

DataImport プロバイダ

Version 1.1.5

ユーザーズ ガイド

January 24, 2023

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0	2016-06-27	初版.
1.0.1	2016-09-02	データの読み出し方式を変数取得から OnMessage イベント通知へ変更. 各オプション設定の変更・追加. Execute コマンド追加. サブフォルダ配下を読み出し対象にできるよう対応.
1.0.2	2018-01-30	読み出し対象 CSV ファイルの注意事項を追記.
1.1.0	2018-05-11	AddController に SleepTime オプションの追加
1.1.1	2018-10-30	メモリーリーク バグ修正
1.1.2	2019-02-14	Execute(“Reset”)コマンド追加
1.1.3	2019-04-22	ファイル読み出し処理修正 Execute(“Reset”)コマンド処理修正
1.1.4	2020-05-19	ファイルの読み出し失敗時の処理修正
	2020-12-01	誤記修正
	2021-11-30	プロバイダ概要を加筆修正 誤記修正
1.1.5	2023-01-24	プロバイダ初期化時に監視フォルダに大量のファイルが存在すると異常終了する不具合を修正.

【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
2.1.1. 読み込みフォルダの動作条件.....	5
2.1.2. プロバイダの基本的な動作.....	7
2.1.2.1. 読出ファイルリスト.....	9
2.1.2.2. 読出し完了判定.....	9
2.1.2.3. 中断データファイル.....	9
2.2. メソッド・プロパティ.....	12
2.2.1. CaoWorkspace::AddController メソッド.....	12
2.2.1.1. ファイルの更新時間フィルタ.....	15
2.2.1.2. ファイル後処理の種類と指定方法.....	18
2.2.1.3. データ区切り文字.....	19
2.2.1.4. ダブルクォーテーションで括られたデータの解釈.....	20
2.2.2. CaoController::AddVariable メソッド.....	21
2.2.3. CaoVariable:get_Value プロパティ.....	21
2.2.4. CaoController::Execute.....	21
2.2.4.1. StartReading コマンド.....	22
2.2.4.2. StopReading コマンド.....	22
2.2.4.3. Reset コマンド.....	22
2.2.5. CaoController::OnMessage イベント.....	23
2.2.5.1. エラーログ.....	23
2.2.5.2. 読み出しデータ通知.....	23
2.2.5.3. データ構造変化通知.....	24
2.3. 変数一覧.....	24
2.3.1. CaoController クラス.....	24
2.4. エラーコード.....	24

1. はじめに

本書は、CSV ファイルに対し横断的にアクセスし、データ取得を行う 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(CaoProvDataImport.dll)を DataImport プロバイダと呼びます。

第 2 章に DataImport プロバイダの概要、変数の詳細を記載しています。

2. プロバイダの概要

2.1. 概要

DataImport プロバイダは、指定フォルダに保存してある CSV ファイルに対し横断的にアクセスし、OnMessage イベント通知として各ファイル内のデータを順番に読み出し通知する 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。DataImport プロバイダを使用するにあたっては ORiN2SDK をインストールするか、下表を参照して手作業でレジストリ登録を行う必要があります。

表 2-1 DataImport プロバイダ

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

2.1.1. 読み込みフォルダの動作条件

DataImport プロバイダは以下の動作が行われているフォルダを読み込み対象とすることを想定して作成されています。下図はプロバイダを動かしているアプリケーションを ORiN アプリ、それ以外のアプリケーションをその他アプリとしています。

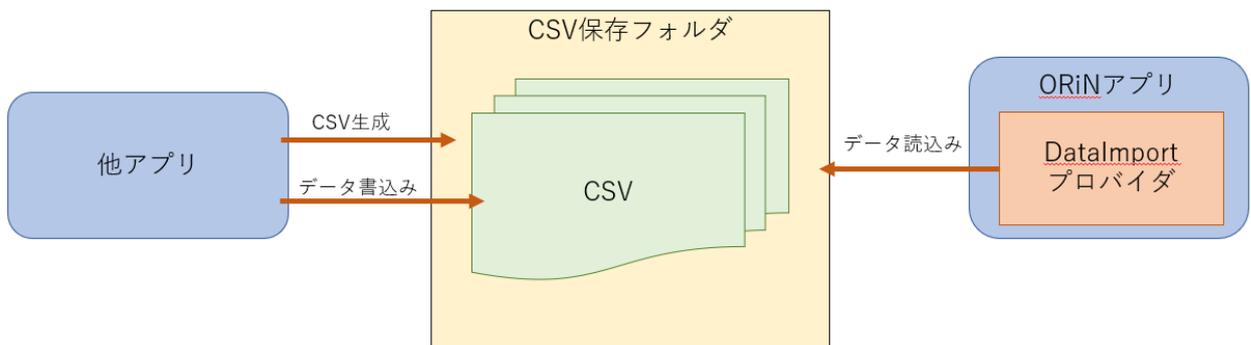


図 2-1 DataImport プロバイダ想定する動作概要図

その他アプリが動的に CSV ファイルを作成し、作成した CSV ファイルにデータを追加しています。ORiN アプリは、CSV 保存フォルダを監視し、CSV ファイルに存在するデータを読み込みます。

以下に、DataImport プロバイダが想定している他アプリの挙動を定義します。

他アプリの挙動

- 1 動的に CSV ファイルが作成されます。ファイル名はルールにより決定されます。(例:プレフィックスが追加される, ファイル名は日付であるなど)
- 2 CSV ファイルには動的にデータが追加されます。
- 3 データが追加される CSV ファイルは、作成されたファイル群では常に 1 つです。

- 4 データの動的追加の停止は他アプリケーション内のルールにより停止されます(例:100 レコード追記で停止など)
- 5 データ追加が停止した CSV ファイルには以降データは追加されません。
- 6 あるファイルへのデータの追加が停止すると、新たに CSV ファイルが作成されます。以降は作成された CSV ファイルへデータが追加されます。
- 7 フォルダ内の CSV ファイルに対して移動/削除/名前の変更が行われません。
- 8 フォルダ内の CSV ファイル内のデータは削除されません。

下図は、上記の動作を表す 1 つの例を示しています。



図 2-2 DataImport プロバイダが監視するフォルダの動作例

あるアプリケーションが、最新の CSV ファイルに、1 秒ごとにデータを書き込みます。アプリケーションは 600 個のデータを CSV ファイルに書き込むと、そのファイルへのデータの書き込みを完了し、次のファイルを作成します。その際のファイル名は、“Data”+”yyyyMMddHHmmss”となります。図中では、書き込み対象ファイルを ■ で、書き込み完了ファイルを ■ で表しています。

2.1.2. プロバイダの基本的な動作

動作開始時

DataImport プロバイダは、動作を開始した際に、指定された読み込み対象フォルダに存在する読出し対象ファイルのリストを作成します(読出しファイルリストと呼称)。読出しファイルリストを元に読出すファイルを決定します。読出しファイルリストの詳細は「2.1.2.1 読出しファイルリスト」を参照ください。

動作終了時

現在の読出し状況をまとめた中断データをファイルとして保存し、次回実行時はその中断データをもとに処理を再開します。中断データの詳細は「2.1.2.3 中断データファイル」を参照ください。

下図は、DataImport プロバイダの基本動作をまとめたフローチャートです。

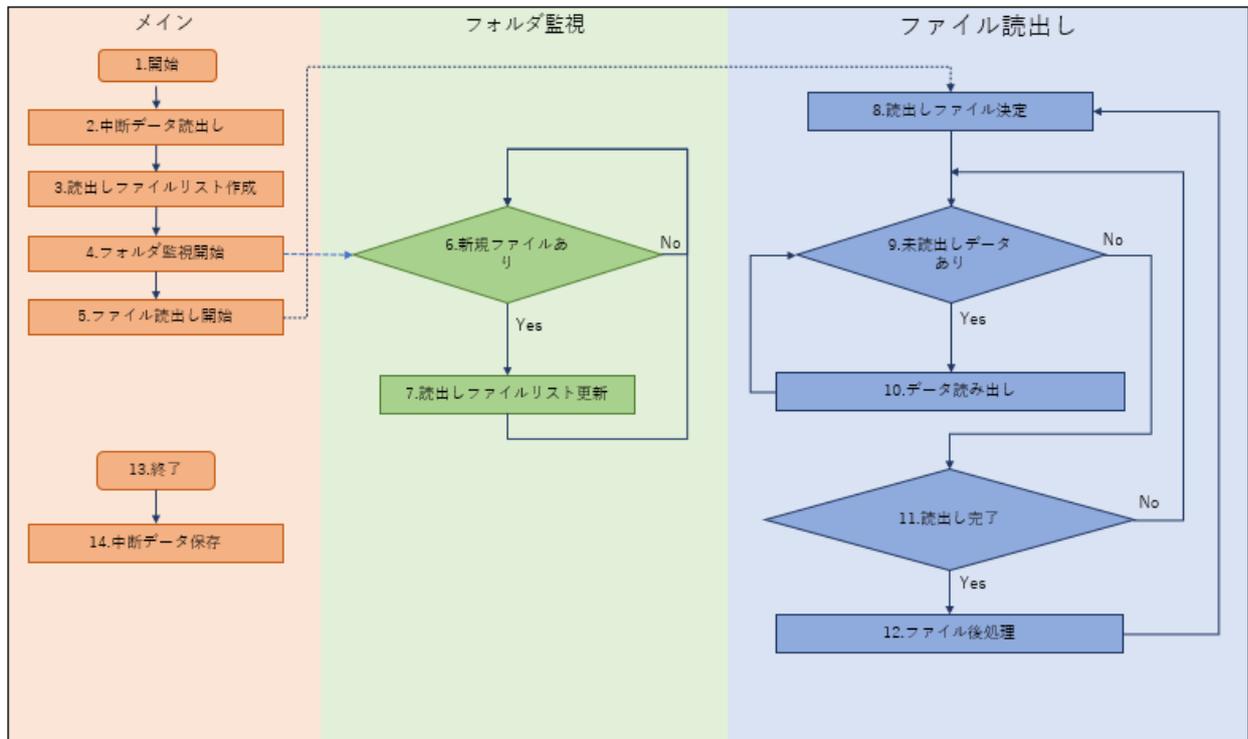


図 2-3 プロバイダの動作概要
実線は同期呼出しを、点線は非同期呼出しを示しています。

表 2-2 処理概要

No	種類	処理名	説明
1	イベント	開始	プロバイダの処理開始イベントを示しています。 「CaoWorkspace::AddController メソッド」の呼び出しが該当します。
2	処理	中断データ読出し	前回終了した際に保存された中断データを読出します。 中断データについては、「2.1.2.3 中断データファイル」を参照してください。

No	種類	処理名	説明
3	処理	読出しファイルリスト作成	中断データを加味し、読出し対象のファイルリストを作成します。DataImport プロバイダはここで作成したファイルリストを元に処理を行います。 ファイル読出し順については、「2.1.2.1.1 ファイルの読出し順」を参照してください。
4	処理	フォルダ監視開始	コントローラ生成時に指定された読み込み対象フォルダの監視を開始します。フォルダの監視は非同期で実行されます。
5	処理	ファイル読出し開始	ファイルの読出し処理を開始します。ファイルの読出しは非同期で実行されます。
6	判断	新規ファイルあり	フォルダ監視スレッドにて、読み込み対象フォルダに新規ファイルが存在するかを監視します。 検出方法については、「2.1.2.1.2 新規読み込みデータの検出（ファイルリストの更新）」を参照してください。
7	処理	読出しファイルリスト更新	新規ファイルを検出すると、読出しファイルリストの末尾に検出したファイルを追加します。 ファイルリスト更新方法については、「2.1.2.1.2 新規読み込みデータの検出」を参照してください。
8	処理	読出しファイル決定	読出しファイルリストの先頭から読出します。
9	判断	未読出しデータあり	ファイルに未読出しデータが存在するかを監視します。 未読出しデータが存在する場合、「データ読出し処理」に進みます。未読出しデータが存在しない場合は、「未読出しファイルあり」判定に進みます。
10	処理	データ読出し	ファイルから1行分のデータを読出し、「未読出しデータあり」判定に戻ります。
11	処理	読出し完了	読出しが完了したかを判定します。 読出し完了時、「ファイル後処理」に進みます。読出し未完了時は、「未読出しデータあり」判定に戻ります。 判定基準については、「2.1.2.2 読出し完了判定」を参照して下さい。
12	処理	ファイル後処理	コントローラ生成時に指定された方法で、ファイルの後処理を行います。 ファイル後処理詳細は、「2.2.1.2 ファイル後処理の種類と指定方法」を参照してください。 ファイル後処理が完了すると、「読出しファイル決定」に戻ります。

No	種類	処理名	説明
13	イベント	終了	プロバイダの処理終了イベントです。 「CaoWorkspace::RemoveController メソッド」の呼び出しが該当します。
14	処理	中断データ保存	現在の読出し状況を中断データとして保存します。 詳細は、「2.1.2.3 中断データファイル」を参照してください。

2.1.2.1. 読出ファイルリスト

2.1.2.1.1. ファイルの読出し順

読出ファイルリストは、コントローラ生成時に指定されたオプションに従ってフィルタリングおよび、ソートされます。データ読出し状態が開始状態になると、このファイルリストの先頭からファイルが読み出され始めます。

注意

- サブフォルダ内を対象時、ソート方法にファイル名を指定している場合、読み込み対象のフォルダからの相対パスが付与されたファイル名でリスト順が決まります。

2.1.2.1.2. 新規読み込みデータの検出(ファイルリストの更新)

読出しファイルリストを生成した後、読み込み対象フォルダを監視し、新規読み込みデータの検出を行います。監視対象は対象フォルダ内のファイルサイズの変化、更新時間の変化です。

新たな読み込み対象のファイルを検出した場合は、ソート設定に関係なく読出しファイルリストの末尾(読出し順の最後尾)に追加されます。複数同時に追加候補が見つかった場合はソートの指定に従った順にファイルリストに追加されます。

2.1.2.2. 読出し完了判定

現在のファイルの末尾まで読出しが完了した際に別の未読出しファイルが存在すると、現在のファイルの読出しを完了し、後処理が実行されます。

現在のファイルの末尾まで読出しが完了していても、読出しファイルリストに、未読出しファイルが存在しない場合、現在のファイルにデータが追加される可能性があるため、後処理は実行されず、ファイル読出しが継続されます。

後処理が実行された後は、ファイルリストも後処理後の状態に更新されます。そのため、後処理前と同名の、異なるファイルが新たに追加された場合は、新規のファイルとみなされます。

2.1.2.3. 中断データファイル

2.1.2.3.1. ファイル構成

中断データファイルには以下の内容が記されています。

表 2-3 中断データ記録ファイルの構成

セクション名	キー名	説明
LastFile	Path	読み込み対象のフォルダパス
	FullPath	最終読み込みファイルのフルパス
	Name	最終読み込みファイル名
	Line	最終読出し行数
	Elem	最終読出し行のデータ要素数
FileFilter	NameFilter	ファイル名フィルタ
TimeFilter	Begin	更新時間フィルタ(絶対時間:開始)
	End	更新時間フィルタ(絶対時間:終了)
	BeginRelative	更新時間フィルタ(相対時間:開始)
	EndRelative	更新時間フィルタ(相対時間:終了)
Sort	Type	ソートタイプ

2.1.2.3.2. 記録

コントローラ解放時に、中断データとして読み込み中のファイル情報、各種フィルタ条件、ソート条件を記録したファイル(コントローラ名+history.ini)を生成します。生成場所は CaoConfig のパラメータ設定で“HistoryPath”オプションで指定します。指定がない場合はプロバイダの DLL と同じフォルダに生成されます。

2.1.2.3.3. 中断データからの再開

次回、同名のコントローラ生成時に、この中断データ記録ファイルが存在し、読出し条件(各種フィルタ、ソート)が同じであり対象のファイルが存在する場合は、前回の続きから読出しを行います。読出し条件が異なっている、あるいは対象のファイルが存在しない場合は、コントローラ生成時に作成した読出しファイルリストの先頭から読出しを行います。

注意

- 再開時に読み込み対象フォルダの状態を初期状態に戻した場合でも、読出し完了済みと判断され、読出し対象にならない場合があります。
- コントローラ名未指定時はランダムなコントローラ名になるため、中断データを利用できません。

2.1.2.3.4. 中断データのリセット

中断データは進行状況をリセットすることも可能です。リセットするためには、Reset コマンドを利用してください。

中断データを正しく残すために、コントローラオブジェクトを解放する際は必ずデータ読み出し状態を停止状態にしてから解放を行うようにして下さい。停止状態にするために、StopReading コマンドを使用して下さい。

データ読み出し状態が開始状態のままコントローラオブジェクトを解放すると解放処理中にデータ読み出し処理が進んでしまい、その結果が中断データとして残ってしまいます。
その結果、再開時に進んでしまった位置からデータ読み出しを開始するためデータが欠落します。

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

DataImport プロバイダは AddController 時にフィルタ条件パラメータを参照し、指定されたフォルダ配下にある読み出し対象のファイルリストを作成します。

また、**コントローラの解放時には StopReading コマンドを実行する必要があります**。(参照:2.2.4.2)



```
AddController(<bstrCtrlName:BSTR>,<bstrProvName:BSTR>,  
              <bstrPCName:BSTR>,<bstrOption:BSTR>))
```

bstrCtrlName : [in] コントローラ名
 bstrProvName : [in] プロバイダ名. 固定値 = "CaoProv.DataImport"
 bstrPcName : [in] プロバイダの実行マシン名
 bstrOption : [in] オプション文字列

以下にオプション文字列に指定するリストを示します。

表 2-4 CaoWorkspace::AddController のオプション文字列

オプション (1)	説明
Path=<フォルダパス> ²	読み込み対象ファイルの格納先フォルダを指定します。フォルダパスは絶対パスで指定する必要があります。(必須)
NameFilter[=<ファイル名フィルタ>]	読み込み対象とするファイル名のフィルタを指定します。 フィルタはワイルドカード(* や ? が使用可能)で指定することができます。 (デフォルト:フィルタ無し)
TimeFilterBegin[=<対象ファイルの更新時間>]	読み込み対象とするファイルの更新時間フィルタ(絶対時間指定:開始時間)を指定します。 (参照:2.2.1.1)
TimeFilterEnd[=<対象ファイルの更新時間>]	読み込み対象とするファイルの更新時間フィルタ(絶対時間指定:終了時間)を指定します。 (参照:2.2.1.1)
TimeFilterBeginRelative[=<対象ファイルの更新時間>]	読み込み対象とするファイルの更新時間フィルタ(相対時間指定:開始時間)を指定します。

¹ 角括弧("[]")内は省略可能を示します。また、各パラメータの解説中の下線部はオプションを指定しなかったときのデフォルト値になります。

² ネットワークドライブを指定した場合、CSV ファイルの追記等の変化を検知ができないことがあります。

	(参照:2.2.1.1)										
TimeFilterEndRelative[=<対象ファイルの更新時間>]	読み込み対象とするファイルの更新時間フィルタ (相対時間指定:終了時間)を指定します. (参照:2.2.1.1)										
Sort[=<ソートタイプ>]	読み込み対象とするファイルの読み込み順番を指定します. (デフォルト:3) <table border="1"> <thead> <tr> <th>値</th> <th>ソートタイプ</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>ファイル名昇順</td> </tr> <tr> <td>2</td> <td>ファイル名降順</td> </tr> <tr> <td>3</td> <td>更新時間昇順</td> </tr> <tr> <td>4</td> <td>更新時間降順</td> </tr> </tbody> </table>	値	ソートタイプ	1	ファイル名昇順	2	ファイル名降順	3	更新時間昇順	4	更新時間降順
値	ソートタイプ										
1	ファイル名昇順										
2	ファイル名降順										
3	更新時間昇順										
4	更新時間降順										
PostProc[=<ファイル後処理タイプ>]	読み込みが完了したファイルの後処理方法を指定します. (デフォルト:1) <table border="1"> <thead> <tr> <th>値</th> <th>ファイル後処理タイプ</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>何もしない</td> </tr> <tr> <td>2</td> <td>リネーム</td> </tr> <tr> <td>3</td> <td>移動</td> </tr> <tr> <td>4</td> <td>削除</td> </tr> </tbody> </table> (参照:2.2.1.2)	値	ファイル後処理タイプ	1	何もしない	2	リネーム	3	移動	4	削除
値	ファイル後処理タイプ										
1	何もしない										
2	リネーム										
3	移動										
4	削除										
PostProcParam[=<パラメータ>]	ファイルの後処理方法毎のパラメータを指定します. (参照:2.2.1.2)										
CSVStartLine[=<読み込み開始行>]	ファイル毎に読み込みを開始する行数を指定します. (デフォルト:1)										
CSVDelimiter[=<区切り文字>]	データの区切り文字を指定します. (デフォルト:1) (参照:2.2.1.3)										
CSVQuotation[=TRUE/FALSE]	データ前後のダブルクォーテーション解釈を行うかどうかを指定します. (デフォルト:FALSE) TRUE:解釈を行う FALSE:解釈を行わない (参照:2.2.1.4)										
CodePage[=<コードページ番号>]	CSV ファイルの文字コードをコードページ番号で指定します. (デフォルト:0) 例1) ANSI Code Page = 0 例2) Shift_JIS = 932										

	例3) UTF-8 = 65001
MessageMax[=<メッセージ最大数>]	<p>データ読み出しイベント通知で生成するメッセージオブジェクトの数を指定します。</p> <p>指定数に通知が達した場合、メッセージオブジェクトが解放され再度通知可能になるまでデータ読み出し処理をブロッキングします。</p> <p>(デフォルト:100)</p>
EnableSubfolder[=TRUE/FALSE]	<p>Path オプションで指定したフォルダ内のサブフォルダ配下を読み込み対象とするかどうかを指定します。(デフォルト:FALSE)</p> <p>TRUE: 対象にする</p> <p>FALSE: 対象にしない</p>
AutoStartDelay[=<自動開始 Delay 時間>]	<p>AddController から何ミリ秒後に読み込み処理を開始するか指定します。(デフォルト:100)</p> <p>-1: 読み込み処理は停止状態です。StartReading コマンドを実行し、任意のタイミングで読み込みを開始します。</p> <p>0 以上: 指定ミリ秒後に読み込みを開始します。指定時間経過前に StartReading コマンドを実行した場合は即座に読み込みを開始します。</p>
SleepTime[=<停止時間>]	<p>監視フォルダの変更を検知した際の停止時間を設定します。(デフォルト:1000)</p>

2.2.1.1. ファイルの更新時間フィルタ

以下にファイルの更新時間フィルタを指定するためのオプションパラメータの詳細を示します。

これらは組み合わせて指定することが可能ですが、絶対時間指定と相対時間指定を混在させて指定した場合は絶対時間指定が優先されます。

オプションを指定しない場合は該当のフィルタ設定なしとして動作します。

[絶対時間指定:開始時間]

“TimeFilterBegin[=<開始時間>]”

<開始時間> : 絶対時間で開始時間を指定します。
指定した時間より更新時間が新しいファイルが読み込み対象となります。
時間は以下の書式で年, 月, 日, 時, 分, 秒を指定します。
書式 : YYYY/MM/DD hh:mm:ss

例) TimeFilterBegin=2016/01/15 12:05:30

- 年月日は”/”(スラッシュ)で区切ります
- 時分秒は”:”(コロン)で区切ります
- 年月日と時分秒の間に半角スペースが必要です

[絶対時間指定:終了時間]

“TimeFilterEnd[=<終了時間>]”

<終了時間> : 絶対時間で終了時間を指定します。
指定した時間より更新時間が古いファイルが読み込み対象となります。
時間は以下の書式で年, 月, 日, 時, 分, 秒を指定します。
書式 : YYYY/MM/DD hh:mm:ss

※ 書式は開始時間と同様です

[相対時間指定:開始時間]

“TimeFilterBeginRelative[=<開始時間>]”

<開始時間> : AddController をした時点からの相対時間で開始時間を指定します。
相対時間は過去方向を正とした 10 進数の数値で秒単位で指定できます。
指定した時間より更新時間が新しいファイルが読み込み対象となります。

例 1) TimeFilterBeginRelative=300

例 2) TimeFilterBeginRelative=-300

- 300 秒= AddController の 5 分前以降に更新されたファイルが対象となります
- マイナスの値を指定した場合は未来の時間を表します
- -300 秒=AddController の 5 分後以降に更新されたファイルが対象となります

[相対時間指定:開始時間]

“TimeFilterEndRelative[=<終了時間>]”

<終了時間> : AddController をした時点からの相対時間で終了時間を指定します。
相対時間は過去方向を正とした 10 進数の数値で秒単位で指定できます。
指定した時間より更新時間が古いファイルが読み込み対象となります。

例 1) TimeFilterEndRelative=300

例 2) TimeFilterEndRelative=-300

- 300 秒= AddController の 5 分前以前に更新されたファイルが対象となります
- マイナスの値を指定した場合は未来の時間を表します
- -300 秒=AddController の 5 分後以前に更新されたファイルが対象となります

■ 組み合わせ指定した場合の例

例1) 絶対時間で開始/終了時間を指定

“TimeFilterBegin=2016/01/15 12:05:30, TimeFilterEnd=2016/01/16 00:00:00”

2016年1月15日12時5分30秒から2016年1月16日0時0分0秒までの間に更新されたファイルが対象。

例2) 相対時間で開始/終了時間を指定

“TimeFilterBeginRelative=3600, TimeFilterEndRelative=300”

AddControllerした時刻の3600秒(=1時間)前から300秒(=5分)前までの間に更新されたファイルが対象。(AddControllerした時刻が13:00とした場合12:00~12:55の間に更新されたファイルが対象となる)

例3) 絶対時間と相対時間を混在させて指定

“TimeFilterBegin=2016/01/15 12:05:30, TimeFilterEndRelative=300”

TimeFilterBeginが指定されているため絶対時間での指定(開始時間指定あり, 終了時間指定なし)が優先され, 相対時間指定のオプションは無効となる。

この場合, 2016年1月15日12時5分30秒以降に更新された全てのファイルが対象となる。

2.2.1.2. ファイル後処理の種類と指定方法

以下にファイル後処理タイプ毎の指定方法を示します。

[何もしない]

“PostProc=1”

1 ファイル読み込み完了後、ファイルに対し何も処理を行いません。

PostProcParam オプションの指定は不要で、指定されている場合は無視されます。

[リネーム]

“PostProc=2, PostProcParam=<変更後拡張子>”

1 ファイル読み込み完了後、ファイルに対し PostProcParam オプションで指定された文字列にファイルの拡張子名をリネームします。

¥/:*? "<>| を文字列中に含むことは出来ません。

拡張子を置き換えた結果ファイル名として不正となってしまうような文字が指定されていた場合、後処理は失敗します。

リネーム後と同名のファイルが存在していた場合、後処理は失敗します(上書きはされません)。

[移動]

“PostProc=3, PostProcParam=<移動先フォルダパス>”

1 ファイル読み込み完了後、ファイルを PostProcParam オプションで指定されたフォルダパスに移動します。

フォルダパスは絶対パスで指定し、存在するフォルダを指定する必要があります。(自動でフォルダの生成は行いません)

移動先のフォルダに同名のファイルが存在していた場合、後処理は失敗します(上書きはされません)。

例) ファイル:c:¥foo¥hoge.csv, 移動先フォルダパス:c:¥bar

c:¥foo¥hoge.csv → c:¥bar¥hoge.csv

[削除]

“PostProc=4”

1 ファイル読み込み完了後、ファイルを削除します。

PostProcParam オプションの指定は不要で、指定されている場合は無視されます。

2.2.1.3. データ区切り文字

以下にデータ区切り文字の指定方法を示します。

“CSVDelimiter[=<区切り文字>]”

データ区切り文字は数値または文字の何れかで指定します。(デフォルト 1)

■ 数値で指定する場合

- 1 : カンマ区切り
- 2 : タブ区切り

■ 文字で指定する場合

データ区切り文字で指定したい文字(1 キャラクタ)をダブルクォーテーションで括って指定します。
ダブルクォーテーション自体をデータ区切り文字として指定することはできません。

[指定例]

例1) カンマ区切り(数値指定)

“CSVDelimiter=1”

例2) カンマ区切り(文字指定)

“CSVDelimiter=(“,”)”

例3) タブ区切り(数値指定)

“CSVDelimiter=2”

例4) スペース区切り(文字指定)

“CSVDelimiter=(“ ”)”

例5) スラッシュ区切り(文字指定)

“CSVDelimiter=(“/”)”

2.2.1.4. ダブルクォーテーションで括られたデータの解釈

ダブルクォーテーションで括られたデータの解釈のされ方について、カンマ区切りを例に以下に示します。

(³)

[csv として見た場合]

	1	2	3
1	AAAA	BBBB	CCCC
2	1111	22 33	4444
3	“ABCD”	EF”GH	55,66

[txt として見た場合]

1	AAAA, BBBB, CCCC
2	1111, “22
3	33”, 4444
4	“””ABCD”””, “EF””GH”, “55,66”

[取得データ]

<ダブルクォーテーション解釈なし>

1 行目	2 行目	3 行目	4 行目
[0] AAAA	[0] 1111	[0] 33”	[0] “””ABCD”””
[1] BBBB	[1] “22	[1] 4444	[1] “EF””GH”
[2] CCCC			[2] “55
			[3] 66”

<ダブルクォーテーション解釈あり>

1 行目	2 行目	3 行目
[0] AAAA	[0] 1111	[0] “ABCD”
[1] BBBB	[1] 22(¥r¥n)33	[1] EF”GH
[2] CCCC	[2] 4444	[2] 55,66

※ (¥r¥n) 改行コード

³ タブ区切りの場合も区切り文字が異なるだけで同等の解釈が成されます。

※

2.2.2. CaoController::AddVariable メソッド

CaoController クラスの AddVariable メソッドは、変数オブジェクトを作成するためのメソッドです。

書式 AddVariable(<bstrVariableName:VT_BSTR>[,<bstrOption:VT_BSTR>])

<bstrVariableName> : [in] 変数名
<bstrOption> : [in] オプション文字列

2.2.3. CaoVariable:get_Value プロパティ

変数オブジェクトの値を取得します。

2.2.4. CaoController::Execute

CaoController クラスの Execute メソッドは、コマンドを実行するためのメソッドです。

書式 [<vntRet:VARIANT> =] Execute(<bstrCmd:BSTR > [,<vntParam:VARIANT>])

<bstrCmd> : [in] コマンド名
<vntParam> : [in] パラメータ

以下に指定可能なコマンドの一覧を示します。

表 2-5 コマンド一覧

コマンド	機能	パラメータ
StartReading	データ読み込み開始	なし
StopReading	データ読み込み停止	なし
Reset	データ読み出し位置のリセット	<p><自動開始 Delay 時間> (VT_I4)</p> <p>Reset から何ミリ秒後に読み込み処理を開始するか指定します。</p> <p>-1: 読み込み処理は停止状態で完了します。</p> <p>StartReading コマンドを実行し、任意のタイミングで読み込みを開始します。</p> <p>0 以上: 指定ミリ秒後に読み込みを開始します。</p> <p>指定時間経過前に StartReading コマンドを実行した場合は即座に読み込みを開始します。</p>

2.2.4.1. StartReading コマンド

データ読み出し状態を開始状態に変更します。

データ読み出し状態が開始状態になると順次対象ファイルから各オプション指定に従ったデータを1行分読み出し OnMessage イベントで読み出しデータ通知(参照:2.2.5.2)を発行します。

読み出し対象のデータがなくなった, 又はメッセージオブジェクト数がオプション指定の最大数に達した場合, データ読み出し処理及び読み出しデータ通知は一時停止しブロッキングされます。読み出し対象のデータが追加された, 又はメッセージオブジェクトが解放され通知可能になった場合, データ読み出し処理及び読み出しデータ通知は自動的に再開します。

2.2.4.2. StopReading コマンド

データ読み出し状態を停止状態に変更します。

再度 StartReading コマンドを実行することで読み出しを再開することが出来ます。

2.2.4.3. Reset コマンド

データ読み出し位置をリセットします。

中断データを無視して再度頭から読み出し直すことができます。

2.2.5. CaoController::OnMessage イベント

以下の契機で CaoController クラスの OnMessage イベントが発生します。

表 2-6 メッセージ種別

メッセージ種別		発生契機
0	エラーログ	内部エラー検出時に INFO レベルでログ出力します。 (データフォーマット異常検出, ファイル後処理失敗, etc...)
1	読み出しデータ通知	各オプション指定に従って解析した 1 行分のデータを 1 データずつ分割し通知します。 メッセージ読み出し状態が開始状態の際に順次通知されます。(メッセージオブジェクトが指定数に達している場合読み出し処理はブロッキングされます)
2	データ構造変化通知	データを読み出した際に直前のデータと比較して 1 行辺りのデータ要素数が異なる際に通知します。

2.2.5.1. エラーログ

エラーログメッセージで得られるデータ形式を以下に示します。

Number : メッセージ種別 (0)
 Value : エラー内容 (VT_BSTR)
 DateTime : タイムスタンプ
 Destination : Null
 Source : Null
 Description : Null

2.2.5.2. 読み出しデータ通知

読み出しデータ通知メッセージで得られるデータ形式を以下に示します。

Number : メッセージ種別 (1)
 Value : 読み出した 1 行分のデータ(VT_BSTR | VT_ARRAY)
 DateTime : タイムスタンプ
 Destination : Null
 Source : ファイル名⁴
 Description : Null

⁴ サブフォルダ配下のファイルの場合は Path オプションで指定したフォルダからの相対パスがファイル名の前に付与されます。

2.2.5.3. データ構造変化通知

データ構造変化通知メッセージで得られるデータ形式を以下に示します。

Number	:	メッセージ種別 (2)
Value	:	新データ要素数 (VT_I4)
DateTime	:	タイムスタンプ
Destination	:	Null
Source	:	Null
Description	:	Null

2.3. 変数一覧

2.3.1. CaoController クラス

表 2-7 CaoController クラス ユーザ変数一覧

変数名	データ型	説明	属性	
			get	put
—	—	非対応	—	—

表 2-8 CaoController 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	プロバイダバージョン情報.	○	—

2.4. エラーコード

DataImport プロバイダでは、固有のエラーコードはありません。

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。