

DBPARTNER2 Client プログラマーズガイド

解説・手引・文法書

3020-6-152

前書き

■ 対象製品

P-2C63-4514 DBPARTNER2 Client (参照版) 02-01 (適用 OS : Windows Vista, Windows Server 2008, Windows 7, Windows 8, Windows 8.1, Windows 2008 R2, Windows Server 2012, Windows 20012 R2, Windows 10)

P-2C63-4614 DBPARTNER2 Client (更新版) 02-01 (適用 OS : Windows Vista, Windows Server 2008, Windows 7, Windows 8, Windows 8.1, Windows 2008 R2, Windows Server 2012, Windows 20012 R2, Windows 10)

P-2C63-4714 DBPARTNER2 Client (定義版) 02-01 (適用 OS : Windows Vista, Windows Server 2008, Windows 7, Windows 8, Windows 8.1, Windows 2008 R2, Windows Server 2012, Windows 20012 R2, Windows 10)

P-2C63-5514 DBPARTNER ODBC Driver 02-00 (適用 OS : Windows Vista, Windows Server 2008, Windows 7, Windows 8, Windows 8.1, Windows 2008 R2, Windows Server 2012, Windows 20012 R2, Windows 10)

■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

HITACHI, DBPARTNER2 Client は、株式会社 日立製作所の商標または登録商標です。

ActiveX は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Internet Explorer は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Microsoft は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Microsoft および Excel は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Microsoft および SQL Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Microsoft および Visual Basic は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Microsoft および Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Microsoft および Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Oracle と Java は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。

UNIX は、The Open Group の米国ならびに他の国における登録商標です。

その他記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

■ 製品構成について

DBPARTNER2 Client は、次に示す製品から構成されます。

- ・ DBPARTNER2 Client
- ・ DBPARTNER ODBC Driver

DBPARTNER2 Client の形名の違いで、次に示す機能の差異があります。

機能	P-2C63-4514 (参照版)	P-2C63-4614 (更新版)	P-2C63-4714 (定義版)
データ抽出	○	○	○
データ更新	—	○	○
表定義	—	—	○
データベースの保存	—	—	○
予約検索	—	—	○

(凡例)

- ：機能を使えます。
- ：機能を使えません。

■ 発行

2016 年 1 月 3020-6-152

■ 著作権

All Rights Reserved. Copyright (C) 2016, Hitachi, Ltd.

はじめに

このマニュアルは、データベースアクセスプログラム DBPARTNER2 Client の使い方について説明したものです。DBPARTNER2 Client は、次に示す製品から構成されます。

- DBPARTNER2 Client
- DBPARTNER ODBC Driver

このマニュアルでは、DBPARTNER2 Client の ActiveX コントロール仕様と DBPARTNER ODBC Driver について説明します。

DBPARTNER2 の最新情報

次に示すアドレスで、DBPARTNER2 の最新情報を参照できます。

<http://www.hitachi.co.jp/soft/dbpart/>

■ 対象読者

DBPARTNER2 Client 又は DBPARTNER ODBC Driver をクライアント PC に組み込んで業務処理をするアプリケーション開発者の方々を対象にしています。

Windows の操作について知識があることを前提としています。さらに、データベースのサーバと接続するときには、サーバの環境に関する知識があることを前提としています。

アプリケーションを作成するときは、次に示すビジュアル開発ツールの使い方の知識を前提としています。

- Microsoft Visual Basic

アプリケーションから SQL 文を使ってデータベースにアクセスするときは、SQL 文の文法の知識を前提としています。

■ マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

第 1 章 DBPARTNER2 Client の紹介

DBPARTNER2 Client の特長と機能、サーバと接続する形態について説明しています。

第 2 章 環境の設定

DBPARTNER2 Client の実行環境を設定する方法について説明しています。

第 3 章 アプリケーションを使ったデータベースアクセス (ODBC 接続)

DBPARTNER2 Client のアプリケーションから ActiveX コントロールのオブジェクトを使ってデータベースにアクセスする方法について説明しています。

第 4 章 DBPARTNER2 Client のオブジェクトの文法 (ODBC 接続)

DBPARTNER2 Client のアプリケーションを開発するときに使う ActiveX コントロールのオブジェクトの文法について説明しています。

第 5 章 ODBC 関数を使ったデータベースアクセス

DBPARTNER ODBC Driver をセットアップすると使える ODBC 関数について説明しています。

第 6 章 アプリケーションを使ったデータベースアクセス (CommuniNet 接続)

DBPARTNER2 Client のアプリケーションから ActiveX コントロールのオブジェクトを使ってデータベースにアクセスする方法について説明しています。

第 7 章 DBPARTNER2 Client のオブジェクトの文法 (CommuniNet 接続)

DBPARTNER2 Client のアプリケーションを開発するときに使う ActiveX コントロールのオブジェクトの文法について説明しています。

第 8 章 CommuniNet 接続を使うときの参照情報

CommuniNet 接続で DBPARTNER2 Client のアプリケーションを開発するときに、データベースとサーバプログラム別で注意する必要がある項目について説明しています。

第 9 章 トラブルが起こったときは

DBPARTNER2 Client にトラブルが起こったときの対処について説明しています。

第 10 章 メッセージの一覧

DBPARTNER2 Client をインストールするときに出力されるメッセージの意味と、DBPARTNER ODBC Driver から通知されるメッセージの意味について説明しています。

付録 A AP 作成ウィザード

ActiveX コントロールを使用したアプリケーション作成を支援するプログラムである AP 作成ウィザードについて説明しています。

付録 B DBPARTNER2 Client の予約語

表名や列名として使用できない、DBPARTNER2 Client の予約語について説明しています。

付録 C DBPARTNER2 Client が扱うデータの形式

DBPARTNER2 Client が扱うデータ (ファイル) の形式について説明しています。

付録 D プロパティとメソッドの制限値

ActiveX コントロールのプロパティとメソッドのパラメタの制限値，及び文字定数について説明しています。

付録 E 列の属性

アプリケーションで使用する文字定数の属性について説明しています。

付録 F 繰り返し列と配列列での制限事項

表の列に繰り返し列や配列列を使用している場合の制限事項について説明しています。

付録 G このマニュアルの参考情報

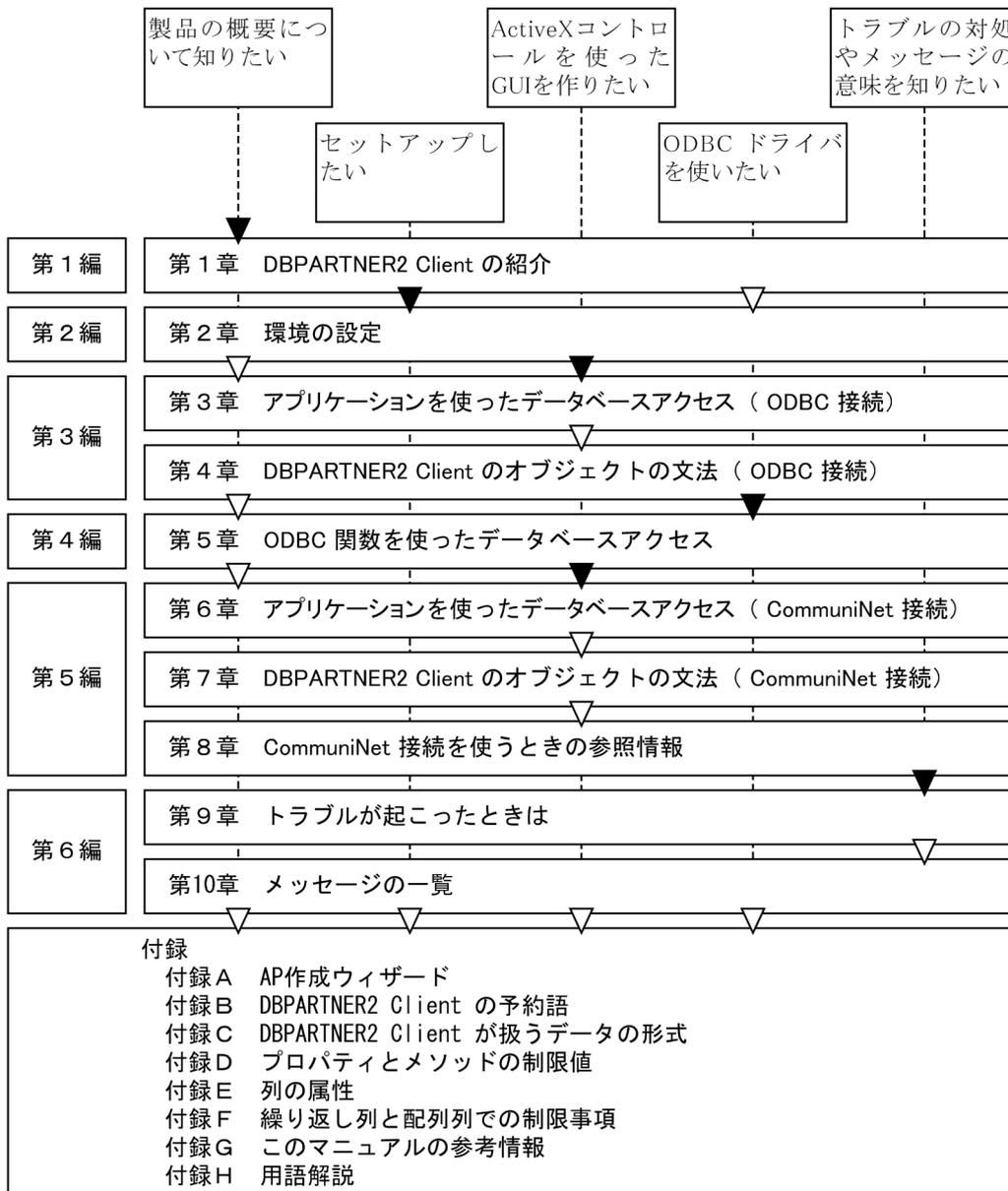
このマニュアルを読むに当たっての参考情報について説明しています。

付録 H 用語解説

このマニュアルで使用している用語について説明しています。

■ 読書手順

このマニュアルは，利用目的に合わせて章を選んで参照できます。利用目的別にお読みいただくことをお勧めします。



(凡例)  : 必ず読む項目  : 必要に応じて読む項目

■ マニュアルで使用する記号

全体で使用する記号

- このマニュアル全体で使う記号を次のように定義します。

< > : 文字キーを示します。

[] : メニューの名称やダイアログボックス又はダイアログボックス中のボタンを示します。

補足 : 操作や機能の補足説明を示します。

- Windows の [スタート] ボタンからメニューをたどって開く操作又はダイアログボックスのメニューを開く操作は、ハイフンでつないで表記します。

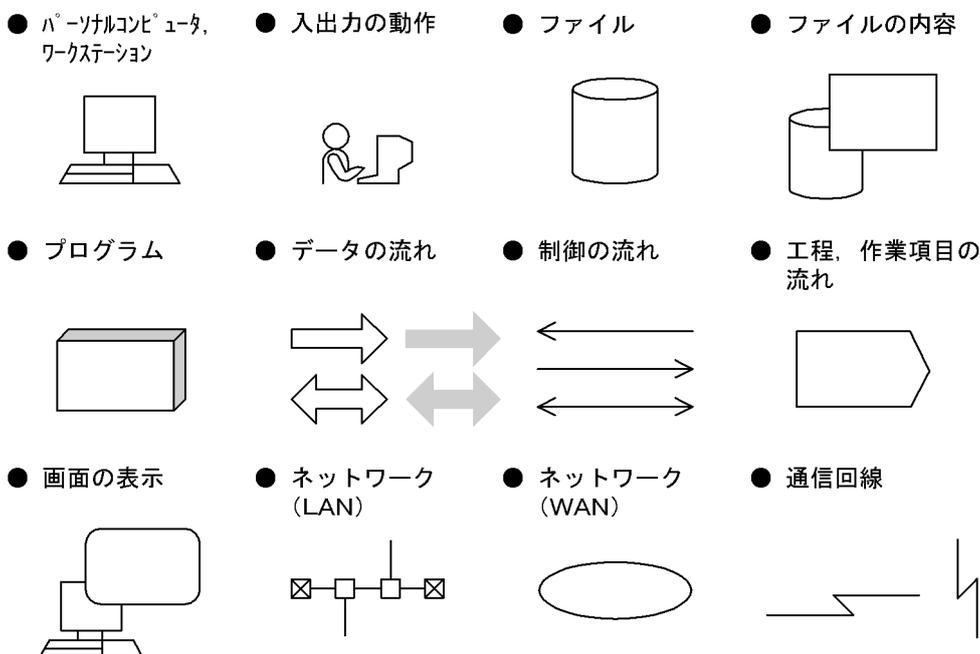
(例)

[スタート] ボタンから [プログラム] メニューの下の [DBPARTNER2] メニューを選ぶ操作は「[スタート] - [プログラム] - [DBPARTNER2] メニューを選ぶ」と表記します。

- Windows を組み込んだドライブやフォルダ名は、標準的な例で記述してあります。標準的な環境と異なる Windows で使うときは、ドライブやフォルダ名をお使いになる Windows の環境に読み替えてください。

図中で使用する記号

このマニュアルの図中に使う記号を次のように定義します。



条件式の文法で使用する記号

このマニュアルの条件式の説明で使う記号を次のように定義します。

記号	意味
 ストローク	横に並べられた複数の項目に対し、項目間の区切りを示し、「又は」の意味を示します。 (例) A B C は、「A, B 又は C」を意味します。
{ } 波括弧	この記号で囲まれている複数の項目のうちから一つを選択します。項目が縦に複数行にわたって記述されている場合は、そのうちの 1 行分を選択します。項目が並べられ、記号 で区切られている場合は、そのうちの一つを選択します。 (例)

記号	意味
	{A B C}では、「A, B 又は C のどれかを指定する」ことを示しています。
[] きっ甲	この記号で囲まれている項目は省略してもよいことを意味します。複数の項目が縦又は横に並べて記述されている場合には、すべてを省略するか、記号{ }と同じく、どれか一つを選択します。 (例) [A] は「何も指定しない」か、「A と指定する」ことを示します。[B C] は「何も指定しない」か、「B 又は C と指定する」ことを示します。
… リーダー	記述が省略されていることを示します。この記号の直前に示された項目を繰り返し複数個指定できます。 (例) A, B, B, …は、「A の後ろに B を必要個指定する」ことを示します。
— 下線	括弧内のすべての項目を省略したときに、システムが仮定する値を示します。複数の項目のうち、1 項目にだけ付けてあります。 (例) [A B <u>C</u>] は、「A, B, C のどれも指定しなかった場合、システムは C を指定したときと同じ処理をする」ことを示します。
《 》 二重角括弧	設定値を省略したときに、システムが仮定する値を示します。 (例) 《HiRDB》は、「値を省略すると、HiRDB という文字列が仮定される」ことを示します。
△ 白三角	空白を空けることを示します。 (例) EDIT△PANEL は、EDIT と PANEL の間に一つ空白を空けることを示します。
△n 白三角	空白を n 回以上繰り返して指定することを示します。 (例) NOT△ ₂ IN は、NOT の後ろに二つ以上の空白を空けることを示します。

コマンドの変数指定で使用する記号

このマニュアルの条件式の変数指定で使う記号を次のように定義します。

記号	意味
< >	項目を記述するときに従わなければならない構文要素を示します。

■ このマニュアルで使用する構文要素

このマニュアルの文法説明で使用する構文要素の種類を次に示します。

種類	定義
<英字>	A~Z a~z
<英大文字>	A~Z @ # ¥
<英小文字>	a~z

種類	定義
〈数字〉	0~9
〈英数字〉	A~Z a~z 0~9
〈カナ文字〉	ア~ン ヲ カナ小文字 ` ° ー
〈記号〉	! " # \$ % & ' () + , - . / : ; < = > @ [] ^ _ { } ー タブ 空白

■ マニュアルとオンラインヘルプの使い分け

DBPARTNER2 Client では、オンラインヘルプを提供しています。マニュアルとオンラインヘルプは、それぞれの用途に応じて使い分けてください。

マニュアル

ActiveX コントロールを使用して初めてアプリケーションを作成するとき。

ActiveX コントロールの全体の機能を理解したいとき。

DBPARTNER2 Client の機能の概要を知りたいとき。

DBPARTNER2 Client を使用するときに必要な環境設定を知りたいとき。

オンラインヘルプ

アプリケーションの作成時に ActiveX コントロールのプロパティ、メソッドの形式やパラメタの指定内容などを参照したいとき。

目次

前書き	2
はじめに	4

第1編 概要

1	DBPARTNER2 Client の紹介	24
1.1	DBPARTNER2 Client の概要	25
1.2	DBPARTNER2 Client の特長	27
1.2.1	ActiveX コントロールのオブジェクトを使ってデータベースにアクセスできます	27
1.2.2	ODBC に対応したデータベースにアクセスできます	30
1.3	DBPARTNER2 Client の機能	31
1.3.1	データの抽出	31
1.3.2	データの更新	33
1.4	DBPARTNER2 Client とサーバを接続する形態	34
1.4.1	DBPARTNER2 Client で使えるサーバプログラム	34
1.4.2	ODBC ドライバの概要	35
1.4.3	CommuniNet 接続の概要	37

第2編 準備

2	環境の設定	40
2.1	DBPARTNER2 Client の環境を設定する手順	41
2.2	DBPARTNER2 Client のインストール	43
2.2.1	DBPARTNER2 Client の製品の種類	43
2.3	ODBC ドライバのセットアップ	45
2.3.1	ODBC ドライバのインストールの手順	45
2.3.2	ODBC ドライバをインストールすると作成されるファイル	45
2.3.3	DABroker ドライバのデータソースと通信ドライバの設定	46
2.4	マクロスクリプトの設定 (CommuniNet 接続の場合)	54
2.4.1	CommuniNet のインストールの手順	54
2.4.2	マクロスクリプトファイルの作成	54

第3編 ODBC 接続の ActiveX コントロール

3	アプリケーションを使ったデータベースアクセス (ODBC 接続)	59
3.1	DBPARTNER2 Client が提供する ActiveX コントロールの概要	60

3.1.1	ActiveX コントロールのオブジェクトの階層	60
3.1.2	プロパティとメソッドの表記	61
3.1.3	アプリケーションのサンプル	62
3.1.4	アプリケーションの作成	62
3.2	データベースアクセスの手順	63
3.2.1	DBPARTNER オブジェクトのプロパティの設定	64
3.3	データベースへの接続と接続の解除 (DBPARTNER オブジェクト)	65
3.3.1	データベースへの接続	65
3.3.2	データベースへのアクセスのキャンセル	65
3.3.3	データベースとの接続の解除	66
3.4	トランザクションと排他制御 (DBPARTNER オブジェクト)	67
3.4.1	トランザクションの制御の設定	67
3.4.2	排他制御	67
3.5	表と列の情報の取得 (TableList オブジェクト, ColumnList オブジェクト)	69
3.5.1	表情報の取得	69
3.5.2	列情報の取得	71
3.6	データの抽出 (QueryDefine オブジェクト)	74
3.6.1	QueryDefine オブジェクトを使ってデータを抽出する手順	74
3.7	抽出データの管理 (Result オブジェクト)	76
3.7.1	抽出データの格納手順	76
3.7.2	Result オブジェクトと取得するデータの関係	77
3.7.3	FieldCodeType プロパティで取得するデータ	78
3.8	データの更新 (UpdateDefine オブジェクト)	80
3.8.1	データを更新する手順	80
3.8.2	更新実行モードとデータ更新の関係	82
3.9	SQL を直接記述してアクセス (SQLDirect オブジェクト)	84
3.9.1	データの抽出 (SQLDirect オブジェクト)	84
3.9.2	データの更新 (SQLDirect オブジェクト)	85
3.10	データの抽出 (Catalog オブジェクト)	87
3.10.1	Catalog オブジェクトを使ってデータを抽出する手順	87
3.11	ストアドプロシジャの実行 (Procedure オブジェクト)	89
3.11.1	Procedure オブジェクトを使ってストアドプロシジャを実行する手順	89
3.12	避けて欲しい処理	91
3.12.1	プロパティとメソッド処理中に、別の入力を許す処理	91
3.12.2	Disconnect メソッドを実行しないでアプリケーションを終了する処理	91
3.12.3	2 フェーズコミットを必要とする更新処理	91
3.12.4	Internet Explorer で ActiveX コントロールを使う処理	92
3.13	プロパティとメソッドの使い方の詳細	93
3.13.1	プロパティとメソッドの実行順序	93

3.13.2	プロパティとメソッドの制限	99
4	DBPARTNER2 Client のオブジェクトの文法 (ODBC 接続)	101
4.1	プロパティページの設定	102
4.2	DBPARTNER オブジェクトの詳細	104
	CommitMode プロパティ	104
	ConnectDialogMode プロパティ	105
	DataSource プロパティ	106
	DictPath プロパティ	107
	TableDefSource プロパティ	108
	IsConnect プロパティ	108
	Password プロパティ	109
	UserID プロパティ	109
	Cancel メソッド (CancelEx メソッド)	110
	Commit メソッド	111
	Connect メソッド	111
	Disconnect メソッド	112
	Rollback メソッド	113
4.3	SQLDirect オブジェクトの詳細	114
	ExecutedRowCount プロパティ	114
	SQLText プロパティ	115
	Execute メソッド	115
4.4	Result オブジェクトの詳細	117
	DataCheck プロパティ	118
	EOF プロパティ	118
	FieldCodeType プロパティ	119
	FieldCount プロパティ	120
	FieldData プロパティ	121
	FieldDataB プロパティ	121
	FieldDataLength プロパティ	122
	FieldHeader プロパティ	123
	FieldNameLength プロパティ	123
	FieldType プロパティ	124
	HeaderOutput プロパティ	125
	QueryCountLimit プロパティ	125
	RecordData プロパティ	126
	RecordHeader プロパティ	127
	Close メソッド	127
	CopyToClipboard メソッド	128
	Get メソッド	129
	SaveLargeObject メソッド	129
	SaveToFile メソッド	130
4.5	TableList オブジェクトの詳細	132
	Count プロパティ	132

	CountLimit プロパティ	133
	SelectDict プロパティ	133
	Pattern プロパティ	134
	TableComment プロパティ	135
	TableCreateDate プロパティ	136
	TableDataSource プロパティ	136
	TableGroupName プロパティ	137
	TableHeader プロパティ	137
	TableName プロパティ	138
	TableOwner プロパティ	139
	TableType プロパティ	139
	Type プロパティ	140
	Get メソッド	141
4.6	ColumnList オブジェクトの詳細	142
	ColumnComment プロパティ	142
	ColumnHeader プロパティ	143
	ColumnLength プロパティ	144
	ColumnName プロパティ	144
	ColumnNameLength プロパティ	145
	ColumnScale プロパティ	145
	ColumnType プロパティ	146
	Count プロパティ	146
	CountLimit プロパティ	147
	SelectDict プロパティ	148
	TableName プロパティ	148
	Get メソッド	149
4.7	QueryDefine オブジェクトの詳細	150
	Exclusive プロパティ	150
	Execute メソッド	151
	QueryCount メソッド	151
	RemoveAll メソッド	152
4.8	Tables コレクションオブジェクトの詳細	153
	Count プロパティ	153
	Add メソッド	153
	Item メソッド	154
	RemoveAll メソッド	154
4.9	Table オブジェクトの詳細	156
	TableName プロパティ	156
4.10	Columns コレクションオブジェクトの詳細	157
	Count プロパティ	157
	Add メソッド	157
	Item メソッド	158
	RemoveAll メソッド	158
4.11	Column オブジェクトの詳細	160

- ColumnName プロパティ 160
- FunctionMode プロパティ 160
- SortMode プロパティ 161
- SortPriority プロパティ 162
- 4.12 Conditions コレクションオブジェクトの詳細 163
 - Count プロパティ 163
 - Relation プロパティ 163
 - Add メソッド 164
 - Item メソッド 165
 - RemoveAll メソッド 166
- 4.13 Condition オブジェクトの詳細 167
 - Text プロパティ 167
- 4.14 UpdateDefine オブジェクトの詳細 168
 - ErrorInformation プロパティ 168
 - Exclusive プロパティ 169
 - ExecutedRowCount プロパティ 170
 - NCharPrefixMode プロパティ 170
 - TableName プロパティ 171
 - UpdateMode プロパティ 171
 - Execute メソッド 172
 - RemoveAll メソッド 173
- 4.15 Updates コレクションオブジェクトの詳細 174
 - Count プロパティ 174
 - Add メソッド 174
 - Item メソッド 175
 - RemoveAll メソッド 175
- 4.16 Update オブジェクトの詳細 176
 - Column プロパティ 176
 - KeyColumn プロパティ 177
 - KeyCount プロパティ 177
 - KeyValue プロパティ 178
 - UpdateValue プロパティ 178
 - UpdateValueB プロパティ 179
 - UpdateValueCount プロパティ 180
 - AddKey メソッド 180
 - AddUpdateValue メソッド 181
 - AddUpdateValueB メソッド 181
- 4.17 Inserts コレクションオブジェクトの詳細 183
 - Count プロパティ 183
 - Add メソッド 183
 - Item メソッド 184
 - RemoveAll メソッド 184
- 4.18 Insert オブジェクトの詳細 185
 - Column プロパティ 185

	InsertValue プロパティ	186
	InsertValueB プロパティ	186
	InsertValueCount プロパティ	187
	AddInsertValue メソッド	187
	AddInsertValueB メソッド	188
4.19	Deletes コレクションオブジェクトの詳細	190
	Count プロパティ	190
	Add メソッド	190
	Item メソッド	191
	RemoveAll メソッド	191
4.20	Delete オブジェクトの詳細	192
	KeyColumn プロパティ	192
	KeyCount プロパティ	193
	KeyValue プロパティ	193
	AddKey メソッド	194
4.21	Catalog オブジェクトの詳細	195
	Convert メソッド	195
	FileLoad メソッド	196
4.22	VariableList オブジェクトの詳細	197
	Count プロパティ	197
	VariableName プロパティ	197
	VariableValue プロパティ	198
4.23	Procedure オブジェクトの詳細	199
	ListPattern プロパティ	199
	Execute メソッド	200
	GetList メソッド	201
	GetParam メソッド	201
4.24	ParamList オブジェクトの詳細	202
	Count プロパティ	202
	ParamCodeType プロパティ	203
	ParamName プロパティ	203
	ParamType プロパティ	204
	ParamComment プロパティ	205
	ParamDataType プロパティ	205
	ParamLength プロパティ	206
	ParamNullable プロパティ	206
	ParamPrecision プロパティ	207
	ParamRadix プロパティ	207
	ParamScale プロパティ	208
	ParamTypeName プロパティ	208
	ParamValue プロパティ	209
	ParamValueB プロパティ	210
	ParamValueMaxLength プロパティ	211
4.25	ProcedureList オブジェクトの詳細	212

Count プロパティ 212
ProcedureComment プロパティ 212
ProcedureName プロパティ 213
ProcedureOwner プロパティ 213
ProcedureType プロパティ 214

第4編 DBPARTNER ODBC Driver

- 5 ODBC 関数を使ったデータベースアクセス 215
 - 5.1 DBPARTNER ODBC Driver が提供する ODBC 関数の概要 216
 - 5.1.1 ODBC ドライバを使ってアクセスできるデータベース 216
 - 5.2 DBPARTNER ODBC Driver で使える ODBC 関数 217
 - 5.2.1 DABroker ドライバで使える ODBC 関数 217
 - 5.3 ODBC とデータベースとのデータ型の対応 223
 - 5.3.1 DABroker ドライバの場合の ODBC とデータベースとのデータ型の対応 223
 - 5.4 ODBC 関数の規則 239
 - 5.4.1 DABroker ドライバの場合に、各データベースで使える機能 239
 - 5.4.2 各 ODBC 関数での繰り返し列と配列列の扱い 242
 - 5.5 DBPARTNER ODBC Driver で取得できる ODBC 関数のオプション値 243
 - 5.5.1 DABroker ドライバで取得できる ODBC 関数のオプション値 243
 - 5.6 ODBC 関数を実行したときのエラーメッセージ 248
 - 5.6.1 エラーメッセージに付く ODBC 関数のプレフィックス 248
 - 5.6.2 HiRDB のエラーメッセージの見方 249
 - 5.7 ODBC 関数の非同期実行 250
 - 5.8 ストアドプロシジャの実行 252

第5編 CommuniNet 接続の ActiveX コントロール

- 6 アプリケーションを使ったデータベースアクセス (CommuniNet 接続) 253
 - 6.1 DBPARTNER2 Client が提供する ActiveX コントロールの概要 254
 - 6.1.1 ActiveX コントロールのオブジェクトの階層 254
 - 6.1.2 プロパティとメソッドの表記 255
 - 6.1.3 アプリケーションのサンプル 256
 - 6.1.4 アプリケーションの作成 256
 - 6.2 データベースアクセスの手順 257
 - 6.2.1 DBPARTNER オブジェクトのプロパティの設定 257
 - 6.3 データベースへの接続と接続の解除 (DBPARTNER オブジェクト) 259
 - 6.3.1 データベースへの接続 259
 - 6.3.2 データベースへのアクセスのキャンセル 259
 - 6.3.3 データベースとの接続の解除 260
 - 6.4 トランザクションと排他制御 (DBPARTNER オブジェクト) 261

6.4.1	トランザクションの制御の設定	261
6.4.2	排他制御	261
6.5	表と列の情報の取得 (TableList オブジェクト, ColumnList オブジェクト)	263
6.5.1	表情報の取得	263
6.5.2	列情報の取得	266
6.6	データの抽出 (QueryDefine オブジェクト)	269
6.6.1	QueryDefine オブジェクトを使ってデータを抽出する手順	269
6.7	抽出データの管理 (Result オブジェクト)	271
6.7.1	抽出データを格納する方法	271
6.8	データの更新 (UpdateDefine オブジェクト)	272
6.8.1	UpdateDefine オブジェクトを使ってデータを更新する手順 (変更)	273
6.8.2	UpdateDefine オブジェクトを使ってデータを更新する手順 (追加)	273
6.8.3	UpdateDefine オブジェクトを使ってデータを更新する手順 (削除)	274
6.9	データの抽出 (Catalog オブジェクト)	275
6.9.1	Catalog オブジェクトを使ってデータを抽出する手順	275
6.10	避けて欲しい処理	277
6.10.1	プロパティとメソッド処理中に、別の入力を許す処理	277
6.10.2	TerminalClose メソッドを実行しないでアプリケーションを終了する処理	277
6.10.3	一つのアプリケーションで複数の ActiveX コントロールを使う処理	277
6.10.4	Internet Explorer で ActiveX コントロールを使う処理	278
6.11	プロパティとメソッドの使い方の詳細	279
6.11.1	プロパティとメソッドの実行順序	279
6.11.2	プロパティとメソッドの制限	284
7	DBPARTNER2 Client のオブジェクトの文法 (CommuniNet 接続)	286
7.1	プロパティページの設定	287
7.1.1	プロパティページの [端末] タブ	287
7.1.2	プロパティページの [サーバ] タブ	288
7.2	DBPARTNER オブジェクトの詳細	290
	CodeChangeMode プロパティ	291
	CommitMode プロパティ	292
	DictPath プロパティ	293
	TableDefSource プロパティ	293
	IsConnect プロパティ	294
	LogoffFileName プロパティ	294
	LogonFileName プロパティ	295
	MacroTimeLimit プロパティ	296
	Password プロパティ	296
	ResponseLimit プロパティ	297
	TargetDatabase プロパティ	297
	TargetServer プロパティ	299

	TerminalMode プロパティ	300
	TerminalName プロパティ	300
	UserID プロパティ	301
	Cancel メソッド (CancelEx メソッド)	301
	Commit メソッド	302
	Connect メソッド	303
	Disconnect メソッド	304
	Rollback メソッド	305
	TerminalClose メソッド	305
7.3	Result オブジェクトの詳細	307
	DataCheck プロパティ	307
	EOF プロパティ	308
	FieldCount プロパティ	309
	FieldData プロパティ	309
	FieldDataLength プロパティ	310
	FieldHeader プロパティ	311
	HeaderOutput プロパティ	311
	QueryCountLimit プロパティ	312
	RecordData プロパティ	313
	RecordHeader プロパティ	313
	Close メソッド	314
	CopyToClipboard メソッド	314
	Get メソッド	315
	SaveToFile メソッド	316
7.4	TableList オブジェクトの詳細	318
	Count プロパティ	319
	CountLimit プロパティ	319
	SelectDict プロパティ	320
	Pattern プロパティ	320
	TableComment プロパティ	321
	TableCreateDate プロパティ	322
	TableDataSource プロパティ	322
	TableDBMName プロパティ	323
	TableDeleteAuth プロパティ	324
	TableGroupName プロパティ	324
	TableHeader プロパティ	325
	TableInsertAuth プロパティ	326
	TableName プロパティ	326
	TableOwner プロパティ	327
	TablePasswordMode プロパティ	328
	TableSelectAuth プロパティ	329
	TableType プロパティ	329
	TableUpdateAuth プロパティ	330
	Get メソッド	331
7.5	ColumnList オブジェクトの詳細	332

	ColumnComment プロパティ	332
	ColumnEditPattern プロパティ	333
	ColumnHeader プロパティ	334
	ColumnLength プロパティ	334
	ColumnName プロパティ	335
	ColumnNumber プロパティ	336
	ColumnScale プロパティ	336
	ColumnType プロパティ	337
	Count プロパティ	337
	CountLimit プロパティ	338
	SelectDict プロパティ	338
	TableName プロパティ	339
	TablePassword プロパティ	339
	Get メソッド	340
7.6	QueryDefine オブジェクトの詳細	341
	Exclusive プロパティ	341
	Execute メソッド	342
	QueryCount メソッド	343
	RemoveAll メソッド	343
7.7	Tables コレクションオブジェクトの詳細	344
	Count プロパティ	344
	Add メソッド	344
	Item メソッド	345
	RemoveAll メソッド	345
7.8	Table オブジェクトの詳細	347
	TableName プロパティ	347
	TablePassword プロパティ	347
7.9	Columns コレクションオブジェクトの詳細	349
	Count プロパティ	349
	Add メソッド	349
	Item メソッド	350
	RemoveAll メソッド	350
7.10	Column オブジェクトの詳細	352
	ColumnName プロパティ	352
	FunctionMode プロパティ	353
	SortMode プロパティ	353
	SortPriority プロパティ	354
7.11	Conditions コレクションオブジェクトの詳細	356
	Count プロパティ	356
	Relation プロパティ	356
	Add メソッド	357
	Item メソッド	358
	RemoveAll メソッド	359
7.12	Condition オブジェクトの詳細	360

- Text プロパティ 360
- 7.13 UpdateDefine オブジェクトの詳細 361
 - ErrorInformation プロパティ 361
 - Exclusive プロパティ 362
 - ExecutedRowCount プロパティ 363
 - TableName プロパティ 363
 - TablePassword プロパティ 364
 - UpdateMode プロパティ 364
 - Execute メソッド 365
 - RemoveAll メソッド 366
- 7.14 Updates コレクションオブジェクトの詳細 367
 - Count プロパティ 367
 - Add メソッド 367
 - Item メソッド 368
 - RemoveAll メソッド 368
- 7.15 Update オブジェクトの詳細 370
 - Column プロパティ 370
 - KeyColumn プロパティ 371
 - KeyCount プロパティ 371
 - KeyValue プロパティ 372
 - UpdateValue プロパティ 373
 - UpdateValueCount プロパティ 373
 - AddKey メソッド 374
 - AddUpdateValue メソッド 374
- 7.16 Inserts コレクションオブジェクトの詳細 376
 - Count プロパティ 376
 - Add メソッド 376
 - Item メソッド 377
 - RemoveAll メソッド 377
- 7.17 Insert オブジェクトの詳細 379
 - Column プロパティ 379
 - InsertValue プロパティ 380
 - InsertValueCount プロパティ 380
 - AddInsertValue メソッド 381
- 7.18 Deletes コレクションオブジェクトの詳細 382
 - Count プロパティ 382
 - Add メソッド 382
 - Item メソッド 383
 - RemoveAll メソッド 383
- 7.19 Delete オブジェクトの詳細 385
 - KeyColumn プロパティ 385
 - KeyCount プロパティ 386
 - KeyValue プロパティ 386
 - AddKey メソッド 387

- 7.20 Catalog オブジェクトの詳細 388
 - Convert メソッド 388
 - FileLoad メソッド 389
- 7.21 VariableList オブジェクトの詳細 390
 - Count プロパティ 390
 - VariableName プロパティ 390
 - VariableValue プロパティ 391
- 8 CommuniNet 接続を使うときの参照情報 392**
- 8.1 条件式, 表名, 列名, 配列列の形式 393
- 8.1.1 条件式を設定するときの規則 393
- 8.1.2 表名を設定するときの規則 394
- 8.1.3 列名を設定するときの規則 395
- 8.1.4 配列列を設定するときの規則 396
- 8.2 クライアント側プログラムでコード変換するときの注意 398
- 8.2.1 サーバプログラムの設定 398
- 8.2.2 空白文字に変換されるコード 399
- 8.2.3 外字の変換規則 399
- 8.2.4 1 列当たりの制限 399
- 8.3 データベースと使えるプロパティ/メソッド 400
- 8.3.1 HOAPDBS がサーバプログラムするとき 400
- 8.3.2 ACE3 E3 がサーバプログラムするとき 409
- 8.4 データベース別の規則 419
- 8.4.1 VOS3 XDM/RD E2 の場合の規則 419
- 8.4.2 VOSK RDB 編成ファイルの場合の規則 425
- 8.4.3 VOS3 XDM/SD E2 の場合の規則 430
- 8.4.4 VOS3 PDMII E2, VOS1 PDMII E2 の場合の規則 434
- 8.4.5 VOS3 RDB1 の場合の規則 439
- 8.4.6 VOS1 ACE3 ユーザファイルの場合の規則 444
- 8.4.7 VOS3 ACE3 ユーザファイルの場合の規則 450
- 8.4.8 VOS1 スプールファイルの場合の規則 456
- 8.4.9 VOSK スプールファイルの場合の規則 457

第 6 編 エラーの対処

- 9 トラブルが起こったときは 459**
- 9.1 トラブルの現象と対処 460
- 9.1.1 インストール時に起こるトラブル 460
- 9.1.2 データベースと接続するときに起こるトラブル 460
- 9.1.3 DBPARTNER2 Client の操作時に起こるトラブル 461

9.1.4 システムに関するトラブル 461

10 メッセージの一覧 462

10.1 メッセージの形式と見方 463

10.1.1 メッセージの形式 463

10.1.2 メッセージの見方 464

10.1.3 注意 464

10.2 メッセージの詳細 465

10.2.1 DBPARTNER2 Client のインストールのメッセージ 465

10.2.2 DBPARTNER ODBC Driver のメッセージ 468

付録 477

付録 A AP 作成ウィザード 478

付録 A.1 アプリケーション作成の流れ 478

付録 A.2 AP 作成ウィザードの起動と終了 479

付録 A.3 AP 作成ウィザードの操作 480

付録 A.4 作成できるアプリケーションの仕様 494

付録 B DBPARTNER2 Client の予約語 497

付録 C DBPARTNER2 Client が扱うデータの形式 499

付録 D プロパティとメソッドの制限値 500

付録 D.1 ODBC 接続のときのプロパティとメソッドの制限値 500

付録 D.2 CommuniNet 接続のときのプロパティとメソッドの制限値 505

付録 E 列の属性 511

付録 E.1 ODBC 接続のときの列の属性 511

付録 E.2 CommuniNet 接続のときの列の属性 512

付録 F 繰り返し列と配列列での制限事項 517

付録 F.1 データの抽出, 更新, 削除時の制限事項 517

付録 F.2 データの追加時の制限事項 518

付録 G このマニュアルの参考情報 520

付録 G.1 関連マニュアル 520

付録 G.2 このマニュアルで使用する略称 521

付録 G.3 略語一覧 523

付録 G.4 KB (キロバイト) などの単位表記について 525

付録 H 用語解説 526

索引 532

1

DBPARTNER2 Client の紹介

この章では、DBPARTNER2 Client の特長と機能、サーバと接続する形態について説明します。

1.1 DBPARTNER2 Client の概要

DBPARTNER2 Client とは、Windows のクライアント PC から UNIX/Windows/メインフレームのサーバのデータベースにアクセスできるようにする製品です。

DBPARTNER2 Client を使うと、クライアント PC からデータベースのデータを取り出したり、データベースのデータを更新したりできます。UNIX や Windows サーバなどで稼働する DBMS にも、メインフレーム (VOS3, VOS1, VOSK) で稼働する DBMS にも、DBPARTNER2 Client からアクセスできます。どの OS のデータベースにアクセスするときも DBPARTNER2 Client の操作は同じなので、クライアント PC でデータベースの OS を意識する必要はありません。

DBPARTNER2 Client に関連する製品を表 1-1 に示します。このマニュアルでは、DBPARTNER2 Client の ActiveX コントロール仕様について説明します。

表 1-1 DBPARTNER2 Client に関連する製品

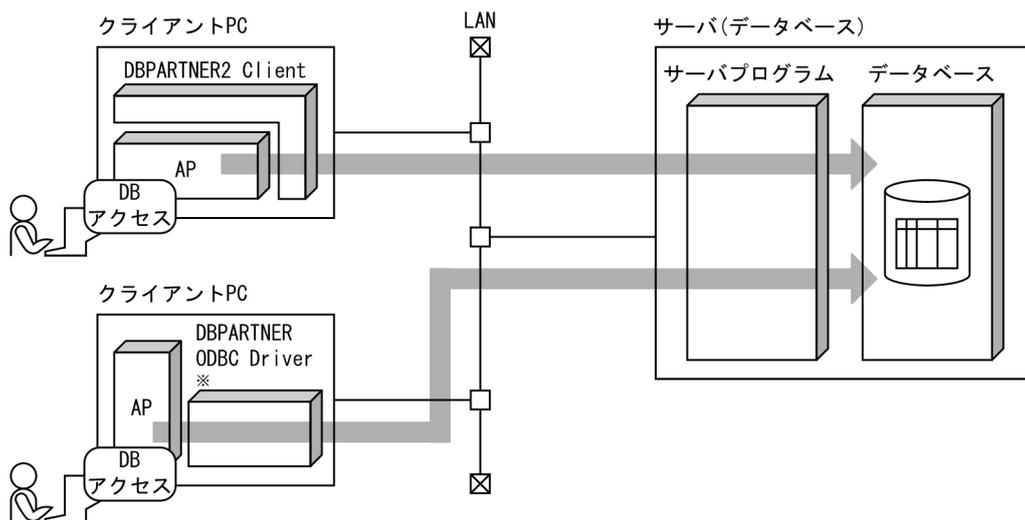
DBPARTNER2 Client の関連製品	機能
DBPARTNER2 Client ※	データベースにアクセスする、次に示す製品の総称です。
DBPARTNER2 Client	クライアント PC からデータベースに、Windows の GUI でアクセスできるようにする製品です。 Visual Basic などのビジュアル開発ツールでアプリケーションを開発できるように、ActiveX コントロールのオブジェクトも提供しています。
DBPARTNER ODBC Driver	クライアント PC からデータベースに、ODBC 接続で通信できるようにする製品です。

注※

DBPARTNER2 Client は、複数の製品を統合しています。製品は、必要に応じてクライアント PC にインストールします。DBPARTNER2 Client の各製品の概要については、[2.2 DBPARTNER2 Client のインストール] を参照してください。

DBPARTNER2 Client の関連製品を使ったデータベースへのアクセスを図 1-1 に示します。

図 1-1 DBPARTNER2 Client の関連製品を使ったデータベースへのアクセス



注※ DBPARTNER ODBC Driverを DBPARTNER2 Client以外のアプリケーションで使った形態です。

DBPARTNER2 Client からサーバのデータベースへのアクセスを仲介するプログラムのことを、**サーバプログラム**といいます。クライアント PC とサーバの接続形態とサーバプログラムの関係については、「[1.4 DBPARTNER2 Client とサーバを接続する形態](#)」を参照してください。

1.2 DBPARTNER2 Client の特長

DBPARTNER2 Client の ActiveX コントロール仕様の特長について説明します。

1.2.1 ActiveX コントロールのオブジェクトを使ってデータベースにアクセスできます

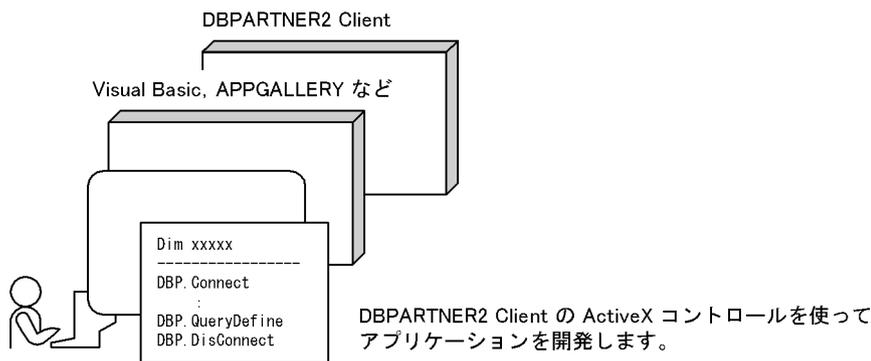
DBPARTNER2 Client は、データベースアクセスの各操作を ActiveX コントロールのオブジェクトで提供しています。このオブジェクトから Visual Basic などのビジュアル開発ツールを使ってデータベースにアクセスできるアプリケーションを開発できます。アプリケーションを作成するときは、データベースアクセスの処理を SQL 文で記述することも、SQL 文を使わないで記述することもできます。

作成したアプリケーションを操作するときは、アクセスするデータベースを意識する必要はありません。異なるデータベースにアクセスするときでも、DBPARTNER2 Client のアプリケーションでは同じユーザインタフェースで操作できます。

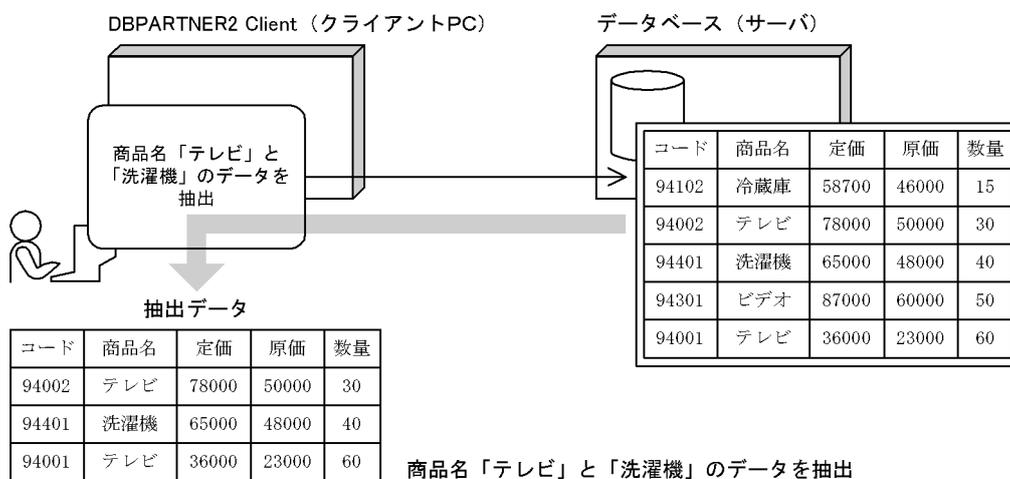
DBPARTNER2 Client を使ったデータベースアクセスの概要を図 1-2 に示します。

図 1-2 DBPARTNER2 Client を使ったデータベースアクセスの概要

- ビジュアル開発ツールで、GUI画面を作成します。



- データベースのデータを抽出します。



(1) 抽出する条件を付けて、効率良くデータを抽出できます

DBPARTNER2 Client のアプリケーションでは、次に示すようにデータを抽出できます。

条件を設定したデータの抽出 (比較条件, LIKE 条件, NULL 条件, AND, OR など)

データベースから表データを抽出するときに、指定した列だけを抽出したり、設定した条件に合う列の行だけを抽出したりできます。また、指定した複数の条件を関連付けた抽出もできます。

指定した件数までのデータの抽出

表から抽出するデータの件数 (行) を指定できます。指定した件数を超えるデータは切り捨てられます。

表名と列名を検索

表名や列名が分からないときは、接続しているデータベースに存在する表名や列名を一度表示させて、必要なものを選択して操作できるような GUI を作成できます。

(2) データベースのデータを更新できます

DBPARTNER2 Client のアプリケーションを使って、データベースのデータを更新できます。データベースの更新には、変更, 追加, 削除があります。

データの変更

データベースの表のデータを変更できます。

データの追加

データベースの表にデータを追加できます。

データの削除

データベースの表のデータを削除できます。

(3) マルチメディアデータ (BLOB 型データ) にアクセスできます

DBPARTNER2 Client を使うと、音声や画像などを使ったマルチメディアデータ (BLOB 型データ) を抽出できます。さらに、抽出したマルチメディアデータを加工して更新することもできます。DBPARTNER2 Client でマルチメディアデータを扱うときは、DBPARTNER2 Client のオブジェクトのプロパティやメソッドに BLOB 型データ用の設定をします。

アクセスするデータベースやサーバプログラムによっては、BLOB 型データを扱えないことがあります。BLOB 型データを使えるかどうかについては、アクセスするデータベースのマニュアルを参照してください。

(4) アプリケーション開発にビジュアル開発ツールを使えます

DBPARTNER2 Client のアプリケーションは、各種のビジュアル開発ツールを使って作成できます。次に示す製品で、動作を確認済みです。

- Microsoft Visual Basic 4.0 (32 ビット版), Microsoft Visual Basic 5.0, Microsoft Visual Basic 6.0
- APPGALLERY 4.0
- Power Builder 5.0

上記以外のビジュアル開発ツールを使うときは、別途確認が必要です。

(5) サンプルを提供しています

DBPARTNER2 Client は、すぐにアプリケーションの動作を体験できるように、サンプル (Visual Basic 4.0 32 ビット版対応) を提供しています。サンプルを使ってアプリケーションを作成したり、サンプルを業務に合わせた処理に修正したりすれば、DBPARTNER2 Client のアプリケーションを最初から作成する手間が省けます。各サンプルについては、Readme.txt を参照してください。サンプルを格納しているフォルダについては、マニュアル「DBPARTNER2 Client 操作ガイド」を参照してください。

(6) AP 作成ウィザードで簡単にアプリケーションが作成できます。

AP 作成ウィザードは、ActiveX コントロールを使用したアプリケーションを開発するためのプログラムです。AP 作成ウィザードを使用すると、データベースからデータを抽出するアプリケーションが簡単に作成できます。AP 作成ウィザードについては、「付録 A AP 作成ウィザード」を参照してください。

1.2.2 ODBC に対応したデータベースにアクセスできます

DBPARTNER2 Client は、ODBC に対応しています。DBPARTNER2 では、日立データベース用の各種 ODBC ドライバを提供しています。ODBC 対応の表計算ソフトからデータベースにアクセスするときは DBPARTNER2 Client の ODBC ドライバだけを経由するので、それぞれの表計算ソフトのユーザインタフェースをそのまま使えます。

ODBC ドライバを使うときは、DBPARTNER ODBC Driver をクライアント PC に組み込みます。DBPARTNER ODBC Driver は、DBPARTNER2 Client と一緒に使うことも、DBPARTNER2 Client 以外のアプリケーションで使うこともできます。

(1) ODBC ドライバの種類

DBPARTNER2 Client が提供する ODBC ドライバには、次のドライバがあります。

- DABroker ドライバ
(DABroker, Database Connection Server 接続用)

DBPARTNER2 Client の ODBC ドライバを使ってアクセスできるデータベースについては、「[1.4 DBPARTNER2 Client とサーバを接続する形態](#)」を参照してください。

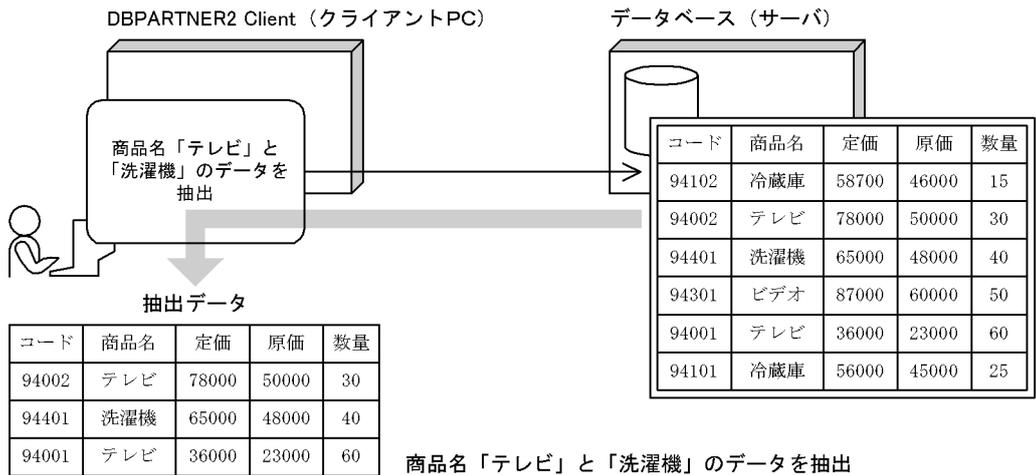
1.3 DBPARTNER2 Client の機能

DBPARTNER2 Client の ActiveX コントロールを使ってアプリケーションを作成すると、データベースから指定した条件に合ったデータを読み込みます（データの抽出）。また、抽出したデータを編集してデータベースを更新できます（データの更新）。

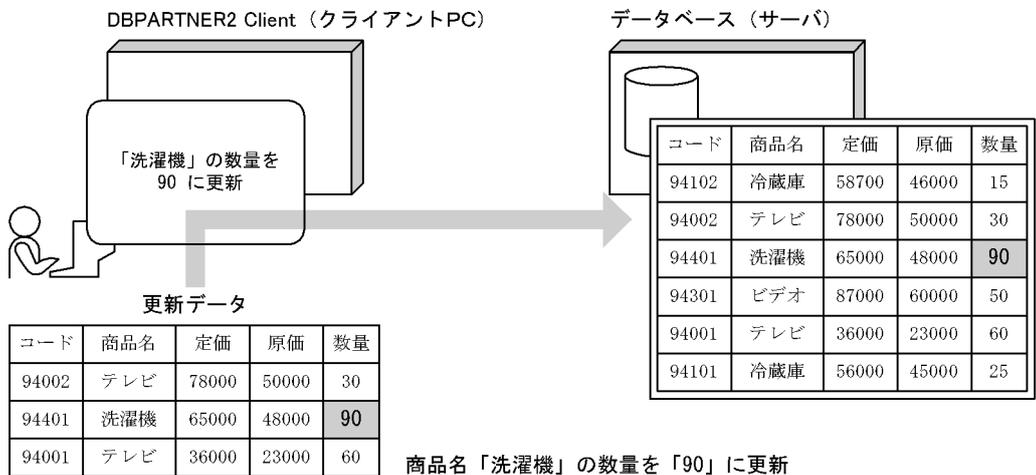
DBPARTNER2 Client を使ったデータの抽出／更新の概要を図 1-3 に示します。

図 1-3 DBPARTNER2 Client を使ったデータの抽出／更新の概要

● データの抽出



● データの更新



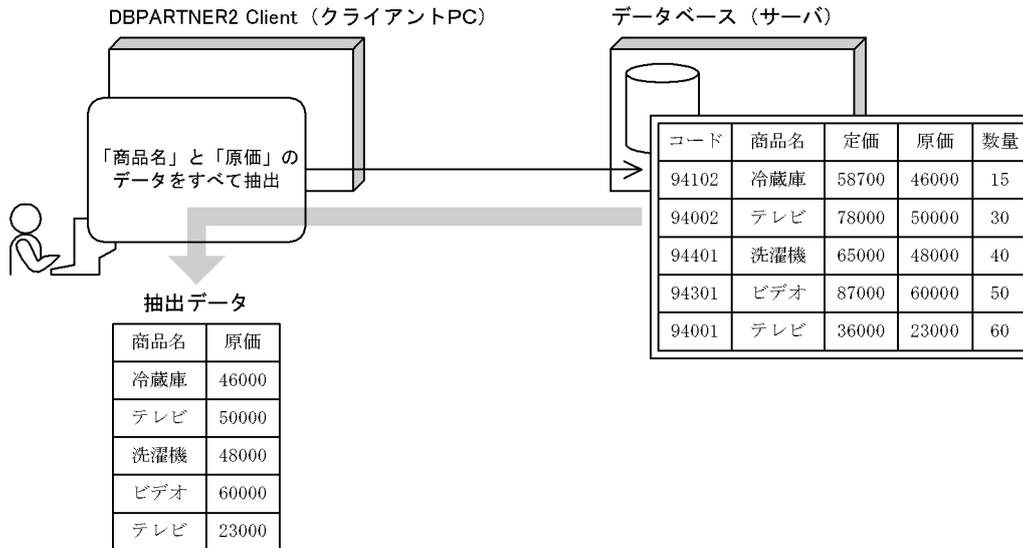
1.3.1 データの抽出

クライアント PC のアプリケーションから、データベースにアクセスしてデータを抽出できます。このとき、複数の表を結合してデータを抽出することもできます。さらに、データベース上の表からデータを抽出するときに特定した列だけを抽出したり、条件に合うデータだけを抽出したりできます。

DBPARTNER2 Client を使ったデータの抽出を図 1-4 に、抽出で使える機能の一覧を表 1-2 に示します。

図 1-4 DBPARTNER2 Client を使ったデータの抽出

● 列を指定して抽出



● 条件を指定して抽出

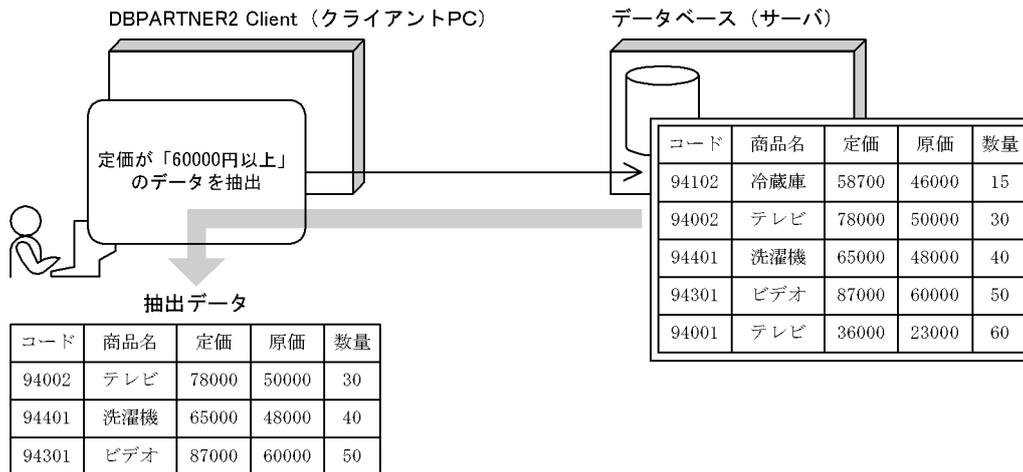


表 1-2 抽出で使える機能の一覧

抽出で使える機能	説明
複数の表を結合して抽出する	表と表で共通する列を結び付けて、一つの表として抽出できます（表結合）。表結合を使うと、関連する表のデータを一括して利用できます。
表を取得するときに絞り込みをする	表を選ぶときに、条件を付けて絞り込みます。抽出対象の表が大量にある場合などに、効率良く表を選べます。
特定の列だけを抽出する	抽出する表から特定の列だけを選んで抽出できます。選択した列のデータをソートしたり、列のデータの最大値、最小値などを抽出したりすることもできます。
条件に合うデータを抽出する	抽出する表から条件に合うデータだけを抽出できます。また、複数の条件式を関連付けることもできます。例えば、定価が 50,000 円以上のデータを抽出する条件と、商品名が冷蔵庫又はテレビのデータを抽出する条件を「AND」で関連付けます。この場合、定価が 50,000 円以上でかつ、商品名が冷蔵庫又はテレビのデータだけを抽出できます。

抽出で使える機能	説明
特定の列で、条件に合うデータを抽出する	列と条件の両方を設定して、データを抽出できます。指定した列のデータのうち、条件に合うデータだけを抽出できます。
抽出するデータの件数を制限する	データを抽出するときに、その件数を指定できます。抽出するデータをソートして上位10件だけを抽出したり、表計算ソフトに表示できるデータ量を考慮して、抽出する件数を制限したりするときなどに使います。

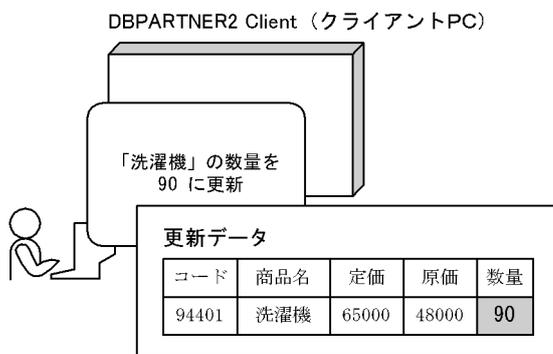
1.3.2 データの更新

クライアント PC のアプリケーションから、データベースにアクセスしてデータを更新（変更、追加、削除）できます。

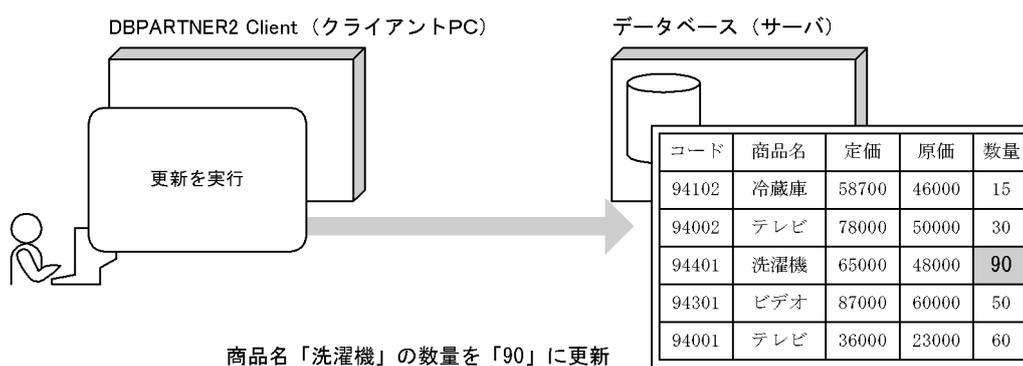
DBPARTNER2 Client を使ったデータの更新を図 1-5 に示します。

図 1-5 DBPARTNER2 Client を使ったデータの更新

- 更新データを作成します



- 更新データをデータベースに反映します



1.4 DBPARTNER2 Client とサーバを接続する形態

DBPARTNER2 Client とサーバを接続する形態について説明します。DBPARTNER2 Client とサーバとの接続形態には、次に示す 2 通りがあります。

- ODBC に準拠した接続形態（ODBC 接続）
- CommuniNet を使った接続形態（CommuniNet 接続）

一つのクライアント PC に複数の DBPARTNER2 Client の製品（例：DBPARTNER2 Client と DBPARTNER/Client32）を組み込んで、ODBC 接続と CommuniNet 接続と組み合わせて使うこともできます。

クライアント PC 上で DBPARTNER2 Client とほかの製品を連携して使うときは、すべて 32 ビット対応のソフトウェアを使ってください。

1.4.1 DBPARTNER2 Client で使えるサーバプログラム

DBPARTNER2 Client からデータベースにアクセスするときは、仲介の役割をするプログラムが必要です。このプログラムをサーバプログラムといいます。DBPARTNER2 Client で使えるサーバプログラムを次に示します。

(1) ODBC 接続で使うサーバプログラム

DABroker

分散オブジェクト環境に準拠したサーバプログラムです。Windows 又は UNIX で稼働します。

Database Connection Server

メインフレーム上で稼働するサーバプログラムです。

XDM/DF

VOS3 で稼働するサーバプログラムです。

DF1

VOS1 で稼働するサーバプログラムです。

DF/K

VOSK で稼働するサーバプログラムです。

(2) CommuniNet 接続で使うサーバプログラム

HOAPDBS E2, ACE3 E3

VOS3 で稼働するサーバプログラムです。

DBPARTNER/Link

VOS1 で稼働するサーバプログラムです。

HOAPDBS/PC

VOSK で稼働するサーバプログラムです。

1.4.2 ODBC ドライバの概要

ODBC 接続でデータベースにアクセスする場合は、それに対応する ODBC ドライバ (DBPARTNER ODBC Driver) を用意する必要があります。

DBPARTNER2 Client で同梱している DBPARTNER ODBC Driver では、次に示す ODBC ドライバを提供しています。

- DABroker ドライバ (DBPARTNER/Server ドライバ)

それぞれの ODBC ドライバは、アクセスするデータベースによって使い分けます。

DBPARTNER ODBC Driver のほかに DBPARTNER2 Client で接続を確認している ODBC ドライバを次に示します。

- DBPARTNER ODBC 3.0 Driver
- HiRDB ODBC Driver

(1) ODBC ドライバを使ってアクセスできるデータベース

DBPARTNER2 Client の ODBC ドライバを使ってアクセスできるデータベースを表 1-3 に示します。

表 1-3 DABroker ドライバ (DBPARTNER/Server ドライバ) の場合

サーバプログラム	アクセスできるデータベース
DABroker	HiRDB *
	Oracle
	Microsoft SQL Server
	Sybase Adaptive Server Anywhere
	VOS3 XDM/RD E2
	VOS3 XDM/SD E2
	VOS1 RDB1 E2
	VOSK RDB 編成ファイル
Database Connection Server	VOS3 XDM/RD E2
	VOS3 XDM/SD E2

サーバプログラム	アクセスできるデータベース
	VOS1 PDMII E2
	VOSK RDB 編成ファイル
	VOSK スプールファイル

注※

HiRDB のデータベース管理で扱えるデータは、接続するサーバプログラムによって決まります。

(2) DBPARTNER2 Client が稼働する環境 (ODBC 接続)

DBPARTNER2 Client のクライアント PC に必要なソフトウェアとドライバ、サーバ側で必要なソフトウェアの関係を表 1-4 に示します。

表 1-4 DBPARTNER2 Client が稼働する環境 (ODBC 接続)

データベース	サーバプログラム	ODBC ドライバ	クライアント PC 側の環境
HiRDB	DABroker	DABroker ドライバ (DBPARTNER/Server ドライバ)	—
Oracle	DABroker	DABroker ドライバ (DBPARTNER/Server ドライバ)	—
Microsoft SQL Server, Sybase Adaptive Server Anywhere	DABroker	DABroker ドライバ (DBPARTNER/Server ドライバ)	—
VOS3 XDM/RD E2	DABroker ※	DABroker ドライバ (DBPARTNER/Server ドライバ)	—
	Database Connection Server		
VOS3 XDM/SD E2	DABroker ※	DABroker ドライバ (DBPARTNER/Server ドライバ)	—
	Database Connection Server		
VOS1 RDB1 E2	DABroker ※	DABroker ドライバ (DBPARTNER/Server ドライバ)	—
	Database Connection Server		
VOS1 PDMII E2	DABroker ※	DABroker ドライバ (DBPARTNER/Server ドライバ)	—
	Database Connection Server		
VOSK RDB 編成ファイル	DABroker ※	DABroker ドライバ (DBPARTNER/Server ドライバ)	—
	Database Connection Server		
VOSK スプールファイル	DABroker ※	DABroker ドライバ	—

データベース	サーバプログラム	ODBC ドライバ	クライアント PC 側の環境
	Database Connection Server	(DBPARTNER/Server ドライバ)	

注※

メインフレーム系データベースにアクセスするときは、RDA Link for Gateway、又は Database Connection Server が必要です。DABroker に必要なソフトウェアについては、マニュアル「DABroker」を参照してください。

(3) 使えるデータベースと表の数

1 回のデータ抽出処理で処理対象にできるデータベースの個数は 1 個です。つまり、複数のデータベースから同時にデータを抽出できません。

1 個のデータベースから使える表の最大数は、データベースによって異なります。また、権限によって処理対象にならない表があります。権限については、システム管理者に確認してください。

(4) データの形式

DBPARTNER2 Client で扱える形式のデータを表 1-5 に示します。データの形式については、「付録 C DBPARTNER2 Client が扱うデータの形式」を参照してください。

表 1-5 転送時のデータ形式

形式	拡張子	出力先		利用できるソフトウェア
		ファイル	クリップボード	
CSV 形式	.csv	○	×	表計算ソフト
DAT 形式	.dat	○	×	表計算ソフト
TEXT 形式	.txt	○	○	表計算ソフト
XLS 形式	.xls	○	○	表計算ソフト (Excel)
XLSB 形式	.xlsb	○	×	表計算ソフト (Excel 2007 以降)
XLSX 形式	.xlsx	○	×	表計算ソフト (Excel 2007 以降)

(凡例)

- ：出力できます。
- ×：出力できません。

1.4.3 CommuniNet 接続の概要

CommuniNet 接続でデータベースにアクセスするときは、DBPARTNER2 Client の PC に CommuniNet が必要です。

DBPARTNER2 Client の ActiveX コントロールで CommuniNet 接続を使うときに必要なプログラムを次に示します。

- CommuniNet Version 3.0 以降
- CommuniNet Extension Version 3.0 以降

さらに、上記の CommuniNet が稼働するために必要なソフトウェアがあります。CommuniNet に関連する製品については、CommuniNet の該当するマニュアルを参照してください。

(1) CommuniNet 接続でアクセスできるデータベース

CommuniNet 接続でアクセスできるデータベースを表 1-6 に示します。

表 1-6 CommuniNet 接続でアクセスできるデータベース

サーバの OS	サーバプログラム	アクセスできるデータベース
VOS3	HOAPDBS E2	XDM/RD E2, XDM/SD E2
	ACE3 E3	PDMII E2, RDB1, ACE3 ユーザファイル
VOSK	HOAPDBS/PC	RDB 編成ファイル, VOSK スプールファイル
VOS1	DBPARTNER/Link	PDMII E2, ACE3 ユーザファイル, VOS1 スプールファイル

(2) DBPARTNER2 Client が稼働する環境 (CommuniNet 接続)

DBPARTNER2 Client のクライアント PC に必要なソフトウェア、サーバ側に必要なソフトウェアの関係を表 1-7 に示します。

表 1-7 DBPARTNER2 Client が稼働する環境 (CommuniNet 接続)

データベース	サーバプログラム	クライアント PC 側の環境
VOS3 XDM/RD E2	HOAPDBS E2	CommuniNet, CommuniNet Extension
VOS3 XDM/SD E2		
VOS3 PDMII E2	ACE3 E3	
VOS3 RDB1		
VOS3 ACE3 ユーザファイル		
VOSK RDB 編成ファイル	HOAPDBS/PC	
VOSK スプールファイル		
VOS1 PDMII E2	DBPARTNER/Link	
VOS1 ACE3 ユーザファイル		
VOS1 スプールファイル		

(3) アクセスするデータベースの制限

CommuniNet 接続のときには、DBPARTNER2 Client からアクセスするデータベースによって、使える機能に制限があります。データベースごとの制限については、「8. CommuniNet 接続を使うときの参照情報」の説明を参照してください。

(4) 使えるデータベースと表の数

1 回のデータ抽出処理で処理対象にできるデータベースの個数は 1 個です。つまり、複数のデータベースから同時にデータを抽出できません。

1 個のデータベースから使える表の最大数は、データベースによって異なります。また、権限によって処理対象にならない表があります。権限については、システム管理者に確認してください。

(5) データの形式

DBPARTNER2 Client で扱える形式のデータを表 1-8 に示します。データの形式については、「付録 C DBPARTNER2 Client が扱うデータの形式」を参照してください。

表 1-8 転送時のデータ形式

形式	拡張子	出力先		利用できるソフトウェア
		ファイル	クリップボード	
CSV 形式	.csv	○	×	表計算ソフト
DAT 形式	.dat	○	×	表計算ソフト
TEXT 形式	.txt	○	○	表計算ソフト
XLS 形式	.xls	○	○	表計算ソフト (Excel)
XLSB 形式	.xlsb	○	×	表計算ソフト (Excel 2007 以降)
XLSX 形式	.xlsx	○	×	表計算ソフト (Excel 2007 以降)

(凡例)

- ：出力できます。
- ×：出力できません。

2

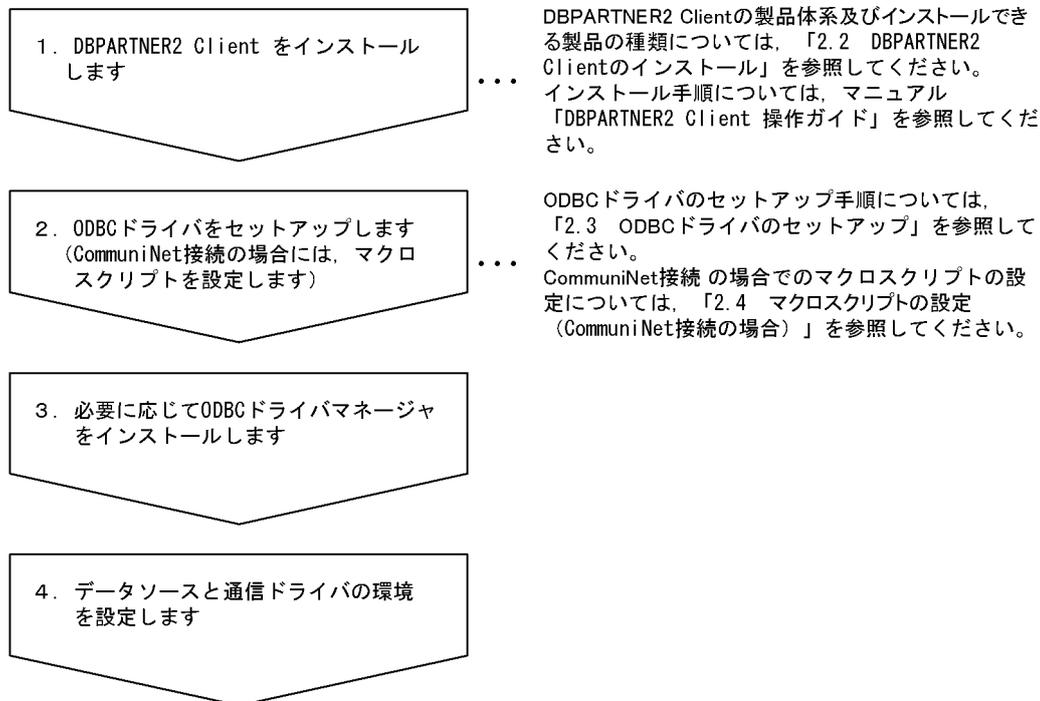
環境の設定

この章では、DBPARTNER2 Client の実行環境を設定する方法について説明します。

2.1 DBPARTNER2 Client の環境を設定する手順

DBPARTNER2 Client の環境を設定する手順について説明します。環境を設定する手順を図 2-1 に示します。

図 2-1 環境を設定する手順



1. DBPARTNER2 Client をセットアップします

DBPARTNER2 Client をクライアント PC に組み込みます。

2. ODBC ドライバ又は CommuniNet をセットアップします

ODBC 接続のとき

接続するサーバに応じた ODBC ドライバを、次に示すどちらかから選んでインストールします。

- DABroker ドライバ (DBPARTNER/Server ドライバ)

HiRDB, ORACLE, Microsoft SQL Server, Sybase Adaptive Server Anywhere, VOS3 XDM/RD E2, VOS3 XDM/SD E2, VOS1 RDB1 E2, VOSK RDB 編成ファイル, VOS1 PDMII E2, VOSK スプールファイル

CommuniNet 接続のとき

CommuniNet をセットアップします。

3. 必要に応じて ODBC ドライバマネージャをインストールします

ODBC ドライバマネージャのインストールについては、「2.3.1 ODBC ドライバのインストールの手順」を参照してください。

4. データソースと通信ドライバの環境を設定します

ODBC ドライバのとき

DABroker ドライバ (DBPARTNER/Server ドライバ) を使うときは、2. でインストールした ODBC ドライバのデータソース名と通信ドライバに関する情報を指定します。

DBPARTNER2 Client と一緒に ODBC ドライバを使うときは、ODBC ドライバで指定するデータソース名と DBPARTNER オブジェクトに設定するデータソース名は同じ名前にしてください。

CommuniNet 接続のとき

CommuniNet の環境を設定して、接続するサーバの環境に応じたマクロスクリプトファイルを作成します。

CommuniNet のマクロスクリプトファイル名は、CommuniNet 接続の DBPARTNER オブジェクトに設定します。

2.2 DBPARTNER2 Client のインストール

インストールできる DBPARTNER2 Client の製品の種類について説明します。インストールの手順については、マニュアル「DBPARTNER2 Client 操作ガイド」を参照してください。

2.2.1 DBPARTNER2 Client の製品の種類

DBPARTNER2 Client には、参照版、更新版、定義版があります。業務で使う DBPARTNER2 Client を選んでインストールしてください。

(1) DBPARTNER2 Client の製品体系

DBPARTNER2 Client は、複数の製品を同梱しています。DBPARTNER2 Client の製品名と機能の一覧を表 2-1 に示します。

表 2-1 DBPARTNER2 Client の製品名と機能の一覧

DBPARTNER2 Client の製品名	機能
DBPARTNER2 Client	クライアント PC からデータベースに、Windows の GUI でアクセスできるようにする製品です。 Visual Basic などのビジュアル開発ツールでアプリケーションを開発できるように、ActiveX コントロールのオブジェクトも提供しています。
DBPARTNER ODBC Driver	クライアント PC からデータベースに、ODBC 接続で通信できるようにする製品です。ODBC 接続を使うときは、必ずクライアント PC に組み込みます。 DBPARTNER ODBC Driver は、DBPARTNER2 Client と一緒に使うことも、DBPARTNER2 Client 以外のアプリケーションで使うこともできます。

(2) DBPARTNER2 Client の製品の種類と使える機能の関係

DBPARTNER2 Client の製品の種類（参照版、更新版、定義版）と使える機能の関係を表 2-2 に示します。

SQLDirect オブジェクトで SQL を直接実行するときは、表 2-2 に示す制限はありません。

表 2-2 DBPARTNER2 Client の製品と使える機能の関係

DBPARTNER2 Client の製品	デザイン ※1		実行 ※2	
	QueryDefine オブジェクト	UpdateDefine オブジェクト	QueryDefine オブジェクト	UpdateDefine オブジェクト
DBPARTNER2 Client (参照版)	○	○	○	—
DBPARTNER2 Client (更新版)	○	○	○	○
DBPARTNER2 Client (定義版)	○	○	○	○

(凡例)

○：機能を使えます。

－：機能を使えません（エラーになります）。

注※1

デバッグを実行するときは、実行時と同じ制限があります。

注※2

デザインした DBPARTNER2 Client のアプリケーションを稼働するクライアント PC の DBPARTNER2 Client を示します。

2.3 ODBC ドライバのセットアップ

DBPARTNER2 Client の ODBC ドライバ (DBPARTNER ODBC Driver) をセットアップする方法について説明します。DBPARTNER ODBC Driver は、DBPARTNER2 Client と一緒に使うことも、DBPARTNER2 Client 以外のアプリケーションで使うこともできます。

DBPARTNER ODBC Driver では、次に示す ODBC ドライバが使えます。

- DABroker ドライバ (DBPARTNER/Server ドライバ)

2.3.1 ODBC ドライバのインストールの手順

ODBC ドライバは、DBPARTNER ODBC Driver の提供媒体の **Setup.exe** を実行してインストールします。

提供媒体の Setup.exe を実行して ODBC ドライバをインストールする手順を次に示します。

1. ODBC ドライバの提供媒体をドライブにセットします。
2. 1. でセットした ODBC ドライバの提供媒体の Setup.exe を実行します。Setup.exe 実行後は、表示される画面に従ってインストールしてください。
3. ODBC ドライバ、及び ODBC ドライバマネージャのインストール完了後、データソースをセットアップします。
データソースのセットアップは、「[2.3.3 DABroker ドライバのデータソースと通信ドライバの設定](#)」で示す手順に従って操作してください。

2.3.2 ODBC ドライバをインストールすると作成されるファイル

ODBC ドライバのインストール時の選択によって作成されるファイルについて説明します。

(1) DABroker ドライバを選択すると作成されるファイル

DABroker ドライバ (DBPARTNER/Server ドライバ) を選択すると作成されるファイルの一覧を表 2-3 に示します。

表 2-3 DABroker ドライバ (DBPARTNER/Server ドライバ) を選択すると作成されるファイルの一覧

ファイル名	ファイルの内容	格納先
Readme.txt	はじめにお読みください	インストール先ディレクトリ
Dbpsvd32.dll	ODBC ドライバ	Windows のフォルダ¥System

ファイル名	ファイルの内容	格納先
DbpWSC32.dll	通信部(Windows Sockets 用)	Windows のフォルダ¥System
Dbpsvl32.dll	共通サービス機能	Windows のフォルダ¥System
Dbpsvs32.dll	ODBC ドライバのセットアップ機能	Windows のフォルダ¥System
Dbpdfh32.hlp	ヘルプ	Windows のフォルダ¥System

2.3.3 DABroker ドライバのデータソースと通信ドライバの設定

DABroker ドライバ (DBPARTNER/Server ドライバ) は、DABroker を経由してデータベースにアクセスするときに必要です。

DABroker ドライバをインストールして環境を設定する方法について説明します。

1. データソースのセットアップ

接続先サーバのセンタ名称を「データソース名」として ODBC ドライバに登録します。また、接続先サーバのホスト名称、使う通信ドライバ、応答監視時間を設定します。使える通信ドライバは「Windows Sockets」だけです。

2. 通信ドライバの環境設定

DABroker ドライバ (DBPARTNER/Server ドライバ) は、TCP/IP を使用してサーバプログラムと通信しています。そのため、TCP/IP に関する通信ドライバの環境を、次の手順に従って設定してください。

- **ホスト名称とインターネットアドレスの登録**

hosts ファイルに、接続するサーバのホスト名称とインターネットアドレスを登録します。ホスト名称には、[DBPARTNER ODBC Driver for DABroker-セットアップ-] ダイアログボックスで指定するホスト名称を指定します。ホスト名称の登録方法については、通信ドライバのマニュアルを参照してください。

- **ポート番号の登録**

services ファイルに、接続するサーバのサービス名、ポート番号および通信プロトコル種別を登録します。

ポート番号に登録する内容を次に示します。

サービス名 : DBPARTNER_SV

ポート番号 : XXXXX

通信プロトコル種別 : tcp

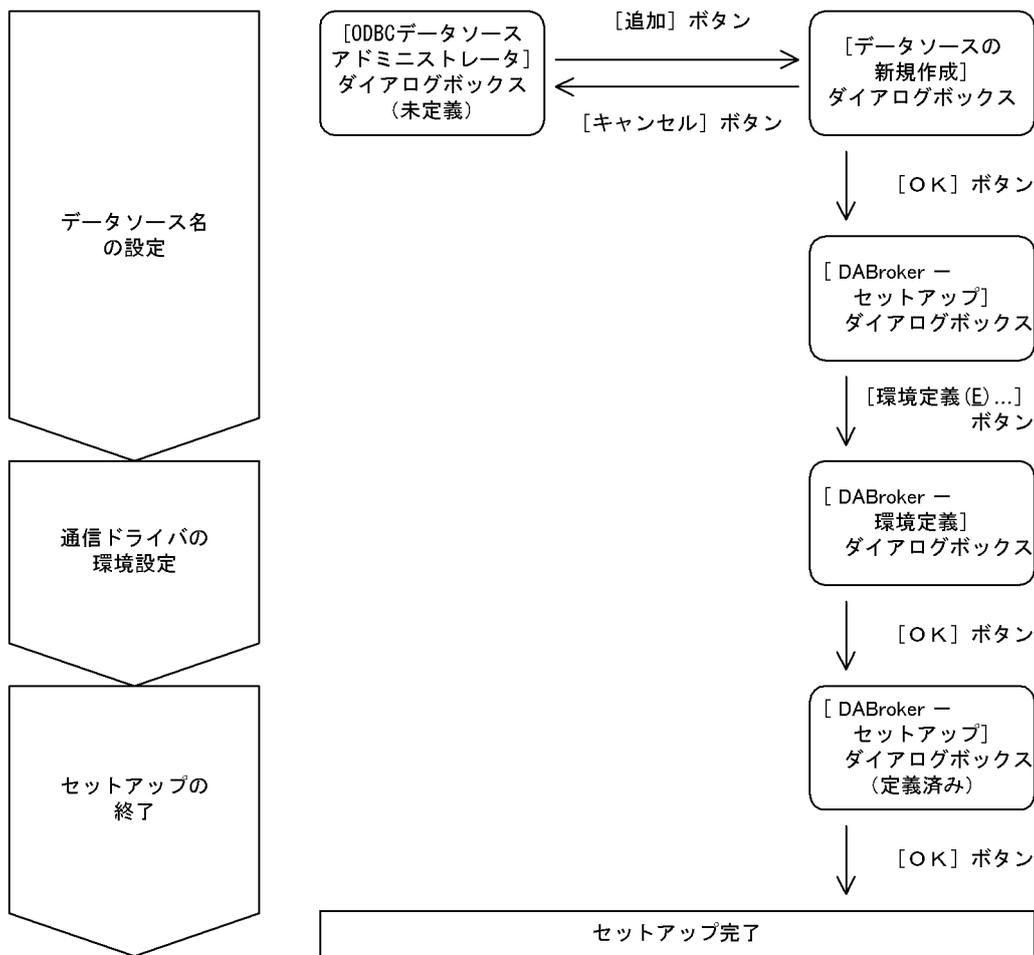
(登録例) DBPARTNER_SV 40179/tcp

XXXXX : サーバ側の services ファイルに定義されている DABROKER_SV の値と一致させます。

DABROKER_SV の値が 40179 の場合、DBPARTNER_SV の登録を省略できます。

DABroker ドライバ (DBPARTNER/Server ドライバ) のセットアップ手順を図 2-2 に示します。

図 2-2 DABroker ドライバ (DBPARTNER/Server ドライバ) のセットアップ手順



(1) データソースのセットアップ (新規追加)

DABroker ドライバ (DBPARTNER/Server ドライバ) を使うときに、データソースを新規で追加する手順を説明します。

データソースをセットアップするときは、ODBC 接続を定義する画面 ([ODBC データソースアドミニストレータ] ダイアログボックス) を開いてください。

(a) [DBPARTNER ODBC Driver for DABroker -セットアップ-] ダイアログボックスに項目を設定

サーバにアクセスするために、接続先のサーバのセンタ名称を「データソース名」として ODBC ドライバに追加します。同時に接続先の環境を設定します。

1. [ODBC データソースアドミニストレータ] ダイアログボックスの [追加(D)...] ボタンをクリックします。[データソースの新規作成] ダイアログボックスが表示されます。
2. 使う通信ドライバに応じて ODBC ドライバの中から選んでから [OK] ボタンをクリックします。ボタンをクリックすると、[DBPARTNER ODBC Driver for DABroker -セットアップ-] ダイアログボックスが表示されます。



[DBPARTNER ODBC Driver for DABroker-セットアップ] ダイアログボックスに設定する項目を次に示します。

- [データソース名] テキストボックス ～ 〈1～32 けたの文字列〉

データソースを識別するために任意の名称を指定します。データソース名は省略できません。

(例) 東京センタ

このデータソース名は、接続するサーバをユーザが認識できるようにするための名称です。ユーザが認識できればどんな名称を指定してもかまいません。すべて全角文字（シフト JIS コード）で入力するときは 16 文字まで、すべて半角文字（JIS8 コード）で入力するときは 32 文字まで入力できます。全角文字と半角文字が混在してもかまいません。

- [ホスト名称] テキストボックス ～ 〈1～32 けたの文字列〉

使う通信ドライバに登録されている接続先システムのインターネットアドレスに対応するホスト名称を指定します。ホスト名称は省略できません。

- [接続データベース] テキストボックス ～ 〈1～32 けたの文字列〉

接続するデータベースの種別を指定します。次に示す項目から選んで指定してください。

HiRDB : HiRDB を示します。

Oracle : ORACLE を示します。

SQL Server : Microsoft SQL Server を示します。

SQL Anywhere : Sybase Adaptive Server Anywhere を示します。

XDM/RD : XDM/RD E2 を示します。

RDB1 E2 : RDB1 E2 を示します。

SQL/K : RDB 編成ファイルを示します。

XDM/SD E2 : XDM/SD E2 を示します。

PDM2 : PDMII E2 を示します。

SQL/K SPOOL : VOSK スプールファイルを示します。

- [データベース識別子] テキストボックス ～ 〈文字列〉

- データベースが HiRDB 又は ORACLE の場合

サーバに複数のデータベースが存在するときに、接続するデータベース識別子を指定します。

HiRDB の場合

接続する HiRDB システムのポート番号を指定します。データベースが一つで PC サーバに HiRDB のポート番号が設定してあるときは、指定しなくても接続できます。省略すると、PDNAMEPORT の値が仮定されます。

ORACLE の場合

ネットサービス名を指定します。データベースが一つしかないときは、指定は不要です。省略すると、ORACLE_SID の値が仮定されます。

- データベースが Microsoft SQL Server, Sybase Adaptive Server Anywhere の場合

DABroker がインストールされているマシンのシステム DSN に設定したデータソース名を指定します。

- サーバプログラムが DABroker 経由-RDA Link for Gateway で、データベースが XDM/RD E2, RDB1 E2, SQL/K の場合

RDA Link for Gateway で定義されている、RD ノード名称を指定します。

- サーバプログラムが DABroker 経由-Database Connection Server で、データベースが XDM/RD E2, RDB1 E2, SQL/K, XDM/SD E2, PDMII E2 の場合

指定しません。指定しても、指定した内容は無視されます。

- サーバプログラムが Database Connection Server で、データベースが XDM/RD E2, RDB1 E2, SQL/K, XDM/SD E2, PDMII E2 の場合

Database Connection Server で定義されている、データベース識別子を指定します。

- データベースが VOSK スプールファイルの場合

指定しません。指定した場合、指定した内容は無視されます。

- [データベースホスト名] テキストボックス ～ 〈文字列〉

接続するデータベースが HiRDB のとき、HiRDB のホスト名 (PDHOST) を指定します。接続するデータベースが HiRDB 以外の場合は不活性となります。

- [送受信領域サイズ] テキストボックス ～ 〈符号なし整数〉 ((1~16,000)) (単位: キロバイト)

データ送受信時に DABroker ドライバが使うデータ受信用領域サイズを、1~16,000kB の範囲で指定します。省略すると、64kB が仮定されます。

指定した値が 1~16,000 の範囲外の場合は、次に示す領域サイズを仮定して処理します。

- 指定値 < 1 のとき: 64
- 指定値 > 16,000 のとき: 16,000

送受信領域サイズは、**注意**に示す理由で、取り扱うデータが BLOB などの長大なデータのとき以外は指定しないことをお勧めします。

注意

1. DABroker ドライバでは、指定したサイズの送受信領域を使う SQL ステートメントごとに送信用、受信用それぞれ 1 個ずつ確保します (1 ステートメントごとの指定サイズ×2 のメモリを使います)。使うマシンの性能を考慮して、適正な値を指定してください。

2. DABroker ドライバでは、指定したサイズの受信領域に格納できる行数分の結果データを、DABroker から一度に受け取って処理します。そのため、受信領域サイズが大きいと、一度に取得できるデータ数は多くなって、通信の発生頻度は減ります。その反面、受信領域サイズの指定が大き過ぎるときは、メモリエラーになって処理が続けられない、又はメモリの圧迫による性能劣化、さらに 1 回のデータ通信量の増大によってバッファ境界での Fetch 性能が劣化するおそれがあります (ただし、結果的には通信オーバーヘッドは変わりません)。この項目を指定するときは、上限 1,000 kB (1MB) 程度に抑えて、マシンの性能を考慮した適正な値に設定してください。受信領域サイズ指定の目安を次に示します。

受信領域長 = (((行データに含まれる列データ長の合計 + (4 × 行データの列数)) + 4) × 一回の通信で取得する行数) + 42

- **[DABroker 経由-Database Connection Server 接続] 欄**

DABroker 経由-Database Connection Server 接続を使用するかどうかを指定します。

DABroker 経由-Database Connection Server 接続を使用する場合、**[使用する] チェックボックス** をオンにします。

[データベース種別名] テキストボックス ~ 〈文字列〉

DABroker の接続先データベース定義で定義されたデータベース種別名を指定します。

[データベース名] テキストボックス ~ 〈文字列〉

DABroker の接続先データベース定義で定義されたデータベース名を指定します。

- **[ユーザ外字] 欄**

自 PC で作成した外字とサーバのデータベースで作成した外字を対応付けるかどうかを指定します。

サーバのシフト JIS コード又は EUC コードにユーザ外字を使って ODBC ドライバで外字コードを変換するときは、**[使用する] チェックボックス** をオンにして、**[選択(S)...] ボタン** をクリックしてからユーザ外字変換ファイルを選びます。ユーザ外字変換ファイルの作成方法については、「[2.3.3 \(5\) ユーザ外字変換ファイルの作成方法](#)」を参照してください。

ユーザ外字を使う指定をすると、ODBC ドライバはユーザ外字変換ファイルの内容に従って、外字コードを変換します。

- **[EUC-SJIS 変換を行わない] チェックボックス**

EUC-SJIS 変換をするかどうかを指定します。

サーバ側の文字コードが EUC コードの場合で ODBC アプリケーションが EUC コード処理をする場合、この項目をオンにします。

この項目をオンにすると、すべての EUC-SJIS 変換が行われません。また、ユーザ外字の使用の指定も無視されます。

- **[Oracle 接続で SQLDescribeParam を使用] チェックボックス**

接続するデータベースが ORACLE の場合、通常 SQLDescribeParam は使用できませんが、このチェックボックスを指定すると、すべてのパラメタの SQL データ型が SQL_VARCHAR 型で出力されます。

デフォルトはチェックなしで、接続するデータベースが ORACLE 以外の場合はグレー表示になります。

- **[検索データの余分な空白文字を削除] チェックボックス**

検索するデータの後ろに含まれる空白文字を削除するかどうかを指定します。削除するときは、[検索データの余分な空白文字を削除] チェックボックスをオンにしてください。

デフォルトはチェックなしです。

(b) [DBPARTNER ODBC Driver for DABroker –環境定義–] ダイアログボックスに項目を設定

[DBPARTNER ODBC Driver for DABroker –セットアップ–] ダイアログボックスの項目を設定した後、[環境定義(E)...] ボタンをクリックします。ボタンをクリックすると、[DBPARTNER ODBC Driver for DABroker –環境定義–] ダイアログボックスが表示されます。



[DBPARTNER ODBC Driver for DABroker –環境定義–] ダイアログボックスに指定する項目を次に示します。

- **[通信ドライバ] テキストボックス** ～ 〈文字列〉

[WINSOCK] で固定です。変更できません。

- **[応答監視時間] テキストボックス** ～ 〈符号なし整数〉 ((0~3600)) 《60》 (単位：秒)

サーバプログラムからの応答監視時間を秒単位で指定します。ここで指定した時間を過ぎてもサーバプログラムから応答が返らないと、処理が打ち切られます。

指定を省略すると、60 (秒) が仮定されます。0 を指定すると、時間監視しません。

- **[非同期キャンセル機能を使用する] チェックボックス**

このチェックボックスは無効です。このチェックボックスのチェックの有無にかかわらず、非同期キャンセルは実行されます。ODBC 関数の非同期実行については、「5.7 ODBC 関数の非同期実行」を参照してください。

(c) 設定の完了

[DBPARTNER ODBC Driver for DABroker –環境定義–] ダイアログボックスの項目を指定した後、[OK] ボタンをクリックします。

[DBPARTNER ODBC Driver for DABroker –セットアップ–] ダイアログボックスに戻ったら、[OK] ボタンをクリックします。

これでデータソースのセットアップは終了しました。

(2) セットアップの中止

データソースのセットアップを中止するときは、[DBPARTNER ODBC Driver for DABroker –セットアップ–] ダイアログボックスの [キャンセル] ボタンをクリックします。ボタンをクリックすると、設定した内容は登録されません。

(3) データソース名の変更

既に設定した内容を変更する場合は、次の手順に従ってください。

1. [ODBC データソースアドミニストレータ] ダイアログボックスで変更するデータソース名を選択し、[構成(C)...] ボタンをクリックします。
[DBPARTNER ODBC Driver for DABroker –セットアップ–] ダイアログボックスが表示されます。
2. データソース名を変更します。
これ以降の操作は、新規追加の 2. からの操作に従ってください。

(4) データソース名の削除

データソース名を削除するときは、[ODBC データソースアドミニストレータ] ダイアログボックスから削除するデータソース名を選んでから [削除(R)] ボタンをクリックします。

ボタンをクリックすると、データソース名が削除されます。

(5) ユーザ外字変換ファイルの作成方法

ユーザ外字変換ファイルの作成方法について説明します。外字コードを変換するときは、次に示すテキスト形式のファイル (~.txt) を作成します。

(a) 形式

```
クライアント PC 側外字コード 1 =サーバ側外字コード 1,  
クライアント PC 側外字コード 2 =サーバ側外字コード 2
```

(b) 規則

- 行の先頭に空白文字を入れないでください。
- クライアント PC 側外字コード、サーバ側外字コードとも、16 進数表記で指定してください。16 進数を示す先頭の「0x」は付けしないでください。
(例)「0xF040」は「F040」と記述してください。
- コードに小文字は使えません。
(例)「0xf040」は「F040」と記述してください。
- 使える文字は、半角英数字と半角スペースです。使える文字数を次に示します。
サーバが EUC コードの場合：4 文字又は 6 文字 (例：F040, 8FA1A1)

サーバがシフト JIS コードの場合：4 文字（例：F040）

- 指定できる外字コードの範囲を次に示します。範囲については、サーバの OS がサポートする外字コードの範囲を確認してください。

クライアント PC の文字コード：F040～F9FC（ただし、外字エディタで使える範囲に限ります）

サーバの文字コード：任意（指定が妥当かどうかは、チェックされません）

- 「=」の前後に空白文字を記述できます。
- 1 行に複数のコードを記述できます。複数のコードを記述するときは、「,」で区切ってください。
- 改行は、区切りを示す「,」の直後だけでできます。
- 同じ変換前外字コードを複数回定義したときは、最後に定義したコードが有効になります。
- 「#」から改行するまでは、コメント行とみなされます。コメント行も、行の先頭に空白文字を入れなくてください。

(c) ユーザ外字変換ファイルの記述例

```
# ユーザ定義文字マッピング (SJIS<->EUC)
F040 = 8FA1A1
F041 = F5A1
# END
```

(d) 注意

- ユーザ外字変換ファイルに指定したコード範囲が妥当かどうかは、チェックしません。また、変換する外字として使える文字コードの範囲は、サーバ側のシステムに依存します。ユーザ外字変換ファイルに文字コードを指定するときは、サーバ側のシステムの OS や DBMS のマニュアルを参照して、サポートする文字コードの範囲を事前に確認してください。
- ユーザ外字変換ファイルは、サーバ側のシステムとの接続を開始するときに読み込まれます。このため、ユーザ外字変換ファイルの内容を変更したときは、サーバ側のシステムとの接続をいったん解除してから、再び接続してください。

2.4 マクロスクリプトの設定 (CommuniNet 接続の場合)

CommuniNet 接続の DBPARTNER2 Client の ActiveX コントロールを使ってアプリケーションを開発するときには、CommuniNet をインストールする必要があります。

2.4.1 CommuniNet のインストールの手順

CommuniNet をインストールするときは、[コントロールパネル] を閉じてから実行してください。

CommuniNet のセットアップについては、該当する CommuniNet のマニュアルを参照してください。

2.4.2 マクロスクリプトファイルの作成

CommuniNet 接続で DBPARTNER2 Client を使うときは、HOAPDBS E2 や ACE3 E3 などのサーバプログラムにログオン、ログオフするためのマクロスクリプトを設定する必要があります。マクロスクリプトを CommuniNet で実行させると、オンライン画面で実行するログオン、ログオフなどのキー入力操作を自動的に実行できます。

マクロスクリプトについては、CommuniNet のマニュアルを参照してください。

(1) マクロスクリプトファイルの例

ログオン、ログオフの操作は TSS の設定で異なるので、次の例を参考に作成してください。

マクロスクリプトファイルのサンプルは、DBPARTNER2 Client のフォルダ¥Samples フォルダに格納してあります。ファイルの内容を必要に応じて修正して使うと、最初からマクロスクリプトファイルを作成する手間が省けます。

(a) サーバプログラムが VOS3 HOAPDBS E2 の例

このサンプルは、VOS3 HOAPDBS E2 を使うときの例です。データベースには VOS3 XDM/RD E2, VOS3 XDM/SD E2 を使えます。

LOGON マクロスクリプト

```
# Begin Logon Macro
# WRITEコマンド, SENDコマンドを実行する前には,
# ステータスが"キーOK"になるのを待ちます。
AWAIT KEYOK
SEND SRQ
AWAIT KEYOK
WRITE (1,1) "LOGON "
WRITE %1
WRITE "/"
WRITE %2
WRITE " S(5000) G(GENKA)"
```

```

SEND EXEC
RECEIVE (1,10) "READY"
# 画面に"READY"が表示されるのを待ちます。
# 環境によってカーソル位置, 文字列を変更してください。
AWAIT KEYOK
WRITE "SETCS &##HOAPDBSPCDBID VALUE(XRDA)" ※
SEND EXEC
AWAIT KEYOK
WRITE "HDBPC" ※
SEND EXEC
# End

```

注※ この部分に指定する値については、マニュアル「VOS3 統合情報管理共通サービス HOAPSERV 運用の手引」を参照してください。

LOGOFF マクロスクリプト

```

# Begin Logoff Macro
AWAIT KEYOK
WRITE "LOGOFF"
SEND EXEC
AWAIT KEYOK
# End

```

(b) サーバプログラムが VOS3 ACE3 E3 の例

このサンプルは、VOS3 ACE3 E3 を使うときの例です。データベースには VOS3 PDMII E2, VOS3 ユーザファイル, VOS3 RDB1 を使えます。

LOGON マクロスクリプト

```

# Begin Logon Macro
# WRITEコマンド, SENDコマンドを実行する前には,
# ステータスが"キーOK"になるのを待ちます。
AWAIT KEYOK
SEND SRQ
AWAIT KEYOK
WRITE (1,1) "LOGON"
WRITE %1
WRITE "/"
WRITE %2
WRITE " S(5000) G(GENKA)"
SEND EXEC
# 画面に"READY"が表示されるのを待ちます。
# 環境によってカーソル位置, 文字列を変更してください。
RECEIVE (1,10) "READY"

※1
AWAIT KEYOK
WRITE "SETCS &##ACE3E3PCUSID V(' "
WRITE %1
WRITE ")'"
SEND EXEC
AWAIT KEYOK
WRITE "SETCS &##ACE3E3PCPSWD V(' "
WRITE %2

```

```

WRITE ")”
SEND EXEC

※2
AWAIT KEYOK
WRITE "SETCS &¥¥ACE3E3PCLINK V(' YES' )" ※3
SEND EXEC
AWAIT KEYOK

AWAIT KEYOK
WRITE "ACE3E3" ※3
SEND EXEC
# End

```

注※1 サーバに TRUST を組み込んでいないときは、この指定が必要です。

注※2 VOS3 ACE3 E3 の環境を作成して、ACE3OPT マクロに PCLINK = YES を指定したときは、この指定は不要です。

注※3 この部分の指定については、マニュアル「VOS3 エンドユーザ言語 ACE3 E3 運用編」を参照してください。

LOGOFF マクロスクリプト

```

# Begin Logoff Macro
AWAIT KEYOK
WRITE "LOGOFF"
SEND EXEC
AWAIT KEYOK
# End

```

(c) サーバプログラムが VOS1 DBPARTNER/Link の例

このサンプルは、VOS1 DBPARTNER/Link を使うときの例です。データベースには VOS1 PDMII E2, VOS1 ユーザファイル, VOS1 スプールファイルを使えます。

LOGON マクロスクリプト

```

# Begin Logon Macro
# 端末とES/IEXを接続します。
SEND EXEC
RECEIVE (1,2) "JCK071I ENTER LOGON"
AWAIT KEYOK
WRITE (1,2) "LOGON IEX          "
SEND EXEC
# セッション選択画面の表示を待ち, ユーザID, パスワード, セッション名を入力
# します。
RECEIVE (1,14) "セッション選択画面"
AWAIT KEYOK
WRITE (21,18) %1
WRITE (21,58) %2
WRITE (22,18) "ACE3001"
SEND EXEC
# マスタメニュー画面の表示を待ち, ACE3 E2を起動します。
RECEIVE (2,2) "JMASTER" ※
AWAIT KEYOK

```

```
WRITE (22,13) "@@ACE3 CPROC=ACE3DCT,DBUSE=P,MODE=DBPARTNER" ※
SEND EXEC
# END
```

注※ この部分の指定については、マニュアル「VOS1 エンドユーザ言語 ACE3 E2 運用の手引」を参照してください。

LOGOFF マクロスクリプト

```
# Begin Logoff Macro
# マスタメニュー画面の表示を待ち,"6. セッションの終了"を入力します。
RECEIVE (2,2) "JMASTER"
AWAIT KEYOK
WRITE (22,13) "6"
SEND EXEC
# セッションの終了画面の表示を待ち,"1. このメニューを実行中のセッションの終
# 了"を選択します。
RECEIVE (2,2) "JUESSI"
AWAIT KEYOK
WRITE (4,46) "1"
SEND EXEC
#セッション選択画面の表示を待ち,入力行へLOGOFFを入力します。
RECEIVE (1,14) "セッション選択画面"
AWAIT KEYOK
WRITE (24,13) "LOGOFF"
SEND EXEC
#"JIC101I"メッセージの表示を待ちます。
RECEIVE (20,3) "JIC101I"
AWAIT KEYOK
SEND EXEC
# オンラインモードに戻ったことを確認します。
RECEIVE (1,2) "JCK071I ENTER LOGON"
AWAIT KEYOK
# END
```

(d) サーバプログラムが VOSK HOAPDBS/PC の例

このサンプルは、VOSK HOAPDBS/PC を使うときの例です。データベースには VOSK RDB 編成ファイル、VOSK スプールファイルを使えます。

LOGON マクロスクリプト

```
# Begin Logon Macro
# "ログオン画面"が表示されるのを待ちます。
RECEIVE (1,36) "ログオン画面"
# 環境によってログオン画面の表示が完了するまでに時間が掛かり,
# 完了しないまま SENDコマンドを実行するとエラーになります。
# ログオン画面が表示完了するまでに時間が掛かる場合は,
# PAUSEコマンドでマクロスクリプトの実行を一時中断してください。
# PAUSE 3
AWAIT KEYOK
WRITE (3,27) "1"
WRITE (6,27) %1
WRITE (6,57) %2
WRITE (7,27) "#HDBSPC"
SEND EXEC
```

```
RECEIVE (22,3) "コマンド"  
# WRITEコマンド, SENDコマンドを実行する前には,  
# ステータスが"キーOK"になるのを待ちます。  
AWAIT KEYOK  
WRITE (22,14) "DEF HDBID,DBID=RDFA" ※  
SEND EXEC  
AWAIT KEYOK  
WRITE (22,14) "HDBPC" ※  
SEND EXEC  
# End
```

注※ この部分の指定については、マニュアル「VOSK システム OA」を参照してください。

LOGOFF マクロスクリプト

```
# Begin Logoff Macro  
AWAIT KEYOK  
WRITE (22,14) "LOGOFF"  
SEND EXEC  
RECEIVE (1,36) "ログオン画面"  
# End
```

3

アプリケーションを使ったデータベースアクセス (ODBC 接続)

この章では、DBPARTNER2 Client のアプリケーションから ActiveX コントロールのオブジェクトを使ってデータベースにアクセスする方法（ODBC 接続）について説明します。

3.1 DBPARTNER2 Client が提供する ActiveX コントロールの概要

DBPARTNER2 Client のアプリケーションを作成するときは、DBPARTNER2 Client が提供する ActiveX コントロールのオブジェクトを使います。オブジェクトを使って、データベースからデータを抽出したり、更新したりできます。

3.1.1 ActiveX コントロールのオブジェクトの階層

ActiveX コントロールのオブジェクトの階層を図 3-1 に、オブジェクトの説明を表 3-1 に示します。複数形のオブジェクト名 (Tables, Columns など) は、下位のオブジェクトを管理するコレクションオブジェクトです。

図 3-1 DBPARTNER2 Client が提供する ActiveX コントロールのオブジェクトの階層

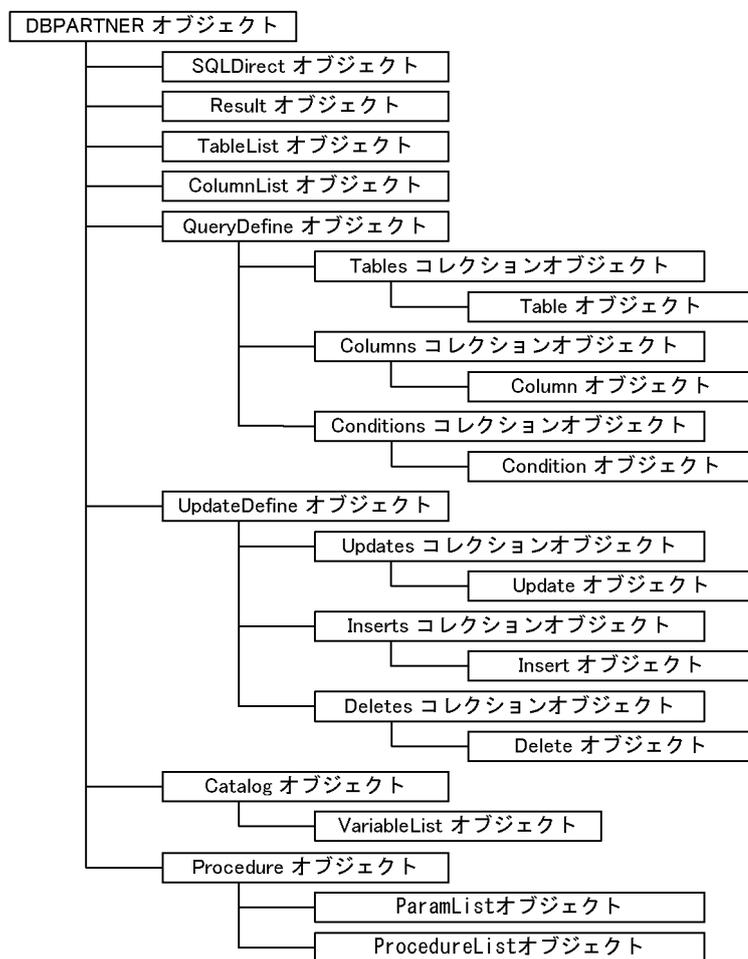


表 3-1 DBPARTNER2 Client が提供する ActiveX コントロールのオブジェクトの説明

オブジェクト	説明
DBPARTNER	サーバプログラムとの接続を管理するオブジェクトです。
SQLDirect	ユーザが作成する SQL 文の実行を管理するオブジェクトです。

オブジェクト	説明
Result	抽出データを管理するオブジェクトです。
TableList	表情報を管理するオブジェクトです。
ColumnList	列情報を管理するオブジェクトです。
QueryDefine	抽出処理を実行するオブジェクトです。
Tables	Table オブジェクトを管理するコレクションオブジェクトです。
Table	抽出表を管理するオブジェクトです。
Columns	Column オブジェクトを管理するコレクションオブジェクトです。
Column	抽出列を管理するオブジェクトです。
Conditions	Condition オブジェクトを管理するコレクションオブジェクトです。
Condition	抽出条件を管理するオブジェクトです。
UpdateDefine	更新処理を実行するオブジェクトです。
Updates	Update オブジェクトを管理するコレクションオブジェクトです。
Update	変更行条件, 変更情報を管理するオブジェクトです。
Inserts	Insert オブジェクトを管理するコレクションオブジェクトです。
Insert	追加情報を管理するオブジェクトです。
Deletes	Delete オブジェクトを管理するコレクションオブジェクトです。
Delete	削除行条件を管理するオブジェクトです。
Catalog	手順ファイルを管理するオブジェクトです。
VariableList	手順ファイルの可変値を管理するオブジェクトです。
Procedure	ストアードプロシジャを実行するオブジェクトです。
ParamList	ストアードプロシジャのパラメータ一覧を管理するオブジェクトです。
ProcedureList	ストアードプロシジャー一覧を管理するオブジェクトです。

■ 注意事項

DBPARTNER2 Client の ActiveX コントロールのオブジェクトには、使い方に制限があります。制限については、「[3.12 避けて欲しい処理](#)」を参照してください。

3.1.2 プロパティとメソッドの表記

ActiveX コントロールのオブジェクトに属するプロパティとメソッドは、「オブジェクト名. プロパティ又はメソッド名」で示します。

(例) DBPARTNER オブジェクトの Connect メソッドの表記

DBPARTNER.Connect メソッド

ビジュアル開発ツールでは、オブジェクト名を「.」（ピリオド）で表記できます。そのため、このマニュアルでプロパティ又はメソッド名にオブジェクト名を付けて表記するときは、オブジェクト名を斜体文字にしています。

このマニュアルでは、Visual Basic でアプリケーションを開発することを想定して説明します。ほかのビジュアル開発ツールを使ってアプリケーションを作成するときは、マニュアルの記述を読み替えてください。

3.1.3 アプリケーションのサンプル

DBPARTNER2 Client では、次に示すフォルダ下にサンプルを提供しています。

- DBPARTNER2 Client のフォルダ¥Ocx¥Samples

サンプルでは、テキストボックスやリストボックスも実装しています。DBPARTNER2 Client の詳しいアプリケーションの処理については、サンプルを参照してください。

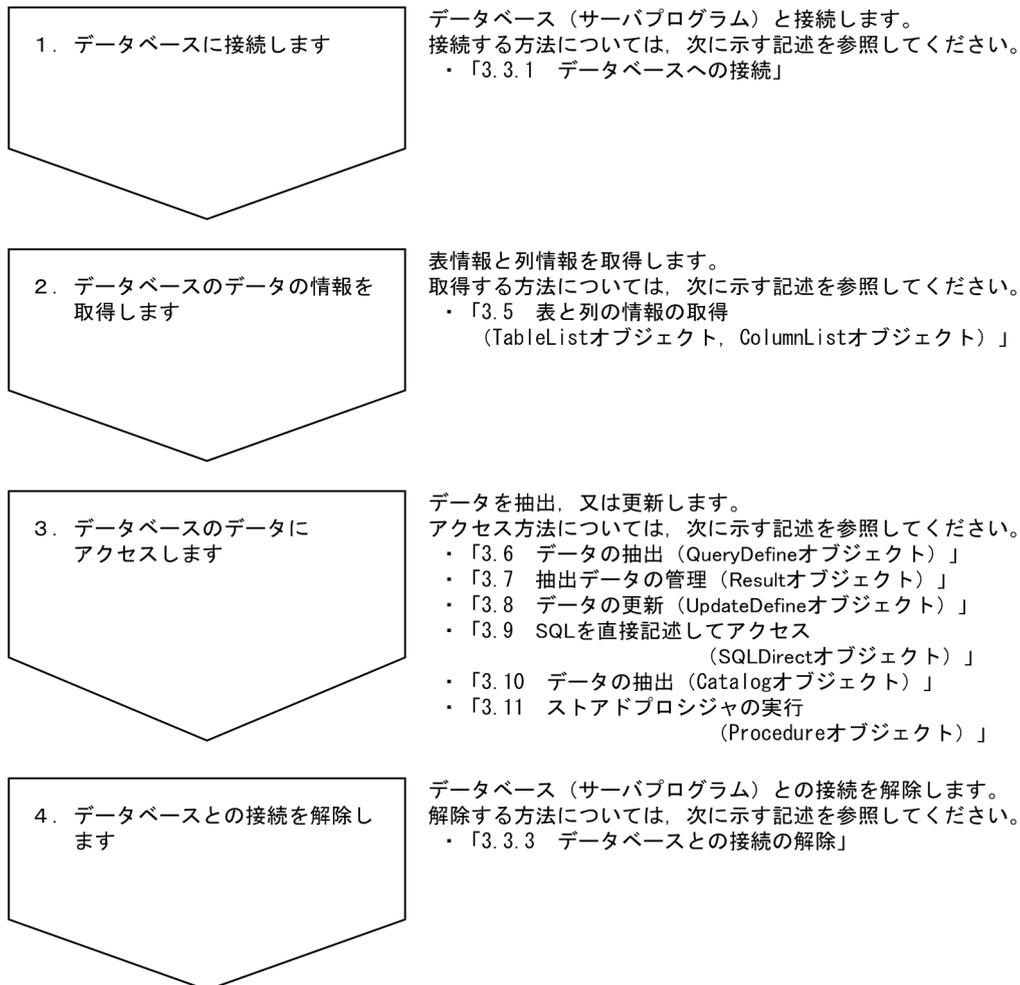
3.1.4 アプリケーションの作成

ActiveX コントロールを使用したアプリケーションを開発するためのプログラムとして、AP 作成ウィザードを提供しています。AP 作成ウィザードを使用すると、データベースからデータを抽出するアプリケーションが簡単に作成できます。AP 作成ウィザードについては、「付録 A AP 作成ウィザード」を参照してください。

3.2 データベースアクセスの手順

DBPARTNER2 Client が提供する ActiveX コントロールのオブジェクトを使って、データベースにアクセスする方法について説明します。DBPARTNER2 Client からデータベースにアクセスする手順の概要を図 3-2 に示します。

図 3-2 DBPARTNER2 Client からデータベースにアクセスする手順の概要



データベースにアクセスするときは、**トランザクション**と**排他制御**の知識が必要です。トランザクションと排他制御については、「3.4 トランザクションと排他制御 (DBPARTNER オブジェクト)」を参照してください。

ActiveX コントロールのオブジェクトの文法については、「4. DBPARTNER2 Client のオブジェクトの文法 (ODBC 接続)」を参照してください。

サンプルの参照先

DBPARTNER オブジェクトは、抽出／更新で使います。サンプルについては、DBPARTNER2 Client のフォルダ¥Ocx¥Samples¥Vb¥Odbc フォルダを参照してください。

3.2.1 DBPARTNER オブジェクトのプロパティの設定

DBPARTNER オブジェクトは、DBPARTNER2 Client のトップオブジェクトです。データベースに接続する前に、DBPARTNER オブジェクトのプロパティを設定してください。

(1) DBPARTNER オブジェクトのプロパティに設定する内容

DBPARTNER オブジェクトのプロパティを次に示します。

- *DBPARTNER.CommitMode* プロパティ
トランザクションの終了モードを設定します。
- *DBPARTNER.ConnectDialogMode* プロパティ
データベースと接続するときに、ユーザ ID とパスワードの入力を要求するダイアログボックスを表示するかどうかを設定します。
- *DBPARTNER.DataSource* プロパティ
データソース名を設定します。ODBC ドライバのセットアップで設定したデータソース名を設定してください。
- *DBPARTNER.UserID* プロパティ
ユーザ ID を設定します。
- *DBPARTNER.Password* プロパティ
パスワードを設定します。

3.3 データベースへの接続と接続の解除 (DBPARTNER オブジェクト)

DBPARTNER2 Client とデータベースを接続する方法と接続を解除する方法について説明します。DBPARTNER2 Client とデータベースを接続するときは、サーバプログラム (DABroker 又は Database Connection Server) が必要です。DBPARTNER2 Client がサーバプログラムに接続すると、データベースに接続されたこととなります。

3.3.1 データベースへの接続

DBPARTNER2 Client からデータベースに接続するときは、*DBPARTNER.Connect* メソッドを使います。*DBPARTNER.Connect* メソッドは、データソース名 (*DBPARTNER.DataSource* プロパティ)、ユーザ ID (*DBPARTNER.UserID* プロパティ)、パスワード (*DBPARTNER.Password* プロパティ) を参照して、データベースに接続します。

それぞれのプロパティを変更すると、接続するデータベースを変更できます。

データベースと接続するときにデータベース又はサーバプログラムが起動されていないときは、エラーメッセージが表示されます。

(1) 複数のデータベースにアクセスするとき

一つのアプリケーションから、複数のデータベースにアクセスできます。例えば、HiRDB の「支店データベース」と「全社データベース」の両方にアクセスしたり、HiRDB と ORACLE など別の種類のデータベースにアクセスしたりできます。

複数のデータベースにアクセスするときは、ODBC ドライバで複数のデータソース名を指定します。そして、それぞれのデータベースごと (データソース名ごと) に DBPARTNER オブジェクトをデータベースの数だけ実行して、*DBPARTNER.Connect* メソッドを実行します。このとき、トランザクションはデータベース単位で管理されます。

(2) データベースとの接続の確認

DBPARTNER2 Client とサーバプログラムが接続されているかどうかは、*DBPARTNER.IsConnect* プロパティを参照して確認できます。

3.3.2 データベースへのアクセスのキャンセル

DBPARTNER2 Client からデータベースへのアクセス処理をキャンセルするときは、*DBPARTNER.Cancel* メソッドを使います。*DBPARTNER.Cancel* メソッドを実行できる条件とキャンセル後の格納データについては、「4. DBPARTNER2 Client のオブジェクトの文法 (ODBC 接続)」の *DBPARTNER.Cancel* メソッドの説明を参照してください。

3.3.3 データベースとの接続の解除

DBPARTNER2 Client とデータベースの接続を解除するときは、*DBPARTNER.Disconnect* メソッドを使います。*DBPARTNER.Disconnect* メソッドを使った後は、接続していたデータベースへの処理はすべてエラーになります。

抽出処理を実行している間に *DBPARTNER.Disconnect* メソッドを使うと、メソッドがエラーになります。また、*DBPARTNER.Connect* メソッドで接続していないデータベースに *DBPARTNER.Disconnect* メソッドを実行しても無視されます（エラーになりません）。

3.4 トランザクションと排他制御 (DBPARTNER オブジェクト)

アプリケーションでトランザクションを制御するときの手順について説明します。

3.4.1 トランザクションの制御の設定

トランザクションを DBPARTNER2 Client で自動的に制御するか、アプリケーションからトランザクションを制御するかは、*DBPARTNER.CommitMode* プロパティに設定します。

(1) DBPARTNER2 Client で自動的にトランザクションを制御するとき

DBPARTNER2 Client で自動的にトランザクションを制御するときは、*DBPARTNER.CommitMode* プロパティに *dbpAuto* を設定します。*dbpAuto* を設定すると、データベースアクセスのトランザクションを DBPARTNER2 Client のアプリケーションから意識する必要はありません。

(2) アプリケーションからトランザクションを制御するとき

アプリケーションからメソッドを使ってトランザクションを制御するときは、*DBPARTNER.CommitMode* プロパティに *dbpManual* を設定して、*DBPARTNER.Commit* メソッド又は *DBPARTNER.Rollback* メソッドを使います。

DBPARTNER2 Client は、データベースへのアクセス処理 (表の参照, SQL の実行など) を *Execute* メソッドで実行してから *DBPARTNER.Commit* メソッド又は *DBPARTNER.Rollback* メソッドの実行までを、1 件のトランザクションとして扱います。*DBPARTNER.Commit* 又は *DBPARTNER.Rollback* が成功すると、その時点から次のトランザクションが開始されます。データベースにアクセスし直す必要はありません。

トランザクションを終了させないでオブジェクトを削除、又はデータベースとの接続を切断すると、DBPARTNER2 Client がアプリケーションの処理が異常と判断して *DBPARTNER.Rollback* メソッドの処理を自動的に実行します。

3.4.2 排他制御

DBPARTNER2 Client のアプリケーションからデータを抽出/更新するときには、ほかのアプリケーションからのアクセスを許すかどうかを設定します。これを排他制御といいます。

(1) 排他制御の設定

排他制御は、データにアクセスするプロパティ (*QueryDefine.Exclusive* プロパティ又は *UpdateDefine.Exclusive* プロパティ) に文字列で設定します。アクセスするデータベースによって、設定する内容が異なります。設定する内容については、アクセスするデータベースのマニュアルを参照してください。

(2) 排他エラーになったときの動作の設定

データにアクセスしようとした場合に、ほかのアプリケーションがデータを確保しているときは、データが解放されるまで待つか、エラーにするかを選べます。どちらにするかは、データにアクセスするプロパティ (*QueryDefine.Exclusive* プロパティ又は *UpdateDefine.Exclusive* プロパティ) に設定します。

データが解放されるまで待つ設定をすると、複数のアプリケーションが互いが確保しているデータの解放を待ち続ける状態 (デッドロック) になるおそれがあります。デッドロックにならないように、複数のアプリケーション間でデータを確保する順番を決めておいてください。さらに、複数のアプリケーションからデータベースにアクセスするときは、確保するデータの範囲を最小限に抑えてください。

エラーにする設定をすることは、データベースアクセスごとにエラーの対処 (一定時間を空けてリトライをするなど) が必要になります。

3.5 表と列の情報の取得 (TableList オブジェクト, ColumnList オブジェクト)

データを抽出する前に表と列の情報を取得して、どのように抽出するかを決めます。情報には、表情報と列情報があります。

3.5.1 表情報の取得

表情報を取得するときは、*DBPARTNER.Connect* メソッドでサーバプログラムと接続した後、*TableList* オブジェクトを使います。

表情報を取得するときのオブジェクトの流れとプロパティとメソッドの実行手順を図 3-3 に示します。

図 3-3 表情報を取得するときのオブジェクトの流れとプロパティとメソッドの実行手順

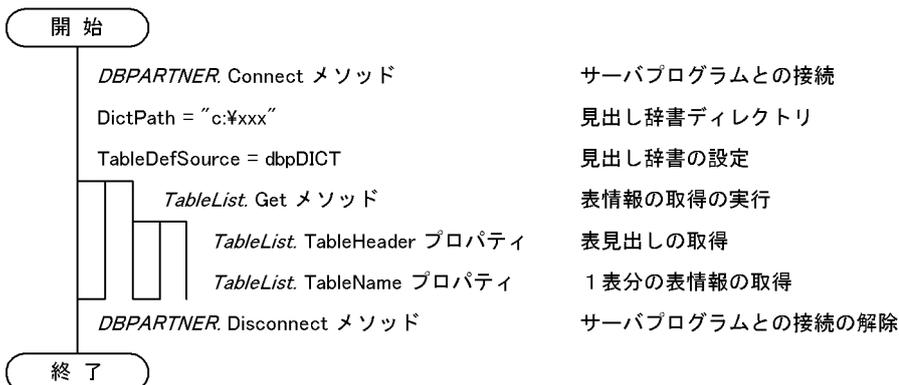
●オブジェクトの流れ



●プロパティとメソッドの流れ (見出し辞書を使用しない場合)



●プロパティとメソッドの流れ (見出し辞書を使用する場合)



サンプルの参照先

TableList オブジェクトのサンプルは、DBPARTNER2 Client のフォルダ¥Ocx¥Samples¥Vb¥Odbc ¥Query フォルダを参照してください。

(1) 表情情報を取得する手順

表情情報を取得する手順を次に示します。

見出し辞書を使用しない場合

- 次に示す `TableList` オブジェクトのプロパティを設定します。
 - `TableList.Pattern` プロパティ：絞り込み条件を設定します。
 - `TableList.CountLimit` プロパティ：取得する表情情報の個数を設定します。
- `TableList.Get` メソッドで、表情情報の取得を実行します。`TableList.Get` メソッドを実行すると、次に示すプロパティの値を参照できます。
 - `TableList.Count` プロパティ：取得した表情情報の個数。
 - `TableList.TableName` プロパティ：取得した表の名前。
 - `TableList.TableType` プロパティ：取得した表の属性。
 - `TableList.TableOwner` プロパティ：取得した表の所有者。
 - `TableList.TableComment` プロパティ：取得した表の注釈。
 - `TableList.TableCreateDate` プロパティ：取得した VOSK スプールファイルの作成日付。
 - `TableList.TableGroupName` プロパティ：取得した表のグループ名。
 - `TableList.TableHeader` プロパティ：取得した表のラベル。上記のプロパティから、必要な情報を参照してください。

見出し辞書を使用する場合

- 次に示すオブジェクトのプロパティを設定します。
 - `DBPARTNER.DictPath` プロパティ：使用する見出し辞書のパスを設定します。
 - `DBPARTNER.TableDefSource` プロパティ：表情情報の取得元を設定します。
 - `TableList.SelectDict` プロパティ：使用する見出し辞書を限定します。
 - `TableList.Pattern` プロパティ：絞り込み条件を設定します。
 - `TableList.CountLimit` プロパティ：取得する表情情報の個数を設定します。
- `TableList.Get` メソッドで、表情情報の取得を実行します。`TableList.Get` メソッドを実行すると次に示すプロパティの値を参照できます。
 - `TableList.TableDataSource` プロパティ：接続先の名称を取得します。
 - `TableList.Count` プロパティ：取得した表情情報の個数。
 - `TableList.TableName` プロパティ：取得した表の名前。
 - `TableList.TableType` プロパティ：取得した表の属性。
 - `TableList.TableOwner` プロパティ：取得した表の所有者。

- *TableList.TableComment* プロパティ：取得した表の注釈。
- *TableList.TableCreateDate* プロパティ：取得した VOSK スプールファイルの作成日付。
- *TableList.TableGroupName* プロパティ：取得した表のグループ名。
- *TableList.TableHeader* プロパティ：取得した表のラベル。

上記のプロパティから、必要な情報を参照してください。

(2) 注意

TableList.Pattern プロパティと *TableList.CountLimit* プロパティは、*TableList.Get* メソッドを実行する前に設定してください。*TableList.Get* メソッドを実行した後に変更しても、無視されます。

3.5.2 列情報の取得

列情報を取得するときは、*DBPARTNER.Connect* メソッドでサーバプログラムと接続した後、*ColumnList* オブジェクトを使います。

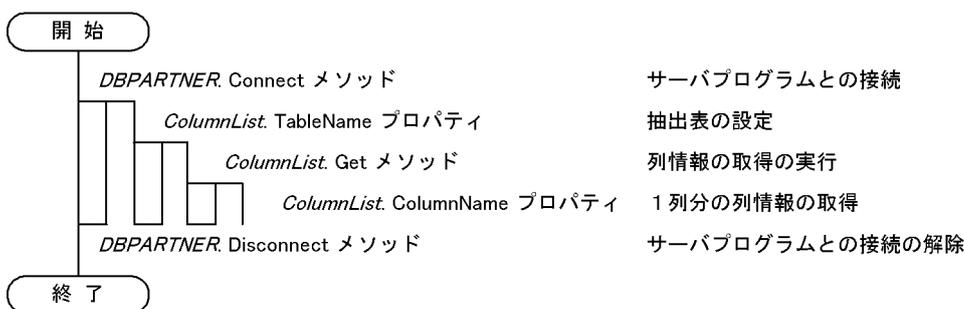
表情報を取得するときのオブジェクトの流れとプロパティとメソッドの実行手順を図 3-4 に示します。

図 3-4 列情報を取得するときのオブジェクトの流れとプロパティとメソッドの実行手順

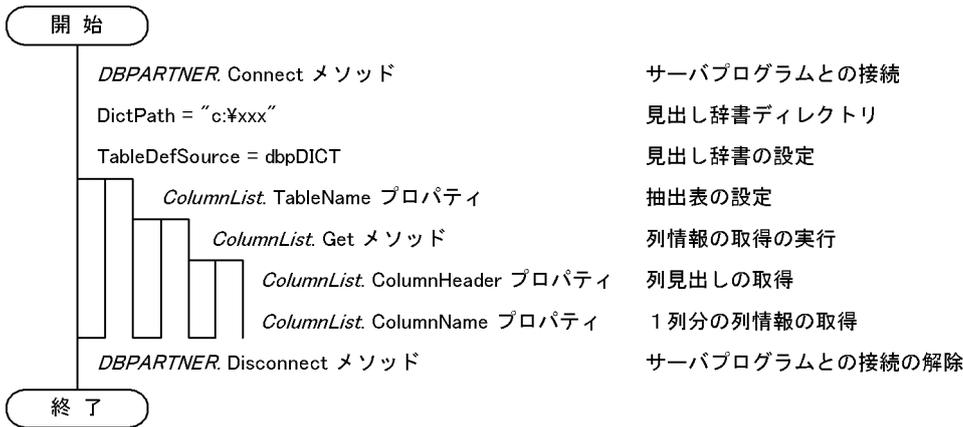
●オブジェクトの流れ



●プロパティとメソッドの流れ（見出し辞書を使用しない場合）



●プロパティとメソッドの流れ（見出し辞書を使用する場合）



サンプルの参照先

ColumnList オブジェクトのサンプルは、DBPARTNER2 Client のフォルダ¥Ocx¥Samples¥Vb¥Odbc ¥Query フォルダを参照してください。

(1) 列情報を取得する手順

列情報を取得する手順を次に示します。

見出し辞書を使用しない場合

1. 次に示す ColumnList オブジェクトのプロパティを設定します。

- *ColumnList.TableName* プロパティ：列情報を取得する表の名前を設定します。
- *ColumnList.CountLimit* プロパティ：取得する列情報の個数を設定します。

2. *ColumnList.Get* メソッドで、列情報の取得を実行します。*ColumnList.Get* メソッドを実行すると、次に示すプロパティの値を参照できます。

- *ColumnList.Count* プロパティ：取得した列の個数。
- *ColumnList.ColumnName* プロパティ：取得した列の名前。
- *ColumnList.ColumnType* プロパティ：取得した列の属性。
- *ColumnList.ColumnLength* プロパティ：取得した列の定義長。
- *ColumnList.ColumnScale* プロパティ：取得した列の小数点以下のけた数。
- *ColumnList.ColumnComment* プロパティ：取得した列の注釈。
- *ColumnList.ColumnHeader* プロパティ：取得した列のラベル。

上記のプロパティから、必要な情報を参照してください。

見出し辞書を使用する場合

1. 次に示すオブジェクトのプロパティを設定します。

- *DBPARTNER.DictPath* プロパティ：使用する見出し辞書のパスを設定します。

- *DBPARTNER.TableDefSource* プロパティ：列情報の取得元を設定します。
- *TableList.SelectDict* プロパティ：使用する見出し辞書を限定します。
- *ColumnList.Tablename* プロパティ：列情報を取得する表の名前を設定します。
- *ColumnList.CountLimit* プロパティ：取得する列情報の個数を設定します。

2. *ColumnList.Get* メソッドで、列情報の取得を実行します。*ColumnList.Get* メソッドを実行すると、次に示すプロパティの値を参照できます。

- *ColumnList.Count* プロパティ：取得した列の個数。
- *ColumnList.ColumnName* プロパティ：取得した列の名前。
- *ColumnList.ColumnType* プロパティ：取得した列の属性。
- *ColumnList.ColumnLength* プロパティ：取得した列の定義長。
- *ColumnList.ColumnScale* プロパティ：取得した列の小数点以下のけた数。
- *ColumnList.ColumnComment* プロパティ：取得した列の注釈。
- *ColumnList.ColumnHeader* プロパティ：取得した列のラベル。

上記のプロパティから、必要な情報を参照してください。

(2) 注意

ColumnList.TableName プロパティと *ColumnList.CountLimit* プロパティは、*ColumnList.Get* メソッドを実行する前に設定してください。*ColumnList.Get* メソッドを実行した後に変更しても、無視されます。

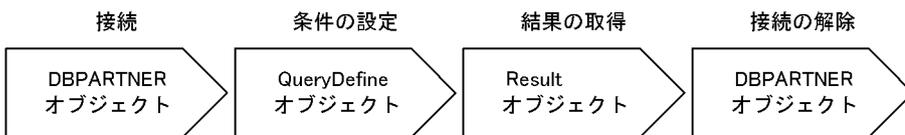
3.6 データの抽出 (QueryDefine オブジェクト)

DBPARTNER2 Client のオブジェクト QueryDefine を使って、データを抽出する方法について説明します。データを抽出するときは、*DBPARTNER.Connect* メソッドでサーバプログラムと接続した後、QueryDefine オブジェクトを使います。

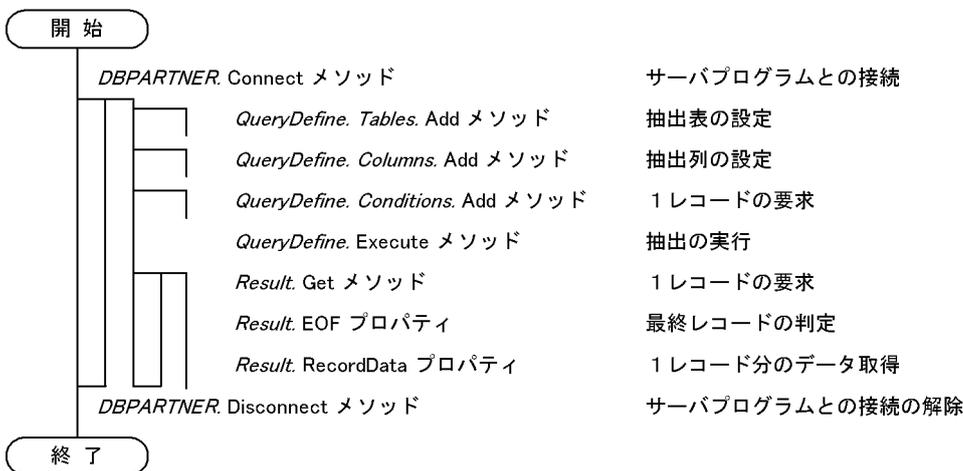
QueryDefine オブジェクトを使った抽出のオブジェクトの流れとプロパティとメソッドの実行手順を図 3-5 に示します。

図 3-5 QueryDefine オブジェクトを使った抽出のオブジェクトの流れとプロパティとメソッドの実行手順

●オブジェクトの流れ



●プロパティとメソッドの流れ



サンプルの参照先

QueryDefine オブジェクトのサンプルは、DBPARTNER2 Client のフォルダ¥Ocx¥Samples¥Vb¥Odbc¥Query フォルダを参照してください。

3.6.1 QueryDefine オブジェクトを使ってデータを抽出する手順

QueryDefine オブジェクトを使ってデータを抽出する手順を次に示します。

1. 次に示すオブジェクトに、抽出データに関する条件を設定します。

- **Tables コレクションオブジェクト**：下位にある複数の Table オブジェクトに、抽出表の名称を設定します。Table オブジェクトの *TableName* プロパティに、抽出表の名称を設定します。

- **Columns コレクションオブジェクト**：下位にある複数の Column オブジェクトに、抽出列の名称を設定します。Column オブジェクトの ColumnName プロパティに、抽出列の名称を、SortMode プロパティと SortPriority プロパティにソートする条件を、FunctionMode プロパティに集合関数を設定します。
- **Conditions コレクションオブジェクト**：下位にある複数の Condition オブジェクトに、抽出条件式を設定します。Condition オブジェクトの Text プロパティに抽出条件式を設定します。

2. *QueryDefine.Exclusive* プロパティに、排他モードを設定します。

3. *QueryDefine.Execute* メソッドで、抽出を実行します。

4. **Result オブジェクト**で、抽出データを処理します。Result オブジェクトの使い方については、「[3.7 抽出データの管理 \(Result オブジェクト\)](#)」を参照してください。

3.7 抽出データの管理 (Result オブジェクト)

`SQLDirect.Execute` メソッド又は `QueryDefine.Execute` メソッドを実行して抽出したデータを処理する方法について説明します。抽出データを処理するときは、Result オブジェクトを使います。

サンプルの参照先

Result オブジェクトのサンプルは、DBPARTNER2 Client のフォルダ¥Ocx¥Samples¥Vb¥Odbc ¥Query フォルダ又は DBPARTNER2 Client のフォルダ¥Ocx¥Samples¥Vb¥Odbc¥Update フォルダを参照してください。

3.7.1 抽出データの格納手順

Result オブジェクトを使って、抽出データを格納する手順について説明します。

(1) 抽出データを格納する方法

抽出データを格納する手順を次に示します。

1. 次に示す Result オブジェクトのプロパティを設定します。

- `Result.QueryCountLimit` プロパティ：抽出処理の結果として出力する行数を設定します。
- `Result.DataCheck` プロパティ：抽出データの文字コードをチェックするかどうかを設定します。
- `Result.HeaderOutput` プロパティ：抽出データの先頭行に、列名を付けるかどうかを設定します。
- `Result.FieldCodeType` プロパティ：抽出データを取得するときのデータ型を設定します。
`Result.FieldCodeType` プロパティの設定内容と抽出できるデータ型の関係については、「[3.7.2 Result オブジェクトと取得するデータの関係](#)」を参照してください。

2. 1行ずつデータを抽出するときは、`Result.Get` メソッドを実行します。`Result.Get` メソッドを実行すると、次に示すプロパティの値を参照できます。

- `Result.RecordHeader` プロパティ：抽出処理の結果として出力される列名。
- `Result.RecordData` プロパティ：抽出データ。1行単位で返します。
- `Result.FieldData` プロパティ：文字列型の抽出データ。1列単位で返します。
- `Result.FieldDataB` プロパティ：バイト型の抽出データ。1列単位で返します。
- `Result.FieldType` プロパティ：列の属性。
- `Result.FieldDataLength` プロパティ：列のデータ長。
- `Result.FieldCount` プロパティ：抽出データの列数。

必要に応じて、次に示すプロパティを設定します。

- `Result.FieldHeader` プロパティ：列にヘッダを設定します。

3. 抽出データを一括してファイルに出力するときは、*Result.SaveToFile* メソッドを実行します。メソッドには、抽出するデータの形式（TXT 形式、CSV 形式、XLS 形式、DAT 形式）を設定します。抽出データが LONG VARBINARY 属性のデータのときは、*Result.SaveLargeObject* メソッドを使ってください。
- いったんクリップボードに退避するときは、*Result.CopyToClipboard* メソッドを実行します。メソッドには、抽出するデータの形式（TXT 形式、XLS 形式）を設定します。クリップボードに退避した抽出データは、Excel などのアプリケーションに貼り付けて使えます。
4. データをすべて取得できたかどうかは、*Result.EOF* プロパティで判定します。抽出処理を途中で終了させるときは、*Result.Close* メソッドを実行します。

3.7.2 Result オブジェクトと取得するデータの関係

Result オブジェクトのプロパティ／メソッドと、取得するデータの関係について説明します。

(1) 列属性とデータ取得の関係

列属性とデータの取得に使うプロパティ／メソッドによって、プロパティ／メソッドが使えるかどうか、又はプロパティ／メソッドに戻る値が異なります。

列属性とデータ取得の関係を表 3-2 に示します。FieldCodeType プロパティを使ってデータを取得するときは、「(2) FieldCodeType プロパティに設定した値とデータ取得の関係」を参照してください。

表 3-2 列属性とデータ取得の関係

プロパティ／メソッド	列属性		
	dbpTypeLONGVARBINARY	dbpTypeVARBINARY dbpTypeBINARY	左記以外の列属性
FieldData プロパティ	データの長さ	×	文字列型のデータ
FieldDataB プロパティ	バイト型のデータ	バイト型のデータ	×
RecordData プロパティ	データの長さ	データの長さ	文字列型のデータ
SaveToFile メソッド	データの長さ	データの長さ	文字列型のデータ
CopyToClipboard メソッド	データの長さ	データの長さ	文字列型のデータ
SaveLargeObject メソッド	バイト型のデータ	×	×

(凡例)

×：プロパティ／メソッドでは使えない列属性です。

(2) FieldCodeType プロパティに設定した値とデータ取得の関係

FieldCodeType プロパティの取得モードに取得するデータ型を設定したときは、取得モードとデータの取得に使うプロパティ/メソッドによって、取得できるかどうか、又は戻る値が異なります。

FieldCodeType プロパティに設定した値とデータ取得の関係を表 3-3 に示します。

表 3-3 FieldCodeType プロパティに設定した値とデータ取得の関係

FieldCodeType プロパティの属性	プロパティ/メソッド	取得モード	
		dbpTypeCHAR	dbpTypeBINARY
dbpTypeLONGVARBINARY	FieldData プロパティ	dbpTypeLONGVARBINARY の列には、dbpTypeCHAR は設定できません。	×
	FieldDataB プロパティ		バイト型のデータ
	RecordData プロパティ		データの長さ
	SaveToFile メソッド		データの長さ
	CopyToClipboard メソッド		データの長さ
	SaveLargeObject メソッド		×
dbpTypeVARBINARY, dbpTypeBINARY	FieldData プロパティ	文字列型のデータ	×
	FieldDataB プロパティ	×	バイト型のデータ
	RecordData プロパティ	文字列型のデータ	データの長さ
	SaveToFile メソッド	文字列型のデータ	データの長さ
	CopyToClipboard メソッド	文字列型のデータ	データの長さ
	SaveLargeObject メソッド	×	×
上記以外の属性	FieldData プロパティ	文字列型のデータ	×
	FieldDataB プロパティ	×	バイト型のデータ
	RecordData プロパティ	文字列型のデータ	×
	SaveToFile メソッド	文字列型のデータ	×
	CopyToClipboard メソッド	文字列型のデータ	×
	SaveLargeObject メソッド	×	×

(凡例)

×：プロパティ/メソッドでは使えない列属性です。

3.7.3 FieldCodeType プロパティで取得するデータ

Result.FieldCodeType プロパティで設定する取得モードと列属性の関係を表 3-4 に示します。

表 3-4 FieldCodeType プロパティで設定する取得モードと列属性の関係

列属性の文字定数	FieldCodeType プロパティで設定する取得モード	
	dbpTypeCHAR	dbpTypeBINARY
dbpTypeCHAR	◎	○
dbpTypeNUMERIC	◎	○
dbpTypeDECIMAL	◎	○
dbpTypeINTEGER	◎	○
dbpTypeSMALLINT	◎	○
dbpTypeFLOAT	◎	○
dbpTypeREAL	◎	○
dbpTypeDOUBLE	◎	○
dbpTypeDATE	◎	○
dbpTypeTIME	◎	○
dbpTypeTIMESTAMP	◎	○
dbpTypeVARCHAR	◎	○
dbpTypeLONGVARCHAR	◎	○
dbpTypeBINARY	○	◎
dbpTypeVARBINARY	○	◎
dbpTypeLONGVARBINARY	×	◎
dbpTypeBIGINT	◎	○
dbpTypeTINYINT	◎	○
dbpTypeBIT	◎	○
dbpTypeNCHAR	◎	○
dbpTypeNVARCHAR	◎	○
dbpTypeLONGNVARCHAR	◎	○
dbpTypeMCHAR	◎	○
dbpTypeMVARCHAR	◎	○
dbpTypeLONGMVARCHAR	◎	○
dbpTypeNUMERICUNS	◎	○

(凡例)

- ◎：設定できます（省略時仮定値を示します）。
- ：設定できません。
- ×：設定できません。

3.8 データの更新 (UpdateDefine オブジェクト)

DBPARTNER2 Client のオブジェクト UpdateDefine を使って、データを更新する方法です。データを更新するときは、*DBPARTNER.Connect* メソッドでサーバプログラムと接続した後、UpdateDefine オブジェクトを使います。

データの更新は、**更新版と定義版**の DBPARTNER2 Client で実行できます。参照版の DBPARTNER2 Client では、データの更新はできません。

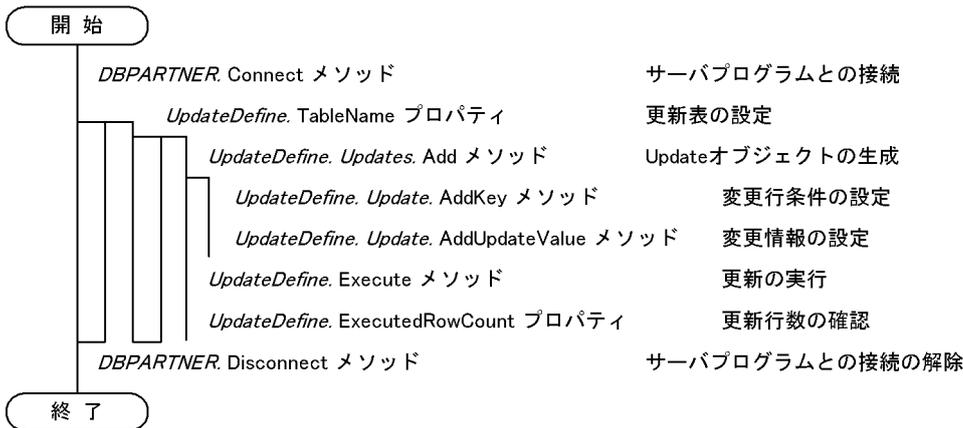
UpdateDefine オブジェクトを使った更新のオブジェクトの流れとプロパティとメソッドの実行手順を図 3-6 に示します。

図 3-6 UpdateDefine オブジェクトを使った更新のオブジェクトの流れとプロパティとメソッドの実行手順

●オブジェクトの流れ



●プロパティとメソッドの流れ



サンプルの参照先

UpdateDefine オブジェクトのサンプルは、DBPARTNER2 Client のフォルダ¥Ocx¥Samples¥Vb ¥Odbc¥Update フォルダを参照してください。

3.8.1 データを更新する手順

UpdateDefine オブジェクトを使ってデータを更新（変更，追加，削除）する手順について説明します。

(1) UpdateDefine オブジェクトを使ってデータを更新する手順（変更）

UpdateDefine オブジェクトを使ってデータを更新（変更）する手順を次に示します。

1. 次に示すプロパティを設定します。

- *UpdateDefine.Exclusive* プロパティ：排他モードを設定します。
- *UpdateDefine.UpdateMode* プロパティ：更新実行モードを設定します。
- *UpdateDefine.TableName* プロパティ：更新する表の名前を設定します。
- *UpdateDefine.NCharPrefixMode* プロパティ：プレフィックス "N" を付けるかどうかを設定します。

2. *UpdateDefine.Updates.Add* メソッドで、Update オブジェクトを生成します。

3. *Updates* コレクションオブジェクト下の Update オブジェクトに、変更するデータに関する条件を設定します。

- *UpdateDefine.Update.UpdateValue* プロパティ：文字列型のデータの変更行条件を設定します。
- *UpdateDefine.Update.UpdateValueB* プロパティ：バイト型のデータの変更行条件を設定します。

4. 変更するデータに関する条件を変更するときは、次に示すメソッドを実行します。

- *UpdateDefine.Update.AddKey* メソッド：変更行条件を設定します。
- *UpdateDefine.Update.AddUpdateValue* メソッド：文字列型のデータを変更します。
- *UpdateDefine.Update.AddUpdateValueB* メソッド：バイト型のデータを変更します。

5. *UpdateDefine.Execute* メソッドを実行します。*UpdateDefine.Execute* メソッドを実行すると、次に示すプロパティの値を参照できます。

- *UpdateDefine.ExecutedRowCount* プロパティ：サーバプログラムが更新した行数。

(2) UpdateDefine オブジェクトを使ってデータを更新する手順 (追加)

UpdateDefine オブジェクトを使ってデータを更新 (追加) する手順を次に示します。

1. 次に示すプロパティを設定します。

- *UpdateDefine.Exclusive* プロパティ：排他モードを設定します。
- *UpdateDefine.UpdateMode* プロパティ：更新実行モードを設定します。
- *UpdateDefine.TableName* プロパティ：更新する表の名前を設定します。
- *UpdateDefine.NCharPrefixMode* プロパティ：プレフィックス "N" を付けるかどうかを設定します。

2. *UpdateDefine.Inserts.Add* メソッドで、Insert オブジェクトを生成します。

3. *Inserts* コレクションオブジェクト下の Insert オブジェクトに、追加するデータに関する条件を設定します。

- *UpdateDefine.Insert.InsertValue* プロパティ：文字列型のデータの追加情報を設定します。
- *UpdateDefine.Insert.InsertValueB* プロパティ：バイト型のデータの追加情報を設定します。

4. 追加するデータに関する条件を変更するときは、次に示すメソッドを実行します。

- *UpdateDefine.Insert.AddInsertValue* メソッド：文字列型のデータを追加します。
 - *UpdateDefine.Insert.AddInsertValueB* メソッド：バイト型のデータを追加します。
5. メソッドを実行した後、*UpdateDefine.Execute* メソッドを実行します。*UpdateDefine.Execute* メソッドを実行すると、次に示すプロパティの値を参照できます。
- *UpdateDefine.ExecutedRowCount* プロパティ：サーバプログラムが更新した行数。更新が正しく実行されたかどうかを確認できます。

(3) UpdateDefine オブジェクトを使ってデータを更新する手順（削除）

UpdateDefine オブジェクトを使ってデータを更新（削除）する手順を次に示します。

- 次に示すプロパティを設定します。
 - *UpdateDefine.Exclusive* プロパティ：排他モードを設定します。
 - *UpdateDefine.UpdateMode* プロパティ：更新実行モードを設定します。
 - *UpdateDefine.TableName* プロパティ：更新する表の名前を設定します。
 - *UpdateDefine.NCharPrefixMode* プロパティ：プレフィックス "N" を付けるかどうかを設定します。
- UpdateDefine.Deletes.Add* メソッドで、Delete オブジェクトを生成します。
- Deletes コレクションオブジェクト下の Delete オブジェクトに、削除するデータに関する条件を設定します。
 - *UpdateDefine.Delete.KeyCount* プロパティ：削除行条件の数を設定します。
 - *UpdateDefine.Delete.KeyColumn* プロパティ：削除行条件の列名を設定します。
 - *UpdateDefine.Delete.KeyValue* プロパティ：削除行条件のデータを設定します。
- 削除するデータに関する条件を変更するときは、次に示すメソッドを実行します。
 - *UpdateDefine.Delete.AddKey* メソッド：削除行条件を設定します。
- メソッドを実行した後、*UpdateDefine.Execute* メソッドを実行します。*UpdateDefine.Execute* メソッドを実行すると、次に示すプロパティの値を参照できます。
 - *UpdateDefine.ExecutedRowCount* プロパティ：サーバプログラムが更新した行数。

3.8.2 更新実行モードとデータ更新の関係

UpdateDefine.UpdateMode プロパティに設定する更新実行モードとデータ更新の関係を表 3-5 に示します。

表 3-5 更新実行モードとデータ更新の関係

更新実行 モード	メソッド	列属性		
		dbpTypeLONGVAR BINARY	dbpTypeVARBINAR Y, dbpTypeBIT	その他の属性
dbpNormal	AddKey メソッド	×	×	○
	AddUpdateValue メソッド	×	×	○
	AddUpdateValueB メソッド	×	×	×
	AddInsertValue メソッド	×	×	○
	AddInsertValueB メソッド	×	×	×
dbpVariable	AddKey メソッド	×	×	○
	AddUpdateValue メソッド	○	×	○
	AddUpdateValueB メソッド	○	○	○
	AddInsertValue メソッド	○	×	○
	AddInsertValueB メソッド	○	○	○

3.9 SQL を直接記述してアクセス (SQLDirect オブジェクト)

アプリケーションに SQL 文を記述して、データを抽出／更新する方法について説明します。SQL 文を直接記述して実行するときは、*DBPARTNER.Connect* メソッドでサーバプログラムと接続した後、SQLDirect オブジェクトを使います。SQLDirect オブジェクトを使うと、データの抽出／更新のほかにも SQL でサポートしている機能を実行できます。

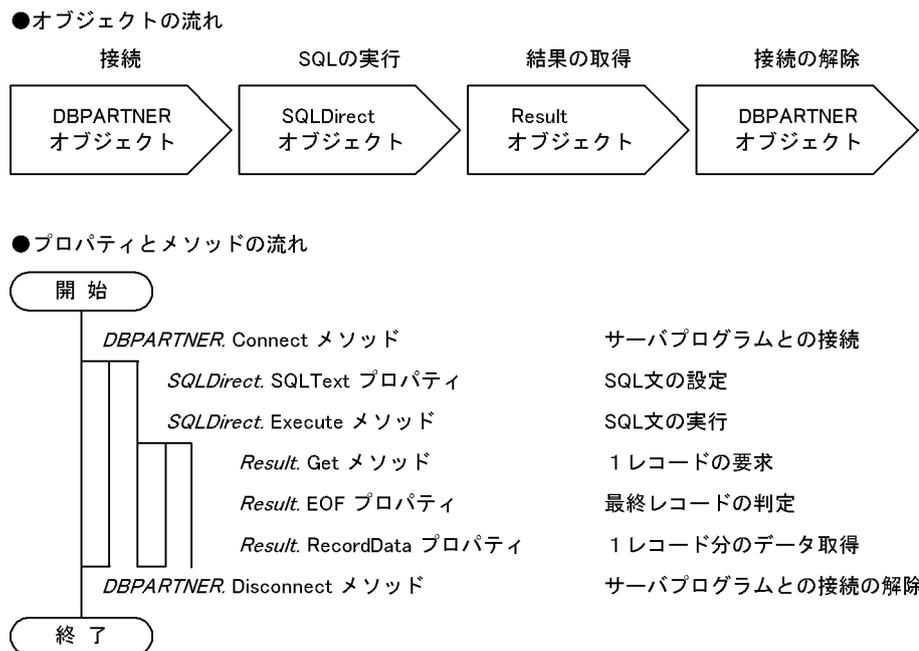
サンプルの参照先

SQLDirect オブジェクトのサンプルは、DBPARTNER2 Client のフォルダ¥Ocx¥Samples¥Vb¥Odbc ¥Sqlirect フォルダを参照してください。

3.9.1 データの抽出 (SQLDirect オブジェクト)

SQLDirect オブジェクトを使った抽出のオブジェクトの流れとプロパティとメソッドの実行手順を図 3-7 に示します。

図 3-7 SQLDirect オブジェクトを使った抽出のオブジェクトの流れとプロパティとメソッドの実行手順



(1) SQLDirect オブジェクトを使ってデータを抽出する手順

SQLDirect オブジェクトを使ってデータを抽出する手順を次に示します。

- 1.*SQLDirect.SQLText* プロパティに、データを抽出する SQL 文を記述します。
- 2.*SQLDirect.Execute* メソッドで、SQL 文を実行します。SQL 文を実行した結果は、次に示す方法で確認できます。

- Select 文の実行結果：Result オブジェクトを使って取得します。
- Select 文以外の実行結果：戻り値で返ります。

SQLDirect.SQLText プロパティに更新処理（SQL 文の Update 文, Insert 文, Delete 文）を記述してあるときは、次に示すプロパティの値を参照できます。

- *SQLDirect.ExecutedRowCount* プロパティ：サーバプログラムが更新した行数。

3. Result オブジェクトで、抽出データを処理します。Result オブジェクトの使い方については、「3.7 抽出データの管理 (Result オブジェクト)」を参照してください。

3.9.2 データの更新 (SQLDirect オブジェクト)

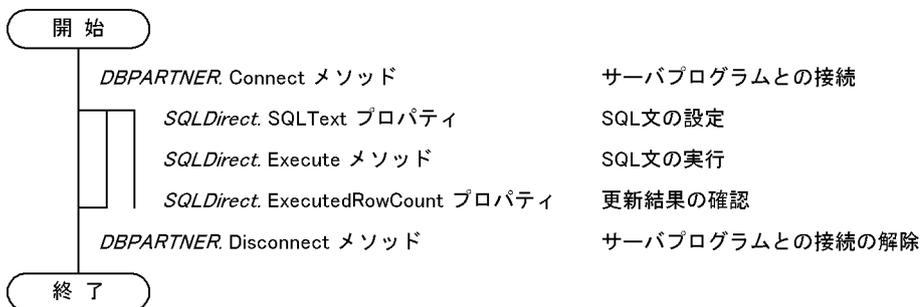
SQLDirect オブジェクトを使った更新のオブジェクトの流れとプロパティとメソッドの実行手順を図 3-8 に示します。

図 3-8 SQLDirect オブジェクトを使った更新のオブジェクトの流れとプロパティとメソッドの実行手順

●オブジェクトの流れ



●プロパティとメソッドの流れ



(1) SQLDirect オブジェクトを使ってデータを更新する手順

SQLDirect オブジェクトを使ってデータを更新する手順を次に示します。

1. *SQLDirect.SQLText* プロパティに、データを更新（変更, 追加, 削除）する SQL 文を記述します。
2. *SQLDirect.Execute* メソッドで、データの更新を実行します。
3. *SQLDirect.SQLText* プロパティに SQL 文の Update 文, Insert 文, Delete 文を記述してあるときは、次に示すプロパティの値を参照できます。
 - *SQLDirect.ExecutedRowCount* プロパティ：サーバプログラムが更新した行数。

SQL 文を実行した結果は、次に示す方法で確認できます。

- Select 文の実行結果：Result オブジェクトを使って取得します。
- Select 文以外の実行結果：戻り値で返ります。

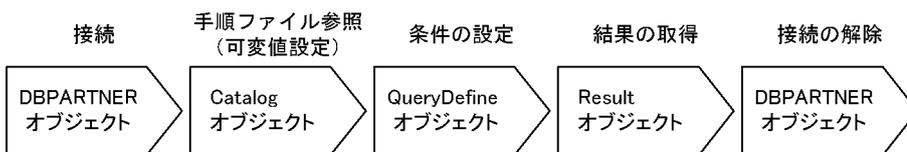
3.10 データの抽出 (Catalog オブジェクト)

DBPARTNER2 Client のオブジェクト Catalog を使って、データを抽出する方法について説明します。データを抽出するときは、*DBPARTNER.Connect* メソッドでサーバプログラムと接続した後、Catalog オブジェクトを使います。

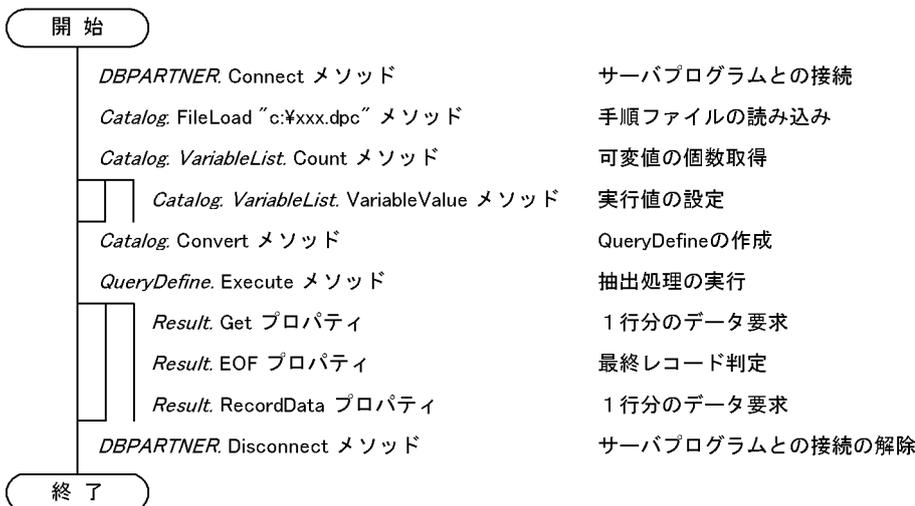
Catalog オブジェクトを使った抽出のオブジェクトの流れとプロパティとメソッドの実行手順を図 3-9 に示します。

図 3-9 Catalog オブジェクトを使った抽出のオブジェクトの流れとプロパティとメソッドの実行手順

●オブジェクトの流れ



●プロパティとメソッドの流れ



3.10.1 Catalog オブジェクトを使ってデータを抽出する手順

Catalog オブジェクトを使ってデータを抽出する手順を次に示します。

- 次に示す Catalog オブジェクトのプロパティを設定します。
 - Catalog.FileLoad* プロパティ：手順ファイル名を設定します。
- Catalog.Convert* メソッドで QueryDefine オブジェクトを作成します。
- QueryDefine.Exclusive* プロパティに、排他モードを設定します。
- QueryDefine.Execute* メソッドで、抽出を実行します。

5. Result オブジェクトで、抽出データを処理します。Result オブジェクトの使い方については、「[3.7 抽出データの管理 \(Result オブジェクト\)](#)」を参照してください。

3.11 ストアドプロシジャの実行 (Procedure オブジェクト)

DBPARTNER2 Client のオブジェクト `Procedure` を使って、ストアドプロシジャを実行する方法について説明します。ストアドプロシジャを実行するときは、`DBPARTNER.Connect` メソッドでサーバプログラムと接続した後、`Procedure` オブジェクトを使います。

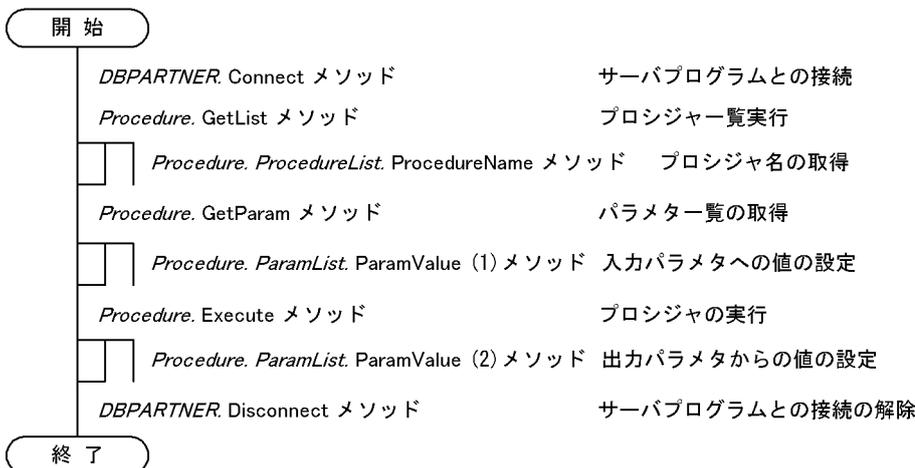
`Procedure` オブジェクトを使ったストアドプロシジャ実行のオブジェクトの流れとプロパティとメソッドの実行手順を図 3-10 に示します。

図 3-10 Procedure オブジェクトを使ったストアドプロシジャ実行のオブジェクトの流れとプロパティとメソッドの実行手順

●オブジェクトの流れ



●プロパティとメソッドの流れ



1 番目のパラメタが入力パラメタ、2 番目のパラメタが出力パラメタの場合

3.11.1 Procedure オブジェクトを使ってストアドプロシジャを実行する手順

`Procedure` オブジェクトを使ってストアドプロシジャを実行する手順を次に示します。

1. 次に示す `Procedure` オブジェクトのプロパティを設定します。

- `Procedure.ListPattern` プロパティ：ストアドプロシジャの一覧を絞り込む条件を設定します。

2. `Procedure.GetList` メソッドで `ProcedureList` オブジェクトを作成します。

3. `ProcedureList.ProcedureName` プロパティに、実行するストアドプロシジャ名を設定します。

4. *Procedure*.GetParam メソッドで ParamList オブジェクトを作成します。
5. *ParamList*.ParamValue プロパティで、ストアドプロシジャに与えるパラメタを設定します。
6. *Procedure*.Execute メソッドで、ストアドプロシジャを実行します。
7. *ParamList*.ParamValue プロパティで、実行結果を取得します。

3.12 避けて欲しい処理

DBPARTNER2 Client のアプリケーションを開発するときには、次に示す処理は作らないでください。このような処理を作ったアプリケーションを DBPARTNER2 Client で使ったときの動作は保証しません。

3.12.1 プロパティとメソッド処理中に、別の入力を許す処理

プロパティとメソッドが処理を実行中に、別の入力を受け取ってプロパティとメソッドを実行する処理は避けてください。プロパティとメソッドが処理中に別のプロパティとメソッドを実行すると、最初に実行したプロパティとメソッド以外はエラーになります。

例えば、ボタンコントロールに対応したコードの中にプロパティとメソッドを記述するときは、ボタンコントロールのコードの最初にコントロールを不活性（入力を受け付けない）に変える処理を記述して、コードの最後に活性（入力を受け付ける）に変える処理を記述してください。

タイマーコントロールなどを使うときは、処理ステータスを管理するフラグを使って、同時に複数のプロパティとメソッドが実行されないようにしてください。

3.12.2 Disconnect メソッドを実行しないでアプリケーションを終了する処理

Disconnect メソッドを実行しないでアプリケーションを終了させると、例外処理が起こるおそれがあります。アプリケーションの終了時には、必ず Disconnect メソッドでサーバプログラムとの接続を終了させてください。

サーバプログラムとの接続状態は、IsConnect プロパティで調べることができます。調べた結果、サーバプログラムとの接続が開始していないときは、Disconnect メソッドを実行する必要はありません。

3.12.3 2 フェーズコミットを必要とする更新処理

一つのアプリケーションで複数の表を更新して、互いの表の整合性を保証する必要があるときは、2 フェーズコミットを使います。しかし、DBPARTNER2 Client の ActiveX コントロールは、2 フェーズコミットをサポートしていません。このため、正常に終了させた更新処理を元の状態に戻せません。

例えば、二つの表を更新する業務で、片方の表に対する更新処理が正常に処理されて、もう片方の更新処理が失敗したときは、二つの表の整合性を保証できなくなります。

3.12.4 Internet Explorer で ActiveX コントロールを使う処理

DBPARTNER2 Client の ActiveX コントロールは、クライアントのアプリケーションを開発するためだけに使ってください。Microsoft ActiveX Control Pad などを使って HTML ファイルに記述を追加できますが、その動作は保証されません。

3.13 プロパティとメソッドの使い方の詳細

プロパティとメソッドを実行するときの、次に示す条件について説明します。

- プロパティとメソッドの実行順序
- プロパティとメソッドの制限

3.13.1 プロパティとメソッドの実行順序

プロパティとメソッドを使うときに前提になるプロパティ又はメソッドを、オブジェクトごとに表 3-6～表 3-14 に示します。

表の項目の「接続」とは、データベースとの接続 (*DBPARTNER.Connect* メソッド) が必要かどうかを示します。必要なときは「○」、特に必要でないときは「-」で示します。

表 3-6 プロパティとメソッドの実行順序 (DBPARTNER オブジェクト)

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
CommitMode	-	-
ConnectDialogMode	-	-
DataSource	-	-
DictPath	-	-
TableDefSource	-	-
UserID	-	-
Password	-	-
IsConnect	-	-
Cancel	○	-
Disconnect	○	-
Commit	○	-
Rollback	○	-
Connect	-	-

表 3-7 プロパティとメソッドの実行順序 (SQLDirect オブジェクト)

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
SQLText	-	-
ExecutedRowCount	○	<i>SQLDirect.Execute</i>

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
Execute	○	<i>SQLDirect</i> .SQLText

表 3-8 プロパティとメソッドの実行順序 (TableList オブジェクト)

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
Pattern	—	—
CountLimit	—	—
Count	○	<i>TableList</i> .Get
SelectDict	—	—
TableName	○	<i>TableList</i> .Get
TableOwner	○	<i>TableList</i> .Get
TableType	○	<i>TableList</i> .Get
TableComment	○	<i>TableList</i> .Get
TableDataSource	○	<i>TableList</i> .Get
TableCreateDate	○	<i>TableList</i> .Get
TableGroupName	○	<i>TableList</i> .Get
TableHeader	○	<i>TableList</i> .Get
Type	—	—
Get	○	—

表 3-9 プロパティとメソッドの実行順序 (ColumnList オブジェクト)

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
TableName	—	—
CountLimit	—	—
Count	○	<i>ColumnList</i> .Get
ColumnName	○	<i>ColumnList</i> .Get
ColumnNameLength	○	<i>ColumnList</i> .Get
ColumnType	○	<i>ColumnList</i> .Get
ColumnLength	○	<i>ColumnList</i> .Get
ColumnScale	○	<i>ColumnList</i> .Get
ColumnComment	○	<i>ColumnList</i> .Get
ColumnHeader	○	<i>ColumnList</i> .Get
SelectDict	—	—

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
Get	○	<i>ColumnList.TableName</i>

表 3-10 プロパティとメソッドの実行順序 (Result オブジェクト)

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
QueryCountLimit	—	—
DataCheck	—	—
HeaderOutput	—	—
RecordHeader	○	<i>QueryDefine.Execute</i> 又は <i>SQLDirect.Execute</i>
FieldHeader	○	<i>QueryDefine.Execute</i> 又は <i>SQLDirect.Execute</i>
FieldNameLength	○	<i>QueryDefine.Execute</i> 又は <i>SQLDirect.Execute</i>
RecordData	○	<i>Result.Get</i>
FieldData	○	<i>Result.Get</i>
FieldDataB	○	<i>Result.Get</i>
FieldCodeType	○	<i>QueryDefine.Execute</i> 又は <i>SQLDirect.Execute</i>
FieldType	○	<i>QueryDefine.Execute</i> 又は <i>SQLDirect.Execute</i>
FieldDataLength	○	<i>Result.Get</i>
FieldCount	○	<i>QueryDefine.Execute</i> 又は <i>SQLDirect.Execute</i>
EOF	○	<i>Result.Get</i>
Get	○	<i>QueryDefine.Execute</i> 又は <i>SQLDirect.Execute</i>
Close	○	<i>QueryDefine.Execute</i> 又は <i>SQLDirect.Execute</i> 又は <i>Result.Get</i>
SaveToFile	○	<i>QueryDefine.Execute</i> 又は <i>SQLDirect.Execute</i>
CopyToClipboard	○	<i>QueryDefine.Execute</i> 又は <i>SQLDirect.Execute</i>
SaveLargeObject	○	<i>Result.Get</i>

表 3-11 プロパティとメソッドの実行順序 (QueryDefine オブジェクト)

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
Exclusive	—	—
QueryCount	○	<i>QueryDefine.Tables.Add</i>
Execute	○	<i>QueryDefine.Tables.Add</i>
RemoveAll	—	—
<i>Tables.Count</i>	—	<i>QueryDefine.Tables.Add</i>
<i>Tables.Add</i>	—	—

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
<i>Tables.RemoveAll</i>	—	—
<i>Table.TableName</i>	—	<i>QueryDefine.Tables.Add</i>
<i>Columns.Count</i>	—	<i>QueryDefine.Columns.Add</i>
<i>Columns.Add</i>	—	—
<i>Columns.RemoveAll</i>	—	—
<i>Column.ColumnName</i>	—	<i>QueryDefine.Columns.Add</i>
<i>Column.SortMode</i>	—	<i>QueryDefine.Columns.Add</i>
<i>Column.SortPriority</i>	—	<i>QueryDefine.Column.SortMode</i>
<i>Column.FunctionMode</i>	—	<i>QueryDefine.Columns.Add</i>
<i>Conditions.Count</i>	—	<i>QueryDefine.Conditions.Add</i>
<i>Conditions.Relation</i>	—	<i>QueryDefine.Conditions.Add</i>
<i>Conditions.Add</i>	—	—
<i>Conditions.RemoveAll</i>	—	—
<i>Condition.Text</i>	—	<i>QueryDefine.Conditions.Add</i>

表 3-12 プロパティとメソッドの実行順序 (UpdateDefine オブジェクト)

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
Exclusive	—	—
UpdateMode	—	—
NCharPrefixMode	—	—
TableName	—	—
ErrorInformation	○	<i>UpdateDefine.Execute</i>
ExecutedRowCount	○	<i>UpdateDefine.Execute</i>
Execute	○	<i>UpdateDefine.TableName</i> と次に示す組み合わせ <ul style="list-style-type: none"> • <i>UpdateDefine.Update.AddKey</i> と <i>UpdateDefine.Update.AddUpdateValue</i> • <i>UpdateDefine.Update.AddKey</i> と <i>UpdateDefine.Update.AddUpdateValueB</i> • <i>UpdateDefine.Insert.AddValue</i> • <i>UpdateDefine.Insert.AddValueB</i> • <i>UpdateDefine.Delete.AddKey</i>
RemoveAll	—	—
<i>Updates.Count</i>	—	<i>UpdateDefine.Updates.Add</i>

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
<i>Updates.Add</i>	—	—
<i>Updates.RemoveAll</i>	—	—
<i>Update.KeyCount</i>	—	<i>UpdateDefine.Update.AddKey</i>
<i>Update.UpdateValueCount</i>	—	<i>UpdateDefine.Update.AddUpdateValue</i>
<i>Update.KeyColumn</i>	—	<i>UpdateDefine.Update.AddKey</i>
<i>Update.KeyValue</i>	—	<i>UpdateDefine.Update.AddKey</i>
<i>Update.Column</i>	—	<i>UpdateDefine.Update.AddUpdateValue</i>
<i>Update.UpdateValue</i>	—	<i>UpdateDefine.Update.AddUpdateValue</i>
<i>Update.UpdateValueB</i>	—	<i>UpdateDefine.Update.AddUpdateValueB</i>
<i>Update.AddKey</i>	—	<i>UpdateDefine.Updates.Add</i>
<i>Update.AddUpdateValue</i>	—	<i>UpdateDefine.Updates.Add</i>
<i>Update.AddUpdateValueB</i>	—	<i>UpdateDefine.Updates.Add</i>
<i>Inserts.Count</i>	—	<i>UpdateDefine.Inserts.Add</i>
<i>Inserts.Add</i>	—	—
<i>Inserts.RemoveAll</i>	—	—
<i>Insert.InsertValueCount</i>	—	<i>UpdateDefine.Insert.AddInsertValue</i>
<i>Insert.Column</i>	—	<i>UpdateDefine.Insert.AddInsertValue</i>
<i>Insert.InsertValue</i>	—	<i>UpdateDefine.Insert.AddInsertValue</i>
<i>Insert.InsertValueB</i>	—	<i>UpdateDefine.Insert.AddInsertValueB</i>
<i>Insert.AddInsertValue</i>	—	<i>UpdateDefine.Inserts.Add</i>
<i>Insert.AddInsertValueB</i>	—	<i>UpdateDefine.Inserts.Add</i>
<i>Deletes.Count</i>	—	<i>UpdateDefine.Deletes.Add</i>
<i>Deletes.Add</i>	—	—
<i>Deletes.RemoveAll</i>	—	—
<i>Delete.KeyCount</i>	—	<i>UpdateDefine.Delete.AddKey</i>
<i>Delete.KeyColumn</i>	—	<i>UpdateDefine.Delete.AddKey</i>
<i>Delete.KeyValue</i>	—	<i>UpdateDefine.Delete.AddKey</i>
<i>Delete.AddKey</i>	—	<i>UpdateDefine.Deletes.Add</i>

表 3-13 プロパティとメソッドの実行順序 (Catalog オブジェクト)

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
Convert	○	<i>Catalog.FileLoad</i>
FileLoad	—	—
<i>Variable.Count</i>	—	<i>Catalog.FileLoad</i>
<i>Variable.VariableName</i>	—	<i>Catalog.FileLoad</i>
<i>Variable.VariableValue</i>	—	<i>Catalog.FileLoad</i>

表 3-14 プロパティとメソッドの実行順序 (Procedure オブジェクト)

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
ListPattern	—	—
Execute	○	<i>Procedure.GetParam</i>
GetList	○	—
GetParam	○	—
<i>ParamList.Count</i>	—	<i>Procedure.GetParam</i>
<i>ParamList.ParamCodeType</i>	○	<i>Procedure.GetParam</i>
<i>ParamList.ParamName</i>	○	<i>Procedure.GetParam</i>
<i>ParamList.ParamType</i>	○	<i>Procedure.GetParam</i>
<i>ParamList.ParamComment</i>	○	<i>Procedure.GetParam</i>
<i>ParamList.ParamDataType</i>	○	<i>Procedure.GetParam</i>
<i>ParamList.ParamLength</i>	○	<i>Procedure.GetParam</i>
<i>ParamList.ParamNullable</i>	○	<i>Procedure.GetParam</i>
<i>ParamList.ParamPrecision</i>	○	<i>Procedure.GetParam</i>
<i>ParamList.ParamRadix</i>	○	<i>Procedure.GetParam</i>
<i>ParamList.ParamScale</i>	○	<i>Procedure.GetParam</i>
<i>ParamList.ParamTypeName</i>	○	<i>Procedure.GetParam</i>
<i>ParamList.ParamValue</i>	○	<i>Procedure.GetParam</i>
<i>ParamList.ParamValueB</i>	○	<i>Procedure.GetParam</i>
<i>ParamList.ParamValueMaxLength</i>	—	<i>Procedure.GetParam</i>
<i>ProcedureList.Count</i>	—	<i>Procedure.GetList</i>
<i>ProcedureList.ProcedureComment</i>	○	<i>Procedure.GetList</i>
<i>ProcedureList.ProcedureName</i>	○	<i>Procedure.GetList</i>

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
<i>ProcedureList.ProcedureOwner</i>	○	<i>Procedure.GetList</i>
<i>ProcedureList.ProcedureType</i>	○	<i>Procedure.GetList</i>

3.13.2 プロパティとメソッドの制限

プロパティとメソッドは、サーバプログラムと接続していないとエラーになることがあります。プロパティとメソッドの状態遷移を表 3-15 に示します。表 3-15 にある「×」は、エラーになることを示します。表 3-15 に記載していないプロパティとメソッドは、状態遷移による制限はありません。

表 3-15 に示す状態は、次に示す 4 種類あります。

- 未接続：サーバプログラムと接続していない状態を示します。
- 接続：サーバプログラムと接続している状態を示します。
- 前処理：Select 系 SQL 文を実行した状態を示します。
- 取得：抽出結果の取得中の状態を示します。

表 3-15 プロパティとメソッドの状態遷移

オブジェクト	プロパティ 又はメソッド	状態			
		1：未接続	2：接続	3：前処理	4：取得
DBPARTNER	Cancel	1※1	2※1	2※2	2※2
	Disconnect	1※3	1	×	×
	Commit	×	2※4	2※5	2※5
	Rollback	×	2※4	2※5	2※5
	Connect	2	2※6	×	×
SQLDirect	Execute	×	3 又は 2※7	×	×
Result	RecordHeader	×	×	3	4
	FieldHeader	×	×	3	4
	RecordData	×	×	×	4
	FieldData	×	×	×	4
	FieldDataB	×	×	×	4
	FieldType	×	×	3	4
	FieldDataLength	×	×	×	4

オブジェクト	プロパティ 又はメソッド	状態			
		1：未接続	2：接続	3：前処理	4：取得
	FieldCount	1※8	2※8	3	4
	Get	×	×	4 又は 2※9	4 又は 2※9
	Close	1※3	2※3	2	2
	SaveLargeObject	×	×	×	4
	SaveToFile	×	×	2	×
	CopyToClipboard	×	×	2	×
TableList	Get	×	2	×	×
ColumnList	Get	×	2	×	×
QueryDefine	QueryCount	×	2	×	×
	Execute	×	3	×	×
UpdateDefine	Execute	×	2	×	×

注※1

Cancel は正常に終了します。

注※2

Cancel が正常に受け付けられたときは遷移します。受け付けられなかったときは遷移しません。

注※3

何も処理しません。

注※4

Commit/Rollback メソッドは正常に終了します。

注※5

Commit/Rollback メソッドを処理した後で、DBPARTNER2 Client が自動的に Close メソッドを実行します。

注※6

現在の接続を終了した後で、接続を再開します。

注※7

Select 系の SQL を実行した場合は「3」。それ以外は「2」に遷移します。

注※8

0 を返します。

注※9

最終データまで取得したとき、又はエラーになったときは、「2」に遷移します。

4

DBPARTNER2 Client のオブジェクトの文法 (ODBC 接続)

この章では、DBPARTNER2 Client のアプリケーションを開発するときに使う ActiveX コントロールのオブジェクトの文法 (ODBC 接続) について説明します。

4.1 プロパティページの設定

DBPARTNER2 Client の ActiveX コントロールで設定できるプロパティページについて説明します。

DBPARTNER2 Client のプロパティページを図 4-1 に示します。

図 4-1 DBPARTNER2 Client のプロパティページ



- **コミットモード (DBPARTNER.CommitMode プロパティ)**

DBPARTNER2 Client が自動的にトランザクションを制御するかどうかを設定します。省略時仮定値は「0-AutoCommit」です。

- 0-AutoCommit

- DBPARTNER2 Client が自動的にトランザクションを制御します。

- 1-ManualCommit

- アプリケーションから Commit/Rollback を実行してトランザクションを制御します。

- **ログイン画面 (DBPARTNER.ConnectDialogMode プロパティ)**

設定した接続先データソースに接続を要求するときに、ODBC が提供する [ログイン] ダイアログボックスを表示するかどうかを設定します。省略時仮定値は「0-非表示」です。

- 0-非表示

- ユーザ ID/パスワードの設定にかかわらず、[ログイン] ダイアログボックスを表示しません。

- 1-省略時表示

- ユーザ ID/パスワードを設定していないときだけ、画面を表示します。

- 2-表示

- ユーザ ID/パスワードの設定にかかわらず、[ログイン] ダイアログボックスを表示します。

- **接続先 (DBPARTNER.DataSource プロパティ)**

接続するサーバプログラムが動作しているサーバマシンに接続する ODBC のデータソース名を設定します。接続先には、ODBC ドライバのセットアップでデータソースに設定/登録してある値を設定してください。文字列は、最大 32 バイト入力できます。

- ユーザ ID (DBPARTNER.UserID プロパティ)

ユーザ ID を設定します。ここに設定した値は、UserID プロパティに設定されます。文字列は、最大 32 バイト入力できます。

- パスワード (DBPARTNER.Password プロパティ)

パスワードを設定します。ここに設定した値は、Password プロパティに設定されます。文字列は、最大 32 バイト入力できます。

4.2 DBPARTNER オブジェクトの詳細

DBPARTNER オブジェクトは、サーバプログラムとの接続を管理するオブジェクトです。このオブジェクトは、DBPARTNER2 Client のトップオブジェクトです。

DBPARTNER オブジェクトのプロパティとメソッドの一覧を次に示します。

●DBPARTNER オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
CommitMode	○	○	トランザクション終了モードを設定します。
ConnectDialogMode	○	○	接続ダイアログボックス表示モードを設定します。
DataSource	○	○	データソース名を文字列で設定します。
DictPath	○	○	見出し辞書を格納しているパスを文字列で設定します。
TableDefSource	○	○	表情報、列情報の取得元を設定します。
IsConnect	—	○	サーバと接続しているかどうかの状態を取得します。
Password	○	○	パスワードを文字列で設定します。
UserID	○	○	ログインするユーザ ID を文字列で設定します。

●DBPARTNER オブジェクトのメソッドの一覧

メソッド	説明
Cancel (CancelEx)	実行中の処理をキャンセルします。
Commit	実行中の処理を有効にして、トランザクションを終了させます。
Connect	サーバとの接続を開始します。
Disconnect	サーバとの接続を終了します。
Rollback	実行中の処理を無効にして、トランザクションを終了させます。

CommitMode プロパティ

機能

サーバプログラムに対するトランザクションを自動的に終了するかどうかを、数値で設定します。このプロパティは、値の取得もできます。

形式

```
Object.CommitMode [= Mode]
```

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

- *Mode*

トランザクション終了モードを、次に示す定数又は数値で設定します。

定数	数値	内容
<code>dbpAuto</code>	0	自動的にトランザクションを終了します。
<code>dbpManual</code>	1	アプリケーションで終了を設定します。

使い方

- CommitMode プロパティに `dbpManual` を設定したときは、トランザクションの終了を Commit メソッド又は Rollback メソッドで設定します。Commit/Rollback メソッドを実行するタイミングを次に示します。
 - Select 文関連以外の SQL 文を `SQLDirect.Execute` メソッドで実行した後
 - `Result.Get` メソッドですべてのデータを取得し終えた後
 - `Result.Close` メソッドでデータ取得を中止した後
 - `TableList.Get` メソッドの後
 - `ColumnList.Get` メソッドの後
 - `QueryDefine.QueryCount` メソッドの後
 - `UpdateDefine.Execute` メソッドの後
- サーバプログラムと接続中 (Connect メソッドから Disconnect メソッドまでの間) は、CommitMode プロパティの値を変更してもトランザクション終了モードを変えることはできません。

ConnectDialogMode プロパティ

機能

Connect メソッドを実行したときに、ODBC が提供する [データソース選択] ダイアログボックスと [ログイン] ダイアログボックスを表示するかどうかを設定します。このプロパティは、値の取得もできます。

形式

```
Object. ConnectDialogMode [= Mode]
```

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

- *Mode*

[データソース選択] ダイアログボックスと [ログイン] ダイアログボックスを表示するかどうかを、次に示す定数又は数値で設定します。

定数	数値	内容
dbpHide	0	ダイアログボックスを表示しません。
dbpCase	1	省略値があるときは、ダイアログボックスを表示します。
dbpShow	2	ダイアログボックスを常に表示します。

使い方

- dbpHide を設定又は*Mode* を省略すると、DataSource/UserID/Password の各プロパティの値に関係なく [データソース選択] ダイアログボックスと [ログイン] ダイアログボックスは表示されません。
- dbpShow を設定すると、DataSource/UserID/Password の各プロパティの値に関係なく [データソース選択] ダイアログボックスと [ログイン] ダイアログボックスが表示されます。
- dbpCase を設定すると、DataSource/UserID/Password の各プロパティの値によって表示するダイアログボックスが異なります。プロパティの設定とダイアログボックスの関係を次に示します。

プロパティ	設定	ダイアログボックスの表示
DataSource	あり	ダイアログボックスは表示しません。
	省略	[データソース選択] ダイアログボックスを表示します。
UserID, Password	両方あり	ダイアログボックスは表示しません。
	どちらかがなし	[ログイン] ダイアログボックスを表示します。

- Connect メソッドを実行した後に ConnectDialogMode プロパティの値を変更しても、接続には影響しません。
- [データソース選択] ダイアログボックスで設定した値は、DataSource プロパティで取得できません。
- [ログイン] ダイアログボックスで設定した値は、UserID プロパティ又は Password プロパティで取得できません。

DataSource プロパティ

機能

Connect メソッドで接続するサーバプログラムが定義されているデータソース名を、文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object.DataSource [= DSN]
```

パラメタ

- *Object*
DBPARTNER オブジェクトへの参照を表すオブジェクト式です。
- *DSN*
データソース名を、0~32 バイトの文字列で設定します。値をクリアするときは、長さ 0 の文字列を設定します。

使い方

- サーバプログラムと接続中（Connect メソッドから Disconnect メソッドまでの間）は、DataSource プロパティの値を変更してもデータソースを変えることはできません。

DictPath プロパティ

機能

見出し辞書を格納しているディレクトリのパスを設定します。このプロパティは、値の取得もできます。

形式

```
Object.DictPath [= Path]
```

パラメタ

- *Object*
DBPARTNER オブジェクトへの参照を表すオブジェクト式です。
- *Path*
パスを、1~256 の文字列で設定します。パスの終端の “¥” は省略できます。

使い方

- 表情報、列情報を見出し辞書から取得する場合は、必ず設定してください。

TableDefSource プロパティ

機能

表情報、列情報を取得するモードを設定します。このプロパティは、値の取得もできます。

形式

```
Object. TableDefSource [= Mode]
```

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

- *Mode*

表情報、列情報の要求先を、定数又は数値で設定します。

定数	数値	情報の要求先
dbpDB	0	表情報、列情報をデータベースから取得します。
dbpDICT	1	表情報、列情報を見出し辞書から取得します。
dbpDBDICT	2	表情報、列情報をデータベースから取得し、ラベル (TableHeader, ColumnHeader) は見出し辞書から取得します。

IsConnect プロパティ

機能

サーバプログラムとの接続状態を BOOLEAN 値で取得します。

形式

```
Object. IsConnect
```

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

戻り値

定数	内容
TRUE (真)	サーバプログラムと接続されています。

定数	内容
FALSE (偽)	サーバプログラムと接続されていません。

使い方

- サーバプログラムの時間監視機能などで強制的に接続が解除されたときは、接続されていなくても TRUE が返ることがあります。

Password プロパティ

機能

UserID プロパティで設定したユーザ ID のパスワードを、文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object.Password [= PWD]
```

パラメタ

- Object*
DBPARTNER オブジェクトへの参照を表すオブジェクト式です。
- PWD*
パスワードを、0~32 バイトの文字列で設定します。値をクリアするときは、長さ 0 の文字列を設定します。

使い方

- サーバプログラムと接続中 (Connect メソッドから Disconnect メソッドまでの間) は、Password プロパティの値を変更しても接続ユーザのパスワードを変えることはできません。

UserID プロパティ

機能

サーバプログラムに接続するためのユーザ ID を、文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object.UserID [= UID]
```

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

- *UID*

ユーザ ID を、0~32 バイトの文字列で設定します。値をクリアするときは、長さ 0 の文字列を設定します。

使い方

- サーバプログラムと接続中 (Connect メソッドから Disconnect メソッドまでの間) は、UserID プロパティの値を変更しても接続ユーザを変えることはできません。

Cancel メソッド (CancelEx メソッド)

機能

サーバプログラムに対して実行中の処理をキャンセルします。

ビジュアル開発ツールの仕様によっては、メソッド名を「Cancel」と記述するとエラーになることがあります。このときは、メソッド名を「CancelEx」と記述してください。

形式

<i>Object</i> .Cancel 又は <i>Object</i> .CancelEx

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

使い方

- Cancel メソッドが有効になる範囲を次に示します。
 - *SQLDirect*.Execute メソッドで Select 文関連の SQL 文を実行してから処理が終了するまで。
 - *QueryDefine*.Execute メソッドを実行してから処理が終了するまで。
 - 1 回目の *Result*.Get メソッドを実行してから処理が終了するまで。ただし、1 回目の *Result*.Get メソッドを発行する直前に Cancel メソッドが発行されている場合、*Result*.Get メソッドの処理をキャンセルします。
 - *TableList*.Get メソッドを実行してから処理が終了するまで。
 - *ColumnList*.Get メソッドを実行してから処理が終了するまで。
 - *QueryDefine*.QueryCount メソッドを実行してから処理が終了するまで。

- *UpdateDefine.Execute* メソッドを実行してから処理が終了するまで。
- 中断する前に取得していたデータは、削除されます。
- 中断した処理の結果は保証されません。
- *UpdateDefine.UpdateMode* プロパティに *dbpVariable* を設定しているときは、Cancel メソッドは使えません。
- *Result.SaveToFile* メソッド、*Result.CopyToClipboard* メソッドでファイル又はクリップボードにデータを出力中に Cancel メソッドを実行すると、キャンセルしたデータまでをファイル又はクリップボードに出力されます。
- AP 開発ツールの仕様によっては、Cancel メソッドを記述するとエラーが発生することがあります。この場合、「CancelEx」と記述してください。

Commit メソッド

機能

サーバプログラムに対して実行したトランザクションを有効にして、トランザクションを終了します。

形式

<i>Object.Commit</i>

パラメタ

- *Object*
DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

使い方

- *CommitMode* プロパティに *dbpAuto* を設定しているときは、正常に処理が終わった時点で自動的に Commit メソッドが実行されるので、アプリケーションから Commit メソッドを実行する必要はありません。
- *CommitMode* プロパティで *dbpAuto* を設定している状態で Commit メソッドを実行しても、無視されます (エラーにはなりません)。

Connect メソッド

機能

DataSource, *UserID*, *Password* の各プロパティに設定された値を使って、サーバプログラムと接続します。

形式

```
Object. Connect
```

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

使い方

- 既にサーバプログラムと接続されている状態で Connect メソッドを実行すると、接続中のセッションを終了させて、新たに接続処理が実行されます。
- 抽出処理の実行中は、Connect メソッドは使えません。使うと Connect メソッドがエラーになります。抽出処理の実行中とは、*QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行してからデータの取得が完了するまで（又は、データの取得を中止してから Close メソッドを実行するまで）のことです。
- Connect メソッドで接続したセッションを終了するときは、Disconnect メソッドを実行してください。

Disconnect メソッド

機能

Connect メソッドで接続したサーバプログラムとの接続を解除します。

形式

```
Object. Disconnect
```

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

使い方

- 抽出処理の実行中は、Disconnect メソッドは使えません。使うと Disconnect メソッドがエラーになります。抽出処理の実行中とは、*QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行してからデータの取得が完了するまで（又は、データの取得を中止してから Close メソッドを実行するまで）のことです。
- Connect メソッドが実行されていない状態で Disconnect メソッドを実行しても無視されます（エラーにはなりません）。

- Connect メソッドでサーバプログラムに接続したときは、アプリケーションを終了させる前に必ず Disconnect メソッドを実行して接続を終了させてください。

Rollback メソッド

機能

サーバプログラムに対して実行したトランザクションを無効 (*QueryDefine.Execute* メソッド, *UpdateDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行する前の状態) にして、トランザクションを終了します。

形式

<i>Object</i> . Rollback

パラメタ

- *Object*
DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

使い方

- CommitMode プロパティに dbpAuto を設定しているときは、エラーで処理が終わった時点で自動的に Rollback メソッドが実行されるので、アプリケーションから Rollback メソッドを実行する必要はありません。
- CommitMode プロパティで dbpAuto を設定している状態で Rollback メソッドを実行しても、無視されます (エラーにはなりません)。
- エラーになった時点までの更新処理を有効にするときは、CommitMode プロパティに dbpManual を設定してください。
- ロールバックをサポートしていないデータベースで Rollback メソッドを発行してもエラーにはなりません。この場合、トランザクションを無効にすることはできません。

4.3 SQLDirect オブジェクトの詳細

SQLDirect オブジェクトは、ユーザが記述した SQL 文を実行するためのオブジェクトです。SQL 文を実行させた結果は、Select 文を実行したときは Result オブジェクトを使って取得できます。Select 文以外を実行したときは、正常に実行されたかどうかの戻り値で返ります。

SQLDirect オブジェクトのプロパティとメソッドの一覧を次に示します。

●SQLDirect オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
ExecutedRowCount	—	○	サーバプログラムが更新した行数を取得します。
SQLText	○	○	SQL 文を文字列で設定します。

●SQLDirect オブジェクトのメソッドの一覧

メソッド	説明
Execute	SQL 文を実行します。

ExecutedRowCount プロパティ

機能

サーバプログラムが更新した行数を、数値で取得します。SQLText プロパティにデータを更新する SQL 文 (Update 文, Insert 文, Delete 文) が記述してあるときに有効です。

形式

```
Object.ExecutedRowCount
```

パラメタ

- *Object*

SQLDirect オブジェクトへの参照を表すオブジェクト式です。

使い方

- *SQLDirect.Execute* メソッドで実行した SQL 文が Update/Insert/Delete 文以外のときは、0 が返ります。
- *SQLDirect.Execute* メソッドを実行する前は、0 を返します。

SQLText プロパティ

機能

サーバプログラムに実行させる SQL 文を、文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object. SQLText [= text]
```

パラメタ

- *Object*
SQLDirect オブジェクトへの参照を表すオブジェクト式です。
- *text*
SQL 文を、1~65,000 バイトの文字列で設定します。長さが 0 の文字列は設定できません。

使い方

- SQLText プロパティの値は、*SQLDirect.Execute* メソッドを実行するときに使われます。*SQLDirect.Execute* メソッドを実行した後に SQLText プロパティの値を変更しても、*SQLDirect.Execute* メソッドの結果を変えることはできません。

Execute メソッド

機能

SQLText プロパティに設定した SQL 文を、サーバプログラムに対して実行します。

形式

```
Object. Execute
```

パラメタ

- *Object*
SQLDirect オブジェクトへの参照を表すオブジェクト式です。

使い方

- Select 文の実行結果は、Result オブジェクトで取得します。結果取得を完了させないで再び *SQLDirect.Execute* メソッドを実行すると、エラーになります。
- Update/Insert/Delete 文を実行してサーバプログラムが更新した行数は、ExecutedRowCount プロパティで取得します。

- Update/Insert/Delete 文以外の SQL 文の実行結果は返りません。

4.4 Result オブジェクトの詳細

Result オブジェクトは、SQLDirect オブジェクトや QueryDefine オブジェクトで実行した抽出処理の結果として出力する抽出データを管理するオブジェクトです。

Result オブジェクトのプロパティとメソッドの一覧を次に示します。

●Result オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
DataCheck	○	○	抽出データの中の文字コードをチェックするかどうかを設定します。
EOF	—	○	データが取得できたかどうかを取得します。
FieldCodeType	○	○	抽出データを取得するときのデータ型を設定します。
FieldCount	—	○	抽出データの列数を返します。
FieldData	—	○	抽出データを1列単位の文字列型データで返します。
FieldDataB	—	○	抽出データを1列単位のバイト型データで返します。
FieldDataLength	—	○	列のデータ長を返します。
FieldHeader	○	○	設定した列にヘッダを設定します。
FieldNameLength	—	○	列の列名長を返します。
FieldType	—	○	列の属性を返します。
HeaderOutput	○	○	抽出データの先頭行に列名を付けるかどうかを設定します。
QueryCountLimit	○	○	抽出処理の結果として出力する行数を、数値で設定します。
RecordData	—	○	抽出データを1行単位の文字列型データで返します。
RecordHeader	—	○	抽出処理の結果として出力される列名を、1行のテキスト形式で取得します。

●Result オブジェクトのメソッドの一覧

メソッド	説明
Close	抽出処理を終了させます。
CopyToClipboard	抽出データをクリップボードに出力します。
Get	1行分のデータを要求します。
SaveLargeObject	dbpTypeLONGVARBINARY 属性のデータをファイルに出力します。
SaveToFile	抽出データをファイルに出力します。

DataCheck プロパティ

機能

抽出データの中に文字コード以外のデータがあったときに、エラーを返すかどうかを設定します。このプロパティは、値の取得もできます。

文字コード以外のデータと判定するデータを、次に示します。

- 1 バイト文字で X'20'~X'7E', 又は X'A1'~X'DF'以外の文字。
- 2 バイト文字で 1 バイト目が, X'81'~X'9F', 又は X'E0'~X'FC'以外の文字。
- 2 バイト文字で 2 バイト目が, X'40'~X'7E', 又は X'80'~X'FC'以外の文字。

形式

```
Object.DataCheck [= CheckMode]
```

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。
- *CheckMode*
文字コードをチェックするかどうかを、次に示す定数で設定します。

定数	内容
FALSE	文字コードをチェックしません。
TRUE	文字コード以外のデータがあれば、エラーを返します。

使い方

- DataCheck プロパティの値は、*QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行するときに使われます。*QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行した後で DataCheck プロパティの値を変更しても、*Result.Get* メソッドの結果を変えることはできません。

EOF プロパティ

機能

直前に実行した *Result.Get* メソッドでデータが取得できたかどうかを、BOOLEAN 値で取得します。

形式

`Object.EOF`

パラメタ

- *Object*

Result オブジェクトへの参照を表すオブジェクト式です。

戻り値

定数	内容
FALSE (偽)	データを読み込みました。
TRUE (真)	読み込むデータがありません。

使い方

- EOF プロパティは、データの最後を知るために使います。*Result.Get* メソッドの次に EOF プロパティを実行すると、すべての抽出データを取得し終えたかどうかを判断できます。EOF プロパティが TRUE を返したときは、直前の *Result.Get* メソッドで読み込むデータがなかったことが分かります。

FieldCodeType プロパティ

機能

抽出データを取得するときのデータ型を設定します。このプロパティは、値の取得もできます。

形式

`Object.FieldCodeType (Index) [= Mode]`

パラメタ

- *Object*

Result オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1~FieldCount プロパティの範囲内で設定します。

- *Mode*

取得モードを、次に示す定数又は数値で設定します。

定数	数値	内容
<code>dbpTypeBINARY</code> , <code>dbpBINARY</code>	-2	バイト型を設定します。

定数	数値	内容
dbpTypeCHAR, dbpCHAR	1	文字列型を設定します。

使い方

- 取得モードの設定によって、データを取得するプロパティ/メソッドが異なります。取得モードとデータを取得するプロパティ/メソッドの関係については、「[3.7.2 Result オブジェクトと取得するデータの関係](#)」を参照してください。
- 列番号は、*QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行した後、FieldCount プロパティで得られた列数の範囲内で指定してください。
- *QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行する前に FieldCodeType プロパティを使うと、エラーになります。
- 列属性によって、設定できない取得モードがあります。設定できない取得モードについては、「[3.7.3 FieldCodeType プロパティで取得するデータ](#)」を参照してください。
- FieldCodeType プロパティで設定した値は、次に *QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行するまで有効です。再び *QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行すると、「[3.7.3 FieldCodeType プロパティで取得するデータ](#)」で示す省略値のモードに戻ります。

FieldCount プロパティ

機能

抽出データの列数を、数値で取得します。

形式

Object.FieldCount

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。

使い方

- *QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行する前は、0 を返します。

FieldData プロパティ

機能

列番号で設定した列のデータを、文字列型で取得します。

形式

Object. FieldData (*Index*)

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的の列の列番号を、数値で設定します。番号は、1~FieldCount プロパティの範囲内で設定します。

使い方

- 列属性が dbpTypeBINARY, dbpTypeVARBINARY のデータをバイト型で取得するときは、FieldDataB プロパティを使ってください。
- 列属性が dbpTypeLONGVARBINARY のデータは、データ長が文字変換されて返されます。バイト型で取得するには、FieldDataB プロパティ又は SaveLargeObject メソッドを使ってください。
- 列番号は、*QueryDefine*.Execute メソッド又は *SQLDirect*.Execute メソッドを実行した後で、FieldCount プロパティで得られた列数の範囲内で指定してください。
- *QueryDefine*.Execute メソッド又は *SQLDirect*.Execute メソッドを実行する前に FieldData プロパティを使うと、エラーになります。

FieldDataB プロパティ

機能

列番号で設定した列のデータを、バイト型で取得します。

形式

Object. FieldDataB (*Index*)

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1～FieldCount プロパティの範囲内で設定します。

使い方

- データを格納する変数は、**バイト型**で宣言してください。データを格納する変数がバイト型以外の場合は、エラーになります。
- Visual Basic を使う場合の例を次に示します（データを DataBuff に格納します）。
宣言の例：Dim DataBuff() As Byte
- 列属性が dbpTypeBINARY, dbpTypeVARBINARY, dbpTypeLONGVARBINARY 以外の列のデータをバイト型で取得するときは、FieldCodeType プロパティを使ってください。
- 列番号は、*QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行した後で、FieldCount プロパティで得られた列数の範囲内で指定してください。
- *QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行する前に FieldDataB プロパティを使うと、エラーになります。

FieldDataLength プロパティ

機能

列番号で設定した列のデータ長を、数値で取得します。

形式

<i>Object</i> .FieldDataLength (<i>Index</i>)

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的の列の列番号を、数値で設定します。番号は、1～FieldCount プロパティの範囲内で設定します。

使い方

- 列番号は、*QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行した後、FieldCount プロパティで得られた列数の範囲内で設定してください。
- *QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行する前に FieldDataLength プロパティを使うと、エラーになります。

FieldHeader プロパティ

機能

列番号で設定した列にヘッダを、数値と文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object. FieldHeader (Index) [= String]
```

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的の列の列番号を、数値で設定します。番号は、1~FieldCount プロパティの範囲内で設定します。
- *String*
列に付けるヘッダを、0~256 バイトの文字列で設定します。値をクリアするときは、長さ 0 の文字列を設定します。

使い方

- 列番号は、*QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行した後、FieldCount プロパティで得られた列数の範囲内で設定してください。
- *QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行する前に FieldHeader プロパティを使うと、エラーになります。

FieldNameLength プロパティ

機能

列番号で設定した列の列名長を、数値（バイト数）で取得します。列が繰り返し列又は配列列以外の場合、0 を返します。

形式

```
Object. FieldNameLength (Index)
```

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1～FieldCount プロパティの範囲内で設定します。

使い方

- Result.FieldHeader プロパティで列名を取得した場合、繰り返し列又は配列列は、"列名(n)" (n は要素番号) の形式で返されます。FieldNameLength プロパティを使うと、列名部分の長さ (括弧と要素番号を除いた部分の長さ) を取得できます。
- 列番号は、*QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行した後、FieldCount プロパティで得られた列数の範囲内で設定してください。
- *QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行する前に FieldNameLength プロパティを使うと、エラーになります。

FieldType プロパティ

機能

列番号で設定した列の属性を、数値で取得します。

形式

```
Object.FieldType (Index)
```

パラメタ

- *Object*

Result オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1～FieldCount プロパティの範囲内で設定します。

使い方

- 列の属性については、「付録 E 列の属性」を参照してください。
- 列番号は、*QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行した後に、FieldCount プロパティで得られた列数の範囲内で設定してください。
- *QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行する前に FieldType プロパティを使うと、エラーになります。

HeaderOutput プロパティ

機能

SaveToFile メソッド、CopyToClipboard メソッドを使うときに、抽出データの先頭行にヘッダを付けるかどうかを設定します。このプロパティは、値の取得もできます。

形式

```
Object.HeaderOutput [= HeaderMode]
```

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。
- *HeaderMode*
ヘッダを付けるかどうかを、次に示す定数で設定します。

定数	内容
FALSE	抽出データにヘッダを付けません。
TRUE	抽出データの 1 行目にヘッダを付けます。

使い方

- HeaderOutput プロパティの値は、SaveToFile メソッド又は CopyToClipboard メソッドを実行するときに使われます。SaveToFile メソッド又は CopyToClipboard メソッドを実行した後で HeaderOutput プロパティの値を変更しても SaveToFile メソッド又は CopyToClipboard メソッドの結果を変えることはできません。
- ヘッダが設定されていない列は、列名を出力します。

QueryCountLimit プロパティ

機能

抽出処理の結果として出力する行数を、数値で設定します。このプロパティは、値の取得もできます。

形式

```
Object.QueryCountLimit [= Number]
```

パラメタ

- *Object*

Result オブジェクトへの参照を表すオブジェクト式です。

- **Number**

抽出する行数の制限値を、次に示す数値で設定します。

数値	内容
0	行数は制限しません。
1~999,999,999	設定した行数まで、データを取得します。

使い方

- QueryCountLimit プロパティの値は、*QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行するときに使われます。*QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行した後で QueryCountLimit プロパティの値を変更しても、*Result.Get* メソッドの結果を変えることはできません。

RecordData プロパティ

機能

Result.Get メソッドで取得した 1 レコード単位の抽出データを、文字列で取得します。取得する文字列は、1 行のテキスト形式になっています。

形式

Object. RecordData

パラメタ

- *Object*

Result オブジェクトへの参照を表すオブジェクト式です。

使い方

- テキストデータのフォーマットについては、「付録 C DBPARTNER2 Client が扱うデータの形式」を参照してください。
- データは、*Result.Get* メソッドを実行するときに更新されます。
- 列属性が *dbpTypeLONGVARBINARY* のデータを RecordData プロパティで取得すると、データ長が文字変換されて返されます。バイト型でデータを取得するときは、*FieldDataB* プロパティ又は *SaveLargeObject* メソッドを使ってください。
- 列属性が *dbpTypeBINARY*, *dbpTypeVARBINARY* のデータを RecordData プロパティで取得すると、データ長が文字変換されて返されます。バイト型でデータを取得するときは、*FieldDataB* プロパティを使ってください。

- *QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行する前に *RecordData* プロパティを使うと、エラーになります。

RecordHeader プロパティ

機能

抽出処理の結果として出力される列名を、文字列で取得します。取得する文字列は、1行のテキスト形式になっています。

形式

```
Object. RecordHeader
```

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。

使い方

- テキスト形式のフォーマットについては、「[付録 C DBPARTNER2 Client が扱うデータの形式](#)」を参照してください。
- *QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行する前に *RecordHeader* プロパティを使うと、エラーになります。

Close メソッド

機能

抽出処理を終了させます。

形式

```
Object. Close
```

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。

使い方

- EOF プロパティが FALSE の状態でほかの処理に移りたいときは、必ず Close メソッドを実行してください。
- Close メソッドを連続して実行しても、不要な Close メソッドは無視されます（エラーにはなりません）。

CopyToClipboard メソッド

機能

抽出データを、設定した形式でクリップボードに出力します。

形式

Object.CopyToClipboard (*Type*)

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。
- *Type*
出力するデータの形式を、次に示す定数又は数値で設定します。

定数	数値	内容
dbpTextFormat	0	TEXT 形式
dbpBiff4Format	1	Excel 4.0 形式
dbpOleBiff8Format	4	Excel 97-2000 ブック形式

使い方

- データの形式については、「[付録 C DBPARTNER2 Client が扱うデータの形式](#)」を参照してください。
- *QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行する前に CopyToClipboard プロパティを使うと、エラーになります。
- 列属性が dbpTypeLONGVARBINARY のデータを CopyToClipboard メソッドで取得すると、データ長が文字変換されて返されます。バイト型でデータを取得するときは、FieldDataB プロパティ又は SaveLargeObject メソッドを使ってください。
- 列属性が dbpTypeBINARY, dbpTypeVARBINARY のデータを CopyToClipboard メソッドで取得すると、データ長が文字変換されて返されます。バイト型でデータを取得するときは、FieldDataB プロパティを使ってください。

- Cancel メソッドを実行して処理を中止すると、Cancel メソッドまでのデータをクリップボードに出力します。
- CopyToClipboard メソッドがエラーになったときは、クリップボードの内容は保証されません。
- 抽出するデータが 1 件もないときは、クリップボードの内容がクリアされます。HeaderOutput プロパティに TRUE を設定して、抽出データの先頭行にヘッダを付けると、先頭行のヘッダだけが出力されます。

Get メソッド

機能

サーバプログラムに対して 1 行分のデータを要求します。

形式

```
Object. Get
```

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。

使い方

- データを取得するときは、RecordData プロパティと FieldData プロパティを使います。
- *Result.Get* メソッドを実行した後は、EOF プロパティで取得状況を確認してください。
- *QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行する前に *Result.Get* メソッドを使うと、エラーになります。
- 1 回目の *Result.Get* メソッドが成功して それ以降でデータの取得を中止するときは、Close メソッドを使ってください。Cancel メソッドでは、データの取得を中止できません。

SaveLargeObject メソッド

機能

列番号で設定した dbpTypeLONGVARBINARY 属性の列のデータを、設定したファイルに格納します。

形式

```
Object. SaveLargeObject Index, FileName
```

パラメタ

- *Object*

Result オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1~FieldCount プロパティの範囲内で設定します。

- *FileName*

データを出力するファイル名を、1~256 バイトの文字列で設定します。この設定で、値はクリアできません。

使い方

- 列番号が dbpTypeLONGVARIABLE 属性の列以外の場合は、エラーを返します。
- 列番号は、*QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行した後、FieldCount プロパティで得られた列数の範囲内で指定してください。
- *QueryDefine.Execute* メソッド又は *SQLDirect.Execute* メソッドを実行する前に SaveLargeObject プロパティを使うと、エラーになります。

SaveToFile メソッド

機能

抽出データを、設定した形式でファイルに出力します。

形式

```
Object. SaveToFile FileName, Type
```

パラメタ

- *Object*

Result オブジェクトへの参照を表すオブジェクト式です。

- *FileName*

抽出データを出力するファイル名を、1~256 バイトの文字列で設定します。値はクリアできません。

- *Type*

出力するデータの形式を、次に示す定数又は数値で設定します。

定数	数値	内容
dbpTextFormat, dbpTextFile	0	TEXT 形式
dbpBiff4Format, dbpBiff4File	1	Excel 4.0 形式

定数	数値	内容
dbpCSVFile	2	CSV 形式
dbpDATFile	3	DAT 形式
dbpOleBiff8File	4	Excel オブジェクトで作成する Excel 97-2003 ブック形式 (XLS 形式)
dbpOleBiff12File	5	Excel オブジェクトで作成する Excel 2007 バイナリファイル形式 (XLSB 形式)
dbpOleXMLWorkbookFile	6	Excel オブジェクトで作成する Excel 2007XML ファイル形式 (XLSX 形式)

使い方

- データの形式については、「[付録 C DBPARTNER2 Client が扱うデータの形式](#)」を参照してください。
- `QueryDefine.Execute` メソッド又は `SQLDirect.Execute` メソッドを実行する前に `SaveToFile` プロパティを使うと、エラーになります。
- 列属性が `dbpTypeLONGVARIABLE` のデータを `SaveToFile` メソッドで取得すると、データ長が文字変換されて返されます。バイト型でデータを取得するときは、`FieldDataB` プロパティ又は `SaveLargeObject` メソッドを使ってください。
- 列属性が `dbpTypeBINARY`, `dbpTypeVARIABLE` のデータを `SaveToFile` メソッドで取得すると、データ長が文字変換されて返されます。バイト型でデータを取得するときは、`FieldDataB` プロパティを使ってください。
- `Cancel` メソッドを実行して処理を中止すると、`Cancel` メソッドまでのデータをファイルに出力します。
- `SaveToFile` メソッドがエラーになったときは、出力されたデータの内容は保証されません。

4.5 TableList オブジェクトの詳細

TableList オブジェクトは、設定したデータソースに格納されている表一覧情報を取得したり、取得した結果を管理したりします。それぞれの表情情報は、プロパティによって返されます。

TableList オブジェクトのプロパティとメソッドの一覧を次に示します。

●TableList オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	取得した表情情報の個数を取得します。
CountLimit	○	○	取得したい表情情報の個数を設定します。
SelectDict	○	○	接続先をキーに、使用する見出し辞書を限定します。
Pattern	○	○	表情情報の絞り込み条件を文字列で設定します。
TableComment	—	○	設定した表の注釈を文字列で取得します。
TableCreateDate	—	○	設定した VOSK スプールファイルの作成日付を文字列で取得します。
TableDataSource	—	○	設定した表の見出し辞書中の接続先を文字列で取得します。
TableGroupName	—	○	設定した表のグループ名を文字列で取得します。
TableHeader	—	○	設定した表のラベルを文字列で取得します。
TableName	—	○	設定した表の表名を文字列で取得します。
TableOwner	—	○	設定した表の所有者を文字列で取得します。
TableType	—	○	設定した表の属性を文字列で取得します。
Type	○	○	取得したい表情情報の表属性を設定します。

●TableList オブジェクトのメソッドの一覧

メソッド	説明
Get	表情情報の取得を要求します。

Count プロパティ

機能

取得した表情情報の個数を、数値で取得します。

形式

Object.Count

パラメタ

- *Object*

TableList オブジェクトへの参照を表すオブジェクト式です。

使い方

- *TableList.Get* メソッドを実行する前は、0 を返します。

CountLimit プロパティ

機能

取得する表情報の最大数を、数値で設定します。このプロパティは、値の取得もできます。

形式

```
Object.CountLimit [= max]
```

パラメタ

- *Object*

TableList オブジェクトへの参照を表すオブジェクト式です。

- *max*

取得する表の制限数を、次に示す数値で設定します。

数値	内容
0	すべての表の情報を取得します。
1~8192	設定した数だけ、表の情報を取得します。

使い方

- CountLimit プロパティの値は、*TableList.Get* メソッドを実行するときに使われます。*TableList.Get* メソッドを実行した後で CountLimit プロパティの値を変更しても、*TableList.Get* メソッドの結果を変えることはできません。

SelectDict プロパティ

機能

接続先をキーに、使用する見出し辞書を限定します。このプロパティは、値の取得もできます。

形式

```
Object. SelectDict = DataSource
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。
- *DataSource*
接続先を 0~32 バイトの文字列で設定します。0 バイトの文字列も設定できます。

使い方

- 接続先には、*TableList.TableDataSource* プロパティと同じ文字列を設定してください。
- 接続先が見出し辞書にない場合、*TableList.Count* プロパティが 0 となります。

Pattern プロパティ

機能

表情情報を取得するときの絞り込み条件を、文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object. Pattern [= text]
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。
- *text*
絞り込み条件を、0~128 バイトの文字列で設定します。文字列には、ワイルドカード (_, %) 文字を含めることができます。値をクリアするときは、長さ 0 の文字列を設定します。絞り込み条件は、次の書式で設定してください。
 - 所有者.表名
 - 表名

使い方

- 絞り込み条件に設定する所有者や表名の最大長は、使うデータベースに依存します。制限を超えた絞り込み条件を設定すると、*TableList.Get* メソッドがエラーになるか、又は正しく絞り込みできません。
- 値がクリアされているときは、%.% が仮定されます。

- 「_」 (アンダースコア文字) は何らかの単一文字を示します。
- 「%」 (パーセント文字) は何らかのゼロ個以上の文字のつながりを示します。
- 使える絞り込み条件を次に示します。
 - 前方一致 (例: ABC%) 文字列の先頭が ABC であり, 以降の文字は任意です。
 - 後方一致 (例: %ABC) 文字列の最後が ABC であり, 以前の文字は任意です。
 - 任意一致 (例: %ABC%) 文字列中の任意の部分が ABC です。
 - 完全一致 (例: ABC) 文字列が ABC と同じです。
 - 部分一致 (例: _ABC_) 5 文字の文字列のうち, 2~4 文字目が ABC です。
(例: A_B%C) 文字列の先頭が A, 3 文字目が B, 最後が C です。
- 「_」 や 「%」 を条件に含めるときは, ワイルドカード文字の前に 「¥」 を付けます。
(例) ABC¥%% : 文字列の先頭が ABC% で, 以降の文字は任意です。
- Pattern プロパティの値は, *TableList.Get* メソッドを実行するときに使われます。*TableList.Get* メソッドを実行した後で Pattern プロパティの値を変更しても, *TableList.Get* メソッドの結果を変えることはできません。

TableComment プロパティ

機能

表番号で設定した表の注釈を, 文字列で取得します。

形式

```
Object.TableComment (Index)
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的の表の表番号を, 数値で設定します。番号は, 1~*TableList.Count* プロパティの範囲内で設定します。

使い方

- 表番号は, *TableList.Get* メソッドを実行した後, *TableList.Count* プロパティで得られた表数の範囲内で設定してください。
- 接続するデータベースが VOSK スプールファイルの場合は, ジョブに付けたコメント情報が返ります。

TableCreateDate プロパティ

機能

表番号で指定した VOSK スプールファイルの作成日付を文字列で取得します。

形式

```
Object. TableCreateDate (Index)
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的の表の表番号を、数値で設定します。番号は、1～*TableList.Count* プロパティの範囲内で設定します。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。
- *TableCreateDate* プロパティは、接続するデータベースが VOSK スプールファイルの場合だけ値が取得できます。それ以外のデータベースに対して発行した場合は NULL が返ります。

TableDataSource プロパティ

機能

表番号で設定した表の見出し辞書の接続先を、文字列で取得します。

形式

```
Object. TableDataSource (Index)
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的の表の表番号を、数値で設定します。番号は、1～*TableList.Count* プロパティの範囲内で設定します。

使い方

- 表番号は、*TableList.Get* プロパティを発行した後、*TableList.Count* プロパティで取得した表の数の範囲内で設定してください。

TableGroupName プロパティ

機能

表番号で指定した表のグループ名を、文字列で取得します。

形式

```
Object.TableGroupName(Index)
```

パラメタ

- Object*
TableList オブジェクトへの参照を表すオブジェクト式です。
- Index*
目的の表を数値で設定します。番号は、1～*TableList.Count* プロパティの範囲内で設定します。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。
- TableDefSource* プロパティに *dbpDICT* を設定している場合、見出し辞書のグループ名を返します。
- TableDefSource* プロパティに *dbpDBDICT* を設定している場合、見出し辞書の中に異なる接続先で同一の表名があるときは *TableList.SelectDict* プロパティで接続先を設定してください。
TableList.SelectDict プロパティで接続先を設定しないと、正しいラベルが取得できないことがあります。

TableHeader プロパティ

機能

表番号で設定した表のラベルを、文字列で取得します。

形式

```
Object.TableHeader (Index)
```

パラメタ

- *Object*

TableList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の表の表番号を、数値で設定します。番号は、1～*TableList.Count* プロパティの範囲内で設定します。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。
- *TableDefSource* プロパティに *dbpDICT*、又は *dbpDBDICT* を設定している場合、見出し辞書のラベルを返します。
- *TableDefSource* プロパティに *dbpDBDICT* を設定している場合、見出し辞書の中に異なる接続先で同一の表名があるときは *TableList.SelectDict* プロパティで接続先を設定してください。
TableList.SelectDict プロパティで接続先を設定しないと、正しいラベルが取得できないことがあります。
- 接続するデータベースが VOSK スプールファイルの場合は、ジョブ名称が返ります。

TableName プロパティ

機能

表番号で設定した表の表名を、文字列で取得します。

形式

```
Object.TableName (Index)
```

パラメタ

- *Object*

TableList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の表の表番号を、数値で設定します。番号は、1～*TableList.Count* プロパティの範囲内で設定します。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。

- 接続するデータベースが VOSK スプールファイルの場合は、ジョブ通し番号とファイル番号を連結した文字列が返ります。

TableOwner プロパティ

機能

表番号で設定した表の所有者を、文字列で取得します。

形式

```
Object.TableOwner (Index)
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的の表の表番号を、数値で設定します。番号は、1 ~ *TableList.Count* プロパティの範囲内で設定します。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。
- 接続するデータベースが VOSK スプールファイルの場合は、ジョブを実行したユーザ ID が返ります。

TableType プロパティ

機能

表番号で設定した表の属性を、文字列で取得します。

形式

```
Object.TableType (Index)
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。
- *Index*

目的の表の表番号を、数値で設定します。番号は、1~*TableList.Count* プロパティの範囲内で設定します。

使い方

- 表属性には、"TABLE", "VIEW", "ALIAS", "SYNONYM", 又はサーバプログラム固有の表属性があります。
- 表番号は、*TableList.Get* メソッドを実行した後で、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。

Type プロパティ

機能

取得する表の属性を文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object.Type [= Text]
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。
- *text*
表の表属性を、0~256 バイトの文字列で設定します。次の書式で設定してください。
 - "'表属性'[, '表属性', ...]"なお、長さ 0 の文字列を設定すると、["'TABLE','VIEW','ALIAS','SYNONYM'"] が仮定されます。

使い方

- 設定できる表属性は次のとおりです。
 - "TABLE"
 - "VIEW"
 - "SYSTEM TABLE"
 - "GLOBAL TEMPORARY"
 - "LOCAL TEMPORARY"
 - "ALIAS"
 - "SYNONYM"
- TableDefSource プロパティに dbpDICT を設定している場合、このプロパティの設定は無視されます。

Get メソッド

機能

サーバプログラムに定義されている表情情報を要求します。

形式

```
Object. Get
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。

使い方

- このメソッドを実行すると、前回 *TableList.Get* メソッドで取得した表の情報は上書きされます。

4.6 ColumnList オブジェクトの詳細

ColumnList オブジェクトは、設定された表名に定義されている列一覧情報を取得したり、取得した結果を管理したりします。それぞれの列情報は、プロパティによって返されます。

ColumnList オブジェクトのプロパティとメソッドの一覧を次に示します。

●ColumnList オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
ColumnComment	—	○	設定した列の注釈を文字列で取得します。
ColumnHeader	—	○	設定した列のラベルを文字列で取得します。
ColumnLength	—	○	設定した列の定義長を数値で取得します。
ColumnName	—	○	設定した列の列名を文字列で取得します。
ColumnNameLength	—	○	設定した列の列名長を数値で取得します。
ColumnScale	—	○	設定した列の小数点以下のけた数を数値で取得します。
ColumnType	—	○	設定した列の属性を取得します。
Count	—	○	取得した列の個数を取得します。
CountLimit	○	○	列情報を取得する個数を設定します。
SelectDict	○	○	接続先をキーに、使用する見出し辞書を限定します。
TableName	○	○	列情報を取得したい表名を、文字列で設定します。

●ColumnList オブジェクトのメソッドの一覧

メソッド	説明
Get	列情報の取得を要求します。

ColumnComment プロパティ

機能

列番号で設定した列の注釈を、文字列で取得します。

形式

```
Object.ColumnComment (Index)
```

パラメタ

- *Object*

ColumnList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1~*ColumnList.Count* プロパティの範囲内で設定します。

使い方

- 列番号は、*ColumnList.Get* メソッドを実行した後、*ColumnList.Count* プロパティで得られた列数の範囲内で設定してください。

ColumnHeader プロパティ

機能

列番号で設定した列のラベルを、文字列で取得します。

形式

Object.ColumnHeader (Index)

パラメタ

- *Object*

ColumnList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1~*ColumnList.Count* プロパティの範囲内で設定します。

使い方

- 列番号は、*ColumnList.Get* メソッドを実行した後、*ColumnList.Count* プロパティで得られた列数の範囲内で設定してください。
- *TableDefSource* プロパティに *dbpDICT*、又は *dbpDBDICT* を設定している場合、見出し辞書のラベルを返します。
- *TableDefSource* プロパティに *dbpDBDICT* を設定している場合、見出し辞書の中に異なる接続先で同一の表名があるときは *TableList.SelectDict* プロパティで接続先を設定してください。
TableList.SelectDict プロパティで接続先を設定しないと、正しいラベルが取得できないことがあります。

ColumnLength プロパティ

機能

列番号で設定した列の定義長を、数値で取得します。

形式

```
Object.ColumnLength (Index)
```

パラメタ

- *Object*

ColumnList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1~*ColumnList.Count* プロパティの範囲内で設定します。

使い方

- 列番号は、*ColumnList.Get* メソッドを実行した後、*ColumnList.Count* プロパティで得られた列数の範囲内で設定してください。

ColumnName プロパティ

機能

列番号で設定した列の列名を、文字列で取得します。

形式

```
Object.ColumnName (Index)
```

パラメタ

- *Object*

ColumnList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1~*ColumnList.Count* プロパティの範囲内で設定します。

使い方

- 列番号は、*ColumnList*.Get メソッドを実行した後、*ColumnList*.Count プロパティで得られた列数の範囲内で設定してください。

ColumnNameLength プロパティ

機能

列番号で設定した列の列名長を、数値（バイト数）で取得します。列が繰り返し列又は配列列以外の場合、0 を返します。

形式

```
Object.ColumnNameLength (Index)
```

パラメタ

- Object*
ColumnList オブジェクトへの参照を表すオブジェクト式です。
- Index*
目的の列の列番号を、数値で設定します。番号は、1 ~ *ColumnList*.Count プロパティの範囲内で設定します。

使い方

- ColumnList*.ColumnName プロパティで列名を取得した場合、繰り返し列又は配列列は、"列名(n)" (n は要素番号) の形式で返されます。ColumnNameLength プロパティを使うと、列名部分の長さ（括弧と要素番号を除いた部分の長さ）を取得できます。
- 列番号は、*ColumnList*.Get メソッドを実行した後、*ColumnList*.Count プロパティで得られた列数の範囲内で設定してください。

ColumnScale プロパティ

機能

列番号で設定した列の小数点以下のけた数を、数値で取得します。

形式

```
Object.ColumnScale (Index)
```

パラメタ

- *Object*

ColumnList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1~*ColumnList.Count* プロパティの範囲内で設定します。

使い方

- 列番号は、*ColumnList.Get* メソッドを実行した後、*ColumnList.Count* プロパティで得られた列数の範囲内で設定してください。

ColumnType プロパティ

機能

列番号で設定した列の属性を、数値で取得します。

形式

<i>Object.ColumnType (Index)</i>

パラメタ

- *Object*

ColumnList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1~*ColumnList.Count* プロパティの範囲内で設定します。

使い方

- 属性については、「付録 E 列の属性」を参照してください。
- 列番号は、*ColumnList.Get* メソッドを実行した後、*ColumnList.Count* プロパティで得られた列数の範囲内で設定してください。

Count プロパティ

機能

取得した列情報の個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

ColumnList オブジェクトへの参照を表すオブジェクト式です。

使い方

- *ColumnList.Get* メソッドを実行する前は、0 を返します。

CountLimit プロパティ

機能

取得する列情報の最大数を、数値で設定します。このプロパティは、値の取得もできます。

形式

```
Object.CountLimit [= max]
```

パラメタ

- *Object*

ColumnList オブジェクトへの参照を表すオブジェクト式です。

- *max*

取得する列数を、次に示す数値で設定します。

数値	内容
0	すべての列を取得します。
1~8192	設定した数だけ、列の情報を取得します。

使い方

- CountLimit プロパティの値は、*ColumnList.Get* メソッドを実行するときに使われます。*ColumnList.Get* メソッドを実行した後で CountLimit プロパティの値を変更しても、*ColumnList.Get* メソッドの結果を変えることはできません。

SelectDict プロパティ

機能

接続先をキーに、使用する見出し辞書を限定します。このプロパティは、値の取得もできます。

形式

```
Object. SelectDict = DataSource
```

パラメタ

- *Object*
ColumnList オブジェクトへの参照を表すオブジェクト式です。
- *DataSource*
接続先を 0~32 バイトの文字列で設定します。0 バイトの文字列も設定できます。

使い方

- 接続先には、*TableList.TableDataSource* プロパティと同じ文字列を設定してください。
- 接続先が見出し辞書にない場合、*TableList.Count* プロパティが 0 となります。

TableName プロパティ

機能

列情報を取得したい表名を、文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object. TableName [= text]
```

パラメタ

- *Object*
ColumnList オブジェクトへの参照を表すオブジェクト式です。
- *text*
表名を、1~256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。表名は、次に示す書式で設定してください。
 - "所有者"."表名"
 - "表名"

使い方

- TableName プロパティの値は、*ColumnList.Get* メソッドを実行するときに使われます。
ColumnList.Get メソッドを実行した後で TableName プロパティの値を変更しても、*ColumnList.Get* メソッドの結果を変えることはできません。

Get メソッド

機能

ColumnList.TableName プロパティで設定した表に含まれる列情報の取得を要求します。

形式

```
Object.Get
```

パラメタ

- *Object*
ColumnList オブジェクトへの参照を表すオブジェクト式です。

使い方

- このメソッドを実行すると、前回 *ColumnList.Get* メソッドで取得した列の情報は上書きされます。

4.7 QueryDefine オブジェクトの詳細

QueryDefine オブジェクトは、抽出処理を実行するオブジェクトです。抽出処理の結果として出力する抽出データは、Result オブジェクトで取得します。

QueryDefine オブジェクトのプロパティとメソッドの一覧を次に示します。

●QueryDefine オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Exclusive	○	○	抽出処理を実行するときの排他モードを、文字列で設定します。

●QueryDefine オブジェクトのメソッドの一覧

メソッド	説明
Execute	抽出処理を実行します。
QueryCount	抽出処理の結果として出力される行数を、数値で取得します。
RemoveAll	すべての Table, Column, Condition オブジェクトをクリアします。

Exclusive プロパティ

機能

抽出する表をほかのユーザが使っているときの動作を、文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object.Exclusive [= String]
```

パラメタ

- *Object*

QueryDefine オブジェクトへの参照を表すオブジェクト式です。

- *String*

排他オプションを、次に示す 0~64 バイトの文字列で設定します。

文字列	内容
長さ 0 バイトの文字列	"WITH SHARE LOCK"の説明を参照してください。
"WITH SHARE LOCK"	一度検索した内容を、トランザクション終了まで、ほかのユーザからの参照は許しますが、更新は許しません。

文字列	内容
"WITH EXCLUSIVE LOCK"	一度検索した内容を、トランザクション終了まで、ほかのユーザからの参照も更新も許しません。
"WITHOUT LOCK WAIT"	一度検索した内容を、トランザクション終了まで、排他制御しません。
"WITHOUT LOCK NOWAIT"	ほかのユーザが更新中のデータでも、更新の完了を待たないで参照して、一度検索した内容をトランザクション終了まで排他制御しません。
"WITH ROLLBACK"	検索の対象になる表をほかのユーザが使っているときは、トランザクションを取り消して無効にします。

排他オプションの文字列については、各データベースのマニュアルを参照してください。

使い方

- Exclusive プロパティの値は、`QueryDefine.Execute` メソッドを実行するときに使われます。`QueryDefine.Execute` メソッドを実行した後で Exclusive プロパティの値を変更しても、`QueryDefine.Execute` メソッドの結果を変えることはできません。

Execute メソッド

機能

データベースに抽出条件を渡して、検索を実行します。

形式

```
Object.Execute
```

パラメタ

- *Object*
QueryDefine オブジェクトへの参照を表すオブジェクト式です。

使い方

- 検索したデータは、Result オブジェクトを使って取得します。
- 結果取得を完了させないで再び `QueryDefine.Execute` メソッドを実行すると、エラーになります。

QueryCount メソッド

機能

抽出条件に一致するデータの件数を、数値で取得します。

形式

```
Object. QueryCount
```

パラメタ

- *Object*

QueryDefine オブジェクトへの参照を表すオブジェクト式です。

使い方

- QueryCountLimit プロパティの設定内容には依存しません。

RemoveAll メソッド

機能

すべての Table, Column, Condition オブジェクトをクリアします。

形式

```
Object. RemoveAll
```

パラメタ

- *Object*

QueryDefine オブジェクトへの参照を表すオブジェクト式です。

4.8 Tables コレクションオブジェクトの詳細

Tables コレクションオブジェクトは、Table オブジェクトをリストオブジェクトとして管理します。

Tables コレクションオブジェクトのプロパティとメソッドの一覧を次に示します。

●Tables コレクションオブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	抽出表の個数を数値で取得します。

●Tables コレクションオブジェクトのメソッドの一覧

メソッド	説明
Add	抽出表を文字列で追加します。
Item	Table オブジェクトのオブジェクト式を取得します。
RemoveAll	すべての Table オブジェクトをクリアします。

Count プロパティ

機能

抽出条件として設定されている表の個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

Tables コレクションオブジェクトへの参照を表すオブジェクト式です。

Add メソッド

機能

抽出条件にする表を、文字列で追加します。

形式

```
Object.Add (String)
```

パラメタ

- *Object*

Tables コレクションオブジェクトへの参照を表すオブジェクト式です。

- *String*

抽出条件にする表の表名を、1~256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。表名は、次に示すどちらかの書式で設定してください。

- "所有者"."表名"
- "表名"

Item メソッド

機能

Table オブジェクトのオブジェクト式を取得します。

形式

```
Object.Item (ObjectNumber)
```

パラメタ

- *Object*

Tables コレクションオブジェクトへの参照を表すオブジェクト式です。

- *ObjectNumber*

目的の抽出表番号 (Table オブジェクトの番号) を、数値で設定します。番号は、1~*Tables.Count* プロパティの範囲内で設定します。

RemoveAll メソッド

機能

すべての Table オブジェクトをクリアします。

形式

```
Object.RemoveAll
```

パラメタ

- *Object*

Tables コレクションオブジェクトへの参照を表すオブジェクト式です。

4.9 Table オブジェクトの詳細

Table オブジェクトは、抽出表を管理するオブジェクトです。

Table オブジェクトのプロパティの一覧を次に示します。Table オブジェクトのメソッドはありません。

●Table オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
TableName	○	○	抽出表として設定されている文字列を設定します。

TableName プロパティ

機能

抽出条件にする表名を、文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object.TableName [= String]
```

パラメタ

- *Object*

Table オブジェクトへの参照を表すオブジェクト式です。

- *String*

抽出する表の表名を、1～256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。表名は、次に示すどちらかの書式で設定してください。

- "所有者"."表名"
- "表名"

4.10 Columns コレクションオブジェクトの詳細

Columns コレクションオブジェクトは、Column オブジェクトをリストオブジェクトとして管理します。

Columns コレクションオブジェクトのプロパティとメソッドの一覧を次に示します。

●Columns コレクションオブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	抽出列の個数を数値で返します。

●Columns コレクションオブジェクトのメソッドの一覧

メソッド	説明
Add	抽出列を文字列で追加します。
Item	Column オブジェクトのオブジェクト式を取得します。
RemoveAll	すべての Column オブジェクトをクリアします。

Count プロパティ

機能

抽出列の個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

Columns コレクションオブジェクトへの参照を表すオブジェクト式です。

Add メソッド

機能

抽出列の列名を、文字列で追加します。

形式

```
Object.Add (String)
```

パラメタ

- *Object*

Columns コレクションオブジェクトへの参照を表すオブジェクト式です。

- *String*

抽出列を、1~256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。

使い方

- 複数の表を使うときは、"所有者.表識別子.列名" の形式で設定してください。

Item メソッド

機能

Column オブジェクトのオブジェクト式を取得します。

形式

```
Object.Item (ObjectNumber)
```

パラメタ

- *Object*

Columns コレクションオブジェクトへの参照を表すオブジェクト式です。

- *ObjectNumber*

目的の抽出列番号 (Column オブジェクトの番号) を、数値で設定します。番号は、1~*Columns.Count* プロパティの範囲内で設定します。

RemoveAll メソッド

機能

すべての Column オブジェクトをクリアします。

形式

```
Object.RemoveAll
```

パラメタ

- *Object*

Columns コレクションオブジェクトへの参照を表すオブジェクト式です。

4.11 Column オブジェクトの詳細

Column オブジェクトは、抽出列を管理するオブジェクトです。

Column オブジェクトのプロパティの一覧を次に示します。Column オブジェクトのメソッドはありません。

●Column オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
ColumnName	○	○	抽出列の列名を設定します。
FunctionMode	○	○	抽出列に集合関数を設定します。
SortMode	○	○	抽出列にソートを設定します。
SortPriority	○	○	ソートを設定した列の優先順位を設定します。

ColumnName プロパティ

機能

抽出列とする列名を、文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object.ColumnName [= String]
```

パラメタ

- *Object*
Column オブジェクトへの参照を表すオブジェクト式です。
- *String*
抽出列を、1~256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。

使い方

- 複数の表を使うときは、"所有者.表識別子.列名" の形式で設定してください。

FunctionMode プロパティ

機能

抽出列に**集合関数**を設定します。このプロパティは、値の取得もできます。

形式

```
Object. FunctionMode [= mode]
```

パラメタ

- *Object*
Column オブジェクトへの参照を表すオブジェクト式です。
- *mode*
集合関数を、次に示す定数又は数値で設定します。

定数	数値	内容
<code>dbpNormal</code> , <code>dbpNoFunction</code>	0	集合関数なし
<code>dbpGroup</code>	1	重複排除
<code>dbpMaximum</code>	2	最大
<code>dbpMinimum</code>	3	最小
<code>dbpCount</code>	4	件数
<code>dbpAverage</code>	5	平均
<code>dbpSum</code>	6	合計

使い方

- 抽出列のどれかに集合関数を付けると、集合関数を設定していない抽出列に「重複排除」が仮定されて、抽出処理が実行されます。ただし、集合関数を設定していない抽出列の集合関数の値は変更されません。

SortMode プロパティ

機能

抽出列をソートするかどうかを設定します。このプロパティは、値の取得もできます。

形式

```
Object. SortMode [= mode]
```

パラメタ

- *Object*
Column オブジェクトへの参照を表すオブジェクト式です。
- *mode*

ソートするかどうかを、次に示す定数又は数値で設定します。

定数	数値	内容
<code>dbpNormal</code> , <code>dbpNoSort</code>	0	データをソートしません。
<code>dbpAscend</code>	1	データを昇順にソートします。
<code>dbpDescend</code>	2	データを降順にソートします。

使い方

- 複数の列にソートを設定したときは、優先するソートを決める必要があります。どの列のソートが優先されるかは、SortMode プロパティの実行順に従います。つまり、最初に設定されたソートが最優先になります。二つめ以降を設定するときには、現在設定されているソートの中で最も低い優先順位の次の順位が割り当てられます。
- 優先順位を参照又は変更したいときは、SortPriority プロパティを使ってください。

SortPriority プロパティ

機能

ソートが設定されている抽出列の優先順位を、数値で設定します。このプロパティは、値の取得もできます。

形式

```
Object.SortPriority [= number]
```

パラメタ

- *Object*
Column オブジェクトへの参照を表すオブジェクト式です。
- *number*
ソートの優先順位を、1～255 の値で設定します。

使い方

- SortPriority プロパティが設定できる列は、SortMode プロパティでソートを設定した列だけです。ソートが設定されていない列には設定できません。
- ソート優先順位の値が小さいほど、優先順位が高くなります。
- 優先順位の値を重複させると、同一番号内での順位は保証されません。
- 優先順位は、連続している値を設定する必要はありません。

4.12 Conditions コレクションオブジェクトの詳細

Conditions コレクションオブジェクトは、Condition オブジェクトをリストオブジェクトとして管理します。

Conditions コレクションオブジェクトのプロパティとメソッドの一覧を次に示します。

●Conditions コレクションオブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	抽出条件式の個数を取得します。
Relation	○	○	条件関連を文字列で設定します。

●Conditions コレクションオブジェクトのメソッドの一覧

メソッド	説明
Add	抽出条件式を文字列で追加します。
Item	Condition オブジェクトのオブジェクト式を取得します。
RemoveAll	すべての Condition オブジェクトをクリアします。

Count プロパティ

機能

抽出条件式の個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

Conditions コレクションオブジェクトへの参照を表すオブジェクト式です。

Relation プロパティ

機能

抽出条件式を関連させる条件関連を、文字列で設定します。このプロパティは、値の取得もできます。

形式

Object. Relation [= *string*]

パラメタ

- *Object*

Conditions コレクションオブジェクトへの参照を表すオブジェクト式です。

- *string*

抽出条件式を関連させる条件を、0~1,024 バイトの文字列で設定します。値をクリアするときは、長さ 0 の文字列を設定します。

使い方

- 値をクリアすると、すべての抽出条件式は AND で関連付けられます。
- 条件はすべて半角文字で設定して、AND, OR, 及び NOT の前後にはスペースを入力してください。

AND, and : 論理積を示します。

「1 AND 2」は、条件式 1 と条件式 2 の両方を満たすデータの抽出を設定します。

OR, or : 論理和を示します。

「1 OR 2」は、条件式 1 か条件式 2 のどちらかを満たすデータの抽出を設定します。

NOT, not : 否定を示します。

「NOT 1」は、条件式 1 を満たさないデータの抽出を設定します。

() : 括弧内の条件を優先します。

「1 AND (2 OR 3)」は、条件式 2 か条件式 3 のどちらかを満たして、かつ条件式 1 を満たすデータの抽出を設定します。

Add メソッド

機能

抽出データを絞り込むための条件式を、文字列で追加します。

形式

Object. Add (*String*)

パラメタ

- *Object*

Conditions コレクションオブジェクトへの参照を表すオブジェクト式です。

- *String*

抽出条件式を、1~30,000 バイトの文字列で設定します。長さ 0 の文字列は設定できません。

使い方

- 代表的な抽出条件式を次に示します。

列名 C1 のデータが NULL 値の行を抽出する (NULL 述語)

```
"C1" IS NULL
```

列名 C2 のデータが 10, 20, 30 の行を抽出する (IN 述語)

```
"C2" IN (10, 20, 30)
```

列名 C3 のデータが ABC から始まる行を抽出する (LIKE 述語)

```
"C3" LIKE 'ABC%'
```

列名 C4 のデータが 10 から 50 の範囲内にある行を抽出する (BETWEEN 述語)

```
"C4" BETWEEN 10 AND 50
```

列名 C5 のデータが 100 の行を抽出する (比較条件)

```
"C5" = 100
```

Item メソッド

機能

Condition オブジェクトのオブジェクト式を取得します。

形式

```
Object.Item (ObjectNumber)
```

パラメタ

- *Object*

Conditions コレクションオブジェクトへの参照を表すオブジェクト式です。

- *ObjectNumber*

目的の抽出条件式番号 (Condition オブジェクトの番号) を、数値で設定します。番号は、1~*Conditions.Count* プロパティの範囲内で設定します。

RemoveAll メソッド

機能

すべての Condition オブジェクトをクリアします。

形式

```
Object.RemoveAll
```

パラメタ

- *Object*

Conditions コレクションオブジェクトへの参照を表すオブジェクト式です。

4.13 Condition オブジェクトの詳細

Condition オブジェクトは、Select 文を生成するための情報になる条件情報を管理します。条件一つにつき、一つの Condition オブジェクトが対応します。

Condition オブジェクトのプロパティの一覧を次に示します。Condition オブジェクトのメソッドはありません。

●Condition オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Text	<input type="radio"/>	<input type="radio"/>	抽出条件式を設定します。

Text プロパティ

機能

抽出データを絞り込むための条件式を、文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object.Text [= String]
```

パラメタ

- *Object*
Condition オブジェクトへの参照を表すオブジェクト式です。
- *String*
抽出の条件式を、1~30,000 バイトの文字列で設定します。長さ 0 の文字列は設定できません。

4.14 UpdateDefine オブジェクトの詳細

UpdateDefine オブジェクトは、更新処理を実行するオブジェクトです。

UpdateDefine オブジェクトのプロパティとメソッドの一覧を次に示します。

●UpdateDefine オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
ErrorInformation	—	○	更新処理でエラーになったオブジェクトの番号を取得します。
Exclusive	○	○	抽出処理を実行するときの排他モードを、文字列で設定します。
ExecutedRowCount	—	○	更新処理の結果、サーバプログラムで更新した行数を取得します。
NCharPrefixMode	○	○	各国文字列のデータにプレフィックスとして"N"を付けるかどうかを設定します。
TableName	○	○	更新表を設定します。
UpdateMode	○	○	更新実行モードを設定します。

●UpdateDefine オブジェクトのメソッドの一覧

メソッド	説明
Execute	更新処理を実行します。
RemoveAll	すべての Update, Insert, Delete オブジェクトをクリアします。

ErrorInformation プロパティ

機能

更新処理でエラーになった Update, Insert, Delete オブジェクトの番号を、数値で取得します。

形式

```
Object. ErrorInformation
```

パラメタ

- *Object*

UpdateDefine オブジェクトへの参照を表すオブジェクト式です。

戻り値

値	内容
0	更新する表又は列として、存在しない表名又は列名を設定しています。
1 以上	エラーになった Update, Insert, Delete オブジェクトの番号を示します。

使い方

- `UpdateDefine.Execute` メソッドを実行する前は、0 を返します。

Exclusive プロパティ

機能

更新する表をほかのユーザが使っているときの動作を、文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object.Exclusive [= String]
```

パラメタ

- *Object*
UpdateDefine オブジェクトへの参照を表すオブジェクト式です。
- *String*
排他オプションを、次に示す 0~64 バイトの文字列で設定します。

文字列	内容
長さ 0 バイトの文字列	更新対象の表をほかのユーザが使っているときは、使っているユーザのトランザクション終了を待ちます。
"WITH ROLLBACK"	更新対象の表をほかのユーザが使っているときは、トランザクションを取り消して無効にします。

排他オプションの文字列については、各データベースのマニュアルを参照してください。

使い方

- Exclusive プロパティの値は、`UpdateDefine.Execute` メソッドを実行するときに使われます。`UpdateDefine.Execute` メソッドを実行した後で Exclusive プロパティの値を変更しても、`UpdateDefine.Execute` メソッドの結果を変えることはできません。

ExecutedRowCount プロパティ

機能

`UpdateDefine.Execute` メソッドを実行 (Update, Insert, Delete) した結果, サーバプログラムが更新した行数を, 数値で取得します。

形式

```
Object. ExecutedRowCount
```

パラメタ

- *Object*

UpdateDefine オブジェクトへの参照を表すオブジェクト式です。

使い方

- `UpdateDefine.Execute` メソッドを実行する前は, 0 を返します。

NCharPrefixMode プロパティ

機能

`dbpTypeNCHAR`, `dbpTypeNVARCHAR`, `dbpTypeLONGNVARCHAR` の各国文字列のデータに, プレフィックスとして "N" を付けるかどうかを設定します。このプロパティは, 値の取得もできます。

形式

```
Object. NCharPrefixMode [= Mode]
```

パラメタ

- *Object*

UpdateDefine オブジェクトへの参照を表すオブジェクト式です。

- *Mode*

プレフィックスを付けるかどうかを, 次に示す定数又は数値で設定します。

定数	数値	内容
<code>dbpNCHARPrefix</code>	0	プレフィックス "N" を付けます。
<code>dbpCHARPrefix</code>	1	プレフィックス "N" を付けません。

使い方

- NCharPrefixMode プロパティの値は、*UpdateDefine.Execute* メソッドを実行するときに使われます。*UpdateDefine.Execute* メソッドを実行した後で NCharPrefixMode プロパティの値を変更しても、*UpdateDefine.Execute* メソッドの処理モードを変えることはできません。

TableName プロパティ

機能

更新する表名を、文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object.TableName [= string]
```

パラメタ

- *Object*
UpdateDefine オブジェクトへの参照を表すオブジェクト式です。
- *string*
更新する表名を、1~256 バイトの文字列で設定します。長さ 0 の文字列は、設定できません。

UpdateMode プロパティ

機能

更新処理を実行するときどのような SQL 文で更新するかを設定します。このプロパティは、値の取得もできます。

形式

```
Object.UpdateMode [= mode]
```

パラメタ

- *Object*
UpdateDefine オブジェクトへの参照を表すオブジェクト式です。
- *mode*
更新を実行するモードを、次に示す定数又は数値で設定します。

定数	数値	内容
<code>dbpNormal</code>	0	変数 (?パラメタ) を使いません。
<code>dbpVariable</code>	1	変数 (?パラメタ) を使います。

使い方

- `dbpNormal` を設定すると、Update/Insert/Delete の各オブジェクト単位に SQL 文を作成して更新処理を実行します。
- `dbpVariable` を設定すると、Update/Insert/Delete の各オブジェクトの値の部分を変数にした SQL 文を作成して更新処理を実行します。`dbpVariable` を使うときは、更新する列の数や、列の設定順が一定でないと、エラーになります。
- 次に示す条件で更新するときは、更新実行モードを `dbpVariable` にしてください。
 - `AddUpdateValue` メソッド、`AddInsertValue` メソッドのデータにファイル名を指定したとき
 - `AddUpdateValueB` メソッド、`AddInsertValueB` メソッドを使ったとき

Execute メソッド

機能

データベースに対して、データを更新します。

形式

Object.Execute (*Type*)

パラメタ

- *Object*
UpdateDefine オブジェクトへの参照を表すオブジェクト式です。
- *Type*
更新種別を、次に示す定数又は数値で設定します。

定数	数値	内容
<code>dbpUpdate</code>	0	変更処理を示します。
<code>dbpInsert</code>	1	追加処理を示します。
<code>dbpDelete</code>	2	削除処理を示します。

使い方

- `UpdateDefine.Execute` メソッドを実行するためには、次に示す条件があります。

変更処理の場合

一つ以上の Update オブジェクトが設定されていること

追加処理の場合

一つ以上の Insert オブジェクトが設定されていること

削除処理の場合

一つ以上の Delete オブジェクトが設定されていること

- トランザクション終了モードに dbpManual を設定しているときは、更新処理が終了した時点で必ず Commit メソッド又は Rollback メソッドを実行してください。更新処理がエラーになったときも、メソッドを実行する必要があります。Commit メソッド又は Rollback メソッドを実行し忘れると、データベースのデータを壊してしまうおそれがあります。

RemoveAll メソッド

機能

Update, Insert, Delete オブジェクトをすべてクリアします。

形式

```
Object.RemoveAll
```

パラメタ

- *Object*
UpdateDefine オブジェクトへの参照を表すオブジェクト式です。

4.15 Updates コレクションオブジェクトの詳細

Updates コレクションオブジェクトは、Update オブジェクトをリストオブジェクトとして管理します。

Updates コレクションオブジェクトのプロパティとメソッドの一覧を次に示します。

●Updates コレクションオブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	Update オブジェクトの個数を数値で取得します。

●Updates コレクションオブジェクトのメソッドの一覧

メソッド	説明
Add	Update オブジェクトを生成します。
Item	Update オブジェクトのオブジェクト式を取得します。
RemoveAll	すべての Update オブジェクトを削除します。

Count プロパティ

機能

`Updates.Add` メソッドで作成した Update オブジェクトの個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

Updates コレクションオブジェクトへの参照を表すオブジェクト式です。

Add メソッド

機能

Update オブジェクトを生成します。

形式

```
Object.Add
```

パラメタ

- *Object*

Updates コレクションオブジェクトへの参照を表すオブジェクト式です。

Item メソッド

機能

Update オブジェクトのオブジェクト式を取得します。

形式

```
Object.Item (ObjectNumber)
```

パラメタ

- *Object*

Updates コレクションオブジェクトへの参照を表すオブジェクト式です。

- *ObjectNumber*

目的の Update オブジェクトの番号を、数値で設定します。番号は、1~*Updates.Count* プロパティの範囲内で設定します。

RemoveAll メソッド

機能

すべての Update オブジェクトをクリアします。

形式

```
Object.RemoveAll
```

パラメタ

- *Object*

Updates コレクションオブジェクトへの参照を表すオブジェクト式です。

4.16 Update オブジェクトの詳細

Update オブジェクトは、サーバプログラムのデータを変更するための情報を管理します。

Update オブジェクトのプロパティとメソッドの一覧を次に示します。

●Update オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Column	○	○	変更情報の列名を設定します。
KeyColumn	○	○	変更行条件の列名を設定します。
KeyCount	—	○	変更行条件の数を取得します。
KeyValue	○	○	変更行条件のデータを設定します。
UpdateValue	○	○	変更情報のデータを文字列型で設定します。
UpdateValueB	○	○	変更情報のデータをバイト型で設定します。
UpdateValueCount	—	○	変更情報の数を取得します。

●Update オブジェクトのメソッドの一覧

メソッド	説明
AddKey	変更行条件を追加します。
AddUpdateValue	文字列型の変更情報を追加します。
AddUpdateValueB	文字列型の変更情報を追加します。

Column プロパティ

機能

変更情報の番号で設定した情報の列名を、文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object.Column (ValueNumber) [= Column]
```

パラメタ

- *Object*
Update オブジェクトへの参照を表すオブジェクト式です。
- *ValueNumber*

目的の変更情報を、数値で設定します。番号は、1~*Update.UpdateValueCount* プロパティの範囲内で設定します。

- *Column*

列名を、1~256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。列名は、次に示す書式で設定してください。

- "列名" (表名で修飾できません)

KeyColumn プロパティ

機能

変更行条件の番号で設定した条件の列名を、文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object.KeyColumn (KeyNumber) [= Column]
```

パラメタ

- *Object*

Update オブジェクトへの参照を表すオブジェクト式です。

- *KeyNumber*

目的の条件の変更行番号を、数値で設定します。番号は、1~*Update.KeyCount* プロパティの範囲内で設定します。

- *Column*

列名を、1~256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。列名は、次に示す書式で設定してください。

- "列名" (表名で修飾できません)

KeyCount プロパティ

機能

変更行条件の数を、数値で取得します。

形式

```
Object.KeyCount
```

パラメタ

- *Object*

Update オブジェクトへの参照を表すオブジェクト式です。

KeyValue プロパティ

機能

変更行条件の番号で設定した条件のデータを設定します。このプロパティは、値の取得もできます。

形式

```
Object.KeyValue (KeyNumber) [= Data]
```

パラメタ

- *Object*

Update オブジェクトへの参照を表すオブジェクト式です。

- *KeyNumber*

目的の変更行条件の番号を、数値で設定します。番号は、1~*Update.KeyCount* プロパティの範囲内で設定します。

- *Data*

条件のデータを、0~30,000 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが仮定されます。

UpdateValue プロパティ

機能

AddUpdateValue メソッドで追加した変更情報のデータを、文字列型で設定します。このプロパティは、値の取得もできます。

形式

```
Object.UpdateValue (ValueNumber) [= Data]
```

パラメタ

- *Object*

Update オブジェクトへの参照を表すオブジェクト式です。

- *ValueNumber*

目的の変更情報を、数値で設定します。番号は、1~*Update.UpdateValueCount* プロパティの範囲内で設定します。

- *Data*

データを、0~30,000 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが仮定されます。データとして *dbpTypeLONGVARIABLE* 属性のデータを設定するときは、データを格納しているファイル名を設定します。

使い方

- ファイルサイズが 2 ギガバイト以上のファイルは、データとして設定できません。

UpdateValueB プロパティ

機能

AddUpdateValueB メソッドで追加した変更情報のデータを、バイト型で設定します。このプロパティは、値の取得もできます。

形式

Object. UpdateValueB (*ValueNumber*) [= *Data*]

パラメタ

- *Object*

Update オブジェクトへの参照を表すオブジェクト式です。

- *ValueNumber*

目的の変更情報を、数値で設定します。番号は、1~*Update.UpdateValueCount* プロパティの範囲内で設定します。

- *Data*

データを、バイト型データで設定します。

使い方

- 更新データを格納する変数は、バイト型で宣言してください。データを格納する変数がバイト型以外のときは、エラーになります。
- Visual Basic を使うときの例を次に示します (データを *DataBuff* に格納します)。宣言の例 : Dim *DataBuff*() As Byte

UpdateValueCount プロパティ

機能

変更情報の数を、数値で取得します。

形式

```
Object.UpdateValueCount
```

パラメタ

- *Object*

Update オブジェクトへの参照を表すオブジェクト式です。

AddKey メソッド

機能

変更行条件を追加します。列名には、*Value* に設定するデータを格納している列名を設定します。データには、変更する行を特定するデータを設定します。

形式

```
Object.AddKey Column, Value
```

パラメタ

- *Object*

Update オブジェクトへの参照を表すオブジェクト式です。

- *Column*

列名を、1~256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。列名は、次に示す書式で設定してください。

- "列名" (表名で修飾できません)

- *Value*

データを、0~30,000 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが假定されます。

使い方

- 一つの変更行条件で変更する行を特定できないときは、同一の Update オブジェクトに対して複数の変更行条件を設定してください。

AddUpdateValue メソッド

機能

変更情報を追加します。列名には、*Value* に設定するデータを格納する列名を設定します。データには、新しく格納する文字列型のデータを設定します。

形式

```
Object. AddUpdateValue Column, Value
```

パラメタ

- *Object*

Update オブジェクトへの参照を表すオブジェクト式です。

- *Column*

列名を、1~256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。列名は、次に示す書式で設定してください。

- "列名" (表名で修飾できません)

- *Value*

データを、0~30,000 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが仮定されます。データとして dbpTypeLONGVARIABLE 属性のデータを設定するときは、データを格納しているファイル名を設定します。

使い方

- 変更行条件で決めた行に対して、複数の列のデータを変更するときは、同じ Update オブジェクトに対して複数の変更情報を設定してください。
- ファイルサイズが 2 ギガバイト以上のファイルは、データとして設定できません。
- AddUpdateValue メソッドで設定したデータを変更/取得するときは、UpdateValue プロパティを使います。

AddUpdateValueB メソッド

機能

変更情報を追加します。列名には、*Value* に設定するデータを格納する列名を設定します。データには、新しく格納するバイト型のデータを設定します。

形式

Object. AddUpdateValueB *Column*, *Value*

パラメタ

- *Object*

Update オブジェクトへの参照を表すオブジェクト式です。

- *Column*

列名を、1~256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。列名は、次に示す書式で設定してください。

- "列名" (表名で修飾できません)

- *Value*

データを、バイト型データで設定します。

使い方

- 更新データを格納する変数は、バイト型で宣言してください。データを格納する変数がバイト型以外のときは、エラーになります。
- Visual Basic を使うときの例を次に示します (データを DataBuff に格納します)。宣言の例：Dim DataBuff() As Byte
- 変更行条件で決めた行に対して、複数の列のデータを変更するときは、同じ Update オブジェクトに対して複数の変更情報を設定してください。
- AddUpdateValueB メソッドで設定したデータを変更/取得するときは、UpdateValueB プロパティを使います。

4.17 Inserts コレクションオブジェクトの詳細

Inserts コレクションオブジェクトは、Insert オブジェクトをリストオブジェクトとして管理します。

Inserts コレクションオブジェクトのプロパティとメソッドの一覧を次に示します。

●Inserts コレクションオブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	Insert オブジェクトの個数を、数値で取得します。

●Inserts コレクションオブジェクトのメソッドの一覧

メソッド	説明
Add	Insert オブジェクトを追加します。
Item	Insert オブジェクトのオブジェクト式を取得します。
RemoveAll	すべての Insert オブジェクトをクリアします。

Count プロパティ

機能

Inserts.Add メソッドで作成した Insert オブジェクトの個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

Inserts コレクションオブジェクトへの参照を表すオブジェクト式です。

Add メソッド

機能

Insert オブジェクトを追加します。

形式

```
Object.Add
```

パラメタ

- *Object*

Inserts コレクションオブジェクトへの参照を表すオブジェクト式です。

Item メソッド

機能

Insert オブジェクトのオブジェクト式を取得します。

形式

```
Object.Item (ObjectNumber)
```

パラメタ

- *Object*

Inserts コレクションオブジェクトへの参照を表すオブジェクト式です。

- *ObjectNumber*

目的の Insert オブジェクトの番号を、数値で設定します。番号は、1~*Inserts.Count* プロパティの範囲内で設定します。

RemoveAll メソッド

機能

すべての Insert オブジェクトをクリアします。

形式

```
Object.RemoveAll
```

パラメタ

- *Object*

Inserts コレクションオブジェクトへの参照を表すオブジェクト式です。

4.18 Insert オブジェクトの詳細

Insert オブジェクトは、表に挿入する 1 行分のデータを管理するオブジェクトです。一つの Insert オブジェクトには列単位に複数の挿入値を設定できます。

Insert オブジェクトのプロパティとメソッドの一覧を次に示します。

●Insert オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Column	○	○	追加情報の列名を設定します。
InsertValue	○	○	追加情報のデータを文字列型で設定します。
InsertValueB	○	○	追加情報のデータをバイト型で設定します。
InsertValueCount	—	○	追加情報の個数を数値で取得します。

●Insert オブジェクトのメソッドの一覧

メソッド	説明
AddInsertValue	文字列型の追加情報を追加します。
AddInsertValueB	バイト型の追加情報を追加します。

Column プロパティ

機能

追加情報の番号で設定した情報の列名を、文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object.Column (ValueNumber) [= Column]
```

パラメタ

- *Object*

Insert オブジェクトへの参照を表すオブジェクト式です。

- *ValueNumber*

目的の追加情報の番号を、数値で設定します。番号は、1~*Insert.InsertValueCount* プロパティの範囲内で設定します。

- *Column*

列名を、1~256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。列名は、次に示す書式で設定してください。

- "列名" (表名で修飾できません)

InsertValue プロパティ

機能

AddInsertValue メソッドで追加した追加情報のデータを、文字列型で設定します。このプロパティは、値の取得もできます。

形式

```
Object.InsertValue (ValueNumber) [= Data]
```

パラメタ

- *Object*
Insert オブジェクトへの参照を表すオブジェクト式です。
- *ValueNumber*
目的の追加情報の番号を、数値で設定します。番号は、1~*Insert.InsertValueCount* プロパティの範囲内で設定します。
- *Data*
データを、0~30,000 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが仮定されます。データとして dbpTypeLONGVARBINARY 属性のデータを設定するときは、データを格納しているファイル名を設定します。

使い方

- ファイルサイズが 2 ギガバイト以上のファイルは、データとして設定できません。

InsertValueB プロパティ

機能

AddInsertValueB メソッドで追加した追加情報のデータを、バイト型で設定します。このプロパティは、値の取得もできます。

形式

```
Object.InsertValueB (ValueNumber) [= Data]
```

パラメタ

- *Object*

Insert オブジェクトへの参照を表すオブジェクト式です。

- *ValueNumber*

目的の追加情報の番号を、数値で設定します。番号は、1~*Insert.InsertValueCount* プロパティの範囲内で設定します。

- *Data*

データを、バイト型データで設定します。

使い方

- 更新データを格納する変数は、バイト型で宣言してください。データを格納する変数がバイト型以外のときは、エラーになります。

- Visual Basic を使うときの例を次に示します（データを *DataBuff* に格納します）。

宣言の例：Dim *DataBuff*() As Byte

InsertValueCount プロパティ

機能

追加情報の数を、数値で取得します。

形式

```
Object.InsertValueCount
```

パラメタ

- *Object*

Insert オブジェクトへの参照を表すオブジェクト式です。

AddInsertValue メソッド

機能

追加情報を追加します。列名には、*Value* に設定するデータを格納する列名を設定します。データには、新しく追加する文字列型のデータを設定します。

形式

```
Object. AddInsertValue Column, Value
```

パラメタ

- *Object*

Insert オブジェクトへの参照を表すオブジェクト式です。

- *Column*

列名を、1~256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。列名は、次に示す書式で設定してください。

- "列名" (表名で修飾できません)

- *Value*

データを、0~30,000 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが仮定されます。データとして dbpTypeLONGVARIABLE 属性のデータを設定するときは、データを格納しているファイル名を設定します。

使い方

- 複数の列にデータを追加するときは、同じ Insert オブジェクトに対して複数の追加情報を設定してください。
- ファイルサイズが 2 ギガバイト以上のファイルは、データとして設定できません。
- AddInsertValue メソッドで設定したデータを変更/取得するときは、InsertValue プロパティを使います。

AddInsertValueB メソッド

機能

追加情報を追加します。列名には、*Value* に設定するデータを格納する列名を設定します。データには、新しく追加するバイト型のデータを設定します。

形式

```
Object. AddInsertValueB Column, Value
```

パラメタ

- *Object*

Insert オブジェクトへの参照を表すオブジェクト式です。

- *Column*

列名を、1～256バイトの文字列で設定します。長さ0の文字列は設定できません。列名は、次に示す書式で設定してください。

- "列名" (表名で修飾できません)
- *Value*
データを、バイト型データで設定します。

使い方

- 更新データを格納する変数は、バイト型で宣言してください。データを格納する変数がバイト型以外の場合は、エラーになります。
- Visual Basic を使うときの例を次に示します (データを DataBuff に格納します)。
宣言の例 : `Dim DataBuff() As Byte`
- 複数の列にデータを追加するときは、同じ Insert オブジェクトに対して複数の追加情報を設定してください。
- AddInsertValueB メソッドで設定したデータを変更/取得するときは、InsertValueB プロパティを使います。

4.19 Deletes コレクションオブジェクトの詳細

Deletes コレクションオブジェクトは、Delete オブジェクトをリストオブジェクトとして管理します。

Deletes コレクションオブジェクトのプロパティとメソッドの一覧を次に示します。

●Deletes コレクションオブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	Delete オブジェクトの個数を数値で取得します。

●Deletes コレクションオブジェクトのメソッドの一覧

メソッド	説明
Add	Delete オブジェクトを追加します。
Item	Delete オブジェクトのオブジェクト式を取得します。
RemoveAll	すべての Delete オブジェクトをクリアします。

Count プロパティ

機能

Deletes.Add メソッドで作成した Delete オブジェクトの個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

Deletes コレクションオブジェクトへの参照を表すオブジェクト式です。

Add メソッド

機能

Delete オブジェクトを生成します。

形式

```
Object.Add
```

パラメタ

- *Object*

Deletes コレクションオブジェクトへの参照を表すオブジェクト式です。

Item メソッド

機能

Delete オブジェクトのオブジェクト式を取得します。

形式

```
Object.Item (ObjectNumber)
```

パラメタ

- *Object*

Deletes コレクションオブジェクトへの参照を表すオブジェクト式です。

- *ObjectNumber*

目的の Delete オブジェクトの番号を、数値で設定します。番号は、1~*Deletes.Count* プロパティの範囲内で設定します。

RemoveAll メソッド

機能

すべての Delete オブジェクトをクリアします。

形式

```
Object.RemoveAll
```

パラメタ

- *Object*

Deletes コレクションオブジェクトへの参照を表すオブジェクト式です。

4.20 Delete オブジェクトの詳細

Delete オブジェクトは、Delete 文を生成するための情報になる削除データの設定値情報を管理します。Delete 文 1 文につき、一つの Delete オブジェクトが対応します。

Delete オブジェクトのプロパティとメソッドの一覧を次に示します。

●Delete オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
KeyColumn	○	○	削除行条件の列名を設定します。
KeyCount	—	○	削除行条件の数を取得します。
KeyValue	○	○	削除行条件のデータを設定します。

●Delete オブジェクトのメソッドの一覧

メソッド	説明
AddKey	削除行条件を追加します。

KeyColumn プロパティ

機能

削除行条件の番号で設定した条件の列名を、文字列で設定します。このプロパティは、値の取得もできます。

形式

```
Object.KeyColumn (KeyNumber) [= Column]
```

パラメタ

- *Object*
Delete オブジェクトへの参照を表すオブジェクト式です。
- *KeyNumber*
目的の削除行条件の番号を、数値で設定します。番号は、1~Delete.KeyCount プロパティの範囲内で設定します。
- *Column*
列名を、1~256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。列名は、次に示す書式で設定してください。
 - "列名" (表名で修飾できません)

KeyCount プロパティ

機能

削除行条件の数を、数値で取得します。

形式

```
Object.KeyCount
```

パラメタ

- *Object*
Delete オブジェクトへの参照を表すオブジェクト式です。

KeyValue プロパティ

機能

削除行条件の番号で設定した条件のデータを、文字列で設定します。このプロパティは、値の取得もできません。

形式

```
Object.KeyValue (KeyNumber) [= Data]
```

パラメタ

- *Object*
Delete オブジェクトへの参照を表すオブジェクト式です。
- *KeyNumber*
目的の削除行条件の番号を、数値で設定します。番号は、1~*Delete.KeyCount* プロパティの範囲内で設定します。
- *Data*
データを、0~30,000 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが仮定されます。

AddKey メソッド

機能

削除行条件を追加します。列名には、*Value* に設定するデータを格納している列名を設定します。データには、削除する行を特定するデータを設定します。

形式

```
Object.AddKey Column, Value
```

パラメタ

- *Object*
Delete オブジェクトへの参照を表すオブジェクト式です。
- *Column*
列名を、1~256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。列名は、次に示す書式で設定してください。
 - "列名" (表名で修飾できません)
- *Value*
データを、0~30,000 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが仮定されます。

使い方

- 一つの削除行条件で削除する行を特定できないときは、同一の Delete オブジェクトに対して複数の削除行条件を設定してください。

4.21 Catalog オブジェクトの詳細

Catalog オブジェクトは、手順ファイルを参照し QueryDefine オブジェクトを生成するための情報を管理します。

Catalog オブジェクトのメソッドの一覧を次に示します。Catalog オブジェクトのプロパティはありません。

●Catalog オブジェクトのメソッドの一覧

メソッド	説明
Convert	読み込んだ手順ファイルの情報から QueryDefine オブジェクトを生成します。
FileLoad	手順ファイルの情報を読み込みます。

Convert メソッド

機能

FileLoad メソッドで読み込まれた手順ファイルの情報から、QueryDefine オブジェクトを生成します。

形式

```
Object. Convert
```

パラメタ

- *Object*

Catalog オブジェクトへの参照を表すオブジェクト式です。

使い方

- 可変値を使用している手順ファイルを読み込んだ場合は、Convert メソッドを発行する前に VariableList オブジェクトに実行値を設定してください。実行値が設定されていない場合は、手順ファイルが格納している初期値を使用して QueryDefine オブジェクトを生成します。
- Convert メソッドを発行すると、現在の QueryDefine オブジェクトの内容は削除されます。

FileLoad メソッド

機能

手順ファイルから抽出に関する情報を読み込みます。抽出条件として可変値が使用されている場合は、*VariableList.Count* プロパティに可変値の個数が返されます。手順ファイルの情報から QueryDefine オブジェクトを生成するには、Convert メソッドを使用します。

形式

```
Object. FileLoad (FileName)
```

パラメタ

- *Object*
Catalog オブジェクトへの参照を表すオブジェクト式です。
- *FileName*
ファイル名を、1~256 バイトの文字列で設定します。

使い方

- ファイル名にパスが含まれていない場合は、カレントディレクトリが仮定されます。
- FileLoad メソッドを発行すると、VariableList オブジェクトの各プロパティの内容が更新されます。
- CommuniNet 接続の抽出を登録した手順ファイルを設定した場合、動作は保証しません。
- 次に示す手順ファイルを設定した場合、エラーとなります。
 - DBPARTNER/Client 03-00 以前のバージョンの手順ファイル
 - 表定義を登録した手順ファイル
 - アウタジョインを登録した手順ファイル
 - SQL を登録した手順ファイル
- 保存、予約、絞り込みを登録した手順ファイルからは、表定義に関する情報は読み込みません。
- メール連携を登録した手順ファイルからは、メール操作に関する情報は読み込みません。
- 更新用抽出を登録した手順ファイルは、抽出操作として読み込みます。
- 手順ファイルに登録されている情報でも、次に示す情報は読み込みません。

ホスト名称、転送先種別、転送先ファイル名、逐次検索、件数確認、抽出件数の制限、仮想列の名称、更新キー解除、BINARY 属性のファイル種別

4.22 VariableList オブジェクトの詳細

VariableList オブジェクトは、手順ファイルに格納されている可変値情報を取得したり、取得した情報を管理したりします。それぞれの情報はプロパティによって返されます。

VariableList オブジェクトのプロパティの一覧を次に示します。VariableList オブジェクトのメソッドはありません。

●VariableList オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	手順ファイルに格納されている可変値の個数を数値で取得します。
VariableName	—	○	可変値の名称を文字列で取得します。
VariableValue	○	○	可変値の値とする実行値を設定します。

Count プロパティ

機能

`Catalog.FileLoad` メソッドで読み込んだ手順ファイルに格納されている可変値の個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

VariableList オブジェクトへの参照を表すオブジェクト式です。

使い方

- 手順ファイルに可変値が使用されていない場合、このプロパティの値は 0 になります。

VariableName プロパティ

機能

可変値の名称を文字列で取得します。

形式

```
Object.VariableName (Index)
```

パラメタ

- *Object*

VariableList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の可変値の番号を数値で設定します。番号は、1~*VariableList.Count* プロパティの範囲で設定します。

使い方

- 可変値の名称の先頭に付いている“&”は返りません。

VariableValue プロパティ

機能

可変値の値とする実行値を設定します。このプロパティは、値の取得もできます。

形式

```
Object.VariableValue (Index) [= String]
```

パラメタ

- *Object*

VariableList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の可変値の番号を数値で設定します。番号は、1~*VariableList.Count* プロパティの範囲で設定します。

- *String*

実行値を、1~30,000 バイトの文字列で設定します。

使い方

- *Catalog.FileLoad* メソッド発行直後は、手順ファイルに格納されている初期値が実行値として設定されます。
- 実行値に長さ 0 の文字列、又は 30,001 バイト以上の文字列が設定された場合、エラーとなります。

4.23 Procedure オブジェクトの詳細

Procedure オブジェクトは、ストアドプロシジャを実行する情報を管理します。

Procedure オブジェクトのプロパティとメソッドの一覧を次に示します。

●Procedure オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
ListPattern	○	○	ストアドプロシジャの一覧を絞り込む条件を文字列で指定します。

●Procedure オブジェクトのメソッドの一覧

メソッド	説明
Execute	ストアドプロシジャを実行します。
GetList	ストアドプロシジャの一覧を取得し、ProcedureList オブジェクトを生成します。
GetParam	ストアドプロシジャを構成するパラメタの一覧を取得し、ParamList オブジェクトを生成します。

ListPattern プロパティ

機能

ストアドプロシジャの一覧を取得するための絞り込み条件を、文字列で設定します。このプロパティは値の取得もできます。

形式

```
Object.ListPattern[= text]
```

パラメタ

- *Object*

Procedure オブジェクトへの参照を表すオブジェクト式です。

- *text*

絞り込み条件を 0~128 バイトまでの文字列で設定します。文字列には、ワイルドカード (_ , %) 文字が使えます。値をクリアする場合は、長さ 0 の文字列を設定します。次の書式で設定してください。

- 所有者.ストアドプロシジャ名
- ストアドプロシジャ名

使い方

- 絞り込み条件に指定する所有者やストアプロシジャ名の最大長は、使用するデータベースに依存します。制限を超えた絞り込み条件を指定した場合は、*Procedure.GetList* メソッドがエラーになるか、絞り込みが正しく行われません。
- 値がクリアされている場合は、`%.%` が仮定されます。
- `_` (アンダースコア文字) は、ある 1 文字を表します。
- `%` (パーセント文字) は、0 個以上のある文字のつながりを表します。
- 使用できる絞り込み条件を次に示します。

前方一致

(例：`ABC%`) ストアドプロシジャ名の先頭が'`ABC`'であり、以降の文字は任意です。

後方一致

(例：`%ABC`) ストアドプロシジャ名の最後が'`ABC`'であり、以前の文字は任意です。

任意一致

(例：`%ABC%`) ストアドプロシジャ名中の任意の部分が'`ABC`'です。

完全一致

(例：`ABC`) ストアドプロシジャ名が'`ABC`'と同じです。

部分一致

(例：`_ABC_`) 5 文字のストアプロシジャ名のうち、2~4 文字目が'`ABC`'です。

(例：`A_B%C`) ストアドプロシジャ名の先頭が `A`、3 文字目が'`B`'、最後が'`C`'です。

- `_` や `%` を条件に含めたい場合は、それらの文字の前に'`¥`'を付けます。
`ABC¥%%` 文字列の先頭が'`ABC%`'であり、以降の文字は任意です。
- `ListPattern` プロパティの値は、*Procedure.GetList* メソッド発行時に使用されます。
Procedure.GetList メソッド発行後に `ListPattern` プロパティの値を変更しても、*Procedure.GetList* メソッドの結果を変えることはできません。

Execute メソッド

機能

Procedure.GetParam メソッドで指定したストアプロシジャを実行します。

形式

```
Object.Execute
```

パラメタ

- *Object*

Procedure オブジェクトへの参照を表すオブジェクト式です。

使い方

- Execute メソッドを実行する前に、*Procedure.GetParam* メソッドを発行しておく必要があります。パラメタがないストアードプロシジャを実行する場合でも、*Procedure.GetParam* メソッドは発行してください。

GetList メソッド

機能

ストアードプロシジャの一覧を取得し、*ProcedureList* オブジェクトを生成します。

形式

```
Object.GetList
```

パラメタ

- *Object*

Procedure オブジェクトへの参照を表すオブジェクト式です。

GetParam メソッド

機能

ストアードプロシジャを構成するパラメタの一覧を取得し、*ParamList* オブジェクトを生成します。

形式

```
Object.GetParam(ProcedureName)
```

パラメタ

- *Object*

Procedure オブジェクトへの参照を表すオブジェクト式です。

- *ProcedureName*

ストアードプロシジャ名を、1~256 バイトの文字列で指定します。

4.24 ParamList オブジェクトの詳細

ParamList オブジェクトは、指定されたストアプロシジャ名に定義されているパラメータを取得したり、取得した結果を管理したりします。それぞれの列情報は、プロパティによって返されます。

ParamList オブジェクトのプロパティの一覧を次に示します。ParamList オブジェクトのメソッドはありません。

●ParamList オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	取得したパラメータの数を数値で取得します。
ParamCodeType	○	○	出力パラメータの値を文字列で取得するかバイト型データで取得するかを数値で設定します。
ParamName	—	○	パラメータの名称を文字列で取得します。
ParamType	○	○	パラメータの種別を数値で取得します。
ParamComment	—	○	パラメータのコメントを文字列で取得します。
ParamDataType	—	○	パラメータの属性を数値で取得します。
ParamLength	—	○	パラメータの定義長を数値で取得します。
ParamNullable	—	○	入力パラメータに NULL が設定できるかどうかを数値で取得します。
ParamPrecision	○	○	パラメータの精度を数値で取得します。
ParamRadix	—	○	パラメータの基数を数値で取得します。
ParamScale	○	○	パラメータの値の小数点以下けた数を数値で取得します。
ParamTypeName	—	○	パラメータのデータベース依存の属性を文字列で取得します。
ParamValue	○	○	入力パラメータに文字列で値を設定します。又は出力パラメータから文字列で値を取得します。
ParamValueB	○	○	入力パラメータにバイト型データで値を設定します。又は出力パラメータからバイト型データで値を取得します。
ParamValueMaxLength	○	○	出力パラメータから値を取得するときのバイト数を数値で設定します。

Count プロパティ

機能

取得したパラメータの数を数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

ParamList オブジェクトへの参照を表すオブジェクト式です。

ParamCodeType プロパティ

機能

出力パラメタの値を、文字列(*ParamList.ParamValue* プロパティ)で取得するかバイト型データ(*ParamList.ParamValueB* プロパティ)で取得するかを、数値で設定します。

形式

```
Object.ParamCodeType(Index)[= Mode ]
```

パラメタ

- *Object*

ParamList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的のパラメタを数値で設定します。番号は、1 ~ *ParamList.Count* プロパティの範囲内で指定します。

- *Mode*

モードを、定数、又は数値で設定します。

定数	値	内容
dbpBINARY	-2	<i>ParamList.ParamValueB</i> プロパティを使用してデータを設定又は取得します。
dbpCHAR	1	<i>ParamList.ParamValue</i> プロパティを使用してデータを設定又は取得します。

ParamName プロパティ

機能

パラメタの名称を文字列で取得します。

形式

```
Object.ParamName(Index)
```

パラメタ

- *Object*
ParamList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的のパラメタを数値で設定します。番号は、1~*ParamList.Count* プロパティの範囲内で指定します。

ParamType プロパティ

機能

パラメタの種別を取得します。取得した値と異なる値を設定することもできます。

形式

```
Object.ParamType(Index)[= Type ]
```

パラメタ

- *Object*
ParamList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的のパラメタを数値で設定します。番号は、1~*ParamList.Count* プロパティの範囲内で指定します。
- *Type*
返されるパラメタ種別には次の種類があります。

定数	値	内容
dbpParamTypeUnknown	0	未知の型のパラメタ
dbpParamInput	1	入力パラメタ
dbpParamInputOutput	2	入出力パラメタ
dbpResultCol	3	結果セットの列
dbpParamOutput	4	出力パラメタ
dbpReturnValue	5	ストアドプロシジャの戻り値

使い方

- ストアドプロシジャに、種別が `dbpResultCol` や `dbpParamTypeUnknown` のパラメタが定義されていると、`Procedure.Execute` メソッド発行時にエラーとなります。
- 取得した種別と異なる種別を設定する必要がある場合は、`dbpResultCol` や `dbpParamTypeUnknown` 以外の種別を設定してください。`dbpResultCol` や `dbpParamTypeUnknown` を設定すると、`Procedure.Execute` メソッド発行時にエラーとなります。

ParamComment プロパティ

機能

パラメタのコメントを文字列で取得します。

形式

```
Object.ParamComment(Index)
```

パラメタ

- *Object*
ParamList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的のパラメタを数値で設定します。番号は、1~`ParamList.Count` プロパティの範囲内で指定します。

ParamDataType プロパティ

機能

パラメタの属性を数値で取得します。

形式

```
Object.ParamDataType(Index)
```

パラメタ

- *Object*
ParamList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的のパラメタを数値で設定します。番号は、1~`ParamList.Count` プロパティの範囲内で指定します。

使い方

- 属性の詳細については、「付録 E.1 ODBC 接続のときの列の属性」を参照してください。

ParamLength プロパティ

機能

パラメタの定義長を数値で取得します。

形式

```
Object.ParamLength(Index)
```

パラメタ

- *Object*
ParamList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的のパラメタを数値で設定します。番号は、1~*ParamList.Count* プロパティの範囲内で指定します。

ParamNullable プロパティ

機能

入力パラメタに NULL が設定できるかどうかを数値で取得します。

形式

```
Object.ParamNullable(Index)
```

パラメタ

- *Object*
ParamList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的のパラメタを数値で設定します。番号は、1~*ParamList.Count* プロパティの範囲内で指定します。

戻り値

定数	値	内容
dbpNoNulls	0	NULL 値を設定できない。
dbpNullable	1	NULL 値を設定できる。
dbpNullableUnknown	2	NULL 値を設定できるかどうか不明。

ParamPrecision プロパティ

機能

パラメタの精度を数値で取得します。

形式

```
Object.ParamPrecision(Index)[= Length]
```

パラメタ

- *Object*
ParamList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的のパラメタを数値で設定します。番号は、1~*ParamList.Count* プロパティの範囲内で指定します。
- *Length*
パラメタの精度を、1~2147483647 の数値で設定します。

ParamRadix プロパティ

機能

パラメタの基数を数値で取得します。

形式

```
Object.ParamRadix(Index)
```

パラメタ

- *Object*
ParamList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的のパラメタを数値で設定します。番号は、1~*ParamList.Count* プロパティの範囲内で指定します。

ParamScale プロパティ

機能

パラメタの値の小数点以下けた数を数値で取得します。

形式

```
Object.ParamScale(Index)[=Length]
```

パラメタ

- *Object*

ParamList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的のパラメタを数値で設定します。番号は、1~*ParamList.Count* プロパティの範囲内で指定します。

- *Length*

パラメタの値の小数点以下けた数を、-2147483648~2147483647 の数値で設定します。

ParamTypeName プロパティ

機能

パラメタのデータベース依存の属性を文字列で取得します。

形式

```
Object.ParamTypeName(Index)
```

パラメタ

- *Object*

ParamList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的のパラメタを数値で設定します。番号は、1~*ParamList.Count* プロパティの範囲内で指定します。

ParamValue プロパティ

機能

入力パラメタに文字列で値を設定します。又は、出力パラメタから文字列で値を取得します。

形式

```
Object.ParamValue(Index)[= Data]
```

パラメタ

- *Object*
ParamList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的のパラメタを数値で設定します。番号は、1~*ParamList.Count* プロパティの範囲内で指定します。
- *Data*
データを、0~30000 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが假定されます。

使い方

- パラメタの属性が dbpTypeBINARY, dbpTypeVARBINARY, dbpTypeLONGVARBINARY のパラメタに対して、このメソッドで値を設定、又は取得する場合は、ParamCodeType プロパティに dbpCHAR を設定してから *Procedure.Execute* メソッドを発行してください。
- パラメタ番号で指定したパラメタの種別によって、使い方が異なります。

パラメタ種別	用途
dbpParamInput	ストアードプロシジャを実行させるために必要な値を文字列で入力するプロパティとして使用します。 <i>Procedure.Execute</i> メソッド発行前に値を設定します。
dbpParamOutput	ストアードプロシジャの実行結果を文字列で取得するプロパティとして使用します。 <i>Procedure.Execute</i> メソッド発行後に値を取得します。
dbpParamInputOutput	<i>Procedure.Execute</i> メソッド発行前は、ストアードプロシジャを実行させるために必要な値を文字列で入力するプロパティとして使用します。 <i>Procedure.Execute</i> メソッド発行後は、ストアードプロシジャの実行結果を文字列で取得するプロパティとして使用します。
dbpReturnValue	ストアードプロシジャを実行させた時の戻り値を文字列で取得するプロパティとして使用します。 <i>Procedure.Execute</i> メソッド発行後に値を取得します。

ParamValueB プロパティ

機能

入力パラメタにバイト型データで値を設定します。又は、出力パラメタからバイト型データで値を取得します。

形式

```
Object.ParamValueB(Index)[= Data]
```

パラメタ

- *Object*
ParamList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的のパラメタを数値で設定します。番号は、1~*ParamList.Count* プロパティの範囲内で指定します。
- *Data*
データを、バイト型データで設定します。

使い方

- このメソッドで、パラメタの属性が dbpTypeBINARY, dbpTypeVARBINARY, dbpTypeLONGVARBINARY 以外のパラメタに対して値を設定、又は取得する場合は、ParamCodeType プロパティに dbpBINARY を設定してから *Procedure.Execute* メソッドを発行してください。
- パラメタ番号で指定したパラメタの種別によって、使い方が異なります。

パラメタ種別	用途
dbpParamInput	ストアードプロシジャを実行させるために必要な値をバイト型データで入力するプロパティとして使用します。 <i>Procedure.Execute</i> メソッド発行前に値を設定します。
dbpParamOutput	ストアードプロシジャの実行結果をバイト型データで取得するプロパティとして使用します。 <i>Procedure.Execute</i> メソッド発行後に値を取得します。
dbpParamInputOutput	<i>Procedure.Execute</i> メソッド発行前は、ストアードプロシジャを実行させるために必要な値をバイト型データで入力するプロパティとして使用します。 <i>Procedure.Execute</i> メソッド発行後は、ストアードプロシジャの実行結果をバイト型データで取得するプロパティとして使用します。
dbpReturnValue	ストアードプロシジャを実行させた時の戻り値をバイト型データで取得するプロパティとして使用します。 <i>Procedure.Execute</i> メソッド発行後に値を取得します。

ParamValueMaxLength プロパティ

機能

出力パラメタから値を取得するときのバイト数を数値で設定します。

機能

入力パラメタにバイト型データで値を設定します。又は、出力パラメタからバイト型データで値を取得します。

形式

```
Object.ParamValueMaxLength(Index)[= Length]
```

パラメタ

- *Object*
ParamList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的のパラメタを数値で設定します。番号は、1~*ParamList.Count* プロパティの範囲内で指定します。
- *Length*
出力パラメタから取得するデータのバイト数を、1~2147483647 の数値で設定します。

使い方

- パラメタの属性が、dbpLONGVARCHAR、又は dbpLONGVARBINARY のパラメタの値を取得する場合に使用します。上記以外の属性では無視されます。

4.25 ProcedureList オブジェクトの詳細

ProcedureList オブジェクトは、指定されたデータソースに格納されているストアードプロシジャの一覧情報を取得したり、取得した結果を管理したりします。それぞれの情報は、プロパティによって返されます。

ProcedureList オブジェクトのプロパティの一覧を次に示します。ProcedureList オブジェクトのメソッドはありません。

●ProcedureList オブジェクトのプロパティの一覧

プロパティ	説明
Count	取得したストアードプロシジャの個数を数値で取得します。
ProcedureComment	ストアードプロシジャのコメントを文字列で取得します。
ProcedureName	ストアードプロシジャの名称を文字列で取得します。
ProcedureOwner	ストアードプロシジャの所有者を文字列で取得します。
ProcedureType	ストアードプロシジャの型を数値で取得します。

Count プロパティ

機能

取得したストアードプロシジャの個数を数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

ProcedureList オブジェクトへの参照を表すオブジェクト式です。

ProcedureComment プロパティ

機能

ストアードプロシジャのコメントを文字列で取得します。

形式

```
Object.ProcedureComment(Index)
```

パラメタ

- *Object*

ProcedureList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的のストアードプロシジャを数値で設定します。番号は、1~*ProcedureList.Count* プロパティの範囲内で指定します。

ProcedureName プロパティ

機能

ストアードプロシジャの名称を文字列で取得します。

形式

```
Object.ProcedureName(Index)
```

パラメタ

- *Object*

ProcedureList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的のストアードプロシジャを数値で設定します。番号は、1~*ProcedureList.Count* プロパティの範囲内で指定します。

ProcedureOwner プロパティ

機能

ストアードプロシジャの所有者を文字列で取得します。

形式

```
Object.ProcedureOwner(Index)
```

パラメタ

- *Object*

ProcedureList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的のストアードプロシジャを数値で設定します。番号は、1~*ProcedureList.Count* プロパティの範囲内で指定します。

ProcedureType プロパティ

機能

ストアードプロシジャの型を数値で取得します。

形式

```
Object.ProcedureType(Index)
```

パラメタ

- *Object*

ProcedureList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的のストアードプロシジャを数値で設定します。番号は、1~*ProcedureList.Count* プロパティの範囲内で指定します。

戻り値

定数	値	内容
dbpPtUnknown	0	ストアードプロシジャが戻り値を返すかどうか不明です。
dbpPtProcedure	1	戻り値を返さないストアードプロシジャです。
dbpPtFunction	2	戻り値を返すストアードプロシジャです。

使い方

- ストアードプロシジャの型が dbpPtProcedure の場合、戻り値は *ParamList.ParamType* プロパティが dbpReturnValue の列に格納されます。

5

ODBC 関数を使ったデータベースアクセス

この章では、DBPARTNER ODBC Driver をセットアップすると使える ODBC 関数について説明します。

5.1 DBPARTNER ODBC Driver が提供する ODBC 関数の概要

DBPARTNER ODBC Driver をセットアップすると、ODBC 関数 (LEVEL1 と LEVEL2 の一部) が使えます。この ODBC 関数を使ったアプリケーションから、サーバのデータベースにアクセスできます。

DBPARTNER ODBC Driver は、DBPARTNER2 Client と一緒にインストールして ActiveX コントロールのアプリケーションと連携して使うことも、DBPARTNER2 Client 以外のアプリケーションで使うこともできます。

ここでは、DBPARTNER2 Client が提供する ODBC 関数、ODBC とデータベースとのデータ型の対応、及び ODBC 関数の規則について説明します。ODBC 関数については、ODBC ドライバのヘルプでも説明しているので、必要に応じて参照してください。

DBPARTNER ODBC Driver で使える ODBC 関数は、ODBC 2.0 の範囲で使ってください。

5.1.1 ODBC ドライバを使ってアクセスできるデータベース

ODBC ドライバを使ってアクセスできるデータベースについては、「[1.4.2 ODBC ドライバの概要](#)」に示す表 1-3 と表 1-4 を参照してください。

5.2 DBPARTNER ODBC Driver で使える ODBC 関数

5.2.1 DABroker ドライバで使える ODBC 関数

DABroker ドライバ (DBPARTNER/Server ドライバ) で使える ODBC 関数を表に示します。DABroker 経由の場合は表 5-1 を、DABroker-DBS 経由又は DBS 直結の場合は表 5-2 を参照してください。

表 5-1 の項目名称にある「RDA 経由」とは、RDA Link for Gateway 経由で接続するデータベースを表しています。RDA Link for Gateway 経由で接続するデータベースには、RDB1 E2, XDM/RD E2, 及び RDB 編成ファイルがあります。「RDA 経由」を含む表の項目名称に使用している略称については、凡例で説明しています。また、表の中で※印で示している注意事項については、この項の末尾にまとめて記述しています。

表 5-1 DABroker ドライバで使える ODBC 関数 (DABroker 経由の場合)

分類	ODBC 関数	DABroker 直結				RDA 経由	拡張レベル
		HiR	ORA	SQL	Adap		
data source との接続	SQLAllocEnv	○	○	○	○	○	Core
	SQLAllocConnect	○	○	○	○	○	Core
	SQLConnect	○	○	○	○	○	Core
	SQLDriverConnect	○	○	○	○	○	Level1
	SQLBrowseConnect	○	○	○	○	○	Level2
	SQLDataSource	—	—	—	—	—	Level2
	SQLDrivers	—	—	—	—	—	Level2
ドライバと data source の情報取得	SQLGetInfo	○	○	○	○	○	Level1
	SQLGetFunction	○	○	○	○	○	Level1
	SQLGetTypeInfo	○	○	○	○	○	Level1
ドライバオプションの設定と取得	SQLSetConnectOption	○	○	○	○	○	Level1
	SQLGetConnectOption	○	○	○	○	○	Level1
	SQLSetStmtOption	○	○	○	○	○	Level1
	SQLGetStmtOption	○	○	○	○	○	Level1
SQL 要求の作成	SQLAllocStmt	○	○	○	○	○	Core
	SQLPrepare	○	○	○	○	○	Core
	SQLBindParameter	△	△	△	△	△	Level1
	SQLSetParam	○	○	○	○	○	Level1

分類	ODBC 関数	DABroker 直結				RDA 経由	拡張レベル
		HiR	ORA	SQL	Adap		
	SQLParamOption※1	×	×	×	×	×	Level2
	SQLGetCursorName	○	○	○	○	○	Core
	SQLSetCursorName	○	○	○	○	○	Core
	SQLSetScrollOption※2	×	×	×	×	×	Level2
SQL の実行	SQLExecute	○	○	○	○	○	Core
	SQLExecDirect	○	○	○	○	○	Core
	SQLNativeSql	○	○	○	○	○	Level2
	SQLDescribeParam	○	×	○	○	×	Level2
	SQLNumParams	○	○ ※4	○	○	○ ※4	Level2
	SQLParamData	○	○	○	○	○	Level1
	SQLPutData	○	○	○	○	○	Level1
実行結果と実行結果情報の取得	SQLRowCount	○	○	○	○	○	Core
	SQLNumResultCols	○	○	○	○	○	Core
	SQLDescribeCol	○	○	○	○	○	Core
	SQLColAttributes	○	○	○	○	○	Core
	SQLBindCol	○	○	○	○	○	Core
	SQLFetch	○	○	○	○	○	Core
	SQLExtendedFetch※2	×	×	×	×	×	Level2
	SQLGetData	○	○	○	○	○	Level1
	SQLSetPos※2	×	×	×	×	×	Level2
	SQLMoreResults※5	○	○	○	○	○	Level1
	SQLError	○	○	○	○	○	Core
data source のシステム情報の取得	SQLColumnPrivileges	○	○	○	○	○	Level2
	SQLColumns	○	○	○	○	○	Level1
	SQLForeignKeys※1	×	×	×	×	×	Level2
	SQLPrimaryKeys※1	×	○ ※7	○ ※3	○ ※3	×	Level2
	SQLProcedureColumns	○	○	○	○	×	Level2
					※7		

分類	ODBC 関数	DABroker 直結				RDA 経由	拡張レベル
		HiR	ORA	SQL	Adap		
	SQLProcedures	○	○	○	○	× ※7	Level2
	SQLSpecialColumns	○ ※5	○ ※6	○ ※5	○ ※5	○ ※5	Level1
	SQLStatistics	○	○	○	○	○	Level1
	SQLTablePrivileges	○	○	○	○	○	Level2
	SQLTables	○	○	○	○	○	Level1
SQL 実行の 終了	SQLFreeStmt	○	○	○	○	○	Core
	SQLCancel	○	○	○	○	○	Core
	SQLTransact	○	○	○	○	○	Core
切り離し	SQLDisconnect	○	○	○	○	○	Core
	SQLFreeConnect	○	○	○	○	○	Core
	SQLFreeEnv	○	○	○	○	○	Core

表 5-2 DABroker ドライバで使える ODBC 関数 (DABroker-DBS 経由又は DBS 直結の場合)

分類	ODBC 関数	RD	SD	RDB	PDM	RDB1	KSP	拡張レベル
data source との接続	SQLAllocEnv	○	○	○	○	○	○	Core
	SQLAllocConnect	○	○	○	○	○	○	Core
	SQLConnect	○	○	○	○	○	○	Core
	SQLDriverConnect	○	○	○	○	○	○	Level1
	SQLBrowseConnect	○	○	○	○	○	○	Level2
	SQLDataSource	—	—	—	—	—	—	Level2
	SQLDrivers	—	—	—	—	—	—	Level2
ドライバと data source の情報取得	SQLGetInfo	○	○	○	○	○	○	Level1
	SQLGetFunction	○	○	○	○	○	○	Level1
	SQLGetTypeInfo	○	○	○	○	○	○	Level1
ドライバオプション の設定 と取得	SQLSetConnectOption	○	○	○	○	○	○	Level1
	SQLGetConnectOption	○	○	○	○	○	○	Level1
	SQLSetStmtOption	○	○	○	○	○	○	Level1
	SQLGetStmtOption	○	○	○	○	○	○	Level1

分類	ODBC 関数	RD	SD	RDB	PDM	RDB1	KSP	拡張レベル
SQL 要求の作成	SQLAllocStmt	○	○	○	○	○	○	Core
	SQLPrepare	○	○	○	○	○	○	Core
	SQLBindParameter	△	△	△	△	△	× ※7	Level1
	SQLSetParam	○	○	○	○	○	× ※7	Level1
	SQLParamOption※1	×	×	×	×	×	×	Level2
	SQLGetCursorName	○	○	○	○	○	○	Core
	SQLSetCursorName	○	○	○	○	○	○	Core
	SQLSetScrollOption※2	×	×	×	×	×	×	Level2
SQL の実行	SQLExecute	○	○	○	○	○	○	Core
	SQLExecDirect	○	○	○	○	○	○	Core
	SQLNativeSql	○	○	○	○	○	○	Level2
	SQLDescribeParam	○	○	○	○	○	× ※7	Level2
	SQLNumParams	○	○※3	○※3	○※3	○※3	× ※7	Level2
	SQLParamData	○	○	○	○	○	× ※7	Level1
	SQLPutData	○	○	○	○	○	× ※7	Level1
	実行結果と実行結果情報の取得	SQLRowCount	○	○	○	○	○	○
SQLNumResultCols		○	○	○	○	○	○	Core
SQLDescribeCol		○	○	○	○	○	○	Core
SQLColAttributes		○	○	○	○	○	○	Core
SQLBindCol		○	○	○	○	○	○	Core
SQLFetch		○	○	○	○	○	○	Core
SQLExtendedFetch※2		×	×	×	×	×	×	Level2
SQLGetData		○	○	○	○	○	○	Level1
SQLSetPos※2		×	×	×	×	×	×	Level2
SQLMoreResults※5		○	○	○	○	○	○	Level1

分類	ODBC 関数	RD	SD	RDB	PDM	RDB1	KSP	拡張レベル
						※5		
	SQLError	○	○	○	○	○	○	Core
data source のシステム情報 の取得	SQLColumnPrivileges	○	×	○	×	○	×	Level2
	SQLColumns	○	○	○	○	○	○	Level1
	SQLForeignKeys※1	×	×	×	×	×	×	Level2
	SQLPrimaryKeys※1	×	×	×	×	×	×	Level2
		※7	※7	※7	※7	※7	※7	
	SQLProcedureColumns	○	×	×	×	×	×	Level2
		※8	※7	※7	※7	※7	※7	
	SQLProcedures	○	×	×	×	×	×	Level2
		※8	※7	※7	※7	※7	※7	
	SQLSpecialColumns	○	○	○	○	○	○	Level1
	※5	※6	※6	※6	※6	※5		
SQLStatistics	○	○	○	×	○	×	Level1	
				※5		※5		
SQLTablePrivileges	○	×	○	×	○	×	Level2	
				※5		※5		
SQLTables	○	○	○	○	○	○	Level1	
SQL 実行の 終了	SQLFreeStmnt	○	○	○	○	○	○	Core
	SQLCancel	○	○	○	○	○	○	Core
	SQLTransact	○	○	○	○	○	○	Core
切り離し	SQLDisconnect	○	○	○	○	○	○	Core
	SQLFreeConnect	○	○	○	○	○	○	Core
	SQLFreeEnv	○	○	○	○	○	○	Core

(凡例)

- ：関数をサポートしています。
- ×
- △：SQLSetParam の機能だけをサポートしています。
- HiR：HiRDB のことです。
- ORA：ORACLE のことです。
- SQL：SQL Server のことです。
- Adap：Adaptive Server Anywhere のことです。

RDA 経由：RDA Link for Gateway 経由で接続するデータベースを表します。このデータベースには、RDB1 E2, XDM/RD E2, 及び RDB 編成ファイルがあります。

RD：XDM/RD E2 のことです。

SD：XDM/SD E2 のことです。

RDB：RDB 編成ファイルのことです。

PDM：PDMII E2 のことです。

KSP：VOSK スプールファイルのことです。

RDB1：RDB1 E2 のことです。

Core：コアレベルのことです。

注※1

DABroker ドライバではこの関数はサポートしていないので、ドライバマネージャがエラーリターンします (SQL STATE : IM001)。

注※2

DABroker ドライバではサポートしていません。ただし、ODBC 2.0 カーソルライブラリで実装しているので、カーソルライブラリで規定された範囲の機能は使えます。

注※3

この関数をサポートしていますが、DBMS 又はサーバプログラムがサポートしていないときは、常に 0 が返ります。

注※4

この関数をサポートしていますが、サーバ側でサポートしていません。関数は使えますが、常に NO_DATA_FOUND が返ります。

注※5

この関数をサポートしていますが、DBMS がサポートしていないときは、常に NO_DATA_FOUND が返ります。

注※6

fColType が SQL_BEST_ROWID に対してだけ情報を返します。

注※7

この関数をサポートしていますが、該当する DBMS でサポートしていないのでエラーリターンします (SQL STATE : S1C00)。

注※8

外部プロシジャだけをサポートしています。

5.3 ODBC とデータベースとのデータ型の対応

ODBC とデータベースとのデータ型の対応を、DBPARTNER2 Client のドライバ別に説明します。ODBC のデータ型とは、ODBC 関数のアーギュメントに指定する SQL データ型のことです。

5.3.1 DABroker ドライバの場合の ODBC とデータベースとのデータ型の対応

(1) ODBC とデータベースとのデータ型の対応

DABroker ドライバ (DBPARTNER/Server ドライバ) の場合の、ODBC とデータベースとのデータ型の対応について表 5-3～表 5-11 に示します。

表 5-3～表 5-11 で使用している略号、及び凡例の意味は次のとおりです。また、表の中で※印で示している注意事項については、この項の末尾にまとめて記述しています。

(略号)

HiR : HiRDB のことです。

ORA : ORACLE のことです。

SQL : SQL Server のことです。

Adap : Adaptive Server Anywhere のことです。

RDB1 : RDB1 E2 のことです。

RDB 編成 : RDB 編成ファイルのことです。

RD : XDM/RD E2 のことです。

SD : XDM/SD E2 のことです。

PDM : PDMII E2 のことです。

KSP : VOSK スプールファイルのことです。

RDA (経由) : RDA Link for Gateway のことです。

DBS (経由, 直結) : Database Connection Server のことです。

(凡例)

— : ODBC にはないデータ型を示します。

○ : データ型を使えます。

● : データベースに該当するデータ型がありません。

× : ODBC ドライバに該当するデータ型がないので使えません。

(a) 文字データ型の対応と各データベースでの使用可否

ODBC とデータベースとの文字データ型の対応を表 5-3 に、各データベースでの文字データ型の使用可否を表 5-4 と表 5-5 に示します。

表 5-4/表 5-5 で使用可否を示す文字データ型については、表 5-3 を参照してください。表 5-3 と表 5-4/表 5-5 は、項番で対応しています。

表 5-3 ODBC とデータベースのデータ型の対応 (文字データ)

項番	ODBC のデータ型	データベースのデータ型	説明
1	SQL_CHAR	CHAR(n), ROWID	固定長文字列
2	SQL_VARCHAR	VARCHAR(n)	可変長文字列
3	SQL_LONGVARCHAR	LONG VARCHAR(n), VARCHAR2, LONG* ¹	可変長長文字列
4	—* ²	NCHAR(n), NATIONAL CHARACTER	固定長各国文字列
5	—* ²	NVARCHAR(n)	可変長各国文字列
6	—* ²	LONG NVARCHAR(n)	可変長長各国文字列
7	—* ²	MCHAR(n)	固定長混在文字列
8	—* ²	MVARCHAR(n)	可変長混在文字列
9	—* ²	LONG MVARCHAR(n)	可変長長混在文字列

表 5-4 各データベースでの文字データ型の使用可否 (DABroker 経由)

項番	DABroker 経由													
	DABroker 直結				RDA 経由				DBS 経由					
	HiR	ORA	SQL	Adap	RD	RDB1	RDB 編成	RD	SD* ⁷	RDB 編成	PDM * ⁸	RDB1	KSP * ⁵	
1	○	○	○	●	○	○	○	○	○	○	○	○	○	●
2	○	○	○	○	○	○	●	○	●	●	●	○	●	
3	●	○	○	○	○	○	●	○	●	●	●	○	●	
4	○	●	●	●	○	○	○	○	○	○	○	○	●	
5	○	●	●	●	○	○	●	○	●	●	●	○	●	
6	●	●	●	●	○	●	●	○	●	●	●	●	●	
7	○	●	●	●	○	●	○	○	●	○	●	●	○	
8	○	●	●	●	○	●	●	○	●	●	●	●	●	
9	●	●	●	●	○	●	●	○	●	●	●	●	●	

表 5-5 各データベースでの文字データ型の使用可否 (DBS 直結)

項番	DBS 直結					
	RD	SD ^{※7}	RDB 編成	PDM ※8	RDB1	KSP ※5
1	○	○	○	○	○	●
2	○	●	●	●	○	●
3	○	●	●	●	○	●
4	○	○	○	○	○	●
5	○	●	●	●	○	●
6	○	●	●	●	●	●
7	○	●	○	●	●	○
8	○	●	●	●	●	●
9	○	●	●	●	●	●

(b) 数データ型の対応と各データベースでの使用可否

ODBC とデータベースとの数データ型の対応を表 5-6 に、各データベースでの数データ型の使用可否を表 5-7 と表 5-8 に示します。

表 5-7/表 5-8 で使用可否を示す数データ型については、表 5-6 を参照してください。表 5-6 と表 5-7/表 5-8 は、項番で対応しています。

表 5-6 ODBC とデータベースのデータ型の対応 (数データ)

項番	ODBC のデータ型	データベースのデータ型	説明
1	SQL_DECIMAL	DEC [IMAL] (p,s) ^{※6}	固定小数点数
2	SQL_NUMERIC	NUMERIC, NUMBER (p,s) ^{※6}	精度 $1 \leq p \leq 15$, $0 \leq s \leq p$
3		LARGE DEC [IMAL] (p,s) ^{※6}	拡張精度固定小数点数 $1 \leq p \leq 29$, $0 \leq s \leq p$
4	SQL_DECIMAL	PACKED DECIMAL FIXED WITH NO SIGN	符号なし PACK 形式
5	— ^{※4}	WZONE	符号なし ZONE 形式
6	SQL_DECIMAL	UNPACKED DECIMAL FIXED WITH NO SIGN	符号なし ZONE 形式
7	SQL_SMALLINT	SMALLINT	値の範囲が-32,768~32,767 の整数
8	SQL_INTEGER	INTEGER	値の範囲が-2,147,483,648~2,147,483,647 の整数

項番	ODBC のデータ型	データベースのデータ型	説明
9	SQL_TINYINT	TINYINT	-256~255 の整数
10	SQL_BIGINT	BIGINT	1 けたの符号と 19 けたの整数
11	SQL_REAL	SMALLFLT, REAL	単精度浮動小数点数
12	SQL_FLOAT	FLOAT, DOUBLE PRECISION	倍精度浮動小数点数
13	SQL_DOUBLE	DOUBLE	倍精度浮動小数点数
14	SQL_BIT	BIT	ビット
15	SQL_BINARY	BINARY	固定長バイナリデータ
16	SQL_VARBINARY	VARBINARY, RAW	可変長バイナリデータ
17	SQL_LONGVARBINARY	LONG VARBINARY, LONG RAW※1	可変長バイナリデータ

表 5-7 DABroker ドライバ経由での数データの使用可否 (DABroker 経由)

項番	DABroker 経由													
	DABroker 直結				RDA 経由			DBS 経由						
	HiR	ORA	SQL	Adap	RD	RDB1	RDB 編成	RD	SD※7	RDB 編成	PDM ※8	RDB1	KSP ※5	
1	●	●	○	○	○	○	○	○	○	○	○	○	○	●
2	●	○	○	○	●	●	○	●	●	○	●	●	●	●
3	○	●	●	●	○	●	●	○	●	●	●	●	●	●
4	●	●	●	●	●	●	●	●	○	●	●	●	●	●
5	●	●	●	●	●	●	●	●	●	●	●	●	●	●
6	●	●	●	●	●	●	●	●	○	●	●	●	●	●
7	○	●	○	○	○	○	○	○	○	○	○	○	○	●
8	○	●	○	○	○	○	○	○	○	○	○	○	○	●
9	●	●	○	○	●	●	●	●	●	●	●	●	●	●
10	●	●	●	○	●	●	●	●	●	●	●	●	●	●
11	○	●	○	○	○	○	●	○	●	●	●	○	●	●
12	○	○	○	○	○	○	●	○	●	●	●	○	●	●
13	●	●	○	○	●	●	●	●	●	●	●	●	●	●
14	●	●	○	○	●	●	●	●	●	●	●	●	●	●
15	●	●	○	○	●	●	○	●	○	○	○	●	●	●
16	●	○	○	●	●	●	●	●	●	●	●	●	●	●

項番	DABroker 経由													
	DABroker 直結				RDA 経由			DBS 経由						
	HiR	ORA	SQL	Adap	RD	RDB1	RDB 編成	RD	SD※7	RDB 編成	PDM ※8	RDB1	KSP ※5	
17	●	○	○	○	●	●	●	○	●	●	●	●	●	

表 5-8 DABroker ドライバ経由での数データの使用可否 (DBS 直結)

項番	DBS 直結					
	RD	SD※7	RDB 編成	PDM ※8	RDB1	KSP ※5
1	○	○	○	○	○	●
2	●	●	○	●	●	●
3	○	●	●	●	●	●
4	●	○	●	●	●	●
5	●	●	●	●	●	●
6	●	○	●	●	●	●
7	○	○	○	○	○	●
8	○	○	○	○	○	●
9	●	●	●	●	●	●
10	●	●	●	●	●	●
11	○	●	●	●	○	●
12	○	●	●	●	○	●
13	●	●	●	●	●	●
14	●	●	●	●	●	●
15	●	○	○	○	●	●
16	●	●	●	●	●	●
17	○	●	●	●	●	●

(c) 日付時刻データ型の対応と各データベースでの使用可否

ODBC とデータベースとの日付時刻データ型の対応を表 5-9 に、各データベースでの日付時刻データ型の使用可否を表 5-10 と表 5-11 に示します。

表 5-10/表 5-11 で使用可否を示す日付時刻データ型については、表 5-9 を参照してください。表 5-9 と表 5-10/表 5-11 は、項番で対応しています。

表 5-9 ODBC とデータベースのデータ型の対応 (日付時刻データ)

項番	ODBC のデータ型	データベースのデータ型	説明
1	SQL_DATE	DATE	日付
2	SQL_TIMESTAMP	DATE	固定長日付/時刻値
3	SQL_TIME	TIME	時刻
4	—※2	INTERVAL YEAR TO DAY	日間隔
5	—※2	INTERVAL HOUR TO SECOND	時間隔

表 5-10 DABroker ドライバ経由での日付時刻データの使用可否 (DABroker 経由)

項番	DABroker 経由													
	DABroker 直結				RDA 経由			DBS 経由						
	HiR	ORA	SQL	Adap	RD	RDB1	RDB 編成	RD	SD ※7	RDB 編成	PDM ※8	RDB1 ※5	KSP ※5	
1	○	●	●	○	○※3	●	●	○※3	●	●	●	●	●	
2	●	○	○	○	●	●	●	●	●	●	●	●	●	
3	○	●	●	○	×	●	●	○	●	●	●	●	●	
4	×	×	●	●	×	●	●	×	●	●	●	●	●	
5	×	×	●	●	×	●	●	×	●	●	●	●	●	

表 5-11 DABroker ドライバ経由での日付時刻データの使用可否 (DBS 直結)

項番	DBS 直結					
	RD	SD※7	RDB 編成	PDM ※8	RDB1 ※5	KSP ※5
1	○※3	●	●	●	●	●
2	●	●	●	●	●	●
3	○	●	●	●	●	●
4	×	●	●	●	●	●
5	×	●	●	●	●	●

注

データ型の最大文字列長と値の範囲は、各データベースで異なります。各データベースのマニュアルを参照してください。

注※1

データの最大長は、DABroker 動作環境定義ファイルの DABORADBA_LONGBUFSIZE で指定した値です。

注※2

「DBPARTNER2 Client と DABroker ドライバ」を使っているときは、データベースのデータ型がそのまま通知されます。それ以外のときは、アプリケーションに固定長文字列 (CHAR) として通知されます。

注※3

XDM/RD E2 01-02以降を使っているときに対応します。

注※4

「DBPARTNER2 ClientとDABrokerドライバ」を使っているときは、データベースのデータ型がそのまま通知されます。それ以外のときは、アプリケーションに固定小数点数（SQL_NUMERIC）として通知されます。

注※5

バージョンが01-02以降のDatabase Connection Server、又はバージョンが02-04以降のDABrokerを使っているときに対応します。

注※6

「(p,s)」の「p」は全体のけた数、「s」は小数点以下のけた数を示します。

注※7

バージョンが02-00以降のDatabase Connection Server、又はバージョンが02-01以降のDABrokerを使っているときに対応します。

注※8

Database Connection Server、又はバージョンが02-04以降のDABrokerを使っているときに対応します。

(2) データベースのデータ型とSQLデータ型の対応

データベースのデータ型に対するSQLデータ型の対応を次に示します。

(a) SQL Server の場合

SQL Server のデータ型に対するSQLデータ型の対応を表5-12に示します。

表 5-12 SQL Server のデータ型とSQLデータ型の対応

データベースのデータ型	SQLデータ型
bit	SQL_BIT
tinyint	SQL_TINYINT
smallint	SQL_SMALLINT
int	SQL_INTEGER
binary	SQL_BINARY
timestamp	SQL_BINARY
varbinary	SQL_VARBINARY
image	SQL_LONGVARBINARY
char	SQL_CHAR
varchar	SQL_VARCHAR
sysname	SQL_VARCHAR
text	SQL_LONGVARCHAR
decimal	SQL_DECIMAL

データベースのデータ型	SQL データ型
money	SQL_DECIMAL
smallmoney	SQL_DECIMAL
numeric	SQL_NUMERIC
real	SQL_REAL
float	SQL_FLOAT 又は SQL_DOUBLE
datetime	SQL_TIMESTAMP
smalldatetime	SQL_TIMESTAMP

(b) Adaptive Server Anywhere の場合

Adaptive Server Anywhere のデータ型に対する SQL データ型の対応を表 5-13 に示します。

表 5-13 Adaptive Server Anywhere のデータ型と SQL データ型の対応

データベースのデータ型	SQL データ型
bit	SQL_BIT
tinyint	SQL_TINYINT
oldbit	SQL_TINYINT
smallint	SQL_SMALLINT
integer	SQL_INTEGER
bigint	SQL_BIGINT
binary	SQL_BINARY
varbinary	SQL_BINARY
java serialization	SQL_LONGVARBINARY
java.lang.Object	SQL_LONGVARBINARY
long binary	SQL_LONGVARBINARY
image	SQL_LONGVARBINARY
char	SQL_VARCHAR
varchar	SQL_VARCHAR
sysname	SQL_VARCHAR
long varchar	SQL_LONGVARCHAR
text	SQL_LONGVARCHAR
decimal	SQL_DECIMAL
numeric	SQL_NUMERIC

データベースのデータ型	SQL データ型
money	SQL_NUMERIC
smallmoney	SQL_NUMERIC
float	SQL_REAL
double	SQL_DOUBLE
date	SQL_DATE
time	SQL_TIME
timestamp	SQL_TIMESTAMP
datetime	SQL_TIMESTAMP
smalldatetime	SQL_TIMESTAMP

(c) XDM/SD E2 の場合

XDM/SD E2 のデータ型に対する SQL データ型の対応を表 5-14 に示します。

表 5-14 XDM/SD E2 のデータ型と SQL データ型の対応

データベースのデータ型	SQL データ型
CHARACTER	SQL_CHAR
NCHARACTER	SQL_CHAR
COMP (2 バイト)	SQL_SMALLINT
COMP (4 バイト)	SQL_INTEGER
PACKED DECIMAL FIXED	SQL_DECIMAL
UNPACKED DECIMAL FIXED	SQL_DECIMAL
PACKED DECIMAL FIXED WITH NO SIGN	SQL_DECIMAL
UNPACKED DECIMAL FIXED WITH NO SIGN	SQL_DECIMAL
BIT	SQL_BINARY
データベースキー	SQL_BINARY

(d) RDB 編成ファイル (DBS 接続) の場合

RDB 編成ファイル (DBS 接続) のデータ型に対する SQL データ型の対応を表 5-15 に示します。

表 5-15 RDB 編成ファイル (DBS 接続) のデータ型と SQL データ型の対応

データベースのデータ型	SQL データ型
CHAR	SQL_CHAR
PACK	SQL_DECIMAL

データベースのデータ型	SQL データ型
ZONE	SQL_NUMERIC
WZONE	SQL_NUMERIC
BINARY (2 バイト)	SQL_SMALLINT
BINARY (4 バイト)	SQL_INTEGER
HEX	SQL_BINARY
NCHAR	SQL_CHAR
MCHAR	SQL_CHAR
BIT	SQL_BINARY

(e) PDMII E2 の場合

PDMII E2 のデータ型に対する SQL データ型の対応を表 5-16 に示します。

表 5-16 PDMII E2 のデータ型と SQL データ型の対応

データベースのデータ型	SQL データ型
CHARACTER	SQL_CHAR
NCHARACTER	SQL_CHAR
COMP (2 バイト)	SQL_SMALLINT
COMP (4 バイト)	SQL_INTEGER
PACKED DECIMAL FIXED	SQL_DECIMAL
UNPACKED DECIMAL FIXED	SQL_DECIMAL
DBK (データベースキー)	SQL_BINARY

(f) RDB1 E2 の場合

RDB1 E2 のデータ型に対する SQL データ型の対応を表 5-17 に示します。

表 5-17 RDB1 E2 のデータ型と SQL データ型の対応

データベースのデータ型	SQL データ型
CHAR	SQL_CHAR
VARCHAR	SQL_VARCHAR
LONG VARCHAR	SQL_LONGVARCHAR
NCHAR	SQL_CHAR
NVARCHAR	SQL_VARCHAR
INTEGER	SQL_INTEGER

データベースのデータ型	SQL データ型
SMALLINT	SQL_SMALLINT
DECIMAL	SQL_DECIMAL
FLOAT	SQL_FLOAT
SMALLFLT	SQL_REAL

(g) VOSK スプールファイルの場合

VOSK スプールファイルのデータ型に対する SQL データ型の対応を表 5-18 に示します。

表 5-18 VOSK スプールファイルのデータ型と SQL データ型の対応

データベースのデータ型	SQL データ型
MCHAR	SQL_CHAR

(3) データベースと接続するときの SQLGetInfo の戻り値

次のデータベースと接続するときの SQLGetInfo の戻り値を表 5-19 に示します。

- SQL Server
- Adaptive Server Anywhere
- XDM/SD E2
- PDMII E2
- RDB1 E2
- VOSK スプールファイル

表 5-19 データベースと接続するときの SQLGetInfo の戻り値

FinfoType	SQL Server	Adaptive Server Anywhere	XDM/SD E2	PDMII E2	RDB1 E2	VOSK スプールファイル
SQL_ACCESSIBLE_PROCEDURES	NO	NO	NO	NO	NO	NO
SQL_ACCESSIBLE_TABLES	YES	NO	YES	YES	YES	YES
SQL_ACTIVE_CONNECTIONS	0	0	0	0	0	0
SQL_ACTIVE_STATEMENTS	10 (01-03 以降では 64)		1	128	DBS 経由：128 DABroker 経由：64	128

FinfoType	SQL Server	Adaptive Server Anywhere	XDM/SD E2	PDMII E2	RDB1 E2	VOSK スプールファイル
SQL_ALTER_TABLE	ADD	DROP	0	0	ADD [0] ※	0
SQL_BOOKMARK_PERSISTENCE	0	0	0	0	0	0
SQL_COLUMN_ALIAS	NO	NO	NO	NO	NO	NO
SQL_CONCAT_NULL_BEHAVIOR	NULL でない値					
SQL_CONVERT_FUNCTIONS	0	0	0	0	0	0
SQL_CONVERT_Type	0	0	0	0	0	0
SQL_CORRELATION_NAME	ANY	ANY	NONE	NONE	DIFFERENT	NONE
SQL_CURSOR_COMMIT_BEHAVIOR	カーソルの CLOSE		DELETE	CLOSE	CLOSE	DELETE
SQL_CURSOR_ROLLBACK_BEHAVIOR	カーソルの CLOSE		DELETE	CLOSE	CLOSE	DELETE
SQL_DATA_SOURCE_NAME	Connect 時に設定					
SQL_DATA_SOURCE_READ_ONLY	NO	NO	NO	NO	NO	YES
SQL_DATABASE_NAME	NULL (データ長は 0)					
SQL_DBMS_NAME	接続したサーバの名称					
SQL_DBMS_VER	接続したサーバのバージョン					
SQL_DEFAULT_TXN_ISOLATION	すべてのトランザクションで変更が認識されます。 (SQL_TXN_READ_UNCOMMITTED)					
SQL_DRIVER_NAME	DBPSVD32.dll					
SQL_DRIVER_ODBC_VER	02.50					
SQL_DRIVER_VER	ドライバのバージョン					
SQL_EXPRESSIONS_IN_ORDERBY	YES	YES	NO	NO	YES	NO
SQL_FETCH_DIRECTION	NEXT					
SQL_FILE_USAGE	NOT SUPPORT					
SQL_GETDATA_	ANY COLUMN					

InfoType	SQL Server	Adaptive Server Anywhere	XDM/SD E2	PDMII E2	RDB1 E2	VOSK スプールファイル
EXTENSIONS						
SQL_GROUP_BY	CONTAINS_SELECT	NO_RELATION	NOT_SUPPORTED	NOT_SUPPORTED	EQUALS_SELECT	NOT_SUPPORTED
SQL_IDENTIFIER_CASE	大小文字混在			SENSITIVE	UPPER	SENSITIVE
SQL_IDENTIFIER_QUOTE_CHAR	" (ダブルクォーテーション)					
SQL_KEYWORDS	NULL (データ長は 0)					
SQL_LIKE_ESCAPE_CLAUSE	YES	YES	NO	NO	YES	NO
SQL_LOCK_TYPES	0	0	0	0	0	0
SQL_MAX_BINARY_LITERAL_LEN	0	0	0	0	0	0
SQL_MAX_CHAR_LITERAL_LEN	0	0	0	0	0	0
SQL_MAX_COLUMN_NAME_LEN	30	128	30	18	18	4
SQL_MAX_COLUMNS_IN_GROUP_BY	16	0	0	0	16	0
SQL_MAX_COLUMNS_IN_INDEX	16	0	0	0	8	0
SQL_MAX_COLUMNS_IN_ORDER_BY	16	0	0	0	16	0
SQL_MAX_COLUMNS_IN_SELECT	4000	0	0	0	255	0
SQL_MAX_COLUMNS_IN_TABLE	250	0	0	0	255	511
SQL_MAX_CURSOR_NAME_LEN	30	128	30	0	18	0
SQL_MAX_QUALIFIER_NAME_LEN	30	0	0	0	0	0
SQL_MAX_INDEX_SIZE	900	0	0	0	254	0
SQL_MAX_OWNER_	30	128	7	8	8	8

InfoType	SQL Server	Adaptive Server Anywhere	XDM/SD E2	PDMII E2	RDB1 E2	VOSK スプールファイル
NAME_LEN						
SQL_MAX_PROCEDURE_NAME_LEN	36	128	0	0	0	0
SQL_MAX_ROW_SIZE	1962	2147483647	32000	32000	4056	32767
SQL_MAX_ROW_SIZE_INCLUDES_LONG	NO	YES	YES	NO	NO	NO
SQL_MAX_STATEMENT_LEN	0	0	30000	30000	16384	8000
SQL_MAX_TABLE_NAME_LEN	30	128	30	8	18	17
SQL_MAX_TABLES_IN_SELECT	16	0	1	1	15	1
SQL_MAX_USER_NAME_LEN	30	128	7	8	8	8
SQL_MULT_RESULT_SETS	NO	NO	NO	NO	NO	NO
SQL_MULTIPLE_ACTIVE_TXN	YES	YES	YES	YES	YES	YES
SQL_NEED_LONG_DATA_LEN	YES	NO	NO	NO	NO	NO
SQL_NON_NULLABLE_COLUMNS	NULL でない値					
SQL_NULL_COLLATION	LOW	START	LOW	0	LOW	0
SQL_NUMERIC_FUNCTIONS	0	0	0	0	0	0
SQL_ODBC_API_CONFORMANCE	Level1					
SQL_ODBC_SAG_CLI_CONFORMANCE	NOT SAG-compliant					
SQL_ODBC_SQL_CONFORMANCE	Core					
SQL_ODBC_SQL_OPT_IEF	NO	NO	NO	NO	NO	NO
SQL_ORDER_BY_COLUMNS_IN_SELECT	NO	NO	NO	NO	YES	NO
SQL_OUTER_JOINS	YES	FULL	NO	NO	NO	NO

InfoType	SQL Server	Adaptive Server Anywhere	XDM/SD E2	PDMII E2	RDB1 E2	VOSK スプールファイル
SQL_OWNER_TERM	所有者	Creator	owner	owner	owner	owner
SQL_OWNER_USAGE	DML, TABLE, INDEX, PRIVILEGE		DML	DML	DML, TABLE, INDEX, PRIVILEGE [DML] ※	DML
SQL_POS_OPERATIONS	0	0	0	0	0	0
SQL_POSITIONED_STATEMENTS	0	0	0	0	0	0
SQL_QUALIFIER_NAME_SEPARATOR	Period	NULL	Period	Period	Period	Period
SQL_QUALIFIER_TERM	データベース	NULL	database	database	database	database
SQL_PROCEDURE_TERM	NULL	NULL	NULL	NULL	NULL	NULL
SQL_PROCEDURES	NO	NO	NO	NO	NO	NO
SQL_QUALIFIER_LOCATION	START	0	0	0	0	0
SQL_QUALIFIER_USAGE	DML, TABLE	0	DML	0	DML	0
SQL_QUOTED_IDENTIFIER_CASE	SENSITIVE	UPPER	UPPER	SENSITIVE	UPPER	SENSITIVE
SQL_ROW_UPDATES	NO	NO	NO	NO	NO	NO
SQL_SCROLL_CONCURRENCY	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only
SQL_SCROLL_OPTIONS	Forward Only	Forward Only	Forward Only	Forward Only	Forward Only	Forward Only
SQL_SEARCH_PATTERN_ESCAPE	'¥'	'-'	NULL	'¥'	'¥'	'¥'
SQL_SERVER_NAME	DABroker, 又は DBS					
SQL_SPECIAL_CHARACTERS	NULL	"@\$#"	NULL	NULL	NULL	NULL
SQL_STATIC_SENSITIVITY	0	0	0	0	0	0
SQL_STRING_FUNCTIONS	0	0	CONCAT, TRIM	0	0	0
SQL_SUBQUERIES	0	0	0	0	0	0

FinfoType	SQL Server	Adaptive Server Anywhere	XDM/SD E2	PDMII E2	RDB1 E2	VOSK スプールファイル
SQL_SYSTEM_FUNCTIONS	0	0	0	0	0	0
SQL_TIMEDATE_ADD_INTERVALS	0	0	0	0	0	0
SQL_TIMEDATE_DIFF_INTERVALS	0	0	0	0	0	0
SQL_TABLE_TERM	テーブル	Table	Table	table	Table	table
SQL_TIMEDATE_FUNCTIONS	0	0	0	0	0	0
SQL_TXN_CAPABLE	ALL	DDL_COMMIT	DML	DML	DML	DML
SQL_TXN_ISOLATION_OPTIONS	未サポート					
SQL_UNION	ALL	ALL	0	0	0	0
SQL_USER_NAME	ログインユーザ ID					

注※

Database Connection Server 経由, DABroker 経由 Database Connection Server 又は DABroker 経由 RDA Link for Gateway の場合。 [] 内の値は DABroker 経由 RDA Link for Gateway の場合。

5.4 ODBC 関数の規則

ODBC 関数を使ったアプリケーションからサーバ上のデータベースにアクセスするときは、データベースによって使える機能が異なります。ODBC 関数の規則について、ドライバ別に説明します。

5.4.1 DABroker ドライバの場合に、各データベースで使える機能

DABroker ドライバ (DBPARTNER/Server ドライバ) の場合に、各データベースで使える機能を表 5-20～表 5-22 に示します。

表 5-20～表 5-22 で使用している略号、及び凡例の意味は次のとおりです。また、表の中で※印で示している注意事項については、この項の末尾にまとめて記述しています。

(略号)

HiR：HiRDB のことです。

ORA：ORACLE のことです。

SQL：SQL Server のことです。

Adap：Adaptive Server Anywhere のことです。

RDB1：RDB1 E2 のことです。

RDB 編成：RDB 編成ファイルのことです。

RD：XDM/RD E2 のことです。

SD：XDM/SD E2 のことです。

PDM：PDMII E2 のことです。

KSP：VOSK スプールファイルのことです。

RDA (経由)：RDA Link for Gateway のことです。

DBS (経由, 直結)：Database Connection Server のことです。

(凡例)

○：データベースで使えます。

×：データベースで使えません。

－：データベースに機能がありません。

表 5-20 データベースで使える機能 (1)

機能	HiR	ORA	SQL	Adap	RDA 経由		
					RD	RDB1	RDB 編成
スペシャルカラム情報の取得	－	○	－	－	－	－	－
インデックス情報の取得	○	○	○	○	○	○	○
日付, 時刻データ型の使用	○※1	○※2	○	○	○※3	－	－

機能	HiR	ORA	SQL	Adap	RDA 経由		
					RD	RDB1	RDB 編成
繰り返し列の使用	×	—	—	—	×	×	×
配列列の使用	×	—	—	—	×	—	×
表ヘッダ, 列ヘッダの取得*4	—	—	—	—	○	×	○
非同期処理	○	○	○	○	○	○	○
LIKE のエスケープ文字の使用	×	×	×	×	×	×	×
更新行数の取得	○	○	○	○	○	○	○
LOGIN タイムアウト時間設定	×	×	×	×	×	×	×
日本語データ型の使用	○*5	—	—	—	○*5	○*5	○*5
定義系 SQL の実行	○	○	○	○	○*6	×	×
外部結合エスケープ処理	○	○	○	○	○*7	—	—

表 5-21 データベースで使える機能 (2)

機能	DABroker 経由					
	RD	SD	RDB 編成	PDM	RDB1	KSP
スペシャルカラム情報の取得	—	○	○	○	○	—
インデクス情報の取得	○	○	○	—	○	—
日付, 時刻データの使用	○*3	—	—	—	—	—
繰り返し列の使用	○	○	○	—	○	—
配列列の使用	○	○	×	—	—	—
表ヘッダ, 列ヘッダの取得*4	○	○	○	○	○	○
非同期処理	○	○	○	○	○	○
LIKE エスケープ文字の使用	×	—	×	—	×	—
更新行数の取得	○	○	○	○	○	○
LOGIN タイムアウト時間設定	×	×	×	×	×	×
日本語データ型の使用*5	○	○	○	○	○	○
定義系 SQL の実行	○*6	×	×	×	○	×
外部結合エスケープ処理	○*7	—	—	—	—	—

表 5-22 データベースで使える機能 (3)

機能	DBS 直結					
	RD	SD	RDB 編成	PDM	RDB1	KSP
スペシャルカラム情報の取得	—	○	○	○	○	—
インデクス情報の取得	○	○	○	—	○	—
日付, 時刻データの使用	○※3	—	—	—	—	—
繰り返し列の使用	○	○	○	—	○	—
配列列の使用	○	○	×	—	—	—
表ヘッダ, 列ヘッダの取得※4	○	○	○	○	○	○
非同期処理	○	○	○	○	○	○
LIKE エスケープ文字の使用	×	—	×	—	×	—
更新行数の取得	○	○	○	○	○	○
LOGIN タイムアウト時間設定	×	×	×	×	×	×
日本語データ型の使用※5	○	○	○	○	○	○
定義系 SQL の実行	○※6	×	×	×	○	×
外部結合エスケープ処理	○※7	—	—	—	—	—

注※1

INTERVAL YEAR TO DAY と INTERVAL HOUR TO SECOND は使えません。

注※2

TIMESTAMP は使えます。

注※3

XDM/RD E2 01-02 以降の場合, 日付データ型は使えます。時刻データ型は使えません。

注※4

XDM/RD E2 の場合は, 取得するかしないかを接続先定義ファイルのパラメタで指定します。XDM/RD E2 以外の場合は, アプリケーションには NULL を返します。

注※5

「DBPARTNER2 Client と DBPARTNER/Server ドライバ」を使っているときは, データベースのデータ型がそのまま通知されます。それ以外のときは, アプリケーションに固定長文字列 (CHAR), 可変長文字列 (VARCHAR), 可変長長文字列 (LONG VARCHAR) として通知されます。

注※6

定義系 SQL のうち, CREATE TABLE を実行できます。

注※7

XDM/RD E2 02-00 以降の場合だけ使えます。

5.4.2 各 ODBC 関数での繰り返し列と配列列の扱い

次に示す ODBC 関数での繰り返し列と配列列の扱いについて説明します。

- SQLColumns 関数
- SQLColumnPrivileges 関数
- SQLStatistics 関数
- SQLDescribeCol 関数
- SQLDescribeParam 関数

なお、SQLColumns 関数、SQLColumnPrivileges 関数、SQLDescribeCol 関数、及び SQLDescribeParam 関数では、繰り返し列と配列列の扱いが同じです。

(1) SQLColumns 関数などでの繰り返し列と配列列の扱い

列の一覧情報を扱う SQLColumns 関数や、列権限一覧情報を扱う SQLColumnPrivileges 関数では、繰り返し列と配列列のそれぞれの要素を一つの列として扱います。同様に、SQLDescribeCol 関数と SQLDescribeParam 関数でも、繰り返し列と配列列のそれぞれの要素を一つの列として扱います。

これらの関数では、要素の数だけ列名が返されます。このとき、列名は次に示す形式で返されます。

列名(n) : 「n」は要素数を示します。

例えば、列名が「C1」で、配列要素数が「3」の繰り返し列（又は配列列）の列情報を SQLColumns 関数で取得する場合、列名は、「C1(1)」、「C1(2)」、「C1(3)」の形式で返されます。

(2) SQLStatistics 関数での繰り返し列と配列列の扱い

列のインデクス一覧情報を扱う SQLStatistics 関数では、同じ列名の繰り返し列全体（又は同じ列名の配列列全体）を一つの列として扱います。したがって、要素が複数ある場合でも、要素が特定されないで列名だけが返されます。

列名が「D1」で、配列要素数が「5」の繰り返し列（又は配列列）のインデクス一覧情報を SQLStatistics 関数で取得する場合、列名は「D1」で返されます。

5.5 DBPARTNER ODBC Driver で取得できる ODBC 関数のオプション値

DBPARTNER ODBC Driver で取得できる ODBC 関数のオプション値について説明します。オプション値を取得できる ODBC 関数を次に示します。

- SQLSetConnectOption 関数, SQLSetStmtOption 関数
- SQLGetConnectOption 関数, SQLGetStmtOption 関数

5.5.1 DABroker ドライバで取得できる ODBC 関数のオプション値

DABroker ドライバ (DBPARTNER/Server ドライバ) で取得できる ODBC 関数のオプション値について説明します。

(1) SQLSetConnectOption 関数, SQLSetStmtOption 関数

SQLSetConnectOption 関数で取得できるオプション値とサポート状況を表 5-23 に、SQLSetStmtOption 関数で取得できるオプション値とサポート状況を表 5-24 に示します。表の備考は、ODBC 関数がサポートされた ODBC のバージョンを示します。

表 5-23 SQLSetConnectOption 関数で取得できるオプション値とサポート状況

fOption	値	vParam	値	サポート状況	指定時の処置	備考
SQL_ACCESS_MODE	101	SQL_MODE_READ_WRITE	0UL	○	正常処理	ODBC 1.0
		SQL_MODE_READ_ONLY	1UL	○	正常処理	
SQL_AUTOCOMMIT	102	SQL_AUTOCOMMIT_ON	1UL	○	正常処理	ODBC 1.0
		SQL_AUTOCOMMIT_OFF	0UL	○	正常処理	
SQL_CURRENT_QUALIFIER	109	データソースが使う修飾語名も含む NULL 終端文字列	—	×	S1C00 エラー	ODBC 2.0
SQL_LOGIN_TIMEOUT	103	秒数に対応する 32 ビット整数値	—	×	S1C00 エラー	ODBC 1.0
SQL_ODBC_CURSORS	110	SQL_CUR_USE_DRIVER	2UL	×	S1C00 エラー	ODBC 2.0
		SQL_CUR_USE_ODBC	1UL	×	S1C00 エラー	
		SQL_CUR_USE_IF_NEEDED	0UL	×	S1C00 エラー	
SQL_OPT_TRACE	104	SQL_OPT_TRACE_OFF	0UL	○	正常処理	ODBC 1.0
		SQL_OPT_TRACE_ON	1UL	×	S1009 エラー	

fOption	値	vParam	値	サポート状況	指定時の処置	備考
SQL_OPT_TRACEFILE	105	トレースファイル名を含む NULL で 終端文字列	—	○	正常処理	ODBC 1.0
SQL_PACKET_SIZE	112	バイト単位のネットワークパケット サイズを示す 32 ビット整数値	—	×	S1C00 エ ラー	ODBC 2.0
SQL_QUIET_MODE	111	32 ビットウィンドウハンドル (hwnd)	—	×	S1C00 エ ラー	ODBC 2.0
SQL_TRANSLATE_DLL	106	DLL 名を含む NULL 終端文字列	—	×	S1C00 エ ラー	ODBC 1.0
SQL_TRANSLATE_OPTION	107	変換 DLL に渡される 32 ビットのフ ラグの値	—	×	S1C00 エ ラー	ODBC 1.0
SQL_TXN_ISOLATION	108	現在の hdbc に関するトランザクショ ン分離レベルを設定する 32 ビットの ビットマスク	—	○	正常処理	ODBC 1.0

(凡例)

- : DABroker ドライバで使える ODBC 関数です。
- × : DABroker ドライバで使えない ODBC 関数です。

表 5-24 SQLSetStmtOption 関数で取得できるオプション値とサポート状況

fOption	値	vParam	値	サポート状況	指定時の処置	備考
SQL_ASYNC_ENABLE	4	SQL_ASYNC_ENABLE_OFF	0UL	○	正常処理	ODBC 1.0
		SQL_ASYNC_ENABLE_ON	1UL	○	正常処理	
SQL_BIND_TYPE	5	バインド方向を設定する 32 ビット整 数値	—	×	S1C00 エ ラー	ODBC 1.0
SQL_CONCURRENCY	7	SQL_CONCUR_READ_ONLY	1	×	S1C00 エ ラー	ODBC 2.0
		SQL_CONCUR_LOCK	2	×	S1C00 エ ラー	
		SQL_CONCUR_ROWVER	3	×	S1C00 エ ラー	
		SQL_CONCUR_VALUES	4	×	S1C00 エ ラー	
SQL_CURSOR_TYPE	6	SQL_CURSOR_FORWARD_ONL Y	0UL	×	S1C00 エ ラー	ODBC 2.0
		SQL_CURSOR_STATIC	3UL	×	S1C00 エ ラー	

fOption	値	vParam	値	サポート 状況	指定時の処置	備考
		SQL_CURSOR_KEYSET_DRIVE N	1UL	×	S1C00 エ ラー	
		SQL_CURSOR_DYNAMIC	2UL	×	S1C00 エ ラー	
SQL_KEYSET_SI ZE	8	キーセット駆動カーソルに対するキー セットの行数を指定する 32 ビット整 数値	—	×	S1C00 エ ラー	ODBC 2.0
SQL_MAX_LEN GTH	3	ドライバが文字又はバイナリ列から 返すデータの最大量を指定する 32 ビット整数値	—	○	正常処理	ODBC 1.0
SQL_MAX_ROW S	1	アプリケーションに返す SELECT ス テートメントの最大行数に相当する 32 ビット整数値	—	○	正常処理	ODBC 1.0
SQL_NOSCAN	2	SQL_NOSCAN_OFF	0UL	○	正常処理	ODBC 1.0
		SQL_NOSCAN_ON	1UL	○	正常処理	
SQL_QUERY_TI MEOUT	0	SQL ステートメントがアプリケー ションに実行して戻るまでの待ち時 間を示す秒数に相当する 32 ビット整 数値	—	×	S1C00 エ ラー	ODBC 1.0
SQL_RETRIEVE _DATA	11	SQL_RD_ON	1UL	×	S1C00 エ ラー	ODBC 2.0
		SQL_RD_OFF	0UL	×	S1C00 エ ラー	
SQL_ROWSET_ SIZE	9	行セットの行数を指定する 32 ビット 整数値	—	×	S1C00 エ ラー	ODBC 2.0
SQL_SIMULATE _CURSOR	10	SQL_SC_NON_UNIQUE	0UL	×	S1C00 エ ラー	ODBC 2.0
		SQL_SC_TRY_UNIQUE	1UL	×	S1C00 エ ラー	
		SQL_SC_UNIQUE	2UL	×	S1C00 エ ラー	
SQL_USE_BOO KMARKS	12	SQL_UB_OFF	0UL	×	S1C00 エ ラー	ODBC 2.0
		SQL_UB_ON	1UL	×	S1C00 エ ラー	

(凡例)

- : DABroker ドライバで使える ODBC 関数です。
- × : DABroker ドライバで使えない ODBC 関数です。

(2) SQLGetConnectOption 関数, SQLGetStmtOption 関数

SQLGetConnectOption 関数で取得できるオプション値とサポート状況を表 5-25 に、SQLGetStmtOption 関数で取得できるオプション値とサポート状況を表 5-26 に示します。表の備考は、ODBC 関数がサポートされた ODBC のバージョンを示します。

表 5-25 SQLGetConnectOption 関数で取得できるオプション値とサポート状況

fOption	値	サポート状況	指定時の処置	備考
SQL_ACCESS_MODE	101	○	正常処理	ODBC 1.0
SQL_AUTOCOMMIT	102	○	正常処理	ODBC 1.0
SQL_CURRENT_QUALIFIER	109	×	S1C00 エラー	ODBC 2.0
SQL_LOGIN_TIMEOUT	103	×	S1C00 エラー	ODBC 1.0
SQL_ODBC_CURSORS	110	×	S1C00 エラー	ODBC 2.0
SQL_OPT_TRACE	104	○	正常処理	ODBC 1.0
SQL_OPT_TRACEFILE	105	○	正常処理	ODBC 1.0
SQL_PACKET_SIZE	112	×	S1C00 エラー	ODBC 2.0
SQL_QUIET_MODE	111	×	S1C00 エラー	ODBC 2.0
SQL_TRANSLATE_DLL	106	×	S1C00 エラー	ODBC 1.0
SQL_TRANSLATE_OPTION	107	×	S1C00 エラー	ODBC 1.0
SQL_TXN_ISOLATION	108	○	正常処理	ODBC 1.0

(凡例)

- : DABroker ドライバで使える ODBC 関数です。
- × : DABroker ドライバで使えない ODBC 関数です。

表 5-26 SQLGetStmtOption 関数で取得できるオプション値とサポート状況

fOption	値	サポート状況	指定時の処置	備考
SQL_ASYNC_ENABLE	4	○	正常処理	ODBC 1.0
SQL_BIND_TYPE	5	×	S1C00 エラー	ODBC 1.0
SQL_CONCURRENCY	7	×	S1C00 エラー	ODBC 2.0
SQL_CURSOR_TYPE	6	×	S1C00 エラー	ODBC 2.0
SQL_KEYSET_SIZE	8	×	S1C00 エラー	ODBC 2.0
SQL_MAX_LENGTH	3	○	正常処理	ODBC 1.0
SQL_MAX_ROWS	1	○	正常処理	ODBC 1.0
SQL_NOSCAN	2	○	正常処理	ODBC 1.0
SQL_QUERY_TIMEOUT	0	×	S1C00 エラー	ODBC 1.0

fOption	値	サポート状況	指定時の処置	備考
SQL_RETRIEVE_DATA	11	×	S1C00 エラー	ODBC 2.0
SQL_ROWSET_SIZE	9	×	S1C00 エラー	ODBC 2.0
SQL_SIMULATE_CURSOR	10	×	S1C00 エラー	ODBC 2.0
SQL_USE_BOOKMARKS	12	×	S1C00 エラー	ODBC 2.0

(凡例)

- : DABroker ドライバで使える ODBC 関数です。
- × : DABroker ドライバで使えない ODBC 関数です。

5.6 ODBC 関数を実行したときのエラーメッセージ

DBPARTNER ODBC Driver から返されるエラーメッセージについて説明します。エラーメッセージについては、「10.2.2 DBPARTNER ODBC Driver のメッセージ」を参照してください。

5.6.1 エラーメッセージに付く ODBC 関数のプレフィックス

データソースのエラーのときは、DBPARTNER ODBC Driver はサーバから返ってきたエラーメッセージを表示します。

ODBC ドライバが検出したエラーのときは、SQLSTATE に関連付けてあるテキストを表示します。

ODBC 関数のエラーメッセージの形式を次に示します。

```
[vender] [ODBC-component] [data-source] error-message
```

[] 内の値は、エラー元を識別できるように表示されます。

DBPARTNER ODBC Driver が返すプレフィックスの一覧を表 5-27 に示します。

表 5-27 DBPARTNER ODBC Driver が返すプレフィックスの一覧

ODBC ドライバの種別	エラー元	プレフィックスに設定される値		
		[vender]	[ODBC-component]	[data-source]
DABroker ドライバ	DABroker ドライバ	[HITACHI]	[DBPARTNER]	—
	DABroker	[HITACHI]	[DBPARTNER]	[DABroker]
	Database Connection Server	[HITACHI]	[DBPARTNER]	[DBS]
	データベース ※	[HITACHI]	[DBPARTNER]	[HiRDB] 又は [ORACLE]
	エラーになったサーバ側のプログラムが不明	[HITACHI]	[DBPARTNER]	[SERVER_UNKNOWN]

注※

DBPARTNER ODBC Driver for DABroker を使って RDA Link for Gateway 経由又は VOS3 Database Connection Server 経由でアクセスできるメインフレーム系データベースでのエラーのとき、及び DABroker の ODBC3.0 経由で接続する Microsoft SQL Server 又は Sybase Adaptive Server Anywhere でのエラーのときは、[data-source] にはデータベース固有の値でなく、各サーバプログラムの値（[DABroker]、[DBS]）が表示されます。

5.6.2 HiRDBのエラーメッセージの見方

ODBC 関数を実行して fOption に「SQL_SUCCESS_WITH_INFO」が返されて、メッセージに「[HITACHI] [DBPARTNER] [HiRDB]」が返されたときは、エラー要因は次の手順に従って見てください。

1. SQLError の pfNativeError に返される 10 進数のコードを 2 進数 16 けたに変換します。
2. 変換した 2 進数は、HiRDB の SQL 領域中の「SQL 連結領域の内容」の SQLWARNO~SQLWARNF に対応します。この内容を基に対処してください。

SQL 連絡領域の SQLWARNO~SQLWARNF の内容については、マニュアル「HiRDB SQL リファレンス」を参照してください。

5.7 ODBC 関数の非同期実行

ODBC 関数を実行するときは、通常 ODBC ドライバは ODBC 関数を同期実行します。このとき ODBC ドライバは、関数の呼び出しが終了するまで、制御をアプリケーションに戻しません。

アプリケーションの処理の効率を上げるため、一部の ODBC 関数は、処理の終了を待たないでアプリケーションに制御を戻せます。このような ODBC 関数の実行方法を**非同期実行**といいます。

ODBC の規約で非同期実行できる関数と DBPARTNER ODBC Driver で非同期実行できるかどうかを表 5-28 に示します。

表 5-28 ODBC の規約で非同期実行できる関数と DBPARTNER ODBC Driver での非同期実行の可否

ODBC の規約で非同期実行できる関数名	DBPARTNER ODBC Driver で非同期実行できるかどうか
SQLColAttributes	×
SQLColumnPrivileges	○
SQLColumns	○
SQLDescribeCol	×
SQLDescribeParam	×
SQLExecDirect	○
SQLExecute	○
SQLFetch	○
SQLForeignKeys	×
SQLGetData	×
SQLGetTypeInfo	×
SQLMoreResults	×
SQLNumParams	×
SQLNumResultCols	×
SQLParamData	○
SQLPrepare	○
SQLPrimaryKeys	○
SQLPutData	×
SQLSpecialColumns	×
SQLStatistics	○
SQLTablePrivileges	○

ODBC の規約で非同期実行できる関数名	DBPARTNER ODBC Driver で非同期実行できるかどうか
SQLTables	○

(凡例)

- : DBPARTNER ODBC Driver で非同期実行できます。
- × : DBPARTNER ODBC Driver で非同期実行できません。

非同期実行のキャンセルと環境定義の非同期キャンセル機能との関係 (DBPARTNER/Client32 で ODBC ドライバを使うときの注意)

[DBPARTNER ODBC Driver for DABroker—環境定義—] ダイアログボックスの [非同期キャンセル機能を使用する] チェックボックスの指定は、DBPARTNER/Client32 を使うときのオプションです。これは、ODBC 規約の正式な非同期キャンセル機能ではありません。したがって、このチェックボックスの指定は、ODBC 関数を非同期実行するときは無効になります。

5.8 ストアドプロシジャの実行

結果セットを返すストアドプロシジャは実行できます。ただし、結果セットは取得できません。

6

アプリケーションを使ったデータベースアクセス (CommuniNet 接続)

この章では、DBPARTNER2 Client のアプリケーションから ActiveX コントロールのオブジェクトを使ってデータベースにアクセスする方法 (CommuniNet 接続) について説明します。

6.1 DBPARTNER2 Client が提供する ActiveX コントロールの概要

DBPARTNER2 Client のアプリケーションを作成するときは、DBPARTNER2 Client が提供する ActiveX コントロールのオブジェクトを使います。オブジェクトを使って、データベースからデータを抽出したり、更新したりできます。

6.1.1 ActiveX コントロールのオブジェクトの階層

ActiveX コントロールのオブジェクトの階層を図 6-1 に、オブジェクトの説明を表 6-1 に示します。複数形のオブジェクト名 (Tables, Columns など) は、下位のオブジェクトを管理するコレクションオブジェクトです。

図 6-1 DBPARTNER2 Client が提供する ActiveX コントロールのオブジェクトの階層

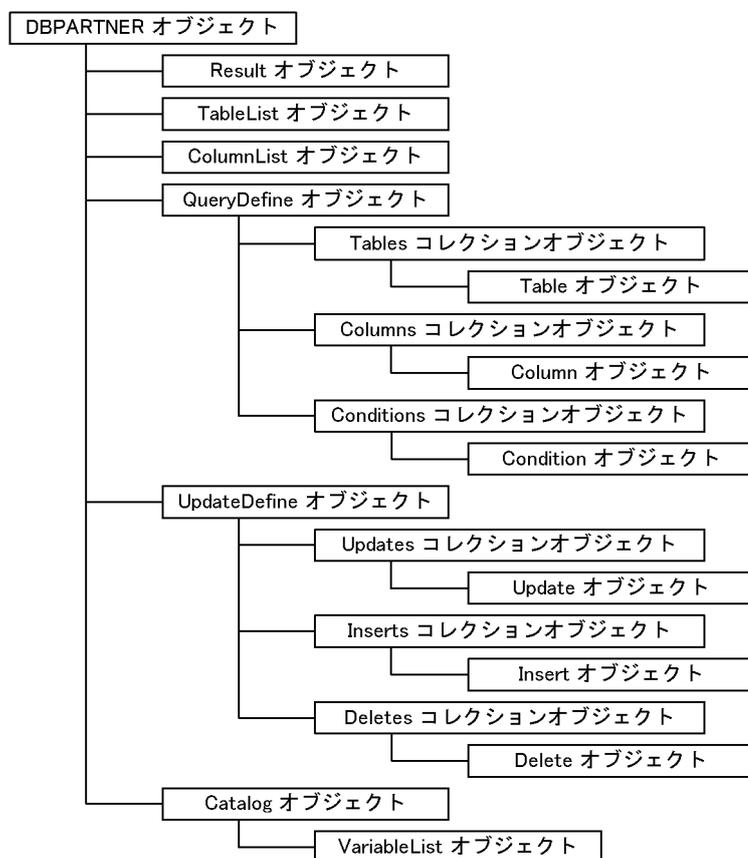


表 6-1 DBPARTNER2 Client が提供する ActiveX コントロールのオブジェクトの説明

オブジェクト	説明
DBPARTNER	サーバプログラムとの接続を管理するオブジェクトです。
Result	抽出データを管理するオブジェクトです。
TableList	表情報を管理するオブジェクトです。
ColumnList	列情報を管理するオブジェクトです。

オブジェクト	説明
QueryDefine	抽出処理を実行するオブジェクトです。
Tables	Table オブジェクトを管理するコレクションオブジェクトです。
Table	抽出表を管理するオブジェクトです。
Columns	Column オブジェクトを管理するコレクションオブジェクトです。
Column	抽出列を管理するオブジェクトです。
Conditions	Condition オブジェクトを管理するコレクションオブジェクトです。
Condition	抽出条件を管理するオブジェクトです。
UpdateDefine	更新処理を実行するオブジェクトです。
Updates	Update オブジェクトを管理するコレクションオブジェクトです。
Update	変更行条件, 変更情報を管理するオブジェクトです。
Inserts	Insert オブジェクトを管理するコレクションオブジェクトです。
Insert	追加情報を管理するオブジェクトです。
Deletes	Delete オブジェクトを管理するコレクションオブジェクトです。
Delete	削除行条件を管理するオブジェクトです。
Catalog	手順ファイルを管理するオブジェクトです。
VariableList	手順ファイルの可変値を管理するオブジェクトです。

注意事項

DBPARTNER2 Client の ActiveX コントロールのオブジェクトには、使い方に制限があります。制限については、「[6.10 避けて欲しい処理](#)」を参照してください。

6.1.2 プロパティとメソッドの表記

ActiveX コントロールのオブジェクトに属するプロパティとメソッドは、「オブジェクト名. プロパティ又はメソッド名」で示します。

(例) DBPARTNER オブジェクトの Connect メソッドの表記

DBPARTNER.Connect メソッド

ビジュアル開発ツールでは、オブジェクト名を「.」（ピリオド）で表記できます。そのため、このマニュアルでプロパティ又はメソッド名にオブジェクト名を付けて表記するときは、オブジェクト名を斜体文字にしています。

このマニュアルでは、Visual Basic でアプリケーションを開発することを想定して説明します。ほかのビジュアル開発ツールを使ってアプリケーションを作成するときは、マニュアルの記述を読み替えてください。

6.1.3 アプリケーションのサンプル

DBPARTNER2 Client では、次に示すフォルダ下にサンプルを提供しています。

- DBPARTNER2 Client のフォルダ¥Ocx¥Samples

サンプルでは、テキストボックスやリストボックスも実装しています。DBPARTNER2 Client の詳しいアプリケーションの処理については、サンプルを参照してください。

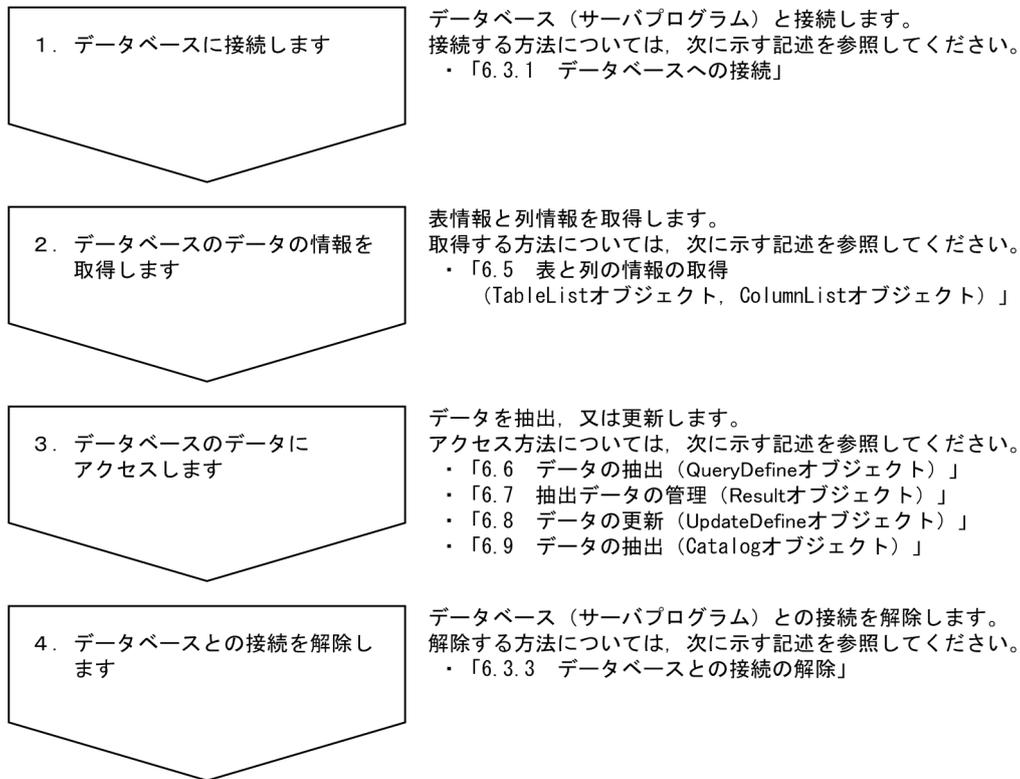
6.1.4 アプリケーションの作成

ActiveX コントロールを使用したアプリケーションを開発するためのプログラムとして、AP 作成ウィザードを提供しています。AP 作成ウィザードを使用すると、データベースからデータを抽出するアプリケーションが簡単に作成できます。AP 作成ウィザードについては、「付録 A AP 作成ウィザード」を参照してください。

6.2 データベースアクセスの手順

DBPARTNER2 Client が提供する ActiveX コントロールのオブジェクトを使って、データベースにアクセスする方法について説明します。DBPARTNER2 Client からデータベースにアクセスする手順の概要を図 6-2 に示します。

図 6-2 DBPARTNER2 Client からデータベースにアクセスする手順の概要



データベースにアクセスするときは、**トランザクション**と**排他制御**の知識が必要です。トランザクションと排他制御については、「6.4 トランザクションと排他制御 (DBPARTNER オブジェクト)」を参照してください。

ActiveX コントロールのオブジェクトの文法については、「7. DBPARTNER2 Client のオブジェクトの文法 (CommuniNet 接続)」を参照してください。

サンプルの参照先

DBPARTNER オブジェクトは、抽出／更新で使います。サンプルについては DBPARTNER2 Client のフォルダ¥Ocx¥Samples¥Vb¥Communinet フォルダを参照してください。

6.2.1 DBPARTNER オブジェクトのプロパティの設定

DBPARTNER オブジェクトは、DBPARTNER2 Client のトップオブジェクトです。データベースに接続する前に、DBPARTNER オブジェクトのプロパティを設定してください。

(1) DBPARTNER オブジェクトのプロパティに設定する内容

DBPARTNER オブジェクトのプロパティを次に示します。

(a) サーバプログラムとの接続に関する設定

- *DBPARTNER.CommitMode* プロパティ
トランザクションの終了モードを設定します。
- *DBPARTNER.CodeChangeMode* プロパティ
コード変換をクライアント側で実行するかサーバ側で実行するかを設定します。

(b) CommuniNet の設定

- *DBPARTNER.TerminalName* プロパティ
CommuniNet の端末名を設定します。
- *DBPARTNER.TerminalMode* プロパティ
CommuniNet の起動モード（クライアント PC のデスクトップに表示される状態）を設定します。
- *DBPARTNER.LogonFileName* プロパティ, *DBPARTNER.LogoffFileName* プロパティ
CommuniNet 接続で使うログオンマクロスクリプトファイル名/ログオフマクロスクリプトファイル名を設定します。
- *DBPARTNER.MacroTimeLimit* プロパティ
CommuniNet 接続で使うマクロスクリプトの実行待ち時間を設定します。

(c) キャンセルの時間監視の設定

- *DBPARTNER.ResponseLimit* プロパティ
キャンセル応答待ち時間（Cancel メソッドの実行待ち時間）を設定します。

(d) サーバに関する設定

- *DBPARTNER.TargetServer* プロパティ
アクセスするサーバプログラムの製品名を設定します。
- *DBPARTNER.TargetDatabase* プロパティ
アクセスするデータベースの製品名を設定します。

(e) 利用者に関する設定

- *DBPARTNER.UserID* プロパティ
ユーザ ID を設定します。
- *DBPARTNER.Password* プロパティ
パスワードを設定します。

6.3 データベースへの接続と接続の解除 (DBPARTNER オブジェクト)

DBPARTNER2 Client とデータベースを接続する方法と接続を解除する方法について説明します。DBPARTNER2 Client とデータベースを接続するときは、サーバプログラム (VOS3 HOAPDBS E2, VOS3 ACE3 E3, VOSK HOAPDBS/PC, VOS1 DBPARTNER/Link) が必要です。DBPARTNER2 Client がサーバプログラムに接続すると、データベースに接続されたことになります。

サンプルの参照先

DBPARTNER オブジェクトのサンプルは、DBPARTNER2 Client のフォルダ `¥Ocx¥Samples¥Vb ¥Communinet` フォルダを参照してください。

6.3.1 データベースへの接続

DBPARTNER2 Client からデータベースに接続するときは、*DBPARTNER.Connect* メソッドを使います。*DBPARTNER.Connect* メソッドは、CommuniNet に関するプロパティの設定、ユーザ ID (*DBPARTNER.UserID* プロパティ)、パスワード (*DBPARTNER.Password* プロパティ) を参照して、データベースに接続します。

それぞれのプロパティを変更すると、接続するデータベースを変更できます。

データベースと接続するときにデータベース又はサーバプログラムが起動されていないときは、エラーメッセージが表示されます。

(1) データベースとの接続の確認

DBPARTNER2 Client とサーバプログラムが接続されているかどうかは、*DBPARTNER.IsConnect* プロパティを参照して確認できます。

6.3.2 データベースへのアクセスのキャンセル

DBPARTNER2 Client からデータベースへのアクセス処理をキャンセルするときは、*DBPARTNER.Cancel* メソッドを使います。

DBPARTNER.Cancel メソッドを実行できる条件とキャンセル後の格納データについては、「7. DBPARTNER2 Client のオブジェクトの文法 (CommuniNet 接続)」の *DBPARTNER.Cancel* メソッドの説明を参照してください。

6.3.3 データベースとの接続の解除

DBPARTNER2 Client とデータベースの接続を解除するときは、次に示すどちらかのメソッドを実行します。

- *DBPARTNER.TerminalClose* メソッド
サーバプログラムとの接続を終了して、CommuniNet を終了させます。
- *DBPARTNER.Disconnect* メソッド
サーバプログラムとの接続だけを終了します。CommuniNet は終了させません。

通常は *DBPARTNER.TerminalClose* メソッドを使います。業務の理由で CommuniNet を終了させないときは、*DBPARTNER.Disconnect* メソッドを使います。

DBPARTNER.TerminalClose メソッド又は *DBPARTNER.Disconnect* メソッドを使った後は、接続していたデータベースへの処理はすべてエラーになります。

抽出処理を実行している間に接続を終了しようとする、メソッドがエラーになります。また、*DBPARTNER.Connect* メソッドで接続していないデータベースに、接続を終了しようとするメソッドを実行しても無視されます（エラーになりません）。

6.4 トランザクションと排他制御 (DBPARTNER オブジェクト)

アプリケーションでトランザクションを制御するときの手順について説明します。

6.4.1 トランザクションの制御の設定

トランザクションを DBPARTNER2 Client で自動的に制御するか、アプリケーションからトランザクションを制御するかは、*DBPARTNER.CommitMode* プロパティに設定します。

(1) DBPARTNER2 Client で自動的にトランザクションを制御するとき

DBPARTNER2 Client で自動的にトランザクションを制御するときは、*DBPARTNER.CommitMode* プロパティに *dbpAuto* を設定します。*dbpAuto* を設定すると、データベースアクセスのトランザクションを DBPARTNER2 Client のアプリケーションから意識する必要はありません。

(2) アプリケーションからトランザクションを制御するとき

アプリケーションからメソッドを使ってトランザクションを制御するときは、*DBPARTNER.CommitMode* プロパティに *dbpManual* を設定してから、*DBPARTNER.Commit* メソッド又は *DBPARTNER.Rollback* メソッドを使います。

DBPARTNER2 Client は、データベースのデータを更新する *Execute* メソッドを実行してから *DBPARTNER.Commit* メソッド又は *DBPARTNER.Rollback* メソッドの実行までを、1 件のトランザクションとして扱います。*DBPARTNER.Commit* 又は *DBPARTNER.Rollback* が成功すると、その時点から次のトランザクションが開始されます。データベースにアクセスし直す必要はありません。

トランザクションを終了させないでオブジェクトを削除、又はデータベースとの接続を切断すると、DBPARTNER2 Client がアプリケーションの処理が異常と判断して *DBPARTNER.Rollback* メソッドの処理を自動的に実行します。

6.4.2 排他制御

DBPARTNER2 Client のアプリケーションからデータを抽出／更新するときには、ほかのアプリケーションからのアクセスを許すかどうかを設定します。これを排他制御といいます。

(1) 排他制御の設定

排他制御は、データにアクセスするプロパティ (*QueryDefine.Exclusive* プロパティ又は *UpdateDefine.Exclusive* プロパティ) に文字列で設定します。アクセスするデータベースによって、設定する内容が異なります。設定する内容については、アクセスするデータベースのマニュアルを参照してください。

(2) 排他エラーになったときの動作の設定

データにアクセスしようとした場合に、ほかのアプリケーションがデータを確保しているときは、データが解放されるまで待つか、エラーにするかを選べます。どちらにするかは、データにアクセスするプロパティ (*QueryDefine.Exclusive* プロパティ又は *UpdateDefine.Exclusive* プロパティ) に設定します。

データが解放されるまで待つ設定をすると、複数のアプリケーションが互いが確保しているデータの解放を待ち続ける状態 (デッドロック) になるおそれがあります。デッドロックにならないように、複数のアプリケーション間でデータを確保する順番を決めておいてください。さらに、複数のアプリケーションからデータベースにアクセスするときは、確保するデータの範囲を最小限に抑えてください。

エラーにする設定をするときは、データベースアクセスごとにエラーの対処 (一定時間を空けてリトライをするなど) が必要になります。

6.5 表と列の情報の取得 (TableList オブジェクト, ColumnList オブジェクト)

データを抽出する前に表と列の情報を取得して、どのように抽出するかを決めます。情報には、表情報と列情報があります。

6.5.1 表情報の取得

表情報を取得するときは、*DBPARTNER.Connect* メソッドでサーバプログラムと接続した後、*TableList* オブジェクトを使います。

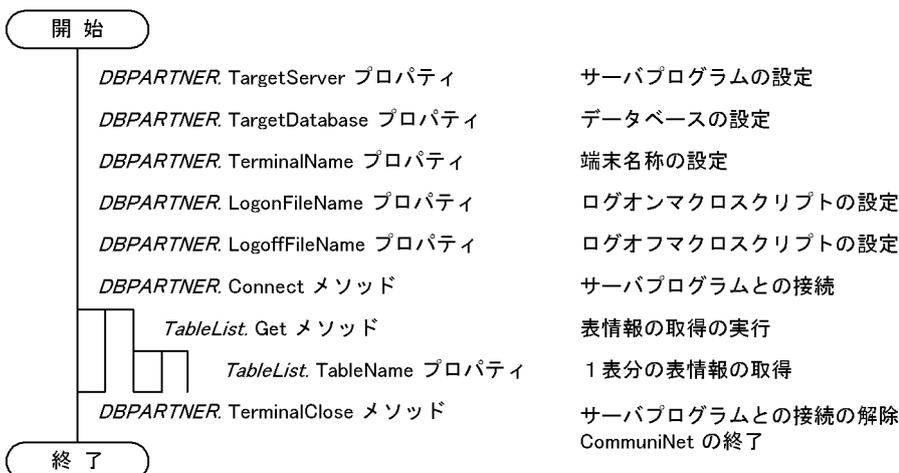
表情報を取得するときのオブジェクトの流れとプロパティとメソッドの実行手順を図 6-3 に示します。

図 6-3 表情報を取得するときのオブジェクトの流れとプロパティとメソッドの実行手順

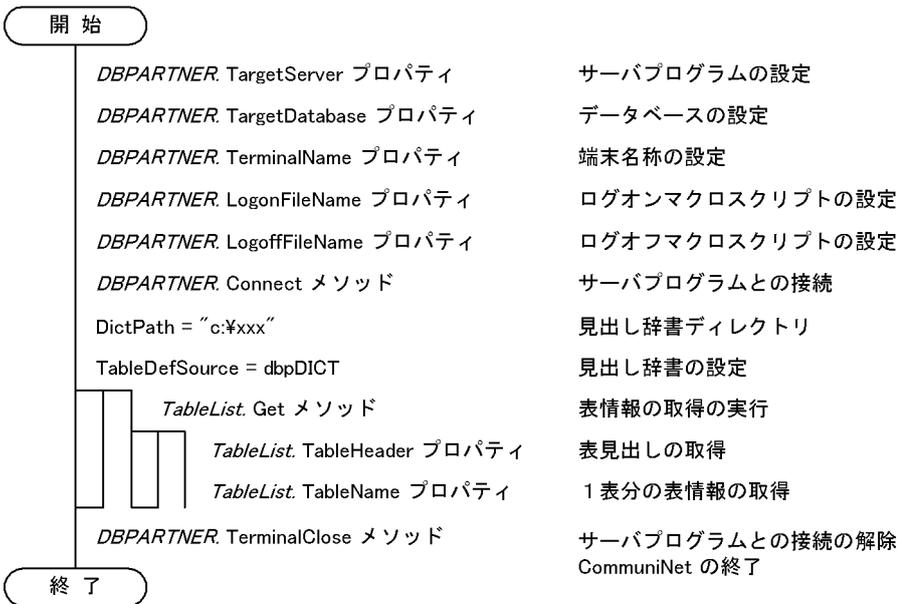
●オブジェクトの流れ



●プロパティとメソッドの流れ (見出し辞書を使用しない場合)



●プロパティとメソッドの流れ（見出し辞書を使用する場合）



サンプルの参照先

ColumnList オブジェクトのサンプルは、DBPARTNER2 Client のフォルダ¥Ocx¥Samples¥Vb ¥CommuniNet¥Query フォルダを参照してください。

(1) 表情情報を取得する手順

表情情報を取得する手順を次に示します。

見出し辞書を使用しない場合

- 次に示す TableList オブジェクトのプロパティを設定します。
 - TableList.Pattern プロパティ：絞り込み条件を設定します。
 - TableList.CountLimit プロパティ：取得する表情情報の個数を設定します。
- TableList.Get メソッドで、表情情報の取得を実行します。TableList.Get メソッドを実行すると、次に示すプロパティの値を参照できます。
 - TableList.Count プロパティ：取得した表情情報の個数。
 - TableList.TableName プロパティ：取得した表の名前。
 - TableList.TableType プロパティ：取得した表の属性。
 - TableList.TableOwner プロパティ：取得した表の所有者。
 - TableList.TableComment プロパティ：取得した表の注釈。
 - TableList.TableGroupName プロパティ：取得した表のグループ名。
 - TableList.TableHeader プロパティ：取得した表のラベル。
 - TableList.TableCreateDate プロパティ：取得した表の作成日付。

- *TableList.TableDBMName* プロパティ：取得した表の DBM 名。
- *TableList.TablePasswordMode* プロパティ：取得した表に設定してある表パスワードの種別。
- *TableList.TableSelectAuth* プロパティ, *TableList.TableUpdateAuth* プロパティ, *TableList.TableInsertAuth* プロパティ, *TableList.TableDeleteAuth* プロパティ：取得した表への抽出／変更／更新／削除の権限があるかどうか。

上記のプロパティから、必要な情報を参照してください。

見出し辞書を使用する場合

1. 次に示すオブジェクトのプロパティを設定します。

- *DBPARTNER.DictPath* プロパティ：使用する見出し辞書のパスを設定します。
- *DBPARTNER.TableDefSource* プロパティ：表情報の取得元を設定します。
- *TableList.SelectDict* プロパティ：使用する見出し辞書を限定します。
- *TableList.Pattern* プロパティ：絞り込み条件を設定します。
- *TableList.CountLimit* プロパティ：取得する表情報の個数を設定します。

2. *TableList.Get* メソッドで、表情報の取得を実行します。*TableList.Get* メソッドを実行すると、次に示すプロパティの値を参照できます。

- *TableList.Count* プロパティ：取得した表情報の個数。
- *TableList.TableName* プロパティ：取得した表の名前。
- *TableList.TableType* プロパティ：取得した表の属性。
- *TableList.TableOwner* プロパティ：取得した表の所有者。
- *TableList.TableComment* プロパティ：取得した表の注釈。
- *TableList.TableGroupName* プロパティ：取得した表のグループ名。
- *TableList.TableHeader* プロパティ：取得した表のラベル。
- *TableList.TableCreateDate* プロパティ：取得した表の作成日付。
- *TableList.TableDBMName* プロパティ：取得した表の DBM 名。
- *TableList.TablePasswordMode* プロパティ：取得した表に設定してある表パスワードの種別。
- *TableList.TableSelectAuth* プロパティ, *TableList.TableUpdateAuth* プロパティ, *TableList.TableInsertAuth* プロパティ, *TableList.TableDeleteAuth* プロパティ：取得した表への抽出／変更／更新／削除の権限があるかどうか。

上記のプロパティから、必要な情報を参照してください。

(2) 注意

TableList.Pattern プロパティと *TableList.CountLimit* プロパティは、*TableList.Get* メソッドを実行する前に設定してください。*TableList.Get* メソッドを実行した後に変更しても、無視されます。

6.5.2 列情報の取得

列情報を取得するときは、*DBPARTNER.Connect* メソッドでサーバプログラムと接続した後、*ColumnList* オブジェクトを使います。

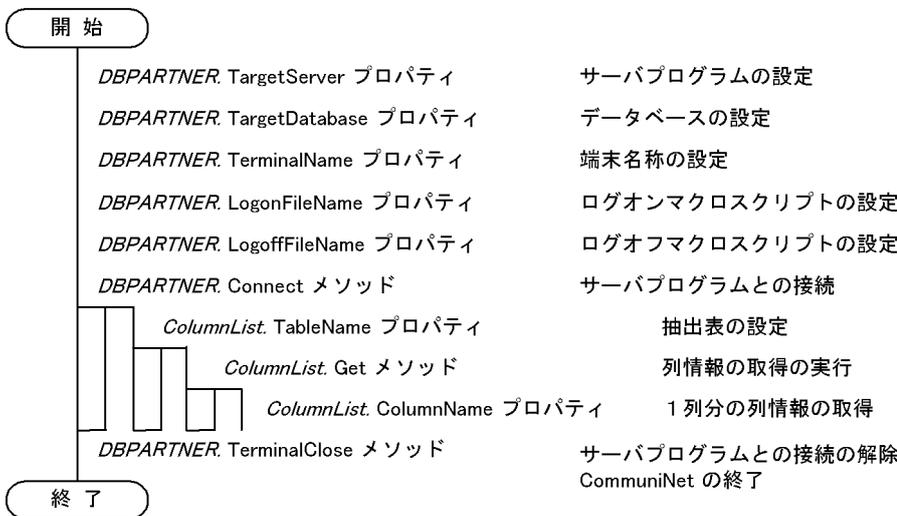
表情報を取得するときのオブジェクトの流れとプロパティとメソッドの実行手順を図 6-4 に示します。

図 6-4 列情報を取得するときのオブジェクトの流れとプロパティとメソッドの実行手順

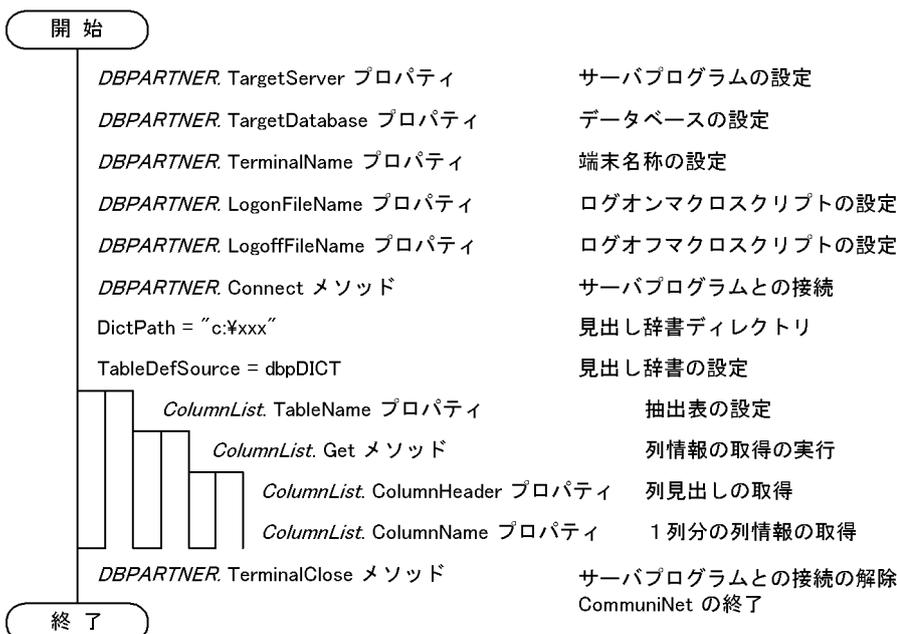
●オブジェクトの流れ



●プロパティとメソッドの流れ（見出し辞書を使用しない場合）



●プロパティとメソッドの流れ（見出し辞書を使用する場合）



サンプルの参照先

ColumnList オブジェクトのサンプルは、DBPARTNER2 Client のフォルダ¥Ocx¥Samples¥Vb ¥Communitnet¥Query フォルダを参照してください。

(1) 列情報を取得する手順

列情報を取得する手順を次に示します。

見出し辞書を使用しない場合

1. 次に示す ColumnList オブジェクトのプロパティを設定します。

- *ColumnList.TableName* プロパティ：列情報を取得する表の名前を設定します。
- *ColumnList.CountLimit* プロパティ：取得する列情報の個数を設定します。
- *ColumnList.TablePassword* プロパティ：取得する列情報の個数を設定します。

2. *ColumnList.Get* メソッドで、列情報の取得を実行します。*ColumnList.Get* メソッドを実行すると、次に示すプロパティの値を参照できます。

- *ColumnList.Count* プロパティ：取得した列の個数。
- *ColumnList.ColumnName* プロパティ：取得した列の名前。
- *ColumnList.ColumnType* プロパティ：取得した列の属性。
- *ColumnList.ColumnLength* プロパティ：取得した列の定義長。
- *ColumnList.ColumnScale* プロパティ：取得した列の小数点以下のけた数。
- *ColumnList.ColumnComment* プロパティ：取得した列の注釈。
- *ColumnList.ColumnHeader* プロパティ：取得した列のラベル。
- *ColumnList.ColumnNumber* プロパティ：取得した列の列番号。
- *ColumnList.ColumnEditPattern* プロパティ：取得した列の編集形式。

上記のプロパティから、必要な情報を参照してください。

見出し辞書を使用する場合

1. 次に示すオブジェクトのプロパティを設定します。

- *DBPARTNER.DictPath* プロパティ：使用する見出し辞書のパスを設定します。
- *DBPARTNER.TableDefSource* プロパティ：列情報の取得元を設定します。
- *TableList.SelectDict* プロパティ：使用する見出し辞書を限定します。
- *ColumnList.TableName* プロパティ：列情報を取得する表の名前を設定します。
- *ColumnList.CountLimit* プロパティ：取得する列情報の個数を設定します。
- *ColumnList.TablePassword* プロパティ：取得する列情報の個数を設定します。

2. *ColumnList.Get* メソッドで、列情報の取得を実行します。*ColumnList.Get* メソッドを実行すると、次に示すプロパティの値を参照できます。

- *ColumnList.Count* プロパティ：取得した列の個数。
- *ColumnList.ColumnName* プロパティ：取得した列の名前。
- *ColumnList.ColumnType* プロパティ：取得した列の属性。
- *ColumnList.ColumnLength* プロパティ：取得した列の定義長。
- *ColumnList.ColumnScale* プロパティ：取得した列の小数点以下のけた数。
- *ColumnList.ColumnComment* プロパティ：取得した列の注釈。
- *ColumnList.ColumnHeader* プロパティ：取得した列のラベル。
- *ColumnList.ColumnNumber* プロパティ：取得した列の列番号。
- *ColumnList.ColumnEditPattern* プロパティ：取得した列の編集形式。

上記のプロパティから、必要な情報を参照してください。

(2) 注意

ColumnList.TableName プロパティと *ColumnList.CountLimit* プロパティは、*ColumnList.Get* メソッドを実行する前に設定してください。*ColumnList.Get* メソッドを実行した後に変更しても、無視されます。

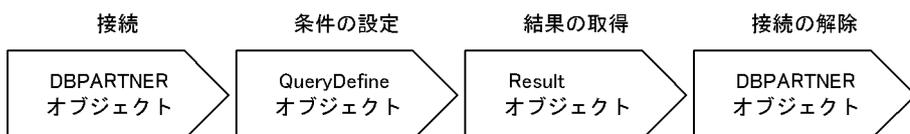
6.6 データの抽出 (QueryDefine オブジェクト)

DBPARTNER2 Client のオブジェクト QueryDefine を使って、データを抽出する方法について説明します。データを抽出するときは、DBPARTNER.Connect メソッドでサーバプログラムと接続した後、QueryDefine オブジェクトを使います。

QueryDefine オブジェクトを使った抽出のオブジェクトの流れとプロパティとメソッドの実行手順を図 6-5 に示します。

図 6-5 QueryDefine オブジェクトを使った抽出のオブジェクトの流れとプロパティとメソッドの実行手順

●オブジェクトの流れ



●プロパティとメソッドの流れ



サンプルの参照先

QueryDefine オブジェクトのサンプルは、DBPARTNER2 Client のフォルダ¥Ocx¥Samples¥Vb ¥CommuniNet¥Query フォルダを参照してください。

6.6.1 QueryDefine オブジェクトを使ってデータを抽出する手順

QueryDefine オブジェクトを使ってデータを抽出する手順を次に示します。

1. 次に示すオブジェクトに、抽出データに関する条件を設定します。

- **Tables コレクションオブジェクト**：下位にある複数の Table オブジェクトに、抽出表の名称を設定します。Table オブジェクトの `TableName` プロパティに、抽出表の名称を設定します。
- **Columns コレクションオブジェクト**：下位にある複数の Column オブジェクトに、抽出列の名称を設定します。Column オブジェクトの `ColumnName` プロパティに、抽出列の名称を、`SortMode` プロパティと `SortPriority` プロパティにソートする条件を、`FunctionMode` プロパティに集合関数を設定します。
- **Conditions コレクションオブジェクト**：下位にある複数の Condition オブジェクトに、抽出条件式を設定します。Condition オブジェクトの `Text` プロパティに抽出条件式を設定します。

2. `QueryDefine.Exclusive` プロパティに、排他モードを設定します。

3. `QueryDefine.Execute` メソッドで、抽出を実行します。

4. `Result` オブジェクトで、抽出データを処理します。Result オブジェクトの使い方については、「[6.7 抽出データの管理 \(Result オブジェクト\)](#)」を参照してください。

6.7 抽出データの管理 (Result オブジェクト)

`QueryDefine.Execute` メソッドを実行して抽出したデータを処理する方法について説明します。抽出データを処理するときは、`Result` オブジェクトを使います。

サンプルの参照先

`Result` オブジェクトのサンプルは、DBPARTNER2 Client のフォルダ¥Ocx¥Samples¥Vb ¥Communitnet¥Query フォルダ又は DBPARTNER2 Client のフォルダ¥Ocx¥Samples¥Vb ¥Communitnet¥Update フォルダを参照してください。

6.7.1 抽出データを格納する方法

抽出データを格納する手順を次に示します。

1. 次に示す `Result` オブジェクトのプロパティを設定します。

- `Result.QueryCountLimit` プロパティ：抽出処理の結果として出力する行数を設定します。
- `Result.DataCheck` プロパティ：抽出データの文字コードをチェックするかどうかを設定します。
- `Result.HeaderOutput` プロパティ：抽出データの先頭行に、列名を付けるかどうかを設定します。

2. 1行ずつデータを抽出するときは、`Result.Get` メソッドを実行します。`Result.Get` メソッドを実行すると、次に示すプロパティの値を参照できます。

- `Result.RecordHeader` プロパティ：抽出処理の結果として出力される列名。
- `Result.RecordData` プロパティ：抽出データ。1行単位で返します。
- `Result.FieldData` プロパティ：抽出データ。1列単位で返します。
- `Result.FieldDataLength` プロパティ：列のデータ長。
- `Result.FieldCount` プロパティ：抽出データの列数。

必要に応じて、次に示すプロパティを設定します。

- `Result.FieldHeader` プロパティ：列にヘッダを設定します。

3. 抽出データを一括してファイルに出力するときは、`Result.SaveToFile` メソッドを実行します。メソッドには、抽出するデータの形式 (TXT 形式, CSV 形式, XLS 形式, DAT 形式) を設定します。

いったんクリップボードに退避するときは、`Result.CopyToClipboard` メソッドを実行します。メソッドには、抽出するデータの形式 (TXT 形式, XLS 形式) を設定します。クリップボードに退避した抽出データは、Excel などのアプリケーションに貼り付けて使えます。

4. データをすべて取得できたかどうかは、`Result.EOF` プロパティで判定します。抽出処理を途中で終了させるときは、`Result.Close` メソッドを実行します。

6.8 データの更新 (UpdateDefine オブジェクト)

DBPARTNER2 Client のオブジェクト UpdateDefine を使って、データを更新する方法です。データを更新するときは、*DBPARTNER.Connect* メソッドでサーバプログラムと接続した後、UpdateDefine オブジェクトを使います。

データの更新は、**更新版と定義版**の DBPARTNER2 Client で実行できます。参照版の DBPARTNER2 Client では、データの更新はできません。

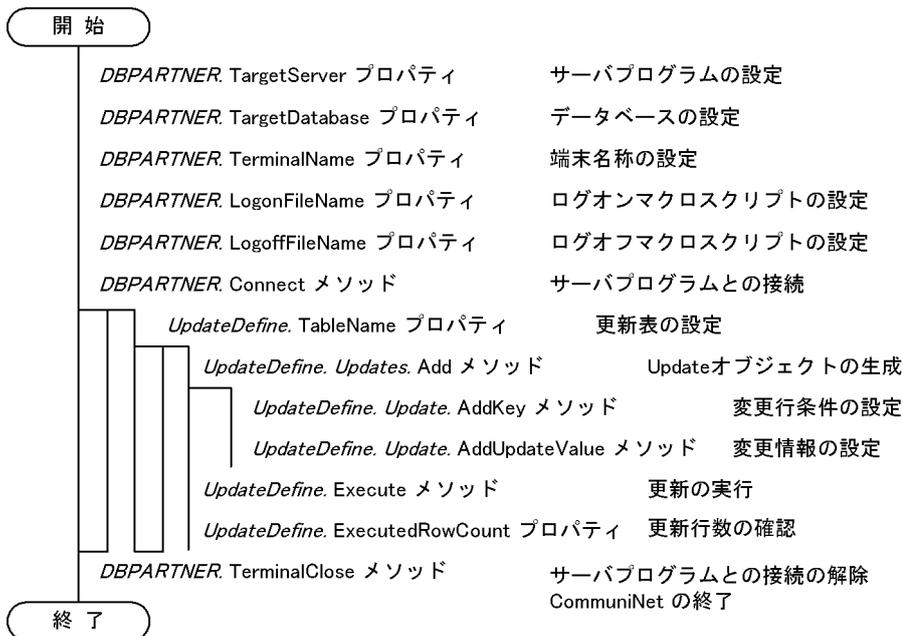
UpdateDefine オブジェクトを使った更新のオブジェクトの流れとプロパティとメソッドの実行手順を図 6-6 に示します。

図 6-6 UpdateDefine オブジェクトを使った更新のオブジェクトの流れとプロパティとメソッドの実行手順

●オブジェクトの流れ



●プロパティとメソッドの流れ



サンプルの参照先

UpdateDefine オブジェクトのサンプルは、DBPARTNER2 Client のフォルダ¥Ocx¥Samples¥Vb ¥CommuniNet¥Update フォルダを参照してください。

6.8.1 UpdateDefine オブジェクトを使ってデータを更新する手順 (変更)

UpdateDefine オブジェクトを使ってデータを更新 (変更) する手順を次に示します。

1. 次に示すプロパティを設定します。

- *UpdateDefine.Exclusive* プロパティ：排他モードを設定します。
- *UpdateDefine.UpdateMode* プロパティ：更新実行モードを設定します。
- *UpdateDefine.TableName* プロパティ：更新する表の名前を設定します。
- *UpdateDefine.TablePassword* プロパティ：更新する表のパスワードを設定します。

2. *UpdateDefine.Updates.Add* メソッドで、Update オブジェクトを生成します。

3. Updates コレクションオブジェクト下の Update オブジェクトに、変更するデータに関する条件を設定します。

変更するデータに関する条件を変更するときは、次に示すメソッドを実行します。

- *UpdateDefine.Update.AddKey* メソッド：変更行条件を設定します。
- *UpdateDefine.Update.UpdateValue* メソッド：変更行条件を設定します。

4. *UpdateDefine.Execute* メソッドを実行します。*UpdateDefine.Execute* メソッドを実行すると、次に示すプロパティの値を参照できます。

- *UpdateDefine.ExecutedRowCount* プロパティ：サーバプログラムが更新した行数。

6.8.2 UpdateDefine オブジェクトを使ってデータを更新する手順 (追加)

UpdateDefine オブジェクトを使ってデータを更新 (追加) する手順を次に示します。

1. 次に示すプロパティを設定します。

- *UpdateDefine.Exclusive* プロパティ：排他モードを設定します。
- *UpdateDefine.UpdateMode* プロパティ：更新実行モードを設定します。
- *UpdateDefine.TableName* プロパティ：更新する表の名前を設定します。
- *UpdateDefine.TablePassword* プロパティ：更新する表のパスワードを設定します。

2. *UpdateDefine.Inserts.Add* メソッドで、Insert オブジェクトを生成します。

3. Inserts コレクションオブジェクト下の Insert オブジェクトに、追加するデータに関する条件を設定します。

追加するデータに関する条件を変更するときは、次に示すメソッドを実行します。

- *UpdateDefine.Insert.AddInsertValue* メソッド：追加情報を設定します。

4. メソッドを実行した後、*UpdateDefine.Execute* メソッドを実行します。*UpdateDefine.Execute* メソッドを実行すると、次に示すプロパティの値を参照できます。

- *UpdateDefine.ExecutedRowCount* プロパティ：サーバプログラムが更新した行数。

更新が正しく実行されたかどうかを確認できます。

6.8.3 UpdateDefine オブジェクトを使ってデータを更新する手順（削除）

UpdateDefine オブジェクトを使ってデータを更新（削除）する手順を次に示します。

1. 次に示すプロパティを設定します。

- *UpdateDefine.Exclusive* プロパティ：排他モードを設定します。
- *UpdateDefine.UpdateMode* プロパティ：更新実行モードを設定します。
- *UpdateDefine.TableName* プロパティ：更新する表の名前を設定します。
- *UpdateDefine.TablePassword* プロパティ：更新する表のパスワードを設定します。

2. *UpdateDefine.Deletes.Add* メソッドで、Delete オブジェクトを生成します。

3. *Deletes* コレクションオブジェクト下の Delete オブジェクトに、削除するデータに関する条件を設定します。

削除するデータに関する条件を変更するときは、次に示すメソッドを実行します。

- *UpdateDefine.Delete.AddKey* メソッド：削除行条件を設定します。

4. メソッドを実行した後、*UpdateDefine.Execute* メソッドを実行します。*UpdateDefine.Execute* メソッドを実行すると、次に示すプロパティの値を参照できます。

- *UpdateDefine.ExecutedRowCount* プロパティ：サーバプログラムが更新した行数。

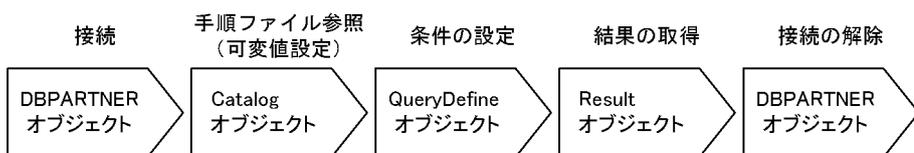
6.9 データの抽出 (Catalog オブジェクト)

DBPARTNER2 Client のオブジェクト Catalog を使って、データを抽出する方法について説明します。データを抽出するときは、*DBPARTNER.Connect* メソッドでサーバプログラムと接続した後、Catalog オブジェクトを使います。

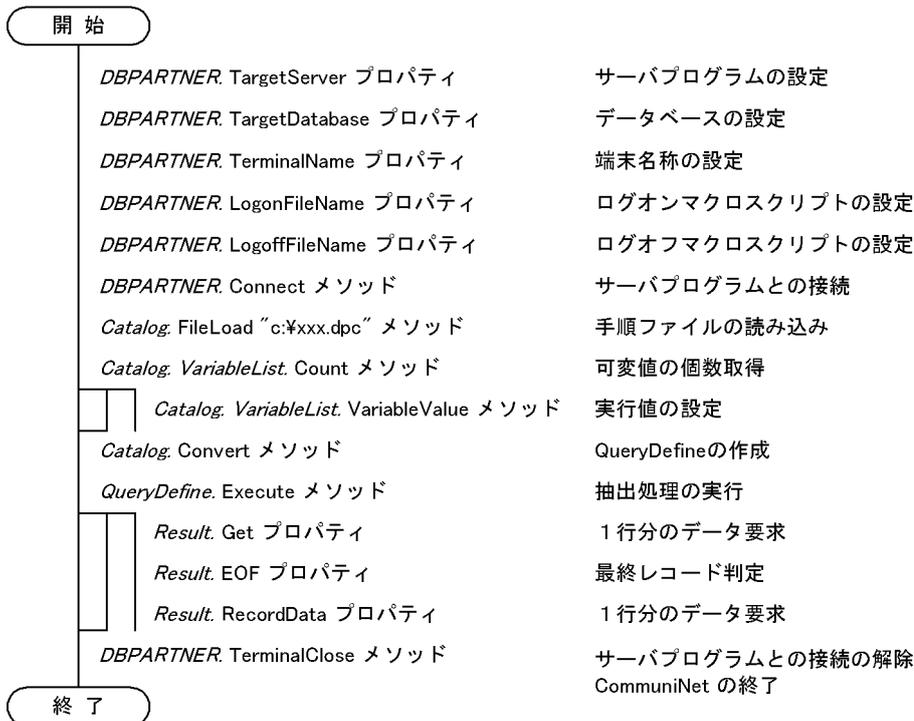
Catalog オブジェクトを使った抽出のオブジェクトの流れとプロパティとメソッドの実行手順を図 6-7 に示します。

図 6-7 Catalog オブジェクトを使った抽出のオブジェクトの流れとプロパティとメソッドの実行手順

●オブジェクトの流れ



●プロパティとメソッドの流れ



6.9.1 Catalog オブジェクトを使ってデータを抽出する手順

Catalog オブジェクトを使ってデータを抽出する手順を次に示します。

1. 次に示す Catalog オブジェクトのプロパティを設定します。

- *Catalog.FileLoad* プロパティ：手順ファイルのファイル名を設定します。

2. *Catalog.Convert* メソッドで *QueryDefine* オブジェクトを作成します。
3. *QueryDefine.Exclusive* プロパティに、排他モードを設定します。
4. *QueryDefine.Execute* メソッドで、抽出を実行します。
5. *Result* オブジェクトで、抽出データを処理します。*Result* オブジェクトの使い方については、「[6.7 抽出データの管理 \(Result オブジェクト\)](#)」を参照してください。

6.10 避けて欲しい処理

DBPARTNER2 Client のアプリケーションを開発するときには、次に示す処理は作らないでください。このような処理を作ったアプリケーションを DBPARTNER2 Client で使ったときの動作は保証しません。

6.10.1 プロパティとメソッド処理中に、別の入力を許す処理

プロパティとメソッドが処理を実行中に、別の入力を受け取ってプロパティとメソッドを実行するような処理は避けてください。プロパティとメソッドが処理中に別のプロパティとメソッドを実行すると、最初に実行したプロパティとメソッド以外はエラーになります。

例えば、ボタンコントロールに対応したコードの中にプロパティとメソッドを記述するときは、ボタンコントロールのコードの最初にコントロールを不活性（入力を受け付けない）に変える処理を記述して、コードの最後に活性（入力を受け付ける）に変える処理を記述してください。

タイマーコントロールなどを使うときは、処理ステータスを管理するフラグを使って、同時に複数のプロパティとメソッドが実行されないようにしてください。

6.10.2 TerminalClose メソッドを実行しないでアプリケーションを終了する処理

TerminalClose メソッドを実行しないでアプリケーションを終了させると、CommuniNet が終了しないことがあります。CommuniNet が終了しないときは、CommuniNet を強制終了してください。強制終了すると、TSS ユーザが残ったままになるので、この TSS ユーザはサーバ側で終了させてください。

アプリケーションを終了するときは、必ず TerminalClose メソッドでサーバプログラムとの接続を終了してから CommuniNet を終了させてください。

サーバプログラムとの接続が開始していないときは、TerminalClose メソッドを実行する必要はありません。

6.10.3 一つのアプリケーションで複数の ActiveX コントロールを使う処理

CommuniNet 接続の ActiveX コントロールは、同時に複数のコントロールからの接続要求を処理できません。一つのアプリケーションで複数のコントロールを使うときは、サーバプログラムと接続中のコントロールに対して CommuniNet の終了処理（TerminalClose メソッドの実行）をしてから、別の ActiveX コントロールで接続を開始してください。

6.10.4 Internet Explorer で ActiveX コントロールを使う処理

DBPARTNER2 Client の ActiveX コントロールは、クライアントのアプリケーションを開発するためだけに使ってください。Microsoft ActiveX Control Pad などを使って HTML ファイルに記述を追加できますが、その動作は保証されません。

6.11 プロパティとメソッドの使い方の詳細

プロパティとメソッドを実行するときの、次に示す条件について説明します。

- プロパティとメソッドの実行順序
- プロパティとメソッドの制限

6.11.1 プロパティとメソッドの実行順序

プロパティとメソッドを使うときに、前提になるプロパティ又はメソッドを表 6-2～表 6-8 に示します。

表の項目の「接続」とは、データベースとの接続 (DBPARTNER.Connect メソッド) が必要かどうかを示します。必要なときは「○」、特に必要でないときは「-」で示します。

表 6-2 プロパティとメソッドの実行順序 (DBPARTNER オブジェクト)

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
CodeChangeMode	-	-
CommitMode	-	-
DictPath	-	-
TableDefSource	-	-
IsConnect	-	-
LogoffFileName	-	-
LogonFileName	-	-
MacroTimeLimit	-	-
Password	-	-
ResponseLimit	-	-
TargetDatabase	-	-
TargetServer	-	-
TerminalMode	-	-
TerminalName	-	-
UserID	-	-
Cancel	○	-
Commit	○	-
Connect	-	-
Disconnect	○	-

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
Rollback	○	—
TerminalClose	○	—

表 6-3 プロパティとメソッドの実行順序 (TableList オブジェクト)

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
Pattern	—	—
CountLimit	—	—
Count	○	<i>TableList.Get</i>
SelectDict	—	—
TableName	○	<i>TableList.Get</i>
TableOwner	○	<i>TableList.Get</i>
TableType	○	<i>TableList.Get</i>
TableComment	○	<i>TableList.Get</i>
TableDataSource	○	<i>TableList.Get</i>
TableGroupName	○	<i>TableList.Get</i>
TableHeader	○	<i>TableList.Get</i>
TableCreateDate	○	<i>TableList.Get</i>
TableDBMName	○	<i>TableList.Get</i>
TableDeleteAuth	○	<i>TableList.Get</i>
TableInsertAuth	○	<i>TableList.Get</i>
TablePasswordMode	○	<i>TableList.Get</i>
TableSelectAuth	○	<i>TableList.Get</i>
TableUpdateAuth	○	<i>TableList.Get</i>
Get	○	—

表 6-4 プロパティとメソッドの実行順序 (ColumnList オブジェクト)

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
TableName	—	—
CountLimit	—	—
Count	○	<i>ColumnList.Get</i>
ColumnName	○	<i>ColumnList.Get</i>
ColumnType	○	<i>ColumnList.Get</i>

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
ColumnLength	○	<i>ColumnList</i> .Get
ColumnScale	○	<i>ColumnList</i> .Get
ColumnComment	○	<i>ColumnList</i> .Get
ColumnHeader	○	<i>ColumnList</i> .Get
ColumnEditPattern	○	<i>ColumnList</i> .Get
ColumnNumber	○	<i>ColumnList</i> .Get
TablePassword	—	—
SelectDict	—	—
Get	○	<i>ColumnList</i> .TableName

表 6-5 プロパティとメソッドの実行順序 (Result オブジェクト)

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
QueryCountLimit	—	—
DataCheck	—	—
HeaderOutput	—	—
RecordHeader	○	<i>QueryDefine</i> .Execute
FieldHeader	○	<i>QueryDefine</i> .Execute
RecordData	○	<i>Result</i> .Get
FieldData	○	<i>Result</i> .Get
FieldDataLength	○	<i>Result</i> .Get
FieldCount	○	<i>QueryDefine</i> .Execute
EOF	○	<i>Result</i> .Get
Get	○	<i>QueryDefine</i> .Execute
Close	○	<i>QueryDefine</i> .Execute 又は <i>Result</i> .Get
SaveToFile	○	<i>QueryDefine</i> .Execute
CopyToClipboard	○	<i>QueryDefine</i> .Execute

表 6-6 プロパティとメソッドの実行順序 (QueryDefine オブジェクト)

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
Exclusive	—	—
QueryCount	○	<i>QueryDefine</i> .Tables.Add
Execute	○	<i>QueryDefine</i> .Tables.Add

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
RemoveAll	—	—
<i>Tables.Count</i>	—	<i>QueryDefine.Tables.Add</i>
<i>Tables.Add</i>	—	—
<i>Tables.RemoveAll</i>	—	—
<i>Table.TableName</i>	—	<i>QueryDefine.Tables.Add</i>
<i>Table.TablePassword</i>	—	<i>QueryDefine.Tables.Add</i>
<i>Columns.Count</i>	—	<i>QueryDefine.Columns.Add</i>
<i>Columns.Add</i>	—	—
<i>Columns.RemoveAll</i>	—	—
<i>Column.ColumnName</i>	—	<i>QueryDefine.Columns.Add</i>
<i>Column.SortMode</i>	—	<i>QueryDefine.Columns.Add</i>
<i>Column.SortPriority</i>	—	<i>QueryDefine.Column.SortMode</i>
<i>Column.FunctionMode</i>	—	<i>QueryDefine.Columns.Add</i>
<i>Conditions.Count</i>	—	<i>QueryDefine.Conditions.Add</i>
<i>Conditions.Relation</i>	—	<i>QueryDefine.Conditions.Add</i>
<i>Conditions.Add</i>	—	—
<i>Conditions.RemoveAll</i>	—	—
<i>Condition.Text</i>	—	<i>QueryDefine.Conditions.Add</i>

表 6-7 プロパティとメソッドの実行順序 (UpdateDefine オブジェクト)

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
Exclusive	—	—
UpdateMode	—	—
TableName	—	—
TablePassword	—	—
ErrorInformation	○	<i>UpdateDefine.Execute</i>
ExecutedRowCount	○	<i>UpdateDefine.Execute</i>
Execute	○	<i>UpdateDefine.TableName</i> と次に示す組み合わせ <ul style="list-style-type: none"> • <i>UpdateDefine.Update.AddKey</i> と <i>UpdateDefine.Update.AddUpdateValue</i> • <i>UpdateDefine.Update.AddKey</i> • <i>UpdateDefine.Insert.AddInsertValue</i> • <i>UpdateDefine.Delete.AddKey</i>

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
RemoveAll	—	—
<i>Updates.Count</i>	—	<i>UpdateDefine.Updates.Add</i>
<i>Updates.Add</i>	—	—
<i>Updates.RemoveAll</i>	—	—
<i>Update.KeyCount</i>	—	<i>UpdateDefine.Update.AddKey</i>
<i>Update.UpdateValueCount</i>	—	<i>UpdateDefine.Update.AddUpdateValue</i>
<i>Update.KeyColumn</i>	—	<i>UpdateDefine.Update.AddKey</i>
<i>Update.KeyValue</i>	—	<i>UpdateDefine.Update.AddKey</i>
<i>Update.Column</i>	—	<i>UpdateDefine.Update.AddUpdateValue</i>
<i>Update.UpdateValue</i>	—	<i>UpdateDefine.Update.AddUpdateValue</i>
<i>Update.AddKey</i>	—	<i>UpdateDefine.Updates.Add</i>
<i>Update.AddUpdateValue</i>	—	<i>UpdateDefine.Updates.Add</i>
<i>Inserts.Count</i>	—	<i>UpdateDefine.Inserts.Add</i>
<i>Inserts.Add</i>	—	—
<i>Inserts.RemoveAll</i>	—	—
<i>Insert.InsertValueCount</i>	—	<i>UpdateDefine.Insert.AddInsertValue</i>
<i>Insert.Column</i>	—	<i>UpdateDefine.Insert.AddInsertValue</i>
<i>Insert.InsertValue</i>	—	<i>UpdateDefine.Insert.AddInsertValue</i>
<i>Insert.AddInsertValue</i>	—	<i>UpdateDefine.Inserts.Add</i>
<i>Deletes.Count</i>	—	<i>UpdateDefine.Deletes.Add</i>
<i>Deletes.Add</i>	—	—
<i>Deletes.RemoveAll</i>	—	—
<i>Delete.KeyCount</i>	—	<i>UpdateDefine.Delete.AddKey</i>
<i>Delete.KeyColumn</i>	—	<i>UpdateDefine.Delete.AddKey</i>
<i>Delete.KeyValue</i>	—	<i>UpdateDefine.Delete.AddKey</i>
<i>Delete.AddKey</i>	—	<i>UpdateDefine.Deletes.Add</i>

表 6-8 プロパティとメソッドの実行順序 (Catalog オブジェクト)

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
Convert	○	<i>Catalog.FileLoad</i>
FileLoad	—	—

プロパティ又はメソッド	接続	前提になるプロパティ又はメソッド
<i>Variable.Count</i>	—	<i>Catalog.FileLoad</i>
<i>Variable.VariableName</i>	—	<i>Catalog.FileLoad</i>
<i>Variable.VariableValue</i>	—	<i>Catalog.FileLoad</i>

6.11.2 プロパティとメソッドの制限

プロパティとメソッドは、サーバプログラムと接続していないとエラーになることがあります。プロパティとメソッドの状態遷移を表 6-9 に示します。表 6-9 にある「×」は、エラーになることを示します。表 6-9 に記載していないプロパティとメソッドは、状態遷移による制限はありません。

表 6-9 に示す状態は、次に示す 4 種類あります。

- **未接続**：サーバプログラムと接続していない状態を示します。
- **接続**：サーバプログラムと接続している状態を示します。
- **前処理**：Select 系 SQL 文を実行した状態を示します。
- **取得**：抽出結果の取得中の状態を示します。

表 6-9 プロパティとメソッドの状態遷移

オブジェクト	プロパティ又はメソッド	状態			
		1：未接続	2：接続	3：前処理	4：取得
DBPARTNER	Cancel	1※1	2※1	2※2	2※2
	Disconnect	1※3	1	×	×
	TerminalClose	1※3	1	×	×
	Commit	×	2※4	2※5	2※5
	Rollback	×	2※4	2※5	2※5
	Connect	2	2※6	×	×
Result	RecordHeader	×	×	3	4
	FieldHeader	×	×	3	4
	RecordData	×	×	×	4
	FieldData	×	×	×	4
	FieldType	×	×	3	4
	FieldDataLength	×	×	×	4
	FieldCount	1※7	2※7	3	4

オブジェクト	プロパティ又は メソッド	状態			
		1：未接続	2：接続	3：前処理	4：取得
	Get	×	×	4 又は 2※8	4 又は 2※8
	Close	1※3	2※3	2	2
	SaveToFile	×	×	2	×
	CopyToClipboard	×	×	2	×
TableList	Get	×	2	×	×
ColumnList	Get	×	2	×	×
QueryDefine	QueryCount	×	2	×	×
	Execute	×	3	×	×
UpdateDefine	Execute	×	2	×	×

注※1

Cancel は正常に終了します。

注※2

Cancel が正常に受け付けられたときは遷移します。受け付けられなかったときは遷移しません。

注※3

何も処理しません。

注※4

Commit/Rollback メソッドは正常に終了します。

注※5

Commit/Rollback メソッドを処理した後で、DBPARTNER2 Client が自動的に Close メソッドを実行します。

注※6

現在の接続を終了した後で、接続を再開します。

注※7

0 を返します。

注※8

最終データまで取得したとき、又はエラーになったときは、2 に遷移します。

7

DBPARTNER2 Client のオブジェクトの文法 (CommuniNet 接続)

この章では、DBPARTNER2 Client のアプリケーションを開発するときに使う ActiveX コントロールのオブジェクトの文法 (CommuniNet 接続) について説明します。

7.1 プロパティページの設定

DBPARTNER2 Client の ActiveX コントロールで設定できるプロパティページについて説明します。

CommuniNet 接続のときは、プロパティページには [端末] タブと [サーバ] タブがあります。

7.1.1 プロパティページの [端末] タブ

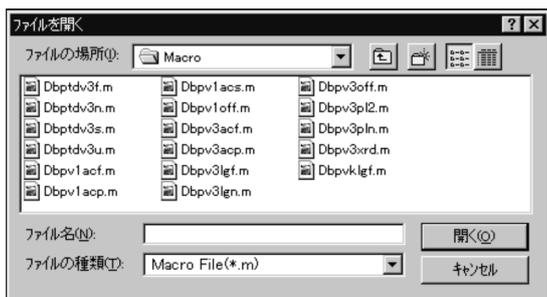
DBPARTNER2 Client のプロパティページ (CommuniNet 接続 [端末] タブ) を図 7-1 に示します。

図 7-1 DBPARTNER2 Client のプロパティページ (CommuniNet 接続 [端末] タブ)



- 端末名称 (*DBPARTNER.TerminalName* プロパティ)
通信に使う CommuniNet の端末名を、文字列で設定します。省略時仮定値はありません。
- 端末起動モード (*DBPARTNER.TerminalMode* プロパティ)
端末起動モードを、コンボボックスから選びます。省略時仮定値は「0-非表示」です。
 - 0-非表示
CommuniNet を非表示モードで起動します。
 - 1-アイコン表示
CommuniNet を最小化モード (タスクバー上のボタン表示) で起動します。
 - 2-表示
CommuniNet を表示モードで起動します。
- ログオンスクリプト (*DBPARTNER.LogonFileName* プロパティ)
サーバプログラムと接続するためのログオン操作を登録した、マクロスクリプトのファイルを設定します。[...] ボタンをクリックすると、マクロスクリプトを選ぶ画面が表示されます。マクロスクリプトを選ぶ画面を図 7-2 に示します。

図 7-2 マクロスクリプトを選ぶ画面



- ログオフスクリプト (*DBPARTNER.LogoffFileName* プロパティ)

サーバプログラムとの接続を終了するためのログオフ操作を登録した、マクロスクリプトのファイルを設定します。[...] ボタンをクリックすると、マクロスクリプトを選ぶ画面が表示されます。マクロスクリプトを選ぶ画面は、図 7-2 と同じです。
- スクリプト実行時間 (*DBPARTNER.MacroTimeLimit* プロパティ)

ログオン/ログオフ操作のマクロスクリプトの実行待ち時間を設定します。省略すると、60 秒が仮定されます。

7.1.2 プロパティページの [サーバ] タブ

DBPARTNER2 Client のプロパティページ (CommuniNet 接続 [サーバ] タブ) を図 7-3 に示します。

図 7-3 DBPARTNER2 Client のプロパティページ (CommuniNet 接続 [サーバ] タブ)



- サーバ種別 (*DBPARTNER.TargetServer* プロパティ)

接続するサーバプログラムを、次に示す候補からコンボボックスで選びます。省略時仮定値は「0-HOAPDBS E2, HOAPDBS/PC」です。

0-HOAPDBS E2, HOAPDBS/PC
1-DBPARTNER/Link, ACE3 E3
- データベース種別 (*DBPARTNER.TargetDatabase* プロパティ)

接続するデータベースを、次に示す候補からコンボボックスで選びます。省略時仮定値は「0-VOS3 XDM/RD E2」です。

0-VOS3 XDM/RD E2
1-VOS3 XDM/SD E2

2-VOS3 PDMII E2

3-VOS3 ACE3 ユーザファイル

4-VOS3 RDB1

5-VOSK RDB 編成ファイル

6-VOSK スプールファイル

7-VOS1 PDMII E2

8-VOS1 ACE3 ユーザファイル

9-VOS1 スプールファイル

- ユーザ ID (*DBPARTNER.UserID* プロパティ)

ユーザ ID を設定します。ここに設定した値は、UserID プロパティに設定されます。文字列は、最大 8 バイト入力できます。

- パスワード (*DBPARTNER.Password* プロパティ)

パスワードを設定します。ここに設定した値は、Password プロパティに設定されます。文字列は、最大 8 バイト入力できます。

- コミットモード (*DBPARTNER.CommitMode* プロパティ)

DBPARTNER2 Client が自動的にトランザクションを制御するかどうかを、次に示す候補からコンボボックスで選びます。省略すると「0-AutoCommit」が仮定されます。

0-AutoCommit

DBPARTNER2 Client が自動的にトランザクションを制御します。

1-ManualCommit

アプリケーションから Commit/Rollback を実行してトランザクションを制御します。

7.2 DBPARTNER オブジェクトの詳細

DBPARTNER オブジェクトは、サーバプログラムとの接続を管理するオブジェクトです。このオブジェクトは、DBPARTNER2 Client のトップオブジェクトです。

DBPARTNER オブジェクトのプロパティとメソッドの一覧を次に示します。

●DBPARTNER オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
CodeChangeMode	○	○	コード変換処理を実行するプログラムを設定します。
CommitMode	○	○	トランザクション終了モードを設定します。
DictPath	○	○	見出し辞書を格納しているパスを文字列で設定します。
TableDefSource	○	○	表情報、列情報の取得元を設定します。
IsConnect	—	○	サーバと接続しているかどうかの状態を取得します。
LogoffFileName	○	○	ログオフに使うマクロスクリプトファイル名を設定します。
LogonFileName	○	○	ログオンに使うマクロスクリプトファイル名を設定します。
MacroTimeLimit	○	○	マクロスクリプトの実行監視時間を設定します。
Password	○	○	パスワードを文字列で設定します。
ResponseLimit	○	○	応答待ち時間を設定します。
TargetDatabase	○	○	接続するデータベースを設定します。
TargetServer	○	○	接続するサーバプログラムを設定します。
TerminalMode	○	○	CommuniNet の起動モードを設定します。
TerminalName	○	○	CommuniNet の端末名を設定します。
UserID	○	○	ログインするユーザ ID を文字列で設定します。

●DBPARTNER オブジェクトのメソッドの一覧

メソッド	説明
Cancel (CancelEx)	実行中の処理をキャンセルします。
Commit	実行中の処理を有効にして、トランザクションを終了させます。
Connect	サーバとの接続を開始します。
Disconnect	サーバとの接続を終了します。
Rollback	実行中の処理を無効にして、トランザクションを終了させます。
TerminalClose	サーバとの接続を終了して、CommuniNet を終了させます。

CodeChangeMode プロパティ

機能

コード変換処理を実行するプログラムを設定します。コード変換処理とは、サーバ側の文字コード (EBCDIK/KEIS) とクライアント側の文字コード (JIS/シフト JIS) の変換処理のことです。

dbpServer を設定すると、TargetServer プロパティで設定したサーバでコード変換処理を実行します。dbpClient を設定すると、TerminalName プロパティで設定した CommuniNet でコード変換処理を実行します。

このプロパティは、値も取得できます。

形式

```
Object.CodeChangeMode [= Mode]
```

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

- *Mode*

コード変換種別を、次に示す定数又は数値で設定します。

定数	数値	内容
<code>dbpServer</code>	0	コード変換を、サーバ側のプログラムで実行します。
<code>dbpClient</code>	1	コード変換を、クライアント側のプログラムで実行します。

使い方

- dbpClient を設定するときは、CommuniNet のコードマッピングユーティリティを使って、シフト JIS コードと KEIS コードを対応付けてください。
- dbpClient を設定するときは、サーバ側の設定を変更する必要があります。dbpClient を設定したときの注意については、「[8.2 クライアント側プログラムでコード変換するときの注意](#)」を参照してください。
- コード変換種別に設定した値は、初回の Connect メソッド実行時 (CommuniNet 起動時) に CommuniNet に渡されるので、CommuniNet が終了するまで変更できません。

CommitMode プロパティ

機能

サーバプログラムに対するトランザクションを自動的に終了するかどうかを設定します。このプロパティは、値も取得できます。

形式

```
Object.CommitMode [= Mode]
```

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

- *Mode*

トランザクション終了モードを、次に示す定数又は数値で設定します。

定数	数値	内容
<code>dbpAuto</code>	0	自動的にトランザクションを終了します。
<code>dbpManual</code>	1	アプリケーションで終了を設定します。

使い方

- `dbpAuto` を設定すると、Commit メソッド又は Rollback メソッドを実行しても無視されます。
- `dbpAuto` を設定したときに一覧情報の取得、データの抽出、データの更新処理が正常に終わると、自動的にトランザクションが Commit されます。
- `dbpAuto` を設定したときに一覧情報の取得、データの抽出、データの更新処理がエラーになると、自動的にトランザクションが Rollback されます。
- `dbpManual` は更新処理のときにだけ有効です。表情報、列情報、データの抽出など、更新以外の処理では有効になりません。
- `dbpManual` を設定したときは、更新処理が終了した時点で Commit メソッド又は Rollback メソッドでトランザクションを終了してください。更新処理が終了した時点でトランザクションを終了しないと、以降の処理の中で自動的に Commit が実行されます。
- サーバプログラムと接続中 (Connect メソッドから Disconnect メソッドまでの間) は、CommitMode プロパティの値を変更してもトランザクション終了モードを変えることはできません。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

DictPath プロパティ

機能

見出し辞書を格納しているディレクトリのパスを設定します。このプロパティは、値の取得もできます。

形式

```
Object.DictPath [= Path]
```

パラメタ

- *Object*
DBPARTNER オブジェクトへの参照を表すオブジェクト式です。
- *Path*
パスを、1~256 の文字列で設定します。パスの終端の “¥” は省略できます。

使い方

- 表情報、列情報を見出し辞書から取得する場合は、必ず設定してください。

TableDefSource プロパティ

機能

表情報、列情報を取得するモードを設定します。このプロパティは、値の取得もできます。

形式

```
Object.TableDefSource [= Mode]
```

パラメタ

- *Object*
DBPARTNER オブジェクトへの参照を表すオブジェクト式です。
- *Mode*
表情報、列情報の要求先を、定数又は数値で設定します。

定数	数値	情報の要求先
dbpDB	0	表情報、列情報をデータベースから取得します。
dbpDICT	1	表情報、列情報を見出し辞書から取得します。

定数	数値	情報の要求先
dbpDBDICT	2	表情報、列情報をデータベースから取得し、ラベル (TableHeader, ColumnHeader) は見出し辞書から取得します。

IsConnect プロパティ

機能

サーバプログラムとの接続状態を BOOLEAN 値で取得します。

形式

Object. IsConnect

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

戻り値

定数	内容
TRUE (真)	サーバプログラムと接続されています。
FALSE (偽)	サーバプログラムと接続されていません。

使い方

- サーバプログラムの時間監視機能などで強制的に接続が解除されたときは、接続されていなくても TRUE が返ることがあります。

LogoffFileName プロパティ

機能

サーバプログラムとの接続を終了するための、ログオフ操作を登録したマクロスクリプトのファイル名を設定します。このプロパティは、値も取得できます。

形式

Object. LogoffFileName [= *FileName*]

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

- *FileName*

ログオフ操作を登録したマクロスクリプトのファイル名を、1~256 バイトの文字列で設定します。マクロスクリプトファイル名は省略できません。

使い方

- マクロスクリプトファイル名には、ドライブ名で始まる絶対パス名で設定してください。ファイル名だけを設定すると、CommuniNet を組み込んだフォルダの¥Macro フォルダ下だけを探します。
- マクロスクリプトファイルの作成方法については、「[2.4.2 マクロスクリプトファイルの作成](#)」を参照してください。

LogonFileName プロパティ

機能

サーバプログラムと接続するための、ログオン操作を登録したマクロスクリプトのファイル名を設定します。このプロパティは、値も取得できます。

形式

Object. LogonFileName [= *FileName*]

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

- *FileName*

ログオン操作を登録したマクロスクリプトのファイル名を、1~256 バイトの文字列で設定します。マクロスクリプトファイル名は省略できません。

使い方

- マクロスクリプトファイル名には、ドライブ名で始まる絶対パス名で設定してください。ファイル名だけを設定すると、CommuniNet を組み込んだフォルダの¥Macro フォルダ下だけを探します。
- マクロスクリプトファイルの作成方法については、「[2.4.2 マクロスクリプトファイルの作成](#)」を参照してください。

MacroTimeLimit プロパティ

機能

ログオン／ログオフ操作のマクロスクリプトの実行待ち時間を設定します。このプロパティは、値も取得できます。

形式

```
Object. MacroTimeLimit [= Time]
```

パラメタ

- *Object*
DBPARTNER オブジェクトへの参照を表すオブジェクト式です。
- *Time*
マクロスクリプトの実行待ち時間を、1～600（秒）の数値で設定します。省略すると、60（秒）が仮定されます。

使い方

- 設定した実行待ち時間を超えてもマクロスクリプトの実行が終了しないときは、Connect メソッド、TerminalClose メソッド、Disconnect メソッドがエラーになります。エラーが頻繁に起こるときは、マクロスクリプトの実行待ち時間を大きくしてください。

Password プロパティ

機能

UserID プロパティで設定したユーザ ID のパスワードを、文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object. Password [= PWD]
```

パラメタ

- *Object*
DBPARTNER オブジェクトへの参照を表すオブジェクト式です。
- *PWD*
パスワードを、0～8 バイトの文字列で設定します。値をクリアするときは、長さ 0 の文字列を設定します。

使い方

- サーバプログラムと接続中（Connect メソッドから Disconnect メソッドまでの間）は、Password プロパティの値を変更しても接続ユーザのパスワードを変えることはできません。

ResponseLimit プロパティ

機能

サーバプログラムで実行中の処理に対するキャンセル（中止）を有効にするかどうかと、キャンセル処理に対するサーバプログラムからの応答監視時間（待ち時間）を設定します。このプロパティは、値も取得できます。

形式

```
Object.ResponseLimit [= Time]
```

パラメタ

- Object**
DBPARTNER オブジェクトへの参照を表すオブジェクト式です。
- Time**
キャンセル応答待ち時間を、次に示す数値（秒）で設定します。

数値（秒）	内容
0	非同期キャンセルしません。
1~3600	非同期キャンセルの応答を待つ時間です。

使い方

- キャンセル応答待ち時間とは、Cancel メソッドを実行してからサーバプログラムが応答するまでの待ち時間です。キャンセル応答待ち時間を超えてもサーバプログラムから応答がないときは、サーバ側で異常が起こったと判断して、Cancel メソッドがエラーになります。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

TargetDatabase プロパティ

機能

接続するデータベースを設定します。このプロパティは、値も取得できます。

形式

Object. TargetDatabase [= *DB*]

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

- *DB*

接続するデータベースを、次に示す定数又は数値で設定します。

定数	数値	内容
dbpV3RD	0	VOS3 XDM/RD E2
dbpV3SD	1	VOS3 XDM/SD E2
dbpV3PDM	2	VOS3 PDMII E2
dbpV3USR	3	VOS3 ACE3 ユーザファイル
dbpV3RDB1	4	VOS3 RDB1
dbpVKRDBF	5	VOSK RDB 編成ファイル
dbpVKSPL	6	VOSK スプールファイル
dbpV1PDM	7	VOS1 PDMII E2
dbpV1USR	8	VOS1 ACE3 ユーザファイル
dbpV1SPL	9	VOS1 スプールファイル

使い方

- このプロパティに設定したデータベースと実際に接続するデータベースが異なると、アプリケーションが正しく動作しないおそれがあります。
- 設定するデータベースと、TargetServer プロパティで設定するサーバプログラムには、組み合わせの規則があります。データベースとサーバプログラムの組み合わせを次に示します。

TargetDatabase プロパティの設定	TargetServer プロパティの設定	
	dbpHoap	dbpAce3
dbpV3RD	○	×
dbpV3SD	○	×
dbpV3PDM	×	○
dbpV3USR	×	○
dbpV3RDB1	×	○
dbpVKRDBF	○	×

TargetDatabase プロパティの設定	TargetServer プロパティの設定	
	dbpHoap	dbpAce3
dbpVKSPL	○	×
dbpV1PDM	×	○
dbpV1USR	×	○
dbpV1SPL	×	○

(凡例)

○：有効な組み合わせです。

×：この組み合わせは設定できません。この組み合わせでプロパティを設定すると、Connect メソッドがエラーになります。

TargetServer プロパティ

機能

接続するサーバプログラムを設定します。このプロパティは、値も取得できます。

形式

Object. TargetServer [= SVR]

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

- *SVR*

接続するサーバプログラムを、次に示す定数又は数値で設定します。

定数	数値	内容
dbpHoap	0	HOAPDBS E2 又は HOAPDBS/PC
dbpAce3	1	ACE3 E3 又は DBPARTNER/Link

使い方

- このプロパティに設定したサーバプログラムと実際に接続するサーバプログラムが異なると、アプリケーションが正しく動作しないおそれがあります。
- 設定するサーバプログラムと、TargetDatabase プロパティで設定するデータベースには、組み合わせの規則があります。データベースとサーバプログラムの組み合わせについては、TargetDatabase プロパティの文法の説明を参照してください。

TerminalMode プロパティ

機能

CommuniNet の端末起動モードを設定します。このプロパティは、値も取得できます。

形式

```
Object. TerminalMode [= Mode]
```

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

- *Mode*

CommuniNet の端末起動モードを、次に示す定数又は数値で設定します。

定数	数値	内容
<code>dbpHide</code>	0	CommuniNet を非表示モードで起動します。
<code>dbpIcon</code>	1	CommuniNet を最小化モード（タスクバーのボタン表示）で起動します。
<code>dbpShow</code>	2	CommuniNet を表示モードで起動します。

使い方

- `dbpIcon` 又は `dbpShow` を設定すると、マクロスクリプトの実行過程を CommuniNet で確認できます。ただし、CommuniNet に直接キー入力するとサーバプログラムとの通信エラーが起こるおそれがあります。そのため、`dbpIcon` 又は `dbpShow` はアプリケーションの動作を確認するためだけに使うようにして、通常は `dbpHide` を設定することをお勧めします。
- 端末起動モードは、アプリケーション起動後に初めて Connect メソッドを実行するときに参照されます。アプリケーションの起動中に端末起動モードを変更するときは、TerminalClose メソッドを実行していったん CommuniNet を終了させてください。

TerminalName プロパティ

機能

通信に使う CommuniNet の端末名を、文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object. TerminalName [= Name]
```

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

- *Name*

CommuniNet の端末名を、1~14 バイトの文字列で設定します。端末名は、省略できません。

使い方

- 端末名については、CommuniNet のマニュアル又はオンラインヘルプを参照してください。
- 端末名の値を変更して Connect メソッドを実行すると、現在起動している CommuniNet を終了させて、新しく設定した端末名の CommuniNet が起動されます。

UserID プロパティ

機能

サーバプログラムに接続するためのユーザ ID を、文字列で設定します。このプロパティは、値も取得できません。

形式

```
Object.UserID [= UID]
```

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

- *UID*

ユーザ ID を、0~8 バイトの文字列で設定します。値をクリアするときは、長さ 0 の文字列を設定します。

使い方

- サーバプログラムと接続中 (Connect メソッドから Disconnect メソッドまでの間) は、UserID プロパティの値を変更しても接続ユーザを変えることはできません。

Cancel メソッド (CancelEx メソッド)

機能

サーバプログラムに対して実行中の処理をキャンセルします。

ビジュアル開発ツールの仕様によっては、メソッド名を「Cancel」と記述するとエラーになることがあります。このときは、メソッド名を「CancelEx」と記述してください。

形式

```
Object.Cancel   又は   Object.CancelEx
```

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

使い方

- Cancel メソッドが有効になる範囲を次に示します。
 - *QueryDefine.Execute* メソッドを実行してから処理が終了するまで。
 - 1 回目の *Result.Get* メソッドを実行してから処理が終了するまで。
 - *TableList.Get* メソッドを実行してから処理が終了するまで。
 - *ColumnList.Get* メソッドを実行してから処理が終了するまで。
 - *QueryDefine.QueryCount* メソッドを実行してから処理が終了するまで。
 - *UpdateDefine.Execute* メソッドを実行してから処理が終了するまで。
 - *Result.SaveToFile* メソッドの実行からメソッドの処理終了まで。
 - *Result.CopyToClipboard* メソッドの実行からメソッドの処理終了まで。
- 中断する前に取得していたデータは、削除されます。
- 中断した処理の結果は保証されません。
- *Result.SaveToFile* メソッド、*Result.CopyToClipboard* メソッドでファイル又はクリップボードにデータを出力中に Cancel メソッドを実行すると、キャンセルしたデータまでをファイル又はクリップボードに出力されます。
- AP 開発ツールの仕様によっては、Cancel メソッドを記述するとエラーが発生することがあります。この場合、「CancelEx」と記述してください。

Commit メソッド

機能

サーバプログラムに対して実行したトランザクションを有効にして、トランザクションを終了します。

形式

```
Object.Commit
```

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

使い方

- CommitMode プロパティに dbpAuto を設定しているときは、正常に処理が終わった時点で自動的に Commit メソッドが実行されるので、アプリケーションから Commit メソッドを実行する必要はありません。
- CommitMode プロパティで dbpAuto を設定している状態で Commit メソッドを実行しても、無視されます（エラーにはなりません）。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

Connect メソッド

機能

CommuniNet を起動して、LogonFileName プロパティで設定したマクロスクリプトを実行します。マクロスクリプトの実行が正常に完了すると、サーバプログラムと接続します。

形式

```
Object.Connect
```

パラメタ

- *Object*

DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

使い方

- 既にサーバプログラムと接続されている状態で Connect メソッドを実行すると、LogoffFileName プロパティで設定したマクロスクリプトが実行されて、一度サーバプログラムとの接続を終了させた後、新たに接続処理が実行されます。
- 抽出処理の実行中は、Connect メソッドは使えません。使うと Connect メソッドがエラーになります。抽出処理の実行中とは、QueryDefine.Execute メソッドを実行してからデータの取得が完了するまで（又は、データの取得を中止してから Close メソッドを実行するまで）のことです。

- Connect メソッドで開始したサーバプログラムとの接続を終了させるには、TerminalClose メソッドを使う方法と Disconnect メソッドを使う方法があります。

サーバプログラムとの接続の終了と一緒に CommuniNet も終了させるときは、TerminalClose メソッドを使います。通常は TerminalClose メソッドを使ってください。サーバプログラムとの接続を終了 (TSS のログオフだけ) を実行するときは、Disconnect メソッドを使います。このとき、アプリケーションを終了させるときに、TerminalClose メソッドを実行する必要があります。

Disconnect メソッド

機能

LogoffFileName プロパティで設定したマクロスクリプトを実行します。マクロスクリプトの実行が正常に完了すると、サーバプログラムとの接続を解除します。

Disconnect メソッドは、サーバプログラムとの接続は終了するが CommuniNet は起動したままにしておくときに使います。CommuniNet を起動したままにしておく、次の Connect メソッドに掛かる時間が短縮されます。

サーバプログラムとの接続の終了と一緒に CommuniNet も終了させるときは、TerminalClose メソッドを使ってください。

形式

<i>Object</i> . Disconnect

パラメタ

- *Object*
DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

使い方

- 抽出処理の実行中は、Disconnect メソッドは使えません。使うと Disconnect メソッドがエラーになります。抽出処理の実行中とは、*QueryDefine.Execute* メソッドを実行してからデータの取得が完了するまで (又は、データの取得を中止してから Close メソッドを実行するまで) のことです。
- Connect メソッドが実行されていない状態で Disconnect メソッドを実行しても無視されます (エラーにはなりません)。

Rollback メソッド

機能

サーバプログラムに対して実行したトランザクションを無効 (*QueryDefine.Execute* メソッド, *UpdateDefine.Execute* メソッドを実行する前の状態) にして, トランザクションを終了します。

形式

```
Object.Rollback
```

パラメタ

- *Object*
DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

使い方

- CommitMode プロパティに dbpAuto を設定しているときは, エラーで処理が終わった時点で自動的に Rollback メソッドが実行されるので, アプリケーションから Rollback メソッドを実行する必要はありません。
- CommitMode プロパティで dbpAuto を設定している状態で Rollback メソッドを実行しても, 無視されます (エラーにはなりません)。
- エラーになった時点までの更新処理を有効にするときは, CommitMode プロパティに dbpManual を設定してください。
- このプロパティは, 使えるデータベースに制限があります。データベースの制限については, 「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

TerminalClose メソッド

機能

Disconnect メソッドと同じ処理を実行した後, Connect メソッドで起動した CommuniNet を終了させます。

形式

```
Object.TerminalClose
```

パラメタ

- *Object*
DBPARTNER オブジェクトへの参照を表すオブジェクト式です。

使い方

- 抽出処理の実行中は、TerminalClose メソッドは実行しないでください。実行すると TerminalClose メソッドがエラーになります。抽出処理の実行中とは、*QueryDefine.Execute* メソッド実行からデータの取得が完了するまで（又は、データの取得を中止して Close メソッド実行まで）のことです。
- Connect メソッドを実行していない状態で TerminalClose メソッドを実行しても無視されます（エラーになりません）。
- TerminalClose メソッドは、サーバプログラムとの接続終了と一緒に CommuniNet も終了させます。サーバプログラムとの接続だけ終了させて CommuniNet を終了させたくないときは、Disconnect メソッドを使ってください。アプリケーションを終了するときは、必ず TerminalClose メソッドを実行して CommuniNet も一緒に終了させてください。
- サーバプログラムとの通信がエラーになったときは、TerminalClose メソッドを実行しても CommuniNet が終了しないことがあります。このときは、CommuniNet の終了コマンドを使って終了させてください。

7.3 Result オブジェクトの詳細

Result オブジェクトは、QueryDefine オブジェクトで実行した抽出処理の結果として出力する抽出データを管理するオブジェクトです。

Result オブジェクトのプロパティとメソッドの一覧を次に示します。

●Result オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
DataCheck	○	○	抽出データの中の文字コードをチェックするかどうかを設定します。
EOF	—	○	データが取得できたかどうかを取得します。
FieldCount	—	○	抽出データの列数を返します。
FieldData	—	○	抽出データを1列単位で返します。
FieldDataLength	—	○	列のデータ長を返します。
FieldHeader	○	○	設定した列にヘッダを設定します。
HeaderOutput	○	○	抽出データの先頭行に列名を付けるかどうかを設定します。
QueryCountLimit	○	○	抽出処理の結果として出力する行数を、数値で設定します。
RecordData	—	○	抽出データを1行単位で返します。
RecordHeader	—	○	抽出処理の結果として出力される列名を、1行のテキスト形式で取得します。

●Result オブジェクトのメソッドの一覧

メソッド	説明
Close	抽出処理を終了させます。
CopyToClipboard	抽出データをクリップボードに出力します。
Get	1行分のデータを要求します。
SaveToFile	抽出データをファイルに出力します。

DataCheck プロパティ

機能

抽出データの中に文字コード以外のデータがあったときに、エラーを返すかどうかを設定します。このプロパティは、値も取得できます。

文字コード以外のデータと判定するデータを、次に示します。

- 1バイト文字で X'20'~X'7E', 又は X'A1'~X'DF'以外の文字。

- 2バイト文字で1バイト目が、X'81'~X'9F'、又はX'E0'~X'FC'以外の文字。
- 2バイト文字で2バイト目が、X'40'~X'7E'、又はX'80'~X'FC'以外の文字。

形式

```
Object.DataCheck [= CheckMode]
```

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。
- *CheckMode*
文字コードをチェックするかどうかを、次に示す定数で設定します。

定数	内容
FALSE	文字コードをチェックしません。
TRUE	文字コード以外のデータがあるときは、エラーを返します。

使い方

- DataCheck プロパティの値は、*QueryDefine.Execute* メソッドを実行するときに使われます。*QueryDefine.Execute* メソッドを実行した後で DataCheck プロパティの値を変更しても、*Result.Get* メソッドの結果を変えることはできません。

EOF プロパティ

機能

直前に実行した *Result.Get* メソッドでデータが取得できたかどうかを、BOOLEAN 値で取得します。

形式

```
Object.EOF
```

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。

戻り値

定数	内容
FALSE (偽)	データを読み込みました。
TRUE (真)	読み込むデータがありません。

使い方

- EOF プロパティは、データの最後を知るために使います。*Result.Get* メソッドの次に EOF プロパティを実行すると、すべての抽出データを取得し終えたかどうかを判断できます。EOF プロパティが TRUE を返したときは、直前の *Result.Get* メソッドで読み込むデータがなかったことが分かります。

FieldCount プロパティ

機能

抽出データの列数を、数値で取得します。

形式

```
Object. FieldCount
```

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。

使い方

- *QueryDefine.Execute* メソッドを実行する前は、0 を返します。

FieldData プロパティ

機能

列番号で設定した列のデータを取得します。

形式

```
Object. FieldData (Index)
```

パラメタ

- *Object*

Result オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1～FieldCount プロパティの範囲内で設定します。

使い方

- 列番号は、*QueryDefine.Execute* メソッドを実行した後で、FieldCount プロパティで得られた列数の範囲内で指定してください。
- *QueryDefine.Execute* メソッドを実行する前に FieldData プロパティを使うと、エラーになります。

FieldDataLength プロパティ

機能

列番号で設定した列のデータ長を、数値で取得します。

形式

```
Object.FieldDataLength (Index)
```

パラメタ

- *Object*

Result オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1～FieldCount プロパティの範囲内で設定します。

使い方

- 列番号は、*QueryDefine.Execute* メソッドを実行した後で、FieldCount プロパティで得られた列数の範囲内で設定してください。
- *QueryDefine.Execute* メソッドを実行する前に FieldDataLength プロパティを使うと、エラーになります。

FieldHeader プロパティ

機能

列番号で設定した列にヘッダを、数値と文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object. FieldHeader (Index) [= String]
```

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的の列の列番号を、数値で設定します。番号は、1~FieldCount プロパティの範囲内で設定します。
- *String*
列に付けるヘッダを、0~256 バイトの文字列で設定します。値をクリアするときは、長さ 0 の文字列を設定します。

使い方

- 列番号は、*QueryDefine.Execute* メソッドを実行した後、FieldCount プロパティで得られた列数の範囲内で設定してください。
- *QueryDefine.Execute* メソッドを実行する前に FieldHeader プロパティを使うと、エラーになります。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

HeaderOutput プロパティ

機能

SaveToFile メソッド、CopyToClipboard メソッドを使うときに、抽出データの先頭行にヘッダを付けるかどうかを設定します。このプロパティは、値も取得できます。

形式

```
Object. HeaderOutput [= HeaderMode]
```

パラメタ

- *Object*

Result オブジェクトへの参照を表すオブジェクト式です。

- *HeaderMode*

ヘッダを付けるかどうかを、次に示す定数で設定します。

定数	内容
<code>FALSE</code>	抽出データにヘッダを付けません。
<code>TRUE</code>	抽出データの 1 行目にヘッダを付けます。

使い方

- `HeaderOutput` プロパティの値は、`SaveToFile` メソッド又は `CopyToClipboard` メソッドを実行するときに使われます。`SaveToFile` メソッド又は `CopyToClipboard` メソッドを実行した後で `HeaderOutput` プロパティの値を変更しても、`SaveToFile` メソッド又は `CopyToClipboard` メソッドの結果を変えることはできません。
- ヘッダが設定されていない列は、列名を出力します。

QueryCountLimit プロパティ

機能

抽出処理の結果として出力する行数を、数値で設定します。このプロパティは、値も取得できます。

形式

```
Object. QueryCountLimit [= Number]
```

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。
- *Number*
抽出する行数の制限値を、次に示す数値で設定します。

数値	内容
<code>0</code>	行数は制限しません。
<code>1~999,999,999</code>	設定した行数まで、データを取得します。

使い方

- QueryCountLimit プロパティの値は、*QueryDefine.Execute* メソッドを実行するときに使われます。*QueryDefine.Execute* メソッドを実行した後で QueryCountLimit プロパティの値を変更しても、*Result.Get* メソッドの結果を変えることはできません。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

RecordData プロパティ

機能

Result.Get メソッドで取得した 1 レコード単位の抽出データを、文字列で取得します。取得する文字列は、1 行のテキスト形式になっています。

形式

```
Object. RecordData
```

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。

使い方

- テキストデータのフォーマットについては、「[付録 C DBPARTNER2 Client が扱うデータの形式](#)」を参照してください。
- データは、*Result.Get* メソッドを実行するときに更新されます。
- *QueryDefine.Execute* メソッドを実行する前に RecordData プロパティを使うと、エラーになります。

RecordHeader プロパティ

機能

抽出処理の結果として出力される列名を、文字列で取得します。取得する文字列は、1 行のテキスト形式になっています。

形式

```
Object. RecordHeader
```

パラメタ

- *Object*

Result オブジェクトへの参照を表すオブジェクト式です。

使い方

- テキスト形式のフォーマットについては、「付録 C DBPARTNER2 Client が扱うデータの形式」を参照してください。
- *QueryDefine.Execute* メソッドを実行する前に *RecordHeader* プロパティを使うと、エラーになります。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「8.3 データベースと使えるプロパティ/メソッド」を参照してください。

Close メソッド

機能

抽出処理を終了させます。

形式

```
Object. Close
```

パラメタ

- *Object*

Result オブジェクトへの参照を表すオブジェクト式です。

使い方

- EOF プロパティが FALSE の状態でほかの処理に移りたいときは、必ず *Close* メソッドを実行してください。
- *Close* メソッドを連続して実行しても、不要な *Close* メソッドは無視されます（エラーにはなりません）。

CopyToClipboard メソッド

機能

抽出データを、設定した形式でクリップボードに出力します。

形式

Object.CopyToClipboard (Type)

パラメタ

- *Object*

Result オブジェクトへの参照を表すオブジェクト式です。

- *Type*

出力するデータの形式を、次に示す定数又は数値で設定します。

定数	数値	内容
dbpTextFormat	0	TEXT 形式
dbpBiff4Format	1	Excel 4.0 形式
dbpOleBiff8Format	4	Excel 97-2000 ブック形式

使い方

- データの形式については、「付録 C DBPARTNER2 Client が扱うデータの形式」を参照してください。
- *QueryDefine.Execute* メソッドを実行する前に *CopyToClipboard* プロパティを使うと、エラーになります。
- *Cancel* メソッドを実行して処理を中止すると、*Cancel* メソッドまでのデータをクリップボードに出力します。
- *CopyToClipboard* メソッドがエラーになったときは、クリップボードの内容は保証されません。

Get メソッド

機能

サーバプログラムに対して 1 行分のデータを要求します。

形式

Object.Get

パラメタ

- *Object*

Result オブジェクトへの参照を表すオブジェクト式です。

使い方

- データを取得するときは、RecordData プロパティと FieldData プロパティを使います。
- *Result.Get* メソッドを実行した後は、EOF プロパティで取得状況を確認してください。
- *QueryDefine.Execute* メソッドを実行する前に *Result.Get* メソッドを使うと、エラーになります。
- 1 回目の *Result.Get* メソッドが成功した以降でデータの取得を中止するときは、Close メソッドを使ってください。Cancel メソッドでは、データの取得を中止できません。

SaveToFile メソッド

機能

抽出データを、設定した形式でファイルに出力します。

形式

```
Object. SaveToFile FileName, Type
```

パラメタ

- *Object*
Result オブジェクトへの参照を表すオブジェクト式です。
- *FileName*
抽出データを出力するファイル名を、1~256 バイトの文字列で設定します。値はクリアできません。
- *Type*
出力するデータの形式を、次に示す定数又は数値で設定します。

定数	数値	内容
dbpTextFormat, dbpTextFile	0	TEXT 形式
dbpBiff4Format, dbpBiff4File	1	Excel 4.0 形式
dbpCSVFile	2	CSV 形式
dbpDATFile	3	DAT 形式
dbpOleBiff8File	4	Excel オブジェクトで作成する Excel 97-2003 ブック形式 (XLS 形式)
dbpOleBiff12File	5	Excel オブジェクトで作成する Excel 2007 バイナリファイル形式 (XLSB 形式)
dbpOleXMLWorkbookFile	6	Excel オブジェクトで作成する Excel 2007XML ファイル形式 (XLSX 形式)

使い方

- データの形式については、「[付録 C DBPARTNER2 Client が扱うデータの形式](#)」を参照してください。
- `QueryDefine.Execute` メソッドを実行する前に `SaveToFile` プロパティを使うと、エラーになります。
- `Cancel` メソッドを実行して処理を中止すると、`Cancel` メソッドまでのデータをファイルに出力します。
- `SaveToFile` メソッドがエラーになったときは、出力されたデータの内容は保証されません。

7.4 TableList オブジェクトの詳細

TableList オブジェクトは、設定したデータソースに格納されている表一覧情報を取得したり、取得した結果を管理したりします。それぞれの表情情報は、プロパティによって返されます。

TableList オブジェクトのプロパティとメソッドの一覧を次に示します。

●TableList オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	取得した表情情報の個数を取得します。
CountLimit	○	○	取得したい表情情報の個数を設定します。
SelectDict	○	○	接続先をキーに、使用する見出し辞書を限定します。
Pattern	○	○	表情情報の絞り込み条件を文字列で設定します。
TableComment	—	○	設定した表の注釈を文字列で取得します。
TableCreateDate	—	○	設定した表の作成日時を文字列で取得します。
TableDataSource	—	○	設定した表の見出し辞書中の接続先を文字列で取得します。
TableDBMName	—	○	設定した表の DBM 名を文字列で取得します。
TableDeleteAuth	—	○	設定した表のデータに対する削除権限があるかどうかを取得します。
TableGroupName	—	○	設定した表のグループ名を文字列で取得します。
TableHeader	—	○	設定した表のラベルを、文字列で取得します。
TableInsertAuth	—	○	設定した表のデータに対する追加権限があるかどうかを取得します。
TableName	—	○	設定した表の表名を文字列で取得します。
TableOwner	—	○	設定した表の所有者を文字列で取得します。
TablePasswordMode	—	○	設定した表の表パスワードの種別を取得します。
TableSelectAuth	—	○	設定した表のデータに対する検索権限があるかどうかを取得します。
TableType	—	○	設定した表の属性を文字列で取得します。
TableUpdateAuth	—	○	設定した表のデータに対する変更権限があるかどうかを取得します。

●TableList オブジェクトのメソッドの一覧

メソッド	説明
Get	表情情報の取得を要求します。

Count プロパティ

機能

取得した表情情報の個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。

使い方

- *TableList.Get* メソッドを実行する前は、0 を返します。

CountLimit プロパティ

機能

取得する表情情報の最大数を、数値で設定します。このプロパティは、値も取得できます。

形式

```
Object.CountLimit [= max]
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。
- *max*
取得する表の制限数を、次に示す数値で設定します。

数値	内容
0	すべての表の情報を取得します。
1~8192	設定した数だけ、表の情報を取得します。

使い方

- CountLimit プロパティの値は、*TableList.Get* メソッドを実行するときに使われます。*TableList.Get* メソッドを実行した後で CountLimit プロパティの値を変更しても、*TableList.Get* メソッドの結果を変えることはできません。

SelectDict プロパティ

機能

接続先をキーに、使用する見出し辞書を限定します。このプロパティは、値の取得もできます。

形式

```
Object. SelectDict = DataSource
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。
- *DataSource*
接続先を 0~32 バイトの文字列で設定します。0 バイトの文字列も設定できます。

使い方

- 接続先には、*TableList.TableDataSource* プロパティと同じ文字列を設定してください。
- 接続先が見出し辞書にない場合、*TableList.Count* プロパティが 0 となります。

Pattern プロパティ

機能

表情情報を取得するときの絞り込み条件を、文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object. Pattern [= text]
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。

- *text*

絞り込み条件を、0~128バイトの文字列で設定します。文字列には、ワイルドカード (? , *) 文字を含めることができます。値をクリアするときは、長さ0の文字列を設定します。絞り込み条件は、次の書式で設定してください。

- 所有者名.表名
- 表名

使い方

- 値がクリアされているときは、*.* が仮定されます。
- 「?」(クエスチョンマーク文字)は何らかの単一文字を示します。
- 「*」(アスタリスク文字)は何らかのゼロ個以上の文字のつながりを示します。
- 使える絞り込み条件を次に示します。
 - 前方一致 (例: ABC*) 文字列の先頭が ABC であり、以降の文字は任意です。
 - 後方一致 (例: *ABC) 文字列の最後が ABC であり、以前の文字は任意です。
 - 任意一致 (例: *ABC*) 文字列中の任意の部分が ABC です。
 - 完全一致 (例: ABC) 文字列が ABC と同じです。
 - 部分一致 (例: ?ABC?) 5文字の文字列のうち、2~4文字目が ABC です。
(例: A?B*C) 文字列の先頭が A, 3文字目が B, 最後が C です。
- Pattern プロパティの値は、*TableList.Get* メソッドを実行するときに使われます。*TableList.Get* メソッドを実行した後で Pattern プロパティの値を変更しても、*TableList.Get* メソッドの結果を変えることはできません。

TableComment プロパティ

機能

表番号で設定した表の注釈を、文字列で取得します。

形式

```
Object.TableComment (Index)
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。
- *Index*

目的の表の表番号を、数値で設定します。番号は、1～*TableList.Count* プロパティの範囲内で設定します。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。
- 接続するサーバプログラムやデータベースによって、返される文字列が異なります。返される文字列については、「[8.4 データベース別の規則](#)」を参照してください。

TableCreateDate プロパティ

機能

表番号で指定した表の作成日付を、文字列で取得します。

形式

```
Object. TableCreateDate (Index)
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的の表の表番号を、数値で設定します。番号は、1～*TableList.Count* プロパティの範囲内で設定します。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。
- 接続するサーバプログラムやデータベースによって、返される文字列が異なります。返される文字列については、「[8.4 データベース別の規則](#)」を参照してください。

TableDataSource プロパティ

機能

表番号で設定した表の見出し辞書の中の接続先を、文字列で取得します。

形式

Object. TableDataSource (Index)

パラメタ

- *Object*

TableList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の表の表番号を、数値で設定します。番号は、1～*TableList.Count* プロパティの範囲内で設定します。

使い方

- 表番号は、*TableList.Get* プロパティを発行した後、*TableList.Count* プロパティで取得した表の数の範囲内で設定してください。

TableDBMName プロパティ

機能

表番号で指定した表の DBM 名を、文字列で取得します。

形式

Object. TableDBMName (Index)

パラメタ

- *Object*

TableList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の表の表番号を、数値で設定します。番号は、1～*TableList.Count* プロパティの範囲内で設定します。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

TableDeleteAuth プロパティ

機能

表番号で指定した表のデータに対して削除権限があるかどうかを、BOOLEAN 値で取得します。

形式

```
Object.TableDeleteAuth (Index)
```

パラメタ

- *Object*

TableList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の表の表番号を、数値で設定します。番号は、1~*TableList.Count* プロパティの範囲内で設定します。

戻り値

定数	内容
TRUE (真)	データを削除する権限があります。
FALSE (偽)	データを削除する権限がありません。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。

TableGroupName プロパティ

機能

表番号で指定した表のグループ名を、文字列で取得します。

形式

```
Object.TableGroupName(Index)
```

パラメタ

- *Object*

TableList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の表を数値で設定します。番号は、1～*TableList.Count* プロパティの範囲内で設定します。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。
- *TableDefSource* プロパティに *dbpDICT* を設定している場合、見出し辞書のグループ名を返します。
- *TableDefSource* プロパティに *dbpDBDICT* を設定している場合、見出し辞書の中に異なる接続先で同一の表名があるときは *TableList.SelectDict* プロパティで接続先を設定してください。
TableList.SelectDict プロパティで接続先を設定しないと、正しいラベルが取得できないことがあります。

TableHeader プロパティ

機能

表番号で設定した表のラベルを、文字列で取得します。

形式

Object. *TableHeader* (*Index*)

パラメタ

- *Object*

TableList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の表の表番号を、数値で設定します。番号は、1～*TableList.Count* プロパティの範囲内で設定します。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。
- *TableDefSource* プロパティに *dbpDICT*、又は *dbpDBDICT* を設定している場合、見出し辞書のラベルを返します。
- *TableDefSource* プロパティに *dbpDBDICT* を設定している場合、見出し辞書の中に異なる接続先で同一の表名があるときは *TableList.SelectDict* プロパティで接続先を設定してください。
TableList.SelectDict プロパティで接続先を設定しないと、正しいラベルが取得できないことがあります。

- 接続するサーバプログラムやデータベースによって、返される文字列が異なります。返される文字列については、「8.4 データベース別の規則」を参照してください。

TableInsertAuth プロパティ

機能

表番号で指定した表のデータに対して追加権限があるかどうかを、BOOLEAN 値で取得します。

形式

Object. TableInsertAuth (*Index*)

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的の表の表番号を、数値で設定します。番号は、1 ~ *TableList.Count* プロパティの範囲内で設定します。

戻り値

定数	内容
TRUE (真)	データを追加する権限があります。
FALSE (偽)	データを追加する権限がありません。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。

TableName プロパティ

機能

表番号で設定した表の表名を、文字列で取得します。

形式

Object. TableName (*Index*)

パラメタ

- *Object*

TableList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の表の表番号を、数値で設定します。番号は、1～*TableList.Count* プロパティの範囲内で設定します。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。
- 接続するサーバプログラムやデータベースによって、返される文字列が異なります。返される文字列については、「[8.4 データベース別の規則](#)」を参照してください。

TableOwner プロパティ

機能

表番号で設定した表の所有者を、文字列で取得します。

形式

```
Object.TableOwner (Index)
```

パラメタ

- *Object*

TableList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の表の表番号を、数値で設定します。番号は、1～*TableList.Count* プロパティの範囲内で設定します。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。
- 接続するサーバプログラムやデータベースによって、返される文字列が異なります。返される文字列については、「[8.4 データベース別の規則](#)」を参照してください。

TablePasswordMode プロパティ

機能

表番号で指定した表の表パスワードの種別を、数値で取得します。

形式

Object. TablePasswordMode (*Index*)

パラメタ

- *Object*

TableList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の表の表番号を、数値で設定します。番号は、1 ~ *TableList.Count* プロパティの範囲内で設定します。

戻り値

次に示す定数又は数値で取得します。

定数	数値	内容
dbpNoPassword	0	表パスワードは設定されていません。
dbpSPassword	1	参照パスワードが設定されています。
dbpUPassword	2	更新パスワードが設定されています。
dbpSUPassword	3	参照パスワードと更新パスワードが設定されています。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。
- 表パスワードの種別は、実際に表に設定されているパスワードと、表に対する権限で決まります。パスワードと権限の関係と返される表パスワードの種別の対応を次に示します。

パスワード設定状況	参照権限がある表	更新権限がある表	参照／更新の両方の権限がある表
パスワード設定なし	dbpNoPassword	dbpNoPassword	dbpNoPassword
参照パスワードだけ設定	dbpSPassword	dbpNoPassword	dbpSPassword
更新パスワードだけ設定	dbpNoPassword	dbpUPassword	dbpUPassword
両方のパスワードが設定	dbpSPassword	dbpUPassword	dbpSUPassword

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

TableSelectAuth プロパティ

機能

表番号で指定した表のデータに対して検索権限があるかどうかを、BOOLEAN 値で取得します。

形式

```
Object.TableSelectAuth (Index)
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的の表の表番号を、数値で設定します。番号は、1 ~ *TableList.Count* プロパティの範囲内で設定します。

戻り値

定数	内容
TRUE (真)	データを抽出する権限があります。
FALSE (偽)	データを抽出する権限がありません。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。

TableType プロパティ

機能

表番号で設定した表の属性を、文字列で取得します。

形式

```
Object.TableType (Index)
```

パラメタ

- *Object*

TableList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の表の表番号を、数値で設定します。番号は、1～*TableList.Count* プロパティの範囲内で設定します。

使い方

- 表番号は、*TableList.Get* メソッドを実行した後で、*TableList.Count* プロパティで得られた表数の範囲内で設定してください。
- 表属性には、"TABLE"又は"VIEW"の表属性があります。
- 属性が "TABLE"又は"VIEW"以外のはきは、1文字の空白文字が返されます。
- VOSK RDB 編成ファイルの併合論理ファイル、連鎖論理ファイル、射影論理ファイル、結合論理ファイルは、"VIEW"として返されます。

TableUpdateAuth プロパティ

機能

表番号で指定した表のデータに対して変更権限があるかどうかを、BOOLEAN 値で取得します。

形式

```
Object.TableUpdateAuth (Index)
```

パラメタ

- *Object*

TableList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の表の表番号を、数値で設定します。番号は、1～*TableList.Count* プロパティの範囲内で設定します。

戻り値

定数	内容
TRUE (真)	データを変更する権限があります。
FALSE (偽)	データを変更する権限がありません。

使い方

- 表番号は、*TableList*.Get メソッドを実行した後、*TableList*.Count プロパティで得られた表数の範囲内で設定してください。

Get メソッド

機能

サーバプログラムに定義されている表情情報を要求します。

形式

```
Object. Get
```

パラメタ

- *Object*
TableList オブジェクトへの参照を表すオブジェクト式です。

使い方

- このメソッドを実行すると、前回 *TableList*.Get メソッドで取得した表の情報は上書きされます。

7.5 ColumnList オブジェクトの詳細

ColumnList オブジェクトは、設定された表名に定義されている列一覧情報を取得したり、取得した結果を管理したりします。それぞれの列情報は、プロパティによって返されます。

ColumnList オブジェクトのプロパティとメソッドの一覧を次に示します。

●ColumnList オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
ColumnComment	—	○	設定した列の注釈を文字列で取得します。
ColumnEditPattern	—	○	設定した列の編集形式を文字列で取得します。
ColumnHeader	—	○	設定した列のラベルを文字列で取得します。
ColumnLength	—	○	設定した列の定義長を数値で取得します。
ColumnName	—	○	設定した列の列名を文字列で取得します。
ColumnNumber	—	○	設定した列の列番号を文字列で取得します。
ColumnScale	—	○	設定した列の小数点以下のけた数を数値で取得します。
ColumnType	—	○	設定した列の属性を取得します。
Count	—	○	取得した列の個数を取得します。
CountLimit	○	○	列情報を取得する個数を設定します。
SelectDict	○	○	接続先をキーに使用する見出し辞書を限定します。
TableName	○	○	列情報を取得したい表名を、文字列で設定します。
TablePassword	○	○	表に定義されているパスワードを設定します。

●ColumnList オブジェクトのメソッドの一覧

メソッド	説明
Get	列情報の取得を要求します。

ColumnComment プロパティ

機能

列番号で設定した列の注釈を、文字列で取得します。

形式

Object. ColumnComment (Index)

パラメタ

- *Object*

ColumnList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1~*ColumnList.Count* プロパティの範囲内で設定します。

使い方

- 列番号は、*ColumnList.Get* メソッドを実行した後、*ColumnList.Count* プロパティで得られた列数の範囲内で設定してください。
- 接続するサーバプログラムやデータベースによって、返される文字列が異なります。返される文字列については、「8.4 データベース別の規則」を参照してください。

ColumnEditPattern プロパティ

機能

列番号で指定した列の編集形式を、文字列で取得します。

形式

Object.ColumnEditPattern (Index)

パラメタ

- *Object*

ColumnList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1~*ColumnList.Count* プロパティの範囲内で設定します。

使い方

- 列番号は、*ColumnList.Get* メソッドを実行した後、*ColumnList.Count* プロパティで得られた列数の範囲内で設定してください。
- 取得した編集形式の内容については、ACE3 のマニュアルを参照してください。

ColumnHeader プロパティ

機能

列番号で設定した列のラベルを、文字列で取得します。

形式

```
Object.ColumnHeader (Index)
```

パラメタ

- *Object*

ColumnList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1~*ColumnList.Count* プロパティの範囲内で設定します。

使い方

- 列番号は、*ColumnList.Get* メソッドを実行した後、*ColumnList.Count* プロパティで得られた列数の範囲内で設定してください。
- *TableDefSource* プロパティに *dbpDICT*、又は *dbpDBDICT* を設定している場合、見出し辞書のラベルを返します。
- *TableDefSource* プロパティに *dbpDBDICT* を設定している場合、見出し辞書の中に異なる接続先で同一の表名があるときは *TableList.SelectDict* プロパティで接続先を設定してください。
TableList.SelectDict プロパティで接続先を設定しないと、正しいラベルが取得できないことがあります。
- 接続するサーバプログラムやデータベースによって、返される文字列が異なります。返される文字列については、「[8.4 データベース別の規則](#)」を参照してください。

ColumnLength プロパティ

機能

列番号で設定した列の定義長を、数値で取得します。

形式

```
Object.ColumnLength (Index)
```

パラメタ

- *Object*

ColumnList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1~*ColumnList.Count* プロパティの範囲内で設定します。

使い方

- 列番号は、*ColumnList.Get* メソッドを実行した後、*ColumnList.Count* プロパティで得られた列数の範囲内で設定してください。

ColumnName プロパティ

機能

列番号で設定した列の列名を、文字列で取得します。

形式

```
Object.ColumnName (Index)
```

パラメタ

- *Object*

ColumnList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の列の列番号を、数値で設定します。番号は、1~*ColumnList.Count* プロパティの範囲内で設定します。

使い方

- 列番号は、*ColumnList.Get* メソッドを実行した後、*ColumnList.Count* プロパティで得られた列数の範囲内で設定してください。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。
- 接続するサーバプログラムやデータベースによって、返される文字列が異なります。返される文字列については、「[8.4 データベース別の規則](#)」を参照してください。

ColumnNumber プロパティ

機能

列番号で設定した列の列番号を、数値で取得します。

形式

```
Object.ColumnNumber (Index)
```

パラメタ

- *Object*
ColumnList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的の列の列番号を、数値で設定します。番号は、1~*ColumnList.Count* プロパティの範囲内で設定します。

使い方

- 繰り返し列の列番号は、同じ番号になります。
- 列番号は、*ColumnList.Get* メソッドを実行した後、*ColumnList.Count* プロパティで得られた列数の範囲内で設定してください。

ColumnScale プロパティ

機能

列番号で設定した列の小数点以下のけた数を、数値で取得します。

形式

```
Object.ColumnScale (Index)
```

パラメタ

- *Object*
ColumnList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的の列の列番号を、数値で設定します。番号は、1~*ColumnList.Count* プロパティの範囲内で設定します。

使い方

- 列番号は、*ColumnList*.Get メソッドを実行した後、*ColumnList.Count* プロパティで得られた列数の範囲内で設定してください。

ColumnType プロパティ

機能

列番号で設定した列の属性を、数値で取得します。

形式

```
Object.ColumnType (Index)
```

パラメタ

- *Object*
ColumnList オブジェクトへの参照を表すオブジェクト式です。
- *Index*
目的の列の列番号を、数値で設定します。番号は、1~*ColumnList.Count* プロパティの範囲内で設定します。

使い方

- 属性については、「付録 E 列の属性」を参照してください。
- 列番号は、*ColumnList*.Get メソッドを実行した後、*ColumnList.Count* プロパティで得られた列数の範囲内で設定してください。

Count プロパティ

機能

取得した列情報の個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

ColumnList オブジェクトへの参照を表すオブジェクト式です。

使い方

- *ColumnList.Get* メソッドを実行する前は、0 を返します。

CountLimit プロパティ

機能

取得する列情報の最大数を、数値で設定します。このプロパティは、値も取得できます。

形式

```
Object.CountLimit [= max]
```

パラメタ

- *Object*
ColumnList オブジェクトへの参照を表すオブジェクト式です。
- *max*
取得する列数を、次に示す数値で設定します。

数値	内容
0	すべての列を取得します。
1~8192	設定した数だけ、列の情報を取得します。

使い方

- CountLimit プロパティの値は、*ColumnList.Get* メソッドを実行するときに使われます。*ColumnList.Get* メソッドを実行した後で CountLimit プロパティの値を変更しても、*ColumnList.Get* メソッドの結果を変えることはできません。

SelectDict プロパティ

機能

接続先をキーに使用する見出し辞書を限定します。このプロパティは、値の取得もできます。

形式

```
Object.SelectDict = DataSource
```

パラメタ

- *Object*
ColumnList オブジェクトへの参照を表すオブジェクト式です。
- *DataSource*
接続先を 0~32 バイトの文字列で設定します。0 バイトの文字列も設定できます。

使い方

- 接続先には、*TableList.TableDataSource* プロパティと同じ文字列を設定してください。
- 接続先が見出し辞書にない場合、*TableList.Count* プロパティが 0 となります。

TableName プロパティ

機能

列情報を取得したい表名を、文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object.TableName [= text]
```

パラメタ

- *Object*
ColumnList オブジェクトへの参照を表すオブジェクト式です。
- *text*
表名を、1~128 バイトの文字列で設定します。長さ 0 の文字列は設定できません。

使い方

- 設定する表名の書式については、「[8.1.2 表名を設定するときの規則](#)」を参照してください。
- *TableName* プロパティの値は、*ColumnList.Get* メソッドを実行するときに使われます。*ColumnList.Get* メソッドを実行した後で *TableName* プロパティの値を変更しても、*ColumnList.Get* メソッドの結果を変えることはできません。

TablePassword プロパティ

機能

表にアクセスするパスワードを設定します。このプロパティは、値も取得できます。

形式

```
Object. TablePassword [= Pass]
```

パラメタ

- *Object*

ColumnList オブジェクトへの参照を表すオブジェクト式です。

- *Pass*

表パスワードを、0~8 バイトの文字列で設定します。値をクリアするときは、長さ 0 の文字列を設定します。

使い方

- 半角文字の「()」, 「[]」, 又は全角文字は設定できません。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

Get メソッド

機能

ColumnList.TableName プロパティで設定した表に含まれる列情報の取得を要求します。

形式

```
Object. Get
```

パラメタ

- *Object*

ColumnList オブジェクトへの参照を表すオブジェクト式です。

使い方

- このメソッドを実行すると、前回 *ColumnList.Get* メソッドで取得した列の情報は上書きされます。

7.6 QueryDefine オブジェクトの詳細

QueryDefine オブジェクトは、抽出処理を実行するオブジェクトです。抽出処理の結果として出力する抽出データは、Result オブジェクトで取得します。

QueryDefine オブジェクトのプロパティとメソッドの一覧を次に示します。

●QueryDefine オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Exclusive	○	○	抽出処理を実行するときの排他モードを、文字列で設定します。

●QueryDefine オブジェクトのメソッドの一覧

メソッド	説明
Execute	抽出処理を実行します。
QueryCount	抽出処理の結果として出力される行数を、数値で取得します。
RemoveAll	すべての Table, Column, Condition オブジェクトをクリアします。

Exclusive プロパティ

機能

抽出する表をほかのユーザが使っているときの動作を、文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object.Exclusive [= String]
```

パラメタ

- *Object*

QueryDefine オブジェクトへの参照を表すオブジェクト式です。

- *String*

排他オプションを、次に示す 0~64 バイトの文字列で設定します。

文字列	内容
長さ 0 バイトの文字列	"WITH SHARE LOCK"の説明を参照してください。
"WITH SHARE LOCK"	一度検索した内容を、トランザクション終了まで、ほかのユーザからの参照は許しますが、更新は許しません。

文字列	内容
"WITH EXCLUSIVE LOCK"	一度検索した内容を、トランザクション終了まで、ほかのユーザからの参照も更新も許しません。
"WITHOUT LOCK WAIT"	一度検索した内容を、トランザクション終了まで、排他制御しません。
"WITHOUT LOCK NOWAIT"	ほかのユーザが更新中のデータでも、更新の完了を待たないで参照して、一度検索した内容をトランザクション終了まで排他制御しません。
"WITH ROLLBACK"	検索の対象になる表をほかのユーザが使っているときは、トランザクションを取り消して無効にします。

排他オプションの文字列については、各データベースのマニュアルを参照してください。

使い方

- Exclusive プロパティの値は、*QueryDefine.Execute* メソッドを実行するときに使われます。*QueryDefine.Execute* メソッドを実行した後で Exclusive プロパティの値を変更しても、*QueryDefine.Execute* メソッドの結果を変えることはできません。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

Execute メソッド

機能

データベースに抽出条件を渡して、検索を実行します。

形式

```
Object.Execute
```

パラメタ

- *Object*
QueryDefine オブジェクトへの参照を表すオブジェクト式です。

使い方

- 検索したデータは、Result オブジェクトを使って取得します。
- 結果取得を完了させないで再び *QueryDefine.Execute* メソッドを実行すると、エラーになります。
- 検索した結果、抽出データがないときは、エラーが返されます。

QueryCount メソッド

機能

抽出条件に一致するデータの件数を、数値で取得します。

形式

```
Object. QueryCount
```

パラメタ

- *Object*
QueryDefine オブジェクトへの参照を表すオブジェクト式です。

使い方

- QueryCountLimit プロパティの設定内容には依存しません。
- FunctionMode プロパティで抽出列に集合関数を設定しているときは、件数を取得しようとするときエラーになります。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

RemoveAll メソッド

機能

すべての Table, Column, Condition オブジェクトをクリアします。

形式

```
Object. RemoveAll
```

パラメタ

- *Object*
QueryDefine オブジェクトへの参照を表すオブジェクト式です。

7.7 Tables コレクションオブジェクトの詳細

Tables コレクションオブジェクトは、Table オブジェクトをリストオブジェクトとして管理します。

Tables コレクションオブジェクトのプロパティとメソッドの一覧を次に示します。

●Tables コレクションオブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	抽出表の個数を数値で取得します。

●Tables コレクションオブジェクトのメソッドの一覧

メソッド	説明
Add	抽出表を文字列で追加します。
Item	Table オブジェクトのオブジェクト式を取得します。
RemoveAll	すべての Table オブジェクトをクリアします。

Count プロパティ

機能

抽出条件として設定されている表の個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

Tables コレクションオブジェクトへの参照を表すオブジェクト式です。

Add メソッド

機能

抽出条件にする表名を、文字列で追加します。

形式

```
Object.Add (String)
```

パラメタ

- *Object*

Tables コレクションオブジェクトへの参照を表すオブジェクト式です。

- *String*

抽出条件にする表の表名を、1~128 バイトの文字列で設定します。長さ 0 の文字列は設定できません。

使い方

- 設定する表名の書式については、「[8.1.2 表名を設定するときの規則](#)」を参照してください。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

Item メソッド

機能

Table オブジェクトのオブジェクト式を取得します。

形式

```
Object.Item (ObjectNumber)
```

パラメタ

- *Object*

Tables コレクションオブジェクトへの参照を表すオブジェクト式です。

- *ObjectNumber*

目的の抽出表番号 (Table オブジェクトの番号) を、数値で設定します。番号は、1~*Tables.Count* プロパティの範囲内で設定します。

RemoveAll メソッド

機能

すべての Table オブジェクトをクリアします。

形式

```
Object.RemoveAll
```

パラメタ

- *Object*

Tables コレクションオブジェクトへの参照を表すオブジェクト式です。

7.8 Table オブジェクトの詳細

Table オブジェクトは、抽出表を管理するオブジェクトです。

Table オブジェクトのプロパティの一覧を次に示します。Table オブジェクトのメソッドはありません。

●Table オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
TableName	○	○	抽出表として設定されている文字列を設定します。
TablePassword	○	○	表に定義されているパスワードを設定します。

TableName プロパティ

機能

抽出条件にする表名を、文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object.TableName [= String]
```

パラメタ

- *Object*

Table オブジェクトへの参照を表すオブジェクト式です。

- *String*

抽出する表の表名を、1~128 バイトの文字列で設定します。長さ 0 の文字列は設定できません。

使い方

- 設定する表名の書式については、「[8.1.2 表名を設定するときの規則](#)」を参照してください。

TablePassword プロパティ

機能

表パスワードを、文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object.TablePassword [= Pass]
```

パラメタ

- *Object*

Table オブジェクトへの参照を表すオブジェクト式です。

- *Pass*

更新のパスワードを、0~8 バイトの文字列で設定します。値をクリアするときは、長さ 0 の文字列を設定します。

使い方

- 半角文字の「(」, 「)」, 又は全角文字は指定できません。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

7.9 Columns コレクションオブジェクトの詳細

Columns コレクションオブジェクトは、Column オブジェクトをリストオブジェクトとして管理します。

Columns コレクションオブジェクトのプロパティとメソッドの一覧を次に示します。

●Columns コレクションオブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	抽出列の個数を数値で返します。

●Columns コレクションオブジェクトのメソッドの一覧

メソッド	説明
Add	抽出列を文字列で追加します。
Item	Column オブジェクトのオブジェクト式を取得します。
RemoveAll	すべての Column オブジェクトをクリアします。

Count プロパティ

機能

抽出列の個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

Columns コレクションオブジェクトへの参照を表すオブジェクト式です。

Add メソッド

機能

抽出列の列名を、文字列で追加します。

形式

```
Object.Add (String)
```

パラメタ

- *Object*

Columns コレクションオブジェクトへの参照を表すオブジェクト式です。

- *String*

抽出列を、1~256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。

使い方

- 設定する列名の書式については、「[8.1.3 列名を設定するときの規則](#)」を参照してください。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

Item メソッド

機能

Column オブジェクトのオブジェクト式を取得します。

形式

```
Object.Item (ObjectNumber)
```

パラメタ

- *Object*

Columns コレクションオブジェクトへの参照を表すオブジェクト式です。

- *ObjectNumber*

目的の抽出列の番号 (Column オブジェクトの番号) を、数値で設定します。番号は、1~*Columns.Count* プロパティの範囲内で設定します。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

RemoveAll メソッド

機能

すべての Column オブジェクトをクリアします。

形式

```
Object.RemoveAll
```

パラメタ

- *Object*

Columns コレクションオブジェクトへの参照を表すオブジェクト式です。

7.10 Column オブジェクトの詳細

Column オブジェクトは、抽出列を管理するオブジェクトです。

Column オブジェクトのプロパティの一覧を次に示します。Column オブジェクトのメソッドはありません。

●Column オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
ColumnName	○	○	抽出列の列名を設定します。
FunctionMode	○	○	抽出列に集合関数を設定します。
SortMode	○	○	抽出列にソートを設定します。
SortPriority	○	○	ソートを設定した列の優先順位を設定します。

ColumnName プロパティ

機能

抽出列とする列名を、文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object.ColumnName [= String]
```

パラメタ

- *Object*
Column オブジェクトへの参照を表すオブジェクト式です。
- *String*
抽出列名を、1～256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。

使い方

- 設定する列名の書式については、「[8.1.3 列名を設定するときの規則](#)」を参照してください。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

FunctionMode プロパティ

機能

抽出列に集合関数を設定します。このプロパティは、値も取得できます。

形式

```
Object.FunctionMode [= mode]
```

パラメタ

- *Object*
Column オブジェクトへの参照を表すオブジェクト式です。
- *mode*
集合関数を、次に示す定数又は数値で設定します。

定数	数値	内容
<u>dbpNormal</u> , <u>dbpNoFunction</u>	0	集合関数なし
dbpGroup	1	重複排除
dbpMaximum	2	最大
dbpMinimum	3	最小
dbpCount	4	件数
dbpAverage	5	平均
dbpSum	6	合計

使い方

- 抽出列のどれかに集合関数を付けると、集合関数を設定していない抽出列に「重複排除」が仮定されて、抽出処理が実行されます。ただし、集合関数を設定していない抽出列の集合関数の値は変更されません。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

SortMode プロパティ

機能

抽出列をソートするかどうかを設定します。このプロパティは、値も取得できます。

形式

```
Object.SortMode [= mode]
```

パラメタ

- *Object*
Column オブジェクトへの参照を表すオブジェクト式です。
- *mode*
ソートするかどうかを、次に示す定数又は数値で設定します。

定数	数値	内容
<code>dbpNormal</code> , <code>dbpNoSort</code>	0	データをソートしません。
<code>dbpAscend</code>	1	データを昇順にソートします。
<code>dbpDescend</code>	2	データを降順にソートします。

使い方

- 複数の列にソートを設定したときは、優先するソートを決める必要があります。どの列のソートが優先されるかは、SortMode プロパティの実行順に従います。つまり、最初に設定されたソートが最優先になります。二つめ以降を設定するときには、現在設定されているソートの中で最も低い優先順位の次の順位が割り当てられます。
- 優先順位を参照又は変更したいときは、SortPriority プロパティを使ってください。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

SortPriority プロパティ

機能

ソートが設定されている抽出列の優先順位を、数値で設定します。このプロパティは、値も取得できます。

形式

```
Object.SortPriority [= number]
```

パラメタ

- *Object*
Column オブジェクトへの参照を表すオブジェクト式です。
- *number*

ソートの優先順位を、1～255 の値で設定します。

使い方

- SortPriority プロパティで優先順位を設定できる列は、SortMode プロパティでソートを設定した列だけです。ソートが設定されていない列には設定できません。
- ソート優先順位の値が小さいほど、優先順位が高くなります。
- 優先順位の値を重複させると、同一番号内での順位は保証されません。
- 優先順位は、連続している値を設定する必要はありません。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

7.11 Conditions コレクションオブジェクトの詳細

Conditions コレクションオブジェクトは、Condition オブジェクトをリストオブジェクトとして管理します。

Conditions コレクションオブジェクトのプロパティとメソッドの一覧を次に示します。

●Conditions コレクションオブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	抽出条件式の個数を取得します。
Relation	○	○	条件関連を文字列で設定します。

●Conditions コレクションオブジェクトのメソッドの一覧

メソッド	説明
Add	抽出条件式を文字列で追加します。
Item	Condition オブジェクトのオブジェクト式を取得します。
RemoveAll	すべての Condition オブジェクトをクリアします。

Count プロパティ

機能

抽出条件式の個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

Conditions コレクションオブジェクトへの参照を表すオブジェクト式です。

Relation プロパティ

機能

複数の抽出条件式の関連を、文字列で設定します。このプロパティは、値も取得できます。

形式

Object. Relation [= *string*]

パラメタ

- *Object*

Conditions コレクションオブジェクトへの参照を表すオブジェクト式です。

- *string*

抽出条件式を関連させる条件を、0~1,024 バイトの文字列で設定します。値をクリアするときは、長さ 0 の文字列を設定します。

使い方

- 抽出条件式の関係は、抽出条件式番号を使って設定します。値をクリアすると、すべての抽出条件式は AND で関連付けられます。

- 条件はすべて半角文字で設定して、AND、OR、及び NOT の前後にはスペースを入力してください。

AND, and : 論理積を示します。

「1 AND 2」は、条件式 1 と条件式 2 の両方を満たすデータの抽出を設定します。

OR, or : 論理和を示します。

「1 OR 2」は、条件式 1 か条件式 2 のどちらかを満たすデータの抽出を設定します。

NOT, not : 否定を示します。

「NOT 1」は、条件式 1 を満たさないデータの抽出を設定します。

() : 括弧内の条件を優先します。

「1 AND (2 OR 3)」は、条件式 2 か条件式 3 のどちらかを満たして、かつ条件式 1 を満たすデータの抽出を設定します。

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

Add メソッド

機能

抽出データを絞り込むための条件式を、文字列で追加します。

形式

Object. Add (*String*)

パラメタ

- *Object*

Conditions コレクションオブジェクトへの参照を表すオブジェクト式です。

- *String*

抽出条件式を、1~256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。

使い方

- 代表的な抽出条件式を次に示します。

列名 C1 のデータが NULL 値の行を抽出する (NULL 述語)

```
"C1" IS NULL
```

列名 C2 のデータが 10, 20, 30 の行を抽出する (IN 述語)

```
"C2" IN (10, 20, 30)
```

列名 C3 のデータが ABC から始まる行を抽出する (LIKE 述語)

```
"C3" LIKE 'ABC%'
```

列名 C4 のデータが 10 から 50 の範囲内にある行を抽出する (BETWEEN 述語)

```
"C4" BETWEEN 10 AND 50
```

列名 C5 のデータが 100 の行を抽出する (比較条件)

```
"C5" = 100
```

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

Item メソッド

機能

Condition オブジェクトのオブジェクト式を取得します。

形式

```
Object.Item (ObjectNumber)
```

パラメタ

- *Object*

Conditions コレクションオブジェクトへの参照を表すオブジェクト式です。

- *ObjectNumber*

目的の抽出条件式の番号 (Condition オブジェクトの番号) を、数値で設定します。番号は、1～*Conditions.Count* プロパティの範囲内で設定します。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

RemoveAll メソッド

機能

すべての Condition オブジェクトをクリアします。

形式

```
Object.RemoveAll
```

パラメタ

- *Object*
Conditions コレクションオブジェクトへの参照を表すオブジェクト式です。

7.12 Condition オブジェクトの詳細

Condition オブジェクトは、Select 文を生成するための情報になる条件情報を管理します。条件一つにつき、一つの Condition オブジェクトが対応します。

Condition オブジェクトのプロパティの一覧を次に示します。Condition オブジェクトのメソッドはありません。

●Condition オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Text	<input type="radio"/>	<input type="radio"/>	抽出条件式を設定します。

Text プロパティ

機能

抽出データを絞り込むための条件式を、文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object.Text [= String]
```

パラメタ

- *Object*
Condition オブジェクトへの参照を表すオブジェクト式です。
- *String*
抽出の条件式を、1~256 バイトの文字列で設定します。長さ 0 の文字列は設定できません。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

7.13 UpdateDefine オブジェクトの詳細

UpdateDefine オブジェクトは、更新処理を実行するオブジェクトです。

UpdateDefine オブジェクトのプロパティとメソッドの一覧を次に示します。

●UpdateDefine オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
ErrorInformation	—	○	更新処理でエラーになったオブジェクトの番号を取得します。
Exclusive	○	○	抽出処理を実行するときの排他モードを、文字列で設定します。
ExecutedRowCount	—	○	更新処理の結果、サーバプログラムで更新した行数を取得します。
TableName	○	○	更新表を設定します。
TablePassword	○	○	表に定義されているパスワードを設定します。
UpdateMode	○	○	更新実行モードを設定します。

●UpdateDefine オブジェクトのメソッドの一覧

メソッド	説明
Execute	更新処理を実行します。
RemoveAll	すべての Update, Insert, Delete オブジェクトをクリアします。

ErrorInformation プロパティ

機能

更新処理でエラーになった Update, Insert, Delete オブジェクトの番号を、数値で取得します。

形式

```
Object.ErrorInformation
```

パラメタ

- *Object*

UpdateDefine オブジェクトへの参照を表すオブジェクト式です。

戻り値

値	内容
0	更新する表又は列として、存在しない表名又は列名を設定しています。

値	内容
1 以上	エラーになった Update, Insert, Delete オブジェクトの番号を示します。

使い方

- *UpdateDefine.Execute* メソッドを実行する前は、0 を返します。

Exclusive プロパティ

機能

更新する表をほかのユーザが使っているときの動作を、文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object.Exclusive [= String]
```

パラメタ

- *Object*
UpdateDefine オブジェクトへの参照を表すオブジェクト式です。
- *String*
排他オプションを、次に示す 0~64 バイトの文字列で設定します。

文字列	内容
長さ 0 バイトの文字列	更新対象の表をほかのユーザが使っているときは、使っているユーザのトランザクション終了を待ちます。
"WITH ROLLBACK"	更新対象の表をほかのユーザが使っているときは、トランザクションを取り消して無効にします。

排他オプションの文字列については、各データベースのマニュアルを参照してください。

使い方

- Exclusive プロパティの値は、*UpdateDefine.Execute* メソッドを実行するときに使われます。*UpdateDefine.Execute* メソッドを実行した後で Exclusive プロパティの値を変更しても、*UpdateDefine.Execute* メソッドの結果を変えることはできません。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「8.3 データベースと使えるプロパティ/メソッド」を参照してください。

ExecutedRowCount プロパティ

機能

`UpdateDefine.Execute` メソッドを実行 (Update, Insert, Delete) した結果、サーバプログラムが更新した行数を、数値で取得します。

形式

```
Object. ExecutedRowCount
```

パラメタ

- *Object*

UpdateDefine オブジェクトへの参照を表すオブジェクト式です。

使い方

- `UpdateDefine.Execute` メソッドを実行する前は、0 を返します。

TableName プロパティ

機能

更新する表名を、文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object. TableName [= string]
```

パラメタ

- *Object*

UpdateDefine オブジェクトへの参照を表すオブジェクト式です。

- *string*

更新する表名を、1~128 バイトの文字列で設定します。長さ 0 の文字列は、設定できません。

使い方

- 設定する表名の書式については、「[8.1.2 表名を設定するときの規則](#)」を参照してください。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

TablePassword プロパティ

機能

表にアクセスするためのパスワードを設定します。このプロパティは、値も取得できます。

形式

```
Object. TablePassword [= Pass]
```

パラメタ

- *Object*

UpdateDefine オブジェクトへの参照を表すオブジェクト式です。

- *Pass*

参照のパスワード又は更新のパスワードを、0~8 バイトの文字列で設定します。値をクリアするときは、長さ 0 の文字列を設定します。

使い方

- 半角文字の「(」, 「)」, 又は全角文字は設定できません。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

UpdateMode プロパティ

機能

更新処理を実行するときどのような SQL 文で更新するかを設定します。このプロパティは、値も取得できます。

形式

```
Object. UpdateMode [= mode]
```

パラメタ

- *Object*

UpdateDefine オブジェクトへの参照を表すオブジェクト式です。

- *mode*

更新を実行するモードを、次に示す定数又は数値で設定します。

定数	数値	内容
<code>dbpNormal</code>	0	変数 (?パラメタ) を使いません。
<code>dbpVariable</code>	1	変数 (?パラメタ) を使います。

使い方

- `dbpNormal` を設定すると、Update/Insert/Delete の各オブジェクト単位に SQL 文を作成して更新処理を実行します。
- `dbpVariable` を設定すると、Update/Insert/Delete の各オブジェクトの値の部分を変数にした SQL 文を作成して更新処理を実行します。`dbpVariable` を使うときは、更新する列の数や、列の設定順が一定でないと、エラーになります。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

Execute メソッド

機能

データベースに対して、データを更新します。

形式

Object.Execute (*Type*)

パラメタ

- *Object*
UpdateDefine オブジェクトへの参照を表すオブジェクト式です。
- *Type*
更新種別を、次に示す定数又は数値で設定します。

定数	数値	内容
<code>dbpUpdate</code>	0	変更処理を示します。
<code>dbpInsert</code>	1	追加処理を示します。
<code>dbpDelete</code>	2	削除処理を示します。

使い方

- `UpdateDefine.Execute` メソッドを実行するためには、次に示す条件があります。

変更処理の場合

一つ以上の Update オブジェクトが設定されていること

追加処理の場合

一つ以上の Insert オブジェクトが設定されていること

削除処理の場合

一つ以上の Delete オブジェクトが設定されていること

- トランザクション終了モードに `dbpManual` を設定しているときは、更新処理が終了した時点で必ず `Commit` メソッド又は `Rollback` メソッドを実行してください。更新処理がエラーになったときも、メソッドを実行する必要があります。`Commit` メソッド又は `Rollback` メソッドを実行し忘れると、データベースのデータを壊してしまうおそれがあります。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

RemoveAll メソッド

機能

Update, Insert, Delete オブジェクトをすべてクリアします。

形式

```
Object.RemoveAll
```

パラメタ

- *Object*

UpdateDefine オブジェクトへの参照を表すオブジェクト式です。

7.14 Updates コレクションオブジェクトの詳細

Updates コレクションオブジェクトは、Update オブジェクトをリストオブジェクトとして管理します。

Updates コレクションオブジェクトのプロパティとメソッドの一覧を次に示します。

●Updates コレクションオブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	Update オブジェクトの個数を数値で取得します。

●Updates コレクションオブジェクトのメソッドの一覧

メソッド	説明
Add	Update オブジェクトを生成します。
Item	Update オブジェクトのオブジェクト式を取得します。
RemoveAll	すべての Update オブジェクトを削除します。

Count プロパティ

機能

`Update.Add` メソッドで作成した Update オブジェクトの個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

Updates コレクションオブジェクトへの参照を表すオブジェクト式です。

Add メソッド

機能

Update オブジェクトを生成します。

形式

```
Object.Add
```

パラメタ

- *Object*

Updates コレクションオブジェクトへの参照を表すオブジェクト式です。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

Item メソッド

機能

Update オブジェクトのオブジェクト式を取得します。

形式

```
Object.Item (ObjectNumber)
```

パラメタ

- *Object*

Updates コレクションオブジェクトへの参照を表すオブジェクト式です。

- *ObjectNumber*

目的の Update オブジェクトの番号を、数値で設定します。番号は、1~*Updates.Count* プロパティの範囲内で設定します。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

RemoveAll メソッド

機能

すべての Update オブジェクトをクリアします。

形式

```
Object.RemoveAll
```

パラメタ

- *Object*

Updates コレクションオブジェクトへの参照を表すオブジェクト式です。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

7.15 Update オブジェクトの詳細

Update オブジェクトは、サーバプログラムのデータを変更するための情報を管理します。

Update オブジェクトのプロパティとメソッドの一覧を次に示します。

●Update オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Column	○	○	変更情報の列名を設定します。
KeyColumn	○	○	変更行条件の列名を設定します。
KeyCount	—	○	変更行条件の数を取得します。
KeyValue	○	○	変更行条件のデータを設定します。
UpdateValue	○	○	変更情報のデータを設定します。
UpdateValueCount	—	○	変更情報の数を取得します。

●Update オブジェクトのメソッドの一覧

メソッド	説明
AddKey	変更行条件を追加します。
AddUpdateValue	変更情報を追加します。

Column プロパティ

機能

変更情報の番号で設定した情報の列名を、文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object.Column (ValueNumber) [= Column]
```

パラメタ

- *Object*

Update オブジェクトへの参照を表すオブジェクト式です。

- *ValueNumber*

目的の変更情報を、数値で設定します。番号は、1~*Update.UpdateValueCount* プロパティの範囲内で設定します。

- *Column*

列名を、1~64バイトの文字列で設定します。長さ0の文字列は設定できません。

使い方

- 設定する列名の書式については、「[8.1.3 列名を設定するときの規則](#)」を参照してください。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

KeyColumn プロパティ

機能

変更行条件の番号で設定した条件の列名を、文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object.KeyColumn (KeyNumber) [= Column]
```

パラメタ

- *Object*
Update オブジェクトへの参照を表すオブジェクト式です。
- *KeyNumber*
目的の条件の変更行番号を、数値で設定します。番号は、1~*Update.KeyCount* プロパティの範囲内で設定します。
- *Column*
列名を、1~64バイトの文字列で設定します。長さ0の文字列は設定できません。

使い方

- 設定する列名の書式については、「[8.1.3 列名を設定するときの規則](#)」を参照してください。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

KeyCount プロパティ

機能

変更行条件の数を、数値で取得します。

形式

```
Object.KeyCount
```

パラメタ

- *Object*

Update オブジェクトへの参照を表すオブジェクト式です。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

KeyValue プロパティ

機能

変更行条件の番号で設定した条件のデータを設定します。このプロパティは、値も取得できます。

形式

```
Object.KeyValue (KeyNumber) [= Data]
```

パラメタ

- *Object*

Update オブジェクトへの参照を表すオブジェクト式です。

- *KeyNumber*

目的の変更行条件の番号を、数値で設定します。番号は、1~*Update.KeyCount* プロパティの範囲内で設定します。

- *Data*

条件のデータを、0~256 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが仮定されます。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

UpdateValue プロパティ

機能

AddUpdateValue メソッドで追加した変更情報のデータを設定します。このプロパティは、値も取得できません。

形式

```
Object.UpdateValue (ValueNumber) [= Data]
```

パラメタ

- *Object*

Update オブジェクトへの参照を表すオブジェクト式です。

- *ValueNumber*

目的の変更情報を、数値で設定します。番号は、1~*Update.UpdateValueCount* プロパティの範囲内で設定します。

- *Data*

データを、0~256 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが仮定されます。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

UpdateValueCount プロパティ

機能

変更情報の数を、数値で取得します。

形式

```
Object.UpdateValueCount
```

パラメタ

- *Object*

Update オブジェクトへの参照を表すオブジェクト式です。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

AddKey メソッド

機能

変更行条件を追加します。列名には、*Value* に設定するデータを格納している列名を設定します。データには、変更する行を特定するデータを設定します。

形式

```
Object.AddKey Column, Value
```

パラメタ

- *Object*
Update オブジェクトへの参照を表すオブジェクト式です。
- *Column*
列名を、1~64 バイトの文字列で設定します。長さ 0 の文字列は設定できません。
- *Value*
データを、0~256 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが仮定されます。

使い方

- 設定する列名の書式については、「[8.1.3 列名を設定するときの規則](#)」を参照してください。
- 一つの変更行条件で変更する行を特定できないときは、同一の Update オブジェクトに対して複数の変更行条件を設定してください。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

AddUpdateValue メソッド

機能

変更情報を追加します。列名には、*Value* に設定するデータを格納する列名を設定します。データには、新しく格納するデータを設定します。

形式

```
Object. AddUpdateValue Column, Value
```

パラメタ

- *Object*

Update オブジェクトへの参照を表すオブジェクト式です。

- *Column*

列名を、1~64 バイトの文字列で設定します。長さ 0 の文字列は設定できません。

- *Value*

データを、0~256 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが仮定されます。

使い方

- 設定する列名の書式については、「[8.1.3 列名を設定するときの規則](#)」を参照してください。
- 変更行条件で決めた行に対して、複数の列のデータを変更するときは、同じ Update オブジェクトに対して複数の変更情報を設定してください。
- AddUpdateValue メソッドで設定したデータを変更/取得するときは、UpdateValue プロパティを使います。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

7.16 Inserts コレクションオブジェクトの詳細

Inserts コレクションオブジェクトは、Insert オブジェクトをリストオブジェクトとして管理します。

Inserts コレクションオブジェクトのプロパティとメソッドの一覧を次に示します。

●Inserts コレクションオブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	Insert オブジェクトの個数を、数値で取得します。

●Inserts コレクションオブジェクトのメソッドの一覧

メソッド	説明
Add	Insert オブジェクトを追加します。
Item	Insert オブジェクトのオブジェクト式を取得します。
RemoveAll	すべての Insert オブジェクトをクリアします。

Count プロパティ

機能

Inserts.Add メソッドで作成した Insert オブジェクトの個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

Inserts コレクションオブジェクトへの参照を表すオブジェクト式です。

Add メソッド

機能

Insert オブジェクトを追加します。

形式

```
Object.Add
```

パラメタ

- *Object*

Inserts コレクションオブジェクトへの参照を表すオブジェクト式です。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

Item メソッド

機能

Insert オブジェクトのオブジェクト式を取得します。

形式

```
Object.Item (ObjectNumber)
```

パラメタ

- *Object*

Inserts コレクションオブジェクトへの参照を表すオブジェクト式です。

- *ObjectNumber*

目的の Insert オブジェクトの番号を、数値で設定します。番号は、1~*Inserts.Count* プロパティの範囲内で設定します。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

RemoveAll メソッド

機能

すべての Insert オブジェクトをクリアします。

形式

```
Object.RemoveAll
```

パラメタ

- *Object*

Inserts コレクションオブジェクトへの参照を表すオブジェクト式です。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

7.17 Insert オブジェクトの詳細

Insert オブジェクトは、表に挿入する 1 行分のデータを管理するオブジェクトです。一つの Insert オブジェクトには列単位に複数の挿入値を設定できます。

Insert オブジェクトのプロパティとメソッドの一覧を次に示します。

●Insert オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Column	○	○	追加情報の列名を設定します。
InsertValue	○	○	追加情報のデータを設定します。
InsertValueCount	—	○	追加情報の個数を数値で取得します。

●Insert オブジェクトのメソッドの一覧

メソッド	説明
AddInsertValue	追加情報を追加します。

Column プロパティ

機能

追加情報の番号で設定した情報の列名を、文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object.Column (ValueNumber) [= Column]
```

パラメタ

- *Object*
Insert オブジェクトへの参照を表すオブジェクト式です。
- *ValueNumber*
目的の追加情報の番号を、数値で設定します。番号は、1~*Insert.InsertValueCount* プロパティの範囲内で設定します。
- *Column*
列名を、1~64 バイトの文字列で設定します。長さ 0 の文字列は設定できません。

使い方

- 設定する列名の書式については、「[8.1.3 列名を設定するときの規則](#)」を参照してください。

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

InsertValue プロパティ

機能

AddInsertValue メソッドで追加した追加情報のデータを設定します。このプロパティは、値も取得できません。

形式

```
Object.InsertValue (ValueNumber) [= Data]
```

パラメタ

- *Object*
Insert オブジェクトへの参照を表すオブジェクト式です。
- *ValueNumber*
目的の追加情報の番号を、数値で設定します。番号は、1~*Insert.InsertValueCount* プロパティの範囲内で設定します。
- *Data*
データを、0~256 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが仮定されます。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

InsertValueCount プロパティ

機能

追加情報の数を、数値で取得します。

形式

```
Object.InsertValueCount
```

パラメタ

- *Object*

Insert オブジェクトへの参照を表すオブジェクト式です。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

AddInsertValue メソッド

機能

追加情報を追加します。列名には、*Value* に設定するデータを格納する列名を設定します。データには、新しく追加するデータを設定します。

形式

```
Object. AddInsertValue Column, Value
```

パラメタ

- *Object*

Insert オブジェクトへの参照を表すオブジェクト式です。

- *Column*

列名を、1~64 バイトの文字列で設定します。長さ 0 の文字列は設定できません。

- *Value*

データを、0~256 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが仮定されます。

使い方

- 設定する列名の書式については、「[8.1.3 列名を設定するときの規則](#)」を参照してください。
- 複数の列にデータを追加するときは、同じ Insert オブジェクトに対して複数の追加情報を設定してください。
- AddInsertValue メソッドで設定したデータを変更/取得するときは、InsertValue プロパティを使います。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

7.18 Deletes コレクションオブジェクトの詳細

Deletes コレクションオブジェクトは、Delete オブジェクトをリストオブジェクトとして管理します。

Deletes コレクションオブジェクトのプロパティとメソッドの一覧を次に示します。

●Deletes コレクションオブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	Delete オブジェクトの個数を数値で取得します。

●Deletes コレクションオブジェクトのメソッドの一覧

メソッド	説明
Add	Delete オブジェクトを追加します。
Item	Delete オブジェクトのオブジェクト式を取得します。
RemoveAll	すべての Delete オブジェクトをクリアします。

Count プロパティ

機能

Deletes.Add メソッドで作成した Delete オブジェクトの個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

Deletes コレクションオブジェクトへの参照を表すオブジェクト式です。

Add メソッド

機能

Delete オブジェクトを生成します。

形式

```
Object.Add
```

パラメタ

- *Object*

Deletes コレクションオブジェクトへの参照を表すオブジェクト式です。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

Item メソッド

機能

Delete オブジェクトのオブジェクト式を取得します。

形式

```
Object.Item (ObjectNumber)
```

パラメタ

- *Object*

Deletes コレクションオブジェクトへの参照を表すオブジェクト式です。

- *ObjectNumber*

目的の Delete オブジェクト番号を、数値で設定します。番号は、1~*Deletes.Count* プロパティの範囲内で設定します。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

RemoveAll メソッド

機能

すべての Delete オブジェクトをクリアします。

形式

```
Object.RemoveAll
```

パラメタ

- *Object*

Deletes コレクションオブジェクトへの参照を表すオブジェクト式です。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

7.19 Delete オブジェクトの詳細

Delete オブジェクトは、Delete 文を生成するための情報になる削除データの設定値情報を管理します。Delete 文 1 文につき、一つの Delete オブジェクトが対応します。

Delete オブジェクトのプロパティとメソッドの一覧を次に示します。

●Delete オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
KeyColumn	○	○	削除行条件の列名を設定します。
KeyCount	—	○	削除行条件の数を取得します。
KeyValue	○	○	削除行条件のデータを設定します。

●Delete オブジェクトのメソッドの一覧

メソッド	説明
AddKey	削除行条件を追加します。

KeyColumn プロパティ

機能

削除行条件の番号で設定した条件の列名を、文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object.KeyColumn (KeyNumber) [= Column]
```

パラメタ

- *Object*
Delete オブジェクトへの参照を表すオブジェクト式です。
- *KeyNumber*
目的の削除行条件の番号を、数値で設定します。番号は、1~Delete.KeyCount プロパティの範囲内で設定します。
- *Column*
列名を、1~64 バイトの文字列で設定します。長さ 0 の文字列は設定できません。

使い方

- 設定する列名の書式については、「[8.1.3 列名を設定するときの規則](#)」を参照してください。

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

KeyCount プロパティ

機能

削除行条件の数を、数値で取得します。

形式

```
Object.KeyCount
```

パラメタ

- *Object*
Delete オブジェクトへの参照を表すオブジェクト式です。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

KeyValue プロパティ

機能

削除行条件の番号で設定した条件のデータを、文字列で設定します。このプロパティは、値も取得できます。

形式

```
Object.KeyValue (KeyNumber) [= Data]
```

パラメタ

- *Object*
Delete オブジェクトへの参照を表すオブジェクト式です。
- *KeyNumber*
目的の削除行条件の番号を、数値で設定します。番号は、1~*Delete.KeyCount* プロパティの範囲内で設定します。
- *Data*

データを、0～256 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが仮定されます。

使い方

- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

AddKey メソッド

機能

削除行条件を追加します。列名には、*Value* に設定するデータを格納している列名を設定します。データには、削除する行を特定するデータを設定します。

形式

```
Object.AddKey Column, Value
```

パラメタ

- *Object*
Delete オブジェクトへの参照を表すオブジェクト式です。
- *Column*
列名を、1～64 バイトの文字列で設定します。長さ 0 の文字列は設定できません。
- *Value*
データを、0～256 バイトの文字列で設定します。長さ 0 の文字列を設定すると、NULL データが仮定されます。

使い方

- 設定する列名の書式については、「[8.1.3 列名を設定するときの規則](#)」を参照してください。
- 一つの削除行条件で削除する行を特定できないときは、同一の Delete オブジェクトに対して複数の削除行条件を設定してください。
- このプロパティは、使えるデータベースに制限があります。データベースの制限については、「[8.3 データベースと使えるプロパティ/メソッド](#)」を参照してください。

7.20 Catalog オブジェクトの詳細

Catalog オブジェクトは、手順ファイルを参照し QueryDefine オブジェクトを生成するための情報を管理します。

Catalog オブジェクトのメソッドの一覧を次に示します。Catalog オブジェクトのプロパティはありません。

●Catalog オブジェクトのメソッドの一覧

メソッド	説明
Convert	読み込んだ手順ファイルの情報から QueryDefine オブジェクトを生成します。
FileLoad	手順ファイルの情報を読み込みます。

Convert メソッド

機能

FileLoad メソッドで読み込まれた手順ファイルの情報から、QueryDefine オブジェクトを生成します。

形式

```
Object. Convert
```

パラメタ

- *Object*
Catalog オブジェクトへの参照を表すオブジェクト式です。

使い方

- 可変値を使用している手順ファイルを読み込んだ場合は、Convert メソッドを発行する前に VariableList オブジェクトに実行値を設定してください。実行値が設定されていない場合は、手順ファイルが格納している初期値を使用して QueryDefine オブジェクトを生成します。
- Convert メソッドを発行すると、現在の QueryDefine オブジェクトの内容は削除されます。

FileLoad メソッド

機能

手順ファイルから抽出に関する情報を読み込みます。抽出条件として可変値が使用されている場合は、*VariableList.Count* プロパティに可変値の個数が返されます。手順ファイルの情報から QueryDefine オブジェクトを生成するには、Convert メソッドを使用します。

形式

```
Object. FileLoad (FileName)
```

パラメタ

- *Object*
Catalog オブジェクトへの参照を表すオブジェクト式です。
- *FileName*
ファイル名を、1～256 バイトの文字列で設定します。

使い方

- ファイル名にパスが含まれていない場合は、カレントディレクトリを仮定します。
- FileLoad メソッドを発行すると、VariableList オブジェクトの各プロパティの内容が更新されます。
- ODBC 接続の抽出を登録した手順ファイルを設定した場合、動作は保証しません。
- 次に示す手順ファイルを設定した場合、エラーとなります。
 - DBPARTNER/Client 03-00 以前のバージョンの手順ファイル
 - 表定義を登録した手順ファイル
 - アウタジョインを登録した手順ファイル
- 保存、予約、絞り込みを登録した手順ファイルからは、表定義に関する情報は読み込みません。
- メール連携を登録した手順ファイルからは、メール操作に関する情報は読み込みません。
- 更新用抽出を登録した手順ファイルは、抽出操作として読み込みます。
- 手順ファイルに登録されている情報でも、次に示す情報は読み込みません。

ホスト名称、転送先種別、転送先ファイル名、逐次検索、件数確認、抽出件数の制限、仮想列の名称、更新キー解除、BINARY 属性のファイル種別

7.21 VariableList オブジェクトの詳細

VariableList オブジェクトは、手順ファイルに格納されている可変値情報を取得したり、取得した情報を管理したりします。それぞれの情報はプロパティによって返されます。

VariableList オブジェクトのプロパティの一覧を次に示します。VariableList オブジェクトのメソッドはありません。

●VariableList オブジェクトのプロパティの一覧

プロパティ	設定	取得	説明
Count	—	○	手順ファイルに格納されている可変値の個数を数値で取得します。
VariableName	—	○	可変値の名称を文字列で取得します。
VariableValue	○	○	可変値の値とする実行値を設定します。

Count プロパティ

機能

`Catalog.FileLoad` メソッドで読み込んだ手順ファイルに格納されている可変値の個数を、数値で取得します。

形式

```
Object.Count
```

パラメタ

- *Object*

VariableList オブジェクトへの参照を表すオブジェクト式です。

使い方

- 手順ファイルに可変値が使用されていない場合、このプロパティの値は0になります。

VariableName プロパティ

機能

可変値の名称を文字列で取得します。

形式

```
Object. VariableName (Index)
```

パラメタ

- *Object*

VariableList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の可変値の番号を数値で設定します。番号は、1~*VariableList.Count* プロパティの範囲で設定します。

使い方

- 可変値の名称の先頭に付いている“&”は返りません。

VariableValue プロパティ

機能

可変値の値とする実行値を設定します。このプロパティは、値の取得もできます。

形式

```
Object. VariableValue (Index) [= String]
```

パラメタ

- *Object*

VariableList オブジェクトへの参照を表すオブジェクト式です。

- *Index*

目的の可変値の番号を数値で設定します。番号は、1~*VariableList.Count* プロパティの範囲で設定します。

- *String*

実行値を、1~256 バイトの文字列で設定します。

使い方

- *Catalog.FileLoad* メソッド発行直後は、手順ファイルに格納されている初期値が実行値として設定されます。
- 実行値に長さ 0 の文字列、又は 257 バイト以上の文字列が設定された場合、エラーとなります。

8

CommuniNet 接続を使うときの参照情報

CommuniNet 接続で DBPARTNER2 Client のアプリケーションを開発するときに、データベースとサーバプログラム別で注意する必要がある項目について説明しています。

8.1 条件式, 表名, 列名, 配列列の形式

CommuniNet 接続で条件式, 表名, 列名, 配列列を設定するときの規則について説明します。

8.1.1 条件式を設定するときの規則

条件式を設定するときの書式と規則を次に示します。

条件式

<条件式> := { <条件><比較演算子><条件> | <述語> }

条件

<条件> := { <定数> | <列> | <条件> | <条件><算術演算子><条件> }

定数

<定数> := { 数値 | '文字列' }

文字列は「'」(アポストロフィ)で囲んでください。文字列に「'」(アポストロフィ)を含めるときは、アポストロフィを二つ「''」設定してください。

列

<列> := [<表>.] "列名"

列名は「"」(ダブルクォーテーション)で囲んでください。

表

<表> := ["所有者."] "表名"

所有者と表名は「"」(ダブルクォーテーション)で囲んでください。

算術演算子

<算術演算子> := { + | - | * | / }

比較演算子

<比較演算子> := { = | <> | < | > | <= | >= }

述語

<述語> := { <列> IS [NOT] NULL |
<列> [NOT] IN (<定数> [, <定数> . . .]) |
<列> [NOT] LIKE <文字パターン> |
<列> [NOT] BETWEEN <定数> AND <定数> }

記号の説明を次に示します。

記号	説明
{ } 波括弧	この記号で囲まれている複数の項目のうちから一つを選択します。 (例) { A B C } は、「A, B 又は C のどれかを指定する」ことを示します。

記号	説明
[] きっ甲	この記号で囲まれている項目は省略してもよいことを示します。 (例) [A.] B は、「A.B と指定する」又は「B を指定する」ことを示します。
.. リーダ	記述を省略していることを示します (例) A, B, ...は、「A の後ろに B を複数指定する」ことを示します。
△ 白三角	空白を空けることを示します。 (例) A△B は、「A と B の間に空白を入れる」ことを示します。

8.1.2 表名を設定するときの規則

表名を設定するときは、所有者と表名を「"」（ダブルクォーテーション）で囲みます。書式を次に示します。

書式

["所有者."] "表名"

表名と列名を設定するプロパティとメソッドを次に示します。

- *ColumnList*.TableName プロパティ
- *Tables.Add* メソッド
- *Table*.TableName プロパティ
- *UpdateDefine*.TableName プロパティ

(1) 所有者と表名の規則

データベースごとの所有者と表名の規則を表 8-1 に示します。

表 8-1 データベースごとの所有者と表名の規則

データベース	所有者	表名
V3RD, V3SD, V3RDB1, V1PDM, V3PDM	表の所有者	表名
V3USR, V1USR	表の所有者	表名（ただし、ACE3 の設定でデータベースと混在させている場合は、表名の後ろに「/F」を付けます）
V1SPL	ユーザ名	「ジョブ通し番号」「分割番号」「出力クラス」を連結した文字列
VKSPL	ユーザ名	「ジョブ通し番号」「ファイル通し番号」を連結した文字列
VKRDBF	表の所有者（所有者が 8 バイト未満のときは、所有者の後ろに空白文字を入れて 8 バイトにします）	表名

8.1.3 列名を設定するときの規則

プロパティとメソッドに列名を設定するときは、列名を「"」（ダブルクォーテーション）で囲みます。列名の書式を次に示します。

(1) 抽出列を設定するときの書式

抽出列を設定するときの書式を次に示します。

通常の前列の書式

["所有者."] ["表名."] "列名"

繰り返し列、配列列の書式

["所有者."] ["表名."] "列名 (n) "

n：要素数

抽出列の設定の対象になるプロパティ/メソッドを次に示します。

- Columns.Add メソッド
- Column.ColumnName プロパティ

(2) 更新する列を設定するときの書式

更新する列を設定するときの書式を次に示します。

通常の前列の書式

"列名"

繰り返し列、配列列の書式

"列名 (n) "

n：要素数

更新する列の設定の対象になるプロパティ/メソッドを次に示します。

- *Update*.KeyColumn プロパティ
- *Update*.KeyValue プロパティ
- *Update*.AddKey メソッド
- *Update*.AddUpdateValue メソッド
- *Insert*.Column プロパティ
- *Insert*.AddInsertValue メソッド
- *Delete*.KeyColumn プロパティ
- *Delete*.AddKey メソッド

8.1.4 配列列を設定するときの規則

VOS3 XDM/RD E2 の配列列にデータを追加するときは、次に示す制限があります。

1. すべての要素にデータを設定してください

配列列のすべての要素にデータを設定してください。データを設定していない要素があると、更新の実行時にエラーになります。

2. 列の要素が昇順になるように設定してください

配列列の要素が昇順になるように設定してください。要素が昇順に並んでいないと、更新の実行時にエラーになります。

3. 列の要素は続けて設定してください

配列列の要素は連続して設定してください。要素の途中で配列列以外の列があると、更新の実行時にエラーになります。

(例)

下記の表 (列 B が配列列) にデータを追加する場合のコーディング例を示します。

列 A	列 B (1)	列 B (2)	列 C
1	x	y	5

正しい設定

```
:  
AddInsertValue "列A", 1  
AddInsertValue "列B (1) ", "x"  
AddInsertValue "列B (2) ", "y"  
AddInsertValue "列C", 5  
:
```

1. に該当する誤り (要素 1 の列にデータが設定されていません)

```
:  
AddInsertValue "列A", 1  
AddInsertValue "列B (2) ", "y"  
AddInsertValue "列C", 5  
:
```

2. に該当する誤り (要素が降順に設定されています)

```
:  
AddInsertValue "列A", 1  
AddInsertValue "列B (2) ", "y"  
AddInsertValue "列B (1) ", "x"  
AddInsertValue "列C", 5  
:
```

3. に該当する誤り (要素が連続して設定されていません)

```
:  
AddInsertValue "列A", 1  
AddInsertValue "列B (1) ", "x"
```

```
AddInsertValue "列C", 5  
AddInsertValue "列B (2) ", "y"  
:
```

8.2 クライアント側プログラムでコード変換するときの注意

CodeChangeMode プロパティで dbpClient を設定するときには注意する項目について説明します。

8.2.1 サーバプログラムの設定

サーバプログラムごとに、コマンド記号の設定が必要です。

コマンド記号は、ログオンマクロスクリプトの中でも指定できます。

(1) VOS3 HOAPDBS E2 の場合

コマンド記号：&¥HOAPDBSPCCONVERTPROC

指定値：NOCONV

VOS3 HOAPDBS E2 のコマンド記号については、マニュアル「VOS3 統合情報管理共通サービス HOAPSERV 運用の手引」を参照してください。

(2) VOSK HOAPDBS/PC の場合

コマンド記号：&@S¥HDBPCCONVERTPROC (セッション階層の指定例)

指定値：NOCONV

VOSK HOAPDBS/PC のコマンド記号については、マニュアル「VOSK システム OA」を参照してください。

(3) VOS3 ACE3 E3 の場合

コマンド記号：&¥ACE3E3PCCDCHG

指定値：PC

VOS3 ACE3 E3 のコマンド記号については、マニュアル「VOS3 エンドユーザ言語 ACE3 E3 運用編」を参照してください。

(4) VOS1 DBPARTNER/Link の場合

コマンド記号：@¥DBPLKPCCDCHG

指定値：PC

VOS1 DBPARTNER/Link のコマンド記号については、マニュアル「VOS1 エンドユーザ言語 ACE3 E2 運用の手引」を参照してください。

8.2.2 空白文字に変換されるコード

次に示す範囲の EBCDIK コードは、JIS の空白文字 ('20'X) に変換されます。

'00'X~'3F'X, 'B2'X~'B9'X, 'CA'X~'CF'X, 'DA'X~'DF'X, 'E1'X, 'EA'X~'EF'X, 'FA'X~'FF'X

8.2.3 外字の変換規則

KEIS コードの外字は、CommuniNet がシフト JIS の外字に変換します。変換の規則は、CommuniNet のコードマッピングユーティリティを使って作成してください。コードマッピングユーティリティで対応できない外字は、シフト JIS の未定義コード ('889E'X) に変換されます。

外字の変換規則については、CommuniNet のコードマッピングユーティリティについて説明しているドキュメントを参照してください。

8.2.4 1 列当たりの制限

1 列のデータが 2,993 バイトを超える漢字データは抽出できません。該当するデータがあると、各オブジェクトの Get メソッド又は Execute メソッドがエラーを返します。

8.3 データベースと使えるプロパティ/メソッド

CommuniNet 接続のときは、データベースによって DBPARTNER2 Client のプロパティ/メソッドが使えないことがあります。

ここでは、サーバプログラム (HOAPDBS, ACE3) ごとに、プロパティ/メソッドが使えるかどうかを表形式で説明します。

8.3.1 HOAPDBS がサーバプログラムするとき

サーバプログラムが HOAPDBS のときの、データベースと使えるプロパティ/メソッドの関係を、オブジェクトごとに示します。表の表記と凡例を次に示します。

(表記)

- dbpV3RD : VOS3 XDM/RD E2 を示します。
- dbpV3SD : VOS3 XDM/SD E2 を示します。
- dbpVKRDBF : VOSK RDB 編成ファイルを示します。
- dbpVKSPL : VOSK スプールファイルを示します。

(凡例)

- : このプロパティ/メソッドを使えます。
- △ : このプロパティ/メソッドを使えますが、注※に示す注意が必要です。
- × : このメソッドを実行すると、エラーになります。
- － : このメソッドを実行しても、何も処理されません。又はこのプロパティに値を設定しても、使われません。

(1) DBPARTNER オブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
CodeChangeMode	○	○	○	○
CommitMode	○	○	○	－
DictPath	○	○	○	○
TableDefSource	○	○	○	○
IsConnect	○	○	○	○
LogoffFileName	○	○	○	○
LogonFileName	○	○	○	○
MacroTimeLimit	○	○	○	○

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
Password	○	○	○	○
ResponseLimit	—	—	—	—
TargetDatabase	○	○	○	○
TargetServer	○	○	○	○
TerminalMode	○	○	○	○
TerminalName	○	○	○	○
UserID	○	○	○	○
Cancel	○	○	○	○
Commit	○	○	○	—
Connect	○	○	○	○
Disconnect	○	○	○	○
Rollback	○	○	○	—
TerminalClose	○	○	○	○

(2) Result オブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
QueryCountLimit	○	○	○	○
DataCheck	○	○	○	○
HeaderOutput	○	○	○	○
RecordHeader	○	○	○	△※
FieldHeader	○	○	○	△※
RecordData	○	○	○	○
FieldData	○	○	○	○
FieldDataLength	○	○	○	○
FieldCount	○	○	○	○
EOF	○	○	○	○
Get	○	○	○	○
Close	○	○	○	○

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
SaveToFile	○	○	○	○
CopyToClipboard	○	○	○	○

注※

FieldHeader プロパティの設定がないときは、空白 1 文字を返します。

(3) TableList オブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
Count	○	○	○	○
CountLimit	○	○	○	○
Pattern	○	○	○	○
SelectDict	○	○	○	○
TableComment	○	○	○	○
TableCreateDate	○	○	○	○
TableDataSource	○	○	○	○
TableDBMName	△ *	△ *	△ *	△ *
TableDeleteAuth	○	○	○	○
TableGroupName	○	○	○	○
TableHeader	○	○	○	○
TableInsertAuth	○	○	○	○
TableName	○	○	○	○
TableOwner	○	○	○	○
TableSelectAuth	○	○	○	○
TableType	○	○	○	○
TableUpdateAuth	○	○	○	○
Get	○	○	○	○

注※

常に長さ 0 の文字列を返します。

(4) ColumnList オブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSP
ColumnComment	○	○	○	×
ColumnEditPattern	○	○	○	×
ColumnHeader	○	○	○	×
ColumnLength	○	○	○	×
ColumnName	○	○	○	×
ColumnNumber	○	○	○	×
ColumnScale	○	○	○	×
ColumnType	○	○	○	×
Count	○	○	○	△ ※1
CountLimit	○	○	○	△ ※2
SelectDict	○	○	○	○
TableName	○	○	○	△ ※2
TablePassword	—	○	—	—
Get	○	○	○	×

注※1

常に 0 を返します。

注※2

Get メソッドがエラーになるため、設定は無意味です。

(5) QueryDefine オブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSP
Exclusive	○	—	—	—
QueryCount	△ ※1	×	△ ※1	×
Execute	○	○	○	△ ※2
RemoveAll	○	○	○	○

注※1

集合関数が設定されているときは、エラーを返します。

注※2

抽出列、抽出条件が設定してあると、エラーを返します。

(6) Tables コレクションオブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
Count	○	○	○	○
Add	○	△※	△※	△※
Item	○	○	○	○
RemoveAll	○	○	○	○

注※

複数の表を設定すると、*QueryDefine.Execute* メソッドでエラーを返します。

(7) Table オブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
TableName	○	○	○	○
TablePassword	—	○	—	—

(8) Columns コレクションオブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
Count	○	○	○	△※
Add	○	○	○	△※
Item	○	○	○	△※
RemoveAll	○	○	○	△※

注※

抽出する列が設定されていると、*QueryDefine.Execute* メソッドでエラーを返します。

(9) Column オブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
ColumnName	○	○	○	△※3
SortMode	○	△※1	○	△※3
SortPriority	○	△※1	○	△※3

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
FunctionMode	○	△ ※2	○	△ ※3

注※1

列にソート種別が設定されていると、*QueryDefine.Execute* メソッドでエラーを返します。

注※2

列に集合関数が設定されていると、*QueryDefine.Execute* メソッドでエラーを返します。

注※3

抽出する列が設定されていると、*QueryDefine.Execute* メソッドでエラーを返します。

(10) Conditions コレクションオブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
Count	○	○	○	△ ※
Relation	○	○	○	△ ※
Add	○	○	○	△ ※
Item	○	○	○	△ ※
RemoveAll	○	○	○	△ ※

注※

抽出条件を設定すると、*QueryDefine.Execute* メソッドでエラーを返します。

(11) Condition オブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
Text	○	○	○	△ ※

注※

抽出条件を設定すると、*QueryDefine.Execute* メソッドでエラーを返します。

(12) UpdateDefine オブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
Exclusive	○	—	—	—
UpdateMode	○	△ ※1	○	△ ※2

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
TableName	○	○	○	△ ※2
TablePassword	—	○	—	—
ExecutedRowCount	○	○	○	△ ※3
ErrorInformation	○	○	○	△ ※3
Execute	○	○	○	×
RemoveAll	○	○	○	○

注※1

dbpVariable を設定すると、*UpdateDefine.Execute* メソッドでエラーを返します。

注※2

更新処理ができないデータベースです。値の設定は無意味です。

注※3

常に 0 を返します。

(13) Updates コレクションオブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
Count	○	○	○	—
Add	○	○	○	△ ※
Item	○	○	○	—
RemoveAll	○	○	○	—

注※

更新処理ができないデータベースです。値の設定は無意味です。

(14) Update オブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
KeyCount	○	○	○	○
UpdateValueCount	○	○	○	○
KeyColumn	○	○	○	△ ※
KeyValue	○	○	○	△ ※
Column	○	○	○	△ ※

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
UpdateValue	○	○	○	△※
AddKey	○	○	○	△※
AddUpdateValue	○	○	○	△※

注※

更新処理ができないデータベースです。値の設定は無意味です。

(15) Inserts コレクションオブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
Count	○	○	○	○
Add	○	○	○	△※
Item	○	○	○	○
RemoveAll	○	○	○	○

注※

更新処理ができないデータベースです。値の設定は無意味です。

(16) Insert オブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
InsertValueCount	○	○	○	○
Column	○	○	○	△※
InsertValue	○	○	○	△※
AddInsertValue	○	○	○	△※

注※

更新処理ができないデータベースです。値の設定は無意味です。

(17) Deletes コレクションオブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
Count	○	○	○	○

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
Add	○	○	○	△※
Item	○	○	○	○
RemoveAll	○	○	○	○

注※

更新処理ができないデータベースです。値の設定は無意味です。

(18) Delete オブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
KeyCount	○	○	○	○
KeyColumn	○	○	○	△※
KeyValue	○	○	○	△※
AddKey	○	○	○	△※

注※

更新処理ができないデータベースです。値の設定は無意味です。

(19) Catalog オブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
Convert	○	○	○	○
FileLoad	○	○	○	○

(20) VariableList オブジェクト

プロパティ/メソッド	データベース			
	dbpV3RD	dbpV3SD	dbpVKRDBF	dbpVKSPL
Count	○	○	○	○
VariableName	○	○	○	○
VariableValue	○	○	○	○

8.3.2 ACE3 E3 がサーバプログラムのとき

サーバプログラムが ACE3 E3 のときの、データベースと使えるプロパティ/メソッドの関係を、オブジェクトごとに示します。表の表記と凡例を次に示します。

(表記)

- dbpV3PDM : VOS3 PDMII E2 を示します。
- dbpV3USR : VOS3 ユーザファイルを示します。
- dbpV3RDB1 : VOS3 RDB1 を示します。
- dbpV1PDM : VOS1 PDMII E2 を示します。
- dbpV1USR : VOS1 ユーザファイルを示します。
- dbpV1SPL : VOS1 スプールファイルを示します。

(凡例)

- : このプロパティ/メソッドを使えます。
- △ : このプロパティ/メソッドを使えますが、注※に示す注意が必要です。
- × : このメソッドを実行すると、エラーになります。
- － : このメソッドを実行しても、何も処理されません。又はこのプロパティに値を設定しても、使われません。

(1) DBPARTNER オブジェクト

プロパティ /メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
CodeChangeMode	○	○	○	○	○	○
CommitMode	－	－	－	－	－	－
DictPath	○	○	○	○	○	○
TableDefSource	○	○	○	○	○	○
IsConnect	○	○	○	○	○	○
LogoffFileName	○	○	○	○	○	○
LogonFileName	○	○	○	○	○	○
MacroTimeLimit	○	○	○	○	○	○
Password	○	○	○	○	○	○
ResponseLimit	－	－	－	○	○	○
TargetDatabase	○	○	○	○	○	○
TargetServer	○	○	○	○	○	○

プロパティ ／メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
TerminalMode	○	○	○	○	○	○
TerminalName	○	○	○	○	○	○
UserID	○	○	○	○	○	○
Cancel	○	○	○	○	○	○
Commit	—	—	—	—	—	—
Connect	○	○	○	○	○	○
Disconnect	○	○	○	○	○	○
Rollback	—	—	—	—	—	—
TerminalClose	○	○	○	○	○	○

(2) Result オブジェクト

プロパティ ／メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
QueryCountLimit	○	○	○	○	○	○
DataCheck	○	○	○	○	○	○
HeaderOutput	○	○	○	○	○	○
RecordHeader	○	○	○	○	○	△ ＊
FieldHeader	○	○	○	○	○	△ ＊
RecordData	○	○	○	○	○	○
FieldData	○	○	○	○	○	○
FieldDataLength	○	○	○	○	○	○
FieldCount	○	○	○	○	○	○
EOF	○	○	○	○	○	○
Get	○	○	○	○	○	○
Close	○	○	○	○	○	○
SaveToFile	○	○	○	○	○	○
CopyToClipboard	○	○	○	○	○	○

注※

FieldHeader プロパティを設定していないと、空白 1 文字を返します。

(3) TableList オブジェクト

プロパティ /メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
Count	○	○	○	○	○	○
CountLimit	○	○	○	○	○	○
SelectDict	○	○	○	○	○	○
Pattern	○	○	○	○	○	○
TableComment	○	○	○	○	○	○
TableCreateDate	○	○	○	○	○	○
TableDataSource	○	○	○	○	○	○
TableDBMName	○	△※	△※	○	△※	△※
TableDeleteAuth	○	○	○	○	○	○
TableGroupName	○	○	○	○	○	○
TableHeader	○	○	○	○	○	○
TableInsertAuth	○	○	○	○	○	○
TableName	○	○	○	○	○	○
TableOwner	○	○	○	○	○	○
TableSelectAuth	○	○	○	○	○	○
TableType	○	○	○	○	○	○
TableUpdateAuth	○	○	○	○	○	○
Get	○	○	○	○	○	○

注※

常に長さ 0 の文字列を返します。

(4) ColumnList オブジェクト

プロパティ /メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
ColumnComment	○	○	○	○	○	×
ColumnEditPattern	○	○	○	○	○	×
ColumnHeader	○	○	○	○	○	×

プロパティ ／メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
ColumnLength	○	○	○	○	○	×
ColumnName	○	○	○	○	○	×
ColumnNumber	○	○	○	○	○	×
ColumnScale	○	○	○	○	○	×
ColumnType	○	○	○	○	○	×
Count	○	○	○	○	○	△ ※1
CountLimit	○	○	○	○	○	△ ※2
SelectDict	○	○	○	○	○	○
TableName	○	○	○	○	○	△ ※2
TablePassword	—	—	—	—	—	—
Get	○	○	○	○	○	×

注※1

常に0を返します。

注※2

Getメソッドがエラーになるため、設定は無意味です。

(5) QueryDefine オブジェクト

プロパティ ／メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
Exclusive	—	—	—	—	—	—
QueryCount	△ ※1	△ ※1	△ ※1	△ ※1	△ ※1	△ ※2
Execute	○	○	○	○	○	△ ※2
RemoveAll	○	○	○	○	○	○

注※1

集合関数を設定していると、エラーを返します。

注※2

抽出列、抽出条件を設定していると、エラーを返します。

(6) Tables コレクションオブジェクト

プロパティ /メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
Count	○	○	○	○	○	○
Add	△ ※1	△ ※1	△ ※1	△ ※1	△ ※1	△ ※2
Item	○	○	○	○	○	○
RemoveAll	○	○	○	○	○	○

注※1

6 個以上の表を設定すると、*QueryDefine.Execute* でエラーを返します。

注※2

複数の表を設定すると、*QueryDefine.Execute* でエラーを返します。

(7) Table オブジェクト

プロパティ /メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
TableName	○	○	○	○	○	○
TablePassword	—	—	—	—	—	—

(8) Columns コレクションオブジェクト

プロパティ /メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
Count	○	○	○	○	○	○
Add	○	○	○	○	○	△ ※
Item	○	○	○	○	○	○
RemoveAll	○	○	○	○	○	○

注※

抽出列を設定すると、*QueryDefine.Execute* メソッドでエラーを返します。

(9) Column オブジェクト

プロパティ /メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
ColumnName	○	○	○	○	○	△※
SortMode	○	○	○	○	○	△※
SortPriority	○	○	○	○	○	△※
FunctionMode	○	○	○	○	○	△※

注※

抽出列を設定すると、*QueryDefine.Execute* メソッドでエラーを返します。

(10) Conditions コレクションオブジェクト

プロパティ /メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
Count	○	○	○	○	○	○
Relation	○	○	○	○	○	△※
Add	○	○	○	○	○	△※
Item	○	○	○	○	○	○
RemoveAll	○	○	○	○	○	○

注※

抽出条件を設定すると、*QueryDefine.Execute* メソッドでエラーを返します。

(11) Condition オブジェクト

プロパティ /メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
Text	○	○	○	○	○	△※

注※

抽出条件を設定すると、*QueryDefine.Execute* メソッドでエラーを返します。

(12) UpdateDefine オブジェクト

プロパティ ／メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
Exclusive	—	—	—	—	—	—
UpdateMode	△ ※1	△ ※1	△ ※1	△ ※1	△ ※1	△ ※1
TableName	△ ※1	△ ※1	△ ※1	△ ※1	△ ※1	△ ※1
TablePassword	—	—	—	—	—	—
ExecutedRowCount	△ ※2	△ ※2	△ ※2	△ ※2	△ ※2	△ ※2
ErrorInformation	△ ※2	△ ※2	△ ※2	△ ※2	△ ※2	△ ※2
Execute	×	×	×	×	×	×
RemoveAll	○	○	○	○	○	○

注※1

更新処理ができないデータベースです。値を設定しても無意味です。

注※2

常に0を返します。

(13) Updates コレクションオブジェクト

プロパティ ／メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
Count	○	○	○	○	○	○
Add	△ ※	△ ※	△ ※	△ ※	△ ※	△ ※
Item	○	○	○	○	○	○
RemoveAll	○	○	○	○	○	○

注※

更新処理ができないデータベースです。値を設定しても無意味です。

(14) Update オブジェクト

プロパティ ／メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
KeyCount	○	○	○	○	○	○

プロパティ ／メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
UpdateValueCount	○	○	○	○	○	○
KeyColumn	△※	△※	△※	△※	△※	△※
KeyValue	△※	△※	△※	△※	△※	△※
Column	△※	△※	△※	△※	△※	△※
UpdateValue	△※	△※	△※	△※	△※	△※
AddKey	△※	△※	△※	△※	△※	△※
AddUpdateValue	△※	△※	△※	△※	△※	△※

注※

更新処理ができないデータベースです。値を設定しても無意味です。

(15) Inserts コレクションオブジェクト

プロパティ ／メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
Count	○	○	○	○	○	○
Add	△※	△※	△※	△※	△※	△※
Item	○	○	○	○	○	○
RemoveAll	○	○	○	○	○	○

注※

更新処理ができないデータベースです。値を設定しても無意味です。

(16) Insert オブジェクト

プロパティ ／メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
InsertValueCount	○	○	○	○	○	○
Column	△※	△※	△※	△※	△※	△※
InsertValue	△※	△※	△※	△※	△※	△※
AddInsertValue	△※	△※	△※	△※	△※	△※

注※

更新処理ができないデータベースです。値を設定しても無意味です。

(17) Deletes コレクションオブジェクト

プロパティ ／メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
Count	○	○	○	○	○	○
Add	△ ※	△ ※	△ ※	△ ※	△ ※	△ ※
Item	○	○	○	○	○	○
RemoveAll	○	○	○	○	○	○

注※

更新処理ができないデータベースです。値を設定しても無意味です。

(18) Delete オブジェクト

プロパティ ／メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
KeyCount	○	○	○	○	○	○
KeyColumn	△ ※	△ ※	△ ※	△ ※	△ ※	△ ※
KeyValue	△ ※	△ ※	△ ※	△ ※	△ ※	△ ※
AddKey	△ ※	△ ※	△ ※	△ ※	△ ※	△ ※

注※

更新処理ができないデータベースです。値を設定しても無意味です。

(19) Catalog オブジェクト

プロパティ ／メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
Convert	○	○	○	○	○	○
FileLoad	○	○	○	○	○	○

(20) VariableList オブジェクト

プロパティ /メソッド	データベース					
	dbp V3PDM	dbp V3USR	dbp V3RDB1	dbp V1PDM	dbp V1USR	dbp V1SPL
Count	○	○	○	○	○	○
VariableName	○	○	○	○	○	○
VariableValue	○	○	○	○	○	○

8.4 データベース別の規則

接続するデータベースごとの規則について説明します。

8.4.1 VOS3 XDM/RD E2 の場合の規則

(1) 表情報の値

表情報で返される値の意味を表 8-2 に示します。

表 8-2 表情報で返される値の意味 (VOS3 XDM/RD E2)

プロパティ/メソッド	値の説明
TableComment	表の注釈
TableCreateDate	'YYMMDDhhmmss' (YY:西暦の下2けた MM:月 DD:日 hh:時 mm:分 ss:秒)
TableDBMName	長さ 0 バイトの文字列
TableGroupName	表のグループ名
TableHeader	表の見出し
TableName	表名
TableOwner	表の所有者

(2) 列情報の値

列情報で返される値の意味を表 8-3 に示します。

表 8-3 列情報で返される値の意味 (VOS3 XDM/RD E2)

プロパティ/メソッド	値の説明
ColumnComment	列の注釈
ColumnEditPattern	編集形式文字列
ColumnHeader	列の見出し
ColumnName	列名

(3) 集合関数 (FunctionMode プロパティ) の規則

- dbpGroup は、255 列まで設定できます。
- 集合関数の設定は、列属性によっては使えない集合関数があります。列属性と使える集合関数の組み合わせを表 8-4 に示します。

表 8-4 列属性と使える集合関数の組み合わせ (VOS3 XDM/RD E2)

列属性	dbpGroup	dbpMaximum	dbpMinimum	dbpAverage	dbpSum	dbpCount
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	○	○	○	○	○	○
dbpTypeCHAR dbpTypeVARCHAR dbpTypeNCHAR dbpTypeNVARCHAR dbpTypeMCHAR dbpTypeMVARCHAR dbpTypeDATE dbpTypeINTERVAL_YEAR_TO_DAY	○	○	○	×	×	○

(凡例)

- ：指定できます。
- ×：指定できません。

(4) ソート (*Column.SortMode* プロパティ) の規則

- ソートは、255 列まで設定できます。

(5) 抽出条件 (*Conditions.Add* メソッド) の規則

(a) 列と定数の比較条件の規則

- 列と定数の比較は、列属性によっては使えない定数があります。列属性と使える定数の組み合わせを表 8-5 に示します。

表 8-5 列属性と使える定数の組み合わせ (VOS3 XDM/RD E2)

列属性	定数					
	整数定数 10 進数定数 浮動小数点定数	文字定数	各国文字定数	混在文字定数	日付の文字表現	日間隔の 10 進表現
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	○	×	×	×	×	○

列属性	定数					
	整数定数 10進数定数 浮動小数点定数	文字定数	各国文字定数	混在文字定数	日付の文字表現	日間隔の10進表現
dbpTypeCHAR dbpTypeVARCHAR	×	○	×	×	○	×
dbpTypeNCHAR dbpTypeNVARCHAR	×	×	○	×	×	×
dbpTypeMCHAR dbpTypeMVARCHAR	×	○	○	○	○	×
dbpTypeDATE	×	×	×	×	○	×
dbpTypeINTERVAL_YEAR_TO_DAY	×	×	×	×	×	○

(凡例)

○：設定できます。

×

(b) 列と列の比較条件の規則

- 列と列の比較は、列属性によっては使えない組み合わせがあります。比較できる列属性の組み合わせを表 8-6 に示します。

表 8-6 比較できる列属性の組み合わせ (VOS3 XDM/RD E2)

左辺の列属性	右辺の列属性					
	dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	dbpTypeCHAR dbpTypeVARCHAR	dbpTypeNCHAR dbpTypeNVARCHAR	dbpTypeMCHAR dbpTypeMVARCHAR	dbpTypeDATE	dbpTypeINTERVAL_YEAR_TO_DAY
dbpTypeINTEGER dbpTypeNUMERIC	○	×	×	×	×	×

左辺の列属性	右辺の列属性					
	dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	dbpTypeCHAR dbpTypeVARCHAR	dbpTypeNCHAR dbpTypeNVARCHAR	dbpTypeMCHAR dbpTypeMVARCHAR	dbpTypeDATE	dbpTypeINTERVAL_YEAR_TO_DAY
dbpTypeFLOAT dbpTypeREAL						
dbpTypeCHAR dbpTypeVARCHAR	×	○	×	○	×	×
dbpTypeNCHAR dbpTypeNVARCHAR	×	×	○	×	×	×
dbpTypeMCHAR dbpTypeMVARCHAR	×	○	×	○	×	×
dbpTypeDATE	×	×	×	×	○	×
dbpTypeINTERVAL_YEAR_TO_DAY	×	×	×	×	×	○

(凡例)

- ：設定できます。
- ×

(c) IN 条件の規則

- 定数の設定規則は、「表 8-5 列属性と使える定数の組み合わせ (VOS3 XDM/RD E2)」に従ってください。
- 定数は、255 個まで設定できます。

(d) LIKE 条件の規則

- LIKE 条件の設定は、列属性によっては使えないことがあります。LIKE 条件が設定できる列属性を表 8-7 に示します。

表 8-7 LIKE 条件が設定できる列属性 (VOS3 XDM/RD E2)

列属性	設定できるかどうか
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	×
dbpTypeCHAR dbpTypeVARCHAR dbpTypeNCHAR dbpTypeNVARCHAR dbpTypeMCHAR dbpTypeMVARCHAR	○
dbpTypeDATE dbpTypeINTERVAL_YEAR_TO_DAY	×

(凡例)

○：指定できます。

×：指定できません。

- 列と文字パターン（定数）の規則は、「表 8-5 列属性と使える定数の組み合わせ (VOS3 XDM/RD E2)」に従ってください。
- 「_」、「%」は、指定する列属性に応じて、全角又は半角を使ってください。

定数	文字種別
文字定数	半角文字の「_」、「%」
各国文字定数	全角文字の「_」、「%」
混在文字定数	半角文字の「_」、「%」

- 文字パターンの指定方法を表 8-8 に示します。

表 8-8 文字パターンの指定方法 (VOS3 XDM/RD E2)

項目	指定例	意味
前方一致	ABC%	先頭が「ABC」で、以降は任意の文字列を検索します。
後方一致	%ABC	最後が「ABC」で、それより前は任意の文字列を検索します。
任意一致	%ABC%	任意の部分が「ABC」の文字列を検索します。
完全一致	ABC	「ABC」と等しい文字列を検索します。
部分一致	__ABC_	6文字のうち、3～5文字目が「ABC」で、ほかの部分は任意の文字列を検索します。
上記以外	ABC%DE	先頭が「ABC」で、最後が「DE」の文字列を検索します。
	%ABC%DE%	任意の部分に「ABC」を含み、以降の任意の部分に「DE」を含む文字列を検索します。

項目	指定例	意味
	ABC_ _DE%	先頭が「ABC」で、2文字の任意の文字列をはさんで「DE」を含む文字列を検索します。

(e) BETWEEN 条件の規則

- 列と定数の規則は、「表 8-5 列属性と使える定数の組み合わせ (VOS3 XDM/RD E2)」に従ってください。

(f) NULL 条件の規則

制限事項はありません。

(g) 列演算条件の規則

- 繰り返し列、配列列は指定できません。
- 演算を指定できる定数を次に示します。
整数定数、10進数定数、浮動小数点定数
- 演算を指定できる列属性を次に示します。
dbpTypeINTEGER, dbpTypeNUMERIC, dbpTypeFLOAT, dbpTypeREAL
- 演算は、各辺の「+」、「-」、「*」、「/」、又は「()」の総和が64個まで指定できます。
- 列と定数を比較するときは、「表 8-5 列属性と使える定数の組み合わせ (VOS3 XDM/RD E2)」に従ってください。

(6) 条件関連 (Conditions.Relation プロパティ) の規則

- NOT, AND 及び OR を組み合わせて、255個まで指定できます。
- 括弧, NOT, AND 及び OR の優先順位は、括弧 > NOT > AND > OR です。
- 括弧, NOT, AND 及び OR は、すべて半角文字で指定してください。
- NOT, AND 及び OR の前後には空白を入力してください。

(7) 各条件で指定する定数の規則

- 各条件で指定する定数の規則を表 8-9 と表 8-10 に示します。

表 8-9 定数の指定方法 (VOS3 XDM/RD E2)

定数の種類	指定方法
整数定数	[符号] 符号なし整数
10進数定数	[符号] 整数部.小数部
浮動小数点定数	[符号] 仮数 E [符号] 指数

定数の種類	指定方法
文字定数	['] 半角の文字列 [']
各国文字定数	['] 全角の文字列 [']
混在文字定数	['] 全角及び半角の文字列 [']
日付の文字表現	['] YYYY-MM-DD ['] 年 (YYYY) 月 (MM) 日 (DD) をハイフンで結ぶ
日間隔の 10 進表現	[符号] YYYYMMDD 年 (YYYY) 月 (MM) 日 (DD) を 10 進数で並べる

表 8-10 定数の指定範囲 (VOS3 XDM/RD E2)

定数の種類	指定範囲	指定例
整数定数	- (10 ³⁰ -1) ~ (10 ³⁰ -1)	123, -123
10 進数定数	10 ⁻²⁹ ~ (10 ³⁰ -1), - (10 ³⁰ -1) ~ -10 ⁻²⁹ , 及び 0	45.6, -45.6
浮動小数点定数	約 1.7E-308 ~ 1.7E+308, 約 -1.7E+308 ~ -1.7E-308, 及び 0	5.1E3, -5.1E-3
文字定数	1 ~ 255 バイト	ABC
各国文字定数	1 ~ 127 文字	テレビ
混在文字定数	1 ~ 255 バイト	テレビ ABC
日付の文字表現	YYYY : 0001 ~ 9999 MM : 01 ~ 12 DD : 01 ~ その月の最終日	1999-03-22
日間隔の 10 進表現	YYYY : 0000 ~ 9999 MM : 00 ~ 99 DD : 00 ~ 99 全体の指定範囲 : -99991199 ~ 99991199	00020304 (2 年 3 か月と 4 日)

8.4.2 VOSK RDB 編成ファイルの場合の規則

(1) 表情報の値

表情報で返される値の意味を表 8-11 に示します。

表 8-11 表情報で返される値の意味 (VOSK RDB 編成ファイル)

プロパティ/メソッド	値の説明
TableComment	表の注釈
TableCreateDate	'YYMMDDhhmmss' (YY : 西暦の下 2 けた MM : 月 DD : 日 hh : 時 mm : 分 ss : 秒)
TableDBMName	長さ 0 バイトの文字列

プロパティ/メソッド	値の説明
TableGroupName	表のグループ名
TableHeader	表の見出し
TableName	表名
TableOwner	表の所有者 所有者が8バイト未満の場合は、後ろに空白文字が埋められます。

(2) 列情報の値

列情報で返される値の意味を表 8-12 に示します。

表 8-12 列情報で返される値の意味 (VOSK RDB 編成ファイル)

プロパティ/メソッド	値の説明
ColumnComment	長さ 0 バイトの文字列
ColumnEditPattern	編集形式文字列
ColumnHeader	列の見出し
ColumnName	列名

(3) 集合関数 (FunctionMode プロパティ) の規則

- ソートが設定されている列には、dbpGroup 以外の集合関数は設定できません。
- dbpGroup は、9 列まで設定できます。
- 集合関数の設定は、列属性によっては使えない集合関数があります。列属性と使える集合関数の組み合わせを表 8-13 に示します。

表 8-13 列属性と使える集合関数の組み合わせ (VOSK RDB 編成ファイル)

列属性	dbpGroup	dbpMaximum	dbpMinimum	dbpAverage	dbpSum	dbpCount
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeNUMERICUNS	○	○	○	○	○	○
dbpTypeCHAR dbpTypeNCHAR dbpTypeMCHAR dbpTypeBINARY	○	○	○	×	×	○

(凡例)

- ：指定できます。
- ×：指定できません。

(4) ソート (*Column.SortMode* プロパティ) の規則

- 9 列まで指定できます。
- dbpGroup 以外の集合関数が設定されている列に、ソートは設定できません。

(5) 抽出条件 (*Conditions.Add* メソッド) の規則

(a) 列と定数の比較条件の規則

- 列と定数の比較は、列属性によっては使えない定数があります。列属性と使える定数の組み合わせを表 8-14 に示します。

表 8-14 列属性と使える定数の組み合わせ (VOSK RDB 編成ファイル)

列属性	定数				
	整数定数 10 進数定数	文字定数	各国文字定数	混在文字定数	16 進定数
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeNUMERICUNS	○	×	×	×	×
dbpTypeCHAR	×	○	×	×	○
dbpTypeNCHAR	×	×	○	×	×
dbpTypeMCHAR	×	○	○	○	○
dbpTypeBINARY	×	×	×	×	○

(凡例)

- ：比較できます。
- ×：比較できません。

(b) 列と列の比較条件の規則

- 列と列の比較は、列属性によっては使えない組み合わせがあります。比較できる列属性の組み合わせを表 8-15 に示します。

表 8-15 比較できる列属性の組み合わせ (VOSK RDB 編成ファイル)

左辺の列属性	右辺の列属性				
	dbpTypeINTE GER dbpTypeNU MERIC dbpTypeNU MERICUNS	dbpTypeCHA R	dbpTypeNCH AR	dbpTypeMCH AR	dbpTypeBINA RY
dbpTypeINTEGER	○	×	×	×	×

左辺の列属性	右辺の列属性				
	dbpTypeINTEGER dbpTypeNUMERIC dbpTypeNUMERICUNS	dbpTypeCHAR	dbpTypeNCHAR	dbpTypeMCHAR	dbpTypeBINARY
dbpTypeNUMERIC dbpTypeNUMERICUNS					
dbpTypeCHAR	×	○	×	×	×
dbpTypeNCHAR	×	×	○	×	×
dbpTypeMCHAR	×	○	○	○	×
dbpTypeBINARY	×	×	×	×	○

(凡例)

- ：指定できます。
- ×

(c) IN 条件の規則

- IN 条件は使えません。

(d) LIKE 条件の規則

- LIKE 条件の設定は、列属性によっては使えないことがあります。LIKE 条件が設定できる列属性を表 8-16 に示します。

表 8-16 LIKE 条件が設定できる列属性 (VOSK RDB 編成ファイル)

列属性	設定できるかどうか
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeNUMERICUNS dbpTypeBINARY	×
dbpTypeCHAR dbpTypeNCHAR dbpTypeMCHAR	○

(凡例)

- ：指定できます。
- ×

- 列と文字パターン (定数) の規則は、「表 8-14 列属性と使える定数の組み合わせ (VOSK RDB 編成ファイル)」に従ってください。
- 「%」は、指定する列属性に応じて、全角又は半角を使ってください。

定数	文字種別
文字定数	半角文字の「%」
各国文字定数	全角文字の「%」
混在文字定数	半角文字の「%」

- 文字パターンの指定方法を表 8-17 に示します。

表 8-17 文字パターンの指定方法 (VOSK RDB 編成ファイル)

項目	指定例	意味
前方一致	ABC%	先頭が「ABC」で、以降は任意の文字列を検索します。
後方一致	%ABC	最後が「ABC」で、それより前は任意の文字列を検索します。
任意一致	%ABC%	任意の部分が「ABC」の文字列を検索します。
完全一致	ABC	「ABC」と等しい文字列を検索します。

(e) BETWEEN 条件の規則

- BETWEEN 条件は使えません。

(f) NULL 条件の規則

- NULL 条件は使えません。

(g) 列演算条件の規則

- 列演算条件は使えません。

(6) 条件関連 (Conditions.Relation プロパティ) の規則

- NOT は使えません。
- 括弧, AND 及び OR の優先順位は、括弧 > AND > OR です。
- 条件関連のネスト数は 64 までです。ネスト数は、AND 又は OR に対して優先順位を表す括弧を省略しないで指定したときの左括弧の数です。

(例)

1 AND ((2 OR 3 AND 4) OR 5) AND 6 のときに、括弧を省略しないと (1 AND (((2 OR 3) AND 4) OR 5) AND 6) になって、ネスト数は 4 です。

- 括弧, AND 及び OR は、すべて半角文字で指定してください。
- AND 及び OR の前後には空白を入力してください。

(7) 各条件で指定する定数の規則

- 各条件で指定する定数の規則を表 8-18 と表 8-19 に示します。

表 8-18 定数の指定方法 (VOSK RDB 編成ファイル)

定数の種類	指定方法
整数定数	[符号] 符号なし整数
10 進数定数	[符号] 整数部.小数部
文字定数	['] 半角の文字列 [']
漢字定数	['] 全角の文字列 [']
混在文字定数	['] 全角及び半角の文字列 [']
16 進定数	['] 0~9, A~F の半角の文字列 [']

表 8-19 定数の指定範囲 (VOSK RDB 編成ファイル)

定数の種類	指定範囲	指定例
整数定数	-9,223,372,036,854,775,808~9,223,372,036,854,775,807	123, -123
10 進数定数	.000,000,000,000,000,001~999,999,999,999,999,999, -999,999,999,999,999,999~-000,000,000,000,000,001, 及び 0	45.6, -45.6
文字定数	1~254 バイト	ABC
漢字定数	1~127 文字	テレビ
混在文字定数	1~254 バイト	テレビ ABC
16 進定数	1~254 バイト	9F

8.4.3 VOS3 XDM/SD E2 の場合の規則

(1) 表情報の値

表情報で返される値の意味を表 8-20 に示します。

表 8-20 表情報で返される値の意味 (VOS3 XDM/SD E2)

プロパティ/メソッド	値の説明
TableComment	表の注釈
TableCreateDate	'YYMMDDhhmmss' (YY:西暦の下2けた MM:月 DD:日 hh:時 mm:分 ss:秒)
TableDBMName	長さ 0 バイトの文字列
TableGroupName	表のグループ名
TableHeader	表の見出し
TableName	表名

プロパティ/メソッド	値の説明
TableOwner	表の所有者

(2) 列情報の値

列情報で返される値の意味を表 8-21 に示します。

表 8-21 列情報で返される値の意味 (VOS3 XDM/SD E2)

プロパティ/メソッド	値の説明
ColumnComment	列の注釈
ColumnEditPattern	編集形式文字列
ColumnHeader	列の見出し
ColumnName	ACE3 で定義した列名, 又は XDM/SD E2 の構成要素名

(3) 集合関数 (FunctionMode プロパティ) の規則

- 集合関数は設定できません。

(4) ソート (Column.SortMode プロパティ) の規則

- ソートは設定できません。

(5) 抽出条件 (Conditions.Add メソッド) の規則

(a) 列と定数の比較条件の規則

- 比較演算子の「<>」は使えません。
- 列と定数の比較は, 列属性によっては使えない定数があります。列属性と使える定数の組み合わせを表 8-22 に示します。

表 8-22 列属性と使える定数の組み合わせ (VOS3 XDM/SD E2)

定数	列属性			
	整数定数 10 進数定数	文字定数	漢字定数	16 進定数
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeNUMERICUNS	○	×	×	×
dbpTypeCHAR	×	○	×	○
dbpTypeNCHAR	×	×	○	×

定数	列属性			
	整数定数 10進数定数	文字定数	漢字定数	16進定数
dbpTypeBINARY	×	×	×	○

(凡例)

○：比較できます。

×：比較できません。

(b) 列と列の比較条件の規則

- 比較演算子の「<>」は使えません。
- 列と列の比較は、列属性によっては使えない組み合わせがあります。比較できる列属性の組み合わせを表 8-23 に示します。

表 8-23 比較できる列属性の組み合わせ (VOS3 XDM/SD E2)

左辺の列属性	右辺の列属性			
	dbpTypeINTEGER dbpTypeNUMERIC dbpTypeNUMERIC UNS	dbpTypeCHAR	dbpTypeNCHAR	dbpTypeBINARY
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeNUMERICUNS	○	×	×	×
dbpTypeCHAR	×	○	×	×
dbpTypeNCHAR	×	×	○	×
dbpTypeBINARY	×	×	×	○

(凡例)

○：比較できます。

×：比較できません。

(c) IN 条件の規則

- IN 条件は使えません。

(d) LIKE 条件の規則

- LIKE 条件は使えません。

(e) BETWEEN 条件の規則

- BETWEEN 条件は使えません。

(f) NULL 条件の規則

- NULL 条件は使えません。

(g) 列演算条件の規則

- 列演算条件は使えません。

(6) 条件関連 (Conditions.Relation プロパティ) の規則

- NOT は指定できません。
- AND, OR を組み合わせて、254 個まで指定できます。
- 括弧, AND 及び OR の優先順位は、括弧 > AND > OR です。
- 括弧は 64 組まで指定できます。
- 括弧, AND 及び OR は、すべて半角文字で指定してください。
- AND 及び OR の前後には空白を入力してください。

(7) 各条件で指定する定数の規則

各条件で指定する定数の規則を表 8-24 と表 8-25 に示します。

表 8-24 定数の指定方法 (VOS3 XDM/SD E2)

定数の種類	指定方法
整数定数 (2 バイト)	[符号] 符号なし整数
整数定数 (4 バイト)	[符号] 符号なし整数
整数定数 (8 バイト)	[符号] 符号なし整数
10 進数定数	[符号] 整数部.小数部
文字定数	['] 半角の文字列 [']
漢字定数	['] 全角の文字列 [']
16 進定数	['] 0~9, A~F の半角の文字列 [']

表 8-25 定数の指定範囲 (VOS3 XDM/SD E2)

定数の種類	指定範囲	指定例
整数定数 (2 バイト)	-32,768 ~ 32,767	123, -123
整数定数 (4 バイト)	-2,147,483,648 ~ 2,147,483,647	
整数定数 (8 バイト)	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807	
10 進数定数	m: 全けた数 n: 小数点以下のけた数 M: 列定義の全けた数 N: 列定義の小数点けた数	45.6, -45.6

定数の種類	指定範囲	指定例
	$1 \leq m \leq M, 1 \leq n \leq N, n \leq m$	
文字定数	1～254 バイト	ABC
漢字定数	1～127 文字	テレビ
16 進定数	1～254 バイト	9F

8.4.4 VOS3 PDMII E2, VOS1 PDMII E2 の場合の規則

(1) 表情報の値

表情報で返される値の意味を表 8-26 に示します。

表 8-26 表情報で返される値の意味 (VOS3 PDMII E2, VOS1 PDMII E2)

プロパティ/メソッド	値の説明
TableComment	表の注釈
TableCreateDate	'YYMMDDhhmmss' (YY:西暦の下2けた MM:月 DD:日 hh:時 mm:分 ss:秒)
TableDBMName	PDMII E2 の DBM 名
TableGroupName	表のグループ名
TableHeader	表の見出し
TableName	表名
TableOwner	表の所有者

(2) 列情報の値

列情報で返される値の意味を表 8-27 に示します。

表 8-27 列情報で返される値の意味 (VOS3 PDMII E2, VOS1 PDMII E2)

プロパティ/メソッド	値の説明
ColumnComment	列の注釈
ColumnEditPattern	編集形式文字列
ColumnHeader	列の見出し
ColumnName	列名

(3) 集合関数 (FunctionMode プロパティ) の規則

- dbpGroup は、21 列まで設定できます。ただし、ソートを設定した列数との和が 22 列以上は設定できません。
- 集合関数の設定は、列属性によっては使えない集合関数があります。列属性と使える集合関数の組み合わせを表 8-28 に示します。

表 8-28 列属性と使える集合関数の組み合わせ (VOS3 PDMII E2, VOS1 PDMII E2)

列属性	dbpGroup	dbpMaximum	dbpMinimum	dbpAverage	dbpSum	dbpCount
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	○	○	○	○	○	○
dbpTypeCHAR dbpTypeVARCHAR dbpTypeNCHAR dbpTypeNVARCHAR	○	○	○	×	×	○

(凡例)

- ：指定できます。
- ×：指定できません。

(4) ソート (Column.SortMode プロパティ) の規則

- 21 列まで指定できます。ただし、dbpGroup を設定した列数との和が 22 列以上は設定できません。

(5) 抽出条件 (Conditions.Add メソッド) の規則

(a) 列と定数の比較条件の規則

- 列と定数の比較は、列属性によっては使えない定数があります。列属性と使える定数の組み合わせを表 8-29 に示します。

表 8-29 列属性と使える定数の組み合わせ (VOS3 PDMII E2, VOS1 PDMII E2)

定数	列属性			
	整数定数 10 進数定数	浮動小数点定数	文字定数	漢字定数
dbpTypeINTEGER dbpTypeNUMERIC	○	×	×	×
dbpTypeFLOAT dbpTypeREAL	○	○	×	×

定数	列属性			
	整数定数 10進数定数	浮動小数点定数	文字定数	漢字定数
dbpTypeCHAR dbpTypeVARCHAR	×	×	○	×
dbpTypeNCHAR dbpTypeNVARCHAR	×	×	×	○

(凡例)

- ：比較できます。
- ×

(b) 列と列の比較条件の規則

- 列と列の比較は、列属性によっては使えない組み合わせがあります。比較できる列属性の組み合わせを表 8-30 に示します。

表 8-30 比較できる列属性の組み合わせ (VOS3 PDMII E2, VOS1 PDMII E2)

左辺の列属性	右辺の列属性		
	dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	dbpTypeCHAR dbpTypeVARCHAR	dbpTypeNCHAR dbpTypeNVARCHAR
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	○	×	×
dbpTypeCHAR dbpTypeVARCHAR	×	○	×
dbpTypeNCHAR dbpTypeNVARCHAR	×	×	○

(凡例)

- ：比較できます。
- ×

(c) IN 条件の規則

- 定数の設定規則は、「表 8-29 列属性と使える定数の組み合わせ (VOS3 PDMII E2, VOS1 PDMII E2)」に従ってください。
- 定数は、255 個まで設定できます。

(d) LIKE 条件の規則

- LIKE 条件の設定は、列属性によっては使えないことがあります。LIKE 条件が設定できる列属性を表 8-31 に示します。

表 8-31 LIKE 条件が設定できる列属性 (VOS3 PDMII E2, VOS1 PDMII E2)

列属性	設定できるかどうか
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	×
dbpTypeCHAR dbpTypeNCHAR dbpTypeVARCHAR dbpTypeNVARCHAR	○

(凡例)

- ：指定できます。
- ×：指定できません。

- 列と文字パターン（定数）の規則は、「表 8-29 列属性と使える定数の組み合わせ (VOS3 PDMII E2, VOS1 PDMII E2)」に従ってください。
- 「_」、「%」は、指定する列属性に応じて、全角又は半角を使ってください。

定数	文字種別
文字定数	半角文字の「_」、「%」
各国文字定数	全角文字の「_」、「%」

- 文字パターンの指定方法を表 8-32 に示します。

表 8-32 文字パターンの指定方法 (VOS3 PDMII E2, VOS1 PDMII E2)

項目	指定例	意味
前方一致	ABC%	先頭が「ABC」で、以降は任意の文字列を検索します。
後方一致	%ABC	最後が「ABC」で、それより前は任意の文字列を検索します。
任意一致	%ABC%	任意の部分が「ABC」の文字列を検索します。
完全一致	ABC	「ABC」と等しい文字列を検索します。
部分一致	_ _ABC_	6文字のうち、3~5文字目が「ABC」で、ほかの部分は任意の文字列を検索します。
上記以外	ABC%DE	先頭が「ABC」で、最後が「DE」の文字列を検索します。
	%ABC%DE%	任意の部分に「ABC」を含み、以降の任意の部分に「DE」を含む文字列を検索します。
	ABC_ _DE%	先頭が「ABC」で、2文字の任意の文字列をはさんで「DE」を含む文字列を検索します。

- 「%」は、二つ以上設定できません。「%」を二つ設定できるのは、任意一致のパターンだけです。「A%B%C」のようなパターンは設定できません。

(e) BETWEEN 条件の規則

- 列と定数の規則は、「表 8-29 列属性と使える定数の組み合わせ (VOS3 PDMII E2, VOS1 PDMII E2)」に従ってください。

(f) NULL 条件の規則

- NULL 条件は使えません。

(g) 列演算条件の規則

- 演算を指定できる定数を次に示します。
整数定数, 10 進数定数, 浮動小数点定数
- 演算を指定できる列属性を次に示します。
dbpTypeINTEGER, dbpTypeNUMERIC, dbpTypeFLOAT, dbpTypeREAL
- 演算は、各辺の「+」, 「-」, 「*」, 「/」, 又は「(」, 「)」の総和が 64 個まで指定できます。
- 列と定数を比較するときは、「表 8-29 列属性と使える定数の組み合わせ (VOS3 PDMII E2, VOS1 PDMII E2)」に従ってください。

(6) 条件関連 (Conditions.Relation プロパティ) の規則

- NOT は使えません。
- 括弧, AND 及び OR の優先順位は、括弧 > AND > OR です。
- 条件関連のネスト数は 64 までです。ネスト数は、AND 又は OR に対して優先順位を表す括弧を省略しないで指定したときの、左括弧の数です。

(例)

1 AND ((2 OR 3 AND 4) OR 5) AND 6 の場合、括弧を省略しないと (1 AND (((2 OR 3) AND 4) OR 5) AND 6) になって、ネスト数は 4 です。

- 括弧, AND 及び OR は、すべて半角文字で指定してください。
- AND 及び OR の前後には空白を入力してください。

(7) 各条件で指定する定数の規則

- 各条件で指定する定数の規則を表 8-33 と表 8-34 に示します。

表 8-33 定数の指定方法 (VOS3 PDMII E2, VOS1 PDMII E2)

定数の種類	指定方法
整数定数 (2 バイト)	[符号] 符号なし整数

定数の種類	指定方法
整数定数 (4 バイト)	[符号] 符号なし整数
10 進数定数	[符号] 整数部.小数部
浮動小数点定数	[符号] 仮数 E [符号] 指数
文字定数	['] 半角の文字列 [']
漢字定数	['] 全角の文字列 [']

表 8-34 定数の指定範囲 (VOS3 PDMII E2, VOS1 PDMII E2)

定数の種類	指定範囲	指定例
整数定数 (2 バイト)	-32,768 ~ 32,767	123, -123
整数定数 (4 バイト)	-2,147,483,648 ~ 2,147,483,647	
10 進数定数	m: 全けた数 n: 小数点以下のけた数 M: 列定義の全けた数 N: 列定義の小数点けた数 $1 \leq m \leq M, 1 \leq n \leq N, n \leq m$	45.6, -45.6
浮動小数点定数	仮数 16 けた以内 指数 -99 ~ 75	5.1E3, 5.1E-3
文字定数	1 ~ 255 バイト	ABC
漢字定数	1 ~ 127 文字	テレビ

8.4.5 VOS3 RDB1 の場合の規則

(1) 表情報の値

表情報で返される値の意味を表 8-35 に示します。

表 8-35 表情報で返される値の意味 (VOS3 RDB1)

プロパティ/メソッド	値の説明
TableComment	表の注釈
TableCreateDate	'YYMMDDhhmmss' (YY: 西暦の下 2 けた MM: 月 DD: 日 hh: 時 mm: 分 ss: 秒)
TableDBMName	長さ 0 バイトの文字列
TableGroupName	表のグループ名
TableHeader	表の見出し
TableName	表名
TableOwner	表の所有者

(2) 列情報の値

列情報で返される値の意味を表 8-36 に示します。

表 8-36 列情報で返される値の意味 (VOS3 RDB1)

プロパティ/メソッド	値の説明
ColumnComment	列の注釈
ColumnEditPattern	編集形式文字列
ColumnHeader	列の見出し
ColumnName	列名

(3) 集合関数 (FunctionMode プロパティ) の規則

- dbpGroup は、21 列まで設定できます。ただし、ソートを設定した列数との和が 22 列以上は設定できません。
- 集合関数の設定は、列属性によっては使えない集合関数があります。列属性と使える集合関数の組み合わせを表 8-37 に示します。

表 8-37 列属性と使える集合関数の組み合わせ (VOS3 RDB1)

列属性	dbpGroup	dbpMaximum	dbpMinimum	dbpAverage	dbpSum	dbpCount
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	○	○	○	○	○	○
dbpTypeCHAR dbpTypeVARCHAR dbpTypeNCHAR dbpTypeNVARCHAR	○	○	○	×	×	○

(凡例)

- ：指定できます。
- ×：指定できません。

(4) ソート (Column.SortMode プロパティ) の規則

- 21 列まで指定できます。ただし、dbpGroup を設定した列数との和が 22 列以上は設定できません。

(5) 抽出条件 (Conditions.Add メソッド) の規則

(a) 列と定数の比較条件の規則

- 列と定数の比較は、列属性によっては使えない定数があります。列属性と使える定数の組み合わせを表 8-38 に示します。

表 8-38 列属性と使える定数の組み合わせ (VOS3 RDB1)

列属性	定数			
	整数定数 10 進数定数	浮動小数点定数	文字定数	漢字定数
dbpTypeINTEGER dbpTypeNUMERIC	○	×	×	×
dbpTypeFLOAT dbpTypeREAL	○	○	×	×
dbpTypeCHAR dbpTypeVARCHAR	×	×	○	×
dbpTypeNCHAR dbpTypeNVARCHAR	×	×	×	○

(凡例)

- ：比較できます。
- ×

(b) 列と列の比較条件の規則

- 列と列の比較は、列属性によっては使えない組み合わせがあります。比較できる列属性の組み合わせを表 8-39 に示します。

表 8-39 比較できる列属性の組み合わせ (VOS3 RDB1)

左辺の列属性	右辺の列属性		
	dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	dbpTypeCHAR dbpTypeVARCHAR	dbpTypeNCHAR dbpTypeNVARCHAR
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	○	×	×
dbpTypeCHAR dbpTypeVARCHAR	×	○	×
dbpTypeNCHAR	×	×	○

左辺の列属性	右辺の列属性		
		dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	dbpTypeCHAR dbpTypeVARCHAR
dbpTypeNVARCHAR			

(凡例)

- ：比較できます。
- ×：比較できません。

(c) IN 条件の規則

- 定数の設定規則は、「表 8-38 列属性と使える定数の組み合わせ (VOS3 RDB1)」に従ってください。
- 定数は、255 個まで設定できます。

(d) LIKE 条件の規則

- LIKE 条件の設定は、列属性によっては使えないことがあります。LIKE 条件が設定できる列属性を表 8-40 に示します。

表 8-40 LIKE 条件が設定できる列属性 (VOS3 RDB1)

列属性	設定できるかどうか
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	×
dbpTypeCHAR dbpTypeVARCHAR dbpTypeNCHAR dbpTypeNVARCHAR	○

(凡例)

- ：指定できます。
- ×：指定できません。

- 列と文字パターン (定数) の規則は、「表 8-38 列属性と使える定数の組み合わせ (VOS3 RDB1)」に従ってください。
- 「_」, 「%」は、指定する列属性に応じて、全角又は半角を使ってください。

定数	文字種別
文字定数	半角文字の「_」, 「%」
各国文字定数	全角文字の「_」, 「%」

- 文字パターンの指定方法を表 8-41 に示します。

表 8-41 文字パターンの指定方法 (VOS3 RDB1)

項目	指定例	意味
前方一致	ABC%	先頭が「ABC」で、以降は任意の文字列を検索します。
後方一致	%ABC	最後が「ABC」で、それより前は任意の文字列を検索します。
任意一致	%ABC%	任意の部分が「ABC」の文字列を検索します。
完全一致	ABC	「ABC」と等しい文字列を検索します。
部分一致	_ _ABC_	6文字のうち、3~5文字目が「ABC」で、ほかの部分は任意の文字列を検索します。
上記以外	ABC%DE	先頭が「ABC」で、最後が「DE」の文字列を検索します。
	ABC_ _DE%	先頭が「ABC」で、2文字の任意の文字列をはさんで「DE」を含む文字列を検索します。

- 「%」は、二つ以上設定できません。「%」を二つ以上設定できるパターンは、任意一致のパターンだけです。A%B%Cのようなパターンは設定できません。

(e) BETWEEN 条件の規則

- 列と定数の規則は、「表 8-38 列属性と使える定数の組み合わせ (VOS3 RDB1)」に従ってください。

(f) NULL 条件の規則

- 制限事項はありません。

(g) 列演算条件の規則

- 演算を指定できる定数を次に示します。
整数定数, 10進数定数, 浮動小数点定数
- 演算を指定できる列属性を次に示します。
dbpTypeINTEGER, dbpTypeNUMERIC, dbpTypeFLOAT, dbpTypeREAL
- 演算は、各辺の「+」, 「-」, 「*」, 「/」, 又は「(」, 「)」の総和が64個まで指定できます。
- 列と定数を比較するときは、「表 8-38 列属性と使える定数の組み合わせ (VOS3 RDB1)」に従ってください。

(6) 条件関連 (Conditions.Relation プロパティ) の規則

- NOT は使えません。
- 括弧, AND 及び OR の優先順位は、括弧 > AND > OR です。
- 条件関連のネスト数は64までです。ネスト数はAND又はORに対して、優先順位を表す括弧を省略しないで指定したときの、左括弧の数です。

(例)

1 AND ((2 OR 3 AND 4) OR 5) AND 6 のときに、括弧を省略しないと (1 AND (((2 OR 3) AND 4) OR 5) AND 6) になって、ネスト数は4です。

- 括弧、AND 及び OR は、すべて半角文字で指定してください。
- AND 及び OR の前後には空白を入力してください。

(7) 各条件で指定する定数の規則

- 各条件で指定する定数の規則を表 8-42 と表 8-43 に示します。

表 8-42 定数の指定方法 (VOS3 RDB1)

定数の種類	指定方法
整数定数 (2 バイト)	[符号] 符号なし整数
整数定数 (4 バイト)	[符号] 符号なし整数
10 進数定数	[符号] 整数部.小数部
浮動小数点定数	[符号] 仮数 E [符号] 指数
文字定数	['] 半角の文字列 [']
漢字定数	['] 全角の文字列 [']

表 8-43 定数の指定範囲 (VOS3 RDB1)

定数の種類	指定範囲	指定例
整数定数 (2 バイト)	-32,768 ~ 32,767	123, -123
整数定数 (4 バイト)	-2,147,483,648 ~ 2,147,483,647	
10 進数定数	m: 全けた数 n: 小数点以下のけた数 M: 列定義の全けた数 N: 列定義の小数点けた数 $1 \leq m \leq M, 1 \leq n \leq N, n \leq m$	45.6, -45.6
浮動小数点定数	仮数 16 けた以内 指数 -99 ~ 75	5.1E3, 5.1E-3
文字定数	1 ~ 255 バイト	ABC
漢字定数	1 ~ 127 文字	テレビ

8.4.6 VOS1 ACE3 ユーザファイルの場合の規則

(1) 表情報の値

表情報で返される値の意味を表 8-44 に示します。

表 8-44 表情報で返される値の意味 (VOS1 ACE3 ユーザファイル)

プロパティ/メソッド	値の説明
TableComment	表の注釈
TableCreateDate	'YYMMDDhhmmss' (YY: 西暦の下 2 けた MM: 月 DD: 日 hh: 時 mm: 分 ss: 秒)
TableDBMName	長さ 0 バイトの文字列
TableGroupName	表のグループ名
TableHeader	表の見出し
TableName	表名 (ただし, ACE3 の設定でデータベースと混在させているときは, 表名の後ろに「/F」が付きます)
TableOwner	表の所有者

(2) 列情報の値

列情報で返される値の意味を表 8-45 に示します。

表 8-45 列情報で返される値の意味 (VOS1 ACE3 ユーザファイル)

プロパティ/メソッド	値の説明
ColumnComment	列の注釈
ColumnEditPattern	編集形式文字列
ColumnHeader	列の見出し
ColumnName	列名

(3) 集合関数 (FunctionMode プロパティ) の規則

- dbpGroup は, 21 列まで設定できます。ただし, ソートを設定した列数との和が 22 列以上は設定できません。
- 集合関数の設定は, 列属性によっては使えない集合関数があります。列属性と使える集合関数を表 8-46 に示します。

表 8-46 列属性と使える集合関数の組み合わせ (VOS1 ACE3 ユーザファイル)

列属性	dbpGroup	dbpMaximum	dbpMinimum	dbpAverage	dbpSum	dbpCount
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	○	○	○	○	○	○
dbpTypeCHAR	○	○	○	×	×	○

列属性	dbpGroup	dbpMaximum	dbpMinimum	dbpAverage	dbpSum	dbpCount
dbpTypeVARCHAR						
dbpTypeNCHAR dbpTypeNVARCHAR	○	○	○	×	×	○
dbpTypeMCHAR dbpTypeMVARCHAR	○	○	○	×	×	○

(凡例)

- ：指定できます。
- ×

(4) ソート (*Column.SortMode* プロパティ) の規則

- 21 列まで指定できます。ただし、dbpGroup を設定した列数との和が 22 列以上は設定できません。

(5) 抽出条件 (*Conditions.Add* メソッド) の規則

(a) 列と定数の比較条件の規則

- 列と定数の比較は、列属性によっては使えない定数があります。列属性と使える定数の組み合わせを表 8-47 に示します。

表 8-47 列属性と使える定数の組み合わせ (VOS1 ACE3 ユーザファイル)

列属性	定数			
	整数定数 10 進数定数 浮動小数点定数	文字定数	漢字定数	混在文字定数
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	○	×	×	×
dbpTypeCHAR dbpTypeVARCHAR	×	○	×	×
dbpTypeNCHAR dbpTypeNVARCHAR	×	×	○	×
dbpTypeMCHAR dbpTypeMVARCHAR	×	○	○	○

(凡例)

- ：比較できます。
- ×

(b) 列と列の比較条件の規則

- 列と列の比較は、列属性によっては使えない組み合わせがあります。比較できる列属性の組み合わせを表 8-48 に示します。

表 8-48 比較できる列属性の組み合わせ (VOS1 ACE3 ユーザファイル)

左辺の列属性	右辺の列属性			
	dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	dbpTypeCHAR dbpTypeVARCHAR	dbpTypeNCHAR dbpTypeNVARCHAR	dbpTypeMCHAR dbpTypeMVARCHAR
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	○	×	×	×
dbpTypeCHAR dbpTypeVARCHAR	×	○	×	○
dbpTypeNCHAR dbpTypeNVARCHAR	×	×	○	×
dbpTypeMCHAR dbpTypeMVARCHAR	×	○	×	○

(凡例)

- ：比較できます。
- ×

(c) IN 条件の規則

- 定数の設定規則は、「表 8-47 列属性と使える定数の組み合わせ (VOS1 ACE3 ユーザファイル)」に従ってください。
- 定数は、255 個まで設定できます。

(d) LIKE 条件の規則

- LIKE 条件の設定は、列属性によっては使えないことがあります。LIKE 条件が設定できる列属性を表 8-49 に示します。

表 8-49 LIKE 条件が設定できる列属性 (VOS1 ACE3 ユーザファイル)

列属性	設定できるかどうか
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeFLOAT dbpTypeREAL	×

列属性	設定できるかどうか
dbpTypeCHAR	○
dbpTypeNCHAR	
dbpTypeVARCHAR	
dbpTypeNCHAR	
dbpTypeNVARCHAR	
dbpTypeMCHAR	
dbpTypeMVARCHAR	

(凡例)

- ：指定できます。
- ×：指定できません。

- 列と文字パターン（定数）の規則は、「表 8-47 列属性と使える定数の組み合わせ（VOS1 ACE3 ユーザファイル）」に従ってください。
- 「_」, 「%」は、指定する列属性に応じて、全角又は半角を使ってください。

定数	文字種別
文字定数	半角文字の「_」, 「%」
各国文字定数	全角文字の「_」, 「%」
混在文字定数	半角文字の「_」, 「%」

- 文字パターンの指定方法を表 8-50 に示します。

表 8-50 文字パターンの指定方法（VOS1 ACE3 ユーザファイル）

項目	指定例	意味
前方一致	ABC%	先頭が「ABC」で、以降は任意の文字列を検索します。
後方一致	%ABC	最後が「ABC」で、それより前は任意の文字列を検索します。
任意一致	%ABC%	任意の部分が「ABC」の文字列を検索します。
完全一致	ABC	「ABC」と等しい文字列を検索します。
部分一致	_ _ABC_	6文字のうち、3~5文字目が「ABC」で、ほかの部分は任意の文字列を検索します。
上記以外	ABC%DE	先頭が「ABC」で、最後が「DE」の文字列を検索します。
	ABC_ _DE%	先頭が「ABC」で、2文字の任意の文字列をはさんで「DE」を含む文字列を検索します。

- 「%」は、二つ以上設定できません。「%」を二つ設定できるのは、任意一致のパターンだけです。「A %B %C」のようなパターンは設定できません。

(e) BETWEEN 条件の規則

- 列と定数の規則は、「表 8-47 列属性と使える定数の組み合わせ（VOS1 ACE3 ユーザファイル）」に従ってください。

(f) NULL 条件の規則

- 制限事項はありません。

(g) 列演算条件の規則

- 演算を指定できる定数を次に示します。
整数定数, 10 進数定数, 浮動小数点定数
- 演算を指定できる列属性を次に示します。
dbpTypeINTEGER, dbpTypeNUMERIC, dbpTypeFLOAT, dbpTypeREAL
- 演算は, 各辺の「+」, 「-」, 「*」, 「/」, 又は「(」, 「)」の総和が 64 個まで指定できます。
- 列と定数を比較するときは, 「表 8-47 列属性と使える定数の組み合わせ (VOS1 ACE3 ユーザファイル)」に従ってください。

(6) 条件関連 (Conditions.Relation プロパティ) の規則

- NOT は使えません。
- 括弧, AND 及び OR の優先順位は, 括弧 > AND > OR です。
- 条件関連のネスト数は 64 までです。ネスト数は AND 又は OR に対して, 優先順位を表す括弧を省略しないで指定したときの, 左括弧の数です。

(例)

1 AND ((2 OR 3 AND 4) OR 5) AND 6 のときは, 括弧を省略しないと (1 AND ((2 OR 3) AND 4) OR 5) AND 6) になって, ネスト数は 4 です。

- 括弧, AND 及び OR は, すべて半角文字で指定してください。
- AND 及び OR の前後には空白を入力してください。

(7) 各条件で指定する定数の規則

- 各条件で指定する定数の規則を表 8-51 と表 8-52 に示します。

表 8-51 定数の指定方法 (VOS1 ACE3 ユーザファイル)

定数の種類	指定方法
整数定数 (2 バイト)	[符号] 符号なし整数
整数定数 (4 バイト)	[符号] 符号なし整数
10 進数定数	[符号] 整数部.小数部
浮動小数点定数	[符号] 仮数 E [符号] 指数
文字定数	['] 半角の文字列 [']
漢字定数	['] 全角の文字列 [']

定数の種類	指定方法
混在文字定数	['] 全角及び半角の文字列 [']

表 8-52 定数の指定範囲 (VOS1 ACE3 ユーザファイル)

定数の種類	指定範囲	指定例
整数定数 (2 バイト)	-32,768 ~ 32,767	123, -123
整数定数 (4 バイト)	-2,147,483,648 ~ 2,147,483,647	
10 進数定数	m : 全けた数 n : 小数点以下のけた数 M : 列定義の全けた数 N : 列定義の小数点けた数 $1 \leq m \leq M, 1 \leq n \leq N, n \leq m$	45.6, -45.6
浮動小数点定数	仮数 16 けた以内 指数 -99~75	5.1E3, 5.1E-3
文字定数	1~254 バイト	ABC
漢字定数	1~127 文字	テレビ
混在文字定数	1~254 バイト	テレビ ABC

8.4.7 VOS3 ACE3 ユーザファイルの場合の規則

(1) 表情報の値

表情報で返される値の意味を表 8-53 に示します。

表 8-53 表情報で返される値の意味 (VOS3 ACE3 ユーザファイル)

プロパティ/メソッド	値の説明
TableComment	表の注釈
TableCreateDate	'YYMMDDhhmmss' (YY : 西暦の下 2 けた MM : 月 DD : 日 hh : 時 mm : 分 ss : 秒)
TableDBMName	長さ 0 バイトの文字列
TableGroupName	表のグループ名
TableHeader	表の見出し
TableName	表名 ACE3 の設定でデータベースと混在させているときは、表名の後ろに「/F」が付きます。
TableOwner	表の所有者 所有者が 8 バイト未満の場合は、後ろに空白文字が埋められます。

(2) 列情報の値

列情報で返される値の意味を表 8-54 に示します。

表 8-54 列情報で返される値の意味 (VOS3 ACE3 ユーザファイル)

プロパティ/メソッド	値の説明
ColumnComment	列の注釈
ColumnEditPattern	編集形式文字列
ColumnHeader	列の見出し
ColumnName	列名

(3) 集合関数 (FunctionMode プロパティ) の規則

- dbpGroup は、21 列まで設定できます。ただし、ソートを設定した列数との和が 22 列以上は設定できません。
- 集合関数の設定は、列属性によっては使えない集合関数があります。列属性と使える集合関数の組み合わせを表 8-55 に示します。

表 8-55 列属性と使える集合関数の組み合わせ (VOS3 ACE3 ユーザファイル)

列属性	dbpGroup	dbpMaximum	dbpMinimum	dbpAverage	dbpSum	dbpCount
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeNUMERICUNS dbpTypeFLOAT dbpTypeREAL	○	○	○	○	○	○
dbpTypeCHAR dbpTypeVARCHAR dbpTypeNCHAR dbpTypeNVARCHAR dbpTypeMCHAR dbpTypeMVARCHAR dbpTypeDATE dbpTypeINTERVAL_YEAR_TO_DAY dbpTypeTIME dbpTypeINTERVAL_HOUR_TO_SECOND	○	○	○	×	×	○

(凡例)

- ：指定できます。
- ×：指定できません。

(4) ソート (Column.SortMode プロパティ) の規則

- 21 列まで指定できます。ただし、dbpGroup を設定した列数との和が 22 列以上は設定できません。

(5) 抽出条件 (Conditions.Add メソッド) の規則

(a) 列と定数の比較条件の規則

- 列と定数の比較は、列属性によっては使えない定数があります。列属性と使える定数の組み合わせを表 8-56 に示します。

表 8-56 列属性と使える定数の組み合わせ (VOS3 ACE3 ユーザファイル)

列属性	定数			
	整数定数 10 進数定数 浮動小数点定数	文字定数	漢字定数	混在文字定数
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeNUMERICUNS dbpTypeFLOAT dbpTypeREAL	○	×	×	×
dbpTypeCHAR dbpTypeVARCHAR	×	○	×	×
dbpTypeNCHAR dbpTypeNVARCHAR	×	×	○	×
dbpTypeMCHAR dbpTypeMVARCHAR	×	○	○	○
dbpTypeDATE dbpTypeINTERVAL_YEAR_T O_DAY dbpTypeTIME dbpTypeINTERVAL_HOUR_T O_SECOND	×	×	×	×

(凡例)

- ：比較できます。
- ×：比較できません。

(b) 列と列の比較条件の規則

- 列と列の比較は、列属性によっては使えない組み合わせがあります。比較できる列属性の組み合わせを表 8-57 に示します。

表 8-57 比較できる列属性の組み合わせ (VOS3 ACE3 ユーザファイル)

左辺の列属性	右辺の列属性				
	dbpTypeINTEGER dbpTypeNUMERIC dbpTypeNUMERICUNS dbpTypeFLOAT dbpTypeREAL	dbpTypeCHAR dbpTypeVARCHAR	dbpTypeNCHAR dbpTypeNVARCHAR	dbpTypeMCHAR dbpTypeMVARCHAR	dbpTypeDATE dbpTypeINTERVAL_YEAR_TO_DAY dbpTypeTIME dbpTypeINTERVAL_HOUR_TO_SECOND
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeNUMERICUNS dbpTypeFLOAT dbpTypeREAL	○	×	×	×	×
dbpTypeCHAR dbpTypeVARCHAR	×	○	×	○	×
dbpTypeNCHAR dbpTypeNVARCHAR	×	×	○	×	×
dbpTypeMCHAR dbpTypeMVARCHAR	×	○	×	○	×
dbpTypeDATE dbpTypeINTERVAL_YEAR_TO_DAY dbpTypeTIME dbpTypeINTERVAL_HOUR_TO_SECOND	×	×	×	×	×

(凡例)

- ：比較できます。
- ×：比較できません。

(c) IN 条件の規則

- 定数の設定規則は、「表 8-56 列属性と使える定数の組み合わせ (VOS3 ACE3 ユーザファイル)」に従ってください。

- 定数は、255 個まで設定できます。

(d) LIKE 条件の規則

- LIKE 条件の設定は、列属性によっては使えないことがあります。LIKE 条件が設定できる列属性を表 8-58 に示します。

表 8-58 LIKE 条件が設定できる列属性 (VOS3 ACE3 ユーザファイル)

列属性	設定できるかどうか
dbpTypeINTEGER dbpTypeNUMERIC dbpTypeNUMERICUNS dbpTypeFLOAT dbpTypeREAL dbpTypeDATE dbpTypeINTERVAL_YEAR_TO_DAY dbpTypeTIME dbpTypeINTERVAL_HOUR_TO_SECOND	×
dbpTypeCHAR dbpTypeNCHAR dbpTypeVARCHAR dbpTypeNVARCHAR dbpTypeMCHAR dbpTypeMVARCHAR	○

(凡例)

- ：指定できます。
- ×：指定できません。

- 列と文字パターン（定数）の規則は、「表 8-56 列属性と使える定数の組み合わせ (VOS3 ACE3 ユーザファイル)」に従ってください。
- 「_」、「%」は、指定する列属性に応じて、全角又は半角を使ってください。

定数	文字種別
文字定数	半角文字の「_」、「%」
各国文字定数	全角文字の「_」、「%」
混在文字定数	半角文字の「_」、「%」

- 文字パターンの指定方法を表 8-59 に示します。

表 8-59 文字パターンの指定方法 (VOS3 ACE3 ユーザファイル)

項目	指定例	意味
前方一致	ABC%	先頭が「ABC」で、以降は任意の文字列を検索します。
後方一致	%ABC	最後が「ABC」で、それより前は任意の文字列を検索します。

項目	指定例	意味
任意一致	%ABC%	任意の部分が「ABC」の文字列を検索します。
完全一致	ABC	「ABC」と等しい文字列を検索します。
部分一致	_ _ABC_	6文字のうち、3~5文字目が「ABC」で、ほかの部分は任意の文字列を検索します。
上記以外	ABC%DE	先頭が「ABC」で、最後が「DE」の文字列を検索します。
	ABC_ _DE%	先頭が「ABC」で、2文字の任意の文字列をはさんで「DE」を含む文字列を検索します。

- 「%」は、二つ以上設定できません。「%」を二つ設定できるのは、任意一致のパターンだけです。「A%B%C」のようなパターンは設定できません。

(e) BETWEEN 条件の規則

- 列と定数の規則は、「表 8-56 列属性と使える定数の組み合わせ (VOS3 ACE3 ユーザファイル)」に従ってください。

(f) NULL 条件の規則

- 制限事項はありません。

(g) 列演算条件の規則

- 演算を指定できる定数を次に示します。
整数定数, 10進数定数, 浮動小数点定数
- 演算を指定できる列属性を次に示します。
dbpTypeINTEGER, dbpTypeNUMERIC, dbpTypeNUMERICUNS, dbpTypeFLOAT, dbpTypeREAL
- 演算は、各辺の「+」, 「-」, 「*」, 「/」, 又は「(」, 「)」の総和が64個まで指定できます。
- 列と定数を比較するときは、「表 8-56 列属性と使える定数の組み合わせ (VOS3 ACE3 ユーザファイル)」に従ってください。

(6) 条件関連 (Conditions.Relation プロパティ) の規則

- NOT は使えません。
- 括弧, AND 及び OR の優先順位は、括弧 > AND > OR です。
- 条件関連のネスト数は64までです。ネスト数はAND又はORに対して、優先順位を表す括弧を省略しないで指定したときの、左括弧の数です。

(例)

1 AND ((2 OR 3 AND 4) OR 5) AND 6 のときに、括弧を省略しないと (1 AND (((2 OR 3) AND 4) OR 5) AND 6) になって、ネスト数は4です。

- 括弧, AND 及び OR は、すべて半角文字で指定してください。

- AND 及び OR の前後には空白を入力してください。

(7) 各条件で指定する定数の規則

- 各条件で指定する定数の規則を表 8-60 と表 8-61 に示します。

表 8-60 定数の指定方法 (VOS3 ACE3 ユーザファイル)

定数の種類	指定方法
整数定数 (2 バイト)	[符号] 符号なし整数
整数定数 (4 バイト)	[符号] 符号なし整数
10 進数定数	[符号] 整数部.小数部
浮動小数点定数	[符号] 仮数 E [符号] 指数
文字定数	['] 半角の文字列 [']
漢字定数	['] 全角の文字列 [']
混在文字定数	['] 全角及び半角の文字列 [']

表 8-61 定数の指定範囲 (VOS3 ACE3 ユーザファイル)

定数の種類	指定範囲	指定例
整数定数 (2 バイト)	-32,768 ~ 32,767	123, -123
整数定数 (4 バイト)	-2,147,483,648 ~ 2,147,483,647	
10 進数定数	m : 全けた数 n : 小数点以下のけた数 M : 列定義の全けた数 N : 列定義の小数点けた数 $1 \leq m \leq M, 1 \leq n \leq N, n \leq m$	45.6, -45.6
浮動小数点定数	仮数 16 けた以内 指数 -99 ~ 75	5.1E3, 5.1E-3
文字定数	1 ~ 254 バイト	ABC
漢字定数	1 ~ 127 文字	テレビ
混在文字定数	1 ~ 254 バイト	テレビ ABC

8.4.8 VOS1 スプールファイルの場合の規則

(1) 表情報の値

表情報で返される値の意味を表 8-62 に示します。

表 8-62 表情報で返される値の意味 (VOS1 スプールファイル)

プロパティ/メソッド	値の説明
TableComment	ジョブに付けたコメント情報
TableCreateDate	'MM/DD hh:mm' (MM:月 DD:日 hh:時 mm:分)
TableDBMName	長さ 0 バイトの文字列
TableGroupName	表のグループ名
TableHeader	ジョブの名称
TableName	「ジョブ通し番号」, 「分割番号」, 「出カクラス」 を連結した文字列
TableOwner	ジョブを実行したユーザ ID

(2) 列情報の値

列情報は取得できません。

(3) 集合関数 (FunctionMode プロパティ) の規則

集合関数は設定できません。

(4) ソート (Column.SortMode プロパティ) の規則

ソートは設定できません。

(5) 抽出条件 (Conditions.Add メソッド) の規則

抽出条件は設定できません。

(6) 条件関連 (Conditions.Relation プロパティ) の規則

条件関連は設定できません。

8.4.9 VOSK スプールファイルの場合の規則

(1) 表情報の値

表情報で返される値の意味を表 8-63 に示します。

表 8-63 表情報で返される値の意味 (VOSK スプールファイル)

プロパティ/メソッド	値の説明
TableComment	ジョブに付けたコメント情報

プロパティ/メソッド	値の説明
TableCreateDate	'YYMMDDhhmmss' (YY:西暦の下2けた MM:月 DD:日 hh:時 mm:分 ss:秒)
TableDBMName	長さ0バイトの文字列
TableGroupName	表のグループ名
TableHeader	ジョブの名称
TableName	「ジョブ通し番号」, 「ファイル通し番号」を連結した文字列
TableOwner	ジョブを実行したユーザID

(2) 列情報の値

列情報は取得できません。

(3) 集合関数 (FunctionMode プロパティ) の規則

集合関数は設定できません。

(4) ソート (Column.SortMode プロパティ) の規則

ソートは設定できません。

(5) 抽出条件 (Conditions.Add メソッド) の規則

抽出条件は設定できません。

(6) 条件関連 (Conditions.Relation プロパティ) の規則

条件関連は設定できません。

9

トラブルが起こったときは

この章では、DBPARTNER2 Client にトラブルが起こったときの対処について説明します。

9.1 トラブルの現象と対処

トラブルの現象と対処を示します。

9.1.1 インストール時に起こるトラブル

- ODBC ドライバのインストールで「上書きできないためインストールできない」ことを知らせるメッセージが出力された。
[無視] ボタンをクリックして、インストールを続行してください。
- ODBC ドライバのインストールで、インストール完了の通知がないまま終了した。
IME97 の対策版がインストールされていません。Microsoft 社に問い合わせ、IME97 の対策版入手してください。

9.1.2 データベースと接続するとき起こるトラブル

- DBPARTNER2 Client のドライバが一覧に表示されない。
ODBC ドライバをインストールしてください。インストール手順については、「[2.3 ODBC ドライバのセットアップ](#)」を参照してください。
- データソースの一覧に、何も表示されない。
ODBC 接続でデータソースに接続先を設定していないときは、[コントロールパネル] - [ODBC データソース] アイコンを開いて [ユーザ DSN] で接続先を設定してください。設定する手順については、「[2.3 ODBC ドライバのセットアップ](#)」を参照してください。
- ODBC ドライバでエラーが表示された。
 - HiRDB に接続する場合
識別子を指定する必要はありません。
 - ORACLE に接続する場合
DABroker と ORACLE が同じマシン (WS 又は PC) で動作している環境のときは、ネットサービス名を指定する必要はありません。
DABroker と ORACLE が異なるマシン (WS 又は PC) で動作している環境、又は複数の ORACLE が動作している環境では、ネットサービス名を指定する必要があります。
DABroker が動作しているマシン (WS 又は PC) から ORACLE のユーティリティを実行するときに指定するネットサービス名を指定してください。
- エラー 10709 (サーバとの接続に失敗) が出力された。
 - CommuniNet と CommuniNet Extension の前提バージョンが正しいかどうかを確認してください。
 - CommuniNet の自動実行マクロを設定しているときは、設定を削除してください。

- CommuniNet を表示させた状態で接続処理を実行して、ログオンマクロスクリプトに設定した文字列が正しく画面に表示されているかどうかを確認してください。
- ログオンマクロスクリプトの実行中にエラーになるときは、マクロスクリプトの実行待ち時間を大きくしてください。
- 10709 「サーバとの接続に失敗しました」 が返される。
 - CommuniNet と CommuniNet Extension の前提バージョンが正しいか確認してください。
 - CommuniNet の自動実行マクロが設定されている場合は、設定を削除してください
 - CommuniNet を表示させた状態から接続処理をして、LOGON マクロスクリプトに設定した文字列が正しく画面に入力されているか確認してください。
 - LOGON マクロスクリプト実行中にエラーが発生する場合は、マクロスクリプト実行待ち時間を大きくしてください。

9.1.3 DBPARTNER2 Client の操作時に起こるトラブル

- データを更新するメソッドを使ったアプリケーションがエラーになる。
インストールしている DBPARTNER2 Client の型名（参照版／更新版／定義版）が正しいかどうかを確認してください。
- 更新機能のメソッドを使った AP がエラーとなる。
インストールしている DBPARTNER2 Client の形名を確認してください。

9.1.4 システムに関するトラブル

- アプリケーションが異常終了した。
アプリケーション又は DBPARTNER2 Client のどこが原因で異常終了したか示すメッセージが表示されます。DBPARTNER2 Client に原因があるときは、保守契約内容に従って対処してください。
- 処理中に DBPARTNER2 Client によって勝手にセッションが切られてしまった。
ODBC 接続の場合は、ドライバの応答監視時間の設定を確認してください。
- アプリケーションが異常終了した。
DBPARTNER が異常終了したメッセージが表示されます。保守契約内容に従い対処してください。
- 処理中にセッションが切られる。
ODBC 接続の場合、応答監視時間の設定を確認してください。

10

メッセージの一覧

この章では、DBPARTNER2 Client をインストールするときに出力されるメッセージの意味と、DBPARTNER ODBC Driver から通知されるメッセージの意味について説明します。

10.1 メッセージの形式と見方

DBPARTNER2 Client のメッセージの形式と見方について説明します。

10.1.1 メッセージの形式

DBPARTNER2 Client で表示されるメッセージは、メッセージのダイアログボックスに表示されます。形式を次に示します。



- シンボルマーク
メッセージの種類を表したマークです。T (タイプコード) に対応しています。
- XXX
メッセージを表示した製品を示します。
INS : DBPARTNER2 Client のインストール関連のメッセージを示します。
DBL : DBPARTNER ODBC Driver のメッセージを示します。
- nnnn
メッセージの通し番号です。
- T
タイプコードです。次のように対応しています。
 - E (エラー)
情報の誤りが通知されるメッセージです。
 - I (インフォメーション)
情報が通知されるメッセージです。
 - W (ワーニング)
警告が通知されるメッセージです。
- [OK] ボタン
メッセージのダイアログボックスを閉じます。

10.1.2 メッセージの見方

このマニュアルでは、メッセージを次の形式で説明しています。

XXX nnnn T

メッセージテキスト

要因：メッセージが表示された原因

処理：DBPARTNER2 の処理

対処：ユーザの処置

10.1.3 注意

- 「保守担当者に連絡してください」とは、システム管理者が購入時の契約に基づき当社問い合わせ窓口へ連絡することを示します。
- エラーコード、シグナル番号などは、保守担当者が使う情報です。
- サーバプログラムから通知されたメッセージについては、次に示すマニュアルを参照してください。
 - 「DABroker」
- データベースから通知されたメッセージについては、該当するデータベース（DBMS）のマニュアルを参照してください。

10.2 メッセージの詳細

このマニュアルでは、次に示すメッセージについて説明します。

- DBPARTNER2 Client のインストールのメッセージ
- DBPARTNER ODBC Driver (ODBC ドライバ) のメッセージ

CommuniNet 接続のときに表示される、サーバからのメッセージについては、マニュアル「DBPARTNER2 Client 操作ガイド」を参照してください。

10.2.1 DBPARTNER2 Client のインストールのメッセージ

DBPARTNER2 Client をインストールするときに出力されるメッセージを示します。

INS0101E

インストールで使用するフロッピーディスク中のファイルが正しくありません。又はファイルがありません。インストールを終了します。

要因：インストールで使うフロッピーディスク中のファイルに誤りがあるか、又はファイルがありません。

対処：正しいインストール用のフロッピーディスクを使って、インストールを再実行してください。

INS0102W

ディレクトリ *aaaa* を作成することができません。

要因：フォルダ (*aaaa*) を作成できません。

対処：フォルダと同名のファイルがあるおそれがあります。その場合、指定するフォルダを変更してください。その後も同じエラーが繰り返し起こるときは、保守担当者に連絡してください。

INS0103E

Windows のバージョンが違います。インストールを終了します。

要因：DBPARTNER2 Client を稼働できる Windows ではありません。

対処：DBPARTNER2 Client を稼働できる Windows をインストールして、DBPARTNER2 Client のインストールを再実行してください。

INS0104E

新しい DBPARTNER2 Client がインストールされています。インストールを終了します。

要因：インストール先に、現在インストール中の DBPARTNER2 Client よりも新しいバージョンのものがインストールされています。

対処：古いバージョンの DBPARTNER2 Client はインストールできません。同じ又は新しいバージョンの DBPARTNER2 Client をインストールしてください。

INS0105W

指定されたディレクトリが正しくありません。

要因：フォルダ名の指定が間違っています。

対処：フォルダ名を正しく指定して、再実行してください。

INS0106W

ドライブ aa に、DBPARTNER2 Client *nnnn* をインストールするだけの容量がありません。インストールには *mmmmm* バイトの容量を確保してください。

要因：aa で示すインストール先のドライブに、DBPARTNER2 Client をインストールするための容量が不足しています。*mmmmm* バイト以上を確保してください。

nnnn : DBPARTNER2 Client のバージョン番号

対処：インストール先ドライブの不要ファイルの削除などで、DBPARTNER2 Client をインストールするために必要な容量を確保して、インストールを再実行してください。

INS0107W

ドライブの指定が誤っています。正しいドライブを指定してください。

要因：指定されたパスのドライブの指定に間違いがあります。

対処：存在するドライブ名を正しく英字で指定して、再実行してください。

INS0108E

Windows 又はインストールプログラムに異常があります。インストールを終了します (*nnnn*)。

要因：Windows のシステム又はインストールプログラムで異常が発生しました。

nnnn : 保守コード

対処：DBPARTNER2 Client を終了して、保守担当者に連絡してください。

INS0109W

組み込みディレクトリに、必要なファイルがありません。ActiveX コントロールを選択してください。

要因：組み込みフォルダに、必要なファイルがありません。

対処：ActiveX コントロールを選択して、再実行してください。

INS0110W

ディレクトリの長さが *nnnn* バイトを超えています。

要因：フォルダ名の長さ（完全パス名から¥ファイル名を除いたもの）が、*nnnn* バイトを超えています。

対処：フォルダ名の長さを *nnnn* バイト以内に指定し直してください。

INS0111E

本プログラムは *nnnn* では動作しません。*mmmm* で実行してください。

要因：現在起動している OS (*nnnn*) では動作しません。このプログラムは OS (*mmmm*) で動作します。

nnnn : 動作できない OS

mmmm : 動作できる OS

対処：このプログラムでは *mmmm* のどれかの OS で起動して、実行してください。

INS0112E

このプログラムは VGA か、それ以上の解像度で実行してください。

要因：解像度が VGA 以上でないため、インストールが実行できません。

対処：VGA か、それ以上の解像度に変更して、再実行してください。

INS0113I

いくつかのファイルを他のプログラムが使用しているため、インストールすることができませんでした。システムを再起動するとファイルを更新します。

要因：一部のファイルをほかのプログラムが使っているため、インストールできませんでした。

対処：インストールできなかったファイルを更新するためには、システムを再起動してください。

INS0114W

ActiveX コントロールの登録ができません。

要因：ActiveX コントロールの登録に失敗しました。

対処：インストールを再実行してください。

INS0115W

インストールするコンポーネントが指定されていません。

要因：インストールするコンポーネントを何も指定していません。

対処：インストールするコンポーネントを一つ以上指定してから、再実行してください。

10.2.2 DBPARTNER ODBC Driver のメッセージ

DBPARTNER ODBC Driver (ODBC ドライバ) のメッセージを示します。

DBL0001E

RDA Link 定義ユーティリティが見つかりませんでした。RDA Link 定義ユーティリティがインストールされているか確認してください。

要因：ネットワーク定義をするためには RDA Link 定義ユーティリティが必要です。RDA Link 定義ユーティリティがインストールされていないか、又はインストールが失敗しているおそれがあります。DBPARTNER ODBC Driver は、処理を終了します。

対処：RDA Link 定義ユーティリティをインストールしていないときは、インストールを実行してください。RDA Link 定義ユーティリティのインストールが失敗していると考えられるときは、インストールを再実行してください。

DBL0002E

RDA Link 定義ユーティリティの起動に必要なメモリが不足しています。他のアプリケーションプログラムを終了し、再度ネットワーク定義をクリックして RDA Link 定義ユーティリティを起動してください。

要因：ネットワーク定義をするための RDA Link 定義ユーティリティの起動に必要なメモリが不足しています。DBPARTNER ODBC Driver は、処理を終了します。

対処：ほかのアプリケーションを終了してから [ネットワーク定義] をやり直した後、RDA Link 定義ユーティリティを起動してください。

DBL0003E

RDA Link のバージョンが違います。

要因：DBPARTNER2 Client を稼働できるバージョンの RDA Link ではありません。DBPARTNER ODBC Driver は、処理を終了します。

対処：DBPARTNER2 Client を稼働できるバージョンの RDA Link をインストールして、再実行してください。

DBL0010E

応答監視時間の指定に誤りがあります。

要因：応答監視時間に 0～3600 の整数以外の値を指定しています。DBPARTNER ODBC Driver は、次の処理の入力待ちとなります。

対処：0～3600 の整数を指定してから、再実行してください。

DBL0020E

送受信領域サイズに数字以外の文字が指定されました。

要因：送受信領域サイズに数字以外の文字を指定しています。ODBC ドライバは、次の処理の入力待ちになります。

対処：送受信領域サイズを数字で指定してから、再実行してください。

DBL0030E

受信データ件数に数字以外の文字が指定されました。

要因：受信データ件数の指定に、数字以外の文字を指定しています。DBPARTNER ODBC Driver は、次の処理の入力待ちになります。

対処：受信データ件数を数字で指定してから、再実行してください。

DBL0100E

通信ドライバの指定が不正なため、処理を中止しました。

要因：DABroker ドライバをインストールしてから 一度も通信ドライバを設定していないか、又は環境定義が破壊されているおそれがあります。DBPARTNER ODBC Driver は、処理を終了します。

対処：[DBPARTNER ODBC Driver for DABroker－環境定義－] ダイアログボックスで、通信ドライバを設定してください。

DBL0101E

指定された通信ドライバに対応した DBPARTNER ODBC Driver ドライバが見つかりませんでした。

要因：環境定義に指定した通信ドライバに対応する DABroker ドライバがインストールされていないか、又は所定のフォルダから移動されているおそれがあります。DBPARTNER ODBC Driver は、処理を終了します。

対処：通信ドライバに対応した DABroker ドライバをインストールした後、再実行してください。

DBL0401E

システムコールでエラーが発生しました。[XXXX (), errno = YYYY]

要因：OS のシステムコール関数がエラーになりました。ODBC ドライバは、処理を終了します。

XXXX：システムコール名

YYYY：エラー番号

対処：操作を再実行してください。その後も同じエラーが繰り返し起こるときは、保守担当者に連絡してください。

DBL0900E

受信バッファが破壊されています。保守担当者に連絡してください。

要因：受信バッファのデータが、途中で破壊されました。DBPARTNER ODBC Driver は、処理を終了します。

対処：[DBPARTNER ODBC Driver for DABroker-セットアップ] ダイアログボックスの設定の、受信データ件数の設定値を少なくしてから、再実行してください。設定値を少なくしても同じエラーになるときは、保守担当者に連絡してください。

DBL1001E

通信ドライバのローディングに失敗しました。

要因：下位の通信ドライバのローディングに失敗しました。DBPARTNER ODBC Driver は、処理を終了します。

対処：次の処置をしてください。

- 通信ドライバをインストールしているかどうかを確認してください。
- DABroker ドライバをセットアップしたときの、通信ドライバの選択が正しいかどうかを確認してください。

DBL1002E

コネクション確立時、ソケットのオープンに失敗しました。[errno = XXXX]

要因：コネクションを確立しようとしたますが、空いているソケットがありません。DBPARTNER ODBC Driver は、処理を終了します。

XXXX：エラーコード

対処：ほかのアプリケーションが終了するのを待ってから、再実行してください。再実行してもエラーが起きるときは、通信ドライバに設定するソケット数を増やしてください。ソケット数の設定については、通信ドライバのマニュアル又はヘルプを参照してください。

DBL1003E

コネクション確立時に TCP コネクション確立処理に失敗しました。[errno = XXXX]

要因：コネクションを確立時に TCP コネクション確立処理に失敗しました。DBPARTNER ODBC Driver は、処理を終了します。

XXXX：エラーコード

対処：次に示す処置をしてください。

- 一定時間後、再実行してください。
- サーバプログラム (DABroker, Database Connection Server) 又はサーバプログラム下の通信プログラムがエラーになっていないかどうかを確認してください。
- 通信ドライバに登録したインターネットアドレスを見直してください。
- ポート番号を見直してください。
- 回線の接続状態を調査してください。
- TCP/IP を使っているほかのアプリケーションを終了させるか、又は通信ドライバに設定する TCP コネクションの数を増やしてください。

DBL1004E

コネクションが切断されました。[errno = XXXX]

要因：データの送受信時にコネクションが切断されました。DBPARTNER ODBC Driver は、処理を終了します。

XXXX：エラーコード

対処：次に示す処置をしてください。

- 一定時間後、再実行してください。
- サーバプログラム (DABroker, Database Connection Server) 又はサーバプログラム下の通信プログラムがエラーになっていないかどうかを確認してください。
- 回線の接続状態を調査してください。
- サーバプログラムの動作環境設定ファイルの内容と DABroker ドライバをセットアップしたときの設定内容を見直してください。

DBL1005E

応答監視時間を超えました。

要因：データの送受信時に応答監視時間を超えました。DBPARTNER ODBC Driver は、処理を終了します。

対処：サーバプログラム (DABroker, Database Connection Server) がエラーになっていないかどうかを確認してください。エラーになっていないときは、DABroker ドライバの [DBPARTNER ODBC Driver for DABroker-環境定義-] ダイアログボックスで設定した応答監視時間の値を大きくしてください。

DBL1006E

インターネットアドレスが取得できません。

要因：接続先のインターネットアドレスが取得できません。DBPARTNER ODBC Driver は、処理を終了します。

対処：通信ドライバの接続先のインターネットアドレスが登録されているかどうかを確認してください。

DBL1007E

通信時に異常が発生しました。[errno = XXXX]

要因：サーバとの通信時に異常が起きました。DBPARTNER ODBC Driver は、処理を終了します。

XXXX：エラーコード

対処：環境が壊れているおそれがあります。ODBC ドライバを再インストールしてください。再インストールしても同じエラーとなる場合は、保守員に連絡してください。

DBL1010E

HITSENER5 以外から DIAPRISM はアクセスできません。

要因：HITSENER5 以外から DIAPRISM はアクセスできません。

処理：処理を終了します。

対処：DIAPRISM にアクセスする場合は HITSENER5 を使用してください。

DBL1011E

接続したデータベースとデータソースに指定したデータベースの種類が異なります。

要因：接続したデータベースとデータソースに指定したデータベースの種類が異なります。

処理：処理を終了します。

対処：データソースに指定したデータベース識別子が正しいかどうか見直してください。

DBL2001E

ユーザ外字ファイルの OPEN に失敗しました。

要因：ファイルが見つからない、オープンされているファイルが多過ぎるなどの理由で、指定したユーザ外字ファイルをオープンできません。DBPARTNER ODBC Driver は、処理を終了します。

対処：ファイルを使える状態かどうかを調べて、使えない状態のときは原因を取り除いてから、再実行してください。

DBL2002E

ユーザ外字ファイルの READ に失敗しました。

要因：ユーザ外字ファイルの読み込み中にエラーが起きました。DBPARTNER ODBC Driver は、処理を終了します。

対処：ファイルを使える状態かどうかを調べて、使えない状態のときは原因を取り除いてから、再実行してください。

DBL2003E

ユーザ外字ファイルの内容が不正です。

要因：ユーザ外字ファイルに指定した内容が間違っています。DBPARTNER ODBC Driver は、処理を終了します。

対処：ユーザ外字ファイルの内容を見直して、間違いがあるときは記述内容を修正してから、再実行してください。

DBL9001E

新しい DBPARTNER ODBC Driver がインストールされています。インストールを終了します。

要因：インストール先に、現在インストール中の DBPARTNER ODBC Driver よりも新しいバージョンのものがインストールされています。DBPARTNER ODBC Driver は、処理を終了します。

対処：現在インストールされているものより古いバージョンの DBPARTNER ODBC Driver はインストールできません。同じ、又は新しいバージョンの DBPARTNER ODBC Driver をインストールしてください。

DBL9002W

ディレクトリの長さが XXXX バイトを超えています。

要因：フォルダ名の長さ（絶対パス名から ¥ファイル名を除いたもの）が、XXXX バイトを超えています。DBPARTNER ODBC Driver は、次の処理要求を待ちます。

対処：フォルダ名の長さを XXXX バイト以内で指定し直してください。

DBL9003E

システム又は、インストールプログラムに異常があります。インストールを終了します。(XXXX)
XXXX：保守コード

要因：システム又はインストールプログラムで異常が発生しました。DBPARTNER ODBC Driver は、処理を終了します。

対処：再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

DBL9004E

DBPARTNER ODBC Driver が稼働できるシステムのバージョンではありません。

要因：DBPARTNER ODBC Driver が稼働できるシステムのバージョンではありません。DBPARTNER ODBC Driver は、処理を終了します。

対処：DBPARTNER ODBC Driver が稼働できるシステムのバージョンをインストールして、DBPARTNER ODBC Driver のインストールを再実行してください。

DBL9005E

XXXX には、DBPARTNER ODBC Driver をインストールするだけの容量がありません。インストールには、YYYY バイトの容量を確保してください。
XXXX：インストール先ディレクトリ

要因：XXXX で示すドライブに、DBPARTNER ODBC Driver をインストールするための容量が不足しています。YYYY バイト以上の容量を確保してください。DBPARTNER ODBC Driver は、次の処理要求を待ちます。

対処：XXXX で示すドライブの不要なファイルの削除などで、DBPARTNER ODBC Driver をインストールするために必要な容量を確保して、インストールを再実行してください。

DBL9006E

DBPARTNER ODBC Driver の DLL ファイル (XXXX) は、現在使用中のためインストールすることができません。
XXXX：ファイル名

要因：XXXX で示すファイルが使用中のため、インストールできません。DBPARTNER ODBC Driver は、処理を終了します。

対処：XXXX で示すファイルを使用しているプログラムを終了して、インストールを再実行してください。

DBL9007E

DBPARTNER ODBC Driver のインストールファイル転送中にエラーが発生しました (XXXX,YYYY)。

XXXX：エラーコード

YYYY：ファイル名

要因：DBPARTNER ODBC Driver のインストールファイル転送中にエラーが発生しました。DBPARTNER ODBC Driver は、処理を終了します。

対処：再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

DBL9008E

DBPARTNER ODBC Driver のインストール起動パラメタ解析中にエラーが発生しました。

要因：DBPARTNER ODBC Driver のインストール起動パラメタ解析中にエラーが発生しました。DBPARTNER ODBC Driver は、処理を終了します。

対処：再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

DBL9009E

OS の画面情報の取得に失敗しました。

要因：OS の画面情報の取得に失敗しました。DBPARTNER ODBC Driver は、処理を終了します。

対処：再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

DBL9010E

ドライバに関する情報のレジストリ登録に失敗しました。

要因：ドライバに関する情報のレジストリ登録に失敗しました。DBPARTNER ODBC Driver は、処理を終了します。

対処：再実行してください。なお、このエラーが繰り返し発生する場合は、システム管理者に連絡してください。

DBL9011W

コンポーネントが一つも選択されていません。一つ以上のコンポーネントを選択してください。

要因：コンポーネントが一つも選択されていません。DBPARTNER ODBC Driver は、次の処理要求を待ちます。

対処：一つ以上のコンポーネントを選択してください。

付録

付録 A AP 作成ウィザード

AP 作成ウィザードは、DBPARTNER2 Client が提供する ActiveX コントロールを使用したアプリケーションを作成するためのプログラムです。AP 作成ウィザードを使用すると、ActiveX コントロールのオブジェクトを意識することなく、データベースからデータを抽出するアプリケーションが作成できます。

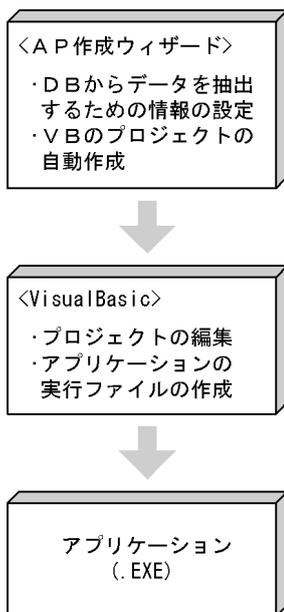
ただし、AP 作成ウィザードでは、アプリケーションの実行ファイルは作成できません。AP 作成ウィザードは、アプリケーションの実行ファイルの作成に必要となる Visual Basic のプロジェクトを作成するだけです。アプリケーションの実行ファイルを作成するには、AP 作成ウィザード以外に Visual Basic が必要です。

AP 作成ウィザードを使用して作成できるアプリケーションは、Visual Basic 5.0 以降に対応します。

付録 A.1 アプリケーション作成の流れ

アプリケーションを作成するには、AP 作成ウィザードと Visual Basic を使用します。次に、AP 作成ウィザードと Visual Basic を使用してアプリケーションを作成する処理の流れについて説明します。

図 A-1 アプリケーションの作成の流れ



(1) AP 作成ウィザードでの処理

AP 作成ウィザードでは、データベースからデータを抽出するための情報（表名、列名、条件など）を設定し、アプリケーションの実行ファイルの作成に必要となる Visual Basic のプロジェクト（*.frm, *.bas, *.vbp などのファイル）を自動作成します。

AP 作成ウィザードでは、表示されるダイアログボックスに従って操作するだけで、データを抽出するための情報設定から Visual Basic のプロジェクトの自動作成までの処理ができます。AP 作成ウィザードのダイアログボックスの名称と、各ダイアログボックスで設定する情報の概要を次に示します。

- [ActiveX 種別設定]：ActiveX コントロールを選択します。
- [接続先設定]：接続先を設定します。
- [抽出表設定]：データを抽出する表名を設定します。
- [抽出条件設定]：データを抽出する条件を設定します。
- [抽出列設定]：データを抽出する列名を設定します。
- [ソート設定]：抽出列にソートを設定します。
- [集合関数設定]：抽出列に集合関数を設定します。
- [プロジェクト設定]：プロジェクト及び AP の作成を確認します。

(2) VisualBasic での処理

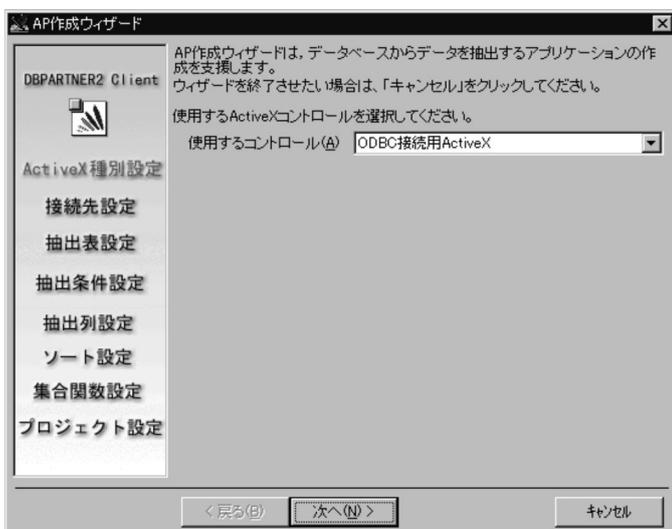
Visual Basic では、AP 作成ウィザードで作成したプロジェクト (*.frm, *.bas, *.vbp などのファイル) を編集してアプリケーションの実行ファイル (*.exe) を作成します。

付録 A.2 AP 作成ウィザードの起動と終了

AP 作成ウィザードを起動、終了する操作について、次に説明します

(1) AP 作成ウィザードの起動

AP 作成ウィザードは、[スタート] メニューから [プログラム] - [DBPARTNER2] - [AP 作成ウィザード] を選択することで起動できます。この操作によって AP 作成ウィザードが起動し、次に示すダイアログボックスが表示されます。



(2) AP 作成ウィザードの終了

AP 作成ウィザードを終了するには、最後に表示される [プロジェクト設定] ダイアログボックスの [終了] ボタンをクリックします。この操作によってダイアログボックスが閉じ、AP 作成ウィザードが終了します。

また、次に示す操作でも AP 作成ウィザードを終了させることができます。

- ダイアログボックスの [キャンセル] ボタンをクリックする。
- ダイアログボックスのタイトル部分の左端にあるアイコンをクリックし、表示されたシステムメニューで [閉じる] メニューをクリックする。
- ダイアログボックスのタイトル部分の右端にある [×] をクリックする。

付録 A.3 AP 作成ウィザードの操作

AP 作成ウィザードは、表示されるダイアログボックスに従って操作します。ダイアログボックスに従って操作すると、プログラミングの知識がなくてもアプリケーションを作成できます。

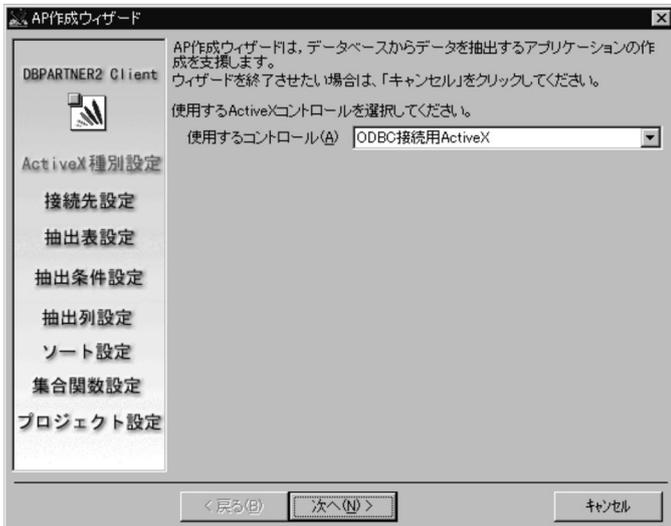
AP 作成ウィザードのダイアログボックスの表示順序、名称、及び各ダイアログボックスで設定する情報の概要を表 A-1 に示します。

表 A-1 AP 作成ウィザードのダイアログボックス

表示順	ダイアログボックス名	設定内容
1	[ActiveX 種別設定]	ActiveX コントロールを選択します
2	[接続先設定]	接続先を設定します
3	[抽出表設定]	データを抽出する表名を設定します
4	[抽出条件設定]	データを抽出する条件を設定します
5	[抽出列設定]	データを抽出する列名を設定します
6	[ソート設定]	抽出列にソートを設定します
7	[集合関数設定]	抽出列に集合関数を設定します
8	[プロジェクト設定]	プロジェクト及び AP の作成を確認します

(1) ActiveX コントロールを選択する：[ActiveX 種別設定]

アプリケーションで使用する ActiveX コントロールの種別を選択します。



●指定項目の説明

- ・ [使用するコントロール(A)]
ActiveX コントロールの種別を選択します。
 - ・ ODBC 接続用 ActiveX
 - ・ CommuniNet 接続用 ActiveX

●ボタンの説明

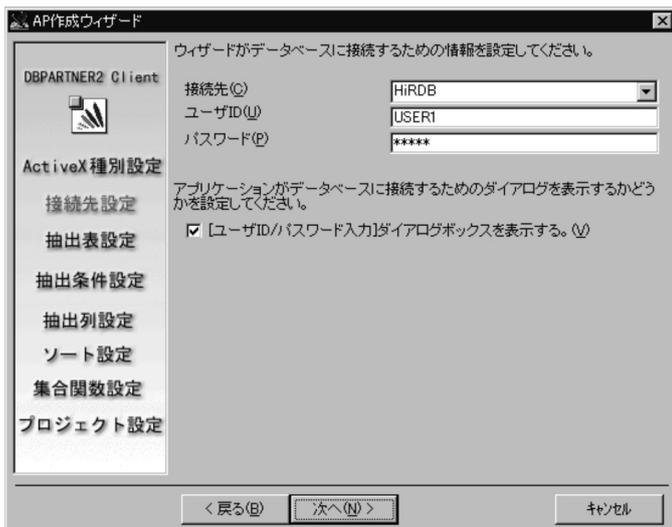
- ・ [戻る(B)]
このダイアログボックスでは無効（不活性）です。
- ・ [次へ(N)]
次のダイアログボックスに進みます。
- ・ [キャンセル]
AP 作成ウィザードを終了します。

(2) 接続先を設定する：[接続先設定]

AP 作成ウィザードがデータベースに接続するための情報を設定します。

データベースに接続するための情報は、使用する ActiveX コントロールによって異なります。

(a) ODBC 接続用 ActiveX コントロールの場合



●指定項目の説明

・ [接続先(C)]

使用するデータソース名を指定又は選択します。
選択肢には端末に登録されているデータソースを表示します。

・ [ユーザ ID(U)]

サーバに接続するユーザ ID (32 文字以内) を指定します。

・ [パスワード(P)]

[ユーザ ID] テキストボックスに指定した ID のパスワード (32 文字以内) を指定します。

・ [[ユーザ ID/パスワード入力] ダイアログボックスを表示する(V)]

アプリケーションの起動時に [ユーザ ID/パスワード入力] ダイアログボックスを表示する場合、チェックします。
チェックがない場合、[ユーザ ID(U)], [パスワード(P)] テキストボックスに設定した内容でデータベースに接続します。

●ボタンの説明

・ [戻る(B)]

前のダイアログボックスに戻ります。この画面で設定した内容は破棄されます。

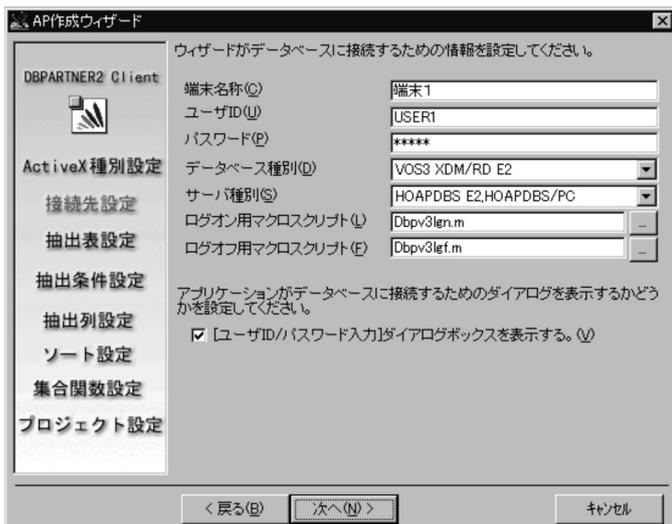
・ [次へ(N)]

次のダイアログボックスに進みます。

・ [キャンセル]

AP 作成ウィザードを終了します。

(b) CommuniNet 接続用 ActiveX コントロールの場合



●指定項目の説明

- [端末名称(C)]
通信に使用する CommuniNet の端末名称（14 文字以内）を指定します。
- [ユーザ ID(U)]
サーバに接続するユーザ ID（8 文字以内）を指定します。
- [パスワード(P)]
[ユーザ ID] テキストボックスに指定した ID のパスワード（8 文字以内）を指定します。
- [データベース種別(D)]
使用するデータベースの種別を選択します。
 - VOS3 XDM/RD E2
 - VOS3 XDM/SD E2
 - VOS3 PDMII E2
 - VOS3 ACE3 ユーザファイル
 - VOS3 RDB1
 - VOSK RDB 編成ファイル
 - VOS1 PDMII E2
 - VOS1 ACE3 ユーザファイル

VOS3 XDM/SD E2 を選択した場合、参照パスワードが設定されている表には [抽出表設定] ダイアログボックスでの設定だけができます。[抽出条件設定]、[抽出列設定]、[ソート設定]、[集合関数設定] ダイアログボックスでの設定はできません。

- [サーバ種別(S)]
使用するサーバプログラムを選択します。

- HOAPDBS E2, HOAPDBS/PC
- DBPARTNER/Link, ACE3 E3
- [ログオン用マクロスクリプト(L)]

サーバプログラムと接続するための、ログオン操作を登録したマクロスクリプトのファイル名を完全パス（256文字以内）で指定します。フォルダを含まないファイル名を指定した場合は、CommuniNetをインストールしたフォルダの¥Macro フォルダを検索します。

右端の [...] ボタンをクリックすることで、ファイル選択の画面から指定できます。
- [ログオフ用マクロスクリプト(F)]

サーバプログラムとの接続を終了するための、ログオフ操作を登録したマクロスクリプトのファイル名を完全パス（256文字以内）で指定します。フォルダを含まないファイル名を指定した場合は、CommuniNetをインストールしたフォルダの¥Macro フォルダを検索します。

右端の [...] ボタンをクリックすることで、ファイル選択の画面から指定できます。
- [[ユーザID/パスワード入力] ダイアログボックスを表示する(V)]

アプリケーションの起動時に [ユーザID/パスワード入力] ダイアログボックスを表示する場合、チェックします。

チェックがない場合、[ユーザID(U)], [パスワード(P)] テキストボックスに設定した内容でデータベースに接続します。

●ボタンの説明

- [戻る(B)]

前のダイアログボックスに戻ります。この画面で設定した内容は破棄されます。
- [次へ(N)]

次のダイアログボックスに進みます。
- [キャンセル]

AP作成ウィザードを終了します。

(3) データを抽出する表名を設定する：[抽出表設定]

データを抽出する表を選択します。



表を選択するには、[表(T)] リストボックスで表の名称を選択した後、[>] ボタンをクリックします。選択した表の名称が[選択表(S)] リストボックスに表示されます。

表の選択を解除するには、[選択表(S)] リストボックスで選択を解除する表の名称をクリックした後、[<] ボタンをクリックします。選択を解除した表の名称が[選択表(S)] リストボックスから消えます。

●表示/指定項目の説明

・ [表名]

[表(T)] リストボックスで選択（反転表示）している表の名称が表示されます。

・ [表(T)]

データベースに定義されている表の名称が表示されます。

・ [選択表(S)]

選択した表の名称が表示されます。

●ボタンの説明

・ [>]

[表(T)] リストボックスで選択（反転表示）している表を選択します。選択した表の名称が[選択表(S)] リストボックスに表示されます。

・ [<]

[選択表(S)] リストボックスで選択（反転表示）している表の選択を解除します。選択を解除した表の名称が[選択表(S)] リストボックスから消えます。

・ [戻る(B)]

前のダイアログボックスに戻ります。この画面で設定した内容は破棄されます。

・ [次へ(N)]

次のダイアログボックスに進みます。

- ・ [完了(X)]

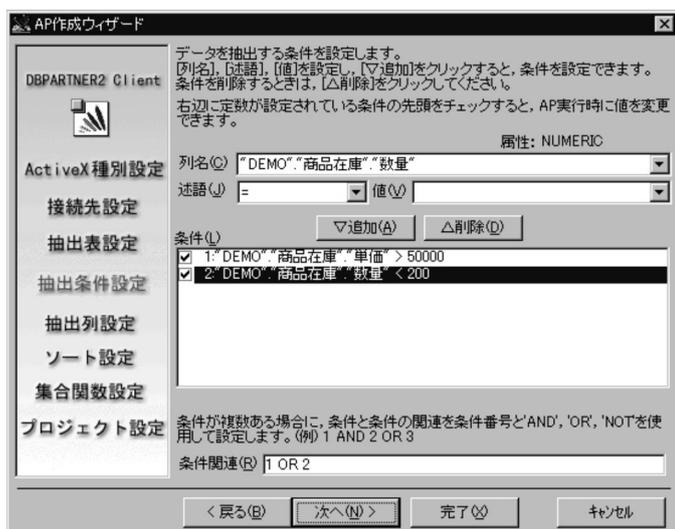
最後のダイアログボックスに進みます。

- ・ [キャンセル]

AP 作成ウィザードを終了します。

(4) データを抽出する条件を設定する：[抽出条件設定]

データを抽出する条件を設定します。



抽出条件を設定するには、[列名(C)]、[述語(J)]、[値(V)] の各コンボボックスで条件を設定した後、[▽追加(A)] ボタンをクリックします。設定した抽出条件が、[条件(L)] リストボックスに表示されます。

設定した抽出条件を削除するには、[条件(L)] リストボックスで削除する条件を選択した後、[△削除(D)] ボタンをクリックします。削除した抽出条件が、[条件(L)] リストボックスから消えます。

●表示/指定項目の説明

- ・ [列名(C)]

抽出条件の左辺とする列名を選択します。

- ・ [述語(J)]

列名と値をつなぐ述語を選択します。

"=", "<>", "<=", "<", ">=", ">",

"LIKE", "NOT LIKE", "IS NULL", "IS NOT NULL"

- ・ [値(V)]

抽出条件の右辺とする値（256 文字以内）を指定します。

列比較条件の右辺とする列名を選択することもできます。

[述語(J)] コンボボックスで"IS NULL"又は"IS NOT NULL"を選択した場合、この項目を指定する必要はありません。

- [属性]

[列名(C)] リストボックスで選択している列の属性が表示されます。

- [条件(L)]

設定した抽出条件が表示されます。

右辺に定数を設定した抽出条件は、先頭をチェックすることでアプリケーションの実行時に定数の値が変更できます。

- [条件関連(R)]

設定した抽出条件の関連を 256 文字以内で設定します。

抽出条件の関連は、抽出条件の番号と"AND", "OR", "NOT"を使用して設定します。

(例)

1 AND 2 OR 3

●ボタンの説明

- [▽追加(A)]

[列名(C)], [述語(I)], [値(V)] の各コンボボックスで指定している抽出条件を設定します。設定した抽出条件は、[条件(L)] リストボックスに表示されます。

- [△削除(D)]

[条件(L)] リストボックスで選択（反転表示）している抽出条件を削除します。削除した抽出条件は、[条件(L)] リストボックスから消えます。

- [戻る(B)]

前のダイアログボックスに戻ります。この画面で設定した内容は破棄されます。

- [次へ(N)]

列を選択した場合は、次のダイアログボックスに進みます。

列を選択していない場合は、最後のダイアログボックスに進みます。

- [完了(X)]

最後のダイアログボックスに進みます。

- [キャンセル]

AP 作成ウィザードを終了します。

(5) データを抽出する列名を設定する：[抽出列設定]

データを抽出する列を選択します。



抽出する列を選択するには、[列(C)] リストボックスで列を選択した後、[>] ボタンをクリックします。選択した列が、[抽出列(S)] リストボックスに表示されます。[>>] ボタンですべての列を一括して選択できます。

列の選択を解除するには、[抽出列(S)] リストボックスで列を選択した後、[<] ボタンをクリックします。選択を解除した列が、[抽出列(S)] リストボックスから消えます。[<<] ボタンですべての列の選択を一括して解除できます。

●表示/指定項目の説明

・ [列名]

[列(C)] リストボックスで選択（反転表示）している列の名称が表示されます。

・ [列(C)]

[抽出表設定] ダイアログボックスで選択した表に定義されている列が表示されます。

・ [抽出列(S)]

選択した列の名称が表示されます。

初期表示は、[列(C)] リストボックスと同一の内容です。

●ボタンの説明

・ [>]

[列(C)] リストボックスで選択（反転表示）している列を、抽出する列として選択します。選択した列が、[抽出列(S)] リストボックスに表示されます。

・ [>>]

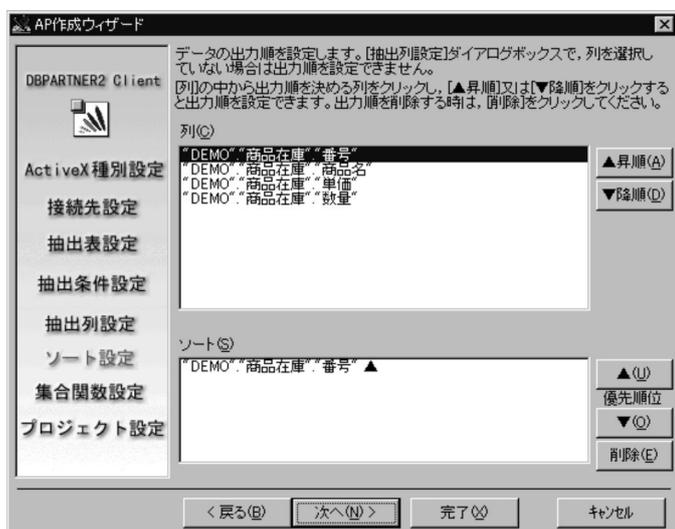
[列(C)] リストボックスに表示されているすべての列を、抽出する列として選択します。選択した列が、[抽出列(S)] リストボックスに表示されます。

- [**<**]
- [抽出列(S)] リストボックスで選択（反転表示）している列を、抽出する列から削除します。削除した列は、[抽出列(S)] リストボックスから消えます。
- [**<<**]
- [抽出列(S)] リストボックスに表示されているすべての列を、抽出する列から削除します。削除した列は、[抽出列(S)] リストボックスから消えます。
- [戻る(B)]
前のダイアログボックスに戻ります。この画面で設定した内容は破棄されます。
- [次へ(N)]
次のダイアログボックスに進みます。
- [完了(X)]
最後のダイアログボックスに進みます。
- [キャンセル]
AP 作成ウィザードを終了します。

(6) 抽出列にソートを設定する：[ソート設定]

抽出したデータ（列）にソートを設定します。

ただし、[抽出列設定] ダイアログボックスで列を選択していない場合はソートを設定できません。



列にソートを設定するには、[列(C)] リストボックスで列を選択した後、[▲昇順(A)] ボタン又は [▼降順(D)] ボタンをクリックします。ソートを設定した列が、[ソート(S)] リストボックスに表示されます。

ソートの設定を解除するには、[ソート(S)] リストボックスで列を選択した後、[削除(E)] ボタンをクリックします。ソートの設定を解除した列が、[ソート(S)] リストボックスから消えます。

●表示項目の説明

- [列(C)]

[抽出列設定] ダイアログボックスで選択した列が表示されます。

- [ソート(S)]

ソートを設定した列の名称が表示されます。

各列の右側に、設定されているソート順 (▲：昇順, ▼：降順) が表示されます。

複数の列にソートを設定している場合、上に表示されている列のソートが優先されます。

●ボタンの説明

- [▲昇順(A)]

[列(C)] リストボックスで選択 (反転表示) している列に、昇順のソートを設定します。ソートを設定した列が、[ソート(S)] リストボックスに表示されます。

- [▼降順(D)]

[列(C)] リストボックスで選択 (反転表示) している列に、降順のソートを設定します。ソートを設定した列が、[ソート(S)] リストボックスに表示されます。

- [▲(U)]

[ソート(S)] リストボックスで選択 (反転表示) している列のソートの優先順位を一つ上げます。

- [▼(O)]

[ソート(S)] リストボックスで選択 (反転表示) している列のソートの優先順位を一つ下げます。

- [削除(E)]

[ソート(S)] リストボックスで選択 (反転表示) している列の、ソートの設定を解除します。解除した列が、[ソート(S)] リストボックスから消えます。

- [戻る(B)]

前のダイアログボックスに戻ります。この画面で設定した内容は破棄されます。

- [次へ(N)]

次のダイアログボックスに進みます。

- [完了(X)]

最後のダイアログボックスに進みます。

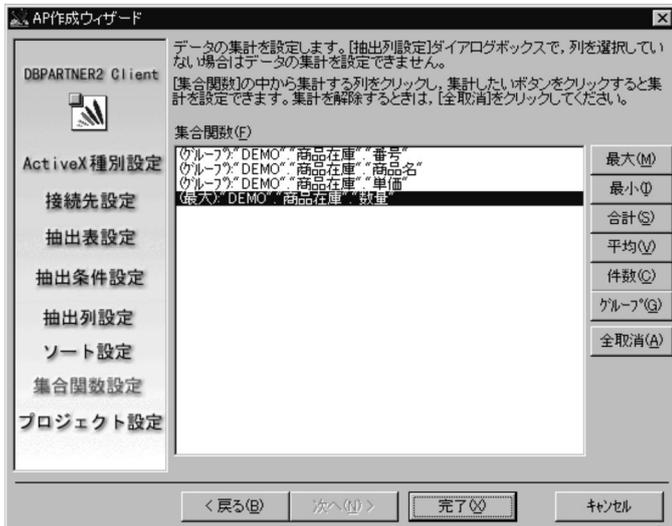
- [キャンセル]

AP 作成ウィザードを終了します。

(7) 抽出列に集合関数を設定する：[集合関数設定]

抽出したデータ (列) に集合関数を設定します。

ただし、[抽出列設定] ダイアログボックスで列を選択していない場合は集合関数を設定できません。



列に集合関数を設定するには、[集合関数(F)] リストボックスで列を選択した後、[最大(M)]、[最小(I)]、[合計(S)]、[平均(V)]、[件数(C)]、又は[グループ(G)] のボタンをクリックします。列の左側に、設定した集合関数が表示されます。

[全取消(A)] ボタンで、すべての集合関数の設定が解除できます。

●表示項目の説明

・ [集合関数(F)]

[抽出列設定] ダイアログボックスで抽出列とした列を表示します。

●ボタンの説明

・ [最大(M)]

[集合関数(F)] リストボックスで選択している列に、集合関数"最大"を設定します。列の左側に"(最大)"と表示されます。

・ [最小(I)]

[集合関数(F)] リストボックスで選択している列に、集合関数"最小"を設定します。列の左側に"(最小)"と表示されます。

・ [合計(S)]

[集合関数(F)] リストボックスで選択している列に、集合関数"合計"を設定します。列の左側に"(合計)"と表示されます。

・ [平均(V)]

[集合関数(F)] リストボックスで選択している列に、集合関数"平均"を設定します。列の左側に"(平均)"と表示されます。

・ [件数(C)]

[集合関数(F)] リストボックスで選択している列に、集合関数"件数"を設定します。列の左側に"(件数)"と表示されます。

- [グループ(G)]

[集合関数(E)] リストボックスで選択している列に、グループ分けを設定します。列の左側に" (グループ) "と表示されます。

- [全取消(A)]

すべての集合関数の設定を解除します。

- [戻る(B)]

前のダイアログボックスに戻ります。この画面で設定した内容は破棄されます。

- [次へ(N)]

このダイアログボックスでは無効 (不活性) です。

- [完了(X)]

最後のダイアログボックスに進みます。

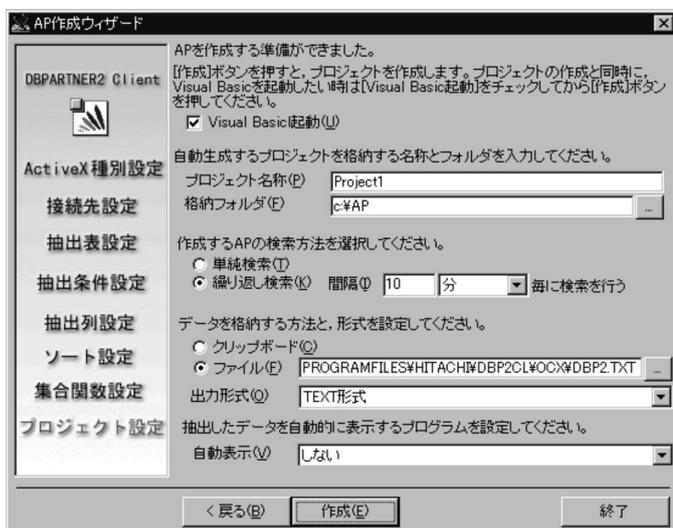
- [キャンセル]

AP 作成ウィザードを終了します。

(8) プロジェクト及び AP の作成を確認する： [プロジェクト設定]

次に示す情報を設定します。

- プロジェクト作成後の Visual Basic の起動
- 作成するプロジェクトの名称と格納フォルダ
- アプリケーションでデータを抽出する際の検索方法、及び抽出したデータの表示、保存方法



●表示/指定項目の説明

- [Visual Basic 起動(U)]

[作成(E)] ボタンをクリックした際、Visual Basic を起動させたい場合にチェックします。

- [プロジェクト名称(P)]

作成するアプリケーションのプロジェクトの名称（40文字以内）を指定します。プロジェクト名称に、/ ¥ < > * ? " | : は使用できません。

- [格納フォルダ(E)]

アプリケーションを格納するフォルダを完全パス（256文字以内）で指定します。
右端の [...] ボタンをクリックすることで、フォルダ選択の画面から指定できます。

- [単純検索(I)]

検索方法を単純検索とします。

単純検索を指定した場合、アプリケーションが表示するダイアログボックスには [抽出] ボタンが表示されます。

- [繰り返し検索(K)]

検索方法を繰り返し検索とします。

繰り返し検索を指定した場合、アプリケーションが表示するダイアログボックスには [開始], [停止] ボタンが表示されます。

- [間隔(I)]

抽出を繰り返す間隔（1～99）を指定します。また、間隔に指定した値の単位（分、時間）を選択します。

- [クリップボード(C)]

抽出結果をクリップボードに格納します。

- [ファイル(F)]

抽出結果をファイルに格納します。

また、抽出結果を格納するファイル名を完全パス（256文字以内）で指定します。
右端の [...] ボタンをクリックすることで、ファイル選択の画面から指定できます。

- [出力形式(O)]

抽出結果を出力する際のデータ形式を選択（TEXT形式、Excel 4.0形式、CSV形式、DAT形式）します。

- [自動表示(V)]

抽出結果を表示させるかどうか（しない、メモ帳、Excel）を選択します。

- ボタンの説明

- [戻る(B)]

前のダイアログボックスに戻ります。この画面で設定した内容は破棄されます。

- [作成(E)]

Visual Basic のプロジェクトを作成します。

プロジェクトの各ファイルの名称は、[プロジェクト名称(P)] テキストボックスに指定した名称に、次に示すサフィックスが付けられます。

- プロジェクトファイル：[プロジェクト名称] + ".Vbp"
- フォームモジュール 1：[プロジェクト名称] + "1.Frm"
- フォームモジュール 2：[プロジェクト名称] + "2.Frm"
- 標準モジュール：[プロジェクト名称] + ".Bas"

• [終了]

AP 作成ウィザードを終了します。

プロジェクトを作成していない場合、終了の確認メッセージが表示されます。

付録 A.4 作成できるアプリケーションの仕様

AP 作成ウィザードで作成できるアプリケーションの仕様は接続形態によって異なります。接続形態別のアプリケーションの仕様を表 A-2 と表 A-3 に示します。

表 A-2 ODBC 接続

指定項目	仕 様	設定ダイアログ
接続先	一覧から選択したデータソース (固定)	[接続先設定]
ユーザ ID	[[ユーザ ID/パスワード入力]ダイアログボックスを表示する]を指定した場合は、入力できる	
パスワード	[[ユーザ ID/パスワード入力]ダイアログボックスを表示する]を指定した場合は、入力できる	
抽出表	一覧から選択した表名 (固定)	[抽出表設定]
抽出列	一覧から選択した列名 (固定)	[抽出列設定]
抽出条件(左辺)	一覧から選択した列名 (固定)	[抽出条件設定]
抽出条件(演算子)	比較演算子, LIKE, NOT LIKE から選択した演算子 (固定)	
抽出条件(右辺)	設定した値 (固定) 条件の先頭をチェックした場合、設定した値をデフォルトで表示し設定できる	
条件関連式	設定した条件関連式 (固定)	
ソート	一覧で指定したソート (固定)	[ソート設定]
集合関数	一覧で指定した集合関数 (固定)	[集合関数設定]
抽出データ出力先	クリップボード, ファイルから選択した出力先 (固定)	[プロジェクト設定]
抽出データ出力形式	クリップボードの場合, TEXT 形式, Excel 4.0 形式から選択した形式 (固定) ファイルの場合, TEXT 形式, Excel 4.0 形式, CSV 形式, DAT 形式から選択した形式 (固定)	
抽出データ表示	しない, メモ帳, Excel から選択した方法 (固定)	

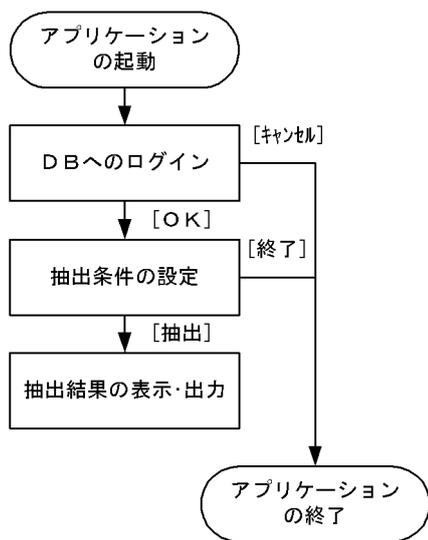
指定項目	仕 様	設定ダイアログ
タイマ監視	単純検索, 繰り返し検索から選択したモード (固定)	

表 A-3 CommuniNet 接続

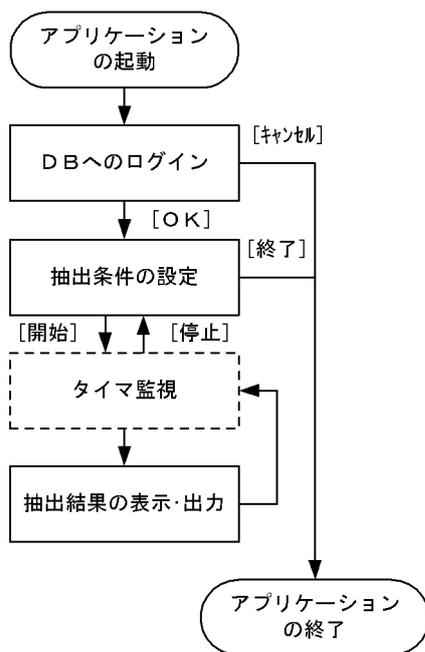
指定項目	仕 様	設定ダイアログ
端末名称	設定した端末名称 (固定)	[接続先設定]
ログオンファイル名	設定したファイル名 (固定)	
ログオフファイル名	設定したファイル名 (固定)	
データベース種別	選択した種別 (固定)	
サーバ種別	選択した種別 (固定)	
ユーザ ID	[[ユーザ ID/パスワード入力]ダイアログボックスを表示する]を指定した場合は, 入力できる	
パスワード	[[ユーザ ID/パスワード入力]ダイアログボックスを表示する]を指定した場合は, 入力できる	
抽出表	一覧から選択した表名 (固定)	[抽出表設定]
抽出列	一覧から選択した列名 (固定)	[抽出列設定]
抽出条件(左辺)	一覧から選択した列名 (固定)	[抽出条件設定]
抽出条件(演算子)	比較演算子, LIKE, NOT LIKE から選択した演算子 (固定)	
抽出条件(右辺)	設定した値 (固定) 条件の先頭をチェックした場合, 設定した値をデフォルトで表示し設定できる	
条件関連式	設定した条件関連式 (固定)	
ソート	一覧で指定したソート (固定)	[ソート設定]
集合関数	一覧で指定した集合関数 (固定)	[集合関数設定]
抽出データ出力先	クリップボード, ファイルから選択した出力先 (固定)	[プロジェクト設定]
抽出データ出力形式	クリップボードの場合, TEXT 形式, Excel 4.0 形式から選択した形式 (固定) ファイルの場合, TEXT 形式, Excel 4.0 形式, CSV 形式, DAT 形式から選択した形式 (固定)	
抽出データ表示	しない, メモ帳, Excel から選択した方法 (固定)	
タイマ監視	単純検索, 繰り返し検索から選択したモード (固定)	

また, 作成したアプリケーションの操作は, [プロジェクト設定] で設定する検索方法 (単純検索, 繰り返し検索) によって異なります。検索方法の違いによるアプリケーションの操作の違いを次に示します。

● 単純検索の場合



● 繰返し検索の場合



付録 B DBPARTNER2 Client の予約語

DBPARTNER2 Client の予約語の一覧を表 B-1 に示します。DBPARTNER2 Client のアプリケーションを作成するときには、ここに示す名称を表名や列名に使わないでください。

SQL を使ってアプリケーションを作成するときは、ここに示す予約語に加えて、SQL の予約語が使えないので注意してください。

アクセスするデータベースやサーバプログラムの予約語については、該当するマニュアルを参照してください。

表 B-1 DBPARTNER2 Client の予約語の一覧

先頭の英字	予約語になる文字列
A	AND, and, ASC, AVG
B	BETWEEN, BIGINT, BINARY, BIT
C	<i>Cn</i> (<i>n</i> は 1~2147483647 の数値), CHAR, CHARACTER, CNT, COUNT, CPT
D	DAT, DATE, DAY, DAYS, DBP <i>xxx</i> (<i>xxx</i> は任意の長さの文字列), DEC, DECIMAL, DESC, DIGITS, DOUBLE
F	FLOAT
G	GROUP, GRP
H	HEX
I	IN, INTEGER, INTERVAL, IS
L	LARGE, LENGTH, LIKE, LONG, LONG MVARCHAR, LONG NVARCHAR, LONG VARBINARY, LONG VARCHAR
M	MAX, MCHAR, MIN, MINUTE, MONTH, MVARCHAR
N	NCHAR, NOT, not, NULL, NUMERIC, NVARCHAR
O	OR, or
P	POL
R	REAL
S	SECOND, SMALLFLT, SMALLINT, SQL <i>xxx</i> (<i>xxx</i> は任意の長さの文字列), SQL_ <i>xxx</i> (<i>xxx</i> は任意の長さの文字列), SUBSTR, SUM
T	<i>Tn</i> (<i>n</i> は 1~2048 の数値), <i>TCm</i> (<i>m</i> は 1~2048 の数値), TIME, TIMESTAMP, TINYINT, TO, TXT
V	VALUE, VARBINARY, VARCHAR
W	WJ2
X	XLS

先頭の英字	予約語になる文字列
Y	YEAR

付録 C DBPARTNER2 Client が扱うデータの形式

DBPARTNER2 Client が扱うデータの形式を表 C-1 に示します。

表 C-1 DBPARTNER2 Client が扱うデータの形式

データの形式	項目			
	データとデータの区切りコード	改行コード (行の終わり)	終端コード (全データの終わり)	データの詳細
CSV 形式	コンマ (0x2c)	CR (0x0d0a)	—	<ul style="list-style-type: none">• 文字列データは、データを「"」（ダブルクォーテーション）で囲みます。• 文字列データに「"」があるときは、「"」2個に変換します。• 数値データは、データを「"」で囲みません。
DAT 形式	コンマ (0x2c)	CR (0x0d0a)	EOF (0x1a)	<ul style="list-style-type: none">• 文字列データは、データを「"」（ダブルクォーテーション）で囲みます。• 文字列データ中に「'」があるときは、「'」（アポストロフィ）に変換します。• 数値データは、データを「"」で囲みません。
TEXT 形式	タブコード ¥t (0x09)	CR (0x0d0a)	—	<ul style="list-style-type: none">• データが文字列か数値かの区別はありません。
XLS 形式	—	—	—	<ul style="list-style-type: none">• Biff4 形式です。

付録 D プロパティとメソッドの制限値

DBPARTNER2 Client の ActiveX コントロールのプロパティとメソッドの制限値を、ODBC 接続と CommuniNet 接続で分けて説明します。

付録 D.1 ODBC 接続のときのプロパティとメソッドの制限値

ODBC 接続のときの、DBPARTNER2 Client の ActiveX コントロールのプロパティとメソッドの制限値を、次の(1)～(20)に示します。表中のアンダラインは、省略時の仮定値を示します。ここでは、制限値があるプロパティ又はメソッドの文字定数について説明しています。制限値がない文字定数は掲載していません。

(1) DBPARTNER オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値/文字定数：値
CommitMode	トランザクション終了モード	<u>dbpAuto</u> : 0 dbpManual : 1
ConnectDialogMode	接続ダイアログ表示モード	<u>dbpHide</u> : 0 dbpCase : 1 dbpShow : 2
DataSource	データソース名	0～32 バイト
DictPath	パス	0～256 バイト
TableDefSource	情報の要求先	<u>dbpDB</u> : 0 dbpDICT : 1 dbpDBDICT : 2
UserID	ユーザ ID	0～32 バイト
Password	パスワード	0～32 バイト

(2) SQLDirect オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
SQLText	SQL 文	1～30,000 バイト

(3) Result オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
QueryCountLimit	抽出行数の制限値	<u>0</u> 1～999,999,999

プロパティ/メソッド	項目	制限値
DataCheck	文字のチェックモード	<u>FALSE</u> , TRUE
HeaderOutput	ヘッダ出力モード	<u>FALSE</u> , TRUE
FieldHeader	ヘッダ	0～256 バイト
FieldCodeType	取得モード	dbpTypeCHAR, dbpCHAR : 1 dbpTypeBINARY, dbpBINARY : -2
SaveToFile	ファイル名	1～256 バイト
	データタイプ	dbpTextFormat, dbpTextFile : 0 dbpBiff4Format, dbpBiff4File : 1 dbpCSVFile : 2 dbpDATFile : 3 dbpOleBiff8File : 4
CopyToClipboard	データタイプ	dbpTextFormat : 0 dbpBiff4Format : 1 dbpOleBiff8Format : 4
SaveLargeObject	ファイル名	1～256 バイト

(4) TableList オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
Pattern	表の絞り込み条件	0～128 バイト
CountLimit	表数の制限値	<u>0</u> 1～8,192
SelectDict	接続先	1～32
Type	表の表属性	0～256 バイト

(5) ColumnList オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
TableName	表名	1～256 バイト
CountLimit	列数の制限値	<u>0</u> 1～8,192
SelectDict	接続先	1～32

(6) QueryDefine オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
Exclusive	排他オプション	0～64 バイト

(7) Tables コレクションオブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
Add	抽出表	1～256 バイト

(8) Table オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
TableName	抽出表	1～256 バイト

(9) Columns コレクションオブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
Add	抽出列	1～256 バイト

(10) Column オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値/文字定数：値
ColumnName	抽出列	1～256 バイト
SortMode	ソートモード	<u>dbpNormal</u> , <u>dbpNoSort</u> : 0 dbpAscend : 1 dbpDescend : 2
SortPriority	ソート優先順位	1～255
FunctionMode	集合関数	<u>dbpNormal</u> , <u>dbpNoFunction</u> : 0 dbpGroup : 1 dbpMaximum : 2 dbpMinimum : 3 dbpCount : 4 dbpAverage : 5 dbpSum : 6

(11) Conditions コレクションオブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
Relation	条件関連	0~1,024 バイト
Add	抽出条件式	1~30,000 バイト

(12) Condition オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
Text	抽出条件式	1~30,000 バイト

(13) UpdateDefine オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値/文字定数：値
Exclusive	排他オプション	0~64 バイト
UpdateMode	更新実行モード	<u>dbpNormal</u> : 0 dbpVariable : 1
TableName	更新表	1~256 バイト
NCharPrefixMode	NCHAR 定数モード	<u>dbpNCHARPrefix</u> : 0 dbpCHARPrefix : 1
Execute	更新種別	dbpUpdate : 0 dbpInsert : 1 dbpDelete : 2

(14) Update オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
KeyColumn	列名	1~256 バイト
KeyValue	データ	0~30,000 バイト
Column	列名	1~256 バイト
UpdateValue	データ	0~30,000 バイト
UpdateValueB	データ	0~2 ギガバイト
AddKey	列名	1~256 バイト
	データ	0~30,000 バイト
AddUpdateValue	列名	1~256 バイト
	データ	0~30,000 バイト

プロパティ/メソッド	項目	制限値
AddUpdateValueB	列名	1～256 バイト
	データ	0～2 ギガバイト

(15) Insert オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
Column	列名	1～256 バイト
InsertValue	データ	0～30,000 バイト
InsertValueB	データ	0～2 ギガバイト
AddInsertValue	列名	1～256 バイト
	データ	0～30,000 バイト
AddInsertValueB	列名	1～256 バイト
	データ	0～2 ギガバイト

(16) Delete オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
KeyColumn	列名	1～256 バイト
KeyValue	データ	0～30,000 バイト
AddKey	列名	1～256 バイト
	データ	0～30,000 バイト

(17) Catalog オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
FileLoad	ファイル名	1～256 バイト

(18) VariableList オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
VariableValue	実行値	1～30,000 バイト

(19) Procedure オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
ListPattern	絞り込み条件	0~128 バイト
GetParam	プロシジャ名	1~256 バイト

(20) ParamList オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値/文字定数：値
ParamCodeType	取得モード	dbpBINARY：-2 dbpCHAR：1
ParamType	パラメタ種別	dbpParamTypeUnknown：0 dbpParamInput：1 dbpParamInputOutput：2 dbpResultCol：3 dbpParamOutput：4 dbpReturnValue：5
ParamPrecision	精度	1~2,147,483,647
ParamScale	小数点以下けた数	-2,147,483,648~2,147,483,647
ParamValue	データ	0~30,000 バイト
ParamValueB	データ	0~2 ギガバイト
ParamValueMaxLength	データ長	0~2,147,483,647

付録 D.2 CommuniNet 接続のときのプロパティとメソッドの制限値

CommuniNet 接続のときの、DBPARTNER2 Client の ActiveX コントロールのプロパティとメソッドの制限値を、次の(1)~(17)に示します。表中のアンダラインは、省略時の仮定値を示します。ここでは、制限値があるプロパティ又はメソッドの文字定数について説明しています。制限値がない文字定数は掲載していません。

(1) DBPARTNER オブジェクト (CommuniNet 接続)

プロパティ/メソッド	項目	制限値/文字定数：値
CodeChangeMode	コード変換種別	<u>dbpServer</u> ：0 dbpClient：1
CommitMode	トランザクション終了モード	<u>dbpAuto</u> ：0 dbpManual：1

プロパティ/メソッド	項目	制限値/文字定数：値
DictPath	パス	0～256 バイト
TableDefSource	情報の要求先	<u>dbpDB</u> ：0 dbpDICT：1 dbpDBDICT：2
LogoffFileName	マクロスクリプトファイル名	1～256 バイト
LogonFileName	マクロスクリプトファイル名	1～256 バイト
MacroTimeLimit	マクロスクリプト実行待ち時間	1～600
ResponseLimit	キャンセル応答待ち時間	0～3600
TargetDatabase	データベース種別	<u>dbpV3RD</u> ：0 dbpV3SD：1 dbpV3PDM：2 dbpV3USR：3 dbpV3RDB1：4 dbpVKRDBF：5 dbpVKSPL：6 dbpV1PDM：7 dbpV1USR：8 dbpV1SPL：9
TargetServer	サーバプログラム種別	<u>dbpHoap</u> ：0 dbpAce3：1
TerminalMode	端末起動モードの種別	<u>dbpHide</u> ：0 dbpIcon：1 dbpShow：2
TerminalName	端末名称	1～14 バイト
UserID	ユーザ ID	0～8 バイト
Password	パスワード	0～8 バイト

(2) Result オブジェクト (CommuniNet 接続)

プロパティ/メソッド	項目	制限値/文字定数：値
QueryCountLimit	抽出行数の制限値	0 1～999,999,999
HeaderOutput	ヘッダ出力モード	<u>FALSE</u> , TRUE
FieldHeader	ヘッダ	0～256 バイト
SaveToFile	ファイル名	1～256 バイト
	データタイプ	<u>dbpTextFormat</u> , <u>dbpTextFile</u> ：0

プロパティ/メソッド	項目	制限値/文字定数：値
		dbpBiff4Format, dbpBiff4File : 1 dbpCSVFile : 2 dbpDATFile : 3 dbpOleBiff8File : 4
CopyToClipboard	データタイプ	dbpTextFormat : 0 dbpBiff4Format : 1 dbpOleBiff8Format : 4

(3) TableList オブジェクト (CommuniNet 接続)

プロパティ/メソッド	項目	制限値/文字定数：値
Pattern	表の絞り込み条件	0~128 バイト
CountLimit	表数の制限値	0 1~8,192
SelectDict	接続先	0~32 バイト
TablePasswordMode	表パスワード種別	dbpNoPassword : 0 dbpSPassword : 1 dbpUPassword : 2 dbpSUPassword : 3

(4) ColumnList オブジェクト (CommuniNet 接続)

プロパティ/メソッド	項目	制限値
TableName	表名	1~128 バイト
CountLimit	列数の制限値	0 1~8,192
SelectDict	接続先	0~32 バイト
TablePassword	表パスワード	0~8 バイト

(5) QueryDefine オブジェクト (CommuniNet 接続)

プロパティ/メソッド	項目	制限値
Exclusive	排他オプション	0~64 バイト

(6) Tables コレクションオブジェクト (CommuniNet 接続)

プロパティ/メソッド	項目	制限値
Add	抽出表	1~128 バイト

(7) Table オブジェクト (CommuniNet 接続)

プロパティ/メソッド	項目	制限値
TableName	抽出表	1~128 バイト
TablePassword	表パスワード	0~8 バイト

(8) Columns コレクションオブジェクト (CommuniNet 接続)

プロパティ/メソッド	項目	制限値
Add	—	1~256 バイト

(9) Column オブジェクト (CommuniNet 接続)

プロパティ/メソッド	項目	制限値/文字定数：値
ColumnName	—	1~256 バイト
SortMode	—	<u>dbpNormal</u> , <u>dbpNoSort</u> : 0 dbpAscend : 1 dbpDescend : 2
SortPriority	ソート優先順位	1~255
FunctionMode	集合関数	<u>dbpNormal</u> , <u>dbpNoFunction</u> : 0 dbpGroup : 1 dbpMaximum : 2 dbpMinimum : 3 dbpCount : 4 dbpAverage : 5 dbpSum : 6

(10) Conditions コレクションオブジェクト (CommuniNet 接続)

プロパティ/メソッド	項目	制限値/文字定数：値
Relation	条件関連	0~1,024 バイト
Add	抽出条件式	1~256 バイト

(11) Condition オブジェクト (CommuniNet 接続)

プロパティ/メソッド	項目	制限値
Text	抽出条件式	1～256 バイト

(12) UpdateDefine オブジェクト (CommuniNet 接続)

プロパティ/メソッド	項目	制限値/文字定数：値
Exclusive	排他オプション	0～64 バイト
UpdateMode	更新実行モード	dbpNormal : 0 dbpVariable : 1
TableName	更新表	1～128 バイト
TablePassword	表パスワード	0～8 バイト
Execute	更新種別	dbpUpdate : 0 dbpInsert : 1 dbpDelete : 2

(13) Update オブジェクト (CommuniNet 接続)

プロパティ/メソッド	項目	制限値
KeyColumn	列名	1～64 バイト
KeyValue	データ	0～256 バイト
Column	列名	1～64 バイト
UpdateValue	データ	0～256 バイト
AddKey	列名	1～64 バイト
	データ	0～256 バイト
AddUpdateValue	列名	1～64 バイト
	データ	0～256 バイト

(14) Insert オブジェクト (CommuniNet 接続)

プロパティ/メソッド	項目	制限値
Column	列名	1～64 バイト
InsertValue	データ	0～256 バイト
AddInsertValue	列名	1～64 バイト
	データ	0～256 バイト

(15) Delete オブジェクト (CommuniNet 接続)

プロパティ/メソッド	項目	制限値
KeyColumn	列名	1～64 バイト
KeyValue	データ	0～256 バイト
AddKey	列名	1～64 バイト
	データ	0～256 バイト

(16) Catalog オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
FileLoad	ファイル名	1～256 バイト

(17) VariableList オブジェクト (ODBC 接続)

プロパティ/メソッド	項目	制限値
VariableValue	実行値	1～256 バイト

付録 E 列の属性

DBPARTNER2 Client のアプリケーションで設定する文字定数の属性を、ODBC 接続と CommuniNet 接続に分けて説明します。

付録 E.1 ODBC 接続のときの列の属性

ODBC 接続のときの、DBPARTNER2 Client の文字定数の値を表 E-1 に示します。

表 E-1 DBPARTNER2 Client の文字定数の値 (ODBC 接続)

列属性の文字定数	数値	説明	ODBC のデータ型
dbpTypeCHAR	1	固定長の文字列	SQL_CHAR
dbpTypeNUMERIC	2	符号付き固定小数点数値 ^{※1}	SQL_NUMERIC
dbpTypeDECIMAL	3	符号付き固定小数点数値 ^{※1}	SQL_DECIMAL
dbpTypeINTEGER	4	符号付き固定小数点数値 ^{※1}	SQL_INTEGER
dbpTypeSMALLINT	5	符号付き固定小数点数値 ^{※1}	SQL_SMALLINT
dbpTypeFLOAT	6	符号付き浮動小数点数値 ^{※1}	SQL_FLOAT
dbpTypeREAL	7	符号付き浮動小数点数値 ^{※1}	SQL_REAL
dbpTypeDOUBLE	8	符号付き浮動小数点数値 ^{※1}	SQL_DOUBLE
dbpTypeDATE	9	日付データ	SQL_DATE
dbpTypeTIME	10	時刻データ	SQL_TIME
dbpTypeTIMESTAMP	11	タイムスタンプデータ	SQL_TIMESTAMP
dbpTypeVARCHAR	12	可変長文字列	SQL_VARCHAR
dbpTypeLONGVARCHAR	-1	可変長文字列	SQL_LONGVARCHAR
dbpTypeBINARY	-2	固定長バイナリデータ	SQL_BINARY
dbpTypeVARBINARY	-3	可変長バイナリデータ	SQL_VARBINARY
dbpTypeLONGVARBINARY	-4	可変長バイナリデータ	SQL_LONGVARBINARY
dbpTypeBIGINT	-5	固定小数点数値 ^{※1}	SQL_BIGINT
dbpTypeTINYINT	-6	符号付き固定小数点数値 ^{※1}	SQL_TINYINT
dbpTypeBIT	-7	1 けたのバイナリ値	SQL_BIT
dbpTypeNCHAR ^{※2}	-1001	固定長の各国文字列	なし
dbpTypeNVARCHAR ^{※2}	-1002	可変長の各国文字列	なし

列属性の文字定数	数値	説明	ODBC のデータ型
dbpTypeLONGNVARCHAR ^{※2}	-1003	可変長長各国文字列	なし
dbpTypeMCHAR ^{※2}	-1004	固定長の混在文字列	なし
dbpTypeMVARCHAR ^{※2}	-1005	可変長の混在文字列	なし
dbpTypeLONGMVARCHAR ^{※2}	-1006	可変長長混在文字列	なし
dbpTypeNUMERICUNS ^{※2}	-2000	符号なし固定小数点数値 ^{※1}	なし

注※1

有効けた数、小数部けた数については、各データベースの制限に従います。

注※2

DBPARTNER2 Client の ODBC ドライバ (DBPARTNER ODBC Driver) を使ったときにだけ使えます。

付録 E.2 CommuniNet 接続のときの列の属性

(1) 列属性の文字定数

CommuniNet 接続のときの、DBPARTNER2 Client の文字定数の値を表 E-2 に示します。

表 E-2 DBPARTNER2 Client の文字定数の値 (CommuniNet 接続)

列属性の文字定数	数値	説明
dbpTypeCHAR	1	固定長の文字列
dbpTypeNUMERIC	2	符号付き固定小数点数値
dbpTypeINTEGER	4	符号付き固定小数点数値
dbpTypeFLOAT	6	符号付き浮動小数点数値
dbpTypeREAL	7	符号付き浮動小数点数値
dbpTypeDATE	9	日付データ
dbpTypeTIME	10	時刻データ
dbpTypeVARCHAR	12	可変長文字列
dbpTypeINTERVAL_HOUR_TO_SECOND	112	時間間隔データ
dbpTypeBINARY	-2	固定長バイナリデータ
dbpTypeNCHAR	-1001	固定長の各国文字列
dbpTypeNVARCHAR	-1002	可変長の各国文字列
dbpTypeMCHAR	-1004	固定長の混在文字列
dbpTypeMVARCHAR	-1005	可変長の混在文字列

列属性の文字定数	数値	説明
dbpTypeNUMERICUNS	-2000	符号なし固定小数点数値
dbpTypeINTERVAL_YEAR_TO_DAY	-2001	日間隔データ

(2) 列属性とデータベースの属性

DBPARTNER2 Client が返す列属性と使うデータベースの列属性の対応を、データベースごとに次の (a) ~ (h) に示します。

(a) VOS3 XDM/RD E2 のとき

列属性の文字定数	ACE3 定義のデータ型	XDM/RD E2 のデータ型
dbpTypeCHAR	C	CHARACTER
dbpTypeVARCHAR	VC	VARCHAR
	LC	LONG VARCHAR
dbpTypeNCHAR	N	NCHAR
dbpTypeNVARCHAR	VN	NVARCHAR
	LN	LONG NVARCHAR
dbpTypeMCHAR	MC	MCHAR
dbpTypeMVARCHAR	VM	MVARCHAR
	LM	LONG MVARCHAR
dbpTypeINTEGER	B	INTEGER
	B	SMALLINT
dbpTypeNUMERIC	P	DECIMAL
	LP	LARGE DECIMAL
dbpTypeFLOAT	D	FLOAT
dbpTypeREAL	F	SMALLFLT
dbpTypeDATE	DT	DATE
dbpTypeINTERVAL_YEAR_TO_DAY	ID	INTERVAL YEAR TO DAY

(b) VOS3 XDM/SD E2 のとき

列属性の文字定数	ACE3 定義のデータ型	XDM/SD E2 のデータ型
dbpTypeCHAR	C	CHARACTER
dbpTypeNCHAR	N	NCHARACTER
dbpTypeINTEGER	M	COMP

列属性の文字定数	ACE3 定義のデータ型	XDM/SD E2 のデータ型
dbpTypeNUMERIC	P	PACKED DECIMAL FIXED
	Z	UNPACKED DECIMAL FIXED
dbpTypeBINARY	X	BIT

(c) VOS3 RDB1 のとき

列属性の文字定数	ACE3 定義のデータ型	RDB1 のデータ型
dbpTypeCHAR	C	CHAR
dbpTypeVARCHAR	VC	VARCHAR
	LC	LONGVARCHAR
dbpTypeNCHAR	N	NCHAR
dbpTypeNVARCHAR	VN	NVARCHAR
dbpTypeINTEGER	B	SMALLINT
	B	INTEGER
dbpTypeNUMERIC	P	DECIMAL
dbpTypeFLOAT	D	FLOAT
dbpTypeREAL	F	SMALLFLT

(d) VOS3 PDMII E2 のとき

列属性の文字定数	ACE3 定義のデータ型	PDMII E2 のデータ型
dbpTypeCHAR	C	X
dbpTypeVARCHAR	VC	—
dbpTypeNCHAR	N	N
dbpTypeNVARCHAR	VN	—
dbpTypeINTEGER	B	B
dbpTypeNUMERIC	P	P
	Z	Z
dbpTypeFLOAT	D	F
dbpTypeREAL	F	F

(e) VOS3 ACE3 ユーザファイルのとき

列属性の文字定数	ACE3 定義のデータ型
dbpTypeCHAR	C
dbpTypeVARCHAR	VC
dbpTypeNCHAR	N
dbpTypeNVARCHAR	VN
dbpTypeMCHAR	MC
dbpTypeMVARCHAR	VM
dbpTypeINTEGER	B
dbpTypeNUMERIC	P
	Z
dbpTypeNUMERICUNS	UP
	UZ
dbpTypeFLOAT	D
dbpTypeREAL	F
dbpTypeDATE	DT
dbpTypeINTERVAL_YEAR_TO_DAY	ID
dbpTypeTIME	TM
dbpTypeINTERVAL_HOUR_TO_SECOND	IS

(f) VOS1 PDMII E2 のとき

列属性の文字定数	ACE3 定義のデータ型	PDMII E2 のデータ型
dbpTypeCHAR	C	X
dbpTypeVARCHAR	VC	X
dbpTypeNCHAR	N	N
dbpTypeNVARCHAR	VN	N
dbpTypeINTEGER	B	B
dbpTypeNUMERIC	P	P
	Z	Z
dbpTypeFLOAT	D	D
dbpTypeREAL	F	F

(g) VOS1 ACE3 ユーザファイルのとき

列属性の文字定数	ACE3 定義のデータ型
dbpTypeCHAR	C
dbpTypeVARCHAR	VC
dbpTypeNCHAR	N
dbpTypeNVARCHAR	VN
dbpTypeMCHAR	MC
dbpTypeMVARCHAR	VM
dbpTypeINTEGER	B
dbpTypeNUMERIC	P
	Z
dbpTypeFLOAT	D
dbpTypeREAL	F

(h) VOSK RDB 編成ファイルのとき

列属性の文字定数	物理フィールド定義でのデータ型
dbpTypeCHAR	C
dbpTypeBINARY	X
dbpTypeNCHAR	N
dbpTypeMCHAR	M
dbpTypeNUMERIC	Z
	P
dbpTypeNUMERICUNS	W
dbpTypeINTEGER	B

付録 F 繰り返し列と配列列での制限事項

表の列に繰り返し列や配列列を使っている場合の制限事項について説明します。

繰り返し列を含む表は、データの更新、削除はできません。

付録 F.1 データの抽出、更新、削除時の制限事項

データの抽出、更新、削除をするときに、繰り返し列又は配列列の列名をメソッドやプロパティに設定する場合、列名を次に示す形式で設定してください。

列名の形式

"列名"(n)

n: 要素数を示します。

要素数を囲む括弧の形式はデータベースにより異なります。各データベースの括弧の形式にあわせて設定してください。

(例)

VOSK RDB 編成ファイルの場合: "列名"[n]

例として、商品コード、売り上げ、及び支店在庫（繰り返し列又は配列列）で構成される在庫管理表（A）から、支店在庫のデータを抽出する場合のコーディングについて説明します。

- 繰り返し列又は配列列が含まれる在庫管理表（A）

商品コード	支店在庫 (1)	支店在庫 (2)	売り上げ
A0100	100	110	1000
A0200	200	220	2000
A0300	300	330	3000

在庫管理表 1 から、支店在庫 (2) の 220 の行を抽出する場合のコーディングは次のとおりです。データの更新や削除をするときも列名の指定は同じ形式になります。

- コーディング例

```

接続などの処理のコーディング
:
:
DBPARTNER1.QueryDefine.Tables.Add "在庫管理表 1"
DBPARTNER1.QueryDefine.Conditions.Add ("支店在庫"(2) = 220")
DBPARTNER1.QueryDefine.Execute
:
:
結果の取得などの処理のコーディング

```

繰り返し列を含む表に、データの追加はできません。

付録 F.2 データの追加時の制限事項

データを追加するときに、配列列の列名をメソッドやプロパティに設定する場合、列名を次に示す形式で設定してください。

列名の形式

```
"列名"
```

注意

- データを追加するときには、すべての要素のデータを設定してください。データを設定していない要素がある場合には、データの追加処理を実行したときにエラーになります。
- 繰り返し列又は配列列の要素を連続して設定してください。設定する要素に配列列以外の列がある場合には、データの追加処理を実行したときにエラーになります。

例として、商品コード、売り上げ、及び支店在庫（配列列）で構成される在庫管理表（B）にデータを追加する場合のコーディングについて説明します。

• 配列列が含まれる在庫管理表（B）

商品コード	支店在庫（1）	支店在庫（2）	売り上げ
B0100	50	55	5000

在庫管理表 2 にデータを追加する場合のコーディングは次のとおりです。追加するデータは、商品コード「B0200」、支店在庫（1）「60」、支店在庫（2）「66」、売り上げ「6000」とします。

• コーディングの例

接続などの処理のコーディング

：

AddInsertValue “商品コード”, B0200

AddInsertValue “支店在庫”, “60”

AddInsertValue “支店在庫”, “66”

AddInsertValue “売り上げ”, 6000

：

結果の取得などの処理のコーディング

●補足

誤りの場合の設定例を次に示します。間違いやすい点なので、ご注意ください。

- ・データを設定していない要素がある

AddInsertValue “商品コード”, B0200

AddInsertValue “支店在庫”, “60”

AddInsertValue “売り上げ”, 6000

- ・要素が連続して設定されていない

AddInsertValue “商品コード”, B0200

AddInsertValue “支店在庫”, “60”

AddInsertValue “売り上げ”, 6000

AddInsertValue “支店在庫”, “66”

付録 G このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報を示します。

付録 G.1 関連マニュアル

関連マニュアルを次に示します。必要に応じてお読みください。

(1) サーバ側

サーバの OS ごとに関連するマニュアルを次に示します。

UNIX/Windows の場合

- DABroker (3020-6-031)
- スケーラブルデータベースサーバ HiRDB Version 8 コマンドリファレンス (UNIX(R)用) (3000-6-355)
- スケーラブルデータベースサーバ HiRDB Version 8 コマンドリファレンス (Windows(R)用) (3020-6-355)
- スケーラブルデータベースサーバ HiRDB Version 8 SQL リファレンス (3020-6-357)
- スケーラブルデータベースサーバ HiRDB Version 8 メッセージ (3020-6-358)
- スケーラブルデータベースサーバ HiRDB Version 9 コマンドリファレンス (UNIX(R)用) (3000-6-455)
- スケーラブルデータベースサーバ HiRDB Version 9 コマンドリファレンス (Windows(R)用) (3020-6-455)
- スケーラブルデータベースサーバ HiRDB Version 9 SQL リファレンス (3020-6-457)
- スケーラブルデータベースサーバ HiRDB Version 9 メッセージ (3020-6-458)

注

このマニュアルの HiRDB マニュアルへの参照指示では、HiRDB のバージョンと OS 名は省略して表記しています。HiRDB マニュアルを参照するときは、該当するバージョンと該当する OS のマニュアルをご利用ください。

VOS3 の場合

- VOS3 データマネジメントシステム XDM E2 系 プログラム作成の手引 (XDM/SD) (6190-6-652)
- VOS3 データマネジメントシステム XDM E2 系 プログラム作成の手引 (XDM/RD E2) (6190-6-638)
- VOS3 データマネジメントシステム XDM E2 系 SQL リファレンス (XDM/RD E2) (6190-6-656)
- VOS3 データマネジメントシステム XDM E2 系 メッセージ (XDM/RD E2) (6190-6-643)

- VOS3 データマネジメントシステム XDM E2 系 運用の手引 -解説編- (6190-6-628)
- VOS3 データマネジメントシステム XDM E2 系 メッセージ (コンソール・端末出力メッセージ) (6190-6-629)
- VOS3/FS システムメッセージ/システムコード (6190-3-103)
- VOS3 データマネジメントシステム RDB1 解説 (6180-6-550)
- VOS3 データマネジメントシステム RDB1 EQL 文法 (6180-6-551)
- VOS3 データマネジメントシステム RDB1 使用の手引 (6180-6-552)
- VOS3 Database Connection Server (6190-6-648)

VOSK の場合

- VOSK システム OA (650-7-321)
- VOSK/FS Database Connection Server (650-3-243)

VOS1 の場合

- VOS1 データマネジメントシステム RDB1 E2 使用の手引 (6150-6-561)
- VOS1 Database Connection Server (6150-6-802)

(2) クライアント側 (クライアント PC 側)

DBPARTNER2 Client を組み込むクライアント側で関連するマニュアルを次に示します。

クライアント用データベースアクセス関連

- DBPARTNER2 Client 操作ガイド (3020-6-151)

ODBC 接続の場合

- DBPARTNER ODBC 3.0 Driver (3020-6-073)

CommuniNet 接続の場合

- CommuniNet Version 3 (3020-3-821)
- CommuniNet Extension Version 3 (3020-3-822)
- CommuniNet Version 4 (3020-3-N71)
- CommuniNet Extension Version 4 (3020-3-N72)

付録 G.2 このマニュアルで使用する略称

このマニュアルでは、製品名を次に示す略称で表記しています。

品名	略称
Microsoft(R) Excel	Excel
Microsoft(R) SQL Server	Microsoft SQL Server 又は SQL Server
Microsoft(R) Visual Basic	Visual Basic
Microsoft(R) Visual C++	Visual C++
Microsoft(R) Windows(R) Sockets	Windows Sockets
Microsoft(R) Internet Explorer	Internet Explorer
ORACLE(R)のデータベース	ORACLE
Sybase Adaptive Server Anywhere	Sybase Adaptive Server Anywhere 又は Adaptive Server Anywhere
VOS3/FS, VOS3/AS, VOS3/ES1	VOS3
VOS1/FS, VOS1/ES2	VOS1
VOSK/FS, VOSK	VOSK

このマニュアルでは、オペレーティングシステムを次に示す略称で表記しています。

製品名	略称	
Microsoft(R) Windows Vista(R) Business	Windows Vista	Windows
Microsoft(R) Windows Vista(R) Enterprise		
Microsoft(R) Windows Vista(R) Ultimate		
Microsoft(R) Windows(R) 7 Professional	Windows 7	
Microsoft(R) Windows(R) 7 Enterprise		
Microsoft(R) Windows(R) 7 Ultimate		
Microsoft(R) Windows(R) 8 Pro	Windows 8	
Microsoft(R) Windows(R) 8 Enterprise		
Microsoft(R) Windows(R) 8.1 Pro		
Microsoft(R) Windows(R) 8.1 Enterprise		
Microsoft(R) Windows(R) 10 Pro	Windows 10	
Microsoft(R) Windows(R) 10 Enterprise		
Microsoft(R) Windows Server(R) 2008 Standard	Windows Server 2008	
Microsoft(R) Windows Server(R) 2008 Enterprise		
Microsoft(R) Windows Server(R) 2008 Datacenter		
Microsoft(R) Windows Server(R) 2008 R2 Standard		

製品名	略称
Microsoft(R) Windows Server(R) 2008 R2 Enterprise	Windows Server 2012
Microsoft(R) Windows Server(R) 2008 R2 Datacenter	
Microsoft(R) Windows Server(R) 2012 Standard	
Microsoft(R) Windows Server(R) 2012 Datacenter	
Microsoft(R) Windows Server(R) 2012 R2 Standard	
Microsoft(R) Windows Server(R) 2012 R2 Datacenter	
Microsoft(R) MS-DOS(R)	MS-DOS

このマニュアルでは、データベースに接続するときを使う通信形態を次に示すように表記しています。通信形態とマニュアルでの表記を次に示します。

通信形態	マニュアルでの表記
ODBC ドライバ	ODBC 接続
CommuniNet Extension	CommuniNet 接続

このマニュアルでは、WWW ブラウザを **Web ブラウザ**、WWW サーバを **Web サーバ**と表記します。

付録 G.3 略語一覧

このマニュアルで使う英略語の一覧を次に示します。

英略語	英字の表記
ACE3	<u>A</u> vailable <u>C</u> ommand language for <u>E</u> nd users <u>3</u>
ACE3 E2	<u>A</u> vailable <u>C</u> ommand language for <u>E</u> nd users <u>3</u> <u>E</u> xtended Version <u>2</u>
ACE3 E3	<u>A</u> vailable <u>C</u> ommand language for <u>E</u> nd users <u>3</u> <u>E</u> xtended Version <u>3</u>
AP	<u>A</u> pplication <u>P</u> rogram
API	<u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface
CD-ROM	<u>C</u> ompact <u>D</u> isc - <u>R</u> ead <u>O</u> nly <u>M</u> emory
DB	<u>D</u> atab <u>a</u> se
DBMS	<u>D</u> atab <u>a</u> se <u>M</u> anagement <u>S</u> ystem
DDL	<u>D</u> ata <u>D</u> efinition <u>L</u> anguage
DF/1	<u>D</u> istributing <u>F</u> acility / for VOS <u>1</u>
DF/K	<u>D</u> istributing <u>F</u> acility / for VOS <u>K</u>
DLL	<u>D</u> ynamic <u>L</u> inking <u>L</u> ibrary

英略語	英字の表記
DML	<u>D</u> ata <u>M</u> anipulation <u>L</u> anguage
EBCDIK	<u>E</u> xtended <u>B</u> inary <u>C</u> oded <u>D</u> ecimal <u>I</u> nterchange <u>K</u> ana code
EUC	<u>E</u> xtended <u>U</u> NIX <u>C</u> ode
GUI	<u>G</u> raphical <u>U</u> ser <u>I</u> nterface
HOAPDBS E2	<u>H</u> igh <u>L</u> evel <u>O</u> bject <u>M</u> anagement and <u>P</u> rocessing <u>D</u> atabase <u>S</u> ervice <u>E</u> xtended Version <u>2</u>
HOAPDBS/PC	<u>H</u> igh <u>L</u> evel <u>O</u> bject <u>M</u> anagement and <u>P</u> rocessing <u>D</u> atabase <u>S</u> ervice / <u>P</u> ersonal <u>C</u> omputer Feature
HTML	<u>H</u> yper <u>T</u> ext <u>M</u> arkup <u>L</u> anguage
JIS	<u>J</u> apanese <u>I</u> ndustrial <u>S</u> tandard code
KEIS	<u>K</u> anji processing <u>E</u> xtended <u>I</u> nformation <u>S</u> ystem
LAN	<u>L</u> ocal <u>A</u> rea <u>N</u> etwork
MAPI	Microsoft <u>M</u> essaging <u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface
OA	<u>O</u> ffice <u>A</u> utomation
ODBC	<u>O</u> pen <u>D</u> atabase <u>C</u> onnectivity
OS	<u>O</u> perating <u>S</u> ystem
OSCF	<u>O</u> perating <u>S</u> ystem <u>C</u> ontrol <u>F</u> acility
OSI	<u>O</u> pen <u>S</u> ystems <u>I</u> nterconnection
PC	<u>P</u> ersonal <u>C</u> omputer
PDM II	<u>P</u> ractical <u>D</u> ata <u>M</u> anager <u>II</u>
PDM II E2	<u>P</u> ractical <u>D</u> ata <u>M</u> anager <u>II</u> <u>E</u> xtended Version <u>2</u>
RD	<u>R</u> elational <u>D</u> atabase
RDB1	<u>R</u> elational <u>D</u> atabase Manager <u>1</u>
RDB1 E2	<u>R</u> elational <u>D</u> atabase Manager <u>1</u> <u>E</u> xtended Version <u>2</u>
SAM	<u>S</u> equential <u>A</u> ccess <u>M</u> ethod
SQL	<u>S</u> tructured <u>Q</u> uery <u>L</u> anguage
SQL/K	<u>S</u> tructured <u>Q</u> uery <u>L</u> anguage / <u>V</u> OS <u>K</u>
TCP/IP	<u>T</u> ransmission <u>C</u> ontrol <u>P</u> rotocol / <u>I</u> nternet <u>P</u> rotocol
UAP	<u>U</u> ser <u>A</u> pplication <u>P</u> rogram
URL	<u>U</u> niform <u>R</u> esource <u>L</u> ocator
VOS1	<u>V</u> irtual-storage <u>O</u> perating <u>S</u> ystem <u>1</u>
VOS3	<u>V</u> irtual-storage <u>O</u> perating <u>S</u> ystem <u>3</u>

英略語	英字の表記
VOSK	<u>V</u> irtual- <u>s</u> torage <u>O</u> perating <u>S</u> ystem <u>K</u> indness
WS	<u>W</u> orkstation
WWW	<u>W</u> orld <u>W</u> ide <u>W</u> eb
XDM/DF	<u>E</u> xtensible <u>D</u> ata <u>M</u> anager / <u>D</u> istributing <u>F</u> acility
XDM/RD E2	<u>E</u> xtensible <u>D</u> ata <u>M</u> anager / <u>R</u> elational <u>D</u> atabase <u>E</u> xtended <u>V</u> ersion <u>2</u>
XDM/SD E2	<u>E</u> xtensible <u>D</u> ata <u>M</u> anager / <u>S</u> equential <u>D</u> atabase <u>E</u> xtended <u>V</u> ersion <u>2</u>

付録 G.4 KB (キロバイト) などの単位表記について

1KB (キロバイト), 1MB (メガバイト), 1GB (ギガバイト), 1TB (テラバイト) はそれぞれ $1,024$ バイト, $1,024^2$ バイト, $1,024^3$ バイト, $1,024^4$ バイトです。

(英字)

NCHAR 定数モード

各国文字列の列に対する更新データに付けるプレフィックスのことです。

SQL 文

データベースを操作したり定義したりするための高級言語のことです。DBPARTNER2 Client では、アプリケーションからデータを抽出するときに使います。

(カ行)

カスタムインストール

作成するファイルや初期値の内容を、任意に設定してインストールする方法です。

クイックインストール

作成するファイルや初期値の内容を、DBPARTNER2 Client が仮定してインストールする方法です。

クライアント/サーバ

プログラムとプログラムの通信方法の一つです。業務処理を依頼する方をクライアント、要求を受けて業務を実行する方をサーバといいます。DBPARTNER2 Client を使った業務では、DBPARTNER2 Client がクライアント、データベースがサーバになります。クライアント/サーバとは、プログラム間の相対的な関係を示す用語です。

更新種別

データベースへ更新データをどのように反映するかを意味します。更新種別には、変更、追加、削除があります。変更とは、データベースの表の行のデータを変更することです。追加とは、データベースの表に行を追加することです。削除とは、データベースの表の行を削除することです。

更新処理

データベースのデータを変更、追加又は削除する処理のことです。

更新表

更新処理の対象になる表のことです。

サーバ

ほかのシステムやプログラムから要求を受けて、業務を実行するシステム又はプログラムのことです。代表的なサーバの業務は、データベース管理です。DBPARTNER2 Client を使った業務では、HiRDB や ORACLE などのデータベース、DABroker などのサーバプログラムを総称してサーバといいます。

サーバプログラム

DBPARTNER2 Client とデータベースの間に位置するサーバのプログラムのことです。DBPARTNER2 Client の ActiveX コントロール仕様では、次に示すサーバプログラムと接続できます。

ODBC 接続のとき

DABroker, VOS3 Database Connection Server, VOS1 Database Connection Server, VOSK Database Connection Server

CommuniNet 接続のとき

VOS3 HOAPDBS E2, VOS3 ACE3 E3, VOSK HOAPDBS/PC, VOS1 DBPARTNER/Link

削除行条件

削除する行を特定するための 1 列分の条件のことです。

削除処理

更新表の 1 行又は複数行を削除する処理のことです。

集合関数

抽出する列中のデータから、最大値、最小値、平均値、合計値、及び件数を抽出する関数です。

条件

データを抽出するときに、データを絞り込む方法です。条件には、=, >, <, ≠, ≥, ≤, NULL, ^NULL, IN, ^IN, LIKE, ^LIKE, BETWEEN, ^BETWEEN, 列比較, 列演算があります。

条件関連

NOT, AND, OR, 又は () を使って、複数の抽出条件式を関連付ける文字列のことです。何も指定しないと、条件式間のつながりは AND になります。

所有者

表を所有しているユーザの認可識別子のことです。

ソートモード

抽出データを昇順又は降順に並び替えるモードのことです。

ソート優先順位

複数の列に設定されているソートモードの優先順位のことです。

属性

データベースに定義している列の属性です。列の属性は、TableList オブジェクト又は Result オブジェクトで取得できます。

(タ行)

抽出条件

抽出データとして出力する行を決める条件のことです。抽出条件は、抽出条件式と条件関連から構成されます。

抽出処理

データベースからデータを取り出す処理のことです。抽出処理の対象になる表のことを抽出表といいます。抽出処理で取り出したデータのことを抽出データといいます。抽出データを絞り込む条件を抽出条件式といいます。

抽出データ格納ファイル

データベースから抽出したデータを格納するファイルです。抽出データ格納ファイルの拡張子(ファイルの種類)は、抽出するときに設定したデータの形式で決まります。

抽出列

抽出データとして出力する列のことです。

追加情報

データを追加するための 1 列分の情報のことです。

追加処理

更新表に 1 行又は複数行を追加する処理のことです。

データソース名

[コントロールパネル] の [ODBC データソース] アイコンに登録されているユーザデータソースに付けてある名称のことです。

データの更新

データベースからデータを抽出して、そのデータを DBPARTNER2 Client 上で編集してデータベースを更新することです。データを更新するときは、UpdateDefine オブジェクトを使います。

データの絞り込み

抽出するデータの対象を狭めることです。

データの抽出

データベースからデータを取り出すことです。データを抽出するときは、QueryDefine オブジェクト又は SQLDirect オブジェクトを使います。抽出したデータは、抽出データ格納ファイルに格納します。

データの転送

サーバにあるデータベースの内容を、クライアント PC のファイルやクリップボードに送ることです。

データベース

論理的に関連のあるデータの集合のことです。

トランザクション

DBPARTNER2 Client からデータベースへの処理要求の単位をトランザクションといいます。トランザクションは、Execute メソッドを実行してから Commit メソッド又は Rollback メソッドを実行するまでを一つの単位として扱います。

(八行)

排他オプション

抽出表、更新表を処理するときの排他制御を設定する文字列のことです。

表

データの集まりの単位のことです。データベースによっては、ファイルやテーブルに相当します。DBPARTNER2 Client ではファイルやテーブルを含めて、表といいます。

表絞り込み条件

取得する表情報を絞り込むための条件のことです。

表情報

データベースに定義されている表の情報のことです。表の情報には、表名、属性、所有者、注釈、ラベルがあります。

表番号

複数の表を識別するための番号のことです。

表名

表を識別するための名称のことです。自分の表と他人の表を区別したいときは、表名を所有者で修飾します。

ヘッダ

データを取得するときに列名の代わりに使う文字列のことです。

ヘッダ出力モード

抽出データの先頭行にヘッダを付けるかどうかの設定のことです。

変更行条件

変更する行を特定するための 1 列分の条件のことです。

変更情報

変更行条件で特定した行のデータを変更する 1 列分の情報のことです。

変更処理

更新表の 1 行又は複数行のデータを変更する処理のことです。

(ラ行)

列一覧

データベースの表を構成する列の一覧のことです。

列情報

データベースに定義されている表を構成する列の情報のこと。列の情報には、列名、属性、定義長、小数点以下けた数、注釈、ラベルがあります。

列番号

複数の列を識別するための番号のことです。

列名

データベースに表を定義するときに指定した列の名前のこと。データベースによっては、カラム名、欄名に相当します。DBPARTNER2 Client ではカラム名や欄名を含めて、列名といいます。複数の表を扱うときは、列名を表名で修飾します。

ワイルドカード

文字の代わりに指定する記号のことです。検索する単語の一部しか分からないとき、分からない部分をワイルドカードに置き換えて指定します。ワイルドカードは抽出する表を絞り込む操作で使えます。

索引

記号

[DBPARTNER ODBC Driver for DABroker –環境定義–] ダイアログボックス 51

[DBPARTNER ODBC Driver for DABroker –セットアップ–] ダイアログボックス 47

[ODBC データソースアドミニストレータ] ダイアログボックス 47

A

ACE3 E3 34, 38, 259

ACE3 E3 がサーバプログラムのとき 409

ACE3 ユーザファイル 38

ACE3 ユーザファイルの場合の規則 444, 450

ActiveX コントロール 25, 27, 43, 60, 254

ActiveX コントロールのオブジェクトの階層 60, 254

ActiveX コントロールのオブジェクトの文法 (ODBC 接続) 101

Adaptive Server Anywhere 35, 36

AddInsertValueB メソッド 82

AddInsertValueB メソッド [ODBC] 188

AddInsertValue メソッド 82, 273

AddInsertValue メソッド [CommuniNet] 381

AddInsertValue メソッド [ODBC] 187

AddKey メソッド 81, 82, 273, 274

AddKey メソッド [CommuniNet] 374, 387

AddKey メソッド [ODBC] 180, 194

AddUpdateValueB メソッド 81

AddUpdateValueB メソッド [ODBC] 181

AddUpdateValue メソッド 81

AddUpdateValue メソッド [CommuniNet] 374

AddUpdateValue メソッド [ODBC] 181

Add メソッド 81, 82, 273, 274

Add メソッド [CommuniNet] 344, 349, 357, 367, 376, 382

Add メソッド [ODBC] 153, 157, 164, 174, 183, 190

AND 28

APPGALLERY 4.0 29

AP 作成ウィザード 478

B

BLOB 型データ 29

C

CancelEx メソッド [CommuniNet] 301

CancelEx メソッド [ODBC] 110

Cancel メソッド 65, 259

Cancel メソッド [CommuniNet] 301

Cancel メソッド [ODBC] 110

Catalog オブジェクト 87

Catalog オブジェクトの詳細 [CommuniNet] 388

Catalog オブジェクトの詳細 [ODBC] 195

Close メソッド 77, 271

Close メソッド [CommuniNet] 314

Close メソッド [ODBC] 127

CodeChangeMode プロパティ [CommuniNet] 291

ColumnComment プロパティ 72, 73, 267, 268

ColumnComment プロパティ [CommuniNet] 332

ColumnComment プロパティ [ODBC] 142

ColumnEditPattern プロパティ 267, 268

ColumnEditPattern プロパティ [CommuniNet] 333

ColumnHeader プロパティ 72, 73, 267, 268

ColumnHeader プロパティ [CommuniNet] 334

ColumnHeader プロパティ [ODBC] 143

ColumnLength プロパティ 72, 73, 267, 268

ColumnLength プロパティ [CommuniNet] 334

ColumnLength プロパティ [ODBC] 144

ColumnList オブジェクト 69, 71, 263, 266

ColumnList オブジェクトの詳細 [CommuniNet] 332

ColumnList オブジェクトの詳細 [ODBC] 142

ColumnNameLength プロパティ [ODBC] 145

ColumnName プロパティ 72, 73, 267, 268

ColumnName プロパティ [CommuniNet] 335, 352
ColumnName プロパティ [ODBC] 144, 160
ColumnNumber プロパティ 267, 268
ColumnNumber プロパティ [CommuniNet] 336
ColumnScale プロパティ 72, 73, 267, 268
ColumnScale プロパティ [CommuniNet] 336
ColumnScale プロパティ [ODBC] 145
Columns コレクションオブジェクトの詳細 [CommuniNet] 349
Columns コレクションオブジェクトの詳細 [ODBC] 157
ColumnType プロパティ 72, 73, 267, 268
ColumnType プロパティ [CommuniNet] 337
ColumnType プロパティ [ODBC] 146
Column オブジェクトの詳細 [CommuniNet] 352
Column オブジェクトの詳細 [ODBC] 160
Column プロパティ [CommuniNet] 370, 379
Column プロパティ [ODBC] 176, 185
CommitMode プロパティ 64, 67, 258, 261
CommitMode プロパティ [CommuniNet] 292
CommitMode プロパティ [ODBC] 104
Commit メソッド 67, 261
Commit メソッド [CommuniNet] 302
Commit メソッド [ODBC] 111
CommuniNet Extension Version 3.0 38
CommuniNet Version 3.0 38
CommuniNet 接続 34, 37
CommuniNet 接続でアクセスできるデータベース 38
CommuniNet 接続のときのプロパティとメソッドの制限値 505
CommuniNet 接続のときの列の属性 512
CommuniNet 接続を使うときの参照情報 392
Conditions コレクションオブジェクトの詳細 [CommuniNet] 356
Conditions コレクションオブジェクトの詳細 [ODBC] 163
Condition オブジェクトの詳細 [CommuniNet] 360
Condition オブジェクトの詳細 [ODBC] 167

ConnectDialogMode プロパティ 64
ConnectDialogMode プロパティ [ODBC] 105
Connect メソッド 65, 259
Connect メソッド [CommuniNet] 303
Connect メソッド [ODBC] 111
Convert メソッド [CommuniNet] 388
Convert メソッド [ODBC] 195
CopyToClipboard メソッド 77, 271
CopyToClipboard メソッド [CommuniNet] 314
CopyToClipboard メソッド [ODBC] 128
CountLimit プロパティ 70, 72, 73, 264, 265, 267
CountLimit プロパティ [CommuniNet] 319
CountLimit プロパティ [ODBC] 133, 147
CountLimit プロパティ 338
Count プロパティ 70, 72, 73, 264, 265, 267, 268
Count プロパティ [CommuniNet] 319, 337, 344, 349, 356, 367, 376, 382, 390
Count プロパティ [ODBC] 132, 146, 153, 157, 163, 174, 183, 190, 197, 202, 212
CSV 形式 37, 39, 77, 271, 499

D

DABroker 34, 35, 65
DABroker (DBPARTNER/Server ドライバ) 25, 30, 35, 41, 43, 45, 216
DABroker ドライバ 35, 41
DABroker ドライバ (DBPARTNER/Server ドライバ) のセットアップ手順 47
DABroker ドライバで使える ODBC 関数 217
DABroker ドライバのデータソースと通信ドライバの設定 46
DABroker ドライバの場合に、各データベースで使える機能 239
DABroker ドライバの場合の ODBC とデータベースとのデータ型の対応 223
DABroker ドライバを選択すると作成されるファイル 45
Database Connection Server 34, 35, 37, 65
DataCheck プロパティ 76, 271
DataCheck プロパティ [CommuniNet] 307

- DataCheck プロパティ [ODBC] 118
- DataSource プロパティ 64
- DataSource プロパティ [ODBC] 106
- DAT 形式 37, 39, 77, 271, 499
- DBPARTNER/Link 35, 38, 259
- DBPARTNER2 Client 25, 43
- DBPARTNER2 Client が扱うデータの形式 499
- DBPARTNER2 Client が稼働する環境 (CommuniNet 接続) 38
- DBPARTNER2 Client が稼働する環境 (ODBC 接続) 36
- DBPARTNER2 Client が提供する ActiveX コントロールの概要 60, 254
- DBPARTNER2 Client からデータベースにアクセスする手順の概要 63, 257
- DBPARTNER2 Client で自動的にトランザクションを制御するとき 67, 261
- DBPARTNER2 Client で使えるサーバプログラム 34
- DBPARTNER2 Client とサーバを接続する形態 34
- DBPARTNER2 Client に関連する製品 25
- DBPARTNER2 Client のインストール 43
- DBPARTNER2 Client のインストールのメッセージ 465
- DBPARTNER2 Client のオブジェクトの文法 (CommuniNet 接続) [CommuniNet] 286
- DBPARTNER2 Client のオブジェクトの文法 (ODBC 接続) 101
- DBPARTNER2 Client の環境を設定する手順 41
- DBPARTNER2 Client の機能 31
- DBPARTNER2 Client の紹介 24
- DBPARTNER2 Client の製品体系 43
- DBPARTNER2 Client の操作時に起こるトラブル 461
- DBPARTNER2 Client の特長 27
- DBPARTNER2 Client の文字定数の値 511, 512
- DBPARTNER2 Client の予約語 497
- DBPARTNER2 Client を使ったデータの更新 33
- DBPARTNER2 Client を使ったデータの抽出 32
- DBPARTNER2 Client を使ったデータベースアクセスの概要 28
- DBPARTNER ODBC 3.0 Driver 35
- DBPARTNER ODBC Driver のメッセージ 468
- DBPARTNER オブジェクト 64, 65, 257, 259
- DBPARTNER オブジェクトの詳細 [CommuniNet] 290
- DBPARTNER オブジェクトの詳細 [ODBC] 104
- dbpTypeLONGVARIABLE 属性の列のデータ [ODBC] 129
- Deletes コレクションオブジェクトの詳細 [CommuniNet] 382
- Deletes コレクションオブジェクトの詳細 [ODBC] 190
- Delete オブジェクト 82, 274
- Delete オブジェクトの詳細 [CommuniNet] 385
- Delete オブジェクトの詳細 [ODBC] 192
- DF/K 34
- DF1 34
- DictPath プロパティ [CommuniNet] 293
- DictPath プロパティ [ODBC] 107
- Disconnect メソッド 66, 260
- Disconnect メソッド [CommuniNet] 304
- Disconnect メソッド [ODBC] 112

E

- EOF プロパティ 77, 271
- EOF プロパティ [CommuniNet] 308
- EOF プロパティ [ODBC] 118
- ErrorInformation プロパティ [CommuniNet] 361
- ErrorInformation プロパティ [ODBC] 168
- Exclusive プロパティ 75, 81, 82, 270, 273, 274
- Exclusive プロパティ [CommuniNet] 341, 362
- Exclusive プロパティ [ODBC] 150, 169
- ExecutedRowCount プロパティ 81, 82, 85, 273, 274
- ExecutedRowCount プロパティ [CommuniNet] 363
- ExecutedRowCount プロパティ [ODBC] 114, 170
- Execute メソッド 75, 81, 82, 84, 85, 270, 273, 274
- Execute メソッド [CommuniNet] 342, 365

Execute メソッド [ODBC] 115, 151, 172, 200

F

FieldCodeType プロパティ 76
FieldCodeType プロパティ [ODBC] 119
FieldCodeType プロパティで取得するデータ 78
FieldCodeType プロパティに設定した値とデータ取得の関係 78
FieldCount プロパティ 76, 271
FieldCount プロパティ [CommuniNet] 309
FieldCount プロパティ [ODBC] 120
FieldDataB プロパティ 76
FieldDataB プロパティ [ODBC] 121
FieldDataLength プロパティ 76, 271
FieldDataLength プロパティ [CommuniNet] 310
FieldDataLength プロパティ [ODBC] 122
FieldData プロパティ 76, 271
FieldData プロパティ [CommuniNet] 309
FieldData プロパティ [ODBC] 121
FieldHeader プロパティ 76, 271
FieldHeader プロパティ [CommuniNet] 311
FieldHeader プロパティ [ODBC] 123
FieldNameLength プロパティ [ODBC] 123
FieldType プロパティ 76
FieldType プロパティ [ODBC] 124
FileLoad メソッド [CommuniNet] 389
FileLoad メソッド [ODBC] 196
FunctionMode プロパティ [CommuniNet] 353
FunctionMode プロパティ [ODBC] 160

G

GetList メソッド [ODBC] 201
GetParam メソッド [ODBC] 201
Get メソッド 70, 72, 264, 265, 267, 268
Get メソッド [CommuniNet] 315, 331, 340
Get メソッド [ODBC] 129, 141, 149

H

HeaderOutput プロパティ 76, 271

HeaderOutput プロパティ [CommuniNet] 311
HeaderOutput プロパティ [ODBC] 125
HiRDB 35, 36
HiRDB ODBC Driver 35
HiRDB のエラーメッセージの見方 249
HOAPDBS/PC 35, 38, 259
HOAPDBS E2 34, 38, 259
HOAPDBS がサーバプログラムのとき 400

I

Inserts コレクションオブジェクトの詳細 [CommuniNet] 376
Inserts コレクションオブジェクトの詳細 [ODBC] 183
InsertValueB プロパティ [ODBC] 186
InsertValueB プロパティ 81
InsertValueCount プロパティ [CommuniNet] 380
InsertValueCount プロパティ [ODBC] 187
InsertValue プロパティ 81
InsertValue プロパティ [CommuniNet] 380
InsertValue プロパティ [ODBC] 186
Insert オブジェクト 81, 273
Insert オブジェクトの詳細 [CommuniNet] 379
Insert オブジェクトの詳細 [ODBC] 185
IsConnect プロパティ 91
IsConnect プロパティ [CommuniNet] 294
IsConnect プロパティ [ODBC] 108
Item メソッド [CommuniNet] 345, 350, 358, 368, 377, 383
Item メソッド [ODBC] 154, 158, 165, 175, 184, 191

K

KeyColumn プロパティ 82
KeyColumn プロパティ [CommuniNet] 371, 385
KeyColumn プロパティ [ODBC] 177, 192
KeyCount プロパティ 82
KeyCount プロパティ [CommuniNet] 371, 386
KeyCount プロパティ [ODBC] 177, 193

KeyValue プロパティ 82
KeyValue プロパティ [CommuniNet] 372, 386
KeyValue プロパティ [ODBC] 178, 193

L

LIKE 条件 28
ListPattern プロパティ [ODBC] 199
LogoffFileName プロパティ 258
LogoffFileName プロパティ [CommuniNet] 294
LogonFileName プロパティ 258
LogonFileName プロパティ [CommuniNet] 295
IsConnect プロパティ 65, 259

M

MacroTimeLimit プロパティ 258
MacroTimeLimit プロパティ [CommuniNet] 296

N

NCharPrefixMode プロパティ 81, 82
NCharPrefixMode プロパティ [ODBC] 170
NCHAR 定数モード [用語解説] 526
NULL 条件 28

O

ODBC 30
ODBC 関数のオプション値 243
ODBC 関数の概要 216
ODBC 関数の規則 239
ODBC 関数の非同期実行 250
ODBC 関数を使ったデータベースアクセス 215
ODBC 接続 34
ODBC 接続のときのプロパティとメソッドの制限値 500
ODBC 接続のときの列の属性 511
ODBC とデータベースとのデータ型の対応 223
ODBC ドライバ 35-37
ODBC ドライバのインストールの手順 45
ODBC ドライバの種類 30
ODBC ドライバのセットアップ 45

ODBC ドライバを使ってアクセスできるデータベース 35, 216
OR 28
Oracle 35, 36

P

ParamCodeType プロパティ [ODBC] 203
ParamComment プロパティ [ODBC] 205
ParamDataType プロパティ [ODBC] 205
ParamLength プロパティ [ODBC] 206
ParamList オブジェクトの詳細 [ODBC] 202
ParamName プロパティ [ODBC] 203
ParamNullable プロパティ [ODBC] 206
ParamPrecision プロパティ [ODBC] 207
ParamRadix プロパティ [ODBC] 207
ParamScale プロパティ [ODBC] 208
ParamTypeName プロパティ [ODBC] 208
ParamType プロパティ [ODBC] 204
ParamValueB プロパティ [ODBC] 210
ParamValueMaxLength プロパティ [ODBC] 211
ParamValue プロパティ [ODBC] 209
Password プロパティ 64, 258
Password プロパティ [CommuniNet] 296
Password プロパティ [ODBC] 109
Pattern プロパティ 70, 264, 265
Pattern プロパティ [CommuniNet] 320
Pattern プロパティ [ODBC] 134
PDMII E2 36, 38
PDMII E2 の場合の規則 434
Power Builder 5.0 29
ProcedureComment プロパティ [ODBC] 212
ProcedureList オブジェクトの詳細 212
ProcedureName プロパティ [ODBC] 213
ProcedureOwner プロパティ [ODBC] 213
ProcedureType プロパティ [ODBC] 214
Procedure オブジェクト 89
Procedure オブジェクトの詳細 [ODBC] 199

Q

- QueryCountLimit プロパティ 76, 271
- QueryCountLimit プロパティ [CommuniNet] 312
- QueryCountLimit プロパティ [ODBC] 125
- QueryCount メソッド [CommuniNet] 343
- QueryCount メソッド [ODBC] 151
- QueryDefine オブジェクト 74, 269
- QueryDefine オブジェクトの詳細 [CommuniNet] 341
- QueryDefine オブジェクトの詳細 [ODBC] 150

R

- RDA Link for Gateway 37
- RDB1 38
- RDB1 E2 35, 36
- RDB1 の場合の規則 439
- RDB 編成ファイル 35, 36, 38
- RDB 編成ファイルの場合の規則 425
- RecordData プロパティ 76, 271
- RecordData プロパティ [CommuniNet] 313
- RecordData プロパティ [ODBC] 126
- RecordHeader プロパティ 76, 271
- RecordHeader プロパティ [CommuniNet] 313
- RecordHeader プロパティ [ODBC] 127
- Relation プロパティ [CommuniNet] 356
- Relation プロパティ [ODBC] 163
- RemoveAll メソッド [CommuniNet] 343, 345, 350, 359, 366, 368, 377, 383
- RemoveAll メソッド [ODBC] 152, 154, 158, 166, 173, 175, 184, 191
- ResponseLimit プロパティ 258
- ResponseLimit プロパティ [CommuniNet] 297
- Result オブジェクト 76, 271
- Result オブジェクトの詳細 [CommuniNet] 307
- Result オブジェクトの詳細 [ODBC] 117
- Rollback メソッド 67, 261
- Rollback メソッド [CommuniNet] 305
- Rollback メソッド [ODBC] 113

S

- SaveLargeObject メソッド 77
- SaveLargeObject メソッド [ODBC] 129
- SaveToFile メソッド [CommuniNet] 316
- SaveToFile メソッド [ODBC] 130
- SelectDict プロパティ [CommuniNet] 320, 338
- SelectDict プロパティ [ODBC] 133, 148
- Setup.exe 45
- SortMode プロパティ [CommuniNet] 353
- SortMode プロパティ [ODBC] 161
- SortPriority プロパティ [CommuniNet] 354
- SortPriority プロパティ [ODBC] 162
- SQLDirect オブジェクト 84, 85
- SQLDirect オブジェクトの詳細 [ODBC] 114
- SQLGetInfo 233
- SQL Server 35, 36
- SQLText プロパティ 84, 85
- SQLText プロパティ [ODBC] 115
- SQL データ型 229
- SQL 文 [ODBC] 115
- SQL 文 [用語解説] 526
- SQL を直接記述してアクセス 84

T

- TableComment プロパティ 70, 71, 264, 265
- TableComment プロパティ [CommuniNet] 321
- TableComment プロパティ [ODBC] 135
- TableGroupName プロパティ 264, 265
- TableCreateDate プロパティ 70, 71, 264, 265
- TableCreateDate プロパティ [CommuniNet] 322
- TableCreateDate プロパティ [ODBC] 136
- TableDataSource プロパティ [CommuniNet] 322
- TableDataSource プロパティ [ODBC] 136
- TableDBMName プロパティ 265
- TableDBMName プロパティ [CommuniNet] 323
- TableDefSource プロパティ [CommuniNet] 293

TableDefSource プロパティ [ODBC] 108

TableDeleteAuth プロパティ 265

TableDeleteAuth プロパティ [CommuniNet] 324

TableGroupName プロパティ 70, 71

TableGroupName プロパティ [CommuniNet] 324

TableGroupName プロパティ [ODBC] 137

TableHeader プロパティ 70, 71, 264, 265

TableHeader プロパティ [CommuniNet] 325

TableHeader プロパティ [ODBC] 137

TableInsertAuth プロパティ 265

TableInsertAuth プロパティ [CommuniNet] 326

TableList オブジェクト 69, 263

TableList オブジェクトの詳細 [CommuniNet] 318

TableList オブジェクトの詳細 [ODBC] 132

Tablename プロパティ 72, 73

TableName プロパティ 70, 81, 82, 264, 265, 267, 273, 274

TableName プロパティ [CommuniNet] 326, 339, 347, 363

TableName プロパティ [ODBC] 138, 148, 156, 171

TableOwner プロパティ 70, 264, 265

TableOwner プロパティ [CommuniNet] 327

TableOwner プロパティ [ODBC] 139

TablePasswordMode プロパティ 265

TablePasswordMode プロパティ [CommuniNet] 328

TablePassword プロパティ 267, 273, 274

TablePassword プロパティ [CommuniNet] 339, 347, 364

TableSelectAuth プロパティ 265

TableSelectAuth プロパティ [CommuniNet] 329

Tables コレクションオブジェクトの詳細 [CommuniNet] 344

Tables コレクションオブジェクトの詳細 [ODBC] 153

TableType プロパティ 70, 264, 265

TableType プロパティ [CommuniNet] 329

TableType プロパティ [ODBC] 139

TableUpdateAuth プロパティ 265

TableUpdateAuth プロパティ [CommuniNet] 330

Table オブジェクトの詳細 [CommuniNet] 347

Table オブジェクトの詳細 [ODBC] 156

TargetDatabase プロパティ 258

TargetDatabase プロパティ [CommuniNet] 297

TargetServer プロパティ 258

TargetServer プロパティ [CommuniNet] 299

TerminalClose メソッド 260

TerminalClose メソッド [CommuniNet] 305

TerminalMode プロパティ 258

TerminalMode プロパティ [CommuniNet] 300

TerminalName プロパティ 258

TerminalName プロパティ [CommuniNet] 300

TEXT 形式 37, 39, 499

Text プロパティ [CommuniNet] 360

Text プロパティ [ODBC] 167

TXT 形式 77, 271

Type プロパティ [ODBC] 140

U

UpdateDefine オブジェクト 80, 272

UpdateDefine オブジェクトの詳細 [CommuniNet] 361

UpdateDefine オブジェクトの詳細 [ODBC] 168

UpdateMode プロパティ 81, 82, 273, 274

UpdateMode プロパティ [CommuniNet] 364

UpdateMode プロパティ [ODBC] 171

Updates コレクションオブジェクトの詳細 [CommuniNet] 367

Updates コレクションオブジェクトの詳細 [ODBC] 174

UpdateValueB プロパティ 81

UpdateValueB プロパティ [ODBC] 179

UpdateValueCount プロパティ [CommuniNet] 373

UpdateValueCount プロパティ [ODBC] 180

UpdateValue プロパティ 81

UpdateValue プロパティ [CommuniNet] 373
UpdateValue プロパティ [ODBC] 178
UpdateValue メソッド 273
Update オブジェクト 81, 273
Update オブジェクトの詳細 [CommuniNet] 370
Update オブジェクトの詳細 [ODBC] 176
UserID プロパティ 64, 258
UserID プロパティ [CommuniNet] 301
UserID プロパティ [ODBC] 109

V

VariableList オブジェクトの詳細 [CommuniNet] 390
VariableList オブジェクトの詳細 [ODBC] 197
VariableName プロパティ [CommuniNet] 390
VariableName プロパティ [ODBC] 197
VariableValue プロパティ [CommuniNet] 391
VariableValue プロパティ [ODBC] 198
Visual Basic 27
Visual Basic 4.0 (32 ビット版) 29
Visual Basic 5.0 29
Visual Basic 6.0 29
VOS1 スプールファイル 38
VOS1 スプールファイルの場合の規則 456
VOSK スプールファイル 38
VOSK スプールファイルの作成日付 [ODBC] 136
VOSK スプールファイルの場合の規則 457

X

XDM/DF 34
XDM/RD E2 35, 36, 38
XDM/RD E2 の場合の規則 419
XDM/SD E2 35, 36, 38
XDM/SD E2 の場合の規則 430
XLSB 形式 37, 39
XLSX 形式 37, 39
XLS 形式 37, 39, 77, 271, 499

あ

アクセスするデータベースの制限 39
アプリケーション 27
アプリケーションからトランザクションを制御するとき 67, 261
アプリケーションのサンプル 62, 256
アプリケーションを使ったデータベースアクセス (CommuniNet 接続) 253
アプリケーションを使ったデータベースアクセス (ODBC 接続) 59

い

インストール 45
インストールすると作成されるファイル 45
インストール時に起こるトラブル 460

え

エラーメッセージに付く ODBC 関数のプレフィックス 248

お

応答監視時間 51
オブジェクト 27, 60, 254
オブジェクトの階層 60, 254
オブジェクトの文法 (CommuniNet 接続) [CommuniNet] 286

か

外字の変換規則 399
カスタムインストール [用語解説] 526
稼働する環境 (CommuniNet 接続) 38
稼働する環境 (ODBC 接続) 36
環境の設定 40
環境を設定する手順 41

き

行数 [CommuniNet] 312
行数 [ODBC] 114, 125

く

- クイックインストール〔用語解説〕 526
- 空白文字に変換されるコード 399
- クライアント/サーバ〔用語解説〕 526
- クライアント PC 25
- クライアント PCに必要なソフトウェア 36, 38
- クライアント側プログラムでコード変換するときの注意 398
- クリップボードに出力〔CommuniNet〕 314
- クリップボードに出力〔ODBC〕 128

け

- 件数を制限する 33

こ

- 更新 33
- 更新した行数〔CommuniNet〕 363
- 更新した行数〔ODBC〕 170
- 更新実行モードとデータ更新の関係 82
- 更新種別〔CommuniNet〕 365
- 更新種別〔ODBC〕 172
- 更新種別〔用語解説〕 526
- 更新処理〔用語解説〕 526
- 更新する表名〔CommuniNet〕 363
- 更新する表名〔ODBC〕 171
- 更新版 43
- 更新表〔用語解説〕 526
- コミットモード〔CommuniNet〕 289
- コミットモード〔ODBC〕 102

さ

- サーバ 25
- サーバ側に必要なソフトウェア 36, 38
- サーバプログラム 26, 34, 36-38, 65, 259
- サーバプログラムと接続 111
- サーバプログラムと接続〔CommuniNet〕 303
- サーバプログラムとの接続状態〔CommuniNet〕 294
- サーバプログラムとの接続状態〔ODBC〕 108

- サーバプログラムとの接続を解除 112
- サーバプログラムとの接続を解除〔CommuniNet〕 304
- サーバプログラム〔用語解説〕 527
- サーバ〔用語解説〕 527
- サーバを接続する形態 34
- 削除 33
- 削除行条件〔用語解説〕 527
- 削除処理〔用語解説〕 527
- 避けて欲しい処理 91, 277
- 参照版 43
- サンプル 29, 62, 256

し

- システムに関するトラブル 461
- 指定した件数までのデータの抽出 28
- 自動的にトランザクションを制御するとき 67, 261
- 絞り込み 32
- 絞り込み条件〔CommuniNet〕 320
- 絞り込み条件〔ODBC〕 134
- 絞り込み条件〔用語解説〕 529
- 集合関数〔CommuniNet〕 353
- 集合関数〔ODBC〕 160
- 集合関数〔用語解説〕 527
- 取得できる ODBC 関数のオプション値 243
- 条件関連〔ODBC〕 163
- 条件関連〔用語解説〕 527
- 条件式〔CommuniNet〕 357, 360
- 条件式〔ODBC〕 164, 167
- 条件式を設定するときの規則 393
- 条件に合うデータを抽出する 32
- 条件〔用語解説〕 527
- 条件を設定したデータの抽出 28
- 所有者〔用語解説〕 527

す

- ストアードプロシジャの実行 89
- スプールファイル 36

せ

- 制限値 500, 505
- 製品の種類 43
- 製品の種類と使える機能の関係 43
- 接続先〔ODBC〕 102

そ

- ソートモード〔用語解説〕 528
- ソート優先順位〔用語解説〕 528
- 属性〔用語解説〕 528

ち

- 抽出 28, 31
- 抽出条件〔用語解説〕 528
- 抽出処理〔用語解説〕 528
- 抽出するデータの形式 77, 271
- 抽出データ〔CommuniNet〕 313
- 抽出データ〔ODBC〕 126
- 抽出データ格納ファイル〔用語解説〕 528
- 抽出データの管理 76, 271
- 抽出で使える機能の一覧 32
- 抽出列の個数〔CommuniNet〕 349
- 抽出列の個数〔ODBC〕 157
- 抽出列の優先順位〔CommuniNet〕 354
- 抽出列の優先順位〔ODBC〕 162
- 抽出列の列名〔CommuniNet〕 349
- 抽出列の列名〔ODBC〕 157
- 抽出列〔用語解説〕 528

つ

- 追加 33
- 追加情報〔用語解説〕 528
- 追加処理〔用語解説〕 528
- 使えるデータベースと表の数 37, 39

て

- 定義版 43
- データ型 229
- データ受信用領域サイズ 49

- データソース 48
- データソースのセットアップ 47
- データソース名 64, 106
- データソース名の削除 52
- データソース名の変更 52
- データソース名〔用語解説〕 528
- データの形式 37, 39, 499
- データの更新 33, 80
- データの更新 (SQLDirect オブジェクト) 85
- データの更新 (UpdateDefine オブジェクト) 272
- データの更新〔用語解説〕 529
- データの削除 29
- データの絞り込み〔用語解説〕 529
- データの抽出 31
- データの抽出 (Catalog オブジェクト) 87
- データの抽出 (QueryDefine オブジェクト) 74
- データの抽出 (SQLDirect オブジェクト) 84
- データの抽出 (Catalog オブジェクト) 275
- データの抽出 (QueryDefine オブジェクト) 269
- データの抽出〔用語解説〕 529
- データの追加 29
- データの転送〔用語解説〕 529
- データの変更 29
- データベース 25, 36-38
- データベースアクセスの概要 27
- データベースアクセスの手順 63, 257
- データベース識別子 49
- データベースと接続するときにかかるトラブル 460
- データベースとの接続の解除 66, 260
- データベースとの接続の確認 65, 259
- データベースとのデータ型の対応 223
- データベース別の規則 419
- データベースへのアクセスのキャンセル 65, 259
- データベースへの接続 65, 259
- データベースへの接続と接続の解除 65, 259
- データベース〔用語解説〕 529
- データを更新 (削除) 82, 274
- データを更新 (追加) 81, 273
- データを更新 (変更) 80, 273

データを更新する手順 80

デッドロック 68, 262

と

特長 27

特定の列だけを抽出する 32

トップオブジェクト 64, 257

トラブルが起こったときは 459

トラブルの現象と対処 460

トランザクション 67, 261

トランザクションと排他制御 67, 261

トランザクションの終了モード 64, 258

トランザクションの制御の設定 67, 261

トランザクション〔用語解説〕 529

は

排他エラーになったときの動作の設定 68, 262

排他オプション 151, 342

排他オプション〔CommuniNet〕 362

排他オプション〔ODBC〕 169

排他オプション〔用語解説〕 529

排他制御 67, 261

排他制御の設定 67, 261

配列列を設定するときの規則 396

パスワード 64, 258

パスワード〔CommuniNet〕 289, 296

パスワード〔ODBC〕 103, 109

ひ

比較条件 28

ビジュアル開発ツール 27, 29

非同期実行 250

表結合 32

表情情報の個数〔CommuniNet〕 319

表情情報の個数〔ODBC〕 132

表情情報の最大数〔CommuniNet〕 319

表情情報の最大数〔ODBC〕 133

表情情報の取得 69, 263

表情情報〔用語解説〕 529

表と列の情報の取得 69, 263

表の個数〔CommuniNet〕 344

表の個数〔ODBC〕 153

表の所有者〔CommuniNet〕 327

表の所有者〔ODBC〕 139

表の属性〔CommuniNet〕 329

表の属性〔ODBC〕 139, 140

表の注釈〔CommuniNet〕 321

表の注釈〔ODBC〕 135

表のラベル〔CommuniNet〕 325

表のラベル〔ODBC〕 137

表番号〔用語解説〕 530

表名〔CommuniNet〕 326, 339

表名〔ODBC〕 138, 148

表名と列名を検索 28

表名〔用語解説〕 530

表名を設定するときの規則 394

表〔用語解説〕 529

ふ

ファイルに出力〔CommuniNet〕 316

ファイルに出力〔ODBC〕 130

複数のデータベースにアクセスするとき 65

複数の表を結合して抽出する 32

プレフィックス〔ODBC〕 170

プロパティとメソッドの実行順序 93, 279

プロパティとメソッドの制限 99, 284

プロパティとメソッドの制限値 500

プロパティとメソッドの表記 61, 255

プロパティページの設定〔CommuniNet〕 287

プロパティページの設定〔ODBC〕 102

へ

ヘッダ〔CommuniNet〕 311

ヘッダ〔ODBC〕 123

ヘッダ出力モード〔用語解説〕 530

ヘッダ〔用語解説〕 530

変更 33

変更行条件〔用語解説〕 530

変更情報〔用語解説〕 530

変更処理〔用語解説〕 530

ほ

ホスト名称 48

ま

マクロスクリプト 54

マクロスクリプトの設定 (CommuniNet 接続の場合) 54

マクロスクリプトファイルの例 54

マルチメディアデータ 29

め

メインフレーム 25

メソッドの実行順序 93, 279

メソッドの制限値 500, 505

メッセージの一覧 462

メッセージの形式 463

メッセージの詳細 465

も

文字定数の値 511, 512

ゆ

ユーザ ID 64, 109, 258

ユーザ ID [CommuniNet] 289, 301

ユーザ ID [ODBC] 103

ユーザ ID とパスワードの入力を要求するダイアログボックス 64

ユーザ外字変換ファイルの作成方法 52

よ

予約語 497

れ

列一覧〔用語解説〕 530

列情報の個数 [CommuniNet] 337

列情報の個数 [ODBC] 146

列情報の最大数 [CommuniNet] 338

列情報の取得 71, 266

列情報〔用語解説〕 530

列数 [CommuniNet] 309

列数 [ODBC] 120

列属性とデータベースの属性 513

列の属性 511

列の属性 [CommuniNet] 337

列の属性 [ODBC] 124, 146

列の注釈 [CommuniNet] 332

列の注釈 [ODBC] 142

列の定義長 [CommuniNet] 334

列の定義長 [ODBC] 144

列のデータ [CommuniNet] 309

列のデータ [ODBC] 121

列のデータ長 [CommuniNet] 310

列のデータ長 [ODBC] 122

列のラベル [CommuniNet] 334

列のラベル [ODBC] 143

列の列名長 [ODBC] 123, 145

列番号 [CommuniNet] 336

列番号〔用語解説〕 530

列名 [CommuniNet] 313, 335

列名 [ODBC] 127, 144

列名〔用語解説〕 530

列名を設定するときの規則 395

ろ

ログイン画面 [ODBC] 102

ログオフスクリプト [CommuniNet] 288

ログオンスクリプト [CommuniNet] 287

わ

ワイルドカード〔用語解説〕 531