4D v11 SQL Release 1 (11.1)
ADDENDUM

4D v11 SQL Release1 にようこそ。このドキュメントでは、新しく追加された機能と変更点について説明しています。

クエリの解析

4D に 3 つのコマンドが追加され、データに対して実行されるクエリの適合率を解析できるようになりました。

これら 3 つの新しいコマンドは " クエリ " テーマに分類されています。

DESCRIBE QUERY EXECUTION

DESCRIBE QUERY EXECUTION (status)

引数	型		説明
status	Boolean	→	True= 内部的なクエリ解析を有効にする False= 内部的なクエリ解析を無効にする

DESCRIBE QUERY EXECUTION は、カレントプロセスで、クエリ解析を有効にするか無効にするかを設定するために使用します。このコマンドは 4D ランゲージおよび SQL で行われるクエリ両方に有効です。

このコマンドを、True を *status* 引数に渡して呼び出すと、クエリ解析モードが有効になります。このモードでは、4D エンジンはデータに対し行われる連続したクエリごとに、内部的に二つの特定の情報を記録します。

- 内部的な実行前の詳細なクエリ情報。言い換えれば実行しようとする計画 (クエリプラン)。
- 内部的な実際に実行されたクエリ情報 (クエリパス)。

記録される情報にはクエリのタイプ (インデックスやシーケンシャル)、見つかったレコードの数、そして実行されるクエリ条件ごとの必要時間が含まれます。

この情報は新しい Get Last Query Plan や Get Last Query Path コマンドを使用して読み出すことができます。

通常、クエリプランとクエリパスの情報は同じです。しかし、パフォーマンスを向上させるために 4D が動的にクエリ実行時に最適化を行うことがあり、その場合は両情報が異なります。例えば、4D エンジンは、インデックスクエリを、そちらのほうが速いと判断すればシーケンシャルクエリに切り替えることがあります。このようなケースは特に、クエリ対象のレコード数が少ないときに発生します。

クエリの解析を終了するには、*status* 引数に False を渡します。クエリの解析はアプリケーションの実行速度を遅くします。

▼ 以下の例は、SQL クエリを行った際に入手することのできる情報のタイプを示します。

```
C_TEXT($vResultPlan;$vResultPath)
ARRAY TEXT(aTitles;0)
ARRAY TEXT(aDirectors;0)
DESCRIBE QUERY EXECUTION(True)`解析モード
Begin SQL
  SELECT ACTORS.FirstName, CITIES.City_Name
    FROM ACTORS, CITIES
    WHERE ACTORS.Birth_City_ID=CITIES.City_ID
    ORDER BY 1
    INTO :aTitles, :aDirectors;
End SQL
$vResultPlan:=Get Last Query Plan(Description in Text Format)
$vResultPath:=Get Last Query Path(Description in Text Format)
DESCRIBE QUERY EXECUTION(False)`解析モード終了
```

このコード実行後、*$vResultPlan* と *$vResultPath* には実行されたクエリの情報が格納されます。例えば:

■ *$vResultPlan*:

```
[Join] : ACTORS.Birth_City_ID = CITIES.City_ID
```

■ *$vResultPath*:

```
And
  [Merge] : ACTORS with CITIES
    [Join] : ACTORS.Birth_City_ID = CITIES.City_ID (1227 records found in 13 ms)
  --> 1227 records found in 13 ms
--> 1227 records found in 14 ms
```

Get Last Query Path に Description in XML Format が渡されると、*$vResultPath* には、XML 形式で、クエリの説明が返されます。

```
<QueryExecution>
  <steps description="And" time="0" recordsfounds="1227">
```

```
        <steps description="[Merge]: ACTORS with CITIES" time="13"
recordsfounds="1227">
          <steps description="[Join]:ACTORS.Birth_City_ID = CITIES.City_ID"
time="13" recordsfounds="1227"/>
        </steps>
      </steps>
  </QueryExecution>
```

Get Last Query Plan

Get Last Query Plan (descFormat) → String

引数	型		説明
descFormat	Longint	→	情報フォーマット：テキストまたはXML
返り値	String	←	最後に実行されたクエリプランの情報

Get Last Query Plan コマンドは、データに対し最後に実行されたクエリプランの情報を返します。クエリ情報についての詳細は DESCRIBE QUERY EXECUTION コマンドの説明を参照してください。

この情報は *descFormat* 引数に渡された値に基づき、テキストまたはXML形式で返されます。"Query" テーマに分類された以下の定数を使用できます：

定数	型	値
Description in Text Format	Longint	0
Description in XML Format	Longint	1

このコマンドを使用する前に、セッション中で DESCRIBE QUERY EXECUTION を実行していなければなりません。

最適化を行うために、クエリプランの情報を Get Last Query Path コマンドによって得られる実際のクエリパスと比較することができます。詳しい情報は DESCRIBE QUERY EXECUTION コマンドの説明を参照してください。

Get Last Query Path

Get Last Query Path (descFormat) → String

引数	型		説明
descFormat	Longint	→	情報フォーマット：テキストまたはXML
返り値	String	←	最後に実行されたクエリパスの情報

Get Last Query Path コマンドは、データに対し最後に実行された実際のクエリパス情報を返します。クエリ情報についての詳細は DESCRIBE QUERY EXECUTION コマンドの説明を参照してください。

この情報は *descFormat* 引数に渡された値に基づき、テキストまたは XML 形式で返されます。"Query" テーマに分類された以下の定数を使用できます:

定数	型	値
Description in Text Format	Longint	0
Description in XML Format	Longint	1

このコマンドを使用する前に、セッション中で DESCRIBE QUERY EXECUTION を実行していなければなりません。

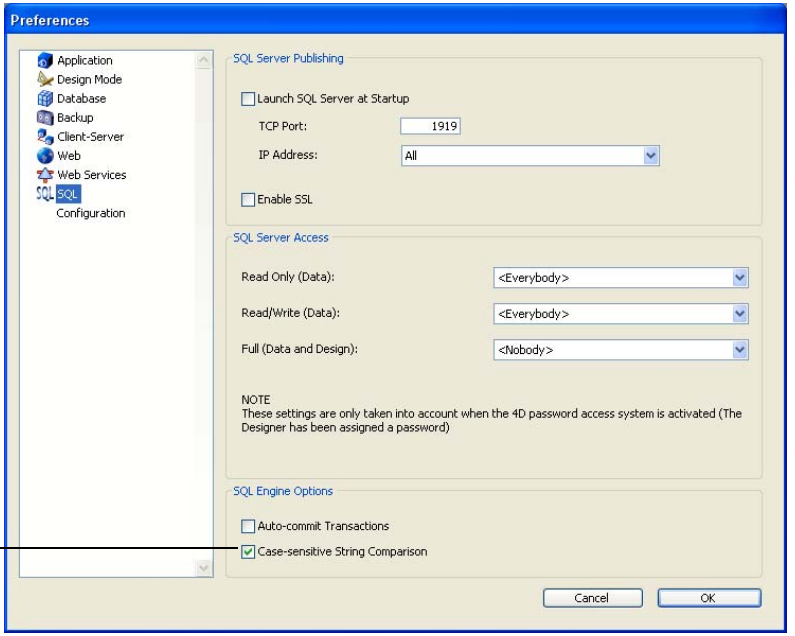
最適化を行うために、クエリプランの情報を Get Last Query Plan コマンドによって得られる実際のクエリパスと比較することができます。詳しい情報は DESCRIBE QUERY EXECUTION コマンドの説明を参照してください。

SQL Engine

この節では 4D に統合された SQL エンジンの新機能について説明します。

大文字小文字の処理

SQL クエリで大文字と小文字の比較方法を指定するためのオプションが環境設定に追加されました。" 大文字小文字を区別した文字列比較 " オプションは 環境設定の SQL/ 設定にあります。



新しいオプション

このオプションはデフォルトでチェックされています。この場合、SQL エンジンは文字列比較 (ソートやクエリ) の際、大文字と小文字を区別します。例えば "ABC"="ABC" ですが、"ABC"#"Abc" となります。

特定のケース、例えば 4D エンジンと SQL エンジンの機能をそろえるために、大文字と小文字を区別しない比較 ("ABC"="Abc") をしたい場合はこのチェックを外します。

SET DATABASE PARAMETER と Get Database Parameter の新しいセレクタ

SQL エンジンの大文字小文字比較オプションは、SET DATABASE PARAMETER と Get database parameter コマンドを使用してプログラムから読み書きできます。この用途に使用するための新しいセレクタが追加されました :

- Selector = 44 (SQL Engine Case Sensitivity)
  - 値 : 0 または 1 (0 = 区別しない、1 = 区別する)
  - 説明 : SQL エンジンによって実行される文字比較において、大文字小文字の区別を有効または無効にする。

ポインタ型の式の参照

SQL クエリで、ポインタを直接参照できるようになりました。

- ポインタを逆参照した変数を渡す :

```
C_LONGINT($vLongint)
C_POINTER($vPointer)
$vLongint:=1
$vPointer:=->$vLongint
Begin SQL
   SELECT Col1 FROM TEST WHERE Col1=:$vPointer->;
End SQL
```

- 逆参照しないポインタ変数を渡す :

```
C_LONGINT($vLongint)
C_POINTER($vPointer)
$vLongint:=1
$vPointer:=->$vLongint
Begin SQL
   SELECT Col1 FROM TEST WHERE Col1=:$vPointer;
End SQL
```

これはすべてのタイプの SQL クエリに有効です : ODBC コマンド、Begin/End SQL タグ、QUERY BY SQL コマンド

Note 1 レベルのポインタのみが有効です。ポインタのポインタは使用できません。

Unicode

4D v11 SQL Release 1 では、いくつかの機能で新しく Unicode サポートが拡張されました。

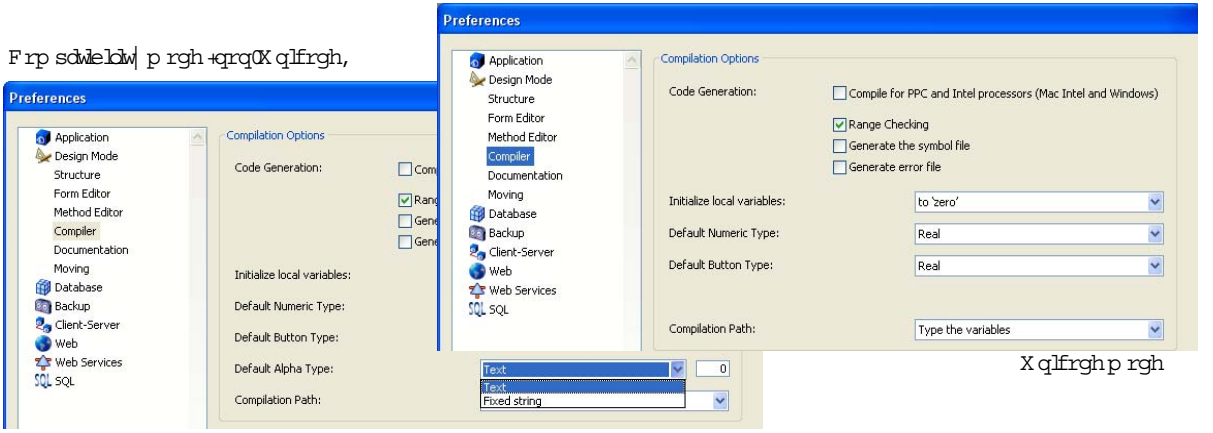
メニュー

4D アプリケーションで、メニューやメニュー項目が Unicode で表示されるようになりました。これにより例えば、日本語とギリシャ語を同じメニューで使用できるようになりました。この新しい機能は、メニューバーエディタで作成されたかプログラムで作成されたかにかかわらず、すべてのメニューで有効です。

この機能は、もちろんデータベースが Unicode モードで動作している場合にのみ有効です。

デフォルト文字変数タイプオプション

環境設定のデザインモード / コンパイラページにある " デフォルト文字変数タイプ " コンパイルオプションは、データベースが Unicode モードで動作している場合表示されません。



Unicode モードではテキスト型が自動で使用されます。

Replace string

Replace string (source; oldString; newString{; howMany{; *}}) → String

引数	型		説明
source	String	→	元の文字列
oldString	String	→	置き換え対象文字列

引数	型		説明
newString	String	→	置き換え後文字列
howMany	Number	→	置き換え数
*	*	→	発音区別符号を評価する (オプション)
返り値	String	←	結果文字列

Replace string は最後の引数にアスタリスク (*) を受け入れるようになりました。この引数を渡すと、文字の評価は発音区別符号を考慮に入れるようになります。つまり、大文字と小文字が区別され、アクセント文字が考慮されるようになります (a#A, a#à, etc.)。

例 :

```
vResult:=Replace string("Crème brûlée";"Brulee";"caramel")
        `vResult は "Crème caramel" になる
vResult:=Replace string("Crème brûlée";"Brulee";"caramel";*)
        `vResult は "Crème brûlée" のまま
```

4D View

新しいバージョンの 4D View プラグインでは、表示と入力が Unicode で行われるようになりました。このモードは 4D の Unicode オプションにかかわらず有効です。

4D View v11.1 は以前のバージョンのドキュメント開くことができます。しかし v11.1 で作成された 4D View ドキュメントをバージョン 11 や 2004.x で開くことはできません。

他の新しい機能

クイックレポートとプリント設定

インターフェースの統一を図るため、4D アプリケーションとクイックレポートのプリント設定は同じものが使用されるようになりました。

以前のバージョンの 4D では、クイックレポートエリアのプリント設定はレポートごとに保存されていました。4D の印刷設定のへの変更はクイックレポートに反映されず、また逆もそうでした。

バージョン 11.1 から、プリント設定は常に 4D とクイックレポートで同じです。4D プリント設定で行われた変更は、クイックレポートのそれにも反映されます。逆にクイックレポートをロードすると、レポートのプリント設定は 4D のそれを置き換えます。

ピクチャのクリック座標

4D で、ピクチャフィールドや変数をクリックした際のローカル座標を、ピクチャがスクロールされていたりズームされていたりしても、取得できるようになりました。

クリック座標は MouseX や MouseY システム変数に返されます。座標はピクチャの左上を (0,0) として、ピクセル単位であらわされます。この座標は On Clicked または On Double Clicked フォームイベントで取り出さなくてはなりません。

このメカニズムを正しく動作させるには、表示フォーマットが"トランケート（中央合わせしない）に設定されていなければなりません。

このピクチャマップと同様のメカニズムは、例えばスクロール可能なボタンバーや地図製作ソフトウェアインターフェースで使用することができます。