4th Dimension 2004.3 追加／修正情報

4th Dimension バージョン 2004.3 へようこそ。本ドキュメントは、4th Dimension プロダクトラインのアプリケーションおよびプラグインの新機能と変更点について説明します。

- **バックアップモジュール：**バックアップモジュールが統合され、論理ミラー、新規コマンドおよび新規オプションなどいくつかの新機能が 4th Dimension に加わりました。
- ランゲージ：**SET DATABASE PARAMETER** および **SET PRINT OPTION** コマンドが変更されました。**SHOW LISTBOX SCROLLBAR** コマンドはコマンド名が **SET SCROLLBAR VISIBLE** コマンドに変更されました。
- 4D Write ：バージョン 2004.3 では内罫線やズーム機能についての動作が変更されました。

バックアップモジュール

4th Dimension バージョン 2004.3 では、バックアップモジュールの統合により、以下の機能やオプションが強化されました。

- ２つの新しいコマンドを使った論理ミラーによるバックアップ設定
- バックアップが成功しなかった場合の新規のバックアップ環境設定
- ログファイルが限界値（critical size）に達すると作動する新たなメカニズムの導入
- **New log file** 関数を採用

論理ミラー

4D Server では、論理ミラーによるバックアップシステムの設定が可能な統合されたソリューションを提供します。このソリューションは、次の２つの新しいコマンドに基づいています。**New log file** 関数および **INTEGRATE LOG FILE** コマンドです。

論理ミラーとは？

論理ミラーは、洗練されたバックアップモードで、主に重要なデータベースや高負荷なデータベースに使用します。

論理ミラーを使用するには、一方のコンピュータでデータベースが操作中の間、そのデータベースのコピーを別のコンピュータで定期的に更新します。双方のコンピュータは操作中のコンピュータのネットワークを通して通信し、データベースで行われたすべての変更は、ログファイルを介して定期的にミラーコンピュータに送信されます。

この方法により、操作中のデータベースに影響を与える問題が生じた場合、ミラーデータベースを使用し、データを失うことなく素早く復元できます。また、バックアップ作業によって操作中のデータベースが“ブロック（妨害)”されることはありません。

なぜ論理ミラーでバックアップした方がよいのか？

論理ミラーを使用することで特定の問題を解決できます。定期的なデータのバックアップとログファイルの使用が基本となる標準のバックアップ方式は、ほとんどの場合、簡単で信頼性も高く、低価格なソリューションを提供することができます。データベースは定期的（通常24時間ごと）にバックアップされ、バックアップ中もread-onlyモードでデータベースを利用できます。データベースが一部利用できない状態になるのは非常に短時間です。サイズが大きいデータベース（2GB以上）でもバックアップに5分以上かかることはありません。このバックアップの作業は、通常データベースを使用する時間以外に設定してプログラムすることもできます。

それでも、例えば病院のような特定の施設では、重要なデータベースが24時間一日中完全に利用できる状態でなければなりません。データベースは、例え短時間でもread-onlyモードにできません。この場合、論理ミラーを設定することで問題を解決することができます。

注：ミラーデータベースにはデータへの変更のみが反映されます。このため、開発中のデータベースに、このバックアップモードは不適当です。頻繁にストラクチャを修正すると、ミラーデータは短期間で陳腐化されてしまいます。またミラーデータベースストラクチャを繰り返し更新しなくてはなりません。

論理ミラーの仕組み

論理ミラーの機能は、4D Serverでのみ利用できます。

論理ミラーによるバックアップは次の２つのコマンドを使って設定できます。**New log file**関数および**INTEGRATE LOG FILE**コマンドです。

以下の原則が適用されます。

■ メインの4D Serverマシン（操作中のマシン）にデータベースをインストールし、同一のデータベースのコピーを4D Serverのミラーマシンにインストールします。

■ アプリケーション起動時のテスト（4D Extensionsのフォルダの中に特定のファイルが存在するかなど）により、２つのバージョン（操作中のものとミラー版）を区別し、適切にバックアップが実行されるようにします。

■ 操作中の4D Serverマシンでは、**New log file**関数によりログファイルが定期的な間隔で"セグメント化"されます。メインサーバでのバックアップが実行されないため、データベースは常にRead-Writeモードで利用できます。

■ ログファイルの各"セグメント"はミラーマシンに送られ、**INTEGRATE LOG FILE**コマンドによりミラーデータベースに統合されます。

このシステムを設定するには、次のような特別なメソッドをプログラムする必要があります。

■ **New log file**関数の実行サイクルを管理するメインサーバ用のタイマー

■ ログファイルの「セグメント」を操作中のマシンとミラーマシンの間で移動させるシステム（FTPもしくはメッセージシステム、Web Services、4D Open for 4Dなどを利用してファイルを移動する場合には、4D Internet Commandsを使います。）

■ 新しく到着したログファイルの"セグメント"を管理し、**INTEGRATE LOG FILE**コマンドによりデータベースに統合させるためのミラーマシン上のプロセス

■ メインサーバとミラーサーバ間の通信やエラー処理を行うシステム

論理ミラーによるバックアップと"標準の"バックアップを同時に使用しないでください。２つのバックアップモードを同時に使用すると、ミラーデータベースが操作中のデータベースに同期化されなくなる可能性があります。論理ミラーを使用する際には、操作中のコンピュータやミラーコンピュータに自動または手動でのバックアップが実行されていないことを確認してください。

New log file

New log file → Text

引数	タイプ		説明
Function result	テキスト	←	閉じられたログファイルのフルパス名

テーマ：Backup

このコマンドは4D Serverのみに使用できます。**Execute on server**コマンドもしくはストアドプロシージャ内でのみで使用することができます。

New log file関数はカレントログファイルを閉じて、そのファイルに別の名前を付けます。そして、以前と同じ名前の新しいログファイルを前のファイルと同じ場所に作成します。この関数は論理ミラーを使用したバックアップシステムを設定するために使用されます（前述の「論理ミラー」の節を参照）。

この関数は、閉じられたログファイル（または「セグメント」）のフルパス名（アクセスパス+パス名）を返します。このファイルは、カレントログファイルと同じ場所に保存されます（これは、「環境設定」の「バックアップ」テーマ内の設定ページで指定します）。この関数は、保存されたファイルではいかなるプロセス（圧縮やセグメント化）も実行しません。ダイアログボックスは表示されません。

ファイルは、データベースやログファイルのカレントバックアップ番号を使用した別名に変更されます。次のようになります。

DatabaseName[BackupNum-LogBackupNum].4DL

■ 例えば、MyDatabase.4DDというデータベースを4回保存すると最後のバックアップファイルの名前は、MyDatabase[0004].4BKとなります。従って最初のログファイルの"セグメント"は、MyDatabase[0004-0000].4DLです。

■ 例えば、MyDatabase.4DDというデータベースが3回保存され、ログファイルが5回保存された場合、6度目に実行されるログファイルのバックアップは、MyDatabase[0003-0005].4DLです。

このコマンドを実行する前に、4D Serverは他に重要な動作（トランザクション や インデックス作成）が実行されていないかを調べます。実行中の動作が見つかった場合、4D Serverは「環境設定」の「バックアップ」テーマ内の「バックアップ」ページで設定された待ち時間に従います。エラーが発生した場合、**ON ERR CALL**コマンドを使用し、エラー処理を置き換えるメソッドを記述することが可能です。

参照：INTEGRATE LOG FILE

INTEGRATE LOG FILE

INTEGRATE LOG FILE(pathName)

引数	タイプ		説明
pathname	テキスト	→	統合されるログファイルのパス名

テーマ：Backup

このコマンドは4D Serverのみに使用できます。**Execute on server**コマンドもしくはストアドプロシージャ内でのみで使用することができます。

INTEGRATE LOG FILEコマンドは、引数＜pathName＞に渡された名前やパス名のログファイルをカレントデータベースに統合します。統合されたファイルがデータベースの新しいカレントログファイルになります。このコマンドは論理ミラーを使用したバックアップシステムを設定するために使用されます（前述の「論理ミラー」の節を参照）。

唯一保存されないログファイルデータ（extension.4DL）は、このコマンドを使用して統合します。ダイアログボックスは表示されませんが、プログレスバーが画面に表示されます。

引数＜pathName＞には、絶対パス名またはデータベースフォルダからの相対パス名を渡すことができます。

空の文字列を渡すと、標準のファイルを開くダイアログボックスが表示され、ユーザはどのファイルを統合するか指定できます。このダイアログボックスをキャンセルした場合、ファイルは統合されず、システム変数OKには0が代入されます。

エラーが発生した場合、**ON ERR CALL**コマンドを使用し、エラー処理を置き換えるメソッドを記述することが可能です。ロックされたレコードがデータベース内に存在する場合、コマンドは何も実行しません。エラーコード1420が生成されます。

注：このコマンドを使用する場合、以下については開発者の判断で行います。

・ミラーマシンにミラーデータベースをインストールし、データファイルが**INTEGRATE LOG FILE**コマンドによるデータベースの統合以外で変更されることのないようにする。ミラー版のデータベースかどうかを調べるには、ファイルを4D Extensionフォルダもしくはデータベースフォルダに置き、On Startupデータベスメソッドなどの実行中にファイルが存在するかどうかをテストする。ファイルが存在する場合、ミラーモードは有効になっている。

・操作中のデータベースとミラーデータベースの間に通信システムをセットアップし、ログファイルセグメントの送受信をオーガナイズする。

・2つのデータベース間で起こりうる送信エラーを処理する。

参照：New log file

論理ミラーの使用方法

以下の例では、ミラーによるバックアップシステムの設定の流れをそれぞれの4D Serverマシンごとに表記します。

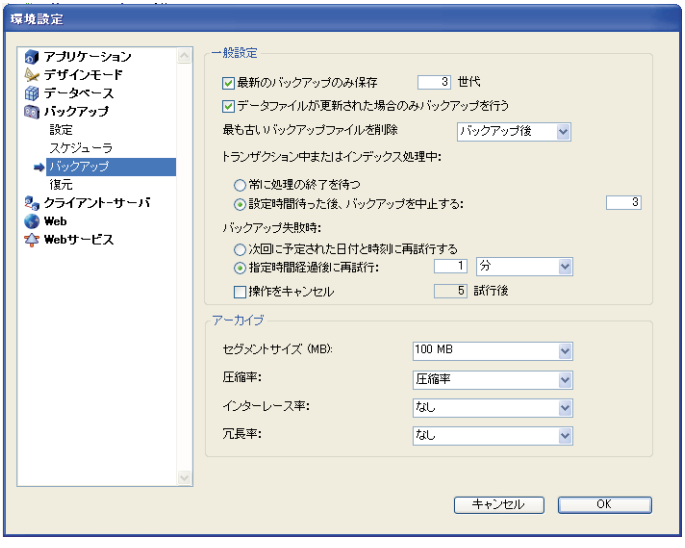
操作中のマシン	ミラーマシン
	

アプリケーションの起動。データファイルのバックアップおよび（必要に応じて）ログファイルのアクティベーションを行います。MyDatabase.4DLファイルが作成されます。	
アプリケーションの終了。	
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。	
アプリケーションの起動。動作開始。	ミラーアプリケーションの起動。ログファイルの選択を要求されます。 操作中のデータベースから送信されたMyDatabase.4DLファイルを選択してください。
ミラー更新の決定（例えば、ある一定の操作の後など）。	
New log fileコマンドを含むメソッドの実行。保存されるファイルの名前はMyDatabase[0001-0000].4DLです。	
MyDatabase[0001-0000].4DLファイルは、（4DIC、4D Open for 4Dなどを使用した）プログラムによりミラーマシンへ送信。	
データベースは操作中の状態。	統合されるファイルの検索。**INTEGRATE LOG FILE**コマンドが含まれるメソッドを実行しMyDatabase[0001-0000].4DLファイルを統合します。今度は、このファイルが新しいカレントログファイルになります。
マシン上で問題が発生し、データベースが使用できない状態。ミラーマシンへの切り替えが必要です。	
通常のコピー先フォルダを使い、カレントログファイルMyDatabase.4DLをミラーマシンにコピーします。	
マシンの修復。	統合されるファイルの検索。**INTEGRATE LOG FILE**コマンドが含まれるメソッドを実行しMyDatabase[0001-0000].4DLファイルを統合します。今度は、このファイルが新しいカレントログファイルになります。

	データベースが操作中の状態。
マシンの修復完了。	データベースの終了。
データベースファイルがミラーデータベースのファイルに差し替えられる。	
アプリケーションの起動。 ログファイルの選択を要求されます。 ミラーデータベースから送信されたMyDatabase.4DLファイルを選択してください。	ミラーアプリケーションの起動。
データベースが操作中の状態。	

新規のバックアップ環境設定

新規の環境設定では、「操作をキャンセル」チェックボックスによりバックアップの試行回数に上限を設定できるようになりました。



この設定は、「指定時間経過後に再試行」のオプションをチェックした場合のみ使用できます。このパラメータは、バックアップモジュールがバックアップを再試行する回数を設定します。

設定された限度を越えてもバックアップが成功しない場合、操作をキャンセルしてエラーメッセージ1401を生成します。この場合、アプリケーションが再起動されるか手動でのバックアップが成功しない限り、自動バックアップが再試行されることはありません。

この設定は、（人的介入を要する）大きな問題が生じて、バックアップが実行されなくなったにもかかわらず、アプリケーションが繰り返しバックアップを試行することで、全体のパフォーマンスに支障が生じることを防ぐことができます。

デフォルトでは、このパラメータはチェックされません。

ログファイルの自動バックアップ

4th Dimensionバージョン2004.3には新しいメカニズムが追加されました。ログファイルが2GBに近づくと、カレントログファイルを終了して新しいファイルを開くために、データベースは自動的にバックアップを開始します。これにより、ファイルのサイズによってアプリケーションのパフォーマンスが変更されることを防ぎます。

この場合、実行されるバックアップにはカレントバックアップパラメータが使われます。手動バックアップでも同じです。

Log File

Log File→文字列

引数	タイプ		説明
このコマンドに引数は不要です。			
Function result	文字列	←	データベースログファイルのフルパス名

新しい**Log File**関数は、開いているデータベースのカレントログファイルのフルパス名（ドライブ名から始まるフォルダ-、ファイル名）を返します。

操作中のデータベースにログファイルがない場合、コマンドは空の文字列を返し、システム変数OKには0が代入されます。

操作中のデータベースにログファイルがある場合、システム変数OKには1が代入されます。コマンドに返されるパス名はカレントプラットフォームのシンタックスで表記されます。

注：4D Clientマシンからこのコマンドを実行した場合、ロングネームではなくログファイル名のみが返されます。

テーマ：Backup

ランゲージ

4th Dimensionバージョン2004.3のランゲージコマンドには、論理ミラーを管理するためのコマンドが追加されたほか、様々な変更が加わりました。

メソッドエディタ

メソッドエディタ上に記述したプロジェクトメソッド名の上でAltキー+ダブルクリック（Windows）もしくはOptionキー+ダブルクリック（Mac）というショートカットを使用すると、新しいウインドウで直接そのプロジェクトメソッドが開きます。これは、従来のプロジェクトメソッド名を文字列選択した後に、Ctrlキー+"P"（Windows）またはコマンドキー＋"P"（Mac）のショートカットとなります。

プロセスのスタックサイズ

プロセスのスタックサイズに割り当てられたメモリの内部管理（internal management of the memory）が4th Dimensionバージョン2004.3で変更されました。このバージョンでは、各プロセスのスタックに64,000バイト以上割り当てるよう推奨されています。この値を下回ると、「スタックスペースが足りません。カレントメソッドを完了できません。」というエラーが表示されます。

プロセスのスタックサイズは**New process**関数および**Execute on server**関数で設定します。以前のバージョンでは、このパラメータは動的に処理されていて、事実、要求されるサイズより大きいサイズが一般的に使われていました。これにより、開発のフレキシビリティは高まりました。しかし、特に4D Serverでは、使用頻度が多くなるとアプリケーションのパフォーマンスが著しく低下する恐れがありました。今後は、設定されたサイズのみが使用されるため、場合によってはスタックサイズの調整が必要になります。

SHOW LISTBOX SCROLLBAR

SHOW LISTBOX SCROLLBARコマンドは、適用範囲の拡大に伴い、コマンド名を**SET SCROLLBAR VISIBLE**に変更し、「List Box」テーマから「オブジェクトプロパティコマンド」テーマに移動しました。

このコマンドは、スクロールバーを含むいくつかのオブジェクトに使用できます。

■ リストボックス

■ スクロール可能なエリア

■ サブフォーム

水平値および垂直値の両方が要求されます。スクロール可能なエリアでは、水平値を渡さなければなりませんが、渡された値は無視される点に留意してください。

SET SCROLLBAR VISIBLE

SET SCROLLBAR VISIBLEコマンドは**SHOW LISTBOX SCROLLAR**コマンドの新しいコマンド名です（前述参照）。

テーマ：オブジェクトプロパティコマンド

SET DATABASE PARAMETER、GET database parameter

SET DATABASE PARAMETER({table;}selector;value)

GET database parameter({table;}selector)→倍長整数

新しいSelector定数が利用できるようになりました。

セレクタ	値	適用範囲
Debug Log Recording	34	4Dアプリケーション

■ Selector=34(Debug Log Recording)

有効な値：0、1または2（0=記録しない、1=記録する、2=詳細モードに記録する（record in detailed mode）

説明：アプリケーションのデバッグを行うために4Dのプログラミングレベルで発生する連続保存のイベントを開始または停止します。デフォルト値は0（イベントを記録しない）です。記録される情報には、以下のように様々な内容が含まれます。

■ 各イベントについては、ファイル番号およびプロセス番号（[n]）が作成されてからの時間（ミリセカンド単位で記録）。

■ 実行された4Dコマンド（[cmd]）および呼び出された各プラグイン名（plugInName）。この場合、スタックレベルが指定されます（[n]）。

■ 呼び出されたそれぞれのプロジェクトメソッド（meth）、オブジェクトメソッド（obj）およびフォームメソッド（form）。

■ 詳細モードが有効な場合（値=2）、プラグインエリア（EventCode）のイベントおよびプラグインによる4Dの呼び出し（externCall）についての追加情報を記録します。

イベントは"4DDebugLog.txt"という名前のファイルに保存され、自動的にデータベースのストラクチャファイルと同じ階層に置かれます。それぞれのイベントは、実行される前にシステム的にファイルに記録されるようになっています。このため、アプリケーションが予期せず終了するようなことがあっても、イベントの存在がファイルに残ります。このファイルはアプリケーションの起動時に毎回上書きされるので、注意してください。

コンパイルモードもしくはインタプリタモードのすべての4Dアプリケーション（4th Dimension、4D Server、4D Client、4D Runtime）でこのオプションを有効にすることができます。

このオプションは、デバッグのみの用途で提供されています。アプリケーションパフォーマンスやハードディスクの彩度が低下する恐れがあるので、製品としては使用しないでください。

以下の例は、デバッガファイルの内容です。

-- Startup on Friday, October 07, 2005 03:29:26 PM--

0[1] Log level: 1
250[1] form: Output; event: OnLoad
250[1] end_form
7050[1] obj: vRecNum; event: OnLoad
7083[1] (1) cmd: Current form table.
7083[1] (1) cmd: Selected record number.
7083[1] (1) cmd: Current form table.
7083[1] (1) cmd: Selected record number.
7083[1] (1) cmd: String.
7083[1] (1) cmd: Current form table.
7083[1] (1) cmd: Records in selection.
7083[1] (1) cmd: String.
7083[1] end_obj
7083[1] form: Input; event: OnLoad
7083[1] end_form
8900[1] obj: Button1; event: OnClicked
8900[1] (1) plugInName: _4D_Pack; cmd: AP AVAILABLE MEMORY.
8900[1] end_obj
10250[1] obj: Button2; event: OnClicked
10250[1] (1) cmd: ALERT.
11666[1] end_obj
18983[6] 4DActionUrl: /4DACTION/testWeb
18983[6] onWebAuthenticationCall: /4DACTION/testWeb
18983[6] end_onWebAuthenticationCalll
18983[6] meth: testWeb

18983[6] (1) cmd: ALERT
19000[6] end_meth
19000[6] end_4DActionUrl
19883[6] 4DActionUrl: /4DACTION/testWeb
19883[6] onWebAuthenticationCall: /4DACTION/testWeb
19883[6] end_onWebAuthenticationCall
19883[6] meth: testWeb
19883[6] (1) cmd: ALERT.
19900[6] end_meth
19900[6] end_4DActionUrl

テーマ： ストラクチャアクセス

SET PRINT OPTION、GET PRINT OPTION

SET PRINT OPTION（option;value1{;value2}）

GET PRINT OPTION（option;value1{;value2}）

4th Dimensionバージョン2004.3では、新しい定数のオプションが利用できるようになりました。

オプション（定数）	value1	value2
印刷の進捗状況を非表示にするオプション（14）	0=表示する（デフォルト）、1=非表示にする	

この新しいオプションにより、ユーザはカレントプロセスのすべての印刷の進捗状況を示すウインドウを非表示にすることができます。印刷の進捗状況を非表示にしたい場合は、＜value1＞に1を代入し、表示する状態に戻したい場合は0（デフォルト値）を代入します。このオプションは、特にMac OS XでPDFファイルに出力する場合などに便利です。

注：環境設定のダイアログボックス内（「アプリケーション／オプション」ページ）には、既に印刷の進捗についての表示オプションがあります。ただし、これはアプリケーション全体に適用されるもので、Mac OS Xではすべてのウインドウが非表示にはなりません。

テーマ：印刷コマンド

4D Write

4D Write 2004.3では、次の点が変更されました。

■ 内罫線の扱い方

■ エリアオープン時のズーム値（zoom value）の設定

■ タブの削除に使用する新しいボタンの追加

ドキュメントの互換性

バージョン2004.3以前の4D Writeで作成されたドキュメントは、すべて新しいバージョンのプラグインに対応しています。ただし、内部に保存されるドキュメントのフォーマットが変更されたため、バージョン2004.3で作成されたドキュメント、もしくは2004.3に変換されたドキュメントを以前のバージョンのプラグインで読み込むことはできません。

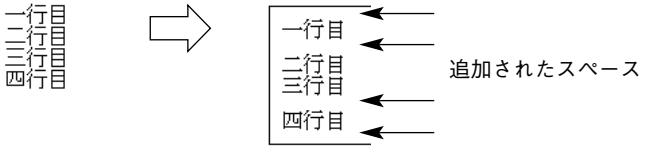
罫線の扱い方

4D Write 2004.3では、段落の内罫線の扱い方が変わりました。この変更には「罫線」のツールバーの中の新規のアイコンや新しい定数などが含まれます。

新しい動作

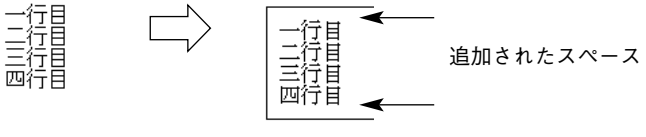
以前の4D Writeでは、罫線を挿入すると各段落の上下にスペースが追加されました。以下の例では、「全罫線」オプションが、選択された4段落に適用されています。

以前の4D Writeバージョン



このことで問題が生じる場合がありましたが、現在の4D Writeでは罫線を使用してもその各段落の上下にはスペースが追加されず、段落全体の一番上と一番下にのみスペースが追加されます。

以前の4D Writeバージョン



新しい動作は、「罫線」のツールバーの「上罫線」および「下罫線」ボタンを使用する際にも適用されます。

互換性の理由から、以前のメカニズムも残されています。新しいバージョンでは、「罫線」のツールバーに追加された新規の「内上罫線」や「内下罫線」ボタンからアクセスできます。

WR Text properties 定数

罫線の動作の変更に伴い、「WR Text properties」テーマの定数も変わりました。

■ 既存の定数（43および44）はそれぞれwr inside top borderおよびwr inside bottom borderと名前が変更しました。

これらの定数の動作（段落の上下にスペースを追加）は、以前のバージョンの4D Writeと同じです。これにより、既存のドキュメントとの互換性が保たれます。

■ ２つの新しい定数が追加されました。wr top border(46)およびwr bottom border(47)です。

これらの定数は、4D Write 2004.3で新しくなった上下の罫線の動作に対応します。

WR Commands 定数

罫線の動作の変更に伴い、「WR Commands」テーマの定数も変更されました。

■ 既存の定数（1006および1008）はそれぞれwr cmd inside top borderおよびwr cmd inside bottom borderに名前が変更されました。

これらの定数は、4D Write 2004.3の「罫線」のツールバーの新しいボタンに対応します。これらの定数の動作（段落の上下にスペースを追加）は、以前のバージョンの4D Writeと同じです。これにより、既存のドキュメントとの互換性が保たれます。

■ ２つの新しい定数が追加されました。wr cmd top border(1015)およびwr cmd bottom border(1016)です。

これらの定数は、4D Write 2004.3で新しくなった上下の罫線の動作に対応します。

オープン時にズーム値（zoom value）を適用

最後にクローズしたときに保存したカスタムズーム値（zoom value）で4D Writeエリアを開くことができるようになりました。

カレントズーム値（zoom value）は、常にエリアに保存されますが、エリアをオープンする際のデフォルトのズーム値（zoom value）は100％です。

前回保存されたズーム値（zoom value）の適用をエリアのオープン時に有効にするためには、新しいwr use saved zoom value(18)定数を**WR SET AREA PROPERTY**コマンドで使用してください。

■ 保存されたズーム値（zoom value）（ドキュメントに保存された値）でエリアを開くには次のステートメントを実行してください。

WR SET AREA PROPERTY（area;wr use saved zoom value;1）

■ ズーム値（zoom value）を100%（デフォルト値）に戻すときには、次のステートメントを実行してください。

WR SET AREA PROPERTY（area;wr use saved zoom value;0）

注：すべての4D Writeエリアにプロパティを適用するには、引数＜area＞に0を代入してください。

タブの削除

フォーマットメニューから呼び出される「タブ...」ダイアログボックスに新しく「全て削除」ボタンが追加されました。このボタンは、ドキュメント内で選択された段落のタブの削除に使用します。