



**SIOS Protection Suite for Linux
PostgreSQL Recovery Kit
v9.0.1**

管理ガイド

2015年12月

本書およびその内容は SIOS Technology Corp. (旧称 SteelEye® Technology, Inc.) の所有物であり、許可なき使用および複製は禁止されています。SIOS Technology Corp. は本書の内容に関していかなる保証も行いません。また、事前の通知なく本書を改訂し、本書に記載された製品に変更を加える権利を保有しています。SIOS Technology Corp. は、新しい技術、コンポーネント、およびソフトウェアが利用可能になるのに合わせて製品を改善することを方針としています。そのため、SIOS Technology Corp. は事前の通知なく仕様を変更する権利を保留します。

LifeKeeper、SteelEye、および SteelEye DataKeeper は SIOS Technology Corp. の登録商標です。

本書で使用されるその他のブランド名および製品名は、識別のみを目的として使用されており、各社の商標が含まれています。

出版物の品質を維持するために、弊社は本書の正確性、明瞭性、構成、および価値に関するお客様のご意見を歓迎いたします。

以下の宛先に電子メールを送信してください。

ip@us.sios.com

Copyright © 2015

By SIOS Technology Corp.

San Mateo, CA U.S.A.

All rights reserved

目次

Chapter 1: はじめに	1
PostgreSQL Recovery Kit のドキュメンテーション	1
概要	1
SIOS Protection Suite のドキュメンテーション	1
PostgreSQL のドキュメンテーション	1
PostgreSQL リソース階層	1
要件	2
PostgreSQL Recovery Kit の要件	2
ハードウェア要件	3
ソフトウェア要件	3
Chapter 2: 設定上の考慮事項	5
PostgreSQL の保護: ベストプラクティス	5
DataKeeper でのミラーファイルシステムの使用	6
Chapter 3: インストール	7
LifeKeeper での PostgreSQL のインストール/ 設定	7
リソース設定作業	7
アップグレード	8
PostgreSQL ソフトウェアのインストール	8
PostgreSQL データベースの作成	9
LifeKeeper ソフトウェアのインストール	10
PostgreSQL リソースのための LifeKeeper 設定項目	10
LKPGSQL_CONN_RETRIES	10
LKPGSQL_DISCONNECT_CLIENT	11
LKPGSQL_SDIRS	11
LKPGSQL_IDIRS	11
PostgreSQL リソース階層の作成	11

PostgreSQL リソース階層の削除	13
PostgreSQL リソース階層の拡張	13
PostgreSQL リソース階層の拡張解除	15
PostgreSQL 設定の参照	15
PostgreSQL Recovery Kit の旧バージョンからのアップグレード	16
Chapter 4: 管理	21
データベース管理者ユーザの更新	21
PostgreSQL リソース階層のテスト	21
EnterpriseDB Postgres Plus Advanced Server 環境	21
LifeKeeper GUI からの手動スイッチオーバーの実行	21
EnterpriseDB Postgres Plus Advanced Server の保護	21
データベース管理者ユーザの更新	22
Chapter 5: トラブルシューティング	25
全般項目	25
設定項目	26

Chapter 1: はじめに

PostgreSQL Recovery Kit のドキュメンテーション

概要

SIOS Protection Suite for Linux PostgreSQL Recovery Kit は、SQLに準拠した、POSTGRESをベースとするオブジェクトリレーショナルデータベース管理システム (ORDBMS) です。PostgreSQL は誕生以来進化を続け、最も高度なオープンソースリレーショナルデータベース管理システムの1つになりました。

SPS for Linux PostgreSQL Recovery Kit は、LifeKeeper 内の PostgreSQL インスタンスを保護する機能を提供します。PostgreSQL ソフトウェア、LifeKeeper Core、および PostgreSQL Recovery Kit はクラスタ内の2台以上のサーバにインストールされます。PostgreSQL データベースインスタンスが LifeKeeper の保護下に入ったら、クライアントは LifeKeeper が保護する IP アドレスを使用してデータベースに接続します。LifeKeeper が保護する IP アドレスは別々に作成し、親 PostgreSQL リソースインスタンスと子 IP アドレスリソースの間に依存関係を手動で作成する必要があります。PostgreSQL サーバに障害が発生した場合、LifeKeeper はまずローカルサーバ上での復旧を試みます。ローカルリカバリに失敗した場合、バックアップサーバに切り替えます。

[PostgreSQL リソース階層](#)

SIOS Protection Suite のドキュメンテーション

SIOS Technology Corp の Web サイトでは以下の SIOS Protection Suite 製品のドキュメンテーションを提供しています。

- SPS for Linux リリースノート
- SPS for Linux テクニカルドキュメンテーション
- Optional Recovery Kit ドキュメンテーション

PostgreSQL のドキュメンテーション

Administration Guide、*User Guide*、*Reference Guide* を含む PostgreSQL のドキュメンテーションは以下の Web サイトで入手できます。

<http://www.postgresql.org/docs>

PostgreSQL リソース階層

以下の例は典型的な PostgreSQL リソース階層を示しています。

要件



上記の例の依存関係は、以下の保護されるリソースに対応しています。

リソース	PostgreSQL ソフトウェアコンポーネント
<i>LKIP.EXAMPLE.COM</i>	クライアント接続に使用される切り替え可能な IP アドレスを保護します。
<i>var/lib/pgsql/data</i>	データベースデータディレクトリ (PGDATA) を保護します。
<i>var/lib/pgsql/exec</i>	PostgreSQL サーバとクライアントの実行可能ファイルを保護します (実行可能ファイルが共有ファイルシステムにインストールされている場合)。
<i>var/lib/pgsql/log</i>	データベースログファイルディレクトリを保護します (ログパスが共有ファイルシステム上にある場合)。
<i>var/lib/pgsql/pg_xlog</i>	データベーストランザクションログディレクトリ (PGDATA/pg_xlog) を保護します。トランザクションディレクトリは Write-Ahead-Log ディレクトリとしても参照されます。
<i>var/lib/pgsql/socket_path</i>	データベースソケットディレクトリを保護します (ソケットパスが共有ファイルシステム上にある場合)。

フェイルオーバーが発生した場合、LifeKeeper はバックアップサーバでファイルシステム、IP アドレス、およびデータベースリソース(すべてのリソース依存関係を含む)を In Service の状態にします。クライアントは切断され、サーバに再接続する必要があります。まだコミットされていない SQL ステートメントはすべて再入力する必要があります。

要件

PostgreSQL Recovery Kit の要件

LifeKeeper for Linux PostgreSQL Recovery Kit をインストールする前に、LifeKeeper 設定が次の要件を満たしている必要があります。LifeKeeper のハードウェアとソフトウェアのインストールと設定に関する具体的な手順については、SPS for Linux インストールガイドを参照してください。

[ハードウェア要件](#)

[ソフトウェア要件](#)

ハードウェア要件

サーバ-サーバは、要件にしたがって設定する必要があります。要件は、SPS for Linux テクニカルドキュメンテーション、および SPS for Linux リリースノートに記載されています。

IP ネットワークインターフェースカード - 各サーバは、イーサネット TCP/IP をサポートするネットワークインターフェースカードを 1 つ以上必要とします。ただし、LifeKeeper クラスタのベストプラクティスでは 2 つ以上のコミュニケーションパスが必要であることを注意してください。LAN を使用した 2 つのコミュニケーションパスは、ハートビート用に別々のサブネットに接続することをお勧めします。また、少なくともどちらか 1 つのサブネットは、プライベートネットワークとして設定します。TCP と TTY のハートビートの組み合わせもサポートされています。

ソフトウェア要件

- **TCP/IP ソフトウェア** - LifeKeeper 環境の各サーバには TCP/IP ソフトウェアが必要です。
- **PostgreSQL ソフトウェア** - クラスタ内の全サーバに必ず同じバージョンの PostgreSQL ソフトウェアをインストールします。PostgreSQL ソフトウェアは、<http://www.postgresql.org/download> に記載されているいずれかのミラーサイトからダウンロードできます。
- **LifeKeeper ソフトウェア** - クラスタ内の各サーバにインストールする LifeKeeper ソフトウェアのバージョン、および、適用する LifeKeeper ソフトウェアパッチのバージョンが同一であることが絶対条件です。
- **LifeKeeper for Linux PostgreSQL Recovery Kit** - PostgreSQL Recovery Kit は、FTP ダウンロードにより SPS for Linux インストールイメージファイル (*sps.img*) で提供されます。Red Hat Package Manager (rpm) でパッケージされており、rpm を使用してインストール/アンインストールを行います。

steeleye-ikPGSQL

Chapter 2: 設定上の考慮事項

本セクションでは、PostgreSQL Recovery Kit の設定と管理を開始する前に考慮すべき事項について説明します。

[DataKeeper でのミラーファイルシステムの使用](#)

[PostgreSQL の保護: ベストプラクティス](#)

PostgreSQL の保護: ベストプラクティス

アクティブ/スタンバイ設定では、バックアップサーバは PostgreSQL をアクティブに実行せず、プライマリサーバに障害が発生した場合に備えてスタンバイします。アクティブ/アクティブ設定では、障害時、他のサーバに対するバックアップとして各サーバがアクティブに PostgreSQL インスタンスを実行します。以下に、アクティブ/スタンバイもしくはアクティブ/アクティブ設定で PostgreSQL リソースインスタンスを保護するときに従わなければならない具体的な要件を示します。

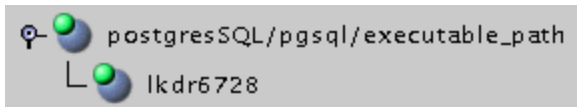
1. PostgreSQL の DataDir および Write-Ahead-LogPath (*PGDATA/pg_xlog*) を 1 つ以上の共有ファイルシステムにインストールしてください。DataDir および WAL-Path は、リソースインスタンスを保護するすべてのサーバ間で共有してください。
 - PostgreSQL オペレーティングシステムユーザは、データディレクトリと、Write-Ahead-LogPath を含むディレクトリを所有する必要があります。
 - PostgreSQL データベースは、*initdb* ユーティリティを使用して作成してください。*initdb* ユーティリティは、PostgreSQL オーナとして、*-D <datadir>* オプションを使用して実行する必要があります。
 - デフォルトの PostgreSQL インスタンスの自動スタートアップは無効にするか、またはデフォルトの PostgreSQL インスタンスが LifeKeeper で使用するポート以外のポートで稼働するように制限する必要があります。
 - LifeKeeper で保護する PostgreSQL インスタンスの自動スタートアップは無効にする必要があります。LifeKeeper は、保護するインスタンスの開始と停止を制御します。
 - PostgreSQL インスタンスは、階層の作成に先だって手動で開始する必要があります。インスタンスは、*pg_ctl* ユーティリティにバックエンドオプション *-o "-p <port>"* を指定して開始してください。
2. *StartupLogPath*、*SocketPath*、*ExecutablePath* は、プライマリサーバの任意の共有ファイルシステム上か、各クラスタノードのローカルファイルシステム上に配置することができます。
 - PostgreSQL オペレーティングシステムユーザはソケットパスを含むディレクトリを所有する必要があります。
 - PostgreSQL オペレーティングシステムユーザは *StartupLogPath* を含むディレクトリでの書き込み権限を所有する必要があります。

3. アクティブ / スタンバイもしくはアクティブ / アクティブのどちらかで複数のインスタンスを起動する場合は、各インスタンスに一意のポートおよびソケットパスを割り当ててください。

DataKeeper でのミラーファイルシステムの使用

PostgreSQL Recovery Kit では、SIOS DataKeeper を共有ファイルシステムとして使用できます。ミラーファイルシステムは、PostgreSQL インストールパス、ログパス、データディレクトリ、および実行可能パスに使用できます。

例えば、PostgreSQL リソースの依存ファイルシステムは、データディレクトリのファイルシステムとその依存関係を示す、以下の SDR リソースミラーと似ています。



Chapter 3: インストール

LifeKeeper での PostgreSQL のインストール / 設定

PostgreSQL データベースとLifeKeeper ソフトウェアのインストールと設定には以下の手順をお勧めします。各ステップは、詳細作業にリンクしています。

1. [PostgreSQL ソフトウェアのインストール](#)
2. [PostgreSQL データベースの作成](#)
3. [LifeKeeper Core と PostgreSQL Recovery Kit のインストール](#)
4. [PostgreSQL リソースのための LifeKeeper 設定項目](#)

これらの作業を実行すると、LifeKeeper リソース階層を作成して PostgreSQL データベースを保護できるようになります。

リソース設定作業

前のセクションで説明したセットアップ作業を終了したら、PostgreSQL リソース階層を作成して拡張できるようになります。

LifeKeeper for Linux PostgreSQL Recovery Kit の設定に関して、以下の作業を行うことができます。

- [Create Resource Hierarchy](#) - PostgreSQL リソース階層を作成します。
- [Delete Resource Hierarchy](#) - PostgreSQL リソース階層を削除します。
- [Extend Resource Hierarchy](#) - PostgreSQL リソース階層をプライマリサーバからバックアップサーバに拡張します。
- [Unextend Resource Hierarchy](#) - LifeKeeper クラスタ内の 1 つのサーバから PostgreSQL リソース階層を拡張解除 (削除) します。
- [Viewing PostgreSQL Configuration Settings](#) - Resource Properties ダイアログを参照します。

LifeKeeper Core のリソース階層 (ファイルシステムや IP リソースなど) の設定方法については、SPS for Linux テクニカルドキュメンテーションの管理作業 セクションを参照してください。

以下の作業については、すべての Recovery Kit で同じ手順を使用する共通の作業であるため、SPS for Linux テクニカルドキュメンテーションの管理 セクションを参照してください。

- リソース依存関係の作成 - 既存のリソース階層と別のリソースインスタンスとの間に親子の依存関係を作成し、クラスタ内のすべての対象サーバに依存関係の変更を反映します。

アップグレード

- リソース依存関係の削除 - リソースの依存関係を削除して、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- リソースを In Service にする - リソース階層を特定のサーバで In Service の状態にします。
- リソースを Out of Service にする - リソース階層を特定のサーバで Out of Service にします。
- リソースのプロパティの表示 / リソースのプロパティの編集 - 特定のサーバでリソース階層のプロパティを表示または編集します。

注記: このセクションでの設定作業は **[Edit]** メニューを使用して実行します。この設定作業の大半は次の方法でも実行できます。

- ツールバーから
- 状況表示の左側ペインにあるグローバルリソースを右クリックする
- 状況表示の右側ペインにあるリソースを右クリックする

右クリックの方法を使用すると、Edit メニューを使用するときに要求される情報を入力せずに済みます。

アップグレード

[PostgreSQL Recovery Kit の旧バージョンからのアップグレード](#)

PostgreSQL ソフトウェアのインストール

同一のパラメータ / 設定を使用してクラスタ内の全サーバに PostgreSQL ソフトウェアをインストールする必要があります。詳細については [PostgreSQL Administration Guide](#) を参照してください。以下は、LifeKeeper を PostgreSQL と連携させるための追加の推奨事項および注意事項です。

- PostgreSQL クライアントソフトウェアパッケージがインストールされている必要があります。これらのパッケージには、PostgreSQL `psql` クライアントユーティリティが含まれていなければなりません。
- PostgreSQL サーバソフトウェアパッケージがインストールされている必要があります。これらのパッケージには、PostgreSQL `pg_ctl` および `initdb` ユーティリティが含まれていなければなりません。
- PostgreSQL クライアントパッケージとサーバパッケージのバージョンは、全サーバで同じでなければなりません。
- 全サーバに、PostgreSQL オペレーティングシステムユーザが以下のように存在している必要があります。
 - PostgreSQL オペレーティングシステムユーザを PostgreSQL ソフトウェアのインストールとサブディレクトリの所有者として指定します。
 - PostgreSQL オペレーティングシステムユーザには `pg_ctl` ユーティリティを使用する権限が必要です。PostgreSQL オペレーティングシステムユーザは `pg_util` コマンドを使用してポストマスターインスタンスを開始および停止できなければなりません。
 - PostgreSQL オペレーティングシステムユーザ名には英数字 (a-z, A-Z, 0-9) のみを使用する必要があります。

- PostgreSQL オペレーティングシステムユーザのユーザ ID とグループ ID は、全サーバで同じでなければなりません。
- PostgreSQL データベース管理者ユーザは、`psql` ユーティリティを使用した LifeKeeper クライアント接続のため、PostgreSQL のデータベース内に存在している必要があります。
 - また PostgreSQL データベース管理者ユーザは、データベース (`template1`) への接続権限を有している必要があります。またインスタンスに定義されたデータベースのリストを取得できる必要があります。
 - この PostgreSQL データベース管理者ユーザは、システムテーブルの閲覧権限および一般的なクエリを作成できる権限を保有している必要があります。
 - PostgreSQL データベース管理者ユーザは PostgreSQL オペレーティングシステムユーザと同じ名前を持つことができますが、異なるものです。
 - 例: PostgreSQL オペレーティングシステムユーザ=`postgres` で PostgreSQL データベース管理者ユーザ=`lkpostgres`、もしくは、PostgreSQL オペレーティングシステムユーザ=`postgres` で PostgreSQL データベース管理者ユーザ=`postgres`。

PostgreSQL データベースの作成

データベースを作成するには、『[PostgreSQL Administration Guide](#)』の手順に従います。さらに次の推奨事項にも留意してください。

- PostgreSQL データディレクトリは、`initdb` ユーティリティを使用し、`-D <data dir>` オプションを指定して初期化する必要があります。`initdb` コマンドは PostgreSQL オペレーティングシステムユーザとして実行する必要があります。
- PostgreSQL インスタンスデータディレクトリは共有ファイルシステムに存在している必要があります。
- PostgreSQL トランザクションログディレクトリは共有ファイルシステム上に存在している必要があります。
- PostgreSQL データベース名には英数字のみを使用する必要があります。
- データベースを作成したら、PostgreSQL データベースインスタンスの自動スタートアップを無効にする必要があります。LifeKeeper の保護下に入ったら、LifeKeeper がデータベースの開始と停止を行います。
- PostgreSQL インスタンスは、階層の作成に先だって手動で開始する必要があります。インスタンスは、`pg_ctl` ユーティリティにバックエンドオプション `-o "-p <port>"` を指定して開始してください。

パスワードの保護を無効にする場合 (インスタンスへの接続に、パスワード保護を使用しない)

- PostgreSQL データベースインスタンスがパスワード保護されない場合や、PostgreSQL の管理ユーザからローカルクライアント接続にパスワードを必要としない場合は、エントリがローカルトラスト接続を許可した状態で存在している必要があります。以下は PostgreSQL のデータベース管理者ユーザのローカルクライアント接続を有効にするために `pg_hba.conf` に記述するエントリのサンプルです。

```
=====  
.  
.  
Local all postgres trust  
.
```

```
=====
```

パスワード保護を有効にする(インスタンスへの接続にパスワードを使用)

- パスワードで保護されるデータベースインスタンスは、リソースが保護されるクラスタの各サーバ上の .pgpass 認証ファイルに存在するための PostgreSQL データベース管理ユーザに対するパスワードエントリが必要になります。pgpass ファイルはパスワードを要求する各 PostgreSQL データベース管理ユーザに対して有効でテスト済みのエントリを保有している必要があります。
- .pgpass ファイルは PostgreSQL オペレーティングシステムユーザのホームディレクトリに配置してください。ファイルへのアクセス権を制限するために適切な権限を設定してください。
- 以下は、<hostname>:<port>:<database>:<user>:<password>のフォーマットで示した有効な .pgpass ファイルの一例です。

```
=====
```

```
*:5443:*:lifekeeper:jh43tmp2009
```

```
=====
```

注記: .pgpass ファイルは、ユーティリティ `psql` による自動の(非ターミナルまたはスクリプト)接続のために必要です。pgpass ファイルは、パスワード保護されたインスタンスが保護される各サーバ上に存在している必要があります。

LifeKeeper ソフトウェアのインストール

PostgreSQL ソフトウェアをインストールし、データベースを作成したら、LifeKeeper Core ソフトウェアと必要なパッチをインストールし、続いて PostgreSQL Recovery Kit をインストールできます。

LifeKeeper パッケージのインストール方法については、SPS for Linux インストールガイドを参照してください。

PostgreSQL リソースのための LifeKeeper 設定項目

PostgreSQL Recovery Kit には、特定のシナリオでリソース保護をカスタマイズできるように、設定項目のための環境変数が用意されています。これらの変数の値を変更するには、ファイル `/etc/default/LifeKeeper` を編集します。新しい設定を反映するためにプロセスを再起動する必要はありません。デフォルト値は PostgreSQL Recovery Kit がインストールされるほとんどの環境で動作する値となっています。

• LKPGSQL_CONN_RETRIES

この設定項目は、データベースが開始するまで PostgreSQL Recovery Kit が待機する時間を制御します。この時間は、「(LKPGSQL_CONN_RETRIES * 5) = データベースインスタンスの開始まで待機する合計時間(秒単位)」の式を用いて Recovery Kit により計算されます。この変数の設定は、リソースの In Service 要求とリソースのローカルリカバリの両方に影響します。

• LKPGSQL_DISCONNECT_CLIENT

この設定項目は、postmaster がクラッシュした場合にアクティブなクライアントが切断されるかどうかを制御します。値を 1 (true) に設定すると、アクティブなクライアントはリソースのローカルリカバリが進行している間は切断されません。値を 0 (false) に設定すると、アクティブなクライアントはリソースのローカルリカバリが進行している間も切断されません。この変数は、リソースのローカルリカバリイベントのみに影響し、postmaster プロセスが稼働していない場合のローカルリカバリイベントの場合にのみ適用されます。

• LKPGSQL_SDIRS

この設定項目は、PostgreSQL データベースをシャットダウンする際のクライアント切断動作を制御します。カンマで区切られた設定項目をデフォルトファイルに追加する必要があります。このオプションを設定することにより、指定されたリソースインスタンスもしくは保護されるデータディレクトリと一致するインスタンスが、クライアントに対してシャットダウン時の切断を要求しません。

```
LKPGSQL_SDIRS=/protected/pgsql-datadir
```

```
LKPGSQL_SDIRS=/protected/pgsql-datadir, /otherprotected/pgsql-datadir
```

/protected/pgsql-datadir および */otherprotected/pgsql-datadir* は、LifeKeeper の保護下の PostgreSQL データディレクトリです。

注記:このオプション LKPGSQL_SDIRS および LKPGSQL_IDIRS は排他的なオプションです。LKPGSQL_SDIRS もしくは LKPGSQL_IDIRS の値は、階層の作成中に選択した保護対象の datadir の値と一致していなければなりません。

• LKPGSQL_IDIRS

この設定項目は、PostgreSQL データベースをシャットダウンする際のクライアント切断動作を制御します。カンマで区切られた設定項目をデフォルトファイルに追加する必要があります。このオプションを設定することにより、指定されたリソースインスタンスもしくは保護されるデータディレクトリと一致するインスタンスが、クライアントに対してシャットダウン時の即時切断を強制します。

```
LKPGSQL_IDIRS=/protected/pgsql-datadir
```

```
LKPGSQL_IDIRS=/protected/pgsql-datadir, /otherprotected/pgsql-datadir
```

/protected/pgsql-datadir および */otherprotected/pgsql-datadir* は、LifeKeeper の保護下の PostgreSQL データディレクトリです。

注記:このオプション LKPGSQL_SDIRS および LKPGSQL_IDIRS は排他的なオプションです。LKPGSQL_SDIRS もしくは LKPGSQL_IDIRS の値は、階層の作成中に選択した保護対象の datadir の値と一致していなければなりません。

PostgreSQL リソース階層の作成

プライマリサーバで次の手順を実行します。

1. [Edit] メニューの [Server] から [Create Resource Hierarchy] を選択します。

[Create Resource Wizard] ダイアログボックスが表示されます。

2. ドロップダウンリストから **[PostgreSQL Database]** を選択し、**[Enter]** をクリックします。
3. 以下の情報を入力するように要求されます。ダイアログボックスで **[Back]** ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、エラーが発生して、前に入力した情報を修正する必要がある場合に便利な機能です。いつでも **[Cancel]** をクリックして、作成処理全体を取り消すことができます。

フィールド	ヒント
Switchback Type	[intelligent] または [automatic] を選択します。フェイルオーバー後、バックアップサーバで PostgreSQL リソースが In Service (アクティブ) になるとき、プライマリサーバにスイッチバックされる方法が決まります。インテリジェントスイッチバック (intelligent) の場合、リソースをプライマリサーバにスイッチバックするときに管理者の介入が必要ですが、自動スイッチバック (automatic) の場合は、プライマリサーバがオンラインに戻り、LifeKeeper のコミュニケーションパスが再確立されるとすぐにスイッチバックが実行されます。 注記: スwitchバック方法は、PostgreSQL リソースと依存関係を構築しているリソースのスイッチバック方法と一致する必要があります。
PostgreSQL Executable Location	このフィールドは、PostgreSQL 実行可能ファイルを含むディレクトリパスを指定するために使用されます。パス名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ ./
PostgreSQL Client Executable Location	このフィールドは、PostgreSQL 実行可能ファイル <code>psql</code> を含むディレクトリパスを指定するために使用されます。パス名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ ./
PostgreSQL Administration Executable Location	このフィールドは、PostgreSQL 実行可能ファイル <code>pg_ctl</code> を含むディレクトリパスを指定するために使用されます。パス名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ ./
PostgreSQL Data Directory	このフィールドは、LifeKeeper の保護下に置かれる PostgreSQL データディレクトリ (datadir) の場所を指定するために使用されます。指定されたディレクトリは、共有ファイルシステムに存在し、常駐している必要があります。パス名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ ./
PostgreSQL Port	このフィールドは、ポスト マスタデーモンがクライアントアプリケーションからの接続をリスニングする TCP/IP ポート番号を指定するために使用されます。
PostgreSQL Socket Path	このフィールドは、ポスト マスタデーモンがクライアントアプリケーションからの接続をリスニングする UNIX ドメインソケットへのフルパスを指定するために使用されます。パス名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ ./
PostgreSQL Database Administrator User	このフィールドは、指定したデータベースインスタンスとそのインスタンスへの接続および管理者権限に対する PostgreSQL データベース管理ユーザ名を指定するために使用されます。
PostgreSQL Logfile	このフィールドは PostgreSQL のログファイルに使用されるログファイルパスを指定するために使用されます。

フィールド	ヒント
PostgreSQL Database Tag	プライマリサーバ上の新しい PostgreSQL データベースリソースに対する固有のタグ名です。デフォルトのタグ名は「pgsql」の後にデータベースインスタンスのポート番号を付けたものです。別の固有のタグ名を入力することもできます。タグ名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ /

4. **[Create]** をクリックします。**[Create Resource Wizard]** によって、PostgreSQL リソース階層が作成されます。LifeKeeper によって入力データが検査されます。問題が検出された場合、情報ボックスにエラーメッセージが表示されます。
5. PostgreSQL リソース階層が正常に作成されたこと、フェイルオーバーによる保護を実現するにはその階層をクラスタ内の別のサーバに拡張する必要があることを示すメッセージが表示されます。**[Next]** をクリックします。
6. **[Continue]** をクリックします。**[Pre-extend Wizard]** が起動されます。リソース階層を別のサーバに拡張する方法の詳細については、[PostgreSQL リソース階層の拡張の手順 2](#) を参照してください。

PostgreSQL リソース階層の削除

LifeKeeper 設定内のすべてのサーバから PostgreSQL リソース階層を削除するには、次の手順を実行します。

1. **[Edit]** メニューの **[Resource]** から **[Delete Resource Hierarchy]** を選択します。
2. **[Target Server]** で、PostgreSQL リソース階層を削除するサーバの名前を選択します。
注記: 左側ペインのグローバルリソースまたは右側ペインの個々のリソースインスタンスを右クリックして **[Delete Resource]** 作業を選択した場合、このダイアログボックスは表示されません。
3. **[Hierarchy to Delete]** を選択します。(このダイアログは、右ペインまたは左ペインのリソースインスタンス上で右クリックして **[Delete Resource]** を選択した場合には表示されません。)**[Next]** をクリックします。
4. 選択したターゲットサーバと、削除の対象として選択した階層を確認する情報ボックスが表示されます。**[Next]** をクリックします。
5. PostgreSQL リソースが正常に削除されたことを確認する別の情報ボックスが表示されます。
6. **[Done]** をクリックして終了します。

PostgreSQL リソース階層の拡張

この操作は **[Edit]** メニューから開始できます。または **[Create Resource Hierarchy]** の処理を完了すると自動的に開始されます。その場合は、下記の **手順 2** を参照してください。

1. **[Edit]** メニューの **[Resource]** から **[Extend Resource Hierarchy]** を選択します。**[Pre-Extend Wizard]** が表示されます。拡張操作に慣れていない場合は、**[Next]** をクリックします。LifeKeeper のリソース階層の拡張作業のデフォルト値が分かっている、入力と確認を省略する場合は、**[Accept Defaults]** をクリックします。
2. **[Pre-Extend Wizard]** で以下の情報を入力します。

注記:最初の2つのフィールドは **[Edit]** メニューの **[Extend]** から操作を開始した場合にだけ表示されません。

フィールド	ヒント
Template Server	PostgreSQL リソースが現在 In Service のサーバを選択します。
Tag to Extend	拡張する PostgreSQL リソースを選択します。
Target Server	拡張先のサーバを入力または選択します。
Switchback Type	フェイルオーバー後、バックアップサーバで PostgreSQL リソースが In Servic (アクティブ) になるとき、プライマリサーバにスイッチバックされる方法が決まります。 [intelligent] または [automatic] を選択できます。スイッチバックタイプは、必要な場合 [Resource Properties] ダイアログボックスの [General] タブで後から変更できます。 注記: スイッチバック方法は、PostgreSQL リソースと依存関係を構築しているリソースのスイッチバック方法と一致する必要があることに注意してください。
Template Priority	テンプレートの優先順位 を選択または入力します。これはサーバで現在 In Servic の PostgreSQL 階層の優先順位です。優先順位は、1 ~ 999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります(数字 1 が最高の優先順位に相当します)。拡張処理時に、別のシステムですでに使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を推奨します。 注記: このフィールドは階層を最初に拡張するときだけ表示されます。
Target Priority	これは、別サーバにある同等の階層に対する、新しく拡張する PostgreSQL 階層の優先順位です。1 ~ 999 の範囲の未使用の値が有効で、リソースのカスケーディングフェイルオーバーシナリオにおけるサーバの優先順位を示します。LifeKeeper のデフォルトでは、階層が作成されたサーバに「1」が割り当てられることに注意してください。優先順位は連続している必要はありませんが、特定のリソースについて2つのサーバに同じ優先順位を割り当てることはできません。

3. 拡張前のチェックが正常に終了したというメッセージが表示されたら、**[Next]** をクリックします。
4. 拡張する階層に応じて、拡張されるリソースタグ(一部編集不可)を示す一連の情報ボックスが表示されます。
5. **[Extend Wizard]** で以下の情報を入力します。

フィールド	ヒント
PostgreSQL Executable Location	このフィールドは、PostgreSQL 実行可能ファイルを含むディレクトリパスを指定するために使用されます。パス名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ . /
PostgreSQL Database Tag	プライマリサーバ上の新しい PostgreSQL データベースリソースに対する固有のタグ名です。デフォルトのタグ名は語「pgsql」の後にデータベースインスタンスのポート番号を付けたものです。別の固有のタグ名を入力することもできます。タグ名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ . /

6. 「Hierarchy extend operations completed」というメッセージが表示されたら、階層を別のサーバに拡張する場合は、**[Next Server]** をクリックし、これ以上拡張作業が必要ない場合は、**[Finish]** をクリックします。
7. 「Hierarchy Verification Finished」のメッセージが表示されたら、**[Done]** をクリックします。

PostgreSQL リソース階層の拡張解除

LifeKeeper クラスタ内にある 1 つのサーバからリソース階層を削除するには、次の手順を実行します。

1. **[Edit]** メニューの **[Resource]** から **[Unextend Resource Hierarchy]** を選択します。
2. **[Target Server]** で、PostgreSQL リソースを拡張解除するサーバを選択します。リソースが現在 In Service のサーバは選択できません。(このダイアログボックスは、右側のペインでリソースインスタンスを右クリックして、拡張解除作業を選択した場合には、表示されません。)**[Next]** をクリックします。
3. 拡張解除する PostgreSQL 階層を選択し、**[Next]** をクリックします。(このダイアログボックスは、いずれかのペインでリソースインスタンスを右クリックして、拡張解除作業を選択した場合には、表示されません。)
4. ターゲットサーバと拡張解除の対象として選択した PostgreSQL リソース階層を確認する情報ボックスが表示されます。**[Unextend]** をクリックします。
5. PostgreSQL リソースが正常に拡張解除されたことを確認する別の情報ボックスが表示されます。**[Done]** をクリックして、**[Unextend Resource Hierarchy]** メニューを終了します。

PostgreSQL 設定の参照

リソースプロパティダイアログは **[Edit]** メニューもしくはリソースコンテキストメニューから参照可能です。このダイアログはサーバ上の特定のリソースに対するプロパティを表示します。**[Edit]** メニューからアクセスした場合は、リソースおよびサーバを選択することができます。リソースコンテキストメニューからアクセスする場合はサーバを選択することができます。

[Configuration] タブからは以下の PostgreSQL 設定を参照することができます。

- Executable Path
- Client Executable Name
- Admin Executable Name
- Bind Setting
- Startup Log Location
- PostgreSQL Operating System User Name
- PostgreSQL Database Administrator User
- Version Number
- Data Directory
- Socket Location

- Port Number
- OS Daemon Name

PostgreSQL Recovery Kit の旧バージョンからのアップグレード

旧バージョンの SPS for Linux の PostgreSQL ソフトウェアからのアップグレード時、既存の SPS PostgreSQL リソースのインスタンスは変更されます。SPS ソフトウェアがサーバ上でアップデートされる際に、次の保持される値が、SPS 内部の情報のフィールドに自動的に追加されます。

- **Client Executable Location (*psql*)** - 保護対象のデータベースインスタンスへの接続に使用される *psql* もしくは同等のクライアントユーティリティの場所。アップグレード後、この値は LifeKeeper GUI のプロパティ表示で確認することが可能です。また、この値は LifeKeeper コマンドラインの *set_value* ユーティリティを使用して確認することも可能です。

set_value は、LifeKeeper PostgreSQL Recovery Kit が内部のリソース情報の項目の値をアップデートするために提供される LifeKeeper ユーティリティの名前です。このユーティリティの使用は本トピックで説明されている問題、もしくは SIOSTechnology Corp. 社のサポートチームによる依頼および指示に限るものとします。

注記: *set_value* ユーティリティは、正確なエラーチェックは行わないので、一般的な使用には向いていません。

- **Administration Executable Location (*pg_ctl*)** - 保護対象のデータベースインスタンスの起動、停止、ステータス確認に使用する *pg_ctl* または同等の管理ユーティリティの場所です。アップグレード後、この値は LifeKeeper GUI のプロパティ表示で確認することが可能です。また、この値は LifeKeeper コマンドラインの *set_value* ユーティリティを使用して確認することも可能です。
- **PostgreSQL Database Administrator User** - LifeKeeper が保護するインスタンスの PostgreSQL データベース管理者ユーザです。このユーザは保護対象のデータベースインスタンスへの接続および管理権限を持っている必要があります。デフォルト値は、PostgreSQL のデータディレクトリを保持している PostgreSQL オペレーティングシステムユーザになります。アップグレード後、この値は LifeKeeper GUI のプロパティ表示で確認することが可能です。また、この値は SPS コマンドラインの *set_value* ユーティリティを使用して確認することも可能です。
- **PostgreSQL Daemon Name (*postmaster*)** - バックエンドで起動しているデーモン名です。この値は、データベースの最初のステータス確認時に決定されます。デフォルト値は *postmaster* です。アップグレード後、この値は LifeKeeper GUI のプロパティ表示で確認することが可能です。また、この値は LifeKeeper コマンドラインの *set_value* ユーティリティを使用して確認することも可能です。
- **Default Test Database (*template1*)** - データベースインスタンスが基本的な接続確認を行っている間に LifeKeeper が使用するデータベースです。アップグレード後、デフォルトのテストデータベースは *template1* に設定されます。
- **PostgreSQL Maximum Monitoring Hangs ([LKPGSQL_QCKHANG_MAX](#))** - この設定は、リストアやリカバリのためのフェイルオーバーアクションが開始される前に、無制限な数の接続がハングすることからの保護を提供します。PostgreSQL Recovery Kit の監視の一部は、保護されたデータベースへの接続を必要とします。どれだけの数の接続のハングを許容するかは、リソース作成時に [LKPGSQL_QCKHANG_MAX](#) によって決定されます。8.1.2 以前のバージョンでは、デフォルトの値は 15 でした。バージョン 8.1.2 (またはそれ以降のバージョン) へのアップグレード後は、デフォルトの値は 2 です。この値はリソース作成時にリソース情報と紐付けて保存されますので、手動でアップデートしない限り、バージョン

8.1.2 にアップグレードする前に作成されたリソースはいずれもデフォルト値 15 のままになります。バージョン 8.1.2 (またはそれ以降のバージョン) で作成されたリソースはデフォルト値の 2 となります。値は [set value ユーティリティ](#) を使用して確認することもできます。



重要な注記: SPS for Linux PostgreSQL Recovery Kit ソフトウェアをアップグレードした後、プライマリサーバからバックアップサーバへリソースインスタンスのフェイルオーバーをシミュレーションするために手動でスイッチオーバーを行い [PostgreSQL リソース階層のテスト](#) を行ってください。

アップグレードに関する重要事項	<p>アップグレードを行った後、リソースを起動できない場合は、以下の事項を確認してください:</p> <p>Client Executable name is not found or incorrect (クライアントの実行可能ファイル名が見つからない、または不正です)</p> <p>この値は <code>set_value</code> ユーティリティを使用してアップデートできます。クライアントの実行可能ファイルをアップデートする構文は以下の通りです。</p> <pre>/opt/LifeKeeper/lkadm/subsys/database/pgsql/bin/set_value <tag> 'clientexe' <full path to the psql utility></pre> <p>例:</p> <pre>/opt/LifeKeeper/lkadm/subsys/database/pgsql/bin/set_value pgsq-5443 'clientexe' '/pgsql/clientutils/psql'</pre> <p>Administration Executable name is not found or incorrect (管理実行可能ファイル名が見つからない、または不正です)</p> <p>この値は <code>set_value</code> ユーティリティを使用してアップデートできます。管理実行可能ファイルをアップデートする構文は以下の通りです。</p> <pre>/opt/LifeKeeper/lkadm/subsys/database/pgsql/bin/set_value <tag> 'osexex' <full path to the pg_ctl utility></pre> <p>例:</p> <pre>/opt/LifeKeeper/lkadm/subsys/database/pgsql/bin/set_value pgsq-5443 'osexex' '/pgsql/adminutils/pg_ctl'</pre>
------------------------	---

複数のハングイベントからの復旧までのインターバルの短縮 ([LKPGSQL_QCKHANG_MAX](#))

8.1.2 以前のバージョンでは、Maximum Monitoring Hangs の値が大きすぎました。

既存の PostgreSQL リソースインスタンスの Maximum Monitoring Hangs の値は、set_value ユーティリティを使用して見る、またはセットすることができます。

Maximum Monitoring Hangs ([LKPGSQL_QCKHANG_MAX](#)) の値をセットする構文は以下の通りです。

```
/opt/LifeKeeper/lkadm/subsys/database/pgsql/bin/set_value <tag> 'hangmax' <number>.
```

例:

```
/opt/LifeKeeper/lkadm/subsys/database/pgsql/bin/set_value pgsq1-5443 'hangmax' 3.
```

注記: クラスタ内の全てのノードで値を更新するためには、-c オプションを付与してください。(set_value -c <tag>...).

この値を確認する構文は以下の通りです。

```
/opt/LifeKeeper/lkadm/subsys/database/pgsql/bin/set_value -l <tag> 'hangmax'
```

例:

```
/opt/LifeKeeper/lkadm/subsys/database/pgsql/bin/set_value -l pgsq1-5443 'hangmax'
```


Chapter 4: 管理

[データベース管理者ユーザの更新](#)

[ユーザの更新](#) オプションにより、LifeKeeper の管理者は、LifeKeeper PostgreSQL リソースインスタンスに対する現在の PostgreSQL データベース管理者ユーザを変更することができます。

PostgreSQL リソース階層のテスト

PostgreSQL リソース階層のテストは、手動でサーバをスイッチオーバーすることで、プライマリサーバからバックアップサーバへのリソースインスタンスのフェイルオーバーをシミュレートして実行できます。

[LifeKeeper GUI からの手動スイッチオーバーの実行](#)

EnterpriseDB Postgres Plus Advanced Server 環境

[EnterpriseDB Postgres Plus Advanced Server の保護](#)

LifeKeeper GUI からの手動スイッチオーバーの実行

LifeKeeper GUI で **[Edit] > [Resource] > [In Service]** を選択すると、手動スイッチオーバーを開始できます。例えば、バックアップサーバで **[In Service]** 要求が実行されると、PostgreSQL リソース階層がバックアップサーバ側で In Service になり、プライマリサーバ側では Out of Service になります。この時点で、元のバックアップサーバがプライマリサーバに、元のプライマリサーバがバックアップサーバに変わります。

[Out of Service] 要求を実行した場合、リソース階層は Out of Service になりますが、別のサーバで In Service にはなりません。

重要: バックアップサーバでリソース階層を In Service にしたら、データベースに接続できるかどうか試す必要があります。パスワード保護されたインスタンスに関しては、`.pgpass` ファイルがバックアップ上で有効であることが重要です。`.pgpass` ファイルが有効であるか確認するためには、データベースへのクライアント接続が `psql` ユーティリティと PostgreSQL データベース管理者ユーザの両方を使用して実行できることを確認してください。パスワード認証を要求されなければ、接続が有効な `.pgpass` ファイルが存在していることが分かります。

EnterpriseDB Postgres Plus Advanced Server の保護

EnterpriseDB Postgres Plus Advanced Server リソースを保護するにあたって、追加の LifeKeeper 設定は必要ありません。

問題	解決方法
EnterpriseDB Postgres Plus Advanced Server のインストール中に、 [Configuration Mode] ダイアログでオプションの [PostgreSQL-compatible defaults and samples] を選択した場合は、LifeKeeper が使用する「edb」データベースが作成されません。	<p>「createdb」ユーティリティを使用して「edb」データベースを手動で追加してください。</p> <pre>createdb -p <port> -h <socket path> edb</pre> <p>コマンドは PostgreSQL オペレーティングシステムユーザとして実行する必要があります。以下はサンプルです。</p> <pre>su - postgres postgres@server1 ~>createdb -p 5435 -h /var/lib/postgres edb</pre>

データベース管理者ユーザの更新

ユーザの更新 オプションは、保護されたリソースが存在する全システム上の PostgreSQL データベース管理者ユーザに対して保存された値を更新します。**[Update User]** オプションは **LifeKeeper のリソースツールバー** もしくは **LifeKeeper のリソースコンテキストメニュー** から実行することが可能です。

PostgreSQL データベース管理者ユーザを更新するためには、プライマリサーバで以下の手順を実行してください。

注記: **[Update User]** メニューおよびツールバーオプションは out of service のリソースに対して無効になります。

1. ツールバーの **[Update User]** アイコンを選択するか、リソースコンテキストメニューから **[Update User]** を選択してください。

[Update User Wizard] のダイアログが表示されます。

2. 以下の情報を入力するように要求されます。ダイアログボックスで **[Back]** ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、エラーが発生して、前に入力した情報を修正する必要がある場合に便利な機能です。いつでも **[Cancel]** をクリックして、作成処理全体を取り消すことができます。

フィールド	ヒント
[PostgreSQL Database Administrator User] と入力	<p>このダイアログは、指定したデータベースインスタンスとそのインスタンスへの接続および管理者権限に対する PostgreSQL データベース管理者ユーザ名を指定するために使用されます。</p> <p>注記: 認証スクリプトは、指定した値を使用して接続確認を行います。パスワード保護されたインスタンスは PostgreSQL データベース管理者ユーザの .pgpass ファイルの有効なエントリを要求します。</p>
Confirm Update Action	このダイアログでは、以前のユーザから新しいユーザに対して行う変更について確認をします。

3. **[Update]** をクリックしてください。現在保護されているリソースがあるすべてのサーバ上で PostgreSQL データベース管理者ユーザが更新されます。

Chapter 5: トラブルシューティング

本セクションでは、SPS PostgreSQL リソース階層の作成時や拡張時、またはリソースの削除時やリストア時に表示される可能性のあるメッセージの一覧を示します。必要に応じて、エラーの原因およびエラー状態を解消するために必要な処置についても説明しています。

他の SPS コンポーネントからメッセージが出されることもあります。そのような場合は、メッセージカタログ (場所は、SIOS テクニカルドキュメンテーションのサイトの「エラーコードの検索」の下) を参照してください。メッセージカタログには、操作、管理、GUI など、SIOS Protection Suite for Linux を使用しているときに遭遇する可能性がある、すべてのエラーコードが列挙されています。また、エラーコードの原因に関する追加の説明や、問題解決のために必要な処置についても、必要に応じて記載されています。この完全なリストを検索すると、受信したエラーコードを見つけることができます。また、関連する SPS コンポーネントの個別のメッセージカタログに直接アクセスすることもできます。

- [全般項目](#)
- [設定項目](#)

全般項目

以下のエラーメッセージおよび状態は PostgreSQL Recovery Kit 使用時に表示される可能性があります。

エラー	解決方法
Unable to protect PostgreSQL database using the same port as another LK protected PostgreSQL database.	PostgreSQL のこのバージョンに <code>postgresql.conf</code> ファイルが含まれていることを確認してください。 <code>postgresql.conf</code> ファイルで、エントリ <code>listen_address=</code> に、データベースインスタンスで使用される IP アドレスを設定してください。 注記: <code>postgresql.conf</code> ファイルの <code>listen_address=</code> のフォーマットは重要です。構文エラーにより、データベースサーバの起動に失敗することもあります。
Unable to perform a manual switchover of version 8.X when clients are connected.	デフォルトの (スマート) シャットダウンオプションがスイッチオーバーでクライアントの切断に失敗しました。接続されたクライアントにより、シャットダウンに失敗し続けてしまう場合は、 <code>LKPGSQL_SDIRS</code> が設定されていないかどうかを確認してください。障害が継続するようであれば、LifeKeeper の設定項目 LKPGSQL_IDIRS を設定してください。
Unable to connect from a remote client to the database server.	PostgreSQL のリモートホストログインを有効にするには、 <i>PostgreSQL Administration Guide</i> の <code>pg_hba.conf</code> ファイルの設定に関する説明を参照してください。

設定項目

エラー	解決方法
psql: connectDBStart() – connect() failed:No such file or directory.Is the postmaster running at 'localhost' and accepting connections on Unix socket '<port>'?"	ソケットファイルが存在し、インスタンスが現在実行されていることを確認します。ソケットファイルが /tmp にある場合、/tmp ディレクトリをクリーンアップする cron ジョブによって削除された可能性があります。リソースを out of service にしてから、in service に戻します。次に、PostgreSQL ソケットファイルを残すように cron ジョブを修正します。
PostgreSQL resource hierarchy fails to come in service but the database is running.	データベースが、指定された期間内に LifeKeeper クライアント要求に応答できなかった可能性があります。/etc/default/LifeKeeper の設定項目 LKPGSQL_CONN_RETRIES を調整して、PostgreSQL データベースインスタンスのリカバリと再起動に許容される秒数を増やします。
PostgreSQL resource hierarchy fails local recovery following a postmaster crash with active client connections.	多数のアクティブクライアントが PostgreSQL に接続しているとき、データベースはクライアント接続が終了するまで正しく再起動できない場合があります。この場合、ローカルリカバリが正常に行われるよう強制的にクライアント接続を終了させることをお勧めします。 LKPGSQL_DISCONNECT_CLIENT 変数を /etc/default/LifeKeeper に設定して、この場合の PostgreSQL リソース階層の動作を制御することができます。値を 1 (true) に設定すると、クライアントプロセスに SIGTERM シグナルが送信され、データベースから強制的に切断されます。この処置は、ポストマスタプロセスがローカルリカバリ中に稼働していない場合のみに取ることができます。
Unable to connect to database with error "WARNING: password file "/home/<osuser>/pgpass" has world or group read access"	.pgpass ファイルの権限を、u=rw(0600) にしてください。.pgpass ファイルの権限および所有者を変更してください。
FATAL: syntax error in file "/<datadir>/postgresql.conf" line 50, near token ".17"	postgresql.conf ファイルの listen_address= のエントリが正しい引用を含んでいません。エントリが正しいかどうか、またはエントリが正しい引用を含んでいるかを確認してください。

設定項目

設定項目	機能
LKPGSQL_KILLPID_TIME	プロセスIDが停止した後、そのプロセスに対する再チェックを行うまでの時間
LKPGSQL_CONN_RETRIES	旧 LKPGSQLMAXCOUNT - 操作 (開始もしくは停止) を行った後、クライアント接続を試みる回数

設定項目

設定項目	機能
LKPGSQL_ACTION_RETRIES	アクションコマンドに失敗するまで、開始と停止を試行する回数
LKPGSQL_STATUS_TIME	status コマンドのタイムアウト値
LKPGSQL_QCKHANG_MAX	データベースインスタンスがフェールオーバー / sendevent を発生させるまでに許容される、クイックチェックがハングする回数
LKPGSQL_CUSTOM_DAEMON	postgres デーモン (デフォルト postmaster) に対する追加の別名の指定を許可します。
LKPGSQL_IDIRS	旧 LKPGSQL_IPORTS - immediate オプションのみを使用してシャットダウンするインスタンスに対する datadir のエントリを含みます。
LKPGSQL_SDIRS	Smart オプションを使用してシャットダウンするインスタンスに対する datadir のエントリを含みます。
LKPGSQL_DISCONNECT_CLIENT	データベースに障害が発生している間の PostgreSQL リソース階層の振る舞いを制御します。値を 1 (true) に設定すると、クライアントプロセスに SIGTERM シグナルが送信され、データベースから強制的に切断されます。この処置は、postmaster プロセスがローカルリカバリ中に稼働していない場合のみに取ることができます。 ※注: PostgreSQL 8.2以降では、本パラメータをご利用いただけません。
LKPGSQL_DISCONNECT_CLIENT_BYTAG	LKPGSQL_DISCONNECT_CLIENT と類似していますが、この設定は、処置を、この設定項目で指定したカンマで区切られたタグのリストに限定します。 ※注: PostgreSQL 8.2以降では、本パラメータをご利用いただけません。
LKPGSQL_RESUME_PROC	プロセスの停止状態が検出されたときに (プロセス状態が T であるとき) 再開するか無視するかを決めます。
LKPGSQLDEBUG	PostgreSQL database kit および postgres データベースのデバッグを有効にします。0 ~ 5 の値が有効です。数字が大きくなるほどデバッグ情報が増えます。 この設定項目は、オプション -d <LKPGSQLDEBUG> を使用して postmaster データベースへと渡されます。

