

拡張機能付き XML→CSV 高性能ツール概要説明 (xceditor V1.0)

[機能]

- ・ XML データを CSV データに変換する高性能ツール。

[特徴]

- ・ 画面で簡単にマッピング機能
- ・ 自動化のバッチ処理機能
- ・ 拡張機能のスク립トで複雑条件の抽出が可能。

[動作環境]

- ・ Windows XP/Vista/Win7

[アンインストール]

- ・ そのまま削除して下さい。

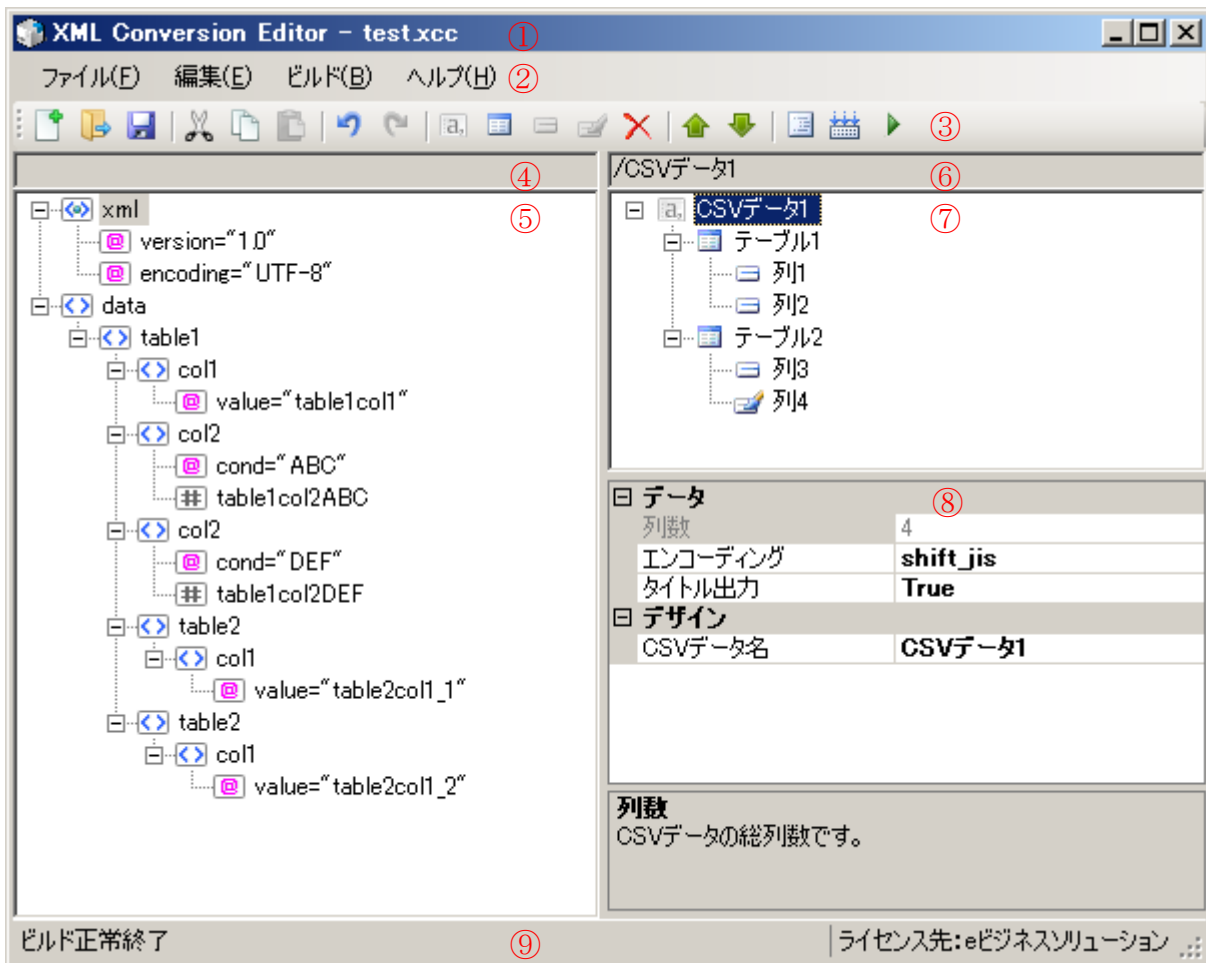
[体験版の制限事項]

- ・ 変換設定ファイルの最大 CSV データ数 : 1
- ・ 変換設定ファイルの最大列数 : 8
- ・ ユーザー定義の共通関数編集機能 : なし
- ・ アドイン関数機能 : なし

XML Conversion Editor (xcditor.exe) の説明

XML Conversion Editor は、XML 変換設定ファイルの作成・編集・テスト実行用ツールです。

1. メイン画面説明



①タイトルバー

「XML Conversion Editor」のツール名前と作業中 XML 変換設定ファイルの名前を表示します。

②メニューバー

・ファイルメニュー

- 新規作成 : 新規に XML 変換設定ファイルの編集を開始します。
つまり、「XML Conversion Editor」起動直後の空白状態に戻します。
- 開く : ファイル参照ダイアログから、既存の XML 変換設定ファイルを選択して開きます。
- 上書き保存 : 現在の編集データを現在のファイル名で保存します。
保存先が存在しない場合は名前を付けて保存します。
- 名前を付けて保存 : ファイル保存ダイアログから、現在の編集データを任意の名前で保存します。
- 終了 : 「XML Conversion Editor」を終了します。

・編集メニュー

- 元に戻す : 直前の操作を取り消します。つまり、作業履歴をひとつ戻ります。
- やり直し : 取り消した操作を再実行します。つまり、作業履歴をひとつ進みます。
- 切り取り : 選択されたオブジェクトを切り取ります。
「コピーして、かつその場から削除する」という操作です。
- コピー : 選択状態のオブジェクトをします。

(クリップボードにコピーします。)

- 貼り付け : 一時的に記憶しているオブジェクトを変換ファイル表示エリアに追加します。
- 追加 : 指定種別の新規オブジェクトを変換ファイル表示エリアに追加します。
- 削除 : 選択状態のオブジェクトを削除します。
- 上移 : 同じ階層で選択状態のオブジェクトを上に移動します。
- 下移 : 同じ階層で選択状態のオブジェクトを下に移動します。
- 名前の変更 : 選択状態のオブジェクトの名前を変更します。

サンプル XML ファイルの指定 : ファイル参照ダイアログから、サンプル XML ファイルを選択して、XML 表示エリアにファイルの内容を表示します。

共通関数編集 : 共通関数編集画面を表示して、共通関数のスクリプトを編集します。

・ビルドメニュー

変換設定ファイルのビルド : 編集中的変換設定ファイルのエラーチェックをします。

エラーなしの場合に変換設定スクリプトを作成します。

テスト実行 : 変換設定スクリプトを実行して、XML ファイルの変換結果を確認します。

・ヘルプメニュー

XML Conversion Editor のバージョン情報 :

XML Conversion Editor のバージョン情報画面を表示します。

ライセンス登録 : ライセンス登録画面を表示して、ライセンス情報を入力します。

③ ツールバー




使用頻度が高いと思われるメニュー操作のアイコンを、横一列に並べたバーです。

④ サンプル XML のノード文字列

XML 表示エリアに選択されたノードの文字列を表示します。(表示のみ)

⑤ XML 表示エリア

サンプル XML ファイルの内容を表示します。

アイコン	XML 型	文字列書式	作成項目 ※1	備考
	要素	/親要素名/……/<要素名	テーブル	
	属性	/親要素名/……/@属性名	列	
	テキスト	/親要素名/……/#	列	CDATA もテキストと認識します。





※1. 選択されたノードを「変換ファイル表示エリア」にドラッグ&ドロップすると、「テーブル」／「列」の自動作成ができます。

⑥ 変換ファイルの項目文字列

変換ファイル表示エリアに選択された項目の階層文字列を表示します。(表示のみ)

⑦ 変換ファイル表示エリア

変換ファイルの内容を表示します。

アイコン	項目型	文字列書式	備考
	CSV データ	/CSV データ名	CSV ファイルに対応します。
	テーブル	/CSV データ名/テーブル名	列・計算列の集合です。
	列	/CSV データ名/テーブル名/列名	XML 属性・テキスト値を取得します。
	計算列	/CSV データ名/テーブル名/計算列名	取得した各列値より計算をする結果です。

⑧ 変換ファイルの項目属性編集エリア

変換ファイル表示エリアに選択された項目の属性の表示・編集用エリアです。

・ CSV データ

列数 : CSV データに所属するテーブルの列数（計算列も含む）合計です。（表示のみ）

エンコーディング : 出力 CSV ファイルの文字集合の種別コードです。（デフォルト : shift_jis）

常用のコード :

shift_jis 日本語（シフト JIS）

euc-jp 日本語（EUC）

utf-8 Unicode（UTF-8）

タイトル出力 : CSV を出力する時、タイトル（列名）を出力するかどうかを示します。

CSV データ名 : CSV データを識別するコードで使われる名前です。

バッチファイルを作成する時、出力対象のファイル名として使用します。

・ テーブル

列数 : 所属する列数（計算列も含む）の合計です。（表示のみ）

取得元 : テーブルのレコードに対応する要素の文字列を指定します。

テーブル名 : テーブルを識別するコードで使われる名前です。

・ 列

取得元 : 列情報に対応する属性（テキスト）の文字列を指定します。

列情報の取得元は、必ずテーブル取得元のサブノードでなければなりません。

取得条件 : 列情報取得の付加条件を指定します。指定しない場合は無条件に取得する。

出力フラグ : CSV ファイルに出力するかどうかを制御します。

出力しない列は他の計算列の構成要素として存在する場合があります。

出力順 : CSV ファイルに列の出力順番を指定します。

出力順の値が同じの場合、列の入力順に従って出力します。

文字種別 : 使用できる文字種別を指定します。

変換する時、文字種別以外の文字を取得する場合、エラーファイル(*.err)に報告を出力します。

選択項は下記です :

制限しない	全角文字	全角カタカナ	全角英数字
半角文字	半角カタカナ	半角英数字	半角数字

最大文字数 : 列の最大文字数を指定します。

変換する時、最大文字数を超える文字列を取得する場合、

エラーファイル(*.err)に報告を出力します。

※半角文字数です。全角 1 文字の「文字数」は 2 です。

※最大文字数=0 の場合、チェックをしません。

固定長 : 最大文字数以外を許可するかどうかを示します。

変換する時、最大文字数以外の文字列を取得する場合、

エラーファイル(*.err)に報告を出力します。

※最大文字数=0 の場合、チェックをしません。

入力必須 : 入力必須であるかどうかを示します。

変換する時、空白の文字列を取得する場合、エラーファイル(*.err)に報告を出力します。

列チェック関数 : 列情報の合理性をチェックする関数です。

戻り値はエラーコード (0:正常 1:エラー) とエラー文字列です。

エラーコード=1 の場合、エラーファイル(*.err)にエラー文字列を出力します。

列名 : 列を識別するコードで使われる名前です。

CSV 出力するとき、列名はタイトルとして出力することができます。

※CSV データに所属する列の列名を重複して設定することができません。

・ 計算列

基本的に列の属性と同じですが、「取得元」と「取得条件」の属性がありません。

代わりに、「計算関数」の属性があります。

計算関数 : 計算列の値の計算する関数です。

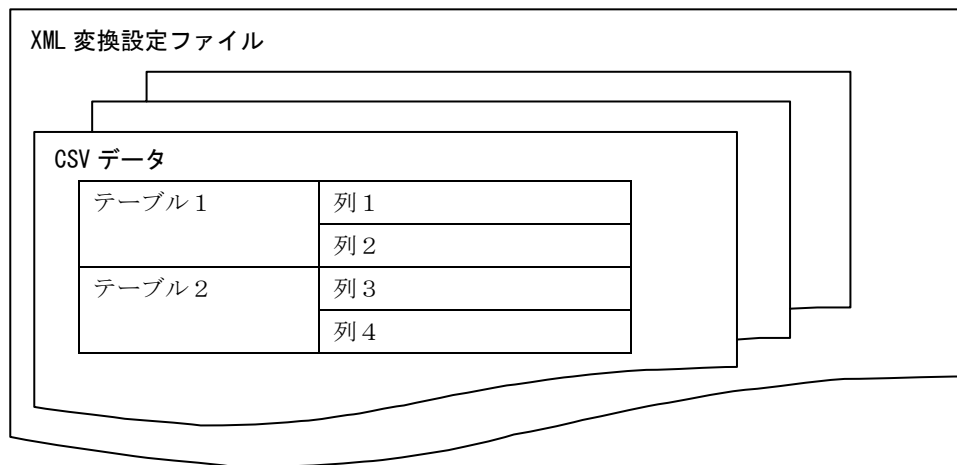
⑨ステータスバー

処理結果とライセンス先を表示します。

2. XML 変換設定ファイルの構成

特定の XML ファイルからデータを抽出する為に、「XML 変換設定ファイル」を作成します。

XML 変換設定ファイル(*.xcc)は、下図のように構成するものです。



①CSV データ

一つ CSV ファイルに対応します。CSV データの中に、複数の「テーブル」を含んでいます。

※基本的に、一つ CSV データに一つテーブルをセットしますが、複数テーブルを含む場合もあります。

※複数テーブルの関係は、「ヘッダ／明細／明細の明細／……」です。

テーブルを結合して CSV に出力することです。

②テーブル

XML 文書では、リレーショナルデータベースのテーブルのように、あるひとかたまりのデータが繰り返し記述されるケースが少なくありません。例えば、以下のような書籍情報一覧 (books.xml) を見てください。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
```

```
<store storename="X X 書店">
```

```
<book level="1">
```

```
<title>X X X 入門教室</title>
```

```
<author>Y. Yamada</author>
```

```
<price>2800</price>
```

```
</book>
```

```
<book level="2">
```

```
<title>サーバサイドプログラミング徹底比較</title>
```

```
<author>H. Takano</author>
```

```
<price>1880</price>
```

```
</book>
```

```
</store>
```

元のデータとなる books.xml は<store>ノードをルートノードとした XML 文書で、各書籍の情報を<book>ノードで表現する、典型的な繰り返し型 XML 文書です。個々の<book>ノード配下には、書籍名を表す<title>ノードや著者名を表す<author>ノードなどが含まれます。

この例に、二つテーブルが作成できます：

1. 書店テーブル（一つレコードしかありませんが）

取得元 : /<store

列定義 : 店名 (/store/@storename)

2. 書籍テーブル（二つレコードがあります）

取得元 : /store/<book

列定義 : 書籍名 (/store/book/title/#)

著者名 (/store/book/author/#)

売単価 (/store/book/price/#)

レベル (/store/book/@level)

変換結果は下記です：

店名	書籍名	著者名	売単価	レベル
X X 書店	X X X 入門教室	Y. Yamada	2800	1
X X 書店	サーバサイドプログラミング徹底比較	H. Takano	1880	2

③列

1. 通常列と計算列の二つ種類の列があります。

通常列は直接に XML 文書のノード（属性・テキスト）内容から取得するのです。

計算列は取得済みの通常列の値より計算（計算関数で）して取得するのです。

2. 通常列を取得する時、「取得条件」を設置することが可能です。

例えば、上記の XML 文書について、「レベル 1 の書籍の売単価を非表示にする」という要求がある場合、「売単価」列に下記の条件を設置すると対応できます：

条件ノード文字列	操作符	常量
/store/book/@level	≠	'1'

3. 下記の例で計算列の運用方法を説明します：

要望：「表示用書籍名」という列を追加したいです。

レベル 1 の表示用書籍名：書籍名_①

レベル 2 の表示用書籍名：書籍名_②

.....

※レベル範囲：1～3

対応方法：「表示用書籍名」の計算列を追加します：

計算関数のソースは下記です：

```
if |レベル| == '1' then
    return |書籍名| .. '_①'
elseif |レベル| == '2' then
    return |書籍名| .. '_②'
else
    return |書籍名| .. '_③'
end
```

※「|列名|」の書き方は LUA 文法ではありませんが、XML Conversion Editor でビルドを行う時、本番の列変数名に変換することになります。

※関数名の定義がありませんが、XML Conversion Editor でビルドを行う時、本番の関数名を自動的に追加します。

4. 下記の例で列のチェック関数の使用方法を説明します：

要望 : 「売単価」 < 0 の場合、「マイナスの売単価です」のエラーを出力したいです。

対応方法 : 「売単価」列に列チェック関数を指定します :

列チェック関数のソースは下記です :

```
if StrToInt(|売単価|) < 0 then
    return 1, 'マイナスの売単価です。'
else
    return 0, ''
end
```

XMLConv コマンドライン

XML の変換はコマンドラインで行うことができます。

コマンドラインの書式は下記のようにです：

XmlConv. exe 変換スクリプトファイル (*. xcs)

XML ファイル (*. xml)

CSV ファイル (*. csv)

[スキーマファイル (*. xsd)]

[エラーファイル (*. err)]

[変換対象 CSVID]

パラメータの順番は任意です（ファイル拡張子で判断するのです）。

1. 変換スクリプトファイル (*. xcs)

XML 変換スクリプトファイル (*. xcs) は LUA 言語のテキストファイルです。

XML Conversion Editor で XML 変換設定ファイル (*. xcc) をビルドする時に、同じフォルダに同名の XML 変換スクリプトファイルを作成します。

任意のテキストエディタで XML 変換スクリプトファイルを修正することは可能ですが、スクリプトの正確性は自分で保証しなければなりません。

2. XML ファイル (*. xml)

変換元 XML ファイル名です。

3. CSV ファイル (*. csv)

変換先 CSV ファイル名です。

4. スキーマファイル (*. xsd)

省略可能のパラメータです。

XML 妥当性検証の必要がある場合、スキーマファイル名を指定します。

妥当性検証の結果は「エラーファイル」に出力します。

5. エラーファイル (*. err)

省略可能のパラメータです。

省略する場合、出力されたエラーファイルの名前 (拡張子の含まない) CSV ファイルの名前と同じです。

エラーファイルは変換エラー/妥当性検証エラーの出力先ファイル (CSV 書式) です。

書式は下記のようにです：

XML 行	XML 列	CSV 行	CSV 列	エラーメッセージ	エラー種別

XML 行・列 : エラー発生 of XML ファイル場所

CSV 行・列 : エラー発生 of CSV ファイル場所

エラーメッセージ : エラー情報の内容

エラー種別 : 1.. エラー 2.. 警告

6. 変換対象 CSVID

省略可能の整数パラメータです (1~)。

省略する場合、1 件目の CSV データ (変換設定ファイルに定義された) の変換を行います。

関数集

function ShowMsg(msg)

指定したテキストを表示するメッセージ ボックスを表示します。

パラメータ :

msg: メッセージ文字列

戻り値 :

なし

function WriteLine(msg)

指定した文字列値を標準出力ストリームに書き込み、続けて現在の行終端記号を書き込みます。

パラメータ :

msg: 出力文字列

戻り値 :

なし

function IsValidFileName(FileName)

指定したファイル名の有効性をチェックします。

パラメータ :

FileName: ファイル名

戻り値 :

true.. 有効 false.. 無効

function FileExist(FileName)

指定したファイルが存在するかを確認します。

パラメータ :

FileName: ファイル名

戻り値 :

true.. 存在 false.. 不存在

function StrToInt(str, DefaultValue)

指定した文字列を整数に変換します。

パラメータ :

str: 変換対象文字列

DefaultValue: str が有効な整数を表していない場合, DefaultValue を返します。

戻り値 :

変換結果の整数

function StrSub(str, StartIndex, len)

インスタンスから部分文字列を取得します。

パラメータ :

str: 対象文字列

StartIndex: 開始インデックス(1~)

len: 指定した文字数

戻り値 :

部分文字列

```
function StrFormat(format, value)
```

指定した書式で数値を文字列に変換します。

パラメータ :

format : 変換書式

value : 変換対象数値

戻り値 :

変換結果文字列

変換書式についてより詳しくは、MSDN の「[標準の数値書式指定文字列](#)」をご覧ください。

```
function RegEx(str, pattern)
```

正規表現と一致する対象が入力文字列内で見つかったかどうかを示します。

パラメータ :

str : 対象文字列

pattern : 正規表現文字列

戻り値 :

true.. 一致 false.. 不一致

```
function StrBLen(str)
```

指定した文字列の byte 文字数を取得します。

全角 1 文字の byte 文字数は 2 です。

パラメータ :

str : 対象文字列

戻り値 :

Byte 文字数

```
function StrLen(str)
```

指定した文字列の文字数を取得します。

全角 1 文字も半角 1 文字も、文字数は 1 です。

パラメータ :

str : 対象文字列

戻り値 :

Byte 文字数

※XML 変換ツールの正式版には、「アドイン関数機能」があります。お客様自分でアドインライブラリを作成して、変換スクリプトファイルに使用することが可能です。

アドインライブラリのサンプルとして、正式版に「xcaddin.WindowsAPI.dll」のソースファイル WindowsAPI.lzh があります。

アドインライブラリの名前 : xcaddin.XXXXXXXX.dll