

DDE Tools 6.7

リファレンス
Windows®



DDE Tools 6.7 リファレンス Windows®

Copyright© 1995 - 2000 4D SA

All rights reserved.

このマニュアルで説明されるソフトウェアは、本製品の 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。このソフトウェアおよびマニュアルは著作権にて保護されており、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、ソフトウェアおよびマニュアルの一部または全部を複製することはできません。電子的媒体の複製や、本製品の License Agreement で認められた以外のいかなる方法でこのソフトウェアを保管、または使用することもできません。

4th Dimension、4D Server、4D、4D ロゴ、4D ロゴ、およびその他の 4D 製品の名称は、4D SA の商標または登録商標です。

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

Apple、Macintosh、Mac、Power Macintosh、Laser Writer、Image Writer、ResEdit、QuickTime は Apple Computer Inc. の登録商標、または商標です。

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。

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。

はじめに	5
DDEとは何か?	5
DDEのクライアント/サーバアーキテクチャ	6
Windows上におけるプラグインのインストール	6
インストール手順	7
ランゲージ	7
DDE_Initiate	8
DDE_Request	9
DDE_Poke	9
DDE_Execute	10
DDE_Terminate	10
DDE_TERMINEALL	11
WinExec	11
DDE_SETTIMEOUT	12
AppActivate	12
エラーコード	13
DDEクライアントとしての4th Dimension - 例	14
データの受け取り	14
データ送信	14
チャンネル経由で実行のリクエストを行う	15
DDEサーバとしての4th Dimension - 例	15
Microsoft Wordでカスタマイズしたメールを作成する	15
変数の表示	15
4th Dimensionへのデータ送信	16
4th Dimension コマンドの実行	17
コマンド索引	19

はじめに

Dynamic Data ExchangeまたはDDEはWindows版バージョン3.0で導入されました。その利点は、ほとんどすべてのWindowsアプリケーションで対応していることです。

DDE とは何か？

DDEとは、2つのアプリケーション間の通信を可能にするプロトコルです。

DDEでは基本的に次の3つの動作をサポートします。

リクエストデータ

例えば、4th DimensionからMicrosoft Excelのセルの内容をリクエストしたり、Microsoft Wordから4Dフィールドの内容をリクエストしたりすることができます。

データ送信

例えば、4th DimensionからExcelのセルをセットしたり、またその逆に4Dフィールドの内容をExcelからセットすることができます。

コマンドの実現

例えば、Excelに対して4th Dimensionで新しいドキュメントを開くようにリクエストしたり、Microsoft Wordから4Dに対してプロシージャを実行するようにリクエストすることができます。

DDE のクライアント / サーバアーキテクチャ

DDEはクライアント / サーバアーキテクチャモデルを基本にしており、クライアントアプリケーションは、DDEサーバとして動作する他のアプリケーションに対してデータをリクエストする際には、特別な言語を使用します。

DDE Toolsにより、DDEサーバ、DDEクライアントの両方に4Dを使用することができます。4D自身の言語を使用して、Excelのスプレッドシートに対してリクエストを送り、セルの値を求めることができます。この場合、4DはDDEクライアント、ExcelはDDEサーバとして動作します。

他のアプリケーションと通信できる状態では、アプリケーションは常にDDEサーバとして動作します。DDE Toolsは4DをDDEサーバにします。したがって、Microsoft Wordから4Dに対し、定まった様式の手紙で空白部分を埋めたい際にフィールドの内容を求めることもできます。この場合、4DはDDEサーバであり、Microsoft WordはDDEクライアントです。

2つのアプリケーション間で通信し、両者が互いにクライアントとサーバとして機能するようなシステムも可能です。例えば、メールマージで、4DはMicrosoft WordにDDEクライアントとして接続し、ドキュメントを開き、Wordに対して4Dデータベースからの参照を更新するようにリクエストすることができます。その際、WordはDDEサーバとして4Dに接続します。この場合、4DとMicrosoft Wordとは、互いにクライアントとサーバの関係にあります。

注：4th Dimensionの、DDEクライアントとDDEサーバとしての動作と、4D Serverと4D Clientとはまったく別なものです。混同しないように注意してください。

Windows 上におけるプラグインのインストール

Windows上におけるDDE Toolsなどのルーチンのインストールは、Win4DXという名前のフォルダにパッケージをコピーして行います。Win4DXはデータベースのストラクチャファイル(*.4db)と同じレベルに配置します。

コピーは、いくつでも作成して任意のデータベースに配置することができます。Win4DXフォルダには、他のいろいろなルーチンやモジュールをともにインストールすることができます。

インストール手順

データベースにDDE Toolsをインストールする方法は、次の通りです。

1. Windowsのファイルマネージャで、新規フォルダを作成します。
2. 新しいフォルダの名前をWin4DXにします。
3. DDE Tools ファイルをこのフォルダにコピーします。
ファイルのコピー（削除なし）は、「Ctrl」キーを押しながらアイコンをドラッグします。

注：DDE Tools パッケージには、DDETOOLS.4DX ファイルとDDETOOLS.RSR ファイルが入っています。

4. このフォルダを、使用するデータベースのストラクチャファイルと同じレベルに配置します。
5. 4Dまたは4D Serverでデータベースを開きます。
DDE Toolsルーチンが使えるようになります。

ランゲージ

DDE Toolsには5個のコマンドが用意されています。

DDE_Initiate：アプリケーションとの通信を開始する。

DDE_Request：対象アプリケーションに対してデータをリクエストする。

DDE_Poke：対象アプリケーションにデータを送信する。

DDE_Execute：対象アプリケーションに対して操作を指示する。

DDE_Terminate：通信を終了する。

ここでは、次のコマンドについても説明します。

DDE_TERMINEALL：以前に開始したチャンネルをすべて閉じる。

DDE_SETTIMEOUT：タイムアウトをセットする。

WinExec：4th Dimension から Windows アプリケーションを立ち上げる。

AppActivate：特定のアプリケーションウィンドウを前面に表示する。

DDE_Initiate

DDE_Initiate (アプリケーション名; 項目名) チャンネル番号

引数	タイプ	説明
アプリケーション名	文字列	使用アプリケーション名
項目名	文字列	ドキュメント名
戻り値	倍長整数	チャンネル番号

説明

通信のセッションは必ず初期化し、セッション終了時にはクローズする必要があります。
この関数は、他のアプリケーションとの通信を開始します。

<項目名> はドキュメント名です。アプリケーションと直接通信するためには、2番目の引数に「System」を渡します。

この関数からは、開かれたチャンネルのIDが返されます。負の値はチャンネルが開かなかったこと、あるいは実行エラーを示します。

参照

DDE_Terminate

DDE_Request

DDE_Request (チャンネル番号; リクエスト; 回答) エラー

引数	タイプ	説明
チャンネル番号	倍長整数	開いているチャンネルの番号
リクエスト	文字列	特定のチャンネルに対する操作
回答	テキスト	ドキュメント名
戻り値	倍長整数	エラー

説明

対象アプリケーションにリクエストを送信します。

リクエストのフォーマットは対象アプリケーションによって異なります。例えば、Excelのセルを求めるためには<リクエスト>にセル名、あるいは3行目の5カラム目であれば“R3C5”などのように表現したセル位置を渡します。

「DDE_Initiate」の引数に「System」を渡して、アプリケーションから直接開かれた通信の場合、DDEの標準では、“SysItems”をリクエストすると、可能なリクエスト全部のリストが返されます。規約として、アプリケーションは“Topics”、“Status”、“Format”リクエストに応答します。

参照

DDE_Poke、DDE_Execute

DDE_Poke

DDE_Poke (チャンネル番号; 対象アプリケーション; データ) エラー

引数	タイプ	説明
チャンネル番号	倍長整数	開いているチャンネルの番号
対象アプリケーション	文字列	対象アプリケーションを指定
データ	テキスト	アプリケーションに送信するデータ
戻り値	倍長整数	エラー

説明

対象アプリケーションへデータを送信します。

参照

DDE_Execute

DDE_Execute

DDE_Execute (チャンネル番号; コマンド) エラー

引数	タイプ	説明
チャンネル番号	倍長整数	開いているチャンネルの番号
コマンド	テキスト	アプリケーションに実行させるコマンド
戻り値	倍長整数	エラー

説明

対象アプリケーションにコマンドを実行させます。コマンドは角カッコ ([]) で囲む必要があります。複数のコマンドを続けて送ることも可能です。

```
Err:=DDE_Execute(Channel;"[NEW()][ZOOM(50)]")
```

参照

DDE_Initiate

DDE_Terminate

DDE_Terminate (チャンネル番号) エラー

引数	タイプ	説明
チャンネル番号	倍長整数	開いているチャンネルの番号
戻り値	倍長整数	エラー

引数として渡されたチャンネル番号のセッションの通信を終了します。

参照

DDE_Initiate

DDE_TERMINEALL

引数	タイプ	説明
----	-----	----

このコマンドには、引数はありません。

説明

以前に開かれたチャンネルをすべて閉じます。

参照

DDE_Terminate, DDE_Initiate

WinExec

WinExec (コマンド) エラー

引数	タイプ	説明
コマンド	文字列	開いているチャンネルの番号
戻り値	倍長整数	エラー

説明

この関数により、4th Dimension から Windows アプリケーションを起動することができます。

例：

```
vChannel := DDE_Initiate("Excel";"SYSTEM")
If(vChannel<0)
    ` Excel から応答が無い場合
    WinExec ("Excel")
    ` Excel アプリケーションを立ち上げる
    vChannel := DDE_Initiate ("Excel";"SYSTEM")
End if
If(vChannel<0)
    ALERT ("Excel からの応答が無い、または見つからない")
Else
    ...
End if
```

参照

なし

DDE_SETTIMEOUT

DDE_SETTIMEOUT (Timeout)

説明

このコマンドで、タイムアウトをセットします。値はミリ秒を表しています。デフォルト値は3000ミリ秒です。アプリケーションの応答までに時間がかかる場合は、この値を増やすことができます。

負の値を渡すと、値はデフォルト値にリセットされます。

参照

なし

AppActivate

AppActivate(WindowTitle) result

引数	タイプ	説明
ウインドウタイトル	文字列	ウインドウ
戻り値	倍長整数	結果

説明

この関数で、特定のアプリケーションウインドウを前面に表示することができます。ウインドウは、ウインドウタイトル名から選択されます。完全なウインドウタイトル名、あるいは最初の数文字を渡すことができます。後者の場合、1文字目が、渡された文字列にマッチした最初のウインドウが前面に表示されます。

ウインドウがアクティブである場合は0を返し、そうでない場合は-1を返します。

参照

なし

エラーコード

次の表は、DDE Tools から返されるエラーコードを一覧にしたものです。

コード	説明
-1	チャンネルが使いません（最高64個まで）
-2	コマンドに渡されたチャンネル番号が無効です
-3	DDE ウィンドウが作成できません
-4	対象アプリケーションから応答がありません
-5	対象から応答がありません
-6	データ形式が不明です
-7	無効なコマンドです

DDE クライアントとしての 4th Dimension - 例

ここでは、DDE クライアントとしての 4D の働きを示す例について説明します。まずはじめに、次のことを心に留めておいてください。

通信には必ず初期化が必要であり、通信終了時には閉じなければならない。

これらの操作を行うのは次の関数です。

DDE_Initiate

DDE_Terminate

注：DDE_TERMINEALL コマンドでも、以前に開かれた接続をすべて閉じることができます。

データの受け取り

次の例では、Excel のドキュメント SPREAD.XLS の “ 営業 ” というセルの内容を求めます。始める前に Excel アプリケーションが立ち上がっているか、また SPREAD.XLS ドキュメントが開かれているかどうかを確認します。データ交換の終了時には、**DDE_Terminate** 関数によりチャンネルをクローズする必要があります。例えば、ボタンに次のようなスクリプトを作成します。

```
vChannel:= DDE_Initiate("Excel";"C:¥SPREAD.XLS")
Err:= DDE_Request(vChannel;"営業";v 営業)
Err:= DDE_Terminate (vChannel)
ALERT("営業 Excel より : "+v 営業)
```

もちろんだキュメン内に「営業」というセルが存在しなければなりません。任意のセルを指定するには、例えば 2 行 3 カラム目を指定するために “ R2C3 ” という形式を使用することもできます。

データ送信

そこで、Excel のスプレッドシート内の “ 営業 ” というセルに値を代入します。

```
vChannel:= DDE_Initiate("Excel";"C:¥SPREAD.XLS")
v 営業:=Ask("営業を Excel に送信しますか?")
Err:= DDE_Poke(vChannel;"営業";v 営業)
Err:= DDE_Terminate (vChannel)
```

チャンネル経由で実行のリクエストを行う

Excel に対し、新しくドキュメントを作成するように求めます。

```
vChannel:= DDE_Initiate("Excel";"SYSTEM")
```

```
Err:= DDE_Execute(vChannel;"[New()]")
```

```
Err:= DDE_Terminate (vChannel)
```

DDE のプロトコルでは、コマンドは角カッコ内に書くことになっています。カッコを並べ、1 回のリクエストで複数のコマンドを送ることができます。

DDE サーバとしての 4th Dimension - 例

4D のフィールドや、プロセス変数、インタープロセス変数の内容を他のアプリケーションから読み込んで、セットすることができます。

Microsoft Word でカスタマイズしたメールを作成する

[顧客]ファイルに個人情報を納めた “顧客.4DB ” という名前のデータベースがあるとします。このデータベースを開き、「ユーザ」モードで手紙を送る人をダブルクリックして選択します。

Microsoft Word を立ち上げ、下書きを作成します。ここで宛名を書く場所に、Word の「挿入」メニューから「フィールド」を選択してフィールドを挿入します。そこで次のような式を入力します。

```
{DDE 4DCONTACTS.4DB[CONTACTS]Name}
```

住所、郵便番号等、他のフィールドについても同じ手順を繰り返し、ドキュメントに必要な項目を挿入します。

フィールド名にスペースが含まれている場合は、二重の不等号で <<[顧客]Zip Code>> のように囲みます。

変数の表示

プロセス変数やインタープロセス変数の内容を表示するには、フィールド名を変数名で置き換えます。

4th Dimension へのデータ送信

社員の給与台帳の計算を行う Excel のスプレッドシートがあり、社員ファイルは 4th Dimension で管理しているとします。

4th Dimension の「ユーザ」モードで、“ DatabaseDDE.4db ” ファイルを開きます。「ユーザ」モードには “ 山田 ” レコードがあります。

Excel では、給与振込額を決定するための典型的な計算用シート処理を行います。簡単なマクロを使って給与合計を計算し、値をレコードへ送信します。例えば、次のようなマクロになります。

```
Sub CalculateSalary ()
    vChannel = DDE_Initiate ("4D"; "DatabaseDDE.4DB")
    Cells(10; 5) .Formula = DDE_Request (vChannel; "[Employee]Salary")
    Set NetSalary = Cells(5; 5)
    DDE_Poke vChannel; "[Employee]NetSalary"; NetSalary
    DDE_Terminate vChannel
End Sub
```

プロシージャの最初の部分は、カレントレコードに入力された総支給額を読み、セル(10; 5)に読み込みます。

```
Sub CalculateSalary()
    '4D との通信を初期化する
    vChannel = DDE_Initiate ("4D"; "DatabaseDDE.4DB")
    '4D から総支給額フィールドの値を得る
    Cells(10;5).Formula = DDE_Request(vChannel; "[Employee]Salary")
```

次の部分で、セル(5; 5)で計算された給与合計を戻し、カレントレコードの給与合計フィールドに書き出します。

```
    Set NetSalary = Cells(5;5)
    DDE_Poke vChannel; "[Employee]NetSalary"; NetSalary
    '通信が中断される可能性があり
    DDE_Terminate vChannel
End Sub
```

4th Dimension コマンドの実行

他のアプリケーションから4th DimensionとのDDE接続を開いている場合、そのアプリケーションから4Dのプロシージャを実行することができます。

4D特有のマルチプロセス構造では、プロシージャの実行はプロセス内で行われる必要があります。そこで、DDE Toolsには、他のアプリケーションから送られたDDEコマンドを実行する“**DDE_SERVER**”コマンドが用意されています。コマンドは、**DDE_SERVER**コマンドを呼んだプロセス内で処理されます。

そこで、定期的に**DDE_SERVER**コマンドが入ったプロシージャを含むプロセスを作る必要があります。

```
ProcDDE プロシージャ
While(true) DDE_SERVER
    DELAY PROCESS (Process number; 60)
End while
```

このプロシージャは、「Startup」プロシージャから起動します。

```
MyProc:=New process ("ProcDDE"; 64*1024; "Process DDE")
```

社員レコードが入っている4Dデータベースがあり、給与支給額の計算を行い、結果を<>Result変数に納めるグローバルプロシージャがあるとします。

```
MPayroll メソッド
ALL RECORDS ([Employee])
<>Result:=Total([Employee]Salary)
```

このプロシージャは、次に示すマクロでExcelから起動することができます。

```
Sub CalculatePayroll()
    vChannel = DDE_Initiate("4D"; "DatabaseDDE.4DB")
    DDE_Execute vChannel; "[MPayroll]"
    Cells(8; 5).Formula = DDE_Request (vChannel; "<>Result")
    DDE_Terminate vChannel
End Sub
```

4DはExcelからの支払給与総額の計算指示を受け取ると、このコマンドをバッファに入れ、4Dの**DDE_SERVER**コマンドが起動されるのを待ちます。**DDE_SERVER**コマンドが呼ばれると、4Dはコマンドを実行し、**DDE_SERVER**を呼び出したプロセスで結果を受け取り、操作を有効にします。

注：**DDE_SERVER**コマンドを定期的に呼び出すプロセスのインストールを誤ると、目的のコマンドは4Dで実行されず、コマンドを起動したアプリケーションでは、その後の4Dからの応答が得られなくなります。

A

AppActivate 12

D

DDE_Execute 10

DDE_Initiate 8

DDE_Poke 9

DDE_Request 9

DDE_SETTIMEOUT 12

DDE_Terminate 10

DDE_TERMINEALL 11

W

WinExec 11

