



Technical Note 05-34

EDIT FORMULA

By Jean-Yves Fock-Hoon, QA Manager, 4D, Inc.
Technical Note 05-34

(原題: The new EDIT FORMULA dialog in 4D 2004)

概要

アプリケーションにおけるフォーミュラの使用全般に関する考察です。

フォーミュラエディタについて

ユーザモードで可能な操作、例えばクエリや並び替え、印刷などの動作には、対応するコマンドがあり、同様の機能をカスタムメニューモードに組み込むことができますが、フォーミュラエディタだけは、そのようなコマンドがありませんでした。

バージョン 2004 では、EDIT FORMULA コマンドにより、フォーミュラエディタを呼び出すことができるようになりました。とはいえ、このダイアログは、ユーザモードの「フォーミュラで更新」と同じものではありません。表示されるのは、純粹にフォーミュラを編集するためのダイアログです。もちろん、適用ボタンのクリックで APPLY TO SELECTION を実行すればよいのですが、フォーミュラの用途は更新だけに限られたものではなく、クエリや並び替え、クイックレポートのカラムや 4D Chart にも適用することができます。APPLY FORMULA ではなく EDIT FORMULA コマンドが採用された理由はそこにあります。

EDIT FORMULA ダイアログ

コマンドにはマスターテーブルとテキスト変数をパラメータとして渡します。テキスト変数については、入出力パラメータであり、戻り値を受け取ります。テキスト変数には、デフォルトのフォーミュラを渡します。空の変数であればエディタのフィールドは空欄です。

ダイアログでは、フォーミュラの構文が正しくないと OK ボタンがクリックできません。その後、OK 変数でダイアログが確定されたことを調べ、返されたテキスト変数を使用して APPLY TO SELECTION あるいは必要な他の処理を実行します。

ダイアログの改良

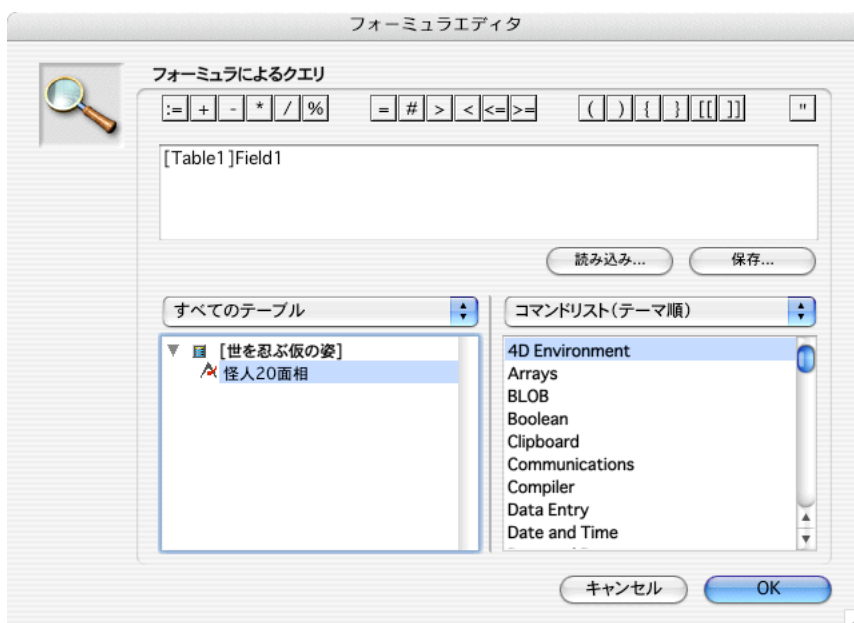
バージョン 2003 と 2004 ではフォーミュラエディタの外観が大きく変更されました。理由にはセキュリティ面での問題とバーチャルストラクチャへの対応が挙げられます。

セキュリティ

バージョン 2003 のフォーミュラエディタでは、すべてのコマンドが実行でき、ユーザのアクセス権限に基づいてこれを制限する手段がありませんでした。エンドユーザがメソッドや DELETE SELECTION、QUIT 4Dなどを勝手に実行できることは問題です。仮に 2004 の LAUNCH EXTERNAL PROCESS を実行できてしまえば、さらに重大な問題が起こる恐れがあります。そのような訳で、バージョン 2004 のフォーミュラエディタには、デフォルトですべてのメソッドおよびコマンドが選択できないようになりました。メソッドについては、後述する SET ALLOWED METHOD によって明示的に許可することができます。

バーチャルストラクチャ

4D はテーブルやフィールドを非表示にしたり、表示上の名前や並び順を変更したりすることによってバーチャルストラクチャを作成することができるようになっています。バージョン 2003 の場合、クエリエディタやフォーミュラエディタに表示されるストラクチャはバーチャルストラクチャを反映していましたが、編集時のフォーミュラをみれば、本当の名前が分かっていました。



バージョン 2004 のフォーミュラエディタでは、編集時のフォーミュラにも選択されたバーチャルストラクチャが使用されるようになりました。フォーミュラが正しく実行されるためには、本当のテーブル名とフィールド名を使用しなくてはなりません。SET TABLE TITLES および SET FIELD TITLES でオプションの引数 *を使用すれば、フォーミュラにバーチャルストラクチャ名を使用できるようになります。

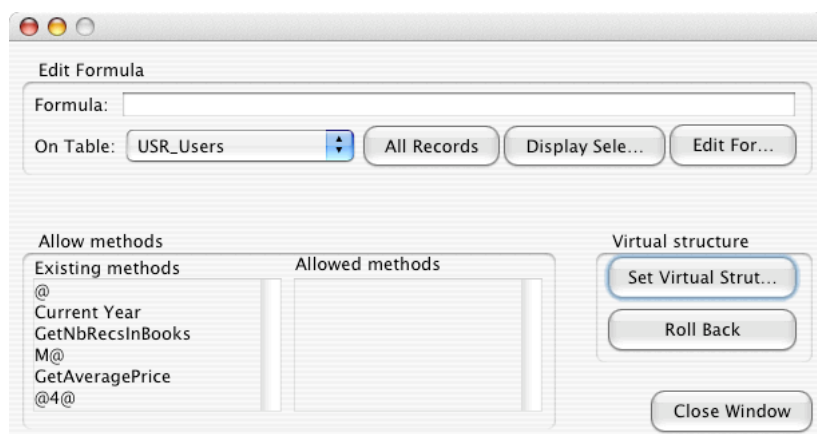
SET ALLOWED METHOD

フォーミュラエディタで選択できるメソッドのリストを設定するコマンドです。パラメータにはテキストの配列を渡し、ワイルドカードも使用できます。「@」だけを渡せば、すべてのメソッドが選択できるようになります。許可されていないメソッドやコマンドは、た

とえタイプしたとしても構文上、認識されないので、OK ボタンをクリックすることができません。このコマンドはプロセスではなく、アプリケーション全体において有効です。

サンプルデータベース

[USR_Users]、[BOK_Books]および[KYB_Keywords]テーブルには、それぞれテーブル名と同じ頭文字で始まる名前のフィールドが定義されています。一部のメソッドはフォーミュラエディタでの使用が許可されており、残りは使用できないようになっています。



Formula フィールドおよび On Table ドロップダウンリストは、EDIT FORMULA に渡すパラメータを設定するために使用します。ALL RECORDS および DISPLAY SELECTION は[BOK_Books]テーブルに対して実行されます。Edit Formula は EDIT FORMULA と APPLY SELECTION を連続して実行します。

ダイアログの下半分には、SET ALLOWED METHODS に渡す引数の候補と、許可されたメソッドのリストが並んでおり、項目をドラッグ&ドロップすればメソッドを許可できるようになっています。リストから取り除くにはダブルクリックをします。

Set Virtual Structure は、テーブルとフィールドに名前を割り当てるボタン、Roll Back は本来の名前に戻るボタンです。

フォーミュラエディタのカスタマイズ

Edit Formula をクリックし、ストラクチャの名前は見え、メソッドがまったく選択できないことを確認します。前の画面に戻り、使用できるメソッドを適当に設定してからエディタを開きます。フォーミュラエディタでは、すべてのプラグインコマンドが禁止されている点に注目できます。EXECUTE を使用すれば、プラグインコマンドを実行できるので、EXECUTE をコールするメソッドを作成して許可すれば、プラグインコマンドをフォーミュラエディタで実行できるようになります。ただし、これには潜在的な危険もあります。

以前のフォーミュラエディタでは、非表示に設定されたメソッドであっても、たまたまタイピング入力で名前が合致すれば実行できてしまいました。バージョン 2004 では、メソッドの非表示属性に関係なく、許可されていないメソッドは OK ボタンのクリックを許さないで、偶然または故意であっても実行されることはありません。そのようなメソッドを実行する唯一の方法は、前述のプラグインコマンドのように EXCUTE(\$1)というメソッド

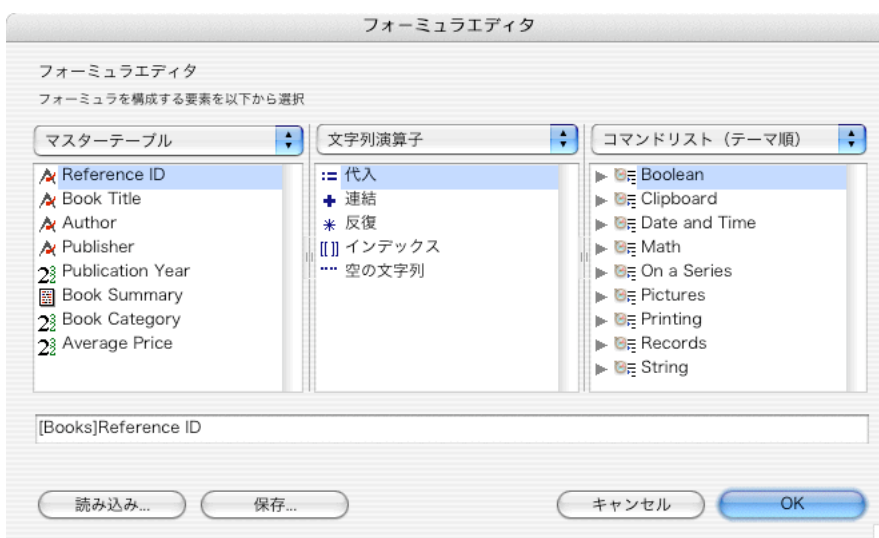
ドを経由する方法です。サンプルの M_Del4Sur メソッドはすべてのレコードを削除する可能性のあるメソッドです。非表示属性であり、許可されていなければ、タイピングが合致しても実行されません。

フォーミュラエディタを表示する前に、ユーザおよび業務内容に基づいて利用できるメソッドを絞り込めば、セキュリティを高められることに加えて、画面表示をすっきりさせることができるというメリットもあります。

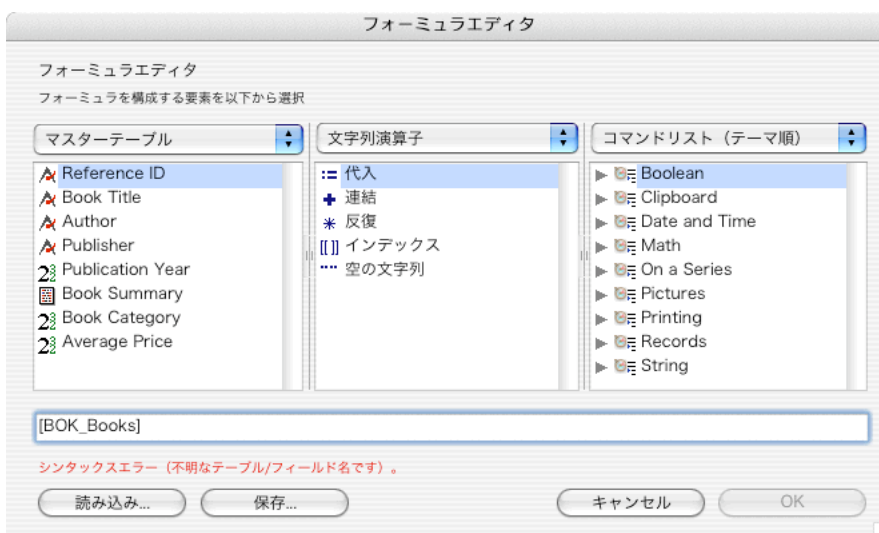
注記: サンプルでは「@」と他の条件の併用はできないようになっています。

バーチャルストラクチャ

サンプルデータベースでは、クリックひとつでバーチャルストラクチャを適用することができます。以前のバージョンとは異なり、フォーミュラエディタのテーブルリストから項目をクリックするとバーチャル名が使用されます。



ほんとうのテーブル名をタイプしてもシンタックスエラー扱いになるので実行できません。



RevertToNonVirtual

コマンドに渡すデフォルトのフォーミュラには、実際のテーブル名を使用しますが、それでもエディタに表示されるのはバーチャル名です。逆に実行後に返されるテキスト変数には、バーチャル名がそのまま返されるので、APPLY TO SELECTION には渡すことができません。サンプルデータベースでは、RevertToNonVirtual メソッドでフォーミュラを変換しています。

テーブルの省略されたフィールド名すべてにマスターテーブル名を補完し、ループでフィールド名を置換しています。実際のフィールド名が他の場所ではバーチャルフィールド名として使用されている可能性もあり、その場合は再変換が生じてしまうので、フォーミュラの他の場所では決して使用されない文字列をタグとして使用し、そのタグを変換します。

フォーミュラが復元できれば、EXECUTE コマンドを使用して実行することができます。

```
EXECUTE("APPLY TO SELECTION(Table(al_TableIDs{at_TableNames})->"+$MyFormula+"")
```

または

```
APPLY TO SELECTION(Table(al_TableIDs{at_TableNames})->;EXECUTE($MyFormula))
```

次の構文はシンタックスエラーになります。(「:=が必要です」エラー)

```
APPLY TO SELECTION(Table(al_TableIDs{at_TableNames})->;$MyFormula)
```

当然、セレクションがなければ、何も起きません。