクトの定義を置換する。
<b>create function (SQLJ)</b>	<b>or replace</b> 句を使用してユーザ定義 SQLJ 関数の定義を置換する。
<b>create index</b>	<b>index_compression</b> 句を使用してインデックスまたはインデックスパーティションを圧縮する。
<b>create procedure</b>	<b>or replace</b> 句を使用してオブジェクトの定義を置換する。
<b>create procedure (SQLJ)</b>	<b>or replace</b> 句を使用して SQLJ プロシージャの定義を置換する。
<b>create rule</b>	<b>or replace</b> 句を使用してオブジェクトの定義を置換する。
<b>create table</b>	<p>状況に応じて次のようになる。</p> <ul style="list-style-type: none"> <li>インデックス圧縮 - <b>index_compression</b> 句を使用して指定テーブル上のインデックスを圧縮する。</li> <li>残存データ削除 - 削除操作の後、自動的に残余データを削除する新しいテーブルを作成する。</li> </ul>

コマンド	説明
<b>create trigger on order</b>	状況に応じて次のようになる。 <ul style="list-style-type: none"> <li>複数のトリガ - 複数のトリガを作成し、コマンド実行後トリガを起動する順序を指定する。</li> <li><b>or replace - or replace</b> 句を使用する create trigger を使用してオブジェクトの定義を置換する。</li> </ul>
<b>create view</b>	<b>or replace</b> 句を使用する create view を使用してオブジェクトの定義を置換する。
<b>drop encryption key</b>	master データベースの sysencryptkeys テーブルから、データベース暗号化キーを削除する。
<b>drop trigger</b>	既存のトリガまたは複数のトリガの削除や置換に使用することができる。
<b>dump database</b>	圧縮で作成されたデータベースダンプまたはトランザクションダンプのローデータに対する偶発的な変更をチェックするための巡回冗長検査が追加されている。
<b>kill</b>	通常の <b>kill.spid</b> パラメータでプロセスを終了できない場合に使用する、 <b>with force</b> パラメータが追加されている。
<b>load database</b>	圧縮されたデータベースダンプまたはトランザクションダンプのローデータに対する偶発的な変更をチェックするための巡回冗長検査が追加されている。
<b>select</b>	SAP ASE バージョン 16.0 以降上で @変数、@@グローバル変数、および定数のみを chained モードで参照する select 文を発行すると、新しいトランザクションは起動しない。
<b>select into</b>	既存テーブルから選択することにより、インデックス圧縮テーブルを作成する。
<b>set</b>	状況に応じて次のようになる。 <ul style="list-style-type: none"> <li>複数トリガ - <b>set show_trigger_execution</b> は、起動されると、トリガの名前をエラーログに出力し、処理の問題がどこで起きているかを特定する支援をする。</li> <li>残余データ削除 - <b>set erase_residual_data {on   off}</b> は、残余データ削除機能を有効にする。</li> </ul>

## 関数

新しい組み込み関数:

関数	説明
<code>dbencryption_status</code>	データベースの暗号化/復号化ステータスと進行状況をレポートする。

システムストアードプロシージャ  
新しいシステムストアードプロシージャ:

プロシージャ	説明
<code>sp_confighistory</code>	<code>ch_events</code> ビューを作成し、SAP ASE 設定に行われた変更を表示する。
<code>sp_dropglockpromote_ptn</code>	パーティションロックプロモーションの値を削除する。
<code>sp_drowlockpromote_ptn</code>	データベースまたはテーブルから、ローパーティションロックプロモーションスレッショルド値を削除する。
<code>sp_helptrigger</code>	以下のため使用する。 <ul style="list-style-type: none"> <li>テーブルへの DML アクションのそれぞれに対するすべてのトリガのリストを表示する</li> <li>作成時に <code>order</code> 句を指定したトリガの順序を確認する</li> </ul>
<code>sp_jsconfigure</code>	ジョブスケジューラエージェントを設定する。
<code>sp_logging_rate</code>	指定期間におけるトランザクションログの増大率を計算する。
<code>sp_setpglockpromote_ptn</code>	サーバ、データベース、テーブルでのパーティションのページロックプロモーションスレッショルドを設定する。
<code>sp_setrowlockpromote_ptn</code>	サーバ、データベース、テーブルでのパーティションのローロックプロモーションスレッショルドを設定する。

変更されたシステムストアードプロシージャ:

プロシージャ	説明
<code>sp_audit</code>	以下の監査オプションが追加されている。 <ul style="list-style-type: none"> <li><code>config_history</code> - 設定履歴の監査を有効または無効にする。</li> <li><code>sproc_auth</code> - システムストアードプロシージャ内で実行される権限チェックの監査を有効または無効にする。</li> </ul>
<code>sp_chgattribute</code>	パーティションレベルロックを有効または無効にする <code>pnt_locking</code> テーブル属性が追加されている。

プロシージャ	説明
<b>sp_clusterlockusage</b>	パーティションロック情報に固有のクラスタロック使用状況を出力するよう、出力が拡張された。
<b>sp_dboption</b>	<b>erase residual data</b> パラメータを使用して、SAP ASE で削除操作から残余データを削除する。
<b>sp_depends</b>	テーブルに関連する複数トリガをリストする。
<b>sp_encryption</b>	暗号化データベースをサポートする。
<b>sp_familylock</b>	partitionid カラムを含めるよう、出力が拡張された。
<b>sp_helppdb</b>	暗号化データベースの情報を含めるよう、出力が拡張された。
<b>sp_lock</b>	partitionid カラムを含めるよう、出力が拡張された。

### システムテーブル

新しいシステムテーブル:

テーブル	説明
<b>ch_events</b>	各設定変更イベントに1つのローを含む。ch_events は、sysmgmtbdb データベース内にある。

変更されたシステムテーブル:

テーブル	説明
<b>sysattributes</b>	新しいカラムはないが、暗号化データベース機能にはデータベース暗号化を意味する新しいクラス 43 が導入されている。
<b>sysconstraints</b>	新しいカラムはないが、複数トリガについての情報が含まれている。
<b>sysobjects</b>	複数トリガを備える sysobjects が変更されている。

### モニタリングテーブル

新しいモニタリングテーブル:

テーブル	説明
<b>monThresholdEvent</b>	SAP ASE により記録された各イベントにつき1行が含まれる。

変更されたモニタリングテーブル:



テーブル	説明
<b>monCachedStatement</b>	<ul style="list-style-type: none"> <li>5 秒ごとにいくつかのカラムの測定基準を更新する。</li> <li>文が実行を開始したときに UseCount カラムを増分する。</li> <li>現在実行中の文の最大値を説明するカラムの値を増分する。</li> </ul>
<b>monDeadLock</b>	partitionid カラムはパーティションのユニークな識別子を表示する。
<b>monLocks</b>	partitionid カラムはパーティションのユニークな識別子を表示する。
<b>monOpenObjectActivity</b>	多数のカラムを追加する。
<b>monOpenPartitionActivity</b>	多数のカラムを追加する。
<b>monProcess</b>	ClientDriverVersion カラムは、クライアントプログラムが使用する接続ドライバを表示する。
<b>monRepLogActivity、 monRepScannersTotalTime、および monRepSenders</b>	監視データの収集を開始するには、 <b>activate monitoring</b> 設定パラメータを有効にする必要がある。
<b>monRepScanners</b>	[Status] カラムを変更する。
<b>monRepScannersTotalTime</b>	MRPBootstrapTime カラムの名前を BootstrapTime に変更する。
<b>monSysExecutionTime</b>	(Cluster Edition のみ) ExecutionTime カラムは、NetworkIO、DeviceIO、および CIPCIO の値を表示する。
<b>monTables</b>	SAP ASE 16.0 以降の Description カラムは 512 文字をサポートする。以前のリリースは 255 文字をサポートしていた。

テーブル	説明
<b>monTableColumns</b>	<ul style="list-style-type: none"> <li>SAP ASE バージョン 16.0 以降の Description カラムは 512 文字をサポートする。以前のバージョンは 255 文字をサポートしていた。</li> <li>SAP ASE バージョン 16.0 以降の Label カラムは 150 文字をサポートする。以前のバージョンは 50 文字をサポートしていた。</li> </ul>

## ユーティリティ

変更されたユーティリティ:

ユーティリティ	説明
<b>ddlgen</b>	暗号化データベースをサポートする。
<b>sybmigrate</b>	暗号化データベースをサポートする。
<b>sybrestore</b>	マスタデータベースの破損後 SAP ASE サーバのリストアをサポートする。

# バージョン 15.7 SP100

Adaptive Server バージョン 15.7 SP100 では、多数の新機能と機能強化が導入されています。

## Adaptive Server 15.7 SP100 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 15.7 SP100 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ています。"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 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
セキュリティおよびディレクトリサービス	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			
SSL (Secure Sockets Layer)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Adaptive Server 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
MIT Kerberos 注:バージョン 4.0.1 Windows 64 版	Y	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	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Web サービス	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 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

## データベースの縮小

Adaptive Server バージョン 15.7 SP100 以降では、**alter database** コマンドを使用してデータベースを縮小し、未使用の領域を解放して再利用または削除します。

縮小中のデータベースの一部にデータが存在する場合は、そのデータを別のロケーションに移動してから、データベースから領域が削除されます。削除する部分が空になったら、物理記憶領域は、null デバイスへの参照に置き換えられます。これにより、領域が解放され、再利用または削除できるようになります。

オンラインで使用中のデータベースを縮小できます。

## バックアップとリストアの機能強化

Adaptive Server 15.7 SP100 では、**dump database** および **load database** コマンドが機能強化されています。

機能強化は次のとおりです。

- **dump database** コマンドを使用して、最後の完全なデータベースダンプ以降に変更されたすべてのページのコピーを作成する累積バックアップを実行できる
- ダンプ履歴ファイルを使用して、1つのデータベースをバックアップ (**dump**) して別のデータベースにリストア (**load**) できる
- Tivoli Storage Manager とダンプ履歴ファイルの統合

- サーバが壊滅的障害にあったとき、Adaptive Server 外のログデバイスから最後のトランザクションログダンプを生成できる **sybdumptran** ユーティリティ
- データベースオプションと属性を含めるための **listonly=create\_sql** の **load database**

## 構文が類似しているクエリから測定基準を集計

---

Adaptive Server 15.7 SP100 には **show\_condensed\_text** 関数が含まれています。これは、監視データにおける、構文的には区別される (個別のコメントやテキスト形式に差異がある) が、意味的には同じクエリである、ステートメントキャッシュ中の複数の文を集計します。

意味的に等価な文を識別する方法を提供することにより、**show\_condensed\_text** は、`monCachedStatement` からの値をクエリ結果の単一行に集計できます。

## 事前計算済み結果セットの更新

---

事前計算済み結果セットの更新には、**select**、**update statistics**、および **delete statistics** コマンドの **immediate refresh** パラメータで作成した事前計算済み結果セットに対するアクセスパーミッションの付与と取り消しが含まれます。また、**sp\_showoptstats** および **sp\_depends** は、事前計算済み結果セットについての追加情報を表示します。

## データロードパフォーマンスの向上

---

Adaptive Server 15.7 SP100 は、**insert** 文にバルクデータロー挿入を使用する、**ins\_by\_bulk** 最適化基準を有効にすることにより、データロードパフォーマンスを向上します。

**set** 文を使用してデータロード最適化基準を有効にすると、Adaptive Server は以降の **insert...select** またはバッチ挿入文の実行時にこの基準を使用します。抽象プランを介して有効にされた文レベルのディレクティブの範囲は、文に限定されます。

この最適化により、ターゲットテーブルのデータロードパフォーマンスが向上します。

## インクリメンタル再編成

---

Adaptive Server 15.7 SP100 に含まれる **reorg** コマンドの新規パラメータ、**defrag** を使用すれば、再編成のスケジュール/再開、および再編成中のデータへの同時読み込み/書き込みができます。

**reorg defrag** は、オブジェクトのロックスキームとして、各ローまたはページをロックし、処理される各データのまとまりについてトランザクションをコミットします。データ再編成に必要な領域は、アロケーションユニットのサイズ (256 データページ) を超えません。再編成データのインデックス更新は、追加の領域を消費しません。インクリメンタル再編成を実行している各データパーティションについて、`sysattributes` に 1 ローが格納されます。

## データアクセスをブロックしないインデックス作成

---

Adaptive Server 15.7 SP100 に含まれる **create index ... online** パラメータを使用すれば、インデックス作成中のデータへのアクセスをブロックすることなく、インデックスを作成できます。

## HTML 形式のクエリプランと実行統計

---

Adaptive Server バージョン 15.7 SP100 以降では、HTML 形式のグラフィカルなクエリプランを生成して Web ブラウザで表示することができます。

クエリプラン実行統計の図を生成するには、クエリプラン実行時に **HTML set statistics** コマンドを使用します。これらの HTML 表現では、ツリー構造とタイミング図を含むクエリプランの説明が 1 ページに示されます。

## create index によるハッシュベースの統計の収集

---

Adaptive Server 15.7 SP100 では、ハッシュベースの統計の収集サポートが **create index** コマンドにも拡張されます。

**create index** のハッシュベースの統計の収集オプションを使用すれば、インデックス作成中に逐次または並列でマイナーインデックスの属性を収集できます。これにより、インデックス作成後にマイナーインデックスの属性について統計を更新する手間が省けます。

## ブルームフィルタによるクエリプラン最適化

---

Adaptive Server 15.7 SP100 では、join のパフォーマンスを向上することにより、クエリプランを最適化するブルームフィルタを導入しています。

ブルームフィルタにより、join 演算子に到達する前にジョインすることのできないローを早い段階でフィルタリングすることができます。Adaptive Server では、ブルームフィルタは、ハッシュジョイン、ソートマージジョイン、および再フォーマットベースのネストループジョインに対して導入されています。ブルームフィルタは、等価ジョインにのみ適用可能で、コスト計算ベースの最終的なプランについてオプティマイザにより選択されます。

## カラム値による複写のサポート

---

Adaptive Server 15.7 SP100 は、データ値に基づいて複写ローを単一テーブルの1つまたは複数のカラムに分散することをサポートする新しい分散モデルを含めることにより、Multi-Path Replication™ を拡張します。

## 複数のスキャナスレッドのサポート

---

テーブルを監視する Adaptive Server は、RepAgent の複数のスキャナスレッドをサポートします。

## Sybase Central Adaptive Server プラグイン

---

Sybase® Central Adaptive Server プラグインはサポートされなくなりました。Adaptive Server バージョン 15.7 SP100 以降では、SAP Sybase Control Center for Adaptive Server を使用します。

## VMware vSphere 上での Adaptive Server の再実行

---

Adaptive Server バージョン 15.7 SP100 は VMware vSphere をサポートします。



## OData のサポート

---

Adaptive Server バージョン 15.7 SP100 は SAP® Sybase OData Server をサポートします。OData Server は、RESTful HTTP を介したデータサービスを実現し、URI (Universal Resource Identifier) を通じた操作の実行により、情報にアクセスしたり情報を変更したりできます。

## Windows 64 ビット版での NTLM および MIT Kerberos サポート

---

MIT Kerberos および NT Lan Manager セキュリティサービスが、Windows 64 ビット版でサポートされます。

## 応答ファイルに追加されたプロパティ

---

すべてのプラットフォームの応答ファイルに新しいプロパティが追加されました。これらプロパティが追加されたのは、以下の応答ファイルです。

- DO\_UPDATE\_ASE\_SERVER
- UPDATE\_ASE\_SERVER\_NAME\_[n]
- UPDATE\_ASE\_PASSWORD\_[ n]

## 自動物理データベース再割り当て

---

個別のログ/データセグメントを持つデータベースのダンプをロードするとき、Adaptive Server 15.7 SP100 以降では、ログ/データセグメントの物理的分離を確実にするために、ターゲットデータベースの物理レイアウトを再割り当てするようになります。

## Adaptive Server バージョン 15.7 SP100 でのシステムの変更点

---

Adaptive Server 15.7 SP100 では、コマンド、関数、システムプロシージャ、設定パラメータ、システムテーブル、モニタリングテーブルが変更されています。

## コマンド

Adaptive Server 15.7 SP100 では、コマンドが変更されています。

表 1 : 変更されたコマンド

コマンド	説明
<b>alter database</b>	領域を解放するデバイスの名前を指定する、 <b>off</b> パラメーターが追加されている。
<b>create index</b>	<ul style="list-style-type: none"> <li>• <b>online</b> パラメータを使用すれば、インデックス作成中のデータへのアクセスをブロックすることなく、インデックスを作成できる。</li> <li>• <b>sorted_data</b> パラメータを使用して、データオンリーロックテーブルでクラスタードインデックスのために、<b>create index</b> を並列で実行できる。</li> <li>• <b>with statistics</b> ハッシュベースオプションおよび追加のハッシュオプション句を使用してテーブルにインデックス統計を収集する。</li> </ul>
<b>create materialized view</b>	<b>immediate refresh</b> パラメータを含む <b>create materialized view</b> と一緒に <b>like</b> 句を含めることはできない。
<b>create precomputed result set</b>	<b>immediate refresh</b> パラメータを含む <b>create precomputed result set</b> と一緒に <b>like</b> 句を含めることはできない。
<b>dbcc</b>	インクリメンタル再編成を実行している各データパーティションについて、 <b>sysattributes</b> に 1 ローが格納される。新しい <b>dbcc</b> パラメータ <b>zapdefraginfo</b> を使用して <b>sysattributes</b> のこの情報を削除してからダウンロードを実行する。
<b>dump database</b>	新しい <b>cumulative</b> を使用すれば、累積インクリメンタルダンプとしてバックアップを指定できる。
<b>load database</b>	新しい <b>cumulative</b> タイプを使用すれば、 <b>dump database cumulative</b> キーワードで作成したバックアップをロードできる。
<b>reorg</b>	<b>defrag</b> パラメータを使用すれば、再編成のスケジュール/再開、および再編成中のデータへの同時読み込み/書き込みができる。
<b>select</b>	<b>ins_by_bulk</b> パラメータは、テーブルにバルクで新しく割り付けられるページにデータローを直接挿入することにより、パフォーマンスを向上させる。特定の <b>insert</b> 文の抽象プランを使用してクエリレベルでパラメータを設定する。

コマンド	説明
set ins_by_bulk	set コマンドは、セッションで新しい <b>ins_by_bulk</b> パラメータを有効にし、テーブルに新しく割り付けられるページにデータローを直接挿入することにより、パフォーマンスを向上させる。
set join_bloom_filter	クエリプラン最適化のブルームフィルタの使用を有効または無効にする。
set statistics  • plan_detail_html • parallel_plan_detail_html • plan_directory_html	<ul style="list-style-type: none"> <li>HTML 形式のグラフィカルなクエリプランを生成する。</li> <li>各実行段階のそれぞれの演算子で費やされるタイミングに関連する実行統計を含む、HTML 形式のグラフィカルなクエリプランを生成する。</li> <li>名前、実行時にキャプチャされた異なるタイムスタンプ、影響のあったロー数、推定ロー数、経過時間など、プラン演算子の情報詳細を含む、HTML 形式のグラフィカルなクエリプランを生成する。</li> <li>複数のワークスレッドを使用して並列実行されるクエリプランのプランフラグメントとスレッドごとの情報詳細を含む、HTML 形式のグラフィカルなクエリプランを生成する。</li> <li>HTML クエリプランの書き込み先のディレクトリパス名を指定する。</li> </ul>
update statistics	<b>update statistics</b> は、セッションがインメモリデバイスにバインドされている tempdb を使用している場合、tempdb バッファキャッシュではなく、デフォルトデータキャッシュからメモリを取得する。

## 関数

Adaptive Server 15.7 SP100 には、新しい関数が含まれています。

表 2 : 新しい関数

コマンド	説明
defrag_status	指定されたオブジェクトまたはパーティション上で開始されたか、継続中のデフラグオペレーションの測定基準を返す。
loginfo	トランザクションログに関する情報を返す。
show_condensed_text	監視データにおける、構文的には区別される (個別のコメントやテキスト形式に差異がある) が、意味的には同じクエリである、ステートメントキャッシュ中の複数の文を集計する。

コマンド	説明
<code>shrinkdb_status</code>	shrink オペレーションのステータスを特定する。

## システムストアプロシージャ

Adaptive Server 15.7 SP100 では、システムストアプロシージャが追加および変更されています。

表 3 : 新しいシステムストアプロシージャ

システムストアプロシージャ	説明
<code>sp_helpdefrag</code>	コマンドが起動されたデータベースの <b>reorg defrag</b> の対象オブジェクトすべて、または、 <b>reorg defrag</b> の対象である特定オブジェクトのデフラグ情報をレポートする。
<code>sp_dump_info</code>	<b>sp_dump_info</b> は、特定の時間に累積ダンプに含まれているログとデータのサイズを表示する。

表 4 : 変更されたシステムストアプロシージャ

システムストアプロシージャ	説明
<code>sp_configure</code>	<ul style="list-style-type: none"> <li><b>enable concurrent dump tran</b> パラメータを 1 に設定することにより、トランザクションダンプと累積データベースダンプの同時ダンプを有効にする。</li> <li><b>utility statistics hashing</b> パラメータを含めることにより、インデックス作成時にハッシュベース統計を使用してインデックス属性の収集を有効にする。</li> </ul>
<code>sp_config_rep_agent</code>	<b>multipath distribution model</b> プロパティの追加分散モデル <b>filter</b> をサポートする。
<code>sp_dboption</code>	累積ダンプのバックアップとリストアをサポートする <b>allow incremental dumps</b> パラメータを含む。
<code>sp_depends</code>	事前計算済み結果セットの追加情報を表示する。
<code>sp_dump_history</code>	<b>@dump_type</b> パラメータの <b>cumulative</b> キーワードを指定したとき、累積データベースダンプに完全ダンプと同じ機能をサポートする。
<code>sp_extrapwdchecks</code>	<b>caller_password</b> および <b>loginame</b> パラメータで NULL 値をサポートする。

システムストアド プロシージャ	説明
<b>sp_helpdb</b>	出力には、ユーザが作成したテンポラリデータベースの持続性についての情報が含まれる。
<b>sp_helprotect</b>	オブジェクト (テーブル、データベースなど) の <code>object</code> カラムに所有者名が含まれる。
<b>sp_listener</b>	AF_UNIX プロトコルのサポートが追加されている。
<b>sp_optgoal</b>	細密なパーミッションが有効の場合、目標の作成/削除をするには、 <code>manage opt goal</code> のあるユーザでなければならない。
<b>sp_replication_path</b>	フィルタのバインド、バインド解除、リストをサポートする。
<b>sp_showoptstats</b>	事前計算済み結果セットの追加情報を表示する。
<b>sp_sjobcreate</b>	<b>continuous_run</b> パラメータを含む。これを使用すると、ジョブを開始時から、指定した期間にしたがって終了時まで実行することができる。
<b>sp_sjobmodify</b>	<b>continuous_run</b> パラメータを含む。これを使用すると、ジョブを開始時から、指定した期間にしたがって終了時まで実行することができる。

## 設定パラメータ

Adaptive Server 15.7 SP100 では、設定パラメータが追加および変更されています。

### 新しい設定パラメータ

設定パラメータ	説明
<b>enable ISM</b>	Adaptive Server による Solaris プラットフォームでの統合サービス管理 (ISM) の使用を有効/無効にする。
<b>enable HugePages</b>	Adaptive Server による Huge Page をサポートする Linux プラットフォームでの Huge Page (ISM) の使用を有効/無効にする。
<b>max util parallel degree</b>	<b>create index with consumers</b> および <b>update stats with consumers</b> コマンドにより使用されるクエリごとに許可される、サーバ全体の最大ワーカープロセス数を指定する。
<b>optimize temp table resolution</b>	

設定パラメータ	説明
<b>solaris async i/o mode</b>	Solaris プラットフォームでさまざまな非同期 IO モードを選択できる。

#### 変更された設定パラメータ

設定パラメータ	説明
<b>enable permissive unicode</b>	有効時、ランダムバイナリデータを含むことができる。
<b>permission cache entries</b>	タスク当たりのキャッシュプロテクタ数を指定し、各ユーザ接続とワーカプロセスのメモリ量を増やす。

## システムテーブル

Adaptive Server 15.7 SP100 では、システムテーブルが追加および変更されています。

#### 新しいシステムテーブル

テーブル	カラム
sysdams	データベースのグローバルアロケーションマップ (DAM) を保存する。

#### 変更されたシステムテーブル

テーブル	カラム
sysloginroles	predid カラムが追加されている。これは、 <b>grant role</b> コマンドの述部のオブジェクト ID である。

## ユーティリティ

Adaptive Server 15.7 SP100 では、ユーティリティが追加および変更されています。

#### 新しいユーティリティ

コマンド	説明
<b>ddlgen</b>	<i>Xextended_object_type</i> 、XDE が -TDB に追加されている。この新しいタイプを使用すると、データベースとそのすべてのオブジェクトを正確な依存順序で生成できる。
<b>optdiag</b>	<b>optdiag</b> には、 <b>prs_name</b> パラメータが追加されている。

コマンド	説明
<b>sybdumptran</b>	以前は Adaptive Server 環境のデータベースによりログデバイスとして使用されていた、オペレーティングシステムファイル/ローデバイスに含まれていたログページからトランザクションログダンプを生成する。

#### 変更されたユーティリティ

コマンド	説明
<b>Backup Server</b>	<b>dump database</b> 、 <b>dump transaction</b> 、 <b>load database</b> 、および <b>load transaction</b> コマンドで <b>mountwait -M</b> パラメータをサポートする。
<b>optdiag prs_name</b>	事前計算済み結果セットの名前を示す。

### モニタリングテーブルの変更

Adaptive Server バージョン 15.7 SP100 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。

#### 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
monRepCoordinator	RAT が複数のスキャナモードで実行される場合、RAT コーディネータプロセスの情報を提供する。
monRepSchemaCache	Replication Server で実行されている各スキャナのスキーマキャッシュ情報をレポートする。
monSysExecutionTime	実行したオペレーションの実行時間 (マイクロ秒) を提供する。

#### 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	説明
monOpenObjectActivity	カラム NumLevel0Waiters および AvgLevel0WaitTime が追加されている。
monRepLogActivity	monRepLogActivity テーブルの MaxHashSchemaSize および NumberOfSchemasReused カラムは削除された。

モニタリングテーブル	説明
monRepScanners	NumberOfTruncPointRequested および NumberOfTruncPointMoved カラムは、ログモデルの単一タスクスキャン使用時のみ入力される。
monRepSenders	<p>以下のカラムが追加されている。</p> <ul style="list-style-type: none"> <li>• NumberOfTruncPointRequested</li> <li>• NumberOfTruncPointMoved</li> <li>• AvgTruncPointInterval</li> </ul>
monSpinlockActivity	SpinlockSlotID カラムが追加されている。
monState	<p>以下のカラムが追加されている。</p> <ul style="list-style-type: none"> <li>• TableAccesses</li> <li>• IndexAccesses</li> <li>• ULCFlushes</li> <li>• ULCFlushFull</li> <li>• WorkTables</li> <li>• TempDBObjects</li> <li>• Rollbacks</li> <li>• Selects</li> <li>• Updates</li> <li>• Inserts</li> <li>• Deletes</li> <li>• Merges</li> <li>• ULCKBWritten</li> <li>• PagesRead</li> <li>• PagesWrite</li> <li>• PhysicalReads</li> <li>• PhysicalWrites</li> <li>• LogicalReads</li> </ul>



## コマンドおよび関数のパーミッション変更

Adaptive Server 15.7 SP100 では、コマンドおよび関数のパーミッションが変更されています。

### 変更されたパーミッション

設定パラメータ	説明
<b>asehostname</b>	細密なパーミッションが有効の場合、データサーバが動作しているホストの名前をクエリするには、 <code>manage server</code> 権限を持っているか、 <b>asehostname</b> で <code>select</code> パーミッションを付与されている必要がある。細密なパーミッションが無効の場合、 <code>sa_role</code> を持つユーザであるか、 <b>asehostname</b> に対する <code>select</code> パーミッションを付与されている必要がある。
<b>dbcc pravailable-tempdbs</b>	細密なパーミッションが有効の場合、 <code>pravailabletempdbs</code> を実行するための <code>manage server</code> 権限を持つユーザである必要がある。細密なパーミッションが無効の場合、 <code>sa_role</code> を持つユーザである必要がある。
<b>dbcc serverlimits</b>	細密なパーミッションが有効の場合、 <code>serverlimits</code> を実行するための <code>manage server</code> 権限を持つユーザである必要がある。細密なパーミッションが無効の場合、 <code>sa_role</code> を持つユーザである必要がある。
<b>dbcc cis showcaps</b>	細密なパーミッションが有効の場合、 <code>cis showcaps</code> を実行するための <code>manage server</code> 権限を持つユーザである必要がある。細密なパーミッションが無効の場合、 <code>sa_role</code> を持つユーザである必要がある。
<b>dbcc cis remcon</b>	細密なパーミッションが有効の場合、 <code>cis remcon</code> を実行するための <code>manage server</code> 権限を持つユーザである必要がある。細密なパーミッションが無効の場合、 <code>sa_role</code> を持つユーザである必要がある。
<b>sp_optgoal</b>	新しいサーバレベル権限である <code>manage opt goal</code> を使用すれば、 <code>sa_role</code> および <code>sa_serverprivs_role</code> を持つユーザは、 <b>sp_optgoal</b> を使用して目標を書き込みまたは削除できる。



# バージョン 15.7 ESD #4

Adaptive Server バージョン 15.7 ESD #4 では、多数の新機能と機能強化が導入されています。

## 15.7 ESD #4.2

---

Adaptive Server Enterprise 15.7 ESD #4.2 の新機能を次に示します。

### ロールバックレポートの向上

Adaptive Server 15.7 ESD #4.2 以降では、**kill with statusonly** コマンドを使用することによってレポートが拡張されます。

## 15.7 ESD #4

---

Adaptive Server Enterprise 15.7 ESD #4 の新機能を次に示します。

### リモート Backup Server 名最大長

**dump** および **load** コマンドのリモート Backup Server 名最大長が 30 文字から 255 文字に増加しました。

### クエリプランの位置決めサポート

設定オプションの動的 SQL プランの位置決めが Adaptive Server 15.7 ESD #4 で導入され、クエリプランマネージャへのアクセスを待つサーバ接続にかかる時間を減らすことにより、Adaptive Server パフォーマンスを向上します。

### AF\_UNIX サポート

Adaptive Server バージョン 15.7 SP100 以降では、**interfaces** ファイルで AF\_UNIX の名前付きパイプエントリをサポートします。これにより、TCP プロトコルでもたらされるオーバーヘッドの一部を回避できます。

AF\_UNIX エントリを作成するには、**sp\_listener** システムプロシージャを使用します。

## sp\_addthreshold および役割への変更

スレシヨルドプロシージャ作成時にアクティブなシステムおよびユーザ定義の役割は `systhresholds` に保存されます。

## Adaptive Server バージョン 15.7 ESD #4 におけるシステム変更

Adaptive Server 15.7 ESD #4 には、ユーティリティおよび設定パラメータに対する変更が含まれています。

### ユーティリティ

Backup Server では、**dump database**、**dump transaction**、**load database**、および **load transaction** コマンドで Tivoli Storage Manager **mountwait -M** パラメータがサポートされるようになりました。

### 設定パラメータ

Adaptive Server 15.7 ESD #4 では 2 つの新しい設定パラメータが導入されました。

設定パラメータ	説明
<b>early row send increment</b>	結果セットの 2 番目以降のパケットに送信される追加ロー数 (最大パケットサイズによる) を設定する。
<b>number of early send rows</b>	Adaptive Server がクライアントの新しい結果セットの最初のパケットに送信するロー数を設定する。

# バージョン 15.7 ESD #3

Adaptive Server バージョン 15.7 ESD #3 では、多数の新機能と機能強化が導入されています。

## sybrestore ユーティリティ

**sybrestore** は、Adaptive Server データベースを最新の完全なデータベースバックアップダンプファイルから障害時の状態にリストアします。

ESD #3 の『ASE ユーティリティガイド』を参照してください。

## Adaptive Server バージョン 15.7 ESD #3 におけるシステム変更

Adaptive Server 15.7 ESD #3 では、関数およびコマンドに対する変更がなされています。

### コマンド

Adaptive Server 15.7 ESD #3 では、コマンドが変更されています。

表 5 : 変更されたコマンド

コマンド	説明
<b>writetext</b>	IRL カラムの <b>writetext</b> logs オペレーション。 <b>writetext</b> コマンド ( <b>log</b> パラメータあり/なし) を、 <b>online</b> パラメータと同時に実行できる。

### 関数

Adaptive Server 15.7 ESD #3 には、新しい関数が含まれています。

表 6 : 新しい関数

コマンド	説明
<b>allocinfo</b>	オブジェクトアロケーションマップ (OAM) ページに格納されているアロケーションページのリストを返す。

コマンド	説明
<code>spaceusage</code>	Adaptive Server の領域使用の測定基準をカンマ区切り文字列として返す。

## システムストアプロシージャ

Adaptive Server 15.7 ESD #3 では、ストアプロシージャが追加および変更されています。

表 7: 変更されたシステムストアプロシージャ

システムストアプロシージャ	説明
<code>sp_helpconfig</code>	<b>estimate</b> パラメータは、メモリを大量に使用する一部の設定パラメータに必要なおおよそのメモリ量を特定する。
<code>sp_dboption</code>	<b>deallocate first text page</b> オプションが <code>sp_dboption</code> に追加された。

## モニタリングテーブルの変更

Adaptive Server バージョン 15.7 ESD #3 では、モニタリングテーブルが追加されています。

モニタリングテーブル	説明
<code>monMemoryUsage</code>	測定基準、そのサイズ、使用パターン、利用可能性などのカーネルメモリプールとサーバについての情報を提供する。

# バージョン 15.7 ESD #2

Adaptive Server® バージョン 15.7 ESD #2 では、多数の新機能と機能強化が導入されています。

## 細密なパーミッション

---

細密なパーミッションを使用すると、システム権限を付与する際に、サイト固有の役割を要件に一致する権限で構成したり、システム管理者やデータベース所有者によるユーザデータへのアクセスを制限したりできます。

付与可能なシステム権限は細密化されているため、“作業の分割”の原則(特定のオペレーションセットにおいて、各個人がセット内のすべてのオペレーションを実行できないようにする必要がある)や“最小権限”の原則(情報システムのすべてのユーザが、そのジョブの実行に必要な権限をできる限り抑えて実行できることが求められる)を適用できます。

付与された権限はいずれも変更できません。つまり、ある権限から別の権限を取り消したり、ある権限を別の権限に付与したりすることはできません。ただし、付与対象者が実行できる内容が権限間で重複しても構いません。ある権限を所有すると、さらに細密な別の権限を所有することになる場合があります。

細密なパーミッションを有効にすると、システム標準の役割(sa\_role、sso\_role、oper\_role、および replication\_role)が、明示的に付与された権限のセットで構成される権限コンテナとして再構成されます。明示的に付与されたシステム権限をシステム標準の役割から取り消して、役割に再付与することができます。

Adaptive Server で細密なパーミッションを設定し、使用方法の詳細については、『セキュリティ管理ガイド』の「細密なパーミッションの使用」を参照してください。細密なパーミッションを Adaptive Server で有効にした場合に個々のコマンドおよびシステムプロシージャに与える影響を確認するには、『リファレンスマニュアル：コマンド』および『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。

## 述部付きの権限

---

述部付きの権限は、ローレベルでの柔軟なアクセス制御の仕組みを提供します。この仕組みを利用することにより、データアクセス時に Adaptive Server が評価する述部に基づいて、さまざまなユーザ、グループ、役割に対して権限を付与、選

択、更新、および削除できるようになります。いずれのデータローも述部で表された条件に一致しない場合は、そのローを結果セットで使用することはできません。

述部付きの権限が提供するデータプライバシーの保護は、データ内容やコンテキスト情報に応じて動的にユーザに権限を付与するローレベルのアクセス制御に基づいています。そのため、クライアントや Web サーバではなく、サーバにプライバシーポリシーを実装することができます。

述部は、テーブル、SQL 関数、組み込み関数など、他のオブジェクトにアクセスする場合があります。このようなアクセスは述部の所有者 (付与者など) のパーミッションや役割に対してチェックされるので、述部がアクセスするオブジェクトに **select**、**update**、または **delete** コマンドを実行するユーザに明示的にパーミッションを与える必要はありません。

述部付きの権限を使用すると、サービスプロバイダが 1 つのデータベースにデータを保存し、同じテーブルを複数の顧客で共有できるので、顧客ごとに個別のビューやトリガを用意せずに済みます。

Adaptive Server で述部付きの権限を設定し、使用方法の詳細については、『セキュリティ管理ガイド』の「述部付きの権限の付与」を参照してください。

## 遅延テーブルの作成

---

**create table...with deferred\_allocation** を使用すると、テーブルのページ割り付けを遅延させることができます。

**create table** コマンドの **with deferred\_allocation** パラメータを使用すると、テーブルのページ割り付けを遅延させることができます。遅延テーブルは、アプリケーションで多数のテーブルを作成してそのうちの少数のテーブルだけを使用する場合に役立ちます。Adaptive Server がテーブルのページを割り付けるまで、そのテーブルは "遅延" と呼ばれます。

システムテーブルには遅延テーブルのエントリが含まれます。これらのエントリを使用すると、遅延テーブルに関連付けられたオブジェクト (ビュー、プロシージャ、トリガなど) を作成できます。

Adaptive Server は最初のローを挿入 (テーブルをマテリアライズ) するときに遅延テーブルに対してページを割り付けます。最初の **insert** が実行される前にテーブルにアクセスすると (選択、削除、更新など)、領域使用量を報告する関数や他のテーブルへの DML 操作で参照整合性制約を適用する関数は、テーブルが空であるかのように動作します。つまり、遅延テーブルに対して **select** を実行すると空の結果セットが生成されるということです。遅延テーブルにインデックスを作成で



きますが、作成したインデックスに対するページ割り付けは、Adaptive Serverがこのテーブルをマテリアライズするまで遅延されます。

## オンラインユーティリティ

---

Adaptive Server バージョン 15.7 ESD #1 以降には **reorg rebuild** に対する **online** パラメータが含まれています。このパラメータを使用すると、ユーザのユーザデータをブロックせずにデータの再編成やテーブルのメンテナンスを実行できます。

## パーティションのマージおよび分割

---

時間の経過とともに、パーティションのデータ分布に偏りが見られるようになる場合や、最初にデータを分割した方法が現在のビジネス要件に適さなくなる場合があります。**alter table** を使用してパーティションをマージ、分割、または移動し、データを再分配すると、パーティションの使用によるパフォーマンス上のメリットが回復します。

例：

- パーティションの分割 - ある企業では、データを地域に応じて4つのパーティション(北部、南部、東部および西部)に分割しています。そのため、顧客担当者は担当する地域と他の地域とを切り離して、高速かつ効率的に顧客にアクセスすることができます。南部で売上が増加し、顧客基盤が大幅に拡大した場合、パーティションスキャンや保守操作に関連するクエリが頻繁に実行されることによって南部パーティションでのパフォーマンスが低下し、効率が悪くなる恐れがあるため、顧客データを分割することのメリットが損なわれます。この場合、南部パーティションのデータを2つのパーティション(南東部および南西部)に分割することによって、他のパーティションのデータに影響を与えずにパフォーマンスを回復させることができます。
- パーティションのマージ - ある企業の売上データが年度の四半期(Q1、Q2、Q3、およびQ4)で分割されています。この企業では年度末にその年のデータをマージしてアーカイブします。過去の年度の売上データに対してはアクセス頻度が低く、また、古いデータは更新ではなく読み取り対象として使用される可能性が高いため、事業年度が終了したパーティションをマージすると効果的です。

## ステートメントキャッシュに保存されるクエリの最大サイズ

---

Adaptive Server バージョン 15.7 ESD #2 以降では、非常に長い SQL 文を保存できません。最大 2MB (64 ビットマシンの場合) の個々の文をステートメントキャッシュに保存できます。

バージョン 15.7 ESD #2 より前の Adaptive Server では、ステートメントキャッシュサイズが大きなサイズで設定されていたとしても、ステートメントキャッシュに保存される個々の文は 16K までに制限されていました。

## show\_cached\_plan\_in\_xml の機能強化

---

Adaptive Server バージョン 15.7 ESD #2 以降には、**show\_cached\_plan\_in\_xml** に対する新しい情報が含まれています。

**show\_cached\_plan\_in\_xml** の出力には以下の情報が含まれます。

- スキャンの範囲
- ワークテーブル
- 動的パーティション削除
- 合計論理 I/O (lio) および合計物理 I/O (pio)

## 高速にログを取るバルクコピー

---

Adaptive Server バージョン 15.7 ESD #2 以降では、完全なログを取る **bcp** を高速モードで使用できます。これにより、データスループットが高速化され、完全なデータリカバリが可能になります。前のバージョンでは、ページ割り付けのログのみが取られていました。

**set logbulkcopy {on | off}** コマンドを使用して、高速にログを取る **bcp** をセッションに対して設定します。**set logbulkcopy {on | off}** に **--initstring 'Transact-SQL\_command'** パラメータを指定することができます。このパラメータを指定すると、Transact-SQL コマンドを Adaptive Server に送信してから、データが転送されます。たとえば、titles.txt のデータを pubs2..titles テーブルに転送する場合にロギングを有効にするには、次のように入力します。

```
bcp pubs2..titles in titles.txt --initstring 'set logbulkcopy on'
```

**fast-logged bcp** を発行する前に、データベースで **select into/bulkcopy/pllsort** を有効にする必要があります。有効にしない場合は、**bcp** で低速モードが使用されます。

## 事前計算済み結果セット

---

Adaptive Server バージョン 15.7 ESD #2 以降では、事前計算済み結果セットを作成できます。

事前計算済み結果セットは、結果が計算され、保存されたビューであり、将来的な使用に備えて用意されるものです。事前計算済み結果セットの設定を行うと、Adaptive Server が事前にクエリを計算し、計算済みの結果を使用して実際のクエリに応答しようとしています。事前計算済み結果セットは、マテリアライズされたビューとも呼ばれます。

概念上、事前計算済み結果セットはビューでもあり (システムテーブルに保存されたクエリ定義を含むため)、テーブルでもあります (永続的なデータを含むため)。インデックスの作成、`update statistics` の実行など、テーブルに実行するさまざまな操作を、事前計算済み結果セットにも実行できます。

事前計算済み結果セットには `create`、`alter`、および `refresh` コマンドを使用します。

## dump database コマンドと dump transaction コマンドの同時実行

---

Adaptive Server バージョン 15.7 ESD #2 以降では、`dump transaction` コマンドと `dump database` コマンドを同時に実行できます。これにより、データベースの更新がダンプポリシーで設定された時間よりも長くかかる場合に、その更新を損失する危険性が低下します。

## ハッシュベースの update statistics

---

Adaptive Server バージョン 15.7 ESD #2 以降では、ソートベースの統計を使用する代わりに、マイナーインデックスの属性およびインデックスのないカラムに対してハッシュベースの統計を収集することにより、経過時間とリソース使用量を大幅に削減することができます。ハッシュベースの統計を使用すると、必要なスキャン回数が減少し、ディスクベースのソートが回避されるため、パフォーマンスが向上します。

ハッシュベースの統計はソートベースの統計よりも、以下の点で柔軟性に優れています。

- ハッシュベースの統計は実行時間が短く済むため、メンテナンス時間中により多くの作業を実行できます。

- ハッシュベースの統計では必要なプロシージャキャッシュが少なく済むため、メンテナンス時間外にデータオンリーロックテーブルに対して **update statistics** を実行できます。また、Adaptive Server の tempdb バッファキャッシュ (通常はデフォルトデータキャッシュを使用) は、一般的にプロシージャキャッシュよりも大容量になるため、**update statistics** による影響が軽減されます。
- 通常、ハッシュベースの統計では、tempdb に対して大容量のディスク割り付けは必要ありません。update statistics からの大規模なソートに対応するために tempdb のサイズを事前に増やしている場合は、この領域を再配備できます。
- ハッシュを使用した **update [index | all] statistics** は、サンプリングを使用した **update [index | all] statistics** よりも実行速度が速くなる場合があります。ただし、**update statistics table\_name (col\_name)** は例外です。
- **update statistics table\_name** ハッシュを使用した *(col\_name1), (col\_name2)...* では、複数のカラムに関する統計を、複数回ではなく 1 回のスキャンで収集できます。

## dump および load の機能強化

Adaptive Server 15.7 ESD #2 では、**dump** および **load** コマンドの機能が強化されています。そのため、データベースのバックアップとリストアを簡単に実行することができます。

機能強化は次のとおりです。

- **dump configuration** コマンドを使用すると、Adaptive Server の設定ファイル、ダンプ履歴ファイル、およびクラスタ設定ファイルをバックアップできます。
- ダンプ設定によって、データベースダンプ作成時のオプションを定義します。その後、Backup Server がこの設定を使用してデータベースダンプを実行します。次の設定を使用できます。
  - ダンプ設定を作成、変更、リストするためのダンプ設定。この設定は **dump database** または **dump transaction** で使用します。
  - **enforce dump configuration** 設定パラメータ。ダンプ操作でダンプ設定を使用できるようにします。
  - 設定グループ "dump configuration"。ユーザが作成したダンプ設定を示します。
- ダンプ履歴:
  - **dump database** および **dump transaction** コマンドの履歴をダンプ履歴ファイルに保持します。このファイルは、Adaptive Server があとで特定の時点までデータベースをリストアする際に使用します。

- ダンプ履歴ファイルを読み込み、データベースのリストア時に必要な SQL 文のロードシーケンスを再生成します。
- **sp\_dump\_history** を使用して、履歴レコードを消去します。
- **enable dump history** 設定パラメータを使用して、各ダンプ操作の最後でダンプ履歴ファイルに対してデフォルトで行われる更新を無効にします。
- **dump history filename** 設定パラメータを使用して、ダンプ履歴ファイルの名前を指定します。
- ダンプヘッダ - **dump with listonly** コマンドの新しいオプションは次のとおりです。
  - **create\_sql** - ソースデータベースと同じレイアウトでターゲットデータベースを作成する際に必要な **disk init**、**sp\_cacheconfig**、**create database**、および **alter database** コマンドのシーケンスをリストします。
  - **load\_sql** - ダンプ履歴ファイルを使用して、特定の時点までデータベースを再移植する際に必要な **load database** および **load transaction** コマンドのリストを生成します。

---

## データコピーなしの **alter table drop column**

---

Adaptive Server バージョン 15.7 ESD #2 以降では、**alter table ... drop column** コマンドに対する **no datacopy** パラメータが追加されています。このパラメータを使用すると、データコピーを実行せずにテーブルからカラムを削除できるので、**alter table ... drop column** の実行に必要な時間が短縮されます。

---

## 最大データベースサイズの拡張

---

Adaptive Server バージョン 15.7 ESD #2 以降では、論理ページ番号が符号付き整数から符号なし整数に変換されたため、データベースの最大サイズが約 64 テラバイトまで拡張されます。

バージョン 15.7 ESD #2 より前の Adaptive Server では、使用可能な最大データベースサイズは約 32 テラバイトでした。

---

## ユーザ定義の最適化目標

---

Adaptive Server バージョン 15.7 ESD #2 以降では、ユーザ定義の最適化目標を作成できます。

ユーザ定義の最適化目標は、以下の機能を提供します。

- オプティマイザの新しい目標の作成
- 一連のアクティブな基準の定義
- サーバレベル、セッションレベル、およびクエリレベルでの目標のアクティブ化
- 目標内容の動的な変更(クライアントセッションの切断および再接続は不要)

ユーザ定義の最適化目標を作成したら、その目標をサーバレベルで、またはユーザセッションに対して呼び出すことができます。

## 共有クエリプラン

---

Adaptive Server バージョン 15.7 ESD #2 以降では、クエリプランを共有できます。このプランは、Adaptive Server が既存のプランと同じクエリプランを作成または再コンパイルする必要がないようにプライマリクエリプランから作成されたクローンです。

Adaptive Server はクエリプランを再使用したり再コンパイルする代わりに共有するため、パフォーマンスが向上します。Adaptive Server が共有クエリを使用している間はプライマリクエリプランがキャッシュに固定されるため、プロシージャキャッシュメモリの使用量に若干の変化が見られる場合があります。

## データベースの非同期的な初期化

---

Adaptive Server バージョン 15.7 ESD #2 以降には、**alter database** および **create database** コマンド用の **async\_init** パラメータが含まれています。このパラメータを使用すると、データベースの使用中にデータベースを非同期的に初期化できます。

データベースの初期化はユーザにとって透過的です。データベースは、初期化が完了した時点ではなく、作成または変更された時点ですぐに使用可能になります。

初期化されていないデータベースのページを使用するタスクは、そのページが存在するアロケーションユニットを初期化します。

非同期的な初期化は、**create database** または **alter database** コマンドで起動されるサービスタスクによって実行されます。Adaptive Server を再起動すると、自動的に新しいサービスタスクが起動され、初期化が完了します。クラスタ環境では、サービスタスクを実行しているインスタンスが障害を起こしたり、停止したりした場合は、コーディネーティングインスタンスによって新しいサービスタスクが起動され、初期化が完了します。

## ロー内ラージオブジェクトの圧縮

Adaptive Server バージョン 15.7 ESD #2 以降では、ロー内ラージオブジェクト (LOB) の圧縮がサポートされます。

## 共有メモリダンプの設定

Adaptive Server バージョン 15.7 ESD #2 以降では、圧縮された共有メモリダンプを、構成可能な特定の条件に従って自動的に実行できます。

共有メモリダンプで実行される圧縮量は、**memory dump compression level** 設定パラメータを使用して設定します。共有メモリダンプを設定するには、**sp\_shmdumpconfig** を使用します。

## Adaptive Server バージョン 15.7 ESD #2 におけるシステム変更

Adaptive Server 15.7 ESD #2 では、コマンド、関数、システムプロシージャ、設定パラメータ、システムテーブル、モニタリングテーブルが変更されています。

### コマンド

Adaptive Server 15.7 ESD #2 では、コマンドが追加および変更されています。

表 8 : 新しいコマンド

コマンド	説明
<b>create {precomputed result set   materialized view}</b>	事前計算済み結果セットを定義する。
<b>alter {precomputed result set   materialized view}</b>	事前計算済み結果セットのプロパティまたはポリシーを変更する。
<b>refresh {precomputed result set   materialized view}</b>	指定された事前計算済み結果セットをリフレッシュする。
<b>drop {precomputed result set   materialized view}</b>	事前計算済み結果セットを削除する。

コマンド	説明
<b>truncate {precomputed result set   materialized view}</b>	事前計算済み結果セットのデータをトランケートする。
<b>dump configuration</b>	Adaptive Server の設定ファイルのバックアップを、指定されたダンプディレクトリに作成する。コピーは Backup Server ではなく Adaptive Server によって作成される。

表 9 : 変更されたコマンド

コマンド	変更点
<b>alter database</b>	<b>noasync_init</b> - データベースを拡張し、拡張された領域を Adaptive Server が非同期に初期化することを示す。
<b>alter table</b>	<ul style="list-style-type: none"> <li>• <b>with immediate_allocation</b> - 通常のテーブル (遅延テーブルではない) を作成する。</li> <li>• <b>split partition</b> - 2 つ以上のパーティションにデータを再分配する。</li> <li>• <b>merge partition</b> - マージと互換性のある 2 つ以上のパーティションを 1 つのパーティションに結合する。</li> <li>• <b>move partition</b> - パーティション (およびそのインデックス) を指定のセグメントに移動する。</li> <li>• <b>drop column</b> - データをコピーせずにテーブルからカラムを削除する。</li> <li>• <b>noasync_init</b> - データベースが同期的に初期化されることを示す。</li> </ul>
<b>create database</b>	<ul style="list-style-type: none"> <li>• <b>noasync_init</b> - データベースが同期的に初期化されることを示す。</li> </ul>
<b>create index</b>	並列形式の <b>create index</b> を発行し、より効率的にコマンドが実行されるようにクエリ実行エンジンを使用できる。
<b>create table</b>	<ul style="list-style-type: none"> <li>• <b>with deferred_allocation</b> - 遅延テーブルを作成する。</li> <li>• <b>with immediate_allocation</b> - 通常のテーブル (遅延テーブルではない) を作成する。</li> </ul>
<b>dump database</b>	ダンプ構成ファイルの設定に従ってデータベースをダンプする。
<b>grant</b>	<ul style="list-style-type: none"> <li>• <b>as pred_name</b> - 述部の名前。</li> <li>• <b>grantby grantor</b> - ユーザまたは役割に権限を付与する付与者を示す。</li> </ul>



コマンド	変更点
<b>grant role</b>	<b>where pred_expression</b> - 名前付き役割をアクティブにする際に満たす必要がある SQL の条件。
<b>load database</b>	新しいパラメータ： <ul style="list-style-type: none"> <li>• <b>listonly=load_sql</b> - データベースを特定の時点までリストアするための <b>load database</b> と <b>load transaction</b> の SQL 文のシーケンスを生成する。</li> <li>• <b>until_time</b> - データベースがこの日時までリストアされる。</li> <li>• <b>listonly=create_sql - create database</b> と <b>alter database</b> の SQL 文のシーケンスを生成する。</li> </ul>
<b>load transaction</b>	<b>listonly=create_sql - create database</b> と <b>alter database</b> の SQL 文のシーケンスを生成する。
<b>merge</b>	<b>merge</b> のターゲットテーブルにトリガを含めることができる。
<b>reorg rebuild</b>	<b>with online</b> - データをオフラインにせずに再編成できる。
<b>set</b>	<ul style="list-style-type: none"> <li>• <b>materialized_view_optimization</b> - クエリ最適化時にどの事前計算済み結果セットを考慮するかを決定する。</li> <li>• <b>mon_stateful_history</b> - 無効にした場合、履歴モニタリングテーブルに対するクエリでは、テーブルバッファ内にあるすべてのローが返される。有効にした場合、履歴モニタリングテーブルに対するクエリでは、<b>mon_stateful_history</b> が無効にされてからテーブルに追加されたローのみが返される。</li> <li>• <b>show_transformed_sql</b> - Adaptive Server の前処理フェーズおける文の SQL テキストを表示する。</li> </ul>
<b>update statistics</b>	<ul style="list-style-type: none"> <li>• <b>no_hashing</b> - (デフォルト) バージョン 15.7 ESD #2 よりも前の Adaptive Server のソートまたはハッシュアルゴリズムを使用する。</li> <li>• <b>partial_hashing</b> - ユニークな件数が低いドメインに対してハッシュを使用する。</li> </ul>

**update index statistics**、**update statistics**、および **update all statistics** には **print\_progress** パラメータがあります。このパラメータを使用すると、これらのコマンドで進行メッセージを表示することができます。

『リファレンスマニュアル: コマンド』を参照してください。

## 関数

Adaptive Server 15.7 ESD #2 では、関数が追加および変更されています。

### 変更された関数

以下の関数は、int の代わりに unsigned の結果を返すようになりました。

- **curunreservedpgs** (lstart および **unreservedpgs** パラメータは符号なし int も返します)
- **used\_pages**
- **data\_pages**
- **reserved\_pages**
- **lct\_admin**

### 新しい関数

Adaptive Server 15.7 ESD #2 では、以下の関数が追加されています。

- **show\_cached\_text** - キャッシュされた文の SQL テキストを表示する
- **show\_cached\_text\_long** - キャッシュされ、長さが 16 K を超える文の SQL テキストを表示する

## システムストアプロシージャ

Adaptive Server 15.7 ESD #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 10 : 新しいシステムストアプロシージャ

システムストアプロシージャ	説明
<b>sp_config_dump</b>	ダンプ設定をリスト、追加、変更できる。
<b>sp_dump_history</b>	ダンプ履歴ファイルからダンプレコードを消去できる。元のダンプ履歴ファイルは、ファイル名にタイムスタンプのサフィックスが付けられて保存される。
<b>sp_optgoal 'show','goal_name'</b>	ユーザが作成したオプティマイザの目標によってアクティブになる個々の基準をすべてレポートする。
<b>sp_restore_system_role</b>	システム標準の役割またはシステムデフォルトのデータベース所有者の権限をレポートする。
<b>sp_shmdumpconfig</b>	共有メモリダンプを設定する。

表 11 : 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
<b>sp_dboption</b>	遅延テーブルの割り付けパラメータを追加し、以降に作成されるすべてのユーザテーブルに対して Adaptive Server がページ割り付けを遅延するようにデータベースを設定する。
<b>sp_checksource</b>	<i>objname</i> に対する述部付きの権限を含めることができる。
<b>sp_hidetext</b>	<i>objname</i> に対する述部付きの権限を含めることができる。

以下のシステムプロシージャは述部付きの権限に関する情報を表示します。

- **sp\_helprotect**
- **sp\_help**
- **sp\_helptext**
- **sp\_checksource**

以下のシステムプロシージャは細密なパーミッションに関する情報を表示します。

- **sp\_help**
- **sp\_helprotect**

『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。

## 設定パラメータ

Adaptive Server 15.7 では、設定パラメータが追加および変更されています。

新しい設定パラメータ	説明
<b>enable concurrent dump tran</b>	ダンプの同時実行を有効または無効にする。
<b>enable predicated privileges</b>	Adaptive Server で述部付きの権限の使用を有効にする。
<b>update statistics hashing</b>	Adaptive Server でハッシュベースの統計の収集を有効にする。
<b>enforce dump configuration</b>	Adaptive Server でダンプ設定を使用するかどうかを決定する。
<b>enable dump history</b>	データベースダンプの最後でのダンプ履歴ファイルへの更新を有効または無効にする。
<b>dump history filename</b>	ダンプ履歴ファイルのパスを指定する。

新しい設定パラメータ	説明
<code>enable plan sharing</code>	共有クエリプランを使用する。
<code>enable async database init</code>	すべての <code>create database</code> および <code>alter database</code> コマンドで、データベースの初期化が非同期に行われるようにする。
<code>memory dump compression level</code>	共有メモリダンプの圧縮レベルを制御する。

Adaptive Server バージョン 15.7 ESD #2 では、以下の設定パラメータに対して必要な役割がシステム管理者からシステムセキュリティ担当者に変更されています。

- `allow updates to system tables`
- `SQL Perfmon Integration`
- `syb_sendmsg port number`

## システムテーブル

Adaptive Server 15.7 ESD #2 では、システムテーブルが変更されています。

表 12 : 変更されたシステムテーブル

システムテーブル	追加されたカラム	説明
<code>sysattributes</code>	<ul style="list-style-type: none"> <li>• <code>object_cinfo2</code></li> <li>• <code>object_datetime</code></li> </ul>	<ul style="list-style-type: none"> <li>• オブジェクトの説明を提供する</li> <li>• オブジェクトの日付および時刻を提供する</li> <li>• <code>SPobject_type</code> は RSA キーペア再生成に関連するオプションを格納し、LR <code>object_type</code> はログインプロファイルに関連するオプションを格納する</li> </ul>
<code>sysobjects</code>	<code>type</code>	以下を追加する <ul style="list-style-type: none"> <li>• RS - 事前計算済み結果セットを示す</li> <li>• PP - 述部権限を示す</li> </ul>

システムテーブル	追加されたカラム	説明
sysprotects	<ul style="list-style-type: none"> <li>pred_id</li> <li>protstatus</li> </ul>	<ul style="list-style-type: none"> <li>述部付き権限のオブジェクト ID</li> <li>次のいずれか <ul style="list-style-type: none"> <li>PROT_PREDICATED - 権限(または拒否)が含まれることを示す</li> <li>PROT_ROW_FILTER - where 句が述部であることを示す</li> </ul> </li> </ul>

以下のテーブルのカラムのデータ型は int か unsignedint に変更されています。

表 13 : int から unsigned int に変更されたカラム

テーブル	カラム
sysusages	<ul style="list-style-type: none"> <li>lstart</li> <li>size</li> <li>unreservedpgs</li> </ul>
sysaltusages	<ul style="list-style-type: none"> <li>lstart</li> <li>size</li> </ul>
syspartitions	<ul style="list-style-type: none"> <li>firstpage</li> <li>rootpage</li> <li>dataoampage</li> <li>indoampage</li> </ul>
systabstats	<ul style="list-style-type: none"> <li>leafcnt</li> <li>pagecnt</li> <li>emptypgcnt</li> <li>warmcachepgcnt</li> <li>unusedcnt</li> <li>oampgct</li> </ul>
syslocks	page
syslogshold	page
systhresholds	free_space

## ユーティリティ

Adaptive Server 15.7 では、以下のユーティリティが変更されています。

コマンド	説明
<b>bcp</b>	Adaptive Server バージョン 15.7 ESD #2 以降では、高速モードで完全なログを取る <b>bcp</b> を使用できる。これにより、完全なデータリカバリが可能になる。
<b>optdiag</b>	バージョン 15.7 ESD #2 の <b>optdiag</b> の出力を前のバージョンの <b>optdiag</b> で読み取ることはできない。 バージョン 15.7 ESD #2 で読み取ることでできる出力ファイルを作成するには、前のバージョンの <b>optdiag</b> で <b>optdiag -T1</b> フラグを使用する。
<b>dataserver</b>	<ul style="list-style-type: none"> <li>• <b>-Asystem_role - enable granular permissions</b> が 0 に設定されており、すべてのユーザが Adaptive Server にログインできない場合、<b>sso_role</b> が付いたログインアカウントをサーバ管理者に提供する</li> <li>• <b>-n system_privileges - enable granular permissions</b> が 1 に設定されており、すべてのユーザが Adaptive Server にログインできない場合、パスワード変更権限が付いたログインアカウントをサーバ管理者に提供する</li> </ul>

## モニタリングテーブルの変更

Adaptive Server 15.7 ESD #2 では、いくつかのモニタリングテーブルが変更されています。

monCachedStatement には、以下のカラムが追加されています。

モニタリングテーブル	説明
AvgScanRows	実行 1 回あたりに読み取られたスキャン済みローの平均数
MaxScanRows	実行 1 回あたりに読み取られたスキャン済みローの最大数
AvgQualifyingReadRows	<b>read</b> コマンドの実行 1 回あたりに該当したデータローの平均数
MaxQualifyingReadRows	<b>read</b> コマンドの実行 1 回あたりに該当したデータローの平均数

モニタリングテーブル	説明
AvgQualifyingWriteRows	<b>write</b> コマンドの実行 1 回あたりに該当したデータローの平均数
MaxQualifyingWriteRows	<b>write</b> コマンドの実行 1 回あたりに該当したデータローの最大数
LockWaits	ロック待機の合計回数
LockWaitTime	ロック待機に費やされた合計時間 (ミリ秒単位)
SortCount	ソートオペレーションの合計回数
SortSpilledCount	ディスクにあふれたソートオペレーションの合計回数
TotalSortTime	ソートに費やされた合計時間 (ミリ秒単位)
MaxSortTime	ソートに費やされた最大時間 (ミリ秒単位)





# バージョン 15.7 ESD #1

Adaptive Server バージョン 15.7 ESD #1 では、多数の新機能と機能強化が導入されています。

## プロキシテーブルにおけるロー内 LOB カラムのサポート

---

Adaptive Server® バージョン 15.7 ESD #1 では、ロー内ラージオブジェクト (LOB) カラムを作成できます。ただし、リモートのロー内 LOB カラムに関する情報は、プロキシテーブルのメタデータに保存されないため、ソーステーブルまたはターゲットテーブルの LOB カラムに関する情報が使用できない場合は、ターゲットテーブルのロー外にデータが保存されます。

## 複写マスターデータベースでのシステムプロシージャのサポート

---

Adaptive Server 15.7 ESD #1 では、複写マスターデータベースで以下のシステムプロシージャを複写できます。

- `sp_addexternlogin`
- `sp_dropexternlogin`
- `sp_maplogin`
- `sp_addrmotelogin`
- `sp_dropremotelogin`
- `sp_addserver`
- `sp_dropserver`

## パラメータ数の増加

---

Adaptive Server 15.7 ESD #1 では、動的 SQL 文およびパラメータ化した言語の文に含めることができるパラメータの最大数が 2048 から 32767 に増えています。

## Windows での複数リスナ

---

Adaptive Server バージョン 15.7 ESD #1 以降では、ローカルコンピュータ上で識別される個々のネットワーク接続でリスナタスクが開始されます。これにより、使

用している物理的なネットワーク接続に関係なく、クライアントが確実に接続できるようになります。

## Windows での Adaptive Server 高可用性のサポート

---

64 ビット版の Windows オペレーティングシステムで稼動する Adaptive Server バージョン 15.7 では高可用性がサポートされます。

## 追加および変更された設定パラメータ

---

Adaptive Server バージョン 15.7 ESD #1 では、**network polling mode** 設定パラメータが追加され、**number of network tasks** の設定が変更されています。

表 14 : 新しい設定パラメータ

設定パラメータ	説明
<b>network polling mode</b>	<b>network polling mode</b> を <b>threaded</b> に設定すると、Adaptive Server は、ポーリングを実行するように設定されているネットワークタスクごとに個別のスレッドを生成する。 <b>inline</b> に設定した場合は、いずれかのエンジンがポーリングを実行する。

表 15 : 変更された設定パラメータ

設定パラメータ	説明
<b>number of network tasks</b>	<b>number of network tasks</b> は、 <b>network polling mode</b> が <b>threaded</b> に設定されている場合にのみ機能する。

## バージョン 15.7

Adaptive Server® バージョン 15.7 では、多数の新機能と機能強化が導入されています。

### アプリケーション機能設定グループ

---

Adaptive Server バージョン 15.7 では、設定ファイルにアプリケーション機能設定グループが追加されています。

アプリケーション機能設定パラメータは次のとおりです。

- `enable functionality group`
- `select for update`
- `streamlined dynamic SQL`
- `enable inline default sharing`
- `enable permissive unicode`
- `quoted identifier enhancements`

このグループのすべての設定パラメータを有効または無効にするには、`enable functionality group` を使用します。グループ値を上書きするには、個々の設定パラメータを有効または無効にします。

『システム管理ガイド 第1巻』の「設定パラメータ」を参照してください。

### 新しい Adaptive Server カーネル

---

Adaptive Server バージョン 15.7 以降では、スレッドカーネルおよびプロセスカーネルという 2 つのカーネルが含まれています。

Adaptive Server を設定したカーネルによって、Adaptive Server で実行するモードが決まります。

- スレッドモード - Adaptive Server は 1 つのマルチスレッドオペレーティングシステムプロセスとして実行され、スレッドプールのスレッドで実行されるエンジンにより SQL クエリが処理されます。スレッドモードでは、エンジンを使用しないスレッドを活用して I/O を管理します。管理者は追加スレッドプールを設定して負荷を管理できます。
- プロセスモード - 前バージョンまでの Adaptive Server が使用していた従来のカーネル。プロセスモードでは、Adaptive Server が 1 つのサーバとして機能す

る複数のオペレーティングシステムプロセスとして実行されます。プロセスモードでは、エンジンを使用して I/O を管理し、管理者はエンジングループを設定して負荷を管理します。

スレッドモードは多くの負荷に対して、プロセスモードよりも大幅に少ない CPU を使用して、同等以上のパフォーマンスを提供します。スレッドモードには、タスクとエンジンとの結び付きがあまり必要ないため、I/O 集中利用負荷と CPU 集中利用負荷が混在する場合でも、より一定したパフォーマンスを発揮します。

スレッドカーネルを使用すると、以前のバージョンのカーネルよりも多くのプロセッサ、プロセッサコア、ハードウェアスレッドを持つ並列ハードウェアとサポートシステムを Adaptive Server で利用することができます。バージョン 15.7 ではカーネルが変更されていますが、クエリプロセッサに変更はありません。スレッドカーネルモードを実行するには、以前のバージョンの Adaptive Server 用に作成したほとんどのスクリプトは変更の必要がありませんが、一部のコマンドとストアードプロシージャが変更されています。アプリケーションはスレッドモードと完全互換です。

スレッドモードは Adaptive Server のデフォルトモードで、Sybase® ではこのモードを推奨しています。Windows プラットフォームの Adaptive Server はスレッドモードのみで実行されます。

詳細については、Adaptive Server をスレッドモードに設定する方法については、『システム管理ガイド 第 2 巻』を参照してください。

## Adaptive Server のデータの圧縮

---

Adaptive Server バージョン 15.7 では、データ圧縮が導入されています。データ圧縮を行うと、同じ容量のデータをより小さい記憶領域に格納して、キャッシュメモリの消費量を削減し、I/O 要求の緩和によってパフォーマンスを向上させることができます。

ラージオブジェクト (LOB) データと通常データを圧縮できます。

圧縮テーブルまたはパーティションを作成すると、それ以降に挿入または更新したデータ (まだ圧縮されていない既存のデータ) はすべて自動的に圧縮されます。挿入されたデータを Adaptive Server が効率的に圧縮できない場合、元のローは保持されます。新しく挿入または更新された LOB データが単一データページと同等かこれ以下の領域を占めている場合、Adaptive Server はこのデータを圧縮しません。

データに対してクエリを実行するには、そのデータを展開する必要があります。圧縮データを挿入、更新、および削除できます。select または readtext 文を圧縮カラムに対して実行すると、圧縮解除されたローが返されます。Adaptive Server に

よって検索されるデータが少なくなるため、I/O が少なくなり、データ記憶領域の効率性が向上します。

データ圧縮は、別途ライセンスするオプションです。『圧縮ユーザズガイド』を参照してください。

## 新しいセキュリティ機能

---

Adaptive Server バージョン 15.7 では、エンドツーエンドの CIS Kerberos 認証、暗号化キーのデュアルコントロールと無人起動、ログイン、役割、パスワードの保護のための管理拡張機能、ログインプロファイルというセキュリティ機能が追加されています。

### エンドツーエンドの CIS Kerberos 認証

Adaptive Server バージョン 15.7 には、コンポーネント統合システム (CIS) を介してリモート Adaptive Server 接続をサポートするためのエンドツーエンドの Kerberos 認証が追加されています。

エンドツーエンドの (CIS) Kerberos 認証を使用すると、Kerberos 認証を使用して Adaptive Server にログインした Kerberos V5 ユーザは、以下を行う際に Kerberos 統一化ログイン認証により、リモート Adaptive Server に接続することができます。

- Adaptive Server に RPC を要求
- CIS パススルー接続を発行
- CIS を使用してリモート Adaptive Server に一般的な分散クエリ処理要求を発行

Adaptive Server では、CIS を使用したリモート ASE サーバーへの Kerberos 接続に対する次のオプションセキュリティサービスをサポートしています。

- メッセージの機密保持
- メッセージの整合性
- 相互認証

『セキュリティ管理ガイド』を参照してください。

### 暗号化キーのデュアルコントロールと無人起動

Adaptive Server バージョン 15.7 では、暗号化キーのデュアルコントロールと無人起動機能が追加されています。

デュアルコントロールと知識分割、および無人起動には、次の変更が加えられています。

- master および dual master システムキーは sso\_role または keycustodian\_role のユーザにより作成されたデータベースレベルのキーで、ユーザが作成した暗

号化キーに対するキー暗号化キー (KEK) として使用することで、データ暗号化キーのセキュリティと知識分割を向上します。master キーは、現在のシステム暗号化パスワードに置き換わるものです。Adaptive Server では、下位互換性のためにシステム暗号化パスワードも引き続きサポートされます。Sybase では、ユーザにシステム暗号化パスワードを使用してデータ暗号化キーを暗号化しないことをおすすめします。

- SQL コマンドとプライベートファイルにより master および dual master キーのパスワードを設定する機能。マスタキーのパスワードは非永続的で、データベースに格納されません。
- デュアルコントロールと知識分割によりユーザが作成したすべてのキーを保護する機能。

『暗号化カラムユーザズガイド』を参照してください。

## ログイン、役割、パスワードの保護のための管理拡張機能

Adaptive Server バージョン 15.7 には、ログイン、役割、パスワードの保護のための管理拡張機能が追加されています。

Adaptive Server では、以下によりログイン、役割、パスワードを強化しています。

- ディスク上に保管される役割パスワードの暗号化の強化
- ロックされた役割アカウント
- ログイン、役割、グローバルパスワードポリシーオプションに関する ISO 8601 の持続時間の指定
- RSA (Rivest-Shamir-Adleman) キーペア生成時間、役割パスワードに対するパスワード複雑性チェックの拡張、高可用性環境でのパスワード管理サポートの拡張などのパスワード管理拡張機能
- 役割の定義、アクティブ化、アクティブ化試行の失敗による役割のロック、RSA キーペア再生成などのパスワード管理拡張機能の監査機能の強化

『セキュリティ管理ガイド』を参照してください。

## ログインプロファイル

Adaptive Server 15.7 では、ログインプロファイルが追加されています。ログインプロファイルはログイン属性とその値の SQL 定義のコンテナです。

ログインアカウントは属性により定義され、制御されています。たとえば、ログインアカウントの追加または変更時に、次の属性がログインアカウントと関連付けられます。

- 使用するデータベース
- 自動的にアクティブ化する役割
- 適用する言語

- Adaptive Server へのログイン時に呼び出すログインスクリプト

デフォルトログインプロファイルに属性を指定することで一部の属性をすべてのログインアカウントに関連付けたり、ログインプロファイルに属性を指定してこのログインプロファイルをログインアカウントと関連付けることで他の属性を特定のログインアカウントに関連付けることができます。

『セキュリティ管理ガイド』を参照してください。

## 従業員のライフサイクル管理

システムセキュリティ担当者またはデータベース所有者は、**alter... modify owner** を使用してデータベースオブジェクトの所有権を譲渡することができます。

データベース管理者は、このコマンドを使用して従業員の変化に応じてオブジェクトの割り当てを管理したり、データベースオブジェクトの作成所有権を分割することができます。

## 外部パスワードと隠しテキスト

Adaptive Server 15.7 では、AES-256 対称暗号化アルゴリズムを使用して外部ログインパスワードと隠しテキストの暗号化を強化しています。

強化された外部パスワードの暗号化は次のパスワードに適用されます。

- Replication Agent - 複写データベースのパスワード。
- CIS - リモート記述子とログインのパスワード。
- Job Scheduler - Job Scheduler Agent のパスワード。
- RTMS - Real Time Messaging Services のパスワード
- SSL (Secure Socket Layer) と LDAP (Lightweight Directory Access Protocol) - SSL と LDAP アクセスアカウントのパスワード。ストアードプロシージャ `sp_ldapadmin` および `sp_ssladmin` を使用して管理するパスワードは保護することができます。

## キャッシュされた文の抽象プラン

Adaptive Server バージョン 15.7 では、ステートメントキャッシュに抽象プラン情報を保存する機能が導入されています。

抽象プランを含むこの例では、ハッシュテーブルは、次の SQL TEXT 行に示されたように **select \* from t1 plan '(use optgoal allrows\_mix)'** を保存します。

15.7 より前のバージョンの Adaptive Server では、SQL TEXT 行には、`select * from t1` コマンドのみが含まれ、`plan` 句はありませんでした。

『パフォーマンス&チューニングシリーズ: クエリ処理と抽象プラン』を参照してください。

## ログ領域の縮小

---

Adaptive Server バージョン 15.7 以降では、**alter database** にデータベースログの不要な部分を削除する **log off** パラメータが含まれ、データベースを再作成しなくてもログ領域を縮小して記憶領域を解放することができます。

**log off** は、**select into**、**alter table**、**reorg rebuild** などのデータベース操作のフルログを取る操作を実行した後、データベースに割り当てられた追加スペースが不要になった場合に特に有益になります。

このドキュメントの「完全にリカバリ可能な DDL」および『システム管理ガイド 第 2 巻』の「ユーザデータベースの作成と管理」を参照してください。

## sysoptions による現在設定されているスイッチの表示

---

Adaptive Server バージョン 15.7 では、現在設定されているスイッチのスイッチ ID を含む `number` カラムが `sysoptions` テーブルに追加されています。

`sysoptions` には以下のスイッチが表示されます。

- `runserver` ファイルに `-T` フラグが設定されたトレースフラグ
- `dbcc traceon(flag_number)` または `set switch serverwide on` が設定されたトレースフラグ
- 特定のシステムプロセス ID (`spid`) に対して `set switch on` が設定されたトレースフラグおよびスイッチ

`sysoptions` には、ユーザが参照可能なスイッチのみが表示されます。つまり、ユーザは他の `spid` によりプライベートに設定されているスイッチを表示できません。`number` の値は、スイッチ以外のすべてのオプションカテゴリに対して Null です。

## ラージオブジェクトの変更

---

Adaptive Server バージョン 15.7 では、小さい `text`、`image`、`unitext` データ型のロー内の LOB カラムの保存、LOB を含む宣言された SQL 文の保存、Transact-SQL 文での LOB の間接的参照、ラージオブジェクトの null 値のチェックの許可など、ラージオブジェクト (LOB) に対する変更が含まれています。



## ロー内/ロー外の LOB

Adaptive Server 15.7 では、ページ内の空き領域に応じて、text、image、unitext データ型の小さなロー内の LOB カラムの保存をサポートしています。

LOB のサイズが拡大するか、空き容量が他のロー内のカラム (varchar や varbinary データ型に使用されるカラムなど) に使用されている場合、Adaptive Server ではロー内の LOB データがロー外の記憶領域に連続的に移行され、データがロー内テキストポインタに自動的に置換されます。

Adaptive Server 15.7 では、以下を使用できます。

- **create table** による LOB カラムのロー内の記憶領域の指定
- **alter table** による LOB カラムの保存方法の変更
- **create database** または **alter database** コマンドによるデータベース全体での LOB カラムのロー内の長さの管理

『Transact-SQL ユーザーズガイド』の「ロー内/ロー外の LOB」を参照してください。

## ストアドプロシージャにおけるラージオブジェクトの text、unitext、image データ型の使用

いったん宣言されると、Adaptive Server は LOB を含む SQL 文を格納します。

Adaptive Server バージョン 15.7 以降では、以下のことができます。

- ローカル変数に対してラージオブジェクト (LOB) の text、image、または unitext データ型を宣言し、その変数を入力パラメータとしてストアドプロシージャに渡すことができます。
- LOB パラメータを含む SQL 文を作成できます。

『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 文における LOB ロケータの使用

ラージオブジェクト (LOB) ロケータを使用すると、LOB 自身を参照する代わりに、Transact-SQL 文で LOB を間接的に参照することができます。

text、unitext、または image の LOB のサイズは数メガバイトになることがあるため、Transact-SQL 文に LOB ロケータを使用することで、クライアントと Adaptive Server 間のネットワークトラフィックを低減し、クライアントによる LOB の処理に必要なメモリ量を低減することができます。

Adaptive Server 15.7 では、クライアントアプリケーションでホスト変数およびパラメータマーカとしてロケータを送受信することができます。

LOB ロケータを作成すると、Adaptive Server でメモリ内に LOB 値がキャッシュされ、それを参照する LOB ロケータが生成されます。

LOB ロケータの作成後は、作成されたトランザクションの期間にわたって有効です。Adaptive Server では、トランザクションのコミット時またはロールバック時にロケータが無効になります。

LOB ロケータは 3 種類のデータ型を使用します。

- `text_locator` - `text` の LOB 用
- `unitext_locator` - `unitext` の LOB 用
- `image_locator` - `image` の LOB 用

『Transact-SQL ユーザーズガイド』の「データ型の使用と作成」を参照してください。

## ラージオブジェクトに対する where 句の拡張機能

ラージオブジェクトの `null` 値のチェックを許可するために、`where` 句が拡張されています。

『リファレンスマニュアル：コマンド』を参照してください。

## XML でのキャッシュされたプランの表示

`show_cached_plan_in_xml` 関数は、キャッシュ内の文に対して XML で `showplan` 出力をリターンします。

`show_cached_plan_in_xml` を使用する前に、ステートメントキャッシュを有効にする必要があります。

『パフォーマンス&チューニングシリーズ：クエリ処理と抽象プラン』の「クエリ最適化方式と見積りの表示」を参照してください。

## str を使用した文字フィールドの埋め込み

Adaptive Server バージョン 15.7 では、`str` 関数の `decimal` パラメータが、指定した文字または数値でフィールドを埋め込むように拡張されています。

『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。

## select for update の変更

---

Adaptive Server バージョン 15.7 では、同じトランザクション内の後続の更新、および更新可能なカーソルのためにローの排他ロックを行うための **select for update** がサポートされています。これにより、同時に実行される他のタスクがこれらのローを更新したり、後続の更新をブロックすることを防止できます。**select for update** は独立性レベル 1、2、3 でサポートされています。

**select for update** は、カーソルコンテキストの外部の言語文として発行できます。言語文とカーソルのいずれの場合でも、**begin transaction** コマンドまたは連鎖モード内で **select for update** を実行する必要があります。

カーソルコンテキストで **select for update** を実行する場合、カーソルの **open** および **fetch** 文がトランザクションのコンテキスト内である必要があります。そうでない場合は、Adaptive Server が 15.7 より前の機能に戻ります。

『Transact-SQL ユーザーズガイド』の「クエリ: テーブルからのデータの選択」を参照してください。

## マテリアライズされていない非 Null カラムの作成

---

Adaptive Server バージョン 15.7 では、マテリアライズされていない非 Null カラムを作成できます。

マテリアライズされていないカラムは仮想的に存在しますが、ロー内に物理的に格納されるわけではありません。マテリアライズされていないカラムは、選択、更新、SQL クエリでの参照、インデックスキーとしての使用で、他のカラムと同じように使用します。

Adaptive Server では、マテリアライズされていないカラムが、null カラムと同様に処理されます。カラムがロー内に物理的に存在しない場合は、Adaptive Server によりデフォルトが提供されます。null 入力可能なカラムのデフォルトは null ですが、マテリアライズされていないカラムのデフォルトはユーザ定義の非 NULL 値になります。

『Transact-SQL ユーザーズガイド』の「データの追加、変更、転送、削除」を参照してください。

## インラインデフォルトの共有

---

Adaptive Server 15.7 では、同じデータベース内にあるテーブル間でのインラインデフォルトの共有がサポートされています。

Adaptive Server では、新しいインラインデフォルトを作成する前に、同じユーザに属するデータベース内で同じ値を持つ既存の共有可能なインラインデフォルトを検索します。既存の共有可能なデフォルトが検出された場合は、新しいデフォルトを作成する代わりに、このオブジェクトがカラムにバインドされます。しかし、既存の共有可能なインラインデフォルトが検出されない場合には、新しいデフォルトが作成されます。

Adaptive Server では、tempdb にあるインラインデフォルトを共有できません。

『Transact-SQL ユーザーズガイド』の「データのデフォルトとルールの定義」を参照してください。

## モニタリングデータの保持

---

Adaptive Server バージョン 15.7 では、メタデータキャッシュに一部のオブジェクトの記述子が格納されません。代わりに、記述子内に格納されたモニタリングデータを保持することによりクエリパフォーマンスを向上しています。

メタデータキャッシュは制限リソースで、制限された数のオブジェクト記述子を格納することができます。キャッシュに記述子を追加する場合、他の記述子がキャッシュからフラッシュされる可能性があります。

Adaptive Server では、以下の関数を実行した場合に、まだキャッシュに存在しないオブジェクトの記述子を廃棄してリソースの浪費を回避します。

- data\_pages
- used\_pages
- reserved\_pages
- object\_id
- row\_count
- datachange
- derived\_stat

『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。

## 動的パラメータの分析

---

Adaptive Server バージョン 15.7 では、クエリの実行前に動的パラメータ (疑問符で表示) を分析することにより、非効率的なクエリプランを避けることができます。

以下を使用した動的パラメータの分析：

- `@@lwpid` グローバル変数 - 動的 SQL prepare 文に該当する、最も最近に準備されたライトウェイトプロシージャのオブジェクト ID をリターンします。
- `@@plwpid` グローバル変数 - 動的 SQL prepare 文に該当する、最後から 2 番目に準備されたライトウェイトプロシージャのオブジェクト ID をリターンします。
- `show_dynamic_params_in_xml` - 動的 SQL 文に関する情報を表示します。

`@@plwpid` により提供される値を `show_dynamic_params_in_xml object_id` パラメータの値として使用することで、クエリ内の動的パラメータに関する情報が表示されます。最適なクエリプランが得られるパラメータを確認するまで、クエリプランの調整を続けます。

『パフォーマンス&チューニングシリーズ: クエリ処理と抽象プラン』1の「クエリ最適化方式と見積もりの表示」を参照してください。

## ロックタイムアウトのモニタ

---

Adaptive Server バージョン 15.7 では、ロックタイムアウトをモニタできます。

Adaptive Server バージョン 15.7 では、ロックのトラッキングに関する情報が追加されています。

- `monLockTimeout` モニタリングテーブルは、ロックタイプ、所有者、ロックステータスなど、ロックタイムアウト要求に関する情報を提供します。『リファレンスマニュアル：テーブル』を参照してください。
- これらの設定パラメータにより、Adaptive Server でロック待機タイムアウト情報を収集し、`monLockTimeout` テーブルで使用できるようになります。
  - `lock timeout pipe active`
  - `lock timeout pipe max messages`

『システム管理ガイド 第1巻』の「設定パラメータ」を参照してください。

## 後続の 0 のトランケート

---

Adaptive Server バージョン 15.7 では、varbinary および binary の null データからの後続の 0 のトランケーションを有効または無効にするための **disable varbinary truncation** 設定パラメータが追加されています。

デフォルトでは **disable varbinary truncation** がサーバでオフになっています。

『Transact-SQL ユーザーズガイド』の「データの追加、変更、転送、削除」を参照してください。

## 完全にリカバリ可能な DDL

---

Adaptive Server バージョン 15.7 では、以前のバージョンの Adaptive Server で最低限のログが取られていた操作を完全にリカバリするために、**dump transaction** を使用することができます。

以下の操作は **dump transaction** でリカバリすることができます。

- **select into**
- データの移動が必要な **alter table** コマンド
- **reorg rebuild**

デフォルトで最低限のログが取られるコマンドのフルログを取るには、master データベースで **sp\_dboption** を実行します。

『リファレンスマニュアル: コマンド』を参照してください。

## merge によるソーステーブルからターゲットテーブルへのローの転送

---

Adaptive Server 15.7 では、ソーステーブルからターゲットテーブルにローを転送するための **merge** コマンドが追加されています。

『リファレンスマニュアル: コマンド』を参照してください。

## sp\_showoptstats による統計とヒストグラムの表示

`sp_showoptstats` により、`systabstats` や `sysstatistics` などのシステムテーブルからさまざまなタイプのデータオブジェクトの統計およびヒストグラムを XML ドキュメントで抽出して表示できます。

『リファレンスマニュアル：プロシージャ』および『パフォーマンス&チューニングシリーズ：統計的分析によるパフォーマンスの向上』の「統計テーブルおよび `optdiag` を使った統計の表示」を参照してください。

## カーソルの変更

Adaptive Server バージョン 15.7 では、カーソルロック、カーソルのトランザクションの管理方法、カーソル文の宣言方法が変更されています。

## カーソルクローズ時のカーソルロックの解放

Adaptive Server 15.7 では、トランザクションがアクティブでもカーソルをクローズしたときに、独立性レベル 2 と 3 でカーソルロックを解放するための `declare cursor ... release_locks_on_close` オプションが追加されています。

『リファレンスマニュアル：コマンド』および『Transact-SQL ユーザーズガイド』の「カーソル：データのアクセス」を参照してください。

## カーソルの拡張トランザクションサポート

Adaptive Server 15.7 以降では、カーソルによるトランザクションのサポート方法が変更されています。

- トランザクションをコミットする場合、`for update` 句で宣言されたオープンカーソルを自動的にクローズしません。トランザクションのコミット時に読み込み専用カーソルをクローズするには、`close on end tran` オプションを設定します。
- トランザクションのコミット後のオープンカーソルに対する `fetch` 操作をサポートしています。

『リファレンスマニュアル：コマンド』および『Transact-SQL ユーザーズガイド』の「カーソル：データのアクセス」を参照してください。

## カーソル文のモニタ

Adaptive Serverバージョン15.7では、monCachedStatement モニタリングテーブルからカーソル文に基づく情報をモニタします。

たとえば、new\_cursor カーソル(「sq0267364184\_1108036110ss」によって指定)を宣言しても、**sp\_cursorinfo** にはそのプランが表示されません。

```
declare new_cursor cursor for select id from sysroles
go
sp_cursorinfo
go
Cursor name 'new_cursor' is declared on procedure
'*sq0267364184_1108036110ss*'
The cursor is declared as NON-SCROLLABLE cursor.
The cursor id is 983044.
The cursor has been successfully opened 0 times.
The cursor will remain open when a transaction is committed or rolled
back.
The number of rows returned for each FETCH is 1.
The cursor is updatable.
This cursor is declared on a stored procedure. It is presently using
'860'
bytes. However, the memory usage will increase when the cursor is
opened because
the query plan will be associated with the cursor at that time.
```

Adaptive Server はカーソルを開いたときにコンパイルします。

カーソルのモニタ文を有効または無効にするには、**enable functionality group** 設定パラメータを使用します。

『Transact-SQL ユーザーズガイド』の「カーソル: データのアクセス」を参照してください。

## ネストされた **select** 文の機能強化

Adaptive Server 15.7 では、アスタリスク (\*) の機能が拡張されています。

Adaptive Server 15.7 以降では、アスタリスクが次の条件である限り、**exists** サブクエリではないネストされた **select** 文にアスタリスクを使用できます。

- **select** 文の唯一の項目である
- ネストされたクエリの1つのテーブルカラムを解決する

さらに、次の操作ができます。



- *qualifier*\* フォーマット (ここで、*qualifier* は from 句内の 1 つのテーブル) を使用して、ネストされたクエリ内で選択するカラムを特定のテーブルに属するカラムのみに制限できます。
- group by 句を含むネストされたクエリにアスタリスクを使用できます。

アスタリスクがネストされたクエリの 1 つのテーブルカラムに解決される場合、このクエリは 1 つのテーブルカラムを明示的に使用する場合と同様になります。

『Transact-SQL ユーザーズガイド』の「クエリ：テーブルからのデータの選択」を参照してください。

## 連鎖トランザクションのコマンドとシステムプロシージャの変更

---

Adaptive Server バージョン 15.7 では、連鎖トランザクションモードを使用するセッションで一部のシステムプロシージャを実行できます。

- 次のシステムプロシージャは、オープントランザクションが存在しない場合に連鎖トランザクションモードを使用するセッションで実行できます。
  - **sp\_configure**
  - **sp\_engine**
  - **sp\_rename**
- 次のシステムプロシージャは、**sp\_procxmode** を使用してトランザクションモードを **anymode** に変更した後に、連鎖トランザクションを使用するセッションで実行できます。
  - **sp\_addengine**
  - **sp\_dropengine**
  - **sp\_showplan**
  - **sp\_sjobcontrol**
  - **sp\_sjobcmd**
  - **sp\_sjobcreate**
- **sp\_sjobdrop** は連鎖トランザクションモードを使用するセッションで実行できますが、オープントランザクション中に実行した場合は失敗します。

これらのストアドプロシージャを実行すると、オープントランザクションが存在しない場合にこれらのストアドプロシージャにより実行された変更が明示的にコミットされるため、**commit** または **rollback** を発行する必要があります。

以下を発行したときにオープントランザクションが存在すると、次のようになります。

- **sp\_rename**、**sp\_configure**、**sp\_engine**、**sp\_addengine**、または **sp\_dropengine** - これらのプロシージャはトランザクション内で実行できないため、エラー 17260 で失敗します。
- **sp\_sjobcontrol**、**sp\_sjobcmd**、**sp\_sjobcreate**、**sp\_sjobdrop**、または **sp\_showplan** - プロシージャの実行後トランザクションが開いたままになります。トランザクション全体に対して **commit** または **rollback** を明示的に発行する必要があります。これらのプロシージャの実行時にエラーが出されると、プロシージャ内で実行された操作のみにロールバックしますが、同じトランザクション内で操作が実行されている場合でも、実行前に行われた操作にロールバックすることはありません。

**set chained {on | off}** を使用してセッションの連鎖モードを設定します。

『リファレンスマニュアル: コマンド』および『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。

## 可変長のローの拡張

---

Adaptive Server バージョン 15.7 では、32767 バイトまでのローオフセットを使用するようにデータオンリーロック (DOL) カラムが再定義されています。長い可変長の DOL ローを作成するには、Adaptive Server を論理ページサイズ 16K に設定する必要があります。

Adaptive Server のデフォルトでは、長い可変長の DOL ローを使用しません。各データベースに対して長い可変長の DOL ローを有効にするには、以下を使用します。

```
sp_dboption database_name, 'allow wide dol rows', true
```

『パフォーマンス&チューニングシリーズ：物理データベースのチューニング』の「データの格納」を参照してください。

## like パターン一致の変更

---

Adaptive Server バージョン 15.7 では、**like** パターン一致アルゴリズム内の角カッコを個別に処理することができます。

たとえば、以前のバージョンの Adaptive Server で、`'[XX]'` のローを一致させるには、次を使用しました。

```
select * from t1 where f1 like '[][]XX[]'
```

Adaptive Server バージョン 15.7 では、以下も使用できます。

```
select * from t1 where f1 like '[][]XX'
```

## 引用符付き識別子の変更

---

Adaptive Server 15.7以降では、テーブル、ビュー、カラム名、インデックス名、システムプロシージャパラメータに引用符付き識別子を使用できます。

15.7より前のバージョンの Adaptive Server では、二重引用符(引用符付き識別子)または角カッコで区切られた非英数字を使用する識別子として「ident」を処理していました。これらの識別子は、テーブル、ビュー、カラムの名前のみに使用できます。

『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。

## Unicode 非文字の許可

---

Adaptive Server バージョン 15.7 では、**enable functionality group** に属する **enable permissive unicode** 設定パラメータで Unicode 非文字を無視することができます。

この機能を有効にすると、次の内部の Unicode 非文字が検出されなくなります。

- パラメータ
  - univarchar および unitext (UTF-16) データ型で示されるもの
  - varchar および text (UTF-8) データ型で示されるもの
  - 動的 SQL 文のパラメータ
  - パラメータ化した言語の文の入力
  - パラメータ化した言語の文の入力
- サーバの文字セットが UTF-8 の場合の文字列リテラル
- サーバの文字セットに関係なく、エスケープされた文字列リテラル(プレフィクスが U& のもの)
- unichar (UTF-16) と varchar (UTF-8) 間の変換プロセス (両方向)

さらに、Unicode 非文字は、有効な Unicode 文字よりも前にソートされる比較などの単純な式に使用することができます。

15.7 よりも前のバージョンの Adaptive Server では、utf-8 デフォルト文字セット内の unichar、univarchar、unitext、char、varchar、text データ型は Unicode 非文字を受け入れませんでした(コードポイントは永久に内部使用のために予約されています)。

『システム管理ガイド 第1巻』の「設定パラメータ」および「クライアント/サーバの文字セット変換の設定」を参照してください。

## クエリ処理遅延時間の短縮

---

Adaptive Server 15.7 のクエリ処理層では、複数のクライアント接続に動的 SQL ライトウェイトプロシージャ (LWP) を再利用または共有させることが可能です。

Adaptive Server では、ステートメントキャッシュを使用して、LWP に変換された動的 SQL 文を保管します。ステートメントキャッシュはすべての接続の間で共有されるため、接続の間で動的 SQL 文を再利用することができます。次の文はキャッシュされません。

- **select into** 文
- すべてのリテラル値を持ち、パラメータのない insert-values 文
- テーブルを参照しないクエリ
- 次の例のように、複数の SQL 文を含む個別に準備された文
- **instead-of** トリガを呼び出す文

**streamlined dynamic SQL** または **enable functionality group** 設定パラメータを使用してこの機能を有効にします。

『パフォーマンス&チューニングシリーズ：基本』の「メモリの使い方とパフォーマンス」および『システム管理ガイド 第1巻』の「設定パラメータ」を参照してください。

## sybdiag ユーティリティ

---

Adaptive Server 15.7 では、包括的な Adaptive Server 設定および環境データを収集する Java ベースのツール、**sybdiag** ユーティリティが追加されています。Sybase サポートセンタはこの情報を使用してサーバの問題を診断することにより、迅速なトラブルシューティングを行えるようにしています。

**sybdiag** は Adaptive Server に接続して **sp\_configure** のようなシステムプロシージャを実行し、**monLicense** のようなテーブルにクエリを送ります。**ps**、**vmstat**、**netstat** のようなコマンドを実行することにより、オペレーティングシステムとプラットフォームの診断情報を収集します。

**sybdiag** からは、HTML ファイルとデータファイルで構成される .zip 出力ファイルが生成されます。このファイルを解凍して Web ブラウザで表示することができます。収集される情報には、オペレーティングシステムと環境データ、Adaptive Server の設定とモニタリングデータ、Adaptive Server ファイルとスクリプトが含まれます。

sybdiag では、ログイン、パスワード、ユーザリストに関する Adaptive Server データやオペレーティングシステムデータに加え、アプリケーションデータベーステーブルからの情報を収集しません。

『ユーティリティガイド』を参照してください。

## オプティマイザの診断ユーティリティ

---

Adaptive Server バージョン 15.7 には、Adaptive Server オプティマイザにより生成されるクエリプランとクエリプランの選択に影響を与える要因を分析するための **sp\_opt\_querystats** システムプロシージャが追加されています。

この分析は、クエリ内の要素または実行環境が Adaptive Server によるクエリの実行方法とパフォーマンスにどのような影響を与えるかを確認するために役立ちます。分析を実行するために、選択したクエリを実行する必要はありません。

**sp\_opt\_querystats** により次の情報が提供されます。

- showplan によって生成されたクエリプラン
- 有効なトレースフラグとスイッチ
- set statistics io によって生成されたクエリの I/O アクティビティ
- クエリに含まれるテーブルに対して検出された欠落統計
- オプティマイザによって計算されたプランコストの見積もり
- 最終的なプランとオプティマイザによって計算されたコストの見積もり
- クエリの抽象プラン
- 結果セットが実行されている場合のクエリの結果 (たとえば、noexec がオンでない場合)
- set オプションの show によって生成されたプランの論理演算子ツリー
- **set statistics time** によって生成されたクエリの実行時間
- クエリの実行後、set statistics time によって生成されたクエリの実行時間

Job Scheduler をインストールして **sp\_opt\_querystats** を実行するように設定する必要があります。

『パフォーマンス&チューニングシリーズ：クエリ処理と抽象プラン』の「最適化の制御」を参照してください。

## Adaptive Server バージョン 15.7 でのシステムの変更点

Adaptive Server 15.7 では、コマンド、関数、システムプロシージャ、設定パラメータ、システムテーブル、モニタリングテーブル、グローバル変数に変更されています。

### コマンド

Adaptive Server 15.7 では、コマンドが追加および変更されています。

表 16 : 新しいコマンド

コマンド	説明
<b>alter login</b>	ログインアカウントの属性を変更する
<b>alter login profile</b>	ログインプロファイルの属性を変更する
<b>alter...modify owner</b>	データベースオブジェクトの所有権を 1 人の所有者から別の所有者に転送する
<b>alter thread pool</b>	スレッドプールを変更する
<b>create login</b>	ログインアカウントを作成して、アカウントに割り当てるパスワード、アカウントのログインプロファイル、ユーザ提供のパラメータを指定する
<b>create login profile</b>	指定した属性でログインプロファイルを作成する
<b>create thread pool</b>	ユーザ定義のスレッドプールを作成する
<b>deallocate locator</b>	メモリに格納されているラージオブジェクト (LOB) を削除し、その LOB ロケータを無効にする
<b>drop login</b>	1 つまたは複数のログインアカウントを削除する
<b>drop login profile</b>	1 つまたは複数のログインプロファイルを削除する
<b>drop thread pool</b>	ユーザ定義のプールを削除する
<b>merge</b>	送信元テーブルからターゲットテーブルにローを転送する
<b>select for update</b>	同じトランザクション内で後続の更新のためにローを排他的にロックする
<b>truncate lob</b>	LOB を指定の長さにトランケートする

表 17 : 変更されたコマンド

コマンド	変更点
<b>alter database</b> の変更点	<ul style="list-style-type: none"> <li>• データベースレベルで圧縮設定を変更できるようにする</li> <li>• <b>alter database .. inrow_LOB_length</b> - データベース全体でロー内の LOB カラム長を変更できるようにする</li> <li>• <b>alter database ... log off</b> - データベースログの不要な部分を削除し、データベースを再作成しなくてもログ領域を縮小して記憶領域を解放できるようにする</li> </ul>
<b>alter encryption key</b>	<ul style="list-style-type: none"> <li>• <b>master</b> および <b>dual master</b> - マスタまたはデュアルマスタの暗号化キーを変更していることを示す</li> <li>• <b>master key</b> - マスタキーで暗号化キーを変更していることを示す</li> <li>• <b>[no] dual_control</b> - デュアルコントロールを使用して新しいキーを暗号化するかどうかを示す</li> <li>• <b>for recovery</b> - パスワードを紛失した場合に、キーコピーを使用してマスタキーをリカバリすることを示す</li> <li>• <b>for automatic_startup</b> - サーバの起動後にキーコピーを使用してマスタキーまたはデュアルマスタキーにアクセスすることを示す</li> <li>• <b>regenerate key</b> - マスタキーまたはデュアルマスタキーのローキー値を新しいローキーに置き換え、マスタキーまたはデュアルマスタキーで暗号化されたすべてのカラム暗号化キーを再暗号化する</li> </ul>
<b>alter table</b>	<ul style="list-style-type: none"> <li>• テーブル、カラム、パーティションの圧縮属性を変更できるようにする</li> <li>• <b>alter table ... not materialized</b> - マテリアライズされていないカラムを作成していることを示す</li> <li>• <b>alter table ... add lob-colname</b> - 新しく追加した null 入力可能な LOB カラムをロー内として定義し、その長さを指定する</li> <li>• <b>alter table ... modify lob-colname</b> - 既存の LOB カラムをロー外からロー内に変更できるようにする</li> <li>• <b>alter table ... modify off row   in row</b> - Java-SQL カラムが、ローから独立して格納されるか、ローに直接割り付けられた記憶領域に格納されるかを指定する</li> </ul>
連結演算子	<p>+ および    Transact-SQL 演算子は、LOB ロケータを連結演算用の式として受け入れる。1 つまたは複数のロケータを伴う連結演算の結果は、入力ロケータにより参照されるロケータと同じデータ型の新しい LOB ロケータになる。</p>

コマンド	変更点
<b>create data-base</b>	<ul style="list-style-type: none"> <li>• <b>compression =</b> - 新しく作成したテーブルまたはパーティションに適用する圧縮レベルを示す。</li> <li>• <b>lob_compression = value</b> - 新しく作成したテーブルの圧縮レベルを決定する。off を選択した場合、テーブルに LOB 圧縮を使用しないことを意味する。</li> <li>• <b>inrow_lob_length = value</b> - バイト数を指定する。inrow_lob_length の有効値の範囲は 0 からデータベースの論理ページサイズまで。</li> </ul>
<b>create encryption key</b>	<ul style="list-style-type: none"> <li>• <b>master</b> および <b>dual master</b> - マスタまたはデュアルマスタの暗号化キーを作成していることを示す</li> <li>• <b>passwd system_encr_passwd   master key</b> - システム暗号化パスワードまたはパスワードのマスタキーを使用していることを示す</li> <li>• <b>[no] dual_control</b> - デュアルコントロールを使用して新しいキーを暗号化するかどうかを示す</li> </ul>
<b>create table ... [ in row [(length)]   off row ]</b>	<p>圧縮テーブルを作成できるようにする。</p> <p><b>create table</b> により、LOB カラムのデータをロー外に格納するのではなく、ロー内に保つことを指定できる。</p>
<b>declare cursor ... [release_locks_on_close]</b>	各カーソルのロック解放動作を設定して、トランザクションがアクティブでもカーソルをクローズしたときに共有ロックを解放できるようにする
<b>drop encryption key</b>	<b>[dual] master</b> - マスタキーまたはデュアルマスタキーを削除していることを示す
<b>dump data-base ... with shrink_log</b>	データベースがダンプシーケンス中かどうかにかかわらず、データベースの最後にある空白を削除できるようにする
<b>where</b> 句内の <b>like</b> 句	<b>where</b> 句は、 <i>variables</i> 式および <i>match_string</i> に対して text および unitext の LOB ロケータを受け入れるが、image の LOB ロケータは受け入れない。
<b>select into ... [in row [(length)]   off row ]</b>	ターゲットテーブルに text カラムのロー内特性を設定するか変更する。長さを指定しない場合、設定されているデフォルトのロー内の長さを使用される。



コマンド	変更点
set	<p>set は以下を追加する。</p> <ul style="list-style-type: none"> <li>• <b>send_locator [on   off]</b> - クライアントに送信する結果セット内に LOB または LOB を参照するロケータを送るかどうかを指定する</li> <li>• <b>cis_rpc_handling {on   off}</b> - CIS のリモートプロシージャコール (RPC) 処理メカニズムを共有ディスククラスタ (SDC) 処理のデフォルトのメカニズムにする</li> <li>• <b>encryption passwd &lt;char_literal&gt; for key [dual] master</b> - マスタキーまたはデュアルマスタキーのパスワードを設定する</li> </ul>
LOB をサポートする where 句拡張機能	select、insert、update、delete 文の where 句には null のラージオブジェクト (LOB) の条件を含めることができる

『リファレンスマニュアル: コマンド』を参照してください。

## 関数

Adaptive Server 15.7 では、関数が追加および変更されています。

表 18 : 新しい関数

関数	説明
<b>dol_downgrade_check</b>	8191 バイトよりも長い可変長カラムを含む指定データベース内のデータオンリーロック (DOL) テーブルの数をリターンする
<b>create_locator</b>	指定したラージオブジェクト (LOB) のロケータを明示的に作成する
<b>locator_literal</b>	バイナリ値をロケータのリテラルとして表す
<b>locator_valid</b>	LOB ロケータが有効かどうかを確認する
<b>lprofile_id</b>	指定したログインプロファイル名のログインプロファイル ID、または現在のログインか指定したログイン名に関連付けられたログインプロファイルのログインプロファイル ID をリターンする
<b>lprofile_name</b>	指定したログインプロファイル ID のログインプロファイル名、または現在のログインかログイン suid に関連付けられたログインプロファイルのログインプロファイル名をリターンする
<b>return_lob</b>	ロケータの参照を解除し、そのロケータによって参照されていた LOB をリターンする

関数	説明
<code>setdata</code>	一部またはすべての LOB を上書きする
<code>show_cached_plan_in_xml</code>	ステートメントキャッシュ内のクエリの実行クエリプランを XML で表示する
<code>show_dynamic_params_in_xml</code>	クエリのテキストを XML フォーマットでリターンする

表 19 : 変更された関数

関数	説明
<code>charindex</code>	<code>charindex</code> は <code>text_locator</code> 、 <code>unitext_locator</code> および <code>image_locator</code> LOB ロケータデータ型と <code>start</code> オプションのサポートを追加する。
<code>charlength</code>	<code>charlength</code> は <code>text_locator</code> および <code>unitext_locator</code> データ型をサポートする。
<code>datalength</code>	<code>datalength</code> は <code>text_locator</code> 、 <code>unitext_locator</code> および <code>image_locator</code> データ型を受け入れる。
<code>patindex</code>	<code>patindex</code> は <code>text_locator</code> および <code>unitext_locator</code> データ型を受け入れる。
<code>show_cached_plan_in_xml</code>	<code>show_cached_plan_in_xml</code> は、ステートメントキャッシュ内のものだけでなく、ライトウェイトプロシージャを参照するオブジェクト ID を受け入れるように <code>statement_id</code> パラメータの範囲を拡大する。
<code>str</code>	<code>str</code> 関数の 10 進パラメータは、文字または数値による指定した長さまでの出力の埋め込みをサポートするように拡張されている。
<code>textptr</code>	<p>ロー内/ロー外の LOB 機能は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のデータページを分割または縮小し、データロー (これらのデータロー内に格納されたロー内の LOB カラムを含む) が別のページに移動することがあるため、このようなロー内の LOB カラムの <code>textptr</code> テキストポインタ値が分割または縮小操作の前後で異なる。</p> <p>Adaptive Server 15.7 以降では、ローを転送したデータオンリーロックのデータロー内に格納されたロー内の LOB カラムに対してリターンされる <code>textptr</code> 値が転送後も有効になる。</p>

関数	説明
<b>textvalid</b>	ロー内とロー外の両方の LOB に対してリターンされるテキストポインタに <b>textvalid</b> を使用することができ、テキストポインタが有効な LOB カラムを指す場合は 1、LOB カラムが無効な場合は 0 がリターンされる。

『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。

## システムストアプロシージャ

Adaptive Server 15.7 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 20 : 新しいシステムストアプロシージャ

システムストアプロシージャ	説明
<b>sp_merge_dup_inline_default</b>	既存の重複インラインデフォルトオブジェクトを削除し、ユニークなインラインデフォルトを共有可能なインラインデフォルトオブジェクトに変換する
<b>sp_opt_querystats</b>	選択したクエリのパフォーマンス分析をリターンする
<b>sp_securityprofile</b>	ログインプロファイルに関連付けられた属性またはバインドをリストする
<b>sp_showoptstats</b>	systabstats や sysstatistics などのシステムテーブルからさまざまなデータオブジェクトの統計およびヒストグラムを抽出して表示する

表 21 : 変更されたシステムストアプロシージャ

システムストアプロシージャ	説明
<b>sp_dboption</b>	<ul style="list-style-type: none"> <li>• <b>enforce dump tran sequence</b> - 後続のダンプトランザクションを禁止するオペレーションを防止する</li> <li>• <b>allow wide rows</b> - 長い可変長のデータオンリーロック (DOL) ローを許可するようにデータベースを設定する</li> <li>• <b>full logging for all</b> - デフォルトで最低限のログを取るコマンド (<b>select into</b>、<b>alter table</b>、<b>reorg rebuild</b>) のフルログコマンド。</li> </ul>

システムストア ドプロシージャ	説明
<b>sp_displaylogin</b>	<p>以下を表示する。</p> <ul style="list-style-type: none"> <li>ログインアカウントに関連付けられたログインプロファイル名。</li> <li>ログインアカウントに直接関連付けられたログインプロファイルがなくても、デフォルトログインプロファイルが存在する場合のデフォルトログインプロファイル名。</li> </ul> <p>ログインは <b>sp_addlogin</b>、<b>sp_modifylogin default database</b>、<b>default language</b>、<b>authenticate with</b> および <b>login script</b> パラメータを上書きする。</p> <p>ログインプロファイルが無視される場合、またはログインアカウントに直接関連付けられるか、デフォルトログインプロファイルにより関連付けられたログインプロファイルがない場合、<b>sp_displaylogin</b> は 15.7 より前のバージョンのフォーマットで情報を表示する。</p>
<b>sp_displayroles</b>	<ul style="list-style-type: none"> <li>現在のログインに対して実行した場合、<b>sp_displayroles</b> は関連付けられているログインプロファイルに付与された役割を表示する。<b>sp_displayroles</b> で他のログインプロファイルに関連付けられた役割を表示するには、<b>sso_role</b> が必要になる。</li> <li>関連付けられたログインプロファイルによりログインに付与された役割を表示する。出力の <b>Grantee</b> カラムには該当するログインプロファイル名が示される。ログインに役割が付与された関連ログインプロファイルがある場合のみ、<b>sp_displayroles</b> で <b>Grantee</b> カラムが表示される。</li> <li>役割がロックされた日付、ロックの理由、役割をロックしたログイン ID を表示する。パスワードで保護されている役割の場合、<b>sp_displayroles</b> では、役割のパスワードの暗号化バージョンが表示される。</li> </ul>

システムストア ドプロシージャ	説明
sp_encryption	<ul style="list-style-type: none"> <li>SSO、キー管理者、または DBO が実行した場合、キーがデュアルコントロールにより保護されていることがレポートされる。</li> <li><b>sp_encryption helpkey, master</b> および <b>sp_encryption helpkey, 'dual master'</b> は、自動起動用に暗号化されたコピーの存在やリカバリコピーの存在を含む、マスタキーとデュアルマスタキーの情報をレポートする。</li> <li><b>mkey_startup_file</b> [, {&lt;new_path&gt;   default_location   null} [, {sync_with_mem   sync_with_qrm}]] - マスタキーのスタートアップファイル名とパスを表示または設定する。</li> <li><b>downgrade_kek_size</b> [, {"true"   "false"}] - サーバの <b>downgrade_kek_size</b> 設定を表示または設定する。</li> </ul>
sp_help	カラム、テーブル、パーティションレベルで圧縮設定を表示する。カラムおよびテーブルレベルでロー内の LOB 設定を表示する。
sp_helpconstraint	共有可能なインラインデフォルトに関する情報を表示するよう更新される
sp_helprotect	<ul style="list-style-type: none"> <li>「master key」 および 「dual master key」 をオブジェクト名として受け入れる。</li> <li>「Set Encryption Passwd」 を有効なパーミッション名として受け入れる。</li> <li>デュアルマスタキーとマスタキーのパーミッションを表示する。</li> </ul>
sp_helpuser	<i>display_object</i> は、現在のデータベースの <i>name_in_db</i> により所有されているすべてのオブジェクトとユーザ定義データ型をリストする。
sp_locklogin	除外されたログインアカウントが非アクティブであるためロックされなくなる。
sp_passwordpolicy	<ul style="list-style-type: none"> <li><b>keypair regeneration period</b> - 最初のキーペア生成を開始する日付と時刻、および後続のキーペア再生成頻度を指定する。</li> <li><b>keypair error retry wait/count</b> - キーペアの生成が失敗した後に、キーペアの再生成に設定できるさまざまな設定を指定する。</li> </ul>

システムストア ドプロシージャ	説明
<b>sp_serveroption</b>	次のオプションの定義を変更する。 <ul style="list-style-type: none"> <li>• use message confidentiality - すべての接続のメッセージの機密性を Kerberos 認証を使用したりリモートサーバに設定する。</li> <li>• use message integrity - すべての接続のメッセージの整合性を Kerberos 認証を使用したりリモートサーバに設定する。</li> </ul>

『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。

## 設定パラメータ

Adaptive Server 15.7 では、新しい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
<b>automatic master key access</b>	Adaptive Server が無人起動モードで動作することを決定する
<b>capture compression statistics</b>	monTableCompression モニタリングテーブルが圧縮統計の取得を開始できるように設定する
<b>column default cache size</b>	マテリアライズされていないカラム用のデフォルトを提供するために Adaptive Server がメモリ内に確保する必要があるキャッシュサイズを決定する
<b>compression info pool size</b>	圧縮に使用するメモリプールのサイズを決定する
<b>disable varbinary truncation</b>	varbinary または binary の null データの最後に後続のゼロを含めるかどうかを制御する
<b>enable console logging</b>	有効にすると、起動後にエラーログからコンソールに個別にメッセージが送信される

設定パラメータ	説明
<b>enable functionality group</b>	Adaptive Server バージョン 15.7 以降で以下の機能を有効または無効にする。 <ul style="list-style-type: none"> <li>共有可能なインラインデフォルト</li> <li>更新に選択</li> <li>引用符付き識別子</li> <li>Unicode 非文字</li> <li>カーソル文のモニタ</li> <li>クエリ処理遅延時間の短縮</li> </ul>
<b>enable hp posix async i/o</b>	HP-UX 11.31 以降で非同期 I/O を有効にする
<b>kernel mode</b>	Adaptive Server カーネルがスレッドまたはプロセスのどちらのモードを使用するかを決定する
<b>kernel resource memory</b>	すべてのスレッドプールと他のカーネルリソースを割り当てるカーネルリソースメモリプールのサイズを 2K ページで決定する
<b>lock timeout pipe active</b>	Adaptive Server でロックタイムアウトメッセージを収集するかどうかを制御する
<b>lock timeout pipe max messages</b>	Adaptive Server が保存するロックタイムアウトメッセージの数、およびタスクに割り当てるメモリ量を決定する
<b>number of disk tasks</b>	ポーリングおよびディスク I/O の完了のための専用タスクの数を制御する
<b>number of network tasks</b>	ポーリングおよびネットワーク I/O の完了のための専用タスクの数を制御する

詳細については、『システム管理ガイド 第 1 巻』を参照してください。

## システムテーブル

Adaptive Server 15.7 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 22 : 変更されたシステムテーブル

システムテーブル	追加カラム	説明
sysattributes	<ul style="list-style-type: none"> <li>object_cinfo2</li> <li>object_datetime</li> </ul>	<ul style="list-style-type: none"> <li>オブジェクトの説明を提供する</li> <li>オブジェクトの日付および時刻を提供する</li> <li><i>sysattributes</i> は、RSA キーペア再生成に関連するオプションを格納する <b>SP</b> <i>object_type</i> とログインプロファイルに関連するオプションを格納する LR <i>object_type</i> を追加する</li> </ul>
sysdatabases		データベース全体の圧縮設定を示すためにステータスビットを <i>status4</i> カラムに追加する
sysoptions	number	スイッチ ID を整数としてリストする
sysobjects	lobcomp_lvl	<ul style="list-style-type: none"> <li>データベース全体の圧縮設定を示すためにステータスビットを <i>status3</i> カラムに追加する</li> <li><i>lobcomp_lvl</i> - ラージオブジェクトに定義されたカラムの圧縮レベル</li> </ul>



システムテーブル	追加カラム	説明
syscolumns	inrowlen	<ul style="list-style-type: none"> <li>カラムが圧縮するように明示的に定義されているかどうかを示すためにステータスビットを status2 カラムに追加する</li> <li>inrowlen - ロー内に作成された LOB カラムのユーザー指定のロー内の長さまたは抽出されたロー内の長さを格納する null 入力可能なカラム</li> <li>lobcomp_lvl - ラージオブジェクトに定義されたカラムの圧縮レベル</li> </ul>
syslogins	crsuid	ログインまたはログインプロファイルの作成者のサーバユーザ ID
syslogins	lpid	ログインプロファイル ID
sysssrvroles	<ul style="list-style-type: none"> <li>lockdate</li> <li>lockreason</li> <li>locksuid</li> </ul>	<ul style="list-style-type: none"> <li>役割がロックされた日付と時刻</li> <li>役割がロックされた理由</li> <li>役割をロックしたユーザの ID</li> </ul>
syssservers	srvprincipal	リモートサーバの Kerberos プリンシパル名を指定する
syscomments	<ul style="list-style-type: none"> <li>syb_syscommkey_ dddddd</li> </ul>	

Adaptive Server バージョン 15.7 では、以下のシステムテーブルが追加されています。これらのシステムテーブルは master データベースのビューで、データキャッシュおよびプールの設定に関する情報を提供します。

システムテーブル	説明
syscacheinfo	データキャッシュに関する情報を提供する。
syspoolinfo	キャッシュプールに関する情報を提供する。

システムテーブル	説明
syscachepoolinfo	データキャッシュの設定情報を含む各データキャッシュプールのローを表示する。このビューは <i>syscacheinfo</i> ビューおよび <i>syspoolinfo</i> ビューを統合したビューである。

『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 15.7 では、ユーティリティが追加および変更されています。

### 新しいユーティリティ

コマンド	説明
<b>sybdiag</b>	<b>sybdiag</b> は、包括的な Adaptive Server 設定および環境データを収集する Java ベースのツールである。Sybase サポートセンタはこの情報を使用してサーバの問題を診断することにより、迅速なトラブルシューティングを行えるようにしている。

### 変更されたユーティリティ

コマンド	説明
<b>sybperf</b>	<b>sybperf</b> の 15.7 バージョンでは、Adaptive Server パフォーマンスのモニタに役立つ一連の Adaptive Server カウンタが公開されている。

## モニタリングテーブルの変更

Adaptive Server バージョン 15.7 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。

### 新しいモニタリングテーブル

Adaptive Server バージョン 15.7 では、モニタリングテーブルが追加されています。

コマンド	説明
monDeviceSpaceUsage	データベースデバイスが割り当てられたファイルシステムに関する情報を提供する。空き領域の情報は、ファイルシステムデバイスについてのみ表示される。ファイルシステムサイズおよび空き領域の値は、ローデバイスに割り当てられたデータベースデバイスの場合 null。
monLockTimeout	ロックタイムアウト要求に関する情報を提供する。

『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の変更

Adaptive Server バージョン 15.7 では、一部のモニタリングテーブルが変更されています。

#### *monCachePool* の変更

モニタリングテーブル	説明
LogicalReads	プールから読み込まれたバッファの数
PhysicalWrites	このプールのデータに実行する書き込み操作の回数（1回の書き込み操作に複数ページが含まれる場合がある）
APFReads	このプールにページをロードした APF 読み込み操作の数
APFPercentage	このプールに設定された非同期プリフェッチ制限
WashSize	メモリプールのウォッシュサイズ (キロバイト単位)

#### *monCachedProcedures* の変更

モニタリングテーブル	説明
ExecutionCount	ストアードプロシージャプランまたはツリーがキャッシュされてから Adaptive Server により実行された回数
CPUTime	費やされた合計 CPU 時間 (ミリ秒単位)
ExecutionTime	Adaptive Server でストアードプロシージャプランまたはツリーの実行に費やした合計経過時間 (ミリ秒単位)
PhysicalReads	実行した物理読み込み回数
LogicalReads	プロシージャキャッシュに保持されたこのストアードプロシージャのインスタンスにより実行された論理読み込みの回数
PhysicalWrites	実行した物理書き込み回数
PagesWritten	書き込まれたページ数

#### *monCachedStatement* の変更

モニタリングテーブル	説明
OptimizationGoal	ステートメントキャッシュに格納された最適化目標

モニタリングテーブル	説明
OptimizerLevel	ステートメントキャッシュに格納されたオプティマイザレベル

*monDataCache* の変更

モニタリングテーブル	説明
ステータス	キャッシュのステータス
タイプ	キャッシュのタイプ
CacheSize	キャッシュの合計サイズ (キロバイト単位)
ReplacementStrategy	キャッシュ置換方式
APFReads	このデータキャッシュの非同期プリフェッチ (APF) 読み込み回数
Overhead	キャッシュオーバーヘッド

*monDeadLock* に追加されたカラム

カラム	説明
HeldClientAppl-Name	ロックを保持しているアプリケーションによって設定された <i>clientapplname</i> プロパティの値
HeldClientName	ロックを保持しているアプリケーションによって設定された <i>clientname</i> プロパティの値
HeldClientHost-Name	ロックを保持しているアプリケーションによって設定された <i>clienthostname</i> プロパティの値
HeldHostName	ロックを保持しているクエリを実行したアプリケーションが実行されているホストマシンの名前
HeldNumLocks	spid を保持することで現在保持されているロックの数
HeldProcDBName	デッドロックの発生時にストアードプロシージャがブロッキングプロセスを実行していたデータベースの名前 (該当する場合)
HeldProcedureName	デッドロックの発生時にブロッキングプロセスが実行していたストアードプロシージャの名前 (該当する場合)

カラム	説明
HeldProgramName	ロックを保持しているプロセスを実行しているプログラムの名前
HeldStmtNumber	ロックを保持している SQL 文の SQL バッチ内の文の番号
ObjectDBName	データベースの名前
ObjectID	オブジェクトのユニークな識別子
WaitApplName	ロックを待機しているアプリケーションの名前
WaitBatchID	ロックタイムアウトの発生時にロックを待機していたプロセスによって実行されている SQL バッチの識別子
WaitClientApplName	ロックを待機しているアプリケーションによって設定された <i>clientapplname</i> プロパティの値
WaitClientHostName	ロックを待機しているアプリケーションによって設定された <i>clienthostname</i> プロパティの値
WaitClientName	ロックを待機しているアプリケーションによって設定された <i>clientname</i> プロパティの値
WaitCommand	プロセスのカテゴリ、またはプロセスがブロックされてタイムアウトになったときに実行していたコマンド
WaitContextID	ロックを待機しているプロセスが別のプロセスにブロックされたときのユニークなコンテキスト識別子
WaitHostName	ロックを待機しているプロセスを実行するホストの名前
WaitLineNumber	ロックを待機している SQL バッチまたはストアードプロシージャ内の SQL 文の行番号
WaitProcDBID	ロックを待機しているストアードプロシージャが存在するデータベースのユニークな識別子 (該当する場合)
WaitProcDBName	ロックを待機しているストアードプロシージャが存在するデータベースの名前 (該当する場合)
WaitProcedureID	ロックを待機しているストアードプロシージャの ID (該当する場合)
WaitProcedureName	ロックを待機しているストアードプロシージャの名前 (該当する場合)
WaitProgramName	プロセスを実行しているプログラムの名前

カラム	説明
WaitStmtNumber	ロックを待機している SQL バッチの行番号
WaitTranName	ロックが要求されたトランザクションの名前

### *monErrorLog* の変更

Adaptive Server バージョン 15.7 以降では、`monErrorLog.Severity` カラムのスタックトレースの値が変更されています。以前のバージョンでは、スタックトレースの値として 0 が使用されていました。Adaptive Server バージョン 15.7 では、スタックトレースを表すすべてのローの重大度値は 99 になります。

### *monLockTimeout* の変更

モニタリングテーブル	説明
HeldProgramName	<code>monLockTimeout</code> から削除された。
WaitProgramName	<code>monLockTimeout</code> から削除された。
HeldProcedureID	タイムアウトの発生時にブロッキングプロセスが実行していたストアードプロシージャのユニークなオブジェクト識別子
WaitProcedureID	ロックを待機しているストアードプロシージャのユニークなオブジェクト識別子 (該当する場合)

### *monOpenObjectActivity* の変更

モニタリングテーブル	説明
SharedLockWaitTime	すべてのタスクが共有ロックを待機するために費やした合計時間 (ミリ秒単位)
ExclusiveLockWaitTime	すべてのタスクが排他ロックを待機するために費やした合計時間 (ミリ秒単位)
UpdateLockWaitTime	すべてのタスクが更新ロックを待機するために費やした合計時間 (ミリ秒単位)
ObjectCacheDate	オブジェクトをキャッシュに追加した日付と時刻を示す

*monOpenPartitionActivity* の変更

モニタリングテーブル	説明
ObjectCacheDate	オブジェクトをキャッシュに追加した日付と時刻を示す

*monProcess* の変更

モニタリングテーブル	説明
HostName	プロセスを開始したアプリケーションが実行されているホストマシンの名前
ClientName	アプリケーションによって設定された <i>clientname</i> プロパティの値
ClientHostName	アプリケーションによって設定された <i>clienthostname</i> プロパティの値
ClientApplName	アプリケーションによって設定された <i>clientapplname</i> プロパティの値

*monProcessActivity* の変更

モニタリングテーブル	説明
HostName	クエリを実行したアプリケーションが実行されているホストマシンの名前
Application	アプリケーションの名前
ClientName	アプリケーションによって設定された <i>clientname</i> プロパティの値
ClientHostName	アプリケーションによって設定された <i>clienthostname</i> プロパティの値
ClientApplName	アプリケーションによって設定された <i>clientapplname</i> プロパティの値

*monProcessLookup* の変更

モニタリングテーブル	説明
ClientName	アプリケーションによって設定された <i>clientname</i> プロパティの値

モニタリングテーブル	説明
ClientHostName	アプリケーションによって設定された <i>clienthostname</i> プロパティの値
ClientApplName	アプリケーションによって設定された <i>clientapplname</i> プロパティの値

#### *monProcessProcedures* の変更

モニタリングテーブル	説明
ExecutionCount	プロシージャキャッシュに保持されたストアドプロシージャのこのインスタンスが Adaptive Server により実行された回数
CPUTime	Adaptive Server でプロシージャキャッシュに保持されたこのストアドプロシージャのインスタンスの実行に費やした CPU 時間 (ミリ秒単位)
ExecutionTime	Adaptive Server でプロシージャキャッシュに保持されたこのストアドプロシージャのインスタンスの実行に費やした時間 (ミリ秒単位)
PhysicalReads	プロシージャキャッシュに保持されたこのストアドプロシージャのインスタンスにより実行された物理読み込みの回数
LogicalReads	プロシージャキャッシュに保持されたこのストアドプロシージャのインスタンスにより実行された論理読み込みの回数
PhysicalWrites	プロシージャキャッシュに保持されたこのストアドプロシージャのインスタンスにより実行された物理書き込みの回数
PagesWritten	プロシージャキャッシュに保持されたこのストアドプロシージャのインスタンスにより読み込まれたページ数

#### *monTableColumns* の変更

*monTableColumns* の変更点は次のとおりです。

- Adaptive Server バージョン 15.7 以降では、*monTableColumns* の Description カラムにカラムの測定単位が含まれます。
- *monTableColumn* は、列に表示されるデータの簡単な説明を含む Label カラム (データ型 `varchar (50)`) を追加します。アプリケーションのユーザインタフェースでは、実際のカラム名の代わりにこれらの値を使用できます。



`monTables`、`monTableColumns`、`monWaitEventInfo`、および  
`monWaitClassInfo` の変更

これらのモニタリングテーブルでは、Language カラム(データ型 `varchar(30)`)  
が追加されています。このカラムにより、Adaptive Server で Description カラムおよ  
び `monTableColumns.Label` カラムの値を返す言語を指定できます。

『リファレンスマニュアル：テーブル』を参照してください。

## グローバル変数

Adaptive Server バージョン 15.7 では、グローバル変数が追加されています。

コマンド	説明
@@plwpid	最も最近に準備されたライトウェイトプロシージャのオブジェクト ID をリ ターンする。
@@lwpid	最後から 2 番目に実行されたライトウェイトプロシージャのオブジェクト ID をリターンする。

『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。



## バージョン 15.5 Cluster Edition

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。

	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
<b>Adaptive Server Cluster Edition のオプション</b>				
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
プラットフォームネイティブ Kerberos				Y

	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
<b>Adaptive Server Cluster Edition のオプション</b>				
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 サービス	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。

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 バージョン 15.5 Cluster Edition におけるシステム変更

---

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。



## バージョン 15.5

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 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 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			
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

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
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 サービス	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
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

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
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、複写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が作成したメモリ内テンポラリデータベース
- 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す。**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、次のストアードプロシージャが導入されています。

- **sp\_deletesobj** - 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 1 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ー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ー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ー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) をストアー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ーズガイド』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 ユーザーズガイド』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 バージョン 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る。秒の小数点以下は 6 桁まで格納される。
<i>bigdatetime</i>	<i>bigdatetime</i> 値は、年、月、日、時、分、秒、秒の小数位 6 桁で構成される。

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 23 : 新しい関数

関数	説明
<b>db_attr</b>	指定したデータベースの <b>durability</b> 、 <b>dml_logging</b> 、 <b>template</b> の設定を返す。

関数	説明
<b>object_attr</b>	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する。
<b>cache_usagecache_usage</b>	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す。
<b>current_bigdatetime</b>	現在の日時を表すマイクロ秒の精度の <i>bigdatetime</i> 値を返す。
<b>current_bigtime</b>	現在の時刻を表すマイクロ秒の精度の <i>bigtime</i> 値を返す。

表 24 : 変更された関数

関数	説明
<b>datepart</b>	指定した日付の指定した <i>datepart</i> 引数を整数として生成する。
<b>datename</b>	指定した <i>datepart</i> を文字列として生成する。
<b>datediff</b>	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する。
<b>dateadd</b>	指定した日付または時刻に間隔を追加する。

『リファレンスマニュアル: コマンド』を参照してください。

## システムストアプロシージャ

追加および変更されたシステムストアプロシージャでは、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 25 : 新しいシステムストアプロシージャ

システムストアプロシージャ	説明
<b>sp_deletesmobj</b>	TSM からバックアップオブジェクトを削除する。
<b>sp_querysmobj</b>	TSM からバックアップオブジェクトの一覧を取得する。

表 26 : 変更されたシステムストアプロシージャ

システムストア プロシージャ	説明
<b>sp_addsegment</b>	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た。
<b>sp_addthreshold</b>	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た。
<b>sp_bindcache</b>	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ない。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ない。
<b>sp_cacheconfig</b>	メモリ内記憶域キャッシュの作成、サイズの拡大、または削除を行う。
<b>sp_cachestrategy</b>	<b>prefetch</b> パラメータと <b>MRU</b> 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ない。
<b>sp_dbextend</b>	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ない。
<b>sp_deviceattr</b>	<b>directio</b> 属性と <b>dsync device</b> 属性は、メモリ内デバイスに適用されない。
<b>sp_downgrade</b>	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する。
<b>sp_diskdefault</b>	<b>sp_diskdefault</b> 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ない。
<b>sp_dropdevice</b>	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する。
<b>sp_dropsegment</b>	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た。
<b>sp_droptreshold</b>	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た。
<b>sp_extendsegment</b>	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た。
<b>sp_help</b>	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する。
<b>sp_helpcache</b>	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する。

システムストアド プロシージャ	説明
<b>sp_helpdb</b>	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する。
<b>sp_helpdevice</b>	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する。
<b>sp_modifythreshold</b>	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た。
<b>sp_plan_dbccdb</b>	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> を設定する。
<b>sp_poolconfig</b>	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ない。
<b>sp_post_xpload</b>	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する。
<b>sp_tempdb</b>	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する。
<b>sp_unbindcache、 sp_unbindcache_all</b>	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ない。

『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 27 : 新しいコマンド

コマンド	説明
<b>transfer table</b>	<p>テーブルの増分転送を開始する。</p> <p>新しい <code>grant with grant option</code> では、<b>transfer table</b> がサポートされている。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する。</p> <pre>grant transfer table on table_name to user with grant option</pre>

表 28 : 変更されたコマンド

コマンド	変更点
<b>alter database</b>	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た。
<b>alter table</b>	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た。  <b>transfer table</b> をサポートするための構文が追加された。  set transfer table [on   off]
<b>create database</b>	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た。
<b>create table</b>	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た。  <b>transfer table</b> をサポートするための構文が追加された。  with transfer table [on   off]
<b>disk init</b>	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た。
<b>dump database</b>	Tivoli Storage Manager (TSM) をサポートするための構文が追加された。キーワード <b>syb_tsm</b> は TSM API (libsyb_tsm) との Sybase インタフェースを呼び出す。  database_name to "syb_tsm::object_name"
<b>dump database ... compression=</b>	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た。
<b>dump transaction</b>	TSM をサポートするための構文が追加された。キーワード <b>syb_tsm</b> は TSM API (libsyb_tsm) との Sybase インタフェースを呼び出す。  database_name to "syb_tsm::object_name"

コマンド	変更点
<b>dump transaction ... compression=</b>	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た。
<b>load database</b>	TSM をサポートするための構文が追加された。キーワード <b>syb_tsm</b> は TSM API (libsyb_tsm) との Sybase インタフェースを呼び出す。 <pre>                                database_name from "syb_tsm:: [[-S source_server_name] [-D source_database_name]::]object_name"</pre>
<b>load transaction</b>	TSM をサポートするための構文が追加された。キーワード <b>syb_tsm</b> は TSM API (libsyb_tsm) との Sybase インタフェースを呼び出す。 <pre>                                database_name from "syb_tsm:: [[-S source_server_name] [-D source_database_name]::]object_name"</pre>
<b>select into</b>	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 ログingleベルを指定するための構文が追加された。
<b>set</b>	次の項目が追加されている。 <ul style="list-style-type: none"> <li>• <b>dml_logging</b> パラメータ (セッションに対するログingの量を指定)</li> <li>• <b>deferred_name_resolution</b> (接続レベルで遅延名前解決をアクティブ化)</li> <li>• <b>builtin_date_strings number</b> 値は次のとおり。 <ul style="list-style-type: none"> <li>• 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> 値として解釈する。これがデフォルト値。</li> <li>• 1 - サーバは引数文字列を <i>bigdatetime</i> として解釈する。これは、日付順のシステム関数の結果に影響する。</li> </ul> </li> </ul>

『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
<b>deferred name resolution</b>	遅延名前解決を使用してプロシージャを作成できる。値は次のとおり。 <ul style="list-style-type: none"> <li>0 - 遅延名前解決を無効にする。これがデフォルト値。</li> <li>1 - 遅延名前解決を有効にする。</li> </ul>
<b>builtin date strings</b>	値は次のとおり。 <ul style="list-style-type: none"> <li>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> 型として解釈される。</li> <li>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> 型として解釈される。</li> </ul>

『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する。

『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 29 : 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する。

表 30 : 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する。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る。
<i>sysdatabases</i>	データベースの持続性レベルを示す <i>durability</i> カラムを追加する。 <i>durability</i> カラムでは、 <i>int</i> データ型が使用される。値は次のとおり。 <ul style="list-style-type: none"> <li>• 1 - full</li> <li>• 5 - at_shutdown</li> <li>• 6 - no_recovery</li> </ul>

『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 31 : 新しいユーティリティ

ユーティリティ	説明
<b>openssl</b>	certreq、certauth、certpk12 で実装されたすべての証明書管理タスクを実行する。Sybase では便宜上このバイナリを組み込んでいるが、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わない。詳細については、 <a href="http://www.openssl.org">www.openssl.org</a> を参照。

表 32 : 変更されたユーティリティ

ユーティリティ	変更点
<b>backupserver</b>	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る。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る。

『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 33 : 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	<b>transfer table</b>	136	キーワードまたはオプション：transfer_table
all、 create	<b>alter table</b>	3	<b>alter table</b> に <b>set transfer table on</b> 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  <b>alter table</b> に <b>set transfer table off</b> 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
all、 create	<b>create table</b>	12	<b>create table</b> に <b>with transfer table on</b> 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  <b>create table</b> に <b>with transfer table off</b> 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
all、 create	<b>create database</b>	9	キーワードまたはオプション：inmemory
all、 create	<b>alter database</b>	2	キーワードまたはオプション：inmemory
all、 create	<b>create procedure</b>	11	キーワードまたはオプション：deferred_name_resolution



## バージョン 15.0.3

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、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 バージョン 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
<b>password_random</b>	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する。
<b>pssinfo</b>	プロセスステータス構造体からの情報を返す (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。

### システムストアプロシージャ

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

表 34 : 新しいシステムストアプロシージャ

ストアプロシージャ	説明
<b>sp_tabsuspectptn</b>	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 になる可能性がある。
<b>sp_jreconfig</b>	これは Java スストアプロシージャである。

表 35 : 変更されたシステムストアプロシージャ

プロシージャ	変更点
<b>sp_idadmin</b>	新しいパラメータ <b>set_failback_interval</b> および <b>reinit_descriptors</b> がサポートされている。

プロシージャ	変更点
<b>sp_addserver</b>	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 <b>filter</b> パラメータがサポートされている。
<b>sp_passwordpolicy</b>	<b>validate password options</b> パラメータがサポートされている。
<b>sp_pciconfig</b>	データベース内での Java がサポートされている。
<b>sp_sysmon</b>	追加のカウンタがサポートされている。

『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
<b>create table</b>	このオプションを指定して作成したテーブルは、 <b>BCP IN</b> および <b>'alter table unpartition'</b> オペレーションでのみ使用できる。
<b>update statistics</b>	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する。

『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
<b>enable pci</b>	これは Java 設定パラメータである。
<b>maximum nesting level</b>	最大ネストレベルが 100 に引き上げられている。
<b>mnc_full_index_filter</b>	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する。 <ul style="list-style-type: none"> <li>インデックス内のカラム</li> <li>ヒストグラムがない述語</li> </ul>
<b>pci memory size</b>	これは Java 設定パラメータである。

『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 36 : 新しいモニタリングテーブル

テーブル	説明
<i>monSQLRepActivity</i>	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する。
<i>monSQLRepMisses</i>	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する。 <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す。

表 37 : 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る。
<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る。

『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"> <li>• <i>dbid</i>, <i>int null</i></li> <li>• <i>qptime</i>, <i>datetime null</i></li> <li>• <i>sprocid</i>, <i>int null</i></li> <li>• <i>hashkey2</i>, <i>int null</i></li> <li>• <i>key1</i>, <i>int null</i></li> <li>• <i>key2</i>, <i>int null</i></li> <li>• <i>key3</i>, <i>int null</i></li> <li>• <i>key4</i>, <i>int null</i></li> </ul> <hr/> <p><b>注意：</b>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> <hr/> <p><b>注意：</b>このカラムは、今後の使用に備えて予約されています。</p>
<i>sysprocesses</i>	<p>新しいカラム：<i>clientport</i></p> <ul style="list-style-type: none"> <li>• 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する。</li> <li>• システムプロセスの場合は0を表示する。</li> <li>• データ型は符号なし <i>smallint</i>。</li> </ul>
<i>sys.servers</i>	カラム <i>srvnetname</i> が <i>varchar(32)</i> から <i>varchar(255)</i> に変更されている。

『リファレンスマニュアル: テーブル』を参照してください。

バージョン 15.0.3



## バージョン 15.0.2

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、読み込み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 38 : 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립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) およびアポル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、ストアー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も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も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% の選択性的見積もられました。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* と新しいストアードプロシージャ **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ードプロシージャ

## 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のログを取ることができない場合は、その情報がエラーログにメッセージとして送信されます。

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セスに変換され、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 バージョン 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、繰り返しのために **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 バージョン 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 39 : 変更されたコマンド

コマンド	変更点
<b>disk init、 disk reinit</b>	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と、警告メッセージが表示される。  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ことが推奨される。
<b>create proxy table、 create table at remote server、 alter table</b>	SQL UDF をサポートしていない。
<b>disk init、 disk resize</b>	<i>skip_alloc</i> を true に設定すると、ページは 0 初期化されない。 <i>skip_alloc</i> のデフォルト設定は false。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る。
<b>dump transaction</b>	<b>dump transaction</b> には、 <b>string</b> および <b>char_variable</b> パラメータを含めることができる。ただし、 <b>string</b> および <b>char_variable</b> オプションを指定して、 <b>begin transaction</b> 、 <b>commit</b> 、 <b>connect to</b> 、 <b>declare cursor</b> 、 <b>rollback</b> 、 <b>dbcc</b> 、 <b>use</b> 、またはネストされた <b>execute</b> コマンドを実行することはできない。

『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
<b>set advanced_aggregation</b> 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する。
<b>set switch</b> 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る。

set コマンドの変更点
set パラメータを使用してストアー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る。

## 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
<b>ddlgen</b> 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、 <b>-P</b> 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た。 <b>ddlgen</b> ではパスワードの入力が要求されるようになった。
<b>ddlgen</b> 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	構文の例を次に示す。 <pre>ddlgen -Usa -P -Sserver -TEK -NSampleKeysDB.dbo.ssn_key</pre>
<b>-XOD</b> フラグ指定なしの <b>ddlgen</b>	次の 2 つのことが発生する可能性がある。 <ul style="list-style-type: none"> <li>暗号化キーの作成時にパスワードを指定しないと、<b>ddlgen</b> によりパスワードなしの DDL が生成される。</li> <li>暗号化キーの作成時にパスワードを指定すると、<b>ddlgen</b> によりデフォルトのパスワードが生成される。</li> </ul>
<b>-XOD</b> フラグを指定した <b>ddlgen</b>	システム暗号化パスワードが生成される。
キーコピーのサポート	<b>ddlgen</b> では、キーコピー用とベースキー用の DDL を生成できる。

『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 40 : 新しいシステムストアプロシージャ

システムストアプロシージャ	説明
<b>sp_downgrade</b>	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かどうかを検証する。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る。
<b>sp_spaceusage</b>	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する。推定値は、データローとインデックスローの平均の長さ、およびテーブル内のローの数を使用して計算される。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る。 <b>sp_spaceusage</b> プロシージャでは、 <b>help</b> 、 <b>display</b> 、 <b>archive</b> 、 <b>report</b> 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る。

表 41 : 変更されたシステムストアプロシージャ

システムストアプロシージャ	変更点
<b>sp_autoformat</b>	<i>int</i> ( <i>smallint</i> 、 <i>bigint</i> 、 <i>tinyint</i> 、 <i>unsignedint</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った。
<b>sp_changedbowner</b>	データベースの所有者を変更する。今回、sa_role 権限または sso_role 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った。そのデータベースのスレッシュホルドの所有者も、指定されたユーザに変更される。
<b>sp_checksource</b>	ユーザ定義関数のテキストを暗号化する。
<b>sp_configure</b>	デフォルト値以外の設定を表示するようになった。
<b>sp_depends</b>	ユーザ定義関数のオブジェクトの依存性の有無をチェックする。

システムストア プロシージャ	変更点
<code>sp_deviceattr</code>	ファイルシステム上のデータベースデバイスに対して <code>dsync</code> 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する。
<code>sp_displaylogin</code>	<p>変更点は次のとおり。</p> <ul style="list-style-type: none"> <li>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する。  <code>sp_displaylogin ['user_id'   '[loginame   wildcard]'</code></li> <li><code>user_id</code> - 表示するログイン情報の所有者の ID (<i>suid</i>)</li> <li><code>wildcard</code> - 検索に使用するワイルドカード文字</li> </ul> <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する。  <code>sp_displaylogin '56'</code></p> <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する。  <code>sp_displaylogin 'st%'</code></p>
<code>sp_droplogin</code>	<code>sp_droplogin</code> 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る。
<code>sp_help</code>	ユーザ定義関数に関する情報を表示する。
<code>sp_helpdevice</code>	<code>sp_helpdevice</code> 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る。デバイスタイプは、raw device、block device、または file system device のいずれか。
<code>sp_helprotect</code>	<code>sp_helprotect</code> の新しいオプション <code>permission_name</code> 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) を表示できる。
<code>sp_hidetext</code>	ユーザ定義関数のテキストを暗号化する。
<code>sp_locklogin</code>	<code>sp_locklogin</code> の <code>lock</code> オプションを <code>number of inactive days</code> 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る。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する。

システムストア プロシージャ	変更点
<b>sp_modifylogin account</b>	'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す。
<b>sp_modifystats</b>	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る。
<b>sp_monitorconfig</b>	<b>result_tabl_name</b> 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する。
<b>sp_passwordpolicy csp_passwordpolicy</b>	<ul style="list-style-type: none"> <li>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。</li> <li>追加の構文は次のとおり。  <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", "date- time"     "expire stale role passwords", "date- time"     "maximum failed logins", -1 </pre> </li> </ul>
<b>sp_fixindex</b>	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った。 <b>sp_fixindex</b> 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する (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する)。
<b>sp_sendmsg</b>	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 文字だった。Adaptive Server リリース 15.0.2 では、 <b>sp_sendmsg</b> で送信可能なメッセージの最大長は 4096 文字となっている。
<b>sp_who</b>	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る。



システムストア プロシージャ	変更点
<b>sp_helptext</b>	ユーザ定義関数のテキストをレポートする。 <b>numlines</b> および <b>printops</b> パラメータが導入されている。
<b>sp_ldapadmin</b>	新しいパラメータ <b>set_max_ldapua_desc</b> 、 <b>set_num_retries</b> 、 <b>set_log_interval</b> が導入されている。
<b>sp_monitor</b>	<b>event</b> および <b>help</b> パラメータの機能が強化されている。
<b>sp_tempdb</b>	<b>show</b> および <b>who</b> パラメータが導入されている。

『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 42：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、 <b>sp_options</b> によるクエリの対象となるフェイクテーブルである。カラム名には大文字と小文字の区別がある。

表 43：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った。
<i>syscolumns</i>	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> カラムに追加する。 0x00001000 (4096) - カラムに復号化のデフォルト値が含まれている。
<i>sysobjects</i>	<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る。

システムテーブル	変更点
<i>sysaudits</i>	<p>変更点は次のとおり。</p> <ul style="list-style-type: none"> <li>• <i>Alter Encryption Key</i> 監査イベント名が <i>AEK As/Not Default</i> に変更された。</li> <li>• 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る。 <ul style="list-style-type: none"> <li>• 118 - AEK Modify Encryption</li> <li>• 119 - AEK Add Encryption</li> <li>• 120 - AEK Drop Encryption</li> <li>• 121 - AEK Modify Owner</li> <li>• 122 - AEK Key Recovery</li> </ul> </li> </ul>
<i>sysattributes</i>	<i>monTableColumns</i> の変更点は次のとおり。
<i>sysencryptkeys</i>	<p><i>sysencryptkeys</i> の変更点は次のとおり。</p> <ul style="list-style-type: none"> <li>• 新しい型を次に示す。 <ul style="list-style-type: none"> <li>• <i>EK_KEYCOPY</i> - 0x0010</li> <li>• <i>EK_KEYBASE</i> - 0x0020</li> <li>• <i>EK_RECOVERY</i> - 0x0040</li> </ul> </li> <li>• 新しいステータスビットを次に示す。 <ul style="list-style-type: none"> <li>• <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</li> <li>• <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</li> <li>• <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</li> <li>• <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</li> </ul> </li> <li>• <i>uid</i> カラムの説明が変更された。ユーザアクセスまたはキーリカバリローが該当する。<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>) が含まれる。以前のバージョンの Adaptive Server では、<i>uid</i> カラムは使用されなかった。</li> </ul>

システムテーブル	変更点
<i>syslogins</i>	以前のリリースでは、 <i>password</i> カラムの最大長は 30 バイトだった。Adaptive Server リリース 15.0.2 では、 <i>password</i> カラムの最大長は 128 バイトとなっている。 新しいカラムは <i>lastlogindate</i> 、 <i>crdate</i> 、 <i>locksuid</i> 、 <i>lockreason</i> 、 <i>lockdate</i> 。
<i>sysssrvroles</i>	以前のリリースでは、 <i>password</i> カラムの最大長は 30 バイトだった。Adaptive Server リリース 15.0.2 では、 <i>password</i> カラムの最大長は 128 バイトとなっている。

『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 44 : 新しい設定パラメータ

設定パラメータ	説明
<b>enable merge join</b>	サーバレベルのマージジョインを有効または無効にする。
<b>cost of a logical io</b>	1 つの論理 I/O のコストを指定する。
<b>cost of a physical io</b>	1 つの物理 I/O のコストを指定する。
<b>cost of a cpu unit</b>	1 つの CPU 処理のコストを指定する。
<b>enable encrypted columns</b>	暗号化カラム機能を有効または無効にする (暗号化カラムのライセンスを保有している場合)。
<b>max online q engines</b>	MQ Series に必要。 <b>max online q engines</b> 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する。
<b>metrics elap max</b>	このクエリの経過時間が <b>metrics elap max</b> 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ない。 <b>enable metrics capture</b> がオンである必要がある。

設定パラメータ	説明
<b>metrics exec max</b>	このクエリの実行時間が <b>metrics exec max</b> 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ない。 <b>enable metrics capture</b> がオンである必要がある。
<b>metrics lio max</b>	このクエリの論理 I/O 時間が <b>metrics lio max</b> 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ない。 <b>enable metrics capture</b> がオンである必要がある。
<b>metrics pio max</b>	このクエリの物理 I/O 時間が <b>metrics pio max</b> 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ない。 <b>enable metrics capture</b> がオンである必要がある。
<b>min pages for parallel scan</b>	大量のクエリを並列でスキャンできる。
<b>not password encryption read</b>	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する。
<b>number of Q engines at startup</b>	MQ Series に必要。オンラインで実行できる Q エンジンの最大数を指定する。
<b>prod-consumer overlap factor</b>	最適化に影響する。Adaptive Server では、group by アルゴリズムが変更されている。並列プランに <b>set statistics IO</b> を使用することはできない。
<b>send doneinproc tokens</b>	<b>dbcc tune</b> オプション <b>doneinproc</b> に取って代わる。

表 45 : 変更された設定パラメータ

設定パラメータ	変更点
<b>max async i/os per engine</b>	デフォルト値が 2147483647 から 1024 に変更された。
<b>maximum failed logins</b>	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す。
<b>print deadlock information</b>	新しいパラメータ値 "2" が追加された。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る。

『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 46 : 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す。
<code>hashbytes()</code>	固定長のハッシュ値式を生成する。

表 47 : 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>in-did = 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る。 <code>in-did = 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る。
関数が作成されると、Adaptive Server は、この関数が SQL UDF か SQLJ UDF かをチェックする。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する。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する。	

## 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 48 : 新しいグローバル変数

グローバル変数	説明
<code>@@lastlogindate</code>	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る。 <code>datetime</code> データ型であり、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。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る。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 となる。

表 49 : 変更されたグローバル変数

グローバル変数	変更点
@@optimeoutlimit	<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 @@optimeout を取り上げていた。</p> <p>これは誤りであり、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 @@optimeoutlimit である。</p>

# バージョン 15.0.1

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 allows_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 allows_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 3 つの構文と同等です。

```
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。

## オペティマイザ基準 `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 バージョン 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> 値かどうかを確認する。
<code>isnumeric</code>	式が有効な <i>numeric</i> データ型かどうかを確認する。
<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 を表示する。

『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 50 : 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する。デフォルトでは、RepAgent は Adaptive Server と同時に起動する。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む。
<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する。
<code>cis idle connection timeout</code>	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 を設定する。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る。  指定する数値は秒単位であるが、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る。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る。

設定パラメータ	説明
<b>sproc optimize timeout limit</b>	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する。

表 51 : 変更された設定パラメータ

設定パラメータ	変更点
<b>optimization timeout limit</b>	<b>optimization timeout limit</b> に指定できる値の範囲が変更された。バージョン 15.0.1 では、0 ~ 1000。値 0 は、最適化のタイムアウトがないことを示す。
<b>max parallel degree</b>	<b>max parallel degree</b> 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る。
<b>number of worker processes</b>	<b>number of worker processes</b> 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する。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る。ただし、十分な数のワーカースレッドがない場合、 <b>alter table</b> および <b>execute immediate</b> コマンドはアボートされる。

『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 52 : 変更されたコマンド

テーブル	変更点
<b>alter table</b>	<p><b>alter table</b> 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する。<b>alter table</b> 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ない。</p> <p>構文は次のとおり。</p> <pre>alter table table_name drop partition partition_name [, partition_name]...</pre>

テーブル	変更点
<b>create index</b>	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る。
<b>create existing table</b>	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る。 <pre>create existing table (&lt;column_list&gt;) EXTERNAL [non_transactional  transactional] PROCEDURE at 'location'</pre> <ul style="list-style-type: none"> <li>• non_transactional - RPC の実行には別の接続を使用する。</li> <li>• transactional - RPC の実行には既存の接続を使用する。</li> </ul> デフォルトは transactional。
<b>update statistics</b>	Adaptive Server 15.0.1 では、グローバルインデックスに対して <b>update statistics</b> を実行する機能が追加されている。 <pre>update table statistics table_name [partition data_partition_name] [index_name [partition index_partition_name]]</pre> <b>update table statistics</b> を実行すると <b>update statistics</b> 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 <b>update statistics</b> を使用する。 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る。

表 53 : 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
<b>set literal_auto_param on off</b>	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する。
<b>set opttimeoutlimit</b>	<b>opttimeoutlimit</b> の値の範囲が 0 ~ 4000 (0 は最適化の制限がない) に変更された。

set コマンドの新しいオプション	説明
set index_union on   off	<p>set index_union を有効にすると、or 句によってテーブルのスキャンが制限される。</p> <p>インデックス和集合(または or 方式)は、or 句を含むクエリに使用される。次に例を示す。</p> <pre data-bbox="431 440 1171 493">select * from titleauthor where au_id="409-56-7008" or title_id="PC8888"</pre> <p>index_union を有効にした場合、この例では au_id のインデックスを使用して au_id = "409-56-7008" を使用するすべての titleauthor タブルのロー ID (RID) が検索され、title_id のインデックスを使用して title_id = "PC8888" を使用するすべての titleauthor タブルの RID が検索される。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する。結果の RID は RidJoin とジョインされてデータテーブルにアクセスする。</p> <p>index_union 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ない。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 titleauthor にテーブルスキャンを使用する)、or 句を scan 演算子のフィルタとして適用する。</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>	<p>プロシージャキャッシュアロケータごとに 1 つのローがある。アロケータはアロケータ ID で識別される。このアロケータ ID は、Adaptive Server の内部 ID である。</p>

モニタリングテーブル	説明
<i>monProcedureCache-ModuleUsage.</i>	プロセス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つのローがある。モジュールは、モジュール ID で識別され、Adaptive Server プロセスキャッシュ管理の内部の機能領域分類である。

『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。

# バージョン 15.0

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る。

## 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 $\sim$ +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  $\sim$  +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ー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ため、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。

## 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せん。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

- アポートカウント - リソース制限を超えたために Resource Governor によってクエリがアポートされた回数

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、カウントとアポ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 バージョン 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 54 : 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
<b>bcp</b>	<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 <b>--sho-fi</b> と <b>--hide-vcc</b> が追加されている。</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 <b>--maxconn</b> が追加されている。</p> <p>特定のパーティションとの間で <b>bcp in</b> および <b>bcp out</b> を実行できるように <b>bcp</b> のインタフェースが変更されている。</p>
<b>dataserver</b>	<b>-b master_database_size</b> パラメータをテラバイト単位で指定できる。
<b>sqlsrvr</b>	<b>-b master_database_size</b> パラメータをテラバイト単位で指定できる。
<b>ddlgen</b>	ユーザ定義 Web サービスをサポートするために、 <b>-T object_type</b> パラメータで指定可能なオブジェクトタイプに <b>WS</b> が追加されている。
<b>preupgrade</b>	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る。また、 <b>load database</b> 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 <b>preupgrade</b> を実行できるように機能が拡張されている。

『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 55 : 新しいグローバル変数

変数	表示対象	値
<b>@@rowcount</b>	<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す。</p>	<p>グローバル変数 <b>@@rowcount</b> の値は、指定されたカーソルタイプの影響を受ける。</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する。最大値は、結果セット内のローの数。</p> <p>スクロール可能カーソルの場合は、<b>fetch</b> コマンドの方向に関係なく、値は増加し続ける。最大値はない。</p>

変数	表示対象	値
@@fetch_status	スクロール可能カーソルに使用される fetch コマンドのステータスを表示する。	0 - fetch 文の実行が成功した。 -1 - fetch 文の実行に失敗した。または、要求されたローが結果セットの範囲外にある。 -2 - 値は予約済み。
@@cursor_rows	カーソル結果セット内のローの総数を表示する。	0 - オープンしているカーソルが1つもない。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ない。 -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ない。カーソルの条件を満たすローの数は不明。 <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る。返される値は、結果セット内のローの総数。

## 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 56 : 新しい設定パラメータ

関数	説明
enable metrics capture	Adaptive Server がサーバレベルで測定基準を取得できるようにする。
enable semantic partitioning	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) 分割を可能にする。
enable web services	Web サービスを有効にする。
enable xml	XML サービスを有効にする。
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する。
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する。これにより、Adaptive Server で水平並列処理を実行できる。

関数	説明
<b>max resource granularity</b>	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する。
<b>number of devices</b>	Adaptive Server が使用できるデータベースデバイスの数を指定する。
<b>number of dump threads</b>	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する。
<b>number of open partitions</b>	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する。
<b>optimization goal</b>	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つの層で指定できる。
<b>optimization time-out limit</b>	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する。
<b>rtm thread idle wait period</b>	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する。
<b>sysstatistics flush interval</b>	<i>sysstatistics</i> をフラッシュする間隔を分単位で指定する。
<b>statement cache size</b>	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する。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る。

表 57 : 変更された設定パラメータ

パラメータ	変更点
<b>default network packet size</b>	以前のバージョンの Adaptive Server では、 <b>default network packet size</b> として 512 が使用されていた。Adaptive Server バージョン 15.0 では、 <b>default network packet size</b> は 2048。

『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 58 : 新しい関数

関数	説明
<b>biginttohex</b>	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 進文字列を返す。
<b>count_big</b>	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> の値で返す。
<b>datachange</b>	update statistics 実行後のデータ分布の変化量を測定する。
<b>data_pages</b>	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す。
<b>hextobigint</b>	16 進文字列に相当する <i>bigint</i> 値を返す。
<b>is_quiesced</b>	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 を返す。
<b>partition_id</b>	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 を返す。
<b>partition_name</b>	新しいパーティションの明示的な名前を返す。
<b>reserved_pages</b>	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する。
<b>row_count</b>	指定したテーブル内のローの予測数を返す。
<b>showplan_in_xml</b>	XML 形式の実行プランを返す。
<b>ssel_message</b>	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す。
<b>tran_dumpable_status</b>	<b>dump transaction</b> を使用できるかどうかを示す true/false の値を返す。
<b>used_pages</b>	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する。
<b>xmlvalidate</b>	XML ドキュメントを検証する。非 ASCII 文字を使用しているドキュメントも対象 (I18N)。『Adaptive Server Enterprise における XML サービス』を参照。

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 59 : 古い関数名と新しい関数名

古い関数名	新しい関数名
data_pgs	data_pages
used_pgs	used_pages
reserved_pgs	reserved_pages
rowcnt	row_count
ptn_data_pgs	data_pages

## 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 60 : 新しいコマンド

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

表 61 : set コマンドの新しいオプション

set のオプション	説明
set delayed_commit	ログレコードをいつディスクに書き込むかを指定できる。 <b>delayed_commit</b> パラメータを true 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る。
set plan optgoal	最適化目標をセッションレベルで設定する。
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する。
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する。

表 62 : 変更されたコマンド

コマンド	変更点
<b>alter table</b>	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た。パーティションに対するサポートが追加された。
<b>create index</b>	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た。パーティションに対するサポートが追加された。
<b>create table</b>	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た。パーティションに対するサポートが追加された。
<b>dbcc</b>	パーティションに対するサポートが追加された。
<b>declare cursor</b>	スクロール可能カーソルの構文が追加された。 <b>semi_sensitive</b> 、 <b>insensitive</b> 、 <b>scrollable</b> カーソルをサポートするための構文が追加された。
<b>delete statistics</b>	パーティションに対するサポートが追加された。
<b>disk init</b>	<b>size</b> パラメータをテラバイト単位で指定できる。 <b>directio</b> パラメータが追加された。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る。
<b>disk reinit</b>	<b>directio</b> パラメータが追加された。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る。
<b>fetch</b>	スクロール可能カーソルをサポートするための <b>fetch_orientation</b> のオプション ( <b>next</b> 、 <b>prior</b> 、 <b>first</b> 、 <b>last</b> 、 <b>absolute</b> 、 <b>relative</b> ) が追加された。
<b>reorg</b>	パーティションに対するサポートが追加された。
<b>select</b>	XML サービスをサポートするための <b>for xml</b> 句が追加された。パーティションに対するサポートが追加された。
<b>truncate table</b>	パーティションに対するサポートが追加された。
<b>update all statistics</b>	パーティションに対するサポートが追加された。
<b>update statistics</b>	パーティションに対するサポートが追加された。
<b>update partition statistics</b>	廃止された。

## システムストアプロシージャ

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

表 63 : 新しいストアプロシージャ

ストアプロシージャ	機能
<code>sp_helpcomputedcolumn</code>	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する。
<code>sp_version</code>	最後に実行されたインストールスクリプト ( <code>installmaster</code> や <code>installdbccdb</code> など) のバージョン情報と、正常に実行されたかどうかを返す。

表 64 : 変更されたシステムストアプロシージャ

ストアプロシージャ	変更点
<code>sp_checksource</code>	計算カラムのソーステキストが存在することを確認する。
<code>sp_help</code>	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する。
<code>sp_helppartition</code>	パーティションの詳細な情報を出力に追加する。
<code>sp_helpindex</code>	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する。
<code>sp_helptext</code>	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する。
<code>sp_hidetext</code>	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す。
<code>sp_modifylogin</code>	オプション " <code>enable logins during recovery</code> " が追加されている。
<code>sp_webservices</code>	ユーザ定義 Web サービスをサポートするための <code>addalias</code> 、 <code>deploy</code> 、 <code>dropalias</code> 、 <code>listudws</code> 、 <code>listalias</code> 、 <code>undeploy</code> オプションが追加されている。
<code>sp_monitorconfig</code>	<code>number of open partitions</code> 設定パラメータがサポートされている。
<code>sp_countmetadata</code>	<code>number of open partitions</code> 設定パラメータがサポートされている。

ストアドプロシージャ	変更点
<code>sp_helpsegment</code>	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する。
<code>sp_objectsegment</code>	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する。
<code>sp_placeobject</code>	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する。
<code>sp_dbcc_faultreport</code>	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する。
<code>sp_sysmon</code>	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する。

『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 65 : 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る。すべてのカラムが新しくなっている。 <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る。
<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る。

表 66 : 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"> <li>• <b>computedcol</b></li> <li>• <b>status3</b></li> </ul> <p>新しいカラム：</p> <ul style="list-style-type: none"> <li>• <i>encrtype</i> - 暗号化のタイプ</li> <li>• <i>encrlen</i> - 暗号化カラムの長さ</li> <li>• <i>encrkeydid</i> - 暗号化キー ID</li> <li>• <i>encrkeydb</i> - 暗号化キーが格納されているデータベースの名前</li> <li>• <i>encrdate</i> - カラムを暗号化した日付</li> </ul> <p><i>status2</i> フィールドの新しいビット：</p> <ul style="list-style-type: none"> <li>• 16 進：0x00000010、10 進：16 - カラムは計算カラム。</li> <li>• 16 進：0x00000020、10 進：32 - カラムは実体化された計算カラム。</li> <li>• 16 進：0x00000040、10 進：64 - カラムはビュー内の計算カラム。</li> </ul>
<i>sysconstraints</i>	<p><i>status</i> 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す。</p>
<i>sysdevices</i>	<p>新しいカラム：</p> <ul style="list-style-type: none"> <li>• <i>vdevno</i> - デバイス ID 番号</li> <li>• <i>crdate</i> - デバイスが作成された日</li> <li>• <i>resizedate</i> - デバイスのサイズが変更された日</li> <li>• <i>status2</i> - 追加の <i>status2</i> ビット</li> </ul>
<i>sysusages</i>	<p>新しいカラム：<i>vdevno</i> - デバイス ID 番号</p>
<i>sysstatistics</i>	<p>新しいカラム：</p> <ul style="list-style-type: none"> <li>• <i>indid</i> - データパーティションのインデックス ID、常に 0</li> <li>• <i>partitionid</i> - データパーティションの ID</li> <li>• <i>ststatus</i> - 内部ステータスビット</li> </ul> <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>

テーブル	変更点
<i>systabstats</i>	<p>新しいカラム：</p> <ul style="list-style-type: none"> <li>• <i>partitionid</i> - データまたはインデックスパーティションの ID</li> <li>• <i>statmoddate</i> - ディスク上で統計値が最後に変更された日</li> <li>• <i>unusedpgcnt</i> - 未使用ページの数</li> <li>• <i>oampagecnt</i> - 各パーティションに対する OAM ページの数</li> </ul>
<i>syspartitions</i>	<p><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る。すべてのカラムが新しくなっている。<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る。</p>
<i>syscomments</i>	<p>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</p> <p>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る。</p>
<i>sysindexes</i>	<p>新しいカラム：</p> <ul style="list-style-type: none"> <li>• <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</li> <li>• <i>conditionid</i> - パーティション条件の ID</li> </ul> <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る。</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。</p>
<i>syslocks</i>	<p>新しいカラム：</p> <ul style="list-style-type: none"> <li>• <i>nodeid</i> - 今後のために予約済み</li> <li>• <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</li> </ul>

テーブル	変更点
<i>sysobjects</i>	<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"> <li>• <i>identburnmax</i> - <i>identity</i> カラムの場合、消去済み <i>identity</i> の最大値</li> <li>• <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)</li> <li>• <i>erlchgts</i> - 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)</li> </ul> <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"> <li>• <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" が追加されている。</li> <li>• <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る。</li> </ul>
<i>sysprocedures</i>	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る。

表 67：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</i> ~ <i>varchar(255) null</i>  <i>varchar(255)</i> ~ <i>varchar(768)</i>	オブジェクトの識別子
<i>sysaudits01</i> - <i>sysaudits08</i>	<i>objname</i>	<i>varchar(30) not null</i> ~ <i>varchar(255) not null</i>	オブジェクト名
<i>syscolumns</i>	<i>name</i>	<i>varchar(30) not null</i> ~ <i>varchar(255) not null</i>	カラム名
	<i>remote_name</i>	<i>varchar(30) null</i> ~ <i>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</i> ~ <i>varchar(255) null</i>	



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>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( <i>if</i> または <b>while</b> 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
<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 68 : 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>monOpenPartitionAcitivity</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る。

表 69 : 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。

## ヘルプと追加情報の取得

この製品リリースの詳細を確認するには、製品マニュアルサイトおよびオンラインヘルプを使用します。

- <http://sybooks.sybase.com/> の Product Documentation - マニュアルのオンライン版であり、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。Web サイトには、ホワイトペーパー、コミュニティフォーラム、メンテナンスリリース、サポートコンテンツ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 ドキュメントを参照または印刷するには Adobe Acrobat Reader が必要です。Adobe Acrobat Reader は Adobe Web サイトから無料でダウンロードできます。

---

**注意：**最新のリリースノートと、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は、Product Documentation Web サイトで確認できません。

---

## サポートセンタ

---

SAP® 製品のサポートを利用してください。

組織でこの製品の保守契約を購入している場合は、サポートセンタとの連絡担当者が指定されています。マニュアルだけでは解決できない問題があった場合には、担当の方を通してサポートセンタまでご連絡ください。

- 地域の Sybase サポートセンタまたは Sybase 子会社
- SAP サポートセンタ

SAP サポート契約を締結しているお客様は、SAP サポートサイト <http://service.sap.com/sybase/support> でこの製品のサポートを取得できます。また、ここから Sybase サポート移行の情報が探せます (ログインが必要なこともあります)。

Sybase サポート契約を締結しているお客様は、<http://www.sybase.com/support> でこの製品のサポートを取得できます (ログインが必要です)。

## 製品更新版のダウンロード

---

メンテナンスリリース、サポートパッケージ/パッチ、関連情報を入手します。

## ヘルプと追加情報の取得

- Sybase 認定販売店から製品を直接購入した場合:
  - a) <http://www.sybase.com/support> を開きます。
  - b) [Support] > [EBFs/Maintenance] を選択します。
  - c) MySybase のユーザ名とパスワードを入力します。
  - d) (オプション) フィルタ、時間枠のいずれかまたはその両方を選択して [Go] をクリックします。
  - e) 製品を選択します。

鍵のアイコンは、認可されたサポートコンタクトとして登録されていないため、一部の EBF/Maintenance リリースをダウンロードする権限がないことを示しています。未登録ではあるが、Sybase 担当者またはサポートセンタから有効な情報を得ている場合は、[My Account] をクリックして、「Technical Support Contact」の役割を MySybase プロファイルに追加します。

  - f) EBF/Maintenance レポートを表示するには [Info] アイコンをクリックします。ソフトウェアをダウンロードするには製品の説明をクリックします。
- Sybase 製品を SAP との契約に基づいて購入した場合:
  - a) <http://service.sap.com/swdc> を開き、必要に応じてログインします。
  - b) [Search for Software Downloads] を選択し、製品名を入力します。[Search] をクリックします。

## 製品およびコンポーネントの動作確認

---

動作確認レポートは、特定のプラットフォームでの Sybase 製品のパフォーマンスを検証します。

動作確認に関する最新情報は次のページにあります。

- パートナー製品の動作確認については、以下を参照してください。 [http://www.sybase.com/detail\\_list?id=9784](http://www.sybase.com/detail_list?id=9784)
- プラットフォームの動作確認については、以下を参照してください。 <http://certification.sybase.com/ucr/search.do>

## アクセシビリティ機能

---

アクセシビリティ機能を使用すると、身体障害者を含むすべてのユーザが電子情報に確実にアクセスできます。

この製品のマニュアルには、アクセシビリティを重視した HTML 版もあります。

オンラインマニュアルは、スクリーンリーダーで読み上げる、または画面を拡大表示するなどの方法により、視覚障害を持つユーザがその内容を理解できるよう配慮されています。

HTML マニュアルは、米国のリハビリテーション法第 508 条のアクセシビリティ規定に準拠していることがテストにより確認されています。第 508 条に準拠しているマニュアルは通常、World Wide Web Consortium (W3C) の Web サイト用ガイドラインなど、米国以外のアクセシビリティガイドラインにも準拠しています。

---

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

---

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

製品マニュアルには、アクセシビリティ機能に関する追加情報も記載されています。

ヘルプと追加情報の取得