



プログラマーズガイド

PHP 用 Adaptive Server[®]
Enterprise 拡張モジュール バージョン 15.7 SP100

ドキュメント ID：DC01819-01-1570100-01

改訂：2013年5月

Copyright © 2013 by Sybase, Inc. All rights reserved.

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

アップグレードは、ソフトウェアリリースの所定の日時に定期的に提供されます。このマニュアルの内容を弊社の書面による事前許可を得ずに、電子的、機械的、手作業、光学的、またはその他のいかなる手段によっても、複製、転載、翻訳することを禁じます。

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

このマニュアルに記載されている SAP、その他の SAP 製品、サービス、および関連するロゴは、ドイツおよびその他の国における SAP AG の商標または登録商標です。

Java および Java 関連のすべての商標は、米国またはその他の国での Oracle およびその関連会社の商標または登録商標です。

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

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

米国政府による使用、複製、開示には、国防総省については DFARS 52.227-7013 のサブパラグラフ (c)(1)(ii)、民間機関については FAR 52.227-19(a)-(d) などの FAR 条項で定められた制約事項が適用されます。

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

目次

| | |
|---|----|
| PHP 用 Adaptive Server Enterprise 拡張モジュール | 1 |
| PHP 用拡張モジュールのインストール | 2 |
| 設定の概要 | 2 |
| サンプル PHP スクリプト | 3 |
| DSN スタイル接続プロパティのサポート | 4 |
| 有効な属性名と属性値 | 4 |
| PHP API リファレンス用拡張モジュール | 7 |
| sybase_affected_rows | 7 |
| sybase_close | 7 |
| sybase_connect | 8 |
| sybase_data_seek | 9 |
| sybase_fetch_array | 9 |
| sybase_fetch_assoc | 10 |
| sybase_fetch_field | 10 |
| sybase_fetch_object | 11 |
| sybase_fetch_row | 11 |
| sybase_field_seek | 12 |
| sybase_free_result | 12 |
| sybase_get_last_message | 13 |
| sybase_get_last_status | 13 |
| sybase_next_result | 13 |
| sybase_num_fields | 14 |
| sybase_num_rows | 14 |
| sybase_pconnect | 15 |
| sybase_query | 15 |
| sybase_rpc_bind_param_ex | 16 |
| sybase_rpc_execute | 17 |
| sybase_rpc_init | 17 |
| sybase_select_db | 18 |
| sybase_set_message_handler | 18 |
| sybase_unbuffered_query | 19 |

目次

| | |
|-----------------------------|-----------|
| sybase_use_result | 20 |
| セキュリティサービスとディレクトリサービス | 20 |
| その他のリソース | 20 |
| 用語解説 | 23 |
| 索引 | 25 |

PHP 用 Adaptive Server Enterprise 拡張モジュール

PHP スクリプト言語用の Sybase[®] Adaptive Server[®] Enterprise 拡張モジュールを使用すると、PHP 開発者は Adaptive Server のデータベースに対してクエリを実行できます。

PHP 用の拡張モジュールは Adaptive Server データベースに対してクエリを実行するベンダー固有のデータベースインタフェースドライバです。オープンソースプログラミング言語である PHP (Hypertext Preprocessor) には、共通のデータベースから情報を取得する機能があります。PHP スクリプト言語用の Adaptive Server 拡張モジュールには必要な PHP API が含まれており、これを使用することにより、PHP 開発者は、Adaptive Server のデータベースに対してクエリを実行するためのスタンドアロンスクリプトを記述することができます。

PHP 用の拡張モジュールはコマンドライン PHP 実装に対してテスト済みです。しかし、Web サーバの CGI モジュールなどでの使用については、Sybase ではテストしていません。

必要なコンポーネント

PHP プログラミング言語を使用して Adaptive Server データベースにアクセスするには、以下のコンポーネントが必要です。

- PHP スクリプト - Adaptive Server データベースサーバに接続するためのアプリケーションスクリプト。
- Adaptive Server Enterprise の PHP 用拡張モジュール - Sybase がサポートしているベンダ固有のドライバ。拡張モジュールは PHP スクリプトで呼び出され、API レイヤ DBCAPI に接続します。
- DBCAPI - PHP 拡張モジュールと CT-Library 間の中間の変換レイヤとして動作する関数ライブラリ。
- CT-Library - Adaptive Server にコマンドを送信し、結果を処理し、データを返すために使用できる Sybase の Open Client ライブラリ。

バージョン要件

プラットフォームのサポートの詳細は、使用しているプラットフォームの『Software Developer's Kit/Open Server インストールガイド』を参照してください。

- Adaptive Server Enterprise - バージョン 15.7 以上。PHP ドライバはバージョン 15.7 で開発しテスト済みです。しかし、15.7 より前の Adaptive Server バージョンには接続できません。

- PHP のインストール - バージョン 5.3.6
- Open Client SDK の CT-Library - バージョン 15.7
- DBCAPI - 同じ SDK インストールからの DBCAPI ライブラリと PHP ドライバを使用することをおすすめします。

PHP 用拡張モジュールのインストール

PHP 用の拡張モジュールは Sybase インストーラでインストールできるコンポーネントです。

インストールタイプとしてカスタムを選択した場合、PHP 用の拡張モジュールはオプションでインストールします。インストールタイプとして標準またはフルを選択した場合、この拡張モジュールはデフォルトでインストールされます。

これは拡張モジュールのインストールの概要です。インストールと設定の完全な手順については、使用しているプラットフォームの『Software Developer's Kit/Open Server インストールガイド』を参照してください。

- インストール前の要件:
 - PHP インストール 5.3.6 - 64 ビットのみ。
 - Open Client SDK が必須。DBCAPI ライブラリは Open Client SDK に含まれており、コアの Open Client (CT-Library) とともにインストールされます。ドライバの一部として DBCAPI をインストールしないでください。
- PHP 用の拡張モジュールは PHP 拡張モジュールとしてダイナミックライブラリにインストールされます。Sybase インストーラは Open Client (CT-Library) ディレクトリに、sybaseasephp.so という名前の共有ライブラリとして拡張モジュールをインストールします。
- 以下のファイルは拡張モジュールとともにインストールされます。
 - \$SYBASE/\$SYBASE_OCS/php/php536_64/lib/sybaseasephp.so
 - \$SYBASE/\$SYBASE_OCS/php/php536_64/devlib/sybaseasephp.so
 - \$SYBASE/\$SYBASE_OCS/sample/php/README
 - \$SYBASE/\$SYBASE_OCS/sample/php/firstapp.php
 - \$SYBASE/\$SYBASE_OCS/sample/php/dsnconnect.php

設定の概要

環境設定を行って、拡張モジュールと OCS インストールの場所を特定します。

環境変数

拡張モジュールを正常に使用するには、PHP 実行可能ファイルを実行する環境内で、Sybase 環境変数を設定します。少なくとも、\$SYBASE 環境変数と

`$$SYBASE_OCS` 環境変数を設定する必要があります。`LD_LIBRARY_PATH` 変数が `$$SYBASE/$$SYBASE_OCS/lib` ディレクトリの正しい場所を指すように設定します。拡張モジュールはこの場所から `DBC-API` ライブラリをロードします。

PHP の設定

システム管理者は拡張モジュールを別の拡張モジュールの特定のディレクトリにコピーし、`php.ini` ファイルを変更してそのディレクトリから拡張モジュールをロードできます。たとえば、次のようにします。

```
; Set the default extension directory.  
extension_dir = "/usr/local/lib/php/extensions"  
; Load the Sybase ASE PHP driver: sybaseasephp  
extension="sybaseasephp.so"
```

ファイルを変更したら、`$$SYBASE/$$SYBASE_OCS/php/php536_64/lib/sybaseasephp.so` または `$$SYBASE/$$SYBASE_OCS/php/php536_64/devlib/sybaseasephp.so` ライブラリを `/usr/local/lib/php/extensions` にコピーできます。

拡張ライブラリ

`php -m` を実行すると、PHP インストール内のアクティブな拡張ライブラリの中に、`sybaseasephp` 拡張ライブラリが表示されます。

注意： 同じディレクトリの特定の PHP インストール内にすべての拡張ライブラリを置く必要はありません。

サンプル PHP スクリプト

サンプル PHP スクリプトは `$$SYBASE/$$SYBASE_OCS/sample/php/` にインストールされます。

サンプルスクリプトの実行方法については、同じディレクトリにある `README` ファイルを参照してください。また、プラットフォームのインストール情報については、『Software Developer's Kit/Open Server インストールガイド』を参照してください。

PHP サンプルスクリプトは次のとおりです。

- **firstapp.php** - サーバに接続し、`select` クエリを送信して、ローを出力する簡単なサンプルプログラムです。
- **dsnconnect.php** - DSN 接続文字列を使用してサーバに接続する簡単なサンプルプログラムです。必要に応じて、サーバ名、アカウント、および現在のデータベースを出力します。

DSN スタイル接続プロパティのサポート

sybase_connect() API と **sybase_pconnect()** API は DSN スタイル接続プロパティをサポートをします。

servername パラメータのみを使用して **sybase_connect()** API または **sybase_pconnect()** API を呼び出す場合、**servername** に有効な DSN (データソース名) 文字列が含まれている必要があります。データソース名は、セミコロン (;) で区切られた名前=値の文字列です。それぞれの構成要素は次のとおりです。

1. 名前 - 等号 (=) またはセミコロン (;) で区切ることができる値。大文字と小文字が区別されます。1 つの属性に複数の同義語がある場合があります。たとえば、**server** と **servername** は同じ属性を指します。
2. 等号 (=) - 名前に割り当てる値の開始を示します。等号がない場合、名前は true 値のブール型とみなされます。
3. 値 - セミコロン (;) で終了する文字列。値内にセミコロンまたは別の円記号 (¥) がある場合は、円記号を使用します。値は、ブール型、整数型、または文字列型で指定できます。ブール型の有効な値は、true、false、on、off、1、0 です。

注意： 値がないブール名がある場合、ブール型は true に設定する必要があります。

次に例を示します。

```
Username=name;Password=pwd;Timeout=10
```

有効な属性名と属性値

dsn キーワード引数に対する有効な属性名とその値を以下の表に示します。

| 名前 | 説明 | 値 |
|-----------------|---|-----------------------------------|
| ANSINull | <p>SQL の等号 (=) または不等号 (!=) の比較で NULL 値のオペランドの評価が ANSI 標準に準拠しているかどうかを判別する。</p> <p>値が true の場合、Adaptive Server は、<i>=NULL</i> と <i>is NULL</i> は同義ではないとする ANSI の動作を適用する。標準の Transact-SQL では、<i>=NULL</i> と <i>is NULL</i> は同義とみなされる。</p> <p>このオプションは、<i><>NULL</i> と <i>is not NULL</i> の動作にも同様に作用する。</p> | <p>ブール値。</p> <p>デフォルトは false。</p> |

| 名前 | 説明 | 値 |
|------------------------------|--|---|
| BulkLogin | 接続でバルクコピーオペレーションを実行できるかどうかを判別する。 | ブール値。 デフォルトは false。 |
| ChainXacts | <p>true の場合、Adaptive Server は連鎖トランザクション動作を使用する。つまり、各サーバコマンドが個別のトランザクションとみなされる。</p> <p>Adaptive Server は、delete、fetch、insert、open、select、update の各文の前に、begin transaction を暗黙的に実行する。ただし、トランザクションを明示的に終了またはロールバックする必要がある。</p> <p>false の場合、アプリケーションは begin transaction 文を commit 文または rollback 文と対で明示的に指定する必要がある。</p> | ブール値。 デフォルトは false。 |
| CharSet | この接続で使用する charset を指定する。 | 文字列値。 |
| Confidentiality | 接続でデータの暗号化サービスを実行するかどうか。 | ブール値。 デフォルトは false。 |
| Credential-Delegation | ユーザの委任クレデンシャルを使用してサーバを他のサーバに接続させるかどうかを判別する。 | ブール値。 デフォルトは false。 |
| DetectReplay | 接続のセキュリティメカニズムがリプレイされた転送を検出するかどうかを判別する。 | ブール値。 デフォルトは false。 |
| DetectOutOfSequence | 接続のセキュリティメカニズムが不正なシーケンスの転送を検出するかどうかを判別する。 | ブール値。 デフォルトは false。 |
| Integrity | 接続のセキュリティメカニズムがデータ整合性チェックを実行するかどうかを判別する。 | ブール値。 デフォルトは false。 |
| Interfaces | interfaces ファイルのパスと名前。 | 文字列値。 |
| Keytab | 接続のセキュリティメカニズムが username 値と一緒に使用されるセキュリティキーを読み込むファイルのパスと名前。 | 文字列値。 デフォルトは NULL。 つまり、接続前にユーザがクレデンシャルを確立しておく必要がある。 |
| Locale | メッセージ、データ型変換、日時フォーマットで使用する言語と文字セットを判別する。 | 文字列値。 |

| 名前 | 説明 | 値 |
|------------------------------|---|--|
| Language | メッセージ、データ型変換、日時フォーマットで使用する言語セットを判別する。 | 文字列値。 |
| LoginTime-out | ログインタイムアウト値を指定する。 | 整数値。 |
| MaxConnect | あるコンテキストで同時にオープン可能な接続の最大数を指定する。 | 整数値。 デフォルト値は 25。負の値および 0 は使用できない。 |
| MutualAuthentication | サーバがクライアントに対して自身を認証する必要があるかどうかを判別する。 | ブール値。 デフォルトは false。 |
| NetworkAuthentication | 接続のセキュリティメカニズムがネットワークベースのユーザ認証を実行するかどうかを判別する。 | ブール値。 デフォルトは false。 |
| PacketSize | TDS パケットサイズを指定する。 | 整数値。 |
| Password | サーバへのログインに使用するパスワードを指定する。 | 文字列値。 |
| PasswordEncryption | 接続が非対称のパスワード暗号化を使用するかどうかを判別する。 | ブール値。 デフォルトは false。 |
| SecurityMechanism | 接続のためのセキュリティサービスを実行するネットワークセキュリティメカニズムの名前を指定する。 | 文字列値。 デフォルト値はセキュリティドライバ設定によって異なる。 |
| Server Servername | 接続先のサーバの名前を指定する。 | 文字列値。 |
| ServerPrincipalName | 接続がオープンされるサーバのネットワークセキュリティプリンシパル名を指定する。 | 文字列値。 デフォルトは NULL。 このとき接続は、サーバのプリンシパル名がその <i>ServerName</i> 値と同じであるとみなす。 |
| Keepalive | KEEPALIVE オプションを使用するかどうかを判別する。 | ブール値。 デフォルトは true。 |
| Timeout | 接続タイムアウト値を指定する。 | 整数値。 |

| 名前 | 説明 | 値 |
|-------------------------|------------------------|-------|
| UID User Username | サーバへのログインに使用する名前を指定する。 | 文字列値。 |

dsnconnect.php サンプルプログラム

`dsnconnect.php` サンプルプログラムは、DSN 接続文字列を使用してサーバに接続します。必要に応じて、サーバ名、ユーザアカウント、および現在のデータベースを出力します。

PHP API リファレンス用拡張モジュール

拡張モジュールインタフェース API。

sybase_affected_rows

`$conn` によって参照されている接続で最後に実行された **INSERT** クエリ、**UPDATE** クエリ、または **DELETE** クエリによって影響を受けているロー数を返します。

この関数は、通常、**INSERT** 文、**UPDATE** 文、または **DELETE** 文で使用されます。**SELECT** 文では、`sybase_num_rows()` 関数の方を使用します。

構文

```
int sybase_affected_rows([resource $conn])
```

パラメータ

- `$conn` – 接続を開く関数によって返される接続リソース。接続リソースが指定されない場合、一番最後に開かれた接続が使用されます。

戻り値

最後の **INSERT** クエリ、**UPDATE** クエリ、または **DELETE** クエリによって影響を受けたロー数。

sybase_close

接続をクローズします。

構文

```
bool sybase_close([resource $conn])
```

パラメータ

- **\$conn** – 接続を開く関数によって返される接続リソース。接続リソースが指定されていない場合、一番最後に開かれた接続を閉じます。

戻り値

成功時は TRUE。

失敗時は FALSE。

sybase_connect

Adaptive Server との接続を開きます。

sybase_pconnect() は、DSN スタイル接続プロパティをサポートします。パラメータ *servername* のみを使用して呼び出す場合、*servername* に有効な DSN 文字列が含まれている必要があります。

構文

```
resource sybase_connect([string $servername  
[, string $username  
[, string $password  
]]) )
```

パラメータ

- **\$servername** – このパラメータ以外も使用して **sybase_connect()** を呼び出す場合、**\$servername** には、関連する Sybase ディレクトリサービスで定義されているサーバが含まれます。このパラメータのみを使用して **sybase_connect()** を呼び出す場合、**\$servername** には、param1=val1;param2=val2;... という構文の有効な DSN 接続文字列が含まれます。
- **\$username** – Adaptive Server へのログイン時に使用するユーザアカウント。
- **\$password** – Adaptive Server へのログイン時に使用するユーザアカウントのパスワード。

戻り値

成功時は、正の接続識別子。

失敗時は FALSE。

sybase_data_seek

指定のロー番号を指すように、結果識別子に関連付けられている結果セット上の内部ローポインタを移動します。

構文

```
bool sybase_data_seek(resource $result, int $row)
```

パラメータ

- **\$result** - **sybase_query()** に対する呼び出しで取得される結果リソース。
- **\$row** - 内部ポインタを設定するロー番号 (0 から開始)。

戻り値

TRUE - 内部ポインタの設定に成功しました。

FALSE - 内部ポインタを正しく設定できませんでした。

sybase_fetch_array

結果ローを関連配列、数値配列、またはその両方としてフェッチします。

構文

```
mixed sybase_fetch_array(resource $result  
[, int $result_type ] )
```

パラメータ

- **\$result** - **sybase_query()** に対する呼び出しで取得される結果リソース。
- **\$result_type** - 返される配列のタイプ。以下の値を受け取ります。

SYB_FETCH_ASSOC - 関連配列

SYB_FETCH_NUM - 数値配列

SYB_FETCH_BOTH - (デフォルト) 関連と数値両方のインデックス

戻り値

\$result_type が SYB_FETCH_ASSOC または SYB_FETCH_BOTH である場合、関連配列 (**sybase_fetch_assoc()** と同じ)。

\$result_type が SYB_FETCH_NUM または SYB_FETCH_BOTH である場合、数値配列 (**sybase_fetch_row()** と同じ)。

FALSE - フェッチするローがなかった場合。

sybase_fetch_assoc

関連配列内で指定された結果識別子に関連付けられた結果セットから、データの1つのローをフェッチします。

内部ポインタの位置を結果セット内でロー1つ分進めます。その後 **sybase_fetch_assoc()** を呼び出すと、結果セット内の次のローを返します。ローがない場合は FALSE を返します。

構文

```
array sybase_fetch_assoc(resource $result)
```

パラメータ

- **\$result** – **sybase_query()** に対する呼び出しで取得される結果リソース。

戻り値

成功時は、結果セットの次のローを返します。

FALSE - フェッチするローがなかった場合。

sybase_fetch_field

フィールド情報を含むオブジェクトを返します。

この関数は、提供されるクエリ結果からフィールド情報を取得するのに使用できます。

構文

```
object sybase_fetch_field(resource $result [, int $field_offset ])
```

パラメータ

- **\$result** – **sybase_query()** に対する呼び出しで取得される結果リソース。
- **\$field_offset** – 情報を取得するフィールド番号 (0 から開始)。フィールドオフセットが指定されていない場合、この関数によってまだ取得されていない次のフィールドが使用されます。

戻り値

プロパティとして次のフィールド情報を含むオブジェクトを返します。

| フィールド名 | フィールドタイプ | フィールドの説明 |
|------------|----------|------------------------------|
| name | string | カラム名。 |
| table | string | カラムが取得されたテーブル。 |
| max_length | int | カラムの最大長。 |
| type | string | cspublic.h で定義されているカラムのデータ型。 |

sybase_fetch_object

オブジェクトとしてローをフェッチします。

この関数は、**sybase_fetch_assoc()** と似ていますが、配列ではなくオブジェクトを返します。内部ポインタの位置を結果セット内でロー 1 つ分進めます。

構文

```
object sybase_fetch_object(resource $result [, mixed $object ])
```

パラメータ

- **\$result** – **sybase_query()** に対する呼び出しで取得される結果リソース。
- **\$object** – 返すオブジェクトのタイプを指定します (オプション)。デフォルトのオブジェクトタイプは stdClass です。

戻り値

フェッチされたローのフィールド名に対応するプロパティを含むオブジェクトを返します。

FALSE - フェッチするローがなかった場合。

sybase_fetch_row

数値配列内の指定結果識別子に関連付けられている結果セットから、1 つのロー分のデータをフェッチします。

内部ポインタの位置を結果セット内でロー 1 つ分進めます。

その後 **sybase_fetch_row()** を呼び出すと、結果セット内の次のローを返します。ローがない場合は FALSE を返します。

構文

```
array sybase_fetch_row(resource $result)
```

パラメータ

- **\$result** – **sybase_query()** に対する呼び出しで取得される結果リソース。

戻り値

数値配列 (0 から開始) - **sybase_fetch_array(\$result, SYB_FETCH_NUM)** と同じ

FALSE - フェッチするローがない。

sybase_field_seek

要求したフィールドオフセットに内部ポインタを設定します。

sybase_fetch_field() の次回呼び出しでフィールドオフセットが指定されない場合、フィールド内部ポインタが使用されます。

構文

```
bool sybase_field_seek(resource $result, int $field_offset)
```

パラメータ

- **\$result** – **sybase_query()** に対する呼び出しで取得される結果リソース。
- **\$field_offset** – フィールドオフセット (0 から開始)。

戻り値

TRUE - 内部ポインタが正しく設定されました。

FALSE - 内部ポインタが正しく設定されませんでした。

sybase_free_result

結果セットに関連付けられているすべてのメモリを解放します。

スクリプトが終了すると、結果メモリは自動的に解放されます。しかしプログラミングの習慣として、必要でなくなったメモリはユーザ自身が解放することをおすすめします。

構文

```
bool sybase_free_result(resource $result)
```

パラメータ

- **\$result** – **sybase_query()** に対する呼び出しで取得される結果リソース。

戻り値

TRUE - メモリが正常に解放されたとき。

FALSE - メモリの開放に失敗したとき。

sybase_get_last_message

Server により返された最後のメッセージ。

構文

```
string sybase_get_last_message(void)
```

パラメータ

なし

戻り値

サーバにより返された最後のメッセージ。

FALSE - 最後のサーバメッセージの取得に失敗した。

sybase_get_last_status

接続 \$conn で送信された最新ステータスの結果を返します。

構文

```
int sybase_get_last_status(resource $conn)
```

パラメータ

- \$conn - 接続を開く関数によって返される接続リソース。

戻り値

接続 \$conn で送信された最新ステータスの結果。

sybase_next_result

接続 \$conn 上の次の結果セットを指す結果セット識別子を返します。

構文

```
mixed sybase_next_result(resource $conn)
```

パラメータ

- **\$conn** – 接続を開く関数によって返される接続リソース。

戻り値

成功時は、正の Sybase 結果セット識別子。

FALSE - 接続上に結果セットがありません。

sybase_num_fields

結果セット内のフィールド数を返します。

構文

```
int sybase_num_fields(resource $result)
```

パラメータ

- **\$result** – **sybase_query()** に対する呼び出しで取得される結果リソース。

戻り値

結果セット内のフィールド数。

FALSE - フィールド数の取得に失敗した。

sybase_num_rows

SELECT 文の結果セット内のロー数を返します。

sybase_num_rows() は、完全な結果セットが読み込まれているときの正しいロー数を返します。

構文

```
int sybase_num_rows(resource $result)
```

パラメータ

- **\$result** – **sybase_query()** に対する呼び出しで取得される結果リソース。

戻り値

結果セット内のロー数。

FALSE - ロー数の取得に失敗した。

sybase_pconnect

Adaptive Server との永続的な接続を開きます。

sybase_pconnect() は、DSN スタイル接続プロパティをサポートします。パラメータ *servername* のみを使用して呼び出す場合、*servername* に有効な DSN 文字列が含まれている必要があります。

注意： 以前、この呼び出しと同じ引数で永続的な接続を開いたことがある場合、新しい接続が開くのではなく、既存の接続の識別子が返されます。

構文

```
resource sybase_pconnect([string $servername  
[, string $username  
[, string $password  
]]) )
```

パラメータ

- **\$servername** – このパラメータ以外も使用して **sybase_connect()** を呼び出す場合、**\$servername** には、関連する Sybase ディレクトリサービスで定義されているサーバが含まれます。このパラメータのみを使用して **sybase_connect()** を呼び出す場合、**\$servername** には、`param1=val1;param2=val2;...` という構文の有効な DSN 接続文字列が含まれます。
- **\$username** – Adaptive Server へのログインに使用されるユーザアカウントの名前。
- **\$password** – Adaptive Server へのログインに使用されるユーザアカウントのパスワード。

戻り値

成功時は、正の接続識別子。

失敗時は FALSE。

sybase_query

接続にクエリを送信します。

構文

```
mixed sybase_query(string $query [, resource $conn])
```

パラメータ

- **\$query** – Adaptive Server に送信するクエリを含む文字列。
- **\$conn** – 接続を開く関数によって返される接続リソース。接続リソースが指定されていない場合、一番最後に開かれた接続を使用します。

戻り値

成功時は、正の Sybase 結果セット識別子。

TRUE - クエリは成功したが、結果セットは返されませんでした。

FALSE - クエリは失敗。

sybase_rpc_bind_param_ex

PHP 変数をリモートプロシージャパラメータにバインドします。

構文

```
bool sybase_rpc_bind_param_ex(resource $stmt,  
int $param_id,  
mixed $var,  
string $type  
[, bool $is_null  
[, int $direction]] )
```

パラメータ

- **\$stmt** – **sybase_rpc_init()** 呼び出しによって返される文識別子リソース。
- **\$param_id** – バインドするストアードプロシージャパラメータの位置を示すインデックス。最初のパラメータは 0 になります。
- **\$var** – バインドする PHP 変数の参照 (アドレス)。
- **\$type** – バインドする PHP 変数のデータ型。以下のいずれかです。

'd' - double

'i' - integer

'b' - binary

's' - string

- **\$is_null** – 変数が NULL を含むか含まないかを示すオプションのブール値。
- **\$direction** – 次のいずれか (オプション)。

SASE_D_INPUT - 入力パラメータ用 (デフォルト)

SASE_D_OUTPUT - 出力パラメータ用

戻り値

TRUE - PHP 変数が正常にバインドしました。

FALSE - PHP 変数のバインドに失敗しました。

sybase_rpc_execute

sybase_rpc_init() で初期化された `$stmt` 内のリモートプロシージャコールを実行します。

構文

```
mixed sybase_rpc_execute(resource $stmt)
```

パラメータ

- `$stmt` - **sybase_rpc_init()** 呼び出しによって返される文識別子リソース。

戻り値

成功時は、正の Sybase 結果セット識別子。

FALSE - RPC の実行に失敗しました。

sybase_rpc_init

接続 `$conn` で `$procedure` に対して初期化されている文を指す文識別子を返します。

構文

```
mixed sybase_rpc_init(resource $conn, string $procedure)
```

パラメータ

- `$conn` - 接続を開く関数によって返される接続リソース。
- `$procedure` - **sybase_rpc_execute()** で実行されるリモート (ストアド) プロシージャの名前。

戻り値

成功時は、正の Sybase 文識別子。

FALSE - RPC 文の初期化に失敗しました。

sybase_select_db

接続リソースにより参照されているサーバ上の現在アクティブなデータベースを設定します。

以降の **sybase_query()** 呼び出しはすべて、現在の接続リソースのアクティブなデータベースに対して行われます。

構文

```
bool sybase_select_db(string $database_name [, resource $conn])
```

パラメータ

- **\$database_name** - 使用するデータベースの名前。
- **\$conn** - 接続を開く関数によって返される接続リソース。接続リソースが指定されていない場合、一番最後に開かれた接続を使用します。

戻り値

TRUE - 現在のデータベースが正しく設定されました。

FALSE - 現在のデータベースの設定に失敗しました。

sybase_set_message_handler

クライアントまたはサーバのメッセージが受信されると呼び出されるユーザ定義済みのコールバック関数を設定します。

構文

```
bool sybase_set_message_handler(callback $handler, int $msg_type [, resource $conn])
```

パラメータ

- **\$handler** - コールバックハンドラには以下の引数があります。

int - message_number

int - severity

int - state

int - line_number

string - description

- **\$msg_type** - 下記のいずれか

SYB_CLIENTMSG_CB - クライアントメッセージコールバック

SYB_SERVERMSG_CB - サーバメッセージコールバック

注意： `$msg_type` は必須ですが、インストールされているメッセージハンドラがクライアントメッセージとサーバメッセージの両方について呼び出されることから、現在は無視されています。

- `$conn` - 接続を開く関数によって返される接続リソース。接続リソースが指定されていない場合、一番最後に開かれた接続を使用します。

戻り値

TRUE - コールバック関数のインストールに成功しました。

FALSE - コールバック関数のインストールに失敗しました。

sybase_unbuffered_query

`$conn` によって参照されている接続にクエリを送信します。

完全結果セットが `sybase_query()` のように自動的にフェッチされ、バッファされることがありません。これにより、特に結果セットが大きい場合、パフォーマンスの向上につながります。

`sybase_fetch_array()` や類似の関数を使用して、必要に応じて多くのローを読み込み、`sybase_data_seek()` を使用してターゲットローにジャンプします。

完全な結果セットが読み込まれているときに正しいロー数を返すには、`sybase_num_rows()` を使用します。

構文

```
mixed sybase_unbuffered_query(string $query [, resource $conn])
```

パラメータ

- `$query` - Adaptive Server に送信するクエリを含む文字列。
- `$conn` - 接続を開く関数によって返される接続リソース。

接続リソースが指定されていない場合、一番最後に開かれた接続を使用します。

戻り値

成功時は、正の Sybase 結果セット識別子。

TRUE - クエリは成功したが、結果セットは返されませんでした。

FALSE - クエリは失敗。

sybase_use_result

接続 \$conn に前回バッファされていないクエリの結果セットを格納して、この格納された結果セットを指す結果セット識別子を返します。

構文

```
mixed sybase_use_result(resource $conn)
```

パラメータ

- **\$conn** - 接続を開く関数によって返される接続リソース。

戻り値

成功時は、正の Sybase 結果セット識別子。

FALSE - 接続上に格納する結果セットがありません。

セキュリティサービスとディレクトリサービス

ocs.cfg ファイルと libtcl.cfg ファイルを使用してセキュリティオプションを設定します。

セキュリティを有効にする拡張モジュールのオプションは、現在サポートされていません。

接続については、ocs.cfg を使用してディレクトリとセキュリティプロパティを設定します。libtcl.cfg を編集して、セキュリティドライバとディレクトリサービスドライバをロードします。

詳細については、『Open Client/Server 設定ガイド UNIX 版』の「設定ファイル」を参照してください。

その他のリソース

拡張モジュールのインストールと設定に関する追加情報は以下を参照してください。

- Open Client/Open Server の設定情報のマニュアル：
『Open Client/Server 設定ガイド UNIX 版』の「設定ファイル」
- すべての Open Client/Server 製品についてのプラットフォームに関連した問題：
『Open Client/Server プログラマーズガイド補足 UNIX 版』

- Open Client/Server ランタイム設定ファイルの使用方法：
『Open Client Client-Library/C リファレンスマニュアル』の「ランタイム設定ファイルの使い方」の「Open Client/Server ランタイム設定ファイルの構文」
- プラットフォームのサポート：
使用しているプラットフォームの『Software Developer's Kit/Open Server インストールガイド』

用語解説

スクリプト言語に特有の用語集

- **Client-Library** – Open Client の一部で、クライアントアプリケーションを記述するためのルーチンの集まり。Client-Library は、Sybase 製品ラインのカーソルや他の高度な機能を取り込むように設計されています。
- **CS-Library** – Client-Library と Server-Library のアプリケーションの両方で役立つユーティリティールーチンの集まり。Open Client および Open Server の両方に含まれています。
- **CT-Library** – (CT-Lib API) は Open Client スイートの一部であり、スクリプトアプリケーションで Adaptive Server に接続するために必要です。
- **PHP** – Hypertext Preprocessor の自己参照的な略。
- **スレッド (thread)** – Open Server アプリケーションからライブラリコードまでの実行のパス。また、スタック領域、ステータス情報およびイベントハンドラに対応するパス。
- **Transact-SQL** – データベース言語 SQL の機能拡張バージョン。アプリケーションは、Transact-SQL を使用して、Adaptive Server Enterprise と通信できます。

索引

D

DBC-API 1

い

インストール

インストールファイル 2

概要 2

要件 2

か

拡張モジュール

PHP API リファレンス 7

拡張モジュール API リファレンス

sybase_affected_rows 7

sybase_close 7

sybase_connect 8

sybase_data_seek 9

sybase_fetch_array 9

sybase_fetch_assoc 10

sybase_fetch_field 10

sybase_fetch_object 11

sybase_fetch_row 11

sybase_field_seek 12

sybase_free_result 12

sybase_get_last_message 13

sybase_get_last_status 13

sybase_next_result 13

sybase_num_fields 14

sybase_num_rows 14

sybase_pconnect 15

sybase_query 15

sybase_rpc_bind_param_ex 16

sybase_rpc_execute 17

sybase_rpc_init 17

sybase_select_db 18

sybase_set_message_handler 18

sybase_unbuffered_query 19

sybase_use_result 20

環境変数 2

こ

コンポーネント

説明 1

必要な 1

さ

サンプルスクリプト 3

そ

その他のリソース 20

て

データフロー図 1

は

バージョン要件 1

よ

用語解説 23

