BALLUFF
IO-Link プロバイダ

Version 1.1.0

ユーザーズ ガイド

January 23, 2019

備考：

【改版履歴】

バージョン	日付	内容
1.0.0	2016-08-30	初版.
1.0.2	2016-11-16	Extension 変数に対する単位取得変数の説明を追記しました.
1.0.3	2017-06-01	プロセスデータの書き込み及びパラメータデータの読み書きに対応しました.
1.0.4	2017-10-12	値の取得と設定にデータ型の指定を対応しました
1.1.0	2019-01-23	ArrayT 型に対応しました. ProcessDataInUnionT 型に対応しました. ProcessDataOutUnionT 型の対応しました. 対応スキーマバージョンを明記しました. XML ファイルの読み込みに関する不具合を修正しました.

【接続確認済み機器】

■ IO-Link マスター

機種名	ファームウェアバージョン	注意事項
BNI EIP-507-005-Z040	4.2.1	*特注バージョンでのチェック

※BALLUFF 社製の API の関係から，接続する IO-Link マスターのファームウェアが 4.3.0 以下のものは接続できない，または，データの通信が上手くいかない可能性があります.

目次

1. はじめに

本書は，BALLUFF 社製の IO-Link マスター(以下，マスター)に対してデータの読み込み，及び書き込みを行なう，BALLUFF IO-Link プロバイダ(以下，IO-Link プロバイダ)のユーザーズガイドです.

IO-Link プロバイダを用いれば，マスター及び，IO-Link デバイス(以下，デバイス)からのデータの取得，書き込みを行うことが容易になります.

本書は，IO-Link プロバイダの機能と，実装されているメソッドについて説明します.

IO-Link 規格の詳細については，以下の IO-Link ホームページをご参照ください.

http://www.io-link.com/en/

2. プロバイダの概要

2.1. 概要

IO-Link プロバイダは，マスターに，EtherNet/IP 通信で接続し，BALLUFF 社製の API「**IOLUDPIF20.dll**」を用いて，マスターのデータ及び，マスターのポートに接続されたデバイスに対しデータの読み書きを行います．

また，本プロバイダを使用するには，マスターについての情報が記載された「**IODM ファイル(※1)**」及びデバイスの情報が記載された「**IODD ファイル(※2)**」が必要です．

なお，本プロバイダで対応している IODD ファイルのスキーマバージョンは，1.1 です．Ver1.1 以外の IODD ファイルは動作保証対象外となりますのでご注意ください．

(**※1**) IODM ファイルは，**Bin¥XML** フォルダに置かれていますので，ご参照ください．

(**※2**) IODD ファイルは，以下の URL からダウンロードが可能です．

https://ioddfinder.io-link.com/#/

また，下図 2-1が本プロバイダとマスター及びデバイスの全体構成図になります．

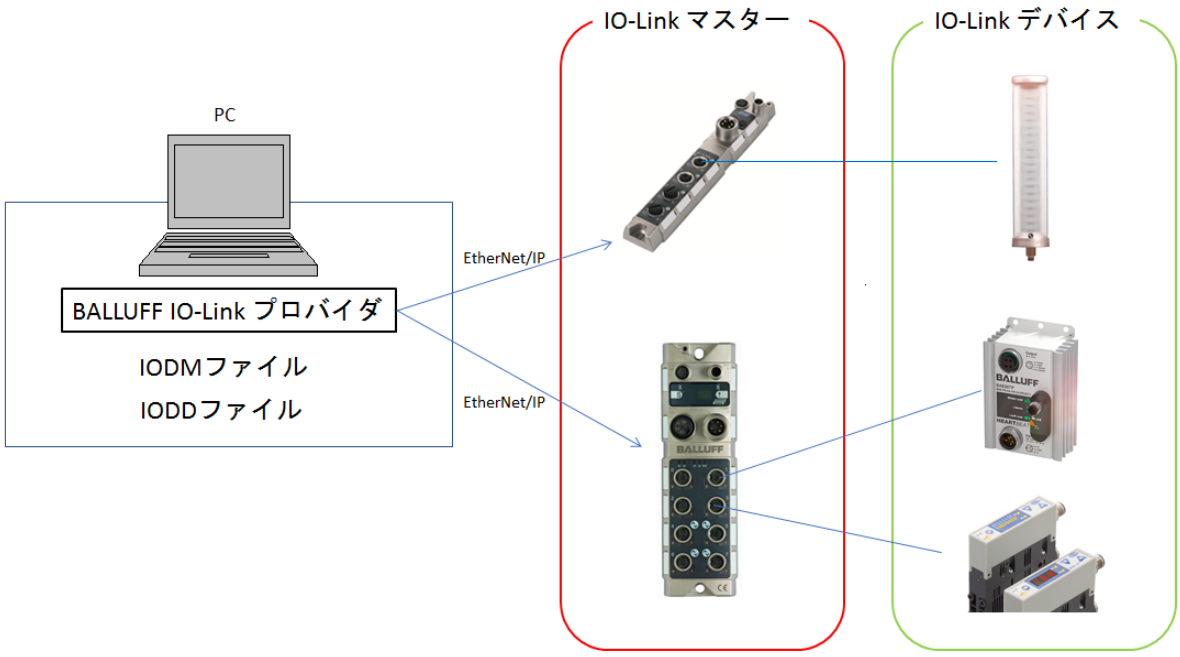

図 2-1 全体構成図

IO-Link プロバイダのファイル形式は DLL(Dynamic Link Library)となっており，その詳細は表 2-1 のようになっています.

表 2-1 BALLUFF IO-Link プロバイダ

ファイル名	CaoProvBALLUFFIOLink.dll
ProgID	CaoProv.BALLUFF.IOLink
レジストリ登録	regsvr32 CaoProvBALLUFFIOLink.dll
レジストリ登録の抹消	regsvr32 /u CaoProvBALLUFFIOLink.dll

また，本プロバイダ及びマスターとデバイスそれぞれの対応を表した図が下図図 2-2となります.
(※一例です．全てを表しているわけではありません．)

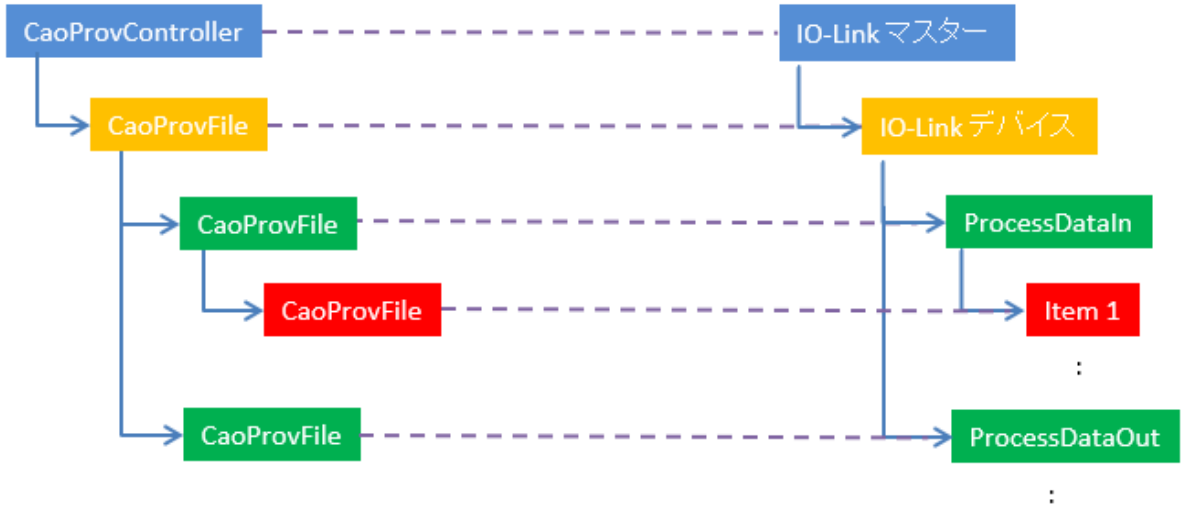

図 2-2 IO-Link プロバイダと IO-Link 機器のデータ対応図

2.2. ファイルクラスとデバイスのデータ関係

この章では，ファイルクラスと，それに接続するデバイスの持つデータの関係を記述します.

下図 2-3 ファイルクラスとデバイスのデータ関係図にファイルクラスのノードごとに割り振った番号順に説明します.

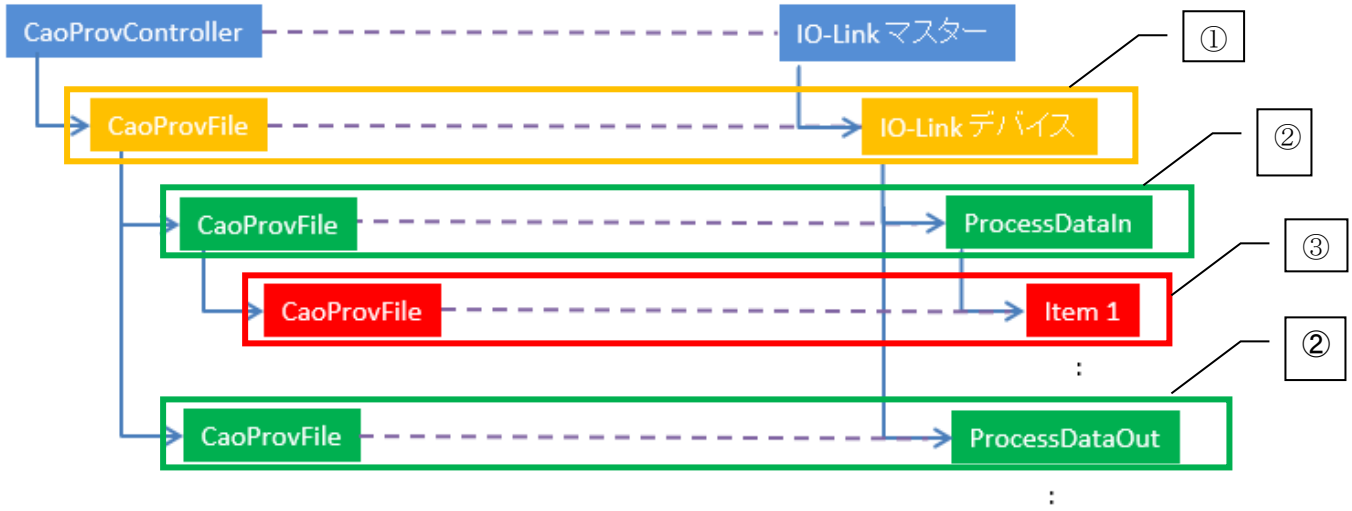

図 2-3 ファイルクラスとデバイスのデータ関係図

① デバイスノード

CaoController::AddFileメソッドで追加されたファイルクラスを，ここでは**デバイスノード**と呼びます.

デバイスノードでは，以下のメソッドを使用できます.

- CaoFile::AddFile
- CaoFile::GetFileNames

デバイスノードは，接続したデバイスが持つプロセスデータ及び各パラメータの名前情報を持ちます.保持している名前情報は,CaoFile::GetFileNames メソッドで取得することができます.

また，デバイスノードの CaoFile::AddFile メソッドで追加できるファイル名は，CaoFile::GetFileNames メソッドで取得した名前のみ使用することができます.

また，CaoFile::GetFileNames で取得したファイル名にはそれぞれ分類があり，以下の表 2-2 に名前と分類の対応を記述します.

表 2-2 CaoFile::GetFileNames メソッドで取得できるファイル名とその分類

名前	分類	説明	備考
"ProcessDataIn"	ProcessDataIn	デバイスの ProcessDataIn とデータをやり取りするファイルクラス名です.	読み込んだ IODD ファイルによっては，この名前が存在しない場合があります.
"ProcessDataOut"	ProcessDataOut	デバイスの ProcessDataOut とデータをやり取りするファイルクラス名です.	読み込んだ IODD ファイルによっては，この名前が存在しない場合があります.
各パラメータ名	Parameter	デバイスのパラメータ情報を持つファイルクラス名です.	

以下に，デバイスノードを追加するまでの，PacScript のサンプルを記述します.

使用例

```
Sub Main
        Dim caoCtrl as Object
        Dim caoSample as Object

        '①. IO-Link Maseter(SampleMaster)と接続
        caoCtrl=cao.AddController("IOLink","caoProv.Balluff.IOLink","",
"server=192.168.1.100,Path=SampleMaster.xml")
        '②. IO-Link デバイス(SampleDev)と接続
        caoSample=caoCtrl.AddFile("Sample","PortNo=0, Path=SampleDevice.xml")
End Sub
```

② データ分類ノード

デバイスノードの CaoFile::AddFile メソッドで追加されたファイルクラスを，ここでは，**データ分類ノード**と呼びます．

データ分類ノードは，IO-Link デバイスの持つデータ「ProcessDataIn」，「ProcessDataOut」，「Parameter」のうち，いずれかを表します．デバイスノードの CaoFile::AddFile メソッドで，ファイルを追加する際に"ProcessDataIn"を指定した場合，追加されたデータ分類ノードの分類は ProcessDataIn になります．

ProcessDataOut も同様に，デバイスノードの CaoFile::AddFile メソッドで，ファイルを追加する際に"ProcessDataOut"を指定した場合は ProcessDataOut となります．

デバイスノードの CaoFile::AddFile メソッドで"ProcessDataIn"，"ProcessDataOut"以外の，**各パラメータ文字列**を指定した場合は，分類は全て **Parameter** となります．

また，データ分類ノードもデバイスノードと同様に，CaoFile::AddFile メソッドで追加できるファイル名は，CaoFile::GetFileNames で取得したファイル名のみ指定できます．

以下にデータ分類ノードの分類毎に使用できるメソッドを表 2-3 に示します．

表 2-3 分類ごとに使用できるメソッド一覧

分類	使用可能メソッド
ProcessDataIn	・CaoFile::GetFileNames ・CaoFile::AddFile
ProcessDataOut	・CaoFile::GetFileNames ・CaoFile::AddFile
Parameter	・CaoFile::GetFileNames ・CaoFile::AddFile ・CaoFile::Execute("GetAccessType")

以下に，ProcessDataIn のデータ分類ノードを追加するまでの PacScript のサンプルを記述します．

使用例

```
Sub Main
        Dim caoCtrl as Object
        Dim caoSample as Object
        Dim caoProcessDataIn as Object

        '①. IO-Link Maseter(SampleMaster)と接続
        caoCtrl=cao.AddController("IOLink","caoProv.Balluff.IOLink","",
"server=192.168.1.100,Path=SampleMaster.xml")
        '②. IO-Link デバイス(SampleDev)と接続
        caoSample=caoCtrl.AddFile("Tower","PortNo=0, Path=SampleDev.xml")
        '③. 分類"ProcessDataIn"を追加
        caoProcessDataIn = caoSample.AddFile("ProcessDataIn")
End Sub
```

③ アイテムノード

データ分類ノードの CaoFile::AddFile メソッドで追加されたファイルクラスを，ここでは**アイテムノード**と呼びます.

アイテムノードは，IO-Link デバイスの持つデータである「**ProcessDataIn**」,「**ProcessDataOut**」及び「**Parameter**」のそれぞれがもつデータアイテムを表します. アイテムノードで実際に IO-Link デバイスとのデータのやり取りを行います.

また，アイテムノードは，どのデータ分類ノードから追加されたかによって，使用できるメソッドが変わります.

以下の表 2-4 に，使用できるメソッドを記述します.

表 2-4 分類毎に使用できるメソッド一覧

データ分類ノードの分類	アイテムノードの使用可能メソッド	備考
ProcessDataIn	・CaoFile::get_Value ・CaoFile::Execute("GetParameterList") ・CaoFile::Execute("GetType") ・CaoFile::Execute("GetRange") ・CaoFile::Execute("GetUnit")	---
ProcessDataOut	・CaoFile::get_Value ・CaoFile::put_Value ・CaoFile::Execute("GetParameterList") ・CaoFile::Execute("GetType") ・CaoFile::Execute("GetRange") ・CaoFile::Execute("GetUnit")	---
Parameter	・CaoFile::get_Value ・CaoFile::put_Value ・CaoFile::Execute("GetParameterList") ・CaoFile::Execute("GetType") ・CaoFile::Execute("GetRange") ・CaoFile::Execute("GetUnit")	※アクセス種類によっては，get，put が実行できない場合があります. 詳細は，ファイルクラスをご参照ください.

以下に，デバイス「SampleDevice」が持つ ProcessDataIn のアイテム「SampleItem」を追加するまでの，PacScript のサンプルを示します.

使用例

```
Sub Main
        Dim caoCtrl as Object
        Dim caoSampleDevice as Object
        Dim caoProcessDataIn as Object
        Dim caoSampleItem as Object

        '①. IO-Link Maseter と接続
        caoCtrl=cao.AddController("IOLink","caoProv.Balluff.IOLink","",
"server=192.168.1.100, Path=SampleMaster.xml")
        '②. IO-Link デバイスと接続
        caoSampleDevice=caoCtrl.AddFile("SampleDevice","PortNo=0, Path=SampleDev.xml")
        '③. "ProcessDataIn"を追加
        caoProcessDataIn = caoSampleDevice.AddFile("ProcessDataIn")
        '④. アイテムを追加
        caoSampleItem = caoProcessDataIn.AddFile("SampleItem")

End Sub
```

2.3. メソッド・プロパティ

2.3.1. CaoWorkspace::AddController メソッド

Controller オブジェクトの生成時に，マスターの IP アドレス及びマスターの情報を記述した IODM ファイルを指定します.

以下に，AddController の仕様を示します.

書式

```
AddController
(
        “<コントローラ名>”,          // コントローラ名(任意)
        “CaoProv.BALLUFF.IOLink”,    // プロバイダ名 (固定)
        “<マシン名>”,                // プロバイダ実行マシン名(未使用)
        “<オプション>”,              // オプション文字列
)
```

以下にオプション文字列に指定する文字列を示します.

表 2-5 CaoWorkspace::AddController のオプション文字列

オプション	必須	説明	値範囲	デフォルト値
Server= <マスターの IP アドレス>	○	接続したいマスターの IP アドレスを指定します.	255 文字まで	----
Path= <IODM ファイルのパス>	○	IODM ファイルへのパスを指定します. 指定方法の詳細については 2.3.1.1をご参照ください.	----	----

接続するマスターの IP アドレスが 192.168.1.100,読み込む IODM ファイルが Sample.xml の場合は,以下のように指定オプション文字列を入力します.

使用例

```
Dim caoCtrl as Object
caoCtrl=cao.AddController("Sample","CaoProv.Balluff.IOLink","","server=192.168.1.100,path=Sample.xml")
```

2.3.1.1. Path オプション

Path オプションに設定する，IODM ファイルの指定方法は，相対パスで指定する方法と，絶対パスで指定する方法の二つがあります. 以下に，相対パス，絶対パスのそれぞれの指定の詳細を記述します.

●相対パス

以下のディレクトリに，読み込む対象の IODM ファイルを置いて下さい.

(※2017 年 6 月時点で Balluff 社から提供されている EIP 対応の IODM ファイルについてはプリインストールしています.)

・Bin/XML

上記ディレクトリに，IODM ファイル「Balluff-BNI-EIP-507-005-Z040-20151110-IOLM1.3.xml」を置いた場合の，Path オプション指定の例を以下に示します.

使用例

```
Path=Balluff-BNI-EIP-507-005-Z040-20151110-IOLM1.3.xml
```

●絶対パス

読み込む IODM ファイルの絶対パスを指定してください.

使用例

```
Path=C:/User/Desktop/Balluff-BNI-EIP-507-005-Z040-20151110-IOLM1.3.xml
```

2.3.2. CaoController::AddVariable メソッド

CaoController から CaoVariable オブジェクトを生成します. 変数名には, 表 2-9 に示す変数名のみ使用できます.

以下に, AddVariable の仕様を示します.

書式

```
AddVariable
(
        “<変数名>”,    // 変数名
        “”,            // オプション文字列（未使用）
)
```

2.3.3. CaoController::GetVariableNames メソッド

表 2-9 コントローラクラス 変数一覧に示す変数名リストを取得します.

2.3.4. CaoController::AddFile メソッド

File オブジェクト生成時に, デバイスが接続されているマスターのポート番号と, IODD ファイルへのパスをオプションで指定します.

以下に, AddFile の仕様を示します.

書式

```
AddFile
(
        “Sample”,                                          // ファイル名(任意)
        “PortNo=<ポート番号>,Path=<IODD ファイルパス>” // オプション文字列
);
```

以下にオプション文字列に指定する文字列を示します.

表 2-6 CaoController::AddFile のオプション文字列

オプション	必須	説明	値範囲	デフォルト値
PortNo=<ポート番号>	○	IO-Link デバイス機器が接続されているマスターのポート番号を指定します. また, 指定ポートが IO-Link に対応していない場合は, エラーとなります.	0~マスターの最大ポート数	----
Path=<IODD ファイルへのパス>	○	IODD ファイルへのパスを指定します. 指定方法の詳細については 2.3.4.1をご参照ください.	----	----

接続するデバイスのポート番号が 0 番ポート, 読み込む IODD ファイル(ExampleDevice.xml)が C:¥Users¥ユーザ名¥Desktop 上に置かれている場合は, 以下の②のように指定オプション文字列を入力します.

使用例

```
Sub Main
        Dim caoCtrl as Object
        Dim caoSampleDevice as Object
        Dim caoProcessDataIn as Object
        Dim caoSampleItem as Object

        'IO-Link Maseter と接続
        caoCtrl=cao.AddController("IOLink","caoProv.Balluff.IOLink","",
        "server=192.168.1.100, Path=SampleMaster.xml")
        'IO-Link デバイスと接続
        caoSampleDevice=caoCtrl.AddFile("SampleDevice",
        "PortNo=0, Path=C:/Users/Desktop/ExampleDevice.xml")

End Sub
```

2.3.4.1. Path オプション

Path オプションに設定する，IODD ファイルの指定方法は，実行環境が **PC** もしくは **RC8** かによって変わります．以下に，実行環境が PC の場合と，RC8 の場合それぞれの指定の詳細を記述します．

●PC 環境

・相対パス指定

→ Bin¥XML

上記のディレクトリに,読み込み対象の IODD ファイルを置き,ファイル名を指定してください．

上記ディレクトリに，IODD ファイル「**Balluff-BNI_IOL-802-102-Z036-20150730-IODD1.1.xml**」を置いた場合の，Path オプション指定の例を以下に示します．

使用例

Path= Balluff-BNI_IOL-802-102-Z036-20150730-IODD1.1.xml

・絶対パス指定

読み込み対象の IODD ファイルの絶対パスを指定してください．

使用例

Path= C:/User/Desktop/Balluff-BNI_IOL-802-102-Z036-20150730-IODD1.1.xml

●RC8 環境

読み込み対象となるIODDファイルを，**WINCAPSⅢ**で**RC8**に転送してください．

その後，転送した IODD ファイル名を，Path オプションに指定すると，転送した IODD ファイルが読み込まれます．

RC8 に，IODD ファイル「**Balluff-BNI_IOL-802-102-Z036-20150730-IODD1.1.xml**」を転送した場合の，Path オプション指定の例を以下に示します．

使用例

Path= Balluff-BNI_IOL-802-102-Z036-20150730-IODD1.1.xml

2.3.5. (デバイスノード)CaoFile::AddFile メソッド

CaoFile から caoFile オブジェクトを生成します. ファイル名を指定することによって, マスターに接続したデバイスから, 取得する値を決定します.

以下に, CaoFile::AddFile の仕様を示します.

書式

```
CaoFile::AddFile
(
        “<ファイル名>”          // 変数名
        “”,                     // オプション文字列（未使用）
)
```

使用できる変数名については,CaoFile::GetFileNamesメソッドで取得できるファイル名のみ使用できます.

2.3.6. (データ分類ノード)CaoFile::AddFile メソッド

CaoFile から caoFile オブジェクトを生成します. ファイル名を指定することによって, マスターに接続したデバイスから, 取得する値を決定します.

以下に, CaoFile::AddFile の仕様を示します.

書式

```
CaoFile::AddFile
(
        “<ファイル名>”                  // 変数名
        “IsGetName=<データ型>”,         // オプション文字列
)
```

使用できる変数名については,CaoFile::GetFileNamesメソッドで取得できるファイル名のみ使用できます.

以下にオプション文字列に指定する文字列を示します.

表 2-7 (データ分類ノード)CaoFile::AddFile のオプション文字列

オプション	必須	説明	値範囲	デフォルト値
IsGetName=<データ型>	-	GetFileNames()で取得した文字列のみを,追加するファイルクラス名として AddFile メソッドに指定できます.	XML 内の値に紐付く定義名を取得するかどうかの指定をします. TRUE : BSTR で定義名を取得 FALSE : BSTR 以外で値を取得	TRUE

2.3.7. (デバイスノード, データ分類ノード)CaoFile::getFileNames メソッド

読み込んだ IODD デバイスファイルから，追加できるファイル名のリストを取得します.
リストの内容は，読み込んだ IODD デバイスファイルによって変化します.

2.3.8. (デバイスノード, データ分類ノード)CaoFile::Execute プロパティ

Execute メソッドを使用することで，アイテム等の情報を取得することができます.
詳細は，Execute コマンド一覧 を参照してください.

書式

```
caoFile::Execute
(
      <bstrCommand:BSTR>          // [in] コマンド名
      [,<vntParam:VARIANT>]       // [in] パラメータ
      [,<pVal:VARIANT>]])         // [in] 実行結果
)
```

2.3.9. (アイテムノード)CaoFile::put_Value プロパティ

作成したアイテムノードのファイルクラスによって，デバイスにデータを設定します.
アイテムノードの分類と，属性の対応については，表 2-16 を参照してください.

また，CaoFile::put_Value プロパティは，caoFile::Execute("GetParameterList")コマンドを実行し，パラメータのリストを取得できた場合(VT_EMPTY ではない)，取得したパラメータ文字列のみを指定できます. CaoFile::Execute("GetParameterList")コマンドで，パラメータのリストを**取得できなかった場合**は，値を**文字列**で設定できます. 設定できる値には，アイテムによっては設定範囲がある場合があります. 設定できる範囲は，CaoFile::Execute("GetRange")コマンドで取得できます. 範囲が存在した場合は，その範囲内で値を設定して下さい. 範囲外の値を設定するとエラーになります.
また，範囲が存在しなかった場合は，CaoFile::Execute("GetType")コマンドで取得した型が，そのまま設定できる値の範囲となります. 取得した型の範囲外の値を設定した場合は，エラーとなります.

2.3.10. (アイテムノード)CaoFile::get_Value プロパティ

作成したアイテムノードのファイルクラスによって，デバイスからデータを取得し，取得したデータは文字列で返されます.
CaoFile::get_Value メソッドは，どの分類のアイテムノードでも実行可能です. アイテムノードの分類と属性の対応については，表 2-16 を参照してください.

2.3.11. CaoVariable::get_Value プロパティ

指定した変数名によって，接続しているマスターからデータを取得します.
詳細は表 2-9 を参照してください.

2.4. Execute コマンド一覧

表 2-8 Execute コマンド一覧に，File クラスで使用できる Execute コマンド一覧を記述します.

表 2-8 Execute コマンド一覧

コマンド名	説明	詳細
GetAccessType	パラメータのアクセス種類を取得します.	P.22
GetUnit	アイテムの単位文字列を取得します.	P.23
GetType	アイテムの型を取得します.	P.23
GetRange	アイテムに設定できる値の範囲を取得します.	P.24
GetParameterList	アイテムに設定できる値を取得します.	P.24

GetAccessType

パラメータのアクセス種類を取得します. このコマンドは，データ分類ノードの分類が **Parameter** の時のみ使用できます. その他のノードで本コマンドを使用した場合は，エラーとなります.

項目	型説明	
vntParam	なし	
pVal	VT_UI1	パラメータのアクセス種類を取得します. 0 : Read / Write 1 : Read Only 2 : Write Only

使用例

```
CaoFile.Execute("GetAccessType")
```

GetUnit

アイテムの単位を文字列で取得します．このコマンドは，アイテムノードのみ使用できます．その他のノードで使用した場合は，エラーとなります．

項目	型説明	
vntParam	なし	
pVal	VT_BSTR	アイテムの単位を文字列で取得します． 例.）"℃"，"msec"など
	VT_EMPTY	単位は存在しません．

使用例

```
CaoFile.Execute("GetUnit")
```

GetType

アイテムの型を取得します．このコマンドは，アイテムノードのみ使用できます．その他のノードで使用した場合は，エラーとなります．

項目	型説明	
vntParam	なし	
pVal	VT_UI2	アイテムの型の値を取得します． 取得された型の VARENUM の値が，pVal の値に設定されます． 16 : VT_I1 2 : VT_I2 3 : VT_I4 20 : VT_I8 17 : VT_UI1 18 : VT_UI2 19 : VT_UI4 21 : VT_UI8 4 : VT_R4 5 : VT_R8 8 : VT_BSTR 11 : VT_BOOL

使用例

```
CaoFile.Execute("GetType")
```

GetRange

アイテムに設定できる値の範囲を取得します.このコマンドは,アイテムノードのみ使用できます. その他のノードで使用した場合は, エラーとなります.

項目	型説明		
vntParam	なし		
pVal	VT_ARRAY | VT_UI8		
	0	VT_UI8	設定できる値の最小値を取得します.
	1	VT_UI8	設定できる値の最大値を取得します.
	VT_ARRAY | VT_I8		
	0	VT_I8	設定できる値の最小値を取得します.
	1	VT_I8	設定できる値の最大値を取得します.
	VT_ARRAY | VT_R4		
	0	VT_R4	設定できる値の最小値を取得します.
	1	VT_R4	設定できる値の最大値を取得します.
	VT_EMPTY		範囲は存在しません.

使用例

```
CaoFile.Execute("GetRange")
```

GetParameterList

アイテムに設定できる値を取得します. 値が取得できた場合は, CaoFile::put_Value プロパティに設定できる値は, GetParameterList コマンドで取得できた値のみを設定することができます.

このコマンドは, アイテムノードのみ使用できます. その他のノードで使用した場合は, エラーとなります.

項目	型説明		
vntParam	なし		
pVal	VT_ARRAY | VT_BSTR		
	i	VT_BSTR	アイテムに設定できる値を取得します.
	VT_EMPTY		パラメータは存在しません.

使用例

```
CaoFile.Execute("GetParameterList")
```

2.5. 変数一覧

2.5.1. コントローラクラス

以下の表 2-9 に，コントローラクラスで使用できる変数一覧を記述します.

表 2-9 コントローラクラス 変数一覧

変数名	データ型	説明	属性	
			get	put
@MAKER_NAME	VT_BSTR	メーカー名(BALLUFF)	○	-
@VERSION	VT_ARRAY | VT_VARIANT	●ファームウェアのバージョン，型式番号情報を取得します. ※以下の順番で配列に格納されています.	○	-
		VT_BSTR: 下記パラメータ A,B,C,D,E,F から成る文字列[A.B.C/D.E.F]		
		VT_UI1: 主要なファームウェアバージョン(A)		
		VT_UI1: 細かいファームウェアバージョン(B)		
		VT_UI1: ファームウェア改定構成(C)		
		VT_UI1: マスター記憶装置の主要な改定(D)		
		VT_UI1: マスター記憶装置の細かい改定 (E)		
		VT_UI1: マスター記憶装置改定の構成 (F)		
@LAST_EVENT	VT_ARRAY | VT_VARIANT	●マスターのイベント情報を取得します. ※以下の順番で配列に格納されています.	○	-
		VT_UI2: DLL へのイベント積算数		
		VT_UI2: イベントが発生したポート番号		
		VT_UI2: イベントコード(表 2-10 イベントコードの		
		VT_UI1: イベントのインスタンス(表 2-11 イベントの照)		
		VT_UI1: イベントモード(表 2-12 イベントモードの		
		VT_UI1: イベントタイプ(表 2-13 イベントタイプの		
		VT_UI1: イベントモード 2(表 2-14 イベントモード 2		
		VT_UI1: IO-Link マスターからイベントが発行されたかの是非(表 2-15 IO-Link マスターからイベント値と説明を参照)		

使用例

マスターの最後のイベントを取得する場合

```
CaoController.AddVariable("@LAST_EVENT","");
```

※変数名は大文字・小文字関係なく入力可能です.

```
AddVariable("@last_event","");
```

表 2-10 イベントコードの取得できる値と説明

取得値	説明
2	フレームエラー受信
16	デバイス接続解除(接続解除,断線,デバイス処理中など)
26	異なるセンサーを検出,予期せぬエラー
27	リトライを検出
30	C/Q（コモン）線上に短絡検出
31	センサー供給にエラー有り
32	アクチュエーター供給にエラー有り
33	IO-Link マスタへの電源供給にエラー有り
34	ポートがリセット場合にイベントが送られます
35	フォールバック成功,デバイスは SIO モード
36	デバイスの使用準備完了
40	データストレージが完了しましたが, CRC が正常であったため何も行いませんでした.
50	パラメータダウンロード完了
51	パラメータアップロード完了
64	入力プロセスデータ長に相違有り
65	出力プロセスデータ長に相違有り
66	デバイスの revision（改定）に相違有り
67	V1.1 センサーの vendorID（ベンダーID）に相違有り
68	V1.1 センサーの DeviceID（デバイス ID）に相違有り
69	V1.0 センサーの VendorID に相違有り
70	V1.0 センサーの DeviceID に相違有り
71	SerialNumber に相違有り
72	サイクルタイムが合いません

表 2-11 イベントのインスタンスの取得できる値と説明

取得値	説明
0	不明なインスタンス
1	インスタンスの物理層
2	インスタンスのデータ層
3	インスタンスアプリケーション層
4	インスタンスアプリケーション

表 2-12 イベントモードの取得できる値と説明

取得値	説明
0	単一メッセージまたは警告(ワーニング)
1	エラー発生
2	エラー解消

表 2-13 イベントタイプの取得できる値と説明

取得値	説明
0	メッセージ
1	警告(ワーニング)
2	エラー

表 2-14 イベントモード 2 の取得できる値と説明

取得値	説明
0	異常
1	プロセスデータ正常

表 2-15 IO-Link マスターからイベントが発行されたかの是非の取得できる値と説明

取得値	説明
0	発行されていない
1	発行された

2.5.2. ファイルクラス

以下の表 2-16 に，アイテムノードのファイルクラスと，その属性を記述します.

表 2-16 アイテムノードファイルクラス 分類毎の属性

データ分類ノードの分類	アクセス種類 (※)	説明	属性	
			get	put
ProcessDataIn	-----	デバイスのプロセスデータインのデータを取得します.	○	-
ProcessDataOut	-----	デバイスのプロセスデータアウトの値を読み書きします.	○	○
Parameter	Read / Write(0)	デバイスのパラメータで，読み書きが可能なアイテムです.	○	○
	Read Only(1)	デバイスのパラメータで，読み込みのみが可能なアイテムです.	○	-
	Write Only(2)	デバイスのパラメータで，書き込みのみが可能なアイテムです.	-	○

※アクセス種類については，CaoFile::Execute("GetAccessType")コマンドで取得できます.

2.6. エラーコード

本プロバイダには，独自のエラーコードが存在します．詳細は以下の表 2-17 独自エラーコード表をご参照ください．

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ご参照ください．

表 2-17 独自エラーコード表

エラー番号	説明
0x80110001	必須オプションが設定されていません．
0x80110002	単位定義ファイルの読み込みに失敗しました． 単位定義ファイルが既定のディレクトリ(Bin¥XML)に存在するかご確認下さい．
0x80110003	"Path="オプションに設定した文字列から，XMLファイルを開くのに失敗しました． 指定したファイルはXMLファイル形式かどうかご確認下さい．
0x80110004	読み込んだXMLファイルが正しくありません． IODMファイルまたは，IODDファイルは正しいものかご確認下さい．
0x80110005	指定したポート番号は存在しません．
0x80110006	指定したポート番号はIO-Linkに対応していません．
0x80110007	ポート番号が指定されていません．
0x80110008	指定された型に変換できません．
0x80110009	指定されたIODDファイルのスキーマバージョンが対象外です．

また，本プロバイダは，Balluff 社製 API「IOLUDPIF20」からのエラーを[0x8010****]でマスクして返します．API からのエラーについては，Balluff 社のインターフェース説明書の「DLL マニュアル 20160608.pdf」をご参照ください．（4.付録に，エラーコードの部分を引用したものを記載します．）

3. サンプルプログラム

この章では, デバイスとのデータ通信を行う PacScript のサンプルを記述します.

前提条件：
・使用する IO-Link マスターは「BNI EIP-507-005-Z040」とする.
・使用する IODM ファイルは
「Balluff-BNI-EIP-507-005-Z040-20151110-IOLM1.3.xml」とする.
・マスター機器の IP アドレスは「192.168.1.100」とする.
・各デバイスが接続されているポートの番号は 0 番とする.

① Schmalz-SCPSi_V2

以下に,本プロバイダと Schmalz-SCPSi_V2 の通信を行い,ProcessDataOut のアイテム「Vacuum」にデータ書き込みを行うサンプルプログラムを記述します.

前提条件：
・使用する IO-Link デバイスは「Schmalz-SCPSi_V2」とする.
・使用する IODD ファイルは「Schmalz-SCPSi_V2-20140829-IODD1.1.xml」とする.

```
Sub Main
        ‘ コントローラ追加
        Dim caoCtrl as Object
        caoCtrl = cao.AddController("IOLink", "CaoProv.Balluff.IOLink", "",
"server=192.168.1.100,path=Balluff-BNI-EIP-507-005-Z040-20151110-IOLM1.3.xml")

        ‘ デバイスノード追加
        Dim caoFileSCPSi as Object
        caoFileSCPSi = caoCtrl.AddFile("Schmalz", "PortNo=0,
Path=Schmalz-SCPSi_V2-20140829-IODD1.1.xml")

        ‘ データ分類ノード追加(ProcessDataOut)
        Dim caoFileSCPSi_ProcessDataOut as Object
        caoFileSCPSi_ProcessDataOut = caoFileSCPSi.AddFile("ProcessDataOut")

        ‘ アイテムノード追加
        Dim caoFileSCPSi_ProcessDataOut_Vacuum as Object
        caoFileSCPSi_ProcessDataOut_Vacuum = caoFileSCPSi_ProcessDataOut.AddFile("Vacuum")

        ‘ 値の書き込み
        Delay 1000
        caoFileSCPSi_ProcessDataOut_Vacuum.Value = "-1"
        Delay 1000
        caoFileSCPSi_ProcessDataOut_Vacuum.Value = "0"
End Sub
```

② BNI_IOL-802-102-Z036

以下に，本プロバイダと BNI_IOL-802-102-Z036 の通信を行い，ProcessDataOut のデータを書き込み，デバイスのライトをオレンジ色に光らせるサンプルプログラムを記述します.

前提条件：
・接続する IO-Link デバイスは「BNI_IOL-802-102-Z036」とする.
・使用する IODD ファイルは「Balluff-BNI_IOL-802-102-Z036-20150730-IODD1.1.xml」とする.

```
Sub Main
        ‘ コントローラ追加
        Dim caoCtrl as Object
        caoCtrl = cao.AddController("IOLink", "CaoProv.Balluff.IOLink", "",
"server=192.168.1.100,path=Balluff-BNI-EIP-507-005-Z040-20151110-IOLM1.3.xml")

        ‘ デバイスノード追加
        Dim caoFileTowerLigth as Object
        caoFileTowerLigth = caoCtrl.AddFile("TowerLight", "PortNo=0,
        Path= Balluff-BNI_IOL-802-102-Z036-20150730-IODD1.1.xml")

        ‘ データ分類ノード追加(ProcessDataOut)
        Dim caoFileTowerLigth_ProcessDataOut as Object
        caoFileTowerLigth_ProcessDataOut = caoFileTowerLigth.AddFile("ProcessDataOut")

        ‘ アイテムノード追加
        Dim caoFileTowerLigth_ProcessDataOut_OperatingMode as Object
        Dim caoFileTowerLigth_ProcessDataOut_BackColor as Object
        caoFileTowerLigth_ProcessDataOut_OperatingMode =
        caoFileTowerLigth_ProcessDataOut.AddFile("Operating mode")
        caoFileTowerLigth_ProcessDataOut_BackColor =
        caoFileTowerLigth_ProcessDataOut.AddFile("Segment 1 color / Background color")

        ‘ 値の書き込み
        Delay 1000
        caoFileTowerLigth_ProcessDataOut_OperatingMode.Value = "Runlight mode"
        Delay 1000
        caoFileTowerLigth_ProcessDataOut_BackColor.Value = "Orange"
End Sub
```

③ BAE-PS-XA-1W

以下に,本プロバイダと BAE-PS-XA-1W の通信を行い,ProcessDataIn のデータ「Input Voltage」の値を読み込むサンプルプログラムを記述します.

前提条件：
・接続する IO-Link デバイスは「BAE-PS-XA-1W」とする.
・使用する IODD ファイルは「Balluff-BAE-PS-XA-1W-24-038-607-20151119-IODD1.1.xml」とする.

```
Sub Main
        ‘ コントローラ追加
        Dim caoCtrl as Object
        caoCtrl = cao.AddController("IOLink", "CaoProv.Balluff.IOLink", "",
"server=192.168.1.100,path=Balluff-BNI-EIP-507-005-Z040-20151110-IOLM1.3.xml")

        ‘ デバイスノード追加
        Dim caoFileSupply as Object
        caoFileSupply = caoCtrl.AddFile("Supply", "PortNo=0,
        Path= Balluff-BAE-PS-XA-1W-24-038-607-20151119-IODD1.1.xml")

        ‘ データ分類ノード追加(Parameter:Input Voltage)
        Dim caoFileSupply_InputVoltage as Object
        caoFileSupply_InputVoltage = caoFileSupply.AddFile("Input Voltage")

        ‘ アイテムノード追加
        Dim caoFileSupply_InputVoltage_Item as Object
        caoFileSupply_InputVoltage_Item = caoFileSupply_InputVoltage.AddFile("Item")

        ‘ 値の読み込み
        Dim Result as Object
        Delay 1000
        Result = caoFileSupply_InputVoltage_Item.Value
End Sub
```

4.付録

Balluff 社のインターフェース説明書「DLL マニュアル 20160608.pdf」から，API のエラーコード値を引用したものを以下に記載します.

RETURN_FIRMWARE_NOT_COMPATIBLE -16
実行できない関数があるためファームウェアをアップデートする必要があります
RETURN_FUNCTION_NOT_IMPLEMENTED -13
接続された IO-Link マスターでは指定の関数が実行できません
RETURN_STATE_CONFLICT -12
現在の IO-Link マスター設定では指定された関数は実行できません
RETURN_WRONG_COMMAND -11
命令コマンドに対して IO-Link マスターから誤ったアンサーが返ってきました
RETURN_WRONG_PARAMETER -10
関数パラメーターが無効です
RETURN_WRONG_DEVICE -9
デバイス名が間違っています，または，サポートされないデバイスが接続されています
RETURN_NO_EVENT -8
イベント読取りが命令されましたが，イベントがありません.
RETURN_UNKNOWN_HANDLE -7
不明な関数の処理です.
RETURN_UART_TIMEOUT -6
タイムアウトになりました，コマンドに対するアンサーがありません
RETURN_CONNECTION_LOST -5
交信中にマスターが取外されました
RETURN_OUT_OF_MEMORY -4
使用可能なメモリーがありません
RETURN_DEVICE_ERROR -3
UDP ドライバーへアクセス中にエラーが発生しました
RETURN_DEVICE_NOT_AVAILABLE -2
一時的にデバイスが使用できない状態です
RETURN_INTERNAL_ERROR -1
内部ライブラリーエラーのためプログラムをリスタートして下さい
RETURN_OK 0
正常完了
RESULT_STATE_CONFLICT 1
現設定に対して命令コマンドが正しくありません

RESULT_NOT_SUPPORTED 2

このデバイスはコマンドに対応していません.

RESULT_SERVICE_PENDING 3

サービス保留中.　新しいサービスは保留中のサービスが完了するまで実行しないで下さい

RESULT_WRONG_PARAMETER_STACK 4

IO-link マスターにパラメーターが拒否されました