



4D バージョン 6.8.3 ランゲージ追加/修正情報

目次

RECEIVE BUFFER	3
デバッグ	5
「カスタム表現式/値」エリア	5
CANCEL	7
PAGE BREAK	8
Count tasks	9
レコードロックコマンド	10
レコードのロード、更新、アンロード	10
UNLOAD RECORD	12
DISPLAY SELECTION	13
GET DOCUMENT PROPERTIES	17
SET DOCUMENT PROPERTIES	18
Open form window	19

RECEIVE BUFFER

RECEIVE BUFFER (受信変数)

引数	タイプ	説明
受信変数	変数	→ 受信データを格納する変数

説明

RECEIVE BUFFER コマンドは、**SET CHANNEL** コマンドで前もって開いたシリアルポートからデータを読み込みます。シリアルポートは、コマンドで読み込まれるまでバッファの内容を保持しています。**RECEIVE BUFFER** コマンドは、バッファ中の文字を<受信変数>に格納して、バッファを消去します。バッファ中に文字が存在しなければ、<受信変数>は何も含みません。

Windows :

Windows のシリアルポートバッファのサイズは 10KB に制限されています。つまり、バッファがオーバーフローする可能性があるということです。バッファがいっぱいになった後、新しい文字を受信すると、最も古いものと置き換えられます。古くなった文字は失われるため、新しい文字を受信する際は、すみやかにバッファを読み込むことが重要です。

Macintosh :

MacOS 9 のシリアルポートバッファのサイズは 10KB に制限されています。バッファがフルの状態 新しいデータを受け取ると、最も古いデータから順に上書き消去していきます。したがって、バッファに文字が入力された時点で迅速に読み込む必要があります。一方、MacOS X 環境では、バッファのサイズは理論上無制限です（ただし利用可能なメモリサイズに制限されます）。

RECEIVE BUFFER コマンドは、バッファ中のデータが何であれ、それを即座に返す点が **RECEIVE PACKET** コマンドと異なります。**RECEIVE PACKET** コマンドはバッファ中に指定した文字を発見するまで、または指定した数の文字が入力されるまで待ちます。

RECEIVE BUFFER コマンドの実行中に、「Ctrl + Alt + Shift」キー（Windows）または「command + option + shift」キー（Macintosh）を押して、受信を中断することができます。中断することにより、エラー-9994が発生します。**ON ERR CALL** を使用してインストールしたエラー処理メソッドにより、このエラーを検出することができます。

▼ 以下のプロジェクトメソッド **LISTEN TO SERIAL PORT** は、**RECEIVE BUFFER** コマンドを使用してシリアルポートからテキストを取得し、それをインタープロセス変数に追加します。

```
` シリアルポートの受信
` シリアルポート受信開始
SET CHANNEL (201; Speed 9600 + Data Bits 8 + Stop Bits One
               + Parity None) ` シリアルポートのオープン
<>IP_Listen_Serial_Port:=True
While (<>IP_Listen_Serial_Port)
    RECEIVE BUFFER ($vtBuffer)
    If ((Length ($vtBuffer)+Length (<>vtBuffer))>MAXTEXTLEN)
        <>vtBuffer:=""
    End if
        <>vtBuffer:=<>vtBuffer+$vtBuffer
End while
<>Buffer:=""
<>Len=0
```

この時点で、他のプロセスからインタープロセス変数 <>vtBuffer を読み込み、シリアルポートから受信したデータの処理を行うことができます。

シリアルポートの監視を中断するには、以下のコードを実行します。

```
` シリアルポート受信の終了
<>IP_Listen_Serial_Port:=False
```

プロセス間でのコンフリクトを避けるために、セマフォを利用してインタープロセス変数 <>vtBuffer へのアクセスを保護する必要がある点に注意してください。詳細は **Semaphore** コマンドを参照してください。

参照

なし

デバッグ

「カスタム表現式／値」エリア

「メソッド連鎖」エリアのすぐ下にあるのは「カスタム表現式／値」エリアです。このエリアは、式を評価するために使用します。フィールド、変数、ポインタ、演算、組み込み関数、カスタム定義関数等、値を戻すものなら何でも、どのようなタイプの式でも評価できます。

テキスト形式で表示できる式であれば、どのような式でも評価することができます。ピクチャやBLOBのフィールドや変数は、対象になりません。一方、デバッガは、割り当てられた階層リストを使用して、配列やポインタを表示できるようにします。BLOBの内容を表示するには、**BLOB to text** 等のようなBLOBコマンドを使用します。

次の例では、2つの変数、1つのフィールドポインタ変数、組み込み関数の結果、演算の項目のうちいくつかが表示されています。

式	値
OK	1
pField	^[Customers]Company
[Customers]Company	""
Records in selection([Customers])	0
\$\$SearchCriteria	"Z@"

新しい式の挿入

次のように「カスタム表現式／値」エリアに式を追加して、評価することができます。

- 「デフォルト表現式／値」エリアからオブジェクトまたは式をドラッグ&ドロップします。
- 「メソッド連鎖」エリアからオブジェクトまたは式をドラッグ&ドロップします。
- 「ソースコード」エリアで、評価できる式をクリックします。

空の式を作成するには、「カスタム表現式／値」エリアの空白スペースの任意の位置をダブルクリックします。すると、新しい式“新規式”が追加された後、編集モードになり、編集することができます。結果を戻す4th Dimensionのフォーミュラを入力できます。

フォーミュラを入力した後、enterまたはreturnキーを押して（またはエリアの任意の位置をクリックして）、式を評価します。

式を変更するには、その式をクリックして選択し、再びクリックすると（またはテンキー上のenterを押す）編集モードになります。

式が必要でなくなった場合には、その式をクリックして選択し、**Backspace** キーまたは **delete** キーを押します。

警告：システム変数（例えば、'OK'など）の値を変更するような式を実行する時は注意して下さい。その後のプログラムコードの実行結果に影響を与える可能性があります。

CANCEL

CANCEL

説明

CANCEL コマンドは、フォームやオブジェクトのメソッド（またはサブルーチン）で以下の目的のために使用されます。

- **ADD RECORD、MODIFY RECORD、ADD SUBRECORD、MODIFY SUBRECORD** を使って開始されたデータ入力による新規または修正レコードやサブレコードをキャンセルするため
- **DIALOG** コマンドで表示されたフォームをキャンセルするため
- **DISPLAY SELECTION** や **MODIFY SELECTION** コマンドを使って、レコードセレクションを表示しているフォームを終了するため

CANCEL コマンドは、ユーザがキャンセルキー（ESCキー）を押した場合と同じ動作を実行します。フォームをキャンセルした後で、システム変数OKは0に設定されます。

一般に **CANCEL** コマンドは、メニューコマンドの結果として実行されます。また、「動作なし」属性ボタンのオブジェクトメソッド内でもよく使用されます。

また、**Open window** 関数におけるオプションの「クローズボックス」メソッド内でもよく使用されます。あるウインドウにコントロールメニューボックス（Macintosh版では、クローズボックス）がある場合、**ACCEPT** や **CANCEL** コマンドはコントロールメニューボックスがクリックされたり、または「閉じる」メニューが選択された際に実行されるメソッドの中で呼び出されます。

CANCEL コマンドは、待ち行列を作成することができません。あるイベントに対してメソッド内で2つの **CANCEL** コマンドを続けて実行しても、1つの **CANCEL** コマンド実行をした場合と同じ効果しか得られません。

参照

なし

PAGE BREAK

PAGE BREAK {(* | >)}

引数	タイプ	説明
* >		→ * : で開始した印刷ジョブをキャンセル > : 1つのプリントジョブを強制する

説明

PAGE BREAK コマンドは、プリンタへ送信したページを印刷し、排出します。**PAGE BREAK** コマンドは、**PRINT FORM** コマンドとともに使用し、ページブレイクを強制し、最終ページを印刷します。**PAGE BREAK** コマンドは、**PRINT SELECTION** コマンドとともに使用しないでください。この代わりに、**Subtotal** 関数または **BREAK LEVEL** コマンドにオプション引数を使用してページブレイクを行ってください。

< * > と < > > 引数は両方とも省略できます。

< * > 引数により、**PRINT FORM** コマンドによって開始したプリントジョブをキャンセルすることができます。このコマンドを実行すると、進行中のプリントジョブが直ちに中止されます。

注：Windows 環境下では、この仕組みはプリントサーバのスプーリング設定状況によっては機能しないことがあります。プリンタがすぐに印刷を開始するように設定されている場合、キャンセルすることはできないでしょう。**PAGE BREAK(*)** コマンドを正しく機能させるためには、プリンタドライバの“最後のページのスプール後に印刷開始”属性を選択しておくといよいでしょう。

< > > 引数は、**PAGE BREAK** コマンドの振る舞いを変更します。この形式は2種類の効果を持ちます。

■ **PAGE BREAK** コマンドが引数なしで再度実行されるまで、プリントジョブの開始を止めます。

■ プリントジョブに優先権を与えます。プリントジョブが終了するまで、他のプリントは行われません。

2番目のオプションは、スプールされるプリントジョブとともに使用すると、特に有効です。< > > 引数により、プリントジョブは1つのファイルにスプールされます。これは、プリント時間を減少させます。

参照

前述の **PRINT FORM** コマンドの例を参照してください。

Count tasks

Count tasks → 整数

説明

Count tasks 関数は、4D Clientまたは4D Server上で開いているプロセス（ストアドプロシージャ含む）の数を返します。

この数には、4th Dimensionが自動的に管理するものも含めてすべてのプロセスが含まれます。この中には、「ユーザ/カスタムメニュー」プロセス、「デザイン」プロセス、「キャッシュマネージャ」プロセス、「インデックス」プロセス、および「Webサーバ」プロセスが含まれます。

Count tasks 関数により返される数にはアボートされたプロセスも含まれます。

Process state および「On Exit」データベースメソッドの例題を参照してください。

参照

Count user processes、Count users、PROCESS PROPERTIES、Process state

レコードロックコマンド

レコードのロード、更新、アンロード

ローカルユーザがレコードを更新するためには、テーブルがリードライト状態であつ、ロードしたレコードがアンロックになっていなければなりません。

NEXT RECORD、**QUERY**、**ORDER BY**、**RELATE ONE**等のコマンドは、カレントレコードをロードし、そのレコードをロックまたはアンロックにします。

レコードは、テーブルのその時の状態（リードオンリーまたはリードライト）に応じてロードされます。レコードが自動リレートされている場合には、自動リレートを実行するコマンドによってリレート先テーブルからもレコードがロードされます。

テーブルがリードオンリー状態になっている場合には、テーブルからロードされたレコードはロックされます。ロックされたレコードは、保存および削除ができません。リードオンリー状態は他のユーザがレコードをロード、更新、保存等の処理を実行している時には望ましい状態です。

テーブルがリードライト状態になっている場合には、テーブルからロードされたレコードは他のユーザが先にロックしていない限りアンロックになります。アンロックされたレコードは、テーブルに保存することができます。テーブルは、レコードをロード、更新、保存等の処理を実行する前にリードライト状態にしなければなりません。

レコードを修正する場合、**Locked**関数を使って、他のユーザがレコードをロックしていないかどうかを調べます。レコードがロックされている場合（**Locked**関数が**True**の場合）には、レコードを**LOAD RECORD**コマンドでロードし、そのレコードがロックされているかどうかを再び調べます。レコードがアンロックになるまで（**Locked**関数が**False**になるまで）この処理を繰り返します。

レコードへの修正が完了したら、**UNLOAD RECORD**コマンドを使って、そのレコードを解放（他のユーザに対してアンロック）しなければなりません。レコードがアンロードされないと、他のカレントレコードが選択されるまで、すべての他のユーザに対してロックされた状態になります。テーブルのカレントレコードを変えると、前のカレントレコードは自動的にアンロックになります。カレントレコードを変更しない場合は、**UNLOAD RECORD**コマンドを明示的に呼び出す必要があります。これは既存レコードの場合だけであり、新しいレコードは作成されると、そのレコードが属するテーブルの状態に関係なく保存することができます。

注：トランザクション中は、UNLOAD RECORD コマンドはカレントプロセスに対してのみレコードをアンロードします。他のプロセスから見た場合は、常にロックされた状態が維持されます。この状態はトランザクションが終了する（VALIDATE TRANSACTION、CANCEL TRANSACTION を実行する）まで続きます。

LOCKED ATTRIBUTES コマンドを使用すると、レコードをロックしているユーザやプロセスを知ることができます。

UNLOAD RECORD

UNLOAD RECORD {(テーブル)}

引数	タイプ	説明
テーブル	テーブル	→ アンロードするレコードの属するテーブル または省略した場合、デフォルトテーブル

説明

UNLOAD RECORD コマンドは、<テーブル>のカレントレコードをアンロードします。

UNLOAD RECORD コマンドは、レコードがローカルユーザに対してアンロックされている場合（他のユーザに対してロックされている場合）は、他のユーザに対してレコードをアンロックします。

UNLOAD RECORD コマンドは、メモリからレコードを解放（アンロード）しますが、そのレコードはカレントレコードのままになります。他のレコードがカレントレコードになると、前のカレントレコードは自動的にアンロードされ、他のユーザに対してアンロックされます。レコードの修正が終わり、そのレコードを自分自身のカレントレコードとしたままで、他のユーザから使えるようにしたい場合は常にこのコマンドを実行します。

レコードに大きなデータやピクチャフィールドまたは4D Draw等の外部ドキュメントが含まれている場合に、修正の必要がなくなるまで、そのカレントレコードをメモリ内に保持したくないかもしれません。こういう場合、**UNLOAD RECORD** コマンドを使用して、メモリ内にレコードを持たずにカレントレコードを保持できるようにします。また、そのフィールドの値にアクセスすることはできませんが、そのレコードによって占有されていたメモリを解放することができます。後に、レコードの値へのアクセスが必要となった場合、**LOAD RECORD** コマンドを使用します。

注：トランザクション中は、**UNLOAD RECORD** コマンドはカレントプロセスに対してのみレコードをアンロードします。他のプロセスから見た場合は、常にロックされた状態が維持されます。この状態はトランザクションが終了する（**VALIDATE TRANSACTION**、**CANCEL TRANSACTION** を実行する）まで続きます。

参照

LOAD RECORD

DISPLAY SELECTION

DISPLAY SELECTION ({テーブル} {;} {;} {;} {;})

引数	タイプ	説明
テーブル	テーブル	→ 表示するテーブル 省略した場合、デフォルトテーブル
*	*	→ レコードが1件の場合にも出力フォームを使用し、入力フォームのスクロールバーを表示しない
*	*	→ 入力フォームのスクロールバーを表示する (最初のオプション<*>を無効にする)

説明

DISPLAY SELECTION コマンドは、出力フォームを使用して、<テーブル>のカレントセレクションを表示します。レコードは、「ユーザ」モードの出力に類似のスクロール可能なリストとして表示されます。ユーザがレコードをダブルクリックすると、そのレコードは入力フォーム上に表示されます。リストは最前面にあるウインドウに表示されます。

セレクションを表示し、かつレコードをダブルクリックして修正を実行するには（「ユーザ」モードのウインドウでの作業と同じように）、**DISPLAY SELECTION** コマンドではなく **MODIFY SELECTION** コマンドを使用します。

次に説明する情報は、レコードの修正に関する情報を除き、すべて **DISPLAY SELECTION** コマンドと **MODIFY SELECTION** コマンドの両方に適用されます。

以下の図に、**DISPLAY SELECTION** コマンドで表示された出力フォームの例を示します。

名字	名前	入社年月日	給与	役職名
柴田	豊	78/04/01	¥6,680,000	次長
刀根	洋二	77/04/01	¥6,235,000	次長
岩島	信次	78/04/01	¥7,652,000	常務
大石	幸治	78/04/01	¥8,965,100	常務
今村	直重	80/09/01	¥6,978,000	
福川	敏一	80/04/01	¥6,240,000	次長代理
渡部	俊夫	80/04/02	¥6,581,400	課長代理
魚住	晃	81/04/05	¥6,532,000	
田中	敬助	81/09/01	¥6,983,000	常務代理
内海	邦章	81/04/01	¥6,238,000	
岸元	卓治	81/04/01	¥6,052,000	
武蔵	剛史	82/04/01	¥6,259,800	係長
白山	敬八朗	82/09/01	¥6,159,000	係長代理
小島	忠男	83/04/01	¥6,352,100	
浦島	能拓	83/09/01	¥6,587,400	
吉野	秀太郎	83/09/01	¥6,852,000	課長代理
藤上	団吉	84/04/01	¥5,872,400	
西山	芳彦	84/06/01	¥5,670,000	
千代田	春重	84/02/15	¥5,897,600	係長
岩澤	五朗	85/06/08	¥5,235,400	
今田	喜美斗	85/04/02	¥5,365,200	係長

DISPLAY SELECTION コマンドを実行した直後は、カレントレコードが存在しない場合があります。レコードを1件選択するために、**FIRST RECORD** コマンドまたは **LAST RECORD** コマンドを使用してください。

MODIFY SELECTION コマンドは入力フォーム中でレコードを修正することができますが、**DISPLAY SELECTION** コマンドは入力フォーム中でレコードを修正することができません。

オプションの<*>引数に関する規則を次に説明します。

- ・ セレクションにレコードが1件しか存在しないときに、1番目のオプションのアスタリスク (*) を指定していなければ、そのレコードは出力フォームではなく入力フォーム上に表示されます。
- ・ 1番目のオプションのアスタリスク (*) を指定した場合は、出力フォームを使用して1レコードのセレクションが表示されます。
- ・ 1番目のオプションのアスタリスク (*) を指定し、ユーザがレコードをダブルクリックしてそれを入力フォームに表示した場合には、入力フォームのスクロールバーは表示されません。これを無効にするには、2番目のオプションのアスタリスクを指定します。

「終了」ボタンが、自動的にリストの最後に組み込まれます。フォーム上に変数、またはアクティブオブジェクトを追加すると「終了」ボタンが消えます。このボタンをクリックするとコマンドを終了します。カスタムボタンを代わりに使用することもできます。出力フォームのフッタエリアにこのボタンを作成ことができます。また、レコード表示を終了させるために「登録」ボタンや「キャンセル」ボタンを使用、あるいは **ACCEPT** や **CANCEL** コマンドを呼び出すオブジェクトメソッドを利用することもできます。

ユーザは、カレントセレクションをスクロールし、該当レコードをクリックして選択することができます。ユーザが別のレコードをクリックした場合は、最初のレコードの選択が解除され、そのレコードが選択されます。また、ユーザは連続するレコードをグループとして選択することもできます。この場合は、選択する先頭のレコードをクリックし、最後のレコードを“shift”キーを押しながらマウスをクリック (shift+クリック) します。不連続のレコードを選択するには、選択したいレコードごとに、“Ctrl (Macintosh版では、command)”キーを押しながらマウスをクリックします。

DISPLAY SELECTION コマンドの実行後、ユーザが選択したレコードは“UserSet”という名前のセットに格納されます。UserSetは、ボタンがクリックされるか、メニュー項目が選択された際に、オブジェクトメソッドに対して表示されたセレクション内で使用できます。また、コマンド終了後に **DISPLAY SELECTION** コマンドを呼び出すプロジェクトメソッドからも利用できます。

例題

- 以下の例は、最初に、[従業員]テーブルのすべてのレコードを選択します。次に、**DISPLAY SELECTION** コマンドを使用してレコードを表示し、ユーザがプリントするレコードを選択します。最後に、**USE SET** コマンドでレコードを選択し、**PRINT SELECTION** コマンドでそのレコードを印刷します。

```

ALL RECORDS ([従業員]) `すべてのレコードの選択
DISPLAY SELECTION ([従業員];*) `レコードの表示
USE SET ("UserSet") `ユーザが選択したレコードだけを使用
PRINT SELECTION ([従業員]) `ユーザが選択したレコードの印刷

```

- Form event** 関数の6番目の例題を参照してください。この例題では、**DISPLAY SELECTION** コマンドの実行中に発生するイベントをすべて監視するためのあらゆるチェックが示されています。
- 「カスタム」モードで **DISPLAY SELECTION** コマンドや **MODIFY SELECTION** コマンドを使用する際に、例えば、「ユーザ」モードの「クエリ」メニューで提供される機能を再現するには、以下の手順にしたがってください。
 - 「デザイン」モードで必要なメニューを備えたメニューバーを作成します。例えば、“すべて表示”、“クエリ”、“並び替え”等のメニューです。
 - このメニューバーに（フォームプロパティダイアログのポップアップメニュー“連結するメニューバー”を使用）、**DISPLAY SELECTION** コマンドや **MODIFY SELECTION** コマンドを使用した出力フォームを関連付けます。
 - 以下のプロジェクトメソッドをメニューに関連付けます。

```

` M_SHOW_ALL ( 「すべて表示」メニュー項目に付属)
$vpCurTable:=Current form table
ALL RECORDS($vpCurTable->)

```

```

` M_QUERY ( 「クエリ」メニュー項目に付属)
$vpCurTable:=Current form table
QUERY($vpCurTable->)

```

```

` M_ORDDER_BY ( 「並び替え」メニュー項目に付属)
$vpCurTable:=Current form table
ORDER BY($vpCurTable->)

```

「カスタム」モードでセレクションの表示や修正を実行するたび、“標準”メニューオプションをすべて提供するために、**PRINT SELECTION**、**REPORT**等、他のコマンドも使用することができます。**Current form table** コマンドを使用すれば、これらのメソッドは汎用コードとなり、このメニューバーをあらゆるテーブルのあらゆる出力フォームに関連付けることができます。

参照

Form event、MODIFY SELECTION、セット

GET DOCUMENT PROPERTIES

GET DOCUMENT PROPERTIES (ドキュメント；ロック；非表示；作成日；作成時間；更新日；更新時間)

引数	タイプ	説明
ドキュメント	文字列	→ ドキュメントの名前、またはドキュメントの完全なパス名
ロック 合はFalse	ブール	← ロックの場合はTrue、アンロックの場合はFalse
非表示 False	ブール	← 非表示の場合はTrue、表示の場合はFalse
作成日	日付	← 作成日
作成時間	時間	← 作成時間
更新日	日付	← 更新日
更新時間	時間	← 更新時間

説明

GET DOCUMENT PROPERTIES コマンドは、引数<ドキュメント>に渡した名前またはパス名を持つドキュメントファイルに関する情報を返します。

呼び出し後、以下の情報が返されます。

- 引数<ロック>には、ドキュメントがロックされていればTrueが返されます。ロックされているドキュメントの内容を書き換えることはできません。
- 引数<非表示>には、ドキュメントが隠されていればTrueが返されます。
- 引数<作成日>および<作成時間>には、ドキュメントが作成された日付と時間が返されます。
- 引数<更新日>および<更新時間>には、ドキュメントが更新された日付と時間が返されます。

SET DOCUMENT PROPERTIES

SET DOCUMENT PROPERTIES (ドキュメント；ロック；非表示；作成日；作成時間；更新日；更新時間)

引数	タイプ	説明
ドキュメント	文字列	→ ドキュメントの名前、またはドキュメントの完全なパス名
ロック	ブール	→ ロックの場合はTrue、アンロックの場合はFalse
非表示	ブール	→ 非表示の場合はTrue、表示の場合はFalse
作成日	日付	→ 作成日
作成時間	時間	→ 作成時間
更新日	日付	→ 更新日
更新時間	時間	→ 更新時間

説明

SET DOCUMENT PROPERTIES コマンドは、引数<ドキュメント>に渡した名前またはパス名を持つドキュメントファイルについての情報を変更します。

呼び出しの前に以下の情報を渡します。

- ドキュメントをロックするには、引数<ロック>にTrueを渡します。ロックされたドキュメントの内容を書き換えることはできません。ドキュメントのロックを解除するには<ロック>にFalseを渡します。
- ドキュメントを隠すには、引数<非表示>にTrueを渡します。デスクトップウィンドウでドキュメントが表示されるようにするには、<非表示>にFalseを渡します。
- 引数<作成日>および<作成時間>に、ドキュメントの作成日および作成時間を渡します。
- 引数<更新日>および<更新時間>に、最新のドキュメント更新日および更新時間を渡します。

作成および最新の更新の日付および時間は、ドキュメントを作成、またはこれにアクセスするたびに、システムのファイルマネージャによって管理されます。このコマンドを使用すると、特別な用途のためにこれらのプロパティを変更することができます。**GET DOCUMENT PROPERTIES** コマンドの例を参照してください。

参照

GET DOCUMENT PROPERTIES、SET DOCUMENT CREATOR、SET DOCUMENT TYPE

Open form window

Open form window ({テーブル;} フォーム名 {; タイプ {; 水平位置 {; 垂直位置 {; *}}}) → ウィンドウ参照番号

引数	タイプ		説明
テーブル	テーブル	→	フォームのテーブル 省略されている場合デフォルトテーブル
フォーム名	文字列	→	フォームの名前
タイプ	倍長整数	→	ウィンドウのタイプ
水平位置	倍長整数	→	ウィンドウの水平位置
垂直位置	倍長整数	→	ウィンドウの垂直位置
*	*	→	ウィンドウの現在位置とサイズをセーブ
ウィンドウ参照番号	倍長整数	←	ウィンドウの参照番号

説明

Open form window コマンドは、フォーム名引数で指定したフォームのサイズおよびサイズ変更プロパティを使用して新しいウィンドウを開きます。

フォームの内容はウィンドウに表示されないことに注意してください。フォームを表示したい場合には、フォームをロードするコマンドを呼び出さなければなりません (例えば、**ADD RECORD**)。

デフォルトでは (タイプ引数が渡されていない場合) クローズボックス付きの標準ウィンドウが開かれます。**Open window** コマンドとは異なり、ウィンドウのクローズボックスには何のメソッドも定義されません。このクローズボックスをクリックすると、**On Close Box** フォームイベントがフォーム用に起動されている場合を除き、ウィンドウをキャンセルして閉じます。この場合、**On Close Box** イベントに定義されたコードが実行されます。

フォームのサイズ変更が可能であれば、開かれたウィンドウはズームボックスならびにグローボックスを持ちます。

注：フォームの主なプロパティを知るには、**GET FORM PROPERTIES** コマンドを使用します。

オプションのタイプ引数はウィンドウのタイプが指定でき、下記の“Open window” テーマ内にある定数の内から1つを渡します。

定数	タイプ	値
Plain window	倍長整数	8
Modal dialog box	倍長整数	1
Movable dialog box	倍長整数	5
Palette window	倍長整数	1984

注：生成されるウィンドウの属性（grow box、close boxなど）は、選択したウィンドウタイプにおけるOS上でのインターフェイス仕様に依存します。したがって、これらの属性の反映状況は実行環境OSによって変わってきます。

オプションの引数水平位置は、ウィンドウの水平位置を定義します。ポイント単位で位置を指定します（**Open window** コマンドを参照してください）。または、下記の“Open form window” テーマ内にある定数の内から1つを渡します。

定数	タイプ	値
Horizontally Centered	倍長整数	65536
On the Left	倍長整数	131072
On the Right	倍長整数	196608

オプションの引数垂直位置は、ウィンドウの垂直位置を定義します。ポイント単位で位置を指定します（**Open window** コマンドを参照してください）。または、下記の“Open form window” テーマ内にある定数の内から1つを渡します。

定数	タイプ	値
Vertically Centered	倍長整数	262144
At the Top	倍長整数	327680
At the Bottom	倍長整数	393216

これらの引数は、ツールバーおよびメニューバーの存在ならびにアプリケーションウィンドウの現在のサイズ（Windowsの場合）を考慮に入れます。

オプションの引数*を渡すと、ウィンドウをクローズした時の位置およびサイズが記憶されます。ウィンドウが再度開かれる時に、その前の位置とサイズが優先されます。この場合、垂直位置と水平位置の引数は最初にウィンドウが開かれる時のみに使用されます。

- ▼下記のステートメントは標準のウィンドウをクローズボックス付きで開き、自動的にそれを「入力」フォームと同じサイズになるように調整します。フォームは、サイズ変更可能なものとして定義されているので、ウィンドウもグローボックスおよびズームボックスを持ちます。

\$winRef := Open form window ([[Table1];"Enter")

- ▼下記のステートメントは、画面の左上部にあるフローティングパレットを開くものです。このパレットは、開かれるたびに前回ユーザが閉じた時の位置に表示されます。

\$winRef := Open form window ([[Table1]; "Tools"; Palette window; On the Left;
At the Top;*)

参照

Open window、GET FORMS PROPERTIES

