

# 4D Open for 4th Dimension 6.7

---

リファレンス  
Windows<sup>®</sup> and Mac<sup>™</sup> OS



---

## 4D Open for 4th Dimension 6.7 リファレンス Windows® and Mac™ OS

Copyright© 1994 - 2000 4D SA

All rights reserved.

---

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 4D SA 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。

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。

<b>第 1 章</b>	<b>はじめに</b> .....	9
	4D Open for 4D とは？ .....	9
	4D Server とは? .....	9
	4D Open for 4th Dimension の使用 .....	10
	4D Server データベースへの接続 .....	11
	ストラクチャ定義へのアクセス .....	11
	検索 .....	11
	命名セレクション .....	11
	配列 .....	11
	マルチユーザ環境の管理 .....	12
	トランザクション .....	12
	算術演算 .....	12
	セキュリティ .....	12
	4D Open for 4D ルーチンの使用 .....	13
	4D Open for 4D ルーチンへ渡す引数の特性 .....	13
	値を受け取らない引数 .....	13
	値を受け取る引数 .....	14
	4D オブジェクトを参照する引数(フィールド、変数、配列) .....	14
	知っておくべきこと .....	15
	必要なこと .....	15
	このマニュアルについて .....	16
	表記方法について .....	16
	例題のコードについて .....	16
<b>第 2 章</b>	<b>ネットワークコンポーネント</b> .....	17
	ネットワークコンポーネント概論 .....	17
	ネットワークコンポーネント ID における定数の使用 .....	17
	複数のネットワークコンポーネントの使用 .....	18
	OP Count network components .....	20
	OP Get network component info .....	21
	OP Load network component .....	23

	OP Unload network component	24
	OP Request	26
<b>第 3 章</b>	<b>ネットワークユーティリティ</b>	<b>27</b>
	ネットワークユーティリティ 序論	27
	OP Get station name	28
	OP Start Remote Connection	29
	OP End Remote Connection	30
	OP Remote Connection Status	31
<b>第 4 章</b>	<b>接続</b>	<b>33</b>
	接続の開始	33
	プロセス	34
	接続の終了	34
	4D Client 以外のアプリケーションに 4D Server への アクセス権を持たせる	34
	4D Open グループへアクセス権を与える	34
	接続、序論	36
	複数の接続、複数のデータベース	36
	4D Server データベースへの接続	37
	OP Select 4D Server	39
	OP Find 4D Server	43
	OP Delete 4D Server	45
	OP Open connection	46
	OP Close connection	48
<b>第 5 章</b>	<b>ストラクチャ情報</b>	<b>49</b>
	ストラクチャ情報、序論	49
	OP Count tables	50
	OP Get all tablename	51
	OP Get field properties	53
	OP Get table properties	56
	OP Get one field number	60
	OP Get all field numbers	61
	OP Get one tablename	63
	OP Count fields	64

<b>第 6 章</b>	<b>検索とソート</b> .....	<b>6 5</b>
	検索とソート、序論 .....	65
	複数条件検索 .....	65
	OP Single query .....	67
	OP Single query selection .....	69
	OP Single order by .....	71
	OP Multi query .....	73
	OP Multi query selection .....	77
	OP Multi order by .....	81
<b>第 7 章</b>	<b>セレクション</b> .....	<b>8 3</b>
	セレクション、序論 .....	83
	OP Records in table .....	84
	OP Records in selection .....	85
	OP All records .....	87
	OP Reduce selection .....	89
	OP Delete selection .....	91
	OP Many to one join .....	92
	OP One to many join .....	94
	OP Scan index .....	96
<b>第 8 章</b>	<b>配列</b> .....	<b>9 9</b>
	配列、序論 .....	99
	OP Selection to array .....	100
	OP Distinct values .....	103
	OP Array to selection .....	105
	OP Subselection to array .....	108
<b>第 9 章</b>	<b>トランザクション</b> .....	<b>1 1 1</b>
	トランザクション、序論 .....	111
	OP Start transaction .....	112
	OP Validate transaction .....	115
	OP Cancel transaction .....	116

<b>第 10 章</b>	<b>命名セクション</b> .....	117
	命名セクション、序論 .....	117
	プロセス/インタープロセス命名セクション .....	117
	OP Copy named selection .....	119
	OP Cut named selection .....	121
	OP Use named selection .....	124
	OP Clear named selection .....	126
<b>第 11 章</b>	<b>一連の値</b> .....	127
	一連の値、序論 .....	127
	OP Sum .....	128
	OP Average .....	130
	OP Min .....	131
	OP Max .....	132
<b>第 12 章</b>	<b>バインド</b> .....	133
	バインド、序論 .....	133
	OP Create bind .....	135
	OP Delete bind .....	136
	OP Define bind by numbers .....	137
	OP Define bind by pointer .....	140
	OP Set format .....	142
<b>第 13 章</b>	<b>レコード</b> .....	145
	レコード、序論 .....	145
	OP Set access mode .....	147
	Read-only モード .....	147
	Read-write モード .....	147
	OP Goto selected record .....	149
	OP Load record .....	152
	OP Unload record .....	155
	OP Update record .....	156
	OP New record .....	160
	OP Sequence number .....	162
	OP Delete record .....	163
	OP Goto record .....	166
	OP Get record numbers .....	168
	OP Current Record Number .....	170

<b>第 14 章</b>	<b>ユーティリティ</b> .....	<b>171</b>
	ユーティリティ、序論 .....	171
	OP Get error text .....	172
	OP Set option .....	174
	OP Get option .....	177
	OP Flush Buffers .....	178
	OP Get version number .....	179
<b>第 15 章</b>	<b>ユーザとグループ</b> .....	<b>181</b>
	ユーザとグループ、序論 .....	181
	OP Get user list .....	182
	OP Enter password .....	184
	OP Get users and groups .....	187
<b>第 16 章</b>	<b>サーバ情報</b> .....	<b>189</b>
	サーバ情報、序論 .....	189
	OP Get server date .....	190
	OP Get server time .....	192
	OP Count connected users .....	193
	OP Count user processes .....	194
	OP Get server version .....	195
	OP Get process list .....	196
<b>第 17 章</b>	<b>プロセス</b> .....	<b>199</b>
	OP Process number .....	199
	OP Set Process Variable .....	200
	OP Get Process Variable .....	201
	OP Execute On Server .....	202
	OP Set Semaphore .....	203
	OP Clear Semaphore .....	204
	OP Check Semaphore .....	205
<b>第 18 章</b>	<b>セット</b> .....	<b>207</b>
	OP Create Empty Set .....	207
	OP Create Set .....	208
	OP Use Set .....	209
	OP Add To Set .....	210
	OP Remove From Set .....	211
	OP Clear Set .....	212

OP Is In Set .....	213
OP Records In Set .....	214
OP Copy Set .....	215
OP Union Set .....	216
OP Intersection Set .....	217
OP Difference Set .....	218
<b>コマンド索引 .....</b>	<b>2 1 9</b>



## 4D Open for 4D とは？

---

4D Open for 4D (4D Open for 4th Dimension)は、4th Dimension、4D Client、4D Serverから複数サーバへ同時接続を可能にする4Dコネクティビティプラグインです。4D Clientで利用する場合は、通常の4D Clientとしての接続方法に加えて、また、4D Serverで利用する場合は、バージョン6以降の機能であるストアードプロシージャ、トリガから4D OPENによる接続が可能になります。

これらのアプリケーションから：

- 4D Serverで稼働中のデータベースストラクチャの情報を取得できます。

- (アプリケーションバージョン、接続中のユーザ、サーバで実行中のプロセスのような)4D Serverの各種ステータス情報を取得できます。

- 4D Server上のデータを検索、取得、更新することができます。

4D Open for 4Dをインストールすれば、新しいコマンドが4th Dimensionに追加されます。

すべての4D Open for 4Dルーチンの名前は接頭辞“OP”が付いていますので、標準の4Dコマンドや他のプラグインコマンドと区別がつかます。

## 4D Server とは？

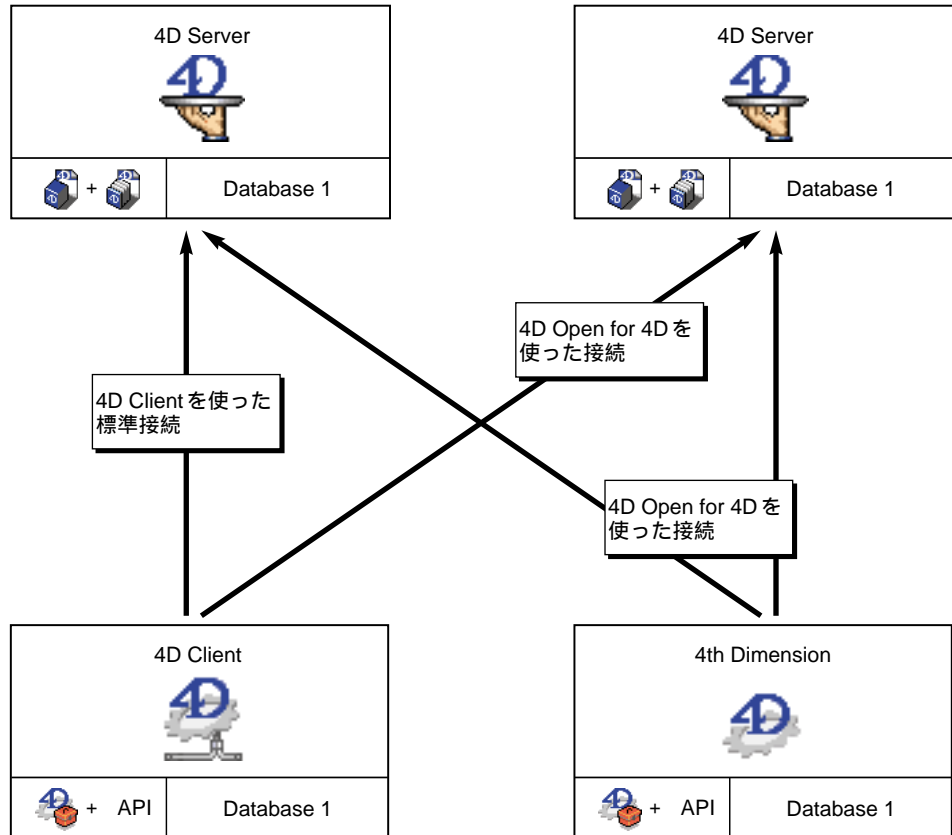
4D Serverは4D社の製品であり、Client/Server型マルチユーザ版4th Dimensionです。4D Serverを使うことによって、マルチユーザデータベースを作り、運用できるだけでなく、client/server構造を持つカスタムアプリケーションを作成することができます。

4D Serverシステムは、4D Serverと4D Clientという2つのアプリケーションで構成されません。

4D Serverは、データベースを管理し、ユーザからのリクエストに応えます。端末機として動作するマシンから4D Clientを使用して4D Server上で稼働するデータベースにアクセスし、データの追加、レポートの作成、データベースの設計変更などのデータベース操作を行うことができます。すなわち、4th Dimensionで実行可能なすべてのことを、4D Serverと4D Clientのシステムでも行うことができるのです。

## 4D Open for 4th Dimension の使用

4D Open for 4th Dimensionルーチンは4th Dimension、4D Client、さらにバージョン6以降の4D Serverで実行できます。下図は4D Clientと4th Dimensionから4D Serverデータベースへの接続を示しています。



4D Open for 4th Dimensionを使えば、4D Client、4th Dimension、4D Serverから別の4D Serverへ複数の接続を行なうことが可能です。また、複数の4D Serverへ同時に接続することも可能です。

4D Open for 4th Dimensionはプラグインルーチンのセットとして構成されています。実行可能なデータベース操作を、いくつかの機能別に紹介します：

### 4D Server データベースへの接続

1つ、もしくは複数の4D Serverデータベースへ同時に接続できます。

このグループの機能としては、クライアントアプリケーションとマシンの初期化、ネットワークコンポーネントの選択、データベースの選択があります。データベースの選択は、メソッド、またはダイアログを使用することにより、ユーザに行わせることも可能です。

### ストラクチャ定義へのアクセス

テーブル名、フィールド名、各フィールドの定義を取得できます。

このグループのルーチンを使うことにより、例えば、テーブル名ポップアップメニューのような、データベースストラクチャを表示するユーザインタフェースを簡単に実装できます。

### 検索

データを更新する前に、全レコードを選択する (**ALL RECORD**) か、または検索を行なって特定のカレントレコード、もしくはセクションを作っておく必要があります。データの更新、削除が可能なのは、そのレコード、もしくはセクションに含まれるレコードだけです。

プロセスごとに各テーブルに1つだけセクションが存在します。このセクションは他のデータベース操作（検索、セット演算など）を行なわない限り、存続し続けます。他のプロセス、他のユーザ（4D OPEN接続の有無に関わらず）からは変更されません。

### 命名セクション

4D Open for 4th Dimensionは命名セクションをサポートするため、検索とセクション作成の最適化に寄与します。検索の結果、得られたセクションを命名セクションとして保持できます。この命名セクションを使えば、再び検索することなしに瞬時にセクションを回復することが可能です。

### 配列

4D Open for 4th Dimensionはデータの送信と受信の際、配列をデータバッファとして使用するルーチンを提供します。

## マルチユーザ環境の管理

マルチユーザ環境でのデータベース操作を管理するのに必要なテーブルごとのアクセス権設定 ( **Read Only** / **Read Write** )、または使い終わったレコードのアンロックを実行できます。

## トランザクション

複数のテーブルに渡る複雑な更新処理を、データの整合性を保ったまま実行するのに必要なトランザクション機能として、start、validate、cancel transactionsの各ルーチンを提供します。

## 算術演算

4D Open for 4th Dimensionは、複数レコードの数値データに対する算術演算を素早く実行するためにsum、average、maximum、minimumのルーチンを提供します。

## セキュリティ

4D Open for 4th Dimensionによる接続も、通常の4th Dimensionパスワードアクセスシステムの管理下に置かれます。データベースがパスワードプロテクトされていれば、正しいユーザ名とパスワードなしではアクセスできません。レコードのロード、追加、更新、削除にも適応するアクセス権を要求されます。

## 4D Open for 4D ルーチンの使用

すべての4D Open for 4th Dimension (4D Open for 4D)プラグインルーチンは、関数の形態を取ります。それぞれの関数は、コマンドの実行が成功したかどうかを知らせるためにエラーコードを返します。エラーコードが「0」ならば、エラーは起きていません。

4D Open for 4D関数の返すエラーコードは、常に倍長整数 (LONGINT) 変数で受け取ってください。以下のコードは、4D Open for 4D関数の使い方を示しています：

```
`何かを行い、エラーコードを受け取る  
$errCode:=OP Load network component (<>ADSPNetComp)
```

**OP Get network component**のような、いくつかの4D Open for 4Dルーチンはエラーコードだけでなく、ある種の情報も返します。そのような値は引数として渡した変数であり、また前述の通り、エラーコードはエラーを受け取る変数で、それぞれ受け取ります。

次の数行のコードでは、エラーコードは\$errCodeに、ルーチンが返す情報は引数 < NCID > と引数 < NCName > に代入されて返ってきます (これらの引数に渡された変数はあらかじめ定義済みとします)、引数 < NC > は、他の2つの引数 < NCID > 、 < NCName > で受け取る情報を得るために、必要なネットワークコンポーネント番号をルーチンへ渡すために使っています。

```
$errCode:=OP Get network component info (NC;NCID;NCName)
```

### 4D Open for 4D ルーチンへ渡す引数の特性

引数の型は、ルーチンがその引数へ返す値によります。また、4Dオブジェクトを参照しているかどうかにもよります。

### 値を受け取らない引数

式、フィールド、変数 (ローカル、プロセス、インタープロセス)、配列 (プロセス、インタープロセス) を渡すことができます。

ローカル配列は渡せません。

マルチプロセス環境で動作させていて、インタープロセス変数、配列を引数として渡す場合は、ローカルセマフォを使ってこれらの変数、配列の使用を (他のプロセスから) プロテクトするような注意が必要です。

使用例：

```
$ErrCode:= OP Load network component (1)
$ErrCode:= OP Load network component (GetCompID("ADSP"))
$ErrCode:= OP Load network component ([Components]Component ID)
$ErrCode:= OP Load network component ($NCID)
$ErrCode:= OP Load network component (NCID)
$ErrCode:= OP Load network component (<>NCID)
$ErrCode:= OP Enter password (arUser;$UserName;$Password)
$ErrCode:= OP Enter password (<>arUser;$UserName;$Password)
```

## 値を受け取る引数

フィールド、変数（ローカル、プロセス、インタープロセス）、配列（プロセス、インタープロセス）を渡すことができます。

ローカル配列は渡せません。

マルチプロセス環境で動作させていて、インタープロセス変数、配列を引数として渡す場合は、ローカルセマフォを使ってこれらの変数、配列の使用を（他のプロセスから）プロテクトするような注意が必要です。

使用例：

```
$ErrCode:= OP Count tables ($ConnID:[Servers]NbFiles)
$ErrCode:= OP Count tables ($ConnID;$NbFiles)
$ErrCode:= OP Count tables ($ConnID;NbFiles)
$ErrCode:= OP Count tables ($ConnID;<>NbFiles)
$ErrCode:= OP Get all tablenames ($ConnID;arFName)
$ErrCode:= OP Get all tablenames ($ConnID;<>arFName)
```

## 4D オブジェクトを参照する引数(フィールド、変数、配列)

以下のような方法で4Dオブジェクトを参照します。実際にどの値が使えるかはルーチンごとに異なります。

フィールド（フィールドポインタ、フィールド番号）、プロセス、インタープロセス変数 / 配列（ポインタ、名前）

配列要素へのポインタは使えません。

ローカル変数 / 配列へのポインタも使えません。

マルチプロセス環境で動作させていて、インタープロセス変数、配列を引数として渡す場合は、ローカルセマフォを使ってこれらの変数、配列の使用を（他のプロセスから）プロテクトするような注意が必要です。

注意：フィールドが使える引数にはBLOBフィールドも使うことができます。

使用例 :

```
$ErrCode:= OP Single query ($ConnID;$FileID;$FieldID;"=";->[Data]Value;  
                                         $NbFound)  
$ErrCode:= OP Single query ($ConnID;$FileID;$FieldID;"=";->vValue;$NbFound)  
$ErrCode:= OP Single query ($ConnID;$FileID;$FieldID;"=";-<>Value;  
                                         $NbFound)  
  
$ErrCode:= OP Define bind by numbers ($BindID;10;4;2;4;"")  
$ErrCode:= OP Define bind by numbers ($BindID;10;4;0;0;"vName")  
$ErrCode:= OP Define bind by numbers ($BindID;10;4;0;0;"<>Name")  
$ErrCode:= OP Define bind by numbers ($BindID;10;4;0;0;"arValue")  
$ErrCode:= OP Define bind by numbers ($BindID;10;4;0;0;"<>arValue")  
$ErrCode:= OP Define bind by pointer ($BindID;10;4;->vName)  
$ErrCode:= OP Define bind by pointer ($BindID;10;4;-<>vName)  
$ErrCode:= OP Define bind by pointer ($BindID;10;4;->arValue)  
$ErrCode:= OP Define bind by pointer ($BindID;10;4;-<>arValue)
```

### 知っておくべきこと

4D Open for 4Dを使用するためには、4th Dimension 言語について慣れ親しんでいる必要があります。

### 必要なこと

4D Open for 4D 6.0を使用するためには、4D Server、4D Client、4th Dimensionのいずれかのバージョン1.5 / 3.5以降を持っていないければなりません。

また、使用する予定のネットワークプロトコルに対応したネットワークコンポーネントをインストールしておくことも必要です。詳しくは、『4D Server ネットワークコンポーネント』マニュアルを参照してください。

## このマニュアルについて

---

このマニュアルでは、4D Open for 4th Dimension ルーチンについて説明しますが、既に4th Dimension プログラミング言語に慣れ親しんでいることを前提としています。4th Dimension プログラミング言語に関する詳細は、『4th Dimension ランゲージリファレンス』マニュアルを参照してください。

第2章以降は、関数カテゴリー（メソッドエディタに現れるルーチンテーマ）ごとに説明していきます。

### 表記方法について

このマニュアルでは、より深く理解できるように、いくつかの決まった書式に従っています。

構文を示している部分での「->」「<-」記号は以下のような意味です：

記号	説明
->	引数は値をルーチンへ渡すためだけに用いられます。値、変数、配列が渡せません。
<-	値を受け取る引数です。したがって、ユーザは変数が配列を渡す必要があります。ローカル配列は渡せません。

このマニュアルで使われる書式は4th Dimension メソッドエディタでメソッドやコマンドを識別するために使われているものと、よく似ています。

4D コマンドのフォントスタイル：

**ALL RECORDS**

4D Open for 4th Dimension のフォントスタイル：

***OP All records***

引数のフォントスタイル：

ConnectionID

### 例題のコードについて

このマニュアルで示される例題のコードは、4D Open for 4th Dimension の機能を説明するためのものです。これらは最適化されたコードではないかもしれませんが。また、これらのコードを実際にストラクチャで使用する場合には、エラーチェックを怠らないようにしてください。



## ネットワークコンポーネント概論

---

4D Open for 4th Dimensionでは、いくつかの異なったネットワークコンポーネントを使用して4D Server上で動作するデータベースにアクセスすることができます。

現在のところ、ADSPプロトコル（MacintoshがAppleTalkネットワークを通じて通信するために設計されたもの）がMac OS 7.x（以降）及びWindows NTの環境で利用できます。

また、Windows 95向けのその他の製品（MacLan connect）の一部としても供給されています。TCP/IPプロトコルは7.1以降のMac OS、Windows NT、Windows 95で利用可能です。

またWindows 3.1では別途、必要なファイルを追加することによって実装可能です。IPX/SPXプロトコルはWindows NT、Windows 95、Windows 3.11に含まれています。

また、Macintoshでも利用可能であり、Macintosh版の4D Server、4D Clientディスクに収録されています（現在の4Dファミリーでは利用できない場合があります。4D社へお問い合わせください）。

それぞれのネットワークプロトコルに準じた通信を開始する前には、対応するネットワークコンポーネントの初期化が必要です。ADSPには、ADSPネットワークコンポーネントがあります。TCP/IPにはTCP/IPネットワークコンポーネントがあります。その他のプロトコルにもそれぞれ個別に対応するネットワークコンポーネントがあります。

まず初めに、使用予定のネットワークコンポーネントをインストールする必要があります（現在では、初めからこれらのネットワークコンポーネントがインストール済みの4D製品もあります）。

### ネットワークコンポーネントIDにおける定数の使用

一度、初期化されたネットワークコンポーネントには、固定的なID番号が割り振られません。したがって、厳密にはネットワークコンポーネントIDは**OP Get network component info**関数を使用して毎回取得する必要はなく、その代わりに最初に取得したID番号を、DB全体を通して、定数として使用することが可能です。

ネットワークコンポーネントとそのIDは下記の表のようになっています：

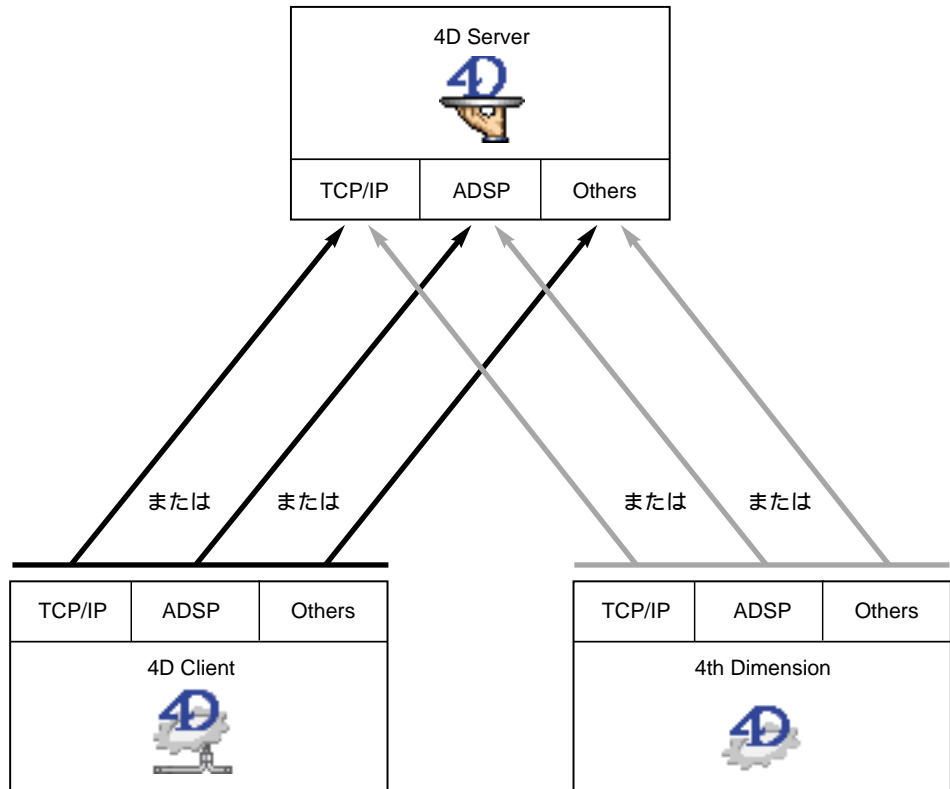
ネットワークコンポーネント	ID	プラットフォーム
ADSP	1	Macintosh/Windows
TCP/IP	29	Macintosh
TCP/IP	2	Windows

しかし、**OP Count network components**関数と**OP Get network component info**関数を使用して汎用コードを書くことは可能です。この方法で書かれたコードは、将来に渡って4D社やサードパーティからリリースされる様々なネットワークコンポーネントと互換性を保てます。

### 複数のネットワークコンポーネントの使用

4D Open for 4th Dimensionを使って4D Serverデータベースと通信するという事は、ネットワークを経由してリクエストを送り、結果を受け取るということです。データはネットワークプロトコルに従ってネットワーク上で運ばれます。

4D Open for 4th Dimensionは特定のネットワークプロトコルに依存しないように設計されています。ご使用の4th Dimension、4D Clientアプリケーションは4D ServerへADSP、TCP/IP、IPX/SPXプロトコルを使って接続できます。



同時に複数のネットワークプロトコルを使用して接続することもできます。例えば、2つのサーバへTCP/IPを使って繋ぎ、3つめにはADSPを使用して繋ぐといったことです。

4D Open for 4th Dimensionを使用するためにADSP、IPX/SPX、TCP/IPなどのネットワークプロトコルそのものに関する知識は必要ありません。4D Open for 4th Dimensionはサーバにリクエストを送り、結果を受け取るために、“内部的に”ネットワークコンポーネントを呼び出すハイレベルデータベースルーチンを提供します。

利用可能なネットワークコンポーネントはインストール済みのネットワークコンポーネントに限られます。4D Open for 4th Dimensionはネットワークコンポーネントに関して4つのルーチンを用意しています。それらは、インストールされたネットワークコンポーネントを“数える”、“情報を得る”、“初期化する”、“破棄する”ルーチンです。

ネットワークコンポーネントルーチン：

**OP Count network components**：インストールされているネットワークコンポーネントの数を返します。

**OP Get network component info**：ネットワークコンポーネントの参照番号（ID）、名前を返します。

**OP Load network component**：指定したネットワークコンポーネントを初期化します。

**OP Unload network component**：指定したネットワークコンポーネントを破棄します。

#### 参照

OP Count network components、OP Get network component info、OP Load network component、OP Unload network component

## OP Count network components

---

### OP Count network components (HowManyNetComps) Error

引数	タイプ	説明
HowManyNetComps	倍長整数	インストールされているネットワークコンポーネントの数
Error	倍長整数	エラーコード

#### 説明

**OP Count network components** は、インストールされているネットワークコンポーネントの数を数えます。

#### エラーコード

関数の実行が成功した場合、**OP Count network components** は0を返します。それ以外の場合は下記のエラーコードを返します：

エラーコード	説明
10128	4D Open for 4th Dimensionがメモリに読み込まれていません。
10154	このコマンドを直ちに実行することができません。

#### 例題

この例題ではインストールされたネットワークコンポーネントの数を数え、個々のネットワークコンポーネントごとにIDと名前を取得し、それらの情報を配列に格納しています。

```
C_STRING(255;$Component)
C_LONGINT($ErrCode;$Count;$CompID;$i)

$ErrCode:=OP Count network components ($Count)

ARRAY LONGINT (aCompID;$Count)
ARRAY STRING (255;aComponent;$Count)

For ($i;1;$Count)
  $ErrCode:=OP Get network component info ($i;$CompID;$Component)
  If ($ErrCode=0)
    aCompID{$i}:=$CompID
    aComponent{$i}:=$Component
  End if
End for
```

#### 参照

OP Get network component info

## OP Get network component info

**OP Get network component info** (NetCompNumber; NetCompID; NetCompName)

Error

引数	タイプ	説明
NetCompNumber	整数	調べたいネットワークコンポーネントのインデックス番号
NetCompID	整数	指定したネットワークコンポーネントの参照番号 (ID)
NetCompName	固定長文字列	指定したネットワークコンポーネントの名前
Error	倍長整数	エラーコード

説明

**OP Get network component info** を使用すれば、現在インストールされているネットワークコンポーネントの情報を得られます。このルーチンの実行後、`< NetCompID >` と `< NetCompName >` 引数として渡した変数に指定したネットワークコンポーネントの参照番号 (ID) と名前を得ることができます。

`< NetCompID >` の値は、**OP Select 4D Server**、**OP Find 4D Server** に渡す値で、それぞれのネットワークコンポーネントごとに別々の値です。例えばADSPネットワークコンポーネントIDは「1」です。

インストールされたすべてのネットワークコンポーネントについて情報を得たい場合は、**OP Count network components** が返す数だけこのルーチンを繰り返し呼び出してください。

**OP Get network component info** を使用して得られるネットワークコンポーネント参照番号 (ID) を使ってコードを書くことが、ご使用のコードと将来の4D OPENのバージョンや追加されるネットワークコンポーネントとの互換性を保証する方法です。

エラーコード

関数の実行が成功した場合、**OP Get network component info** は0を返します。それ以外の場合は下記のエラーコードを返します：

エラーコード	説明
-192	指定されたインデックス番号に対応するネットワークコンポーネントがありません。
10128	4D Open for 4th Dimension がメモリに読み込まれていません。
10154	このコマンドを直ちに実行することができません。

## 例題

下記の"GetCompID"メソッドは、引数で渡した文字列を名前とするネットワークコンポーネントの参照番号 ( ID ) を返します :

```
`...関数名: GetCompID
`...GetCompID (Component name) -> Component ID

C_STRING (255;$1;$Component)
C_LONGINT ($ErrCode;$Count;$CompID;$0;$i)

$ErrCode:=OP Count network components ($Count)

For ($i;1;$Count)
  $ErrCode:=OP Get network component info ($i;$CompID;$Component)
  If ($ErrCode=0) & (Position($1;$Component)>0)
    $0:=$CompID
    $i:=$Count+1
  End if
End for
```

このメソッド「GetCompID」の例題は :

```
C_LONGINT($CompID)
$CompID:=GetCompID ("MacTCP")

If ($CompID>0)
  `...エラーが起きていなければ
  `...ネットワークコンポーネントIDが返される。
End if
```

## 参照

OP Count network components

## OP Load network component

### OP Load network component (NetCompID) Error

引数	タイプ	説明
NetCompID	倍長整数	初期化したいネットワークコンポーネントの参照番号 (ID)
Error	倍長整数	エラーコード

#### 説明

**OP Load network component** は、 < NetCompID > 引数で指定したネットワークコンポーネントを初期化します。

4D Serverとの接続を開始する前には、必ずネットワークコンポーネントを初期化しなければなりません。既に初期化済みのネットワークコンポーネントに対して、再度 **OP Load network component** を実行しても何も行われません。また、その場合はエラーも返しません。

#### エラーコード

関数の実行が成功した場合、**OP Load network component** は0を返します。それ以外の場合は下記のエラーコードを返します：

エラーコード	説明
10128	4D Open for 4th Dimensionがメモリに読み込まれていません。
10129	指定されたネットワークコンポーネントが見つかりません。
10130	ネットワークコンポーネントの初期化に失敗しました。
10154	このコマンドを直ちに実行することができません。

#### 例題

この例題ではADSPネットワークコンポーネントを初期化しています。

```

C_LONGINT ($ErrCode;$CompID)
$CompID:=GetCompID ("ADSP")
`ADSPネットワークコンポーネントを選択している。
If ($CompID>0)
  $ErrCode:=OP Load network component ($CompID)
  `...ネットワークコンポーネントが初期化されれば
  `...4D Serverへの接続手順に移ることができる。
End if
    
```

#### 参照

OP Load network component、OP Unload network component

## OP Unload network component

---

### OP Unload network component (NetCompID) Error

引数	タイプ	説明
NetCompID	倍長整数	破棄したいネットワークコンポーネントの参照番号 ( ID )
Error	倍長整数	エラーコード

#### 説明

**OP Unload network component** は、 < NetCompID > 引数で指定された ID に対応するネットワークコンポーネントを破棄します。4D Server との接続を切断した後は、使用していたネットワークコンポーネントを破棄すべきです。

これにより、ネットワークコンポーネントによって占有されていたメモリブロックが開放されます。4D、4D Client 自体を終了する場合は、4D Open for 4th Dimension がすべてのネットワークコンポーネントを破棄します。

接続で使用中のネットワークコンポーネントを破棄しようとした場合には何も行ないません。また、この場合はエラーも返しません。

#### エラーコード

関数の実行が成功した場合、**OP Unload network component** は 0 を返します。それ以外の場合は下記のエラーコードを返します：

エラーコード	説明
10132	指定されたネットワークコンポーネントは使用中です。破棄できません。
10154	このコマンドを直ちに実行することができません。



## 例題

この例題では、ADSP ネットワークコンポーネントを初期化し、4D Server データベースを選択し、その4D Server との接続を開き、接続を閉じ、最後にネットワークコンポーネントを破棄しています。

```
C_STRING (31;$Server)
C_LONGINT ($ErrCode;$ServerID;$CompID;$ConnID)

$CompID:=GetCompID ("ADSP")      \...ADSP を選ぶ。

$ErrCode:=OP Load network component ($CompID)
$ErrCode:=OP Select 4D Server ($CompID;$Server;$ServerID)
$ErrCode:=OP Open connection ($ServerID;$ConnID;"PowerPC";"Designer";"";
                                "Data Synchronization")

If ($ErrCode=0)
    \データ同期を行う。
End if
$ErrCode:=OP Close connection ($ConnID)
$ErrCode:=OP Unload network component ($CompID)
```

## 参照

OP Load network component

## OP Request

---

### OP Request (connectID; status) Error

引数	タイプ	説明
connectID	倍長整数	接続ごとに割り振られたユニークID
status	ブール	リクエストを処理中か否か
Error	倍長整数	エラーコード

### 説明

**OP Request** を使用すると、非同期モードで実行されたリクエストに対する処理が完了しているかどうかを調べることができます。

4D Open for 4D は、同時に複数のリクエストを実行できません。**OP Request** が < status > 引数に TRUE を返せばリクエストに対する処理は完了していることを示し、そうでない場合は FALSE が返ります。

バージョン 6.0.6 以前の 4D Open for 4D では、非同期モードで実行中にリクエストに対する処理が終わっているかどうかを調べる方法がなかったため、新しいリクエストを送り、返ってきたエラーコードを見て判断するしかありませんでした(-32003;-32001;10154)。

### 参照

OP Open connection

## ネットワークユーティリティ序論

---

この章では、以下のルーチンについて説明します：

**OP Get station name**：動作しているマシンの名前を取得します。

**OP Start remote connection**：ARA ( Macintosh ) もしくはRAS ( Windows ) を使用してリモート接続を開始します。

**OP End Remote connection**：リモート接続を終了します。

**OP Remote Connection Status**：リモート接続の状態を調べます。

## OP Get station name

---

### OP Get station name Error

引数	タイプ	説明
----	-----	----

このコマンドには、引数はありません。

Error	固定長文字列	マシンの名称
-------	--------	--------

### 説明

**OP Get station name** は、動作しているマシンの名称を返します。

Macintosh 上では「ファイル共有」コントロールパネルで設定される名称です。

Windows 上では「ネットワーク」コントロールパネルで設定される名称です。

### 例題

```
C_LONGINT ($errCode;netCompID;$servID;$connID)
C_STRING (32;$WS)

$errCode := LoadNetComp ("TCP/IP";-> netCompID)
If ($errCode = 0)
    `4D Server のサーバIDを得る。
    $errCode := OP Find 4D Server (netCompID;"accounting.acius.com";$servID)
End if
If ($errCode =0)
    `マシン名を得る。
    $WS := OP Get station name
    `4D Serverへ接続する。
    $errCode := OP Open connection ($servID;$connID;$WS;"Jerry";"YRREJ";
                                     "DevSearch")
End if
```

## OP Start Remote Connection

**OP Start Remote Connection** (path; user; password; async) Error

引数	タイプ	説明
path	固定長文字列	パス
user	固定長文字列	ユーザ名
password	固定長文字列	パスワード
async	ブール	非同期接続するか否か
Error	倍長整数	エラーコード

### 説明

**OP Start Remote Connection** は、新たにリモート接続を開始します。Macintosh 上では ARA、Windows では RAS を使用します。

引数 < path > は、ARA では ARA クライアント書類へのファイルパス文字列です。RAS では電話帳の登録見出しです。引数 < path > が空だった場合は、オープンファイルダイアログを表示します (Macintosh のみ)。引数 < user > は ARA では使用されません。引数 < password > が空だった場合、**OP Start Remote Connection** コマンドは引数 < path > で指定された書類の中を探します。

### 例題

非同期接続の場合：

```

C_LONGINT (status1;$ErrCode;error)
C_STRING (100;devType;devName)
C_TEXT ($errorText)
$ErrCode:= OP Start Remote Connection ("Home";"MyName";"MyPassword";True)
Repeat
    $ErrCode:= OP Remote Connection Status (status1;error;devType;devName)
Until (status#1)
$errorText:= OP Get Error Text ($ErrCode)
...
$errorText:= OP End Remote Connection ()

```

同期接続の場合：

```

$errorCode:= OP Start Remote Connection ("Home";"MyName";"MyPassword";False)
`接続が完了するか、エラーが起きるかのどちらかまでは制御が戻ってこない。

```

### 参照

OP End Remote Connection、OP Remote Connection Status

## OP End Remote Connection

---

### OP End Remote Connection Error

引数	タイプ	説明
----	-----	----

このコマンドには、引数はありません。

Error	倍長整数	エラーコード
-------	------	--------

### 説明

**OP End Remote Connection** は **OP Start Remote Connection** で開始した接続を終了します。Macintosh 上では **OP End Remote Connection** は ARA の ioResultCode を返します。Windows 上では RAS ErrorCode を返します。

### 例題

```
C_LONGINT (status1;$ErrCode;error)
C_STRING (100;devType;devName)
C_TEXT ($ErrorText)
```

```
$ErrCode:= OP Start Remote Connection ("Home";"MyName";"MyPassword";True)
```

#### Repeat

```
$ErrCode:= OP Remote Connection Status (status1;error;devType;devName)
```

```
Until (status#1)
```

```
$ErrorText:= OP Get Error Text ($ErrCode)
```

```
...
```

```
$ErrorText:= OP End Remote Connection ()
```

### 参照

OP Remote Connection Status、OP Start Remote Connection

## OP Remote Connection Status

**OP Remote Connection Status** (phase; error; connectedTo; lastMessage) Error

引数	タイプ	説明
phase	倍長整数	リモート接続状況
error	倍長整数	リモート接続エラーコード
connectedTo	固定長文字列	接続先
lastMessage	固定長文字列	最新のメッセージ
Error	倍長整数	エラーコード

### 説明

**OP Remote Connection Status** は、**OP Start Remote Connection** で開始を試みたリモート接続の状況を調べるルーチンです。接続の開始時点でエラーが発生せず、接続手順が継続中の場合は、引数 < phase > は「1」を示します。

その他の値とその意味は：

- 0：接続完了
- 1：接続処理実行中
- 2：切断された

### 例題

```
C_LONGINT (status1;$ErrCode;error)
C_STRING (100;devType;devName)
C_TEXT ($errorText)
```

```
$ErrCode:= OP Start Remote Connection ("Home";"MyName";"MyPassword";True)
```

#### Repeat

```
$ErrCode:= OP Remote Connection Status (status1;error;devType;devName)
Until (status#1)
```

```
$errorText:= OP Get Error Text ($ErrCode)
```

```
...
```

```
$errorText:= OP End Remote Connection ()
```

### 参照

OP End Remote Connection、OP Start Remote Connection

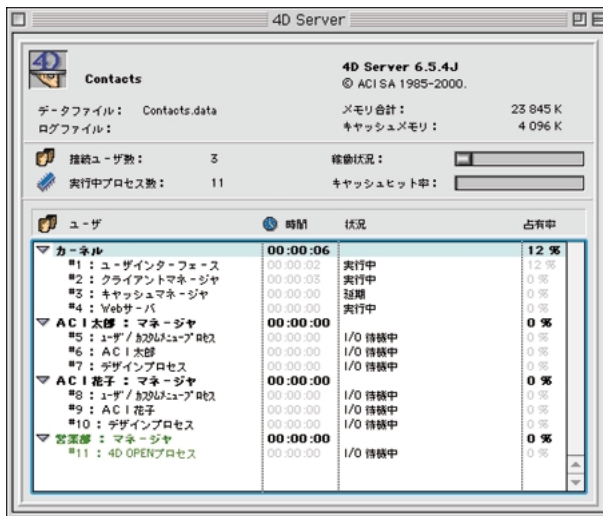




## 接続の開始

ネットワーク上の4D Serverデータベースが特定できたら、ルーチン **OP Open connection** を使用して接続を開始することができます。単一のデータベースへ複数の接続を行うこと、または同時に複数のデータベースを開くことが可能です。

サーバ側ではそれぞれの接続に対して新しいプロセスが起動します。カラーモニタを使用している場合、4D OPENによる接続に対応するプロセスが緑色で表示され、4D Clientによる接続と区別が付きやすくなっています。4D Serverのウィンドウにはこのように表示されます：



4D Serverのウィンドウには、クライアントマシン名、ユーザ名、プロセス名がクライアントごとに階層化されて表示されます。ここで表示される情報は**OP Open connection** ルーチンを使って指定します。

## プロセス

4D Open for 4Dを使って開始した最初の接続で、クライアントとしてサーバへ登録されず。また、同時にサーバ側で接続に対応する新しいプロセスがスタートします。同じクライアントから再び同じデータベースへ接続を行うと、サーバ側に、更にもう1つ新しいプロセスが起動します。1つのデータベースへ複数回接続を繰り返すことにより、複数のプロセスを4D Server上で実行させることができます。

4D Server側から見ればそれぞれのプロセスは別々の接続として振る舞うため、それぞれの接続で異なるデータベース処理を行うために、一対のクライアントとサーバの間であっても、複数の接続を開始することが可能です。例えば、1つのプロセスでレコードの一覧を取得、表示を行い、別のプロセスでユーザにレコード追加をさせるといったこともできるのです。

## 接続の終了

接続の終了には**OP Close connection**を使用します。

### 4D Client 以外のアプリケーションに 4D Server へのアクセス権を持たせる

4D Client以外のアプリケーションも、この4D Open for 4Dを使用することによって4D Serverのデータベースへ接続できるようになりますが、そのためには「4D Openの接続を許可する」のチェックボックスをONにしておく必要があります。このチェックボックスは「データベースプロパティ」ダイアログの中にあります。

このチェックボックスがONになっていなければ、4D OPENを使用した接続はできず、4D OPENルーチンで接続を試みた場合にはエラーとなります。エラーコードは以下の通りです：

- 4D Serverバージョン1.1：error -9956
- 4D Serverバージョン1.1.1以降：error -9947
- 4D Serverバージョン1.1.1以降：接続を試みたユーザの所属するパスワードグループに4D OPENを使用する権限が与えられていない場合にもエラーとなります。

### 4D Open グループへアクセス権を与える

4D Serverバージョン1.1.1以降では、デザイナーは4D OPEN経由でデータベースへ接続できる特別なアクセス権限をパスワードグループに対して与えることができます。

忘れてはならないことは、あくまでデータベースプロパティの「4D Openの接続を許可する」のチェックボックスがONになっていることが前提で、これがOFFのデータベースへは、いかなるアクセス権を持つグループも接続ができないということです。あるグループへ4D OPEN経由の4D Serverアクセス権を与えるためには、「パスワードエディタ」ダイアログを開き、「プラグインアクセス権...」のメニュー項目を選択して現れるアクセス管理画面で4D OPENを選択し、「グループアクセス」ポップアップで目的のグループを選択してください。グループが1つも作られていない場合は、デフォルトですべてのグループが4D OPENのアクセス権を持ちます。1つでもグループが作られていれば、デフォルトではどのグループへも4D OPENアクセス権は与えられません。この場合、4D OPEN経由の接続を試みると-9944のエラーが返ってきます。

接続ダイアログボックスを表示させるためには、**OP Select 4D Server**を使います。この場合、ユーザは接続先データベースを、その時点で起動中の4D Serverで動いているデータベースの一覧から選択することになります。

ユーザに選択操作をさせずに接続先データベースを探すためには、**OP Find 4D Server**を使用します。

データベースを選択したら、**OP Open connection**を使用して接続を開始できます。接続を閉じるためには**OP Close connection**を使用します。

**OP Select 4D Server**、**OP Find 4D Server**で指定されたデータベースサーバの記録を保持するために割り当てられたメモリスペースを開放するためには、**OP Delete 4D Server**を使います。

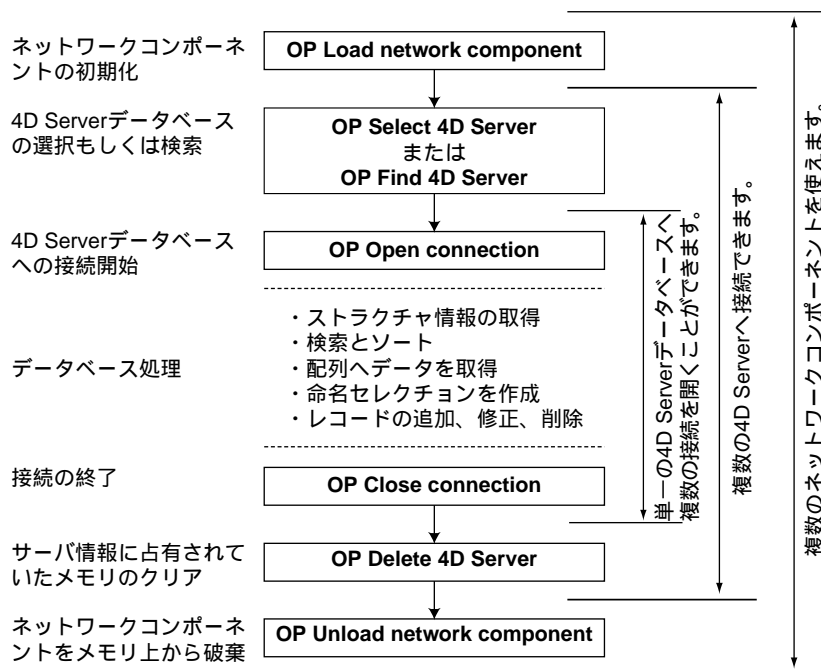
#### 参照

OP Close connection、OP Delete 4D Server、OP Find 4D Server、OP Open connection、OP Select 4D Server

## 接続、序論

この章で説明するルーチンを使って4D Serverデータベースを選択します。

下図は、どのような流れで4th Dimension、4D Client、場合によってはストアードプロシージャを使った4D Server自身が、目的の4D Serverデータベースへ接続していくのかを説明しています。



### 複数の接続、複数のデータベース

4D Openルーチンを使用すれば、単一のクライアントからでも複数の接続を実行できます。同時に複数のデータベースへ接続することも可能です。それぞれの接続には個々に対応する4D Serverプロセスがあります。

4D OPENを使用すれば、1ユーザあたり同時に最大6つまで接続を開くことが可能です。6つの接続を実行中にさらに接続するには、別のユーザ権限で接続を開始してください。そのユーザ権限で、さらに最大6接続まで実行できます。

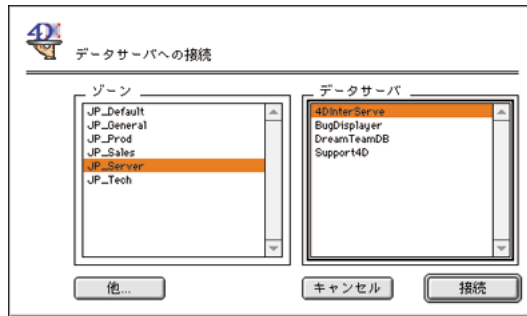
最大ユーザ数は4D Serverにインストールされたユーザライセンス数に制限されます。

## 4D Server データベースへの接続

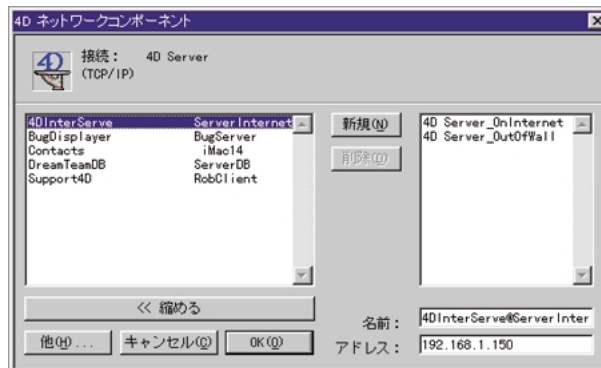
4D Openを使用すれば、サーバマシン上で動作中の4D Serverデータベースへ接続できます。この段階では、使用したいネットワークコンポーネントを読み込み、データベースを選択します。

**OP Select 4D Server**を実行すれば、ユーザに4D Serverデータベース選択のためのダイアログボックスを提示できます。**OP Select 4D Server**の代わりに**OP Find 4D Server**を使う場合は、データベース名を指定しなければなりません。

**OP Select 4D Server**は、「データベースサーバへの接続」というダイアログボックスを表示します。複数のゾーンが設定されたネットワークでADSPネットワークコンポーネントを使用すると、ダイアログボックスは以下のような表示になります：



TCP/IPネットワークコンポーネントを使用していれば、以下のように表示されます：



複数のネットワークコンポーネントをインストールしている場合は、ダイアログの中に「他...」ボタンが現れます。「他...」ボタンをクリックすれば、**OP Select 4D Server**は戻り値< Error >に「2」を返します。このコードをチェックし、それに対応した処理を行うのはプログラマの責任であることに注意してください。例えば、他のネットワークコンポーネントを選び、**OP Select 4D Server**を再実行するという処理です。

4D Serverへの接続の開始についての詳細は、『4D Serverリファレンス』マニュアルを参照してください。

## OP Select 4D Server

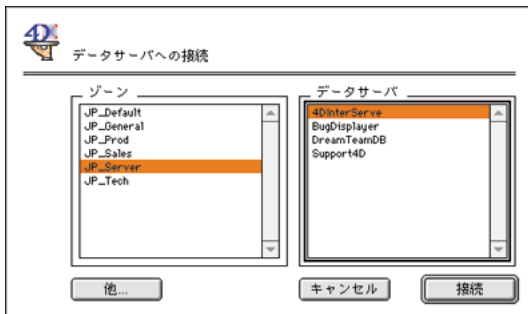
**OP Select 4D Server** ( NetCompID; serverName; serverID; otherButton ) Error

引数	タイプ	説明
NetCompID	倍長整数	ネットワークコンポーネントID
serverName	テキスト	ダイアログで選択されたサ - バの名称
serverID	倍長整数	ダイアログで選択されたサ - バのID
otherButton	ブール	「他...」ボタンを表示するか？
Error	倍長整数	エラーコード

### 説明

**OP Select 4D Server** は、接続するデータベースを選択するためのダイアログを表示します。ダイアログは引数 < netCompID > で指定したネットワークコンポーネント用のものになります。

ADSPネットワークコンポーネントを使用した場合のダイアログの例題です：



選択したサーバ名が、引数 < serverName > に返されます。得られたサーバ名をどこか（フィールド、変数など）に保存しておけば、後で **OP Find 4D Server** を実行する際に利用することができます。

- ・ ADSPを使用した場合、サーバ名文字列はストラクチャファイル名 + “:” + ネットワーク公開名 + “@” + ネットワークゾーン名の文字列です。ネットワーク公開名とは 4D Server がネットワーク上に公開する名前です。デフォルトの名前は「4D Server 1.5」です。4D Server のバージョンが 6.5 の場合は「4D Server 6.5」です。公開名のデフォルトとしての文字列には「4D Server 6.0」は存在しません。例えば、“サポート”ゾーンにある“顧客データベース”を 4D Server 6.0 で起動している場合であれば、以下のようになります：

"顧客データベース:4D Server 1.5@サポート"

- ・ 単一のゾーンで構成されたネットワークで動作している場合のゾーン名文字列は“\*”（アスタリスク）になります。

- ・ TCP/IPを使用してサーバを選択した場合、引数 < serverName > にはストラクチャファイル名 + タブ + サーバマシン名が返ってきます。「データベース選択」ダイアログを拡張し、IPアドレスを手入力してサーバを選択した場合は、IPアドレス文字列が返ってきます。ドメイン名で入力した場合はDNS文字列が返ってきます。例えばIPアドレス 195.4.210.25、マシン名「ServerMac\_1」にあるデータベース「顧客DB」を選択すると以下の文字列が返ってきます：

"顧客DB ServerMac\_1"

- ・ IPアドレスを直接入力した場合は：

"195.4.210.25"

インターネット上に公開されたマシン「www.4d.com」で公開されたデータベースへ接続した場合は：

"www.4d.com"

が返ってきます。

ダイアログで選択されたサーバのIDは引数 < serverID > に返ってきます。このIDは後の接続段階で **OP Open connection** へ渡す値です。

ダイアログの「他...」ボタンを押せば、このコマンドの実行は終了し、戻り値として「2」が返ってきます。この場合、ユーザに他のネットワークコンポーネントを選択させるために、カスタムダイアログを表示しても構いません。例えば、このようなダイアログです：





## エラーコード

**OP Select 4D Server**の実行に成功した場合、戻り値 < Error > は「0」です。そうでなければ、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
-10020	サーバが選択されませんでした。 (「キャンセル」ボタンが押されました)
2	「サーバ選択」ダイアログで「他...」ボタンが押されました。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10129	指定されたネットワークコンポーネントが見つかりません。
10131	ネットワークコンポーネントが読み込まれていません。
10154	このコマンドを今すぐには実行できません。

## 例題

以下のメソッドは、リストからデータベースを選び、接続を開始します。

```

C_STRING (255;$Server)
C_LONGINT ($ErrCode;$CompID;$ServerID)
$ErrCode:= OP Set option (2;0)           \...エラーダイアログ非表示設定。
Repeat
    $CompID:=SelectComp                 \...ネットワークコンポーネント選択。
    $ErrCode:= OP Load network component ($CompID)
    $ErrCode:= OP Select 4D Server ($CompID;$Server;$ServerID)
Until ($ErrCode#2)
    
```

接続ダイアログボックスでは、他のネットワークコンポーネントを選ぶために「他...」ボタンを押すことができます。ユーザが「他...」ボタンを押せば、\$ErrCode=2となり、再びネットワークコンポーネント選択のためのメソッド"SelectComp"を実行する流れになっています。

```

\...関数: SelectComp
\...SelectComp -> 選択したネットワークコンポーネント ID。
C_STRING (255;$Component)
C_LONGINT ($ErrCode;$Count;$CompID;$i;$0)
$ErrCode:= OP Count network components ($Count)
ARRAY LONGINT (aCompID;$Count)
ARRAY STRING (255;aComponent;$Count)
For ($i;1;$Count)
    $ErrCode:= OP Get network component info ($i;$CompID;$Component)
    If ($ErrCode=0)
        aCompID{$i}:=$CompID
        aComponent{$i}:=$Component
    End if
End for
    
```

*Center Window (187;129)*

**DIALOG** ([Dialogs;"Select Comp")

　　`...ネットワークコンポーネント選択のためのダイアログを表示。

**CLOSE WINDOW**

**If** (OK=1) & (aComponent>0)

　　\$0:=aCompID{aComponent}

**End if**

参照

OP Get network component info、OP Load network component

## OP Find 4D Server

### OP Find 4D Server (netCompID; serverName; serverID) Error

引数	タイプ	説明
netCompID	倍長整数	ネットワークコンポーネントID
serverName	固定長文字列	探したいサーバ名
serverID	倍長整数	発見したサーバを示す固有のID
Error	倍長整数	エラーコード

### 説明

**OP Find 4D Server** は、引数 < netCompID > で指定したネットワークコンポーネントを使用して、引数 < serverName > で指定したデータベースを探します。

- ADSPを使用する場合は、引数 < serverName > には、ストラクチャファイル名 + “:” + ネットワーク公開名 + “@” + ネットワークゾーン名のフォーマットの文字列を指定します。ネットワーク公開名とは4D Serverがネットワーク上に公開する名前で、デフォルトの名前は「4D Server 1.5」です。4D Serverのバージョンが6.5の場合は「4D Server 6.5」です。公開名のデフォルトとしての文字列には「4D Server 6.0」は存在しません。例えば、「サポート」ゾーンで4D Server 6.0で稼働中のデータベース「顧客DB」を探している場合は、

"顧客DB:4D Server 1.5@サポート"

になります。

- 単一のゾーンで構成されたネットワークで動作している場合のゾーン名文字列は“\*” (アスタリスク) になります。
- TCP/IPを使用してサーバを探す場合、引数 < serverName > にはストラクチャファイル名 + タブ + サーバマシン名のフォーマットで文字列を渡します。直接IPアドレスを指定してデータベースを探す場合は、IPアドレス文字列を渡します。

例えば、IPアドレス192.9.200.13にあるマシン「PCforServer」で稼働中のデータベース「顧客DB」を探す場合は、

"顧客DB.4db PCforServer"

IPアドレス指定の場合は、

"192.9.200.13"

インターネット上に公開されているマシンへ接続する場合には、ドメインネームでマシンを指定することも可能です。

例えば、インターネット上に公開されたマシン「www.4d.com」で稼働中データベースへ接続する場合は、

```
"www.4d.com"
```

になります。

サーバIDは引数 < serverID > に返ってきます。この値はその後、このサーバへ接続を開始する段階でコマンド **OP Open connection** へ渡します。

## エラーコード

**OP Find 4D Server** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
-10021	サーバが見つかりません。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10129	指定されたネットワークコンポーネントが見つかりません。
10131	ネットワークコンポーネントが読み込まれていません。
10154	このコマンドを今すぐには実行できません。

## 例題

以下のメソッドでは、ネットワークコンポーネントを読み込み、「Example DB」を探し、接続を行っています。

```
C_STRING (80;$Server;$Zone;Database;$NBPName)
C_LONGINT ($ADSP;$ErrCode;$ServerID;vConnectID)

$Zone:= "*"           `カレントゾーン
$NBPName:= "4D Server 6.5" `ネットワーク公開名
$Database:= "Example DB"
$ADSP:= GetCompID ("ADSP")
$Server:= $Database+ "."+ $NBPName+ "@"+ $Zone

$ErrCode:= OP Load network component ($ADSP)
$ErrCode:= OP Find 4D Server ($ADSP;$Server;$ServerID)
$ErrCode:= OP Open connection ($ServerID;vConnectID;"4D Server用マシン";
                                "Hajime Kobayashi";";"データ同期")
`...データベース操作を行う...

$ErrCode:= OP Close connection (vConnectID)
$ErrCode:= OP Delete 4D Server ($ServerID)
```

## 参照

OP Delete 4D Server、OP Open connection、OP Select 4D Server

## OP Delete 4D Server

---

### OP Delete 4D Server (serverID) Error

引数	タイプ	説明
serverID	倍長整数	消去したいサーバを示す固有のID
Error	倍長整数	エラーコード

#### 説明

**OP Delete 4D Server**は4D Open クライアントマシンのメモリ上から、ネットワーク上のサーバの位置を追跡するための情報を消去し、メモリを開放します。

このルーチンは対象となるサーバへのすべての接続を切断した後に実行してください。

1つでも接続が残っているサーバに対してこのルーチンを実行した場合は、このルーチンは何も行いませんが、エラーコードは返します。

#### エラーコード

**OP Delete 4D Server**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10133	指定されたサーバの記録がありません。
10134	サーバは使用中です。サーバ情報を消去できません。
10154	このコマンドを今すぐには実行できません。

#### 参照

OP Find 4D Server、OP Select 4D Server

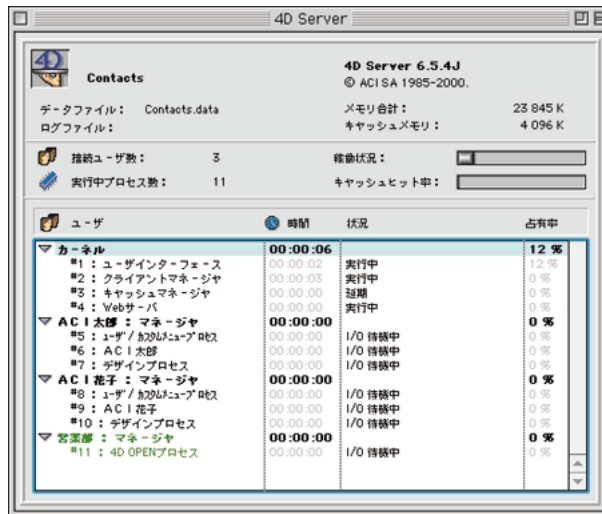
## OP Open connection

**OP Open connection** (serverID; connectionID; station; userName; password; taskName) Error

引数	タイプ	説明
serverID	倍長整数	対象とするサーバを示す固有のID
connectionID	倍長整数	開始された接続を識別する固有のID
station	固定長文字列	4D Serverの画面に表示させるログインマシン名
userName	固定長文字列	ログイン権限認証に使用するユーザ名 (4Dパスワードシステムで登録されているもの)
password	固定長文字列	ログイン権限認証に使用するパスワード (4Dパスワードシステムで登録されているもの)
taskName	固定長文字列	4D Serverの画面に表示させるプロセス名
Error	倍長整数	エラーコード

### 説明

**OP Open connection** は、引数 < serverID > で指定された4D Serverに対し、引数 < userName >、< password > の権限で接続を開始します。接続が成功した場合、サーバから引数 < stationName > で指定した名称の端末機からログインし、引数 < taskName > で指定した名称のプロセスを実行する1クライアントとして認識されます。引数 < connectionID > には接続を識別する固有のID番号が返されます。



## エラーコード

**OP Open connection**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-12	ライセンスされた最大ユーザ数に達しました。
-13	ネットワーク上の問題で接続に失敗しました。再度接続してください。
-108	このコマンドを実行するのに必要なメモリが足りません。
-1277	接続リクエストが取り消されました (ADSP タイムアウトエラー)。
-9944	4D OPEN アクセス権のないユーザです。
-9947	データベースプロパティの「4D OPEN の接続を許可する」チェックボックスがOFFです。
-9956	接続先の4D Serverとバージョン間の互換性がありません。バージョン1.1の4D Serverへ接続しようとした場合はデータベースプロパティの「4D OPEN の接続を許可する」チェックボックスがOFFです。
-9978	< password > が間違っています。
-9979	未登録USERです。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10129	指定されたネットワークコンポーネントが見つかりません。
10133	指定されたサーバの記録がありません。
10154	このコマンドを今すぐには実行できません。

## 参照

OP Close connection、OP Find 4D Server

## OP Close connection

---

### OP Close connection (connectionID) Error

引数	タイプ	説明
connectionID	倍長整数	閉じようとしている接続 ID
Error	倍長整数	エラーコード

#### 説明

**OP Close connection** は、引数 < connectionID > で指定された接続を終了します。

#### エラーコード

**OP Close connection** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

#### 参照

OP Open connection



## ストラクチャ情報、序論

---

この章で説明するルーチンを使用すれば、4D Server上で稼働しているデータベースのストラクチャ情報を取得することが可能です。名前、テーブルやフィールドの数、各種属性などを、以下のルーチンを使用して取得することが可能です：

**OP Count tables**：テーブルの総数を返します。

**OP Get all tablenames**：すべてのテーブルの名前を返します。

**OP Get field properties**：テーブルとそのテーブルに含まれるフィールドについて基本的な情報を返します。

**OP Get table properties**：テーブルとそのテーブルに含まれるフィールドについて、より詳細な情報を返します。

**OP Get one field number**：指定したフィールドのフィールド番号と所属するテーブル番号を返します。

**OP Get all field numbers**：指定した複数のフィールドのフィールド番号と所属するテーブル番号を返します。

**OP Get one tablename**：指定したテーブルの名前を返します。

**OP Count fields**：指定したテーブルに含まれるフィールドの数を返します。

**OP Cache structure**：クライアントマシンのメモリ上に4D Serverデータベースのストラクチャ定義をキャッシュします。

### 参照

OP Cache structure、OP Count fields、OP Count tables、OP Get all field numbers、OP Get all tablenames、OP Get field properties、OP Get one field number、OP Get one tablename、OP Get table properties

## OP Count tables

---

### OP Count tables (connectionID; numOfTables) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
numOfTables	倍長整数	テーブルの数
Error	倍長整数	エラーコード

### 説明

**OP Count tables** は、引数 < numOfTables > にテーブルの数を返します。このコマンドが対象とするデータベースは引数 < connectionID > で示される接続が対象としている 4D Server で稼働中のものです。

### エラーコード

**OP Count tables** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 例題

この例題では、まず接続先データベースのテーブル数を調べています。得られた値はその後で行うフィールド数を調べる処理のループ数として使われています。

```
C_LONGINT ($i;vTablesInDB;vFieldNum)
```

```
`... (サーバへの接続処理部分は省略)
```

```
$ErrCode:= OP Count tables ($ConnectID;vTablesInDB)
```

```
ARRAY STRING (15;aFieldNum;vTablesInDB)
```

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

```
    $ErrCode:= OP Count fields ($ConnectID;$i;vFieldNum)
```

```
    aNumFields{$i}:=$vFieldNum
```

```
End for
```

### 参照

Count tables、OP Count fields

## OP Get all tablenamees

**OP Get all tablenamees** (connectionID; tableNames) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableNames	配列	すべてのテーブルの名前
Error	倍長整数	エラーコード

### 説明

**OP Get all tablenamees** は、引数 < tableNames > にすべてのテーブルの名前を返します。配列の型は固定長文字列でも TEXT でも構いません。このコマンドが対象とするデータベースは、引数 < connectionID > で示される接続が対象としている 4D Server で稼働中のものです。

### エラーコード

**OP Get all tablenamees** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10135	引数型が不正です。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 例題

この例題では配列 "aTableNames" にすべてのテーブルの名前を取得しています。その後で開いているダイアログで配列 "aTableNames" をスクロールエリア（もしくは各種ポップアップ）で提示し、ユーザにテーブルを選択させるインタフェースとして使用しています。

```
ARRAY STRING (15;aTableNames;0)
$ErrCode:= OP Get all tablenamees (vConnectID;aTableNames)
```

```
CenterWindow (150;191)
DIALOG ([Dialogs];"Show Tables")
CLOSE WINDOW
```

```
If (aTableNames>0)
  \...ユーザはテーブルを選択した。
  \...選択したテーブルに関する処理を行う。
End if
```

参照

OP Get one tablename、Table name

## OP Get field properties

---

**OP Get field properties** (connectionID; tableID; tableInvisible; fieldNames;

fieldTypes;fieldLengths; fieldIndexed; fieldInvisible) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	対象とするテーブル番号
tableInvisible	倍長整数	テーブルの不可視属性
fieldNames	配列	対象とするテーブルに属する全フィールド名
fieldTypes	配列	対象とするテーブルに属する全フィールドのタイプ
fieldLengths	配列	対象とするテーブルに属する全フィールドの割り当て文字列長
fieldIndexed	配列	対象とするテーブルに属する全フィールドのインデックス属性
fieldInvisible	配列	対象とするテーブルに属する全フィールドの非表示属性
Error	倍長整数	エラーコード

### 説明

**OP Get field properties** は、指定したテーブルのすべてのフィールドの名前と型を返します。また、すべての固定長文字列型フィールドの割当文字数、インデックス、非表示属性の有無なども取得できます。

引数 < tableInvisible > に「1」が返った場合、引数 < tableID > で指定したテーブルは非表示です。そうでない場合は、引数 < tableInvisible > は「0」になっています。

引数 < fieldNames > は固定長文字列型、もしくはテキスト型の配列でなければなりません。

引数 < fieldTypes > は固定長文字列型、テキスト型、実数型、整数型、倍長整数型のいずれかの配列でなければなりません。返ってくる値は渡した配列の型によって以下のようになります：

文字型配列	数値型配列
(固定長文字列、TEXT型)	(実数、整数、倍長整数型)
"Alphanumeric"	0
"Real"	1
"Text"	2
"Picture"	3
"Date"	4
"Boolean"	6
"Subfile"	7
"Integer"	8
"Long Integer"	9
"Time"	11

引数 < fieldLengths > は固定長文字列型、テキスト型、実数型、整数型、倍長整数型のいずれかの配列でなければなりません。固定長文字列型、テキスト型の配列を渡した場合に、返される値は数値を表す文字列です。実数型、整数型、倍長整数型の配列を渡した場合に、返される値は数値です。引数 < Indexed > には、渡した配列の型によって以下のような値が返されます：

	文字型配列	数値型配列	ブール配列
インデックス付き	"Indexed"	1	True
インデックスなし	""(空文字列)	0	False

引数 < InvisibleField > には、渡した配列の型によって以下のような値が返されます：

	文字型配列	数値型配列	ブール配列
"非表示"ON	"Invisible"	1	True
"非表示"OFF	""(空文字列)	0	False

## エラーコード

**OP Get fields properties** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10135	引数型が不正です。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

## 例題

この例題では、配列は特定のテーブルに属するフィールドの情報を収めておくのに使用されています。

```
C_LONGINT ($Table;$ErrCode)
```

```
ARRAY LONGINT (aTypes;0) `フィールド型
ARRAY LONGINT (aLength;0) `フィールド長(固定長文字列型フィールドのために)
ARRAY LONGINT (aIndexes;0) `インデックス付き?
ARRAY LONGINT (aInvisible;0) `非表示?
ARRAY STRING (15;aNames;0) `フィールド名
```

```
$Table:=1 `番号 1 のテーブルについて情報を取得する。
```

```
$ErrCode:= OP Load network component ($CompID)
```

```
$ErrCode:= OP Open connection ($ServerID;$ConnectID;"iMac";"デザイナ";
                                $Password;"Field 情報取得")
```

```
$ErrCode:=OP Get field properties ($ConnectID;$Table;$Invisible;aNames;
                                    aTypes;aLength;aIndexes;aInvisible)
```

## 参照

GET FIELD PROPERTIES

## OP Get table properties

---

**OP Get table properties** (connectionID; tableID; tableInvisible; tableLeft; tableTop; tableRight; tableBottom; fieldNames; fieldTypes; fieldLength; fieldIndexed; fieldUnique; fieldInvisible; fieldEnterable; fieldModifiable; fieldMandatory; fieldRelatedTable; fieldRelatedField; fieldRelatedWild; fieldManyToOne; fieldOneToMany; fieldDelControl; fieldWhatControl; fieldAutoAssign) Error

引数	タイプ	説明
connectionID	倍長整数	対象となるデータベースへの接続ID
tableID	倍長整数	対象となるテーブルの番号
tableInvisible	倍長整数	テーブルの"不可視"属性
tableLeft	倍長整数	ストラクチャウィンドウでの位置(左)
tableTop	倍長整数	ストラクチャウィンドウでの位置(上)
tableRight	倍長整数	ストラクチャウィンドウでの位置(右)
tableBottom	倍長整数	ストラクチャウィンドウでの位置(下)
fieldNames	配列	フィールド名前
fieldTypes	配列	フィールドの型
fieldLength	配列	フィールドの割当文字列長
fieldIndexed	配列	フィールドのインデックス属性
fieldUnique	配列	フィールドの重複不可属性
fieldInvisible	配列	フィールドの不可視属性
fieldEnterable	配列	フィールドの入力可能属性
fieldModifiable	配列	フィールドの修正可属性
fieldMandatory	配列	フィールドの必須入力属性
fieldRelatedTable	配列	リレート先テーブル番号
fieldRelatedField	配列	リレート先フィールド番号
fieldRelatedWild	配列	リレート先ワイルドカードフィールド番号
fieldManyToOne	配列	リレートの"n対1"の自動リレート属性
fieldOneToMany	配列	リレートの"1対n"の自動リレート属性
fieldDelControl	配列	リレートの"削除制限"属性
fieldWhatControl	配列	リレートの"削除制限"属性の動作
fieldAutoAssign	配列	リレートの"サブフォームにリレート値を自動代入"属性
Error	倍長整数	エラーコード



## 説明

**OP Get table properties** は、テーブルとフィールドについて詳細な情報を返します。

引数 `< fieldTypes >` にはテキスト、固定長文字列、整数、倍長整数、実数型のいずれかの配列を渡します。

返ってくる値は渡した配列型に応じて以下ようになります：

文字配列	数値配列
"Alphanumeric"	0
"Real"	1
"Text"	2
"Picture"	3
"Date"	4
"Boolean"	6
"Subfile"	7
"Integer"	8
"Long Integer"	9
"Time"	11

引数 `< fieldLength >` にはテキスト、固定長文字列、整数、倍長整数、実数型のいずれかの配列を渡します。テキスト型か固定長文字列型を渡した場合は、文字として割当文字列長の数値が返されます。

引数 `< fieldRelatedTable >`、`< fieldRelatedField >`、`< fieldRelatedWild >` には整数、倍長整数、実数型のいずれかの配列を渡します。引数 `< TableID >` で指定されたテーブルがリレーの設定された「1テーブル」に当たる場合は、引数 `< fieldRelatedTable >` にはリレー先のテーブル番号だけを返します。

引数 `< fieldIndexed >`、`< fieldInvisible >`、`< fieldUnique >`、`< fieldEnterable >`、`< fieldModifiable >`、`< fieldMandatory >` にはテキスト、固定長文字列、整数、倍長整数、実数、ブール型のいずれかの配列を渡します。返ってくる値は渡した配列型に応じて以下ようになります：

	文字	数値	ブール
インデックス付き	"Indexed"	1	True
インデックスなし	""(空文字列)	0	False
"非表示"ON	"Invisible"	1	True
"非表示"OFF	""(空文字列)	0	False
"重複不可"ON	"Unique"	1	True
"重複不可"OFF	""(空文字列)	0	False
"表示のみ"ON	""(空文字列)	0	False
"表示のみ"OFF	"Enterable"	1	True
"修正不可"ON	""(空文字列)	0	False

"修正不可"OFF	"Modifiable"	1	True
"必須入力"ON	"Mandatory"	1	True
"必須入力"OFF	""(空文字列)	0	False

引数 < fieldManyToOne >、< fieldOneToMany >、< fieldDelControl >、< fieldWhatControl >、< fieldAutoAssign > には整数、倍長整数、実数、ブール型のいずれかの配列を渡します。4D Open は型に応じて 1 (true) もしくは 0 (false) を数値配列へ、True もしくは False をブール配列へ返します。

## エラーコード

**OP Get table properties** の実行に成功した場合、戻り値 < Error > は「0」です。そうでなければ、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10135	引数型が不正です。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

## 例題

以下のコードでは非表示属性が「ON」に設定されたテーブルとフィールドを一覧表示します。

```
C_LONGINT ($Invisible;$L;$T;$R;$B;$ErrCode;$FileID;$FldID)
```

```
ARRAY STRING (31;artName;0)
```

```
`ConnID には既に有効な接続 ID が入っているものとして ...
$ErrCode:= OP Get all tablenamees (ConnID;artName)
```

```
If ($ErrCode=0)
```

```
  ARRAY STRING (31;arFldName;0) ` 指定可能な型は固定長文字列、テキスト
  ARRAY INTEGER (arFldType;0)   ` 指定可能な型は整数、倍長整数、実数、
                                固定長文字列、テキスト
  ARRAY INTEGER (arFldLen;0)    ` 指定可能な型は整数、倍長整数、実数、
                                固定長文字列、テキスト
  ARRAY INTEGER (arFldInd;0)    ` 指定可能な型は整数、倍長整数、実数、
                                ブール、固定長文字列、テキスト
  ARRAY BOOLEAN (arFldInv;0)    ` 指定可能な型は整数、倍長整数、実数、
                                ブール、固定長文字列、テキスト
  ARRAY INTEGER (arFldUni;0)    ` 指定可能な型は整数、倍長整数、実数、
                                ブール、固定長文字列、テキスト
  ARRAY BOOLEAN (arFldEnt;0)    ` 指定可能な型は整数、倍長整数、実数、
                                ブール、固定長文字列、テキスト
```

- ARRAY BOOLEAN (arFldMod;0) ` 指定可能な型は整数、倍長整数、実数、  
ブール、固定長文字列、テキスト
- ARRAY BOOLEAN (arFldMan;0) ` 指定可能な型は整数、倍長整数、実数、  
ブール、固定長文字列、テキスト
- ARRAY INTEGER (arFldRelFl;0) ` 指定可能な型は整数、倍長整数、実数
- ARRAY INTEGER (arFldRelFd;0) ` 指定可能な型は整数、倍長整数、実数
- ARRAY INTEGER (arFldDis;0) ` 指定可能な型は整数、倍長整数、実数
- ARRAY BOOLEAN (arFldAutM2O;0) ` 指定可能な型は整数、倍長整数、実数、  
ブール
- ARRAY BOOLEAN (arFldAutO2M;0) ` 指定可能な型は整数、倍長整数、実数、  
ブール
- ARRAY BOOLEAN (arFldDel;0) ` 指定可能な型は整数、倍長整数、実数、  
ブール
- ARRAY BOOLEAN (arFldRej;0) ` 指定可能な型は整数、倍長整数、実数、  
ブール
- ARRAY BOOLEAN (arFldUpd;0) ` 指定可能な型は整数、倍長整数、実数、  
ブール

```

For ($tableID;1;Size of array (artName))
  $ErrCode:= OP Get table properties (ConnID;$tableID;$Invisible;$L;$T;$R;
    $B; arFldName;arFldType;arFldLen;arFldInv;arFldInd;
    arFldUni;arFldEnt; arFldMod;arFldMan;arFldRelFl;
    arFldRelFd;arFldDis;arFldAutM2O; arFldAutO2M;
    arFldDel;arFldRej;arFldUpd)
  If ($ErrCode=0)
    If ($Invisible#0)
      MESSAGE ("テーブル["+artName{$tableID}+"]は非表示です.")
    Else
      For ($FldID;1;Size of array (arFldName))
        If (arFldInv{$FldID})
          MESSAGE ("フィールド["+artName{$tableID}+"]"+
            arFldName{$FldID}+"]は非表示です")
        End if
      End for
    End if
  End for
End if

```

参照  
OP Get all tablenamees

## OP Get one field number

---

**OP Get one field number** (connectionID; tableField; tableID; fieldID) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続 ID
tableField	固定長文字列	フィールド名 (" [table]field" のフォーマットで)
tableID	倍長整数	指定したフィールドの属するテーブル番号
fieldID	倍長整数	指定したフィールドの番号
Error	倍長整数	エラーコード

### 説明

**OP Get one field number** は、引数 < tableField > に "[TableName]fieldName" の形式で指定したフィールドのテーブル番号とフィールド番号を引数 < tableID > と < fieldID > に得ます。テーブル名 / フィールド名はケースセンシティブではありません。

### エラーコード

**OP Get one field number** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10146	不正なフィールド名です。
10154	このコマンドを今すぐには実行できません。

### 例題

この例題では、"[Customer]Full name" フィールドのテーブル番号とフィールド番号を調べています：

```
C_LONGINT ($ErrCode;vTable;vField)

$ErrCode:= OP Get one field number (vConnectID;"[Customer]Full Name";vTable;
                                     vField)

If ($ErrCode=0)
    ALERT ("テーブル番号:"+String (vTable)+Char (13)+"フィールド番号:"+String
          (vField))
End if
```

### 参照

Field、OP Get all field numbers、OP Get field properties

## OP Get all field numbers

---

**OP Get all field numbers** (connectionID; tableFields; tableIDs; fieldIDs) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableFields	配列	フィールド名 (" [table]field" フォーマットで)
tableIDs	配列	テーブル番号
fieldIDs	配列	フィールド番号
Error	倍長整数	エラーコード

### 説明

**OP Get all field numbers** は、引数 < tableFields > に渡したテーブルとフィールドの名前 (" [table]field" フォーマットで) に対応するテーブル番号、フィールド番号を引数 < tableIDs >、< fieldIDs > に返します。

引数 < tableFields > はテキストか固定長文字列型の配列でなければなりません。また、その配列へ渡すフィールドを表す文字列は、"[table]field" のフォーマットに従っている必要があります。

テーブルとフィールドの名前は大文字 / 小文字の区別はしません。

引数 < tableIDs > と < fieldIDs > は整数、倍長整数、実数型のいずれかの配列でなければなりません。

### エラーコード

**OP Get all field numbers** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
-9981	不正なテーブル / フィールド指定配列が送られました。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10135	引数型が不正です。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。
10159	配列要素数の最大値を越えました。

## 例題

この例題では、配列"aFldNames"の各要素に対応したフィールドのテーブル番号、フィールド番号が返されます：

```
C_LONGINT ($ErrCode)
```

```
ARRAY LONGINT (aTables;6)
```

```
ARRAY LONGINT (aFields;6)
```

```
ARRAY STRING (15;aFldNames;6)
```

```
aFldNames{1}:="[People]First Name"
```

```
aFldNames{2}:="[People]Last Name"
```

```
aFldNames{3}:="[Company]Address"
```

```
aFldNames{4}:="[Company]City"
```

```
aFldNames{5}:="[Company]ZIP"
```

```
aFldNames{6}:="[Invoices]Amount"
```

```
$ErrCode:= OP Get all field numbers (vConnectID;aFldNames;aTables;aFields)
```

## 参照

Field、OP Count fields、OP Get field properties、OP Get one field number

## OP Get one tablename

---

**OP Get one tablename** (connectionID; tableID; nameOfTable) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	対象とするテーブルの番号
nameOfTable	固定長文字列	対象とするテーブルの名前
Error	倍長整数	エラーコード

### 説明

**OP Get one tablename** は、引数 < nameOfTable > に引数 < tableID > で指定したテーブルの名前を返します。

### エラーコード

**OP Get one tablename** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続IDに対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 参照

OP Get all tablenamees、Table name

## OP Count fields

---

**OP Count fields** (connectionID; tableID; nbOfFields) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	対象とするテーブルの番号
nbOfFields	倍長整数	対象とするテーブルのフィールド数
Error	倍長整数	エラーコード

### 説明

**OP Count fields** は引数 < nbOfFields > に引数 < tableID > で指定したテーブルのフィールド数を返します。

### エラーコード

**OP Count fields** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 参照

Count fields、OP Count tables



## 検索とソート、序論

---

データを処理する場合、操作したいレコードのグループを選択しておくことができます。その方法として、テーブル内の全レコードを選択する方法と、検索を行って特定のいくつかのレコードを選択する方法とがあります。

テーブルごとに常に1つずつのセレクションが存在します。またプロセスごとにも個別に存在します。カレントセレクションはもっとも最近選択されたレコードの集まりです。カレントセレクションを保存するためには後の章で取り上げる命名セレクションが使用できます。

検索については単一条件でも複数条件でも実行できます。

### 複数条件検索

2つ以上のフィールドを対象に検索を行うことを「複合検索」「複数条件検索」と呼ぶことがあります。この種の検索を行うには、複数の検索条件式それぞれに論理演算子を付加します。

論理演算子は、個々の検索条件での検索結果をどのように組み合わせるかを 4th Dimension に伝えます。論理演算子を使用すれば “ ニューヨークがカリフォルニアに勤務する従業員を探せ ” のような複合検索条件を作ることができます。

検索範囲については、テーブルのレコード全体か、カレントセレクションだけか、どちらにするかを選ぶことができます。

検索、ソートルーチンには以下のものがあります：

**OP Single query**：全レコードを対象に単一条件の検索を実行します。

**OP Single query selection**：カレントセレクション内のレコードを対象に単一条件の検索を行います。

**OP Single order by**：単一のフィールドをソートキーにしてカレントセレクションのレコードをソートします。

**OP Multi query**：全レコードを対象により複雑な検索を実行します。

**OP Multi query selection**：カレントセレクション内のレコードを対象により複雑な検索を実行します。

**OP Multi order by**：複数のフィールドをソートキーにしてカレントセレクションのレコードをソートします。

#### 参照

OP Multi order by、OP Multi query、OP Multi query selection、OP Single order by、OP Single query、OP Single query selection

## OP Single query

**OP Single query** (connectionID; tableID; fieldID; queryOperator; queryValue; recordsFound) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	検索条件とするテーブルの番号
fieldID	倍長整数	検索条件とするフィールドの番号
queryOperator	固定長文字列	比較演算子
queryValue	ポインタ	比較値を保持したフィールド、変数へのポインタ
recordsFound	倍長整数	検索にヒットしたレコード数
Error	倍長整数	エラーコード

### 説明

**OP Single query** ルーチンは、引数 < tableID > で指定されたテーブルのレコードを、引数 < queryValue > の指す値を基準にして検索します。対象となるレコードはテーブル内の全レコードです。検索実行後、引数 < recordsFound > には検索によって生じた新しいセレクションのレコード数が返されます。

引数 < queryValue > は検索したい値を保持しているフィールド、変数（プロセス/インタースタート変数）へのポインタを渡します。この値には “@” マークを含めることができるので、前方一致、後方一致、含む検索が可能です。引数 < queryValue > の指す値は、必要に応じて自動的に値の型が変換されます。例えば、ブールフィールドの値を検索対象にする場合、False を探したければ “false” を、True を探したければ “true” を指定しなければなりません。

比較演算子には以下のものがあります：

演算子	説明
"="	イコール
"#"	異なる
">"	超える
">="	以上
"<"	未満
"<="	以下

## エラーコード

**OP Single query**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9969	不正なフィールド型が要求されました。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10144	ポインタの参照先がありません (NIL ポインタ)。
10145	ポインタを指定すべき個所がポインタではありません。
10151	検索条件式に不正な演算子が使われています。
10153	型変換の不可能な引数データがあります。
10154	このコマンドを今すぐには実行できません。

## 例題

この例題では、[納品書]テーブルから総合計が10,000を超えるレコードを検索しています。

```
C_LONGINT (vRecords; $ErrCode)
C_REAL (vValue)
C_LONGINT (vTable;vField)

vValue:=10000
$ErrCode:= OP Get one field number (vConnectID;"[納品書]総合計";vTable;vField)
$ErrCode:= OP Single query (vConnectID;vTable;vField;">";->vValue;vRecords)

If ($ErrCode=0)
    ALERT (String (vRecords)+"件見つかりました.")
End if
```

## 参照

OP Get one field number、OP Multi query、OP Single order by、ORDER BY

## OP Single query selection

**OP Single query selection** (connectionID; tableID; fieldID; queryOperator; queryValue; recordsFound) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続 ID
tableID	倍長整数	検索条件とするテーブルの番号
fieldID	倍長整数	検索条件とするフィールドの番号
queryOperator	固定長文字列	比較演算子
queryValue	ポインタ	比較値を保持したフィールド、変数へのポインタ
recordsFound	倍長整数	検索にヒットしたレコード数
Error	倍長整数	エラーコード

### 説明

**OP Single query selection** は、引数 < tableID > で指定されたテーブルのレコードを、引数 < queryValue > の指す値を基準にして検索します。対象となるレコードはカレントセクション内のレコードのみです。検索実行後、引数 < recordsFound > には検索によって生じた新しいセクションのレコード数が返されます。

引数 < queryValue > は検索したい値を保持しているフィールド、変数（プロセス/インタープロセス変数）へのポインタを渡します。この値には “ @ ” マークを含めることができるので、前方一致、後方一致、含む検索が可能です。引数 < queryValue > の指す値は、必要に応じて自動的に値の型が変換されます。例えば、ブールフィールドの値を検索対象にする場合、False を探したければ “ false ” を、True を探したければ “ true ” を指定しなければなりません。

比較演算子には以下のものがあります：

演算子	説明
"="	イコール
"#"	異なる
">"	超える
">="	以上
"<"	未満
"<="	以下

## エラーコード

**OP Single query selection**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9969	不正なフィールド型が要求されました。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10144	ポインタの参照先がありません (NIL ポインタ)。
10145	ポインタを指定すべき個所がポインタではありません。
10151	検索条件式に不正な演算子が使われています。
10153	型変換の不可能な引数データがあります。
10154	このコマンドを今すぐには実行できません。

## 例題

この例題では、[納品書]テーブルから総合計が10,000を超えるレコードを検索しています。その後で、得られたレコードの中から今年の1月1日以降のレコードだけに絞り込んでいきます。

```
C_LONGINT (vRecords; $ErrCode)
C_REAL (vValue)
C_LONGINT (vTable;vField1;vField2)

$errCode:= OP Get one field number (vConnectID;"[Invoices]Total";vTable;vField1)
$errCode:= OP Get one field number (vConnectID;"[Invoices]Invoice date";vTable;
                                         vField2)

vValue:="10000"
$errCode:= OP Single query (vConnectID;vTable;vField1;">";->vValue;vRecords)
vValue:= String ( year of ( Current date );"#####") + "/01/01"
$errCode:= OP Single query selection (vCon3nectID;vTable;vField2;">";-
                                         >vValue;vRecords)

If ($ErrCode=0)
    ALERT (String(vRecords)+" record(s) were found.")
End if
```

## 参照

OP Get one field number、OP Multi query、OP Single order by、OP Single query、  
QUERY SELECTION

## OP Single order by

**OP Single order by** (connectionID; tableID; fieldID; orderDirection) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	検索条件とするテーブルの番号
fieldID	倍長整数	検索条件とするフィールドの番号
orderDirection	固定長文字列	ソート方向(“>”が昇順)
Error	倍長整数	エラーコード

### 説明

**OP Single order by** は、引数 < tableID > で指定されたテーブルのカレントセレクションを引数 < fieldID > で指定されたフィールドをキーにしてソートします。

もちろんソート方向も引数 < orderDirection > に記号 “>”、“<” を渡すことで指定できます。それぞれ、“>” は昇順、“<” は降順を表します。

インデックス付きフィールドをソートキーに指定した場合には、4D Server はインデックス情報を使用してソートを行います。

### エラーコード

**OP Single order by** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9969	不正なフィールド型が要求されました。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。
10155	不正なソート方向指示文字列です。

## 例題

この例題では納品書テーブルのカレントセレクションを日付フィールドをキーにして昇順でソートします：

```
C_LONGINT (vTable;vField)
```

```
C_LONGINT ($errCode)
```

```
$errCode:= OP Get one field number (vDconnectID;"[Invoices]Invoice date";vTable;  
vField)
```

```
$errCode:= OP Single order by (ConnectID;vTable;->Field;">")
```

## 参照

OP Get one field number、OP Multi order by、OP Multi query、OP Single query、ORDER BY



## OP Multi query

**OP Multi query** (connectionID; tableID; tableIDs; fieldIDs; logicalOperators; queryOperators; queryValues; recordsFound) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	検索を行いセクションを生成するテーブルの番号
tableIDs	配列	検索条件ごとの検索対象フィールドの属するテーブル番号を保持した配列
fieldIDs	配列	検索条件ごとの検索対象フィールドの番号を保持した配列
logicalOperators	配列	各検索条件間の組み合わせを指示する論理演算子を保持した配列
queryOperators	配列	検索条件ごとの比較演算子を保持した配列
queryValues	配列	検索条件ごとの検索比較値を保持したフィールド、変数へのポインターを保持した配列
recordsFound	倍長整数	検索にヒットしたレコード数
Error	倍長整数	エラーコード

### 説明

**OP Multi query** は、引数 < tableID > で指示されたテーブルのレコードを検索します。検索フィールドは引数 < fieldIDs > に渡された配列の各要素に収められたフィールド番号で示され、それらのフィールドは複数のテーブルのものでも構いません。検索はテーブルの全レコードを対象に実行されます。引数 < tableIDs >、< fieldIDs >、< logicalOperators >、< queryOperators >、< queryValues > に渡す配列は最大20まで要素を含むことができます。したがって、検索条件数の最大数は20です。要素数が20を超えていても20番目までの要素しか使われません。また、その際エラーも出ません。

引数 < tableIDs > と < fieldIDs > は整数か倍長整数の配列で、検索条件に使用するフィールドとテーブルの番号を入れておきます。

引数 < logicalOperator > と < queryOperator > は整数、倍長整数、実数、固定長文字列、テキスト型いずれかの配列で、各検索条件ごとの論理演算子と比較演算子を渡します。

論理演算子の値として使用可能なものを以下に列挙します。使用する配列の型によって使用する値が異なります：

説明	文字配列	数値配列
And	"&"	1
Or	" "	2
Except	"#"	0

論理演算子は、各検索条件ごとに得られる検索結果をどのように組み合わせるかを指定する演算子です。組み合わせは1つ前の条件で得られた結果との間で行われます。したがって、最初の検索条件（引数 < tableIDs >、< fieldIDs >、< logicalOperators >、< queryOperators >、< queryValues > の各配列 1 番目の要素の値をもとに構成される検索条件）には論理演算子は必要ないため、指定しても無視されます。

比較演算子に使える値は以下の通りです：

説明	文字	数値
等号	"="	1
不等号	"#"	2
超える	">"	3
以上	">="	4
未満	"<"	5
以下	"<="	6

引数 < queryValue > は、検索したい値を保持する固定長文字列かテキスト型の配列です。この値には “ @ ” マークを含めることができるので、前方一致、後方一致、含む検索が可能です。

値は文字として渡す必要があります。例えば、日付「1999年12月31日」を渡す場合、“1999/12/31” とします。また、ブールフィールドの値を検索対象にする場合、False を探したければ “ false ” を、True を探したければ “ true ” を指定しなければなりません。

## エラーコード

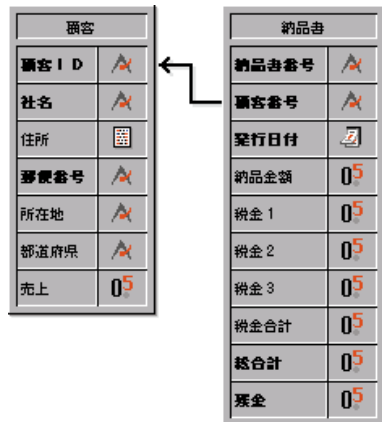
**OP Multi query** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9969	不正なフィールド型が要求されました。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10135	引数型が不正です。
10136	指定された接続 ID に対応する接続がありません。
10151	検索条件式に不正な演算子が使われています。
10152	不正な論理演算子です。
10153	型変換の不可能な引数データがあります。
10154	このコマンドを今すぐには実行できません。
10156	空の検索条件もしくはソート条件がありました。

注：検索条件として渡した配列要素にエラーがあると、**OP Multi query**はエラーの発生した行（要素番号）を配列の選択行とします。例えば、検索条件配列の5番目の要素にエラーがあった場合、すべての検索条件配列の選択行は5になっています。スクロールエリアで表示した場合はその行が反転表示され、ポップアップメニューであれば、その行が表示されているはずですが。

## 例題

この例題では東京と大阪の顧客で、総合計が2,000を超える納品書を検索しています。例題で使用するストラクチャは以下の通りです。



```
C_LONGINT ($ErrCode;$TrgTable;vRecords)
ARRAY LONGINT (arTableID;3)
ARRAY LONGINT (arFieldID;3)
ARRAY TEXT (arLogicalOp;3)
ARRAY TEXT (arQueryOp;3)
ARRAY TEXT (arValues;3)
```

```
$TrgTable:=2           `検索してセレクションを作るテーブルは[納品書]
arTableID{1}:=1        `[顧客]
arTableID{2}:=1        `[顧客]
arTableID{3}:=2        `[納品書]

arFieldID{1}:=6        `[顧客]所在地
arFieldID{2}:=6        `[顧客]所在地
arFieldID{3}:=4        `[納品書]総合計

arQueryOp{1}:= "="     `[顧客]所在地 =
arQueryOp{2}:= "="     `[顧客]所在地 =
arQueryOp{3}:= ">"     `[納品書]総合計 >

arValues{1}:= "東京"   `[顧客]所在地 = "東京"
arValues{2}:= "大阪"   `[顧客]所在地 = "大阪"
arValues{3}:= "2000"   `[納品書]総合計 > 2000
```

```
arLogicalOp{1}:=""      `[顧客]所在地 = "東京"  
arLogicalOp{2}:="|"    `[または[顧客]所在地 = "大阪"  
arLogicalOp{3}:="&"   `[かつ[納品書]総合計 > 2000
```

```
$ErrCode:=OP Multi query (vConnectID;$TrgFile;arFileID;arFieldID;arLogicalOp;  
                           arQueryOp;arValues;vRecords)
```

```
If ($ErrCode=0)
```

```
    ALERT(String(vRecords)+"件見つかりました。")
```

```
End if
```

#### 参照

OP Multi query selection、OP Single order by、OP Single query、QUERY

## OP Multi query selection

**OP Multi query selection** (connectionID; tableID; tableIDs; fieldIDs; logicalOperators; queryOperators; queryValues; recordsFound) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	検索を行いセレクションを生成するテーブルの番号
tableIDs	配列	検索条件ごとの検索対象フィールドの属するテーブル番号を保持した配列
fieldIDs	配列	検索条件ごとの検索対象フィールドの番号を保持した配列
logicalOperators	配列	各検索条件間の組み合わせを指示する論理演算子を保持した配列
queryOperators	配列	検索条件ごとの比較演算子を保持した配列
queryValues	配列	検索条件ごとの検索比較値を保持したフィールド、変数へのポインタを保持した配列
recordsFound	倍長整数	検索にヒットしたレコード数
Error	倍長整数	エラーコード

### 説明

**OP Multi query selection** は、引数 < tableID > で指示されたテーブルのレコードを検索します。検索フィールドは引数 < fieldIDs > に渡された配列の各要素に収められたフィールド番号で示され、それらのフィールドは複数のテーブルのものでも構いません。検索はカレントセレクション内のレコードを対象に実行されます。

引数 < tableIDs >、< fieldIDs >、< logicalOperators >、< queryOperators >、< queryValues > に渡す配列は最大20まで要素を含むことができます。したがって、検索条件数の最大数は20です。要素数が20を超えていても20番目までの要素しか使われません。また、その際エラーも出ません。

引数 < tableIDs > と < fieldIDs > は整数か倍長整数の配列で、検索条件に使用するフィールドとテーブルの番号を入れておきます。

引数 < logicalOperator > と < queryOperator > は整数、倍長整数、実数、固定長文字列、テキスト型いずれかの配列で、各検索条件ごとの論理演算子と比較演算子を渡します。

論理演算子の値として使用可能なものを以下に列挙します。使用する配列の型によって使用する値が異なります：

説明	文字配列	数値配列
And	"&"	1
Or	" "	2
Except	"#"	0

論理演算子は、各検索条件ごとに得られる検索結果をどのように組み合わせるかを指定する演算子です。組み合わせは1つ前の条件で得られた結果との間で行われます。したがって、最初の検索条件（引数 < tableIDs >、< fieldIDs >、< logicalOperators >、< queryOperators >、< queryValues > の各配列1番目の要素の値をもとに構成される検索条件）には論理演算子は必要ないため、指定しても無視されます。

比較演算子の値として使用可能なものを以下に列挙します。使用する配列の型によって使用する値が異なります：

説明	文字	数値
等号	"="	1
不等号	"#"	2
超える	">"	3
以上	">="	4
未満	"<"	5
以下	"<="	6

引数 < queryValue > は検索したい値を保持する固定長文字列かテキスト型の配列です。

この値には “ @ ” マークを含めることができるので、前方一致、後方一致、含む検索が可能です。値は文字として渡す必要があります。例えば、日付「1999年12月31日」を渡す場合、“ 1999/12/31 ” とします。また、ブールフィールドの値を検索対象にする場合、Falseを探したければ“ false ”を、Trueを探したければ“ true ”を指定しなければなりません。

## エラーコード

**OP Multi query** の実行に成功した場合、戻り値 < Error > は「0」です。そうでなければ、以下のエラーコードが返されます：

エラーコード	説明
-9969	不正なフィールド型が要求されました。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10135	引数型が不正です。
10136	指定された接続 ID に対応する接続がありません。
10151	検索条件式に不正な演算子が使われています。

10152	不正な論理演算子です。
10153	型変換の不可能な引数データがあります。
10154	このコマンドを今すぐには実行できません。
10156	空の検索条件もしくはソート条件がありました。

注：検索条件として渡した配列要素にエラーがあると、**OP Multi query** はエラーの発生した行（要素番号）を配列の選択行とします。

例えば、検索条件配列の5番目の要素にエラーがあった場合、すべての検索条件配列の選択行は5になっています。スクロールエリアで表示した場合はその行が反転表示され、ポップアップメニューであれば、その行が表示されているはずです。

### 例題

この例題では東京と大阪の顧客当りの納品書をユーザが前もって検索していたカレントセレクションのレコードの中から検索しています：

```
C_LONGINT ($ErrCode;$TrgTable;vRecords)
ARRAY LONGINT (arTableID;0)
ARRAY LONGINT (arFieldID;0)
ARRAY TEXT (arLogicalOp;0)
ARRAY TEXT (arQueryOp;0)
ARRAY TEXT (arValues;0)
```

*UserSelectInvoices*      `ダイアログを表示してユーザに検索条件を指定させる

```
`メソッド「UserSelectInvoices」で指定された検索条件で検索実行
$ErrCode:=OP Multi query (vConnectID;2;arFileID;arFieldID;arLogicalOp;
arQueryOp;arValues;vRecords)
```

```
ARRAY LONGINT (arTableID;3)
ARRAY LONGINT (arFieldID;3)
ARRAY TEXT (arLogicalOp;2)
ARRAY TEXT (arQueryOp;2)
ARRAY TEXT (arValues;2)
```

\$TrgTable:=2      `検索してセレクションを作るテーブルは[納品書]

```
arTableID{1}:=1      `[顧客]
arTableID{2}:=1      `[顧客]
arFieldID{1}:=6      `[顧客]所在値
arFieldID{2}:=6      `[顧客]所在値
arQueryOp{1}:= "="      `[顧客]所在値 =
arQueryOp{2}:= "="      `[顧客]所在値 =
arValues{1}:= "東京"      `[顧客]所在値 = "東京"
arValues{2}:= "大阪"      `[顧客]所在値 = "大阪"
arLogicalOp{1}:= ""      `[顧客]所在値 = "東京"
arLogicalOp{2}:= "|"      `または[顧客]所在値 = "大阪"
```

`前段階においてユーザ指定の検索条件で作成されたセレクションを対象に  
`新しい条件で絞り込み検索を行う

```
$ErrCode:= OP Multi query selection (vConnectID;$TrgFile;arFileID;arFieldID;  
arLogicalOp;arQueryOp;arValues;vRecords)
```

```
If ($ErrCode=0)  
  ALERT(String(vRecords)+"件発見しました。")  
End if
```

#### 参照

OP Multi query、OP Single order by、OP Single query、QUERY SELECTION



## OP Multi order by

**OP Multi order by** (connectionID; tableID; tableIDs; fieldIDs; orderDirections)

Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	検索条件とするテーブルの番号
tableIDs	配列	ソートキーのフィールドの属するテーブル番号を保持した配列
fieldIDs	配列	ソートキーのフィールド番号を保持した配列
orderDirections	配列	ソート方向(">"が昇順)文字を保持した配列
Error	倍長整数	エラーコード

説明

**OP Multi order by** は、引数 < tableID > で指定されたテーブルのカレントセレクションを対象にレコードをソートします。ソート条件は、引数 < tableIDs >、< fieldIDs >、< orderDirections > の各配列で指定できます。

配列は最大20まで要素を含むことができます。したがって、ソート条件数の最大数は20です。要素数が20を超えていても20番目までの要素しか使われません。また、その際エラーも出ません。

引数 < tableIDs > と < fieldIDs > は整数、倍長整数、実数型のいずれかの配列で、検索条件に使用するフィールドとテーブルの番号を入れておきます。

引数 < orderDirections > は整数、倍長整数、実数、固定長文字列、テキスト、ブール型いずれかの配列で、ソート方向を示す値を渡します。

文字で値を渡す場合は、">" (昇順) か "<" (降順) を使用します。

数値で値を渡す場合は、正の値 (昇順)、ゼロまたは負の値 (降順) を渡します。

ブール値で値を渡す場合は、"true" (昇順)、"false" (降順) の値を渡します。

引数チェック機能がONであり、文字でソート方向を指示しようとしていた場合に、">"、"<"以外の値を渡すとエラーと判断されます。

引数チェック機能がOFFであり、文字でソート方向を指示しようとしていた場合に、"<"以外の文字で始まる文字列を渡すと昇順と判断されます。

このルーチンに渡すソートキーフィールドがすべてインデックス付けされていても、4D Server はシーケンシャルソートを実行します。

## エラーコード

**OP Multi order by**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10135	引数型が不正です。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。
10155	不正なソート方向指示文字列です。
10156	空の検索条件もしくはソート条件がありました。

注：ソート条件として渡した配列要素にエラーがある場合、**OP Multi order by**はエラーの発生した行（要素番号）を配列の選択行とします。例えば、ソート条件配列の5番目の要素にエラーがあった場合、すべてのソート条件配列の選択行は5になっています。スクロールエリアで表示するとその行が反転表示され、ポップアップメニューであれば、その行が表示されているはずですが。

## 例題

この例題では納品書テーブルのカレントセクションを社名、発行日付でソートします。使用するストラクチャは**OP Multi query**と同様です。

```
C_LONGINT ($ErrCode;$TrgFile)
ARRAY LONGINT (aTableID;2)
ARRAY LONGINT (aFieldID;2)
ARRAY STRING (1;aOrder;2)
$Trgtable:=2      `ソート対象テーブルは[納品書]
aTableID{1}:=1    `[顧客]
aTableID{2}:=1    `[納品書]
aFieldID{2}:=2    `[顧客]社名
aFieldID{3}:=3    `[納品書]発行日付
aOrder{1}:=">"
aOrder{2}:=">"
$ErrCode:= OP Multi order by (vConnectID;$TrgFile;aFileID;aFieldID;aOrder)
```

## 参照

OP Get one field number、OP Multi query、OP Order by、OP Single query、ORDER BY

## セクション、序論

---

この章で取り上げるルーチン群は、セクションを操作する機能があります：

**OP Records in table**：指定したテーブルに含まれる全レコード数を返します。

**OP Records in selection**：指定したテーブルのカレントセクションに含まれる全レコード数を返します。

**OP All records**：指定したテーブルに含まれる全レコードをカレントセクションにします。

**OP Reduce selection**：指定した数にセクションのレコード数を減らします。

**OP Delete selection**：カレントセクションのレコードを削除します。

**OP Many to one join**：nテーブル側にセクションを作ります。

**OP One to many join**：1（ワン）テーブル側にセクションを作ります。

**OP Scan index**：インデックス情報を元にカレントセクションの最初のn件もしくは最後のn件のレコードを新しいセクションとして選択します。

### 参照

OP All records、OP Delete selection、OP Many to one join、OP One to many join、OP Records in

## OP Records in table

---

### OP Records in table (connectionID; tableID; recordsInTable) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	対象とするテーブル番号
recordsInTable	倍長整数	対象とするテーブルのレコード数
Error	倍長整数	エラーコード

#### 説明

**OP Records in table** は、引数 < tableID > で指定したテーブルの全レコード数を引数 < recordsInTable > に渡した変数に返します。

#### エラーコード

**OP Records in table** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

#### 例題

```
C_LONGINT (vTableID;vFieldID)
C_LONGINT (vRecords)
C_LONGINT ($errCode;$nbHours)
  `テーブル番号を得る
  $errCode:= OP Get one field number (vConnectID;"[Invoices]Amount";vTableID;
                                          vFieldID)
  `全レコード数を得る
  $errCode:= OP Records in table (vConnectID;vTableID;vRecords)
  `平均処理時間をもとに所要時間を予測する。
  $nbHours:=vRecords*<>avgTimePerRecord
  If ($nbHours > 8) `もし8時間を超えそうであればユーザの許可を請う。
    CONFIRM ("この処理は8時間以上掛かりそうです。
              "+Char (13)+"続行しますか?")
  If (OK=1)
    processInvoices
  End if
End if
```

#### 参照

OP Get one field number、OP Records in selection、Records in table

## OP Records in selection

**OP Records in selection** (connectionID; tableID; recordsInTable) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	対象とするテーブル番号
recordsInSelection	倍長整数	対象とするテーブルのカレントセレクションに含まれるレコード数
Error	倍長整数	エラーコード

### 説明

**OP Records in selection** は、引数 < tableID > で指定したテーブルのカレントセレクションに含まれるレコード数を引数 < recordsInSelection > に渡した変数に返します。

### エラーコード

**OP Records in table** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 例題

```
C_LONGINT (vTableID;vFieldID)
C_LONGINT (vRecords)
C_LONGINT ($errCode;$nbHours)
```

```
UserSelectInvoices    `ダイアログを表示してユーザに検索条件を指定させる。
`メソッド"UserSelectInvoices"で指定された検索条件で検索実行。
UserQueryRun          `セレクションに含まれるレコード数を得る。
```

```
$errCode:= OP Records in selection (vConnectID;2;vRecords)
```

```
`処理所要時間を予測する
```

```
$nbHours:=vRecords*`<>avgTimePerRecord
```

```
If ($nbHours > 8)    `もし8時間を超えそうであればユーザの許可を請う。
```

```
    CONFIRM ("この処理は8時間以上掛かりそうです。
```

```
                "+Char (13)+"続行しますか?")
```

```
    If (OK=1)
```

```
        processInvoices
```

```
    End if
```

```
End if
```

## 参照

OP All records、OP Multi query、OP Records in table、OP Single query、OP Use named selection、Records in selection

## OP All records

### OP All records (connectionID; tableID) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	対象とするテーブル番号
Error	倍長整数	エラーコード

### 説明

**OP All records** は、引数 < tableID > で指定したテーブルの全レコードをカレントセレクションにします。

### エラーコード

**OP All records** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 例題

**C\_LONGINT** (\$errCode;\$nbHours)

*UserSelectInvoices*     `ダイアログを表示してユーザに検索条件を指定させる。

#### Case of

¥(userClickedQuery= 1)     `「検索」ボタンをクリックした。  
 UserQueryRun     `メソッド"UserSelectInvoices"で指定された検索条件で  
 検索実行。

¥(userClickedCancel=1)     `検索はキャンセルされた。  
 \$errCode:= **OP Reduce selection** (vConnectID;vTableID;0)

¥(userClickedAllInvoices=1)     `すべての納品書を選択した。  
 \$errCode:= **OP All records** (vConnectID;vTableID)

#### End case

`カレントセレクションに含まれる全レコード数をカウント。  
 \$errCode:= **OP Records in selection** (vConnectID;vTableID;vRecords)

`処理所要時間を予測する。  
 \$nbHours:=vRecords\*<>avgTimePerRecord

**If** (\$nbHours > 8)     `もし 8 時間を超えそうであればユーザの許可を請う。

**CONFIRM** ("この処理は 8 時間以上掛かりそうです。  
"+Char (13)+"続行しますか？")

**If** (OK=1)  
    *processInvoices*

**End if**

**End if**

#### 参照

ALL RECORDS、 OP Records in selection、 OP Records in table、 OP Reduce selection、  
OP Single query



## OP Reduce selection

**OP Reduce selection** (connectionID; tableID; selectionSize) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	対象とするテーブル番号
selectionSize	倍長整数	カレントセレクションにするレコード数
Error	倍長整数	エラーコード

### 説明

**OP Reduce selection** は、引数 < tableID > で指定したテーブルに新しいカレントセレクションを作ります。新しいセレクションは、現在のセレクションの最初の n 件と等しくなり、件数は引数 < selectionSize > で指定された数とされます。

引数 < selectionSize > がカレントセレクションのレコード数を超えていた場合は、セレクションは変わりません。

### エラーコード

**OP Reduce selection** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 例題

この例題では、昨年の金額トップ 10 の納品書の平均金額を求めています。

**C\_LONGINT** (vRecords; \$ErrCode)

**C\_REAL** (vValue; vMeanAmount)

**C\_LONGINT** (vTable; vFieldDate; vFieldAmount)

\$errCode:= **OP Get one field number** (vConnectID;"[納品書]発行日付";vTable;  
vFieldDate)

\$errCode:= **OP Get one field number** (vConnectID;"[納品書]合計金額";vTable;  
vFieldAmount)

` 昨年の 1 月 1 日を求める。

vValue:=String (year of ( Current date ) -1;"####")+"/01/01/"

` 昨年の納品書を検索する。  
\$errCode:= **OP Single query** (vConnectID;vTable;vFieldDate;">">->vValue;vRecords)

　　` 合計金額をキーに降順でソート(最高金額が最初の行に表れるように)  
\$errCode:= **OP Single order by** (vConnectID;vTable;vFieldAmount;"<")

　　` 金額の高い方から 10 件を選ぶ。  
\$errCode:= **OP Reduce selection** (vConnectID;vTable;10)

　　` 平均値を求める。  
\$errCode:= **OP Sum** (vConnectID;vTable;vFieldAmount;vMeanAmount)

**ALERT** ("昨年の上位 10 件の納品書合計金額の平均値 : "+ **String** (vMeanAmount;  
"### ##.##"))

#### 参照

OP All records、OP Single order by、OP Records in selection、OP Single query、  
REDUCE SELECTION

## OP Delete selection

**OP Delete selection** (connectionID; tableID) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	対象とするテーブル番号
Error	倍長整数	エラーコード

### 説明

**OP Delete selection** は、引数 < tableID > で指定したテーブルのカレントセレクションに含まれるレコードをすべて削除します。カレントセレクションが空である場合は、このルーチンは何もしません。

### エラーコード

**OP Delete selection** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 例題

この例題では、3年以上前の納品書を選択し、一括して別ファイルに保存し、削除しています。

```

C_LONGINT (vRecords; $ErrCode)
C_STRING (50;vValue)
C_LONGINT (vTable;vFieldDate)

$errCode:= OP Get one field number (vConnectID;"[納品書]発行日付";vTable;
                                         vFieldDate)

`3年前の12月31日を計算する。
vValue:=String ( Year of ( Current date ) -3 ;"####") + "/12/31/"
`3年以上前の納品書を検索する。
$errCode:= OP Single query (vConnectID;vTable;vFieldDate;"<=";->vValue;
                                         vRecords)

ArchiveInvoices
`もう必要ないのでこれら古い納品書を削除する。
$errCode:= OP Delete selection (vConnectId;vTable)
    
```

### 参照

DELETE SELECTION、OP All records、OP Records in selection、OP Single query

## OP Many to one join

---

**OP Many to one join** (connectionID; manyTableID; oneTableID) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
manyTableID	倍長整数	nテーブルのID
oneTableID	倍長整数	1テーブルのID
Error	倍長整数	エラーコード

### 説明

**OP Many to one join** は、引数 < oneTableID > で指定したテーブルに、引数 < manyTableID > で指定した n テーブルのセレクションのレコードの値を元にしてカレントセレクションを作ります。例えば、n ファイルにセレクションがあり、その n ファイルのセレクションの各レコードにリレートした 1 テーブルのレコードを作ることができます。[ 顧客 ] / [ 納品書 ] ストラクチャで例を挙げれば、[ 納品書 ] のセレクションに対して **OP Many to one Join** を実行して、[ 顧客 ] テーブルにそれらの納品書の宛先である顧客レコードを選択する場合などに使用されます。

このコマンドは複数レベルのリレーションを横断して動作します。また、リレーションは、自動、マニュアル共に動作します。

### エラーコード

**OP Many to one join** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

## 例題

以下のコードでは、今年の金額上位10件の納品書の宛先顧客の一覧を印刷しています：

```

C_LONGINT (vRecords; $ErrCode)
C_STRING (10;vValue)
C_LONGINT (vTable 納品書;vTable 顧客;vField 発行日;vField 合計;vField 顧客番号)

$errCode:= OP Get one field number (vConnectID;"[納品書]発行日";vTable 納品書;
                                     vField 発行日)
$errCode:= OP Get one field number (vConnectID;"[納品書]合計金額";
                                     vTable 納品書;vField 合計)
$errCode:= OP Get one field number (vConnectID;"[顧客]番号";vTable 顧客;
                                     vField 顧客番号)

`今年の1月1日を計算。
vValue:=String ( year of ( Current date ) ;"####") + "/01/01/"

`今年の納品書を検索。
$errCode:= OP Single query (vConnectID;vTable 納品書;vField 発行日;">";->vValue;
                             vRecords)

`金額をキーに降順でソート(最高金額が最初の行に表れるように)。
$errCode:= OP Single order by (vConnectID;vTable 納品書;vField 合計;"<")

`金額の高い方から10件を選ぶ。
$errCode:= OP Reduce selection (vConnectID;vTable 納品書;10)

`見つかった納品書に関連する顧客のセレクションを作る。
$errCode:= OP Many to one join (vConnectID;vTable 納品書;vTable 顧客)

`該当する顧客の数を調べる。
$errCode:= OP Records in selection (vConnectID;vTable 顧客;vRecords)
ALERT ("今年の納品書金額上位10件の宛先顧客数：" + String (vRecords) + "社")

Print 顧客

```

## 参照

OP One to many join、OP Records in selection、OP Relate one、RELATE ONE SELECTION

## OP One to many join

---

**OP One to many join** (connectionID; manyTableID; manyFieldID) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
manyTableID	倍長整数	nテーブルのID
manyFieldID	倍長整数	nテーブルのリレートされたフィールドID
Error	倍長整数	エラーコード

### 説明

**OP One to many join** は、引数 < manyFieldID > で指定したフィールドとリレートされている1テーブルのセレクションを元に、引数 < manyTableID > で指定したテーブルに新しいカレントセレクションを作ります。

**OP Many to one join** の例で取り上げた [ 顧客 ] / [ 納品書 ] ストラクチャを再び例にとれば、[ 顧客 ] テーブルのカレントセレクションを元に、その顧客へ宛てられた納品書だけを選択するのに **OP Many to one join** を使います。引数で指示するフィールドはnテーブルである [ 納品書 ] で、フィールドは1テーブルとリレートしている必要があるため、この場合 [ 納品書 ] 顧客IDフィールドを指定します。

**OP Many to one join** は自動リレーション、マニュアルリレーション共に動作します。

### エラーコード

**OP Many to one join** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9951	フィールドがリレートしていません。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続IDに対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

## 例題

以下のコードでは、昨年の金額上位10件の納品書の宛先顧客へ今年1年に送ったすべての納品書一覧を得ようとしています。つまり昨年の金額上位10件の納品書を受け取った顧客の今年度の実績を調べているのです：

```

C_LONGINT (vRecords; $ErrCode)
C_STRING (10;vValue1;vValue2;vValue3)
C_LONGINT (vTable 納品書;vTable 顧客;$unused1)
C_LONGINT (vFieldDate;vFieldAmount;vFieldCustID;$unused2)
    ` [納品書]のテーブル番号と[納品書]発行日のフィールド番号を得る
    $errCode:= OP Get one field number (vConnectID;"[納品書]発行日";vTable 納品書;
                                                vFieldDate)

    ` [納品書]合計金額のフィールド番号を得る。
    $errCode:= OP Get one field number (vConnectID;"[納品書]合計金額";$unused1;
                                                vFieldAmount)

    ` [納品書]顧客IDのフィールド番号を得る。
    $errCode:= OP Get one field number (vConnectID;"[納品書]顧客ID";$unused1;
                                                vFieldCustID)

    ` [顧客]のテーブル番号を得る。
    $errCode:= OP Get one field number (vConnectID;"[顧客]Ref";vTable 顧客;
                                                $unused2)

    ` 昨年の1月1日を求める。
    vValue1:=String ( year of ( Current date )-1 ;"####") + "/01/01/"
    ` 昨年の12月31日を求める。
    vValue2:= String ( year of ( Current date )-1 ;"####") + "/12/31/"
    ` 今年の1月1日を求める。
    vValue3:= String ( year of ( Current date ) ;"####") + "/01/01/"
    ` 去年の1月1日以降の納品書を探す。
    $errCode:= OP Single query (vConnectID;vTable 納品書;vFieldDate;">=";->vValue1;
                                                vRecords)

    ` 上の行で行った検索の結果を去年の12月31日までの納品書のみ絞り込む。
    $errCode:= OP Single query selection (vConnectID;vTable 納品書;vFieldDate;"<=";
                                                ->vValue2;vRecords)

    ` 合計金額をキーに降順でソート(最高金額が最初の行になるように)。
    $errCode:= OP Single order by (vConnectID;vTable 納品書;vFieldAmount;"<")
    ` 金額の高い方から10件を選ぶ。
    $errCode:= OP Reduce selection (vConnectID;vTable 納品書;10)
    ` 見つかった納品書の宛先顧客のセレクションを [顧客] テーブルに作る。
    $errCode:= OP Many to one join (vConnectID;vTable 納品書;vTable 顧客)
    ` 上の行で作成された [顧客] テーブルのセレクションを元に
    ` [納品書] テーブルにセレクションを作る。
    $errCode:= OP One to many join (vConnectID;vTable 納品書;vFieldCustID)
    ` でき上がった [納品書] テーブルのセレクションから、
    ` 更に発行日付が今年の1月1日以降のレコードのみに絞り込んでいる。
    $errCode:= OP Single query selection (vConnectID;vTable 納品書;vFieldDate;">=";
                                                ->vValue3;vRecords)
    
```

Print 納品書

## 参照

OP Many to one join、OP Single query、RELATE MANY SELECTION

## OP Scan index

---

**OP Scan index** (connectionID; tableID; fieldID; selectionSize; scanOrigin)

Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	対象とするテーブルのID
fieldID	倍長整数	対象とするフィールドのID
selectionSize	倍長整数	新しいカレントセレクションのサイズ
scanOrigin	倍長整数	スキャン開始位置(0ならインデックスの最初)
Error	倍長整数	エラーコード

説明

**OP Scan index** は、引数 < tableID > で指定したテーブルに引数 < selectionSize > 件分のセレクションを作成します。引数 < scanOrigin > が負の値の場合、**OP Scan index** ルーチンはインデックスの最後から引数 < selectionSize > 件分のレコードをセレクションにします。引数 < scanOrigin > が0以上の場合、**OP Scan index** ルーチンはインデックスの先頭から引数 < selectionSize > 件分のレコードをセレクションにします。

**OP Scan index** ルーチンは、インデックスフィールドにのみ使用できます。

テーブル内のレコード数より多くのレコードを指定した場合には、**OP Scan index** ルーチンはすべてのレコードをセレクションにします。

エラーコード

**OP Scan index** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9970	フィールドにインデックスが付いていません。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。



## 例題

以下のコードでは、金額上位100件の納品書の宛先顧客へ今年1年に送ったすべての納品書一覧を得ようとしています。OP Reduce selection との違いに注意してみてください。

```
C_LONGINT (vRecords; $ErrCode)
C_REAL (vValue;vMeanAmount)
C_LONGINT (vTable;vField 日付;vField 金額;vFromEnd)

$errCode:= OP Get one field number (vConnectID;"[Invoices]Invoice date";
                                         vTable;vField 日付)
$errCode:= OP Get one field number (vConnectID;"[Invoices]Invoice amount";
                                         vTable;vField 金額)

` スキャンはインデックスの最後から開始する。
vFromEnd:= -1
` 金額上位 100 件の納品書を選択する。
$errCode:= OP Scan index (vConnectID;vTable;vField 金額;100;vFromEnd)
` 今年の1月1日を求める。
vValue:= String ( year of ( Current date ) ;"####" ) + "/01/01"
` 今年の納品書だけに絞り込む。
$errCode:= OP Single query selection (vConnectID;vTable;vField 日付;">";
                                         ->vValue;vRecords)

PrintInvoices
```

## 参照

OP All records、OP Order by、OP Reduce selection、OP Single query、SCAN INDEX



## 配列、序論

---

この章で取り上げるルーチン群は、配列とレコードデータをやり取りする機能を持つものです。

**OP Selection to array** : カレントセレクションのレコードデータを配列へロードします。

**OP Distinct values** : フィールドの値からダブりを削除した結果を配列へ返します (インデックス付き固定長文字列フィールドのみ)

**OP Array to selection** : 配列の値をカレントセレクションのレコードへ代入します。

**OP Subselection to array** : カレントセレクションの一部のレコードの値を配列へコピーします。

### 参照

OP Array to selection、OP Distinct values、OP Selection to array、OP Subselection to array

## OP Selection to array

---

### OP Selection to array (connectionID; bindID{; tableID}) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
bindID	倍長整数	バインドID
tableID	倍長整数	ターゲットテーブルのID (セレクションを利用するテーブル)
Error	倍長整数	エラーコード

### 説明

**OP Selection to array** ルーチンは、フィールドの値をローカルの (クライアント側の) 配列へコピーします。使用されるフィールドと配列は、引数 < bindID > で指定されたバインドで定義済みのものを使用します。バインドは **OP Define bind by pointer** や **OP Define bind by numbers** を使用して定義します。

**OP Selection to array** は、単一のテーブルのに属するフィールドからデータを取得するだけでなく、リレート先のテーブルのフィールドからも値をコピーできます。

- ・ ターゲットテーブルに指定できるのはnテーブルです。
- ・ 値を取得できるリレート先テーブルは1テーブルです。
- ・ 自動リレートでなければなりません。
- ・ 必要なすべてのフィールドが1つのバインドへまとめられている必要があります。

引数 < tableID > は、ターゲットテーブルのIDを渡します。この引数を省略するか0を渡すと、自動的にバインドの最初のフィールドが属するテーブルをターゲットテーブルとして扱います。したがって、そのテーブルのカレントセレクションを元にリレーションが実行されます。

以下のストラクチャを使用して説明します：

顧客		納品書	
顧客ID	✖	納品書ID	✖
社名	✖	顧客番号	✖
住所	📄	発行日付	📅
郵便番号	✖	合計金額	05
住所都市	✖	税金1	05
都道府県	✖	税金2	05
売上	05	税金3	05
		税金合計	05
		総合計	05
		残金	05

例1：

[ 納品書 ] 納品書ID  
 [ 納品書 ] 発行日付  
 [ 納品書 ] 合計金額  
 [ 顧客 ] 社名  
 [ 顧客 ] 住所都市

以上のフィールドの値が欲しいものとします。

2つのテーブルに渡っているので、リレートを活用します。ターゲットテーブルはnテーブルである必要があるため、この場合は [ 納品書 ] テーブルになります。上記のフィールド順にバインドを定義していれば、最初のバインド行は [ 納品書 ] テーブルのフィールドになるため、引数 < tableID > を省略することもできます。

例2：

[ 顧客 ] 社名  
 [ 顧客 ] 住所都市

以上のフィールドの値が欲しいものとします。

ただし、[ 納品書 ] テーブルのカレントセレクションに関連のある顧客だけが必要です。

この場合、リレートの元になるカレントセレクションは [ 納品書 ] テーブルのセレクションであるため、ターゲットテーブルは [ 納品書 ] ということになります。[ 納品書 ] のフィールドのデータを取得する必要がないとしても、ターゲットテーブルには [ 納品書 ] テーブルを指定しなければなりません。

## エラーコード

**OP Selection to array**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
-9969	不正なフィールドタイプが要求されました。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10135	引数型が不正です。
10136	指定された接続 ID に対応する接続がありません。
10137	バインドが存在しません。
10139	バインドが未定義です。
10153	型変換の不可能な引数データがあります。
10154	このコマンドを今すぐには実行できません。

### 例題

**OP Array to selection** の例題を参照してください。

### 参照

OP Define bind by numbers、OP Define bind by pointer、OP Selection to array、  
SELECTION TO ARRAY

## OP Distinct values

**OP Distinct values** (connectionID; tableID; fieldID; distinctValues) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	ターゲットテーブルの番号 (セクションを利用するテーブル)
fieldID	倍長整数	対象とするフィールドの番号
distinctValues	配列	カレントセクションの範囲で出現するすべての値を保持した配列
Error	倍長整数	エラーコード

### 説明

**OP Distinct values** は、引数 < distinctValues > に渡した配列に値を埋めます。

値は、カレントセクションの範囲で引数 < tableID >、< fieldID > で指定したフィールドに出現するすべての値で構成されます。重複した値は配列へコピーされません。したがって、ほとんどの場合、カレントセクションのレコード数より配列の要素数は少なくなります。

指定できるフィールドはインデックス付きの固定長文字列のみですが、引数 < distinctValues > に指定できる配列の型は固定長文字列、テキスト、整数、倍長整数、実数、日付、ブールのいずれでも構いません。

このルーチンは対象とするテーブルのカレントセクション、カレントレコードを変更しません。

### エラーコード

**OP Distinct values** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
-9969	不正なフィールド型が要求されました。
-9970	フィールドにインデックスが付いていません。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10135	引数型が不正です。
10136	指定された接続 ID に対応する接続がありません。

10139                    バインドが未定義です。  
10153                    型変換の不可能な引数データがあります。  
10154                    このコマンドを今すぐには実行できません。  
10158                    インデックス付きの固定長文字列フィールドではありません。

## 例題

この例題では、ダイアログを表示して、今日の受注元顧客の住所（都市名）をスクロールエリアに表示します：

```
C_LONGINT (vRecords; $ErrCode)  
C_STRING (10;vValue)  
ARRAY STRING (25; aCities;0)
```

```
  ` 今日の日付を求める。  
vValue:= String ( Current date )
```

```
  ` 発行日が今日の納品書を検索する。  
$ErrCode:= OP Single query (vConnectID;2;3;"=";->vValue;vRecords)
```

```
  ` 納品書のセレクションにリレートしているすべての顧客レコードを  
  ` セレクションにする。  
$ErrCode:= OP Many to one join (vConnectID;1;2)
```

```
  ` 顧客の住所（都市名）を配列へ読み込む。  
$ErrCode:= OP Distinct values (vConnectID;1;5;aCities)
```

## 参照

DISTINCT VALUES、OP Get one field number、OP Selection to array、OP Single query



## OP Array to selection

---

### OP Array to selection (connectionID; bindID) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
bindID	倍長整数	バインドID
Error	倍長整数	エラーコード

#### 説明

**OP Array to selection** は、ローカルの（クライアント側の）配列の値をレコードへコピーします。配列とフィールドの対応は、引数 < bindID > で指定されるバインドにあらかじめ定義された組み合わせを使用します。

このルーチンの場合、使用するバインドに登録されるフィールドはすべて同じテーブルに属していなければなりません。そのテーブルにカレントセクションがない場合は何も行いません。データのコピーは配列とレコードの順序に沿って実行されます。配列の要素数がカレントセクションのレコード数より多い場合は、新しいレコードが作成されます。

バインドに登録された配列の要素数が一致していない場合は、もっとも少ない要素数が有効になります。

## エラーコード

**OP Array to selection**の実行に成功した場合、Errorは「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
-9969	不正なフィールド型が要求されました。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10135	引数型が不正です。
10136	指定された接続 ID に対応する接続がありません。
10137	バインドが存在しません。
10138	バインドにあるフィールドがターゲットテーブルにリレートしていません。
10139	バインドが未定義です。
10153	型変換の不可能な引数データがあります。
10154	このコマンドを今すぐには実行できません。
10162	配列要素が0です。

## 例題

この例題では [ 顧客 ] 売上フィールドの値を配列へコピーし、その日の納品書の合計金額を合算して書き戻しています。

```
C_LONGINT ($i;$ErrCode)
C_LONGINT ($custSalesBind;$invAmountBind)

ARRAY REAL (aCustSales;0)
ARRAY REAL (aInvAmount;0)
ARRAY STRING (10;aCustID;0)
ARRAY STRING (10;aInvCustID;0)

`新しいバインドを作成。
$ErrCode:= OP Create bind ($custSalesBind)
` [ 顧客 ] 売上 ( 1;7 ) と配列 aCustSales をバインド。
$ErrCode:= $ErrCode + OP Define bind by pointer ($custSalesBind;1;7;->aCustSales)

` [ 顧客 ] 顧客 ID ( 1;1 ) と配列 aCustID をバインド。
$ErrCode:= $ErrCode + OP Define bind by pointer ($custSalesBind;1;1;->aCustID)

`新しいバインドを作成。
$ErrCode:= OP Create bind ($invAmountBind)
` [ 納品書 ] 合計 ( 2;4 ) と配列 aInvAmount をバインド。
$ErrCode:= $ErrCode+OP Define bind by pointer($invAmountBind;2;4;->aInvAmount)

` [ 納品書 ] 顧客 ID ( 2;2 ) と配列 aInvCustID をバインド。
$ErrCode:= $ErrCode + OP Define bind by pointer ($invAmountBind;2;2;->aInvCustID)
```

```
C_LONGINT (vRecords; $ErrCode)
C_STRING (10;vValue)
    `今日の日付。
vValue:= String (Current date)
    `今日の納品書を探す。
$errCode:= OP Single query (vConnectID;2;3;"=";->vValue;vRecords)
    `今日の納品書レコードの合計金額フィールド値を配列 aInvAmountへロード。
$errCode:= OP Selection to array (vConnectID;$invAmountBind;2)
    `納品書の宛先顧客を調べるために [顧客] テーブルにセレクションを作成。
$errCode:=OP Many to one join (vConnectID;1;2)
    `顧客ごとの売上合計金額を配列 aCustSalesへロード。
$errCode:= OP Selection to array (vConnectID;$custSalesBind;1)
    `納品書一枚一枚に対して ...
For ($i;1; Size of array (aInvAmount) )
    $cust:= Find in array (aCustID;aInvCustID{$i})
    If ($cust>0)
        `納品書ごとの合計金額を顧客ごとの売上合計へ合算していく。
        aCustSales{$cust} := aCustSales{$cust} + aInvAmount{$i}
    End if
End for
    `顧客テーブルへ書き戻す。
$errCode:= OP Array to selection (vConnectID;$custSalesBind)
```

### 参照

ARRAY TO SELECTION、OP Array to selection、OP Create bind、OP Many to one join、OP Selection to array、OP Single query

## OP Subselection to array

---

**OP Subselection to array** (connectionID; bindID; firstRecord; lastRecord{; tableID}) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
bindID	倍長整数	バインドID
firstRecord	倍長整数	取り出したい範囲の最初のレコードの位置
lastRecord	倍長整数	取り出したい範囲の最後のレコードの位置
tableID	倍長整数	対象とするテーブルの番号
Error	倍長整数	エラーコード

### 説明

**OP Subselection to array** は、ロードするレコードの範囲がカレントセレクション全体ではなく、引数で指定した範囲に限定できる点以外は、**OP Selection to array** と同じように動作します。

ロードするレコードの範囲は、引数 < firstRecord >、< lastRecord > にレコードの選択位置番号で指定します。

### 例題

この例題では、顧客名をテキストファイルへ書き出しています。大きすぎる配列を作らないために、配列へコピーするレコード数を100件ごとに小分けにしています。

```
C_LONGINT ($errCode;vRecords;$custNamesBind)
ARRAY STRING (40;aCustName;0)
  `新しいバインドを作成。
$errCode:= OP Create bind ($custNamesBind)
  `[顧客]社名(1;2)と配列aCustNameをバインド。
$errCode:= $errCode + OP Define bind by pointer ($custNamesBind;1;2;
                                                    ->aCustName)
  `[顧客]テーブル(#1)の全件を選択。
$errCode:= OP All records (vConnectID;1)
  `レコード数を得る
$errCode:= OP Records in selection (vConnectID;1;vRecords)
$i:=0
While ($i<vRecords) `100件の顧客名を配列へコピー。
  $errCode:= OP Subselection to array (vConnectID; $custNamesBind ; $i+1;
                                          $i+100)
  ExportCustNames (->aCustName)
  $i := $i + 100
End while
$errCode:= OP Delete bind ($custNamesBind)
```

**参照**

OP Define bind by pointer、OP Get one field number、OP Records in selection、OP Selection to array、OP Subselection to array



## トランザクション、序論

---

トランザクションとはリレートされたデータに関してサーバとの接続が保たれた間に行われた一連の更新作業を指します。トランザクションが受け入れられるまでは、更新内容は保存されません。トランザクションがキャンセルされた場合や他のイベントによって完了しなかった場合も、更新内容は保存されません。

トランザクションの間、すべての更新内容はローカルのテンポラリバッファに保存されます。トランザクションが受け入れられて初めて、更新内容がデータベースに恒久的に保存されます。トランザクションがキャンセルされると、更新内容は保存されません。

トランザクションがキャンセルもしくは受け入れられた後、トランザクションを実行していたコネクションのカレントセレクションは、すべてのテーブルにわたってクリアされます。これはトランザクションがテンポラリのレコード番号を使用していたためです。同じ理由でトランザクション中に命名セレクションを使用する場合も注意が必要です。トランザクションがキャンセルまたは受け入れられた後、トランザクションの実行前、もしくは実行中に作成された命名セレクションは無効なレコードアドレスを含んでいるかもしれません。例えば、命名セレクションは、トランザクション中に削除されたレコードのアドレスや作成されたレコードのテンポラリのアドレスを含んでいるかもしれません。

以下のルーチンはトランザクションを管理します。

**OP Start transaction** : トランザクションを開始します。

**OP Validate transaction** : **OP Start transaction** で開始したトランザクション中に行った更新内容を保存します。

**OP Cancel transaction** : **OP Start transaction** で開始したトランザクション中に行った更新内容を破棄し、トランザクションをキャンセルします。

### 参照

OP Cancel transaction、OP Start transaction、OP Validate transaction

## OP Start transaction

---

### OP Start transaction (connectionID) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
Error	倍長整数	エラーコード

#### 説明

**OP Start transaction** は引数 < connectionID > で指定された接続を使用してトランザクションを開始します。

すべての更新内容はトランザクションが受け入れられる（コミットされる）か、キャンセルされる（ロールバックされる）まで、仮の保存状態で保たれます。

接続ごとに1つずつのトランザクションを開始できます。トランザクションのネストは行えません。トランザクション中に、更にトランザクションを開始しても4D Serverはそれを無視します。

#### エラーコード

**OP Start transaction** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続IDに対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

#### 例題

以下の例題では新しい納品書を作成しようとしています。また [ 顧客 ] テーブルの総売上高フィールドの更新も行っています。この2つのデータベース処理を不可分なものにするため（処理の原子性を保つため）にトランザクションを使用しています。

```
C_LONGINT ($BindInvID;$BindInvID;$ErrCode)
C_LONGINT ($transErr)
```

```
C_DATE (vInvDate)
C_REAL (vInvTotal;vCustSales)
C_STRING (30;vCustID;vInvoiceID)
```

```
C_LONGINT (vTableInvoices;vTableCustomers;vRecords;vIsLocked;$unused1)
C_LONGINT (vFieldAmount;vFieldInvCustID;vFieldInvDate;vFieldInvID;$unused2)
C_LONGINT (vFieldCustID;vFieldCustName;vFieldSales)
```



```

`[納品書]テーブル番号と[納品書]納品書番号フィールド番号を取得する。
$ErrCode:= OP Get one field number (vConnectID;"[納品書]納品書番号";
                                         vTableInvoices;vFieldInvID)

`[納品書]顧客番号フィールド番号を取得する。
$ErrCode:= OP Get one field number (vConnectID;"[納品書]顧客番号";$unused1;
                                         vFieldInvCustID)

`[納品書]納品書日付フィールド番号を取得する。
$ErrCode:= OP Get one field number (vConnectID;"[納品書]納品書日付";$unused1;
                                         vFieldInvDate)

`[納品書]納品書合計フィールド番号を取得する。
$ErrCode:= OP Get one field number (vConnectID;"[納品書]納品書合計";$unused1;
                                         vFieldAmount)

`[顧客]テーブル番号と[顧客]顧客番号フィールド番号を取得する。
$ErrCode:= OP Get one field number (vConnectID;"[納品書]顧客番号";
                                         vTableCustomers;vFieldCustID)

`[顧客]総売上高フィールド番号を取得する。
$ErrCode:= OP Get one field number (vConnectID;"[顧客]総売上高";$unused1;
                                         vFieldSales)

`[顧客]社名フィールド番号を取得する。
$ErrCode:= OP Get one field number (vConnectID;"[顧客]社名";$unused1;
                                         vFieldCustName)

`...サーバへ接続
`新しい納品書作成のためのバインドを作成。
$ErrCode:= OP Create bind ($BindInvID)
$ErrCode:= OP Define bind by numbers ($BindInvID;vTableInvoices;vFieldInvID;
                                         0;0;"vInvoiceID")
$ErrCode:= OP Define bind by numbers ($BindInvID;vTableInvoices;
                                         vFieldInvCustID;0;0;"vCustID")
$ErrCode:= OP Define bind by numbers ($BindInvID;vTableInvoices;vFieldInvDate;
                                         0;0;"vInvDate")
$ErrCode:= OP Define bind by numbers ($BindInvID;vTableInvoices;vFieldAmount;
                                         0;0;"vInvTotal")

`顧客レコードのロードと更新のためのバインドを作成。
$ErrCode:= OP Create bind ($BindCustID)
$ErrCode:= OP Define bind by numbers ($BindCustID;vTableCustomers;
                                         vFieldSales;0;0;"vCustSales")
$ErrCode:= OP Define bind by numbers ($BindCustID;vTableCustomers;
                                         vFieldCustID;0;0;"vCustID")

`すべてのデータベース処理をトランザクションで囲む。
$transErr:= OP Start transaction (vConnectID)
`顧客レコードのカレントレコードを指定する。
vValue:= "ACME Inc."
$ErrCode:= OP Single query (vConnectID;vTableCustomers;vFieldCustName;"=";
                               ->vValue;vRecords)

```

```

`目的のレコードがあったと仮定して...
  `レコードを排他的に使用するために Read/write モードにする。
  $errCode:= OP Set access mode (vConnectID;vTableCustomers;1)

`レコードをロード
Repeat
  vIsLocked:= 0
  `納品書には顧客番号が必要なので、先に [ 顧客 ] テーブルを読み込んでいる。
  `Unlock 状態で読み込めるまでループするのは、現在の総売上高を調べた後に
  `他でクライアントから更新されることを防ぐため。
  $errCode:= OP Load record (vConnectID;$BindCustID;vTableCustomers;
                                vIsLocked)
Until (vIsLocked =0)  `ロック状態が長く続いていることはないという前提で ...

  `納品書レコードを作成、データを収集し、納品書情報を作成する。
  vInvoiceID:= "0102030405"
  vInvDate:= Current date
  vInvTotal:= 100
  `バインド $BindCustID に含まれる vCustID はバインド $BindCustID にも
  `含まれており、既に $BindCustID を使用した上記の OP Load record で値が
  `セットされている。

  `新しい納品書をサーバへ送る。
  $transErr:= $transErr + OP New record (vConnectID;$BindInvID)

  `今回は [ 顧客 ] レコードを更新。
  vCustSales:= vCustSales + vInvTotal
  $transErr:= $transErr + OP Update record (vConnectID;$BindCustID)

If ($transErr = 0)  `一連の処理中にエラーが発生していなければ ...
  `[ 顧客 ] [ 納品書 ] 双方の更新内容を保存する。
  $errCode:= OP Validate transaction (vConnectID)
Else
  `OP Start transaction以降のすべての処理(New record, Update record)を
  `キャンセルする。
  $errCode:= OP Cancel transaction (vConnectID)
End if

  $errCode:= OP Delete bind ($BindInvID)
  $errCode:= OP Delete bind ($BindCustID)

```

## 参照

OP Cancel transaction、OP Validate transaction、START TRANSACTION

## OP Validate transaction

---

### OP Validate transaction (connectionID) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
Error	倍長整数	エラーコード

#### 説明

**OP Validate transaction** は引数 < connectionID > で指定された接続内で **OP Start transaction** を使って開始されたトランザクションを受け入れます。4D 標準コマンド **START TRANSACTION** で開始したトランザクションは受け入れません。

**OP Validate transaction** はトランザクション中に発生したすべての更新内容を保存しません。

#### エラーコード

**OP Validate transaction** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続IDに対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

#### 例題

**OP Start transaction** の例題を参照してください。

#### 参照

OP Cancel transaction、OP Start transaction、VALIDATE TRANSACTION

## OP Cancel transaction

---

### OP Cancel transaction (connectionID) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
Error	倍長整数	エラーコード

#### 説明

**OP Cancel transaction** は引数 < connectionID > で指定された接続内で **OP Start transaction** を使って開始されたトランザクションをキャンセルします。4D 標準コマンド **START TRANSACTION** で開始したトランザクションはキャンセルできません。

**OP Cancel transaction** はトランザクション中に発生したすべての処理をキャンセルし、データベースを変更しません。

#### エラーコード

**OP Cancel transaction** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

#### 例題

**OP Start transaction** の例題を参照してください。

#### 参照

CANCEL TRANSACTION、OP Start transaction、OP Validate transaction

## 命名セクション、序論

---

命名セクションは、複数のセクションを同時に操作する簡単な手段を提供します。命名セクションはレコードのリストで、カレントセクションのレコードのソート順序に沿って並んでいます。また同時にカレントレコードも記録して、名前を付けて保存されます。

セクションを変更したとしても、あらかじめ命名セクションを作っておけば、すぐに元のセクション状態に戻ることができます。同一接続内の単一のテーブルに対しても複数の命名セクションを持つことができます。

この章では以下のルーチンについて解説を行います：

**OP Copy named selection**：カレントセクションを命名セクションとして記憶します。

**OP Cut named selection**：カレントセクションを命名セクションとして記憶し、カレントセクションをクリアします。

**OP Use named selection**：あらかじめ**OP Copy named selection**、**OP Cut named selection**で作成しておいた命名セクションをカレントセクションにします。

**OP Clear named selection**：**OP Copy named selection**、**OP Cut named selection**で作成した命名セクションをメモリ上から消去します。

### プロセス/インタープロセス命名セクション

4D OPENでは1人のユーザが同じサーバに何度も接続することが可能で、それぞれの接続はお互いに独立して動いています。したがって、ユーザは各々の接続で異なるデータベース操作を実行可能です。例えば、ある1つの接続でレコードの登録を行い、別の接続では印刷を行うといった具合にです。

命名セクションを作成する際には、それをすべてのプロセスで共有するかどうかを選択できます。

共有する命名セレクションは“ インタープロセス命名セレクション ”と呼ばれます。それはすべてのプロセスから利用可能で、別のプロセスで作成された命名セレクションを使用してカレントセレクションを復元することが可能です。

共有できない命名セレクションは“ プロセス命名セレクション ”と呼ばれます。このタイプの命名セレクションは、作成したプロセス内でだけ利用可能です。

プロセス命名セレクションとインタープロセス命名セレクションの区別はその名前で行います。インタープロセス命名セレクションは“ <> ”(2つの不等号記号)で始まります。例えば、“ <>上得意顧客 ”はインタープロセス命名セレクションを意味します。

#### 参照

OP Clear named selection、OP Copy named selection、OP Cut named selection、OP Use named selection

## OP Copy named selection

**OP Copy named selection** (connectionID; tableID; selectionName) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	テーブル番号
selectionName	固定長文字列	命名セレクションに付ける名前
Error	倍長整数	エラーコード

### 説明

**OP Copy named selection** は、引数 < tableID > で指定されたテーブルのカレントセレクションを引数 < SelectionName > の名前で命名セレクションとして保存します。このルーチンは引数 < tableID > で示されるテーブルのカレントセレクションは変更しません。

### エラーコード

**OP Copy named selection** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続IDに対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 例題

この例題では納品書の命名セレクションを作り、バックグラウンドプロセスを起動します。ユーザが納品書の作業を続行する間、起動したプロセスでは命名セレクションに含まれる納品書の印刷を行っています。

```
C_LONGINT ($ErrCode;vRecords)
```

```
ARRAY LONGINT (arTableID;0)
```

```
ARRAY LONGINT (arFieldID;0)
```

```
ARRAY TEXT (arLogOp;0)
```

```
ARRAY TEXT (arQueOp;0)
```

```
ARRAY TEXT (arValues;0)
```

```
UserSelectInvoices `ダイアログを表示してユーザに検索条件を指定させる。
```

```
$errCode:= OP Multi query (<>vConnectID;2;arFileID;arFieldID;arLogOp;arQueOp;  
arValues;vRecords)
```

```

    ` [納品書]テーブルのカレントセレクションをインタープロセス命名セレクション
    ` として記憶する。
    $errCode:= OP Copy named selection (<>vConnectID;2;"<>Invoices to print")
    ` OP Cut named selection を使う代わりに OP Copy named selection を使って
    ` カレントセレクションを維持しておく。
    $printProcess:= New process ("PrintInvoices";32000;"Background printing")

    ` 納品書の一覧を表示する。
    UserDisplayInvoices
    ...

```

以下にメソッド"PrintInvoices"で行う処理の典型例を示します。

```

    ` メソッド"PrintInvoices"はバックグラウンドで命名セレクション
    ` "<>Invoices to print"に含まれる納品書を印刷する。

    ` カレントセレクションの"レプリカ"として命名セレクションを作る。
    $errCode:= OP Use named selection (<>vConnectID;2;"<>Invoices to print")

    If ($errCode=0)
        ` サーバマシン上のメモリを開放するために命名セレクションを
        ` クリアする。
        $errCode:= OP Clear named selection (<>vConnectID;"<>Invoices to print")
        ` カレントセレクションのレコードを数える。
        $errCode:= OP records in selection (<>vConnectID;2;vRecords)

        For ($i;1;vRecords)
            $errCode:= OP Goto selected record (<>VconnectID;2;$i)
            PrintOneInvoice (<>vConnectID)
        End for
    End if

```

#### 参照

OP Clear named selection、OP Cut named selection、OP Use named selection



## OP Cut named selection

**OP Cut named selection** (connectionID; tableID; selectionName) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	テーブル番号
selectionName	固定長文字列	命名セクションに付ける名前
Error	倍長整数	エラーコード

### 説明

**OP Cut named selection** は引数 < tableID > で指定されたテーブルのカレントセクションを、引数 < SelectionName > の名前で命名セクションとして保存します。このルーチンの実行後は、引数 < tableID > で示されるテーブルのカレントセクションは空になります。

**OP Use named selection** を用いて上記の命名セクションを使用した場合、命名セクションは消去されます。

### エラーコード

**OP Cut named selection** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続IDに対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 例題

この例題は **OP Copy named selection** の例題によく似ていますが、複数の命名セクションを取り扱っている点が異なります。ユーザはセクションを作成しても表示しないので、カレントセクションを維持するために **OP Copy named selection** を使用する必要はありません。代わりに **OP Cut named selection** を使用してサーバのメモリを節約することをお勧めします。

```
C_LONGINT ($ErrCode;vRecords)
```

```
ARRAY LONGINT (arTableID;0)
```

```
ARRAY LONGINT (arFieldID;0)
```

```
ARRAY TEXT (arLogOp;0)
```

ARRAY TEXT (arQueOp;0)  
ARRAY TEXT (arValues;0)

ARRAY STRING (0;<>arPrintRequests;0)

Repeat

` 検索条件入力用ダイアログを表示。  
 UserSelectInvoices

` ユーザの指定した条件で検索実行する。  
 \$errCode:= **OP Multi query** (<>vConnectID;2;arFileID;arFieldID;arLogOp;  
 arQueOp;arValues;vRecords)  
 ` まず、次のようなフォーマットでユニークな名前を作る:  
 ` <>InvoicesToPrint0001, <>InvoicesToPrint0002...  
 \$selectionName:= "<>InvoicesToPrint" + **String (Size of array**  
 (<>arPrintRequest)+1;"0000")

` Print リクエスト配列へ要素を追加していく。  
 **INSERT ELEMENT** (<>arPrintRequest;**Size of array** (<>arPrintRequest)+1;1)

` [納品書]テーブルのカレントセレクションを  
 ` インタープロセス命名セレクションとして保存。  
 ` サーバ上の消費メモリを節約するために **OP Copy named selection** の  
 ` 代わりに **OP Cut named selection** を使う。  
 \$errCode:= **OP Cut named selection** (<>vConnectID;2;\$selectionName)  
 <>arPrintRequest{**Size of array** (<>arPrintRequest)} := \$selectionName

Until (UserClickedPrintAndQuit=1)

...

以下はメソッド"BackgroundPrinter"の内容として典型的な実装例です。

` バックグラウンドで配列 <>arPrintRequest によって指定される納品書を  
 ` 印刷する。  
 ` Startup でプロセスは起動し、バックグラウンドでリクエストを待って  
 ` 待機する。

While (True) ` 4th Dimension を quit するまで LOOP し続ける。

If (**Size of array** (<>arPrintRequest)=0)  
 ` 処理すべき印刷リクエストがなければ 10 秒 Delay する。  
 **DELAY PROCESS (Current process;60\*10)**  
 Else  
 ` セレクションの名前を得る。  
 \$SelectionName:= <>arPrintRequest{1}  
 ` 印刷リクエスト配列から該当要素を取り除く。  
 **DELETE ELEMENT** (<>arPrintRequest;1;1)  
 ` 命名セレクションをカレントセレクションにする。

- ・ **OP Cut named selection** で作った命名セクションなので
- ・ **OP Use named selection** でカレントセクションに変更するだけで、
- ・ 自動的に命名セクションはクリアされる。

```
$errCode:= OP Use named selection (<>vConnectID;2;$SelectionName)
```

```
If ($errCode=0)
```

- ・ カレントセクションのレコード数を数える。

```
$errCode:= OP records in selection (<>vConnectID;2;vRecords)
```

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

```
    $errCode:= OP Goto selected record (<>VconnectID;2;$i)
```

```
    PrintOneInvoice (<>vConnectID)
```

```
End for
```

```
End if
```

```
End if
```

```
End while
```

### 参照

OP Clear named selection、OP Copy named selection、OP Single query、OP Use named selection

## OP Use named selection

---

**OP Use named selection** (connectionID; tableID; selectionName) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	テーブル番号
selectionName	固定長文字列	使用する命名セレクションの名前
Error	倍長整数	エラーコード

### 説明

**OP Use named selection** は引数 < selectionName > で指定された命名セレクションをカレントセレクションにします。引数 < tableID > には対応するテーブルの番号が返ってきます。引数 < SelectionName > の命名セレクションを作成した時のカレントレコードも復元されません。

カレントセレクション同様、命名セレクションはレコードの実データは含みません。単にレコード番号のリストとして存在しています。そのため、命名セレクションを作った後にレコードの削除、修正を施した場合、命名セレクションからカレントセレクションを復元しようとした時に直面する問題としては以下のものが挙げられます。

- ・削除したレコードが空のレコードとしてカレントセレクションに含まれているかもしれません。
- ・命名セレクション作成後に新たに追加したレコードがカレントセレクションに含まれているかもしれません。
- ・命名セレクション作成後に施した変更がそのまま現れているかもしれません。

**OP Copy named selection** で作成した命名セレクションは、**OP Clear named selection** を呼んでクリアしてください。**OP Cut named selection** で作成した命名セレクションは、**OP Use named selection** を実行した時点で自動的にクリアされます。

### エラーコード

**OP Use named selection** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9977	存在しない命名セレクションです。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続IDに対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

## 例題

OP Copy named selection と OP Cut named selection の例題を参照してください。

## 参照

OP Clear named selection、OP Copy named selection、OP Cut named selection

## OP Clear named selection

---

**OP Clear named selection** (connectionID; tableID; selectionName) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	テーブル番号
selectionName	固定長文字列	使用する命名セレクションの名前
Error	倍長整数	エラーコード

### 説明

**OP Clear named selection** は引数 < selectionName > で指定された命名セレクションをサーバ上のメモリからクリアします。このルーチンはカレントセレクションには影響を与えません。

**OP Cut named selection** で作成し、**OP Use named selection** で使用した命名セレクションはクリアする必要はありません。

### エラーコード

**OP Clear named selection** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9946	存在しない命名セレクションなのでクリアできません。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続IDに対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 例題

**OP Copy named selection** の例題を参照してください。

### 参照

OP Copy named selection、OP Cut named selection、OP Use named selection

## 一連の値、序論

---

この章のルーチンは、カレントセクションのレコードの一連の値について計算を行います。これらのルーチンは数値フィールドに対してのみ有効です。

数値フィールドには整数、倍長整数、実数があります。

**OP Sum** : カレントセクションを対象に、特定のフィールド値の合計を求めます。

**OP Average** : カレントセクションを対象に、特定のフィールド値の平均を求めます。

**OP Min** : カレントセクションを対象に、特定のフィールド値の最小値を求めます。

**OP Max** : カレントセクションを対象に、特定のフィールド値の最大値を求めます。

### 参照

OP Average、OP Max、OP Min、OP Sum

## OP Sum

---

### OP Sum (connectionID; tableID; fieldID; Result) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	テーブル番号
fieldID	倍長整数	計算対象フィールド番号
Result	変数	合計値
Error	倍長整数	エラーコード

### 説明

**OP Sum** は、カレントセクションに含まれる全レコードの引数 < fieldID > で示されるフィールドの値の合計を求めます。引数 < fieldID > で示されるフィールドがインデックス付きであった場合、インデックスが使われます。所要時間はより短くなります。

### エラーコード

**OP Sum** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9969	不正なフィールド型が要求されました。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 例題

この例題では、その日の最小額、最大額だった納品書の金額及び平均額、総合計をメッセージと共に表示します。

```
C_LONGINT (vTable;vFieldAmount;vFieldDate;$errCode;vRecords)
C_STRING (10;vValue)
C_REAL (vAverage;vMax;vMin;vSum)
  `テーブル番号、フィールド番号を得る。
$errCode := OP Get one field number (vConnectID;"[Invoices]Total";vTable;
                                         vFieldAmount)
$errCode := OP Get one field number (vConnectID;"[Invoices]Invoice date";vTable;
                                         vFieldDate)
```



```
`今日の日付を得る。
vValue:= String ( Current date )
`今日の納品書だけのセレクションを作る。
$errCode:= OP Single query selection (vConnectID;vTable;vFieldDate;"=";->vValue;
vRecords)

`総合計を得る。
$errCode := $ErrCode + OP Sum (vConnectID;vTable;vFieldAmount;vSum)

`最少額を得る。
$errCode := $ErrCode + OP Min (vConnectID;vTable;vFieldAmount;vMin)

`最大額を得る。
$errCode := $ErrCode + OP Max (vConnectID;vTable;vFieldAmount;vMax)

`平均金額を得る。
$errCode := $ErrCode + OP Average (vConnectID;vTable;vFieldAmount;vAverage)

If ($ErrCode=0)
  $mes := " Today ' Errors are " + Char (Carriage return)
  $mes := $mes + " 最小金額 : " + String (vMin;"### ###.00 ) +
  Char (Carriage return)
  $mes := $mes + " 最大金額 : " + String (vMax;"### ###.00 ) +
  Char (Carriage return)
  $mes := $mes + " 平均金額 : " + String (vAverage;"### ###.00 ) +
  Char (Carriage return)
  $mes := $mes + " 総合計 : " + String (vSum;"### ###.00 )
  ALERT ($mes)
End if
```

参照

OP Average、OP Max、OP Min

## OP Average

---

**OP Average** (connectionID; tableID; fieldID; Result) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	テーブル番号
fieldID	倍長整数	計算対象フィールド番号
Result	変数	平均値
Error	倍長整数	エラーコード

### 説明

**OP Average**は、カレントセクションに含まれる全レコードについて引数 < fieldID > で示されるフィールドの値の平均を求めます。引数 < fieldID > で示されるフィールドがインデックス付きであった場合、インデックスが使われます。所要時間はより短くなります。

### エラーコード

**OP Average**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9969	不正なフィールド型が要求されました。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続IDに対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 例題

**OP Sum**の例題を参照してください。

### 参照

OP Max、OP Min、OP Sum

## OP Min

---

**OP Min** (connectionID; tableID; fieldID; Result) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	テーブル番号
fieldID	倍長整数	計算対象フィールド番号
Result	変数	最小値
Error	倍長整数	エラーコード

### 説明

**OP Min** は、カレントセクションに含まれる全レコードについて引数 < fieldID > で示されるフィールドの値の最小値を求めます。引数 < fieldID > で示されるフィールドがインデックス付きであった場合、インデックスが使われます。所要時間はより短くなります。

### エラーコード

**OP Min** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9969	不正なフィールド型が要求されました。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 例題

**OP Sum** の例題を参照してください。

### 参照

OP Average、OP Max、OP Sum

## OP Max

---

**OP Max** (connectionID; tableID; fieldID; Result) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	テーブル番号
fieldID	倍長整数	計算対象フィールド番号
Result	変数	最大値
Error	倍長整数	エラーコード

### 説明

**OP Max** は、カレントセクションに含まれる全レコードについて引数 < fieldID > で示されるフィールドの値の最大値を求めます。引数 < fieldID > で示されるフィールドがインデックス付きであった場合、インデックスが使われます。所要時間はより短くなります。

### エラーコード

**OP Max** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9969	不正なフィールド型が要求されました。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 例題

**OP Sum** の例題を参照してください。

### 参照

OP Average、OP Min、OP Sum

## バインド、序論

---

バインドは、独自に定義できるサーバオブジェクトとローカルオブジェクトのリンク（組み合わせ）です。サーバオブジェクトはテーブル/フィールド、ローカルオブジェクトはテーブル/フィールド、変数、配列を指定できます。

レコードを変更する、新しいレコードを作る、カレントレコードをロードする場合はデータを表示する/変更するためのローカル側フィールド、変数とサーバ側フィールドをバインドします。

配列を使って4D Serverデータベースヘデータを戻す、送出する場合は、データを保持したローカル側配列とサーバ側フィールドをバインドします。

データ操作のためにどのルーチンを使うべきかは、使用するバインドにフィールド、変数、配列のどれを含めるかによります。以下の表では、4th Dimensionオブジェクトの型とそれに適合したルーチンを示しています。

オブジェクト	ルーチン
変数 / フィールド	<b>OP New record</b>
	<b>OP Update record</b>
	<b>OP Load record</b>
	<b>OP Goto record</b>
	<b>OP Goto selected record</b>
配列	<b>OP Selection to array</b>
	<b>OP Array to selection</b>
	<b>OP Subselection to array</b>

以下のルーチンを使ってバインドの作成、内容定義、消去を行います。

**OP Create bind** : メモリ上に空の新規バインドを作ります。

**OP Define bind by numbers / OP Define bind by pointer** : バインド内容の定義を行います。

注 : 同一バインド中の内容定義に **OP Define bind by numbers** と **OP Define bind by pointer** を混在して使用することが可能です。

**OP Delete bind** : バインドを消去し、占有されていたメモリを開放します。

**OP Set format** : 文字型データとそれ以外のデータを相互に変換するためのフォーマットを設定します。

#### 参照

OP Create bind、OP Define bind by numbers、OP Define bind by pointer、OP Delete bind、OP Set format

## OP Create bind

---

### OP Create bind (bindID) Error

引数	タイプ	説明
bindID	倍長整数	新しいバインドのID
Error	倍長整数	エラーコード

### 説明

**OP Create bind** は、メモリ上に空のバインドを作成し、その参照番号としてバインドIDを引数 < bindID > へ返します。

既存のバインドへの新たなバインド定義の追加は、**OP Define bind by numbers** や **OP Define bind by pointer** を使用します。

データ操作に際して、このバインドIDを渡すことによって、サーバ側のオブジェクトに抱きあわせるローカル側オブジェクトを指示します。

バインドの使用が終了したら、**OP Delete bind** を呼び、占有していたメモリを開放します。

**OP Define bind by numbers** と **OP Define bind by pointer** の例題を参照してください。

### エラーコード

**OP Create bind** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 参照

OP Define bind by numbers、OP Define bind by pointer、OP Delete bind

## OP Delete bind

---

### OP Delete bind (bindID) Error

引数	タイプ	説明
bindID	倍長整数	消去するバインドのID
Error	倍長整数	エラーコード

### 説明

**OP Delete bind**はバインドを消去し、占有していたメモリを開放します。このルーチンの使用は、対象とするバインドを使い終わった後だけにしてください。

### エラーコード

**OP Delete bind**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10137	バインドが存在しません。
10154	このコマンドを今すぐには実行できません。

### 参照

OP Create bind、OP Define bind by numbers、OP Define bind by pointer



## OP Define bind by numbers

**OP Define bind by numbers** (bindID; tableID; fieldID; localTableID; localFieldID;

variableName) Error

引数	タイプ	説明
bindID	倍長整数	定義を追加するバインドのID
tableID	倍長整数	サーバ側テーブル番号
fieldID	倍長整数	サーバ側フィールド番号
localTableID	倍長整数	ローカル側テーブル番号 (変数をバインドする時は「0」を指定)
localFieldID	倍長整数	ローカル側フィールド番号 (変数をバインドする時は「0」を指定)
variableName	固定長文字列	変数、配列の名前 (フィールドをバインドする時は""を指定)
Error	倍長整数	エラーコード

### 説明

**OP Define bind by numbers**は、引数 < bindID > で示される既存バインドへ、新たなバインド定義を追加します。バインドするサーバ側オブジェクトの指定にはテーブル/フィールド番号を使用します。

例えば、[納品書]番号フィールドをバインドする場合 - [納品書]テーブルが2番目のテーブルで[納品書]番号フィールドが[納品書]テーブルの1番目のフィールドとすると - 引数 < tableID > には「2」を、引数 < fieldID > には「1」を渡します。

さらにローカル側オブジェクトとしてフィールドを使用する場合には、同様にテーブル番号、フィールド番号を引数 < localTableID >、< localFieldID > へ渡します。この場合、引数 < variableName > へは"" (空) を渡しておきます。

ローカルオブジェクトに変数や配列を使用する場合、変数名/配列名を引数 < variableName > へ渡し、引数 < localTableID >、< localFieldID > へは「0」を渡します。

インタープロセス/プロセス変数、配列を使用できますが、ローカル変数/配列は使用できません。

## エラーコード

**OP Define bind by numbers**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10137	バインドが存在しません。
10140	ローカル側フィールドが存在しません。

## 例題

この例題では[納品書]テーブルのフィールドと配列をバインドしています。配列はフォーム上にスクロールエリアとして配置しているものです。これは納品書の一覧表を提示する簡単な方法です。

```
C_LONGINT ($BindID;$ErrCode;$ConnectID)
ARRAY DATE (aInvDate;0)
ARRAY REAL (aInvTotal;0)
ARRAY STRING (30;aCity;0)
ARRAY STRING (30;aCustID;0)
ARRAY STRING (30;aCompany;0)
C_LONGINT (vTableInvoices;vTableCustomers;$unused1)
C_LONGINT (vFieldDate;vFieldAmount;vFieldCustID;$unused2)
C_LONGINT (vFieldCompany;vFieldCity)

`[納品書]テーブルの番号と[納品書]納品書日付フィールドの番号を得る。
$ErrCode:= OP Get one field number (vConnectID;"[納品書]納品書日付";
                                       vTableInvoices;vFieldDate)

`[納品書]合計額フィールドの番号を得る。
$ErrCode:= OP Get one field number (vConnectID;"[納品書]合計額";$unused1;
                                       vFieldAmount)

`[納品書]顧客番号フィールドの番号を得る。
$ErrCode:= OP Get one field number (vConnectID;"[納品書]顧客番号";$unused1;
                                       vFieldCustID)

`[顧客]テーブルの番号と[顧客]社名フィールドの番号を得る。
$ErrCode:= OP Get one field number (vConnectID;"[顧客]社名";vTableCustomers;
                                       vFieldCompany)

`[顧客]都市名フィールドの番号を得る。
$ErrCode:= OP Get one field number (vConnectID;"[顧客]社名";$unused1;
                                       vFieldCity)

`サーバへ接続し、納品書を検索し、セレクションを作る。
$ErrCode:= OP Create bind ($BindID)
$ErrCode:= $ErrCode + OP Define bind by numbers ($BindID;vTableInvoices;
                                       vFieldCustID;0;0;"aCustID")
$ErrCode:= $ErrCode + OP Define bind by numbers ($BindID;vTableInvoices;
                                       vFieldCustID;0;0;"aInvDate")
$ErrCode:= $ErrCode + OP Define bind by numbers ($BindID;vTableInvoices;
```

```
vFieldAmount;0;0;"aInvTotal")
$ErrCode:= $ErrCode + OP Define bind by numbers ($BindID;vTableCustomers;
vFieldCity;0;0;"aCity")
$ErrCode:= $ErrCode + OP Define bind by numbers ($BindID;vTableCustomers;
vFieldCompany;0;0;"aCompany")
`もし、サーバ側ストラクチャが不変であると分かっているれば、
`以下のようにハードコーディングしても構いません。
`$ErrCode:= $ErrCode+ OP Define bind by numbers ($BindID;1;2;0;0;"aCompany")
If ($ErrCode=0)
    $ErrCode:= OP Selection to array ($ConnectID;$BindID)
End if
$ErrCode:= OP Delete bind ($BindID)
```

参照  
なし

## OP Define bind by pointer

---

**OP Define bind by pointer** (bindID; tableID; fieldID; boundObject) Error

引数	タイプ	説明
bindID	倍長整数	定義を追加するバインドのID
tableID	倍長整数	サーバ側テーブル番号
fieldID	倍長整数	サーバ側フィールド番号
boundObject	ポインタ	ローカル側オブジェクトを指すポインタ
Error	倍長整数	エラーコード

### 説明

**OP Define bind by pointer** は **OP Define bind by numbers** とよく似たルーチンですが、**OP Define bind by pointer** ではローカルオブジェクトを指定するのにポインタを使用します。このシンタクスはより柔軟であるため、前もってバインドするローカルオブジェクトが分からない場合には便利です。

**OP Define bind by pointer** は、引数 < bindID > で示される既存バインドへ新たなバインド定義を追加します。バインドするサーバ側オブジェクトの指定には、テーブル/フィールド番号を使用します。

引数 < boundObject > に、はローカル側のフィールド、変数、配列へのポインタを渡します。例えば、->[テーブル]フィールド、->Var、-><>Var、->Array1、-><>Array1等を渡せます。インタープロセス/プロセス変数、配列を使用できますが、ローカル変数/配列は使用できません。

### エラーコード

**OP Define bind by pointer** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10137	バインドが存在しません。
10140	ローカル側フィールドが存在しません。
10141	ローカル側テーブルが存在しません。
10142	この型のフィールドはバインドできません。
10144	ポインタの参照先がありません (NIL ポインタ)。
10145	ポインタを指定すべき個所がポインタではありません。
10154	このコマンドを今すぐには実行できません。
10157	同じオブジェクトを2重定義しています。

## 例題

この例題では**OP Define bind by numbers**の例題と同じことを、変数名の代わりにポインタでローカルオブジェクトを指定して、行っています。

```

C_LONGINT ($BindID;$ErrCode;$ConnectID)
ARRAY DATE (aInvDate;0)
ARRAY REAL (aInvTotal;0)
ARRAY STRING (30;aCity;0)
ARRAY STRING (30;aCustID;0)
ARRAY STRING (30;aCompany;0)
C_LONGINT (vTableInvoices;vTableCustomers;$unused1)
C_LONGINT (vFieldDate;vFieldAmount;vFieldCustID;$unused2)
C_LONGINT (vFieldCompany;vFieldCity)
    ` [納品書]のテーブル番号と[納品書]納品書日付のフィールド番号を得る。
    $ErrCode:= OP one field number (vConnectID;"[納品書]納品書日付";vTableInvoices;
                                                vFieldDate)

    ` [納品書]合計金額のフィールド番号を得る。
    $ErrCode:= OP one field number (vConnectID;"[納品書]合計金額";$unused1;
                                                vFieldAmount)

    ` [納品書]顧客番号のフィールド番号を得る。
    $ErrCode:= OP one field number (vConnectID;"[納品書]顧客番号";$unused1;
                                                vFieldCustID)

    ` [顧客]のテーブル番号と[顧客]社名のフィールド番号を得る。
    $ErrCode:= OP one field number (vConnectID;"[顧客]社名";vTableCustomers;
                                                vFieldCompany)

    ` [顧客]所在都市名のフィールド番号を得る。
    $ErrCode:= OP one field number (vConnectID;"[顧客]社名";$unused1;vFieldCity)
    ` サーバへ接続し、納品書を検索し、セレクションを作る。
    $ErrCode:= OP Create bind ($BindID)
    $ErrCode:= $ErrCode + OP Define bind by pointer ($BindID;vTableInvoices;
                                                vFieldCustID;->aCustID)
    $ErrCode:= $ErrCode + OP Define bind by pointer ($BindID;vTableInvoices;
                                                vFieldCustID;->aInvDate)
    $ErrCode:= $ErrCode + OP Define bind by pointer ($BindID;vTableInvoices;
                                                vFieldAmount;->aInvTotal)
    $ErrCode:= $ErrCode + OP Define bind by pointer ($BindID;vTableCustomers;
                                                vFieldCity;->aCity)
    $ErrCode:= $ErrCode + OP Define bind by pointer ($BindID;vTableCustomers;
                                                vFieldCompany;->aCompany)
    ` もしサーバ側ストラクチャが不変であると分かていれば、
    ` 以下のようにハードコーディングしても構いません。
    ` $ErrCode:= $ErrCode + OP Define bind by pointer ($BindID;1;2;0;0;->aCompany)
If ($ErrCode=0)
    $ErrCode:= OP Selection to array ($ConnectID;$BindID)
End if
$ErrCode:= OP Delete bind ($BindID)]
    
```

## 参照

なし

## OP Set format

---

### OP Set format (bindID; tableID; fieldID; conversionFormat) Error

引数	タイプ	説明
bindID	倍長整数	データ型/バインドのID
tableID	倍長整数	サーバ側テーブル番号
fieldID	倍長整数	サーバ側フィールド番号
conversionFormat	固定長文字列	変換に使うフォーマット文字列
Error	倍長整数	エラーコード

### 説明

**OP Set format** は、文字データとそれ以外の型のデータを変換する場合に使用するフォーマット文字列を指定します。

変換は双方向で可能です。フォーマットをクリアするためには、引数 < conversionFormat > へ空文字を渡します。

このルーチンは2種類の構文を許容しています。

- ・ 引数 < bindID > に有効なバインドIDを渡します。
- ・ 引数 < bindID > にデータ型を意味する負の値を渡します。

引数 < bindID > に有効なバインドIDを渡した場合は、引数 < tableID >、< fieldID > へも適切な値を渡して、フォーマット設定の対象となるサーバ側フィールドを特定してください。

フォーマットが有効になるのは、引数 < bindID > で指定したバインドを使用してデータを取得/送出した場合だけです。

引数 < bindID > に負の数を渡し、データ型を指定した場合は、引数 < tableID >、< fieldID > に渡した値は無視されます(双方へ「0」を渡しておいても構いません)。

フォーマットは、指定したデータ型のすべてのフィールドとの変換処理に適用されます。ただし、上記の方法で特定のフィールドにフォーマットを設定していた場合は、そちらの方が優先されます。

以下の表では、データ型番号とそれに対応するフォーマットの例を示しています。

データ型	データ型番号	フォーマット例
実数	-1	###.00
倍長整数	-1	¥###,###,###.-
整数	-1	00000
日付	-4	Char(3) ( 3番の日付フォーマットを指定 )
ブール値	-6	男;女
時間	-11	Char(4) ( 4番の時間フォーマットを指定 )

フォーマットに使用できる文字列についての詳細は、『4th Dimension ランゲージリファレンス』の「**SET FORMAT**」コマンドの説明を参照してください。

## エラーコード

**OP Set format**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10137	バインドが存在しません。
10148	未定義の機能オプションを指示しました。
10154	このコマンドを今すぐには実行できません。
10163	空のバインドです。

## 例題

この例題では、サーバ側日付フィールドを固定長文字列配列へバインドしています。フォーマット文字列を指定することによって、どのように変換するかを示しています。

```
C_LONGINT ($BindID;$ErrCode;$ConnectID)
ARRAY STRING (35;alnvDate;0)
C_LONGINT (vTableInvoices)
C_LONGINT (vFieldDate)
```

```
`[納品書]テーブル番号と[納品書]発行日付フィールド ID を得る。
$ErrCode:= OP Get one field number (vConnectID;"[納品書]発行日付";
vTableInvoices;vFieldDate)
```

```
$ErrCode:= OP Create bind ($BindID)
$ErrCode:= $ErrCode + OP Define bind by pointer ($BindID;vTableInvoices;
vFieldDate;->alnvDate)
```

`もしサーバ側ストラクチャが不変であると分かっているならば、

`以下のようにハードコーディングしても構いません。

```
`$ErrCode:= $ErrCode + OP Define bind by pointer ($BindID;2;3;0;0;->aCompany)
```

`このバインドでだけ適用する変換フォーマットを指定する。

`納品書発行日付を日付フォーマット7番（定数ではMM DD YYYY Forced）の  
`書式で表される文字列にする。  
\$ErrCode:= **OP Set format** (vConnectID;\$BindID;vTableInvoices;vFieldDate;  
**Char** (MM DD YYYY Forced))

**If** (\$ErrCode=0)  
    \$ErrCode:= **OP Selection to array** (vConnectID;\$BindID)  
**End if**

\$ErrCode:= **OP Delete bind** (\$BindID)

#### 参照

OP Array to selection、OP Define bind by numbers、OP Define bind by pointer、OP  
Selection to array、OP Set format



## レコード、序論

レコードを変更する、新しいレコードを作成する、カレントレコードをロードする場合は、データを表示する / 変更するためのローカル側フィールド、変数とサーバ側フィールドのバインドを作成し、サーバ側オブジェクトとローカル側オブジェクトの対応関係を確立しておきます。

下図では、バインドの作成から破棄までを含めて、データベース操作との処理手順を示しています。

### 1. バインドの作成

OP Create bind

### 2. サーバ側フィールドとローカル側フィールド、変数、配列との対応関係を定義

OP Define bind by numbers  
且つ / または  
OP Define bind by pointer

### 3. サーバ側の変更対象レコードをカレントレコードにする メモ：場合によってはこのステップをバインド作成前に実行しても良いでしょう。

OP Load record

OP Goto selected record  
または  
OP Goto record

### 4. レコードの追加、カレントレコードの修正、カレントレコードのデータの取得

OP New record

OP Update record

OP Load record  
(オプション)

### 5. レコードの使用が終わればUnload

OP Unload record

### 6. この後で再び使用する機会がなければバインドを消去

OP Delete bind

以下のルーチンを使用してレコードを操作します：

**OP Set access mode**：テーブルごとにアクセスモード（read/write、read-only）を設定します。

**OP Goto selected record**：カレントレコードの中の特定のレコードへ移動し、そのレコードをカレントレコードにします。オプション機能として同時にフィールドデータの取得も可能です。

**OP Load record**：OP Goto selected record、OP Goto record、検索コマンドでカレントレコードを設定した後、このコマンドを使えば、使用するバインド定義にしたがってフィールドデータを取得できます。

**OP Unload record**：カレントレコードをアンロードします。

**OP Update record**：カレントレコードの特定のフィールド（あらかじめバインドの中で指定しておく）の内容を修正する時にこのルーチンを呼びます。

**OP New record**：新しいレコードを作成するルーチンです。新規レコード作成と同時に各フィールドに希望の値を入れておくこともできます。

**OP Sequence number**：シーケンス番号を取得します。

**OP Delete record**：カレントレコードを削除します。

**OP Goto record**：カレントレコードをレコード番号で指定します。カレントセクションはそのレコード1件だけになります。

**OP Get record numbers**：カレントセクションのすべてのレコードのレコード番号を取得します。

**OP Current record number**：カレントレコードのレコード番号を取得します。

## 参照

OP Delete record、OP Get record numbers、OP Goto record、OP Goto selected record、OP Load record、OP New record、OP Sequence number、OP Set access mode、OP Unload record、OP Update record

## OP Set access mode

---

**OP Set access mode** (connectionID; tableID; accessMode) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	テーブル番号 (0 = 全テーブル)
accessMode	倍長整数	アクセスモード (0 = Read-only、それ以外 = read-write)
Error	倍長整数	エラーコード

### 説明

**OP Set access mode** は、引数 < connectionID > で指定される接続において、引数 < tableID > で指定されたテーブルのアクセスモード (Read-only、Read-write) を設定します。

引数 < tableID > に「0」を渡すと、すべてのテーブルに対してアクセスモードを設定します。

Read-only モードでは、レコードは常にロックされているため、レコード内容の変更とレコードの削除はできません。ただし新規レコードの作成は可能です。

接続開始直後は、すべてのテーブルのアクセスモードが read-write になっています。

### Read-only モード

テーブルが Read-only モードにセットされている間は、レコードの表示、印刷は可能ですが、内容の変更はできません。レコードを単に表示したいだけならば、Read-only モードにすることをお勧めします。そうすることで、他のユーザからの Read-write モードでのアクセスを保證できます。

注意すべきことは、Read-only モードは既存レコードにおいてのみ適用されるという点です。新規レコードの作成には影響しません。Read-only モードのテーブルへもレコードの追加は可能です。

### Read-write モード

レコード内容を修正しようとするならば、テーブルを Read-write モードにセットする必要があります。接続を開いた時点では、デフォルトですべてのテーブルは Read-write モードにセットされています。

Read-write モードであっても常にレコード修正ができるわけではありません。先に他のユーザやプロセスにカレントレコードをロックされている場合は、表示はできても、修正はできません。

レコードがロックされているかどうかは、**OP Goto selected record**、**OP Goto record**、**OP Load record**のいずれかのルーチンを実行して、引数 < LockStatus > を調べれば分かります。

## エラーコード

**OP Set access mode**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

## 例題

この例題では、顧客レコードのセレクションを作り、印刷しています。レコード内容の修正は行わないので、他のユーザ/プロセスのアクセシビリティを確保するために、[顧客]テーブルのアクセスモードを Read-only にセットしています。

```
C_LONGINT (vTableID)
C_LONGINT (vFieldCompany;vCompValue)
C_LONGINT ($errCode;vRecords)

`[顧客]テーブルの番号と[顧客]社名フィールドの番号を得る。
$errCode:= OP Get one field number (vConnectID;"[顧客]社名";vTableID;
                                         vFieldCompany)

`"Ab"で始まる社名の会社を検索。
vCompValue:= "Ab@"
$errCode:= OP Single query (vConnectID;vTableID;vFieldCompany;"=";
                               ->vCompValue;vRecords)

`社名でソート。
$errCode:= OP Single order by (vConnectID;vTableID;vFieldCompany;">")

`アクセスモードを Read-only にセット。アクセスモードの設定は
`レコードをロードする前に実行する。
$errCode:= OP Set access mode (vConnectID;vTableID;0)

PrintCustomerRecords
```

## 参照

OP Load record、OP Unload record、READ ONLY、READ WRITE

## OP Goto selected record

**OP Goto selected record** (connectionID; tableID; recordNumber{; bindID{; lockStatus{}}) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	テーブル番号
recordNumber	倍長整数	カレントセクションにおける移動先を示すレコード位置番号
bindID	倍長整数	バインドID
lockStatus	倍長整数	レコードのロック状況 (1ならロックされています)
Error	倍長整数	エラーコード

### 説明

**OP Goto selected record** は、引数 < tableID > で示されたテーブルのカレントレコードを変更します。カレントセクションのレコードの中で、その時点のレコード順序で引数 < recordNumber > で示された位置のレコードをカレントレコードにします。カレントセクション自体は変更しません。

引数 < bindID > に「0」を渡す、もしくは、引数 < bindID >、< lockStatus > の2つを省略した場合、**OP Goto selected record** はカレントレコードを変更するだけで、フィールド内容のダウンロードは行いません。有効なバインドIDを渡した場合は、フィールド内容をダウンロードします。この場合、もしアクセスモードがread-writeなら、レコードを修正可能な状態で読み込んでロックしようと試みます。先に他のユーザ/プロセスにロックされていないければ、引数 < lockStatus > に「0」が返り、無事レコードを修正可能な状態でロードできたことが分かります。「1」が返って来た場合は、そのレコードは既にロックされています。

このように、引数 < lockStatus > をチェックすることで、その後に続けて、レコードの更新、削除の処理をできるか否かを判断できます。

引数 < lockStatus > の値 = 0 でこのルーチンの実行を完了した (=レコードをロックした) 後、レコードを操作する処理を終了するのなら、他のユーザ/プロセスの為に、レコードをアンロードすることをお勧めします (**OP Unload record** を使用します)。

## エラーコード

**OP Goto selected record**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9968	レコード位置番号が範囲外です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

## 例題

この例題では、他の例題で使われているメソッド "PrintCustomerRecords" の実装を示しています。このメソッドは、[顧客] テーブルのカレントセレクションの全レコードを印刷しません。

```
C_LONGINT (vTable;$unused1)
C_LONGINT (vFieldID;vFieldCompany;vFieldAddress;vFieldZip;vFieldCity;
           vFieldState)

C_STRING (10;vCustID)
C_STRING (30;vCustName;vCustCity)
C_STRING (5;vCustZip)
C_STRING (2;vCustState)
C_TEXT (vCustAddress)
C_LONGINT (vRecords;lockStatus)

`[顧客]のテーブル番号と[顧客]番号のフィールド ID を得る。
$errCode:= OP one field number (vConnID;"[顧客]番号";vTable;vFieldID)
`[顧客]社名のフィールド ID を得る。
$errCode:= OP one field number (vConnID;"[顧客]社名";$unused1;vFieldCompany)
`[顧客]住所のフィールド ID を得る。
$errCode:= OP one field number (vConnID;"[顧客]住所";$unused1;vFieldAddress)
`[顧客]郵便番号のフィールド ID を得る。
$errCode:= OP one field number (vConnID;"[顧客]郵便番号";$unused1;vFieldZip)
`[顧客]都市名のフィールド ID を得る。
$errCode:= OP one field number (vConnID;"[顧客]都市名";$unused1;vFieldCity)
`[顧客]都道府県のフィールド ID を得る。
$errCode:= OP one field number (vConnID;"[顧客]都道府県";$unused1;
           vFieldPrefecture)

`バインド定義
$errCode:= OP Create bind ($BindID)
$errCode:= $errCode + OP Define bind by pointer ($BindID;vTable;vFieldID;
           ->vCustID)

$errCode:= $errCode + OP Define bind by pointer ($BindID;vTable;
           vFieldCompany;->vCustName)
$errCode:= $errCode + OP Define bind by pointer ($BindID;vTable;vFieldAddress;
           ->vCustAddress)
$errCode:= $errCode + OP Define bind by pointer ($BindID;vTable;vFieldZip;
```

```
                                ->vCustZip)
$ErrCode:= $ErrCode + OP Define bind by pointer ($BindID;vTable;vFieldCity;
                                ->vCustCity)
$ErrCode:= $ErrCode + OP Define bind by pointer ($BindID;vTable;
                                vFieldPrefecture;->vCustPrefecture)
```

　「顧客情報」フォームにはバインドに定義された変数が配置されている。

```
OUTPUT FORM ([Dialogs];"顧客情報")
```

```
$ i:= 1
```

　[顧客]テーブルに含まれるレコードの数を調べている。

```
$ErrCode:= OP Records in selection (vConnID;vTable;vRecords)
```

```
While ($i <= vRecords)
```

```
    $ErrCode:= OP Goto selected record (vConnID;vTable;$i;$BindID;LockStatus)
```

　ロードした値を印刷する。

```
    PRINT RECORD ([Dialogs])
```

```
    $i := $i +1
```

```
End while
```

```
$ErrCode:= OP Delete bind ($BindID)
```

参照

OP record numbers、OP Goto record

## OP Load record

---

### OP Load record (connectionID; bindID; tableID; lockStatus) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
bindID	倍長整数	バインドID
tableID	倍長整数	テーブル番号
lockStatus	倍長整数	レコードのロック状況 (1ならロックされています)
Error	倍長整数	エラーコード

### 説明

**OP Load record** は、カレントレコードをロードし、同時に引数 < bindID > で指定したバインドに含まれるすべてのテーブルのカレントレコードをロックしようとします。ロードするのはバインドの中で定義されたフィールドのデータだけです。

例えば、テーブルの中の3つのフィールドだけをバインドに定義したとします。ネットワークを超えて送られてくるデータは、レコード内容全体ではなく、バインドに定義した3つのフィールド内容だけです。

引数 < lockStatus > に「0」が返ってくれば、Read-write モードでレコードをロードしたことを示しており、レコード内容の変更とレコード自体の削除を行うことができます。Read-write モードでレコードを処理した後は必ず、**OP Unload Record** してください。これは、他のユーザ / プロセスに Read-write モードでのレコードアクセスを可能にするために必要な処理です。

Read-only モードでレコードをロードした場合、または **OP Load record** 実行時点で他のユーザ / プロセスがカレントレコードを Read-write モードで使用していた場合、引数 < lockStatus > には「1」が返ってきます。処理の都合上、read-write モードでアクセスする必要がある場合は、引数 < lockStatus > に値「0」が返ってくるまで (= 他のユーザ / プロセスがカレントレコードをアンロードするまで) 繰り返し **OP Load record** を呼び出すこととなります。

アクセスモードがいずれであれ、指定したフィールドデータの取得は行われます。

カレントレコードが存在しない場合は **OP Load record** は何も行いません。



## エラーコード

**OP Load record**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
-9969	不正なフィールド型が要求されました。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
4004	ローカル側のテーブルにカレントレコードがありません。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10137	バインドが存在しません。
10139	バインドが未定義です。
10153	型変換の不可能な引数データがあります。
10154	このコマンドを今すぐには実行できません。

## 例題

メソッド"LoadAndLock"はレコードを更新目的でロードします。

```

`メソッド LoadAndLock (Conn,TableID;BindID;MaxWait) -> LockStatus
`Conn,接続 ID
`TableID,カレントレコードをロードしたいサーバ側テーブルの番号
`BindID,値のロードに使用するバインドの ID
`MaxWait,タイムアウト扱いにするまでのロード試行回数

C_LONGINT (Conn,TableID,BindID,MaxWait)
C_LONGINT ($errCode;Lock;$0;$i)

$Conn:= $1
$TableID:= $2
$BindID:= $3
$MaxWait:= $4
$i := 0
$Lock:=0

`read-write モードにします。この後のレコードロード処理から有効。
$errCode:= OP Set access mode (Conn;TableID;1)

Repeat
  $errCode:= OP Load record ($Conn;$TableID;$BindID;$Lock)
  If ( ($i<=$maxWait) & ($lock=1) )
    `既にレコードがロックされていたら、すぐに続けてロードし直さず、
    `しばらく待つ。
    DELAY PROCESS (Current process;15)           `1/4 秒待つ。
  End if

```

```
$ i:= $i + 1  
Until ( ($i >=$maxWait) | ($lock = 0) )
```

```
  `ロックステータスを返す。  
$0 := $lock
```

参照

OP Goto selected record、OP Set access mode、OP Unload record

## OP Unload record

---

**OP Unload record** (connectionID; tableID) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
tableID	倍長整数	テーブル番号
Error	倍長整数	エラーコード

### 説明

**OP Unload record** は、引数 < tableID > で示されるテーブルのカレントレコードをアンロードします。

**OP Load record**、**OP Goto selected record**、**OP Goto record** によってロックされていたレコードを開放することになります。

### エラーコード

**OP Unload record** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 参照

OP Load record、OP Set access mode、OP Update record、UNLOAD RECORD

## OP Update record

---

### OP Update record ( connectionID; bindID ) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続のID
bindID	倍長整数	使用するバインドのID
Error	倍長整数	エラーコード

#### 説明

**OP Update record**は、カレントレコードに対して行った変更を保存します。変更されるのは、引数 < bindID > で指定されたバインドに含まれているフィールドのみです。それらのフィールドは単一のテーブルに属するフィールドでなければなりません。

**OP Update record**を実行する前に、あらかじめ、**OP Goto selected record**か**OP Goto record**を使用してカレントレコードを指定してください。その後のレコードロードや既存のフィールド値の取得には**OP Load record**を使用することもできます。その後、変更を使用する新しい値を割り振って、**OP Update record**を呼び出します。これがレコード更新のおおまかな手順です。

**OP Goto selected record**、**OP Goto record**、**OP Load record**を実行する場合、引数 < lockStatus > を忘れずにチェックしてください。ロックされていたら**OP Update record**を実行してもレコードは更新できません。その場合はエラーコードが返ってきます。

レコードのロックステータスを調べる場合に使用するバインドと、レコード更新に使うバインドは同じである必要はありません。例えば、ロックステータスを調べる場合には1フィールドだけのバインドを使用し、レコード更新には10フィールドを含むバインドを使用する、といったことも可能です。

#### エラーコード

**OP Update record**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
-9967	レコードがロードされていなかったので変更内容は保存されませんでした。
-9969	不正なフィールド型が要求されました。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。

-9998	重複不可属性の設定されたフィールドに重複した値を代入しようとしました。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10137	バインドが存在しません。
10138	バインドにあるフィールドがターゲットテーブルにリレートしていません。
10139	バインドが未定義です。
10153	型変換の不可能な引数データがあります。
10154	このコマンドを今すぐには実行できません。

### 例題

以下の例題ではその日の納品書を反映させて顧客レコードの総売上フィールドを更新しています。

```
C_LONGINT ($BindID_ 納品書;$BindID_ 顧客;$ErrCode)
C_LONGINT ($lock;vRecords)
```

```
C_STRING (8;v 発行日付)
C_REAL (v 顧客別総売上)
C_STRING (30;vCustID)
```

```
C_LONGINT (vTNo 納品書;vTNo 顧客;$unused1)
C_LONGINT (vFNo 合計金額;vFNo 納品先顧客番号;vFNo 発行日付;$unused2)
C_LONGINT (vFNo 顧客番号;vFNo 顧客総売上)
```

```
ARRAY STRING (30;a 顧客番号;0)
ARRAY STRING (30;a 納品先顧客番号;0)
ARRAY REAL (a 顧客総売上;0)
ARRAY REAL (a 納品書金額;0)
```

```
`[納品書]のテーブル番号と[納品書]顧客番号 fieldID を得る。
$errCode:= OP Get one field number (vConnectID;"[納品書]顧客番号";vTNo 納品書;
vFNo 納品先顧客番号)
```

```
`[納品書]発行日付のフィールド番号を得る。
$errCode:= OP Get one field number (vConnectID;"[納品書]発行日付";$unused1;
vFNo 発行日付)
```

```
`[納品書]合計金額 のフィールド番号を得る。
$errCode:= OP Get one field number (vConnectID;"[納品書]合計金額";$unused1;
vFNo 合計金額)
```

```
`[顧客]のテーブル番号と[顧客]顧客番号のフィールド番号を得る。
$errCode:= OP Get one field number (vConnectID;"[納品書]顧客番号";vTNo 顧客;
vFNo 顧客番号)
```

```

`[顧客]総売上のフィールド番号を得る。
$errCode:= OP Get one field number (vConnectID;"[顧客]総売上";$unused1;
                                         vFNo 顧客総売上)

`[顧客]社名のフィールド番号を得る。
$errCode:= OP Get one field number (vConnectID;"[顧客]社名";$unused1;
                                         vFNo 社名)

`顧客レコードをロード、更新するためのバインドを作る。
$errCode:= OP Create bind ($BindID_顧客)
$errCode:= OP Define bind by pointer ($BindID_顧客;vTNo 顧客;vFNo 顧客総売上;
                                         ->v 顧客別総売上)
$errCode:= OP Define bind by pointer ($BindID_顧客;vTNo 顧客;vFNo 顧客番号;
                                         ->vCustID)

`[納品書]合計金額と[納品書]顧客番号をロードするためのバインドを作成。
$errCode:= OP Create bind ($BindID_納品書)
$errCode:= OP Define bind by pointer ($BindID_納品書;vTNo 納品書;
                                         vFNo 合計金額;->a 納品書金額)
$errCode:= OP Define bind by pointer ($BindID_納品書;vTNo 納品書;
                                         vFNo 納品先顧客番号;->a 納品先顧客番号)

`今日発行された納品書をカレントセレクションにする。
v 発行日付:= String (Current date)
$errCode:= OP Single query (vConnectID;vTNo 納品書;vFNo 発行日付;"=";
                                         ->v 発行日付;vRecords)

`納品書合計額と納品先顧客番号を配列へロード。
$errCode:= OP Selection to array (vConnectID;$BindID_納品書)
`[納品書]カレントセレクションからダブりのない納品先顧客番号の配列を作成。
$errCode:= OP Distinct values (vConnectID;vTNo 納品書;vFNo 納品先顧客番号;
                                         a 顧客番号)

`得られた顧客数と等しい要素を持つ顧客別総売上額配列を作る。
ARRAY REAL (a 顧客総売上;Size of array(a 顧客番号))

`すべての納品書を調べて顧客ごとに売上額を集計する。
For ($i;1; Size of array (a 納品先顧客番号) )
    $要素番号:= Find in array (a 顧客番号;a 納品先顧客番号{$i})
    a 顧客総売上{$要素番号}:=a 顧客総売上 {$要素番号}+a 納品書金額{$i}
End for

`[納品書]テーブルのセレクション（今日発行された納品書）から
`[顧客]テーブルのセレクションを作る。
$errCode:= OP Many to one join (vConnectID;vTNo 納品書;vTNo 顧客)

$errCode:= OP Records in selection (vConnectID;vTNo 顧客;vRecords)

`[顧客]テーブルを Read/write モードに設定。
$errCode:= OP Set access mode (vConnectID;vTNo 顧客;1)

```

```
`[顧客]テーブルのカレントセレクションを巡回。
For ($i;1; vRecords)
  $errCode:= OP Goto selected record (vConnectID;vTNo 顧客;$i;
                                         $BindID_顧客;$lock)
    `レコードを Read-write モードでロードできたと仮定する。

    `また同時にバインド $BindID_顧客 で定義した変数 vCustID にも
    `無事に値を取得できたと仮定する。
  $要素番号:= Find in array (a 顧客番号;vCustID)

  v 顧客別総売上:= a 顧客総売上{$要素番号}

  `変数「v 顧客別総売上」を含むバインド「$BindID_顧客」を使って
  `[顧客]レコードを更新。
  $errCode:= OP Update record (vConnectID;$BindID_顧客)
End for

`Read-write モードでロードしたレコードはロックされるが、
`他のレコードをした時点で自動的にアンロードされる。
`結果的に、最後にロードしたレコード以外はすべて開放されている。
$errCode:= OP Unload record (vConnectID;vTNo 顧客)
`最後にロードしたレコードを開放。

$errCode:= OP Delete bind ($BindID_納品書)
$errCode:= OP Delete bind ($BindID_顧客)
```

### 参照

OP Goto selected record、OP Load record、OP Set access mode、OP Unload record、  
SAVE RECORD

## OP New record

---

### OP New record (connectionID; bindID) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続のID
bindID	倍長整数	使用するバインドのID
Error	倍長整数	エラーコード

#### 説明

**OP New record** は、引数 < bindID > で示すバインドに定義されたレコードに、新しいレコードを追加します。したがって、このコマンドを実行する前には、バインドがあらかじめ作成、定義されている必要があります。

この時、バインドに定義したローカルオブジェクトに値を入れておけば、作成された新しいレコードのフィールドにはその値が設定された状態で新規レコードができ上がります。バインドに含まれないサーバ側フィールドはそれぞれの型に見合ったデフォルト値が設定されます。

作成された新規レコードは自動的にアンロードされます。

#### エラーコード

**OP New record** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
-9967	レコードがロードされていなかったので変更内容は保存されませんでした。
-9969	不正なフィールド型が要求されました。
-9971	存在しないフィールド番号です。
-9972	テーブル番号が範囲外です。
-9998	重複不可属性の設定されたフィールドに重複した値を代入しようとしました。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10137	バインドが存在しません。
10138	バインドにあるフィールドがターゲットテーブルにリレートしていません。
10139	バインドが未定義です。
10153	型変換の不可能な引数データがあります。
10154	このコマンドを今すぐには実行できません。



## 例題

この例題ではローカルデータベースの顧客レコードをサーバデータベースへコピーしています。いわゆるレプリケーション処理です。前提としてローカル側とサーバ側に同じフィールド構成の[顧客]テーブルがあるものとします。

```

C_LONGINT ($BindID_顧客)
C_LONGINT (ServerTableID;ServerFieldID)
C_LONGINT ($ErrCode;vRecords;$unused)
  `念のため、ローカル側テーブル名と番号を取得。
$CustTableName:= Table name (->[顧客])
$CustTable:= Table (->[顧客])
$fieldName:= $CustTableName + Field name ($CustTable;1)

  `サーバ側のテーブル番号を得る。
$ErrCode:= OP Get one field number (vConnectID;$fieldName;ServerTableID;
                                         $unused)

  `バインド作成。ローカル側のフィールドとバインドさせる。
$ErrCode:= OP Create bind ($BindID_顧客)
For ($i;1; Count fields($CustTable))
  "[Table]Field"の形式でローカル側のフィールド名文字列を作る。
  `サーバ側と同一の名称であることを前提にしている。
  $fieldName:=$CustTableName+ Field name ($CustTable;$i)
  `ローカルフィールドへのポインタを使ってバインド定義している。
  $ErrCode:= OP Define bind by pointer ($BindID_顧客;ServerTableID;$i;
                                         Field ($CustTable;$i))

End for
ALL RECORDS ([顧客])
$PcustID:= ->[顧客]顧客番号
  `ローカル側の全レコードを巡回。
While ( Not ( End selection ([顧客])))
  `サーバ側に同じレコードがあるかどうかを調べる。
  $ErrCode:= OP Single query (vConnectID;ServerTableID;1;"=";$PcustID;
                                         vRecords)

  If (vRecords = 0)
    `もしサーバ側に同じレコードがなければ、
    `ローカル側のカレントレコードのフィールド内容で
    `新規レコードを作る。
    $ErrCode:= OP New record (vConnectID;$BindID_顧客)
  Else
    `ここに書くコードでは、ローカル側のカレントレコード内容で
    `サーバ側レコードを更新する処理ができるでしょう。
  End if
NEXT RECORD ([顧客])
End while
$ErrCode:= OP Delete bind ($BindID_顧客)

```

## 参照

OP Load record、OP Set access mode、OP Update record、SAVE RECORD

## OP Sequence number

---

**OP Sequence number** (connectionID; tableID; sequenceNumber) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続のID
tableID	倍長整数	シーケンス番号を調べるテーブル番号
sequenceNumber	倍長整数	シーケンス番号
Error	倍長整数	エラーコード

### 説明

**OP Sequence number** は、引数 < tableID > で示されるテーブルの次のシーケンス番号を引数 < sequenceNumber > に返します。

### エラーコード

**OP Sequence number** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 参照

OP New record、OP Update record、Sequence number

## OP Delete record

---

### OP Delete record (connectionID; tableID) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続のID
tableID	倍長整数	レコードを削除するテーブル番号
Error	倍長整数	エラーコード

### 説明

**OP Delete record** は、引数 < tableID > で示されるテーブルのカレントレコードを削除します。

カレントレコードがない、もしくは、引数 < tableID > で示されるテーブルが Read-only モードの場合は、**OP Delete record** は何もしません。

カレントレコードがロックされている場合、レコードは削除されず、**OP Delete record** はエラーコードを返します。

あらかじめ **OP Load record**、**OP Goto record**、**OP Goto selected record** を実行した時点で、引数 < LockStatus > をチェックしておく必要があります。

### エラーコード

**OP Delete record** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9972	テーブル番号が範囲外です。
-9986	リレートされたレコードの自動削除中にロックされたレコードがありました。
-9987	このレコードにリレートしているレコードがあります。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 例題

この例題では、ローカル側に存在しないサーバ側の顧客レコードを削除しています。前提として、ローカル側とサーバ側に同じフィールド構成の[顧客]テーブルがあるものとします。

```
C_LONGINT (ServerTableID;ServerFieldID)
C_LONGINT ($errCode;vRecords;$unused;$lock)
```

```

C_LONGINT ($CustBind;$custLockBind)
C_STRING (10;vCustIDs_Server)
ARRAY STRING (10;aCustIDs_Local;0)
ARRAY STRING (10;aCustIDs_Server;0)
ARRAY LONGINT (aRecNum_Server;0)
    `ローカル側のテーブル名と番号を調べる。
$CustTableName:= Table name (->[顧客])
$CustTable:= Table (->[顧客])
$fieldName:= $CustTableName + Field name (->[顧客]顧客番号)
    `サーバ側のテーブル番号を調べる。
$errCode:= OP Get one field number (vConnectID;$fieldName;ServerTableID;
                                     ServerFieldID)
    `サーバ側顧客テーブルから顧客番号フィールドをロードするバインドを作る。
$errCode:= OP Create bind ($custBind)
$errCode:= OP Define bind by pointer ($custBind;ServerTableID;ServerFieldID;
                                     ->aCustIDs_Server)
    `顧客番号フィールドのダウンロードとレコードロックチェック専用の
    `バインドを作る。
$errCode:= OP Create bind ($custLockBind)
$errCode:= OP Define bind by pointer ($custLockBind;ServerTableID;
                                     ServerFieldID;->vCustIDs_Server)

```

#### **ALL RECORDS** ([顧客])

```

    `ローカル側の顧客番号を配列にコピー。
SELECTION TO ARRAY ([顧客]顧客番号;aCustIDs_Local)

$errCode:= OP All records (vConnectID;ServerTableID)
    `サーバ側の顧客番号を配列にダウンロード。
$errCode:= OP Selection to array (vConnectID;$custBind)
    `より高速なレコードアクセスのためにレコード番号も取得しておく。
$errCode:= OP Get record numbers (vConnectID;ServerTableID;aRecNum_Server)
$deleteCount:= 0
    `サーバ側の全顧客レコードを巡回。
For ($i;1; Size of array (aCustIDs_Server))
    `サーバ側顧客番号がローカル側にもあるかどうか調べる。
    $local:= Find in array (aCustIDs_Local;aCustIDs_Server{$i})
    If ($local = -1)
        `見つからなかった場合：サーバ側レコードを削除。
        `まず、サーバ側の該当レコードをカレントレコードにし、
        `顧客番号フィールドをロード、ロックする。
        $errCode:= OP Goto record (vConnectID;ServerTableID;
                                   aRecNum_Server{$i};$custLockBind;$lock)
        `念のためロードした顧客番号の値が前もって配列へ
        `ダウンロードしていた値と一致しているかどうかチェックする
        `(配列へロードした後、レコードが削除され、新規レコードが
        `他のユーザによって追加されていた場合には同じレコード番号でも
        `同一のレコードである保証はないから )
        If (vCustIDs_Server = aCustIDs_Local {$i})
            `レコードは他のユーザ/プロセスから
            `ロックされていないという前提で
            $errCode:= OP Delete record (vConnectID;ServerTableID)

```

```
        $deleteCount:= $deleteCount +1
    End if
End if
End for
ALERT ("サーバ側の" + String ($deleteCount) + "件のレコードを削除しました。")
$errCode:= OP Delete bind ($custBind)
$errCode:= OP Delete bind ($custLockBind)
```

#### 参照

DELETE RECORD、OP Delete selection、OP Single query、OP Update record

## OP Goto record

---

**OP Goto record** (connectionID; tableID; recordNumber; bindID; lockStatus)

Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続のID
tableID	倍長整数	カレントレコードにしたいレコードのテーブル番号
recordNumber	倍長整数	カレントレコードにしたいレコードのレコード番号
bindID	倍長整数	フィールド値ロードに使うバインドID
lockStatus	倍長整数	ロック状況。「1」だったらレコードはロックされている。
Error	倍長整数	エラーコード

説明

**OP Goto record** は、引数 < tableID >、< recordNumber > で示されるレコードをカレントレコードにし、カレントセクションの唯一のレコードにします。引数 < bindID > へ「0」を渡すか、引数 < bindID >、< lockStatus > を省略してしまえば、**OP Goto record** はカレントレコードとカレントセクションの変更だけを行います。引数 < bindID > へ有効なバインドIDを渡した場合は、そのバインドに従ってデータのダウンロードを行います。この場合、テーブルのアクセスモードがRead-writeに設定されていれば、Read-writeモードでのロードを試みます。

既に他のユーザ/プロセスがロックしていなければ、引数 < lockStatus > には「0」が返され、レコードの更新/削除を行うことができます。「1」が返された場合は、他のユーザ/プロセスによって既にレコードがロックされているため、レコードの更新/削除を行うことはできません。

このように、引数 < lockStatus > に返される値をチェックすることで、その後、**OP Update record** や **OP Delete record** を使用して、更新や削除作業を行えるか否かを判断することができます。

Read-write モードでレコードをロードした後は、そのレコードに用がなくなった時点で、必ず **OP Unload record** でレコードを開放するようにしてください。もちろん、削除してしまったレコードに対しては必要ありません。

## エラーコード

**OP Goto record**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9972	テーブル番号が範囲外です。
-10003	レコード番号が範囲外です。
-9987	このレコードにリレートしているレコードがあります。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

## 例題

**OP Delete record**の例題を参照してください。

## 参照

GOTO RECORD、OP Delete record、OP Current Record Number、OP Get record numbers、OP Goto selected record、OP Load record、OP Single query

## OP Get record numbers

---

### OP Get record numbers (connectionID; tableID; recordNumbers

{;firstRecordNumber {;lastRecordNumber}}) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続のID
tableID	倍長整数	レコード番号を調べたいテーブルのテーブル番号
recordNumbers	配列	レコード番号
firstRecordNumber	倍長整数	対象範囲の最初のレコードの位置番号
lastRecordNumber	倍長整数	対象範囲の最後のレコードの位置番号
Error	倍長整数	エラーコード

### 説明

**OP Get record numbers** は、引数 < tableID > で示されたテーブルのカレントセクションに含まれるすべてのレコードのレコード番号を引数 < recordNumbers > の配列 (倍長整数もしくは実数型) へ返します。

引数 < firstRecordNumber > と < lastRecordNumber > に数値を指定することで、対象範囲をカレントセクションの一部に制限することも可能です。引数 < firstRecordNumber > に「0」を渡すと範囲制限は無効になり、カレントセクション全体のレコードを対象とします。

引数 < firstRecordNumber >、< lastRecordNumber > に渡す値はレコード位置番号ですが、引数 < recordNumbers > の配列へ返される値はレコード番号であることに注意が必要です。

### エラーコード

**OP Get record numbers** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10135	引数型が不正です。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。



**例題**

**OP Delete record**の例題を参照してください。

**参照**

OP Delete record、OP Current Record Number、OP Selection to array、OP Single query、  
SELECTION TO ARRAY

## OP Current Record Number

---

**OP Current Record Number** (connectionID; tableID; recordNumber; recordPosition) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続のID
tableID	倍長整数	テーブル番号
recordNumber	倍長整数	カレントレコードのレコード番号
recordPosition	倍長整数	カレントレコードのレコード位置番号
Error	倍長整数	エラーコード

### 説明

**OP Current Record Number** は、引数 < tableID > で示されたテーブルのカレントレコードについて、引数 < recordNumber > にレコード番号を、引数 < recordPosition > にカレントセクション内でのレコード位置番号を返します。

### エラーコード

**OP Current Record Number** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-9956	接続先の 4D Server とバージョン間の互換性がありません。バージョン 1.1 の 4D Server へ接続しようとした場合は、データベースプロパティの "4D OPEN の接続を許可する" チェックボックスが OFF です。
-32001	非同期モードのクエリが既に実行中です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 参照

OP Get record numbers

## ユーティリティ、序論

---

この章では以下のルーチンについて説明します：

**OP Get error text** : 4D Open for 4D ルーチンから返されるエラーの説明テキストを取得します。エラー番号を渡せば、対応するエラーメッセージが返ってきます。

**OP Set option** : 4D Open for 4D のオプション機能の設定を行います。

**OP Get option** : OP Set option で設定された値を得ます。

**OP Flush Buffers** : バッファのデータを直ちにディスクへ書き込みます。

**OP Get version number** : 4D Open for 4D のバージョン番号を取得します。

### 参照

OP Get error text、OP Get option、OP Set option

## OP Get error text

---

### OP Get error text (errorCode) Error

引数	タイプ	説明
errorCode	倍長整数	エラーコード
Error	固定長文字列	エラーメッセージ

### 説明

**OP Get error text** は、引数 < tableID > で示されたテーブルのカレントセクションに含まれるすべてのレコードのレコード番号を引数 < recordNumbers > の配列（倍長整数もしくは実数型）へ返します。引数 < firstRecordNumber > と < lastRecordNumber > に数値を指定することで、対象範囲をカレントセクションの一部に制限することも可能です。引数 < firstRecordNumber > に「0」を渡すと範囲制限は無効になり、カレントセクション全体のレコードを対象とします。

注意すべき点は、引数 < firstRecordNumber >、< lastRecordNumber > に渡す値はレコード位置番号であることです。

### エラーコード

**OP Get error text** は、< errorCode > の説明テキストを返します。

エラーコード	説明
-9972	テーブル番号が範囲外です。
-10003	レコード番号が範囲外です。
-9987	このレコードにリレートしているレコードがあります。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 例題

この例題では、エラーが起きた時にメソッド "WhichError" を呼び出しています。

```
$ErrCode:= OP All records (ConnectID;vFile)
If ($ErrCode=0)
    `処理を続行。
Else
    WhichError ($ErrCode;"OP All records")
    `エラーが起きた時の処理。
End if
```

以下の"WhichError"メソッドでは2つの引数を受け取ります。1つはエラーコード、残りの1つはルーチン名です。エラーコードは**OP Get error text**に引き渡され、エラーテキストを得ています。ルーチン名はエラーを起こした4D OPENルーチン名を表示するエラーメッセージを生成するために使用します。

```
C_LONGINT ($1)
```

```
ALERT (OP Get error text ($1)+Char(Carriage return)+"エラー発生ルーチン : "  
+ $2+".")
```

参照

OP Set option

## OP Set option

---

### OP Set option (optionNumber; optionValue) Error

引数	タイプ	説明
optionNumber	倍長整数	オプション番号
optionValue	倍長整数	設定値
Error	倍長整数	エラーコード

### 説明

**OP Set option** は 4D OPEN の運用上のオプション機能の設定を行います。引数 < optionNumber > と引数 < optionValue > へ渡せる値は以下のようになります：

オプション番号	設定値	説明
1	1、0	引数チェックモード ON / OFF。デフォルトは ON。
2	1、0	4D OPEN エラーダイアログ ON / OFF。 デフォルトは ON
3	1、0	サーバ側ストラクチャ情報のキャッシュ ON / OFF。
4	n	レコード関連ルーチンで配列が使われた時、特別な用途に使用される要素番号。デフォルトの要素番号は 0。
(NCID×65536) + コード	1、0	ネットワークコンポーネントオプションを設定。コードについては後述。

#### オプション番号 1：引数チェックモード

引数チェックモードを ON にした場合は、4D Open for 4D は引数の妥当性をチェックします。例えば、テーブル番号引数の場合、4D OPEN はサーバ側データベースに該当するテーブルがあるかどうかをチェックする要求を送信します。この機能はデフォルトで ON になっていますが、処理速度を低下させる側面もあるため、**OP Set option** (1;0) とすることで OFF にすることもできます。

#### オプション番号 2：4D OPEN エラーダイアログ

4D OPEN ルーチン呼び出しでエラーが起きた時に、エラーダイアログを表示します。デフォルトは ON です。

#### オプション番号 3：サーバ側ストラクチャ情報のキャッシュ機能

サーバ側ストラクチャ情報のキャッシュ機能を使用するかどうかの設定です。デフォルトは OFF です。**OP Set option** (3;1) としてこの機能を ON にすれば、**OP Cache structure** ルーチンの実行以降のストラクチャアクセスルーチンの実行結果はキャッシュされます。

オプション番号4：レコードルーチンのための要素

配列をバインドしてレコードルーチンを実行すると、番号0の要素はレコードルーチンが使用します。4番オプションを使ってこの要素番号を変更できます。デフォルトは「0」です。

ネットワークコンポーネントオプション

**OP Set option**は、いくつかのネットワークコンポーネントオプションの設定も行うことができます。この機能は上級者のためのもので、接続 / 通信に関するパラメータを変更できます。

4D OPENは与えられた設定値をチェックしないため、場合によっては、トラブルを引き起こすことにもなりかねません。機能を十分理解した上で使用してください。

オプション番号は以下のように計算します：

$$\text{optionValue} = (\text{NCID} \times 65536) + \text{Code}$$

NCID（ネットワークコンポーネントID）は**OP Get network component info**を使用して取得します。

Codeとして使用できる値は以下の通りです。

Code	名称	説明
1	Version	ネットワークコンポーネントのVersion（読み取り専用）。
2	LockDelay	4D Client接続の時にA D S Pネットワークコンポーネントによって使用されます。（読み取り専用）
3	IdleDelay	必ず0にする。
4	BufferSize	通信バッファサイズ（バイト単位）。
5	PackOption	データ送信時に自動圧縮機能を使うか否か。
6	MinPackSize	PackOptionが1の時（＝自動圧縮機能ON）、圧縮する最小データサイズ（バイト単位）。
7	MaxPackSize	PackOptionが1の時（＝自動圧縮機能ON）、圧縮する最大データサイズ（バイト単位）。
8	DelayVoyant	必ず0にする。
9	Product ID	このネットワークコンポーネントを使用中のアプリケーションタイプ。
10	Flags	32ビットのフラグ。ネットワークコンポーネントオプションが内部使用。

ネットワークコンポーネントオプションを設定できるのは、対象とするネットワークコンポーネントが使用されていない時だけです（**OP Load network component**する前、**OP Unload network component**した後）。

## エラーコード

**OP Set option**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-192	リソースが見つかりません。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10132	指定されたネットワークコンポーネントは使用中です。
10148	未定義の機能オプションを指示しました。
10154	このコマンドを今すぐには実行できません。
10165	未知のネットワークコンポーネントオプションです。

## 例題

```
If (<>vDebugLevel >= checkParams)
  `デバック工程中。
  `引数チェック機能を使用する。
  $errCode:= OP Set option (1;1)
Else
  `デバック工程も終わりにさしかかった。
  `引数チェック機能をOFFにしてパフォーマンスを向上。
  $errCode:= OP Set option (1;0)
  If (<>vDebugLevel= final)
    `デバック工程終了。
    `サーバストラクチャもファイナル版となった。
    `ストラクチャキャッシュ機能を使用する。
    $errCode:= OP Set option (3;1)
    $errCode:= OP Cache structure (vConnectID;1)
  Else
    $errCode:= OP Set option (3;0)
  End if
End if
```

## 参照

OP Get option



## OP Get option

---

**OP Get option** (optionNumber; optionValue) Error

引数	タイプ	説明
optionNumber	倍長整数	オプション番号
optionValue	倍長整数	既定値
Error	倍長整数	エラーコード

### 説明

**OP Get option** は4D OPENの運用上の各種オプション機能の現在値を調べます。

このコマンドはいつでも使えます。

引数 < optionNumber > に使用する番号と引数 < optionValue > に返ってくる値の意味は、**OP Set option** の記述を参照してください。

### エラーコード

**OP Get option** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-192	リソースが見つかりません。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10132	指定されたネットワークコンポーネントは使用中です。
10148	未定義の機能オプションを指示しました。
10154	このコマンドを今すぐには実行できません。
10165	未知のネットワークコンポーネントオプションです。

### 参照

OP Set option

## OP Flush Buffers

---

OP Flush Buffers (connectionID)		Error
引数	タイプ	説明
connectionID	倍長整数	使用する接続のID
Error	倍長整数	エラーコード

### 説明

**OP Flush Buffers** は、直ちにバッファのデータをディスクへ書き込みます。データベースに対して実行されたすべての変更がディスクに保存されます。

### エラーコード

**OP Flush Buffers** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-32001	非同期モードのクエリが既に実行中です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続IDに対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

## OP Get version number

---

**OP Get version number**    Version number

引数                      タイプ                      説明

このコマンドには、引数はありません。

Version number              倍長整数                      4D Open for 4D のバージョン

### 説明

**OP Get version number** は 4D Open for 4D のバージョン番号を返します。戻り値 < Version number > (4バイト倍長整数) のハイワードにはバージョン番号 (6xx)、ローワードにはビルド番号が格納されています。

### 例題

1. Version 3 :

```
$Version:= OP Get Version Number
`ハイワードを抽出。
StringVers:= String ($Version//65536)
`ローワードを抽出。
StringVers:=StringVers+"ビルド番号:"+ String ($Version%65536)
```

2. Version 6 :

```
$Version:= OP Get Version Number
`ハイワードを抽出。
StringVers:= String ($Version >> 16)
`ローワードを抽出。
StringVers:=StringVers+"ビルド番号:"+ String ($Version&65535)
```

### 参照

なし



## ユーザとグループ、序論

---

この章では以下のルーチンについて説明します：

**OP Get user list**：ユーザリストを取得します。

**OP Enter password**：4th Dimension パスワードダイアログボックスを表示します。

**OP Get users and groups**：ユーザとグループについて詳細な情報を取得します。

### 参照

OP Enter password、OP Get user list、OP Get users and groups

## OP Get user list

---

**OP Get user list** (connectionID; listSelect; userNames; userConnections;

userLastLogin ) Error

引数	タイプ	説明
connectionID	倍長整数	使用する接続ID
listSelect	倍長整数	管理者(1)かデザイナー(0)
userNames	配列	ユーザ名
userConnections	配列	ユーザごとの接続回数
userLastLogin	配列	ユーザごとの最終ログイン日付
Error	倍長整数	エラーコード

### 説明

**OP Get user list** は、ユーザリストを取得します。引数 < listSelect > に渡す値で、管理者 (1) が作成したユーザのリストを得るのか、デザイナー (0) が作成したユーザのリストを得るのかを使い分けます。また、同時にユーザごとのログイン回数と最終ログイン日付も得られます。

引数 < UserNames > は固定長文字列かテキストの配列です。

引数 < UserConnections > は固定長文字列、テキスト、実数、整数、倍長整数の配列です。

引数 < UserLastLogins > は固定長文字列、テキスト、日付の配列です。

エラーコード	説明
-9968	レコード位置番号が範囲外です。
-9972	テーブル番号が範囲外です。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

## 例題

ユーザのログイン記録を調べ、最近アクセスのないユーザがいれば管理者に知らせようとしています。

```
ARRAY TEXT (arUserNames;0)
ARRAY DATE (arLastLog;0)
ARRAY INTEGER (arUserConns;0)
C_LONGINT ($errCode)
C_TEXT ($message)

、 「Administrator」 が作成したユーザについて情報を得る。
$errCode:= OP Get user list (vConnectID;1;arUserNames;arUserConns;arLastLog)

$message:= ""

For ($i;1;Size of array (arUserNames))
  If (arLastLog {$i} < Current date -7)
    `1週間ログインの記録がないユーザがいれば警告を出す。
    $message:= $message + arUserNames {$i} + ","
  End if
End for

$message:= $message + Char (Carriage return) + "...以上のユーザは1週間の間、
                                         一度もログインしていません。"
$message:= $message + Char (Carriage return) +
               "彼らが帰ってくるまでアクセス権を停止しておいた方が良いでしょう。"

SendMessage ($mess;"admin_4D@acme.com")
```

## 参照

OP Get User Properties、OP Get users and groups

## OP Enter password

**OP Enter password** ( userNames; userName; userPassword )    Error

引数	タイプ	説明
userNames	配列	ユーザ名リスト
userName	倍長整数	選択されたユーザ名
userPassword	配列	入力されたパスワード
Error	倍長整数	エラーコード

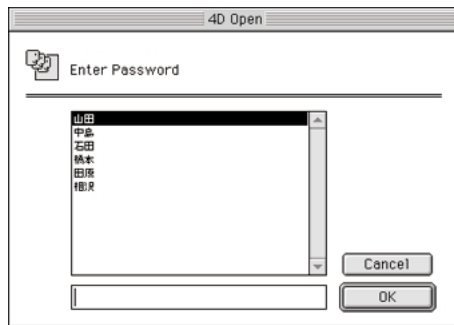
### 説明

**OP Enter password** を使用して 4th Dimension 標準のパスワードダイアログを表示させることができます。ユーザがダイアログに表示されたユーザ名一覧から選択したログインユーザ名と、入力したパスワードを取得できます。

このルーチン自体は、アクセス権限のチェックを行わないため、得られたユーザ名とパスワードを使用して、**OP Open connection** ルーチンをコールし、サーバ側ストラクチャのパスワードシステムにログインを試みることでアクセスの可否を問い合わせてください。

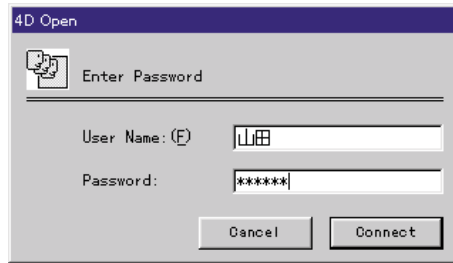
引数 < userName > はユーザが選択したユーザ名が、引数 < userPassword > にはユーザが入力したパスワードが返ってきます。ユーザ名の最大長は30バイト、パスワードの最大長は15バイトです。引数 < userNames > に渡す配列は、固定長文字列かテキスト型です。

配列が空でなければ以下のようなダイアログボックスが表示されます。





配列が空の場合は以下のようなダイアログボックスが表示されます。



## エラーコード

ユーザが「OK」、または「Connect」ボタンをクリックした場合、**OP Enter password**は「0」を返します。そうでない場合は、以下のエラーコードを返します。

エラーコード	説明
10147	「パスワード」ダイアログでユーザが「Cancel」ボタンを押しました。
10154	このコマンドを今すぐには実行できません。

## 例題

この例題では、ユーザ名とパスワードをローカル側の[Preference]テーブルに保存しています。4D OPEN経由の接続を行う度にユーザ名とパスワードの入力をさせるのではなく、保存しておいたログイン情報を使用して、いつでも、何回でも自動ログインを行うことができます。

```

`メソッド"GetUserPasswd"
C_LONGINT ($errcode)
ARRAY TEXT (arUserNames;0)
ARRAY DATE (arLastLog;0)
ARRAY INTEGER (arUserConns;0)
C_STRING (30;vUserName)
C_STRING (15;vUserPasswd)

`ユーザ名リストを得る。
$errCode:= OP Get user list (vConnectID;arUserNames;arLastLog;arUserConns)

`パスワードダイアログを表示して、ログインユーザ名とパスワードを
`入力させる。
$errCode:= OP Enter password (arUserNames;vUserName;vUserPasswd)

READ WRITE ([Preferences])
`[Preferences]テーブルのレコードを更新、もしくは新規登録ユーザ名
QUERY ([Preferences];[Preferences]KeyWord= "RemoteLoginName")
If (Records in selection ([Preferences])=0)
  NEW RECORD ([Preferences])
End if
[Preferences]KeyWord:= "RemoteLoginName"
[Preferences]Value:= vUserName

```

```

SAVE RECORD ([Preferences])
  `パスワード
QUERY ([Preferences];[Preferences]KeyWord = "RemoteLoginPasswd")
If (Records in selection ([Preferences])=0)
  NEW RECORD ([Preferences])
End if
[Preferences]KeyWord:= "RemoteLoginPasswd"
[Preferences]Value:= vUserPasswd
SAVE RECORD ([Preferences])
UNLOAD RECORD ([Preferences])

ARRAY TEXT (arUserNames;0)
ARRAY DATE (arLastLog;0)
ARRAY INTEGER (arUserConns;0)

  `メソッド"OpenRemoteConnection" -> connectionID
C_LONGINT (netCompID;$serverID;$connectID)
C_LONGINT ($0)
READ ONLY ([Preferences])
QUERY ([Preferences];[Preferences]KeyWord = "RemoteServerAdress")
$address:= [Preferences]Value
QUERY ([Preferences];[Preferences]KeyWord = "RemoteServerProtocol")
$protocol:= [Preferences]Value
QUERY ([Preferences];[Preferences]KeyWord = "RemoteLoginName")
$username:= [Preferences]Value
QUERY ([Preferences];[Preferences]KeyWord = "RemoteLoginPasswd")
$userPasswd:= [Preferences]Value

If ( ($address="" ) | ($protocol="" ) | ($username="" ) | ($userPasswd="" ) )
  `Preference]テーブルにログイン情報がない。
  $0:= -1
Else
  $errCode:= LoadNetComp ($protocol;->netCompID)
  If ($errCode = 0)
    `サーバアドレスからサーバIDを導き出す。
    $errCode:= OP Find 4D Server (netCompID;$address;serverID)
  End if
  If ($errCode =0)
    `接続を開く。
    $errCode:= OP Open connection ($serverID; $connID; "CustService";
      $username;$userPasswd;"OrderEntry")
  End if
  If ($errCode = 0)
    $0:=$connID
  Else
    $0:= $errCode
  End if
End if

```

参照

OP Open connection

## OP Get users and groups

**OP Get users and groups** (connectionID; listSelect; userNames; userConnections; userStartMethod; userLastLogins; ownerList; groupNames; groupSizes; groupOwners; members ) Error

引数	タイプ	説明
connectionID	倍長整数	接続 ID
listSelect	倍長整数	取得したいユーザリストの作成者 (管理者 = 1、デザイナー = 0)
userNames	配列	ユーザ名
userConnections	配列	ユーザごとの接続回数
userStartMethod	配列	ユーザごとのスタートアップメソッド
userLastLogins	配列	ユーザごとの最終ログイン日付
ownerList	配列	オブジェクトオーナー
groupNames	配列	グループ名
groupSizes	配列	グループごとのユーザ数
groupOwners	配列	グループオーナー
members	配列	グループごとのユーザリスト
Error	倍長整数	エラーコード

### 説明

**OP Get users and groups** は、ユーザとグループについての情報を取得します。

このコマンドを使用できるのは、デザイナーが管理者の権限でデータベースへ接続している時だけです。

4th Dimension のユーザには 2 種類あります。デザイナーと管理者それぞれ別個にユーザのリストを管理しています。引数 < listSelect > は、そのどちらについて調べるのかを指示します。

引数 < userNames >、< userStartMethod >、< groupNames > は、固定長文字列かテキストの配列です。

引数 < userConnections >、< ownerList >、< groupSizes >、< groupOwners > は、固定長文字列、テキスト、実数、整数、倍長整数の配列です。

引数 < userLastLogins > は、固定長文字列、テキスト、日付の配列です。

引数 < members > は固定長文字列、テキスト、整数、倍長整数の 2 次元配列です。

1次元目の各要素は、グループを示す配列です。2次元目の各要素は、1次元目のグループに属するユーザやグループのIDです。値は数値で、その種別により正の値と負の値に別れています。

以下の表がその特性と値の対応です。

値	指し示すもの	説明
1 ~ 15000	Designerの作成したユーザ	引数 < listSelect > に0を指定した時に、引数 < userNames > の配列に得られたユーザ名の要素番号を示している。
15001 以上	Designerの作成したグループ	引数 < listSelect > に0を指定した時に、引数 < groupNames > の配列に得られたグループ名の要素番号 + 15000を示している。
-11 ~ -15000	Administratorが作成したユーザ	引数 < listSelect > に1を指定した時に、引数 < userNames > の配列に得られたユーザ名の要素番号 - 10を示している。
-15000 未満	Administratorが作成したグループ	引数 < listSelect > に1を指定した時に、引数 < groupNames > の配列に得られたグループ名の要素番号 - 15000を示している。

## 参照

OP Get User List

## サーバ情報、序論

この章では以下のルーチンについて説明します：

**OP Get server date**：サーバマシンの日付を得ます。

**OP Get server time**：サーバマシンの時間を得ます。

**OP Get server version**：4D Serverのバージョンを調べます。

**OP Count connected users**：現在サーバに接続中のユーザ数を調べます。

**OP Count user processes**：ユーザプロセスの数を調べます。

**OP Get process list**：サーバマシンで実行中のプロセスの情報を得ます。

The screenshot shows the '4D Server' administration window. At the top, it displays 'Contacts' and '4D Server 6.5.4J' with copyright information. Below this, it shows system statistics: 'データファイル: Contacts.data', 'メモリ合計: 23 845 K', 'ログファイル:', and 'キャッシュメモリ: 4 096 K'. There are also indicators for '接続ユーザ数: 3' and '実行中プロセス数: 11'. A progress bar for '稼働状況' is visible. The main part of the window is a table with columns for 'ユーザ', '時間', '状況', and '占有率'.

ユーザ	時間	状況	占有率
カーネル	00:00:06		12 %
#1: ユーザインターフェース	00:00:02	実行中	12 %
#2: クライアントマネージャ	00:00:03	実行中	0 %
#3: キャッシュマネージャ	00:00:00	延期	0 %
#4: Webサーバ	00:00:00	実行中	0 %
ACI太郎: マネージャ	00:00:00		0 %
#5: ユーザ/おかげユア対応	00:00:00	I/O 待機中	0 %
#6: ACI太郎	00:00:00	I/O 待機中	0 %
#7: デザインプロセス	00:00:00	I/O 待機中	0 %
ACI花子: マネージャ	00:00:00		0 %
#8: ユーザ/おかげユア対応	00:00:00	I/O 待機中	0 %
#9: ACI花子	00:00:00	I/O 待機中	0 %
#10: デザインプロセス	00:00:00	I/O 待機中	0 %
空室部: マネージャ	00:00:00		0 %
#11: 4D OPENプロセス	00:00:00	I/O 待機中	0 %

### 参照

OP Count connected users、OP Count user processes、OP Get process list、OP Get server date、OP Get server time、OP Get server version

## OP Get server date

---

### OP Get server date (connectionID; serverDate) Error

引数	タイプ	説明
connectionID	倍長整数	接続 ID
serverDate	倍長整数	サーバマシンの日付
Error	倍長整数	エラーコード

#### 説明

**OP Get server date** は、引数 < connectionID > で示される接続の対象サーバマシンのシステム日付を返します。

#### エラーコード

**OP Get server date** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

#### 例題

発注記録のタイムスタンプにサーバマシンの日付と時間を利用している例題です。メソッド "getTimeStamp" は、クライアントマシンのクロックが狂っていても影響を受けないようにサーバマシンから日付、時間を得ています。

```
`GetTimeStamp (connectionID) -> Timestamp
`ConnectionID number -> connection ID with target server
`Timestamp String -> YYYYMMDDHHMMSS
C_LONGINT ($1;$connID)
C_STRING (14;$ts;$0)
C_LONGINT ($errCode)
C_DATE ($serverDate)
C_TIME ($serverTime)

$ConnId:= $1
$errCode:= OP Get server date ($connID;$serverDate)
$errCode:= OP Get server time ($connID;$serverTime)

$ts:= String ( Year of ($serverDate); "####")
$ts:= $ts + String ( Month of ($serverDate); "##")
$ts:= $ts + String ( Day of ($serverDate); "##")
$ts:= $ts + Substring ( Time string ($serverTime); 1 ; 2)
```

```
$ts:= $ts + Substring ( Time string ($serverTime); 4 ; 2)  
$ts:= $ts + Substring ( Time string ($serverTime); 7 ; 2)  
$0:= $ts
```

#### 参照

Current date、Current time、OP Get server time

## OP Get server time

---

### OP Get server time (connectionID; serverTime) Error

引数	タイプ	説明
connectionID	倍長整数	接続 ID
serverTime	倍長整数	サーバマシンの現在時刻
Error	倍長整数	エラーコード

#### 説明

**OP Get server time** は、引数 < connectionID > で示される接続の対象サーバマシンの現在時刻を返します。

#### エラーコード

**OP Get server time** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

#### 例題

**OP Get server date** の例題を参照してください。

#### 参照

Current time、OP Get server date



## OP Count connected users

---

**OP Count connected users** (connectionID; countUsers)    Error

引数	タイプ	説明
connectionID	倍長整数	接続 ID
countUsers	倍長整数	現在接続中のユーザ数
Error	倍長整数	エラーコード

### 説明

**OP Count connected users** は、引数 < connectionID > で示される接続の対象サーバマシンに現在接続中のユーザ数を返します。

### エラーコード

**OP Count connected users** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

### 参照

Count users

## OP Count user processes

---

### OP Count user processes (connectionID; countProcesses) Error

引数	タイプ	説明
connectionID	倍長整数	接続 ID
countProcesses	倍長整数	現在実行中のユーザプロセス数
Error	倍長整数	エラーコード

#### 説明

**OP Count user processes** は、引数 < connectionID > で示される接続の対象サーバマシンで現在実行中のユーザプロセス数を返します。

#### エラーコード

**OP Count user processes** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

#### 参照

Count user processes、OP Count connected users

## OP Get server version

---

**OP Get server version** (connectionID; serverVersion; serverRelease;

serverUpdate) Error

引数	タイプ	説明
connectionID	倍長整数	接続ID
serverVersion	倍長整数	サーバのバージョン
serverRelease	倍長整数	サーバのリリース番号
serverUpdate	倍長整数	サーバのアップデート番号
Error	倍長整数	エラーコード

### 説明

**OP Get server version** は、引数 < connectionID > で示される接続の 4D Server のバージョン、リリース、アップデートの番号を返します。例えば、4D Server 6.5.4 なら、< serverVersion > = 6、< serverRelease > = 5、< serverUpdate > = 4 です。

### エラーコード

**OP Get server version** の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

## OP Get process list

---

**OP Get process list** (connectionID; countProcesses; userNames; stationNames; taskNames; timeSpent; taskStatus) Error

引数	タイプ	説明
connectionID	倍長整数	接続ID
countProcesses	倍長整数	サーバ上で稼働中のプロセス数
userNames	配列	プロセスを起動したユーザ名
stationNames	配列	プロセスを起動したマシン名
taskNames	配列	プロセス名
timeSpent	配列	プロセスの経過時間
taskStatus	配列	プロセスステータス
Error	倍長整数	エラーコード

### 説明

**OP Get process list** は、引数 < connectionID > で示される接続のターゲットサーバ上で実行中のプロセスについて各種情報を得ます。

引数 < userNames >、< stationNames >、< taskNames > は、固定長文字列かテキストの配列です。

引数 < timeSpent >、< taskStatus > は、固定長文字列、テキスト、実数、整数、倍長整数の配列です。

引数 < taskStatus > の配列に返される実際の値は、渡した配列の型によって以下のようなバラエティを持ちます。

数値	文字
0	Executing
1	Delayed
2	Waiting for user event
3	Waiting for Input/Output
4	Waiting for internal flag
5	Paused
6	Hidden modal dialog

## エラーコード

**OP Get process list**の実行に成功した場合、戻り値 < Error > は「0」です。そうでない場合は、以下のエラーコードが返されます：

エラーコード	説明
-108	このコマンドを実行するのに必要なメモリが足りません。
10128	4D Open for 4th Dimension プラグインが読み込まれていません。
10135	引数型が不正です。
10136	指定された接続 ID に対応する接続がありません。
10154	このコマンドを今すぐには実行できません。

## 参照

PROCESS PROPERTIES、OP Count user processes



## OP Process number

---

**OP Process number** (connectID; processName; processID) Error

引数	タイプ	説明
connectionID	倍長整数	接続 ID
processName	固定長文字列	プロセス名
processID	整数	プロセス ID
Error	倍長整数	エラーコード

### 説明

**OP Process number** は、引数 < processName > で指定されたプロセスのプロセス番号 (プロセス ID) を返します。

### 参照

OP Execute On Server

## OP Set Process Variable

---

**OP Set Process Variable** (connectID; processID; procVar; locVar) Error

引数	タイプ	説明
connectionID	倍長整数	接続 ID
processID	整数	プロセス番号
procVar	固定長文字列	サーバ側の変数名
locVar	固定長文字列	ローカル側の変数名
Error	倍長整数	エラーコード

### 説明

**OP Set Process Variable** は、ローカル側のプロセス変数 < locVar > の値を、引数 < processName > で指定されたサーバ側プロセスのプロセス変数 < procVar > へ代入します。

### 参照

OP Get Process Variable



## OP Get Process Variable

---

**OP Get Process Variable** (connectID; processID; procVar; locVar) Error

引数	タイプ	説明
connectionID	倍長整数	接続 ID
processID	整数	プロセス番号
procVar	固定長文字列	サーバ側の変数名
locVar	固定長文字列	ローカル側の変数名
Error	倍長整数	エラーコード

### 説明

**OP Get Process Variable** は、ローカル側のプロセス変数 < locVar > へ、引数 < processName > で指定されたサーバ側プロセスのプロセス変数 < procVar > の値を代入します。

### 参照

OP Set Process Variable

## OP Execute On Server

---

**OP Execute On Server** (connectID; methodName; stackSize; processName; processID{;param}{; param2; ...; paramN}) Error

引数	タイプ	説明
connectionID	倍長整数	接続ID
methodName	固定長文字列	プロセスメソッド名
stackSize	倍長整数	スタックサイズ
processName	固定長文字列	プロセス名
processID	倍長整数	プロセス番号 (ID)
param	固定長文字列	引数として渡す変数名 (最大十個まで)
Error	倍長整数	エラーコード

### 説明

**OP Execute On Server** は、サーバマシンで新しいプロセスを起動します。起動するプロセスには10個まで引数を渡すことができます。値は変数に入れてから渡してください。

サーバ側に引数 < methodName > で指定したメソッドがない場合、プロセスは起動されず、引数 < processID > にも「0」が返されます。この場合、戻り値 < Error > にはエラーコードは返されないことに注意してください。

### 参照

OP Process number

## OP Set Semaphore

---

**OP Set Semaphore** (connectID; semaphoreName; semaphoreState)    Error

引数	タイプ	説明
connectionID	倍長整数	接続 ID
semaphoreName	固定長文字列	セマフォ名
semaphoreState	倍長整数	セマフォの状態
Error	倍長整数	エラーコード

### 説明

**OP Set Semaphore** は、同名のセマフォがまだ発行されていない場合は、セマフォを発行します。この場合は、引数 < semaphoreState > に「1」が返ってきます。

既にその名前のセマフォが発行されていた場合は、引数 < semaphoreState > に「0」が返ってきます。

### 参照

OP Check Semaphore、OP Clear Semaphore

## OP Clear Semaphore

---

**OP Clear Semaphore** (connectID; semaphoreName)    Error

引数	タイプ	説明
connectionID	倍長整数	接続 ID
semaphoreName	固定長文字列	セマフォ名
Error	倍長整数	エラーコード

### 説明

**OP Clear Semaphore** は、セマフォを消去します。

### 参照

OP Check Semaphore、OP Set Semaphore

## OP Check Semaphore

---

**OP Check Semaphore** (connectID; semaphoreName; semaphoreState) Error

引数	タイプ	説明
connectID	倍長整数	接続 ID
semaphoreName	固定長文字列	セマフォ名
semaphoreState	倍長整数	セマフォの状態
Error	倍長整数	エラーコード

### 説明

**OP Check Semaphore** は、引数 < semaphoreName > で指定したセマフォの現在の状態を引数 < semaphoreState > に返します。

**OP Set Semaphore** との違いは、**OP Check Semaphore** はセマフォを新たに発行せずに、既存のセマフォの状態を調べるだけであるところです。

### 参照

OP Clear Semaphore、OP Set Semaphore



## OP Create Empty Set

---

**OP Create Empty Set** (connectID; tableID; setName) Error

引数	タイプ	説明
connectID	倍長整数	接続 ID
tableID	整数	テーブル番号
setName	固定長文字列	作成するセットの名前
Error	倍長整数	エラーコード

### 説明

**OP Create Empty Set** は、指定されたテーブルのために新しい空のセットを作ります。

### 参照

OP Create Set

## OP Create Set

---

**OP Create Set** (connectID; tableID; setName)    Error

引数	タイプ	説明
connectID	倍長整数	接続 ID
tableID	整数	テーブル番号
setName	固定長文字列	作成するセットの名前
Error	倍長整数	エラーコード

### 説明

**OP Create Set** は、指定されたテーブルのカレントセクションから引数 < setName > のセットを作ります。

### 参照

OP Create Empty Set



## OP Use Set

---

OP Use Set (connectID; setName)		Error
引数	タイプ	説明
connectID	倍長整数	接続 ID
setName	固定長文字列	使用するセットの名前
Error	倍長整数	エラーコード

### 説明

**OP Use Set**は、指定されたセットからカレントセクションを復元します。

対象となるテーブルは、引数 < setName > で示されるセットが作られた時に対象としていたテーブルと同じです。

### 参照

OP Clear Set

## OP Add To Set

---

### OP Add To Set (connectID; tableID; setName) Error

引数	タイプ	説明
connectID	倍長整数	接続 ID
tableID	整数	テーブル番号
setName	固定長文字列	対象とするセットの名前
Error	倍長整数	エラーコード

### 説明

**OP Add To Set** は、指定されたテーブルのカレントレコードを指定したセットへ追加します。

### 例題

**OP Remove From Set** の例題を参照してください。

### 参照

OP Is In Set、OP Remove From Set

## OP Remove From Set

---

**OP Remove From Set** (connectID; tableID; setName)    Error

引数	タイプ	説明
connectID	倍長整数	接続 ID
tableID	整数	テーブル番号
setName	固定長文字列	対象とするセットの名前
Error	倍長整数	エラーコード

### 説明

**OP Remove From Set**は、指定されたテーブルのカレントレコードを指定したセットから取り除きます。

### 例題

以下にセットルーチンの典型的な使用例を示します。

```

$ErrCode:= OP Single query (vConnectID;$Server_Table;$Server_Field;"=";
                                ->vValue;vRecords)
$ErrCode:= OP Create Set (vConnectID;$Server_Table;"SetA")
$ErrCode:= OP All records (vConnectID;$Server_Table)
$ErrCode:= OP Create Set (vConnectID;$Server_Table;"SetNoA")
ARRAY LONGINT (vRecNums;0)
$ErrCode:= OP All records (vConnectID;$Server_Table)
$ErrCode:= OP Get record numbers (vConnectID;$Server_Table;vRecNums)
For ($k;1;Size of array(vRecNums))
    $ErrCode:= OP Goto record (vConnectID;1;vRecNums{$k})
    $ErrCode:= OP Is In Set (vConnectID;"SetA";vRecNums{$k};vBool)
    If (vBool=1)
        $ErrCode:= OP Remove From Set (vConnectID;$Server_Table;"SetNoA")
    End if
End for

```

### 参照

OP Add To Set、OP Is In Set

## OP Clear Set

---

### OP Clear Set (connectID; setName) Error

引数	タイプ	説明
connectID	倍長整数	接続 ID
setName	固定長文字列	消去するセットの名前
Error	倍長整数	エラーコード

### 説明

**OP Clear Set** は、指定されたセットを消去し、セットによって占有されていたメモリを解放します。

### 参照

OP Remove From Set

## OP Is In Set

---

**OP Is In Set** (connectID; setName; recordNumber; isInside)    Error

引数	タイプ	説明
connectID	倍長整数	接続 ID
setName	固定長文字列	対象とするセットの名前
recordNumber	倍長整数	レコード番号
isInside	倍長整数	レコードがセットに含まれているか否か。 1 = yes、0 = no
Error	倍長整数	エラーコード

### 説明

**OP Is In Set** は、指定されたセットに指定されたレコードが含まれているかどうかを調べます。

引数 < recordNumber > に渡すレコード番号は、**OP Get records numbers** で調べられます。

調べるレコードはカレントレコードではなく、あくまで引数 < recordNumber > で指定したレコードなので、注意が必要です。

### 参照

OP Get record numbers、OP Records In Set、OP Remove From Set

## OP Records In Set

---

**OP Records In Set** (connectID; setName; totalRecords)    Error

引数	タイプ	説明
connectID	倍長整数	接続 ID
setName	固定長文字列	対象とするセットの名前
totalRecords	倍長整数	セットに含まれるレコード数
Error	倍長整数	エラーコード

### 説明

**OP Records In Set** は、セットに含まれるレコード数を返します。

### 参照

OP Is In Set、OP Remove From Set

## OP Copy Set

---

**OP Copy Set** (connectID; sourceSet; destinationSet)    Error

引数	タイプ	説明
connectID	倍長整数	接続 ID
sourceSet	固定長文字列	コピー元セット名
destinationSet	固定長文字列	コピー先セット名
Error	倍長整数	エラーコード

### 説明

**OP Copy Set** は、セット < sourceSet > をセット < destinationSet > へコピーします。

### 参照

OP Difference Set、OP Intersection Set、OP Union Set

## OP Union Set

---

**OP Union Set** (connectID; sourceSet1; sourceSet2; destSet)    Error

引数	タイプ	説明
connectID	倍長整数	接続 ID
sourceSet1	固定長文字列	ソースセット1の名前
sourceSet2	固定長文字列	ソースセット2の名前
destSet	固定長文字列	論理演算結果を格納するセット名
Error	倍長整数	エラーコード

### 説明

**OP Union Set** は、 $\langle \text{sourceSet1} \rangle$  と  $\langle \text{sourceSet2} \rangle$  に含まれるすべてのレコードでセット  $\langle \text{destSet} \rangle$  を作成します。

### 参照

OP Difference Set、OP Intersection Set



## OP Intersection Set

---

**OP Intersection Set** (connectID; sourceSet1; sourceSet2; destSet) Error

引数	タイプ	説明
connectID	倍長整数	接続 ID
sourceSet1	固定長文字列	ソースセット1の名前
sourceSet2	固定長文字列	ソースセット2の名前
destSet	固定長文字列	論理演算結果を格納するセット名
Error	倍長整数	エラーコード

### 説明

**OP Intersection Set** は、`< sourceSet1 >` と `< sourceSet2 >` のいずれにも含まれるレコードだけでセット `< destSet >` を作成します。

### 参照

OP Difference Set、OP Union Set

## OP Difference Set

---

**OP Difference Set** (connectID; sourceSet1; sourceSet2; destSet)    Error

引数	タイプ	説明
connectID	倍長整数	接続 ID
sourceSet1	固定長文字列	ソースセット1の名前
sourceSet2	固定長文字列	ソースセット2の名前
destSet	固定長文字列	論理演算結果を格納するセット名
Error	倍長整数	エラーコード

### 説明

**OP Difference Set** は、< sourceSet1 > から < sourceSet2 > にも含まれるレコードを取り除いた残りのレコードだけでセット < destSet > を作成します。

### 参照

OP Intersection Set、OP Union Set

## A

OP Add To Set(connectID;tableID;setName) Error .....	210
OP All records(connectionID;tableID) Error .....	87
OP Array to selection(connectionID;bindID) Error .....	105
OP Average(connectionID;tableID;fieldID;Result) Error .....	130

## C

OP Cancel transaction(connectionID) Error.....	116
OP Check Semaphore(connectID;semaphoreName;semaphoreState) Error	205
OP Clear named selection(connectionID;tableID;selectionName) Error .....	126
OP Clear Semaphore(connectID;semaphoreName) Error .....	204
OP Clear Set(connectID;setName) Error.....	212
OP Close connection(connectionID) Error.....	48
OP Copy named selection(connectionID;tableID;selectionName) Error.....	119
OP Copy Set(connectID;sourceSet;destinationSet) Error .....	215
OP Count connected users(connectionID;countUsers) Error .....	193
OP Count fields(connectionID;tableID;nbOfFields) Error.....	64
OP Count network components(HowManyNetComps) Error.....	20
OP Count tables(connectionID;numOfTables) Error.....	50
OP Count user processes(connectionID;countProcesses) Error.....	194
OP Create bind(bindID) Error .....	135
OP Create Empty Set(connectID;tableID;setName) Error.....	207
OP Create Set(connectID;tableID;setName) Error .....	208
OP Current Record Number(connectionID;tableID;recordNumber; recordPosition) Error.....	170
OP Cut named selection(connectionID;tableID;selectionName) Error .....	121

## D

OP Define bind by numbers(bindID;tableID;fieldID;localTableID; localFieldID;variableName) Error.....	137
OP Define bind by pointer(bindID;tableID;fieldID;boundObject) Error .....	140
OP Delete 4D Server(serverID) Error .....	45
OP Delete bind(bindID) Error.....	136
OP Delete record(connectionID;tableID) Error .....	163
OP Delete selection(connectionID;tableID) Error .....	91
OP Difference Set(connectID;sourceSet1;sourceSet2;destSet) Error.....	218
OP Distinct values(connectionID;tableID;fieldID;distinctValues) Error .....	103

## E

OP End Remote Connection Error .....	30
OP Enter password( userNames;userName;userPassword) Error.....	184
OP Execute On Server(connectID;methodName;stackSize;processName; processID{;param};;param2;...;paramN)) Error .....	202

## F

OP Find 4D Server(netCompID;serverName;serverID) Error.....	43
OP Flush Buffers(connectionID) Error .....	178

## G

OP Get all field numbers(connectionID;tableFields;tableIDs;fieldIDs) Error....	61
OP Get all tablenames(connectionID;tableNames) Error .....	51
OP Get error text(errorCode) Error .....	172
OP Get field properties(connectionID;tableID;tableInvisible;fieldNames; fieldTypes;fieldLengths;fieldIndexed;fieldInvisible) Error.....	53
OP Get network component info (NetCompNumber;NetCompID; NetCompName) Error.....	21
OP Get one field number(connectionID;tableField;tableID;fieldID) Error.....	60
OP Get one tablename(connectionID;tableID;nameOfTable) Error.....	63
OP Get option(optionNumber;optionValue) Error .....	177
OP Get process list(connectionID;countProcesses;userNames; stationNames;taskNames;timeSpent;taskStatus) Error.....	196
OP Get Process Variable(connectID;processID;procVar;locVar) Error .....	201
OP Get record numbers(connectionID;tableID;recordNumbers{;	

firstRecordNumber{;lastRecordNumber}}) Error .....	168
OP Get server date(connectionID;serverDate) Error .....	190
OP Get server time(connectionID;serverTime) Error .....	192
OP Get server version(connectionID;serverVersion;serverRelease; serverUpdate) Error .....	195
OP Get station name Error .....	28
OP Get table properties(connectionID;tableID;tableInvisible;tableLeft; tableTop;tableRight;tableBottom;fieldNames;fieldTypes;fieldLength; fieldIndexed;fieldUnique;fieldInvisible;fieldEnterable;fieldModifiable; fieldMandatory;fieldRelatedTa .....	56
OP Getuser list(connectionID;listSelect;userNames;userConnections ;userLastLogin) Error .....	182
OP Get users and groups(connectionID;listSelect;userNames; userConnections;userStartMethod;userLastLogins;ownerList; groupNames;groupSizes;groupOwners;members) Error .....	187
OP Get version number Versionnumber .....	179
OP Goto record(connectionID;tableID;recordNumber;bindID;lockStatus) Error .....	166
OP Goto selected record(connectionID;tableID;recordNumber{; bindID{;lockStatus}}) Error .....	149

## I

OP Intersection Set(connectID;sourceSet1;sourceSet2;destSet) Error .....	217
OP Is In Set(connectID;setName;recordNumber;isInside) Error .....	213

## L

OP Load network component(NetCompID) Error .....	23
OP Load record(connectionID;bindID;tableID;lockStatus) Error .....	152

## M

OP Many to one join(connectionID;manyTableID;oneTableID) Error .....	92
OP Max(connectionID;tableID;fieldID;Result) Error .....	132
OP Min(connectionID;tableID;fieldID;Result) Error .....	131
OP Multi order by(connectionID;tableID;tableIDs;fieldIDs;orderDirections) Error .....	81
OP Multi query(connectionID;tableID;tableIDs;fieldIDs;logicalOperators; queryOperators;queryValues;recordsFound) Error .....	73

OP Multi query selection(connectionID;tableID;tableIDs;fieldIDs;  
logicalOperators;queryOperators;queryValues;recordsFound) Error .....77

## N

OP New record(connectionID;bindID) Error.....160

## O

OP One to many join(connectionID;manyTableID;manyFieldID) Error.....94

OP Open connection(serverID;connectionID;station;userName;  
password;taskName) Error .....46

## P

OP Process number(connectID;processName;processID) Error.....199

## R

OP Records in selection(connectionID;tableID;recordsInTable) Error.....85

OP Records In Set(connectID;setName;totalRecords) Error.....214

OP Records in table(connectionID;tableID;recordsInTable) Error .....84

OP Reduce selection(connectionID;tableID;selectionSize) Error .....89

OP Remote Connection Status(phase;error;connectedTo;lastMessage)  
Error.....31

OP Remove From Set(connectID;tableID;setName) Error .....211

OP Request(connectID;status) Error .....26

## S

OP Scan index(connectionID;tableID;fieldID;selectionSize;scanOrigin)  
Error.....96

OP Select 4D Server(NetCompID;serverName;serverID;otherButton)  
Error.....39

OP Selection to array(connectionID;bindID{;tableID}) Error .....100

OP Sequence number(connectionID;tableID;sequenceNumber) Error .....162

OP Set access mode(connectionID;tableID;accessMode) Error .....147

OP Set format(bindID;tableID;fieldID;conversionFormat) Error .....142

OP Set option(optionNumber;optionValue) Error.....174

OP Set Process Variable(connectID;processID;procVar;locVar) Error.....200

OP Set Semaphore(connectID;semaphoreName;semaphoreState) Error ....203

OP Single order by(connectionID;tableID;fieldID;orderDirection) Error .....	71
OP Single query(connectionID;tableID;fieldID;queryOperator;queryValue; recordsFound) Error .....	67
OP Single query selection(connectionID;tableID;fieldID;queryOperator; queryOperator;queryValue;recordsFound) Error .....	69
OP Start Remote Connection(path;user;password;async) Error.....	29
OP Start transaction(connectionID) Error .....	112
OP Subselection to array(connectionID;bindID;firstRecord;lastRecord {;tableID}) Error.....	108
OP Sum(connectionID;tableID;fieldID;Result) Error .....	128

## U

OP Union Set(connectID;sourceSet1;sourceSet2;destSet) Error.....	216
OP Unload network component(NetCompID) Error.....	24
OP Unload record(connectionID;tableID) Error .....	155
OP Update record(connectionID;bindID) Error .....	156
OP Use named selection(connectionID;tableID;selectionName) Error.....	124
OP Use Set(connectID;setName) Error.....	209

## V

OP Validate transaction(connectionID) Error.....	115
--	-----

