

KVCOM プロバイダ  
KEYENCE プログラマブルコントローラ  
KV シリーズ用プロバイダ  
Version 1.0.0

ユーザーズ ガイド

Oct 04, 2016

【備考】

**【改版履歴】**

バージョン	日付	内容
1.0.0	2016-10-04	初版.

## 目次

1. はじめに.....	4
1.1. PC 環境のセットアップ.....	4
2. プロバイダの概要.....	5
3. リファレンス.....	6
3.1. PLC 制御.....	6
3.1.1. CaoWorkspace::AddController メソッド.....	6
3.1.2. CaoController::AddVariable メソッド.....	8
3.1.2.1. VT オプション.....	10
3.1.3. CaoVariable:put_Value プロパティ.....	10
3.1.4. CaoVariable:get_Value プロパティ.....	10
3.1.5. CaoController::Execute("ReadComment")メソッド.....	11
3.1.6. CaoController::Execute("ClearError")メソッド.....	11
3.1.7. CaoController::Execute("SetSystemTime")メソッド.....	12
3.1.8. CaoController::Execute("GetAlarmLog")メソッド.....	13
3.1.9. CaoController::get_VariableNames プロパティ.....	13
3.1.10. システム変数一覧.....	14
3.2. メモリーカードアクセス.....	15
3.2.1. CaoController::Execute("MemoryCardFileCopyTo") メソッド.....	15
3.2.2. CaoController::Execute("MemoryCardFileCopyFrom") メソッド.....	16
3.2.3. CaoController::AddFile メソッド.....	16
3.2.4. CaoFile::AddFile メソッド.....	17
3.2.5. CaoController::get_FileNames プロパティ.....	17
3.2.6. CaoFile::get_FileNames プロパティ.....	17
3.2.7. CaoFile::get_Path プロパティ.....	17
3.2.8. CaoFile::Delete メソッド.....	18
3.2.9. CaoFile::get_VariableNames プロパティ.....	18
3.2.10. システム変数一覧.....	19
3.3. エラーコード.....	21
付録 A. 接続パラメータ作成方法.....	22

## 1. はじめに

本書は、KEYENCE 製 PLC(プログラマブルコントローラ)に対しデータの書込み/読出しを行う CAO プロバイダのユーザーズガイドです。本書で扱う CAO プロバイダ(CaoProvKEYENCEKVCOM.dll)を KVCOM プロバイダと呼びます。

KVCOM プロバイダは、KV COM+ Library を使用して KEYENCE 製 PLC と通信を行うため、KEYENCE 製ライブラリ KV COM+ Library を別途インストールする必要があります。KVCOM プロバイダで対応している PLC 及び接続方法については KV COM+ Library のマニュアルを参照してください。

本プロバイダは下記に示すバージョンの KV COM+ Library を使用して開発を行っています。バージョンが異なる場合プロバイダが正しく機能しない可能性がありますので、同じバージョンのソフトウェアをインストールしてご使用ください。

KV COM+ Library Ver.1.3

### 1.1. PC 環境のセットアップ

KVCOM プロバイダを使用するためには表 1-1 の手順で PC 環境のセットアップを行ってください。

表 1-1PC 環境のセットアップ手順

手順	詳細
(1)KV COM+ Library のインストール	KV COM+ Library ユーザーズマニュアル 「1-3 ソフトウェアのインストール」を参照してください。
(2)環境変数(Path)の設定	KV COM+ Library ユーザーズマニュアル 「5-1 DLL のインストール」を参照してください。
(3)KVCOM プロバイダの登録	本書「2 プロバイダの概要」を参照してください。

## 2. プロバイダの概要

KVCOM プロバイダは, KEYENCE 製 PLC に対し KV COM+ Library を使用してデータの書き込み/読出しを行う CAO プロバイダです. そのファイル形式は DLL(Dynamic Link Library)であり, CAO エンジンから使用時に動的にロードされます. プロバイダを使用するには 1.1 PC 環境のセットアップにて示す手順(1)(2)を実施後, ORiN2SDK をインストールするか下表を参照してレジストリ登録を行う必要があります.

表 2-1 KVCOM プロバイダ

ファイル名	CaoProvKEYENCEKVCOM.dll
ProgID	CaoProv.KEYENCE.KVCOM
レジストリ登録	regsvr32 CaoProvKEYENCEKVCOM.dll
レジストリ登録の抹消	regsvr32 /u CaoProvKEYENCEKVCOM.dll

## 3. リファレンス

### 3.1. PLC 制御

#### 3.1.1. CaoWorkspace::AddController メソッド

PLC に接続してデータの書込み/読出しを行うコントローラオブジェクトを作成します。オプション文字列で PLC への接続方法を指定します。

書式	<code>&lt;objRet:CaoController&gt; = AddController(&lt;bstrCtrlName:BSTR&gt;,&lt;bstrProvName:BSTR&gt;,&lt;bstrPCName:BSTR&gt;,&lt;bstrOption:BSTR&gt;))</code>	
戻り値	<code>&lt;objRet&gt;</code>	: [out] コントローラオブジェクト
引数	<code>&lt;bstrCtrlName&gt;</code>	: [in] コントローラ名(任意)
	<code>&lt;bstrProvName &gt;</code>	: [in] プロバイダ名. 固定値 ="CaoProv.KEYENCE.KVCOM" を指定する
	<code>&lt;bstrPCName &gt;</code>	: [in] プロバイダの実行マシン名
	<code>&lt;bstrOption&gt;</code>	: [in] オプション文字列(表 3-1 を参照)

表 3-1 にオプション文字列に指定するリストを示します。

**表 3-1 CaoWorkspace::AddController のオプション文字列**

オプション	説明
<code>PlcId=&lt;PLCID&gt;</code>	<p>接続機種を示す数値. DBPlcDef.h で定義された列挙型 DBPlcId の数値を 10 進数で指定してください.</p> <p>DBPlcDef.h は KVCOM+ Library のインストールフォルダ内の「dlluser¥include」に配置されています.</p>
<code>DestName=&lt;接続パラメータ&gt;</code>	<p>PLC への接続方法を示す文字列.</p> <p>詳細は KVCOM+ Library のユーザーズマニュアル「5-2 DLL 関数」-「3 コネクション制御」にて示されている lpszDestName の内容を参照してください.</p> <p>接続パラメータ自身にカンマ記号が含まれる場合は不等号記号で接続パラメータを囲ってください. (使用例 2 を参照)</p> <p>GUI から接続パラメータを作成する方法は付録 A を参照してください.</p>



**使用例 1** PC から USB で KV5500 に接続する場合

```
Dim caoEng as CaoEngine
Dim caoCtrl as CaoController

Set caoEng = New caoEngine
Set caoCtrl = caoEng.Workspaces(0).AddController("KV5500", "CaoProv. KEYENCE. KVCOM", "",
"PlcId=4611, DestName=USB")
```

**使用例 2** PC から USB で接続した PLC を経由し、Ethernet/IP ネットワークを使用して接続する場合

```
Dim caoEng as CaoEngine
Dim caoCtrl as CaoController

Set caoEng = New caoEngine
Set caoCtrl = caoEng.Workspaces(0).AddController("KV5500", "CaoProv. KEYENCE. KVCOM", "",
"PlcId=4611, DestName=<USB:Via, 13, 2, 4, 0, 4, 1, 0, 1, 0, 0, 192. 168. 0. 10, 53>")
```

**3.1.2. CaoController::AddVariable メソッド**

CaoController クラスの AddVariable メソッドは、PLC のデバイスに対しデータの書き込み/読出しを行うための変数オブジェクトを作成します。作成した変数オブジェクトの Value プロパティに対して書き込み/読出しを行うことで、デバイスのデータの書き込み/読出しを実行できます。

表 3-2 オプション文字列に示す書式の文字列を指定することで、アクセス対象とする PLC 内のデバイスのデバイス種別/デバイス番号/データ型/データ数を決定します。

**書式** <objRet:CaoVariable> = AddVariable(<bstrVariableName:VT\_BSTR>  
,<bstrOption:VT\_BSTR>)

戻り値 <objRet> : [out] 変数オブジェクト  
 引数 <bstrVariableName> : [in] 変数名(任意)  
 <bstrOption> : [in] オプション文字列(表 3-2 を参照)

**使用例**

```
' DM00000 - DM00015 にアクセスする CaoVariable オブジェクトを作成
Set caoVar = caoCtrl.AddVariable("DM", "DeviceKind=18, DeviceNo=0, Elem=4")

' 読出しを実行
vVal = caoVar.Value
' 書き込みを実行
caoVar.Value = vVal
```

表 3-2 にオプション文字列に指定するリストを示します。

表 3-2 CaoController::AddVariable のオプション文字列

オプション	説明
DeviceKind=<デバイス種別>	必須。 デバイス種別の値を指定してください。値は 10 進数, 16 進数表記で指定可能です。 値の詳細は KV COM+ Library ユーザーズマニュアル「4-2 プロパティ・メソッド・イベント詳細」 - 「DBCommManager オブジェクト」 - 「ReadDevice」を参照してください。
DeviceNo=<デバイス番号>	必須。 アクセスする先頭のデバイス番号の値を指定してください。値は 10 進数, 16 進数表記で指定可能です。 詳細は KV COM+ Library ユーザーズマニュアル「4-2 プロパティ・メソッド・イベント詳細」 - 「DBCommManager オブジェクト」 - 「ReadDevice」を参照してください。
VT=<変数型>	省略可能。 Put/Get するデータ型を指定してください。詳細は 3.1.2.1 を参照してください。 省略した時, デバイス種別がビットの場合「BIT」, デバイス種別がワードの場合「I2」となります。
Elem=<要素数>	省略可能。 Put/Get するデータ数を指定してください。 要素数は 10 進数または 16 進数で指定可能です。 「VT=BSTR」の場合は文字列のバイト数(Null 終端文字を含む)となり, 最大値は 256 になります。 Elem を省略した時「VT=BSTR」以外の場合は 1, 「VT=BSTR」の場合は 256 になります。
Array=<True or False>	省略可能。 Put/Get するデータが 1 要素の場合に配列として扱うかを指定してください。 (デフォルト:False)

VT オプションと Elem オプションの指定により、一度に書き込み/読出しを行うデータ量を指定することが出来ます。データ量が DeviceKind オプションで指定するデバイス種別のデータ量を超える場合は、DeviceNo オプションで指定するデバイス番号から連続したデバイスに対して書き込み/読出しを行います。デバイスに対して書き込み/読出したデータはリトルエンディアン形式(下位ビットから順にセット)で格納するものとします。

一度に読出し/書き込みするデバイス数が 8192 個を超える場合はデータ書き込み/読出し時にエラーが発生します。

### 3.1.2.1. VT オプション

Value プロパティで Put/Get するデータ型を指定します。指定可能なデータ型の一覧を表 3-3 に示します。

表 3-3 VT オプションで指定可能なデータ型一覧

VT の文字列	データ型	説明
BIT	VT_UI1	データを 0/1 の 1 ビットデータとして書き込み/読出し。
BOOL	VT_BOOL	データを True/False の 1 ビットデータとして書き込み/読出し。
BSTR	VT_BSTR	BSTR を ASCII 文字列 (Null 終端) に変換して書き込み/読出し。 デバイス種別がワードの場合のみ指定可能。
I1	VT_I1	1 バイトデータとして書き込み/読出し。
I2	VT_I2	2 バイトデータとして書き込み/読出し。
I4	VT_I4	4 バイトデータとして書き込み/読出し。
I8	VT_I8	8 バイトデータとして書き込み/読出し。
UI1	VT_UI1	1 バイトデータとして書き込み/読出し。
UI2	VT_UI2	2 バイトデータとして書き込み/読出し。
UI4	VT_UI4	4 バイトデータとして書き込み/読出し。
UI8	VT_UI8	8 バイトデータとして書き込み/読出し。
R4	VT_R4	4 バイトデータとして書き込み/読出し。
R8	VT_R8	8 バイトデータとして書き込み/読出し。

### 3.1.3. CaoVariable:put\_Value プロパティ

引渡された値を AddVariable のオプション指定に従って変換し、デバイスに対して書き込みを行います。引渡された値がオプション指定のデータ型に変換できない場合、またはオプション指定の要素数に満たない場合はエラーが発生します。

### 3.1.4. CaoVariable:get\_Value プロパティ

デバイスに対して読出しを行い、AddVariable のオプション指定に従ったデータ形式に変換して返します。

### 3.1.5. GaoController::Execute("ReadComment")メソッド

PLC 内のデバイスからコメントを取得します。

#### 書式

```
<bstrRet:VT_VBSTR> = ReadComment(<iDeviceKind:VT_I4>, <iDeviceNo:VT_I4>
                                     [, <iCommnetNo:VT_I4>])
```

戻り値 <bstrRet> : [out] 取得したデバイスのコメント。

引数 <iDeviceKind> : [in] デバイス種別。

詳細は KV COM+ Library ユーザーズマニュアル  
「4-2 プロパティ・メソッド・イベント詳細」 -  
「DBCommManager オブジェクト」 -  
「ReadDevice」を参照してください。

<iDeviceNo> : [in] デバイス番号。

<iCommnetNo> : [in] コメント番号(デフォルト値:1)

KV-7500/7300 の場合のみコメント番号を 1~8 の  
値で指定してください。

#### 使用例

```
bstrRet = caoCtrl.Execute("ReadComment", Array(18, 10000))
```

### 3.1.6. GaoController::Execute("ClearError")メソッド

PLC 内で発生しているエラーを 1 件クリアします。

#### 書式

```
ClearError()
```

戻り値 : なし

引数 : なし

#### 使用例

```
caoCtrl.Execute "ClearError"
```

### 3.1.7. CaoController::Execute("SetSystemTime")メソッド

PLC の内蔵時計を設定します。

#### 書式

SetSystemTime(<iTime:VT\_ARRAY | VT\_I4>)]

戻り値 : なし

引数 <iTime> : [in] 設定する時刻情報の配列.

[0] 年

[1] 月

[2] 日

[3] 時

[4] 分

[5] 秒

#### 使用例

---

```
caoCtrl.Execute "SetSystemTime", Array(2016, 10, 4, 12, 0, 0)
```

---

### 3.1.8. CaoController::Execute("GetAlarmLog")メソッド

PLC からアラーム履歴を取得します。

#### 書式

<vntRet:VT\_VARIANT | VT\_ARRAY> = GetAlarmLog()

戻り値 <vntRet> : [out] 取得したアラーム履歴. 次に示す配列を 1 件分のデータとし, 取得した件数分の配列で返します. アラーム履歴が 0 件の場合は VT\_EMPTY を返します.

[0] リレー番号(VT\_I4)

[1] True:立ち上がり, False:立ち下がり(VT\_BOOL)

[2] 時刻(VT\_ARRAY | VT\_I4)

[0] 年

[1] 月

[2] 日

[3] 時

[4] 分

[5] 秒

引数 : なし

#### 使用例

```
vntRet = caoCtrl.Execute("GetAlarmLog")
```

### 3.1.9. CaoController::get\_VariableNames プロパティ

CaoController::AddVariable で指定できるシステム変数名の一覧を返します. システム変数名の詳細は表 3-4 および表 3-5 を参照してください.

#### 書式

<bstrRet:VT\_BSTR | VT\_ARRAY> = VariableNames()

戻り値 <bstrRet> : [out] システム変数名の一覧

引数 : なし

#### 使用例

```
bstrRet = caoCtrl.VariableNames
```

## 3.1.10. システム変数一覧

表 3-4 GaoController クラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@TYPE	VT_BSTR	接続した PLC の機種名を返します。 詳細は KV COM+ Library ユーザーズマニュアル 「5-2 DLL 関数」 - 「DBQueryType」を参照してください。	○	—
@MODE	VT_I4	接続した PLC の動作モード。 1:RUN モード, 2:PROG モード	○	○
@ERROR_NUMBER	VT_UI1	PLC で現在発生しているエラー番号。 エラーが発生していないときは 0 を返します。 詳細は PLC のユーザーズマニュアルを参照してください。	○	—
@LIBRARY_ERROR	VT_I4	各メソッド、プロパティ、変数の呼び出しにて KV COM+ Library エラー(HRESULT = 0x80100000) が発生した場合、この変数に KV COM+ Library の エラー番号が格納されます。 エラー番号の詳細は KV COM+ Library ユーザーズマ ニュアル「付録 5 エラーメッセージ一覧」の「DLL」を参 照してください。 この変数に「0」代入することで値をクリアできます。	○	○

## 3.2. メモリーカードアクセス

### 3.2.1. GaoController::Execute("MemoryCardFileCopyTo") メソッド

PC から PLC に挿入しているメモリーカードにファイルをコピーします。  
コピーできるファイルサイズは最大 256MB です。

#### 書式

MemoryCardFileCopyTo(<bstrPathFrom:VT\_BSTR>, <bstrPathTo:VT\_BSTR>  
[,<bstrOption:VT\_BSTR>])

戻り値 : なし

引数 <bstrPathFrom> : [in] PC 上のコピー元となるファイルのパスを指定してください。

<bstrPathTo> : [in] メモリーカード上のコピー先となるファイルのパスを指定してください。

<bstrOption> : [in] "OverWrite"を指定した時、コピー先に同名のファイルがあった場合に上書きします。  
指定がない時、コピー先に同名のファイルがあった場合はエラーが発生します。

#### 使用例

---

```
caoCtrl.Execute "MemoryCardFileCopyTo", Array("C:¥Temp¥data1.txt", "data1.txt", "OverWrite")
```

---

### 3.2.2. CaoController::Execute("MemoryCardFileCopyFrom") メソッド

PLC に挿入しているメモリーカードから PC にファイルをコピーします。  
コピーできるファイルサイズは最大 256MB です。

#### 書式

MemoryCardFileCopyFrom(<bstrPathFrom:VT\_BSTR>, <bstrPathTo:VT\_BSTR>  
[,<bstrOption:VT\_BSTR>])

戻り値 : なし

引数 <bstrPathFrom> : [in] メモリーカード上のコピー元となるファイルのパスを指定してください。

<bstrPathTo> : [in] PC 上のコピー先となるファイルのパスを指定してください。

<bstrOption> : [in] "OverWrite"を指定した時、コピー先に同名のファイルがあった場合に上書きします。  
指定がない時、コピー先に同名のファイルがあった場合はエラーが発生します。

#### 使用例

```
caoCtrl.Execute "MemoryCardFileCopyFrom", Array("data1.txt", "C:¥Temp¥data1.txt", "OverWrite")
```

### 3.2.3. CaoController::AddFile メソッド

PLC に挿入しているメモリーカード内のファイルまたはディレクトリにアクセスする CaoFile オブジェクトを作成します。

#### 書式

<objRet:CaoFile> = AddFile(<bstrName:VT\_BSTR>[, <bstrOption:VT\_BSTR>])

戻り値 <objRet> : なし

引数 <bstrName> : [in] メモリーカード上のアクセスしたいファイル名またはディレクトリ名のパスを指定してください。  
( (IP アドレス)/MMC 以降を指定してください。 )  
ディレクトリの場合は終端に"¥"記号を付加してください。

<bstrOption> : [in] "@Create=2"を指定した場合、bstrName に指定したディレクトリを作成します。

**使用例**


---

```
Set caoFl = caoCtrl.AddVariable("data1.txt")
```

---

**3.2.4. CaoFile::AddFile メソッド**

前述 3.2.3 と同様にファイルオブジェクトを作成します。bstrName には、このメソッドを実行する CaoFile オブジェクトに対応しているディレクトリ内のファイルまたはディレクトリのみを指定することができます。このメソッドは実行する CaoFile オブジェクトがディレクトリに対応していないときはエラーが発生します。

**3.2.5. CaoController::get\_FileNames プロパティ**

PLC に挿入しているメモリーカード内のファイル名/ディレクトリ名の一覧を取得します。

**書式**

```
<bstrRet:VT_BSTR | VT_ARRAY> = FileNames([<bstrOption:VT_BSTR>])
```

戻り値 <bstrRet> : [out] 取得したファイル名/ディレクトリ名の配列。

ディレクトリ名の終端には"¥"記号が付きます。

引数 <bstrOption> : [in] 一覧を取得したいディレクトリのパスを指定します。省略した場合はルートディレクトリの一覧を取得します。

**使用例**


---

```
bstrRet = caoCtrl.FileNames
```

---

**3.2.6. CaoFile::get\_FileNames プロパティ**

前述 3.2.5 と同様にメモリーカード内のファイル名/ディレクトリ名の一覧を取得します。CaoFile オブジェクトと対応しているディレクトリ内の一覧を取得します。bstrOption でディレクトリを指定することはできません。このメソッドは実行する CaoFile オブジェクトがディレクトリに対応していないときはエラーが発生します。

**3.2.7. CaoFile::get\_Path プロパティ**

CaoFile オブジェクトと対応しているファイルまたはディレクトリのパスを取得します。

**書式**

```
<bstrRet:VT_BSTR> = Path
```

戻り値 <bstrRet> : [out] CaoFile オブジェクトに対応するファイルまたはディレクトリのパス。

引数 : なし。

**使用例**


---

```
bstrRet = caoFl.Path
```

---

**3.2.8. CaoFile::Delete メソッド**

CaoFile オブジェクトと対応しているファイルまたはディレクトリを削除します。

**書式**

Delete([<bstrOption:VT\_BSTR])

戻り値 : なし。

引数 : [in] 対象がファイルの場合, "Force"を指定すると読み取

り専用属性のファイルを削除できます。

対象がディレクトリの場合, "Force"を指定すると,

ディレクトリ内が空でない時でも削除できます。

オプション無しで読み取り専用ファイル, または空

で

ないディレクトリを削除しようとしたときはエラーとなります。

**使用例**


---

```
caoFl.Delete "Force"
```

---

**3.2.9. CaoFile::get\_VariableNames プロパティ**

CaoFile::AddVariable で指定できるシステム変数名の一覧を返します。システム変数名の詳細は表 3-6 を参照してください。

**書式**

<bstrRet:VT\_BSTR | VT\_ARRAY> = VariableNames()

戻り値 <bstrRet> : [out] システム変数名の一覧

引数 : なし

**使用例**


---

```
bstrRet = caoFl.VariableNames
```

---

## 3.2.10. システム変数一覧

表 3-5 GaoController クラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@MEMORYCARD_SPACE	VT_I4   VT_ARRAY	メモリーカードの総容量, 空き容量を Byte 単位で返します. 64bit 整数値を 32bit ごとに分割し, 下記の配列で返します. [0] 総容量 上位 32bit [1] 総容量 下位 32bit [2] 空き容量 上位 32bit [3] 空き容量 下位 32bit	○	—

表 3-6 CaoFile クラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@MEMORYCARD_ FILE_STATE	VT_VARIANT   VT_ARRAY	<p>ファイルまたはディレクトリの状態を下記の配列で返します.</p> <p>[0] 最終アクセス日時(VT_ARRAY   VT_I4)</p> <p>[0] 年 [1] 月 [2] 日 [3] 時 [4] 分 [5] 秒</p> <p>[1] 作成日時(VT_ARRAY   VT_I4)</p> <p>[0] 年 [1] 月 [2] 日 [3] 時 [4] 分 [5] 秒</p> <p>[2] 最終更新日時(VT_ARRAY   VT_I4)</p> <p>[0] 年 [1] 月 [2] 日 [3] 時 [4] 分 [5] 秒</p> <p>[3] ファイルサイズ[Byte] (VT_I4)</p> <p>[4] 属性(VT_I4 0:ファイル 1:ディレクトリ)</p>	○	—

### 3.3. エラーコード

KVCOM プロバイダでは, 表 3-7 に示す固有エラーコードが定義されています.

ORiN2 共通エラーについては, 「[ORiN2 プログラミングガイド](#)」のエラーコードの章を参照してください.

表 3-7 固有エラーコード

エラー名	エラー番号	説明
KV COM+ Library エラー	0x80100000	KV COM+ Library の API 呼び出しにてエラーが発生した場合に返されます. このエラーが発生した場合は CaoController クラスシステム変数「@LIBRARY_ERROR」を使用して詳細なエラー情報を取得してください.
文字コード変換エラー	0x80100001	文字コード変換 (ASCII⇔UNICODE) を行うときに変換できない文字がある場合に返されます.

## 付録A. 接続パラメータ作成方法

AddController に必要となる接続パラメータ”DestName”を GUI から作成する方法を紹介します。

- 手順 1 「Microsoft Excel」を起動します。
- 手順 2 「Alt + F11」キーを押下します。
- 手順 3 「Microsoft Visual Basic for Applications」ウィンドウのメニューから「挿入(I)」→「ユーザーフォーム(U)」を選択します。
- 手順 4 「ツール(T)」→「その他のコントロール(A)」を選択します。
- 手順 5 「コントロールの追加」ウィンドウの「利用可能なコントロール」のリストから「DBComm Manager Class(Library)」にチェックを付けて、「OK」ボタンをクリックします。
- 手順 6 「ツールボックス」ウィンドウから「DBComm Manager」を選択して「UserForm1」に貼り付けます。
- 手順 7 「プロパティ」ウィンドウから「(プロパティページ)」を選択後、「...」ボタンをクリックします。
- 手順 8 「プロパティページ」ウィンドウの「設定」ボタンをクリックします。
- 手順 9 「通信設定」ウィンドウにて通信方法を設定し、「OK」ボタンをクリックします。
- 手順 10 「プロパティページ」のテキストボックスの文字列を接続パラメータとして使用します。