

LANDSCAPE スクリプト・リファレンス・マニュアル

森 洋久

joshua @ globalbase.org

2009-12-19 版

目次

第 1 章	はじめに	2
1.1	目的と概要	2
1.2	このマニュアルを読むために必要な知識	2
1.3	前提となるシステム要件	2
1.4	サーバ操作の注意事項	2
第 2 章	LANDSCAPE スクリプト・リファレンス	3
2.1	概要	3
2.2	継承情報	3
2.3	スクリプト	4
2.3.1	launch-xl(サーバの起動)	4
2.3.2	makefile.xl(コンテンツの登録)	5
2.3.3	p4m.xl(4点以上のマッピングの作成)	6
2.3.4	matrix.xl(マトリックス型画像フォーマット処理)	15
2.3.5	worldfile-mtx.xl(ワールドファイル・マトリックス・コンバータ)	21
2.3.6	import-images.xl(個別画像ファイルの変換)	23
2.3.7	utm2bl-mtx.xl(UTM 画像を楕円体緯度経度座標へ変換する)	25
2.3.8	worldfile.xl(ワールドファイルの変換)	27
2.3.9	wfutils.xl(ワールドファイル関連ツール)	29
2.3.10	prog.xl(進行状況表示)	33
2.3.11	csv-plot.xl(CSV プロットデータのマトリックスへの変換)	34
第 3 章	worldfile.xl スクリプト用 CSV コマンド・リファレンス	39
3.1	概要	39
3.2	継承情報	39
3.3	スクリプト	40
3.3.1	#(コメント)	40
3.3.2	.character(文字種決定)	41
3.3.3	.pre.string(前置)	42
3.3.4	.post.string(後置)	43
3.3.5	.qualifier(クオリファイヤー指定)	44
3.3.6	.qtype(データ型指定)	45
3.3.7	.qdata(データ指定)	46
3.3.8	.wf.mapping(マッピング先指定)	47
3.3.9	.wf.processing(ファイル名の列の指定)	48

第1章 はじめに

1.1 目的と概要

LANDSCAPE の各種機能はサーバ付属の各種スクリプトによって実現されている。特に、画像データなどの公開用データへの変換等の基本的操作について説明する。この他にも、古いマニュアル (<http://>) のコンテンツ作成の章も利用可能である。今後、この古いマニュアルも順次、当該マニュアルへ繰り込んで行く予定である。

1.2 このマニュアルを読むために必要な知識

このマニュアルを読む前に、LANDSCAPE サーバがインストールされている環境が必要です。インストールの方法は、「LANDSCAPE スタートアップ・マニュアル [1]」を参照してください。

XL スクリプトを詳細に操れる必要はありませんが、参考程度に理解していることをお勧めします。xl スクリプト [2][UNDEF REF (protocol)] を参照してください。また、GLOBALBASE の座標系の検索、重ね合わせの原理について概略をわかっている必要があります。これらの技術的理解については、GLOBALBASE 技術資料 [UNDEF REF (GLOBALBASE-tech)] に書かれています。手短に GLOBALBASE の機能を理解するためには、COSMOS GLOBALBASE VIEWER [3] を動かしてみることをお勧めします。

1.3 前提となるシステム要件

POSIX 系マシンが必要です。これに、LANDSCAPE GLOBALBASE VIEWER [1] をインストールすることによって xlsv の機能が利用可能となります。

1.4 サーバ操作の注意事項

LANDSCAPE GLOBALBASE サーバは、インストール後、`launch-xl` (2.3.1 節)(MacOSX 以外) や `XLServer` (2.3.1 節) で起動、終了します。このマニュアルで説明されるコマンドの中には、LANDSCAPE GLOBALBASE サーバが起動しているときに有効なものが含まれています。各項目にその旨書かれていますので、サーバが起動しているかどうかを確かめて利用してください。

第2章 LANDSCAPE スクリプト・リファレンス

2.1 概要

このマニュアルで説明されるコマンドの中には、LANDSCAPE GLOBALBASE サーバが起動しているときに有効なものが含まれています。各項目にその旨書かれていますので、サーバが起動しているかどうかを確かめて利用してください。

環境の情報にエージェントが指定されている場合、エラー表示が可能です。たとえば、

```
xl matrix.xl / import-pnm [pnm-filename] [mtx-filename]
```

と表示されているスクリプトであれば、

```
xl matrix.xl - - / import-pnm [pnm-filename] [mtx-filename]
```

とすることによりエラー表示が可能です。

2.2 継承情報

各スクリプトコマンドは何らかのエージェントの機能を利用しています。エージェントの項目も参照してください。

2.3 スクリプト

2.3.1 launch-xl(サーバの起動)

プロトタイプ

```
/etc/init.d/launch-xl start
```

```
/etc/init.d/launch-xl stop
```

```
/Library/StartupItems/XLServer/XLServer start ( MacOSX)
```

```
/Library/StartupItems/XLServer/XLServer stop ( MacOSX)
```

引数

start LANDSCAPE サーバの起動

stop LANDSCAPE サーバの停止

環境

シェルコマンド

説明

LANDSCAPE サーバの起動、停止を行います。基本的に、各 OS の起動スクリプトのフォーマットに準拠しています。またサーバのインストール直後はハードウェア起動時に自動的に LANDSCAPE サーバも起動するように設定されます、この自動起動の on/off については各 OS のマニュアルを参照してください。

戻り値

なし。

エラー

参考

バグ

2.3.2 makefile.xl(コンテンツの登録)

プロトタイプ

xl makefile.xl

引数

環境

エージェント xl [2]

LANDSCAPE が起動している必要あり。

説明

LANDSCAPE サーバ xldocs の下にコンテンツを格納した時点、およびコンテンツを削除した時点で、コンテンツの登録、削除を行うコマンドです。コンテンツを置くだけでこのコマンドを発行しなかった場合はコンテンツの配信は正常に行われません。

利用手順は、

1. コンテンツを、xldocs の下に格納する。
2. cd コマンド等で、コンテンツを格納したディレクトリへ移動
3. コマンドを発行。

```
xl makefile.xl
```

xldocs ディレクトリ直下でこのコマンドを発行した場合、あるいは、/usr/local/xl-gbs ディレクトリ直下でこのコマンドを発行した場合は、xldocs 配下のすべてのコンテンツに対して登録を行います。

戻り値

なし。

エラー

参考

バグ

xldocs の直下にあるコンテンツに対して正常にはたらい来ません。かならず一つディレクトリを作り、その中にコンテンツを置くことで回避することができます。

2.3.3 p4m.xl(4点以上のマッピングの作成)

プロトタイプ

xl p4m.xl / [p4m-file]

引数

[p4m-file] point-map と meta 情報だけのマッピングファイル

オプション

- reverse= none/src/dest デフォルトは none

ver.B.b16.05 以降のサポート

環境

エージェント xl [2], エージェント netmapper [UNDEF REF (netmapper)]

説明

4点以上の対応点をもったマッピングでは、1次変換で座標を対応させることができません。そのため、全体を複数の三角形の領域 (TIN) に分割し、おのおのの TIN 内部で異なる一次変換を適用する補完方式を行います。また、座標系の無限遠点でも変換出来るように、近似的に無限遠転用の一次変換を指定する必要があります。これらのパラメータを手で指定し、マッピングファイルを作ることも可能ではありますが、これを対応点を与えると自動的に生成するスクリプトがこの p4m.xl です。

ver.B.b16.05 以降の LANDSCAPE では、reverse オプションが新たに加われました。マッピング元とマッピング先の軸の方向が異なる場合、reverse オプションが必要です。コンピュータ座標系 (x 軸右方向、y 軸下向き) を基準にして、数学座標系 (x 軸右方向 y 軸上向き) の座標系を reverse オプションで指定します。例えば、マッピングもとが数学座標系であれば、reverse=src と指定します。マッピング先が数学座標系であれば reverse=dest と指定します。もし、両方とも数学座標系の場合は reverse=none で問題ありません。

まず、拡張子 p4m をもったファイルを作ります。基本的には、拡張子 map 以外であればかまいません。p4m の構造は map ファイルの構造と同じです。point-map と meta 情報のみを記述したものです。いわば、3点までの map ファイルの4番目以降の point-map をそのまま付け加えたものと考えてよいでしょう。従って、すでに3点までマッピングしたマッピングファイルがあり、それに4点目を追加しようと考えているときには、そのまま4点目を追加し、ファイルの拡張子を.p4m に書き換えれば、p4m ファイルは完成です。以下の例は1系 (九州) のための p4m ファイルの中身です。(01.p4m)

```
<?xml version="1.0" encoding="EUC-JP"?>
<map>
  <meta>
    <file type="xl"/>
    <src>
      xlp://isjhp2.nichibun.ac.jp:8080/kokudo/coord/01.crd
    </src>
    <dest>
      xlp://isjhp2.nichibun.ac.jp:8080/kokudo20000/coord/00.crd
    </dest>
    <dp>
      181
    </dp>
  </meta>
  <point-map>
```

```
b1
<list>
6577.435059m
30896.992188m
</list>
<list>
146482.000000m
1294111.625000m
</list>
</point-map>
<point-map>
b2
<list>
-16864.001953m
34593.703125m
</list>
<list>
149568.734375m
1315266.750000m
</list>
</point-map>
<point-map>
b3
<list>
-20587.638672m
81065.656250m
</list>
<list>
189254.359375m
1318867.375000m
</list>
</point-map>
<point-map>
b4
<list>
3998.532959m
12131.631836m
</list>
<list>
130398.726563m
1296403.500000m
</list>
</point-map>
<point-map>
b5
<list>
```

```

-38258.191406m
24350.640625m
</list>
<list>
140765.75000m
1334532.000000m
</list>
</point-map>
<point-map>
b6
<list>
-15451.980469m
56954.250000m
</list>
<list>
168696.531250m
1314098.375000m
</list>
</point-map>
</map>

```

この内容を 01.p4m というファイルにセーブし、このスクリプトを適用します。

```
% x1 p4m.x1 - - / 01.p4m
```

すると拡張子のみが.map に変化した、01.map が生成されます。TIN や無限遠点一次変換が指定された流通可能なマッピングファイルです。中身は以下のようになります。

```

<map>
<meta>
<file type="x1"/>
<src>
xlp://isjhp2.nichibun.ac.jp:8080/kokudo/coord/01.crd
</src>
<dest>
xlp://isjhp2.nichibun.ac.jp:8080/kokudo20000/coord/00.crd
</dest>
<dp>
181
</dp>
</meta>
<horizontal-affen>
<List>
<List>

```

```
0.002363
0.854563
</List>
<List>
-0.901667
0.004435
</List>
</List>
<List>
120044.569714
1299925.616167
</List>
</horizontal-affen>
<point-map>
^"1"
<list>
-127929.444336
-125736.416992
</list>
<list>
12292.601562
1414717.750000
</list>
</point-map>
<point-map>
^"2"
<list>
-38258.191406m
24350.640625m
</list>
<list>
140765.750000m
1334532.000000m
</list>
</point-map>
<point-map>
^"3"
<list>
-127929.444336
218933.705078
</list>
<list>
306835.000000
1416246.250000
</list>
</point-map>
```

```

<triangle-map>
  ^"1"
  ^"2"
  ^"3"
</triangle-map>
<triangle-map>
  ^"2"
  ^"3"
  ^"1"
</triangle-map>
<triangle-map>
  ^"3"
  ^"2"
  ^"1"
</triangle-map>
<point-map>
  ^"4"
  <list>
    6577.435059m
    30896.992188m
  </list>
  <list>
    146482.000000m
    1294111.625000m
  </list>
</point-map>
<point-map>
  ^"5"
  <list>
    96248.687988
    218933.705078
  </list>
  <list>
    307364.656250
    1214112.375000
  </list>
</point-map>
<point-map>
  ^"6"
  <list>
    96248.687988
    -125736.416992
  </list>
  <list>
    12822.257812
    1212583.750000

```

```

</list>
</point-map>
<triangle-map>
  ^"4"
  ^"5"
  ^"6"
</triangle-map>
<triangle-map>
  ^"6"
  ^"4"
  ^"5"
</triangle-map>
<triangle-map>
  ^"5"
  ^"4"
  ^"6"
</triangle-map>
<point-map>
  ^"7"
  <list>
    3998.532959m
    12131.631836m
  </list>
  <list>
    130398.726563m
    1296403.500000m
  </list>
</point-map>
<triangle-map>
  ^"1"
  ^"7"
  ^"6"
</triangle-map>
<triangle-map>
  ^"7"
  ^"6"
  ^"1"
</triangle-map>
<triangle-map>
  ^"6"
  ^"7"
  ^"1"
</triangle-map>
<point-map>
  ^"8"
  <list>

```

```
-20587.638672m
81065.656250m
</list>
<list>
  189254.359375m
  1318867.375000m
</list>
</point-map>
<triangle-map>
  ^"5"
  ^"8"
  ^"3"
</triangle-map>
<triangle-map>
  ^"3"
  ^"8"
  ^"5"
</triangle-map>
<triangle-map>
  ^"8"
  ^"3"
  ^"5"
</triangle-map>
<triangle-map>
  ^"2"
  ^"8"
  ^"3"
</triangle-map>
<triangle-map>
  ^"3"
  ^"2"
  ^"8"
</triangle-map>
<triangle-map>
  ^"5"
  ^"8"
  ^"4"
</triangle-map>
<triangle-map>
  ^"8"
  ^"4"
  ^"5"
</triangle-map>
<triangle-map>
  ^"1"
  ^"2"
```

```

    ^"7"
  </triangle-map>
  <triangle-map>
    ^"2"
    ^"7"
    ^"1"
  </triangle-map>
  <triangle-map>
    ^"7"
    ^"4"
    ^"6"
  </triangle-map>
  <triangle-map>
    ^"6"
    ^"7"
    ^"4"
  </triangle-map>
  <point-map>
    ^"9"
    <list>
      -15451.980469m
      56954.250000m
    </list>
    <list>
      168696.531250m
      1314098.375000m
    </list>
  </point-map>
  <triangle-map>
    ^"9"
    ^"2"
    ^"8"
  </triangle-map>
  <triangle-map>
    ^"9"
    ^"8"
    ^"4"
  </triangle-map>
  <point-map>
    ^"10"
    <list>
      -16864.001953m
      34593.703125m
    </list>
    <list>
      149568.734375m

```

```
1315266.750000m
</list>
</point-map>
<triangle-map>
  ^"2"
  ^"7"
  ^"10"
</triangle-map>
<triangle-map>
  ^"9"
  ^"2"
  ^"10"
</triangle-map>
<triangle-map>
  ^"9"
  ^"10"
  ^"4"
</triangle-map>
<triangle-map>
  ^"7"
  ^"10"
  ^"4"
</triangle-map>
</map>
```

このファイルをサーバに認識させれば、マッピングは完成です。

```
% xl makefile.xl - -
```

戻り値

なし。

エラー

参考

バグ

2.3.4 matrix.xl(マトリックス型画像フォーマット処理)

プロトタイプ

```
xl matrix.xl / mtz-status [mtx-filename]
xl matrix.xl - - / pnm-status [pnm-filename]
xl matrix.xl / create-mtx [mtx-filename] [width] [height]
xl matrix.xl / import-pnm [pnm-filename] [mtx-filename]
xl matrix.xl / export-pnm [mtx-filename] [pnm-filename] splitmode=all
xl matrix.xl / export-pnm [mtx-filename] [pnm-filename] [x-offset] [y-offset] [width] [height]
splitmode=part
xl matrix.xl / export-pnm [mtx-filename] [pnm-filename] [x-divide-nos] [y-divide-nos] split-
mode=split
xl matrix.xl / import-pnm-over [pnm-filename] [mtx-filename] [x-offset] [y-offset]
xl matrix.xl / import-dted1 [target-dir] [dted1-filepath] [bib]
xl matrix.xl / block [mtx-filename] [jpg-filename] [level] [x] [y]
xl matrix.xl / copy-mtx [mtx-src-filename] [mtx-dest-filename] [channelNo.] ...
xl matrix.xl / scan-network [mtx-filename] standard [start-warp-point-file] [end-point-x] [end-
point-y]
xl matrix.xl / scan-network-offset [mtx-filename] standard [start-warp-point-file] [end-point-x]
[end-point-y] [dim-code-level] [dim-code-x] [dim-code-y]
xl matrix.xl - - / leveling-mtx [mtx-filename]
```

引数

[mtx-filename] マトリックスファイル名
[pnm-filename] PNM ファイル名
[jpg-filename] JPEG ファイル名
[dted1-filepath] DTED1 形式ファイルのファイルパス
[bib] crd ファイルの bib 要素を抜き出した bib ファイル
[width] 画像の横幅、ピクセル数
[height] 画像の縦幅 ピクセル数
[x-offset] 画像の保存開始位置。マトリックスファイルの x 方向のオフセットピクセル数
[y-offset] 画像の保存開始位置。マトリックスファイルの y 方向のオフセットピクセル数
[x-divide-nos] export-pnm 用 画像の x 方向の分割個数。
[y-divide-nos] export-pnm 用 画像の y 方向の分割個数。
[level] 画像の階層レベル
[x] 画像の座標位置 マトリックスファイルの x 座標
[y] 画像の座標位置 マトリックスファイルの y 座標
[mtx-src-filename] コピー元マトリックスファイル名
[mtx-dest-filename] コピー先マトリックスファイル名
[channelNo.] チャンネル番号または「COSMOS リファレンス・マニュアル」[4]の「チャンネルフラグ(マトリックス)」
[start-warp-point-file] ワープポイントデータを保存したファイル。ワープポイントデータとは、COSMOSにて「編集」「場所情報をコピー」で得られるテキストデータ「COSMOS リファレンス・マニュアル」[4]の「編集メニュー」。
[end-point-x] スキャンデータのおわりの X 座標。 [start-warp-point-file] で指定されるワープポイントデータのベース座標系の座標値。

[end-point-y] スキャンデータのおわりの Y 座標。 [start-warp-point-file] で指定されるワープポイントデータのベース座標系の座標値。

オプション

- **create-mtx** におけるオプション
default_color= [default_color] デフォルト 0xffffffff
- **import-pnm-over** におけるオプション
transparent= [transparent_color] デフォルト 0xffffffff
leveling= on/off デフォルト on
- **scan-network** におけるオプション
meta= [bib ファイル名]
leveling= [on/off] デフォルト off
- **export-pnm** におけるオプション
splitmode= [all/part/split] デフォルト all
level= 整数 デフォルト 0

環境

エージェント xl [2], gbm [5]gbview [6]

説明

マトリックスに関連した操作を行うスクリプト。現在対象としているのは、8 bitRGB 型の画像のみである。最初の引数が操作を表す。以下のように定義されている。

- **mtx-status**

与えられたマトリックスファイルのステータス情報を返します。詳細はマトリックスフォーマット仕様書を参照してください。ステータスは XL スクリプトの戻り値となるため、標準出力を有効にしておく必要があります。

```
% xl matrix.xl -- / mtx-status
```

と -- を入れるのはそのためです。

- **pnm-status**

与えられた PNM ファイルのステータス情報を返します。詳細は netpbm のマニュアルを参照してください。

- **create-mtx**

与えられた大きさ (ピクセルサイズ) 空のマトリックスファイルを生成します。参照するとすべて透明な画像と等価です。

default_color オプションには、書き込み前に概念上書き込まれていると仮定されている色を指定します。RGBA の 32bit の値です。A=00 が不透明。A=FF が透明。

- **import-pnm**

与えられた PNM ファイルを与えられたファイル名のマトリックスファイルに変換します。ピクセル数の大きさは同じ画像です。

- **export-pnm**

ver.B.b17.01 以前のバージョンでは、与えられたマトリックスファイルを与えられたファイル名の PPM ファイルに変換します。ピクセル数の大きさは同じ画像です。ver.B.b17.02 以降のバージョンでは、他に様々な大きさの画像を出力する方法を追加しました。それは以下の通りです。

ver.B.b16.14 より、level オプションを指定すると、マトリックス中各階層のデータを変換することが可能です。デフォルトは level=0 です。マトリックスの各階層のピクセル数は、マトリックスの設定により異なります。

与えられたマトリックスファイルを与えられたファイル名の PPM ファイルに変換します。書式は三つあり、入力ファイルと出力ファイルのみを指定すると、入力のマトリックスファイルのピクセル数の同じ大きさ画像が出力されます。splitmode=all を指定してもかまいません。level オプションを指定すると、任意のレベルの画像が出力できます。

入力ファイルと出力ファイルの他に、[x-offset], [y-offset], [width], [height] の4つ値を指定し、splitmode=part を指定すると、与えられたオフセットから、[width] [height] の大きさの画像を切り出します。level オプションの使用は可能です。また、splitmode オプションは省略可能です。

入力ファイルと出力ファイルの他に、[x-divide-nos], [y-divide-nos], splitmode=split を指定すると、x 方向に [x-divide-nos] 分割、y 方向に [y-divide-nos] 分割の画像を出力します。画像ファイルには、[pnm-filename]- [x 方向番号]- [y 方向番号].ppm という名前が振られるので、[pnm-filename] には括弧を指定してはいけません。split-mode=on オプションは省略することは出来ません。

- **import-pnm-over**

現存するマトリックスファイル([mtx-filename])の一部に、与えた PNM ファイル([pnm-filename])を埋め込みます。埋め込む位置は、[x-offset], [y-offset] で与えます。

transparent オプションには、書き込まない色を指定します。RGB の 24bit です。つまり、transparent で指定された色の部分はもともと書き込まれている色となります。このオプションを指定しないと、すべての色を上書きします。

leveling=off オプションは、ver.B.b16.14 で有効なオプションで、これを指定すると、階層化を行いません。インポートをまとめて行い、最後に階層化を行うときに便利です。

- **import-dted1**

DTED1 形式ファイルを読み込み、16bit 符号付き(2の補数表現)整数の2次元配列のマトリックスファイルと、それを参照する.crdをはじめとしたオブジェクトファイルを生成します。たとえば、以下のようなコマンドを実行します。

```
x1 matrix.x1 -- / import-dted1 test 'area01/dted/*/*.dted1' bib.x1
```

[dted1-filepath] は一連の DTED1 形式ファイルが収まっているディレクトリまでのパスとファイル名までを含めたファイルパスで指定します。特に DTED1 形式では、ファイル名とファイルを含むディレクトリ名は緯度経度を表しているため、この情報が欠かせません。ファイルパスで指定しないとエラーとなります。また、上記例では緯度経度に相当する部分にワイルドカードを使って複数のファイルを指定しています。

[bib] に相当する bib.x1 は worldfile-mtx.x1 などと同じ形式の書誌情報ファイルです。公開する情報の書誌情報を書き込みます。例えば、以下のような形です。

```
<?xml version="0.1" encoding="EUC-JP"?>
<bib xmlns:gb="xlp://isjhp1.nichibun.ac.jp:8080/gb_metadata">
  <gb:title type="text" data="Japan Ortho"/>
  <gb:creator type="text" data="Osaka City University"/>
  <gb:content.period type="W3C-DTF" data="2003-01-01 / 2003-01-01"/>
  <gb:issue.period type="W3C-DTF" data="2006-06-01"/>
  <gb:property type="gb-prop" data="photo"/>
  <gb:homepage type="URL" data="http://www.osaka-cu.ac.jp"/>
</bib>
```

- **block**

マトリックスファイル [mtx-filename] の、与えた座標 ([level] [x] [y]) の位置のノード JPEG 画像を取り出します。取り出した画像は、 [jpg-filename] へセーブします。

- **copy-mtx**

マトリックスファイル [mtx-src-filename] と同一のヘッダ情報をもったマトリックスファイル [mtx-dest-filename] を生成し、 [mtx-src-filename] 内のすべてのノードについて、 [channelNo.] で与えられたチャンネルのみをコピーします。 [channelNo.] は複数指定可能です。

生成したマトリックスファイルから、送信のみのデータを含んだマトリックスファイルを生成したいときには、

```
% x1 matrix.xml -- / copy-mtx srcfile.mtx destfile.mtx MF_SEND MF_SEND_VISU MF_SEND_FILE
```

とするとよい。また、現在、matrix_RGB8.xml で定義しているチャンネルは、

- 8 送信用ブロック (jpeg 圧縮)
- 9 非圧縮 RGB 画像

- **scan-network**

ネットワーク上の地図をスキャンし [mtx-filename] で指定した名前一枚のマトリックス画像にする。スキャンの開始位置は、 [start-warp-point-file] に示されるファイルに保存したワープポイントデータである。ワープポイントデータは以下のような形をしており、COSMOS の「編集」「場所情報をコピー」「COSMOS リファレンス・マニュアル」[4] の「編集メニュー」で得ることが出来る。COSMOS でブラウズし、スキャンしたい画像を表示したあと、スキャンしたい解像度でスキャン開始位置をズームインし、このメニューを選択し、新しいファイルにペーストすれば、必要なワープポイントデータが得られる。

```
<warp-point>
  <tracking-time>197748854sec</tracking-time>
  <title>China</title>
  <type>2</type>
  <query><OR>
    <query qtype="URL" title="test-C118" id="0" active="off">
      <URL>xlp://gbs2.itakura.toyo.ac.jp:8080/wakashima/C118/C118.crd</URL>
```

```

</query>
<query qtype="property" id="0" active="off">
  <qualifier cond="part">xlp://isjhp1.nichibun.ac.jp:8080/gb_metadata 0 property
</query>
<query qtype="property" title="考古学" id="0" active="off">
  <AND>
    <qualifier cond="part">xlp://isjhp1.nichibun.ac.jp:8080/gb_metadata 0 prope
    <qualifier cond="boundary">xlp://isjhp1.nichibun.ac.jp:8080/gb_metadata 0 c
  </AND>
</query>
.....
</OR>
</query>
<resolution>2.500000 </resolution>
<rotate>0.000000 </rotate>
<center>130.000000 0.000000 </center>
<base>xlp://localhost:8080/test/China.crd</base>
<layers>
  <entry>xlp://localhost:8080/test/China.crd 256</entry>
</layers>
</warp-point>

```

matrix.xml スクリプトは、ワーブポイントデータを読み込み、ベース座標系を、base 要素で指定される座標系にセットする。この座標系上で、ワーブポイントデータの center よそで指定される座標から、resolution 要素で与えられる解像度でスキャンを開始し、[end-point-x], [end-point-y] で指定される座標でスキャンを停止する。[end-point-x], [end-point-y] は、center 要素の座標値より大きくなければならない。また、rotate は無視され、x 軸方向、y 軸方向と平行にスキャンを行う。

query 要素は、座標系の検索条件を指定するもので、この条件にあった座標系がすべて重ねられる。つまり複数の座標系をつなぎ合わせた大きな座標系をスキャンしたい場合によい。

[mtx-filename] で指定されるマトリックスファイルには、スキャン開始位置が、マトリックスの (0,0) の位置に保存され、スキャン終了位置が、マトリックスの X,Y 両ピクセル最大値の位置に保存される。なにもオプションがない場合は、セーブされたマトリックスは階層化されていないことに注意。この時点でつなぎ合わされた画像を export-pnm でデータを pnm に変換することは出来る。逆に、階層化の時間を節約してつなぎ合わせられた画像をつくる事が出来る。

階層化を行う場合には、最初にオプションとして、leveling=on を指定するか、または、オプションなしの実行のあと、matrix.xml の leveling-mtx を利用して階層化を行う事が出来る。階層化を行うとネットワークで公開可能なマトリックスを作ることができる。

meta= [メタデータファイル名] を指定すると、crd,map,lst ファイルを生成し、作業後、makefile.xml (2.3.2 節) を実行すればすぐに公開可能なすべての情報を生成する。meta オプションを指定すると、leveling は自動的に on になる。以下にメタデータファイル (bib ファイル) の例をあげる。

```

<?xml version="0.1" encoding="EUC-JP"?>
<bib xmlns:gb="xlp://isjhp1.nichibun.ac.jp:8080/gb_metadata">

```

```
<gb:title type="text" data="Japan Ortho"/>
<gb:creator type="text" data="Osaka City University"/>
<gb:content.period type="W3C-DTF" data="2003-01-01 / 2003-01-01"/>
<gb:issue.period type="W3C-DTF" data="2006-06-01"/>
<gb:property type="gb-prop" data="photo"/>
<gb:homepage type="URL" data="http://www.osaka-cu.ac.jp"/>
</bib>
```

- **scan-network-offset**

ネットワーク上の地図をスキャンし [mtx-filename] で指定した名前の一枚のマトリックス画像にする。ただし、scan-network と異なるのは、スキャンしたデータを引数最後に指定する「COSMOS リファレンス・マニュアル」 [4] の「次元コード (マトリックス)」の位置から保存する。

- **leveling-mtx**

ver.B.b16.14 からのサポートです。与えられたマトリックス [mtx-filename] を階層化します。たとえば、import-pnmなどで、階層化なしでマトリックスを合成した場合、その結果をあとでまとめて階層化するためなどで使います。

マトリックス型フォーマットはロングファイルに対応しています。ロングフォーマットを利用する場合はOSおよびファイルシステムにおいてロングフォーマットが許可されている必要があります。

今後、GLOBALBASEの初期の段階で採用していた、階層型ラスタフォーマットである r64/cr フォーマットおよび、階層型ベクタフォーマットである pdbp フォーマットは徐々にマトリックス型フォーマットへと統合して行きます。

戻り値

エラー

参考

バグ

2.3.5 worldfile-mtx.xl(ワールドファイル・マトリックス・コンバータ)

プロトタイプ

```
exl worldfile-mtx.xl / [image-path] [worldfile-prefix] [destination-directory] [base-unit] [bib-file]
  引数
```

[image-path] イメージファイルの指定。
[worldfile-prefix] ワールドファイルの拡張子
[destination-directory] 保存先のディレクトリ
[base-unit] 座標系ファイルの単位
[bib-file] 書誌情報の記述されたファイル

オプション

- **bgcolor=** [background-color] デフォルト 0xffffffff
- **backup=on,off,recover** デフォルト off
- **backuploop=** [backuploop-count] デフォルト 1
- **add=on,off,info** デフォルト off
- **leveling=on,off** デフォルト on

環境

エージェント exl [UNDEF REF (exl)], エージェント gbmex [5]

説明

ワールドファイルをマトリックスフォーマットに変換し、閲覧可能なように座標系ファイル、マッピングファイルを生成します。

[image-path] はイメージファイルの指定ですが、ワイルドカードを使うことができます。ワイルドカードを使うときには、シェルがそれを認識しないように”で囲う必要があります。[worldfile-prefix] は「.」から付けてください。[base-unit] は座標系の単位の指定で、ワールドファイルの仮定している座標系の単位と一致します。[bib-file] は座標系ファイルに埋め込む bib 情報です。

```
% xl worldfile-mtx - - / '*/image/*.bmp' .bpw image m image/bib.xl
```

などします。image/bib.xl の中身の例は以下の通り。

```
<?xl version="0.1" encoding="EUC-JP"?>
<bib xmlns:gb="xlp://isjhp1.nichibun.ac.jp:8080/gb_metadata">
  <gb:title type="text" data="Japan Ortho"/>
  <gb:creator type="text" data="Osaka City University"/>
  <gb:content.period type="W3C-DTF" data="2003-01-01 / 2003-01-01"/>
  <gb:issue.period type="W3C-DTF" data="2006-06-01"/>
  <gb:property type="gb-prop" data="photo"/>
  <gb:homepage type="URL" data="http://www.osaka-cu.ac.jp"/>
</bib>
```

worldfile-mtx は大きく分けて二つのフェーズになっています。

1. 第一フェーズ

与えられた画像ファイルとワールドファイルをスキャンして、その配置場所、大きさから全体の座標系の大きさを決め、img.crd,img.map,img.lst ファイルを生成します。

2. 第二フェーズ

次に、画像ファイルを実際に読み込み、マトリックスファイル img.mtx に保存して行きます。

上記、生成されるファイル名は、img.crd,img.map,img.lst,img.mtx と決まっているので、保存場所は空のディレクトリである必要があります。

bgcolor オプション

bgcolor オプションを指定すると、mtx の default_color を $RGBA = 256 * bgcolor + 255$ を指定し、pnm を書き込むときには、transparent=bgcolor とします。指定しないと bgcolor=0xffff と等価です。

backup オプション

backup は、第二フェーズにおいて、img.mtx のバックアップを採るかどうかを指定します。off を指定するとバックアップをとりません。on を指定すると定期的にバックアップを img.mtx.b に採って行き、同時に backup-list.xml にバックアップ情報を保存します。バックアップをとる頻度は、backuploop オプションで指定します。backup=recover を指定すると、第一フェーズを飛ばし、backup-list.xml を読み込み、最後にバックアップをとったところから作業を開始します。しかしそのときに、img.mtx.b はあらかじめ img.mtx へコピーしておく必要があります。

backup=recover は、バックアップを採ります。バックアップを採らずに行う recover は存在しません。

backuploop オプション

backuploop は、画像ファイルの img.mtx への書き込み処理の、何回に一回バックアップをとるかを指定するオプションです。backuploop=1 以下を指定すると、毎回バックアップをとることになります。

異常終了、強制終了した場合は、作成中のデスティネーションフォルダーの中の img.mtx は破壊している可能性があるため、バックアップファイル img.mtx.b をコピーし復旧させる必要があります。

add オプション

add オプションは以前に、worldfile-mtx で作成した既にあるマトリックスに対して、新たに画像データを追加（上書き）するための機能を提供します。add=info を指定すると、第一フェーズを実行し、まず上書きするために必要な情報を取得し、backup-list.xml へ保存します。もし、以前の作業で backup オプションを指定し、すでに backup-list.xml がある場合、add=info を実行する必要はありません。次に、add=on を指定すると、第一フェーズを飛ばし、backup-list.xml を読み込み、第二フェーズから実行を開始し、上書き作業を行います。

leveling オプション

ver.B.b16.14 以降で有効なオプションです。leveling オプションを off にすると、階層化を行いません。合成した画像を、pnm で書き出したい場合などに余計な作業を行わないために有効です。

戻り値

- [distination-directory]/イメージファイル。
- backup-list.xml バックアップ情報ファイル。

エラー

参考

バグ

ver.B.b15 時点で、座標軸に平行にマッピングされるワールドファイルのみサポートしています。座標軸に対してななめである場合は、動作を保証しません。

ver.B.b16.12 で、worldfile-mtx.xl スクリプトがインストーラに入っていないことが判明。ver.B.b16.13 より解決する。

2.3.6 import-images.xl(個別画像ファイルの変換)

プロトタイプ

`xl import-images.xl / [image-format] [path-to-images] [destination-directory] [bib-file]`

引数

[image-format] 変換後のイメージフォーマット r64 または mtX が指定可能。

[path-to-images] 変換元のイメージファイルへのパス。ワイルドカードが利用可能。

[destination-directory] 保存先のディレクトリ

[bib-file] 書誌情報の記述されたファイル

オプション

- process=all,wp デフォルトは all

環境

エージェント exl [UNDEF REF (exl)]

説明

tif, jpeg, bmp のファイルを GLOBALBASE で閲覧可能な形式へ変換します。 .crd, .map, .lst ファイルも同時に生成し、そのまま閲覧可能な状態で保存します。

また、画像ファイルを保存するだけでなく、保存したファイルを参照した、ワーポイントメニューファイルも生成します。このファイルを COSMOS のワーポイントメニューから参照すると、生成した画像の座標系がリストとしてワーポイントメニューに追加出来ます。

以下に [bib-file] の内容の例をあげます。

```
<?xl version="0.1" encoding="EUC-JP"?>
<bib xmlns:gb="xlp://isjhp1.nichibun.ac.jp:8080/gb_metadata">
  <gb:title type="text" data="Japan Ortho"/>
  <gb:creator type="text" data="Osaka City University"/>
  <gb:content.period type="W3C-DTF" data="2003-01-01 / 2003-01-01"/>
  <gb:issue.period type="W3C-DTF" data="2006-06-01"/>
  <gb:property type="gb-prop" data="photo"/>
  <gb:homepage type="URL" data="http://www.osaka-cu.ac.jp"/>
</bib>
```

[bib-file] を用意し、以下のように実行すると、 [destination-directory]=kochizu1 に画像座標系ファイルも生成します。

```
% xl import-images.xl - - / r64 '/home/gbs/data/nichibun/kochizu1/2813459.tif' kochizu1 bib.xl
```

ファイルフォーマットとして、r64 を選ぶか、mtX を選ぶかは、無圧縮画像ファイルで 4Gbyte を越えない場合、r64、越える場合は mtX を選ぶという目安があります。r64 フォーマットは 4Gbyte を越えることはできませんが、圧縮率が高く、また、画像転送容量が小さいフォーマットです。一方、mtX は画像転送容量は r64 に比べて大きいですが、ファイルの大きさが 4Gbyte より大きくとれる特徴があります。今後 mtX ファイルの画像転送容量や性能、機能も向上してくとみられ、徐々に mtX 一本へとシフトしてくと考えられます。

ワーブポイントメニューファイルは、このスクリプトを実行したディレクトリに、wp.xl というファイル名で保存されます。各画像座標系への URL はこの wp.xl の位置から計算されるので、このスクリプトを実行するディレクトリの配下に [destination-directory] があるのが望ましいです。

オプションの process については、process=all のときはすべてのプロセスを実行します。process=wp のときは、ワーブポイントメニューの生成のみを行います。

戻り値

- [distination-directory]/イメージファイル。
- wp.xl ワーブポイントメニューファイル。

エラー

参考

バグ

ver.B.b16.10 以前のバージョンにおいて、mtx フォーマットではエラーが出るバグがある。ver.B.b16.11 以降を利用のこと。

2.3.7 utm2bl-mtx.xl(UTM 画像を楕円体緯度経度座標へ変換する)

プロトタイプ

```
xl utm2bl-mtx.xl / [dest-mtx] [longitude-min] [longitude-max] [latitude-min] [latitude-max]
[longitude-resolution] [latitude-resolution] [src-directory] [elipsoid-a] [elipsoid-b]
```

引数

[dest-mtx] 変換先のマトリックスファイル名。
[longitude-min] 作成する bl の範囲、経度の最小値 -180degree ~ 180degree
[longitude-max] 作成する bl の範囲、経度の最大値 -180degree ~ 540degree
[latitude-min] 作成する bl の範囲、緯度の最小値 -90degree ~ 90degree
[latitude-max] 作成する bl の範囲、緯度の最大値 -90degree ~ 90degree
[longitude-resolution] 経度増加方向の解像度。dot/degree
[latitude-resolution] 緯度増加方向の解像度。dot/degree
[src-directory] 変換元のディレクトリ
[elipsoid-a] 楕円体長径 (m)
[elipsoid-b] 楕円体短径 (m)

オプション

- leveling=on,off デフォルトは on
- part-step= [整数] デフォルトは 1
- part-offset= [整数] デフォルトは 1

環境

エージェント xl [2]

説明

UTM 参照系によるマトリックスデータを回転楕円体緯度経度座標系に変換する。

[longitude-min] [longitude-max] は返還後の画像の経度方向の範囲をしめす。日付変更線を含んだ範囲へ変換することも可能とするため、[longitude-max] は最大 540 までとれる。しかし、[longitude-max]-[longitude-min] は 360 度を超えることは出来ない。同様に、[latitude-min] [latitude-max] は緯度方向の範囲を示す。

[longitude-resolution] [latitude-resolution] はそれぞれ経度方向、緯度方向の解像度を与える。高緯度地域と低緯度地域を同一の解像度では、経度方向の解像度を同一で処理するのは望ましくない。高緯度では、経度方向の解像度を下げるべきである。このため、通常は、マトリックスファイルを南高緯度部分、赤道を中心とした部分、北高緯度部分の三つの地域にわけてそれぞれ [latitude-resolution] を変更して変換するのが望ましい。

[src-directory] は変換元の UTM データのあるディレクトリをしめす。UTM データは各 UTM のゾーンごとに、ゾーン名のサブディレクトリに入っている必要がある。このサブディレクトリの中身は、img.crd, img.map, img.lst, img.mtx からなり、これが UTM 座標に準拠している必要がある。ただし、サブディレクトリの名前は、[ゾーン番号] または、[ゾーン番号]S, [elipsoid-a]s, [elipsoid-a]N, [elipsoid-a]n という形をしている必要がある。数字末尾にアルファベットなし、または、N,n がついているものは北半球のゾーンである。一方、S,s がついている者は南半球のゾーンを示している。

[elipsoid-a] [elipsoid-b] は回転楕円体の長径と短径である。UTM の依拠する標準楕円体を調べ、設定する。

オプション leveling=off では最後の階層化を行いません。何回かに分けて utm2bl-mtx.xl を実行し、最後にまとめて階層化する場合にこのオプションを利用することが可能です。

オプション、part-step,part-offset はすべての UTM を取り込むのではなく、いくつかごとの UTM を取り込む場合に利用します。part-step=2 とすると、1つ飛ばしに UTM を変換します。part-offset=3 とすると、UTM3 から、変換を始めます。一般に極地部分では、UTM 座標系の歪みが少ないため、北緯 60 度以上は、偶数 UTM のデータは奇数番号の UTM へ結合したデータになっているといった場合に、利用可能です。

変換方法

UTM データが複数のファイルの世界ファイルデータであたえられている場合について、変換方法を簡単に述べる。

1. ワールドファイルデータを UTM のゾーンごとにまとめる。
2. ワールドファイルをゾーンごとに、worldfile-mtx.xl (2.3.5 節) スクリプトを利用し、一枚のマトリックスデータファイルセットに変換する。そのときの変換先のディレクトリは、上記所定のサブディレクトリ名になるようにする。

出来上がった UTM データは閲覧することが出来る必要はない。このコンバートに leveling=off オプションをつけると、最後のデータの階層化作業を行わないので、処理時間短縮になる。

3. 最後に、当該スクリプトを使い一つのデータに変換する。

```
[geosage]$ ls
52 54 53n b1
[geosage]$ x1 utm2b1-mtx.xl - - / b1/img0.mtx -180 180 -60 60 3600 3600 . 6378137 6356752
....
[geosage]$ x1 utm2b1-mtx.xl - - / b1/img1.mtx -180 180 60 90 1800 3600 . 6378137 6356752
....
[geosage]$ x1 utm2b1-mtx.xl - - / b1/img2.mtx -180 180 -90 -60 1800 3600 . 6378137 6356752
....
[geosage]$ ls b1
img0.mtx
[geosage]$
```

4. 最後に、crd ファイルを一つ作り、さらに、map ファイル、lst ファイルを、img0.mtx ~ img2.mtx にあわせて作成し、先の crd ファイルにすべてマッピングする。

戻り値

- [dest-mtx] マトリックスファイル

エラー

参考

バグ

2.3.8 worldfile.xl(ワールドファイルの変換)

プロトタイプ

```
xl worldfile.xl / direct [prefix] [unit] [crd]
```

```
xl worldfile.xl / csv [encoding] [filename]
```

引数

[prefix] 変換元データの拡張子

[unit] 変換先座標系の長さの単位

[crd] 変換後のマッピングする先の座標系ファイル。

[encoding] CSV ファイルの文字コーディング名

[filename] CSV ファイル名

オプション

-

環境

エージェント xl [2]

説明

最初の引数に `direct` を指定した場合、一つの座標系 [crd] に複数のワールドファイルをマッピングします。最初の引数に `csv` を指定すると、所定のフォーマットに従って記述された csv ファイル [filename] からよみとられたワールドファイルを一つ一つ別々の座標系にマッピングし、その座標系をさらにワールドファイルに基づいて [crd] で指定した座標系にマッピングします。

`direct` による変換は、複数のワールドファイルが同一の画像を分割したものと考えられる場合に使用します。一方、`csv` による変換は複数のワールドファイルが全く独立した別々の図面である場合に利用することができます。

`direct` による変換の場合、[prefix] はワールドファイルの画像の拡張子を指定します。他の拡張子の画像は大正となりません。[unit] はワールドファイル中のデータの単位を示します。これがマッピング先の座標系のベース unit でもある必要があります。[crd] はマッピング先の座標系を表します。

```
$ xl worldfile.xl - - / direct tif m ginowan.crd
....
$
```

するとワーキングディレクトリにあるすべての tif ワールドファイルについて、.lst,.cr ファイルが生成され、さらに、.lst ファイルをワールドファイルに基づいて ginowan.crd にマッピングする.map ファイルが生成されます。

`csv` による変換の場合、[encoding] は CSV ファイルの記述に使われている文字コード名を与えます。[filename] は対象とする CSV ファイルを示します。

`csv` ファイルの中身の記述については、各ワールドファイルを各行に割当て、ワールドファイル名(幹の部分のみの名前)、ワールドファイルごとの生成される書誌情報を与えます。ファイルの終わりにコントロールコマンドを挿入します。これについて `csv` ファイルを Excel で開いたところを図 2.1 に示します。

”#”ではじまる行はコメントであり、処理対象とはなりません。ファイルの後半、”.”から始まる行がすべてコマンドの部分です。コマンドの詳細内容は 3 節を参照されたい。

原則として、コマンドは、一行にコマンドで、最初の列にコマンド名があり、B 列から右は各列に書かれている書誌データに固有なオプションを与えるという形式になっています。ここでつかわれているコマンドについて若干説明をすると、以下のようになります。

	A	B	C	D	E	F	G	H	I	J	K
1	#										
2	#	平成2年度	京都市埋蔵文化財調査概要								
3	#										
4	#		PDFファイル	内容	TIFF・ワールドファイル	精度(cm/px)					
5	#	序	pro.pdf								
6	#	第1章	H2_100.pdf								
7	#		H2_101.pdf	平安京朝堂礎跡							
8	#		H2_102.pdf	平安京中務省跡1							
9	#		H2_103.pdf	平安京中務省跡2							
10	#		H2_104.pdf	平安京西園跡1							
11	#		H2_105.pdf	平安京西園跡2							
12	#		H2_106.pdf	平安京左京三条一へ四坊	H2_106.FIG08-1	126					
13	#		H2_106.pdf	平安京左京三条一へ四坊	H2_106.FIG08-2	126					
14	#		H2_106.pdf	平安京左京三条一へ四坊	H2_106.FIG08-3	126					
15	#		H2_106.pdf	平安京左京三条一へ四坊	H2_106.FIG08-4	126					
89	#		H2_304.pdf	復原							
90	#	第4章	H2_401.pdf	律及格式および租税考養役事業							
91	#		H2_402.pdf	京都市考古資料館状況							
92	#		H2_403.pdf	役員員名簿							
93	#										
94	#										
95	character		small			small					
96	pre.string		http://www.nichibun.ac.jp/heiankyo/								
97	qualifier		ghomepage	gh:title							
98	dtype		url	text							
99	adata										
100	wf.mapping		no&crd								
101	wf.processing					tf					
102	#										

図 2.1: CSV ファイル

.character の行では、アルファベットの文字種を限定します。small が書かれてある列はすべて小文字に変換されます。なにも書かれていない列は変換されません。

.pre.string の行では、書誌データの前に必ずつける文字列を与えます。なにも書かれていないところはなにも先頭につけません。

.qualifier は書誌データのどの列をどのクオリファイヤーへ対応させるかを示します。これが指定されていない列は最終的に書誌データには含まれません。

.qtype は、書誌データのデータタイプを与えます。

.qdata は、すべての座標系に共通な書誌データを与えます。

.wf.mapping は、他の行とは関係なく、ワールドファイルをマッピングする先の座標系 URL を与えます。

.wf.processing は、処理対象となるワールドファイル名のある列に画像ファイル拡張子を与えます。

戻り値

- direct の場合

各ワールドファイルに対応した、.lst,.cr,.map ファイル

.map は、.lst ファイルから [crd] へのマッピング。

- csv の場合

各ワールドファイルに対応した、.lst,.cr,.map(1),.map(2),.crd ファイル

.map(1) は.lst ファイルから.crd ファイルへのマッピング。

.map(2) は、.crd ファイルから [crd] ファイルへのマッピング。

エラー

参考

バグ

2.3.9 wfutils.xl(ワールドファイル関連ツール)

プロトタイプ

`exl wfutils.xl / generate` [.map URL] [.crd URL]

`exl wfutils.xl / shift` [load-worldfile] [save-worldfile] [x-rate] [y-rate] [x-shift] [y-shift]

`exl wfutils.xl / split` [load-worldfile] [splited-files trunc] [x-split-nos] [y-split-nos]

引数

[.map URL] マップファイル URL

[.crd URL] 座標系ファイル URL

[load-worldfile] 読み込むワールドファイル名

[save-worldfile] 書き込むワールドファイル名

[splited-files trunc] 分割されたファイル名の幹名

[x-rate] X 方向解像度増大率

[y-rate] Y 方向解像度増大率

[x-shift] X 方向平行移動ピクセル数

[y-shift] Y 方向平行移動ピクセル数

[x-split-nos] X 方向分割数

[y-split-nos] Y 方向分割数。

オプション

- `dir` = [forward/reverse] デフォルト forward
- `prefix` = プリフィックス (. を含む) デフォルト .tfw
-

環境

エージェント `exl` [UNDEF REF (exl)]

説明

ver.B.b17.02 よりサポートされている機能です。

GLOBALBASE コンテンツからワールドファイルを作るツール。 `matrix.xl` (2.3.4 節) の `export-pnm` などと一緒に利用する。 `export-pnm` と一緒にワールドファイルが生成される仕組みにではないのは、既に生成された画像ファイルにあとからワールドファイルを付加できるようにするためです。

`generate`

このスクリプトに `generate` を指定すると、GLOBALBASE のコンテンツからワールドファイルを生成する。 [.map URL] に指定すべきマッピングは、ソースが画像オブジェクト、.lst ファイルであり、デスティネーションが座標系 (.crd) である必要がある。一方、 [.crd URL] に指定されるファイルは、ワールドファイルの基づく参照系に等価な座標系である。このように指定すると、 [.crd URL] の座標系で見た、.lst 画像データのワールドファイルを生成する。

`dir` オプションを指定することができ、`dir=reverse` とすると、X,Y 軸の順番を入れ替えることが出来る。また、`prefix` はワールドファイル名につくプリフィックスを指定する。

`shift`

既にあるワールドファイル [load-worldfile] があるマトリックスから生成された画像に対するワールドファイルであったと仮定する。このマトリックスから、解像度の低いデータを出力したり、また、全体の一部の画像を出路した場合のワールドファイルを作る。

level のレベルの画像の、そのレベルで見た場合の原点からのオフセットを [x-shift] [y-shift] とする。 [x-rate] [y-rate] は $1/(1 \ll (\text{level} * \text{dim_divide bit}))$ の値を指定する。現在ですと、dim.divide bit=4 であることを考慮してこの値を計算する。たとえば、1/4 という値になった場合は、 /4 と書くことが出来る。

[x-shift], [y-shift] は解像度を落としたあとの画像でどのくらいシフト場所であるかを指定する。

split

matrix.xl export-pnm などを使って一つの画像を複数の画像に分割した場合に対応して、それぞれの画像ファイルに対するワールドファイルを作る。もとの画像のワールドファイルが、[load-wroldfile] であったとする。それをよこ [x-split-nos] 個、たて [y-split-nos] 個に分割したと考える。分割後の画像の大きさをチェックするために、[splited-file trunc] を指定する。これは、matrix.xl (2.3.4 節) の export-pnm で指定する [pnm-filename] に対応する。ファイルの幹名のみを指定する。これにファイルの番号とプリフィックスが加えられている。一連のファイルを参照しながら、複数のワールドファイルを生成する。

使い方 (一枚のワールドファイル画像を作る)

1. マトリックス画像を含む座標系を作り、COSMOS で見られるようにする。そのときの座標系を test.crd, マトリックス画像をこの座標系に結びつけているマッピングを、test.map とする。

2.

```
exl wfutils.xl - - / generate test..map test.crd prefix=.tfw
```

を実行すると、ワールドファイル test.tfw が生成される。

3. test.map により結びつけられているマトリックス test.mtx から、matrix.xl export-pnm で ppm を作成。

```
x1 matrix.xl - - / export-pnm test.mtx test.ppm
```

4. 生成された test.ppm から tiff 画像を作る。

```
pnmtotiff test.ppm > test.tif
```

5. test.tif と test.tfw とペアでワールドファイルの出来上がり。

使い方 (画像の一部の解像度が低いワールドファイル画像を作る)

1. マトリックス画像を含む座標系を作り、COSMOS で見られるようにする。そのときの座標系を test.crd, マトリックス画像をこの座標系に結びつけているマッピングを、test.map とする。

2.

```
exl wfutils.xl - - / generate test.map test.crd prefix=.tfw
```

を実行すると、ワールドファイル test.tfw が生成される。

3. test.map により結びつけられているマトリックス test.mtx から、level=1 として、また、部分画像の指定をして、matrix.xl export-pnm で ppm を作成。

```
x1 matrix.xl - - / export-pnm test.mtx test.ppm 1000 2000 500 500 level=1 splitmode=part
```

4. 生成された test.ppm から tiff 画像を作る。

```
pnmtotiff test.ppm > test.tif
```

5. test.tif から、上記部分出力に合わせてワールドファイルを作りなおす。

```
exl wfutils.xl - - / shift test.tif tmp.tif /4 /4 250 500
mv tmp.tif test.tif
```

level=1 の場合、画像の解像度は 1/4 になっている。また、それにともない画像オフセットも 1/4 になる。この比率は、厳密には、マトリックスの生成時の仕様により異なるので注意が必要。

6. test.tif と test.tif とペアでワールドファイルの出来上がり。

使い方（一枚のマトリックス画像を複数のワールドファイル画像に分割）

1. マトリックス画像を含む座標系を作り、COSMOS で見られるようにする。そのときの座標系を test.crd, マトリックス画像をこの座標系に結びつけているマッピングを、test.map とする。

- 2.

```
exl wfutils.xl - - / generate test.map test.crd prefix=.tif
```

を実行すると、ワールドファイル test.tif が生成される。

3. test.map により結びつけられているマトリックス test.mtx から、X 方向 4 分割、Y 方向 8 分割の分割画像を書き出し。

```
xl matrix.xl - - / export-pnm test.mtx test 4 8 splitmode=split
```

test-0-0.ppm, test-0-1.ppm, というファイルが出来上がる。

4. test.tif から、上記部分出力に合わせてワールドファイルを作りなおす。

```
exl wfutils.xl - - / split test.tif test 4 8 prefix=.tif
mv tmp.tif test.tif
```

生成された ppm ファイルが読み込まれ、対応する tif が生成される。

5. ppm から tiff 画像を作る。

```
pnmtotiff test-0-0.ppm > test.tif
pnmtotiff test-0-1.ppm > test.tif
.....
```

6. test-X-Y.tif と test-X-Y.tif とペアでワールドファイルの出来上がり。

戻り値

- **generate** の場合

[save-worldfile] で指定したファイル。

- **shift** の場合

[save-worldfile] で指定したファイル。

- **split** の場合

[splited-file trun] で指定したファイル群。

エラー

参考

バグ

2.3.10 prog.xl(進行状況表示)

プロトタイプ

xl prog.xl

引数

環境

エージェント xl [2]

説明

ver.B.b17.02 よりサポートされている機能です。

進行状況表示機能が備わっている他のスクリプトでは、進行状況を保存したファイル.progressive を生成するものがあります。このスクリプトは、この内容を定期的に表示します。

使い方としては、長時間かかる他のスクリプトを一つのコンソールで動かし、他のコンソールで、同じワーキングディレクトリへ移動し、そこで、このコマンドを実行します。終了は、ctrl+C で終了してください。

戻り値

なし。

エラー

参考

バグ

2.3.11 csv-plot.xl(CSV プロットデータのマトリックスへの変換)

プロトタイプ

xl csv-plot.xl [CSV-FILE] [CORRESPOND] [MATRIX] [X-DOTS] [Y-DOTS]

引数

[CSV-FILE] プロットデータの CSV ファイル

[CORRESPOND] 対応変換ファイル

[MATRIX] 保存先マトリックスファイル

[X-DOTS] X 方向ドット数。

[Y-DOTS] Y 方向ドット数。

オプション

•

環境

エージェント exl [UNDEF REF (exl)]

説明

LANDSCAPE をインストールされている場合、/usr/local/xl-gbs/xlsamples/gbs/plot1 にサンプルデータがあるのでこれを参考に以下を読んでいただきたい。

このスクリプトは、CSV ファイル [CSV-FILE] に列挙されたプロットデータを [MATRIX] で与えられたマトリックスフォーマットのファイルへ変換する。マトリックスとなったプロットデータは、全体を見ていて、多量のプロットが画面に表示しきれない場合は、次のような動作となる。

1. アイコンを小さくする。
2. アイコンを小さくしても表示しきれない場合は、単なる色のついた点にする。
3. 画面の 1 ピクセルにいくつものプロットが入るような場合は、点をマージして領域表示にする。

このような操作により、表示可能プロット数はほぼ無制限となる。

マトリックスは基本的には整数座標系であるのに対して、プロットはたとえば緯度経度に対するプロットなど、連続(実数)座標系のプロットである場合が多い。そのため、マトリックスでは解像度を指定し、プロットの座標を整数に変換する。言ってみれば、実数座標系を非常に細かいグリッドに分割し、プロットの位置をそのグリッドの整数座標で表す。[X-DOTS] [Y-DOTS] はその整数グリッドの大きさを表す。

マトリックス内部の 1 プロットに与えられた属性は 0 から始まる番号で識別される。座標 XX,YY のプロットの N 番の属性といった具合である。各属性にはデータ型が存在する。このデータ型は、マトリックスの画像(グリッド)解像度を下げていく過程で、プロット同士のマージが生じた場合にどのようにマージするかを指定するものでもある。このような観点から以下のデータ型が定義されている。まず基本データ型である。

- mxPgTYPE_NONE
NONE 型。値を持たない型。
- mxPgTYPE_STRING
文字列型
- mxPgTYPE_INT
実数型。
- mxPgTYPE_RGBA
透明チャンネルを持った 8bit フルカラー。

さらに、各基本データ型に詳細データ型が存在する。

- **mxPgNONE**
NONE 型。
- **mxPgSTRING**
文字列型。マトリックス上位レベルノードの文字列を無視し、かつ新しい文字列を割り当てる。
- **mxPgSTRING_INHERIT**
文字列型。マトリックス上位レベルノードの文字列を継承する。(ユーザは使っては行けない。)
- **mxPgSTRING_ERASE**
文字列型。マトリックス上位レベルノードの文字を無視かつからのもじれるにする。(ユーザは使っては行けない。)
- **mxPgINT_ADD**
整数型。下位レベルノードのマージ対象のプロットの値の合計値をもって上位レベルノードの値とする。
- **mxPgINT_AVG**
整数型。下位レベルノードのマージ対象のプロットの値の平均値をもって上位レベルノードの値とする。
- **mxPgINT_MAX**
整数型。下位レベルノードのマージ対象のプロットの値の最大値をもって上位レベルノードの値とする。
- **mxPgINT_MIN**
整数型。下位レベルノードのマージ対象のプロットの値の最小値をもって上位レベルノードの値とする。
- **mxPgRGBA**
RGBA 型。各色チャンネルの平均値を持って上位レベルノードの値とする。

CSV ファイル [CSV-FILE] は単なる属性の列挙なので、どれが座標なのかわからない。また同様に CSV ファイルの属性は型をもっていない。そこで、

1. CSV ファイルのどの列を座標値の列とみなすか。
2. CSV ファイルの残りの列をマトリックスファイルの何番の属性に割り当てるか。
3. 割り当てるときに、型をなににするか。

を指定しなければならない。この指定を行うのが、 [CORRESPOND] で指定されるファイルである。[CORRESPOND] は、XL スクリプトファイルであるが、実際に使う命令は少ない。 /usr/local/xl-gbs/xlsamples/plot1/gaz.xl がこれにあたる。これを開くと、使われている命令は、?xl,Define,gmxPgPlotField のみである。

Define で定義されている変数

```
<Define> ^reso      3600 </Define>
```

は、解像度を変数 `reso` に定義しており、これは、3600dot/degree という意味である。例に挙げられている、`gaz.csv` では、列 A,B に `degree=度` で座標が指定されており、マトリックスファイルでは、1秒を1ドットとして定義しているためである。ファイルの最後にある、

```
<Define> ^StartLine 1 </Define>
```

は CSV の最初の無視する行数を示している。例では最初の1行を無視する。

gmxPgPlotField

次につづく、`gmxPgPlotField` によって `csv` の各列と、マトリックスファイルの座標、および属性番号の対応を実現している。`gmxPgPlotField` はエージェント `gbmx` の関数になっており詳しい機能「`gbmx エージェント・リファレンス・マニュアル`」[5]の「`gmxPgPlotField`」を参照されたい。`csv-plot.xml` スクリプトでは以下2種類のように使われる。

(A)

```
<gmxPgPlotField id="^mtx" csv="A" merge-type="^mxPgTYPE_INT" dim="0">
  ^Fields ^mxPgFDIT_DD ^reso (* 180 reso) 0</gmxPgPlotField>
```

(B)

(B-1)

```
<gmxPgPlotField id="^mtx" csv="C" merge-type="^mxPgTYPE_STRING" label="5">
  ^Fields utf-8 ([quote type="direct"] '([field name="LABEL"] ($ ___convert)))</gmxPgPlotField>
```

(B-2)

```
<gmxPgPlotField id="^mtx" merge-type="^mxPgTYPE_RGBA" label="0">
  ^Fields 0 0xff000000</gmxPgPlotField>
```

(B-3)

```
<gmxPgPlotField id="^mtx" csv="D" merge-type="^mxPgTYPE_STRING" label="1">
  ^Fields utf-8 () test-http </gmxPgPlotField>
```

(A) は座標を指定する方法であり、(B) は属性の対応を指定する方法である。XML 属性は以下のとおりである。

- `id=" ^mtx"`

マトリックスの ID を指定するものであり、`csv-plot.xml` の場合はこの文字列に固定。

- `csv`

`csv` の属性に対応する列の番号、A,B,C,... を指定する。この XML 属性を省略することができる。

- `label`

(B) のパターンで利用され、`label` XML 属性はマトリックスの属性番号を指定する。`csv` XML 属性にある列を、この `label` XML 属性で指定されるプロット属性番号に対応される。`csv` XML 属性が省略されていると `gmxPgPlotField` の要素データで与えられるデータ値が共通にすべてのプロットの `label` 属性番号に保存される。

`label` XML 属性は省略可能であるが、`dim` XML 属性とどちらかは存在している必要がある。

- dim

(A) のパターンで利用され、マトリックスの座標軸番号を指定する。マトリックスの本来の機能としては、マトリックスの次元-1 までの値が指定できるが、plot-csv.xml は 2 次元マトリックスのみサポートしているので、0,1 のいずれかが指定可能である。X 軸は 0 であり、Y 軸は 1 である。

dim XML 属性は省略可能であるが、label XML 属性とどちらかは存在している必要がある。

- merge-type マトリックスの詳細属性型を与える。

最後に gmxPgPlotField の要素データの与え方は merge-type XML 属性で与えたデータ型によって異なる。

- mxPgNONE

なし。

- mxPgSTRING

[作業領域へのポインタ] [保存文字コード名] [変換式] [初期化文字列：任意]

[変換式] が () の場合、特に CSV のデータを加工せずそのまま属性値とする。 [変換式] が XL 式で与えられる変換式として与えられていた場合、これにより CSV の値を変換した者を属性値とする。上記 (B-1) の例がそれに相当する。...convert の部分に、CSV からの値が格納される。実際の変換は、CSV のテーブルを初めて引いたときに行われるので、この時点では、シンボル\$は解釈されてはならないので、ダイレクト quote を施す必要がある。

csv 属性が指定されていない場合、 [初期化文字列：任意] が必要となり、この文字列が [変換式] に与えられ、一定値としてすべてのプロットの属性にセットされる。

- mxPgINT_ADD mxPgINT_MAX mxPgINT_MIN

[作業領域へのポインタ] [CSV 中のデータ型] [変換解像度] [変換オフセット] [カウント] [初期化数値：任意]

CSV 内のデータは、まず、 [CSV 中のデータ型] に示されるデータ型として認識され、それを倍精度浮動小数点として読み込む。 [CSV 中のデータ型] の詳細は追って記述する。読み込まれたデータは、 ([読み込まれたデータ] \times [変換解像度] + [変換オフセット]) という式で変換され、変換値は端数切り捨てにより整数値となり、マトリックスの属性データあるいは、座標値となる。 [変換解像度] , [変換オフセット] は倍精度浮動小数点として与える。

[カウント] はこの三つのデータ型では無視される。0 をおいておけばよい。

csv 属性が指定されていない場合、 [初期化数値：任意] が必要となり、この数値が、上記変換に供され、一定値としてすべてのプロットの属性にセットされる。

- mxPgINT_AVG

[作業領域へのポインタ] [CSV 中のデータ型] [変換解像度] [変換オフセット] [カウント] [初期化数値：任意]

属性データの計算方法は、 mxPgINT_ADD mxPgINT_MAX mxPgINT_MIN と同じである。このデータとともに、 [カウント] も属性データとして保存される。二つの整数値のペアとなる。この型の場合、マトリックスの上位レベルへプロットをマージする場合、各プロットの平均をとり新しいプロットの属性値とする。その平均はこの [カウント] 重み付けで計算され、さらに、上位レベルにおける [カウント] は下位レベルの [カウント] の合計値となる。

csv 属性が指定されていない場合、 [初期化数値：任意] が必要となり、この数値が、上記変換に供され、一定値としてすべてのプロットの属性にセットされる。

- mxPgRGBA

[作業領域へのポインタ] [カウント] [初期化 RGBA : 任意]

RGBA を CSV から取得し、プロットの属性へ割り当てる。16 進数で与える場合は、0xff000000 のように C 言語と同じ書式となっているひつようがある。[カウント] は、整数型の [カウント] と同様の働きをし、上位レベルの色を計算するとき、重み付けに使われる。0 を指定すると、重み付けを行わない。

csv 属性が指定されていない場合、[初期化 RGBA : 任意] が必要となり、この数値が一定値としてすべてのプロットの属性にセットされる。

整数型における [CSV 中のデータ型] これに指定する値としては以下のものがある。

- mxPgFDIT_INT

CSV のデータ型は整数型である。(64bit 整数)

- mxPgFDIT_FLOAT

CSV のデータ型は倍精度浮動小数点である。

- mxPgFDIT_DD

CSV のデータ型は緯度経度の十進小数点表記である。たとえば、135.50 は 135 度 30 分となる。

- mxPgFDIT_DDMMSS

CSV のデータ型は緯度経度の 60 進表記である。たとえば、135.5012 は、135 度 50 分 12 秒である。

/usr/local/xl-gbs/xlsamples/gbs/plot1 の使い方

プロットデータを公開したい場合、

1. /usr/local/xl-gbs/xlsamples/gbs/plot1 を公開ポイントへコピーする。たとえばそのディレクトリを、myplot とすると、
2. myplot の中身 img.crd の書誌データを編集する。
3. 公開したいプロットデータの CSV ファイルを用意し、それを、myplot 内にたとえば、myplot.csv というファイルとして保存する。
4. gaz.xl ファイルを myplot.csv 用に編集し、属性の対応をはかる。
- 5.

```
xl csv-plot.xl - - / myplot.csv gaz.xl img.mtx [X-DOTS] [Y-DOTS]
```

を実行する。

- 6.

```
xl makefile.xl - -
```

を実行する。

という流れで公開可能である。

戻り値

エラー

参考

/usr/local/xl-gbs/xlsamples/gbs/plot1

バグ

第3章 worldfile.xl スクリプト用 CSV コマンド・リファレンス

3.1 概要

worldfile.xl (2.3.8 節) スクリプトにおいて `csv` を指定した場合に参照される CSV ファイルに設定するコマンドのリファレンスです。

CSV で使われるコマンドは、すべて、A 列にコマンド名、各行にオプションを記述します。各行に書かれたオプションが、それ以前の行に表れる同じ行のデータに作用します。

このリファレンスでは「プロトタイプ」の項目にコマンド名、「引数」にオプションを記述します。

3.2 継承情報

2.3.8 節を参照する必要があります。

3.3 スクリプト

3.3.1 #(コメント)

プロトタイプ

#

引数

任意

環境

説明

この行はなんの作用も起こしません。

戻り値

エラー

参考

バグ

3.3.2 .character(文字種決定)

プロトタイプ

.character

引数

- small
- capital

環境

説明

small を指定した列にあらわれる英文字はすべて小文字に変換されます。capital を指定した列に表れる英文字はすべて大文字に変換されます。

戻り値

エラー

参考

バグ

3.3.3 .pre.string(前置)

プロトタイプ

.character

引数

前置する文字列

環境

説明

前置する文字列を指定した列では、データの前にその文字列が接続されます。

戻り値

エラー

参考

バグ

3.3.4 .post.string(後置)

プロトタイプ

.character

引数

後置する文字列

環境

説明

後置する文字列を指定した列では、データの前にその文字列が接続されます。

戻り値

エラー

参考

バグ

3.3.5 .qualifier(クオリファイヤー指定)

プロトタイプ

.qualifier

引数

クオリファイヤー名

環境

説明

クオリファイヤー名を与えた列は、各座標系の書誌情報に指定したクオリファイヤーの書誌情報として含まれます。他は無視されます。

戻り値

エラー

参考

バグ

3.3.6 .qtype(データ型指定)

プロトタイプ

.qtype

引数

型名

環境

説明

書誌情報の型を指定します。オプションを与えられた列に対応するクオリファイヤーがオプションで与えられた型となります。 .qualifier (3.3.5 節) でオプションが与えられている列以外に当該コマンドのオプションを与えてはいけません。

戻り値

エラー

参考

バグ

3.3.7 .qdata(データ指定)

プロトタイプ

.qdata

引数

データ

環境

説明

すべての座標系に共通なデータを指定します。オプションを与えられた列に対応するクオリファイヤーがどの座標系においてもこのデータとなります。 .qualifier (3.3.5 節) でオプションを与えられている列以外に当該コマンドのオプションを与えてはいけません。

戻り値

エラー

参考

バグ

3.3.8 .wf.mapping(マッピング先指定)

プロトタイプ

.wf.mapping

引数

座標系 URL

環境

説明

オプションをおく列は問いません。またこのオプションは他の行には影響を与えません。与えられた URL の座標系に、ワールドファイルに基づき、すべてのデータの座標系がマッピングされます。

戻り値

エラー

参考

バグ

3.3.9 .wf.processing(ファイル名の列の指定)

プロトタイプ

.wf.processing

引数

画像ファイル拡張子

環境

説明

画像ファイル拡張子がオプションとして指定された列が、返元となる画像ファイル名の格納された行であることを示します。この行の画像ファイル名は拡張子なしのファイル名でなければなりません。

戻り値

エラー

参考

バグ

関連図書

- [1] 森洋久. LANDSCAPE スタートアップ・マニュアル. GLOBALBASE PROJECT, 2006.
- [2] 森洋久. xl(standard) エージェント・リファレンス・マニュアル. GLOBALBASE PROJECT, 2006.
- [3] 森洋久. COSMOS ユーザーズ・マニュアル. GLOBALBASE PROJECT, 2006.
- [4] 森洋久. COSMOS リファレンス・マニュアル. GLOBALBASE PROJECT, 2006.
- [5] 森洋久. gbm_x エージェント・リファレンス・マニュアル. GLOBALBASE PROJECT, 2006.
- [6] 森洋久. gbview エージェント・リファレンス・マニュアル. GLOBALBASE PROJECT, 2006.

履歴

1. 日時: 2009-12-19
マニュアル生成。(2009-12-19 版)
2. 日時: 2006-10-15
著者: 森 洋久 反映されたバージョン: ver.B.b13
このマニュアルを作成
3. 日時: 2007-04-08
著者: 森 洋久 反映されたバージョン: ver.B.b15
スクリプト matrix.xl の機能追加。
4. 日時: 2007-04-18
著者: 森 洋久 反映されたバージョン: ver.B.b15
スクリプト p4m.xl、worldfile-mtx.xl の追加。matrix.xl の機能追加。
5. 日時: 2007-04-26
著者: 森 洋久 反映されたバージョン: ver.B.b16
スクリプト import-images.xl の追加。matrix.xl
6. 日時: 2007-04-26
著者: 森 洋久 反映されたバージョン: ver.B.b16.05
スクリプト p4m.xl の reverse おぷしょんの 追加。
7. 日時: 2007-07-19
著者: 森 洋久 反映されたバージョン: ver.B.b16.05
スクリプト import-images.xl の記述訂正
8. 日時: 2007-07-22
著者: 森 洋久 反映されたバージョン: ver.B.b16.10
スクリプト worldfile.xl の記述追加。
9. 日時: 2007-09-07
著者: 森 洋久 反映されたバージョン: ver.B.b16.13
スクリプト csv-plot.xl の記述追加。
10. 日時: 2007-10-06
著者: 森 洋久 反映されたバージョン: ver.B.b16.14
スクリプト matrix.xl の export-pnm のオプション level の追加。

11. 日時: 2007-10-18

著者: 森 洋久 反映されたバージョン: ver.B.b16.14

スクリプト utm2bl-mtx.xl の記述追加。

12. 日時: 2008-02-21

著者: 森 洋久 反映されたバージョン: ver.B.b17.02

スクリプト wfutils.xl の追加。matrix.xl の export-pnm の強化。prog.xl の追加。