

Generic full data export/import routine

By Thomas Maul, General Manager, 4D Germany.

TN 06-07

Introduction

このテクニカルノートでは、データファイルをまるごとインポート/エクスポートするための汎用的なメソッドを紹介しています。メソッドを実行すれば、ピクチャ、BLOB、サブテーブルフィールドを含むすべてのデータが XML 形式でインポート/エクスポートできます。

フルテキストインポート/エクスポートの実行は、データベースの耐久性を高め、壊れかけたデータを修復する点においても優れた効果があることがこれまで実証されてきました。すべてのデータをテキストベースでエクスポートし、同じくテキストベースで新規データファイルにインポートすれば、きれいなデータと真新しいインデックスが手に入ります。それだけでなく、完全に断片化解消されたデータファイルが作られるので、シーケンシャルアクセスのスピードが向上するというメリットもあります。

テーブルごとのシーケンスナンバーの設定は、自動的に復元されるようになっています。

今回のコードはすべて単一のメソッドにまとめられているので、移植がとても簡単です。

Installation

サンプルデータベース唯一のメソッド"ExportImport"をメソッドエディタで開いてコピー、またはテキストファイル"ExportImport.rtf"を適当なエディタで開いてコピーしてください。移植先のデータベースでメソッド"ExportImport"を作成します。メソッド名は厳密に一致していなければなりません。別のメソッド名を使用するのであれば、それに合わせてコードを修正する必要があります。クリップボードにコピーされたテキストをペーストしてください。未定義変数タイプチェックの仕方を「すべて定義させる」にしてコンパイルするのであれば、Compiler_XXX メソッドに次のコードを加えてください：

```
C_TEXT(ExportImport ;$0;$1;$2)
C_POINTER(ExportImport ;$3)
C_LONGINT(ExportImport ;$4)
C_BOOLEAN(EXPORTIMPORT_STOP)
```

メソッドで使用しているプロセス変数は ExportImport_Stop だけです。インポート/エクスポートをキャンセルしたい場合、ON ERR CALL などからこの変数にアクセスしてください。

Usage

メソッドはユーザモードから直接実行することができます。パラメータを渡さない場合、ダイアログが表示され、実行するアクションの種類およびファイルパスが尋ねられます。

エクスポートを実行する場合、1 個のテーブルにつきひとつのファイルが作成されます。Windows XP であれば4Dは2GB以上のファイルを扱えますが、Mac OS との互換性に配慮し、1.5GB を超えるファイルについては自動的にセグメントを使用するようになっています。ファイルはXML形式であり、ピクチャやBLOBはBASE64にエンコードされて書き出されます。インポート実行時の注意：

今回のメソッドは、自動的に既存レコードをすべて削除してからインポートを開始します。基本的に新規データファイルが使用されることを想定しています。

既存のインデックスは破棄され、インポート後に再構築されます。これにはインポートの速度を向上させるだけでなく、レコードやインデックスページの断片化を最少限にとどめる効果があります。シーケンスナンバーは自動的に復元されます。

インタフェースに統合できるように、メソッドにはパラメータを渡せるようになっています：

Result := ExportImport(direction; path)

Parameter	Type	Description
direction	文字	Import または Export
path	文字	ファイルパス

使用例：

```
$result := ExportImport ("Export";"C:¥¥myxmlfolder")
$result := ExportImport ("Import";"C:¥¥myxmlfolder")
```

Table/Field names

メソッドは自動的にテーブル名およびフィールド名をXMLに適合したタグに変換します。特殊文字は_に置換され、数字から始まるフィールド名も同じように処理されます。

置換例：

```
Straße    -> Stra_e
2address  -> _2address
```

History

かつてテクニカルノート 01-43 で紹介された原型は、4D 6.7 のコマンドだけで作成され、コンポーネント形式だったのでインストールに制限がありました。今回はサブテーブルに対応し、シーケンスナンバーもサポートされるようになりました。