



Ultra Light アプリケーションにおける 「Persistent Name」パラメータを使用した データベースの状態管理

はじめに

Palm Computing デバイスでは、1 度に1 つのアプリケーションしか実行できません。しかし、プログラマーというのは、ユーザーが別のアプリケーションに切り替えたときに、以前のアプリケーションを終了させないようにしたいと考えるものです。プログラムによっては、ユーザーの現在の設定を格納するだけでこれを実現できるものもありますが、データベース・アプリケーションでは、アプリケーションを再起動した時に前回開いていた結果セットと同じ場所を再度開くことはかなり難しいことです。そこで、Ultra Light ではこれを支援する機能をユーザーに提供しています。

これから説明する方法によって、テーブル内の位置をリストアして、ユーザーが別のアプリケーションに切り替えた際、それまで使用していたアプリケーションが終了するのではなく中断しているような状態にすることができます。これは、ULTable オブジェクトの Open メソッドで「persistent name」パラメータの値を指定することで実現できます。



目次

はじめに.....	表紙
目次.....	3
Palm Computing プラットフォーム上のデータベース	4
Palm Computing プラットフォーム上の Ultra Light データベース	4
persistent name の使用.....	5
persistent name に関する疑問.....	5
接続およびテーブルを閉じる	8
例： persistent name を使用してステータス情報を保持する.....	9
結論.....	10
法的注意	11

Palm Computing プラットフォーム上のデータベース

Ultra Light が Palm デバイス上に情報を格納する方法から見ていきましょう。Palm オペレーティング・システムでは、保持データはすべてデータベースに格納されます。純粹主義者からは反論があるかもしれませんが、Palm データベースは、おおよそ Windows のようなオペレーティング・システムでいうファイルと同じようなものです。このホワイトペーパーでは「データベース」という用語は、Ultra Light リレーショナル・データベースの意味で使用しているため、この 2 つを対比させるとわかりやすくなるでしょう。Palm データベースとリレーショナル・データベースには類似点もありますが、Palm データベースはレコードを 1 テーブルしか格納しないため、関係を管理する必要はないという大きな相違点があります。したがって、ここでは、特に明記しないかぎり、PalmDB という用語は Palm データベースを表し、「データベース」は Ultra Light リレーショナル・データベースを示すものとしします。

Palm Computing プラットフォーム上の Ultra Light データベース

Palm Computing プラットフォーム用の Ultra Light アプリケーションを作成するときには、Ultra Light データベースは通常 Palm 作成者の ID で識別します。実際には複数の PalmDB にデータベース情報を格納しますが、それらの PalmDB の名前は与えられた作成者の ID を使用して構成されます。たとえば、ABCD という作成者 ID で作成されたデータベースを使用すると、以下のファイルが作成されます。

```
ul_state_ABCD  
ul_udb_ABCD
```

Ultra Light では、アプリケーションが終了するときに、ul_state_ABCD PalmDB を使用して、開いているカーソルまたはテーブルのステータス情報を保持します。「ステータス情報」とは、ユーザーが現在開いているローのことです。この方法を使用することにより、開発者は起動したときに、前回終了したときの状態から開始できるアプリケーションを Ultra Light で作成することができます。

PalmDB には、状態を保存する対象のテーブル名と、テーブルをその状態にリストアするための十分な情報を格納します。テーブルに関連付ける名前は、テーブルの名前にすることもできますが、必ずしもそうする必要はありません。また、これは、PalmDB に保持する名前であるため、「persistent name」と呼んでいます。

今後のリリースでは、動的 SQL がサポートされ、この同じ機能を結果セットで使用できるようになります。

persistent nameの使用

テーブルを開くという要求を実行するときに、persistent nameを渡すことができます。Ultra Light では、まず PalmDB でpersistent nameを検索し、それに関連するテーブルが存在するか確認します。存在する場合は、テーブルを開いて、適切なローを表示します。関連するテーブルが存在しない場合は、テーブルが開き、1 行目の手前の部分を表示します。

アプリケーションが終了するときには、Ultra Light 接続と開いているすべてのテーブルが、明示的に閉じられる場合とそうでない場合があります。明示的に閉じられない場合には、開いているテーブルの中でpersistent nameが指定されているものに関しては、現在のローを記録します。persistent nameが指定されていないテーブルは閉じられます。

MobileVB コードの例を見てみましょう。接続オブジェクトのタイプは ULConnection で、ULCustomer というテーブルがデータベースに存在すると仮定します。

```
DimtableAsULTable
Settable=Connection.GetTable("ULCustomer")
table.Open("", "customer")
```

コードの 2 行目は、ULCustomer テーブルを表すテーブル・オブジェクトを取得します。このテーブルは、この時点では開かれてはいないため、まだ読み取りまたは書き込みはできません。テーブルを開くのは次の行です。

Open 呼び出し (コードの 3 行目) で、最初のパラメータは空の文字列で、データがプライマリ・キーに従った順序で並べられることを示します。第 2 のパラメータは、テーブルに割り当てられている persistent nameです。このテーブルがまだ開いている状態でアプリケーションを終了すると、PalmDB は "customer" を "ULCustomer" テーブルと関連付けて、現在の位置を保存します。

persistent nameは、開いているテーブルに割り当てる名前です。テーブル名と必ずしも一致させる必要はなく、スキーマ定義に存在するべきものでもありません。あくまでも、このテーブルと関連付けるために状態 PalmDB に格納する名前です。

persistent nameに関する疑問

この時点で、persistent nameの使用に関していくつか疑問が生じていることでしょうか。この項では、そうした疑問にお答えします。

persistent nameを空にするとどうなりますか？

persistent nameを空にすることは、Ultra Light にこのテーブルのステータス情報を格納しても、テーブルを開くときにステータス情報を検索しようとはしないと指示することになります。

Ultra Light でpersistent nameを指定する必要があるのはなぜですか？

テーブルの状態を格納する必要がない場合は、データベースの名前の検索を省略できます。上述したように、この処理は、persistent nameを空にすることで実行できます。

テーブル名をpersistent nameにして、テーブルのプール・プロパティを保持しないのはなぜですか？

Ultra Light を使用すると、アプリケーションで同じテーブルを複数回同時に開くことが可能ですが、この場合、テーブル名はそれほどユニークでないため、この状態情報は PalmDB には格納できません。この処理を行うアプリケーションでは、コードは以下ようになります。

```
Settable1=Connection.GetTable("ULCustomer")
table1.Open("","customer1")
.
.
.
Settable2=Connection.GetTable("ULCustomer")
table2.Open("","customer2")
```

HotSync 同期は、開いているテーブルの状態にどのように影響しますか？

一言で言うと、影響はありません。ユーザーがデバイスの [HotSync] ボタンを押すと、OSでは、他のアプリケーションを起動した場合と同じ方法でアプリケーションを閉じます。その結果、開いているテーブルの状態は状態 PalmDB に記録され、ユーザーがそのアプリケーションに戻ってテーブルを再度開くと、期待どおりのローを表示します。そのローが同期の一環として削除されている場合は、その次のロー(該当するローが最後のローであった場合は、最後のローの後)を表示します。

アプリケーション終了時に未処理のトランザクションがある場合、どうなりますか？

オートコミットがオフになっているアプリケーションの場合は、コミットが済んでいないトランザクションが残っている状態でアプリケーションが終了することがあります。Ultra Light はこのようなトランザクションを保持し、アプリケーションを再起動したときに再び処理するようにします。

テーブルを開くときに、既存のpersistent nameを使用したものの、インデックスが異なる場合はどうなりますか？

指定したpersistent nameに一致するテーブルがPalmDB 内で見つかった場合、Ultra Light は、テーブルとインデックスが、位置情報を記録したときに使用したテーブルとインデックスと同じであるかどうかチェックします。テーブルとインデックスが異なる場合、Open 呼び出しが失敗します。

MobileVB を使用して Palm 以外のプラットフォーム (Pocket PC など) で動作する Ultra Light アプリケーションを作成することができますが、この動作は、Palm 以外のプラットフォームでも可能ですか ?

現時点 (バージョン 8.0) ではできません。

接続およびテーブルを閉じる

ULTable オブジェクトを明示的に閉じてしまう (Close メソッドを呼び出す) と、現在のローは保持されません。ULConnection オブジェクトを明示的に閉じると、ULTable オブジェクトはすべて暗黙的に閉じられるため、現在のローはすべてのテーブルで保持されないことになります。

現在の状態がPalmDB に格納できるのは、閉じられていないテーブルのみで、またULConnection オブジェクトの Close メソッドが呼び出されない場合のみです。つまり、ULConnection オブジェクトで Close を呼び出さずにアプリケーションを終了する必要がある、あるいはULConnection オブジェクトを定義したルーチンが終了している、またはULConnection オブジェクトの変数が Nothing に割り当てられているか、のいずれかです。

オートコミットがオフになっているアプリケーションの場合、接続を閉じると、コミットされていないトランザクションがロール・バックされます。ULConnection オブジェクトを閉じなければ、未処理のトランザクションは保存され (コミットはされない)、アプリケーションを再起動したときに、それらのトランザクションを表示してコミットまたはロール・バックすることができます。

例：persistent nameを使用してステータス情報を保持する

プログラム例の persistent name は、保持されたステータス情報の使用方法がわかる比較的シンプルなプログラムで、<http://www.sybase.com/detail?id=1022734> にあります。また、以下は、主な部分をそのサンプルから抜粋したものです。

```
CustomerTable.Open
AddRow"John", "Doe", "Atlanta"
AddRow"Mary", "Smith", "Toronto"
AddRow"Jane", "Anderson", "NewYork"
AddRow"Margaret", "Billington", "Vancouver"
AddRow"Fred", "Jones", "London"
AddRow"Jack", "Frost", "Dublin"
AddRow"David", "Reiser", "Berlin"
AddRow"Kathy", "Stevens", "Waterloo"
AddRow"Rebecca", "Gable", "Paris"
AddRow"George", "Jenkins", "Madrid"
CustomerTable.Close
```

このコードを使用すると、ULCustomer テーブルにローが10 個追加されます。テーブル上で Open を呼び出しますが、この場合データを挿入しただけなので、テーブル内の位置は重要ではなく、テーブル内の位置を保持する必要はないため、persistent nameは指定しません。persistent nameを指定してもかまいませんが、便宜上、ここでは指定していません。

CustomerTable.Open "", "customer" というコードを使用すると、プライマリ・キーに従った順序でローが並べられ、"customer" のpersistent nameの確認や割り当てが行われると同時に、ULCustomer テーブルが開かれます。アプリケーションを事前に実行している場合は、"customer" のデータベースの検索が行われます。それ以外の場合は、"customer" はこのテーブルに関連付けられません。

アプリケーション実行中は、customer テーブルは開いたままで、ユーザーが別のアプリケーションに切り替えると、切り替え時に開いていたテーブル内の位置をUltra Light が記録します。アプリケーションを再度実行し、そのテーブルが開かれると、persistent nameが "customer" であるテーブルの位置情報を把握していると判断するため、そのローが表示されます。

ユーザーが [End] ボタンをクリックすると、customer テーブルと接続が閉じられ、アプリケーションが画面から消えます。これには、customer テーブルのステータス情報をすべて廃棄してしまう効果があるため、アプリケーションを再起動したときには、ユーザーは最初のローに位置されています。

結論

ステータス情報の保持は、Ultra Light の一連の機能の大きな特徴です。アプリケーションの切り替え時に、事前に使用していたアプリケーションに終了ではなく中断するような状態を保たせたい場合に、この組み込み機能がなければ、開いているテーブル内の現在の位置をトラッキングするための論理を新たに実装する必要がでてきます。ですから、Palm Computing プラットフォーム対応の Ultra Light アプリケーションを設計する場合は、この機能をぜひお役立てください。

法的注意

Copyright(C) 2000-2003 iAnywhere Solutions, Inc. All rights reserved.

iAnywhere、iAnywhere Solutions、Adaptive Server、SQL Anywhereは、iAnywhere Solutions, Inc.またはSybase, Inc.とその系列会社の商標です。その他の商標はすべて各社に帰属します。

Mobile Linkの技術には、Certicom, Inc.より供給を受けたコンポーネントが含まれています。これらのコンポーネントは特許によって保護されています。

本書に記載された情報、助言、推奨、ソフトウェア、文書、データ、サービス、ロゴ、商標、図版、テキスト、写真、およびその他の資料(これらすべてを"資料"と総称する)は、iAnywhere Solutions, Inc.とその供給元に帰属し、著作権や商標の法律および国際条約によって保護されています。また、これらの資料はいずれも、iAnywhere Solutions, Inc.とその供給元の知的所有権の対象となるものであり、iAnywhere Solutions, Inc.とその供給元がこれらの権利のすべてを保有するものとしします。

資料のいかなる部分も、iAnywhere Solutionsの知的所有権のライセンスを付与したり、既存のライセンス契約に修正を加えることを認めるものではないものとしします。

資料は無保証で提供されるものであり、いかなる保証も行われません。iAnywhere Solutions, Inc.は、資料に関するすべての陳述と保証を明示的に拒否します。これには、商業性、特定の目的への整合性、非侵害性の黙示的な保証を無制限に含みます。

iAnywhere Solutionsは、資料自体の、または資料が依拠していると思われる内容、結果、正確性、適時性、完全性に関して、いかなる理由であろうと保証や陳述を行いません。iAnywhere Solutionsは、資料が途切れていないこと、誤りがないこと、いかなる欠陥も修正されていることに関して保証や陳述を行いません。ここでは、「iAnywhere Solutions」とは、iAnywhere Solutions, Inc.またはSybase, Inc.とその部門、子会社、継承者、および親会社と、その従業員、パートナー、社長、代理人、および代表者と、さらに資料を提供した第三者の情報元や提供者を表します。

* 本書は、米国iAnywhere Solutions, Inc.が作成・テストしたものを日本語に翻訳したものです。



アイエニウェア・ソリューションズ株式会社
<http://www.ianywhere.jp>