



4D v11 SQL Release 6 (11.6)

ADDENDUM

4D v11 SQL Release 6 によろこそ。このドキュメントではこのバージョンの新機能や変更点について説明します。

暗号化能力の向上

4D v11 SQL Release 6 は新しい 2 つの機能により、暗号化に関連する 4D アプリケーションの能力を拡張します：

- 新しいデータベースパラメーターを使用して、4D が使用する暗号リストを変更できます。
- GENERATE ENCRYPTION KEYPAIR コマンドを使用して 2048 ビット長までのキーを生成できます。

SET DATABASE PARAMETER, Get database parameter

新しいセレクター (定数) が SET DATABASE PARAMETER と Get database parameter コマンドで利用できます。

Selector = 64 (SSL Cipher List)

- 値: 文字
- 説明: このセレクターを使用して、4D が SSL プロトコルで使用する暗号リストを変更したり取得したりできます。例えばこのセレクターを使用して SSL 3.0 暗号化アルゴリズムを実装でき、そして SSL 2.0 による接続を拒否できます。暗号リストはコロンで区切られた一連の文字列です: "RC4-MD5:RC4-64-MD5:...."
この設定はアプリケーション全体に適用されます (HTTP サーバーや SQL サーバー、および SSL プロトコルを使用するすべての 4D の機能に関連します) が、その効果は一時的であり、セッションをまたいで保持されません。
暗号リストが変更されると、新しい設定を有効にするために、関連するサーバー (Web サーバーや SQL サーバー、あるいは 4D Server のサーバー機能) を再起動しなければなりません。

暗号リストを (SLI ファイルに恒久的に格納された) デフォルト値に再設定するには、value 引数に空の文字列 ("") を渡して SET DATABASE PARAMETER コマンドを呼び出します。

デフォルトで、4D は RC4 暗号化アルゴリズムを使用します。(より最新の) AES アルゴリズムを使用するには、value 引数に以下の文字列を渡します: "AES:ALL:!aNULL:!eNULL:+RC4:@STRENGTH"

Note: Get database parameter コマンドを使用すると、暗号リストはオプションの stringValue 引数に返され、戻り値は常に 0 です。

- ▶ 4D が SSL 3.0 のみで動作し、SSL 2.0 接続を拒否するようにする設定:

C_TEXT(\$ssl3)

\$ssl3:="NULL-MD5:NULL-SHA:EXP-RC4-MD5:RC4-MD5:RC4-SHA:

EXP-RC2-CBC-MD5:IDEA-CBC-SHA:EXP-DES-CBC-SHA:DES-CBC-SHA:

DES-CBC3-SHA:EXP-EDH-DSS-DES-CBC-SHA:EDH-DSS-CBC-SHA:EDH-DSS-

DES-CBC3-SHA:EXP-EDH-RSA-DES-CBC-SHA:EDH-RSA-DES-CBC-

SHA:EDH-RSA-DES-CBC3-SHA:EXP-ADH-R"+"C4-MD5:ADH-RC4-MD5:

EXP-ADH-DES-CBC-SHA:ADH-DES-CBC-SHA:ADH-DES-CBC3-SHA"

SET DATABASE PARAMETER(SSL Cipher List;\$ssl3)

GENERATE ENCRYPTION KEYPAIR

GENERATE ENCRYPTION KEYPAIR コマンドを使用して、4D は 2048 ビットの RSA キーを生成できるようになりました。これを行うには length 引数に 2048 を渡します。

Web エリア

WA Execute JavaScript

(Mac OS と同様) Windows において、WA Execute JavaScript コマンドは JavaScript コードの結果を戻すようになりました。

論理ミラーアーキテクチャの変更

4D v11 Release 6 より、論理ミラーを通じたデータバックアップソリューションの実装を可能にするメカニズムが変更されました。以下の原則が適用されるようになりました。

今後は、ミラーマシンにおけるカレントログファイルを無効にすることを推奨します。これを行うには、4D Server ミラーの環境設定ダイアログを開き、"バックアップ / 設定" ページを表示します。"ログファイルを使用" オプションのチェックを外し、続けて表示される警告ダイアログ中で停止ボタンをクリックして、処理を確認します。

この原則の実装を行うために、カレントログファイルを有効にしなくても INTEGRATE LOG FILE コマンドを使用できるようになりました。

これらの新しい原則により、以前のバージョンの 4D v11 SQL で発生していた、事故発生時のログファイルの非同期に関するリスクの可能性を避けることができます。すでに 4D v11 SQL で論理ミラーソリューションを実装している場合、サーバーとミラーマシンに Release 6 をインストールし、後述するシナリオに沿って開発の変更を行うことを強くお勧めします。

新しい処理シナリオ

ミラーによるバックアップシステムを実装する典型的なシナリオは、以下ようになります (ゴシック体のテキストは、以前のバージョンから変更された点を示します):

Step	本番環境	ミラーマシン
1	<ul style="list-style-type: none"> - アプリケーションを開始。 - データファイルをバックアップ。 - (まだそうしていなければ) ログファイルを有効にする。4D は MyDatabase.journal ファイルを作成。安全のために、このファイルは別のハードディスクに格納します。 - アプリケーションを終了。 - (ログファイルを含む) すべてのデータベースファイルをミラーマシンにコピー。 	

2	<ul style="list-style-type: none"> - アプリケーションを再起動 (フルバックアップがプログラムされていないことを検証)。 - 実行を開始。 	<ul style="list-style-type: none"> - ミラーアプリケーションを開始。4D Server はカレントログファイルを要求 : 本番環境から転送した MyDatabase.journal ファイルを選択。 - 環境設定のバックアップ / 設定ページでカレントログファイルを無効にする。
3	<ul style="list-style-type: none"> - ミラーの更新を決定する (例えば特定の時間経過後)。 - New log file コマンドを含むメソッドを実行。保存されるファイル名は MyDatabase[0001-0000].journal。 - (4DIC や Web サービスなど) プログラムを使用して MyDatabase[0001-0000].journal ファイルをミラーマシンに送信。 - データベース実行。 	
4		<ul style="list-style-type: none"> - 統合待ちのファイルを検知。 - INTEGRATE LOG FILE コマンドを含むメソッドを実行して MyDatabase[0001-0000].journal ファイルを統合。
5	<ul style="list-style-type: none"> - マシン上で事故発生。データファイルが利用不可になる。ミラーマシンへの移行を決定。 - MyDatabase.journal カレントログファイルをミラーマシンのいつもの送信先フォルダにコピー。 	
6	<ul style="list-style-type: none"> 事故を分析して、修復。 	<ul style="list-style-type: none"> - 統合待ちのファイルを検知。 - INTEGRATE LOG FILE コマンドを含むメソッドを実行して MyDatabase.journal ファイルを統合。 - 念のため、環境設定のバックアップ / 設定ページでカレントログファイルを作成する。 - データベースが動作。

7	<ul style="list-style-type: none">- マシンが復旧。- データベースファイルをミラーデータベースのものと入れ替える。- アプリケーションを開始。4D Server がログファイルを要求。ミラーデータベースから転送したログファイルを選択する。	<ul style="list-style-type: none">- データベースを終了。- Step2 に戻る。
---	---	---

