

4D ChartTM

ランゲージリファレンス
MacTM OS and Windows[®] 版



4th Dimension
by
Laurent Ribardière
Adapted by Bernard Gallet

4D Chart ランゲージリファレンス

Copyright© 1994- 1997 ACI SA/ACI US, Inc.

All rights reserved

注意

このソフトウェアの使用に際し、本製品に同梱のLicense Agreement（使用許諾契約書）に同意する必要があります。ソフトウェアを使用する前に、License Agreementを注意深くお読みください。

このマニュアルに記載されている事項は、将来予告なしに変更されることがあり、いかなる変更に関してもACI SAおよびACI USは一切の責任を負いかねます。このマニュアルで説明されるソフトウェアは、本製品に同梱のLicense Agreement（使用許諾契約書）のもとでのみ使用することができます。

ソフトウェアおよびマニュアルの一部または全部を、ライセンス保持者がこの契約条件を許諾した上での個人使用目的以外に、いかなる目的であれ、電子的、機械的、またどのような形であっても、無断で複製、配布することはできません。

ACI®、4D®、4D First™、4th Dimension®、4D Runtime®、4D Server™、4D Calc®、4D Insider™、4Dロゴ、4th Dimensionロゴは、ACI SAの登録商標または商標です。

Microsoft®とWindows®はMicrosoft Corporation社の登録商標です。

Apple®、Macintosh®、Power Macintosh™、LaserWriter®、Image Writer®、QuickTime®はApple Computer Inc.の登録商標または商標です。

Mac2Win Software Copyright © 1990 - 1996 はAltura Software社の製品です。

XTND Copyright 1992 - 1996 © ACI. All rights reserved.

XTND Technology Copyright 1989 - 1996 © Claris Corporation. All rights reserved.

ACROBAT © Copyright 1987 - 1996 ©, Secret Commercial Adobe Systems Inc. All rights reserved.
ACROBATはAdobe Systems社の登録商標です。

その他、記載されている会社名、製品名は、各社の登録商標または商標です。

本書は『4D Chart ランゲージリファレンス』マニュアルです。

4D Chartプラグインは、4th Dimensionバージョン6に統合されるようになりました。このプラグインが4D自体の一部になったので、『4D Chart ユーザリファレンス』マニュアルは『4th Dimension / 4D First ユーザリファレンス』マニュアルに統合されました。

ACIのソフトウェアは主要な進化を実現しているので、このマニュアルに統合できなかった変更点もあります。以下のコマンドの名称が変更されていることに注意してください。

旧コマンド名	4D Chart V6でのコマンド名
CT_SET PREFERENCES	CT SET PROPERTIES
CT_GET PREFERENCES	CT GET PROPERTIES

これらの変更はユーザが作成するコードの実行には何ら影響を与えません。ユーザが作成したコードは、4Dバージョン6でオープンすると自動的に変換されます。

注：旧バージョンの4D Chartは、日本では発売されていません。

4D Chartは、4th Dimensionプログラム言語に100を超えるコマンドを追加します。4D Chartコマンドを使用すると、通常は手動で行っている作業を制御することができます。

例えば、文書のオープンや保存、グラフの作成や変更、4D Chartメニューコマンドの実行、または他の4th Dimensionモジュールとのホットリンクによる連動などができます。

4D Chartコマンドには、すべて先頭に「CT」が付くので、標準の4th Dimensionコマンドや他のモジュールから追加されたコマンドと区別できます。

このマニュアルでは、4D Chartコマンドは **CT OPEN DOCUMENT** のように特殊フォントの大文字で表記します。4D Chart関数は **CT Get object type** のように最初の文字だけを大文字にして表記します。

このマニュアルの例の中には、スペースの関係でコードの行が2行目や3行目に継続するものもあります。ただし、これらの例を入力するときには、コードの行は1行で入力し、フローを分断するReturnキーは押さないでください。

「メソッド」エディタにおけるコマンド

4D Chartコマンドは、「メソッド」エディタではテーマごとにグループ化され、「ルーチン」リストの最後に表示されます。



4D Chartテーマ

複数のモジュールをインストールしている場合には、リストに表示されるテーマの順序はモジュールがインストールされた順序と同じになります。

4D Chartコマンドは、4th Dimensionのコマンドがアルファベット順でも、テーマごとにグループ分けされていても、テーマごとのグループ分けされて表示されます。

4D Chartコマンドのメソッドへの配置は、4th Dimensionコマンドに対して行うのと同様、メソッドの中に入力するか、コマンドを「ルーチン」リスト内のポップアップメニューから選択して行います。



ポップアップメニューから
任意のコマンドを選択する

4D Chartコマンドは、プロジェクト、トリガ、フォーム、オブジェクトメソッドといった任意の種類のメソッドで使用できます。コマンドは、4D Chartエリアと同じフォーム上にあるオブジェクトから起動されたオブジェクトメソッドで使用すると特に便利です。

文書への参照

4D Chart文書を制御するためにコマンドを使用するときには、文書のエリアID番号で文書を特定する必要があります。エリアID番号は4D Chart内部のものであり、通常は変数内に格納されています。

4D Chart文書は、フォーム上のエリア、外部ウィンドウ、オフスクリーンエリアの3つの場所のいずれかに存在できます。文書の場所が確定すると、4D Chartは文書を配置して、処理を行うために文書のエリアID番号が必要になります。

エリアID番号とエリア変数

4D Chartは、4D Chartエリア、外部ウィンドウ、オフスクリーンエリアの場所を格納するために変数を使用します。コマンドや関数の引数（パラメータ）としてエリアのID番号を含む変数を渡すことで、処理を実行するエリアを参照してください。

この概説の後に記載するコマンド説明では、文書エリアを特定する変数を<エリア>引数といいます。

<エリア>変数には、2つの種類があります：

外部エリアオブジェクト名

外部ウィンドウまたはオフスクリーンエリア用にユーザが作成した変数

これらの2種類の変数については、これ以降の節で説明します。

外部オブジェクト名

フォーム内で4D Chartエリアを作成し、名前を付けると、4th Dimensionが自動的に4D Chartエリアの名前をそのエリアを参照する変数として認識します。例えば、引数<エリア>に「利益」を指定することによって、「利益」エリアを参照できます。

外部ウィンドウとオフスクリーンエリア

External window関数または**CT New offscreen area**関数を使用して外部ウィンドウやオフスクリーンエリアを作成すると、関数から返るエリアID番号が変数に格納されます。その後、この変数を他のコマンドや関数で使用すると、外部ウィンドウやオフスクリーンエリアを参照できます。変数に値を格納するには、変数名と代入演算子(=)をコード行で関数の右側に指定します。

次の例は、4D Chart外部ウィンドウを作成し、エリアID番号を「MyArea」変数に格納します。

```
MyArea:=External window (30 ; 30 ; 450 ; 8 ; "利益" ; "_4D Chart")
```

4D Chartの配置場所

4D Chartは、次の場所で使用できます：

- フォーム上の4D Chartエリア
- 4D Chart外部ウィンドウ
- 4D Chartオフスクリーンエリア

ここでは、これらの場所をデータベースで作成する方法について説明します。

フォーム上の4D Chartエリア

4D Chartは、任意のフォームに配置できます。通常、4D Chartエリアは文書进行操作できるように入力フォーム内に配置しますが、情報を表示、印刷するために出力フォームに4D Chartを配置することもできます。

4D Chartはフォーム全体を使用できます。あるいは、フィールドや他のフォーム要素と一緒にスペースを共有することもできます。

4D Chart用には「外部ルーチンエリア」タイプのアクティブオブジェクトエリアを使用します。外部ルーチンエリアは、ボタン、入力可能エリア、スクロール可能エリアなどの4th Dimensionアクティブオブジェクトタイプの1つです。外部ルーチンエリアに関する詳細は『4th Dimension / 4D First デザインリファレンス』と『4th Dimension ランゲージリファレンス』を参照してください。

フォーム上の文書を参照する必要があるときには、4D Chart外部オブジェクトを作成したときに使用したオブジェクト名を使用してください。

フォームへの4D Chartエリアの配置については、『4th Dimension / 4D First ユーザリファレンス』を参照してください。

4D Chart外部ウィンドウ

4th Dimensionの**External window**関数を使用すると、外部ウィンドウがオープンされ、その中に空の4D Chart文書が表示されます。以下は、**External window**関数の説明です。

External window

External window (左 ; 上 ; 右 ; 下 ; タイプ ; タイトル ; メソッド) 倍長整数

引数	タイプ	説明
左	数値	ウィンドウ左端の画面上の位置(ピクセル)
上	数値	ウィンドウ上端の画面上の位置(ピクセル)
右	数値	ウィンドウ右端の画面上の位置(ピクセル)
下	数値	ウィンドウ下端の画面上の位置(ピクセル)
タイプ	数値	ウィンドウのタイプ
タイトル	文字列	ウィンドウのタイトル
メソッド	文字列	外部ルーチン名

External window関数は、新しいウィンドウを開きます。外部ルーチンエリア<メソッド>を表示し、その外部ルーチンエリアにアクセスするための番号を倍長整数で返します。最初の6個の引数は**Open window**関数と同じです。『4th Dimension ランゲージリファレンス』の**Open window**関数の説明を参照してください。

External window関数は、<エリア>で指定した外部ルーチンエリアを参照するための番号を返します。例えば、4th Dimension外部パッケージ中の複数のコマンドを1つの外部ルーチンエリアに作用させた場合でも、**External window**関数の返した番号を使用して対応するウィンドウの外部ルーチンエリアに対してコマンドを適用することができます。<エリア>は、外部ルーチンエリアに付けられた名前であればなりません。

4D Chartの<メソッド>は、次のように記述します：

_4D Chart

先頭のアングスコア (_) および「4D」と「Chart」の間の空白文字は省略しないでください。これらの両方の表記は必須シンタックス要素です。

External window関数は、モードレスウィンドウを作成します。このコマンドはユーザの入力を待ちません。従って、複数のアクティブウィンドウを一度に開くことができます。各ウィンドウは、クリックすることによって最前面に移動して操作することができます。タイトルバーにクローズボックスのあるウィンドウタイプの場合は、ユーザの操作によって閉じることができます。プロシージャでウィンドウを閉じる場合には、**CLOSE WINDOW**コマンドを使用します。この場合に、**External window**関数の返した番号を**CLOSE WINDOW**コマンドに引数として渡します。

次の例は、外部ウインドウを開き、空の4D Chart文書を表示します：

```
vChart:=External window (50 ; 50 ; 350 ; 450 ; 8 ; "利益グラフ" ; "_4D Chart")
```

これに引き続いて、この文書に対するエリアを指定する必要があるときには、いつでもvChartを使用できます。以下の例を参照してください：

```
CT GET DEPTH (vChart ; vObject ; v水平 ; v垂直)
```

4D Chartオフスクリーンエリア

オフスクリーンエリアはメモリ上に格納され、プログラマやユーザからは見えません。オフスクリーンエリアを使用すると、必要があればユーザが文書を見る前に変更を行ったり、オリジナルに戻せるように文書を保存したりできます。

画面が再表示される必要がないので、4D Chartの操作は、オフスクリーンエリアの方がより迅速に実行できます。

CT New offscreen area関数を使用するとオフスクリーンエリアを作成できます。**CT PICTURE TO AREA**コマンドを使用すると、(オフスクリーンエリアにできる)4D Chartエリアに(4D Chartエリアを含むことができる)4Dのピクチャフィールドを配置できます。これらのコマンドに関する詳細は、第3章の「エリアコマンド」を参照してください。

オフスクリーンエリアの使用が終わったときには、忘れずにオフスクリーンエリアを削除して、使用しているメモリを解放してください。オフスクリーンエリアをすべてクリアしていないと、データベースをクローズするときに4th Dimensionからエラーメッセージが表示されます。

次の例のコードをプロジェクトメソッドに配置すると、4D Chart文書の保存用オフスクリーンエリアを作成できます。フォーム上のボタンを使用すると、保存したオリジナルの文書に戻すこともできます。

```
エリア:= CT New offscreen area  
QUERY([売上];[売上]顧客ID=vCustID)  
If (Records in selection([売上]=1)  
    CT PICTURE TO AREA (エリア;[売上]利益)  
    `オフスクリーンエリアにグラフを格納する  
    MODIFY RECORD([売上])  
    `売上レコードを変更する  
    CT DELETE OFFSCREEN AREA (エリア)  
    `オフスクリーンエリアで使用されたメモリを解放する  
End if
```

入力フォーム上にボタンを作成し、それを以下のコードに割り当てます：

```
利益:= CT Area to picture (エリア ; -2)
```

```
`オリジナル文書を含むオフスクリーンエリアを利益フォーム内にある外部ルーチンエリアに配置する
```

デフォルトエリアを使用する

デフォルトエリアは、すべての新規4D Chartと外部ウィンドウのデフォルト属性を設定するために使用できるRAM内のテンプレートです。4D Chartエリアに対して実行できるコマンドは、その<エリア>引数を-1に設定すると、デフォルトエリアで実行できます。メソッドを使用すると、他の任意のエリアで行う処理をデフォルトエリアで実行できます。

デフォルトエリアを使用すると4D Chartエリアに対する不要なコードの実行を軽減することができます。例えば、新規の4D Chartエリアと外部ウィンドウすべてをスクロールバーなしで表示させる場合は、エリアや外部ウィンドウで個別にスクロールバーをオフにする必要がありません。

ユーザは4D Chartエリアと外部ウィンドウの両方の属性を設定できます。フォーム上の新しい4D Chartエリアや新しい外部ウィンドウがオープンされたときには、デフォルトエリアはいつでもテンプレートとして自動的に使用されます。コードを実行する必要がまったくないので、デフォルトエリアはチャートエリアを高速にカスタマイズする方法を提供します。

デフォルトエリアをすべての新しい4D Chartエリアに適用したくない場合には、4D Chartエリア用のディスク上にテンプレートを作成するか、フォームのBeforeフェーズに適切なコードを配置することによってデフォルトエリアを上書きできます。ディスク上のテンプレートまたはフォームのBeforeフェーズのコードは、デフォルトエリアよりも優先されます。

オブジェクトへの参照

4D Chart文書のすべてのオブジェクトには、一意な番号が与えられています。この番号は、オブジェクトのIDであり、オブジェクトが作成されるときに割り当てられます。

これは、グラフの作成、「ツール」パレットでのオブジェクトの描画、クリップボードからのオブジェクトの貼り付け、複数のオブジェクトのグループ化、既存のオブジェクトの複製、ホットリンクの引用、フィールド参照の貼り付けなどをするたびに、新しいIDが割り当てられることを意味します。オブジェクトIDは一意なので、IDはオブジェクトを参照する便利な方法です。オブジェクトIDが文書内部で再使用されることはありません。たとえオブジェクトが削除されても、その番号は文書がある間は「廃棄」されています。

オブジェクトIDは転送可能ではありません。ある4D Chart文書のIDが5であるオブジェクトは、別の文書に貼り付けられると必ずしも同じIDにはなりません。

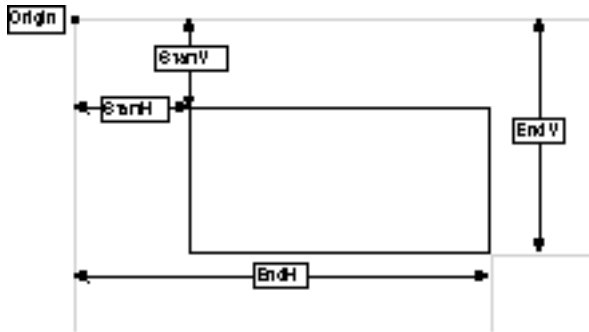
標準オブジェクト作成コマンドはすべて関数であり、結果のオブジェクトのIDを戻しません。パレットでユーザが作成したオブジェクトとは違い、コマンドで作成されたオブジェクトは自動的に選択されません。

CT Get ID関数を使用するとオブジェクトのIDを取得できます。この関数に関する詳細は、第7章の「オブジェクトコマンド」を参照してください。

オブジェクトの配置場所（座標）を指定する

オブジェクトの位置とサイズをそのオブジェクトの「座標」といいます。座標の説明または指定を行うすべてのコマンドは、ポイント単位で行います。

位置の説明や指定を行うコマンドは、原点に対して行います。原点は横のルーラと縦のルーラのゼロ点の交差です。次の図は、座標システムを表しています。



スコープへのコマンド指定

多くの4D Chartコマンドには<スコープ>という引数があります。<スコープ>は4D Chart文書のどのオブジェクトまたはテキスト文字がコマンドから影響を受けるのかを指定するものです。

次の表は、<スコープ>の一般的な規則を説明しています。<スコープ>が与えられたコマンドにどのように影響するかは、第3章から第9章までのコマンドごとの説明を参照してください。

スコープ	影響を受けるテキストまたはオブジェクト
>0	オブジェクトID
0	選択されたオブジェクト
-1	文書のすべてのオブジェクト
-2	デフォルト値
-3	テキストオブジェクトで選択された文字

4D Chartエラーの取り扱い

4D Chartは、コードの実行中に発生するエラーを管理する複数の方法を提供します。次に示す方法を任意に組み合わせて使用してください。

CT Error関数を使用すると、処理を実行した後にエラーをチェックできます。**CT Error**関数は4D Chartが実行した最後の処理の状態を表すエラーコードを返します。**CT Error**関数に関する詳細は、第14章の「エリア制御コマンド」を参照してください。

CT ON ERRORコマンドを使用すると、4D Chartエラーを管理するプロシージャをインストールできます。プロシージャを**CT ON ERROR**コマンドでインストールした後、4D Chartは4D Chartのエラーが発生したときにそのプロシージャをコールします。**CT ON ERROR**コマンドに関する詳細は、第4章の「エリア制御コマンド」を参照してください。

引数に値を返す関数やコマンドを実行した後は、エラーのチェックができます。関数の実行中にエラーが発生した場合、関数は-32000を返します。引数に値を返すコマンドが特定の引数に対する値を取り出しているときにエラーを検出した場合は、その引数に-32000を返します。

この章は、4D Chartのコマンドと関数を使ったプログラムによるグラフの作成について、6つの実例を紹介します。

各セクションでは、以下を掲載します。

例に使用される状況の説明

サンプルデータベースの構造

サンプルデータを使用して、すべてメソッドで作成されたグラフ

サンプルグラフの作成に使用されたコード

4D Chartコマンドの完全な説明については、このマニュアルの第3章から第9章までを参照してください。

データベース内のレコードを使ってグラフを作成する

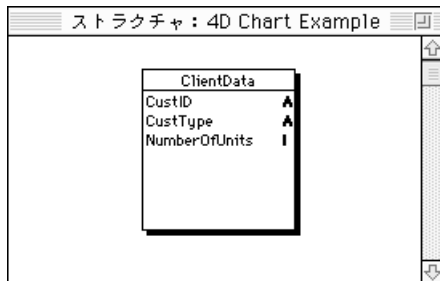
次に示す4つの節は、**CT Chart selection**関数と**CT Chart data**関数を使用した2次元グラフと3次元グラフの作成例です。

フィールドで定義された系列を使って2次元グラフを作成する

あなたがソフトウェア会社を営し、政府、教育機関、小売りなど、多岐に渡る分野の顧客を抱えているとします。あなたのデータベースは顧客と営業の履歴を取っています。

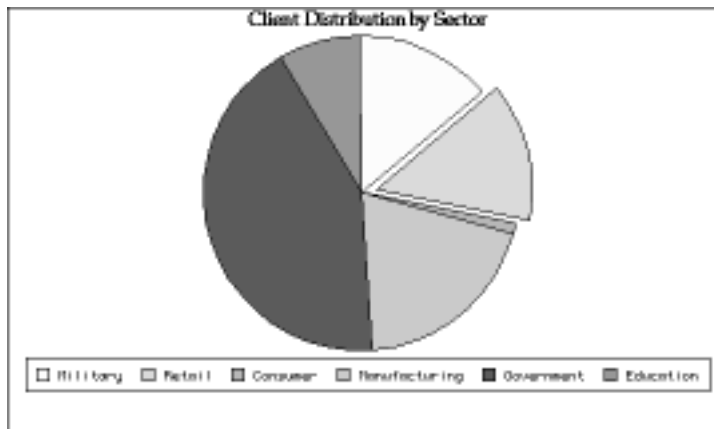
将来の製品開発とマーケティング用に顧客全体の履歴記録を続行するために、それぞれの分野でソフトウェアが何単位売れたのかを知りたいと思っています。

次のストラクチャファイルは、サンプル用に簡略化したものです。



ClientData	
CustID	A
CustType	A
NumberOfUnits	I

4D Chartを使用して、次のグラフを生成します。これは、各分野での全営業成績の割合を示したものです。



次はサンプルグラフを作成するために使用された「顧客別利益」メソッドのコードです。

```

`メソッド：顧客別利益
`項目：顧客タイプ(グループ化)
`数値：購入された単位の数
C_LONGINT ($Left ; $Top ; $Right ; $Bottom)
C_LONGINT ($Left2 ; $Top2 ; $Right2 ; $Bottom2)
C_LONGINT ($Area ; $Chart ; $Title ; $Locate)

`チャート用のレコード選択を生成
ALL RECORDS([ClientData])
`数値データを含むフィールドの配列
`この配列の要素数は系列の数と同じ
ARRAY LONGINT($aFields ; 1)
$aFields{1}:=3 [ClientData]NumberOfUnitsフィールドの番号
`チャートを表示する外部ウィンドウをオープンする
$Area:=External window(2 ; 40 ; 550 ; 400 ; 8 ; "My Chart" ; "_4D Chart")
`2D 円グラフを作成する
$Chart:= CT Chart selection ($Area ; 6 ; 2 ; 1 ; 1 ; 2 ; $aFields)
`凡例を配置する場所を設定する (下, 水平方向)
CT SET LEGEND ATTRIBUTES ($Area ; $Chart ; 1 ; 0 ; 0 ; 0 ; 8 ; 0 ; 0)
`左上隅にチャートタイトルを追加する
$Title:= CT Draw text ($Area ; 1 ; 1 ; 210 ; 3 ; "Client Distribution by Sector")
`チャートタイトルをフォーマットする (Palatino, 14ポイント, 太字, 中央揃え, 黒)
CT SET TEXT ATTRIBUTES ($Area ; $Title ; CT Font number("Palatino") ; 14 ;
1 ; CT Index to color (16) ; 1)
`エリア内のオブジェクトを中央に揃えるためにエリア境界を取得する
CT GET AREA BOUNDARY ($Area ; 1 ; $Left ; $Top ; $Right ; $Bottom)
`チャートを中央揃えする
CT GET BOUNDARY ($Area ; $Chart ; $Left2 ; $Top2 ; $Right2 ; $Bottom2)
$Locate:=((($Right-$Left)-($Right2-$Left2))/2)
CT MOVE ($Area ; $Chart ; $Locate ; $Top2)
`タイトルを中央揃えする
CT GET BOUNDARY ($Area ; $Title ; $Left2 ; $Top2 ; $Right2 ; $Bottom2)
$Locate:=((($Right-$Left)-($Right2-$Left2))/2)
CT MOVE ($Area ; $Title ; $Locate ; $Top2)
`チャートを9ポイント分下げ、タイトルの下に表示されるようにする
CT GET BOUNDARY ($Area ; $Chart ; $Left ; $Top ; $Right ; $Bottom)
CT MOVE ($Area ; $Chart ; $Left ; $Top+9)
    
```

フィールドで定義された系列を使って3次元グラフを作成する

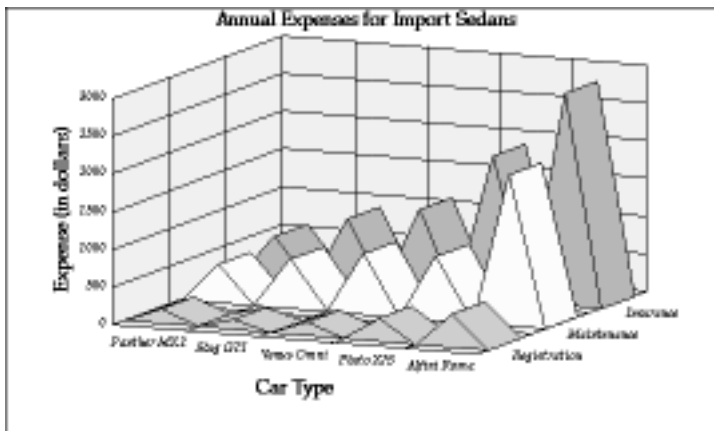
あなたが新車の市場におり、データベースを使用して手持ちの金で買える最高の車を決定しようとしています。5台の車に選択を絞り込み、それぞれの車で年間の登録、保険、メンテナンスの経費がいくらかかるかを比べたいと思っています。

データベースのストラクチャは、次のとおりです。

Expenses	
Car	A
Cost	R
Registration	R
Insurance	R
Maintenance	R

収集した情報を解析するために、3次元グラフで情報をグラフ化することにしました。

4D Chartを使用して、次のグラフを生成します。



「車種グラフ」プロジェクトメソッドは前述のグラフ作成に使われたものです。このメソッドは、**CT Chart selection**関数を使用します。**CT Chart selection**関数を使用するときには、この関数にフィールドの配列を渡します。フィールド名は系列名になり、フィールドの値が数値軸上でグラフ化されます。この例では、Registration(登録)、Insurance(保険)、Maintenance(メンテナンス)の各フィールドは、系列と数値のために使用されています。

次は、「車種グラフ」プロジェクトメソッドのコードです。

メソッド：車種グラフ

項目：車種

系列：経費カテゴリ

数値；ドル換算の経費

C_LONGINT(\$Left ; \$Top ; \$Right ; \$Bottom)

C_LONGINT(\$Area ; \$Chart ; \$Title ; \$Locate ; \$i)

C_LONGINT(\$Left2 ; \$Top2 ; \$Right2 ; \$Bottom2)

`チャート用のレコード選択を生成する

ALL RECORDS([Expenses])

ORDER BY([Expenses];[Expenses]Cost ; >)

`系列と数値のデータをフィールド配列に入れる

ARRAY LONGINT(\$aFields ; 3)

\$aFields{1}:=**Field**(>>[Expenses]Registration)

\$aFields{2}:=**Field**(>>[Expenses]Maintainance)

\$aFields{3}:=**Field**(>>[Expenses]Insurance)

`チャートを表示するために外部ウィンドウをオープンする

\$Area:=**External window**(2 ; 40 ; 550 ; 400 ; 8 ; "My Chart" ; "_4D Chart")

`インタフェース要素を隠す

CT SET DISPLAY (\$Area ; 1 ; 0) `メニューを隠す

CT SET DISPLAY (\$Area ; 2 ; 0) `チャートツールを隠す

CT SET DISPLAY (\$Area ; 3 ; 0) `オブジェクトツールを隠す

CT SET DISPLAY (\$Area ; 6 ; 0) `スクロールバーを隠す

CT SET DISPLAY (\$Area ; 9 ; 0) `ルーラを隠す

`2D 円グラフを作成する

\$Chart:= **CT Chart selection** (\$Area ; 100 ; 2 ; 1 ; **File**(>>[Expenses]) ; 1 ; \$aFields)

`スケールを設定する

CT SET REAL SCALE (\$Area ; \$Chart ; 0 ; 0 ; 0 ; 0 ; 3000 ; 500 ; 100)

`凡例は表示しない

CT SET LEGEND ATTRIBUTES (\$Area ; \$Chart ; 0 ; 0 ; -1 ; -1 ; 0 ; 0 ; 0)

`チャートタイトルを追加する

CT SET TITLE ATTRIBUTES (\$Area ; \$Chart ; 0 ; 3 ; 0 ; "車種")

CT SET TITLE ATTRIBUTES (\$Area ; \$Chart ; 1 ; 0 ; 0 ; "") `表示しない

CT SET TITLE ATTRIBUTES (\$Area ; \$Chart ; 2 ; 2 ; 3 ; "経費(ドル換算)")

```

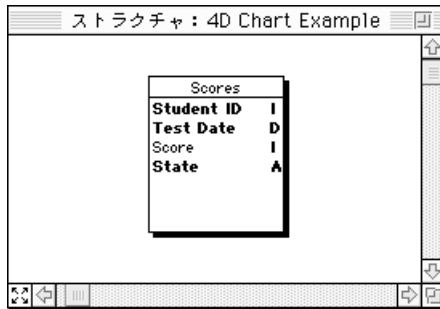
`左上隅にタイトルを追加する
$Title:= CT Draw text ($Area ; 1 ; 1 ; 300 ; 3 ; "輸入セダンの年間経費")
`タイトルをフォーマットする (Palatino , 14ポイント, 太字, 中央揃え, 黒)
$Color:= CT Index to color (16)
$Font:= CT Font number ("Palatino")
CT SET TEXT ATTRIBUTES ($Area ; $Title ; $Font ; 14 ; 1 ; $Color ; 1)
`1番目の系列の色を緑に設定する
$Color:= CT Index to color (10)
CT SET CHART FILL ATTRIBUTES ($Area ; $Chart ; 8 ; 100 ; 3 ; $Color)
`1番目の系列の色を黄に設定する
$Color:= CT Index to color (2)
CT SET CHART FILL ATTRIBUTES ($Area ; $Chart ; 8 ; 200 ; 3 ; $Color)
`1番目の系列の色を赤に設定する
$Color:= CT Index to color (4)
CT SET CHART FILL ATTRIBUTES ($Area ; $Chart ; 8 ; 300 ; 3 ; $Color)
`すべてのプロット矩形用に塗りつぶし属性を設定する
$Color:= CT Index to color (8)
For ($ i ; 0 ; 2)
    CT SET CHART FILL ATTRIBUTES ($Area ; $Chart ; 1 ; $i ; 5 ; $Color)
End for
`テキスト属性ラベルを設定する (Palatino, 9ポイント, 太字斜体)
$Font:= CT Font number ("Palatino")
For ($ i ; 0 ; 2)
    CT SET CHART FILL ATTRIBUTES ($Area ; $Chart ; 4 ; $i ; $Font ; 9 ; 3 ; -1)
End for
`タイトル用のテキスト属性を設定する (Palatino, 12ポイント, 太字)
$Font:= CT Font number ("Palatino")
For ($ i ; 0 ; 2)
    CT SET CHART FILL ATTRIBUTES ($Area ; $Chart ; 5 ; $i ; $Font ; 14 ; 1 ; -1)
End for
`中央揃え用のエリア寸法を取得する
CT GET AREA BOUNDARY ($Area ; 1 ; $Left ; $Top ; $Right ; $Bottom)
`チャートを中央揃えする
CT GET BOUNDARY ($Area ; $Chart ; $Left2 ; $Top2 ; $Right2 ; $Bottom2)
$Locate:=((($Right-$Left)-($Right2-$Left2))/2)
CT MOVE ($Area;$Chart;$Locate;$Top2)
`タイトルを中央揃えする
CT GET BOUNDARY ($Area ; $Title ; $Left2 ; $Top2 ; $Right2 ; $Bottom2)
$Locate:=((($Right-$Left)-($Right2-$Left2))/2)
CT MOVE ($Area ; $Title ; $Locate ; $Top2)
`チャートを9ポイント下げる
CT GET BOUNDARY ($Area ; $Chart ; $Left ; $Top ; $Right ; $Bottom)
CT MOVE ($Area ; $Chart ; $Left ; $Top+9)

```

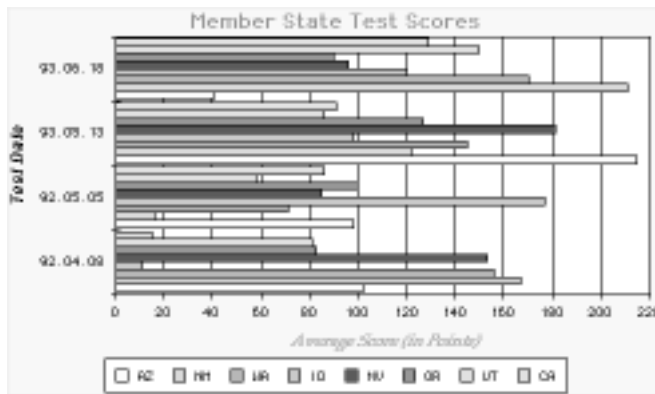
レコードのデータによって定義された系列を使って2次元グラフを作成する

あなたが教育プログラムの西部地区担当者であり、あなたの地区にいる各州の生徒達の標準テストスコアを別のテスト日付と比較したいと仮定します。

データベースには、テスト日付と出身地にタグが付いて、あなたの地区の生徒のスコアが保持されています。



4D Chartを使用して、次のグラフを作成します。



前述のグラフは、「スコアグラフ」プロジェクトメソッドによって作成されています。次は、「スコアグラフ」メソッドのコードです。

メソッド：スコアグラフ

項目：テスト日付

系列；メンバーの州

数値：テストスコア

C_LONGINT (\$Left ; \$Top ; \$Right ; \$Bottom)

C_LONGINT (\$Left2 ; \$Top2 ; \$Right2 ; \$Bottom2)

C_LONGINT (\$Area ; \$Chart ; \$Title ; \$Locate ; \$Score ; \$Color ; \$Font ; \$Rect)

ALL RECORDS([Scores])

ORDER BY([Scores];[Scores]Test Date ; >)

`チャートを表示するために外部ウィンドウをオープンする

\$Area:=**External window**(2 ; 40 ; 550 ; 400 ; 8 ; "My Chart" ; "_4D Chart")

`インタフェース要素を隠す

CT SET DISPLAY (\$Area ; 1 ; 0) `メニューを隠す

CT SET DISPLAY (\$Area ; 2 ; 0) `チャートツールを隠す

CT SET DISPLAY (\$Area ; 3 ; 0) `オブジェクトツールを隠す

CT SET DISPLAY (\$Area ; 9 ; 0) `ルーラを隠す

`2D 棒グラフを作成する

\$Chart:= **CT Chart data** (\$Area ; 2 ; 1 ; 1 ; 1 ; **File**(>>[Scores]) ; 2 ; 4 ; 3)

`水平棒グラフをにする

ARRAY LONGINT(\$aOptions ; 4)

\$aOptions{1}:=1 `方向：水平

\$aOptions{2}:=0 `積み重ねなし

\$aOptions{3}:= 0 `重ねるなし

\$aOptions{4}:=50 `間隔 = 50%

CT SET CHART OPTIONS (\$Area;\$Chart;\$aOptions)

`軸タイトルを表示する

CT SET TITLE ATTRIBUTES (\$Area ; \$Chart ; 0 ; 2 ; 3 ; "テスト日付")

CT SET TITLE ATTRIBUTES (\$Area ; \$Chart ; 2 ; 3 ; 0 ; "平均スコア(ポイント換算)")

`項目軸タイトルをフォーマットする (Palatino, 緑, 太字, 12ポイント)

\$Color:= **CT Index to color** (9)

\$Font:= **CT Font number** ("Palatino")

CT SET CHART TEXT ATTRIBUTES (\$Area ; \$Chart ; 5 ; 0 ; \$Font ; 12 ; 1 ; \$Color)

`数値軸タイトルをフォーマットする (Helvetica, 赤, 太字斜体, 12ポイント)

\$Color:= **CT Index to color** (4)

\$Font:= **CT Font number** ("Helvetica")

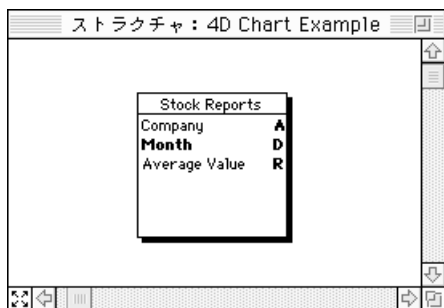
CT SET CHART TEXT ATTRIBUTES (\$Area ; \$Chart ; 5 ; 2 ; \$Font ; 10 ; 3 ; \$Color)

```

`凡例位置を下部中央、横方向にする
CT SET LEGEND ATTRIBUTES ($Area ; $Chart ; 1 ; 0 ; 0 ; 0 ; 8 ; 0 ; 0)
`チャートタイトルを左上隅に追加する
$Title:= CT Draw text ($Area ; 1 ; 1 ; 350 ; 3 ; "Member State Test Scores")
`チャートタイトルをフォーマットする (Geneva, 14ポイント, 太字, 中央揃え, 青)
$Color:= CT Index to color (7)
$Font:= CT Font number ("Geneva")
CT SET TEXT ATTRIBUTES ($Area ; $Title ; $Font ; 14 ; 1 ; $Color ; 1)
`中央揃え用に使用するウィンドウ寸法を取得する
CT GET AREA BOUNDARY ($Area ; 1 ; $Left ; $Top ; $Right ; $Bottom)
`チャートをウィンドウサイズから50ポイント小さく変更する
CT SIZE ($Area ; $Chart ; $Right-50 ; $Bottom-50)
`チャートを横方向に中央揃えする
CT GET BOUNDARY ($Area ; $Chart ; $Left2 ; $Top2 ; $Right2 ; $Bottom2)
$Locate:=((($Right-$Left)-($Right2-$Left2))/2)
CT MOVE ($Area ; $Chart ; $Locate ; $Top2)
`タイトルを横方向に中央揃えする
CT GET BOUNDARY ($Area ; $Title ; $Left2 ; $Top2 ; $Right2 ; $Bottom2)
$Locate:=((($Right-$Left)-($Right2-$Left2))/2)
CT MOVE ($Area ; $Title ; $Locate ; $Top2)
`チャートをタイトルから10ポイント分下げる
CT GET BOUNDARY ($Area;$Chart;$Left;$Top;$Right;$Bottom)
CT MOVE ($Area;$Chart;$Left;$Top+10)
`チャートとタイトルに灰色の矩形で枠組みをする
$Rect:= CT Draw rectangle ($Area ; $Left-2 ; $Top2-2 ; $Right+2 ; $Bottom+2+10 ; 0)
CT SET FILL ATTRIBUTES ($Area ; $Rect ; 3 ; CT Index to color (13))
CT SET LINE ATTRIBUTES ($Area ; $Rect ; 3 ; CT Index to color (15) ; 1)
`すべてのオブジェクトを縦方向に中央揃えする
CT GET AREA BOUNDARY ($Area ; 1 ; $Left ; $Top ; $Right ; $Bottom)
CT GET BOUNDARY ($Area ; -1 ; $Left2 ; $Top2 ; $Right2 ; $Bottom2)
$Locate:=((($Bottom-$Top)-($Bottom2-$Top2))/2)
CT MOVE ($Area ; -1 ; $Left2 ; $Locate)
`矩形を背面に送る
CT SELECT ($Area ; -1 ; 0) `すべての選択を解除する
CT SELECT ($Area ; $Rect ; 1) `矩形を選択する
CT DO COMMAND ($Area ; 24002) `背面に送る
CT SELECT ($Area ; -1 ; 0) `すべての選択を解除する
    
```

レコードのデータによって定義された系列を使って3次元グラフを作成する

自分の会社の株式が同業他社と比べてどのような動きをしているのかを知りたいと仮定します。各企業の株式の終値を4th Dimensionデータベースで記録する場合があります。



次のグラフは、Joe Bob's Chemicals社の株価を業界の平均と比較しています。グラフ上のデータは何の操作もせずにデータベースから直接取られたものです。



「在庫チャート」プロジェクトメソッドは、データベースから直接取られたフィールド値を使用してグラフを生成します。データベースに格納された値はすでに平均として計算されているので、追加の計算を行う必要はありません。代わりに日々の値を記録していた場合には、月ごとの平均値を計算し、それを配列に格納してから**CT Chart arrays**関数を使用してグラフを作成する必要があるでしょう。

注：月ごとの値を平均化する例として、この章で後述する「配列を使って2次元グラフを作成する」を参照してください。配列から3次元のグラフを作成する例は「配列を使って3次元グラフを作成する」を参照してください。

次は、「在庫チャート」メソッドのコードです。

メソッド：ストックチャート

項目：月

系列：企業と業界のインデックス

数値：平均株価

C_LONGINT (\$Left ; \$Top ; \$Right ; \$Bottom)

C_LONGINT (\$Left2 ; \$Top2 ; \$Right2 ; \$Bottom2)

C_LONGINT (\$Area ; \$Chart ; \$Title ; \$Locate ; \$Font ; \$Color)

`チャート用のレコード選択を生成する

ALL RECORDS([Stock Reports])

`チャートを表示するために外部ウィンドウをオープンする

\$Area:=External window(2 ; 40 ; 550 ; 400 ; 8 ; "My Chart" ; "_4D Chart")

`3D線グラフを作成する

\$Chart:= CT Chart data (\$Area ; 101 ; 2 ; 1 ; 1 ; **File**(>>[Stock Reports]) ; 2 ; 1 ; 3)

`チャートの回転と仰角をそれぞれ1度に設定する

CT SET 3D VIEW (\$Area ; \$Chart ; 1 ; 1)

`系列のカラーを設定する (緑, 赤)

CT SET CHART FILL ATTRIBUTES (\$Area ; \$Chart ; 8 ; 100 ; 3 ; **CT Index to color** (10))

CT SET CHART FILL ATTRIBUTES (\$Area ; \$Chart ; 8 ; 200 ; 3 ; **CT Index to color** (4))

`グラフ軸ラベルを表示する

CT SET LABEL ATTRIBUTES (\$Area ; \$Chart ; 2 ; 3 ; 0 ; "###,##0")

CT SET LABEL ATTRIBUTES (\$Area ; \$Chart ; 1 ; 0 ; 0 ; "") `このラベルを隠す

CT SET LABEL ATTRIBUTES(\$Area ; \$Chart ; 0 ; 3 ; 3 ; "#####")

`項目軸と数値軸のタイトルを追加する

CT SET TITLE ATTRIBUTES (\$Area ; \$Chart ; 2 ; 2 ; 3 ; "Average Stock Price in Dollars")

CT SET TITLE ATTRIBUTES (\$Area ; \$Chart ; 0 ; 3 ; 0 ; "Date")

`凡例位置を上部中央、横方向に設定する

CT SET LEGEND ATTRIBUTES (\$Area ; \$Chart ; 1 ; 0 ; 0 ; 0 ; 7 ; 0 ; 0)

`チャートタイトルを左上隅に追加する

\$Title:= **CT Draw text** (\$Area ; 1 ; 1 ; 350 ; 3 ; "Company Performance vs Industry Average")

`チャートタイトルをフォーマットする(Palatino, 14ポイント, 太字, 中央揃え, 青)

\$Color:= **CT Index to color** (7)

\$Font:= **CT Font number** ("Palatino")

CT SET TEXT ATTRIBUTES (\$Area ; \$Title ; \$Font ; 14 ; 1 ; \$Color ; 1)

CT SET TEXT ATTRIBUTES (\$Area ; \$Title ; \$Font ; 14 ; 1 ; \$Color ; 1)

`エリアのオブジェクトを中央揃えするためにエリア境界を取得する

CT GET AREA BOUNDARY (\$Area ; 1 ; \$Left ; \$Top ; \$Right ; \$Bottom)

`チャートを中央揃えする

CT GET BOUNDARY (\$Area ; \$Chart ; \$Left2 ; \$Top2 ; \$Right2 ; \$Bottom2)

\$Locate:=(((\$Right-\$Left)-(\$Right2-\$Left2))/2)

CT MOVE (\$Area ; \$Chart ; \$Locate ; \$Top2)

`チャートタイトルを中央揃えする

CT GET BOUNDARY (\$Area ; \$Title ; \$Left2 ; \$Top2 ; \$Right2 ; \$Bottom2)

\$Locate:=(((\$Right-\$Left)-(\$Right2-\$Left2))/2)

CT MOVE (\$Area ; \$Title ; \$Locate ; \$Top2)

`チャートを10ポイント下げる

CT GET BOUNDARY (\$Area ; \$Chart ; \$Left ; \$Top ; \$Right ; \$Bottom)

CT MOVE (\$Area ; \$Chart ; \$Left ; \$Top+10)

配列を使ってグラフを作成する

次の2つの節では、CT Chart arrays関数を使用して、2次元グラフと3次元グラフを作成する例を示します。

配列を使って2次元グラフを作成する

あなたが部品を製造する企業のエンジニアリング部門の新しいマネージャになったと仮定してください。あなたの新しい業務の1つとして、あなたが管理する部署が顧客からの支援要求に回答するときの時間を短縮することがあります。

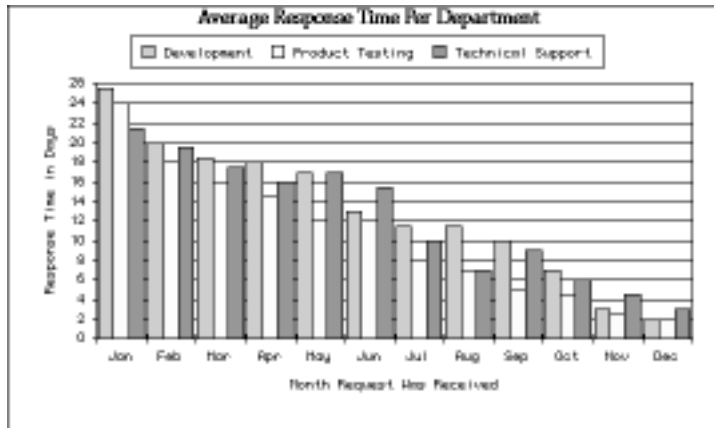
あなたは、各部署の応答時間を追跡するためにデータベースを設計することにしました。データベースの構造は以下のとおりです。

ストラクチャ: 4D Chart Example

Request	
Department	A
Request Date	D
Completion Date	D

収集した情報を解析するために、去年の各月について、各部署の平均応答時間((完了日 - 要求日) / 要求数) をグラフ化することにしました。

4D Chartを使用して、次のグラフを生成します。



前述のグラフが生成され、「応答グラフ」プロジェクトメソッドでフォーマットされました。このメソッドは配列にデータを入れ、その配列からグラフを生成してから、軸タイトルや系列のカラーなどのグラフの特定の外観をカスタマイズします。

メソッドの最初の部分は4th Dimensionコマンドと関数を使用して配列の作成とデータの指定を行っています。項目配列の内容はコードで指定され、系列配列の内容はデータベースから直接取り込まれ、数値配列はデータ操作の結果です。数値配列のサイズは系列配列のサイズを項目配列のサイズで乗算したものと同じです。

このメソッドは、レコードの選択を操作するためにセットを使用します。セットを作成した後は、必要とされるレコードの選択を検索によって変更し、何度でもレコードのオリジナルセットに戻すことができます。このプロシージャでは、セットはグラフ全体のレコードの選択に使用されます。各月の各部署のレコードの検索の結果行われる選択で値が決まるので、選択は数値配列へのデータの指定中に変更されます。

配列にデータが指定された後、4D Chartコマンドがグラフの作成とその機能の変更のために使用されます。

次は、「応答グラフ」メソッドのコードです。

メソッド：応答グラフ

項目：月

系列：部門名

数値：作業期間の平均(日単位)

C_LONGINT (\$x ; \$y ; \$z ; \$Counter)

C_LONGINT (\$Left ; \$Top ; \$Right ; \$Bottom)

C_LONGINT (\$Left2 ; \$Top2 ; \$Right2 ; \$Bottom2)

C_LONGINT (\$Area ; \$Chart ; \$Title ; \$Locate ; \$Duration)

`項目配列を定義し、値を入れる

ARRAY STRING(3 ; \$aCategories ; 12)

\$aCategories{1}:="Jan"

\$aCategories{2}:="Feb"

\$aCategories{3}:="Mar"

\$aCategories{4}:="Apr"

\$aCategories{5}:="May"

\$aCategories{6}:="Jun"

\$aCategories{7}:="Jul"

\$aCategories{8}:="Aug"

\$aCategories{9}:="Sep"

\$aCategories{10}:="Oct"

\$aCategories{11}:="Nov"

```

`チャート用のレコード選択を生成する
QUERY BY FORMULA([Request] ; Year of([Request]Request Date)=1993)
`後で使用するためにレコードをセットに格納する
CREATE SET([Request] ; "sChartData")
`系列配列を部門名で定義し、データを入れる
ARRAY STRING(20 ; $aSeries ; 0)
DISTINCT VALUES([Request]Department ; $aSeries)
`数値の数でチャートを判断する
`(数値の数=項目の数*系列の数)
`数値配列を設定する
ARRAY REAL($aValues ; 12*Size of array($aSeries))
`数値配列にデータを入れる
`各部門で、月ごとの平均作業期間を検出する
$Counter:=0 `カウンタは数値の数を追跡する
For ($x ; 1 ; Size of array($aSeries)) 部門の数だけループする
    For($y ; 1 ; 12) `12ヵ月分ループする
        $Counter:=$Counter+1
        `各部門で、部門のレコードを検出する
        QUERY SELECTION([Request];[Request]Department=$aSeries{$x})
        `この選択内で、月ごとのレコードを検出する
        QUERY SELECTION BY FORMULA([Request] ; Month of([Request]Request
        Date)=$y)
        If (Records in selection([Request])>0)
            $Duration:=0 `間隔を受け取る
            `レコードをループして、応答時間を合計する
            For($z; 1 ; Records in selection([Request]))
                GOTO SELECTED RECORD([Request] ; $z)
                $Duration:=$Duration+([Request]Completion Date-[Request]Request Date)
            End for
            `平均時間を取得し、それを$aValues配列に入れる
            $aValues{$Counter}:=$Duration / Records in selection([Request])
        End if
        `レコードの元のセレクションを復元する
        USE SET("sChartData")
    End for
End for
CLEAR SET("sChartData") `終了したときにセットをクリアする

```

```

`チャートを表示するために外部ウィンドウをオープンする
$Area:=External window (2 ; 40 ; 550 ; 400 ; 8 ; "My Chart" ; "_4D Chart")
`棒グラフを作成する
$Chart:= CT Chart arrays ($Area ; 2 ; 2 ; $aCategories ; $aSeries ; $aValues)
`項目軸と数値軸のタイトルを追加する
CT SET TITLE ATTRIBUTES ($Area ; $Chart ; 2 ; 2 ; 3 ; "Response Time in Days")
CT SET TITLE ATTRIBUTES ($Area ; $Chart ; 0 ; 3 ; 0 ; "Month Request Was Received")
`系列のカラーを設定する(緑、黄色、青)
CT SET CHART FILL ATTRIBUTES ($Area ; $Chart ; 8 ; 100 ; 3 ; CT Index to color (10))
CT SET CHART FILL ATTRIBUTES ($Area ; $Chart ; 8 ; 200 ; 3 ; CT Index to color (2))
CT SET CHART FILL ATTRIBUTES ($Area ; $Chart ; 8 ; 300 ; 3 ; CT Index to color (7))
`凡例位置を上部中央、横方向に設定する
CT SET LEGEND ATTRIBUTES ($Area ; $Chart ; 1 ; 0 ; 0 ; 0 ; 7 ; 0 ; 0)
`チャートタイトルを左上隅に追加する
$Title:= CT Draw text ($Area ; 1 ; 1 ; 275 ; 3 ; "Average Response Time Per Department")
`チャートタイトルをフォーマットする (Palatino, 14ポイント, 太字, 中央揃え, 黒)
CT SET TEXT ATTRIBUTES ($Area ; $Title ; CT Font number ("Palatino");
14 ; 1 ; CT Index to color (16) ; 1)
`ウィンドウのオブジェクトを中央揃えするためにチャートエリアの寸法を検出する
CT GET AREA BOUNDARY ($Area ; 1 ; $Left ; $Top ; $Right ; $Bottom)
`チャートを中央揃えする
CT GET BOUNDARY ($Area ; $Chart ; $Left2 ; $Top2 ; $Right2 ; $Bottom2)
$Locate:=((($Right-$Left)-($Right2-$Left2))/2)
CT MOVE ($Area ; $Chart ; $Locate ; $Top2)
`チャートタイトルを中央揃えする
CT GET BOUNDARY ($Area ; $Title ; $Left2 ; $Top2 ; $Right2 ; $Bottom2)
$Locate:=((($Right-$Left)-($Right2-$Left2))/2)
CT MOVE ($Area ; $Title ; $Locate ; $Top2)
`チャートを10ポイント下げ、チャートタイトル用に空間を開ける
CT GET BOUNDARY ($Area ; $Chart ; $Left ; $Top ; $Right ; $Bottom)
CT MOVE ($Area ; $Chart ; $Left ; $Top+10)

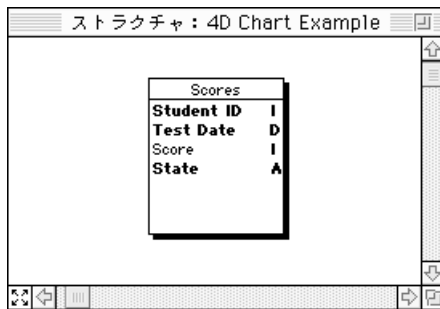
```

配列を使って3次元グラフを作成する

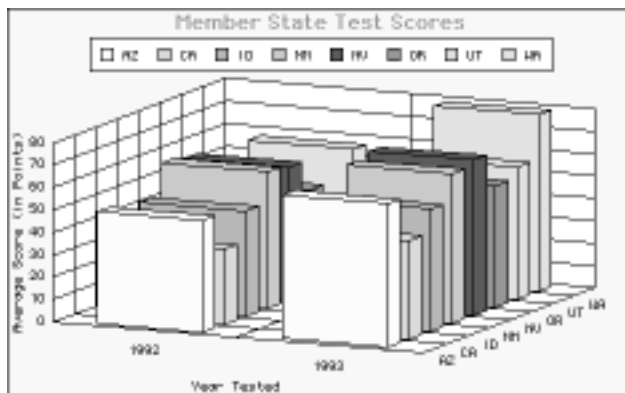
あなたが教育プログラムの西日本地区担当者であり、あなたの地区の生徒達のテストスコアが向上したかどうかを判断したいと仮定します。

あなたのデータベースには、テスト日付と出身地にタグが付いて、あなたのエリアの生徒のスコアが保持されています。

注：これはこの章で前述した「レコードのデータによって定義された系列を使って2次元グラフを作成する」で使用したのと同じサンプルデータベースです。



毎年複数のテスト日があるので、データベースから直接データをグラフ化することはできません。その代わりに、データを配列に入れるプロシージャを作成し、各年を“項目”、各州を“系列”として使用することができます。



前述のグラフは「スコア3Dグラフ」プロジェクトメソッドを使用して生成されています。このメソッドは配列にデータを入れ、配列からグラフを生成した後、グラフの特定の外観をカスタマイズします。

項目配列の内容はコードに直接指定されます。系列配列の内容はデータベースから取られます。数値配列の内容は年ごとの各都道府県のテストスコアの平均です。数値配列のサイズは系列配列のサイズと項目配列のサイズを乗算したものと同じです。

4D Chartコマンドを使用すると、4D Chartメニューバー、スクロールバー、ルーラ、ツールパレットは隠されます。4D Chartエリアは「入力不可」に設定され、ユーザはそのエリアのオブジェクトを選択したり、それに対する変更をまったく行えなくなります。

次は、「スコア3Dグラフ」メソッドのコードです。

メソッド：スコア3Dグラフ

項目：調査年

系列：メンバの州

数値：テストスコア

C_LONGINT (\$x ; \$y ; \$z ; \$Counter)

C_LONGINT (\$Left ; \$Top ; \$Right ; \$Bottom)

C_LONGINT (\$Left2 ; \$Top2 ; \$Right2 ; \$Bottom2)

C_LONGINT (\$Area ; \$Chart ; \$Title ; \$Locate ; \$Score ; \$Color ; \$Font ; \$Rect)

`項目配列を定義し、データを指定する

ARRAY STRING(4 ; \$aCategories ; 2)

\$aCategories{1}:="1992"

\$aCategories{2}:="1993"

`系列配列に州名を定義し、そのデータを指定する

ALL RECORDS([Scores])

ARRAY STRING (2 ; \$aSeries ; 0)

DISTINCT VALUES ([Scores]State ; \$aSeries)

`数値配列を設定する (グラフへの数値の数=項目*系列)

ARRAY REAL(\$aValues ; 2***Size of array**(\$aSeries))

`数値配列を判断し、データを指定する

`各都道府県に対して、平均テストスコアを検出する

\$Counter:=0 `カウンタは数値を追跡する

For (\$x ; 1 ; **Size of array**(\$aSeries))

For (\$y ; 1 ; 2) `2年間だけループする

 \$Counter:=\$Counter+1

QUERY([Scores];[Scores]State=\$aSeries{\$x}) `州の数だけループする

QUERY SELECTION BY FORMULA([Scores];**String**(**Year of**([Scores]Test Date))=\$aCategories{\$y}))


```

If (Records in selection([Scores])>0)
  $Score:=0 `スコアの合計を保持する
  `レコードをループし、スコアを合計する
  For ($z ; 1 ; Records in selection([Scores]))
    GOTO SELECTED RECORD([Scores] ; $z)
    $Score:=$Score+[Scores]Score
  End for
  `スコアの平均を出す
  $aValues{$Counter}:=$Score / Records in selection([Scores])
End if
End for
End for

`チャートを表示するために外部ウィンドウをオープンする
$Area:=External window (2 ; 40 ; 500 ; 350 ; 4 ; "My Chart" ; "_4D Chart")
`インタフェース要素を隠す
CT SET DISPLAY ($Area ; 1 ; 0) `4D Chartメニューバーを隠す
CT SET DISPLAY ($Area ; 2 ; 0) `チャートツールを隠す
CT SET DISPLAY ($Area ; 3 ; 0) `オブジェクトツールを隠す
CT SET DISPLAY ($Area ; 6 ; 0) `スクロールバーを隠す
CT SET DISPLAY ($Area ; 9 ; 0) `ルーラを隠す
`3D 棒グラフを作成する
$Chart:= CT Chart arrays ($Area ; 100 ; 2 ; $aCategories ; $aSeries ; $aValues)
`別の軸タイトルを表示または隠す
CT SET TITLE ATTRIBUTES ($Area ; $Chart ; 0 ; 3 ; 0 ; "Year Tested")
CT SET TITLE ATTRIBUTES ($Area ; $Chart ; 1 ; 1 ; 0 ; "State") `Hide title
CT SET TITLE ATTRIBUTES ($Area ; $Chart ; 2 ; 2 ; 3 ; "Average Score (in Points)")
`凡例位置を上部中央、横方向に設定する
CT SET LEGEND ATTRIBUTES ($Area ; $Chart ; 1 ; 0 ; 0 ; 0 ; 7 ; 0 ; 0)
`チャートタイトルを左上隅に追加する
$Title:= CT Draw text ($Area ; 1 ; 1 ; 350 ; 3 ; "Member State Test Scores")
`チャートタイトルをフォーマットする (Geneva, 14ポイント, 太字, 中央揃え, 青)
$Color:= CT Index to color (7)
$Font:= CT Font number ("Geneva")
CT SET TEXT ATTRIBUTES ($Area ; $Title ; $Font ; 14 ; 1 ; $Color ; 1)
`中央揃えのためにウィンドウの寸法を取得する
CT GET AREA BOUNDARY ($Area ; 1 ; $Left ; $Top ; $Right ; $Bottom)
`チャートをウィンドウサイズから50ポイント小さく変更する
CT SIZE ($Area ; $Chart ; $Right-50 ; $Bottom-50)

```

```

`チャートを横方向に中央揃えする
CT GET BOUNDARY ($Area ; $Chart ; $Left2 ; $Top2 ; $Right2 ; $Bottom2)
$Locate:=((($Right-$Left)-($Right2-$Left2))/2)
CT MOVE ($Area ; $Chart ; $Locate ; $Top2)
`タイトルを横方向に中央揃えする
CT GET BOUNDARY ($Area ; $Title ; $Left2 ; $Top2 ; $Right2 ; $Bottom2)
$Locate:=((($Right-$Left)-($Right2-$Left2))/2)
CT MOVE($Area ; $Title ; $Locate ; $Top2)
`チャートをタイトルから10ポイント分下げる
CT GET BOUNDARY ($Area ; $Chart ; $Left ; $Top ; $Right ; $Bottom)
CT MOVE($Area ; $Chart ; $Left ; $Top+10)
`チャートとタイトルに灰色の矩形で枠組みをする
$Rect:= CT Draw rectangle ($Area ; $Left-2 ; $Top2-2 ; $Right+2 ; $Bottom+2+10 ; 0)
CT SET FILL ATTRIBUTES ($Area ; $Rect ; 3 ; CT Index to color (13))
`すべてのオブジェクトを縦方向に中央揃えする
CT GET AREA BOUNDARY ($Area ; 1 ; $Left ; $Top ; $Right ; $Bottom)
CT GET BOUNDARY ($Area;-1;$Left2;$Top2;$Right2;$Bottom2)
$Locate:=((($Bottom-$Top)-($Bottom2-$Top2))/2)
CT MOVE ($Area;-1;$Left2;$Locate)
`灰色の矩形を奥に下げる
CT SELECT ($Area ; -1 ; 0) `すべての選択を解除する
CT SELECT ($Area ; $Rect ; 1) `矩形を選択する
CT DO COMMAND ($Area ; 24002) `背面を送る
CT SELECT ($Area ; -1 ; 0) `すべての選択を解除する
`エリアを入力不可にする
CT SET ENTERABLE ($Area ; 0)

```

次のコマンドと関数は、「ルーチン」エリアの「Area」テーマ内にあります：

CT AREA TO AREA
CT AREA TO FIELD
CT FIELD TO AREA
CT Area to picture
CT PICTURE TO AREA
CT NEW DOCUMENT

CT OPEN DOCUMENT
CT SAVE DOCUMENT
CT New offscreen area
CT DELETE OFFSCREEN AREA
CT GET AREA BOUNDARY

このテーマ内のコマンドと関数によって、フォーム、オフスクリーンエリア、ピクチャフィールド上の4D Chartエリア間で情報の転送ができます。

また、上記の「エリア」コマンドで変更や新しいエリアの作成もできます。作業が完了したときには忘れずにオフスクリーンエリアを削除してください。

CT AREA TO AREA

CT AREA TO AREA (ソース ; コピー先 ; コピーコード)

引数	タイプ	説明
ソース	倍長整数	コピー元となる4D Chartエリア
コピー先	倍長整数	コピーの受け取り先となる4D Chartエリア
コピーコード	整数	コピー内容を表すコード 1=設定内容、2=オブジェクト、3=両方

CT AREA TO AREAコマンドは、4D Chartエリアの引数<ソース>の内容を4D Chartエリアの引数<コピー先>の中にコピーします。

転送する内容は、引数<コピーコード>を基にしています。

<コピーコード>が1の場合は、表示オプションなどの文書設定が転送されます。

<コピーコード>が2の場合は、<ソース>内のすべてのオブジェクトが<コピー先>に転送されます。

<コピーコード>が3の場合は、オブジェクトと文書設定の両方が<コピー先>に転送されます。

文書設定は転送されると、<コピー先>の文書設定を置き換えます。オブジェクトが転送されるときには、<コピー先>内のオブジェクトに付加されます。**CT AREA TO AREA**コマンドは、特にオフスクリーンエリアを操作するときに便利です。

次の例は、4D Chartエリア「売上チャート」の内容を新規オフスクリーンエリアにコピーします。

vOffscreen:= **CT New offscreen area**

CT AREA TO AREA(売上チャート ; vOffscreen ; 3)

参照 : CT New offscreen area、CT DELETE OFFSCREEN AREA

CT AREA TO FIELD

CT AREA TO FIELD (エリア ; スコープ ; テーブル ; フィールド ; {保存方法})

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	コマンドのスコープ -2=文書、-1=すべて 0=選択されたオブジェクト、>0=オブジェクトID
テーブル	整数	テーブル番号
フィールド	整数	フィールド番号
保存方法	整数	引数<エリア>の内容を保存する際の方法 1=ピクチャのみ、2=データのみ 3=ピクチャとデータ、4=変更なし

CT AREA TO FIELDコマンドは、引数<エリア>の内容を<テーブル>と<フィールド>で指定されたピクチャフィールドの中にコピーします。<フィールド>はピクチャフィールドにはできません。

CT AREA TO FIELDコマンドは、リレートしているテーブルのフィールドにオブジェクトを格納したいときや特定のオブジェクトだけを格納したいときに便利です。**CT AREA TO FIELD**コマンドは、単にオブジェクトをフィールド>に割り当てるだけです。<テーブル>のレコードは保存されたままである必要があります。

引数<スコープ>は、コピー内容を制御します。

オプション引数の<保存方法>は、4D Chartエリア内の文書の保存方法を決定します。

<保存方法>が1の場合は、ピクチャ(PICT)だけが保存されます。この設定をすると、オブジェクトを個別に操作することはできなくなります。

<保存方法>が2の場合は、4D Chartエリアのオブジェクトに関連するデータだけが保存されます。イメージは保存されたデータの情報を使用して後で再構築されます。この保存オプションは最速の方法であり、メモリの使用量が最も少ないものです。選択した保存方法に対するメモリが十分ないときには、別の方法を選択できるダイアログボックスが表示されます。

<保存方法>が3の場合は、イメージの再構築に使用されたピクチャと内部データの両方が保存されます。これは、文書を保存する際の通常の方法です。

次の例は、<エリア>にオブジェクト用のリレートレコードを作成します。

```
CREATE RECORD([オブジェクト])`レコードを作成してオブジェクトを格納する
[オブジェクト]キー:=[チャート]名前`リレート値を割り当てる
$Temp:= CT Get ID(エリア ; -1 ; 3)`オブジェクトのIDを取得する
CT AREA TO FIELD(エリア ; $Temp ; 3 ; 2 ; 1)`オブジェクトをレコードにコピーする
SAVE RECORD([オブジェクト])`レコードを保存する
```

参照 : CT FIELD TO AREA

CT FIELD TO AREA

CT FIELD TO AREA (エリア ; テーブル ; フィールド)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
テーブル	整数	テーブル番号
フィールド	整数	フィールド番号

CT FIELD TO AREAコマンドは、引数<テーブル>と<フィールド>で指定されたピクチャフィールドに含まれる文書を<エリア>に配置します。

ピクチャ値は、<テーブル>のカレントレコードから取得されます。

<フィールド>はピクチャタイプである必要があります。<フィールド>には以前に保存された4D Chart文書やピクチャが含まれていることもあります。<フィールド>の内容は<エリア>の内容を置き換えます。<フィールド>が空の場合には、このコマンドは何も行いません。

次の入力フォームのオブジェクトメソッドは、2番目のテーブルの5番目のフィールドに含まれる4D Chart文書をオープンします。

If (Before)

CT FIELD TO AREA (エリア ; 2 ; 5)

End If

参照 : CT AREA TO FIELD

CT Area to picture

CT Area to picture (エリア ; スコープ) ピクチャ

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	整数	コマンドのスコープ -2=文書、-1=すべて、0=選択されたオブジェクト >0=オブジェクトID

CT Area to picture関数は、引数<エリア>にあるオブジェクトの4th Dimensionピクチャを返します。

ピクチャに含まれるオブジェクトは、引数<スコープ>によって制御されます。

<スコープ>が-2の場合は、文書全体がコピーされます。これには表示オプションなどの文書設定も含まれます。

<スコープ>が-1の場合は、文書設定を除く<エリア>内のすべてのオブジェクトがコピーされます。

<スコープ>が0の場合は、選択されたオブジェクトだけがコピーされます。

<スコープ>が0よりも大きい場合は、それが特定オブジェクトIDと同じである必要があり、そのオブジェクトだけがコピーされます。

次の例は、新しいオフスクリーンエリアをオープンして既存の配列から棒グラフを作成し、グラフをピクチャ変数に格納してから最後にオフスクリーンエリアを削除します。

エリア:= **CT New offscreen area**

vChart:= **CT Chart arrays** (エリア ; 2 ; 2 ; aCategory ; aSeries ; aValues)

vPict:= **CT Area to picture** (エリア ; vChart)

CT DELETE OFFSCREEN AREA (エリア)

参照 : CT Get ID、CT PICTURE TO CLIPBOARD、CT PLACE PICTURE

CT PICTURE TO AREA

CT PICTURE TO AREA (エリア ; ピクチャ)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
ピクチャ	ピクチャ	ピクチャ

CT PICTURE TO AREAコマンドは、引数<ピクチャ>に含まれる文書を<エリア>に配置します。

<ピクチャ>は正しい4th Dimensionピクチャ式である必要があります。<エリア>の内容は<ピクチャ>で置き換えられます。<ピクチャ>が空の場合には、このコマンドは何も行いません。

次のオブジェクトメソッドは、ピクチャフィールドから4D Chartエリアにグラフをコピーできます。

```
$Name:=Request ("ロードするチャート名を入力してください。")  
If (OK=1)  
  QUERY([チャート];[チャート]ラベル:=$Name)  
  If (Records in selection([チャート])>0)  
    CT PICTURE TO AREA (エリア ; [チャート]MyChart)  
  End if  
End if
```

参照 : CT Area to picture

CT NEW DOCUMENT

CT NEW DOCUMENT (エリア)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア

CT NEW DOCUMENTコマンドは、引数<エリア>内の文書の内容をクリアします。**CT NEW DOCUMENT**コマンドは確認ダイアログボックスが提示されない以外は、「ファイル」メニューから「新規...」メニューコマンドを選択することと同じです。**CT NEW DOCUMENT**コマンドはすべてのオブジェクトと、文書サイズやルーラの目盛り指定などのすべての文書設定をクリアします。

次の例は、<エリア>内の文書をクリアします。

CT NEW DOCUMENT (エリア)

警告：このコマンドを使用すると、<エリア>内の現在の文書は保存されません。現在の文書を保存する場合には、**CT NEW DOCUMENT**コマンドをコールする前に**CT SAVE DOCUMENT**コマンドをコールする必要があります。

参照：CT SAVE DOCUMENT

CT OPEN DOCUMENT

CT OPEN DOCUMENT (エリア ; 文書 ; {モード})

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
文書	文字列	文書の名前 (パスは最大255バイト)
モード	整数	文書置き換え、または文書追加 0=置き換え、1=文書に追加する

CT OPEN DOCUMENTコマンドは、引数<文書>をオープンし、その内容を<エリア>に配置します。

<文書>が空の文字列の場合には、**CT OPEN DOCUMENT**コマンドは、ユーザが文書の選択をできる標準「ファイルを開く」ダイアログボックスを表示します。

<文書>に文字列が入っている場合には、指定された文書のオープンが行われます。<文書>が存在しない場合には、<エリア>の内容は変更されず、**CT Error**関数がシステムエラーコードを返します。

4D Chartはデータベースストラクチャを持つフォルダに<文書>があることを想定しています。データベースフォルダ以外の文書をオープンする場合には、完全なパス名を指定してください。パス名に関する詳細は、『4th Dimension ランゲージリファレンス』を参照してください。<文書>がすでにオープンされている場合には、**CT Error**関数がシステムエラーコードを返します。

オプション引数の<モード>は、文書のオープン方法を制御します。<モード>は、<文書>が空の文字列ではなく、かつ4D Chart文書ではないときにだけ使用されます。<モード>が0の場合、または指定されていない場合は、<文書>は<エリア>の内容を置き換えます。<モード>が1の場合には、<文書>は<エリア>の現在の内容に組み入れられます。

次の例は、「形態」フィールドの値に基づいて、異なる文書をオープンします。

Case of

¥ ([顧客]形態="販売店") `形態が "販売店" の場合

CT OPEN DOCUMENT (エリア ; "販売店") ` "販売店" 文書を開く

¥ ([顧客]形態="Constructor") `形態が "建設" の場合

CT OPEN DOCUMENT (エリア ; "建設") ` "建設" 文書を開く

¥ ([顧客]形態="エンドユーザ") `形態が "エンドユーザ" の場合

CT OPEN DOCUMENT (エリア ; "エンドユーザ") ` "エンドユーザ" 文書を開く

End case

参照 : CT SAVE DOCUMENT

CT SAVE DOCUMENT

CT SAVE DOCUMENT (エリア ; 文書 ; タイプ ; {スコープ})

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
文書	テキスト	文書の名前 (パスを持った)
タイプ	文字列	文書タイプ
スコープ	倍長整数	コマンドのスコープ 0=全オブジェクト、選択されたオブジェクト

CT SAVE DOCUMENTコマンドは、引数<エリア>の内容を<文書>に保存します。

<文書>が空の文字列の場合は、CT SAVE DOCUMENTコマンドは、文書名、タイプ、スコープをユーザが指定できる標準「ファイル保存」ダイアログボックスを表示します。<文書>が空の文字列ではない場合は、CT SAVE DOCUMENTコマンドは<タイプ>で<文書>を保存します。

<文書>が存在しない場合は、CT SAVE DOCUMENTコマンドがその文書を作成します。<文書>が存在する場合にはCT SAVE DOCUMENTコマンドはその文書を上書きします。

<タイプ>が空の文字列の場合は、標準の4D Chart文書が作成されます。PICTとして文書を保存するには、<タイプ>が「PICT」である必要があります。

オプション引数の<スコープ>は、<文書>に保存するものを制御します。<文書>が空文字列ではなく、かつ文書をPICTとして保存するときだけにだけ<スコープ>を使用してください。

デフォルトでは、<文書>はデータベース構造を含むフォルダに保存されます。データベースフォルダ以外に文書を保存する場合には、完全なパス名を指定します。パス名に関する詳細は、『4th Dimension ランゲージリファレンス』を参照してください。

次の例は、4D Chart文書を会社名と年度を名前とする文書として保存します。

```
$Year := Request ("何年?") `年度を要求する
If (OK=1) `要求が正当である場合
    $SaveName := [会社]会社名+" " + $Year `文書名を連結する
    CT SAVE DOCUMENT (エリア ; $SaveName ; "") `文書を保存する
End if
```

参照 : CT OPEN DOCUMENT

CT New offscreen area

CT New offscreen area 倍長整数

CT NEW offscreen area関数は4D Chartオフスクリーンエリアを作成し、そのエリアのIDを返します。**CT NEW offscreen area**関数から返される値は、4D Chartエリアを必要とする任意の4D Chartコマンドで使用できます。

次の例は、レコードを検索し、オフスクリーンエリアを作成し、レコードからエリアに文書をコピーしてから、そのエリアを印刷します。

```
QUERY ([テーブル];[テーブル]フィールド1="レベル1")`レコードを検索する
$Offscreen := CT New offscreen area`新しいオフスクリーンエリアを作成する
CT FIELD TO AREA ($Offscreen ; 3 ; 2)`フィールドに格納された文書をコピーする
CT PRINT ($Offscreen ; 0)`エリアを印刷する
CT DELETE OFFSCREEN AREA ($Offscreen)`オフスクリーンエリアの使用をやめる
```

参照 : CT DELETE OFFSCREEN AREA、CT AREA TO AREA

CT DELETE OFFSCREEN AREA

CT DELETE OFFSCREEN AREA (エリア)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア

CT DELETE OFFSCREEN AREAコマンドは、**CT New offscreen area**関数で作成された4D Chartオフスクリーンエリアを廃棄し、使用されたメモリを解放します。

引数<エリア>はフォーム上やウィンドウ内のエリアではなくオフスクリーンエリアである必要があります。オフスクリーンエリアの使用を完了したときには常に**CT DELETE OFFSCREEN AREA**コマンドをコールしてください。

次の例は、**CT New offscreen area**コールと対応する**CT DELETE OFFSCREEN AREA**コールをペアにする方法を示しています。

```
$NewArea := CT New offscreen area`新しいオフスクリーンエリアを作成する
`ここで何らかの処理を行う
CT DELETE OFFSCREEN AREA ($NewArea)`オフスクリーンエリアを消去する
```

参照 : CT AREA TO AREA、CT New offscreen area

CT GET AREA BOUNDARY

CT GET AREA BOUNDARY (エリア ; 境界コード ; 左 ; 上 ; 右 ; 下)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
境界コード	整数	境界コード 0=文書全体の境界、1=切り取られた境界
左	実数	エリアの左の境界を受け取る
上	実数	エリアの上の境界を受け取る
右	実数	エリアの右の境界を受け取る
下	実数	エリアの下の境界を受け取る

CT GET AREA BOUNDARYコマンドは、引数<左>、<上>、<右>、<下>の各変数に<エリア>矩形の座標を返します。

<境界コード>が0の場合、**CT GET AREA BOUNDARY**コマンドは文書全体の境界を返します。<境界コード>が1の場合は、**CT GET AREA BOUNDARY**コマンドはフォーム上の4D Chartエリアまたは4D Chart外部ウィンドウの現在のサイズの境界を返します。

次の例は、既存のチャートエリアで複数の線を組み合わせたジオメトリックオブジェクトを作成し、エリアの境界座標を取得してエリア内のオブジェクトを中央に配置します。

```

For ($i ; 0 ; 360 ; 5)
  vLine:= CT Draw line (エリア ; 50*Cos($i) ; 50*Sin($i) ; 0 ; 0 ; 0)
End for
CT GET AREA BOUNDARY (エリア ; 1 ; $left ; $top ; $right ; $bottom)
CT MOVE (エリア ; -1 ; (($right-$left)/2)-50 ; (($bottom-$top)/2)-50)
    
```


次のコマンドと関数は、「ルーチン」エリアの「Area Control」テーマ内にあります：

CT DO COMMAND	CT Last event
CT MENU STATUS	CT SET ENTERABLE
CT ON ERROR	CT Get display
CT ON EVENT	CT SET DISPLAY
CT ON MENU	CT GET DOCUMENT SIZE
CT Error	CT SET DOCUMENT SIZE
CT EVENT FILTER	CT GET PROPERTIES
CT EXPERT COMMAND	CT SET PROPERTIES
CT EXPERT MODE	

これらのコマンドと関数を使用すると、4D Chartエリアとそのエリアで実行されるアクションの現在のメニューステータスを追跡できます。さらに、4D Chart文書に変更を行うユーザの機能も制御できます。また、メソッドによって4D Chartのメニューコマンドを実行することもできます。

CT DO COMMAND

CT DO COMMAND (エリア ; コマンド)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
コマンド	倍長整数	コマンド番号

CT DO COMMANDコマンドは、引数<コマンド>で指定されるメニューコマンドを実行します。メニューコマンドは、ユーザが4D Chartメニューから選択したのと同じように実行されます。このコマンドを使用すると、プログラムで同等のものがないアクションでも実行できます。

<コマンド>に指定できる値は、付録Bにリストされています。4D Chartの将来のバージョンでメニューコマンドが変更されたり、位置が変わってもこれらの数値は変わりません。

次の例は、指定されたエリアのすべてのオブジェクトを選択し、複製します。

`「編集」メニューから「すべてを選択」を選択するのと同じ

CT DO COMMAND (エリア ; 2009)

`「編集」メニューから「複製」を選択するのと同じ

CT DO COMMAND (エリア ; 2007)

CT MENU STATUS

CT MENU STATUS (エリア ; コマンド ; チェック ; 使用可 ; 名前)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
コマンド	倍長整数	コマンド番号
チェック	整数	メニューコマンドがチェックされているか? 0=チェックなし、1=チェックあり
使用可	整数	メニューコマンドが使用可能かどうか? 0=使用不可、1=使用可能
名前	文字列	メニューコマンドの名前

CT MENU STATUSコマンドは、引数<コマンド>によって表される<エリア>内のメニューコマンドに関する情報を<チェック>、<使用可>、<名前>変数に返します。

<コマンド>に指定可能な値は、付録Bにリストされています。

<使用可>が0の場合は、そのメニューコマンドは使用不可です。<使用可>が1の場合は、そのメニューコマンドは使用可能です。

<チェック>が0の場合は、そのメニューコマンドはチェックされていません。<チェック>が1の場合はそのメニューコマンドはチェックされています。

<名前>は、メニューコマンドの名前です。

次の例は、エリアが「参照表示」モードまたは「値表示」モードかどうかを確認するためにメニューコマンドをチェックします。エリアが「参照表示」モードの場合は、「値表示」モードをオンにします。

```
CT MENU STATUS (エリア ; 6006 ; $Checked ; $Available ; $Name)
If ($Name="参照表示")
    CT DO COMMAND (エリア ; 6006)
End if
```

参照 : CT DO COMMAND

CT ON ERROR

CT ON ERROR (メソッド)

引数	タイプ	説明
メソッド	文字列	実行するメソッド

CT ON ERRORコマンドは、4D Chartエラーの管理用メソッドとして引数<メソッド>をインストールします。エラー処理メソッドをインストールすると、4D Chartは4D Chartエラーが発生したときに<メソッド>をコールします。

<メソッド>が空の文字列の場合は、メソッドはコールされず、エラー処理は4D Chartに戻ります。

4D Chartは、<メソッド>をコールするときにエラー管理用に使用できる3つのパラメータ(\$1、\$2、\$3)を返します。

\$1	倍長整数	エラーが発生した4D Chartエリアを表す。エラーが特定の4D Chartエリアに対するものではない場合には、\$1は0になる。
\$2	倍長整数	エラー番号を保持する。 CT Error 関数へのコールと同じである。
\$3	テキスト	エラーのテキストが含まれる。 CT Error 関数へのコールと同じである。

データベースをコンパイルする予定がある場合は、以下のようにこれらのパラメータのタイプを宣言してください。

C_LONGINT (\$1 ; \$2)

C_TEXT (\$3)

次の例は、エラー処理メソッドのインストレーションを示しています。

CT ON ERROR ("チャートエラー")

次のメソッドは、「チャートエラー」です。「チャートエラー」メソッドは、\$1をテストして、エラーが<エリア>内で発生したかどうかを判断し、次にエラー番号とメッセージを含む警告ボックスを提示します。

C_LONGINT (\$1 ; \$2)

C_TEXT (\$3)

If (\$1 = エリア)

ALERT ("4D Chartエリア内でエラーが発生しました。")

End if

ALERT ("エラーコード " + **String**(\$2) + **Char**(13) + \$3)

参照 : CT Error

CT ON EVENT

CT ON EVENT (メソッド)

引数	タイプ	説明
メソッド	文字列	実行するメソッド

CT ON EVENTコマンドは、事前に指定されたイベントが発生するといつでも引数<メソッド>を実行します。

<メソッド>を実行させるイベントは、**CT ON EVENT**コマンドで記述されます。<メソッド>が空の文字列の場合は、メソッドは実行されません。イベントが発生するエリアにオブジェクトメソッドとイベントメソッドの両方がある場合には、オブジェクトメソッドが後で実行されます。**CT ON EVENT**コマンドは、オブジェクトメソッドを持ってない外部ウィンドウ内の4D Chartエリアで利用すると特に便利です。

4D Chartは、<メソッド>をコールするときにイベント管理のために使用できる4つのパラメータ(\$1、\$2、\$3、\$4)を返します。

\$1	倍長整数	イベントが発生した4D Chartエリアを表す。
\$2	倍長整数	イベント番号を保持する。 CT Last event 関数へのコールと同じである。
\$3	倍長整数	エリアが存在するフォームのテーブル番号。 \$3が-1の場合、そのエリアは外部ウィンドウにある。
\$4	倍長整数	エリアが自動保存されているフィールドの番号。 \$4が0の場合、そのエリアは自動保存されていない。

データベースをコンパイルする予定がある場合は、以下のようにこれらのパラメータのタイプを宣言してください。

C_LONGINT (\$1 ; \$2 ; \$3 ; \$4)

次の例は、イベントメソッドのインストールを示しています。外部ウィンドウをオープンし、イベントとして「Ctrl(コマンド) - クリック」を指定してから、イベントメソッド“EventProc”をインストールします。

外部ウィンドウを開く

```
vArea := External window (20 ; 50 ; 400 ; 350 ; 0 ; "チャート"; "_4D Chart")
```

```
CT ON EVENT ("EventProc") `メソッド “EventProc” をインストールする
```

```
CT EVENT FILTER (vArea ; 64) `「Ctrl(コマンド) - クリック」がメソッドをコールする
```

参照 : **CT EVENT FILTER**、**CT Last event**

CT ON MENU

CT ON MENU (エリア ; メソッド)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
メソッド	文字列	実行するメソッド

このコマンドは、「ユーザ」モードまたは「カスタム」モードのいずれかで、メニューコマンドが起動されるたびに引数<メソッド>を実行します。また、メニューコマンドは、<メソッド>内でコールされる限り、**CT DO COMMAND**コマンドを使用してコールすることもできます。

コールされたメソッドは、次の3つのパラメータを返します：

- \$1 4D ChartエリアのIDを含む倍長整数
- \$2 メニューコマンド番号を含む倍長整数
- \$3 モディファイアキーが押されたときの番号を含む倍長整数

\$3パラメータは、次のモディファイアキーのいずれか1つ(またはモディファイアキーの組み合わせ)に対応します。

- 0 なし
- 1 コマンドキー (Windows版では、Ctrl)
- 2 shiftキー
- 4 optionキー (Windows版では、Alt)
- 8 Controlキー (Windows版には、対応するキーはありません。)

モディファイアキーの組み合わせが押されたときには、それらの値は加算されてパラメータとして渡されます。例えば、値10はメニューコマンドを選択しているときに「shiftキーとcontrolキー」の両方を押したことを示しています。

データベースをコンパイルする予定がある場合は、次のようにこれらのパラメータのタイプを宣言してください。

C_LONGINT (\$1 ; \$2 ; \$3)

次の例は、「MenuProc」イベントメソッドを起動します。

CT ON MENU (エリア ; "MenuProc")

「MenuProc」メソッドは、メニューコマンドへのユーザのアクセスを制御します。「テンプレートとして保存」または「プロパティ...」のいずれかのメニューコマンドが選択されると、ダイアログボックスが表示され、メニュー選択は無効になります。他のすべてのメニューコマンドは中断なく実行されます。

次は、「MenuProc」メソッドのコードです。

C_LONGINT(\$1 ; \$2 ; \$3)

Case of

¥ (\$2=1006) `「テンプレートとして保存」メニューコマンド番号

ALERT("テンプレートを保存できません。")

¥ (\$2=2011) `「プロパティ...」メニューコマンド番号

ALERT("「プロパティ」ダイアログボックスにアクセスすることはできません。")

Else

CT DO COMMAND (vArea ; \$2)

End case

CT Error

CT Error (メッセージ) 整数

引数	タイプ	説明
メッセージ	倍長整数	エラーメッセージ

CT Error関数は、4D Chartで実行された最後の処理のステータスを示す番号を返します。

CT Error関数が0を返した場合には、最後の処理はエラーの原因ではありません。**CT Error**関数が0以外の番号を返した場合は、最後の処理中にエラーが発生しています。1つのフォーム上で複数のエリアがアクティブである場合は、**CT Error**関数はエリアを区別せずに最後のエラーを返します。エラーコードのリストについては、付録Cを参照してください。

オプション引数の<メッセージ>が**CT Error**関数に渡された場合には、コール後のエラーのテキストを含むテキスト変数のはずです。

次の例は、以前のコマンドでエラーが発生したかどうかをチェックします。

If (**CT Error** #0)

`最後の処理がエラーの原因になった

End if

参照 : **CT ON ERROR**

CT EVENT FILTER

CT EVENT FILTER (エリア ; フィルタ)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
フィルタ	倍長整数	使用するイベント

CT EVENT FILTERコマンドは、引数<エリア>のオブジェクトメソッドを発生させるイベントまたは実行するイベントメソッドを指定します。

デフォルトでは、4D Chartエリアに付着したオブジェクトメソッドは、ユーザがエリア以外のオブジェクトを選択したときに実行されます。**CT EVENT FILTER**コマンドでは、オブジェクトメソッドを実行する他のイベントを指定できます。さらに、**CT ON EVENT**コマンドでインストールされたメソッドも実行できます。

引数<フィルタ>はイベントコードの加算値として使用するイベントを指定します。下記にイベントコードを示します：

値	イベント
-1	すべてのイベント
0	イベントなし
1	エリア作成
2	エリア削除
4	エリアのアクティブ化 (クリックされた、または前面に押し出された)
8	エリア非アクティブ化 (エリアはすでにアクティブではない)
16	オブジェクト生成 (作成、貼り付け、複製)
32	オブジェクト削除 (削除、切り取り、消去)
64	コマンド (Ctrl) - クリック (オブジェクトに対するものである必要はない)
128	オブジェクトが移動された (微調整、整列、移動など)
256	オブジェクトのサイズが変更された (矢印キー、ドラッグなど)
1024	選択済みオブジェクトでの変更
2048	ダブルクリック
4096	オブジェクトが作り直された

<エリア>に対して-1を指定した場合、イベントフィルタはフォーム上および外部ウィンドウ内に新しく作成されるすべての4D Chartエリアに対するデフォルトフィルタになります。これによって、「ユーザ」モードの「プラグイン」メニューから作成されたエリアに対してトラップすることができます。また、すべてのエリアで一貫したイベント処理ができます。

次の例はチャートエリアのオブジェクトメソッドによってトラップされたデフォルトイベントのリストに対して「コマンド-クリック」および「ダブルクリック」を追加します。

CT EVENT FILTER (エリア ; 64+2048)

参照 : CT Last event、 CT ON EVENT

CT EXPERT COMMAND

CT EXPERT COMMAND (エリア ; コマンド ; ステータス)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
コマンド	倍長整数	コマンド番号
ステータス	整数	エキスパートモードのメニューコマンドの状態 -1=カレント値を返す 0=使用可、1=使用不可

CT EXPERT COMMANDコマンドは、4D Chartエキスパートモードに対するメニューコマンドの使用の可否を設定します。

引数<ステータス>が0の場合は、<コマンド>で指定されるメニューコマンドはエキスパートモードで使用可能になります。<ステータス>が0よりも大きな場合は、そのメニューコマンドは使用不可になります。<ステータス>が-1の変数である場合は、**CT EXPERT COMMAND**コマンドは、<ステータス>(0=使用可、1=使用不可)にメニューコマンドの現在のステータスを返します。

<コマンド>に指定できる値は、付録Bにあります。

メニューコマンドが**CT EXPERT COMMAND**コマンドで使用不可である場合でも、**CT DO COMMAND**コマンドでコールすることで実行できます。

次の例は、「データベース」メニューの「フィールド貼り付け...」メニューコマンドを使用不可にします。

CT EXPERT COMMAND (エリア ; 6001 ; 1)

CT EXPERT MODE (エリア ; 1)

参照 : CT EXPERT MODE

CT EXPERT MODE

CT EXPERT MODE (エリア ; モード)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
モード	整数	エキスパートモードを切り替える -1=カレント値を返す 0=オフ、1=オン

CT EXPERT MODEコマンドは、エキスパートモードをオンまたはオフにします。4D Chartがエキスパートモードのときには、4D Chartメニュー上の特定の項目が使用できなくなっている可能性があります。エキスパートモードは、**CT EXPERT COMMAND**コマンドで設定されます。

引数<モード>が1の場合は、エキスパートモードが起動されます。エキスパートモードが起動されると、以前に**CT EXPERT COMMAND**コマンドで指定されたメニューコマンドは使用不可になります。

<モード>が0の場合は、エキスパートモードはオフになります。

<モード>が-1の場合は、**CT EXPERT MODE**コマンドは<モード>にそのモード(0=オフ、1=オン)を返します。

次の例は、「ファイル」メニューの「フルウインドウ」メニューコマンドを使用不可にします。

CT EXPERT COMMAND (エリア ; 1012 ; 1)

CT EXPERT MODE (エリア ; 1)

参照 : **CT EXPERT COMMAND**

CT Last event

CT Last event (エリア) 倍長整数

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア

CT Last event関数は、引数 <エリア> で発生した最後のイベントのコードを返します。

CT Last event関数は、4D Chartエリアのオブジェクトメソッド、またはCT ON EVENTコマンドでインストールされたイベントプロシージャで使用できます。CT Last event関数はオブジェクトメソッドやメソッドを実行させたイベントを特定します。CT EVENT FILTERコマンドと共に使用した場合は、CT Last event関数でユーザのアクションに基づいたアクションを行うことができます。

下記にイベントコードを示します：

値	イベント
-1	すべてのイベント
0	イベントなし
1	エリア作成
2	エリア削除
4	エリアのアクティブ化 (クリックされた、または前面に押し出された)
8	エリア非アクティブ化 (エリアはすでにアクティブではない)
16	オブジェクト生成 (作成、貼り付け、複製)
32	オブジェクト削除 (削除、切り取り、消去)
64	コマンド (Ctrl) - クリック (オブジェクトに対するものである必要はない)
128	オブジェクトが移動された (微調整、整列、移動など)
256	オブジェクトのサイズが変更された (矢印キー、ドラッグなど)
1024	選択済みオブジェクトでの変更
2048	ダブルクリック
4096	オブジェクトが作り直された

次の例では、CT Last event関数はCT ON EVENTコマンドでインストールされたメソッドで使用され、ダブルクリックを特定します。ユーザがグラフをダブルクリックすると、カスタムダイアログボックスが表示されるのでグラフに対して変更を行えます。

```

If ( CT Last event (エリア)=2048) `ダブルクリックの場合
  If ( CT Get object type (エリア ; 0)=5) `グラフの場合
    `カスタム「チャート変更」ダイアログボックスをオープンする
    CHANGE CHART (エリア ; CT Get ID (エリア ; 0 ; 1))
  End if
End if

```

参照：CT EVENT FILTER、CT Last event

CT SET ENTERABLE

CT SET ENTERABLE (エリア ; モード)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
モード	整数	エキスパートモードを切り替える -1=カレント値を返す 0=オフ、1=オン

CT SET ENTERABLE コマンドは、引数<エリア>内の文書へのアクセスを制御します。

<モード>が1の場合は、<エリア>が使用可能になり、正常に作動します。<モード>が0の場合は、<エリア>は使用不可になります。

使用不可になったエリアに対しては、ユーザは変更はできませんが、言語で変更できません。エリアが使用不可になると、ユーザはそのエリアをスクロールして選択したオブジェクトをクリップボードにコピーできます。ユーザは選択したものを変更したり、4D Chartメニューバーやツールパレットを使用したりすることはできません。

次の例は、<エリア>を入力不可にするフォームメソッドです。

If (Before) `Beforeフェーズにいる場合

CT SET ENTERABLE (エリア ; 0) `エリアを入力不可にする

End if

参照 : CT SET DISPLAY

CT Get display

CT Get display (エリア ; 項目) 整数

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
項目	整数	情報を取得する項目

CT Get display関数を使用すると、4D Chartウィンドウの特定の機能が表示されるかどうかわかります。

CT Get display関数は、指定された項目が表示されないのであれば 0、表示されるのであれば 1を返します。

メニューバー、「チャートツール」パレット、「オブジェクトツール」パレット、スクロールバー、ルーラは**CT SET DISPLAY**コマンドを使用して「ユーザ」モードまたはメソッドによって隠したり、表示したりすることができます。

下記は、引数<項目>のコードです：

コード	項目
1	メニューバー
2	チャートツール
3	オブジェクトツール
6	スクロールバー
9	ルーラ

次の例は、メニューバーが使用不可かどうかを確認し、使用不可になっていないのであれば、使用不可にします。

```
If ( CT Get display (エリア ; 1)=1)
  CT SET DISPLAY (エリア ; 1 ; 0)
End if
```

参照：CT SET DISPLAY

CT SET DISPLAY

CT SET DISPLAY (エリア ; 項目 ; 表示コード)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
項目	整数	表示 / 非表示を行う項目 (下記の表参照)
表示コード	整数	項目を表示する? 0=隠す、1=表示、2=切り替え

CT SET DISPLAYコマンドは、指定した項目が4D Chartウィンドウで表示されるのが隠されるのを設定します。

メニューバー、「チャートツール」パレット、「オブジェクトツール」パレット、スクロールバー、ルーラは**CT SET DISPLAY**コマンドを使用して隠したり、表示したりすることができます。

下記は、引数<項目>のコードです：

コード	項目
1	メニューバー
2	チャートツール
3	オブジェクトツール
6	スクロールバー
9	ルーラ

次の例は、4D Chartメニューバー、「チャートツール」パレット、「オブジェクトツール」パレット、ルーラを隠します。

CT SET DISPLAY(エリア ; 1 ; 0)

CT SET DISPLAY(エリア ; 2 ; 0)

CT SET DISPLAY(エリア ; 3 ; 0)

CT SET DISPLAY(エリア ; 9 ; 0)

参照 : CT Get display

CT GET DOCUMENT SIZE

CT GET DOCUMENT SIZE (エリア ; 幅 ; 高さ)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
幅	実数	文書の幅を受け取る(ポイント単位)
高さ	実数	文書の高さを受け取る(ポイント単位)

CT GET DOCUMENT SIZEコマンドは文書エリアのサイズを返します。4D Chart文書は最大3500 × 3500ポイントまでを測定できます。

引数 <幅> は、ポイント単位の文書エリアの幅です。

引数 <高さ> は、ポイント単位の文書エリアの長さです。

次の例は、**CT GET DOCUMENT SIZE**コマンドを使用して現在の文書を変更する前にそのサイズを取得します。

```

CT GET DOCUMENT SIZE (エリア ; $Width ; $Height)
If ($Width < 2208)
    CT SET DOCUMENT SIZE (エリア ; 2208 ; 730)
End if
    
```

参照 : CT SET DOCUMENT SIZE

CT SET DOCUMENT SIZE

CT SET DOCUMENT SIZE (エリア ; 幅 ; 高さ)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
幅	実数	文書の幅を受け取る(ポイント単位) -1=変更なし
高さ	実数	文書の高さを受け取る(ポイント単位) -1=変更なし

CT SET DOCUMENT SIZEコマンドは、文書エリアのサイズを設定します。4D Chart文書は、最大3500 × 3500ポイントまで測定できます。

引数 <幅> は、ポイント単位の文書エリアの幅です。

引数 <高さ> は、ポイント単位の文書エリアの長さです。

次の例は、「On Startup」データベースメソッドに配置できるものであり、すべての新規文書用のデフォルト文書サイズを2208 × 1460ポイントに設定します。

CT SET DOCUMENT SIZE (-1 ; 2208 ; 1460)

参照 : CT GET DOCUMENT SIZE

CT GET PROPERTIES

CT GET PROPERTIES (エリア ; プリント順 ; 変更メッセージ ; ホットリンクタイプ ; 保存メッセージ)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
プリント順	整数	プリント順を受け取る 0=行順、1=列順
変更メッセージ	整数	グラフタイプ変更メッセージを受け取る 0=メッセージなし、1=メッセージあり
ホットリンクタイプ	整数	ホットリンク用のデフォルトグラフタイプを受け取る
保存メッセージ	整数	保存メッセージを受け取る 0=メッセージなし、1=メッセージあり

CT GET PREFERENCES コマンドは、指定された4D Chartエリアに設定されているプロパティ情報を取得します。

引数<プリント順>は、文書のどのページが印刷されるのかの順序です。プリント順は、文書が印刷される順序だけに影響し、ページの方向には影響しません。

引数<変更メッセージ>は、グラフのタイプを変更しようとしたときに「アラート」ボックスが表示されるかどうかを指定します。ユーザは、キャンセルするか変更を続行するかのオプションを選択できます。

引数<ホットリンクタイプ>は、ホットリンクに格納されたデータから生成されるグラフのデフォルトタイプです。この<ホットリンクタイプ>のコードを下記に示します：

コード	チャートタイプ
1	2D 面
2	2D 棒
3	2D ピクチャ
4	2D 線
5	2D 散布図
6	2D 円
7	2D ポーラー
100	3D 棒
101	3D 線
102	3D 面
103	3D 等高線
104	3D 三角形
105	3D ピン

引数<保存メッセージ>は、変更を保存していない4D Chart文書をクローズするときに「アラート」ボックスをユーザに表示するかどうかを指定します。

<保存メッセージ>が1の場合は、4D Chartは変更を保存していない4D Chart文書をユーザがクローズすると通常の「アラート」ボックスを表示します。「アラート」ボックスには、変更を保存する、変更を保存しない、クローズせずに文書に戻るというオプションがあります。

<保存メッセージ>が0の場合は、4D Chartは変更を保存しないか、またはユーザに「アラート」ボックスを表示しません。変更を保存するかどうかは設定者の責任になります。例外は、ピクチャフィールドに保存される途中のフォーム上にある4D Chartエリアです。これらのエリアの内容はピクチャフィールドに自動的に保存されません。

次の例は、\$POrder、\$CAAlert、\$HType、\$SAAlertの変数にプリント順、グラフタイプ変更メッセージ、ホットリンクチャートタイプといったエリアのプロパティを返す例です。

CT GET PROPERTIES (エリア ; \$POrder ; \$CAAlert ; \$HType ; \$SAAlert)

参照 ; CT SET PROPERTIES、CT SAVE DOCUMENT

CT SET PROPERTIES

CT SET PROPERTIES (エリア ; プリント順 ; 変更メッセージ ; ホットリンクタイプ ; 保存メッセージ)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
プリント順	整数	プリント順を受け取る 0=行順、1=列順
変更メッセージ	整数	グラフタイプ変更メッセージを受け取る 0=メッセージなし、1=メッセージあり
ホットリンクタイプ	整数	ホットリンク用のデフォルトグラフタイプを受け取る
保存メッセージ	整数	保存メッセージを受け取る 0=メッセージなし、1=メッセージあり

CT SET PREFERENCES コマンドは、指定された4D Chartエリアに設定されているプロパティ情報を取得します。

引数<プリント順>は、文書のどのページが印刷されるのかの順序です。プリント順は、文書が印刷される順序だけに影響し、ページの方向には影響しません。

引数<変更メッセージ>は、グラフのタイプを変更しようとしたときに「アラート」ボックスが表示されるかどうかを指定します。ユーザは、キャンセルするか変更を続行するかかのオプションを選択できます。

引数<ホットリンクタイプ>は、ホットリンクに格納されたデータから生成されるグラフのデフォルトタイプです。この<ホットリンクタイプ>のコードについて、**CT GET PROPERTIES**コマンドの節を参照してください。

引数<保存メッセージ>は、変更を保存していない4D Chart文書をクローズするときに「アラート」ボックスをユーザに表示するかどうかを指定します。

<保存メッセージ>が1の場合は、4D Chartは変更を保存していない4D Chart文書をユーザがクローズすると通常の「アラート」ボックスを表示します。「アラート」ボックスには、変更を保存する、変更を保存しない、クローズせずに文書に戻るというオプションがあります。

<保存メッセージ>が0の場合は、4D Chartは変更を保存しないか、またはユーザに「アラート」ボックスを表示しません。変更を保存するかどうかは設定者の責任になります。例外は、ピクチャフィールドに保存される途中のフォーム上にある4D Chartエリアです。これらのエリアの内容はピクチャフィールドに自動的に保存されます。

次の例は、他のプロパティは変更せずに、デフォルトホットリンクチャートタイプを3D 等高線グラフに設定します。

CT SET PROPERTIES (エリア ; -1 ; -1 ; 103 ; -1)

参照 ; CT GET PROPERTIES、CT SAVE DOCUMENT

次のコマンドと関数は、「ルーチン」エリアの「CHART」テーマ内にあります：

CT Chart arrays	CT Get legend text
CT Chart selection	CT SET LEGEND TEXT
CT Chart data	CT GET REAL SCALE
CT GET CHART FILL ATTRIBUTES	CT SET REAL SCALE
CT SET CHART FILL ATTRIBUTES	CT GET DATE SCALE
CT GET CHART TEXT ATTRIBUTES	CT SET DATE SCALE
CT SET CHART TEXT ATTRIBUTES	CT GET DEPTH
CT GET CHART LINE ATTRIBUTES	CT SET DEPTH
CT SET CHART LINE ATTRIBUTES	CT GET 3D VIEW
CT SHOW GRID LINES	CT SET 3D VIEW
CT GET AXIS ATTRIBUTES	CT GET CHART PART
CT SET AXIS ATTRIBUTES	CT GET CHART OPTIONS
CT GET LABEL ATTRIBUTES	CT SET CHART OPTIONS
CT SET LABEL ATTRIBUTES	CT Get chart type
CT GET TITLE ATTRIBUTES	CT SET CHART TYPE
CT SET TITLE ATTRIBUTES	CT UPDATE CHART
CT GET LEGEND ATTRIBUTES	CT Get chart picture
CT SET LEGEND ATTRIBUTES	CT SET CHART PICTURE
CT EXPLODE PIE	

このテーマのコマンドと関数によって、4D Chart文書にグラフを作成し、そのグラフを操作することができます。

この章で最初に説明する3つの関数を使用すると、データベース内のフィールドに格納されたデータ、または配列に格納されたデータのいずれかから2次元グラフと3次元のグラフを作成できます。

このテーマにあるコマンドに関する詳細例については、第2章の「プログラムでグラフを作成する」を参照してください。

プログラムでグラフを作成する

この節の3つの関数を使用すると、プログラムでグラフを作成することができます。

注：また、ホットリンクからもグラフをプログラムで作成できます。詳細については第6章の「ホットリンクコマンド」の**CT Subscribe**関数の説明を参照してください。

次の表は、項目、系列、数値が各関数に渡されると、どのようにデータがグラフ化されるのかを説明しています。

	CT Chart arrays	CT Chart selection	CT Chart data
項目	項目名の配列	指定されたフィールドのレコード	指定されたフィールドのレコード
系列	系列名の配列	指定されたフィールドの名前	指定されたフィールドの名前
数値	数値の配列	指定されたフィールドのレコード	指定されたフィールドのレコード

CT Chart selection関数と**CT Chart data**関数では、系列が指定される方法が違います。

CT Chart selection関数では、系列はフィールド名です。それらのフィールドのデータは、数値軸にプロットされます。

CT Chart data関数では、系列軸用に1つのフィールドを指定します。系列名はフィールドに格納されたデータから取得されます。次にプロットされる値を含む別のフィールドを指定します。

CT Chart selection関数を使用した例は、第2章の「フィールドで定義された系列を使ってグラフを作成する」の節を参照してください。**CT Chart data**関数を使用する例については、第2章の「レコードのデータで定義された系列を使ってグラフを作成する」の2節を参照してください。

付録Eには、各軸上にグラフ化できるフィールドタイプのリストがあります。

ユーザは、各関数の引数<タイプ>に指定するグラフタイプによってグラフを2次元にするのか、3次元にするのかを指定します。

CT Chart array

CT Chart array (エリア; タイプ; サイズ; 項目配列; 系列配列; 数値配列) 数値

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
タイプ	整数	グラフタイプ (下記の表参照)
サイズ	整数	グラフの初期サイズオプション 1=可変、2=自動-ウインドウ、3=自動-文書
項目配列	配列	項目配列
系列配列	配列	系列配列
数値配列	配列	数値配列

CT Chart arrays関数は、指定された配列に基づいてグラフを作成し、そのグラフのオブジェクトIDを返します。このコマンドは2次元グラフまたは3次元グラフのどちらを作成するときにも使用できます。

次の表は、引数<タイプ>のコードを示しています：

コード	チャートタイプ
1	2D 面
2	2D 棒
3	2D ピクチャ
4	2D 線
5	2D 散布図
6	2D 円
7	2D ポーラー
100	3D 棒
101	3D 線
102	3D 面
103	3D 等高線
104	3D 三角形
105	3D ピン

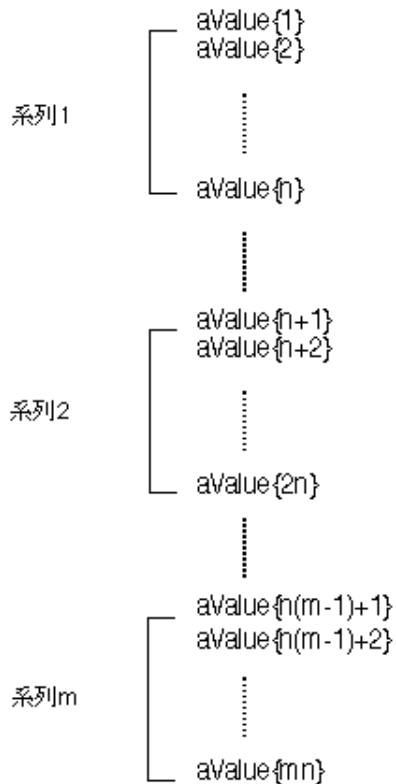
引数<サイズ>は、グラフが生成されたときにそのグラフがどの程度のスペースを取るのか、ウインドウのサイズ変更をしたときにグラフのサイズはどのように変更するのかを決定します。各オプションの詳細については、『4th Dimension / 4D First ユーザリファレンス』を参照してください。

<項目配列>には、X軸項目があります。

<系列配列>には、系列が含まれます。2次元グラフでは、系列は項目軸上に表示されま
す。3次元グラフでは、系列は数値軸上に表示されます。

<数値配列>は、数値軸上でグラフ化されるすべての値を含む1次元の配列です。<項目
配列>と<系列配列>の各要素の値が存在するように<数値配列>には値が入っている
必要があります。つまり、n項目とm系列がある場合には、<数値配列>にはn*m個の要
素があることになります。

次の図は、<数値配列>に入れられるものの順序を示しています。nは項目の合計数を表
します。mは系列の合計数を表します。



例えば、次のデータを参照してください。

学校名 (項目)	年度 (系列)	生徒数 (数値)
千代田中学	1993	1000
千代田中学	1995	1250
千代田中学	1997	800
東京第1中学	1993	600
東京第1中学	1995	975
東京第1中学	1997	1100

```
aValues
aValues{1}:=1000
aValues{2}:=600
aValues{3}:=1250
aValues{4}:=975
aValues{5}:=800
aValues{6}:=1100
```

CT Chart arrays関数の使用例については、第2章の「プログラムでグラフを作成する」を参照してください。

参照 : CT Chart selection、CT Chart data

CT Chart selection

CT Chart selection (エリア; タイプ; サイズ ; グループ項目 ; テーブル番号 ; 項目フィールド ; 系列 / 数値フィールド) 倍長整数

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
タイプ	整数	グラフタイプ (下記の表参照)
サイズ	整数	グラフの初期サイズオプション 1=可変、2=自動-ウインドウ、3=自動-文書
グループ項目	整数	項目データをグループ化する? 0=なし、1=グループ化
テーブル番号	整数	グラフのテーブル番号
項目フィールド	整数	項目軸上にプロットするフィールドの番号
系列 / 数値フィールド	整数配列	フィールド番号の配列

CT Chart selection関数は、引数 <テーブル番号> のカレントセレクションのグラフを作成します。この関数は、チャートのオブジェクトIDを返します。

次の表は、引数<タイプ>のコードを示しています：

コード	チャートタイプ
1	2D 面
2	2D 棒
3	2D ピクチャ
4	2D 線
5	2D 散布図
6	2D 円
7	2D ポーラー
100	3D 棒
101	3D 線
102	3D 面
103	3D 等高線
104	3D 三角形
105	3D ピン

引数<サイズ>は、グラフが生成されたときにそのグラフがどの程度のスペースを取るのか、ウィンドウのサイズ変更をしたときにグラフのサイズはどのように変更するのかを決定します。各オプションの詳細については、『4th Dimension / 4D First ユーザリファレンス』を参照してください。

引数<グループ項目>は、項目軸のデータがグループ化されるのかどうかを指定します。<グループ項目>が1の場合には、各項目は一意になり、重複した項目の値があればまとめられます。<グループ項目>が0の場合には、各項目の値は別々にグラフ化されません。

注：系列をグループ化する必要はありません。これは、系列はフィールド名なので一意だからです。

引数<テーブル番号>は、データをグラフ化するテーブルの番号です。テーブルへのポイントをTable関数の引数として渡すことによって、テーブルの番号を検索することができます。

<項目フィールド>は、グラフ化する項目軸上のフィールドの番号です。フィールドへのポイントをField関数の引数として渡すことによってフィールドの番号を検索することができます。

<系列 / 数値フィールド>は系列と数値としてグラフ化するフィールドのフィールド番号配列です。フィールド名は系列になります。フィールドに格納された値は数値軸上でグラフ化されます。

2次元グラフでは、系列は項目軸上に表示されます。3次元グラフでは、系列は数値軸上に表示されます。

CT Chart selection関数の使用例は、第2章の「プログラムでグラフを作成する」を参照してください。

参照 ; CT Chart arrays、CT Chart data

CT Chart data

CT Chart data (エリア; タイプ; サイズ; グループ項目; グループ系列; テーブル番号; 項目フィールド; 系列フィールド; 数値フィールド) 倍長整数

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
タイプ	整数	グラフタイプ (下記の表参照)
サイズ	整数	グラフの初期サイズオプション 1=可変、2=自動-ウインドウ、3=自動-文書
グループ項目	整数	項目データをグループ化する? 0=なし、1=グループ化
グループ系列	整数	系列データをグループ化する? 0=なし、1=グループ化
テーブル番号	整数	データをグラフ化するテーブルの番号
項目フィールド	整数	項目軸上にプロットするフィールドの番号
系列フィールド	整数	系列としてプロットするフィールドの番号
数値フィールド	整数	数値軸上にプロットするフィールドの番号

CT Chart data関数は、引数<テーブル番号>のカレントセレクションのグラフを作成します。この関数はチャートのオブジェクトIDを返します。

次の表は、引数<タイプ>のコードを示しています：

コード	チャートタイプ
1	2D 面
2	2D 棒
3	2D ピクチャ
4	2D 線
5	2D 散布図
6	2D 円
7	2D ポーラー
100	3D 棒
101	3D 線
102	3D 面
103	3D 等高線
104	3D 三角形
105	3D ピン

引数<サイズ>は、グラフが生成されたときにそのグラフがどの程度のスペースを取るのか、ウィンドウのサイズ変更をしたときにグラフのサイズはどのように変更するのかを決定します。各オプションの詳細については、『4th Dimension / 4D First ユーザリファレンス』を参照してください。

引数<グループ項目>は、項目軸のデータがグループ化されるのかどうかを指定します。<グループ項目>が1の場合には、各項目は一意になり、重複した項目の値があればまとめられます。<グループ項目>が0の場合には、各項目の値は別々にグラフ化されません。

引数<グループ系列>は、系列軸上のデータをグループ化するかどうかを指定します。<グループ系列>が1の場合には、各シリーズは一意になり、重複したシリーズの値があればまとめられます。<グループ系列>が0の場合には、各シリーズの値は別々にグラフ化されます。

引数<テーブル番号>は、データをグラフ化するテーブルの番号です。テーブルへのポインタをTable関数の引数として渡すことによって、テーブルの番号を検索することができます。

引数<系列フィールド>は、グラフ化する系列軸上のフィールド番号です。2次元グラフでは、系列は項目軸上に表示されます。3次元グラフでは、系列は系列軸上に表示されません。

引数<数値フィールド>は、グラフ化する数値軸上のフィールド番号です。

フィールドへのポインタを**Field**関数の引数として渡すと、フィールドの番号を検索することができます。

CT Chart data関数の使用例は、第2章の「プログラムでグラフを作成する」を参照してください。

参照 ; CT Chart arrays、CT Chart selection

グラフ属性を変更する

この章の後半で説明するコマンドと関数は、すでに作成されたグラフを変更するときに使用します。

これらのコマンドと関数の<オブジェクト>引数には、この章の前半で説明したチャート作成関数から戻される倍長整数が必要です。

CT GET CHART FILL ATTRIBUTES

CT GET CHART FILL ATTRIBUTES (エリア; オブジェクト; タイプ ; 特定部分 ; パターン ; カラー)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
タイプ	整数	属性を取得するオブジェクトのタイプ
特定部分	倍長整数	属性を取得するオブジェクトの特定の部分
パターン	整数	パターン番号(1から36)を受け取る
カラー	倍長整数	カラーの値を受け取る

CT GET CHART FILL ATTRIBUTESコマンドは、引数<エリア>、<オブジェクト>、<タイプ>、<特定部分>で指定されたチャートオブジェクトの塗りつぶし属性を取得します。

<タイプ>と<特定部分>は、属性を取得するグラフの部分を指定します。<タイプ>と<特定部分>のコードは、付録Dに掲載されています。

引数<パターン>は、パターンパレット上で利用可能なパターンの1つを指定する1から36までの整数です。<パターン>のコードは、付録Dに掲載されています。

引数<カラー>は、オブジェクトの色を指定する倍長整数です。**CT Index to color**関数や**CT RGB to color**関数を使用することで <カラー>の値を指定できます。これらの関数の詳細については、第9章の「ユーティリティコマンド」を参照してください。

次の例は、\$ChartIDに指定されたチャートの最初の系列の塗りつぶし属性を\$Pattern変数と\$Color変数に返します。

CT GET CHART FILL ATTRIBUTES (エリア; \$ChartID; 8; 100; \$Pattern; \$Color)

参照; CT SET CHART FILL ATTRIBUTES

注: 描画ツールや描画関数を使用して文書に追加されたオブジェクトの属性を取得するには、第7章で説明している「オブジェクトコマンド」を使用してください。

CT SET CHART FILL ATTRIBUTES

CT SET CHART FILL ATTRIBUTES (エリア; オブジェクト; タイプ; 特定部分; パターン; カラー)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
タイプ	整数	属性を取得するオブジェクトのタイプ
特定部分	倍長整数	属性を取得するオブジェクトの特定の部分
パターン	整数	パターン番号(1から36)を受け取る
カラー	倍長整数	カラーの値を受け取る

CT SET CHART FILL ATTRIBUTESコマンドは、引数<エリア>、<オブジェクト>、<タイプ>、<特定部分>で指定されたチャートオブジェクトの塗りつぶし属性を取得します。

<タイプ>と<特定部分>は、属性を取得するグラフの部分を指定します。<タイプ>と<特定部分>のコードは、付録Dに掲載されています。

引数<パターン>は、パターンパレット上で利用可能なパターンの1つを指定する1から36までの整数です。<パターン>のコードは、付録Dに掲載されています。

引数<カラー>は、オブジェクトの色を指定する倍長整数です。**CT Index to color**関数や**CT RGB to color**関数を使用することで<カラー>の値を指定できます。これらの関数の詳細については、第9章の「ユーティリティコマンド」を参照してください。

次の例は、\$ChartIDに指定されたチャートの最初の系列の塗りつぶし属性を変更します。色は赤に、パターンは塗りつぶしに設定されます。

CT SET CHART FILL ATTRIBUTES (エリア; \$ChartID; 8; 100; 3; **CT Index to color**(4))

参照; CT GET CHART FILL ATTRIBUTES

注: 描画ツールや描画関数を使用して文書に追加されたオブジェクトの属性を取得するには、第7章で説明している「オブジェクトコマンド」を使用してください。

CT GET CHART TEXT ATTRIBUTES

CT GET CHART TEXT ATTRIBUTES (エリア; オブジェクト; タイプ; 特定部分; フォントID; フォントサイズ; スタイル; カラー)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
タイプ	整数	属性を取得するオブジェクトのタイプ
特定部分	倍長整数	属性を取得するオブジェクトの特定の部分
フォントID	整数	フォントIDを受け取る
フォントサイズ	整数	フォントサイズを受け取る
スタイル	整数	フォントスタイルを受け取る
カラー	倍長整数	カラーの値を受け取る

CT GET CHART TEXT ATTRIBUTESコマンドは、引数<エリア>、<オブジェクト>、<タイプ>、<特定部分>で指定されたチャートテキストの属性を取得します。

<タイプ>と<特定部分>は、属性を取得するグラフの部分を指定します。<タイプ>と<特定部分>のコードは、付録Dに掲載されています。

引数<フォントID>は、システム内にあるフォントのIDです。フォントのID番号は**CT Font number**関数を使用することによって取得できます。これらの関数の詳細については、第9章の「ユーティリティコマンド」を参照してください。

引数<フォントサイズ>は、反転表示されたテキストやテキストオブジェクトのポイント単位のサイズです。

引数<スタイル>は、複数のスタイル番号の加算の結果を混合した番号です。次の表はスタイル番号を示しています：

値	スタイル
0	標準
1	太字 (ボールド)
2	斜体 (イタリック)
4	下線 (アンダーライン)
8	アウトライン
16	シャドウ

引数<カラー>は、オブジェクトの色を指定する倍長整数です。**CT Index to color**関数や**CT RGB to color**関数を使用することで<カラー>の値を指定できます。これらの関数の詳細については、第9章の「ユーティリティコマンド」を参照してください。

次の例は、項目軸のテキスト属性がカスタマイズされたかどうかを確認し、カスタマイズされている場合にはリセットします。

```
CT GET CHART TEXT ATTRIBUTES (エリア ; $ChartID ; 8 ; 100 ; $FontID ; $FontSize  
; $Style ; $Color)  
If (($FontSize#10) | ($FontID# CT Font number ("Geneva"))) |  
($Color# CT Index to color (10)))  
  CT SET CHART TEXT ATTRIBUTES (Area;$ChartID;5; 0; CT Font number("Geneva")  
; 10 ; 1 ; CT Index to color (10))  
End if
```

参照： CT SET CHART TEXT ATTRIBUTES

注：テキストツールや**CT Draw text**関数を使用して文書に追加されたオブジェクトの属性を取得するには、第7章で説明している「オブジェクトコマンド」を使用してください。

CT SET CHART TEXT ATTRIBUTES

CT SET CHART TEXT ATTRIBUTES (エリア; オブジェクト; タイプ; 特定部分; フォントID; フォントサイズ; スタイル; カラー)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
タイプ	整数	属性を取得するオブジェクトのタイプ
特定部分	倍長整数	属性を取得するオブジェクトの特定の部分
フォントID	整数	フォントID、-1=変更なし
フォントサイズ	整数	フォントサイズ、-1=変更なし
スタイル	整数	フォントスタイル、-1=変更なし
カラー	倍長整数	カラーの値、-1=変更なし

CT SET CHART TEXT ATTRIBUTES コマンドは、引数 <エリア>、<オブジェクト>、<タイプ>、<特定部分> で指定されたチャートテキストの属性を取得します。

<タイプ> と <特定部分> は、属性を取得するグラフの部分指定します。<タイプ> と <特定部分> のコードは、付録Dに掲載されています。

引数 <フォントID> は、システム内にあるフォントのIDです。フォントのID番号は **CT Font number** 関数を使用することによって取得できます。これらの関数の詳細については、第9章の「ユーティリティコマンド」を参照してください。

引数 <フォントサイズ> は、反転表示されたテキストやテキストオブジェクトのポイント単位のサイズです。

引数 <スタイル> は、複数のスタイル番号の加算の結果を混合した番号です。次の表はスタイル番号を示しています：

値	スタイル
0	標準
1	太字 (ボールド)
2	斜体 (イタリック)
4	下線 (アンダーライン)
8	アウトライン
16	シャドウ

引数 <カラー> は、オブジェクトの色を指定する倍長整数です。 **CT Index to color** 関数や **CT RGB to color** 関数を使用することで <カラー> の値を指定できます。これらの関数の詳細については、第9章の「ユーティリティコマンド」を参照してください。

CT GET CHART TEXT ATTRIBUTES コマンドの例を参照してください。

参照：CT GET CHART TEXT ATTRIBUTES

注：テキストツールや**CT Draw text**関数を使用して文書に追加されたオブジェクトの属性を取得するには、第7章で説明している「オブジェクトコマンド」を使用してください。

CT GET CHART LINE ATTRIBUTES

CT GET CHART LINE ATTRIBUTES (エリア; オブジェクト; タイプ; 特定部分; パターン; カラー; 線幅)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
タイプ	整数	属性を取得するオブジェクトのタイプ
特定部分	倍長整数	属性を取得するオブジェクトの特定の部分
パターン	整数	パターン番号(1から36)を受け取る
カラー	倍長整数	カラーの値 (0以上) を受け取る
線幅	実数	ポイント単位で線の太さを受け取る(0以上)

CT GET CHART LINE ATTRIBUTESコマンドは、引数<パターン>、<カラー>、<線幅>で指定された線の属性を返します。

属性を検索する対象の線は、引数<エリア>、<オブジェクト>、<タイプ>、<特定部分>によって指定されます。

<タイプ>と<特定部分>は、属性を取得するグラフの部分を指定します。<タイプ>と<特定部分>のコードは、付録Dに掲載されています。

引数<パターン>は、パターンパレット上で利用可能なパターンの1つを指定する1から36までの整数です。<パターン>のコードは、付録Dに掲載されています。

引数<カラー>は、オブジェクトの色を指定する倍長整数です。**CT Index to color**関数や**CT RGB to color**関数を使用することで <カラー>の値を指定できます。これらの関数の詳細については、第9章の「ユーティリティコマンド」を参照してください。

引数<線幅>は、ポイント単位で計測される線の太さです。

注：線ツールや**CT Draw line**関数を使用して文書に追加された線の属性を取得するには、第7章で説明している「オブジェクトコマンド」を使用してください

次の例は、\$ChartID に指定されたチャートの矩形の線属性のプロットを\$Pattern、\$Color、\$Line変数に返します。

CT GET CHART LINE ATTRIBUTES (エリア; \$ChartID ; 1 ; 0 ; \$Pattern ; \$Color ; \$Line)

参照：CT SET CHART LINE ATTRIBUTES

CT SET CHART LINE ATTRIBUTES

CT SET CHART LINE ATTRIBUTES (エリア; オブジェクト; タイプ; 特定部分; パターン; カラー; 線幅)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
タイプ	整数	属性を取得するオブジェクトのタイプ
特定部分	倍長整数	属性を取得するオブジェクトの特定の部分
パターン	整数	パターン番号(1から36)、-1=変更なし
カラー	倍長整数	カラーの値(0以上)、-1=変更なし
線幅	実数	ポイント単位で線の太さ(0以上)、-1=変更なし

CT SET CHART LINE ATTRIBUTESコマンドは、引数<パターン>、<カラー>、<線幅>で指定された線の属性を設定します。

属性を検索する対象の線は、引数<エリア>、<オブジェクト>、<タイプ>、<特定部分>によって指定されます。

<タイプ>と<特定部分>は、属性を取得するグラフの部分を指定します。<タイプ>と<特定部分>のコードは、付録Dに掲載されています。

引数<パターン>は、パターンパレット上で利用可能なパターンの1つを指定する1から36までの整数です。<パターン>のコードは、付録Dに掲載されています。

引数<カラー>は、オブジェクトの色を指定する倍長整数です。**CT Index to color**関数や**CT RGB to color**関数を使用することで <カラー>の値を指定できます。これらの関数の詳細については、第9章の「ユーティリティコマンド」を参照してください。

引数<線幅>は、ポイント単位で計測される線の太さです。

注：線ツールや**CT Draw line**関数を使用して文書に追加された線の属性を取得するには、第7章で説明している「オブジェクトコマンド」を使用してください

次の例は、\$ChartIDに指定されたチャートに対して、矩形の線属性のプロットを変更します。色は緑に変更され、線の幅は3ポイントに、パターンは塗りつぶしに設定されます。

CT SET CHART LINE ATTRIBUTES (エリア; \$ChartID; 1; 0; 3;**CT Index to color**(10); 3)

参照：CT GET CHART LINE ATTRIBUTES

CT SHOW GRID LINES

CT SHOW GRID LINES (エリア; オブジェクト; 軸; グリッド; 表示)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
軸	整数	目盛線の表示 / 非表示に対する軸 0=項目、1=系列、2=数値
グリッド	整数	表示 / 非表示対象のグリッド線、0=補助、1=主
表示	整数	グリッド線の非表示 / 表示、0=非表示、1=表示

CT SHOW GRID LINESコマンドは、引数<エリア>、<オブジェクト>、<軸>で指定される軸に対する主グリッド線また補助グリッド線あるいはその両方を表示または非表示にします。

引数<グリッド>によって、コマンドによってどちらのグリッド線が影響を受けるのかを指定できます。主グリッド線は主増分の間隔になり、補助グリッド線は補助増分の間隔になります。

引数<表示>によって、指定したグリッド線を見えるようにするかどうかを指定できません。

次の例は、\$ChartIDに指定されたチャート用のValues軸の補助グリッド線を表示します。

CT SHOW GRID LINES (エリア ; \$ChartID ; 2 ; 0 ; 1)

参照 : CT GET CHART LINE ATTRIBUTES、CT SET CHART LINE ATTRIBUTES

CT GET AXIS ATTRIBUTES

CT GET AXIS ATTRIBUTES (エリア; オブジェクト; 軸; 補助目盛; 主目盛; 位置; 反転)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
軸	整数	グラフ軸 0=項目、1=系列、2=数値
補助目盛	整数	補助目盛りマークの種類を受け取る 0=なし、1=内側、2=外側、3=交差
主目盛	整数	目盛線の表示 / 非表示に対する軸 0=なし、1=内側、2=外側、3=交差
位置	実数	軸の位置を受け取る(軸が配置される位置)
反転	整数	逆順を受け取る 0=反転なし、1=反転

CT GET AXIS ATTRIBUTESコマンドは、引数<補助目盛>、<主目盛>、<位置>、<反転>の変数に<エリア>、<オブジェクト>、<軸>で指定されたグラフ軸の属性を返します。このコマンドは、2次元グラフだけに適用されます。

<補助目盛>と<主目盛>とは、<エリア>上の目盛りのことです。目盛りオプションは、各軸用の「軸」ダイアログボックスに設定するか、**CT SET AXIS ATTRIBUTES**コマンドを使用して設定できます。

<位置>とは、軸が別の軸と交差するときの数値のことです。<軸>が横軸の場合には、<位置>は縦軸の一番下からの増分数になります。<軸>が縦軸の場合には、<位置>は横軸の左側からの増分数になります。

引数<反転>が1の場合には、軸上にグラフ化される項目の順序は逆になります。<反転>が0の場合には、項目は本来の順序のままになります。

次の例は、\$ChartIDに指定されたチャートの項目軸属性を\$MajorTick、\$MinorTick、\$Location、\$Reverse変数に返します。

```
$ChartID:= CT Get ID (エリア ; 0 ; 1)
CT GET AXIS ATTRIBUTES (エリア ; $ChartID ; 0 ; $MinorTick ; $MajorTick ; $Location ; $Reverse)
```

参照 : CT SET AXIS ATTRIBUTES

CT SET AXIS ATTRIBUTES

CT SET AXIS ATTRIBUTES (エリア; オブジェクト; 軸; 補助目盛; 主目盛; 位置; 反転)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
軸	整数	グラフ軸 0=項目、1=系列、2=数値
補助目盛	整数	補助目盛りマークの種類を受け取る 0=なし、1=内側、2=外側、3=交差
主目盛	整数	目盛線の表示 / 非表示に対する軸 0=なし、1=内側、2=外側、3=交差
位置	実数	軸の位置を受け取る(軸が配置される位置)
反転	整数	逆順を受け取る 0=反転なし、1=反転

CT SET AXIS ATTRIBUTES コマンドは、引数 <補助目盛>、<主目盛>、<位置>、<反転> の変数に <エリア>、<オブジェクト>、<軸> で指定されたグラフ軸の属性を設定します。このコマンドは、2次元グラフだけに適用されます。

<補助目盛> と <主目盛> とは、<エリア> 上の目盛りのことです。

<位置> とは、軸が別の軸と交差するときの数値のことです。<軸> が横軸の場合には、<位置> は縦軸の一番下からの増分数になります。<軸> が縦軸の場合には、<位置> は横軸の左側からの増分数になります。

引数 <反転> が1の場合には、軸上にグラフ化される項目の順序は逆になります。<反転> が0の場合には、項目は本来の順序のままになります。

次の例は、\$ChartIDに指定されたチャートの項目軸属性を変更します。

```
$ChartID:= CT Get ID (エリア ; 0 ; 1)  
CT SET AXIS ATTRIBUTES (エリア ; $ChartID ; 0 ; 0 ; 3 ; 160 ; 1)
```

参照 : CT GET AXIS ATTRIBUTES

CT GET LABEL ATTRIBUTES

CT GET LABEL ATTRIBUTES (エリア; オブジェクト; 軸 ; 位置 ; 方向 ; フォーマット)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
軸	整数	属性を受け取るための軸 0=項目、1=系列、2=数値
位置	整数	ラベルの位置を受け取る 0=なし、1=上、2=左、3=下、4=右
方向	整数	ラベルの方向を受け取る 0=通常、1=垂直、2=右回転、3=左回転 4=互い違い、5=ワードラップ
フォーマット	文字列	ラベルフォーマットを受け取る

CT GET LABEL ATTRIBUTES コマンドは、引数<エリア>、<オブジェクト>、<軸>で指定された軸ラベルの属性を<位置>、<方向>、<フォーマット>変数に返します。

引数<位置>は、グラフに相対的な軸のラベル位置です。

引数<方向>は、各ラベルの方向です。各方向オプションを示す表については、『4th Dimension / 4D First ユーザリファレンス』を参照してください。

引数<フォーマット>は、ラベルテキストの表示フォーマットです。表示フォーマットが「通常」のときには<フォーマット>変数に空の文字列 " " が返ります。表示フォーマットで使用される特殊文字に関する詳細は、『4th Dimension / 4D First デザインリファレンス』を参照してください。

次の例は、\$ChartIDに指定されたチャートの項目軸ラベル属性を\$Position、\$Orient、\$Format変数に返します。

CT GET LABEL ATTRIBUTES (エリア ; \$ChartID ; 0 ; \$Position ; \$Orient ; \$Format)

参照 : CT SET LABEL ATTRIBUTES、CT GET CHART TEXT ATTRIBUTES

注 : 軸ラベルのテキスト属性を取得するには、**CT GET CHART TEXT ATTRIBUTES** コマンドを使用してください。

CT SET LABEL ATTRIBUTES

CT SET LABEL ATTRIBUTES (エリア; オブジェクト; 軸; 位置; 方向; フォーマット)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
軸	整数	属性を受け取るための軸 0=項目、1=系列、2=数値
位置	整数	ラベルの位置を受け取る 0=なし、1=上、2=左、3=下、4=右
方向	整数	ラベルの方向を受け取る 0=通常、1=垂直、2=右回転、3=左回転 4=互い違い、5=ワードラップ
フォーマット	文字列	ラベルフォーマットを受け取る

CT SET LABEL ATTRIBUTES コマンドは、引数<エリア>、<オブジェクト>、<軸>で指定された軸ラベルの属性を<位置>、<方向>、<フォーマット>変数に設定します。

引数<位置>は、グラフに相対的な軸のラベル位置です。

引数<方向>は、各ラベルの方向です。各方向オプションを示す表については、『4th Dimension / 4D First ユーザリファレンス』を参照してください。

引数<フォーマット>は、ラベルテキストの表示フォーマットです。表示フォーマットが「通常」のときには<フォーマット>変数に空の文字列 " " が返ります。表示フォーマットで使用される特殊文字に関する詳細は、『4th Dimension / 4D First デザインリファレンス』を参照してください。

次の例は、\$ChartIDに指定されるチャートの項目軸ラベル属性を変更します。ラベル位置は下、フォーマットは通常、方向は右回転に設定されます。

CT SET LABEL ATTRIBUTES (エリア; \$ChartID; 0; 3; 2; "")

参照: CT GET LABEL ATTRIBUTES、CT SET CHART TEXT ATTRIBUTES

注: 軸ラベルのテキスト属性を取得するには、**CT SET CHART TEXT ATTRIBUTES** コマンドを使用してください。

CT GET TITLE ATTRIBUTES

CT GET TITLE ATTRIBUTES (エリア; オブジェクト; 軸; 位置; 方向; タイトル)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
軸	整数	属性を受け取るための軸 0=項目、1=系列、2=数値
位置	整数	タイトルの位置を受け取る 0=なし、1=上、2=左、3=下、4=右
方向	整数	タイトルの方向を受け取る 0=通常、1=垂直、2=右回転、3=左回転 4=互い違い、5=ワードラップ
タイトル	文字列	タイトルのテキストを受け取る

CT GET TITLE ATTRIBUTESコマンドは、引数<エリア>、<オブジェクト>、<軸>で指定された軸タイトルの位置、方向、テキストを返します。

引数<位置>は、グラフに相対的なタイトルの位置です。

引数<方向>は、タイトルの方向です。

引数<タイトル>は、タイトルのテキストです。タイトルの長さは255文字以内です。

次の例は、\$Position、\$Orient、\$Title変数に項目軸タイトル属性を返します。

CT GET TITLE ATTRIBUTES (エリア ; \$ChartID ; 0 ; \$Position ; \$Orient ; \$Title)

参照 : CT SET TITLE ATTRIBUTES、CT GET CHART TEXT ATTRIBUTES

注 : タイトルのテキスト属性を取得するには、**CT GET CHART TEXT ATTRIBUTES**コマンドを使用してください。

CT SET TITLE ATTRIBUTES

CT SET TITLE ATTRIBUTES (エリア; オブジェクト; 軸; 位置; 方向; タイトル)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
軸	整数	属性を受け取るための軸 0=項目、1=系列、2=数値
位置	整数	タイトルの位置を受け取る 0=なし、1=上、2=左、3=下、4=右
方向	整数	タイトルの方向を受け取る 0=通常、1=垂直、2=右回転、3=左回転 4=互い違い、5=ワードラップ
タイトル	文字列	タイトルのテキストを受け取る

CT SET TITLE ATTRIBUTES コマンドは、引数 <エリア>、<オブジェクト>、<軸> で指定された軸タイトルの位置、方向、テキストを設定します。

引数 <位置> は、グラフに相対的なタイトルの位置です。

引数 <方向> は、タイトルの方向です。

引数 <タイトル> は、タイトルのテキストです。タイトルの長さは255文字以内です。

次の例は、数値軸タイトルのテキスト、位置、方向を設定します。

CT SET TITLE ATTRIBUTES (エリア; \$ChartID; 2; 2; 3; "総雨量")

参照: CT GET TITLE ATTRIBUTES、CT SET CHART TEXT ATTRIBUTES

注: タイトルのテキスト属性を取得するには、**CT SET CHART TEXT ATTRIBUTES** コマンドを使用してください。

CT GET LEGEND ATTRIBUTES

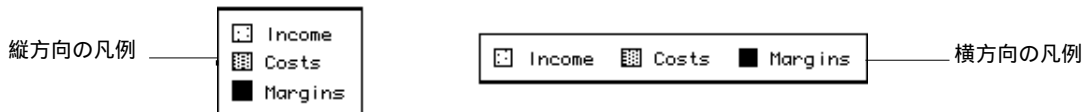
CT GET LEGEND ATTRIBUTES (エリア; オブジェクト; 表示; 方向; 逆順; 反転; 位置; 水平オフセット; 垂直オフセット)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
表示	整数	凡例を表示するか? 0=表示しない、1=表示する
方向	整数	凡例内の系列の方向を受け取る 0=水平、1=垂直
逆順	整数	順番を逆にするか? 0=逆順にしない、1=逆順にする
反転	整数	キーとテキストは逆になっているか? 0=反転しない、1=反転する
位置	整数	位置コードを受け取る
水平オフセット	整数	プロットの左側からの水平方向のオフセットを ポイント単位で受け取る
垂直オフセット	整数	プロットの上側からの垂直方向のオフセットを ポイント単位で受け取る

CT GET LEGEND ATTRIBUTESコマンドは、引数<表示>、<方向>、<逆順>、<反転>、<位置>、<水平オフセット>、<垂直オフセット>の変数に引数<エリア>、<オブジェクト>で指定された凡例の属性を返します。

引数<表示>は、凡例が表示されるかどうかを指定します。

引数<方向>は、系列が凡例内で縦方向に表示されるか、横方向に表示されるのかを指定します。次の図は、縦方向の凡例と横方向の凡例を示しています。



引数<逆順>は、凡例内の系列の順序が逆順になるかどうかを指定します。

引数<凡例>は、系列の一意なパターンと色を説明する系列ラベルとキーが逆になるかどうかを指定します。デフォルトでは、キーがラベルの左側に表示されます。

次の表は、引数 <位置> のコードを示しています：

コード	位置
0	凡例の位置は、組み込み位置オプションの1つではありません。
1	左上
2	左下
3	右上
4	右下
5	左
6	右
7	上
8	下

引数 <水平オフセット> と <垂直オフセット> は、<位置> が組み込み凡例位置にはないとき (<位置> = 0) に使用されます。<水平オフセット> は、グラフの左側から凡例の左側までをポイント単位で表します。<垂直オフセット> は、グラフの上の端から凡例の上の端までをポイント単位で表します。

次の例は、\$ChartIDに指定されたチャートの凡例テキスト属性を取得します。

CT GET LEGEND ATTRIBUTES (エリア ; \$ChartID ; \$Display ; \$Orient ; \$ReverseO ; \$ReverseK ; \$Location ; \$HorizOff ; \$VertOff)

参照：CT SET LEGEND ATTRIBUTES、CT Get legend text、
CT SET LEGEND TEXT、CT GET CHART TEXT ATTRIBUTES

注：凡例テキストのテキスト属性を取得するには、**CT GET CHART TEXT ATTRIBUTES** コマンドを使用してください。

CT SET LEGEND ATTRIBUTES

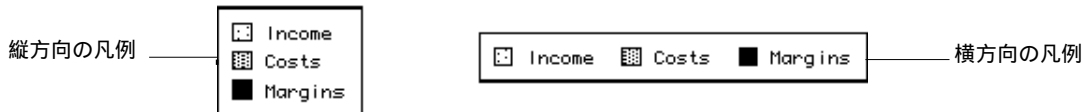
CT SET LEGEND ATTRIBUTES (エリア; オブジェクト; 表示; 方向; 逆順; 反転; 位置; 水平オフセット; 垂直オフセット)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
表示	整数	凡例を表示するか? 0=表示しない、1=表示する
方向	整数	凡例内の系列の方向を受け取る 0=水平、1=垂直
逆順	整数	順番を逆にするか? 0=逆順にしない、1=逆順にする
反転	整数	キーとテキストは逆になっているか? 0=反転しない、1=反転する
位置	整数	位置コードを受け取る
水平オフセット	整数	プロットの左側からの水平方向のオフセットを ポイント単位で受け取る
垂直オフセット	整数	プロットの上側からの垂直方向のオフセットを ポイント単位で受け取る

CT SET LEGEND ATTRIBUTESコマンドは、引数<表示>、<方向>、<逆順>、<反転>、<位置>、<水平オフセット>、<垂直オフセット>の変数に引数<エリア>、<オブジェクト>で指定された凡例の属性を設定します。

引数<表示>は、凡例が表示されるかどうかを指定します。

引数<方向>は、系列が凡例内で縦方向に表示されるか、横方向に表示されるのかを指定します。次の図は、縦方向の凡例と横方向の凡例を示しています。



引数<逆順>は、凡例内の系列の順序が逆順になるかどうかを指定します。

引数<凡例>は、系列の一意なパターンと色を説明する系列ラベルとキーが逆になるかどうかを指定します。デフォルトでは、キーがラベルの左側に表示されます。

次の表は、引数 <位置> のコードを示しています：

コード	位置
0	凡例の位置は、組み込み位置オプションの1つではありません。
1	左上
2	左下
3	右上
4	右下
5	左
6	右
7	上
8	下

引数 <水平オフセット> と <垂直オフセット> は、<位置> が組み込み凡例位置にはないとき (<位置> = 0) に使用されます。<水平オフセット> は、グラフの左側から凡例の左側までをポイント単位で表します。<垂直オフセット> は、グラフの上の端から凡例の上の端までをポイント単位で表します。

次の例は、グラフの上端中央に凡例を表示します。。

CT SET LEGEND ATTRIBUTES (エリア ; \$ChartID ; 1 ; 0 ; 0 ; 0 ; 7 ; 0 ; 0)

参照 : CT GET LEGEND ATTRIBUTES、CT Get legend text、
CT SET LEGEND TEXT、CT SET CHART TEXT ATTRIBUTES

注 : 凡例テキストのテキスト属性を取得するには、**CT SET CHART TEXT ATTRIBUTES** コマンドを使用してください。

CT Get legend text

CT Get legend text (エリア; オブジェクト; 凡例項目) テキスト

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
凡例項目	整数	凡例項目番号

CT Get legend text関数は、指定された凡例項目にテキストを返します。

引数 < 凡例項目 > は、凡例内での系列の番号(あるいは、円グラフでの項目)です。ただし、凡例が逆順になった場合には、< 凡例項目 > は逆順ではなく正順を反映します。

次の例は、\$ChartIDに指定されたチャート内の最初の系列の凡例テキストを返します。

```
$Text:= CT Get legend text (エリア ; $ChartID ; 1)
```

参照 : CT SET LEGEND TEXT、CT GET LEGEND ATTRIBUTES、CT SET LEGEND ATTRIBUTES

CT SET LEGEND TEXT

CT SET LEGEND TEXT(エリア; オブジェクト; 凡例項目 ; 凡例テキスト)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
凡例項目	整数	凡例項目番号
凡例テキスト	テキスト	凡例項目のテキスト

CT SET LEGEND TEXTコマンドは、指定された凡例項目にテキストを設定します。

引数 < 凡例項目 > は、凡例内での系列の番号(あるいは、円グラフでの項目)です。ただし、凡例が逆順になった場合には、< 凡例項目 > は逆順ではなく正順を反映します。

次の例は、\$ChartIDに指定されたチャートの凡例テキストを変更します。

```
ARRAY STRING(20 ; aLegend ; 3)
```

```
aLegend{1}:="売上"
```

```
aLegend{2}:="マーケティング"
```

```
aLegend{3}:="エンジニアリング"
```

```
For ($i ; 1 ; 3)
```

```
    CT SET LEGEND TEXT (エリア ; $ChartID ; $i;aLegend{$i})
```

```
End for
```

参照 : CT Get legend text、CT GET LEGEND ATTRIBUTES、CT SET LEGEND ATTRIBUTES

CT GET REAL SCALE

CT GET REAL SCALE(エリア; オブジェクト; 最小自動; 最大自動; 主増分自動; 補助増分自動; 最小; 最大; 主増分; 補助増分)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
最小自動	整数	デフォルトの最小値を使用しているか? 0=使用しない、1=使用する
最大自動	整数	デフォルトの最大値を使用しているか? 0=使用しない、1=使用する
主増分自動	整数	デフォルトの主増分を使用しているか? 0=使用しない、1=使用する
補助増分自動	整数	デフォルトの補助増分を使用しているか? 0=使用しない、1=使用する
最小	実数	最小値を受け取る
最大	実数	最大値を受け取る
主増分	実数	主増分を受け取る
補助増分	実数	補助増分を受け取る

CT GET REAL SCALEコマンドは、デフォルト値が使用されるのかどうか、数値軸目盛りに設定されている補助値が何であるのかを返します。**CT GET REAL SCALE**コマンドは、値が実数、整数、または倍長整数のときに使用されます。

引数 <最小自動> と <最大自動> は、グラフがデフォルトの最小値と最大値を現在使用しているかどうかを示します。

引数 <主増分自動> と <補助増分自動> は、グラフがデフォルトの主増分と補助増分を現在使用しているかどうかを示します。

引数 <最小> と <最大> は、「軸」ダイアログボックスでユーザが設定するか、または設計者が **CT SET REAL SCALE** コマンドを使用するかによって設定される最小値と最大値です。

引数 <主増分> と <補助増分> は、「軸」ダイアログボックスでユーザが設定するか、または設計者が **CT SET REAL SCALE** コマンドを使用するかによって設定される主増分と補助増分です。

次の例は、\$ChartIDに指定されたチャートの目盛りデータを返します。

CT GET REAL SCALE (エリア ; \$ChartID ; \$MinA ; \$MaxA ; \$MajA ; \$MinA ; \$Minimum ; \$Maximum ; \$MajorInc ; \$MinorInc)

参照 : CT SET REAL SCALE、CT GET DATE SCALE、CT SET DATE SCALE

CT SET REAL SCALE

CT SET REAL SCALE(エリア; オブジェクト; 最小自動; 最大自動; 主増分自動; 補助増分自動; 最小; 最大; 主増分; 補助増分)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
最小自動	整数	デフォルトの最小値を使用しているか? -1=変更なし、0=使用しない、1=使用する
最大自動	整数	デフォルトの最大値を使用しているか? -1=変更なし、0=使用しない、1=使用する
主増分自動	整数	デフォルトの主増分を使用しているか? -1=変更なし、0=使用しない、1=使用する
補助増分自動	整数	デフォルトの補助増分を使用しているか? -1=変更なし、0=使用しない、1=使用する
最小	実数	最小値を受け取る
最大	実数	最大値を受け取る
主増分	実数	主増分を受け取る
補助増分	実数	補助増分を受け取る

CT SET REAL SCALE コマンドは、デフォルト値が使用されるのかどうか、数値軸目盛りに設定されている補助値が何であるのかを設定します。**CT SET REAL SCALE** コマンドは、値が実数、整数、または倍長整数のときに使用されます。

引数 <最小自動> と <最大自動> は、グラフがデフォルトの最小値と最大値を現在使用しているかどうかを示します。

引数 <主増分自動> と <補助増分自動> は、グラフがデフォルトの主増分と補助増分を現在使用しているかどうかを示します。

引数 <最小> と <最大> は、代替最小値と代替最大値です。

引数 <主増分> と <補助増分> は、代替主増分と代替補助増分です。

次の例は、配列からチャートを生成し、目盛り値を設定します。

`$ChartID:= CT Chart arrays (エリア ; 2 ; 1 ; aCategories ; aSeries ; aValues)`

`CT SET REAL SCALE (エリア ; $ChartID ; 0 ; 0 ; 0 ; 0 ; -100 ; 300 ; 100 ; 20)`

参照 : CT GET REAL SCALE、CT GET DATE SCALE、CT SET DATE SCALE

CT GET DATE SCALE

CT GET DATE SCALE(エリア; オブジェクト; 最小自動; 最大自動; 主増分自動; 補助増分自動; 最小; 最大; 主増分タイプ; 主増分; 補助増分タイプ; 補助増分)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
最小自動	整数	デフォルトの最小値を使用しているか? 0=使用しない、1=使用する
最大自動	整数	デフォルトの最大値を使用しているか? 0=使用しない、1=使用する
主増分自動	整数	デフォルトの主増分を使用しているか? 0=使用しない、1=使用する
補助増分自動	整数	デフォルトの補助増分を使用しているか? 0=使用しない、1=使用する
最小	日付	最小値を受け取る
最大	日付	最大値を受け取る
主増分タイプ	整数	主増分タイプはなにか? 1=日、2=週、3=月、4=年
主増分	整数	主増分を受け取る
補助増分タイプ	整数	補助増分を受け取る
補助増分	整数	補助増分を受け取る

CT GET DATE SCALEコマンドは、デフォルト値が使用されるのかどうか、数値軸目盛り用に設定されている補助値が何であるのかを返します。**CT GET DATE SCALE**コマンドは、値が日付のときに使われます。

引数<最小自動>と<最大自動>は、グラフがデフォルトの最小値と最大値を現在使用しているかどうかを示します。

引数<主増分自動>と<補助増分自動>は、グラフがデフォルトの主増分と補助増分を現在使用しているかどうかを示します。

引数<最小>と<最大>は、「軸」ダイアログボックスでユーザが設定するか、または設計者が**CT SET DATE SCALE**コマンドを使用するかによって設定される最小値と最大値です。

引数<主増分タイプ>と<補助増分タイプ>は、<主増分>と<補助増分>が指定される単位用のコードです。

引数<主増分>と<補助増分>は、「軸」ダイアログボックスでユーザが設定するか、または設計者が**CT SET DATE SCALE**コマンドを使用するかによって設定される主増分と補助増分です。

次の例は、\$ChartIDに指定されたチャートの目盛りデータを返します。

CT GET DATE SCALE (エリア ; \$ChartID ; \$MinA ; \$MaxA ; \$MajA ; \$MinA ; \$Minimum ; \$Maximum ; \$MajType ; \$MajorInc ; \$MinType ; \$MinorInc)

参照 : CT SET DATE SCALE、CT GET REAL SCALE、CT SET REAL SCALE

CT SET DATE SCALE

CT SET DATE SCALE(エリア; オブジェクト; 最小自動; 最大自動; 主増分自動; 補助増分自動; 最小; 最大; 主増分タイプ; 主増分; 補助増分タイプ; 補助増分)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
最小自動	整数	デフォルトの最小値を使用しているか? 0=使用しない、1=使用する
最大自動	整数	デフォルトの最大値を使用しているか? 0=使用しない、1=使用する
主増分自動	整数	デフォルトの主増分を使用しているか? 0=使用しない、1=使用する
補助増分自動	整数	デフォルトの補助増分を使用しているか? 0=使用しない、1=使用する
最小	日付	最小値を受け取る
最大	日付	最大値を受け取る
主増分タイプ	整数	主増分タイプはなにか? 1=日、2=週、3=月、4=年
主増分	整数	主増分を受け取る
補助増分タイプ	整数	補助増分を受け取る 1=日、2=週、3=月、4=年
補助増分	整数	補助増分を受け取る

CT SET DATE SCALEコマンドは、デフォルト値を使用するかどうかの指定、数値軸目盛り用の代替値の設定に使用します。**CT SET DATE SCALE**コマンドは、値が日付のときに使われます。

引数<最小自動>と<最大自動>は、グラフがデフォルトの最小値と最大値を現在使用しているかどうかを示します。

引数<主増分自動>と<補助増分自動>は、デフォルトの最小値と最大値を使用するかどうかを示します。

引数<最小>と<最大>は、代替最小値と代替最大値です。

引数<主増分タイプ>と<補助増分タイプ>は、<主増分>と<補助増分>が指定される単位用のコードです。

引数<主増分>と<補助増分>は、代替主増分と代替補助増分です。

次の例は、データベースからチャートを作成し、目盛り値を設定します。

```
ARRAY INTEGER(aYFields ; 2)
```

```
aYFields{1}:=2
```

```
aYFields{2}:=3
```

```
$ChartID:= CT Chart selection (エリア ; 2 ; 1 ; 1 ; File(>>[顧客]));
```

```
    Field(>>[顧客]形態) ; aYFields)
```

```
CT SET DATE SCALE (エリア ; $ChartID ; 0 ; 0 ; 0 ; 0 ; 0 ; !95.01.01!;!97.06.30!;4;1;3;1)
```

参照 : CT GET DATE SCALE、CT GET REAL SCALE、CT SET REAL SCALE

CT GET DEPTH

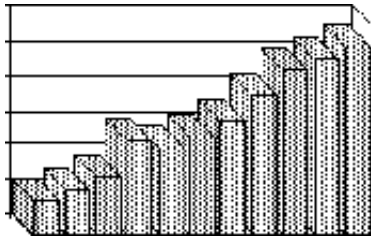
CT GET DEPTH (エリア; オブジェクト; 水平オフセット; 垂直オフセット)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
水平オフセット	整数	ポイント単位で横方向のオフセットを受け取る
垂直オフセット	整数	ポイント単位で縦方向のオフセットを受け取る

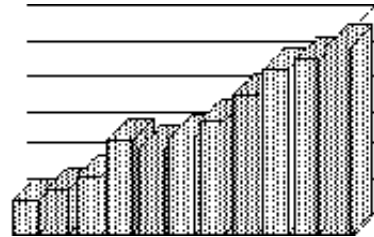
CT GET DEPTHコマンドは、引数<エリア>、<オブジェクト>で指定されたグラフの横方向と縦方向のオフセット(奥行き)を返します。このコマンドは、2次元グラフだけで有効です。

引数<水平オフセット>は、ポイント単位で測定した横方向のオフセットです。正の値はオフセットが右方向になることを示し、負の値はオフセットが左方向になることを示します。

水平オフセット=5、垂直オフセット=5

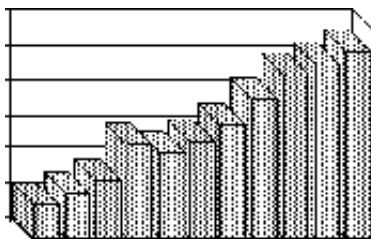


水平オフセット=-5、垂直オフセット=5

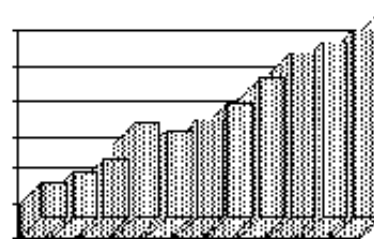


引数<垂直オフセット>は、ポイント単位で測定した縦方向のオフセットです。正の値はX軸から奥に入る距離を示し、負の値はX軸から前に入る距離を示します。

水平オフセット=5、垂直オフセット=5



水平オフセット=5、垂直オフセット=-5



次の例は、\$ChartIDに指定されたチャートの奥行きをチェックし、ユーザによって変更されていない場合には奥行きを設定します。

```
CT GET DEPTH (エリア ; $ChartID ; $Horiz ; $Vert)
If ($Horiz=0) & ($Vert=0)
    CT SET DEPTH (エリア ; $ChartID ; 10 ; 10)
End if
```

参照 : CT SET DEPTH

CT SET DEPTH

CT SET DEPTH (エリア; オブジェクト; 水平オフセット ; 垂直オフセット)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
水平オフセット	整数	ポイント単位で横方向のオフセットを受け取る (-32000よりも大きい必要がある)
垂直オフセット	整数	ポイント単位で縦方向のオフセットを受け取る (-32000よりも大きい必要がある)

CT GET DEPTHコマンドは、引数<エリア>、<オブジェクト>で指定されたグラフの横方向と縦方向のオフセット(奥行き)を設定します。このコマンドは、2次元グラフだけで有効です。

引数<水平オフセット>は、ポイント単位で測定した横方向のオフセットです。正の値はオフセットが右方向になることを示し、負の値はオフセットが左方向になることを示します。

引数<垂直オフセット>は、ポイント単位で測定した縦方向のオフセットです。正の値はX軸から奥に入る距離を示し、負の値はX軸から前に入る距離を示します。

横方向と縦方向の奥行きの実例については、**CT GET DEPTH**コマンドを参照してください。

CT GET DEPTHコマンドの例を参照してください。

参照 : CT SET DEPTH

CT GET 3D VIEW

CT GET 3D VIEW (エリア; オブジェクト; 回転; 仰角)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
回転	実数	度数単位で回転を受け取る(0から90まで)
仰角	実数	度数単位で仰角を受け取る(0から90まで)

CT GET 3D VIEWコマンドは、引数<エリア>と<オブジェクト>で指定されたグラフの回転と仰角を返します。このコマンドは、3次元グラフだけで有効です。

引数<回転>は、Z軸を中心としたグラフの回転です。

引数<仰角>は、X軸を中心としたグラフの回転です。

次の例は、\$ChartIDに指定されたチャートの回転と仰角を\$Rotationと\$Elevation変数に返します。

CT GET 3D VIEW(エリア; \$ChartID; \$Rotation; \$Elevation)

参照: CT SET 3D VIEW

CT SET 3D VIEW

CT SET 3D VIEW (エリア; オブジェクト; 回転; 仰角)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
回転	実数	度数単位の回転 (0から90まで、それ以外の値のは何も行わない)
仰角	実数	度数単位の仰角 (0から90まで、それ以外の値のは何も行わない)

CT SET 3D VIEWコマンドは、引数<エリア>と<オブジェクト>で指定されたグラフの回転と仰角を設定します。このコマンドは、3次元グラフだけで有効です。

引数<回転>は、Z軸を中心としたグラフの回転です。

引数<仰角>は、X軸を中心としたグラフの回転です。

次の例は、\$ChartIDに指定されているチャートの回転と仰角を30度に設定します。

CT SET 3D VIEW(エリア; \$ChartID; 30; 30)

参照: CT GET 3D VIEW

CT GET CHART PART

CT GET CHART PART (エリア; オブジェクト; タイプ; 特定部分)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
タイプ	整数	エリアで選択されたオブジェクトのタイプを受け取る
特定部分	倍長整数	選択されたオブジェクトの特定の部分を受け取る

CT GET CHART PARTコマンドは、引数<エリア>と<オブジェクト>で指定されたグラフで現在選択されているチャートオブジェクトのコードを<タイプ>と<特定部分>変数に返します。

引数<タイプ>と<特定部分>は、ユーザが選択しているグラフの部分を示します。<タイプ>と<特定部分>のコードは、付録Dに掲載されています。

次の例は、選択された系列用の部分コードを取得し、その系列の塗りつぶしパターンを無地、塗りつぶしカラーを緑に設定します。

```
CT GET CHART PART (エリア ; $ChartID ; $Type ; $Specifics)
```

```
If ($Type=8) `それは系列か？
```

```
    CT SET CHART FILL ATTRIBUTES (エリア ; $ChartID ; $Type ; $Specifics ;3;
```

```
        CT Index to color (10))
```

```
End if
```


CT GET CHART OPTIONS

CT GET CHART OPTIONS (エリア; オブジェクト; オプション)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
オプション	整数配列	オプションコードを受け取るための変数を含む配列

CT GET CHART OPTIONS コマンドは、選択されたグラフのオプションを返します。これらのオプションは、各チャートタイプに対して「オプション」ダイアログボックスでユーザが設定できるオプションと同じです。

各グラフタイプのコードについては、付録Dを参照してください。

次の例は、\$ChartIDに指定されたチャートのオプションをOptionsに返します。

ARRAY INTEGER(aOptions ; 0)

CT GET CHART OPTIONS (エリア ; \$ChartID ; aOptions)

参照 : CT SET CHART OPTIONS

CT SET CHART OPTIONS

CT SET CHART OPTIONS (エリア; オブジェクト; オプション)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
オプション	整数配列	オプションコードを含む配列

CT SET CHART OPTIONSコマンドは、選択されたグラフのオプションを設定します。これらのオプションは、各チャートタイプに対して「オプション」ダイアログボックスでユーザが設定できるオプションと同じです。

各グラフタイプのコードについては、付録Dを参照してください。

次の例は、\$ChartIDで指定されたカラムチャートのチャートオプションを横方向、スタック比例、100パーセントオーバーラップ、25パーセントのギャップに設定します。

ARRAY INTEGER (aOptions ; 4)

aOptions{1}:=1

aOptions{2}:=2

aOptions{3}:=100

aOptions{4}:=25

CT SET CHART OPTIONS (エリア ; \$ChartID ; aOptions)

参照 : CT GET CHART OPTIONS

CT Get chart type

CT Get chart type (エリア; オブジェクト) 整数

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID

CT Get chart type関数は、引数<エリア>と<オブジェクト>で指定されたチャートのタイプを返します。

次の表は、チャートタイプのコードです：

コード	チャートタイプ
1	2D 面
2	2D 棒
3	2D ピクチャ
4	2D 線
5	2D 散布図
6	2D 円
7	2D ポーラー
100	3D 棒
101	3D 線
102	3D 面
103	3D 等高線
104	3D 三角形
105	3D ピン

次の例では、**CT Get chart type**関数は\$ChartIDに指定されているチャートのチャートタイプを取得するために使用されます。カラムチャートではない場合には、**CT SET CHART TYPE**コマンドはチャートをカラムチャートに変更します。

```
If (CT Get chart type (エリア ; $ChartID)#2)
    CT SET CHART TYPE (エリア; $ChartID ; 2)
End if
```

参照：CT SET CHART TYPE

CT SET CHART TYPE

CT SET CHART TYPE (エリア; オブジェクト; タイプ)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
タイプ	整数	チャートタイプ

CT SET CHART TYPEコマンドは、指定されたチャートのタイプを<タイプ>に変更します。

引数<タイプ>は、2次元グラフに対しては2Dグラフタイプ、3次元グラフに対しては3Dグラフタイプである必要があります。

CT Get chart type関数の例を参照してください。

参照 : CT Get chart type

CT UPDATE CHART

CT UPDATE CHART (エリア; オブジェクト; 警告表示)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
警告表示	整数	ユーザに警告を表示するか?

CT UPDATE CHARTコマンドは、データベース内のデータから作成されたグラフを更新します。このコマンドは、「チャート」メニューの「更新」メニューコマンドと同じです。

このコマンドはデータベースのフィールドからユーザが作成したグラフ、または開発者が**CT Chart selection**関数または**CT Chart data**関数を使用して作成したグラフだけを更新します。

グラフは、グラフ化されるテーブルのカレントセレクション内のレコードを使用して更新されます。

引数 <警告表示> が1である場合には、ユーザに「アラート」ボックスが表示されるので、アクションの受け付けかキャンセルを選択できます。 <警告表示> が0の場合は、「アラート」ボックスは表示されません。

次の例は、グラフ化されるテーブルのセレクションを変更し、変更を表示するようにグラフを更新します。

```
REDUCE SELECTION([テーブル]; 350)  
CT UPDATE CHART(エリア; $ChartID; 0)
```

CT Get chart picture

CT Get chart picture (エリア; オブジェクト; タイプ; 特定部分) ピクチャ

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
タイプ	整数	ピクチャを取得するオブジェクトのタイプ
特定部分	倍長整数	ピクチャを取得するオブジェクトの特定の部分

CT Get chart picture関数は、ピクチャチャートの指定されたシリーズに表示されているピクチャを返します。ピクチャはピクチャ変数内に返されます。

引数 <タイプ> は8である必要があり、これはチャート内の系列です。

引数 <特定部分> は、系列の数値に100を乗算したものと同等である必要があります。

次の例はピクチャグラフの最初の系列からピクチャをコピーし、それをクリップボードに配置します。

```
$Pict:= CT Get chart picture (エリア ; $ChartID ; 8 ; 100)  
CT PICTURE TO CLIPBOARD ($Pict)
```

参照 : CT SET CHART PICTURE

CT SET CHART PICTURE

CT SET CHART PICTURE (エリア; オブジェクト; タイプ; 特定部分; ピクチャ)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
タイプ	整数	属性を取得するオブジェクトのタイプ
特定部分	倍長整数	属性を取得するオブジェクトの特定の部分
ピクチャ	ピクチャ	「ピクチャ」グラフの系列に貼り付けるピクチャ

CT SET CHART PICTUREコマンドは、指定された系列に引数<ピクチャ>を貼り付けます。

引数<オブジェクト>は、ピクチャグラフである必要があります。

引数<タイプ>は8である必要があります、これはグラフ内の系列です。

引数<特定部分>は、系列の数値に100を乗算したものと同等である必要があります。

次の例は、グラフがピクチャグラフである場合には、クリップボードから選択したグラフの最初の系列にピクチャをコピーします。

```

$ChartID:= CT Get ID (エリア ; 0 ; 1)
If (CT Get chart type (エリア ; $ChartID)=3)
    $Pict:= CT Clipboard to picture
    CT SET CHART PICTURE (エリア ; $ChartID ; 8 ; 100 ; $Pict)
End if
    
```

参照 : CT Get chart picture

CT EXPLODE PIE

CT EXPLODE PIE (エリア; オブジェクト; 項目 ; パーセンテージ)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
オブジェクト	倍長整数	オブジェクトID
項目	整数	切り離す円グラフのパイウェッジの項目番号 (0=すべてのパイウェッジ)
パーセンテージ	整数	半径の長さのパーセンテージ(0から1000まで)

CT EXPLODE PIEコマンドは、円グラフの中心から指定されたパイウェッジを移動します。

引数<項目>は、移動させるパイウェッジの項目番号です。<項目>が0の場合には、すべてのパイウェッジが移動されます。

引数<パーセンテージ>は、半径の長さのパーセンテージとして指定された、パイウェッジの移動距離です。このため、円グラフのサイズが変更されると、移動距離は新しい半径に従って変わります。

次の例は\$ChartIDに指定された円グラフの最初の3つの部分を切り離します。各パイウェッジは円の半径の5%の距離で外側に移動されます。

```
For ($i;1;3)  
  CT EXPLODE PIE (エリア ; $ChartID ; $i ; 5)  
End for
```


次のコマンドと関数は、「ルーチン」エリアの「Hot Links」テーマ内にあります：

CT PUBLISH
CT ADD TO HOT LINK
CT UNPUBLISH

CT Subscribe
CT UNSUBSCRIBE

これらのコマンドと関数によって、4D Chartホットリンクの発行、変更、発行解除がメソッドでできます。また、任意のプラグインからのピクチャホットリンクの引用、引用解除ができます。

CT PUBLISH

CT PUBLISH (エリア; スコープ; 名前)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	コマンドのスコープ -1=すべて、0=選択されたオブジェクト >0=オブジェクトID
名前	文字列	ホットリンクの名前

CT PUBLISHコマンドは、引数<エリア>内の<スコープ>で記述されるオブジェクトを<名前>という新しいホットリンクとして発行します。

<スコープ>が-1の場合には、**CT PUBLISH**コマンドは<エリア>にある全オブジェクトを発行します。

<スコープ>が0の場合には、**CT PUBLISH**コマンドは<エリア>内で選択されたオブジェクトを発行します。

<スコープ>が0よりも大きな値の場合には、それが特定オブジェクトのIDと同じである必要があり、そのオブジェクトだけが発行されます。

引数<名前>が空の文字列("")の場合には、ユーザが名前を選択できる「ホットリンクを発行」ダイアログボックスが表示されます。

次の例は、ユーザにホットリンク名を要求し、そこで選択されたオブジェクトをホットリンクとして発行します。入力した名前がすでに存在する場合にはユーザに警告しません。

```
$Name := Request ("選択したオブジェクトのホットリンク名を入力してください。")
If (OK = 1)
  CT PUBLISH (エリア ; 0 ; $Name)
  If (CT Error= 37)
    ALERT ("その名前を持ったホットリンクがすでに存在します!")
  End if
End if
```

参照 : CT Get ID、CT Subscribe、CT UNPUBLISH

CT ADD TO HOT LINK

CT ADD TO HOT LINK (エリア; スコープ; 名前)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	コマンドのスコープ -1=すべて、0=選択されたオブジェクト >0=オブジェクトID
名前	文字列	ホットリンクの名前

CT ADD TO HOT LINKコマンドは、ホットリンク名<名前>に<スコープ>で記述された<エリア>のオブジェクトを追加します。

<スコープ>が-1の場合には、**CT ADD TO HOT LINK**コマンドは文書内の全オブジェクトをホットリンクに追加します。

<スコープ>が0の場合には、**CT ADD TO HOT LINK**コマンドは現在選択されているオブジェクトをホットリンクに追加します。

<スコープ>が0よりも大きな値の場合には、それが特定オブジェクトのIDと同じである必要があり、そのオブジェクトがホットリンクに追加されます。

引数<名前>が空の文字列("")の場合には、ユーザがホットリンクを選択できる「ホットリンク選択」ダイアログボックスが表示されます。

<名前>が<エリア>のホットリンクではない場合には、**CT ADD TO HOT LINK**コマンドは何も行いません。

次の例は、ユーザにホットリンク名を要求し、そこで指定されたホットリンクに選択されたオブジェクトを追加します。ホットリンクが存在しない場合にはユーザに警告します。

```
$Name := Request ("選択したオブジェクトをどのホットリンクに追加しますか？")
If (OK = 1)
    CT ADD TO HOT LINK (エリア ; 0 ; $Name)
    If (CT Error=5)
        ALERT ("その名前のホットリンクは存在しません。")
    End if
End if
```

参照 : CT PUBLISH、CT Subscribe

CT UNPUBLISH

CT UNPUBLISH (エリア ; 名前)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
名前	文字列	ホットリンクの名前

CT UNPUBLISHコマンドは、引数<エリア>の<名前>というホットリンクの発行を解除します。このコマンドは、「ホットリンク解除」ダイアログボックスでホットリンクを選択することと同じです。

引数<名前>が空の文字列("")の場合には、発行を解除するホットリンクをユーザが選択できる「ホットリンク解除」ダイアログボックスが表示されます。

<名前>が存在しないか、あるいは<名前>が<エリア>から発行されたホットリンクではない場合には、**CT UNPUBLISH**コマンドは何も行いません。

次の例は、ユーザにホットリンク名を要求し、そこで指定されたホットリンクの発行を解除します。

```
$Name := Request ("発行を解除するホットリンクの名前を入力してください。")
If (OK = 1)
    CT UNPUBLISH (Area;$Name)
    If (CT Error=5)
        ALERT("その名前のホットリンクは存在していません。")
    End if
End if
```

参照 : CT PUBLISH、CT Subscribe

CT Subscribe

CT Subscribe (エリア ; 名前) 倍長整数

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
名前	文字列	ホットリンクの名前

CT Subscribe関数は、すでに存在しているホットリンク名<名前>を引用し、新しいオブジェクトのIDを返します。

この関数は、引数<エリア>内に新しいオブジェクトを作成します。

引数<名前>が空の文字列("")の場合には、ユーザがホットリンクを選択できる「ホットリンク引用」ダイアログボックスが表示されます。

<名前>が存在しない場合には、**CT Subscribe**関数は-32000を返します。

指定されたホットリンクがピクチャタイプのホットリンクである場合には、PICTが4D Chartエリアのピクチャとして表示されます。指定されたホットリンクが値タイプのホットリンクである場合には、4D Chartが「プロパティ」ダイアログボックスで指定されたデフォルトホットリンクタイプを使用してデータからグラフを生成します。

次の例は、ユーザにホットリンク名を要求し、そこで指定されたホットリンクを引用します。引用が正常に行われた場合には、表示可能なエリアでホットリンクが中央に配置されます。

```
$Name := Request ("どのホットリンクを引用しますか?") `ホットリンクを取得する
If (OK = 1) `要求が受け付けられたら、
    $Hotlink := CT Subscribe (エリア ; $Name) `そのホットリンクを引用する
    If (CT Error=0) `エラーがなければ、
        `エリアのクリップされた境界を取得する
        CT GET AREA BOUNDARY (エリア ; 1 ; $left ; $top ; $right ; $bottom)
        `ホットリンクの境界を取得する
        CT GET BOUNDARY (エリア ; $HotLink ; $left2 ; $top2 ; $right2 ; $bottom2)
        `ホットリンクをエリアの中央に配置する
        CT MOVE (エリア ; $HotLink ; (($right-$left)-($right2-$left2)) / 2 ; (($bottom-$top)-($bottom2-$top2)) / 2)
    End if
End if
```

参照 : CT PUBLISH、CT UNPUBLISH

CT UNSUBSCRIBE

CT UNSUBSCRIBE (エリア ; 名前)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
名前	文字列	ホットリンクの名前

CT UNSUBSCRIBEコマンドは、引数<名前>で記述されたホットリンクへの<エリア>の引用を解除します。

<名前>のすべてのコピーは<エリア>から消去されます。このコマンドは、「ホットリンク引用解除」ダイアログボックスから<名前>を選択することと同じです。ホットリンクの1つのインスタンスだけを消去する場合は、そのオブジェクトを削除するだけで構いません。

<名前>が空の文字列("")の場合には、ユーザがホットリンクを選択できる「ホットリンク引用解除」ダイアログボックスが表示されます。

次の例は、ユーザにホットリンク名を要求し、そこで指定されたホットリンクのすべてのインスタンスを消去します。

```
$Name := Request ("どのホットリンクを消去しますか？")
If (OK = 1)
    CT UNSUBSCRIBE (エリア ; $Name)
End if
```

参照 : CT PUBLISH、CT Subscribe

次のコマンドと関数は、「ルーチン」エリアの「Objects」テーマ内にあります：

CT Draw text	CT GET LINE ATTRIBUTES
CT Draw line	CT SET LINE ATTRIBUTES
CT Draw rectangle	CT GET FILL ATTRIBUTES
CT Draw oval	CT SET FILL ATTRIBUTES
CT Array to polygon	CT SELECT
CT Place picture	CT MOVE
CT Get ID	CT SIZE
CT Get object type	CT Count
CT Get refnum	CT ALIGN
CT SET REFNUM	CT GET HIGHLIGHT
CT GET BOUNDARY	CT SET HIGHLIGHT
CT GET TEXT ATTRIBUTES	CT INSERT FIELD
CT SET TEXT ATTRIBUTES	CT INSERT EXPRESSION

このテーマにあるコマンドと関数を使用すると、線、矩形、テキストなどのオブジェクトの描画および変更を行うことができます。

CT Draw text

CT Draw text (エリア ; 左 ; 上 ; 右 ; 下 ; テキスト) 倍長整数

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
左	実数	開始点の横方向の位置 (ポイント単位)
上	実数	開始点の縦方向の位置 (ポイント単位)
右	実数	終了点の横方向の位置 (ポイント単位)
下	実数	終了点の縦方向の位置 (ポイント単位)
テキスト	実数	新しいテキストオブジェクトのテキスト

CT Draw text関数は、引数<エリア>に新しいテキストオブジェクトを作成し、新しいオブジェクトのオブジェクトIDを返します。オブジェクトは、<左>、<上>、<右>、<下>の座標に従って配置されます。

次のメソッドは、チャートエリアの左上隅に「Hello World」というテキストを描画します。

```
$Text:= CT Draw text (エリア ; 0 ; 0 ; 300 ; 10 ; "Hello World")
```


CT Draw line

CT Draw line (エリア ; 左 ; 上 ; 右 ; 下 ; 矢印形態) 倍長整数

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
左	実数	開始点の横方向の位置 (ポイント単位)
上	実数	開始点の縦方向の位置 (ポイント単位)
右	実数	終了点の横方向の位置 (ポイント単位)
下	実数	終了点の縦方向の位置 (ポイント単位)
矢印形態	実数	矢印形態コード (下記の表参照)

CT Draw line関数は、引数<エリア>に新しい線オブジェクトを作成し、新しいオブジェクトのオブジェクトIDを返します。オブジェクトは、<左>、<上>、<右>、<下>の座標に従って配置されます。

下記は、引数<矢印形態>用のコードです。

コード	矢印形態
-1	デフォルト値
0	なし
1	始点
2	終点
3	両端

次のメソッドは、チャートエリアに矢印付きの線を描画します。

```
$Line:= CT Draw line (エリア ; 10 ; 10 ; 50 ; 50 ; 3)
```

CT Draw rectangle

CT Draw rectangle (エリア ; 左 ; 上 ; 右 ; 下 ; 角の丸み) 倍長整数

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
左	実数	開始点の横方向の位置 (ポイント単位)
上	実数	開始点の縦方向の位置 (ポイント単位)
右	実数	終了点の横方向の位置 (ポイント単位)
下	実数	終了点の縦方向の位置 (ポイント単位)
矢印形態	実数	角の丸みの度合い (ポイント単位)

CT Draw rectangle関数は、引数<エリア>に新しい矩形オブジェクトを作成し、新しいオブジェクトのオブジェクトIDを返します。オブジェクトは、<左>、<上>、<右>、<下>の座標に従って配置されます。

引数<角の丸み>は、新しい矩形の角の丸みの度合いを調整します。<角の丸み>が0ならば、角は丸くなりません。

次のメソッドは、丸みのある矩形をチャートエリアに描画します。

```
$Rect:= CT Draw rectangle (エリア ; 5 ; 5 ; 200 ; 200 ; 5)
```

CT Draw oval

CT Draw oval (エリア ; 左 ; 上 ; 右 ; 下) 倍長整数

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
左	実数	開始点の横方向の位置 (ポイント単位)
上	実数	開始点の縦方向の位置 (ポイント単位)
右	実数	終了点の横方向の位置 (ポイント単位)
下	実数	終了点の縦方向の位置 (ポイント単位)

CT Draw oval関数は、引数<エリア>に新しい楕円オブジェクトを作成し、その新しいオブジェクトのオブジェクトIDを返します。オブジェクトは、<左>、<上>、<右>、<下>の座標に従って配置されます。

次のメソッドは、チャートエリアに円を描画します。

```
$Oval:= CT Draw oval (エリア ; 5 ; 5 ; 100 ; 100)
```

CT Array to polygon

CT Array to polygon (エリア ; 水平配列 ; 垂直配列) 倍長整数

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
水平配列	実数配列	頂点の横方向の値の配列
垂直配列	実数配列	頂点の縦方向の値の配列

CT Array to polygon関数は、引数<水平配列>と<垂直配列>の配列を基にした新しい多角形を<エリア>に作成し、その新しいオブジェクトのIDを返します。

<水平配列>と<垂直配列>は、多角形の各頂点の位置を記述しています。この2つの配列にはポイント単位で実数、倍長整数、整数のいずれかのデータタイプを指定できます。多角形が正常に作成されるようにそれぞれの配列には少なくとも3つの要素が必要です。配列の要素が同じ数ではない場合には、大きい方の配列の余分の要素は無視されます。閉じた多角形を作成するには、各偉いの最後の値と最初の値を一致させる必要があります。

次の例は、2つの配列に値を入れ、そこから多角形を作成します。次にその多角形を移動し、サイズを変更します。

```

$Vertices:=Num(Request("頂点の数を入力してください。"))
If (OK=1)
  `配列を宣言する
  ARRAY REAL(aVerticeH ; $Vertices)
  ARRAY REAL(aVerticeV ; $Vertices)
  `配列に値を入れる
  For ($i ; 1 ; $Vertices)
    aVerticeH{$i}:=Sin($i)
    aVerticeV{$i}:=Cos($i)
  End for
  `多角形を描く
  $Poly:= CT Array to polygon (エリア ; aVerticeH ; aVerticeV)
  `多角形をエリア座標 (10,10) に移動する)
  CT MOVE (エリア ; $Poly ; 10 ; 10)
  `多角形のサイズを200 x 200に変更する
  CT SIZE (エリア ; $Poly ; 200 ; 200)
End if

```

CT Place picture

CT Place picture (エリア ; ピクチャ ; 左 ; 上) 倍長整数

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
ピクチャ	ピクチャ	配置する4Dのピクチャ
左	実数	左端からポイント単位で表した左座標
上	実数	上端からポイント単位で表した上座標

CT Place picture関数は、引数<左>と<上>によって決定される点で<エリア>に<ピクチャ>を貼り付けてから、そのピクチャのオブジェクトIDを返します。

引数<ピクチャ>は、正しい14th Dimensionピクチャ式である必要があります。

次の例は、指定された企業の<エリア>にピクチャフィールド[ロゴ]ロゴの内容を貼り付けます。

```
MyRequest:=Request ("どの会社のロゴがほしいですか？")
If (OK=1)
  QUERY ([ロゴ];[ロゴ]会社=MyRequest)
  If (Records in selection([ロゴ]>0)
    $NewPict:= CT Place Picture(エリア ; [ロゴ]ロゴ ; 10 ; 10)
  Else
    ALERT("この会社は存在しません。")
  End if
End if
```

CT Get ID

CT Get ID (エリア ; スコープ ; インデックス) 倍長整数

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	関数のスコープ -1=すべて、0=選択されたオブジェクト >0=オブジェクトID
インデックス	倍長整数	<スコープ>でのオブジェクトの数値

CT Get ID関数は、引数<スコープ>と<インデックス>で記述された<エリア>のオブジェクトに対する一意なIDを返します。この数値は他の多くの4D Chartコマンドで使用され、オブジェクトのIDになります。

オブジェクトのIDを取得するには、まず最初に対象のオブジェクトセットを指定し、次にそのセット内のオブジェクトの順序を指定します。オブジェクトは後ろのものから前のものへの順になっています。一番後ろのオブジェクトにはインデックス1がついていません。

<スコープ>が-1の場合には、<インデックス>は文書全体のオブジェクトの順序を指します。

<スコープ>が0の場合には、<インデックス>は現在選択されているオブジェクト内のオブジェクトの順序を指します。

<スコープ>が0よりも大きな場合には、それがグループに対するIDである必要があり、<インデックス>はそのグループ内部のオブジェクトの順序を指します。この最後の構文ではグループを解体せずにグループ内のオブジェクトを扱うことができます。

次の例は、選択したオブジェクトのIDを取り出す方法を示しています。

`vID := CT Get ID (エリア ; 0 ; 1)` `最初に選択したオブジェクトの IDを取得する

CT Get object type

CT Get object type (エリア ; スコープ) 倍長整数

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	関数のスコープ -1=すべて、0=選択されたオブジェクト >0=オブジェクトID

CT Get object type関数は、引数<スコープ>で記述された<エリア>内のオブジェクトのオブジェクトタイプを返します。

オブジェクトのタイプは整数コードによって記述され、一度オブジェクトが作成されると変更はできません。

<スコープ>が-1の場合には、**CT Get object type**関数は文書内のすべてのオブジェクトのオブジェクトタイプを返します。オブジェクトのタイプがすべて同じではないときには、**CT Get object type**関数は-32000を返します。

<スコープ>が0の場合には、**CT Get object type**関数は選択されたオブジェクトのオブジェクトタイプを返します。オブジェクトのタイプがすべて同じではないときには、**CT Get object type**関数は-32000を返します。

<スコープ>が0よりも大きな場合には、それが特定オブジェクトのIDと同じである必要があり、そのオブジェクトのタイプが返されます。オブジェクトが存在しない場合には、**CT Get object type**関数は-32000を返します。

次の表は、すべてのオブジェクトコードです。

コード	オブジェクトタイプ
1	テキスト
2	ホットリンク
3	ピクチャ
4	使用せず
5	チャート
6	矩形
7	多角形
8	楕円
9	使用せず
10	線
11	グループ

次のコマンドは、選択されたオブジェクトのIDを\$ID変数に返します。

\$ID:= **CT Get object type** (エリア ; 0)

参照 : CT Get ID

CT Get refnum

CT Get refnum (エリア ; スコープ) 倍長整数

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	関数のスコープ -2=デフォルト、-1=すべて、 0=選択されたオブジェクト、>0=オブジェクトID

CT Get refnum関数は、引数<スコープ>で記述された<エリア>にオブジェクトの参照番号を返します。

参照番号はオブジェクトに対応する倍長整数であり、一意である必要性はありません。参照番号はプロシージャだけに従って操作されます。参照番号は、ユーザがオブジェクトに割り当てるものです。これに対して、オブジェクトIDは4D Chartによって割り当てられるものです。

<スコープ>が-2の場合には、**CT Get refnum**関数はデフォルトの参照番号を返します。

<スコープ>が-1の場合には、**CT Get refnum**関数は文書内の全オブジェクトに対する参照番号を返します。オブジェクトの参照番号がすべて同じではないときには、**CT Get refnum**関数は、-32000を返します。

<スコープ>が0の場合には、**CT Get refnum**関数は選択されたオブジェクトの参照番号を返します。オブジェクトの参照番号がすべて同じではないときには、**CT Get refnum**関数は、-32000を返します。

<スコープ>が0よりも大きな場合には、それが特定オブジェクトのIDと同じである必要があり、そのオブジェクトの参照番号が返されます。オブジェクトが存在しない場合には、**CT Get refnum**関数は-32000を返します。

次の例は、<エリア>を含むフォーム上のボタン用オブジェクトメソッドです。オブジェクトメソッドは、実行されると1つのオブジェクトだけが選択されているかどうかをチェックし、対応するレコードを[部分]テーブルで検索し、その記述を表示します。

QUERY ([部分];[部分]参照番号 = **CT Get refnum** (エリア ; 0))

ALERT ("このオブジェクトは" + [部分]記述 + "です。")

参照 : CT Get ID, CT SET REFNUM

CT SET REFNUM

CT SET REFNUM (エリア ; スコープ ; 参照番号)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	コマンドのスコープ -2=デフォルト、-1=すべて、 0=選択されたオブジェクト、>0=オブジェクトID
参照番号	倍長整数	参照番号

CT SET REFNUMコマンドは、引数<参照番号>を<スコープ>で記述された<エリア>にあるオブジェクト用の参照番号にします。参照番号は、それがオブジェクトを識別する方法であり、一意ではないという点でオブジェクト名と似ています。参照番号はオブジェクトID番号ではありません。オブジェクトID番号は文書内の各オブジェクトに対して4D Chartで割り当てられた一意の番号です。

<スコープ>が-2の場合は、**CT SET REFNUM**コマンドがデフォルトの参照番号を設定します。これは、任意の新しいオブジェクトに使用される参照番号です。

<スコープ>が-1の場合は、**CT SET REFNUM**コマンドは文書内の全オブジェクトに対する参照番号を設定します。

<スコープ>が0の場合は、**CT SET REFNUM**コマンドは選択されたオブジェクトに対する参照番号を設定します。

<スコープ>が0よりも大きな場合は、それが特定オブジェクトのIDと同じである必要があります。オブジェクトが存在しない場合は、**CT SET REFNUM**コマンドは何も行いません。

参照番号は、オブジェクトに対応する一意ではない倍長整数です。参照番号はメソッドだけに従って操作できます。参照番号のデフォルト値は、0です。

次の例は、選択したオブジェクトの参照番号をvNumber変数に含まれる値に変更します。

CT SET REFNUM (エリア ; 0 ; vNumber)

参照 : CT Get ID, CT Get refnum

CT GET BOUNDARY

CT GET BOUNDARY (エリア ; スコープ ; 左 ; 上 ; 右 ; 下)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	コマンドのスコープ -1=すべて、0=選択されたオブジェクト >0=オブジェクトID
左	実数	開始点の横方向の位置を受け取る (ポイント単位)
上	実数	開始点の縦方向の位置を受け取る (ポイント単位)
右	実数	終了点の横方向の位置を受け取る (ポイント単位)
下	実数	終了点の縦方向の位置を受け取る (ポイント単位)

CT GET BOUNDARYコマンドは、引数<スコープ>で記述された<エリア>にあるオブジェクトの境界を<左>、<上>、<右>、<下>変数に返します。

境界とは、オブジェクトを含む最も小さな矩形領域の座標です。

<スコープ>が-1の場合には、CT GET BOUNDARYコマンドは文書内のすべてのオブジェクトに対する境界を返します。

<スコープ>が0の場合には、CT GET BOUNDARYコマンドは選択されたオブジェクトの境界を返します。

<スコープ>が0よりも大きな場合には、それは特定オブジェクトのIDと同じである必要があり、そのオブジェクトの境界が返されます。オブジェクトが存在しない場合には、CT GET BOUNDARYコマンドは各座標に対して-32000を返します。

次の線は、選択したオブジェクトの境界を\$left、\$top、\$right、\$bottom変数に返します。

CT GET BOUNDARY (エリア ; 0 ; \$left ; \$top ; \$right ; \$bottom)

参照 : CT GET AREA BOUNDARY

CT GET TEXT ATTRIBUTES

CT GET TEXT ATTRIBUTES (エリア; スコープ; フォントID; フォントサイズ; スタイル; カラー; 位置揃え)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	コマンドのスコープ -3=選択されたテキスト、-2=デフォルト、-1=すべて 0=選択されたオブジェクト、>0=オブジェクトID
フォントID	整数	フォントIDを受け取る
フォントサイズ	整数	フォントサイズを受け取る
スタイル	整数	フォントスタイルを受け取る
カラー	倍長整数	カラーの値を受け取る
位置揃え	整数	テキストの位置揃えを受け取る

CT GET TEXT ATTRIBUTESコマンドは、引数<エリア>、<スコープ>で指定されたテキストの属性を引数<フォントID>、<フォントサイズ>、<カラー>、<位置揃え>に返します。

引数<フォントID>は、システム内にあるフォントのIDです。フォントのID番号は**CT Font number**関数を使用することによって取得できます。これらの関数の詳細については、第9章の「ユーティリティコマンド」を参照してください。

引数<フォントサイズ>は、反転表示されたテキストやテキストオブジェクトのポイント単位のサイズです。

引数<スタイル>は、複数のスタイル番号の加算の結果を混合した番号です。次の表はスタイル番号を示しています：

値	スタイル
0	標準
1	太字 (ボールド)
2	斜体 (イタリック)
4	下線 (アンダーライン)
8	アウトライン
16	シャドウ

引数<カラー>は、オブジェクトの色を指定する倍長整数です。**CT Index to color**関数や**CT RGB to color**関数を使用することで <カラー>の値を指定できます。これらの関数の詳細については、第9章の「ユーティリティコマンド」を参照してください。

引数<位置揃え>は、テキストの位置揃えです。

次の例は、選択したテキストオブジェクトの属性を\$Font、\$Size、\$Style、\$Color、\$Justify変数に返します。

CT GET TEXT ATTRIBUTES (エリア ; 0 ; \$Font ; \$Size ; \$Style ; \$Color ; \$Justify)

参照 : CT SET TEXT ATTRIBUTES、CT GET CHART TEXT ATTRIBUTES

注 : このコマンドは「テキスト」ツールや**CT Draw text**関数を使用して文書に追加されたテキストの属性を取得するために使用してください。軸ラベルなどのチャートテキストの属性を取得するには、第5章の「チャートコマンド」で説明するコマンドを使用してください。

CT SET TEXT ATTRIBUTES

CT SET TEXT ATTRIBUTES (エリア; スコープ; フォントID ; フォントサイズ ; スタイル ; カラー ; 位置揃え)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	コマンドのスコープ -3=選択されたテキスト、-2=デフォルト、-1=すべて 0=選択されたオブジェクト、>0=オブジェクトID
フォントID	整数	フォントIDを受け取る
フォントサイズ	整数	フォントサイズを受け取る
スタイル	整数	フォントスタイルを受け取る
カラー	倍長整数	カラーの値を受け取る
位置揃え	整数	テキストの位置揃えを受け取る

CT GET TEXT ATTRIBUTESコマンドは、引数<エリア>、<スコープ>で指定されたテキストにフォント、フォントサイズ、フォントスタイル、カラー、位置揃えを設定します。

引数<フォントID>は、システム内にあるフォントのIDです。フォントのID番号は**CT Font number**関数を使用することによって取得できます。これらの関数の詳細については、第9章の「ユーティリティコマンド」を参照してください。

引数<フォントサイズ>は、反転表示されたテキストやテキストオブジェクトのポイント単位のサイズです。

引数<スタイル>は、複数のスタイル番号の加算の結果を混合した番号です。次の表はスタイル番号を示しています：

値	スタイル
0	標準
1	太字 (ボールド)
2	斜体 (イタリック)
4	下線 (アンダーライン)
8	アウトライン
16	シャドウ

引数<カラー>は、オブジェクトの色を指定する倍長整数です。**CT Index to color**関数や**CT RGB to color**関数を使用することで <カラー>の値を指定できます。これらの関数の詳細については、第9章の「ユーティリティコマンド」を参照してください。

引数<位置揃え>は、テキストの位置揃えです。

次の例は、選択したテキストをTimes、14ポイント、太字、斜体、緑、中央揃えに設定します。

CT SET TEXT ATTRIBUTES (エリア ; 0 ; **CT Font number** ("Times") ; 14 ; 3 ; **CT Index to color** (10) ; 1)

参照 : CT GET TEXT ATTRIBUTES、CT SET CHART TEXT ATTRIBUTES

注 : このコマンドは「テキスト」ツールや**CT Draw text**関数を使用して文書に追加されたテキストの属性を取得するために使用してください。軸ラベルなどのチャートテキストの属性を取得するには、第5章の「チャートコマンド」で説明するコマンドを使用してください。

CT GET LINE ATTRIBUTES

CT GET LINE ATTRIBUTES (エリア; スコープ; パターン; カラー; 線幅)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	コマンドのスコープ -2=デフォルト、-1=すべて 0=選択されたオブジェクト、>0=オブジェクトID
パターン	整数	パターンインデックスを受け取る
カラー	倍長整数	カラーの値(0以上)を受け取る
線幅	実数	ポイント単位で線の太さを受け取る(0以上)

CT GET LINE ATTRIBUTESコマンドは、引数<スコープ>で記述された<エリア>内のオブジェクトに対する線属性を変数に返します。線以外のオブジェクトには、線属性がそのオブジェクトの境界線に適用されます。

引数<パターン>は、パターンパレット上で利用可能なパターンの1つを指定する1から36までの整数です。<パターン>のコードについては、付録Dを参照してください。

引数<カラー>は、オブジェクトの色を指定する倍長整数です。**CT Index to color**関数や**CT RGB to color**関数を使用することで<カラー>の値を指定できます。これらの関数の詳細については、第9章の「ユーティリティコマンド」を参照してください。

引数<線幅>は、ポイント単位で計測される線の太さです。

次の例は、選択したオブジェクトの線属性を\$Pattern、\$Color、\$Width変数に返します。

CT GET LINE ATTRIBUTES (エリア ; 0 ; \$Pattern ; \$Color ; \$Width)

参照 : CT SET LINE ATTRIBUTES、CT GET CHART LINE ATTRIBUTES

注 : このコマンドは「線」ツールや**CT Draw line**関数を使用して文書に追加された線の属性を取得するために使用してください。グリッド線などのチャート線属性を取得するには、第5章の「チャートコマンド」で説明するコマンドを使用してください。

CT SET LINE ATTRIBUTES

CT SET LINE ATTRIBUTES (エリア; スコープ; パターン; カラー; 線幅)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	コマンドのスコープ -2=デフォルト、-1=すべて 0=選択されたオブジェクト、>0=オブジェクトID
パターン	整数	パターン(0から36まで)、-1=変更なし
カラー	倍長整数	カラーの値(0以上)、-1=変更なし
線幅	実数	線の幅(ポイント単位)、-1=変更なし

CT SET LINE ATTRIBUTESコマンドは、引数<スコープ>で記述された<エリア>内のオブジェクトに対する線属性を変更します。線以外のオブジェクトには、線属性がそのオブジェクトの境界線に適用されます。

引数<パターン>は、パターンパレット上で利用可能なパターンの1つを指定する1から36までの整数です。<パターン>のコードについては、付録Dを参照してください。

引数<カラー>は、オブジェクトの色を指定する倍長整数です。**CT Index to color**関数や**CT RGB to color**関数を使用することで<カラー>の値を指定できます。これらの関数の詳細については、第9章の「ユーティリティコマンド」を参照してください。

引数<線幅>は、ポイント単位で計測される線の太さです。

次の例は、選択されたオブジェクトの線属性を実線、青、3ポイントに設定します。

CT SET LINE ATTRIBUTES (エリア ; 0 ; 3 ; **CT Index to color** (6) ; 3)

参照 : CT GET LINE ATTRIBUTES、CT SET CHART LINE ATTRIBUTES

注 : このコマンドは「線」ツールや**CT Draw line**関数を使用して文書に追加された線の属性を取得するために使用してください。グリッド線などのチャート線属性を取得するには、第5章の「チャートコマンド」で説明するコマンドを使用してください。

CT GET FILL ATTRIBUTES

CT GET FILL ATTRIBUTES (エリア; スコープ; パターン; カラー)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	コマンドのスコープ -2=デフォルト、-1=すべて 0=選択されたオブジェクト、>0=オブジェクトID
パターン	整数	パターン番号(1から36)を受け取る
カラー	倍長整数	カラーの値を受け取る

CT GET FILL ATTRIBUTESコマンドは、引数<スコープ>で記述される<エリア>内のオブジェクトに対する塗りつぶし属性を変数に返します。塗りつぶし属性はオブジェクトの内部によって決定されます。

引数<パターン>は、パターンパレット上で利用可能なパターンの1つを指定する1から36までの整数です。<パターン>のコードについては、付録Dを参照してください。

引数<カラー>は、オブジェクトの色を指定する倍長整数です。**CT Index to color**関数や**CT RGB to color**関数を使用することで<カラー>の値を指定できます。これらの関数の詳細については、第9章の「ユーティリティコマンド」を参照してください。

次の例は、選択されたオブジェクトの塗りつぶし属性を\$Pattern変数と\$Color変数に返します。

CT GET FILL ATTRIBUTES (エリア ; 0 ; \$Pattern ; \$Color)

参照 : CT SET FILL ATTRIBUTES、CT GET CHART FILL ATTRIBUTES

注 : このコマンドは、この章にある描画ツールや描画関数を使用して文書に追加されたオブジェクトの属性を取得するために使用してください。系列カラムなどのチャートオブジェクト属性を取得するには、第5章の「チャートコマンド」で説明するコマンドを使用してください。

CT SET FILL ATTRIBUTES

CT SET FILL ATTRIBUTES (エリア; スコープ; パターン; カラー)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	コマンドのスコープ -2=デフォルト、-1=すべて 0=選択されたオブジェクト、>0=オブジェクトID
パターン	整数	パターン番号(1から36)を受け取る
カラー	倍長整数	カラーの値を受け取る

CT SET FILL ATTRIBUTESコマンドは、引数<スコープ>で記述される<エリア>内のオブジェクトに対する塗りつぶし属性を変更します。塗りつぶし属性はオブジェクトの内部によって決定されます。

引数<パターン>は、パターンパレット上で利用可能なパターンの1つを指定する1から36までの整数です。<パターン>のコードについては、付録Dを参照してください。

引数<カラー>は、オブジェクトの色を指定する倍長整数です。**CT Index to color**関数や**CT RGB to color**関数を使用することで<カラー>の値を指定できます。これらの関数の詳細については、第9章の「ユーティリティコマンド」を参照してください。

次の例は、選択されたオブジェクトの塗りつぶし属性を黄色の塗りつぶしに設定します。

CT SET FILL ATTRIBUTES (エリア; 0; 3; **CT Index to color** (2))

参照 : CT GET FILL ATTRIBUTES、CT SET CHART FILL ATTRIBUTES

注 : このコマンドは、この章にある描画ツールや描画関数を使用して文書に追加されたオブジェクトの属性を設定するために使用してください。系列カラムなどのチャートオブジェクト属性を設定するには、第5章の「チャートコマンド」で説明するコマンドを使用してください。

CT SELECT

CT SELECT(エリア; スコープ; 動作)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	コマンドのスコープ -2=デフォルト、-1=すべて
動作	整数	0=選択されたオブジェクト、>0=オブジェクトID オブジェクトを選択するか、選択解除するか？ 0=選択解除、1=選択、2=切り替え

CT SELECTコマンドは、引数<スコープ>で記述された<エリア>にあるオブジェクトの選択または選択解除を行います。

<スコープ>が-1の場合には、**CT SELECT**コマンドは文書内の全オブジェクトを対象にします。

<スコープ>が0の場合には、**CT SELECT**コマンドは選択されたオブジェクトを対象にします。

<スコープ>が0よりも大きな場合には、それが特定オブジェクトのIDと同じである必要があり、そのオブジェクトが対象になります。オブジェクトが存在しない場合には、**CT SELECT**コマンドは何も行いません。

<スコープ>で記述されたオブジェクトは、引数<動作>に従って選択または選択解除されます。<動作>が0の場合には、<スコープ>で記述されたオブジェクトの選択は解除されます。<動作>が1の場合には、<スコープ>で記述されたオブジェクトが選択されます。<動作>が2の場合には、オブジェクトの現在の状態が切り替わります。つまり、選択されたものは選択が解除され、選択解除されているものは選択されます。

<スコープ>以外のオブジェクトは、**CT SELECT**コマンドの対象にはなりません。つまり、<エリア>ですでに選択されており、<スコープ>で指定されないオブジェクトは、選択されたままになります。

次の例は、文書の全オブジェクトの選択を解除してから、ID番号が1であるオブジェクトを選択します。

```
CT SELECT (エリア ; -1 ; 0)
```

```
CT SELECT (エリア ; 1 ; 1)
```

参照 : CT Get ID

CT MOVE

CT MOVE(エリア; スコープ; 左; 上)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	コマンドのスコープ -1=すべて、0=選択されたオブジェクト >0=オブジェクトID
左	実数	新しい左端の座標
上	実数	新しい上端の座標

CT MOVEコマンドは、引数<スコープ>で記述された<エリア>にあるオブジェクトの位置を変更します。

<スコープ>が-1の場合には、**CT MOVE**コマンドは文書内の全オブジェクトの位置を変更します。

<スコープ>が0の場合には、**CT MOVE**コマンドは選択されたオブジェクトの位置を変更します。

<スコープ>が0よりも大きな場合には、それが特定オブジェクトのIDと同じである必要があり、そのオブジェクトの位置を変更します。

<スコープ>で記述されたオブジェクトは、現在の原点からのオフセットとして指定される引数<左>と<上>に従って移動されます。

次の例は、選択されたオブジェクトをチャートエリアの左上隅に移動します。

CT MOVE (エリア; 0; 0; 0)

参照 : CT Get ID, CT SIZE

CT SIZE

CT SIZE(エリア; スコープ; 幅; 高さ)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	コマンドのスコープ -1=すべて、0=選択されたオブジェクト >0=オブジェクトID
幅	実数	新しい幅 (ポイント単位) -1=変更なし
高さ	実数	新しい高さ (ポイント単位) -1=変更なし

CT SIZEコマンドは、引数<スコープ>で<エリア>に記述されたオブジェクトのサイズを変更します。オブジェクトのサイズを変更するときに、オブジェクトの左上隅は固定されています。

<スコープ>が-1の場合には、**CT SIZE**コマンドは文書内の全オブジェクトのサイズを変更します。

<スコープ>が0の場合には、**CT SIZE**コマンドは選択されたオブジェクトのサイズを変更します。

<スコープ>が0よりも大きな場合には、それは特定オブジェクトのIDである必要があり、そのオブジェクトのサイズが変更されます。

<スコープ>で記述されたオブジェクトは、ポイント単位で指定される引数<幅>と<高さ>に従ってサイズが変更されます。

次の例は、IDが5であるオブジェクトのサイズを変更します。

CT SIZE (エリア ; 5 ; 10 ; 10)

参照 : CT Get ID、CT MOVE

CT Count

CT Count(エリア; スコープ) 整数

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	コマンドのスコープ -1=すべて、0=選択されたオブジェクト >0=グループID

CT Count関数は、引数<スコープ>で指定された<エリア>にオブジェクトの数を返します。

<スコープ>が-1の場合には、**CT Count**関数はグループ内にはない文書中のオブジェクトの数を返します。グループは単一のオブジェクトになります。

<スコープ>が0の場合には、**CT Count**関数はグループ内にはない現在選択されているオブジェクトの数を返します。グループは単一のオブジェクトになります。

<スコープ>0よりも大きな場合には、それはグループ用のIDである必要があり、**CT Count**関数はそのグループ内部のオブジェクト数を返します。この構文で、グループを解体せずにグループ内のオブジェクトについての情報を取得できます。ネストしているグループは、再度**CT Count**関数をコールして調査できます。

次の例は、現在選択されているオブジェクトの数を表示する警告ボックスをオープンします。

```
$Count := CT Count (エリア ; 0)  
ALERT ( String ($Count) + " 個のオブジェクトが選択されました。")
```

参照 : CT Get ID

CT ALIGN

CT ALIGN(エリア; スコープ; 水平; 垂直)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	コマンドのスコープ -1=すべて、0=選択されたオブジェクト
水平	整数	0=なし、1=左、2=中、3=右
垂直	整数	0=なし、1=上、2=中、3=下




CT ALIGNコマンドは、引数<スコープ>で記述された<エリア>にオブジェクトを整列させます。

<スコープ>が-1の場合には、CT ALIGNコマンドは文書内の全オブジェクトを整列させます。

<スコープ>が0の場合には、CT ALIGNコマンドは選択されたオブジェクトを整列させます。




<スコープ>で記述されたオブジェクトは、引数<水平>と<垂直>に従って配列されます。

以下は、引数<水平>用の値、その効果の内容、「オブジェクト整列」ダイアログボックスでの対応するアイコンを示しています。

値	整列	アイコン
0	なし	
1	左揃え	
2	中揃え	
3	右揃え	

<水平>が0の場合には、CT ALIGNコマンドは横方向の整列は行いません。<水平>が1の場合には、CT ALIGNコマンドは左側に整列させます。<水平>が2の場合は、CT ALIGNコマンドは横方向の中央に整列させます。<水平>が3の場合は、CT ALIGNコマンドは右側に整列させます。

以下は、引数<垂直>用の値、その効果の内容、「オブジェクト整列」ダイアログボックスでの対応するアイコンを示しています。

値	整列	アイコン
0	なし	
1	上揃え	
2	中揃え	
3	下揃え	

<垂直>が0の場合には、**CT ALIGN**コマンドは横方向の整列は行いません。<垂直>が1の場合には、**CT ALIGN**コマンドは上側に整列させます。<水平>が2の場合は、**CT ALIGN**コマンドは縦方向の中央に整列させます。<垂直>が3の場合は、**CT ALIGN**コマンドは下側に整列させます。

次の例は、選択したオブジェクトを縦横両方の中央に整列させます。

CT ALIGN (エリア ; 0 ; 2 ; 2)

参照 : CT Get ID

CT GET HIGHLIGHT

CT GET HIGHLIGHT (エリア; 先頭; 最終)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
先頭	整数	先頭の文字の位置から1を引いたものを受け取る
最終	整数	最終の文字の位置を受け取る

CT GET HIGHLIGHTコマンドは、引数<エリア>内で反転表示されているテキストの文字位置を<先頭>変数と<最終>変数に返します。

引数<先頭>は反転表示された先頭の文字の位置から1を引いたものであり、<最終>は反転表示された最終の文字です。<先頭>と<最終>が同じ場合には、反転表示されている文字はないので、挿入点は<先頭>と<先頭>+1の間になります。

反転表示されるテキストを持つのは一度に1つのオブジェクトだけなので、引数<スコープ>は必要ありません。<エリア>に反転表示されているテキストが存在しない場合には、**CT GET HIGHLIGHT**コマンドは、<先頭>と<最終>に-32000を返します。

次の例は、反転表示されたテキストの位置を返し、テキストが選択されていない場合にはユーザに警告を表示します。

```

CT GET HIGHLIGHT (エリア ; $First ; $Last)
If (CT Error=46)
  ALERT ("反転表示されているテキストがありません。")
End if

```

参照 : CT SET HIGHLIGHT

CT SET HIGHLIGHT

CT SET HIGHLIGHT (エリア; スコープ; 先頭; 最終)

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	-1=文書内の先頭のオブジェクト 0=セレクション内の先頭のオブジェクト >0=オブジェクトID
先頭	整数	先頭の文字の位置から1を引いたもの
最終	整数	最終の文字の位置

CT SET HIGHLIGHTコマンドは、引数<スコープ>で記述された<エリア>にあるテキストオブジェクト内の文字を反転表示します。

<スコープ>が-1の場合には、**CT SET HIGHLIGHT**コマンドは文書の最初のオブジェクトにある文字を反転表示します。

<スコープ>が0の場合には、**CT SET HIGHLIGHT**コマンドは最初に選択されたオブジェクトの文字を反転表示します。

<スコープ>が0よりも大きな場合は、それが特定テキストオブジェクトのIDと同じである必要があり、そのテキストオブジェクト内部の文字が反転表示されます。オブジェクトが存在しない場合は、**CT SET HIGHLIGHT**コマンドは何も行いません。

CT SET HIGHLIGHTコマンドでは、<スコープ>で記述されたオブジェクトが<エリア>で選択される唯一のオブジェクトです。

<スコープ>で記述されたオブジェクトがテキストオブジェクトではない場合には、**CT SET HIGHLIGHT**コマンドは何も行いません。

引数<先頭>と<最終>は、どの文字を反転表示するのかを決定します。<先頭>は反転表示される先頭の文字位置から1を引いたものです。<最終>は反転表示される最終の文字の位置です。<先頭>と<最終>が同じ場合には、選択されている文字はないので、挿入点は<先頭>と<先頭>+1の間になります。Lastがテキストオブジェクトの文字数よりも大きい場合には、**CT SET HIGHLIGHT**コマンドはテキストオブジェクトの最終まで文字を反転表示します。

CT SET HIGHLIGHTコマンドは参照の一部だけの反転表示は行いません。参照の任意の部分が反転表示されると、**CT SET HIGHLIGHT**コマンドは参照全体を含めるように反転表示を調整します。

次の例は、選択されたテキストオブジェクトのテキストを取得し、「4th Dimension」という名前を検索します。「4th Dimension」が見つかった場合には、反転表示されて太字（ボールド）になります。

```
$Find := Position ("4th Dimension" ; $Text)
If ($Find # 0)
CT SET HIGHLIGHT (エリア ; 0 ; $Find - 1 ; $Find + 12)
CT SET TEXT ATTRIBUTES (エリア ; -3 ; -1 ; -1 ; 1 ; -1 ; -1)
End if
```

参照 : CT GET HIGHLIGHT

CT INSERT FIELD

CT INSERT FIELD (エリア; スコープ; 先頭; 最終; テーブル; フィールド; {フォーマット})

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	-1=文書内の先頭のオブジェクト 0=セレクション内の先頭のオブジェクト >0=オブジェクトID
先頭	整数	先頭の文字の位置から1を引いたもの
最終	整数	最終の文字の位置
テーブル	整数	参照のテーブル番号
フィールド	整数	参照のフィールド番号
フォーマット	文字列	参照のフォーマット

CT INSERT FIELDコマンドは、引数<スコープ>で記述された<エリア>にあるテキストオブジェクトにフィールド参照を挿入します。

<スコープ>が-1の場合には、**CT INSERT FIELD**コマンドは文書の最初のオブジェクトに参照を挿入します。

<スコープ>が0の場合には、**CT INSERT FIELD**コマンドは最初に選択されたオブジェクトに参照を挿入します。

<スコープ>が0よりも大きな場合には、それが特定テキストオブジェクトのIDと同じである必要があり、参照はそのテキストオブジェクト内部に挿入されます。オブジェクトが存在しない場合には、**CT INSERT FIELD**コマンドは何も行いません。

<スコープ>で記述されたオブジェクトがテキストオブジェクトではない場合には、**CT INSERT FIELD**コマンドは何も行いません。

引数<先頭>と<最終>は、参照が挿入される場所を決定します。<先頭>は置き換えられる先頭の文字の位置から1少ないものであり、<最終>は置き換えられる最終の文字位置です。<先頭>がLastと同じ場合には、文字は置き換えられず、参照は<先頭>と<先頭>+1の間に挿入されます。<最終>がテキストオブジェクトにある文字の数よりも大きい場合には、**CT INSERT FIELD**コマンドはテキストオブジェクトの<先頭>から最終の文字まで文字を置き換えます。

<先頭>と<最終>は参照されるフィールドを決定します。<テーブル>はテーブルの番号であり、<フィールド>はフィールドの番号です。テーブルとフィールドは作成された順に番号付けされています。

オプション引数の<フォーマット>は、参照用の表示フォーマットです。このオプションは、「フォーマット」ダイアログボックスからフォーマットを選択するのと同じです。フォーマットはその番号またはその名前から参照されます。フォーマットは、「フォーマット」ダイアログボックスのリストに表示される順に番号付けされています。

<フォーマット>が1桁か2桁の文字列の場合には、<フィールド>に適用されたフォーマットはリストからのものです。<フォーマット>が1桁か2桁の文字列ではない場合には、リスト内の各フォーマットのテキスト値と比較されます。リスト内の値のどれかと一致した場合には、そのフォーマットが適用されます。つまり、最初の日付フォーマットは「19」または「Short」のどちらかで参照できるということです。

<フォーマット>がフォーマットのリストにない場合は、カスタム数値フォーマットとして解釈されます。<フォーマット>が参照の結果の値としては不適切である場合には、無効になります。例えば、数値に対して日付フォーマットを使用した場合には、その数値はフォーマットされずに表示されます。

次の例は、IDが1であるテキストオブジェクトに最初のテーブルの最初のフィールドへの参照を挿入し、オブジェクトにある任意のテキストを置き換えてから、リストの11番目のフォーマットに従ってフォーマットを行います。

CT INSERT FIELD (エリア ; 1 ; 0 ; 32000 ; 1 ; 1 ; "11")

4th Dimensionの**Field**関数と**Table**関数を使用すると、フィールドやテーブルの番号を判断できます。これによって、コードは読みやすくなります。例えば、前述の例で使用されたフィールドが[顧客]名前である場合、コードは以下のようになります。

CT INSERT FIELD (エリア ; 1 ; 0 ; 32000 ; **Table**(>>[顧客]) ; **Field**(>>名前) ; "11")

参照 : CT INSERT EXPRESSION

CT INSERT EXPRESSION

CT INSERT EXPRESSION (エリア; スコープ; 先頭; 最終; 式; {フォーマット})

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
スコープ	倍長整数	-1=文書内の先頭のオブジェクト 0=セクション内の先頭のオブジェクト >0=オブジェクトID
先頭	整数	先頭の文字の位置から1を引いたもの
最終	整数	最終の文字の位置
式	文字列	式
フォーマット	文字列	式のフォーマット

CT INSERT EXPRESSION コマンドは、引数 <スコープ> で記述された <エリア> にある <式> への参照を挿入します。

<スコープ> が -1 の場合には、CT INSERT EXPRESSION コマンドは文書の最初のオブジェクトに参照を挿入します。

<スコープ> が 0 の場合には、CT INSERT EXPRESSION コマンドは最初に選択されたオブジェクトに参照を挿入します。

<スコープ> が 0 よりも大きな場合には、それが特定テキストオブジェクトのIDと同じである必要があり、参照はそのテキストオブジェクト内部に挿入されます。オブジェクトが存在しない場合には、CT INSERT EXPRESSION コマンドは何も行いません。

<スコープ> で記述されたオブジェクトがテキストオブジェクトではない場合には、CT INSERT EXPRESSION コマンドは何も行いません。

引数 <先頭> と <最終> は、参照が挿入される場所を決定します。<先頭> は置き換えられる先頭の文字の位置から1少ないものであり、<最終> は置き換えられる最終の文字位置です。<先頭> が Last と同じ場合には、文字は置き換えられず、参照は <先頭> と <先頭> +1 の間に挿入されます。<最終> がテキストオブジェクトにある文字の数よりも大きい場合には、CT INSERT EXPRESSION コマンドはテキストオブジェクトの <先頭> から最終の文字まで文字を置き換えます。

引数 <式> は値を返す正しい 14th Dimension 式であるテキストです。<式> は、フィールド、変数、4th Dimension 関数、ユーザ定義関数(プロジェクトメソッド)、外部ルーチン、ステートメントのいずれかへの参照になります。

次の表は、それぞれの式タイプの例を示しています。

例	式のタイプ
[描画]オブジェクト	フィールド
vCriteria	変数
Current date	4 th Dimension関数
GetNum	ユーザ定義関数 (プロジェクトメソッド)
CT Count	4D Chart 関数
3 * "こんにちは"	ステートメント

オプション引数の<フォーマット>は、参照用の表示フォーマットです。このオプションは、「フォーマット」ダイアログボックスからフォーマットを選択するのと同じです。フォーマットはその番号またはその名前から参照されます。フォーマットは、「フォーマット」ダイアログボックスのリストに表示される順に番号付けされています。

<フォーマット>が1桁か2桁の文字列の場合には、<フィールド>に適用されたフォーマットはリストからのものです。<フォーマット>が1桁か2桁の文字列ではない場合には、リスト内の各フォーマットのテキスト値と比較されます。リスト内の値のどれかと一致した場合には、そのフォーマットが適用されます。つまり、最初の日付フォーマットは「19」または「Short」のどちらかで参照できるということです。

<フォーマット>がフォーマットのリストにない場合は、カスタム数値フォーマットとして解釈されます。<フォーマット>が参照の結果の値としては不適切である場合には、無効になります。例えば、数値に対して日付フォーマットを使用した場合には、その数値はフォーマットされずに表示されます。

次の例は、新しいテキストオブジェクトを作成し、それに4th Dimension関数の**Current date**への参照を入れ、Long日付フォーマットを使用してフォーマットします。

```
$ID := CT Draw text (エリア ; 0.5 ; 0.5 ; 3.5 ; 1 ; "今日は : ")  
CT INSERT EXPRESSION (エリア ; $ID ; 32000 ; 32000 ; "Current date" ; "Long")
```

参照 : CT INSERT FIELD

次のコマンドと関数は、「ルーチン」エリアの「Printing」テーマ内にあります：

CT PRINT

CT PRINT MERGE

これらのコマンドによって、さまざまな方法で4D Chartエリアを印刷することができます。

CT PRINT

CT PRINT (エリア；キャンセル可；{プリントダイアログ})

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
キャンセル可	整数	印刷のキャンセルを許可するか？ 0=印刷処理のキャンセルを許可しない 1=印刷処理のキャンセルを許可する
プリントダイアログ	整数	「プリント」ダイアログボックスを表示するか？ 0=ダイアログボックスを表示する 1=ダイアログボックスを表示しない

CT PRINTコマンドは、引数<エリア>内の文書を印刷します。**CT PRINT**コマンドをコールすることは、「プリント設定」ダイアログボックスがユーザに提示されない以外は、「ファイル」メニューから「プリント...」を選択することに類似した関数を実行します。印刷前に「プリント設定」ダイアログボックスを表示するには、**CT DO COMMAND**コマンドを使用します。

引数<キャンセル可>が1の場合には、4D Chartはユーザが「Ctrl (Macintosh版では、コマンド) - ピリオド」を押すことによって印刷をキャンセルできるダイアログボックスを表示します。ユーザが印刷をキャンセルした場合には、**CT Error**関数はエラー番号 20を返します。<キャンセル可>が0の場合には、このダイアログボックスは表示されず、ユーザは印刷をキャンセルできません。

オプション引数の<プリントダイアログ>が0の場合は、標準の「プリント」ダイアログボックスは表示されず、印刷ジョブが即座に開始されます。<プリントダイアログ>が1である場合には、標準の「プリント」ダイアログボックスが表示されます。

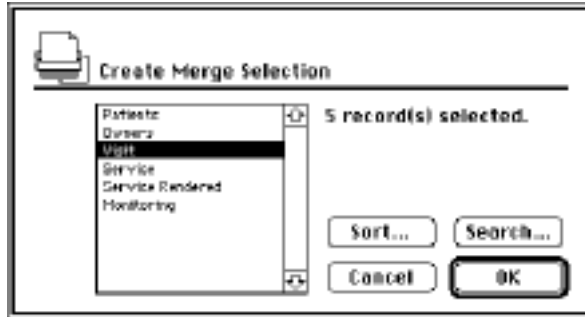
CT PRINT MERGE

CT PRINT MERGE (エリア; テーブル番号; キャンセル可; {プリントダイアログ})

引数	タイプ	説明
エリア	倍長整数	4D Chartエリア
テーブル番号	整数	テーブル番号
キャンセル可	整数	印刷のキャンセルを許可するか? 0=印刷処理のキャンセルを許可しない 1=印刷処理のキャンセルを許可する
プリントダイアログ	整数	「プリント」ダイアログボックスを表示するか? 0=ダイアログボックスを表示する 1=ダイアログボックスを表示しない

このコマンドは、引数<テーブル番号>のカルレントセクション用にプリントマージを実行できます。プリントマージ用に使用された文書は、<エリア>で指定されます。

<テーブル番号>が0の場合には、「マージセクション作成」ダイアログボックスが表示されます。



引数<キャンセル可>が1の場合には、4D Chartはユーザが「Ctrl (Macintosh版では、コマンド) - ピリオド」を押すことによって印刷をキャンセルできるダイアログボックスを表示します。ユーザが印刷をキャンセルした場合には、**CT Error**関数はエラー番号 20を返します。<キャンセル可>が0の場合には、このダイアログボックスは表示されず、ユーザは印刷をキャンセルできません。

オプション引数の<プリントダイアログ>が0の場合は、標準の「プリント」ダイアログボックスは表示されず、印刷ジョブが即座に開始されます。<プリントダイアログ>が1である場合には、標準の「プリント」ダイアログボックスが表示されます。

次の例は、4D Chartエリアが「値表示」モード用に設定され、プリントマージを実行することを確認する例です。

`プリントマージ用の選択を生成する

ALL RECORDS ([MyFile])

`表示モードが「値表示」に設定されていることを確認する

CT MENU STATUS (エリア ; 6006 ; \$Checked ; \$Available ; \$Name)

If (\$Name="値表示")

CT DO COMMAND (エリア ; 6006) `表示を「値表示」モードに設定する

End if

`ユーザの操作なしにプリントマージを実行する

CT PRINT MERGE (エリア ; Table(>>[MyFile]) ; 0 ; 0)

次のコマンドと関数は、「ルーチン」エリアの「Utilities」テーマ内にあります：

CT Font name	CT RGB to color
CT Font number	CT COLOR TO RGB
CT Clipboard to picture	CT Index to color
CT PICTURE TO CLIPBOARD	CT Color to index

このテーマのコマンドと関数によって、ユーザはフォント、カラー、クリップボードを扱うことができます。

CT Font name

CT Font name (フォント番号) 文字列

引数	タイプ	説明
フォント番号	整数	フォントID番号

CT Font name関数は、IDが<フォント番号>であるフォントの名前を返します。

引数<フォント番号>が存在しない場合には、**CT Font name**関数は空の文字列を返しません。

次の例は、IDが3のフォントの名前を返します。

```
vName:= CT Font name (3)
```

参照：CT Font number

CT Font number

CT Font number (フォント名) 整数

引数	タイプ	説明
フォント名	文字列	フォントの名前

CT Font number関数は、名前が<フォント名>であるフォントのIDを返します。

次の例は、フォント「Times」の数値を返します。

```
vNumber:= CT Font number("Times")
```

参照 : CT Font name

CT Clipboard to picture

CT Clipboard to picture ピクチャ

CT Clipboard to picture関数は、クリップボードの内容のコピーである4th Dimensionピクチャを返します。

クリップボードにピクチャがない場合には、**CT Clipboard to picture**関数は空のピクチャを返します。

次の例は、クリップボードから「vPict」変数にピクチャをコピーします。

```
vPict:= CT Clipboard to picture
```

参照 : CT PICTURE TO CLIPBOARD

CT PICTURE TO CLIPBOARD

CT PICTURE TO CLIPBOARD (ピクチャ)

引数	タイプ	説明
ピクチャ	ピクチャ	ピクチャ

CT PICTURE TO CLIPBOARDコマンドは、引数<ピクチャ>をクリップボードにコピーします。

クリップボード上にコピーされると、ピクチャがペーストできるところならどこにでも<ピクチャ>を貼り付けることができます。

次の例は、“ [チャート]オブジェクト ” フィールドの内容をクリップボードにコピーします。

CT PICTURE TO CLIPBOARD ([チャート]オブジェクト)

参照 : CT Clipboard to picture

4D Chartコマンドでカラーを指定する

4D Chartでカラーを指定するには、次の3種類の方法があります。

RGB : 赤、緑、青のカラー(3つの倍長整数)。

インデックス : 4th Dimensionパレットの1から256までのカラーで、モニタが256色に設定されていれば、「塗りつぶしカラー」、「線カラー」、「テキストカラー」の各サブメニューで表示されます(1つの倍長整数)。

カラー : 4D Chart内部で使用される値(1つの倍長整数)。

この節で説明するユーティリティコマンドは、これらの3種類の表記の間でカラーの値を変換します。第5章と第7章(「チャートコマンド」と「オブジェクトコマンド」)のコマンドは、倍長整数タイプの<カラー>を引数として想定しています。

CT RGB to color

CT RGB to color (赤 ; 緑 ; 青) 倍長整数

引数	タイプ	説明
赤	倍長整数	赤の値 (0から65535)
緑	倍長整数	緑の値 (0から65535)
青	倍長整数	青の値 (0から65535)

CT RGB to color関数は、引数<赤>、<緑>、<青>のコンポーネントを表す値を返します。この数値は、いくつかの4D Chartコマンドで使用されます。

引数<赤>、<緑>、<青>は、システムのカラーピッカーで使用されているものと同じです。

次の表は、一般的に使用される3つのカラーでの<赤>、<緑>、<青>の値を示しています：

Color	Red	Green	Blue
Red	56683	2242	1698
Green	0	32768	4528
Blue	0	0	54272

次の例は、赤用の色変数を設定します。

```
$color:= CT RGB to color(56683 ; 2242 ; 1698)
```

参照 : CT COLOR TO RGB、CT Color to index、CT Index to color

CT COLOR TO RGB

CT COLOR TO RGB (カラー ; 赤 ; 緑 ; 青)

引数	タイプ	説明
カラー	倍長整数	カラーの値(4D Chart内部の値)
赤	倍長整数	赤の値を受け取る (0から65535)
緑	倍長整数	緑の値を受け取る (0から65535)
青	倍長整数	青の値を受け取る (0から65535)

CT COLOR TO RGBコマンドは、引数<カラー>にあるカラーの値を分解して、<赤>、<緑>、<青>のコンポーネントを表す変数に値を返します。

次の例は、カラーの値用のコンポーネントRGB値を取得し、それらをいくつかのローカル変数に配置します。

```
$color:=100000
CT COLOR TO RGB($color ; $red ; $green ; $blue)
```

参照 : CT RGB to color、CT Color to index、CT Index to color

CT Index to color

CT Index to color (インデックス) 倍長整数

引数	タイプ	説明
インデックス	整数	パレットインデックス

CT Index to color関数は、引数<インデックス>で記述されたカラーを返します。

<インデックス>は4th Dimensionカラーパレットに特定の色を指定する整数です。4th Dimensionパレット上のカラーは、1から256までの番号がついています。

CT Index to color関数は、個別のコンポーネントを知らなくても色を指定できる便利な方法です。

次の例は、「vColor」変数にシアンの色(パレットインデックス番号8)の色の値を配置します。

```
vColor:= CT Index to color (8)
```

参照 : CT Color to index、CT RGB to color、CT COLOR TO RGB

CT Color to index

CT Color to index (カラー) 整数

引数	タイプ	説明
カラー	整数	カラーの値(4D Chart内部の値)

CT Color to index関数は、4th Dimension カラーパレット上に引数 <カラー> に最も近い色のインデックスを返します。

例えば、特定の青の影が指定された場合に、**CT Color to index**関数は4th Dimensionパレットに最も近い青の影を返します。4th Dimensionパレット上の色は1から256までの番号が付けられます。

次の例は、赤に最も近い色のパレットインデックスを「vColor」変数に配置します。

```
vColor:= CT Color to index (CT RGB to color (56683 ; 2242 ; 1698))
```

参照 : CT Index to color、CT RGB to color、CT COLOR TO RGB

多角形の描画

操作	Windows	MacOS
最後の頂点を削除する	Delete	delete
1番目の頂点で多角形を閉じる	Alt - Enter	option - enter
多角形を終了する	Enter	enter

テキストの入力

操作	Windows	MacOS
カレントテーブルのフィールドを表示する	Alt - クリック	option - クリック
すべてのテーブルとフィールドを表示する	Shift - Alt - クリック	shift - option- クリック
テキスト入力モードと選択テキストオブジェクトを終了する	Enter	enter

選択オブジェクトの移動

操作	Windows	MacOS
1ピクセル上に移動する		
1ピクセル下に移動する		
1ピクセル左に移動する		
1ピクセル右に移動する		

オブジェクトサイズの変更

操作	Windows	MacOS
1ピクセル高さを低くする	Ctrl -	コマンド -
1ピクセル高さを高くする	Ctrl -	コマンド -
1ピクセル幅を狭くする	Ctrl -	コマンド -
1ピクセル幅を広くする	Ctrl -	コマンド -

オブジェクトの選択

選択範囲	Windows	MacOS
1つのオブジェクト	クリック	クリック
1つのオブジェクト(または選択解除)	Shift - クリック	shift - クリック
マーカー内の全オブジェクト	クリック - ドラッグ	クリック - ドラッグ
マーカー内部およびマーカーで交差された全オブジェクト	Ctrl - クリック - ドラッグ	コマンド - クリック - ドラッグ

メニューコマンドのショートカット

メニューコマンド	Windows	MacOS
新規	Ctrl - Shift - N	コマンド - shift - N
開く...	Ctrl - Shift - O	コマンド - shift - O
保存...	Ctrl - Shift - S	コマンド - shift - S
プリント	Ctrl - Shift - P	コマンド - shift - P
フルウィンドウ/フォームに戻る	Ctrl - Shift - Q	コマンド - shift - Q
すべてを選択	Ctrl - Shift - A	コマンド - shift - A
コピー	Ctrl - Shift - C	コマンド - shift - C
貼り付け	Ctrl - Shift - V	コマンド - shift - V
切り取り	Ctrl - Shift - X	コマンド - shift - X
複製	Ctrl - Shift - D	コマンド - shift - D
取り消し	Ctrl - Shift - Z	コマンド - shift - Z
メニューバー表示	Ctrl - Shift - M	コマンド - shift - M
オブジェクトツール表示	Ctrl - Shift - T	コマンド - shift - T
スクロールバー表示	Ctrl - Shift - B	コマンド - shift - B
グループ化	Ctrl - Shift - G	コマンド - shift - G
グループ解除	Ctrl - Shift - U	コマンド - shift - U
オブジェクト整列	Ctrl - Shift - L	コマンド - shift - L
最前面へ	Ctrl - Shift -]	コマンド - shift -]
最背面へ	Ctrl - Shift - [コマンド - shift - [
前面へ	Ctrl - Shift - '	コマンド - shift - '
背面へ	Ctrl - Shift - ;	コマンド - shift - ;

次の表は、引数<コマンド>用のコードです：

コマンド番号	メニュー	メニューコマンド
1	ツールパレット	ポインタ
2	ツールパレット	テキスト
3	ツールパレット	線
4	ツールパレット	矩形
5	ツールパレット	角の丸い
6	ツールパレット	楕円
7	ツールパレット	多角形
1001	ファイル	新規...
1002	ファイル	開く...
1003	ファイル	保存
1004	ファイル	新規保存...
1006	ファイル	テンプレートとして保存...
1008	ファイル	用紙設定...
1009	ファイル	プリント...
1010	ファイル	プリントマージ...
1012	ファイル	フルウインドウ/フォームに戻る
2001	編集	取り消し
2003	編集	切り取り
2004	編集	コピー
2005	編集	貼り付け
2006	編集	消去
2007	編集	複製
2009	編集	すべてを選択
2011	編集	プロパティ
4016	編集	表示メニュー

コマンド番号	メニュー	メニューコマンド
3001	テキスト	フォントメニュー
3002	テキスト	サイズメニュー
3003	テキスト	書体メニュー
3004	テキスト	カラーメニュー
3005	テキスト	文字揃えメニュー
4001	チャート	新規2Dチャートメニュー
4002	チャート	新規3Dチャートメニュー
4003	チャート	軸メニュー
4004	チャート	目盛線メニュー
4005	チャート	タイトルメニュー
4006	チャート	凡例メニュー
4007	チャート	凡例編集メニュー
4008	チャート	奥行きメニュー
4009	チャート	数値メニュー
4010	チャート	オプションメニュー
4012	チャート	3Dビューメニュー
4014	チャート	更新メニュー
5001	オブジェクト	塗りつぶしパターンメニュー
5002	オブジェクト	塗りつぶしカラーメニュー
5004	オブジェクト	線パターンメニュー
5005	オブジェクト	線カラーメニュー
5007	オブジェクト	線幅メニュー
5008	オブジェクト	矢印形態メニュー
5010	オブジェクト	角丸めメニュー
5012	オブジェクト	調整メニュー
6001	データベース	フィールド貼り付け...
6002	データベース	フォーマット...
6003	データベース	データ参照
6004	データベース	データ参照解除
6006	データベース	値表示 / 参照表示...
6008	データベース	ホットリンク引用...
6009	データベース	ホットリンク発行...
6010	データベース	ホットリンク引用解除...

コマンド番号	メニュー	メニューコマンド
6011	データベース	ホットリンク発行解除...
6012	データベース	ホットリンクへ追加...
7001 7999	フォント	個々のフォント名
8001 8009	サイズ	個々のフォントサイズ
8010	サイズ	その他のサイズ
9001	書体	標準
9001	書体	太字 (ボールド)
9001	書体	斜体 (イタリック)
9001	書体	下線 (アンダーライン)
9001	書体	アウトライン
9001	書体	シャドウ
10001 10999	カラー	個々のカラー
11001	文字揃え	左
11002	文字揃え	中央
11003	文字揃え	右
12001	新規2Dチャート	2D 面
12002	新規2Dチャート	2D 棒
12003	新規2Dチャート	2D 線
12004	新規2Dチャート	2D 散布図
12005	新規2Dチャート	2D 円
12006	新規2Dチャート	2D ピクチャ
12007	新規2Dチャート	2D ポーラー
13001	新規3Dチャート	3D 棒
13002	新規3Dチャート	3D 線
13003	新規3Dチャート	3D 面
13004	新規3Dチャート	3D 等高線
13005	新規3Dチャート	3D 三角形
13006	新規3Dチャート	3D ピン
14001	軸	項目軸
14002	軸	数値軸 (2Dグラフ)
		系列軸 (3Dグラフ)
14003	軸	数値軸 (3Dグラフ)

コマンド番号	メニュー	メニューコマンド
15001	目盛線	項目軸
15002	目盛線	数値軸 (2Dグラフ) 系列軸 (3Dグラフ)
15003	目盛線	数値軸 (3Dグラフ)
16001	タイトル	項目軸
16002	タイトル	数値軸 (2Dグラフ) 系列軸 (3Dグラフ)
16003	タイトル	数値軸 (3Dグラフ)
17001	表示	メニューバー
17002	表示	オブジェクトツール
17003	表示	チャートツール
17004	表示	スクロールバー
17005	表示	ルーラ
18001 18036	塗りつぶしパターン	個々の塗りつぶしパターン
19001 19256	塗りつぶしカラー	個々の塗りつぶしカラー
20001 20036	線パターン	個々の線パターン
21001 21256	線カラー	個々の線カラー
22001	線幅	極細線
22002	線幅	1ポイント
22003	線幅	2ポイント
22004	線幅	4ポイント
22005	線幅	6ポイント
22006	線幅	その他の線幅
23001	矢印形態	なし
23001	矢印形態	始点
23001	矢印形態	終点
23001	矢印形態	両端
24001	調整	最前面へ
24002	調整	最背面へ
24003	調整	前面へ
24004	調整	背面へ
24005	調整	オブジェクトの整列...
24006	調整	グループ化
24007	調整	グループ解除

次の表は、4D Chartエラーメッセージのコードです：

コード	メッセージ
1	4D Chartのエリア参照が正しくありません。
2	セグメントをロードすることができません。
3	コマンド番号が正しくありません。
4	メニューコマンドを使用できません。
5	ホットリンクが見つかりません。
6	4Dテーブル番号が正しくありません。
7	4Dフィールド番号が正しくありません。
8	ピクチャを作成できませんでした。
9	CT AREA TO AREA コマンドのスコープが正しくありません。
10	送信元と送信先エリアは異なわなければなりません。
11	スコープが正しくありません。
12	オブジェクトIDが正しくありません。
13	この操作はこのオブジェクトタイプで実行することはできません。
14	オブジェクトインデックスが正しくありません。
15	選択されたオブジェクトがありません。
16	この文書内にオブジェクトがありません。
17	ファイルタイプが正しくありません。
18	この4D Chart文書のバージョンでは、サポートされていません。
19	この4D Chart文書は最新バージョンで作成されています。
20	ユーザがダイアログボックスをキャンセルしました。
21	ホットリンクのタイプが正しくありません。
22	このホットリンクへの追加はホットリンク連鎖で反復して生成されます。
23	この操作はオブジェクトを文書外に移動しました。
24	4Dテーブルがありません。
25	ビットマップイメージが大きすぎます。
26	この操作はオブジェクトの最大番号を越えました。
27	チャートを作成するための項目数が十分ではありません。

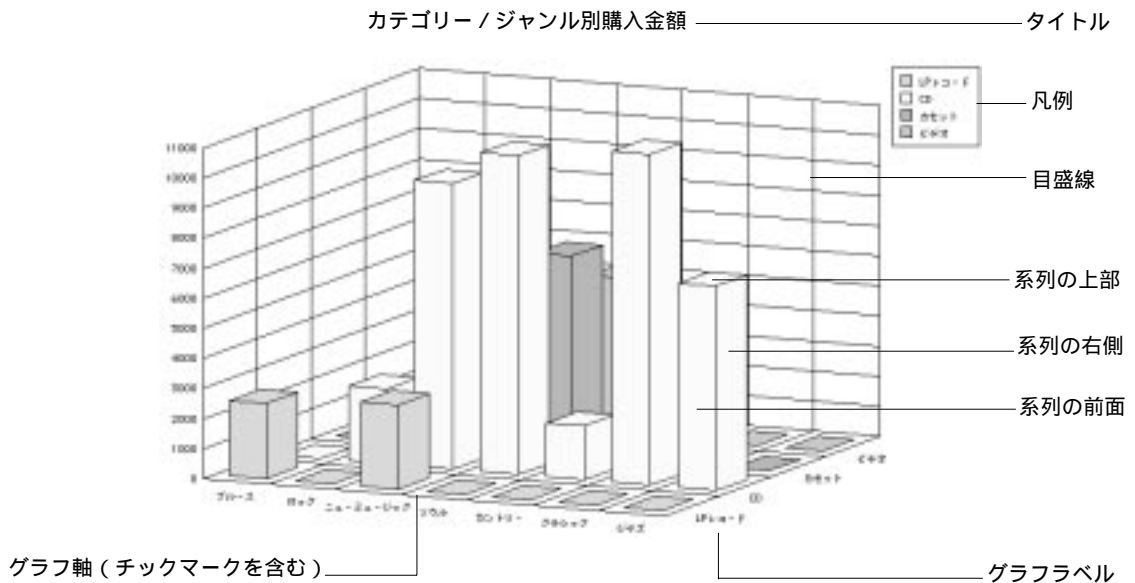
コード	メッセージ
28	チャートを作成するには項目数が多すぎます。
29	チャートを作成するための系列数が十分ではありません。
30	チャートを作成するには系列数が多すぎます。
31	チャートタイプが正しくありません。
32	チャートサイズが正しくありません。
33	項目軸に設定するフィールドタイプが正しくありません。
34	数値軸に設定するフィールドタイプが正しくありません。
35	配列タイプが正しくありません。
36	値が複数あります。
37	そのホットリンク名またはホットリンクタイプはすでに存在します。
38	少なくとも1文字必要です。
39	オブジェクト境界が正しくありません。
40	角の丸みが正しくありません。
41	RGB値が正しくありません。
42	カラー値が正しくありません。
43	カラーインデックスが正しくありません。
44	クリップボードにピクチャがありません。
45	すべての値が範囲外です。
46	テキスト編集モードではありません。
47	軸インデックスが正しくありません。
48	目盛線インデックスが正しくありません。
49	多角形の頂点数が正しくありません。
50	座標が正しくありません。
51	オブジェクトサイズが正しくありません。
52	指定されたオブジェクトにはこの属性はありません。
53	フルパス名が255バイトを越えました。
54	フィールドタイプが正しくありません。
55	この種のチャートには適用できません。
56	クリップボードからの値が正しくありません。
57	クリップボードからの次元数が正しくありません。
58	セレクション値が正しくありません。
59	プリンタリソースを初期化できませんでした。
60	オフスクリーンエリアへのメモリ割り当てが十分ではありません。
61	カラーリソースで使用するメモリが十分ではありません。
62	表示項目番号が正しくありません。
63	テキストオブジェクトをコピーするメモリが十分ではありません。
64	フォントが正しくありません。
65	選択されたレコードがありません。

ここ付録には、引数<タイプ>と<特定部分>、<パターン>、<オプション>用のコードが含まれています。

引数<タイプ>と<特定部分>のコード

グラフ軸の1つまたは凡例などのグラフの一部をプログラムで修正する場合、引数<タイプ>と<特定部分>を使って、修正するグラフの一部を指定します。

グラフの主要部分を次の図に示します：



下記は、引数<タイプ>用のコードです；

コード	グラフ要素
1	プロット矩形
2	凡例
3	軸
4	ラベル
5	タイトル
6	主目盛線
7	補助目盛線
8	系列
9	系列のラベル

下記は、引数<特定部分>用のコードです；

グラフ要素	特定部分コード
プロット矩形	2Dグラフ：0=矩形全体 3Dグラフ：1=背面、2=側面、3=下
凡例	0=凡例矩形
軸	0=項目、1=系列、2=数値
ラベル	0=項目、1=系列、2=数値
タイトル	0=項目、1=系列、2=数値
主目盛線	0=項目、1=系列、2=数値
補助目盛線	0=項目、1=系列、2=数値
系列 / 系列のラベル	(系列番号*100) + (側面番号)
側面番号 (系列側面用)	0=すべての側面 (すべての2Dグラフに使用)、1=前面 2=左、3=右、4=上、5=下

注：2Dグラフの系列は、1つの側面（前面）を持っています：3Dグラフでは、1度に3つの側面を表示することができます。

パターンコード

次の図は、塗りつぶしパターンで使用するものです。図の右側にある数字は左の図の各パターンに対応するパターンコードを示しています。例えば、左上隅（なし）はパターンコード1です。そこから右方向に順番にプラス1していき、右下隅がパターンコード36になります。引数<パターン>は、1から36までのパターンコード指定します：

なし		■	▨	▩	▧	1	2	3	4	5	6
▧	▨	▩	▧	▨	▩	7	8	9	10	11	12
▧	▨	▩	▧	▨	▩	13	14	15	16	17	18
▧	▨	▩	▧	▨	▩	19	20	21	22	23	24
▧	▨	▩	▧	▨	▩	25	26	27	28	29	30
▧	▨	▩	▧	▨	▩	31	32	33	34	35	36

グラフタイプのオプションコード

下記は、CT GET CHART OPTIONコマンドとCT SET CHART OPTIONコマンドで使用される引数<オプション>用のコードです。

グラフタイプ	要素番号	コード
2D 面	1	方向：0=垂直、1=水平
	2	積み重ね？ 0=通常、1=積み重ね、2=比率
2D 棒	1	方向：0=垂直、1=水平
	2	積み重ね？ 0=通常、1=積み重ね、2=比率
	3	重ねる比率（0から100%）
	4	間隔比率（0から100%）
2D 線	1	方向：0=垂直、1=水平
	2	積み重ね？ 0=通常、1=積み重ね、2=比率
2D 散布図	1	方向：0=垂直、1=水平
	2	積み重ね？ 0=通常、1=積み重ね、2=比率
2D 円	1	方向：0=垂直、1=水平
	2	開始角度（0から360度）

グラフタイプ	要素番号	コード
2D ピクチャ	1	方向：0=垂直、1=水平
	2	積み重ね？ 0=通常、1=積み重ね、2=比率
	3	重ねる比率（0から100%）
	4	間隔比率（0から100%）
	5	水平整列 0=左、1=中央、2=右
	6	垂直整列 0=上、1=中央、2=下
	7	水平方向表示 0=切り捨て、1=拡大縮小、2=積み重ね
	8	垂直方向表示 0=切り捨て、1=拡大縮小、2=積み重ね
2D ポーラー		なし
3D 棒	1	項目軸(X)を比率で指定：0=幅、1=間隔
	2	系列軸(Y)を比率で指定：0=幅、1=間隔
	3	項目軸(X)比率（0から100%）
	4	系列軸(Y)比率（0から100%）
	5	表示範囲：0=すべて、1=上部のみ
3D 線	1	系列軸(Y)を比率で指定：0=幅、1=間隔
	2	系列軸(Y)比率（0から100%）
3D 面	1	系列軸(Y)を比率で指定：0=幅、1=間隔
	2	系列軸(Y)比率（0から100%）
3D 等高線	1	表示範囲：0=すべて、1=上部のみ
3D 三角形	1	上下反転：0=なし、1=あり
	2	数値0も描画する：0=なし、1=あり
	3	系列軸(Y)を比率で指定：0=幅、1=間隔
	4	系列軸(Y)比率（0から100%）
3D ピン	1	ピン先頭の形：0=楕円形、1=四角形

次の表は、各項目、系列、数値のグラフ軸に割り当てることができるデータタイプの情報を示しています。

データタイプ	項目軸または系列軸	数値軸	数値軸と互換性のあるデータタイプ
文字		×	...
テキスト		×	...
実数			整数、倍長整数
整数			実数、倍長整数
倍長整数			実数、整数
日付			...
時間		×	...
ブール		×	...
ピクチャ	×	×	...
Blob	×	×	...

CT A

CT ADD TO HOT LINK (エリア; スコープ; 名前)	110
CT ALIGN(エリア; スコープ; 水平; 垂直)	137
CT AREA TO AREA (ソース; コピー先; コピーコード)	32
CT AREA TO FIELD (エリア; スコープ; テーブル; フィールド; {保存方法})	33
CT Area to picture (エリア; スコープ) ピクチャ	35
CT Array to polygon (エリア; 水平配列; 垂直配列) 倍長整数	119

CT C

CT Chart array (エリア; タイプ; サイズ; 項目配列; 系列配列; 数値配列) 数値	65
CT Chart data (エリア; タイプ; サイズ; グループ項目; グループ系列; テーブル番号; 項目フィールド; 系列フィールド; 数値フィールド) 倍長整数	69
CT Chart selection (エリア; タイプ; サイズ; グループ項目; テーブル番号; 項目フィールド; 系列 / 数値フィールド) 倍長整数	67
CT Clipboard to picture ピクチャ	150
CT Color to index (カラー) 整数	154
CT COLOR TO RGB (カラー; 赤; 緑; 青)	153
CT Count(エリア; スコープ) 整数	136

CT D

CT DELETE OFFSCREEN AREA (エリア)	40
CT DO COMMAND (エリア; コマンド)	44
CT Draw line (エリア; 左; 上; 右; 下; 矢印形態) 倍長整数	117
CT Draw oval (エリア; 左; 上; 右; 下) 倍長整数	118
CT Draw rectangle (エリア; 左; 上; 右; 下; 角の丸み) 倍長整数	118
CT Draw text (エリア; 左; 上; 右; 下; テキスト) 倍長整数	116

CT E

CT Error (メッセージ) 整数	49
CT EVENT FILTER (エリア; フィルタ)	50

CT EXPERT COMMAND (エリア ; コマンド ; ステータス)	51
CT EXPERT MODE (エリア ; モード)	52
CT EXPLODE PIE (エリア ; オブジェクト ; 項目 ; パーセンテージ)	108

CT F

CT FIELD TO AREA (エリア ; テーブル ; フィールド)	34
CT Font name (フォント番号) 文字列	149
CT Font number (フォント名) 整数	150

CT G

CT GET 3D VIEW (エリア ; オブジェクト ; 回転 ; 仰角)	98
CT GET AREA BOUNDARY (エリア ; 境界コード ; 左 ; 上 ; 右 ; 下)	41
CT GET AXIS ATTRIBUTES (エリア ; オブジェクト ; 軸 ; 補助目盛 ; 主目盛 ; 位置 ; 反転)	79
CT GET BOUNDARY (エリア ; スコープ ; 左 ; 上 ; 右 ; 下)	125
CT GET CHART FILL ATTRIBUTES (エリア ; オブジェクト ; タイプ ; 特定部分 ; パターン ; カラー)	71
CT GET CHART LINE ATTRIBUTES (エリア ; オブジェクト ; タイプ ; 特定部分 ; パターン ; カラー ; 線幅)	76
CT GET CHART OPTIONS (エリア ; オブジェクト ; オプション)	101
CT GET CHART PART (エリア ; オブジェクト ; タイプ ; 特定部分)	100
CT Get chart picture (エリア ; オブジェクト ; タイプ ; 特定部分) ピクチャ	106
CT GET CHART TEXT ATTRIBUTES (エリア ; オブジェクト ; タイプ ; 特定部分 ; フォントID ; フォントサイズ ; スタイル ; カラー)	73
CT Get chart type (エリア ; オブジェクト) 整数	103
CT GET DATE SCALE (エリア ; オブジェクト ; 最小自動 ; 最大自動 ; 主増分自動 ; 補助増分自動 ; 最小 ; 最大 ; 主増分タイプ ; 主増分 ; 補助増分タイプ ; 補助増分)	93
CT GET DEPTH (エリア ; オブジェクト ; 水平オフセット ; 垂直オフセット)	96
CT Get display (エリア ; 項目) 整数	55
CT GET DOCUMENT SIZE (エリア ; 幅 ; 高さ)	57
CT GET FILL ATTRIBUTES (エリア ; スコープ ; パターン ; カラー)	131
CT GET HIGHLIGHT (エリア ; 先頭 ; 最終)	139
CT Get ID (エリア ; スコープ ; インデックス) 倍長整数	121
CT GET LABEL ATTRIBUTES (エリア ; オブジェクト ; 軸 ; 位置 ; 方向 ; フォーマット)	81

CT GET LINE ATTRIBUTES (エリア; スコープ; パターン; カラー; 線幅)	129
CT GET LEGEND ATTRIBUTES (エリア; オブジェクト; 表示; 方向; 逆順; 反転; 位置; 水平オフセット; 垂直オフセット)	85
CT Get legend text (エリア; オブジェクト; 凡例項目) テキスト	89
CT Get object type (エリア; スコープ) 倍長整数	122
CT GET PROPERTIES (エリア; プリント順; 変更メッセージ; ホットリンクタイプ; 保存メッセージ)	59
CT GET REAL SCALE(エリア; オブジェクト; 最小自動; 最大自動; 主増分自動; 補助増分自動; 最小; 最大; 主増分; 補助増分)	90
CT Get refnum (エリア; スコープ) 倍長整数	123
CT GET TEXT ATTRIBUTES (エリア; スコープ; フォントID; フォントサイズ; スタイル; カラー; 位置揃え)	126
CT GET TITLE ATTRIBUTES (エリア; オブジェクト; 軸; 位置; 方向; タイトル)	83
 CT I	
CT Index to color (インデックス) 倍長整数	153
CT INSERT EXPRESSION (エリア; スコープ; 先頭; 最終; 式; {フォーマット})	143
CT INSERT FIELD (エリア; スコープ; 先頭; 最終; テーブル; フィールド; {フォーマット})	141
 CT L	
CT Last event(エリア) 倍長整数	53
 CT M	
CT MENU STATUS (エリア; コマンド; チェック; 使用可; 名前)	45
CT MOVE(エリア; スコープ; 左; 上)	134
 CT N	
CT NEW DOCUMENT (エリア)	37
CT New offscreen area 倍長整数	40
 CT O	
CT ON ERROR (メソッド)	46
CT ON EVENT (メソッド)	47
CT ON MENU (エリア; メソッド)	48
CT OPEN DOCUMENT (エリア; 文書; {モード})	38

CT P

CT PICTURE TO AREA (エリア; ピクチャ)	36
CT PICTURE TO CLIPBOARD (ピクチャ)	151
CT Place picture (エリア; ピクチャ; 左; 上) 倍長整数	120
CT PRINT (エリア; キャンセル可; {プリントダイアログ})	145
CT PRINT MERGE (エリア; テーブル番号; キャンセル可; {プリントダイアログ})	146
CT PUBLISH (エリア; スコープ; 名前)	109

CT R

CT RGB to color (赤; 緑; 青) 倍長整数	152
--	-----

CT S

CT SAVE DOCUMENT (エリア; 文書; タイプ; {スコープ})	39
CT SELECT (エリア; スコープ; 動作)	133
CT SET 3D VIEW (エリア; オブジェクト; 回転; 仰角)	99
CT SET AXIS ATTRIBUTES (エリア; オブジェクト; 軸; 補助目盛; 主目盛; 位置; 反転)	80
CT SET CHART FILL ATTRIBUTES (エリア; オブジェクト; タイプ; 特定部分; パター ン; カラー)	72
CT SET CHART LINE ATTRIBUTES (エリア; オブジェクト; タイプ; 特定部分; パタ ーン; カラー; 線幅)	77
CT SET CHART OPTIONS (エリア; オブジェクト; オプション)	102
CT SET CHART PICTURE (エリア; オブジェクト; タイプ; 特定部分; ピクチャ)	107
CT SET CHART TEXT ATTRIBUTES (エリア; オブジェクト; タイプ; 特定部分; フ ォントID; フォントサイズ; スタイル; カラー)	75
CT SET CHART TYPE (エリア; オブジェクト; タイプ)	104
CT SET DATE SCALE (エリア; オブジェクト; 最小自動; 最大自動; 主増分自動; 補 助増分自動; 最小; 最大; 主増分タイプ; 主増分; 補助増分タイプ; 補助増分)	94
CT SET DEPTH (エリア; オブジェクト; 水平オフセット; 垂直オフセット)	97
CT SET DISPLAY (エリア; 項目; 表示コード)	56
CT SET DOCUMENT SIZE (エリア; 幅; 高さ)	58
CT SET ENTERABLE (エリア; モード)	54
CT SET FILL ATTRIBUTES (エリア; スコープ; パターン; カラー)	132
CT SET HIGHLIGHT (エリア; スコープ; 先頭; 最終)	140
CT SET LABEL ATTRIBUTES (エリア; オブジェクト; 軸; 位置; 方向; フォーマット)	82
CT SET LEGEND ATTRIBUTES (エリア; オブジェクト; 表示; 方向; 逆順; 反転; 位 置; 水平オフセット; 垂直オフセット)	87
CT SET LEGEND TEXT (エリア; オブジェクト; 凡例項目; 凡例テキスト)	89
CT SET LINE ATTRIBUTES (エリア; スコープ; パターン; カラー; 線幅)	130

CT SET PROPERTIES (エリア; プリント順; 変更メッセージ; ホットリンクタイプ; 保存メッセージ)	60
CT SET REAL SCALE(エリア; オブジェクト; 最小自動; 最大自動; 主増分自動; 補助増分自動; 最小; 最大; 主増分; 補助増分)	91
CT SET REFNUM (エリア; スコープ; 参照番号)	124
CT SET TEXT ATTRIBUTES (エリア; スコープ; フォントID; フォントサイズ; スタイル; カラー; 位置揃え)	127
CT SET TITLE ATTRIBUTES (エリア; オブジェクト; 軸; 位置; 方向; タイトル)	84
CT SHOW GRID LINES (エリア; オブジェクト; 軸; グリッド; 表示)	78
CT SIZE(エリア; スコープ; 幅; 高さ)	135
CT Subscribe (エリア; 名前) 倍長整数	112
CT U	
CT UNPUBLISH (エリア; 名前)	111
CT UNSUBSCRIBE (エリア; 名前)	113
CT UPDATE CHART (エリア; オブジェクト; 警告表示)	105

第3章 エリアコマンド

CT AREA TO AREA (ソース; コピー先; コピーコード)	32
CT AREA TO FIELD (エリア; スコープ; テーブル; フィールド; {保存方法})	33
CT FIELD TO AREA (エリア; テーブル; フィールド)	34
CT Area to picture (エリア; スコープ) ピクチャ	35
CT PICTURE TO AREA (エリア; ピクチャ)	36
CT NEW DOCUMENT (エリア)	37
CT OPEN DOCUMENT (エリア; 文書; {モード})	38
CT SAVE DOCUMENT (エリア; 文書; タイプ; {スコープ})	39
CT New offscreen area 倍長整数	40
CT DELETE OFFSCREEN AREA (エリア)	40
CT GET AREA BOUNDARY (エリア; 境界コード; 左; 上; 右; 下)	41

第4章 エリア制御コマンド

CT DO COMMAND (エリア; コマンド)	44
CT MENU STATUS (エリア; コマンド; チェック; 使用可; 名前)	45
CT ON ERROR (メソッド)	46
CT ON EVENT (メソッド)	47
CT ON MENU (エリア; メソッド)	48
CT Error (メッセージ) 整数	49
CT EVENT FILTER (エリア; フィルタ)	50
CT EXPERT COMMAND (エリア; コマンド; ステータス)	51
CT EXPERT MODE (エリア; モード)	52
CT Last event(エリア) 倍長整数	53
CT SET ENTERABLE (エリア; モード)	54
CT Get display (エリア; 項目) 整数	55
CT SET DISPLAY (エリア; 項目; 表示コード)	56
CT GET DOCUMENT SIZE (エリア; 幅; 高さ)	57
CT SET DOCUMENT SIZE (エリア; 幅; 高さ)	58
CT GET PROPERTIES (エリア; プリント順; 変更メッセージ; ホットリンクタイプ; 保存メッセージ)	59
CT SET PROPERTIES (エリア; プリント順; 変更メッセージ; ホットリンクタイプ; 保存メッセージ)	60

第5章 エリア制御コマンド

CT Chart array (エリア; タイプ; サイズ; 項目配列; 系列配列; 数値配列)	数値	・ 65
CT Chart selection (エリア; タイプ; サイズ; グループ項目; テーブル番号; 項目フィールド; 系列/数値フィールド)	倍長整数	・ 67
CT Chart data (エリア; タイプ; サイズ; グループ項目; グループ系列; テーブル番号; 項目フィールド; 系列フィールド; 数値フィールド)	倍長整数	・ 69
CT GET CHART FILL ATTRIBUTES (エリア; オブジェクト; タイプ; 特定部分; パターン; カラー)		・ 71
CT SET CHART FILL ATTRIBUTES (エリア; オブジェクト; タイプ; 特定部分; パターン; カラー)		・ 72
CT GET CHART TEXT ATTRIBUTES (エリア; オブジェクト; タイプ; 特定部分; フォントID; フォントサイズ; スタイル; カラー)		・ 73
CT SET CHART TEXT ATTRIBUTES (エリア; オブジェクト; タイプ; 特定部分; フォントID; フォントサイズ; スタイル; カラー)		・ 75
CT GET CHART LINE ATTRIBUTES (エリア; オブジェクト; タイプ; 特定部分; パターン; カラー; 線幅)		・ 76
CT SET CHART LINE ATTRIBUTES (エリア; オブジェクト; タイプ; 特定部分; パターン; カラー; 線幅)		・ 77
CT SHOW GRID LINES (エリア; オブジェクト; 軸; グリッド; 表示)		・ 78
CT GET AXIS ATTRIBUTES (エリア; オブジェクト; 軸; 補助目盛; 主目盛; 位置; 反転)		・ 79
CT SET AXIS ATTRIBUTES (エリア; オブジェクト; 軸; 補助目盛; 主目盛; 位置; 反転)		・ 80
CT GET LABEL ATTRIBUTES (エリア; オブジェクト; 軸; 位置; 方向; フォーマット)		・ 81
CT SET LABEL ATTRIBUTES (エリア; オブジェクト; 軸; 位置; 方向; フォーマット)		・ 82
CT GET TITLE ATTRIBUTES (エリア; オブジェクト; 軸; 位置; 方向; タイトル)		83
CT SET TITLE ATTRIBUTES (エリア; オブジェクト; 軸; 位置; 方向; タイトル)		・ 84
CT GET LEGEND ATTRIBUTES (エリア; オブジェクト; 表示; 方向; 逆順; 反転; 位置; 水平オフセット; 垂直オフセット)		・ 85
CT SET LEGEND ATTRIBUTES (エリア; オブジェクト; 表示; 方向; 逆順; 反転; 位置; 水平オフセット; 垂直オフセット)		・ 87
CT Get legend text (エリア; オブジェクト; 凡例項目)	テキスト	・ 89
CT SET LEGEND TEXT (エリア; オブジェクト; 凡例項目; 凡例テキスト)		・ 89

CT GET REAL SCALE(エリア; オブジェクト; 最小自動; 最大自動; 主増分自動; 補助増分自動; 最小; 最大; 主増分; 補助増分)	90
CT SET REAL SCALE(エリア; オブジェクト; 最小自動; 最大自動; 主増分自動; 補助増分自動; 最小; 最大; 主増分; 補助増分)	91
CT GET DATE SCALE(エリア; オブジェクト; 最小自動; 最大自動; 主増分自動; 補助増分自動; 最小; 最大; 主増分タイプ; 主増分; 補助増分タイプ; 補助増分)	93
CT SET DATE SCALE(エリア; オブジェクト; 最小自動; 最大自動; 主増分自動; 補助増分自動; 最小; 最大; 主増分タイプ; 主増分; 補助増分タイプ; 補助増分)	94
CT GET DEPTH (エリア; オブジェクト; 水平オフセット; 垂直オフセット)	96
CT SET DEPTH (エリア; オブジェクト; 水平オフセット; 垂直オフセット)	97
CT GET 3D VIEW (エリア; オブジェクト; 回転; 仰角)	98
CT SET 3D VIEW (エリア; オブジェクト; 回転; 仰角)	99
CT GET CHART PART (エリア; オブジェクト; タイプ; 特定部分)	100
CT GET CHART OPTIONS (エリア; オブジェクト; オプション)	101
CT SET CHART OPTIONS (エリア; オブジェクト; オプション)	102
CT Get chart type (エリア; オブジェクト) 整数	103
CT SET CHART TYPE (エリア; オブジェクト; タイプ)	104
CT UPDATE CHART (エリア; オブジェクト; 警告表示)	105
CT Get chart picture (エリア; オブジェクト; タイプ; 特定部分) ピクチャ	106
CT SET CHART PICTURE (エリア; オブジェクト; タイプ; 特定部分; ピクチャ)	107
CT EXPLODE PIE (エリア; オブジェクト; 項目; パーセンテージ)	108

第6章 ホットリンクコマンド

CT PUBLISH (エリア; スコープ; 名前)	109
CT ADD TO HOT LINK (エリア; スコープ; 名前)	110
CT UNPUBLISH (エリア; 名前)	111
CT Subscribe (エリア; 名前) 倍長整数	112
CT UNSUBSCRIBE (エリア; 名前)	113

第7章 オブジェクトコマンド

CT Draw text (エリア; 左; 上; 右; 下; テキスト) 倍長整数	116
CT Draw line (エリア; 左; 上; 右; 下; 矢印形態) 倍長整数	117
CT Draw rectangle (エリア; 左; 上; 右; 下; 角の丸み) 倍長整数	118
CT Draw oval (エリア; 左; 上; 右; 下) 倍長整数	118
CT Array to polygon (エリア; 水平配列; 垂直配列) 倍長整数	119
CT Place picture (エリア; ピクチャ; 左; 上) 倍長整数	120
CT Get ID (エリア; スコープ; インデックス) 倍長整数	121

CT Get object type (エリア ; スコープ) 倍長整数	122
CT Get refnum (エリア ; スコープ) 倍長整数	123
CT SET REFNUM (エリア ; スコープ ; 参照番号)	124
CT GET BOUNDARY (エリア ; スコープ ; 左 ; 上 ; 右 ; 下)	125
CT GET TEXT ATTRIBUTES (エリア ; スコープ ; フォントID ; フォントサイズ ; スタイル ; カラー ; 位置揃え)	126
CT SET TEXT ATTRIBUTES (エリア ; スコープ ; フォントID ; フォントサイズ ; スタイル ; カラー ; 位置揃え)	127
CT GET LINE ATTRIBUTES (エリア ; スコープ ; パターン ; カラー ; 線幅)	129
CT SET LINE ATTRIBUTES (エリア ; スコープ ; パターン ; カラー ; 線幅)	130
CT GET FILL ATTRIBUTES (エリア ; スコープ ; パターン ; カラー)	131
CT SET FILL ATTRIBUTES (エリア ; スコープ ; パターン ; カラー)	132
CT SELECT (エリア ; スコープ ; 動作)	133
CT MOVE (エリア ; スコープ ; 左 ; 上)	134
CT SIZE (エリア ; スコープ ; 幅 ; 高さ)	135
CT Count (エリア ; スコープ) 整数	136
CT ALIGN (エリア ; スコープ ; 水平 ; 垂直)	137
CT GET HIGHLIGHT (エリア ; 先頭 ; 最終)	139
CT SET HIGHLIGHT (エリア ; スコープ ; 先頭 ; 最終)	140
CT INSERT FIELD (エリア ; スコープ ; 先頭 ; 最終 ; テーブル ; フィールド ; {フォーマット})	141
CT INSERT EXPRESSION (エリア ; スコープ ; 先頭 ; 最終 ; 式 ; {フォーマット})	143

第8章 印刷コマンド

CT PRINT (エリア ; キャンセル可 ; {プリントダイアログ})	145
CT PRINT MERGE (エリア ; テーブル番号 ; キャンセル可 ; {プリントダイアログ})	146

第9章 エリア制御コマンド

CT Font name (フォント番号) 文字列	149
CT Font number (フォント名) 整数	150
CT Clipboard to picture ピクチャ	150
CT PICTURE TO CLIPBOARD (ピクチャ)	151
CT RGB to color (赤 ; 緑 ; 青) 倍長整数	152
CT COLOR TO RGB (カラー ; 赤 ; 緑 ; 青)	153
CT Index to color (インデックス) 倍長整数	153
CT Color to index (カラー) 整数	154