

Pervasive PSQL v11 SP1

What's New in Pervasive PSQL

An Overview of New Features and Changed Behavior



免責事項

Pervasive Software Inc. は、本ソフトウェアおよびドキュメントの使用を、利用者またはその会社に対して「現状のまま」で、かつ同梱の使用許諾契約書に記載の契約条件によってのみ許諾するものです。Pervasive Software Inc. は、いかなる場合にも本ソフトウェアおよび本マニュアルに記載された内容に関するその他の一切の保証を、明示的にも黙示的にも行いません。Pervasive Software Inc. は、市場性、権利、特定の目的に対する適合性、あるいは一連の取引業務や職業的な使用に関する問題などに対し、一切の保証を行わないことを明示するとともに、利用者およびその会社がこれに同意したものとします。

商標

Btrieve、Client/Server in a Box、Pervasive、Pervasive Software および Pervasive Software のロゴは、Pervasive Software Inc. の登録商標です。

Built on Pervasive Software、DataExchange、MicroKernel Database Engine、MicroKernel Database Architecture、Pervasive.SQL、Pervasive PSQL、Solution Network、Ultralight、ZDBA は Pervasive Software Inc. の商標です。

Microsoft、MS-DOS、Windows、Windows 95、Windows 98、Windows NT、Windows Me、Windows 2000、Windows Server 2003、Windows 2008、Windows 7、Windows XP、Win32、Win32s、および Visual Basic は、Microsoft Corporation の登録商標です。

NetWare および Novell は Novell, Inc. の登録商標です。

NetWare Loadable Module、NLM、Novell DOS、Transaction Tracking System、TTS は、Novell, Inc. の商標です。

Sun、Sun Microsystems、Java、および Sun、Solaris、Java を含むすべての商標やロゴは、Sun Microsystems の商標または登録商標です。

すべての会社名および製品名は各社の商標または登録商標です。

© Copyright 2011 Pervasive Software Inc. All rights reserved. このマニュアルの全文、一部に関わりなく複製、複写、配布をすることは、前もって発行者の書面による同意がない限り禁止します。

本製品には、Powerdog Industries により開発されたソフトウェアが含まれています。

© Copyright 1994 Powerdog Industries. All rights reserved.

本製品には、KeyWorks Software により開発されたソフトウェアが含まれています。

© Copyright 2002 KeyWorks Software. All rights reserved.

本製品には、DUNDAS SOFTWARE により開発されたソフトウェアが含まれています。

© Copyright 1997-2000 DUNDAS SOFTWARE LTD. All rights reserved.

本製品には、Apache Software Foundation Foundation (<http://www.apache.org/>) により開発されたソフトウェアが含まれています。

本製品ではフリー ソフトウェアの unixODBC Driver Manager を使用しています。これは Peter Harvey (pharvey@codebydesign.com) によって作成され、Nick Gorham (nick@easysoft.com) により変更および拡張されたものに Pervasive Software が一部修正を加えたものです。Pervasive Software は、unixODBC Driver Manager プロジェクトの LGPL 使用許諾契約書に従って、このプロジェクトの現在の保守管理者にそのコード変更を提供します。unixODBC Driver Manager の Web ページは www.unixodbc.org にあります。このプロジェクトに関する詳細については、現在の保守管理者である Nick Gorham (nick@easysoft.com) にお問い合わせください。

GNU Lesser General Public License (LGPL) は本製品の配布メディアに含まれています。LGPL は www.fsf.org/licenses/licenses/lgpl.html でも見ることができます。

What's New in Pervasive PSQL

一般リリース 2011 年 7 月

138-004435-002

目次

このマニュアルについて	vii
このマニュアルの読者	viii
このマニュアルの構成	ix
表記上の規則	x
1 Pervasive PSQL v11 SP1 の新機能	1
新機能および変更された機能の概要	
Pervasive PSQL 用のパフォーマンス モニター カウンター	2
メッセージ ログ	3
ライセンス メッセージ	3
Pervasive Notification Viewer	4
オペレーティング システムのイベント ログ	4
Pervasive PSQL のイベント ログ	5
関連トピック	5
インストール	6
ファイアウォール	6
Pervasive Notification Viewer	8
各種ユーティリティ	9
License Administrator	9
Pervasive Notification Viewer	9
Pervasive System Analyzer (PSA)	10
リレーショナル インターフェイス	11
システム カタログ関数	11
スカラー関数	12
SDK アクセス方法	13
Btrieve API	13
DTI	13
DTO	14
PDAC	14
Linux	15
64 ビット ODBC ドライバー	15
Red Hat Enterprise Linux 6	22
A 以前の Pervasive PSQL v11 の新機能	23
Pervasive PSQL v11 の新機能の概要	
マルチコア サポート	24

目次

マルチコアをサポートする理由	24
マルチコアのジレンマ	26
現時点で得られる利点と将来的な計画	28
IPv6 のサポート	29
IPv6 を用いた Pervasive PSQL の使用	29
IPv6 サポートについてよく寄せられる質問	33
Pervasive PSQL のユーティリティと IPv6	34
アプリケーションプログラマーにとっての IPv6	35
64 ビット ODBC ドライバー	39
ODBC およびデータ ソース名 (DSN)	39
ODBC の変更によって影響を受けるユーティリティ	45
.NET Framework 3.5 SP1 および 4.0 のサポート	46
Pervasive PSQL ADO.NET データ プロバイダー 3.2	46
Pervasive PSQL ADO.NET データ プロバイダー 3.5	46
PDAC 開発環境	48
その他の SDK アクセス方法の機能拡張	49
DTO	49
製品認証	50
認証の方法	50
認証解除方法	51
仮想マシン	52
クラスター環境	52
障害復旧	53
OEM 向け製品認証	53
設定	54
通信スレッド数	54
リッスン IP アドレス	54
ユーティリティの変更点	55
Pervasive PSQL Control Center	55
ODBC アドミニストレーター	55
廃止予定および廃止された機能	56
廃止予定の機能	56
廃止された機能	56

表

1	Vista 以上のオペレーティング システム用のインストール動作	6
2	Linux 用の Pervasive PSQL ODBC ドライバー記述	16
3	Linux 向けの ODBC および DSN サポートに関する FAQ	18
4	Pervasive でサポートする IPv6 ユニキャスト アドレス形式	30
5	Pervasive PSQL での IPv6 の制限事項	32
6	IPv6 サポートについての FAQ	33
7	Windows 用の Pervasive PSQL ODBC ドライバー	40
8	ODBC および DSN の変更に関する FAQ	41

このマニュアルについて

このマニュアルには、本リリースの Pervasive PSQL における新機能や拡張機能に関する情報が記載されています。本リリースとは、Pervasive PSQL v11 SP1 を指します。

このマニュアルの読者

このマニュアルは、Pervasive PSQL に精通し、このソフトウェアの本リリースにおける変更点をお知りになりたいユーザー向けに作成されています。

本ソフトウェアの全般的な使用手順は説明していません。この製品の本リリースにおける新機能や変更点のみを説明することを目的としています。

このマニュアルの構成

この『What's New in Pervasive PSQL』の構成は以下のとおりです。

- 第1章「[Pervasive PSQL v11 SP1 の新機能](#)」
この章では、本ソフトウェアの現リリースにおける変更点の概要について説明します。
- 付録 A「[以前の Pervasive PSQL v11 の新機能](#)」
この章では、本ソフトウェアの初期リリースにおける変更点の概要について説明します。

このマニュアルの巻末には索引が用意されています。

表記上の規則

特段の記述がない限り、コマンド構文、コード、およびコード例では、以下の表記が使用されます。

大文字小文字の 区別	通常、コマンドと予約語は、大文字で表記されます。本書で別途記述がない限り、これらの項目は大文字、小文字、あるいはその両方を使って入力できます。たとえば、MYPROG、myprog、またはMYprog と入力することができます。
太字	太字で表示される単語には次のようなものがあります。メニュー名、ダイアログ ボックス名、コマンド、オプション、ボタン、ステートメントなど。
固定幅フォント	固定幅フォントは、コマンド構文など、ユーザーが入力するテキストに使われます。
[]	省略可能な情報には、 <code>[log_name]</code> のように、角かっこが使用されます。角かっこで囲まれていない情報は必ず指定する必要があります。
	縦棒は、 <code>[file name @file name]</code> のように、入力する情報の選択肢を表します。
< >	< > は、 <code>/D=<5 6 7></code> のように、必須項目に対する選択肢を表します。
変数	<i>file name</i> のように斜体で表されている語は、適切な値に置き換える必要のある変数です。
...	<code>[parameter...]</code> のように、情報の後に省略記号が続く場合は、その情報を繰り返し使用できます。
::=	記号 ::= は、ある項目が別の項目用語で定義されていることを意味します。たとえば、 <code>a::=b</code> は、項目 <i>a</i> が <i>b</i> で定義されていることを意味します。
% 文字列 %	Windows オペレーティング システムによって定義される変数です。「文字列」部分は可変テキストを表します。たとえば、 <code>%ProgramFiles%</code> は <code>C:\Program Files</code> という場所に用いる変数です。
\$ 文字列	Linux オペレーティング システムで定義される環境変数。「文字列」部分は可変テキストを表します。たとえば、 <code>\$PATH</code> には、シェルがコマンド（名前にスラッシュを含まない）を検索する際に用いるディレクトリが、コロン区切りで格納されています。

Pervasive PSQL v11 SP1 の新機能

1

新機能および変更された機能の概要

この一般リリースでは、以下の新機能と変更が含まれています。

- 「[Pervasive PSQL 用のパフォーマンス モニター カウンター](#)」
- 「[メッセージ ログ](#)」
- 「[インストール](#)」
- 「[各種ユーティリティ](#)」
- 「[リレーショナル インターフェイス](#)」
- 「[SDK アクセス方法](#)」
- 「[Linux](#)」

Pervasive PSQL 用のパフォーマンス モニター カウンター

Pervasive PSQL v11 SP1 では Windows パフォーマンス モニター ユーティリティで使用されるパフォーマンス カウンターを提供します。Pervasive PSQL のパフォーマンス カウンターはデータベース エンジンの状態や動作を測定します。これによりアプリケーションのパフォーマンスを分析することができます。Windows パフォーマンス モニターは、指定の時間間隔でパフォーマンス カウンターの現在の値を要求します。

パフォーマンス カウンターは、以下のプラットフォームで実行する Pervasive PSQL サーバーでのみ使用できます。

- Windows Vista 以上 (Windows 7、Windows Server 2008 および Windows Server 2008 R2 など)。
- 32 ビットおよび 64 ビット版の両オペレーティング システムがサポートされます。

デフォルトで、Pervasive PSQL のインストールでは Pervasive PSQL パフォーマンス カウンターをパフォーマンス モニターに登録します。このカウンターはインストール完了後に使用できるようになります。

パフォーマンス カウンターの使用は、主にアプリケーション開発者およびその他の技術スタッフ向けの高度な機能です。Windows パフォーマンス モニターおよび一般的なカウンターの使用の詳細については、Microsoft の関連ドキュメントを参照してください。

Pervasive PSQL パフォーマンス カウンターの詳細については、『Advanced Operations Guide』の「[パフォーマンスの分析](#)」を参照してください。

メッセージ ログ

Pervasive PSQL ではメッセージ用のさまざまなログ リポジトリをサポートするようになりました。ログは大きく分けて 2 つのカテゴリに分類されます。

- **すべてのメッセージ**：これらのメッセージにはステータス、エラー、警告および情報メッセージが含まれます。これらは、ライセンス管理コンポーネントを含め、あらゆる Pervasive PSQL コンポーネントから発生します。
- **ライセンス メッセージ**：これらのメッセージは、ライセンスに関する問題を警告し、そのトラブルシューティング情報を提供します。これらはライセンス管理コンポーネントから発生します。

次の表にログ リポジトリの概要を示します。

リポジトリ	記録元	注記
「 Pervasive Notification Viewer 」	ライセンス管理コンポーネント	Pervasive PSQL v11 SP1 での新機能
「 オペレーティング システムのイベント ログ 」 (Windows)	ライセンス管理コンポーネント	Windows イベント ログへの書き込みは Pervasive PSQL v11 SP1 での新機能です。
「 オペレーティング システムのイベント ログ 」 (Linux)	Pervasive PSQL の全コンポーネント	ライセンスに関するメッセージが追加され、記録されるようになりました。
「 Pervasive PSQL のイベント ログ 」 (Windows のみ)	Pervasive PSQL の全コンポーネント	ライセンスに関するメッセージが追加され、記録されるようになりました。

ライセンス メッセージ

上記の表で示したログ リポジトリのいくつかはライセンス メッセージに重点を置いています。これは、Pervasive PSQL のライセンス コンポーネントが、データベース エンジン用のキーの有効性を定期的に検証するからです。キーが無効と判定されると、そのキーの状態は "アクティブ" から "検証失敗" に変わります。この状態でも一定の期間内はデータベース エンジンが正常に機能するので、検証失敗を修正するための時間は十分にあります。

この期間が終わる前に検証失敗を修正しなかった場合は、キーの状態が "無効" に変わります。これにより、このキーは無効となりデータベース エンジンはデータ ファイルにアクセスできなくなります。

検証失敗への対応を適時に行う必要があるため、キーの状態の変更はさまざまなログ リポジトリからできるだけ早く通知されるようになっています。

メッセージ ログは一方方向の階層に従って記録されます。Pervasive Notification Viewer へ記録されるライセンス メッセージは、オペレーティング システムのイベント ログおよび Pervasive PSQL イベント ログにも記録されます。オペレーティング システムのイベント ログに記録されるライセンス メッセージは、Pervasive PSQL イベント ログにも記録されます。

Pervasive Notification Viewer

Pervasive Notification Viewer は、ライセンス コンポーネントによって記録されるメッセージを表示するアプリケーション ユーティリティです。このユーティリティは、注目すべきライセンス メッセージをわかりやすく通知することを目的としています。

Pervasive Notification Viewer は、Pervasive PSQL サーバー（Windows および Linux バージョンの 32 ビットおよび 64 ビット版）および Pervasive PSQL ワークグループのインストール時に（デフォルトで）一緒にインストールされます。

Windows プラットフォームの場合、この実行可能ファイルの名前は notifyviewer.exe です。Pervasive PSQL をインストールすると、このユーティリティはスタートアップに登録される（また [スタート] メニューにも追加される）ので、Windows の再起動時にこのユーティリティも再開します。

Linux ディストリビューションの場合、このユーティリティは notifyviewer という名前のシェル スクリプトです。Linux を再起動した場合は、Pervasive Notification Viewer を手動で再起動する必要があります。Linux の再起動時、このシェル スクリプトは自動的に実行されません。

Pervasive Notification Viewer では、システム トレイ アイコンとグラフィカル ユーザー インターフェイス (GUI) の 2 つのインターフェイスを提供します。

詳細については、『Advanced Operations Guide』の「[Pervasive Notification Viewer](#)」を参照してください。

オペレーティング システムのイベント ログ

Pervasive PSQL のライセンス管理コンポーネントは、Windows プラットフォーム上でオペレーティング システムのイベント ログにメッセージを書き込みます。Linux ディストリビューションの場合、Pervasive PSQL の全コンポーネントがオペレーティング システムのイベント ログへメッセージを書き込みます。

ライセンス メッセージは "エラー" または "警告" として分類され記録されます。さらに、特定の情報メッセージも記録されます。

Windows プラットフォームの場合、Pervasive PSQL は Windows のアプリケーション イベント ログにライセンス メッセージを記録します。これらは PVSU.LOG に書き込まれるライセンス メッセージのサブセットです。現在、Pervasive PSQL サーバー、ワークグループおよびクライアント 32

ビットではメッセージを記録します。Pervasive PSQL クライアント 64 ビットは記録しません。詳細については、『Advanced Operations Guide』の「[Windows プラットフォームのイベント ログ](#)」を参照してください。

Linux ディストリビューションの場合は、Pervasive PSQL の全コンポーネントが Linux の標準ログ システム syslog にメッセージを書き込みます。詳細については、『Advanced Operations Guide』の「[Linux ディストリビューションのイベント ログ](#)」を参照してください。

Pervasive PSQL のイベント ログ

Windows プラットフォームの場合、Pervasive PSQL の全コンポーネントが Pervasive PSQL のイベント ログ PVSU.LOG にステータス、エラー、警告および情報メッセージを書き込みます。ライセンス メッセージは " エラー " または " 警告 " として分類され記録されます。さらに、特定の情報メッセージも記録されます。詳細については、『Advanced Operations Guide』の「[Pervasive PSQL のイベント ログ](#)」を参照してください。

関連トピック

「[各種ユーティリティ](#)」と「[SDK アクセス方法](#)」も参照してください。

インストール

Pervasive PSQL v11 SP1 ではインストールに関して以下の点に変更されています。

- 「ファイアウォール」
- 「[Pervasive Notification Viewer](#)」

ファイアウォール

Pervasive PSQL サーバーおよび Pervasive PSQL ワークグループで、ファイアウォールに関するインストール動作が変更されています。この変更が適用されるのは Windows オペレーティング システムのみです。詳細については、『[Getting Started with Pervasive PSQL](#)』の「[Windows ファイアウォール](#)」を参照してください。

Windows Vista 以上

Windows Vista 以上のオペレーティング システムには、ファイアウォール プロファイル（セキュリティ設定のグループ）を提供する「セキュリティが強化された Windows ファイアウォール」があります。次の表では、アクティブなプロファイルに関するインストール動作についてまとめています。

表 1 Vista 以上のオペレーティング システム用のインストール動作

アクティブなファイアウォール プロファイル ¹	Pervasive PSQL サービス用に追加された規則	インストール後の規則の状態 ²
以下のプロファイル（複数の場合あり）。 ◆ ドメイン ◆ プライベート ◆ パブリック	◆ ドメインーはい ◆ プライベートーはい ◆ パブリックーはい	◆ ドメインー有効 ◆ プライベートー有効 ◆ パブリックー無効
パブリックのみ	はい	有効
¹ "アクティブ" とはプロファイルがネットワーク接続を監視していることを意味します。 ² "有効" とは、そのファイアウォールプロファイルによって管理されるネットワーク接続用の全ポートにおいて、受信 TCP および受信 UDP トラフィックと Pervasive PSQL サービスが通信できることを意味します。		

上の表でわかるように、パブリック プロファイルがその他の 1 つ以上のアクティブ プロファイルと共にアクティブな場合、Pervasive PSQL 規則はパブリック プロファイル用に追加されますが無効になっています。Pervasive PSQL サーバーまたはワークグループのインストールが対話式またはサイレントであってもこの動作を変更することはできません。パブリックプロ

ファイルの規則を有効にしたい場合は、手動で行う必要があります。『Getting Started with Pervasive PSQL』の「[Pervasive PSQL 規則をパブリックプロファイルに対して有効にするには](#)」を参照してください。



メモ コーポレートポリシーによっては、ローカル管理者が特定のマシンに対するファイアウォールプロファイルを変更できない（つまり、プロファイルがロックされている）こともあります。そのような場合、ロックされたプロファイルによって監視されるネットワーク接続経路でデータベースエンジンが通信するために必要なファイアウォール規則を、Pervasive PSQL のインストールによって追加したり有効にしたりすることはできません。そのような状況の場合は、権限を持つシステム管理者へファイアウォールポリシーの変更を要請してください。この変更で Pervasive PSQL でインストールされるすべてのサービスと通信する全ポートに対し受信 TCP および UDP トラフィックを許可します。

また、ターゲットシステムがドメインに参加している場合は、グループポリシーのみが、自身で制御するファイアウォールポリシーに対する規則を Pervasive PSQL インストールによって追加および有効化させないようにします。Pervasive PSQL をインストールしているユーザーが、ドメインユーザーとしてではなくローカルユーザーとしてターゲットシステムにログインしていた場合、そのインストールによってファイアウォールプロファイルに対する規則の追加および有効化が行われます。ただし、ターゲットシステムが後でグループポリシーを制御するドメインへ参加する場合はその規則が**無効**になります。

Windows Vista より前の Windows オペレーティング システム

Windows Server 2003 および Windows XP にはファイアウォールプロファイルが含まれていません。これらのオペレーティングシステムでは、インストールによって Pervasive PSQL の各サービスが Windows ファイアウォールの例外リストに追加されます。この結果、Windows ファイアウォールを介した全ポートで Pervasive PSQL サービスと通信する受信 TCP および受信 UDP トラフィックを許可する 1 つ以上のファイアウォール規則が追加されます。Pervasive PSQL で使用されるデフォルトのポートを変更する必要がある場合に備えてすべてのポートが使用されます。

Pervasive Notification Viewer

新しいユーティリティである Pervasive Notification Viewer は、Pervasive PSQL サーバー（Windows および Linux バージョンの 32 ビットおよび 64 ビット版）および Pervasive PSQL ワークグループのインストール時にデフォルトで一緒にインストールされます。

Windows プラットフォームの場合、Pervasive PSQL をインストールするとこのユーティリティはスタートアップに登録される（また [スタート] メニューにも追加される）ので、Windows の再起動時にこのユーティリティも再開します。Linux ディストリビューションの場合、このユーティリティはシェルスクリプトです（「メッセージ ログ」セクションの「[Pervasive Notification Viewer](#)」も参照してください）。

Ptksetup.ini には Pervasive Notification Viewer をインストールするかどうかを制御する新しいプロパティが含まれています。『Installation Toolkit Guide』の「[PTKSetup.ini](#)」を参照してください。

各種ユーティリティ

Pervasive SQL v11 SP1 には新しいユーティリティと変更されたユーティリティがあります。

- 「[License Administrator](#)」
- 「[Pervasive Notification Viewer](#)」
- 「[Pervasive System Analyzer \(PSA\)](#)」

License Administrator

License Administrator では以下の新しい機能を提供します。

機能	説明
キーの認証残り回数	キーの認証残り回数をチェックする新しい機能が追加されています。『 Pervasive SQL User's Guide 』の「 認証残り回数を表示するには 」(GUI) および「 認証残り回数を表示するには 」(CLI) を参照してください。
検証機能	キーを検証する新しい機能が追加されています。License Administrator ではこの検証機能により、すべてのキーの状態が更新されます。この機能を使用すると、キーの状態が変わったかどうかを見ることができます (この表の次の行を参照)。 『 Pervasive SQL User's Guide 』の「 検証を実行するには 」(GUI) および「 検証を実行するには 」(CLI) を参照してください。 「 DTI 」も参照してください。
キーの新しい状態 および検証失敗の 修正期間	キーの状態に " 検証失敗 " という新しい状態が追加されています。 Pervasive SQL のライセンス コンポーネントは、データベース エンジン用のキーが有効かどうかを定期的に検証します。キーが無効と判定されると、そのキーの状態は " アクティブ " から " 検証失敗 " に変わります。この状態でも一定の期間内はデータベース エンジンが正常に機能するので、検証失敗を修正するための時間は十分にありま。デフォルトで、この期間は 14 日間です。 この期間が終わる前に検証失敗を修正しなかった場合は、キーの状態が " 無効 " に変わります。これにより、このキーは無効となりデータベース エンジンはデータファイルにアクセスできなくなります (キーの状態が " 無効 " に変わるのはローカルタイム ゾーンの火曜日から木曜日のみとなっています。このため、修正期間は必要に応じて自動的に延長されます)。 『 Pervasive SQL User's Guide 』の「 キーの状態 」を参照してください。

Pervasive Notification Viewer

Pervasive Notification Viewer は、ライセンス コンポーネントによって記録されるメッセージを表示する新しいトレイ アプリケーション ユーティリティです。「メッセージ ログ」セクションの「[Pervasive Notification Viewer](#)」を参照してください。

Pervasive System Analyzer (PSA)

PSA では IPv6 についてネットワーク テスト用に数字および名前のアドレスをサポートするようになりました。Pervasive PSQL がサポートするのは、ユニキャスト タイプの IPv6 アドレス形式 "ループバック"、"グローバル" および "リンク ローカル" のみであることを注意してください。詳しい説明については、「[以前の Pervasive PSQL v11 の新機能](#)」の「[IPv6 のサポート](#)」を参照してください。

リレーショナル インターフェイス

このセクションでは、リレーショナル インターフェイスをサポートするための新しい機能と修正された機能について説明します。

- 「[システム カタログ関数](#)」
- 「[スカラー関数](#)」

システム カタログ関数

Pervasive PSQL v11 SP1 では、以下の新しいシステム カタログ関数が含まれています。

- `dbo.fSQLColumns`
- `dbo.fSQLForeignKeys`
- `dbo.fSQLPrimaryKeys`
- `dbo.fSQLProcedures`
- `dbo.fSQLProcedureColumns`
- `dbo.fSQLSpecialColumns`
- `dbo.fSQLStatistics`
- `dbo.fSQLTables`

システム カタログ関数を使用すると、データ辞書ファイル（「カタログ」とも呼ばれます）からデータベース メタデータを取得することができます。これは FROM 句でのみ使用することができます。各システム カタログ関数のテンポラリ ビュー スキーマは、SQLPrepare で作成されます。SQLExecute で対応する ODBC カタログ API を呼び出すことにより、そのビューにデータが格納されます。

『SQL Engine Reference』の「[システム カタログ関数](#)」の章を参照してください。

スカラー関数

以下のスカラー関数は、新規、パラメーターが追加された、あるいはこれまでドキュメントに記載されていなかった関数です。

機能	説明	『SQL Engine Reference』での参照セクション
DAY (date_exp)	指定された date_exp の日にちを返します。	「時刻と日付の関数」
DATEPART (datepart, date_exp)	TZoffset は値へ加算されるタイムゾーンオフセットで datepart 引数に使用できます。TZoffset は、タイムゾーンのオフセットを分数（符号付き）で返します。	「時刻と日付の関数」
ISNUMERIC(string)	string が数値として評価できる場合は 1 (TRUE) を返し、そうでない場合は 0 (FALSE) を返します。	「文字列関数」
REVERSE (string)	string の文字の並びが反転された文字列を返します。	「文字列関数」
SYSDATETIMEOFFSET	現在の日付と時刻に加えて、Pervasive PSQL データベース エンジンが起動しているコンピューターの現在のタイムゾーンと UTC の間の時間と分のオフセットを返します。	「時刻と日付の関数」
WEEKDAY (date_exp)	指定された date_exp の曜日を返します。1 = 日曜日、7 = 土曜日です。	「時刻と日付の関数」

SDK アクセス方法

Pervasive PSQL v11 SP1 では、以下の SDK アクセス方法を改善しました。

- 「Btrieve API」
- 「DTI」
- 「DTO」
- 「PDAC」

Btrieve API

Pervasive PSQL のダウンロードには、VB.NET 環境での Btrieve API の使用を示すサンプル アプリケーションとファイルが含まれるようになりました。このサンプルは、Pervasive Web サイト http://www.agtech.co.jp/support/reference/pervasive/psqlib/access_methods/btrieve.html からダウンロードできます。

DTI

DTI には以下の新しい関数および変更された関数があります。

関数	説明	新規または変更
「PvValidateLicenses()」	キーの妥当性をチェックする検証動作を開始します。	新規
「PvGetProductsInfo()」	License Manager で検出されたすべての Pervasive Software 製品の一覧を取得します。	変更。ライセンスに関する追加情報（猶予期間など）が含まれるようになりました。これは XML 文字列で返されます。



メモ PvValidateLicenses は、検証動作の要求から生じた結果のみを返します。キーの状態に関する情報は何も返しません。キーの状態に関する情報も含め、製品情報の XML 文字列を取得するには、別に PvGetProductsInfo() を呼び出す必要があります。

DTO

Pervasive PSQL DTO SDK では、.NET Framework を含む 64 ビット環境をサポートできるようになりました。64 ビット アプリケーションで DTO を利用するためには、Pervasive PSQL サーバー 64 ビットまたはクライアント 64 ビットをインストールして 64 ビット ライブラリ W64DTO2.DLL を取得する必要があります。『Distributed Tuning Objects Guide』の「[W64DTO2](#)」を参照してください。

"AnyCPU" としてコンパイルされた .NET アプリケーションで DTO を使用する場合、64 ビット版 DTO を使用すればアプリケーションは 32 ビットおよび 64 ビットの両オペレーティングシステムで DTO 呼び出しを行うことができます。『Distributed Tuning Objects Guide』の「[アプリケーションと DLL の相互作用に関する理解](#)」も参照してください。

DTO には以下の新しい関数および変更されたメソッドがあります。

DTO オブジェクト	メソッド	説明	新規または変更
DtoDatabase	RemoveUserFromGroup	既存のグループから既存のユーザーを削除します。	新規
DtoLicenseMgr オブジェクト	ValidateLicenses	すべてのキーの妥当性をチェックする検証動作を開始します。	新規
DtoLicenseMgr オブジェクト	GetProductInfo	License Manager で検出されたすべての Pervasive Software 製品の一覧を取得します。	変更。ライセンスに関する追加情報（猶予期間など）が含まれるようになりました。これは XML 文字列で返されます。



メモ ValidateLicenses は、検証動作の要求から生じた結果のみを返します。キーの状態に関する情報は何も返しません。キーの状態に関する情報も含め、製品情報の XML 文字列を取得するには、別に GetProductInfo を呼び出す必要があります。

PDAC

Pervasive PSQL v11 SP1 には、RAD Studio XE 開発環境と完全に統合された Pervasive Direct Access Components が含まれています。開発者用ドキュメントの『Pervasive Direct Access Components Guide』も参照してください。

Linux

Pervasive SQL v11 SP1 では Linux オペレーティング システム向けに以下のものがサポートされます。

- 「[64 ビット ODBC ドライバー](#)」
- 「[Red Hat Enterprise Linux 6](#)」

64 ビット ODBC ドライバー

Pervasive SQL v11 SP1 は、Linux で 64 ビット アプリケーション用 ODBC インターフェイスをサポートするようになりました。64 ビット ODBC ドライバーは Pervasive SQL サーバー 64 ビットおよび Pervasive SQL クライアント 64 ビットでインストールされます。

Linux ディストリビューションの場合、個々の ODBC ドライバーはドライバー マネージャー UNIXODBC を介してロードされます。このドライバー マネージャーはデータ ソース名 (DSN) から特定の Pervasive SQL ODBC ドライバーへのマッピングを保持します。

Pervasive SQL サーバー 64 ビットまたはクライアント 64 ビットがインストールされても、既に存在するユーザー定義の 32 ビット DSN に変更はありません。これは、これらの DSN は 64 ビット アプリケーションからは直接アクセスできないということです。新しい DSN については、上記の両製品のインストールで 32 ビットおよび 64 ビット ODBC ドライバーが `odbcinst.ini` に割り当てられます。この割り当てによって、1 つの DSN を 32 ビット アプリケーションと 64 ビット アプリケーションの両方で使用することができます。

既存の 32 ビット DSN を 32 ビット アプリケーションと 64 ビット アプリケーションの両方にアクセスさせたい場合は、それらを新しいスタイルの DSN として再作成する必要があります (表 2 の「[Pervasive ODBC Interface](#)」を参照)。

ODBC およびデータ ソース名 (DSN)

クライアント アプリケーションのビット数が Pervasive SQL サーバー製品のビット数と一致している必要はありません。たとえば、64 ビット ODBC ドライバーや 32 ビット ODBC ドライバーは、Pervasive SQL サーバー 64 ビットまたは Pervasive SQL サーバー 32 ビットのどちらの接続にも使用することができます。

ただし、Pervasive SQL クライアントの場合は、アプリケーションのビット数がクライアント マシン上の Pervasive SQL クライアントのビット数と一致している必要があります。つまり、クライアント上で 64 ビット アプリケーションを使用するには、Pervasive SQL 64 ビット クライアントがインストールされている必要があります。

Pervasive PSQL v11 は次の表で説明するように 3 つの ODBC ドライバーを提供します。

表 2 Linux 用の Pervasive PSQL ODBC ドライバー記述

INI ファイルにおける ODBC ドライバー記述	当該ドライバーがインストールされる PSQL 製品	一緒にインストールされる製品の動作
Pervasive ODBC Engine Interface	サーバー 64 ビット サーバー 32 ビット	<ul style="list-style-type: none"> ◆ 既存のエンジン DSN との互換性を維持するため ODBC ドライバーを <code>odbc.ini</code> に割り当てます。 ◆ ローカルの名前付きデータベースへ接続します。 ◆ 「<code>dsnadd</code>」ユーティリティを使用する場合、デフォルトでは作成されなくなりました (<code>-engdsn</code> オプションを指定する必要があります)。「ODBC の変更によって影響を受けるユーティリティ」も参照してください。 ◆ Windows クライアント上で実行している ODBC アドミニストレーターでエンジン DSN を表示すると見ることができます。 ◆ エンジン DSN を使用するため既にコード化された 32 ビット アプリケーションに使用されます。 ◆ Pervasive PSQL v11 で 32 ビット エンジン DSN を使用することは推奨しません。「以前の Pervasive PSQL v11 の新機能」を参照してください。

表 2 Linux 用の Pervasive PSQL ODBC ドライバー記述

INI ファイルにおける ODBC ドライバー記述	当該ドライバーがインストールされる PSQL 製品	一緒にインストールされる製品の動作
Pervasive ODBC Client Interface	サーバー 64 ビット サーバー 32 ビット クライアント 32 ビット	<ul style="list-style-type: none"> ◆ 既存のクライアント DSN との互換性を維持するため ODBC ドライバーを <code>odbc.ini</code> に割り当てます。 ◆ ローカルまたはリモートの名前付きデータベース、あるいはエンジン DSN へ接続します。 ◆ クライアント DSN を使用するため既にコード化された 32 ビット アプリケーションに使用されます。 ◆ <code>dsnadd</code> ユーティリティで <code>-host</code> を指定する場合、デフォルトでは作成されなくなりました (<code>-clntdsn</code> オプションを指定する必要があります)。「ODBC の変更によって影響を受けるユーティリティ」も参照してください。
Pervasive ODBC Interface	サーバー 64 ビット クライアント 64 ビット	<ul style="list-style-type: none"> ◆ 新しい 32 ビットおよび 64 ビット DSN で使用するため ODBC ドライバーを <code>odbcinst.ini</code> に割り当てます。 ◆ ローカルまたはリモートの名前付きデータベースへ接続します。 ◆ 32 ビットおよび 64 ビット アプリケーション用に推奨されるドライバー記述です。 ◆ 「dsnadd」ユーティリティを使用する場合、デフォルトで作成されます。「ODBC の変更によって影響を受けるユーティリティ」も参照してください。 ◆ Windows クライアント上で実行している ODBC アドミニストレーターで、エンジン DSN としては表示されません。

よく寄せられる質問

次の表では、Pervasive PSQL v11 SP1 における Linux 向けの ODBC および DSN のサポートについてよく寄せられる質問 (FAQ) の回答を記載しています。

表 3 Linux 向けの ODBC および DSN サポートに関する FAQ

質問	回答
32 ビット エンジン DSN は使用しない方がよいですか？	はい。新規アプリケーションまたは修正を施す 32 ビット アプリケーションの場合、エンジン DSN を使用するよりも、名前付きデータベースに接続する DSN を作成することをお勧めします。「以前の Pervasive PSQL v11 の新機能 」も参照してください。
"DSN レス" 接続とはなんですか？	DSN レス接続とは、ODBC ドライバー "Pervasive ODBC Client Interface" (既存の 32 ビット アプリケーション用) または "Pervasive ODBC Interface" (32 ビットまたは 64 ビット アプリケーション用) を使用して名前付きデータベースへ接続するものを指します。
Pervasive PSQL v11 SP1 サーバーまたはクライアントにアップグレードするときに、既存の 32 ビット エンジン DSN はどうなりますか？	移行のために必要な手続きはありません。既存のユーザー定義の 32 ビット エンジン DSN は引き続き適切に機能し、設定どおりに動作します。PSQL サーバーまたはクライアントマシン上のアプリケーションは、引き続き 32 ビット エンジン DSN を使用して動作します。 ただし、Pervasive PSQL サーバー 64 ビットでインストールされる DEMODATA サンプル データベースについては異なります。DEMODATA 用の DSN は、インストール時に "Pervasive ODBC Interface" を使用して再作成され、名前付きデータベースに接続します。 既存の 32 ビット DSN を 32 ビット アプリケーションと 64 ビット アプリケーションの両方にアクセスさせたい場合は、"Pervasive ODBC Interface" を使用してそれらの DSN を再作成し、ローカルまたはリモートの名前付きデータベースに接続する必要があります。
Pervasive PSQL v11 SP1 クライアントにアップグレードするときに、既存の 32 ビット クライアント DSN はどうなりますか？	移行のために必要な手続きはありません。既存のクライアント DSN は引き続きリモート エンジン DSN に接続します。 ただし、エンジン DSN の使用は推奨されないので、新規アプリケーションまたは修正を施す 32 ビット アプリケーションは、エンジン DSN ではなく名前付きデータベースに接続することをお勧めします。
エンジン DSN やクライアント DSN はまだ作成することができますか？	はい、作成できます。しかし、「 dsnadd 」では <code>-engdsn</code> または <code>-clntdsn</code> オプションを指定する必要があります。「 ODBC の変更によって影響を受けるユーティリティ 」を参照してください。いずれかのオプションで作成された DSN は 32 ビット アプリケーションのみに対応します。

表 3 Linux 向けの ODBC および DSN サポートに関する FAQ

質問	回答
<p>以前のリリースの PSQL クライアント (PSQL v10.x クライアントなど) からの接続はどうなりますか？</p>	<p>Pervasive PSQL はリモート クライアント DSN を引き続きサポートするので、以前のバージョンのクライアントも接続することができます。</p> <p>ただし、Pervasive PSQL サーバー 32 ビットおよび 64 ビットのどちらの場合もエンジン DSN は 32 ビットのみであることに注意してください。64 ビットエンジン DSN は Pervasive PSQL で作成することはできません。</p>
<p>Pervasive PSQL DSN 用の ODBC 接続文字列はどのようなものですか？</p>	<p>『SQL Engine Reference』の「ODBC 接続文字列」を参照してください。</p>
<p>32 ビット アプリケーションを 64 ビットに移植する場合、DSN について何か行う必要がありますか？</p>	<p>"Pervasive ODBC Interface" を使用して、名前付きデータベースに接続する新しい DSN を作成する必要があります。</p>
<p>データベース エンジンと一緒にインストールされる DEMODATA サンプル データベース用の DSN はどうなりますか？</p>	<p>Pervasive PSQL サーバー 64 ビットまたはクライアント 64 ビットをインストールすると、"Pervasive ODBC Interface" を使用して DEMODATA 用の DSN が作成され、名前付きデータベースに接続します。これは、Windows の ODBC アドミニストレーターを使用するクライアントの場合、Pervasive PSQL サーバーから [エンジン DSN] の一覧で DEMODATA が見えなくなることを意味します。このクライアントは [データベース名] の一覧を見る必要があります。64 ビットアプリケーション場合、エンジン DSN を介して DEMODATA に接続することはできません。</p>
<p>作成したアプリケーションが DTI を使用して DSN を管理する場合はどうなりますか？</p>	<p>DSN 用の DTI 関数で管理するのは 32 ビットエンジン DSN のみです。そのため、将来的には DSN 用の DTI 関数は 32 ビットエンジン インターフェイス ODBC ドライバーと共に使用できなくなる可能性があります。「以前の Pervasive PSQL v11 の新機能」を参照してください。</p> <p>(DSN を管理する DTI を使用する代わりに、SQL や ODBC ステートメントを使用することができます。たとえば、「CREATE DATABASE」を使用して名前付きデータベースを作成し、SQLConfigDataSource を使用して DSN を設定することができます)。</p>
<p>Pervasive PSQL v11 SP1 には 64 ビット ODBC および DSN をサポートする新しいユーティリティが含まれていますか？</p>	<p>はい。「ODBC の変更によって影響を受けるユーティリティ」を参照してください。</p>
<p>64 ビット ODBC および DSN をサポートするために既存のユーティリティが変更されていますか？</p>	<p>はい。「ODBC の変更によって影響を受けるユーティリティ」を参照してください。</p>

表 3 Linux 向けの ODBC および DSN サポートに関する FAQ

質問	回答
<p>一部の記述子フィールドは ODBC のさまざまな SQLSet... および SQLGet... 関数を介して設定できますが、これらの関数が 64 ビット値対応されている一方、それ以外の関数はまだ 32 ビット値対応ですか？</p>	<p>はい。64 ビット ODBC ドライバーを使用している場合はそうなります。記述子フィールドを設定および取得するときは、適切なサイズの変数を使用するようにしてください。詳細については、次の Web サイト（英語サイト）を参照してください。</p> <p>http://www.unixodbc.org/doc/ODBC64.html</p> <p>説明の要点は、SQL_ROWSET_SIZE は SQLGetStmtOption と SQLGetStmtAttr の両方でサポートされるということです。64 ビット ODBC ドライバーを使用し、SQLGetStmtOption または SQLGetStmtAttr を呼び出した場合、属性パラメーターが SQL_ROWSET_SIZE に設定されている場合には、*ValuePtr に 64 ビット値が返されます。</p>
<p>将来的に、ODBC 接続で推奨される方策はありますか？</p>	<p>はい。新規アプリケーションまたは修正を施す 32 ビットアプリケーションは、ローカルでもリモートでも、名前付きデータベースへ接続するようにしてください。</p> <p>将来的に Pervasive PSQL でエンジン DSN がサポートされなくなった場合には、アプリケーションで適切に対処してください。</p> <p>(SQL や ODBC ステートメントを使用することで DSN を管理することができます。たとえば、「CREATE DATABASE」を使用して名前付きデータベースを作成し、SQLConfigDataSource を使用して DSN を設定することができます)。</p>

ODBC ヘッダー ファイル

ODBC 用の sql.h、sqltypes.h および sqlxext.h ヘッダー ファイルには、32 ビットアプリケーションと 64 ビットアプリケーションのコンパイルに違いがあります。64 ビット ODBC の説明については、UNIXODBC Web サイトで ODBC に関するドキュメントを参照してください。たとえば、次の Web サイト（英語サイト）の情報が役立ちます。

<http://www.unixodbc.org/doc/ODBC64.html>

ODBC の変更によって影響を受けるユーティリティ

- 64 ビット版の isql、isql64 は、Pervasive PSQL サーバー 64 ビットおよびクライアント 64 ビットでインストールされます。『Pervasive PSQL User's Guide』の「[isql](#)」を参照してください。

- dsnadd ユーティリティは、デフォルトで "Pervasive ODBC Interface" を使用して DSN を作成するよう変更されています。このスタイルの DSN を使用すれば、1 つの DSN を 32 ビット アプリケーションと 64 ビット アプリケーションの両方で使用することができます。この DSN は ODBC アドミニストレーターでエンジン DSN として表示されません。また、odbc.ini ファイルには "Pervasive ODBC Engine Interface" という文字列は存在しません。

使用が推奨されないレガシースタイルのエンジン DSN またはクライアント DSN を作成したい場合は、-engdsn または -clntdsn オプションをそれぞれ指定する必要があります。『Pervasive PSQL User's Guide』の「[dsnadd](#)」を参照してください。

- Pervasive PSQL Control Center (PCC) の [データベースの新規作成] ダイアログで DSN を作成するためのオプションは、32 ビット限定となりました (オプション名は [32- ビット エンジン DSN の作成])。『Pervasive PSQL User's Guide』の「[データベースの新規作成 GUI リファレンス](#)」を参照してください。

PCC は 32 ビット アプリケーションです。64 ビットバージョンは使用できません。「[以前の Pervasive PSQL v11 の新機能](#)」も参照してください。

- Windows プラットフォーム用の Pervasive ODBC DSN セットアップ GUI は変更されました。「[以前の Pervasive PSQL v11 の新機能](#)」を参照してください。

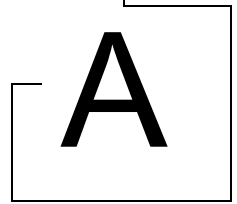
Red Hat Enterprise Linux 6

Pervasive PSQL は Red Hat Enterprise Linux 6 をサポートします。この 64 ビット版の Linux ディストリビューションは、デフォルトでは 64 ビットのコンポーネントと実行可能ファイルのみをサポートします。64 ビット Red Hat Enterprise Linux 6 の配布メディアには 32 ビット ライブラリも含まれますが、デフォルトでこれらはインストールされません。

次の表では、Red Hat Enterprise Linux 6 について、Pervasive PSQL に必要な特定の 32 ビット ライブラリの説明など、インストールに関する注意事項をまとめています。

Pervasive PSQL 製品	64 ビット版 Red Hat Enterprise Linux 6 のインストールに関する注記
<ul style="list-style-type: none"> ◆ 64 ビット サーバー ◆ 32 ビット サーバー ◆ 32 ビット クライアント 	<p>Pervasive PSQL の RPM および TAR インストール ファイルは、32 ビット GLIBC および LIBSTDC++ ライブラリが存在しているか確認します。これらのライブラリが存在しない場合、Pervasive PSQL のインストールは停止し、メッセージを表示します。Pervasive PSQL をインストールするには、最初に glibc.i686 および libstdc++.i686 をインストールしておく必要があります。</p> <p>Pervasive PSQL の RPM および TAR インストール ファイルは、32 ビット GTK が存在しているかどうか確認します。GTK が存在しない場合、Pervasive PSQL は警告メッセージを表示します。このメッセージは、32 ビット GTK がインストールされなければ PCC、DDF Builder および GUI 手動認証ユーティリティが機能しないことを通知します。このメッセージの表示後、Pervasive PSQL のインストールは最後まで続行します。</p> <p>PCC、DDF Builder および GUI 手動認証ユーティリティが動作するようにするには、以下の Linux パッケージをインストールします。</p> <ul style="list-style-type: none"> ◆ gtk2.i686 ◆ PackageKit-gtk-module.i686 ◆ libcanberra-gtk2.i686 ◆ gtk2-engines.i686 <p>たとえば、次のように指定します。</p> <pre>yum install gtk2.i686 PackageKit-gtk-module.i686 libcanberra-gtk2.i686 gtk2-engines.i686</pre> <p>上記の 3 つのユーティリティを除けば、32 ビット GTK が存在しなくても Pervasive PSQL は機能することに留意してください。つまり、データベースエンジンは機能し、すべてのコマンド ライン ユーティリティも動作します。また、Pervasive PSQL ライブラリは存在するのでアプリケーションからのアクセスも可能です。クライアント アプリケーションは、32 ビット GTK を必要としなければ機能します。</p>
<ul style="list-style-type: none"> ◆ 64 ビット サーバー ◆ 32 ビット サーバー 	<p>Pervasive PSQL のインストールでは Samba の構成を試みます。ただし、Samba は、デフォルトでその一部のみがインストールされ Red Hat Enterprise Linux 6 で有効になっていません。</p> <p>Samba 共有 PSQDATA の作成に関するインストール エラーを無視することができません (smb.conf ファイルは存在しますが、それをサポートするものは何もありません)。このエラーは無害なので Pervasive PSQL は正常に機能します。</p>

以前の Pervasive PSQL v11 の 新機能



Pervasive PSQL v11 の新機能の概要

この一般リリースでは、以下の新機能と変更が含まれています。

- 「マルチコア サポート」
- 「IPv6 のサポート」
- 「64 ビット ODBC ドライバー」
- 「.NET Framework 3.5 SPI および 4.0 のサポート」
- 「PDAC 開発環境」
- 「その他の SDK アクセス方法の機能拡張」
- 「製品認証」
- 「設定」
- 「ユーティリティの変更点」
- 「廃止予定および廃止された機能」

マルチコア サポート

Pervasive PSQL v11 は、特にマルチコア マシンにおけるスケーラビリティとパフォーマンスを向上させることを目的に設計されています。マルチコア マシンに Pervasive PSQL v11 をインストールすると、マルチユーザー環境ですぐにその効果が得られます。

どれほどの効果があるのか知りたいと思われるでしょう。ハードウェアテクノロジーの進歩によってスケーラビリティとパフォーマンスは明らかに向上しており、それらを利用できると考えられます。これまでは、ハードウェアテクノロジーの進歩といえば速度の高速化を指していました。つまり、アプリケーションの実行速度が速くなるということです。近年、コンピューターテクノロジーにおける進歩は、クロック速度の高速化ではなく並列処理効率の向上を指すようになってきました。これにより、アプリケーションに対し、おそらく今まで取り組むことがなかった課題が出てきました。

この進歩は、マルチコア環境のためにだけに変わったのではなく、劇的に変わっています。たとえば、複数のユーザーでデータを共有し、トランザクションの整合性を維持する必要があるデータベースを使用するアプリケーションの場合は、マルチコアプロセッサ上では実行速度が低下する可能性があります。

Pervasive PSQL v11 を使用する大多数のアプリケーションがこの状況に当てはまるので、マルチコアのサポートが Pervasive PSQL における主要な機能となっています。これはマルチユーザー アプリケーションをマルチコア環境に移行する場合に最も重要です。

マルチコアをサポートする理由

ほとんどのソフトウェア アプリケーションは変更することなくマルチコア マシンで実行できます。しかし、次のような状況では変更を検討する必要があります。これは実在のフィードバックに基づいています。

旧式の製品サーバーを最新のものにリプレースすることになりました。新しいマルチコア マシン（互換性のあるオペレーティング システム搭載）にマルチユーザー アプリケーションをインストールしました。これで、今までより動作が快適になると思われました。しかし、応答時間が遅くなってしまい、パフォーマンスはハードウェアのアップグレード前よりも低下してしまいました。

どうしてでしょう？これは、ビジネス ソリューションの重要な各要素が、マルチコアの新しい世界では互いに最適化されていないからです。

これを次のように考えてみましょう。" アプリケーション " は、作成したコード（一般的な定義では " アプリケーション "）、**データベース**、**オペレーティング システム**および**ハードウェア**という 4 つの主要な要素で構成されています。**ハードウェア**の変更は大きな影響があります。ただし、

これはそのハードウェアが変更前のものと根本的に異なっていた場合です。

しかし、適切な調整を行うことで処理速度が上がるアプリケーションであれば、ハードウェアの変更の利点を活用し、パフォーマンスを大幅に向上させることができます。多くの場合、データベースなどアプリケーションスタックの一部をスワップアウトすればマルチコアの問題を解決することができます。アプリケーションをすぐに変更する必要はありません。この手段は、アプリケーション開発の長期的な方策を講じる間、時間を稼ぐためのリスクの低い方法として用いることができます。

Pervasive PSQL v11 をデータベースとして使用すれば、マルチコアマシン上でパフォーマンスとスケーラビリティが向上することを実感することができます。

パフォーマンス

Pervasive PSQL v11 は、複数の類似動作を実行する並列スレッドを提供するように設計されています。並列処理が増加したことで、マルチプロセッサに関わる地点までのスループットが改善します。この結果、複数のクライアントがセントラルサーバーにアクセスするマルチコア環境で、データベースエンジンのパフォーマンスが向上します。マルチクライアントアプリケーションは、コードを再コンパイルしたり再設計することなく、このパフォーマンスの向上による利点を得ることができます。

また、**Pervasive PSQL v11** はトランザクショナルインターフェイスにおける低レベルの同期メカニズムについても機能強化しています。複数のユーザーが、キャッシュされた同じファイルページを同時に読み込むことができ、またそれらの操作はそれぞれ別々のサーバー CPU で処理することができます。チェックポイントやログ管理などの非ユーザー操作には、さらに別のサーバー CPU を使用することもできます。

スケーラビリティ

Pervasive PSQL v11 は、特にマルチコアハードウェア向けに行ったアーキテクチャ設計によってスケーラビリティも強化されました。たとえば、複数のユーザーがそれぞれ別々のファイルにアクセスする場合、それらの操作を別々のサーバー CPU で処理することができます。また、データベースエンジンは、以前に比べ少ないオーバーヘッドで多くのユーザーロードを処理することもできるので、より安定したスループットが実現します。

パフォーマンスの向上と同様に、コードを再コンパイルしたり再設計することなく、このスケーラビリティの向上によるすべての利点を得ることができます。

各種設定

Pervasive PSQL v11 におけるマルチコアに関する改善点の大部分は透過的です。その最適化をより高めるために調整する必要がある設定は特にありません。なお、[通信スレッド数] 設定は変更されています。この設定を使用すると、パフォーマンスを微調整することができます。「[設定](#)」を参照してください。

マルチコアのジレンマ

マルチコア環境におけるハードウェアとソフトウェアの相互作用によっていくつかの問題が出てきました。それによりアプリケーションでパフォーマンスが低下する可能性もあります。その問題の中には、マルチスレッドとメモリの競合などがあります。これらの問題およびその他の問題の詳細については、CITO Research の CTO（最高技術責任者）Dan Woods 氏によるホワイト ペーパー「マルチコアのジレンマ」(The Multi-core Dilemma) を参照してください。このホワイト ペーパーは Pervasive Web サイトから入手可能です。

本ドキュメントではマルチ スレッドとメモリの競合について簡単に説明します。これによりマルチコアのサポートが Pervasive PSQL v11 の主要な機能となっている理由を明らかにします。

マルチ スレッド

マルチスレッド化されたアプリケーションは、必ずしもマルチコア マシンで実行速度が速くなるわけではありません。事実、マルチスレッド化されたアプリケーションの実行速度は低下する場合があります。

スレッドが正しく並列に動作するためには、それらのスレッドを同期させる必要があります。アプリケーションがマルチスレッド化できても、スレッド自体が同期するわけではありません。実際、この状況はごく一般的なことで、古いアプリケーションは必要に応じて追加のスレッドを分離独立させています。これは、効率性を確保する設計に基づくよりも便利なためです。そのようなアプリケーションはスレッドが互いに競合するため、マルチコア マシンで効率よく実行されません。スレッドの競合により、マルチコアが関わっていない地点までのスループットが阻害されるので、マルチコアが提供されてもメリットがありません。

また、マルチコア アーキテクチャは、マルチ スレッドを一連のシングル スレッドとして分離独立させるサブタスクを認識することがあります。そうすると、シングルスレッド化されたプログラムと同じように、そのスレッドは単独のキューに強制的に入れられ、1 つずつ処理されます。キャッシュではこの問題を改善できません。より悪化してしまいます（「[メモリの競合](#)」を参照してください）。

可能であれば、コアごとに個別のデータを処理するようにします。そうでなければ、同期に伴うオーバーヘッドによってパフォーマンスが著しく低下する恐れがあります。Pervasive PSQL v11 は、同期される並列スレッドを提供するよう設計されていることを思い出してください。

メモリの競合

開発者はアプリケーションの作成時に、並列処理または非並列処理のどちらにするか決定する必要はありませんでした。アプリケーションの大半はシーケンシャルに作成されます。つまり、これらは逐次的または連続的に情報へアクセスするということです。メモリの競合に伴う問題は、マルチコアシステムで非並列（一般的）アプリケーションを実行したときに発生します。

人々が一群となって一斉に1つの出入り口を通り抜けようとする様子を描く（喜劇的）寸劇を思い浮かべてください。出入り口は人々でいっぱいになり、ぎゅうぎゅう詰めに押し込められている様子は面白いと思われるでしょう。では次に、これを複数のスレッド（人々）を同時に処理（出入り口）しようとする状況に置き換えてみてください。4個から16個（またはそれ以上）のスレッドが一度に同じプロセッサを通り抜けようとする場合は混雑が生じ、オペレーティングシステムがそれらを整理する必要があります。

複数のコアまたはプロセッサに同じデータを指すキャッシュがある場合、1つのコアがそのデータを変更すると、別のコアにキャッシュされたデータは無効になります。このため、キャッシュを同期させる必要があります。また、あるプロセッサのタスクが、別のプロセッサのタスクによって作成された古いデータに対して実行されないようにするために、各プロセッサがキャッシュを繰り返しチェックする場合にも競合が発生します。このチェックでは、各プロセッサがメモリ キャッシュを個別かつ順次にチェックするので、処理速度が低下します。

メモリの競合を減らす方法として、Pervasive PSQL v11 では複数のユーザーによる操作を別々のサーバー CPU 上で処理することを思い出してください。複数のユーザーが、キャッシュされた同じファイルページを同時に読み込むことができ、また別々のファイルにアクセスすることができます。

オペレーティング システムの役割

オペレーティング システム (OS) はマルチ コアの問題をどのように支援するのか、知りたいと思われるでしょう。最新の 64 ビット OS にもかかわらず、期待するほどの効果はありません。

リソースの競合が発生した場合、OS がその問題の解決にあたります。大多数のアプリケーションで、マルチコアシステム上の OS がスレッドの競合を処理するときには速度が低下します。つまり、マルチコアシステム上の OS では、競合ポイントの解決に時間がかかります。

これには次のような理由があります。アプリケーションがオペレーティングシステムに対し、単一ファイルで複数のタスクを実行する方法を今もなお要求している場合、マルチコア用に最適化されたオペレーティングシステムであっても問題を解決することはできません。

マルチコア処理用の命令が組み込まれていないアプリケーションからのリクエストを OS が受ける場合、その OS にとってリクエストを処理する順序の並び替えは大きな負荷となります。これは車の交通渋滞に似ています。概念的に言えば、OS はそれぞれの車を通す前に、待機している車の各運転手に対し、進行する準備ができていのかどうか尋ねます。そのような処理渋滞が OS レベルで起こっていたとしても、ユーザーは、その処理速度の低下をアプリケーションのパフォーマンスの問題としてとらえます。

マルチコア用に最適化されたアプリケーションは、共有リソースの管理方法およびリソースへのアクセス優先度について、OS へ指示を与えることができます。情報の要求は、キャッシュラインの競合を行わない、または中央メモリにアクセスしないよう整理されます。

Pervasive PSQL v11 には、特にマルチコアハードウェア向けのアーキテクチャ設計が含まれていることを思い出してください。低レベルロックは、マルチコアマシン向けに最適化されました。

現時点で得られる利点と将来的な計画

マルチコアマシンは今や標準仕様となっているので、現行または将来のハードウェアアップグレードには複数のコアが含まれます。一方、オペレーティングシステムは最良のパフォーマンスを支援するためのマルチコアマシンに対応できていません。これらの状況に対処するにはどのような方法が一番よいでしょうか。

最終的には、マルチコアマシンで最適に実行されるようアプリケーションを再設計する必要があります。そうすれば、アプリケーションはマルチプロセッサにおける並列スレッドを利用し、さらに同期の問題を回避することができます。

再設計には周到な計画と実装するための時間を要します。場合によっては数年かかるかもしれません。それまでの間は従来の状況が続くことになるでしょう。このセクションの冒頭でも述べたように、アプリケーションをマルチコア環境に移行する上で、マルチコアのサポートは最も重要です。

"アプリケーション" はコード、データベース、オペレーティングシステムおよびハードウェアで構成されています。ハードウェアシステムは既にマルチコアサポートに対応しています。アプリケーションがマルチコアを活用することによって、オペレーティングシステムからなんらかの支援が得られます。それはデータベースにゆだねられます。

Pervasive PSQL v11 のマルチコア機能は、マルチコア環境向けに最適化されていないアプリケーションでユーザーやエンドユーザーが経験するパフォーマンスの低下を補うのに役立ちます。ほとんどの場合、アプリケーションコードの再コンパイルや変更を行わなくても、アプリケーションのパフォーマンスを向上させることができます。

IPv6 のサポート

インターネット プロトコル バージョン 6 (IPv6) は IPv4 の後継として設計された次世代インターネット プロトコルです。ここでは、以下の項目について説明します。

- 「[IPv6 を用いた Pervasive PSQL の使用](#)」
- 「[IPv6 サポートについてよく寄せられる質問](#)」
- 「[Pervasive PSQL のユーティリティと IPv6](#)」
- 「[アプリケーション プログラマーにとっての IPv6](#)」

IPv6 を用いた Pervasive PSQL の使用

Pervasive PSQL v11 は Windows オペレーティング システムにおける以下のアクセス方法に IPv6 をサポートします。

- トランザクショナル (別名「Btrieve」)
- DTI (Distributed Tuning Interface)

どちらのアクセス方法も、IPv4 環境、IPv6 環境、またはこれら 2 つを兼ね備えた環境で正しく機能します。Pervasive PSQL で特別な設定を行う必要はありません。

クライアント接続

Pervasive PSQL クライアントは、Pervasive PSQL データベース エンジンを実行している IPv6 ホストへ接続します。これは IPv4 の場合も同じです。つまり、クライアントは DTI を介して、あるいは URI または UNC の記述によってサーバーを指定し接続します。サーバーの指定は、Pervasive PSQL サーバーまたはワークグループが実行されているマシンのマシン名または IP アドレスのどちらを用いても可能です。

以下のドキュメントも参照してください。

- 『Pervasive PSQL Programmer's Guide』の「[データベース URI](#)」
- 『Getting Started with Pervasive PSQL』の「[UNC \(Universal Naming Convention\) パス形式](#)」
- 『Distributed Tuning Interface Guide』の「[DTI を使ってサーバーへの接続を行う](#)」

次に IPv6 アドレスを使用してサーバーを指定する方法について説明します。

IPv6 アドレスの形式

未加工の IPv6 アドレスは、コロンで区切られた 8 個のセグメントで構成されます。各セグメントは 4 桁の 16 進数値として記述できます。たとえば、「1234:5678:90ab:cdef:1234:5678:90ab:cdef」と表記されます。

Pervasive PSQL がサポートするのはユニキャスト アドレスのみです。Pervasive PSQL で使用できるユニキャスト アドレス形式は以下のとおりです。

表 4 Pervasive でサポートする IPv6 ユニキャスト アドレス形式

ユニキャスト アドレス形式	説明
ループバック	<p>IPv6 でローカル ループバックアドレスは 0:0:0:0:0:0:1 と表記されます。このループバック アドレスは ::1 と省略表記することができます。</p> <p>IPv6 のループバック アドレスは、IPv4 のループバック アドレス 127.0.0.1 に相当します。</p>
グローバル	<p>グローバル アドレスは 64 ビット プレフィックスを持ちます。先頭から 3 ビットは常に 001 で、次の 45 ビットはグローバル ルーティング プレフィックス、その次の 16 ビットにはサブネット ID が設定され、最後の 64 ビットはインターフェイス ID となります。</p> <p>例 : 2001:db8:28:3:f98a:5b31:67b7:67ef</p>
リンク ローカル	<p>リンク ローカルアドレスは、同じリンク上の近隣ノードと通信を行う際にノードによって使用されます。リンク ローカル アドレスは 64 ビット プレフィックスを持ちます。先頭から 10 ビットには 1111 1110 10、次の 54 ビットには 0 が設定され、最後の 64 ビットはインターフェイス ID となります。このリンク ローカル アドレスのプレフィックスはたいてい FE80::/64 と表します。</p> <p>例 : fe80:0:0:0:713e:a426:d167:37ab (fe80::713e:a426:d167:37ab と指定することもできます)。</p> <p>「制限事項」も参照してください。</p>

IPv6 アドレスの修飾子

IPv6 にはアドレス識別子が含まれています。この識別子はショートカットとして機能したり、また詳細な宛先の指定に用いたりすることができます。Pervasive PSQL は IPv6 に以下の修飾子をサポートします。

修飾子	説明
::	1 つ以上のゼロがコロンで区切られていることを表します。たとえば、::1 は 0:0:0:0:0:0:0:1 に相当します。この修飾子 :: は IPv6 アドレス内で 1 回のみ使用できます。
%	宛先ノードのゾーン ID またはインターフェイスを表します。ゾーン ID は IPv6 トラフィックの宛先のゾーンを指定する整数値です。 「 制限事項 」を参照してください。

UNC パスおよび URI 接続を使用した IPv6

UNC パスでは、コロンなど特定の文字を使用することはできません。未加工の IPv6 アドレスではコロンを使用するので、UNC パスの扱いには異なる方法が使用できます。Pervasive PSQL は以下の方法をサポートします。

- 「[IPv6-literal.net 名](#)」
- 「[かっこ付き IPv6 アドレス](#)」

IPv6-literal.net 名

ipv6-literal.net 名は以下に示す 3 つの変更を施した未加工の IPv6 アドレスです。

- ":" は "-" に置き換えられます。
- "%" は "s" に置き換えられます。
- アドレスの末尾に ".ipv6-literal.net" が追加されます。

例：

```

当初のアドレス      fe80::713e:a426:d167:37ab%4
                    2001:db8:28:3:f98a:5b31:67b7:67ef

変更されたアドレス  fe80--713e-a426-d167-37abs4.ipv6-literal.net
                    2001-db8-28-3-f98a-5b31-67b7-67ef.ipv6-literal.net

```

Ipv6-literal.net 名は Pervasive PSQL で使用される URI または UNC で受け入れられます。

かっこ付き IPv6 アドレス

かっこ付き IPv6 アドレスとは角かっこで囲まれた未加工の IPv6 アドレスです。この形式は UNC で正しく動作するアドレスとしても参照されます。

例：

当初のアドレス	fe80::713e:a426:d167:37ab%4 2001:db8:28:3:f98a:5b31:67b7:67ef
変更されたアドレス	[fe80::713e:a426:d167:37ab%4] [2001:db8:28:3:f98a:5b31:67b7:67ef]

角かっこの使用は、Pervasive PSQL における URI または UNC で使用する未加工の IPv6 アドレスには必要です。「[制限事項](#)」を参照してください。URI で、ゾーン ID を含めたアドレスを使用する場合、ゾーン ID 文字 "%" はエスケープ文字 "%25" に置き換えられることに注意してください。「[制限事項](#)」を参照してください。

制限事項

Pervasive PSQL で IPv6 を使用する際の制限事項を次の表に示します。

表 5 Pervasive PSQL での IPv6 の制限事項

制限事項	説明
Windows Server 2003 および Windows XP オペレーティングシステムにおける Pervasive PSQL での IPv6 の使用	Windows Server 2003 および Windows XP オペレーティングシステム上の Pervasive PSQL では IPv6 の使用がサポートされません。Windows 環境の Pervasive PSQL で IPv6 を使用できるようにするには、Windows Vista 以降のオペレーティングシステムを利用してください。
Linux ディストリビューションにおける Pervasive PSQL での IPv6 の使用	Linux ディストリビューション上の Pervasive PSQL では IPv6 の使用がサポートされません。
URI または UNC で未加工の IPv6 アドレスを使用する場合、その IPv6 アドレスには角かっこが必要	未加工の IPv6 アドレスは、URI または UNC で使用する場合は角かっこで囲む必要があります。これはその IPv6 アドレスが省略表記されているかどうかにかかわらず必要です。 例： ◆ btrv://czjones@[2001:b1::23]/demodata ◆ btrv://abanderas@[2001:12:34:56:78:90:12:23]/demodata ◆ ¥¥[2001:12:34:56:78:90:12:23]¥acctsvr1¥Domestic¥file.mkd IPv6 アドレスを角かっこで囲まなかった場合、URI を使用した Btrieve 呼び出しにはステータスコード 3014 または 3103 が返され、UNC を使用した Btrieve 呼び出しにはステータスコード 11、94 または 170 が返されます。

表 5 Pervasive PSQL での IPv6 の制限事項

制限事項	説明
URI で、サーバー アドレスにゾーン ID を含める場合、ゾーン ID 文字 "%" は "%25" でエスケープされる	IPv6 アドレスを含めた btrv:// 接続を使用する場合、ホスト名に対するゾーン ID をエスケープする必要があります。 例： UNC 形式のアドレスは次のように表記されます。 btrv://@[fe80::20c:29ff:fe67:2ee4%4] このアドレスは、次のように変更されます btrv://@[fe80::20c:29ff:fe67:2ee4%254]
IPv6 のみの環境における PCC の使用	IPv6 のみの環境の場合、PCC で利用できるのはトランザクショナルまたは DTI アクセス方法によってサポートされる機能のみです。たとえば、IPv6 のみのマシン上の PSQL クライアントを IPv6 のみのサーバー マシン上のデータベース エンジンへ接続することができます。PCC ではエンジンおよびクライアントのプロパティの表示や設定が行えます。これが可能なのはそれらの機能が DTI を使用しているからです。ただし、データベースを参照したり Table Designer を使用することはできません。それらの機能はリレーショナル インターフェイスなどの別のアクセス方法を使用しているからです。このアクセス方法は IPv6 でまだサポートされていません。
License Administrator (および clicadm)	Pervasive ライセンス サーバーはまだ IPv6 をサポートしていません。このため、IPv6 上で License Administrator を使用しライセンスを管理することはできませんが、このユーティリティでライセンスを認証することはできません。ライセンスを認証するには、IPv4 ネットワークまたはリモート認証を使用する必要があります。

IPv6 サポートについてよく寄せられる質問

次の表では、Pervasive PSQL v11 における IPv6 のサポートについてよく寄せられる質問 (FAQ) の回答を記載しています。

表 6 IPv6 サポートについての FAQ

質問	回答
IPv6 を用いて自動再接続機能の Pervasive Auto Reconnect (PARC) を使用できますか？	はい。
仮想マシン環境で Pervasive PSQL は IPv6 通信をサポートしますか？	はい。
IPv6 のサポートはリレーショナル アクセス方法 (SRDE) に対しても適用されますか？	いいえ。トランザクショナルおよび DTI アクセス方法のみがサポートされます。
IPv6 は Linux ディストリビューションや Macintosh OS X でサポートされますか？	いいえ。Windows プラットフォームのみサポートされます。

以前の Pervasive PSQL v11 の新機能

表 6 IPv6 サポートについての FAQ

質問	回答
IPv6 は Pervasive DataExchange および Backup Agent でサポートされますか？	いいえ。
IPv4 と IPv6 が共存するネットワーク環境は Pervasive PSQL ユーザー カウントに影響がありますか？	いいえ。Pervasive PSQL サーバーまたはワークグループは、同じクライアント コンピューター セッション (TCP/IP および SPX など) から受信する一意のプロトコルごとに 1 つのユーザー カウントを使用します。IPv4 および IPv6 は単に TCP/IP のアドレス形式の異なったものです。
[リッスン IP アドレス] 設定に複数のアドレスを設定することはできますか？	はい。「 リッスン IP アドレス 」を参照してください。

Pervasive PSQL のユーティリティと IPv6

以下の Pervasive PSQL ユーティリティで IPv6 がサポートされます。これらユーティリティで特別な設定を行う必要はありません。

ユーティリティ	関連項目
bcfg	『Advanced Operations Guide』の「 設定リファレンス 」
Function Executor	『Advanced Operations Guide』の「 Btrieve オペレーションのテスト 」
License Administrator (および clilcadm)	『Pervasive PSQL User's Guide』の「 ライセンス管理 」 「 制限事項 」を参照してください。
Monitor (および bmon)	『Advanced Operations Guide』の「 データベース リソースのモニター 」
Pervasive PSQL Control Center (PCC)	『Pervasive PSQL User's Guide』の「 Pervasive PSQL Control Center の使用 」 IPv6 のみの環境で PCC を使用する場合は、「 制限事項 」を参照してください。

アプリケーション プログラマーにとっての IPv6

IPv6 は広く採用されていないため、ここではアプリケーション プログラマーにとって有用と思われるいくつかの側面について説明します。ネットワークの概念または IPv6 の詳しい説明は行いませんが、IPv6 について簡単に紹介します。IPv6 の詳しい説明については、www.ipv6.org（英語サイト）に記載の「IPv6 specification（仕様）」、また、さまざまなオペレーティング システム ベンダーおよびネットワーク ハードウェア ベンダーが提供している IPv6 ドキュメントを参照してください。

IPv6 の重要性

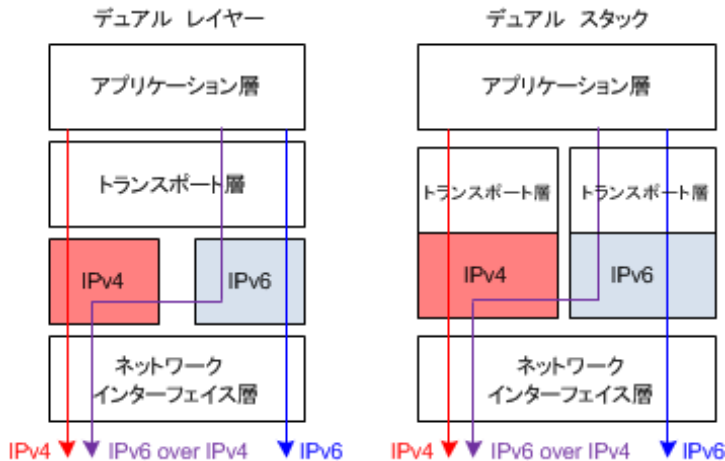
IPv6 は IPv4 の後継として設計された次世代インターネット プロトコルです。IPv4 はインターネットで使用された最初のインプリメンテーションであり、現在も最も多く使用されています。IPv4 には、その使用年数やネットワークにおける世界的環境の変化のため、将来的なニーズに対応できない限界がいくつかあります。

最も深刻な限界は、アドレス空間がいずれ枯渇するということでしょう。現在でさえ、パブリック IPv4 アドレスは相対的に十分とはいえません。さらに、全世界に広がるネットワークでは、構成性能の簡素化、セキュリティの強化および拡張性など、IPv4 が提供できる限度を超えた要件を導入しています。

IPv6 アドレスは、IPv4 の欠点を解決するだけでなく多くの利点をもたらします。最近のハードウェアやオペレーティング システムでは IPv6 サポートを提供します。特定の分野のアプリケーションでは既に IPv6 サポートが必要となっています。たとえば、米国および日本の政府は IPv6 への対応を義務付けています。IPv4 はいずれ IPv6 に切り替えられるため、この切り替えが早ければ早いほど IPv6 の利点が見えるようになります。

クライアント / サーバー通信

IPv4 から IPv6 サポートへの移行期間中は、いくつかのオペレーティング システムで両方のプロトコルが機能する可能性があります。オペレーティング システムによって、これは "デュアルレイヤー" または "デュアルスタック" と呼ばれます。ただし、IPv4 および IPv6 トラフィックは別々にルーティングされます。2 つのホスト間で通信する場合、双方が共に IPv4 対応、あるいは IPv6 対応である必要があります。



デュアル レイヤー

デュアル スタック

- ◆ Windows Vista、Windows Server 2008 および Windows 7 で使用可能
 - ◆ IPv6 はオペレーティング システムと共に自動的にインストール済み
 - ◆ IPv6 はアンインストール不可
 - ◆ IPv6 は停止可能
 - ◆ IPv4 は停止可能
- ◆ Windows Server 2003 および Windows XP (および Linux ディストリビューション) で使用可能
 - ◆ IPv6 は Windows プラットフォーム用のアドオンとしてインストール
 - ◆ IPv6 は Windows プラットフォームでアンインストール可能
 - ◆ IPv6 は停止可能
 - ◆ IPv4 は停止不可

オペレーティング システム レベルでネットワーク設定を構成する場合は、以下の点に留意してください。

オペレーティング システム	IPv6 に関する注記
Windows Server 2003 および Windows XP	IPv6 は手動でインストールする必要があります。 使用できる ネットワーク GUI ユーティリティはありませんが、ipconfig、netsh および nsupdate というコマンドライン ユーティリティが提供されます。
Windows Vista 以上	ネットワーク GUI 設定ユーティリティが使用できます。 また、コマンドライン ユーティリティ ipconfig、netsh および nsupdate も使用可能です。

ホスト ファイル、ゾーン ID および名前の検出

hosts ファイル内で、各 IP は互換性のあるアドレス形式を持つ行のみを使用します。たとえば、あるホスト名の IPv4 アドレスを要求した場合、IPv6 の行は無視されます。また、互換性のあるアドレスが localhost に適用されるので、通常 hosts ファイルには 127.0.0.1 (IPv4) および ::1 (IPv6) という localhost 行があります。

ホスト名を IP アドレスへ変換するために検索を行う場合、アプリケーションプログラマーが IPv4、IPv6 またはその両方を使用するかどうかを指定します。オペレーティング システムのネットワーク コンポーネントは、管理者レベルの環境設定を使用して、ローカル hosts ファイル、ローカル DNS キャッシュ、リモート DNS サーバーなど、検索の順序を決定します。IPv6 では、DNS を使用しないでリモート マシンを見つけることができる新しい自動検出プロトコルがあります。

以下の制限事項に従い、hosts ファイルに IPv6 アドレスを指定できます。

- hosts ファイルのレコードにはゾーン ID を含めることはできません。
- hosts ファイルには、同じノード名で IPv4 用および IPv6 用に別々の行を持つことができます。

hosts ファイルが最も役立つのは、ゾーン ID が不要な場合です。

ゾーン ID

ゾーン ID はネットワーク インターフェイスに割り当てられます。単独のネットワーク インターフェイス カード (NIC) およびゲートウェイを用いる場合、ゾーン ID は必要とされません。これは、単独のゲートウェイであれば 1 つのルートのみで到達するからです。IPv6 に対応しているほとんどマシンには、ISATAP、6to4、または Teredo のような変換ルーターのビルトイン サポートのため、複数のインターフェイスが搭載されています。

netsh コマンドでは、インターフェイス名 ("Local Area Connection 2" または "eth0" など) を使用する必要があります (interface= パラメーターを使用します)。IPv6 アドレスで ping を使用する場合は、ゾーン ID を使用する必要があるかもしれません。たとえば、「fe80::abcd%10」と指定される場合、10 進数の 10 がゾーン ID です。

Windows プラットフォームでは、ipconfig コマンドを使用するとインターフェイスごとのゾーン ID を表示できます。

名前の検出

IPv6 では、DNS (Domain Name System) を使用しないでリモート マシンを見つけることができる自動検出プロトコルがあります。LLMNR (Link Local Multicast Name Resolution : リンク ローカル マルチキャスト名前解決) は、DNS パケット形式に基づくプロトコルです。LLMNR を使用すると、IPv4 ホストおよび IPv6 ホストのどちらも、DNS サーバーを使用することなく単一のサブネット上のホストの名前解決を行うことができます。各 IPv6 マ

以前の Pervasive PSQL v11 の新機能

シンはリンク ローカルアドレスを持つため、そのマシンがサブネット上に存在する場合は、リンク グローバル アドレスで DNS 検索を行う前に、LLMNR によってそのマシンの場所を特定（名前解決）します。

64 ビット ODBC ドライバー

Pervasive PSQL v11 は、64 ビット アプリケーション用 ODBC インターフェイスをサポートするようになりました。64 ビット ODBC ドライバーは Pervasive PSQL サーバー 64 ビットおよび Pervasive PSQL クライアント 64 ビットでインストールされます。

ODBC およびデータ ソース名 (DSN)

64 ビット Windows オペレーティング システムの場合、Windows レジストリ設計により 64 ビット DSN と 32 ビット DSN は区別されています。Windows ODBC データ マネージャーの場合、作成されたアプリケーションのビット アーキテクチャ (ビット数) がわかっていること、またその同じビット数を用いて DSN を作成していることが要求されます。Pervasive PSQL v11 はこの同じモデルを採用しています。このため、64 ビット アプリケーションは 64 ビット ODBC ドライバーを使用し、32 ビット アプリケーションは 32 ビット ODBC ドライバーを使用します。

アプリケーションのビット数が Pervasive PSQL サーバー製品のビット数と一致している必要はありません。たとえば、64 ビット ODBC ドライバーや 32 ビット ODBC ドライバーは、Pervasive PSQL サーバー 64 ビットまたは Pervasive PSQL サーバー 32 ビットのどちらでも使用することができます。

以前の Pervasive PSQL v11 の新機能

Pervasive PSQL v11 は次の表で示すように 3 つの ODBC ドライバーを提供します。

表 7 Windows 用の Pervasive PSQL ODBC ドライバー

ODBC ドライバー	当該ドライバーがインストールされる PSQL 製品	一緒にインストールされる製品の動作
Pervasive ODBC エンジン インターフェイス	サーバー 64 ビット サーバー 32 ビット ワークグループ	<ul style="list-style-type: none"> ◆ 32 ビット エンジン DSN を作成します。 ◆ ローカルの名前付きデータベースへ接続します。 ◆ 32 ビット アプリケーション向け。 ◆ 後述のとおり、Pervasive PSQL v11 では使用を推奨しません。
Pervasive ODBC クライアント インターフェイス	サーバー 64 ビット サーバー 32 ビット クライアント 32 ビット ワークグループ	<ul style="list-style-type: none"> ◆ 32 ビット クライアント DSN を作成します。 ◆ ローカルまたはリモートの名前付きデータベース、あるいはエンジン DSN へ接続します。 ◆ GUI では、名前付きデータベースとエンジン DSN の両方を一覧に表示します。 ◆ 32 ビット アプリケーション向け。
Pervasive ODBC インターフェイス	サーバー 64 ビット クライアント 64 ビット	<ul style="list-style-type: none"> ◆ 64 ビット DSN を作成します。 ◆ ローカルまたはリモートの名前付きデータベースへ接続します。 ◆ 64 ビット アプリケーション向け。

名前付きデータベースへの接続方法を単純化するため、Pervasive PSQL v11 には以下の点を改善しています。

- 32 ビット エンジン DSN は廃止予定のため使用を推奨しません。32 ビット エンジン インターフェイス ドライバーは、主に以前のバージョンとの互換性を保持する目的で本リリースでも提供されます。Pervasive では、新規または修正を施す 32 ビット アプリケーションは、エンジン DSN を使用するより、クライアント DSN を介して、あるいは "Pervasive ODBC Client Interface" の指定による DSN レス (DSN を使用しない) 接続を使用して名前付きデータベースに接続することをお勧めします。
- 32 ビット エンジン DSN を管理する DTI 関数は廃止予定のため使用を推奨しません。「DTI」を参照してください。
- 64 ビット インターフェイス ドライバーは名前付きデータベースにのみ提供します。64 ビット ODBC インターフェイスはローカルの名前付きデータベースに接続できる (エンジン DSN の機能に置き換わる)、あるいはリモートの名前付きデータベースに接続できます。エンジン DSN への接続はサポートされていません。

よく寄せられる質問

次の表では、Pervasive PSQL v11 における ODBC および DSN のサポートについてよく寄せられる質問 (FAQ) の回答を記載しています。

表 8 ODBC および DSN の変更に関する FAQ

質問	回答
64 ビット ODBC ドライバーは Linux ディストリビューションや Macintosh OS X でサポートされますか？	いいえ。表 7 で示したように、Windows プラットフォームのみでサポートされます。
Pervasive PSQL v11 サーバーまたはワークグループにアップグレードするときに、既存の 32 ビット エンジン DSN はどうなりますか？	移行のために必要な手続きはありません。既存の 32 ビット エンジン DSN は引き続き適切に機能し、設定どおりに動作します。 PSQL サーバーまたはワークグループ マシン上のアプリケーションは、引き続き 32 ビット エンジン DSN を使用して動作します。
Pervasive PSQL v11 クライアントにアップグレードするときに、既存の 32 ビット クライアント DSN はどうなりますか？	移行のために必要な手続きはありません。既存のクライアント DSN は引き続きリモート エンジン DSN に接続します。 ODBC アドミニストレーターでクライアント DSN を編集する場合、リモート エンジン DSN を引き続き使用するか、またはリモートの名前付きデータベースを使用するか、どちらかを選択できます。「ODBC DSN セットアップの GUI」を参照してください。 ただし、エンジン DSN の使用は推奨されないので、新規または修正を施す 32 ビット アプリケーションは、エンジン DSN ではなく名前付きデータベースに接続することをお勧めします。
Pervasive ODBC Client Interface を使用する接続 ("DSN レス" 接続と呼ばれます) に何か影響がありますか？	いいえ。Pervasive ODBC Client Interface を使って接続する DSN レス接続は今までどおり動作します。
以前のリリースの PSQL クライアント (PSQL v10.x クライアントなど) からの接続はどうなりますか？	Pervasive PSQL v11 はリモート クライアント DSN を引き続きサポートするので、以前のバージョンのクライアントも接続することができます。 ただし、Pervasive PSQL サーバー 32 ビットおよび 64 ビットのどちらの場合もエンジン DSN は 32 ビットのみであることに注意してください。64 ビット エンジン DSN は Pervasive PSQL で作成することはできません。
Pervasive PSQL DSN 用の ODBC 接続文字列はどのようなものですか？	『SQL Engine Reference』の「ODBC 接続文字列」を参照してください。
32 ビット アプリケーションを 64 ビットに移植する場合、DSN について何か行う必要がありますか？	アプリケーションが DSN レス接続 (Pervasive ODBC Client Interface を使用した接続) を利用する場合、Pervasive ODBC Interface への接続文字列を変更します。『SQL Engine Reference』の「ODBC 接続文字列」を参照してください。 アプリケーションが DSN を使用する場合は、名前付きデータベースに接続する 64 ビット DSN を作成する必要があります。

表 8 ODBC および DSN の変更に関する FAQ

質問	回答
<p>データベース エンジンと一緒にインストールされる Demodata サンプル データベース用の DSN はどうなりますか？</p>	<p>Pervasive PSQL サーバー 32 ビットまたは Pervasive PSQL ワークグループのインストールでは、Demodata 用にエンジン DSN ではなくクライアント DSN を作成します。Pervasive PSQL サーバー 64 ビットのインストールでは、Demodata 用に 32 ビットクライアント DSN と 64 ビット DSN の両方が作成されます。</p> <p>Pervasive PSQL サーバー 32 ビットまたは Pervasive PSQL ワークグループのインストールに加えて Pervasive PSQL クライアント 64 ビットをインストールした場合、64 ビット DSN は作成されません。32 ビット データベース エンジンのインストールで作成される DSN のみが存在します。</p> <p>同様に、Pervasive PSQL クライアント 64 ビットのインストールに加えて Pervasive PSQL サーバー 32 ビットまたは Pervasive PSQL ワークグループをインストールした場合も、64 ビット DSN は作成されません。32 ビット データベース エンジンのインストールで作成される DSN のみが存在します。</p>
<p>64 ビット オペレーティング システム上で 32 ビットの ODBC アドミニストレーターはどのように実行すればよいですか？</p>	<p>『SQL Engine Reference』の「ODBC アドミニストレーター」を参照してください。</p>
<p>ODBC アドミニストレーターで DSN が見えないのはなぜでしょう？</p>	<p>64 ビット Windows オペレーティング システムの場合、レジストリ設計により 64 ビット システム DSN と 32 ビット システム DSN は区別されています。</p> <p>64 ビット ODBC アドミニストレーターを使用している場合は、32 ビット システム DSN は見えません。逆もまた同様です。</p> <p>なお、64 ビット オペレーティング システム上のリレーショナル サービス インターフェイスが、クライアントから エンジン DSN への接続を受け取った場合、データベース エンジンは要求されたエンジン DSN を 32 ビット レジストリでのみ検索するので注意してください。</p> <p>「ODBC DSN セットアップの GUI」 を参照してください。</p>
<p>作成したアプリケーションが DTI を使用して DSN を管理する場合はどうなりますか？</p>	<p>「DTI」 を参照してください。</p>
<p>ODBC アドミニストレーターにはどのような変更がありますか？</p>	<p>「ODBC DSN セットアップの GUI」 を参照してください。</p>
<p>ODBC アドミニストレーター以外で、64 ビット ODBC および DSN をサポートする新しいユーティリティが Pervasive PSQL v11 に含まれていますか？</p>	<p>いいえ。</p>

表 8 ODBC および DSN の変更に関する FAQ

質問	回答
64 ビット ODBC および DSN をサポートするために既存のユーティリティが変更されていますか？	はい。「ODBC の変更によって影響を受けるユーティリティ」を参照してください。
一部の記述子フィールドは ODBC のさまざまな SQLSet... および SQLGet... 関数を介して設定できますが、これらの関数が 64 ビット値対応されている一方、それ以外の関数はまだ 32 ビット値対応ですか？	<p>はい。64 ビット ODBC ドライバーを使用している場合はそうなります。記述子フィールドを設定および取得するときは、適切なサイズの変数を使用するようにしてください。詳細については、Microsoft の ODBC ドキュメントを参照してください。特に http://msdn.microsoft.com/en-us/library/ms716287%28VS.85%29.aspx (英語) をご覧ください。</p> <p>説明の要点は、SQL_ROWSET_SIZE は SQLGetStmtOption と SQLGetStmtAttr の両方でサポートされるということです。64 ビット ODBC ドライバーを使用していて、SQLGetStmtOption または SQLGetStmtAttr を呼び出した場合、属性パラメーターが SQL_ROWSET_SIZE に設定されている場合には、*ValuePtr に 64 ビット値が返されます。</p>
将来的に、ODBC 接続で推奨される方策はありますか？	<p>はい。ローカルまたはリモートの、新規または修正を施す 32 ビット アプリケーションは、エンジン DSN ではなくクライアント DSN を介した名前付きデータベースに接続するようにしてください。この代わりに、Pervasive ODBC Client Interface を指定することによってアプリケーションが DSN レス接続を使用するという方法もあります。</p> <p>将来的に Pervasive PSQL でエンジン DSN がサポートされなくなった場合には、アプリケーションで適切に対処してください。</p>

DTI

DSN 用の DTI 関数で管理するのは 32 ビット エンジン DSN のみです。そのため、以下の DTI 関数は 32 ビット エンジン インターフェイス ODBC ドライバーとともに廃止される予定です。

- 「PvCreateDSN()」
- 「PvCreateDSN2()」
- 「PvGetDSN()」
- 「PvGetDSNEx()」
- 「PvGetDSNEx2()」
- 「PvDeleteDSN()」
- 「PvListDSNs()」
- 「PvModifyDSN()」
- 「PvModifyDSN2()」

これらの関数の操作はすべて 32 ビット レジストリのみで行います。64 ビット データベース エンジンが 64 ビット オペレーティング システムに

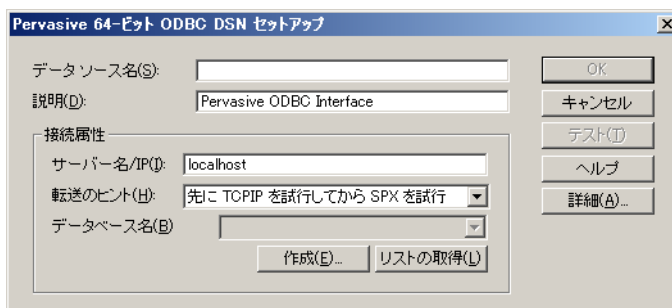
インストールされた場合にもこれは適用されます。32 ビット ODBC アドミニストレーターはエンジン DSN 用の DTI 関数を使用します。そのため、既存の エンジン DSN および新規作成されたエンジン DSN の一覧は 32 ビット レジストリ用のみです。

DSN を管理する関数の説明については、『Distributed Tuning Interface Guide』を参照してください。

ODBC DSN セットアップの GUI

ODBC アドミニストレーターによる DSN のセットアップには、以下のよう
な変更が行われました。

- 新しいグラフィカル ユーザー インターフェイス (GUI) では、64 ビット DSN のセットアップが可能です。表 7 「[Windows 用の Pervasive PSQL ODBC ドライバー](#)」も参照してください。



- 32 ビット クライアント DSN のセットアップ用の GUI は次のように変更されました。
 - この GUI を使用すれば、ローカルまたはリモートのサーバー名または IP アドレスの選択が行えます。表 7 「[Windows 用の Pervasive PSQL ODBC ドライバー](#)」も参照してください。



- [サーバー] セクション ラベルは [接続属性] という名称になりました。
 - [アドレス] オプション ラベルは [サーバー名/IP] という名称になりました。
 - [データソース名] オプション ラベルは [エンジン DSN] という名称になりました。
 - [オプション] ボタン ラベルは [詳細] という名称になり、このボタンをクリックすると詳細な接続属性が表示されます。この詳細な接続属性では、前バージョンの当該 [オプション] ダイアログで使用できた同じオプションを提供します。
- エンジン DSN のセットアップ用の GUI は次のように変更されました。
- [データベース] セクション ラベルは [接続属性] という名称になりました。
 - [オプション] ボタン ラベルは [詳細] という名称になり、このボタンをクリックすると詳細な接続属性が表示されます。この詳細な接続属性では、前バージョンの当該 [オプション] ダイアログで使用できた同じオプションを提供します。

GUI の新しいコントロールの説明については、『SQL Engine Reference』の「[DSN と ODBC アドミニストレーター](#)」の章を参照してください。

ODBC ヘッダー ファイル

ODBC 用の `sql.h` および `sqltypes.h` ヘッダー ファイルには、32 ビット アプリケーションと 64 ビット アプリケーションのコンパイルに違いがあります。64 ビット ODBC の説明については、Microsoft Web サイトで ODBC に関するドキュメントを参照してください。たとえば、以下の情報 (英語) が有用です。

[http://msdn.microsoft.com/en-us/library/ms716287\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms716287(VS.85).aspx)

ODBC の変更によって影響を受けるユーティリティ

64 ビット オペレーティング システムにインストールされた Pervasive PSQL サーバーおよびクライアントの場合、Pervasive PSQL Control Center (PCC) で 32 ビットまたは 64 ビット用に別々の ODBC アドミニストレーターを選択することができます。この選択は [ツール] メニューから行うことができます。『Pervasive PSQL User's Guide』の「[その他のユーティリティ](#)」を参照してください。

また、[データベースの新規作成] ダイアログで DSN を作成するためのオプションは、32 ビット限定となりました (オプション名は [32-ビット エンジン DSN の作成])。『Pervasive PSQL User's Guide』の「[データベースの新規作成 GUI リファレンス](#)」を参照してください。(PCC は 32 ビット アプリケーションです。64 ビット バージョンは使用できません。)

Pervasive ODBC DSN セットアップ GUI は変更されました。「[ODBC DSN セットアップの GUI](#)」を参照してください。

.NET Framework 3.5 SP1 および 4.0 のサポート

Pervasive PSQL v11 は ADO.NET データ プロバイダーでバージョン 3.2 および 3.5 の 2 つのバージョンを提供します。デフォルトで、どちらのバージョンもデータベース エンジンおよび Pervasive PSQL クライアントと一緒にインストールされます。

Pervasive PSQL の 32 ビット版または 64 ビット版のどちらをインストールした場合でも、両方のデータ プロバイダーが **Program Files (x86)** ディレクトリ下に置かれます。ただし、ビット数によってデータ プロバイダーはそれぞれ異なります (同じものではありません)。各データ プロバイダーは 32 ビットおよび 64 ビットの両方の .NET Framework で動作します。

Pervasive PSQL ADO.NET データ プロバイダー 3.2

Pervasive PSQL ADO.NET データ プロバイダー 3.2 には前のバージョンからの新機能はありません。これは Pervasive PSQL v11 でそのバージョン 3.2 を使用したいアプリケーション開発者向けに含まれています。

Pervasive PSQL ADO.NET データ プロバイダー 3.5

Pervasive PSQL ADO.NET データ プロバイダー 3.5 は .NET Framework 3.5 SP1 における新機能をサポートします。このプロバイダーは、.NET Framework バージョン 2.0、3.0、3.5、3.5 SP1、および 4.0 に準拠しています。バージョン 3.5 のプロバイダーは、.NET Framework 4.0 で導入された新機能をサポートしませんが、Entity Framework 1.0 の全機能をサポートする .NET Framework 4.0 で実行します。

さらに、Pervasive PSQL ADO.NET Data Provider 3.5 には以下の主な機能が含まれます。

- LINQ、EntitySQL および ObjectServices など、新しい Entity Framework コンシューマーに応じたメソッド セットを使用した開発
- Pervasive Bulk Load。DbBulkCopy クラスは共通プログラミング モデルでのデータのバルク ロードに対応しています。さらに、このデータ プロバイダーにはプロバイダー固有のバルク ロード クラスがあります。
- 接続統計情報のサポート
- 返されるスキーマ メタデータを指定する、スキーマ オプションの接続文字列オプション
- ネイティブ パラメーター マーカーおよびパラメーター バインディングのサポート
- Microsoft Enterprise Library 4.1 (2008 年 10 月) のサポート。Data Access Application Block (DAAB) のサポートを含みます。

- 接続確立時の初期コマンドタイムアウトを指定するための Initial Command Timeout 接続文字列オプション
- Microsoft Visual Studio 2008 および Visual Studio 2010 のサポート

詳しい説明については、開発者用ドキュメントの『Pervasive Data Provider for .NET Guide and Reference』を参照してください。

PDAC 開発環境

Pervasive PSQL v11 では以下の新たな開発環境向けの PDAC が含まれています。

- RAD Studio 2009
- RAD Studio 2010

RAD Studio 2009 および 2010 のサポートは、Pervasive PSQL v11 でサポートされる Delphi および C++ Builder コンポーネントのみを対象としています。

Pervasive PSQL v11 では、バージョン 6 より古い Delphi および C++ Builder 開発環境向けの PDAC 統合は提供されません。「[廃止予定および廃止された機能](#)」を参照してください。

開発者用ドキュメントの『Pervasive Direct Access Components Guide』も参照してください。

その他の SDK アクセス方法の機能拡張

Pervasive PSQL v11 では SDK アクセス方法である DTI (Distributed Tuning Objects) の機能を拡張しています。

DTO

Pervasive PSQL v11 には以下の新しいメソッドが含まれています。

DTO オブジェクト	メソッド	説明
DtoDatabase	「 AddUserToGroup 」	既存ユーザーをデータベースの既存グループに追加します。
	「 AlterUserName 」	指定されたデータベースの既存のユーザーの名前を変更します。
	「 AlterUserPassword 」	指定されたデータベースの既存のユーザーのパスワードを変更します。
	「 CreateGroup 」	既存のデータベースに新しいユーザーグループを作成します。
	「 CreateUser 」	既存のデータベースに新しいユーザーを作成します。
	「 DropGroup 」	データベースから既存のグループを削除します。
	「 DropUser 」	データベースから既存のユーザーを削除します。
DtoLicenseMgr	「 GetProductInfo 」	License Manager で検出された Pervasive Software 製品をすべて XML 形式のリストで返します。

『Distributed Tuning Objects Guide』でこの新しいメソッドを参照してください。

製品認証

製品の認証とは、特定のハードウェア構成アイテムを製品のライセンスに関連付けるキーの検証処理です（このキーは「製品キー」と呼びます）。この関連付けの結果、インストール ID が固有なものとなります。これにより、ソフトウェアが正当なものであること、また適切なハードウェアおよびソフトウェアプラットフォームに存在することを保証します。

製品キーを認証すると、その固有のインストール ID が Pervasive へ送られます。Pervasive ではキーの信頼性を検証し、そのキーが複数のインストールに使用されていないことを確認します。

認証の方法

デフォルトでは、評価版キーが Pervasive PSQL と一緒にインストールされ、このキーは一定期間を過ぎると失効します。データベース エンジンを使用し続けるためには、評価期間が終わる前に期限なしキーを用いて製品を認証しておく必要があります。製品の認証には、以下の 3 種類の方法があります。

- オンライン
- リモート
- オフライン

オンライン

Pervasive PSQL データベース エンジンインストールしているマシンがインターネットへ直接アクセスできる場合は、オンライン 認証が利用できます。

これは最も一般的な認証方法で、インストール処理の一部として透過的に行うことができます。インストール時に適切な時点でキーを入力すると製品の認証が処理されます。

インストール後でも、License Administrator を使用することで製品の認証が行えます。『Pervasive PSQL User's Guide』の「[キーを適用するには](#)」を参照してください。製品を認証した後に、データベース エンジンを再起動する必要はありません。

インターネットへ直接アクセスできない場合は、リモート認証またはオフライン認証を使用できます。

リモート

データベース エンジンがあるマシンはインターネット接続できないが、そのマシンがインターネット接続可能な別のマシンとネットワーク接続して

いる場合は、リモート認証が使用できます。PSQL クライアントは、インターネット アクセスが可能なマシンにあることが必要です。

インターネットへ接続できないマシンにデータベース エンジンをインストールする場合は、インストール時にキーを提供することができません。この場合は一時ライセンスを使って製品をインストールします。

製品のインストール後、クライアント マシンで PSQL のライセンス ユーティリティを使用して、インターネット接続不可のマシンで製品を認証します。リモート認証は、PSQL クライアントがデータベース エンジンとインターネット アクセスとの間を仲介する役目を果たすことを除けば、オンライン認証と同様です。

『Pervasive PSQL User's Guide』の「[キーをリモートで適用するには](#)」を参照してください。製品を認証した後に、データベース エンジンを再起動する必要はありません。

オフライン

データベース エンジンがインストールされているマシンが、ネットワークへもインターネットへも接続できない場合は、オフライン認証を使用します。

オフライン認証には2つのマシンを必要とします。どちらかのマシンにインターネット アクセスおよび PSQL クライアントが必要です。もう一方のマシンは孤立しており（クライアント マシンまたはインターネットへ接続できない）、データベース エンジンがあります。

オフライン認証に使用するユーティリティおよびその使用手順については、『Pervasive PSQL User's Guide』の「[キーをオフラインで適用するには](#)」を参照してください。製品を認証した後に、データベース エンジンを再起動する必要はありません。

認証解除方法

状況によっては、製品の認証を何度も行う必要があるかもしれません。たとえば、サーバー マシンを新しいものにリプレースする場合、その新しいマシンで Pervasive PSQL を認証したいと思われるでしょう。1つの製品キーを初めて認証解除した場合は、その後、最大5回まで認証することができます。

認証解除は、製品キーと特定のマシンのハードウェア構成との関連付けを切り離します。これにより、別のマシンで製品を認証することができ、新しいハードウェア構成と関連付け直すことができます。

Pervasive PSQL で提供しているライセンス ユーティリティを使用すれば、オンラインまたはリモートによる認証解除を使用してキーを認証解除することができます。オンラインで認証解除する場合は、「[キー削除するには](#)」を参照してください。リモートで認証解除する場合は、「[キーをリモートで](#)

削除するには」を参照してください。これらのタスクについては、『Pervasive PSQL User's Guide』を参照してください。

オフラインの認証解除については、弊社サポート部門へのご連絡が必要です。

製品キーを認証解除すると、そのキーに関連付けられているすべての追加ユーザー カウントも認証が解除されるので注意してください。一時ライセンスを認証解除することはできません。これは、認証の有効期限が過ぎると無効になります。



ヒント マシン上の Pervasive PSQL を認証した後に、そのマシンのある特定のハードウェア構成アイテムが変更されると、キーが無効になることを覚えておいてください(ハードドライブのシリアル番号、NIC カードの MAC アドレス、BIOS ファームウェア、CPU のタイプ、およびそのハードウェアで実行しているオペレーティング システムなどの構成の変更)。ハードウェア構成の変更を行う必要がある場合は、まずキーを認証解除しておいてください。認証解除は、製品キーとハードウェア構成との関連付けを切り離します。ハードウェア構成の変更が完了したら、再び製品キーを認証することができます。

仮想マシン

Pervasive PSQL v11 では、製品を仮想マシン内で認証することができます。Pervasive PSQL は物理マシンで直接使用しようと、仮想マシン内で実行しようと、同じライセンス要件が適用されます。各製品キーに関連付けられるソフトウェアは 1 つだけなので、仮想イメージ (クローンやコピーも含む) ごとに個別の製品キーが必要です。

仮想マシン用の認証および認証解除は、このセクションの前で説明した同じ方法で動作します。



ヒント 仮想マシンの構成を変更すると、物理マシンでの変更と同様にマシン ID が変わります。仮想イメージの構成 (メモリ割り当ての変更を除く) をコピー、移動または変更する場合は事前に、製品キーを認証解除しておきます。仮想イメージへの操作が完了したら、製品キーを再び認証します。

クラスター環境

仮想イメージと同様に、1 つのクラスター ノードで製品キーと関連付けられる Pervasive PSQL は 1 つだけです。クラスター環境内のノードごとにも、それぞれキーが必要です。

障害復旧

サーバーのハードウェアに障害が生じた場合、そのサーバー上の Pervasive PSQL 製品に関連付けられた製品キーはまだ認証されている状態です。これは、本製品を別のシステムで使用することができないことを意味します。製品キーを再認証できる状態にするには、弊社サポート部門までお問い合わせください。

直ちにシステムを作動させる必要がある場合は、製品 CD-ROM またはダウンロードしたインストール イメージを使用して、Pervasive PSQL v11 を 30 日間評価版として代替機にインストールします。

OEM 向け製品認証

Pervasive PSQL v11 では、製品の認証テクノロジーを弊社の OEM パートナーにも拡張します。OEM パートナーである場合、以下のリソースを参照してください。

- Pervasive Web サイト上の製品認証情報。
- Pervasive Web サイト上の OEM Web ポータル。この Web ポータルを使用すると、製品キーの生成、および製品キーに関するさまざまな管理者機能を実行できます。このポータルは 24 時間いつでも利用可能で、使いやすいインターフェイスを提供します (Pervasive PSQL 販売代理店はこのポータルに関してより多くの情報を提供できます)。このポータルで以下のドキュメントも参照してください。
 - ◆ 『OEM Partner Handbook』 – これは大幅に改訂されています。
 - ◆ ホワイト ペーパー – 「Pervasive PSQL Product Activation for OEM Partners」 (Pervasive PSQL OEM パートナー向けの製品の認証)。
 - ◆ 「Product Activation Troubleshooting Guide for OEM Support Staff」 (OEM サポート スタッフ向け製品認証トラブルシューティング)

設定

Pervasive PSQL v11 では以下の設定が変更されています。

- 「[通信スレッド数](#)」
- 「[リッスン IP アドレス](#)」

通信スレッド数

通信スレッドの設定範囲およびデフォルト値が変更されました。

- この範囲は、**プロセッサのコア数** ~ 256 になりました。プロセッサのコア数とは、データベース エンジンが起動しているマシンのプロセッサ数です。
- デフォルト値は、プロセッサのコア数です

前のバージョンでこの範囲は 1 ~ 1,024 で、デフォルトは 16 でした。

通信スレッドの設定は、ある特定の状況下における改善に役立ちます。たとえば、1 つのファイルに対して多くのクライアントが操作（主に書き込み）を行うような場合、より低い値を設定すればスケーラビリティが向上するはずです。スレッド数の値をより低くすると、システム リソースでコンテキスト スイッチを行わないようにします。このほか、大量のワーカー スレッド間でスラッシングによって速度低下が発生する状況も、この通信スレッドの設定によって改善する可能性があります。Pervasive PSQL v11 で、ワーカー スレッドは、既存の全スレッドがレコードまたはファイルのロックで待機している場合にのみ動的に作成されます。

『Advanced Operations Guide』の「[通信スレッド数](#)」を参照してください。

リッスン IP アドレス

[[リッスン IP アドレス](#)] 設定では、複数の IP アドレスを受け付けるようになりました。各アドレスはカンマで区切って指定できます。このアドレス文字列は IPv4 アドレスと IPv6 アドレスの組み合わせも可能です。IPv6 アドレスにサポートされるすべての形式が使用できます。「[IPv6 アドレスの形式](#)」を参照してください。

[TCP/IP マルチホーム] 設定が**オフ**（チェックボックスがオフ）の場合、データベース エンジンが受信待ちする（単一または複数の）IP アドレスを [[リッスン IP アドレス](#)] 設定に指定します。『Advanced Operations Guide』の「[リッスン IP アドレス](#)」および「[TCP/IP マルチホーム](#)」を参照してください。

ユーティリティの変更点

Pervasive PSQL v11 では、以下のユーティリティに対して変更が行われています。

- [「Pervasive PSQL Control Center」](#)
- [「ODBC アドミニストレーター」](#)

Pervasive PSQL Control Center

Pervasive PSQL Control Center (PCC) では DSN に関して、以下の点が変わっています。

- Pervasive PSQL サーバー 64 ビットの場合、PCC の [ツール] メニューには、32 ビットまたは 64 ビット用に別々の ODBC アドミニストレーターを選択できるサブメニューがあります。
- [データベースの新規作成] ダイアログで DSN を作成するためのオプションは、32 ビットであることが条件となりました (オプション名は [32-ビット エンジン DSN の作成])。

[「ODBC およびデータ ソース名 \(DSN\)」](#) も参照してください。

ODBC アドミニストレーター

32 ビット DSN 用の Pervasive ODBC セットアップ GUI が変更されました。64 ビット DSN 用の Pervasive ODBC セットアップ GUI が新たに追加され、利用できます。[「ODBC DSN セットアップの GUI」](#) を参照してください。

廃止予定および廃止された機能

廃止予定の機能

Pervasive PSQL v11 では、以下のカテゴリの機能を使用することは推奨しません。これらの機能は Pervasive PSQL v11 でも利用できますが、将来的には製品から削除される予定です。今後、新たにアプリケーションを開発したり、既存のアプリケーションを修正する際の参考にしてください。

ODBC

以下の ODBC 機能は Pervasive PSQL v11 でも利用できますが、将来的には製品から削除される予定です。

- 32 ビット エンジン DSN (32 ビット クライアント DSN がこれに匹敵する機能を提供するようになりました)。「[ODBC およびデータ ソース名 \(DSN\)](#)」を参照してください。
- 32 ビット エンジン DSN を管理する DTI 関数。「[DTI](#)」を参照してください。

Pervasive Direct Access Components (PDAC)

Delphi 2006 (および 2006 と互換性のある 2007) 用の PDAC 動的ライブラリは Pervasive PSQL v11 でも利用できますが、将来的には製品から削除される予定です。

廃止された機能

以下の機能は Pervasive PSQL v11 でサポートされなくなりました。

- Windows 2000 のサポート
- バージョン 6 より古い Delphi および C++ Builder 開発環境。Pervasive PSQL v11 は、バージョン 6 より古い開発環境との PDAC 統合を提供しません。
- Pervasive PSQL ADO.NET データ プロバイダー バージョン 2.1 および 3.0。Pervasive PSQL v11 データベース エンジンのインストールで、このバージョンのプロバイダーが検出された場合は自動的にアンインストールされます。

索引

P

Pervasive PSQL v11 SP1 での新機能

Btrieve API

VB.NET サンプル アプリケーション 13

DTI 関数

PvAuditLicenses 13

PvGetProductsInfo 13

DTO

64 ビット環境のサポート 14

DTO メソッド

AduditLicenses 14

GetProductInfo 14

RemoveUserFromGroup 14

License Administrator

認証残り回数 9

ライセンスの検証機能 9

Linux

64 ビット ODBC のサポート 15

Red Hat Enterprise Linux 6 のサポート 22

Notification Viewer 4

PDAC

RAD Studio XE との統合 14

PSA

IPv6 のサポート 10

PVSW.LOG への追加ログ 5

オペレーティング システムのイベント ログ
への記録 4

検証に失敗したキーの状態 3

パフォーマンス モニター カウンター 2

ファイアウォールに関するインストール動
作の変更 6

メッセージ ログ 3

猶予期間 3

リレーショナル インターフェイス

システム カタログ関数 11

新規および変更されたスカラー関数 12

Pervasive PSQL v11 での新機能

ADO.NET

.NET Framework 3.5 SP1 のサポート . 46

DSN

エンジン DSN と廃止予定の DTI 関数 43

セットアップ GUI 44

廃止予定の DTI 関数 43

DTI

廃止予定の関数 43

DTO メソッド

Pervasive PSQL v11 で新規 49

IPv6

FAQ 33

literal.net 名 31

UNC パスと URI 接続 31

アプリケーション プログラマーの観点
35

かっこ付きアドレス 32

クライアント接続 29

サポートしているアドレス形式 30

制限 32

.NET Framework

バージョン 3.5 SP1 のサポート 46

ODBC

64 ビット アーキテクチャ 39

DSN 設定 GUI 44

Pervasive PSQL で使用が推奨されない 40

Pervasive PSQL ドライバー 40

よく寄せられる質問 41

PDAC

新しい開発環境 48

RAD Studio 48

UNC パス

IPv6 を使用 31

URI 接続

IPv6 を使用 31

アーキテクチャ

64 ビット ODBC 39

キー

製品認証 50

製品認証 50

テクノロジーを OEM へ拡張 53

設定

Pervasive PSQL v11 用の変更 54

通信スレッド

マルチコア サポート用に変更 26

ドライバー

ODBC.....	40
廃止された機能.....	56
廃止予定の機能.....	56
マルチコア	
影響を受ける設定.....	26
影響を及ぼす主要なコンポーネント	24
サポート.....	24
パフォーマンスとスケーラビリティ	25
パフォーマンスと並列スレッド.....	25
マルチスレッドの問題点.....	26
メモリ競合の問題.....	27
ユーティリティ	
Pervasive PSQL v11 用の変更.....	55