

LocalFile プロバイダ 汎用ファイルアクセス

Version 1.1.1

ユーザーズ ガイド

April 10, 2020

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0.0	2006-02-24	初版
1.0.1.0	2008-02-06	マニュアル修正
1.0.2.0	2009-08-20	マニュアル修正 AddController の引数
1.0.2.1	2010-02-11	エラーコード追加
1.0.3.0	2011-12-12	Append コマンド追加, TextMode オプション追加
1.0.3	2012-07-17	ドキュメントのバージョンルールを変更
1.1.0	2020-01-30	CaoController の GetFileList コマンド追加
1.1.1	2020-04-10	GetFileList のバグ修正

【対応機器】

機種	バージョン	注意事項

目次

1. はじめに	4
2. プロバイダの概要	5
2.1. 概要	5
2.2. メソッド・プロパティ	6
2.2.1. CaoWorkspace::AddController メソッド	6
2.2.2. CaoController::AddFile メソッド	6
2.2.3. CaoController::Execute コマンド	7
2.2.3.1. GetFileList のパラメータ詳細	8
2.2.3.2. GetFileList の返り値詳細	9
2.2.4. CaoFile::AddFile メソッド	10
2.2.5. CaoFile::Copy メソッド	10
2.2.6. CaoFile::Delete メソッド	11
2.2.7. CaoFile::Move メソッド	11
2.2.8. CaoFile::Execute メソッド	11
2.2.9. CaoFile::get_FileNames プロパティ	12
2.2.10. CaoFile::GetAttribute プロパティ	12
2.2.11. CaoFile::get_Value プロパティ	12
2.2.12. CaoFile::put_Value プロパティ	13
2.3. 変数一覧	14
2.3.1. コントローラクラス	14
2.4. エラーコード	14
3. サンプルプログラム	15

1. はじめに

このドキュメントは、ローカルマシン上にあるファイルにアクセスするためのプロバイダである LocalFile プロバイダのユーザーズガイドです。

CAO の CaoFile オブジェクトと実際のファイルを 1 対 1 に対応しています。これにより CaoFile オブジェクトが実際のファイルのツリー構造を再現できるように実装されています。

本書は、この LocalFile 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。

2. プロバイダの概要

2.1. 概要

LocalFile プロバイダは、CAO のインタフェースを用いてローカルマシン上のファイルにアクセスします。

また、Windows のシステムエラーが発生した場合は、エラー番号を“0x8010000”でマスクした値を CAO API は返します。

例) Windows のシステムエラー: 2 (0x0002) → CAO API のエラー: 0x80100002

表 1 LocalFile プロバイダ

ファイル名	CaoProvLocalFile.dll
ProgID	CaoProv.LocalFile
レジストリ登録 ¹	regsvr32 CaoProvLocalFile.dll
レジストリ登録の抹消	regsvr32 /u CaoProvLocalFile.dll

¹ ORiN SDK でインストールした場合は手動で登録/抹消する必要はありません。

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

このメソッドで CaoController オブジェクトを生成します。コントローラ名は任意の名前を使用することができます。以下に AddController の引数仕様を示します。

```
AddController
(
    "<コントローラ名>",           // 仮想コントローラ名
    "GaoProv. LocalFile",       // プロバイダ名。固定。
    "<マシン名>",               // (未使用)
    "<オプション>"              // オプション文字列
)
```

2.2.2. CaoController::AddFile メソッド

この AddFile メソッドでは CaoFile オブジェクトを作成します。そのオブジェクトに対応するファイルをオブジェクト名として指定します。以下に AddFile の引数仕様を示します。

```
AddFile
(
    "<ファイル名>",           // ファイル名
    "<オプション>"           // オプション文字列
)
```

オプションは以下の文字列を使用します。

表 2 CaoController::AddFile のオプション文字列

オプション	意味
Path[=<パス名>]	ファイル名として指定したファイルのあるディレクトリのパスを記述します。(デフォルト:システムドライブ) 絶対パスか UNC で入力して下さい。相対パスのときは、AddFile メソッドは失敗します。このとき、パスのデリミタは“¥”, “/”のどちらでも使用できます。
@Create[=<0~2>]	指定したファイルがないとき、このオプション値に従ってファイルを作成します。 0:ファイルを作成しません。(デフォルト) 1:ファイルを作成します。 2:ディレクトリを作成します。 指定したファイルが存在する、またはファイル名がドライブときはこのオプションは無視されます。
TextMode [=<TRUE/FALSE>]	指定したファイルの内容をテキスト形式で取得するかを指定します。 TRUE:テキスト形式でデータを取得します。 FALSE:バイナリ形式でデータを取得します。(デフォルト)

2.2.3. CaoController::Execute コマンド

コマンドを実行します。

使用可能なコマンドの詳細は以下を参照してください。

表 3 CaoController::Execute メソッドのコマンド実装一覧

コマンド	パラメータ	返り値	動作
GetFileList	<ul style="list-style-type: none"> ・パス ・フィルタ ・モード <p>詳細は GetFileList のパラメータ詳細 を参照してください。</p>	<ul style="list-style-type: none"> ・ファイル情報一覧 <p>詳細は GetFileList の返り値詳細 を参照してください。</p>	<p>指定したパス内のファイル・ディレクトリの一覧を取得します。</p> <p>取得する一覧の最大数は 10,000 個となります。</p> <p>10,000 個を超える結果となった場合は、E_CAOP_LISTLIMITOVER (エラーコード参照)が発生します。</p> <p>結果が 0 個だった場合は VT_EMPTY が返ります。</p> <p>ワイルドカードを使用してフィルタリングを行うことが可能です。</p> <p>検索結果は、</p> <ul style="list-style-type: none"> ・ファイル+ディレクトリ ・ファイルのみ ・ディレクトリのみ <p>を指定することが可能です。</p>

2.2.3.1. GetFileList のパラメータ詳細

VT_ARRAY | VT_VARIANT

要素	項目	型	必須	説明
第 1 要素	パス	VT_BSTR	○	取得対象のパスを指定します。 詳細は CaoController::AddFile メソッド の Path オプションを参照してください。
第 2 要素	フィルタ	VT_BSTR	—	取得対象のフィルタリングを行います。 ワイルドカード ・「*」:長さ 0 文字以上の任意の文字列にマッチするパターン ・「?」:任意の一文字にマッチするパターン が使用可能です。 8.3 形式の別名もチェックの対象となります。 指定しなかった場合は「*」が指定されたものとします。
第 3 要素	モード	VT_I4	—	検索対象を指定します。 以下の値を指定することができます。 0:ファイル+ディレクトリ(デフォルト) 1:ファイルのみ 2:ディレクトリのみ 指定しなかった場合は 0 が指定されたものとします。

2.2.3.2. GetFileList の返り値詳細

VT_ARRAY | VT_VARIANT – ファイル情報の一覧

└VT_ARRAY | VT_VARIANT – ファイル情報 ([ファイル情報詳細](#)参照)²

| └...

└...ファイル・ディレクトリ数分繰り返し

2.2.3.2.1. ファイル情報詳細

要素	項目	型	説明
第 1 要素	ファイル名	VT_BSTR	検出されたファイルの名前(拡張子を含む)を取得します。
第 2 要素	完全パス	VT_BSTR	検出されたファイルの完全なパスを取得します。
第 3 要素	ディレクトリ属性	VT_BOOL	見つかったファイルがディレクトリであるかどうかを判断します。 TRUE: ディレクトリ FALSE: ファイル
第 4 要素	サイズ	VT_UI8	検出されたファイルの長さをバイト単位で取得します。
第 5 要素	作成日時	VT_DATE	ファイルが作成された時刻を取得します。
第 6 要素	更新日時	VT_DATE	ファイルが最後に変更されて保存された時刻を取得します。
第 7 要素	アクセス日時	VT_DATE	ファイルが最後にアクセスされた時刻を取得します。
第 8 要素	読み取り専用属性	VT_BOOL	検出されたファイルが読み取り専用かどうかを判断します。 TRUE: 読み取り専用 FALSE: 読み取り専用でない
第 9 要素	隠しファイル属性	VT_BOOL	検出されたファイルが非表示かどうかを判断します。 TRUE: 非表示 FALSE: 非表示でない

² ファイルの検出順番はファイルシステムによって異なります。
通常、NTFS や CDFS の場合はアルファベット順、FAT の場合は書き込まれた順ですが、必ずしも保証されているものではありません。

2.2.4. CaoFile::AddFile メソッド

前述 2.2.2 と同様にファイルオブジェクトを作成します。ファイル名には、このメソッドを実行する CaoFile オブジェクトと対応しているディレクトリ内のファイルのみを指定することができます。このため前述 2.2.2 のオプションである Path オプションは無視されます。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と対応していないときは失敗します。

2.2.5. CaoFile::Copy メソッド

このメソッドを実行すると、ファイルを指定した場所にコピーします。但し、CaoFile オブジェクトがドライブ、若しくは共有ディレクトリである場合にはメソッドは失敗します。

表 4 対応ファイルごとの Copy メソッドの動作

対応しているファイル	Copy メソッドの動作
ファイル	ファイルをコピーします。
ディレクトリ	ディレクトリとその中のファイル、サブディレクトリをコピーします。
ドライブ、共有ディレクトリ	コピーしないで、エラーを返します。

以下に Copy の引数仕様を示します。

```
Copy
(
    “<コピー先>”,           // ファイルのコピー先
    “<オプション>”         // オプション文字列
)
```

オプションは以下の文字列を使用します。

表 5 CaoFile::Copy のオプション文字列

オプション	意味
Overwrite[=<True/False>]	コピー先にファイルがあるときの上書き設定。 True : 上書きします。 False : 上書きしません。(デフォルト値)

コピー先の指定方法でコピーする場所は以下のようになります。

表 6 CaoFile::Copy のコピー先の指定方法とそのコピー先

コピー先の指定方法	結果	例:指定方法	コピー結果
絶対パス	コピー先に指定した場所にコピーします。	/TestDir2/Test2.txt	C:¥TestDir2¥Test2.txt
相対パス	コピー元のファイルのあるディレクトリを基準とした場所にコピーします。	./TestDir2/Test.txt	C:¥TestDir¥TestDir2¥Test.txt
ファイル名のみ	コピー元と同じディレクトリにコピーします。	Test2.txt	C:¥TestDir¥Test2.txt
パスの最後にデリミタ	コピー元と同名のファイルをコピー先のディレクトリにコピーします。	/TestDir2/	C:¥TestDir2¥Test.txt

※ このときコピー元のパスは“C:¥TestDir¥Test.txt”とします。

またコピー先に以下の条件のものを入力したときはエラーを返します。

- ・ コピー先とコピー元が同じとき。
- ・ コピー先がコピー元のサブディレクトリであるとき。

2.2.6. CaoFile::Delete メソッド

オブジェクトに対応しているファイルを削除します。ファイルを削除した後もオブジェクトは消去されないの
で、不必要なときはクライアントでオブジェクトを消す必要があります。

2.2.7. CaoFile::Move メソッド

Move メソッドは前述 2.2.5Copy メソッドを実行した後、2.2.6Delete メソッドを実行します。以下に Move の引
数仕様を示します。

2.2.8. CaoFile::Execute メソッド

コマンドを実行します。

使用可能なコマンドの詳細は以下を参照してください。

表 7 CaoFile::Execute メソッドのコマンド実装一覧

コマンド	パラメータ	返り値	動作
Refresh	なし	なし	ファイルの属性, 作成日時, アクセス日時, 更新日時の情報を更新します。

Append	<DATA:VT_BSTR> = 追加文字列	なし	指定した文字列を追記します。
--------	---------------------------	----	----------------

2.2.9. CaoFile::get_FileNames プロパティ

CaoFile オブジェクトに対応しているディレクトリの中にあるファイル名リストを取得します。CaoFile オブジェクトに対応しているのがファイルのときはこのプロパティは失敗します。

表 8 CaoFile::get_FileName のオプション文字列

オプション	意味
Filter[=<フィルタ文字列>]	取得リストのフィルタの設定。(デフォルト: "*.*)"

2.2.10. CaoFile::GetAttribute プロパティ

オブジェクトに対応しているファイルの属性値を取得します。取得できる属性とその値は以下のようになっています。複数の属性を持つファイルの場合は値の総計をとります。

表 9 GetAttribute で取得できる属性とその値

属性名	値
読み取り専用	0x001
隠しファイル	0x002
システムファイル	0x004
ディレクトリ	0x010
アーカイブ	0x020
ノーマル	0x080
一時ファイル	0x100
圧縮ファイル	0x800

2.2.11. CaoFile::get_Value プロパティ

オブジェクトに対応しているファイルの内容を取得します。ディレクトリのときは失敗します。

取得時のデータ形式は、AddFile の TextMode オプションで決定します。詳細は以下のようになります。

表 10 TextMode オプションと取得データ形式

TextMode オプションの値	取得形式
------------------	------

TRUE	テキスト形式 (VT_BSTR) データの中に NULL 文字が含まれている場合は、その直前までのデータを取得します。
FALSE	バイナリ形式 (VT_UI1 VT_ARRAY)

2.2.12. CaoFile::put_Value プロパティ

オブジェクトに対応しているファイルの内容を設定します。クライアントはプロバイダに対し文字列、バイナリ配列のどちらでデータを送っても設定することができます。ディレクトリのときは失敗します。

2.3. 変数一覧

2.3.1. コントローラクラス

表 2-11 コントローラクラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@Version	VT_BSTR	バージョン情報	○	-

2.4. エラーコード

LocalFile プロバイダでは、以下の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 2-12 独自エラーコード一覧

エラー名	エラー番号	説明
E_CAOP_FILENOTFOUND	0x80100800	ファイルが見つかりません
E_CAOP_LISTLIMITOVER	0x80100801	リストの制限を超えました

3. サンプルプログラム

以下にファイルの作成, 内容の取得, 設定をするサンプルを示します.

ファイルパス : “C:¥Sample.txt”

List 3-1**Sample.frm**

```
Private eng As CaoEngine
Private ctrl As CaoController
Private file As CaoFile

Private Sub Form_Load()
    Dim ws As CaoWorkspace

    Set eng = New CaoEngine
    Set ws = eng.Workspaces(0)

    ' ローカルファイルプロバイダの起動
    Set ctrl = ws.AddController("sample", "CaoProv.LocalFile")
    ' ファイルの取得/作成
    Set file = ctrl.AddFile("Sample.txt", "Path=C:¥, @Create=1, TextMode=True")

End Sub

' ファイル内容の設定
Private Sub Command1_Click()
    file = Text1.Text
End Sub

' ファイル内容の取得
Private Sub Command2_Click()
    Text2.Text = file
End Sub
```