



# 4D v11 SQL Release 5 (11.5)

## ADDENDUM

---

4D v11 SQL Release 5 によろこそ。このドキュメントではこの最新バージョンの新機能や変更点について説明します。

### 新しいプラットフォームのサポート

4D v11 SQL Release 5 は以下のオペレーティングシステムをサポートします：

- Mac OS X Snow Leopard
- Windows 7

### ランゲージ

新しい [GET CACHE STATISTICS](#) コマンドは 4D キャッシュの状態に関する情報を提供します。また [SET DATABASE PARAMETER](#), [Get database parameter](#)、[Application version](#) そして [REDRAW](#) コマンドが更新されました。

#### GET CACHE STATISTICS

GET CACHE STATISTICS (infoType; arrNames; arrValues; arrCount)

引数	型		説明
infoType	倍長整数	→	取得する情報のセレクタ
arrNames	テキスト配列	←	情報のタイトル
arrValues	実数配列	←	情報の値
arrCount	実数配列	←	関連するオブジェクトの数 (利用可能な場合)

新しい [GET CACHE STATISTICS](#) コマンドを使用して、4D のデータキャッシュに関連する情報を取得できます。この情報はアプリケーションの動作を検証するために使用できます。

infoType 引数には取得したい情報タイプの指定する値を渡します：

- 1 = 一般的なメモリ情報。この情報は物理、仮想、空き、使用メモリサイズ等、ランタイムエクスペローラでも見ることができるものです。
- 2 = データベースキャッシュの占有に関する統計のサマリ。

これらの値を加算して、すべての情報を一度の呼び出しで取得することもできます。

コマンド実行後、リクエストされた統計は arrNames、arrValues、そして arrCount 配列に返されます。

このデータの解釈に関する詳細情報は、お住まいの地域の技術サポート部にお問い合わせください。

テーマ : 4D 環境

## SET DATABASE PARAMETER, Get database parameter

2つの新しいセレクタ (定数) が SET DATABASE PARAMETER と Get database parameter コマンドで利用できるようになりました。

**Selector = 53 (Server Base Process Stack Size)**

- 値 : 正の倍長整数
- 説明: サーバ上のプリエンティブシステムプロセス毎に割り当てるスタックのサイズ (バイト単位) です。デフォルトでこの値は 1 000 000 (1 MB) です。  
プリエンティブシステムプロセスはメインの 4D クライアントプロセスを制御するためにロードされます。デフォルトでそれぞれのプリエンティブプロセスに割り当てられるサイズはおよその場合最適なサイズですが、何百ものプロセスが作成されるようなケースではこのサイズが適切かどうか検討する必要が出てくるかもしれません。  
データベースが実行する処理がそれを許す限り、最適化の目的でこのサイズを大幅に減らすことができます (例えばデータベースで大量のレコードの並び替えなどを行わない場合)。512 や 256 KB でさえも設定可能です。スタックサイズを小さくしすぎることは致命的であり、4D Server の動作に害を及ぼすことになるので注意してください。このパラメタの設定は注意を持って行い、データベースの利用状況 (レコード数や行う処理など) を考慮しながら行わなければなりません。このパラメタの設定を行うには、On Server Startup データベースメソッドなどを使用してサーバ上でコマンドが実行されなければなりません。

**Selector = 61 (Maximum Temporary Memory Size)**

- 値: 正の倍長整数
- 説明: 4D がそれぞれのプロセスに割り当てることのできる一時的なメモリの最大サイズ (MB)。デフォルトで値は 0 (最大サイズの設定なし) です。

4D はインデックスやソート処理のために特別な一時的メモリを使用します。このメモリは大量の処理を行う間、"標準" キャッシュメモリの保護を意図したものです。これは必要な時にのみ有効になります。デフォルトで、一時的なメモリのサイズは、(システムメモリ設定に基づく) 利用可能なリソースによってのみ制限されます。

このメカニズムはほとんどのアプリケーションで適しています。しかし特定の特別なコンテキスト、特に同時に多数のシーケンシャルソートを行うようなサーバ/クライアントアプリケーションでは、一時的なメモリのサイズが、システムが不安定になるほどに致命的に増加するかもしれません。このような場合は、一時的メモリの最大サイズを設定することで、アプリケーションが正しく動作するようにできます。その代わりに、実行速度に影響が出ます。プロセスに対する最大サイズに達すると、4D はディスクファイルを使用し、そのために処理が遅くなります。

先のようなケースの場合、だいたい 50 MB が一般的なサイズとしてよいと思われます。しかし適切な値はアプリケーションの特性、そして実際の環境でのテスト結果に基づき決定されるべきです。

テーマ: 4D 環境

**Application version**

Application Version { (buildNum{;\*}) } → 文字列

引数	型	説明
buildNum	倍長整数	← ビルド番号
*	*	→ 指定時 = 長いバージョン番号 省略時 = 短いバージョン番号
戻り値	文字列	← エンコードされたバージョン番号

Application version コマンドはオプションの buildNum 引数に追加の情報として、現在称している 4D アプリケーションのビルド番号を返すようになりました。これは内部的なコンパイル番号であり、4D の技術サポートチームにご利用の 4D バージョンを知らせる際に使用できる情報です。

テーマ: 4D 環境

**REDRAW**

REDRAW (object)

セレクション表示モードのリストボックスのコンテキストにおいて、リストボックスタイプのオブジェクトに REDRAW 文を適用すると、オブジェクトに表示されているデータをリフレッシュします。  
この呼び出しは特にセレクション中のレコードのデータ更新が発生した場合に呼び出されなければなりません。

テーマ : ユーザインタフェース

## 4D アプリケーションの操作

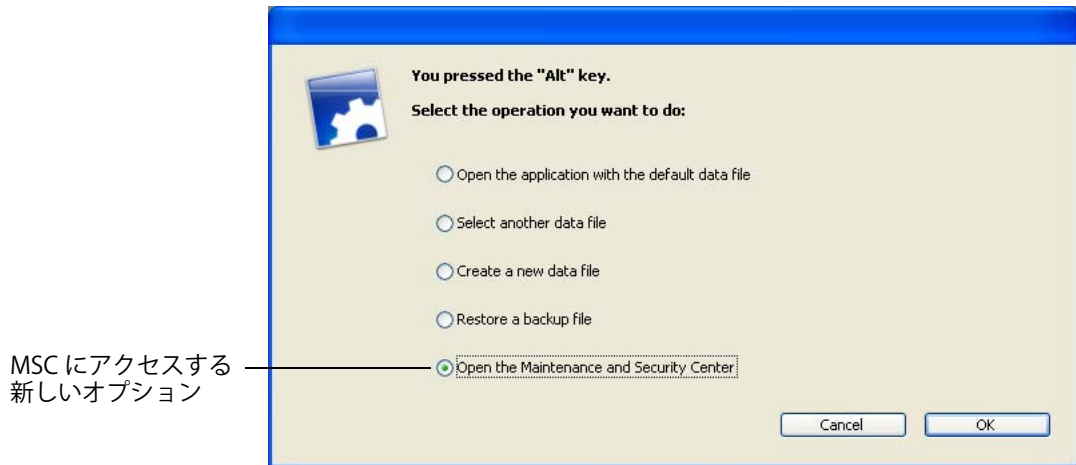
**アプリケーションビルダ (HardLink キー)** バージョン 11.5 より、4D における XML HardLink キーのアプリケーションルールが強化されました。本バージョンより、アプリケーション設定ファイル (デフォルト名が BuildApp.XML) 内でこのキーに値が割り当てられたとき、ビルドされるサーバアプリケーションは、指定されたキー値を持たないクライアントアプリケーションからの接続を拒否します。これは同時に、HardLink キーを持たずにビルドされたクライアントアプリケーションは接続も更新もできないことを意味します。

クライアント / サーバアプリケーションがアプリケーションビルダダイアログボックスでビルドされる時、XML HardLink 要素のみが設定ファイルにデフォルト (空) で挿入されます。HardLink を有効にするには明示的に値を割り当て、BUILD APPLICATION コマンドを使用しなければなりません。

このメカニズムは、サーバアプリケーションでこのキーの値が変更されたとき、クライアントアプリケーションを再ビルドしなければならないことを意味します。

**スタンドアロンアプリケーションからの MSC へのアクセス** 4D スタンドアロンアプリケーション (シングルユーザあるいはクライアントサーバアプリケーション) を開始したとき、直接 Maintenance & Security Center にアクセスするために新しいオプションを利用できます。

このオプションは **Alt** (Windows) または **Option** (Mac OS) キーを押しながらアプリケーションを起動すると表示されるオプションダイアログボックスにあります：



この新しい機能は特に、4D を経由する必要なく、MSC で直接スタンドアロンアプリケーションを修復するために使用できます。

## キャッシュメモリの管理

データキャッシュの管理が 4D v11.5 で最適化されました。特に新しいメカニズムは大抵一時的メモリを消費する処理を実行します (これによりメインキャッシュの使用量が減少します)。一時的メモリの利点は、必要な時にのみ使用され、マシンリソースを要求しないというところにあります。必要であればこのメモリを [SET DATABASE PARAMETER, Get database parameter](#) のセレクタ 61 で設定できます (前述)。

結果アプリケーションのパフォーマンスを犠牲にすることなく、キャッシュに割り当てるサイズを減らすことができます。キャッシュサイズを過大にとりすぎれば全体的なシステムのパフォーマンスが落ち、さらにはシステムが不安定となるリスクが発生します。安全のため、キャッシュの最大サイズは 2.3 GB (2 500 000 KB) に制限されます。

## インデックスの互換性

文字列処理に関して 4D v11 SQL Release 5 で行われた修正のため、テーブルにインデックスが設定された文字フィールドがあり、その中に空の文字と " 無視可能な " 文字 (例えば Char(27)) が含まれる場合、異常な動作が見られるかもしれません。

この場合、MSC を使用してインデックスを検証することをお勧めします。"Table\_n.Field\_n インデックス中、B-tree インデックス n のページ #N に問題があります : キーが正しい順序ではありません" といった結果が表示

された場合、動作を通常に戻すためにはインデックスを再作成する必要があります。無視可能な文字すべてが問題を引き起こすわけではない点に留意してください。

## テキスト比較 (日本語)

---

この新しい動作はデータ言語が日本語に設定された 4D v11 データベースにのみ関連します。

---

最適化のため、日本語のデータファイル利用時にクエリやソートで使用されるテキスト比較のメカニズムが、4D v11 release 5 で変更されました。

互換性に関する注意 データファイルの言語は環境設定のデータベース/インターナショナルページ内、"テキスト比較の言語" で設定できます。

---

4D v11 SQL ではテキスト比較のアルゴリズムに ICU (Unicode) ライブラリを使用しています。このアルゴリズムについては以下のページで説明されています:

<http://userguide.icu-project.org/collation/concepts>

規則として、4D は 2 つの文字列の同等性を " 第一レベル " で、ソートを " 第三レベル " でテストします。

Release 5 より、データベースで日本語が選択されている場合、同等性のテストは " 第二レベル " で、ソートは " 第四レベル " かつひらがなを先に配置する比較を使用して行います。

この場合、4D はキーワードのインデックスも、ICU でない特別なアルゴリズムを使用して作成します。

これらの新しい動作を既存の日本語データベースで有効にするためには、インデックスを再作成する必要があります。この処理は自動では行われません。インデックスの再作成を行うには環境設定ダイアログボックスを開き、データベース/インターナショナルページでテキスト比較の言語が現在のデータベースで日本語になっているのを確認して、OK ボタンをクリックします。次回データベースを開く際にインデックスを再作成する旨のダイアログが表示されるので、OK をクリックし、データベースを再起動します。

互換性に関する注意 この操作は 11.5 で設定したデータファイルを 11.4 以前の 4D で開く際にも必要です。

---