

## 電子国土 Web システム API リファレンス (Ver1.0) Denshi Kokudo Web System API Reference (Ver1.0)

地理空間情報部 志田忠広・西城祐輝・村岡清隆・安藤暁史・伊藤裕之

Geospatial Information Department

Tadahiro SHIDA, Yuuki SAIJO, Kiyotaka MURAOKA, Akifumi ANDO and Hiroyuki ITO

### 要 旨

近年、インターネットを通じて、様々な情報をやり取りすることが可能な時代である。地図情報についても例外ではなく、インターネットを利用できる環境さえあれば、日本全国どの地域の地図情報でも、利用することができる。

このような状況の下、国土地理院では、いつでも、誰でも、どこでも、国土に関する様々な情報を統合し、国土の管理や災害対策、行政・福祉、観光など幅広い分野で活用できるよう、電子国土 Web システムを開発・公開してきた。電子国土 Web システムは、国土地理院から提供している 1/25,000 地形図等を Web 上で閲覧するだけのシステムではなく、地図上にさまざまな情報を重ね合わせることができ、それらは、インターネットを通じて発信することができるシステムとなっている。

国土地理院では、平成 15 年 7 月からプラグインのインストールを必要とする電子国土 Web システム（プラグイン版）の提供を開始し、その後も機能追加などのシステム改良を行ってきた。平成 16 年 6 月にバージョン 0.8.1、平成 19 年 1 月には、Internet Explorer のみの対応であるがバージョン 1.0.0 を公開した。その一方で、プラグインを必要としない電子国土 Web システム（非プラグイン版）の開発も行っており、平成 17 年 7 月に 1 版を、平成 19 年 9 月に機能追加した 2 版を公開した。電子国土 Web システム（非プラグイン版）は、プラグインを必要としないため、サイトにアクセスさえすれば、すぐに地図を閲覧することができ、また、様々なブラウザ上で動作させることができる。

電子国土 Web システムでは、容易に電子国土 Web システムを利用した Web サイト（電子国土サイト）を構築し、情報を発信できるよう、様々な API(Application Program Interface) 及びツールなどを用意している。

本稿では、これらの電子国土 API の機能や使い方などについて解説する。なお、現在公開中の電子国土 Web システム（非プラグイン版） 2 版では、本稿で紹介する電子国土 API の一部を利用することができない。

### 1 . はじめに

本稿は、電子国土 Web システム第 1.0 版が提供する API (Application Programming Interface) 関数の詳細を解説するものである。

すべての電子国土 Web システム API は JavaScript を用いて呼び出すことができるようになっている。電子国土 Web システムを利用するページでは「map」という名前のフレームの内容を、電子国土 Web システム API を使って制御することによって、地図及びそこに重ね合わせ表示された地理情報を操作することができる。なお、フレーム名は、任意に指定することが可能であるが、本稿では「map」として解説する。

本稿では、それぞれの電子国土 Web システム API についての解説のみを記載したものであり、その利用方法に関しては「電子国土 Web システムプログラミングガイド（以下、「プログラミングガイド」という。）」を参照のこと。

本稿の構成は、以下となっている。

- 2 . 1 必須関数
- 2 . 2 地図表示関数
- 2 . 3 地図操作関数
- 2 . 4 地図情報関数
- 2 . 5 レイヤ操作関数
- 2 . 6 印刷関連関数
- 2 . 7 イベント関連関数
- 2 . 8 オブジェクト操作関数
- 2 . 9 振発レイヤ関連関数
- 2 . 10 プラグイン関連関数
- 2 . 11 グラフ関連関数

### 2 . 電子国土 Web システム API リファレンス

#### 2 . 1 必須関数

必須関数は、電子国土 Web システムを動作させるために必要な関数であり、ここに挙げられた関数が存在しない場合、電子国土 Web システムが起動しない、あるいは異常終了する。

電子国土 Web システムを利用したサイト（以下、「電子国土サイト」という。）構築者は、ここに挙げられた関数を必ずページ内の適当な箇所に記述しなければならない。

## 一覧

- **app\_main()**  
電子国土 Web システム起動関数

### 関数名

**app\_main()**

### 機能

電子国土 Web システムの初期化と必要な初期設定を行うユーザ定義の必須関数である。電子国土 Web システムプラグインが起動後、最初に呼び出され、この関数の中に記述された関数群が実行される。

C 言語プログラムにおける main() 関数のような位置づけの関数である。

### 条件

この関数は、map フレーム等を定義する HTML ファイル内に記述しなければならない。

### 引数

引数はなし。

### 戻り値

戻り値はなし。

### 使用方法

通常は、この関数の中で、ページがロードされた時に最初に表示する地図の位置や縮尺の決定、上乗せデータのロード、スケールバーや経緯度情報等の画面デザイン関数の実行等を行う。

#### さらに詳細な技術情報

電子国土 Web システムが起動後は、背景となる各種レベルの地図データを読み出す URL の定義などが行われる。それらの定義後、初期化処理が行われ、フレームサイズに合わせた描画サイズ設定と再描画を行った後、初期化関数の最後にこの app\_main() 関数が呼び出される。関数は、top.app\_main() の形で呼び出されるため、フレーム定義時に src= " " によってロードされる HTML ソース内に記述するとエラーになるので、必ず map フレームを定義している HTML ファイル内に記述をしなければならない。

## 2.2 地図表示関数

地図表示関数は、map フレームへの地図表示及び表示する地図の設定を行う関数である。

## 一覧

- **openMap()**  
指定したパラメータで地図を読み出す
- **openMap(url)**  
ネイティブデータの地図を読み出す
- **openJSGIXML()**  
ローカルから電子国土 XML データを読み出すダイアログを表示し、選択されたデータを表示する

- **openJSGIXML(url, fit)**  
電子国土 XML データを読み出して表示する
- **openJSGIXMLex(url, second, fit)**  
電子国土 XML データを読み出す（タイムアウトチェックつき）
- **setMapRect()**  
表示する地図の範囲を指定する
- **setMapCenter()**  
表示する地図の中心位置と表示縮尺を指定する
- **setMapScale()**  
表示する地図の表示縮尺を指定する
- **clearMap()**  
地図表示領域をクリアする
- **redrawMap()**  
地図表示領域を再描画する
- **createScaleBar()**  
スケールバーを表示する
- **enablePopup()**  
地図オブジェクトの属性をポップアップ表示する
- **disablePopup()**  
地図オブジェクトの属性ポップアップ表示を OFF にする
- **setLoadMcIClass(class)**  
openJSGIXML 関数で読み出すクラス名を指定する
- **existMcIClass(class)**  
クラス名の存在を調べる

### 関数名

**top.map.openMap()**

### 機能

あらかじめ指定されたパラメータに従って、地図データを地図サーバから読み出す。

### 引数

引数はなし。

### 戻り値

戻り値はなし。

### 解説

openMap 関数を実行したときに、指定されている位置と縮尺から必要な地図データを地図サーバから読み出す。

位置や縮尺は、setMapCenter、setScale、setMapRect 関数などをあらかじめ呼び出すことによって指定する。位置や縮尺が指定されていない場合は、なにも読み出されない。

openMap 関数が読み出す地図データのレベルと表示縮尺の関係は、以下のとおりである。

表示縮尺

ロードされる地図データ

~1/1,000,000	全国レベル (1/2000万相当)
1/1,000,000 ~ 1/220,000	都道府県レベル (1/300万相当)
1/220,000 ~ 1/32,000	市町村レベル (1/20万相当)
1/32,000 ~ 1/15,000	概略図 (1/5万相当)
1/15,000 ~ 1/2,500	詳細地形図 (1/2.5万相当)
1/2,500 ~	大縮尺図 (1/2500 ~ 1/500相当)

ただし、大縮尺図については、市町村が発信しているので、発信している市町村がない地域では、何も表示されない。

#### 使用例

```
top.map.setMapCenter(140.084,36.104,10000);
top.map.openMap();
// 中心位置（経度、緯度）と縮尺(1/10000)を
// 指定し地図を表示する
```

#### 関数名

**top.map.openMap(url)**

#### 機能

指定したネイティブデータの地図を読み出す。

#### 引数

ネイティブデータファイルの URL をフルパスで指定する。

#### 戻り値

戻り値はなし。

#### 解説

電子国土 Web システム用ネイティブデータを読み出す。ネイティブデータとは、電子国土 XML データ形式をネイティブ化ツール (xml2htm.exe) で変換したデータファイルである。ネイティブ化により、データ量を概ね 1/10 にできる。

#### 使用例

```
top.map.openMap("http://domain/folders/
htmdata.htm");
// 電子国土ネイティブデータの htmdata.htm
// ファイルを読み出す
```

#### 関数名

**top.map.openJSGIXML(url [,fit])**

#### 機能

指定した URL から電子国土 XML データを読み出し表示する。

#### 引数

url 引数は、読み出す電子国土 XML データファイルまたは CGI を示す URL を「http://」で始まるフルパスで指定する。

fit 省略した場合、読み出したデータの全体を

表示する。

表示範囲を変更することなく読み出したデータを表示する場合は、1を指定する。

#### 戻り値

読み出されたオブジェクトの配列を返す。

#### 解説

この関数は、電子国土 XML データ仕様に基づいて作成されたデータを読み出し、データ内に記述された描画法および地理情報の位置情報に従って、描画される。データ全体を表示させた場合、データ範囲に合わせて必要な範囲の背景データもロードされる。

#### 使用例

```
ob=top.map.openJSGIXML("http://domain/
folders/data.xml");
// data.xml データ全体が表示されるよう縮尺、
// 範囲を調整し表示する。
ob=top.map.openJSGIXML("http://domain/
folders/data.xml",1);
// 表示範囲、縮尺を変更せずに data.xml を読み出し表示する
```

#### 関数名

**top.map.openJSGIXML()**

#### 機能

ローカルディスクから電子国土 XML データを読み出すダイアログを表示し、選択されたデータを、画面に表示する。

#### 引数

引数はなし。

#### 戻り値

戻り値はなし。

#### 解説

この関数は、top.map.openJSGIXML(url) 関数の url で読み出すデータを指定する代わりに、ファイルを開くダイアログを表示してローカルディスクのデータを読み込む。

#### 関数名

**top.map.openJSGIXMLEx(url,second,fit)**

#### 機能

指定した URL から、指定した時間内に読み出しが完了するかチェックしながら電子国土 XML データを読み出し、表示する。

#### 引数

url 読み出す電子国土 XML データファイルまたは CGI を示す URL を「http://」で始まるフルパスで指定する。

Second タイムアウトを判断する時間を秒単位の整数で指定する。

fit 0: 表示範囲を変更しない

1: 読み出したデータの全体を表示する

## 戻り値

指定された時間内に読み出しが終了した場合、読み出されたオブジェクトの配列が返る。  
指定された時間内に読み出しが終了しない場合、文字列「timeout」を返す。

## 解説

この関数は、電子国土 XML データ仕様に基づいて作成されたデータを読み出し、データ内に記述された描画法および地理情報の位置情報に従って、描画される。データ全体を表示させた場合、データ範囲に合わせて必要な範囲の背景データがロードされる。

基本的には openJSGIXML 関数にタイムアウト処理を追加したものであるが、タイムアウトの計測は、url に指定するサーバとのコネクションが確立してからの秒数で行われる。サーバが発見できないことによるタイムアウト処理は行われない。

## 使用例

```
ob=top.map.openJSGIXML("http://domain/
folders/data.xml",30,0);
// タイムアウト時間を 30 秒に設定し、表示範
囲を変更せずに XML データを読み出す
```

## 関数名

**top.map.setMapRect(left,bottom,right,top)**

## 機能

map フレームに表示する地図の範囲を指定する。  
引数

表示する範囲の西端の経度、南端の緯度、東端の経度、北端の緯度を度単位の 10 進数の浮動小数点数で指定する。これら 4 つの引数は、セットで指定しなければならない。

left 表示範囲の西端の経度値  
bottom 表示範囲の南端の緯度値  
right 表示範囲の東端の経度値  
top 表示範囲の北端の緯度値

## 戻り値

戻り値はなし。

## 解説

引数として指定するのは、表示したい対象範囲に外接する矩形（矩形の辺は東西南北の方向に沿ったものとする）のそれぞれの経度値又は緯度値である。

この関数は、map フレームに表示する地図の範囲をパラメータとしてセットするだけであり、実際の地図データの読み出しあは、openMap 関数が実行された時点で行われる。また、この関数の後に、setScale 関数や、setMapCenter 関数などが実行されると、パラメータが更新され、指定した範囲とは異なる範囲の地図データが読み出される。

## 使用例

```
top.map.setMapRect(140.5,35.75,140.7,36.2);
top.map.openMap();
// 表示範囲(西端, 南端, 東端, 北端)を度単
位で指定し地図を表示する
```

## 関数名

**top.map.setMapCenter(x,y [,scale])**

## 機能

map フレームに表示する地図の中心位置と表示縮尺を指定する。

## 引数

表示する範囲の中心の緯度、経度を度単位の 10 進数の浮動小数点数で、さらに表示縮尺の分母を整数で指定する。scale は、省略することができます。

x 表示範囲の中心位置の経度値  
y 表示範囲の中心位置の緯度値  
scale 表示縮尺の分母数字（1/10,000 の場合は、10000 を指定）（省略可能）

## 戻り値

戻り値はなし。

## 解説

引数として指定するのは、表示したい対象範囲の中心位置と、表示縮尺の分母である。表示縮尺が省略された場合、表示縮尺は変更されない。

この関数は、map フレームに表示する地図の位置をパラメータとしてセットするだけであり、実際の地図データの読み出しあは、openMap() 関数が実行された時点で行われる。また、この関数の後に、setScale 関数や、setMapRect 関数などが実行されると、パラメータが更新されるため、指定した範囲とは異なる地図データが読み出される。

## 使用例

```
top.map.setMapCenter(140.084640,
36.104638,10000);
top.map.openMap();
// 地図の中心位置と縮尺(1/10000)を指定し地
図を表示する
```

## 関数名

**top.map.setScale(scale)**

## 機能

map フレームに表示する地図の表示縮尺を指定する。

## 引数

表示縮尺の分母を整数で指定する。  
scale 表示縮尺の分母数字（1/10,000 の場合は、10000 を指定）

## 戻り値

戻り値はなし。

## 解説

引数として指定するのは、表示縮尺の分母数値

である。

この関数は、map フレームに表示する地図の位置をパラメータとしてセットするだけであり、実際の地図データの読み出しは、openMap()関数が実行された時点で行われる。また、この関数の後に、setMapCenter 関数や、setMapRect 関数などが実行されると、パラメータが更新され、指定した範囲とは異なる地図データが読み出される。

#### 使用例

```
top.map.setMapScale(10000);
top.map.openMap();
// 地図の表示縮尺を 1/10000 に指定し地図を
// 表示する
```

#### 関数名

**top.map.clearMap()**

#### 機能

地図表示領域をクリアする。

#### 引数

引数はなし。

#### 戻り値

戻り値はなし。

#### 解説

この関数は、map フレームに表示されているすべての重ね合わせ情報、背景地図をクリアし、初期状態に戻す。初期状態とは、地図が読み込まれていない状態であり、map フレームには地図が表示されない。

#### 関数名

**top.map.redrawMap()**

#### 機能

地図表示領域を再描画する。

#### 引数

引数はなし。

#### 戻り値

戻り値はなし。

#### 解説

この関数は、map フレームに表示されているすべての重ね合わせ情報、背景地図を再描画する。地図データの新たなロードは行われない。

#### 関数名

**top.map.createScaleBar()**

#### 機能

map フレームにスケールバーを表示する。

#### 引数

引数はなし。

#### 戻り値

戻り値はなし。

#### 解説

map フレームにスケールバーを表示したい場合に、app\_main 関数内で最初に 1 回だけ呼び出す。

スケールバーを非表示にする関数は用意されていないので、この関数が呼び出された後は、常にスケールバーが表示され、表示位置は、map フレームの右下隅固定となる。

map フレームのサイズによっては、共通規約等を示すアイコンと重複することがあり、その場合には共通規約等を示すアイコンの表示が優先され、非表示となる。

#### 関数名

**top.map.enablePopup()**

#### 機能

電子国土 XML データ形式で上乗せされたデータおよび電子国土背景データの属性値を吹き出し形式でポップアップ表示する。

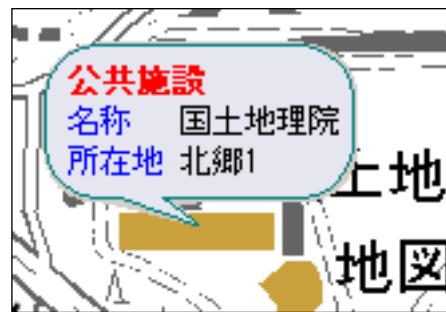


図 - 1 属性吹き出しの表示

#### 引数

引数はなし。

#### 戻り値

戻り値はなし。

#### 解説

電子国土 XML データ形式で読み込まれたデータの<attribute>要素に記述されたデータが、表示される。詳細は、objFieldVal 関数の項を参照のこと。

1/25000 レベルの背景データは、公共施設属性を吹き出し表示することができる。

電子国土 Web システム起動後のデフォルトは、ポップアップ表示機能は無効になっている。

#### 関数名

**top.map.disablePopup()**

#### 機能

電子国土 XML データ形式で上乗せされたデータおよび基盤地図データの属性値を吹き出し形式のポップアップ表示を OFF にする。

#### 引数

引数はなし。

#### 戻り値

戻り値はなし。

#### 解説

enablePopup 関数によって有効にされた属性ポ

ツップアップ表示機能を無効にする .

電子国土 Web システム起動後のデフォルトでは ,  
ツップアップ表示機能は無効になっている .

関数名

**top.map.setLoadXmlClass(class)**

機能

openJSGIXML 関数で読み出すクラス名を指定す  
る .

引数

クラス名を指定する . 省略した場合は , デフォ  
ルトの "JSGIXML" となる .

class クラス名

戻り値

戻り値はなし .

解説

openJSGIXML 関数で読み出す電子国土 XML デー  
タを , 指定したクラス名で読み出す . 指定したク  
ラスが無い場合は , 新たにクラス名を作成する .  
クラス名の変更又は作成した場合 , もう一度 , 関  
数を実行しクラス名を変更するまで変わらない .  
openJSGIXML 関数で読み出すクラス名のデフォル  
トは , "JSGIXML" である . 指定したクラス名は ,  
saveJSGIXML 関数で指定するクラス名に利用する .

使用例

```
top.map.setLoadXmlClass("DATAXML");
// 読み出すクラス名を"DATAXML"に指定する
top.map.openJSGIXML("http://domain/
                      folders/data.xml ");
// 電子国土 XML データをクラス名"DATAXML"で  
読み出す
```

関数名

**top.map.existMcIClass(class)**

機能

クラス名の存在を調べる .

引数

存在を調べるクラス名を指定する .

class クラス名

戻り値

0 : クラス名が存在する

1 : クラス名が存在しない

解説

クラス名を指定しクラス名が存在するか調べる .

使用例

```
cl = top.map.existMcIClass("DATAXML");
// クラス名"DATAXML"の存在を調べる
```

## 2 . 3 地図操作関数

地図操作関数は , 地図の拡大 , 縮小 , 移動を操作  
するための関数である .

一覧

- **startZooming(updown)**

ズーミング処理を開始する

- **stopZooming()**

ズーミング処理を終了する

- **startScroll(direction)**

スクロール処理を開始する

- **stopScroll()**

スクロール処理を終了する

関数名

**top.map.startZooming(updown)**

機能

ズーミング処理を開始する .

引数

ズームインまたはズームアウトを示す以下の文  
字列のいずれかを指定する .

updown ズームインまたはズームアウトを指定

in : ズームイン (拡大)

out : ズームアウト (縮小)

戻り値

戻り値はなし .

解説

この関数は , 地図表示領域を拡大 / 縮小するズ  
ーミング処理を開始する . いったんズーミング処  
理が開始されると , 次に startZooming 関数が実行  
されるか , stopZooming 関数によってズーミング  
処理を終了するまで , 拡大 / 縮小を続ける .

使用例

```
top.map.startZooming('in');
// 地図を拡大する
```

関数名

**top.map.stopZooming()**

機能

ズーミング処理を終了する .

引数

引数はなし .

戻り値

戻り値はなし .

解説

この関数は , startZooming 関数によって開始さ  
れたズーミング処理を終了させる .

関数名

**top.map.startScroll(direction)**

機能

スクロール処理を開始する .

引数

スクロールする方向を次の文字列で指定する .

左上	上	右上
左		右
左下	下	右下

**戻り値**

戻り値はなし。

**解説**

この関数は、地図表示領域をスクロールする処理を開始する。いったんスクロール処理が開始されると、次に startScroll 関数が実行されるか、stopScroll 関数によってスクロール処理を終了するまで、スクロール処理を続ける。

スクロールによって新しく読み出す必要が生じた地域の背景データは、電子国土 Web システムによって自動的にロードされる。

**使用例**

```
top.map.startScroll('右');
// 地図を右にスクロールさせる
```

**関数名**

**top.map.stopScroll()**

**機能**

スクロール処理を終了する。

**引数**

引数はなし。

**戻り値**

戻り値はなし。

**解説**

この関数は、startScroll 関数によって開始されたスクロール処理を終了させる。

## 2.4 地図情報関数

地図情報関数は、表示されている地図の縮尺や座標を取得するための関数である。

**一覧**

- **getMapScale()**  
表示されている地図の縮尺を取得する
- **getPositionMaiIXY()**  
地図中心位置と表示縮尺を組み合わせた文字列を生成する
- **getRect()**  
表示されている地図データの範囲を取得する
- **getLeft()**  
表示されている地図西端の経度を取得する
- **getBottom()**  
表示されている地図南端の緯度を取得する
- **getRight()**

表示されている地図東端の経度を取得する

- **getTop()**

表示されている地図北端の緯度を取得する

- **getCenter()**

表示されている地図中央の経度緯度を取得する

- **getCx()**

表示されている地図中央の経度を取得する

- **getCy()**

表示されている地図中央の緯度を取得する

- **copyCenter()**

表示されている地図中央の経度緯度をクリップボードにコピーする

- **pasteText()**

クリップボードから文字列を取得する

**関数名**

**top.map.getMapScale()**

**機能**

map フレームに表示されている地図の縮尺を取得する。

**引数**

引数はなし。

**戻り値**

表示されている地図の縮尺分母数値を返す。

**解説**

map フレームに表示されている地図の縮尺の分母数字を返す。表示縮尺が 1/15,000 の場合は、15,000 が返される。

**関数名**

**top.map.getPositionMaiIXY()**

**機能**

map フレームに表示されている地図の中心位置と表示縮尺を「xy=x,y,scale」の形のテキストに整形する。

**引数**

引数はなし。

**戻り値**

表示されている地図の中心位置と表示縮尺を「xy=経度値,緯度値,縮尺の分母」という形で組み合わせた文字列となり、経度緯度値は、度単位の10進数で表した浮動小数点数となる。

**解説**

map フレームに表示されている地図の中心位置を経度緯度と表示縮尺の分母数字をひとつの文字列に整形し返す。その値は、「getCx()」「getCy()」「getScale()」で取得するものを「xy=」の後ろにカンマ区切りでマージしたものになる。

たとえば、以下のような文字列が返される。

xy=140.22584,36.44581,8500

**関数名**

**top.map.getRect()**

**機能**

`map` フレームに表示されている地図範囲の経度  
緯度値を取得する。

**引数**

引数はなし。

**戻り値**

`map` フレームに表示されている地図範囲の経度  
緯度値が, 度単位の 10 進数で表した浮動小数点数  
が配列変数で返す。

- [0] 地図西端の経度値
- [1] 地図南端の緯度値
- [2] 地図東端の経度値
- [3] 地図北端の緯度値

**解説**

地図の表示範囲を知りたい場合などに使用する  
関数で, 同様の関数として `getLeft`, `getBottom`,  
`getRight`, `getTop` 関数がある。

**関数名**

`top.map.getLeft()`

**機能**

`map` フレームに表示されている地図西端の経度  
値を取得する。

**引数**

引数はなし。

**戻り値**

`map` フレームに表示されている地図西端の経度  
値を度単位の 10 進数で表した浮動小数点数とな  
る 地図が読み込まれていない場合は, 0 を返す。

**解説**

地図の表示範囲を知りたい場合などに使用する  
関数で, 同様の関数として `getRect`, `getBottom`,  
`getRight`, `getTop` 関数がある。

**関数名**

`top.map.getBottom()`

**機能**

`map` フレームに表示されている地図南端の緯度  
値を取得する。

**引数**

引数はなし。

**戻り値**

`map` フレームに表示されている地図南端の緯度  
値を度単位の 10 進数で表した浮動小数点数とな  
る 地図が読み込まれていない場合は, 0 を返す。

**解説**

地図の表示範囲を知りたい場合などに使用する  
関数で, 同様の関数として `getRect`, `getLeft`,  
`getRight`, `getTop` 関数がある。

**関数名**

`top.map.getRight()`

**機能**

`map` フレームに表示されている地図東端の経度  
値を取得する。

**引数**

引数はなし。

**戻り値**

`map` フレームに表示されている地図東端の経度  
値を度単位の 10 進数で表した浮動小数点数とな  
る 地図が読み込まれていない場合は, 0 を返す。

**解説**

地図の表示範囲を知りたい場合などに使用する  
関数で, 同様の関数として `getRect`, `getLeft`,  
`getBottom`, `getTop` 関数がある。

**関数名**

`top.map.getTop()`

**機能**

`map` フレームに表示されている地図北端の緯度  
値を取得する。

**引数**

引数はなし。

**戻り値**

`map` フレームに表示されている地図北端の緯度  
値を度単位の 10 進数で表した浮動小数点数とな  
る 地図が読み込まれていない場合は, 0 を返す。

**解説**

地図の表示範囲を知りたい場合などに使用する  
関数で, 同様の関数として `getRect`, `getLeft`,  
`getBottom`, `getRight` 関数がある。

**関数名**

`top.map.getCenter()`

**機能**

`map` フレームに表示されている地図中央の経緯  
度を取得する。

**引数**

引数はなし。

**戻り値**

`map` フレームに表示されている地図中央の経緯  
度が, 度単位の 10 進数で表した浮動小数点数を配  
列変数で返す。

- [0] 地図の中央の経度値
- [1] 地図の中央の緯度値

**解説**

地図の表示範囲を知りたい場合などに使用する  
関数で, 同様の関数として `getCx`, `getCy` 関数があ  
る。

**関数名**

`top.map.getCx()`

**機能**

`map` フレームに表示されている地図中央の経度  
値を取得する。

**引数**

引数はなし .

#### 戻り値

map フレームに表示されている地図中央の経度  
値を度単位の 10 進数で表した浮動小数点数とな  
る 地図が読み込まれていない場合は ,0 を返す .

#### 解説

地図の表示範囲を知りたい場合などに使用する  
関数で , getCy 関数と組み合わせることで , 画面  
中央の位置を知ることができる .

#### 関数名

**top.map.getCy()**

#### 機能

map フレームに表示されている地図中央の緯度  
値を取得する .

#### 引数

引数はなし .

#### 戻り値

map フレームに表示されている地図中央の緯度  
値を度単位の 10 進数で表した浮動小数点数とな  
る 地図が読み込まれていない場合は ,0 を返す .

#### 解説

地図の表示範囲を知りたい場合などに使用する  
関数で , getCx 関数と組み合わせることで , 画面  
中央の位置を知ることができる .

#### 関数名

**top.map.copyCenter()**

#### 機能

表示されている地図中央の経度緯度をクリップ  
ボードにコピーする .

#### 引数

引数はなし .

#### 戻り値

戻り値はなし .

#### 解説

表示されている地図中央の経度緯度を「経度値,  
緯度値」という文字列でクリップボードにコピー  
する . 値の表現は , 度単位 10 進数の浮動小数点数  
である .

#### 関数名

**top.map.pasteText()**

#### 機能

copyCenter 関数でクリップボードにコピーさ  
れた文字列を取得する .

#### 引数

引数はなし .

#### 戻り値

クリップボードの文字列を返す .

#### 解説

copyCenter 関数で , クリップボードにコピーさ  
れた地図中央の経度緯度を , pasteText 関数を実

行することにより取得することができる .

## 2.5 レイヤ操作関数

レイヤ操作関数は , 表示されている地図のレイヤ  
に対して , 操作をするための関数である .

#### 一覧

- **displayLayer(layer, onoff)**  
レイヤごとに表示 / 非表示を切り替える ( 前方  
一致 )
  - **displayStyle(layer, onoff)**  
レイヤごとに表示 / 非表示を切り替える ( 完全  
一致 )
  - **focusLayer(layer)**  
指定するレイヤを最前面に表示する
  - **clearLayers(layer)**  
指定するレイヤをクリアする ( 前方一致 )
  - **clearLayer(layer)**  
指定するレイヤをクリアする ( 完全一致 )
  - **setLayerRGB(layer, r, g, b)**  
指定するレイヤの色を変更する
  - **setLayerHRGB(layer, r, g, b)**  
指定するレイヤの塗りつぶし色あるいはハッチ  
ングの前景色を変更する
  - **setLayerBRGB(layer, r, g, b)**  
指定するレイヤのハッチングの背景色を変更す  
る
  - **getLayers()**  
上乗せデータすべてのレイヤ名リストを取得す  
る
  - **getDescription()**  
上乗せデータすべてのレイヤ説明文を取得する
  - **getLayerDescription(layer)**  
指定するレイヤの説明文を取得する
  - **openDisplayLayerWindow()**  
表示されているレイヤの一覧を表示する
  - **setMapColorWhitish(plus)**  
1/25,000 レベルの地図の表示色を薄くする
  - **setMapColorNormal()**  
setMapColorWhitish 関数で変更した表示色を  
もとに戻す
- 関数名
- top.map.displayLayer(layer [,onoff])**
- 機能
- レイヤごとに表示 / 非表示を切り替える .
- 引数
- 切り替えたいレイヤ名と , 表示 / 非表示を表す  
フラグを指定する . ここで指定するレイヤ名とは ,  
電子国土 XML データ形式の<GI><dataset><layer>  
<name>の内容と<GI><dataset><layer><style><na  
me>の内容を , ピリオド「.」でつなぎるものであ

る。

**layer** レイヤ名（前方一致で検索される。）  
(\*が使用可能)  
**onoff** 表示 / 非表示を表すフラグ（省略可能）  
0：非表示  
1：表示

#### 戻り値

**onoff** が与えられず、レイヤ名のみを引数として呼び出された場合は、そのレイヤの表示 / 非表示の状態表すフラグを返す。

0：非表示  
1：表示  
2：レイヤが複数検索され表示と非表示が混在している

#### 解説

この関数は、指定されたレイヤの表示 / 非表示を切り替える。指定されたレイヤが存在しない場合は、何も行われない。また、**onoff** が与えられない場合は、そのレイヤの状態を返す。

レイヤ名'電子国土 XML.面1' と '電子国土 XML.面2' があり、引数レイヤ名に'電子国土 XML.面' と指定した場合、前方一致で検索されるため、両方のレイヤの表示が切り替わる。**displayLayer** 関数は前方一致、**displayStyle** 関数は完全一致で検索されるところの違いがある。

#### 使用例

```
top.map.displayLayer('電子国土 XML.面',0);
// レイヤ名'電子国土 XML'を非表示にする
layonoff =
    top.map.displayLayer('電子国土 XML.面');
// レイヤ名'電子国土 XML.面'の表示状態を調べる
```

#### 関数名

**top.map.displayStyle(layer [,onoff])**

#### 機能

レイヤごとに表示 / 非表示を切り替える。

#### 引数

切り替えたいレイヤ名と、表示か非表示かを表す数値を指定する。ここで指定するレイヤ名とは、電子国土 XML データ形式の<GI><dataset><layer><name>の内容と<GI><dataset><layer><style><name>の内容を、ピリオド「.」でつないだものである。

**layer** レイヤ名（完全一致で検索される）  
(\*が使用可能)  
**onoff** 表示 / 非表示を表すフラグ（省略可能）  
0：非表示  
1：表示

#### 戻り値

**onoff** が与えられず、レイヤ名のみを引数とし

て呼び出された場合は、そのレイヤの表示 / 非表示の状態表すフラグを返す。

0：非表示  
1：表示  
2：レイヤが複数検索され表示と非表示が混在している

#### 解説

この関数は、指定されたレイヤ名を持つオブジェクトの表示 / 非表示を切り替える。指定されたレイヤが存在しない場合は、何も行われない。また、**onoff** が与えられない場合は、そのレイヤの状態を返す。

レイヤ名'電子国土 XML.面1'がある場合、表示を切り替えるには、引数レイヤ名に'電子国土 XML.面1' と指定し完全一致させる必要がある。**displayLayer** 関数は前方一致、**displayStyle** 関数は完全一致で検索されるところの違いがある。

#### 使用例

```
top.map.displaystyle('電子国土 XML.面',1);
// レイヤ名'電子国土 XML.面'を表示する
styonoff =
    top.map.displayStyle('電子国土 XML.面');
// レイヤ名'電子国土 XML.面'の表示状態を調べる
```

#### 関数名

**top.map.focusLayer(layer)**

#### 機能

指定するレイヤを最前面に移動する。

#### 引数

最前面に移動したいレイヤ名を指定する。ここで指定するレイヤ名とは、電子国土 XML データ形式の<GI><dataset><layer><name>の内容と<GI><dataset><layer><style><name>の内容を、ピリオド「.」でつないだものである。

**layer** レイヤ名（完全一致で検索される）

#### 戻り値

戻り値はなし。

#### 解説

**openJSGIXML**, **openJSGIXMLex** 関数で読み込まれたレイヤは、読み込まれた順に重なっている。同じ場所に異なるレイヤのオブジェクトが重なる場合に、下側のレイヤを最前面に移動させることによって、選択しやすくなることができる。

レイヤ名'電子国土 XML.面1'がある場合、このレイヤを最前面に移動するには、引数レイヤ名に'電子国土 XML.面1' と指定し完全一致させる必要がある。**focusLayers** 関数は前方一致、**focusLayer** 関数は完全一致で検索されるところの違いがある。

#### 使用例

```
top.map.focusLayer('電子国土 XML.面');
```

// レイヤ名'電子国土 XML.面'を最前面に移動する

## 関数名

**top.map.clearLayers(layer)**

## 機能

指定するレイヤをクリアする。

## 引数

クリアしたいレイヤ名を指定する。ここで指定するレイヤ名とは、電子国土 XML データ形式の<G I><dataset><layer><name>の内容と<G I><dataset><layer><style><name>の内容を、ピリオド「.」でつないだものである。

layer レイヤ名（前方一致で検索される）  
（\*が使用可能）

## 戻り値

戻り値はなし。

## 解説

この関数は、指定されたレイヤをクリアし、再描画を行う。指定されたレイヤのデータが表示されていた場合は、消去される。また、指定されたレイヤの、下位のレイヤもすべてクリアされる。

この関数でレイヤを削除した場合、ブラウザを再起動しない限り、削除したレイヤを元に戻す方法はない。

レイヤ名'電子国土 XML.線 1' と '電子国土 XML.線 2' があり、引数レイヤ名に'電子国土 XML.線' と指定した場合、前方一致で検索されるため、両方のレイヤがクリアされる。clearLayers 関数は前方一致、clearLayer 関数は完全一致で検索されるところの違いがある。

## 使用例

```
top.map.clearLayers('電子国土 XML.線');
// レイヤ名'電子国土 XML.線'をクリアする
```

## 関数名

**top.map.clearLayer(layer)**

## 機能

指定するレイヤをクリアする。

## 引数

クリアしたいレイヤ名を指定する。ここで指定するレイヤ名とは、電子国土 XML データ形式の<G I><dataset><layer><name>の内容と<G I><dataset><layer><style><name>の内容を、ピリオド「.」でつないだものである。

layer レイヤ名（完全一致で検索される）  
（\*が使用可能）

## 戻り値

戻り値はなし。

## 解説

この関数は、指定されたレイヤをクリアし、再

描画を行う。指定されたレイヤのデータが表示されていた場合は、消去される。

この関数でレイヤを削除した場合、ブラウザを再起動しない限り、削除したレイヤを元に戻す方法はない。

レイヤ名'電子国土 XML.線 1' がある場合、このレイヤをクリアするには、引数レイヤ名に'電子国土 XML.線 1' と指定し完全一致させる必要がある。clearLayers 関数は前方一致で、clearLayer 関数は完全一致で検索されるところの違いがある。

## 使用例

```
top.map.clearLayer('電子国土 XML.線');
// レイヤ名'電子国土 XML.線'をクリアする
```

## 関数名

**top.map.setLayerRGB(style,r,g,b)**

## 機能

指定するレイヤの色を変更する。

## 引数

色を変更したいレイヤ名と、表示する色を指定する。ここで指定するレイヤ名とは、電子国土 XML データ形式の<G I><dataset><layer><name>の内容と<G I><dataset><layer><style><name>の内容を、ピリオド「.」でつないだものである。

layer レイヤ名（完全一致で検索される）

r 赤色の強さ（0～255 の整数値）

g 緑色の強さ（0～255 の整数値）

b 青色の強さ（0～255 の整数値）

## 戻り値

true : 成功

false : 失敗（指定したレイヤ名が存在しない）

## 解説

指定されたレイヤ名の色を変更する。なお、色変更を反映させるには、openMap 関数を実行する必要がある。

## 使用例

```
tr = top.map.setLayerRGB('電子国土 XML.線',
,0,255,255);
```

top.map.openMap();

// レイヤ名'電子国土 XML.線'を黄色に変更し  
表示する

## 関数名

**top.map.setLayerHRGB(style,r,g,b)**

## 機能

指定するレイヤの塗りつぶし色、あるいはハッチングを行っている場合の前景色を変更する。

## 引数

色を変更したいレイヤ名と、表示する色を指定する。ここで指定するレイヤ名とは、電子国土 XML データ形式の<G I><dataset><layer><name>の内容と<G I><dataset><layer><style><name>の内容

を，ピリオド「.」でつないだものである。  
 layer レイヤ名（完全一致で検索される）  
 r 赤色の強さ（0～255の整数値）  
 g 緑色の強さ（0～255の整数値）  
 b 青色の強さ（0～255の整数値）

**戻り値****true** : 成功**false** : 失敗（指定したレイヤ名が存在しない）**解説**

指定されたレイヤ名の色を変更する。なお，色変更を反映させるには，openMap 関数を実行する必要がある。

**使用例**

```
tr = top.map.setLayerHRGB('電子国土 XML.面'
                           ,0,255,255);

top.map.openMap();
// レイヤ名'電子国土 XML.面'を黄色で塗りつぶし表示する
```

**関数名****top.map.setLayerBRGB(layer,r,g,b)****機能**

指定するレイヤがハッチングを行っている場合の背景色を変更する。

**引数**

色を変更したいレイヤ名と，表示する色を指定する。ここで指定するレイヤ名とは，電子国土 XML データ形式の<GI><dataset><layer><name>の内容と<GI><dataset><layer><style><name>の内容を，ピリオド「.」でつないだものである。

layer レイヤ名（完全一致で検索される）  
 r 赤色の強さ（0～255の整数値）  
 g 緑色の強さ（0～255の整数値）  
 b 青色の強さ（0～255の整数値）

**戻り値****true** : 成功**false** : 失敗（指定したレイヤ名が存在しない）**解説**

指定されたレイヤ名の背景色を変更する。なお，色変更を反映させるには，openMap 関数を実行する必要がある。

**使用例**

```
tr = top.map.setLayerBRGB('電子国土 XML.面'
                           ,200,200,200);

top.map.openMap();
// レイヤ名'電子国土 XML.面'の背景色をグレーに変更し表示する
```

**関数名****top.map.getLayers()****機能**

上乗せデータすべてのレイヤ名リストを取得す

る。

**引数**

引数はなし。

**戻り値**

レイヤ名の配列を返す。ここで返されるレイヤ名とは，電子国土 XML データ形式の<GI><dataset><layer><name>の内容と<GI><dataset><layer><style><name>の内容を，ピリオド「.」でつないだものである。

**解説**

この関数は，関数実行時に読み込まれている上乗せデータのレイヤを配列変数に代入して返す。レイヤ名は，電子国土 Web システム内部ではレイヤ名が使われているため，戻り値はレイヤ名のリストになる。

返されるレイヤ名は，読み込まれた順番に並んでいる。

**関数名****top.map.getDescription()****機能**

上乗せデータすべてのレイヤ説明文を取得する。

**引数**

引数はなし。

**戻り値**

説明文の配列を返す。返される配列の格納順番は，getLayers 関数で返されるレイヤの順番と同じである。

**解説**

この関数は，関数実行時に読み込まれている上乗せデータの<layer><description>要素に記述されたレイヤの説明文を，配列変数に代入して返す。

**関数名****top.map.getLayerDescription(layer)****機能**

指定するレイヤの説明文を取得する。

**引数**

レイヤ名を指定する。ここで指定するレイヤ名とは，電子国土 XML データ形式の<GI><dataset><layer><name>の内容と<GI><dataset><layer><style><name>の内容を，ピリオド「.」でつないだものである。

**戻り値**

レイヤの説明文を文字列で返す。

**解説**

この関数は，指定するレイヤの説明文を取得する。説明文は，電子国土 XML データ形式の<layer>要素の中に含まれる<description>要素で定義された文字列である。

**使用例****top.map.getLayerDescription('電子**

```
// レイヤ名'電子国土 XML.線'の説明文を取得する
```

**関数名****top.map.openDisplayLayerWindow()****機能**

現在、表示されているレイヤー一覧を表示する。

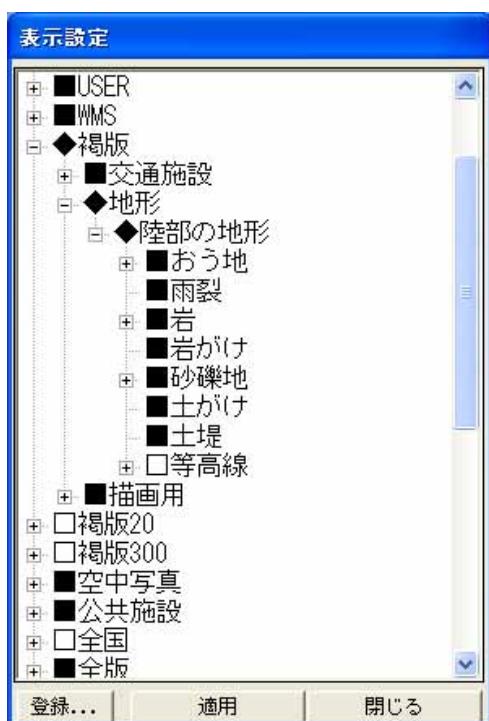


図 - 2 レイヤー一覧表示

**引数**

引数はなし。

**戻り値**

戻り値はなし。

**解説**

文字列の左にある **や** をクリックすることで、レイヤーの表示設定が変わる（**が非表示**、**が表示**で、**が下位レイヤーの一部表示**を意味する）。設定を変更したあと、「適用」を押すことで、電子国土上の表示が変更される。これにより、設定を変更したいレイヤーがどのレイヤーであるか調べることができる。

レイヤー名は、このウィンドウで出ているツリーを上から読んでつなげたものである。例えば、等高線の場合、「**褐版.地形.陸部の地形.等高線**」となる。

なお、map フレームで表示されている地図の種類が入れ替わっても、すでに表示されている表示設定ウィンドウには、**や** の変化が反映されない。そのため、拡大縮小で地図の種類が入れ替わったときには、表示設定ウィンドウを「閉じる」で閉じ、もう一度 openDisplayLayerWindow 関数を呼び出す必要がある。

この関数は、電子国土サイト作成者のためのツールとして提供しているもので、電子国土サイトの閲覧者向けに設計されたものではない。

**関数名****top.map.setMapColorWhitish(plus)****機能**

1/25,000 レベルの背景地図の表示色を薄く（明るく）する。

**引数**

表示をどの程度薄くするか値を指定する。値が大きいほど表示色が薄くなる。

**plus0 ~ 250 の整数値**

**戻り値**

戻り値はなし。

**解説**

表示されている 1/25,000 レベル背景地図の表示色を薄くする。setMapColorNormal 関数で初期状態に戻さずに、再度、この関数を実行すると、さらに表示色は薄くなる。色変更を反映させるには、openMap() 関数を実行する必要がある。

setMapColorNormal 関数で、初期状態に戻る。

**使用例**

```
top.map.setMapColorWhitish(100);
```

```
top.map.openMap();
```

// 1/25,000 レベル背景地図を指定した値(100)

だけ薄くし表示する

**関数名****top.map.setMapColorNormal()****機能**

setMapColorWhitish 関数で変更した表示色をもとに戻す。

**引数**

引数はなし。

**戻り値**

戻り値はなし。

**解説**

setMapColorWhitish 関数で、変更された 1/25,000 レベルの背景地図の表示色をもとに戻す。

## 2.6 印刷関連関数

印刷関連関数は、表示されている地図を印刷するための関数である。

**一覧**

- **printMap()**

画面のハードコピーをプリンタに出力する

- **printStatus()**

印刷用ダイアログが開かれているか調べる

**関数名****top.map.printMap()****機能**

map フレームに表示されている画面のハードコピーを、A4 サイズでプリンタに出力する。

**引数**

引数はなし。

**戻り値**

戻り値はなし。

**解説**

この関数が呼び出されると、図 - 3 のような印刷設定のダイアログが表示され、プリンタの選択、用紙の方向を選択し、プレビューの表示および印刷を行うことができる。



図 - 3 印刷設定ダイアログ

関数実行時に、map フレームに表示されている範囲を含む領域を、A4 用紙にハードコピーする。用紙の方向によって印刷される範囲は異なる。

A4 用紙以外のサイズへの出力および縮尺を指定しての印刷は、現時点では許可していない。

**関数名****top.map.printStatus()****機能**

印刷用のダイアログが開かれているか調べる。

**引数**

引数はなし。

**戻り値**

印刷用ダイアログの状態を返す。

0：印刷用のダイアログは開かれていません

1：印刷用のダイアログが開かれています

**解説**

`printMap` 関数によって印刷用のダイアログが開かれているか調べ、その状態を返す。

## 2.7 イベント関連関数

イベント関連関数は、マウスのイベントモードの切り替えとイベントの処理関数をイベントハンドラに登録する関数である。

**一覧**

- `setMouseMode(mode)`

マウスマードを切り替える

- `event(eventname,procname)`

指定するイベントの処理関数をイベントハンドラに登録する

**関数名****top.map.setMouseMode(mode)****機能**

マウスのイベント割り当てモードを切り替える。

**引数**

mode に、以下の文字列を指定することができます。

zoom 拡大縮小モード

selection 選択モード

pan 移動モード

measure 計測モード

edit 編集モード

**戻り値**

戻り値はなし。

**解説**

この関数を呼び出すと、その時点からマウスへのイベントの割り当てが変わることになる。

拡大縮小モードは、電子国土 Web システムのデフォルトのモードであり、地図の拡大 / 縮小を行うモードである。ズーミングモードともいう。

選択モードは、地図オブジェクトを選択するためのモードである。

移動モードは、地図のスクロールのためのモードで、マウスカーソルが掌の形になり、地図をつかんでずらす感覚でスクロールすることができる。

計測モードは、距離、面積計測用のダイアログが開く。

編集モードは、上乗せデータを編集することができるモードである。下の表に各モードでのマウスボタンへの機能割付けを示す。なお、どのモードでもマウスのホイールをクリックするとスクロールする機能が提供されている。

マウスマード [設定文字列]	マウスボタンへの機能割り付け			
	左	左 ドラッグ	右	右 ドラッグ
拡大縮小 [zoom]	拡大	範囲拡大	縮小	範囲縮小
選択 [selection]	選択	範囲選択		
移動 [pan]		移動		
計測 [measure]	経由点 決定		経由点 消去	
編集 [edit]	選択	移動	キャンセル	

**使用例**

```
top.map.setMouseMode('selection');
// 選択モードに設定する
```

**関数名**

```
top.map.event(eventname,procname)
```

**機能**

指定するイベントに対する処理関数をイベントハンドラに登録する。

**引数**

フックしたいイベントの名称 (eventname) と、それを処理する関数名 (procname) を指定する。

**戻り値**

戻り値はなし。

**解説**

電子国土 Web システムは、以下のイベントを処理することができる。

イベント名	内容
mousedown	マウスの左ボタンが押し下げられた
mouseup	マウスの左ボタンが解放された
rightdown	マウスの右ボタンが押し下げられた
rightup	マウスの右ボタンが解放された
dblclick	マウスがダブルクリックされた
move	マウスポインタが移動した
mapload	マップファイルがロードされた
repaint	再描画された
selection	一つのオブジェクトが選択された
mselection	複数のオブジェクトが選択された

mousedown, mouseup, rightdown, rightup, dblclick の各イベントでは、イベントが発生したマウスポインタの位置が経度、緯度の順で処理関数に渡される。

selection イベントでは、選択状態のオブジェクトが処理関数に渡される。

mselection イベントでは、選択状態のオブジェクトハンドルが処理関数に渡される。

この関数の使用方法に関する詳細は、「電子国土 Web システムプログラミングガイド」を参照のこと。

**使用例**

```
top.map.event('mousedown',
    'top.mouseleftdown');
// マウスの左ボタンが押されたとき、
mouseleftdown を実行する
```

**2.8 オブジェクト操作関数**

オブジェクト操作関数は、上乗せデータに関するオブジェクトを操作するための関数である。

**一覧**

- queryRect(left,bottom,right,top,layer)
 バッファリング矩形を定義し、その中に含まれるオブジェクトのリストを取得する
- queryWindow(layer)
 表示されている範囲に含まれるオブジェクトのリストを取得する
- queryLayer(layer)
 指定するレイヤのオブジェクトリストを取得する
- objLayer(obj)
 オブジェクトが属するレイヤ名を取得する
- objGeometry(obj)
 オブジェクトの座標値を取得する
- objRange(obj)
 オブジェクトの外接矩形の範囲を取得する
- objFieldNames(obj)
 オブジェクトの属性項目リストを取得する
- objFieldVal(obj,attribute)
 オブジェクトの属性値を取得する
- objType(obj)
 オブジェクトの図形型を取得する
- objName(obj)
 オブジェクトの name 値を取得する
- objFromList(objHandle)
 オブジェクトハンドルからオブジェクトのリストを取得する
- changeSelection(obj,redraw)
 オブジェクトを選択状態のオブジェクトに変更する
- appendSelection(obj,redraw)
 オブジェクトを選択状態のオブジェクトに追加する
- clearSelection(redraw)
 選択状態のオブジェクトをクリアにする
- selectionNum()
 選択状態のオブジェクトの個数を取得する
- selectionList()
 選択状態のオブジェクトハンドルを取得する
- selectionObj(index)
 選択状態の複数オブジェクトから指定する順位のオブジェクトを取得する
- deleteObject()
 選択状態のオブジェクトを削除する
- cutClipboard()
 選択状態のオブジェクトをクリップボードに切り取る
- copyClipboard()
 選択状態のオブジェクトをクリップボードにコピーする

- `pastClipboard()`  
クリップボードにコピーされたオブジェクトを貼り付ける
- `undoEdit()`  
`deleteObject` 等の関数の動作をやり直す
- `redoEdit()`  
`undoEdit()`でやり直した動作を再度実行する
- `saveJSGIXML(class)`  
オブジェクトを電子国土 XML データ形式で保存する

**関数名**

```
top.map.queryRect(left,bottom,
                   right,top [,layer])
```

**機能**

抽出処理を行う矩形を定義し、その中に含まれるオブジェクトのリストを取得する。

**引数**

抽出処理を行う矩形の南西角と北東角の経度緯度であり、いずれも度単位 10 進数の浮動小数点数である。傾いた矩形は定義できない。指定のレイヤからのみ取得する場合は、レイヤ名を指定する必要がある。レイヤ名は、電子国土 XML データ形式の`<GI><dataset><layer><name>`の内容である。

<code>left</code>	抽出範囲の西端の経度値
<code>bottom</code>	抽出範囲の南端の緯度値
<code>right</code>	抽出範囲の東端の経度値
<code>top</code>	抽出範囲の北端の緯度値
<code>layer</code>	レイヤ名（省略可能）（*が使用可能）

**戻り値**

指定した範囲に含まれるオブジェクトリストの配列を返す。

該当するオブジェクトが一つも無い場合は、長さが 1 でかつ空白がセットされた配列が返る。

**解説**

この関数は、読み込まれた上乗せデータの抽出処理（バッファリング）を行うための抽出範囲（バッファ）を定義し、その中に読み込まれている上乗せデータの`<selection>`が「on」となっているデータを抽出して、そのオブジェクトリストの配列を返す。抽出範囲にオブジェクトの一部が含まれると抽出の対象となる。

表示範囲からの抽出を行う場合は、`queryWindow` 関数を使うことができる。

**使用例**

```
objlist=top.map.queryRect(140.5,35.75,
                           140.75,36.25);
// 指定した範囲（西端、南端、東端、北端）の
// オブジェクトリストを取得する
```

**関数名**

```
top.map.queryWindow([layer])
```

**機能**

表示されている範囲に含まれるオブジェクトのリストを取得する。

**引数**

指定するレイヤからのみ取得する場合は、レイヤ名を定義する。レイヤ名は、電子国土 XML データ形式の`<GI><dataset><layer><name>`の内容である。

`layer` レイヤ名（省略可能）（\*が使用可能）

**戻り値**

表示されている範囲に含まれるオブジェクトリストの配列を返す。

該当するオブジェクトが一つも無い場合は、長さが 1 でかつ空白がセットされた配列が返る。

**解説**

この関数は、表示範囲に含まれ、読み込まれている上乗せデータの`<selection>`が「on」となっているデータを抽出して、そのオブジェクトリストの配列を返す。抽出範囲にオブジェクトの一部が含まれると抽出の対象となる。

指定範囲からの抽出を行う場合は、「`queryRect()`」関数を使うことができる。

**使用例**

```
objlist = top.map.queryWindow('電子
                                 國土 XML');
// 表示範囲でレイヤ名'電子國土 XML'のオブ
// ジェクトリストを取得する
```

**関数名**

```
top.map.queryLayer(layer)
```

**機能**

指定するレイヤのオブジェクトのリストを取得する。

**引数**

オブジェクトリストを取得するレイヤ名を指定する。レイヤ名は、電子国土 XML データ形式の`<GI><dataset><layer><name>`の内容である。

`layer` レイヤ名（\*が使用可能）

**戻り値**

オブジェクトリストの配列を返す。

**解説**

この関数は、指定したレイヤに含まれ、読み込まれている上乗せデータの`<selection>`が「on」となっているデータを抽出して、そのオブジェクトリストの配列を返す。

**使用例**

```
objlist = top.map.queryLayer('電子國土 XML');
// レイヤ名'電子國土 XML'のオブジェクトトリ
// ストを取得する
```

**関数名**

```
top.map.objLayer(obj)
```

**機能**

オブジェクトから属するレイヤ名を取得する .

**引数**

オブジェクトを指定する . 実際は , queryRect , queryWindow 関数等により得たオブジェクトリストの配列変数の要素を指定する .

**戻り値**

オブジェクトが属するレイヤ名を文字列で返す .

**解説**

この関数は , queryRect , queryWindow 関数等で取得したオブジェクトリストのうちの指定するオブジェクトが属するレイヤ名を取得する .

直接オブジェクトを引数に指定してそのレイヤ名を取得することもできる .

**使用例**

```
layname = top.map.objLayer(objlist[1]);
// オブジェクト'objlist[1]'が属するレイヤ名を取得する
```

**関数名**

**top.map.objGeometry(obj)**

**機能**

オブジェクトから座標値を取得する .

**引数**

オブジェクトを指定する . 実際は , queryRect , queryWindow 関数等により得たオブジェクトリストの配列変数の要素を指定する .

**戻り値**

オブジェクトの座標値が配列で返る .

- [0] 1点目の経度
- [1] 1点目の緯度
- [2] 2点目の経度
- [3] 2点目の緯度

...

**解説**

この関数は , queryRect , queryWindow 関数等で取得したオブジェクトリストのうちの指定するオブジェクトの座標値を取得する .

戻り値は , 度単位 10 進数の浮動小数点数 , 「1 点目の経度 , 1 点目の緯度 , 2 点目の経度 , 2 点目の緯度 ... 」の配列で返る .

直接オブジェクトを引数に指定してその座標値を取得することもできる .

**使用例**

```
xy = top.map.objGeometry(objlist[1]);
// オブジェクト'objlist[1]'の座標値を取得する
```

**関数名**

**top.map.objRange(obj)**

**機能**

オブジェクトから外接矩形の対角座標値を取得

する .

**引数**

オブジェクトを指定する . 実際は , queryRect , queryWindow 関数等により得たオブジェクトリストの配列変数の要素を指定する .

**戻り値**

オブジェクトの外接矩形の南西角と北東角の経度緯度を配列で返す .

- [0] 西端の経度値
- [1] 南端の緯度値
- [2] 東端の経度値
- [3] 北端の緯度値

**解説**

この関数は , queryRect , queryWindow 関数等で取得したオブジェクトリストのうちの指定するオブジェクトの外接矩形の対角座標値を取得する . 座標値は , 度単位 10 進数の浮動小数点数 , 「西端の経度値 南端の緯度値 東端の経度値 北端の緯度値」の順で返る .

直接オブジェクトを引数に指定してその外接矩形の対角座標値を取得することもできる .

**使用例**

```
kukei = top.map.objRange(objlist[1]);
// オブジェクト'objlist[1]'が外接する矩形の対角座標を取得する
```

**関数名**

**top.map.objFieldNames(obj)**

**機能**

オブジェクトから属性項目名称のリストを取得する .

**引数**

オブジェクトを指定する . queryRect , queryWindow 関数等で取得したオブジェクトリストの配列変数の要素を指定する .

**戻り値**

オブジェクトの<attribute>要素に記述されている「属性項目 = 属性値」の属性項目のリストを配列変数で返す .

**解説**

この関数は , 上乗せデータとして読み出した電子国土 XML データ形式の<attribute>要素に記述されている「属性項目 1 = 属性の値 1 , 属性項目 2 = 属性の値 2 , ... 以下連続 ... 」の属性項目のリストを配列変数として取得する .

**使用例**

```
att = top.map.objFieldNames(objlist[1]);
// オブジェクト'objlist[1]'の属性項目を取得する
```

**関数名**

**top.map.objFieldVal(obj,attribute)**

**機能**

オブジェクトから属性値を取得する .

**引数**

オブジェクトと , 属性項目名を文字列で指定する . オブジェクトは , queryRect , queryWindow 関数等で取得したオブジェクトリストの配列変数の要素を指定する .

**obj** オブジェクト  
**attribute** 属性項目名

**戻り値**

オブジェクトの<attribute>要素に記述されている「属性項目 = 属性値」の属性値を文字列で返す .

**解説**

この関数は , 上乗せデータとして読み出した電子国土 XML データ形式の<attribute>要素に記述されている「属性項目 1 = 属性の値 1 , 属性項目 2 = 属性の値 2 , . . . 以下連続 . . . 」で , 指定された属性項目名の属性値を文字列として取得する .

**使用例**

```
att = top.map.objFieldVal(objlist[1],  
                           '属性 1');  
// オブジェクト'objlist[1]'の属性項目'属性  
1'の属性値を取得する
```

**関数名**

**top.map.objType(obj)**

**機能**

オブジェクトの図形型を取得する .

**引数**

オブジェクトを指定する . queryRect , queryWindow 関数等で取得したオブジェクトリストの配列変数の要素を指定する .

**戻り値**

以下の図形型の文字列を返す .

<b>string</b>	線型
<b>polygon</b>	面型
<b>circle</b>	円型
<b>symbol</b>	点型
<b>text</b>	注記型
<b>bmp</b>	画像型

**解説**

指定するオブジェクトの図形型を取得する .

**使用例**

```
zukeitype = top.map.objType(objlist[1]);  
// オブジェクト'objlist[1]'の図形型を取得  
する
```

**関数名**

**top.map.objName(obj)**

**機能**

オブジェクトの name 値を取得する .

**引数**

オブジェクトを指定する . queryRect , queryWindow 関数等で取得したオブジェクトリストの配列変数の要素を指定する .

**戻り値**

取得した文字列を返す .

**解説**

上乗せデータとして読み出した電子国土 XML データ形式のオブジェクト<name>に記述された文字列を返す .

**使用例**

```
objname = top.map.objName(objlist[1]);  
// オブジェクト'objlist[1]'の<name>値を取  
得する
```

**関数名**

**top.map.objFromList(objhandle)**

**機能**

オブジェクトハンドルからオブジェクトのリストを取得する .

**引数**

オブジェクトハンドルを指定する .

**戻り値**

オブジェクトハンドルの各要素から得たオブジェクトをリストにした配列変数を返す . オブジェクトが含まれない場合には ,長さ 0 の配列が返る .

**解説**

引数として与えるオブジェクトハンドルは , 複数オブジェクトを選択したときに発生する mselection イベントで渡される値または selectionList 関数で取得した値のみである .

**使用例**

```
objh = top.map.selectionList();  
objlist = top.map.objFieldVal(objh);  
// selectionList 関数で取得したオブジェクト  
ハンドルを引数とし , オブジェクトリストを  
取得する
```

**関数名**

**top.map.changeSelection(obj, redraw)**

**機能**

指定するオブジェクトを選択状態に変更する .

**引数**

オブジェクトと表示の更新を指定する .

**obj** オブジェクト

**redraw** true:表示を更新する

false:表示を更新しない

**戻り値**

戻り値はなし .

**解説**

指定するオブジェクトを選択状態に変更する . 既に選択状態のオブジェクトがある場合 , 選択状

態のオブジェクトは解除され、指定するオブジェクトのみ選択状態となる。

この関数は、マウス操作からではなく、プログラムからオブジェクトを選択状態にできる。

#### 使用例

```
top.map.changeSelection(objList[0], true);
// オブジェクト 'objList[0]' を選択状態にする
```

#### 関数名

**top.map.appendSelection(obj, redraw)**

#### 機能

指定するオブジェクトを選択状態のオブジェクトに追加する。

#### 引数

オブジェクトと表示の更新を指定する。

obj オブジェクト

redraw true:表示を更新する

false:表示を更新しない

#### 戻り値

戻り値はなし。

#### 解説

指定するオブジェクトを選択状態のオブジェクトに追加する。既に選択状態のオブジェクトには変更はなく選択状態のままとなる。

この関数は、マウス操作からではなく、プログラムから選択状態のオブジェクトに追加することができる。

SHIFT キーを押した状態での、マウスクリックによる選択オブジェクト追加操作に相当する。

#### 使用例

```
top.map.appendSelection(objList[0], true);
// 選択状態のオブジェクトに、オブジェクト
'objList[0]' を追加し選択状態にする
```

#### 関数名

**top.map.clearSelection(redraw)**

#### 機能

選択状態のオブジェクトをクリアにする。

#### 引数

表示を更新するか指定する。

redraw true:表示を更新する

false:表示を更新しない

#### 戻り値

戻り値はなし。

#### 解説

選択状態のオブジェクトを選択されていない状態にする。

この関数は、マウス操作からではなく、プログラムから選択状態のオブジェクトにクリアすることができます。

#### 使用例

```
top.map.clearSelection(true);
// 選択状態のオブジェクトをクリアし、表示を
更新する
```

#### 関数名

**top.map.selectionNum()**

#### 機能

選択されているオブジェクトの個数を取得する。

#### 引数

引数はなし。

#### 戻り値

選択されているオブジェクトの個数を返す。

#### 解説

この関数は、選択されているオブジェクトの個数を取得する。

#### 関数名

**top.map.selectionList()**

#### 機能

選択状態のオブジェクトハンドルを取得する。

#### 引数

引数はなし。

#### 戻り値

選択状態のオブジェクトハンドルを返す。

#### 解説

選択状態のオブジェクトハンドルを取得する。

この関数で取得したオブジェクトハンドルは、`objFromList` 関数のみで使用可能である。

#### 関数名

**top.map.selectionObj(index)**

#### 機能

選択状態の複数オブジェクトから指定する順位のオブジェクトを取得する。

#### 引数

取得するオブジェクトのリスト内順位を指定する。先頭は 0 からとなる。

#### 戻り値

オブジェクトを返す。

#### 解説

この関数は、選択されている複数オブジェクトから指定する順位のオブジェクトを取得する。取得した値は、オブジェクトを指定する `objLayer`, `objGeometry` 関数等で利用できる。

#### 使用例

```
ob = top.map.selectionObj(1);
// 選択されているオブジェクトから 2 番目の
オブジェクトを取得する
```

#### 関数名

**top.map.deleteObject()**

#### 機能

選択状態のオブジェクトを削除する。

**引数**

引数はなし .

**戻り値**

戻り値はなし .

**解説**

この関数は、選択状態のオブジェクトを削除する。選択されたオブジェクトが無い場合は、実行されない。編集履歴が保持されるため、undoEdit 関数によるやり直しや、redoEdit 関数によるやり直しの再実行が可能である。

**関数名**

**top.map.cutClipboard()**

**機能**

選択状態のオブジェクトをクリップボードに切り取る .

**引数**

引数はなし .

**戻り値**

戻り値はなし .

**解説**

この関数は、選択状態のオブジェクトをクリップボードに切り出す。切り出したオブジェクトは表示されなくなる。

**関数名**

**top.map.copyClipboard()**

**機能**

選択状態のオブジェクトをクリップボードにコピーする .

**引数**

引数はなし .

**戻り値**

戻り値はなし .

**解説**

この関数は、選択状態のオブジェクトをクリップボードにコピーする。cutClipboard 関数と異なり、切り出したオブジェクトは表示されたままとなる。

**関数名**

**top.map.pasteClipboard()**

**機能**

クリップボードのオブジェクトを貼り付ける .

**引数**

引数はなし .

**戻り値**

戻り値はなし .

**解説**

この関数は、cutClipboard, copyClipboard 関数で、クリップボードにコピーされたオブジェクトを貼り付ける。

**関数名**

**top.map.undoEdit()**

**機能**

deleteObject, cutClipboard, pastClipboard 関数の動作をやり直す .

**引数**

引数はなし .

**戻り値**

戻り値はなし .

**解説**

この関数は、deleteObject, cutClipboard, pastClipboard 関数で行った動作をやり直す。やり直しを取り消すには、redoEdit 関数を使用する。いったんマウスモードを変更すると動作しなくなる。

**関数名**

**top.map.redoEdit()**

**機能**

undoEdit 関数でやり直した動作を再度実行する .

**引数**

引数はなし .

**戻り値**

戻り値はなし .

**解説**

この関数は、undoEdit 関数でやり直した関数の動作を再実行する。いったんマウスモードを変更すると動作しなくなる。

**関数名**

**top.map.saveJSGIXML(class)**

**機能**

オブジェクトを電子国土 XML データ形式で保存する .

**引数**

オブジェクトのクラス名を指定する。openJSGIXML 関数により読み込まれたオブジェクトのクラス名は、デフォルトでは "JSGIXML" である。複数指定する場合は、ブランクで区切る .

**戻り値**

戻り値はなし .

**解説**

関数を実行すると xml ファイルを選択するダイアログが表示され、オブジェクトを電子国土 XML データ形式で保存される。揮発レイヤ関連関数で作成した揮発図形は保存できない。openJSGIXML 関数で読み込まれたいくつかのファイルを、saveJSGIXML 関数を利用することにより、一つにまとめることもできる。

**使用例**

```
top.map.saveJSGIXML("JSGIXML");
// openJSGIXML 関数で読み込まれたクラス名
```

"JSGIXML"のオブジェクトを電子国土 XML データ形式でファイルに保存する

## 2.9 振発レイヤ関連関数

振発レイヤ関連関数は、振発レイヤに図形を描画する機能を提供する。振発レイヤとは、メモリ上に一時的に保持されるレイヤで、ページをリロードすると消えてしまう。

振発レイヤに描画されたオブジェクトは、`openJSGIXML()`, `openJSGIXMLEx()`で読み込まれたオブジェクトとは異なり、オブジェクト操作関数の引数に指定することはできない。

### 一覧

- `draw_group(groupname)`  
図形グループを宣言する
- `draw_level(level)`  
図形を表示する地図レベルを指定する
- `draw_xyunit(unit)`  
図形を描画する単位を宣言する
- `draw_width(width)`  
図形の描画線幅を指定する
- `draw_color(r,g,b)`  
図形の描画色を指定する
- `draw_color2(r,g,b)`  
図形の塗りつぶし色を指定する
- `draw_circle(cx,cy,r)`  
円を描画する
- `draw_disc(cx,cy,r)`  
塗りつぶし円を描画する
- `draw_line([xy 配列])`  
折れ線を描画する
- `draw_poly([xy 配列])`  
ポリゴンを描画する
- `draw_symbol(url,size,x,y)`  
シンボルを描画する
- `draw_text(font,style,size,origin,text,x,y)`  
文字列を描画する
- `draw_display(group,onoff)`  
図形グループ毎に表示を ON / OFF する
- `draw_remove(group)`  
図形グループ毎に図形を削除する

### 関数名

`top.map.draw_group(groupname)`

#### 機能

振発レイヤに描画する図形のグループ名を宣言する。

#### 引数

振発レイヤに描画する図形のグループ名を文字列で指定する。

#### 戻り値

戻り値はなし。

### 解説

揮発レイヤに描画する図形のグループ名を宣言する。グループ名は擬似的にレイヤ名として扱われ、一度図形グループが宣言されると、次の図形グループが宣言されるまでは現在のグループのレイヤで描画が行われる。

グループ名を宣言することによって、揮発レイヤに描画される中からグループ毎に表示を ON/OFF したり、一括して削除したりすることができる。

### 使用例

```
top.map.draw_group('kihatsu');
// 図形グループ名を'kihasu'に指定する
```

### 関数名

`top.map.draw_level(level)`

#### 機能

揮発レイヤに描画する図形の地図レベルを指定する。

#### 引数

表示したい背景地図のレベルを文字列で指定する。

- |       |   |                    |
|-------|---|--------------------|
| all   | : | すべてのレベルの地図で表示する    |
| 10000 | : | 10000k レベルの地図で表示する |
| 3000  | : | 3000k レベルの地図で表示する  |
| 200   | : | 200k レベルの地図で表示する   |
| 50    | : | 50k レベルの地図で表示する    |
| 25    | : | 25k レベルの地図で表示する    |
| 0.5   | : | 大縮尺レベルの地図で表示する     |

#### 戻り値

戻り値はなし。

### 解説

カレントグループの描画内容が表示される背景地図を指定する。`'all'`が指定されると、すべてのレベル上で表示される。`'10000'`から`'0.5'`までの文字列を表示すると、それぞれのレベルの背景地図上で表示され、`draw_display` 関数による表示 ON/OFF は、指定された背景地図上で有効である。また、複数の背景地図レベルを指定するときは、`'10000,3000'`のように、カンマで複数のレベルをつなげた文字列を引数として指定する。デフォルトは`'25'`である。

### 使用例

```
top.map.draw_level('25,50');
// 25k レベルと 50k レベルのときに図形を描画する
```

### 関数名

`top.map.draw_xyunit(unit)`

#### 機能

揮発レイヤに描画する図形を指定する際に使用

する単位を宣言する .

#### 引数

揮発レイヤに描画する図形を指定する際に使用する単位を指定する .

単位として指定できるのは ,以下のものである .

引数文字列 内容

m	メートル
mm	ミリメートル
deg	度単位
screen	画面のピクセル単位

#### 戻り値

戻り値はなし .

#### 解説

揮発レイヤに描画する図形を指定する際に使用する単位を宣言する .一度宣言した単位は , 次に単位が宣言されるまで有効である .

「m」「mm」は , 現実世界の長さで表され , そのときの表示縮尺に応じて動的に表示サイズが調整される .

「deg」は , 経度 , 緯度を直接指定する際に使用するもので , 10進数の度単位で表される .例えば , 133 度 30 分は , 133.5 という表記になる .

「screen」は , 画面のピクセル単位で位置を指定したい場合に使用する .現実世界と関係ない単位であり , 例えば画面中心を表す十字線を描くなど , どの場所の地図が表示されているかに関係なく map フレームの決まった位置に図形を描画することができる .

#### 使用例

```
top.map.draw_xyunit('screen');
// 使用する単位を画面のピクセル単位に設定
する
```

#### 関数名

**top.map.draw\_width(width)**

#### 機能

揮発レイヤに描画する線の線幅をピクセル数で指定する .

#### 引数

揮発レイヤに描画する線の線幅 ( ピクセル数 ) を表す整数を指定する .

#### 戻り値

戻り値はなし .

#### 解説

揮発レイヤに描画する線の線幅をピクセル数で指定する .draw\_xyunit 関数で宣言した座標単位とは関係なく , 引数は , 常にピクセル数で指定しなければならない .

#### 使用例

```
top.map.draw_width(10);
top.map.draw_circle(250,150,50);
```

// 線幅が 10 ピクセルの円を描く

#### 関数名

**top.map.draw\_color(r,g,b)**

#### 機能

揮発レイヤに描画する図形の描画色を指定する .

#### 引数

描画色の三原色の値を指定する .

r 赤色の強さ ( 0 ~ 255 の整数値 )

g 緑色の強さ ( 0 ~ 255 の整数値 )

b 青色の強さ ( 0 ~ 255 の整数値 )

#### 戻り値

戻り値はなし .

#### 解説

揮発レイヤに描画する図形の描画色を指定する .

赤 , 緑 , 青それぞれの値を整数値で指定する .

#### 使用例

```
top.map.draw_color(0,0,255);
top.map.draw_line(["250 250", "350 250"]);
// 青色の線を描く
```

#### 関数名

**top.map.draw\_color2(r,g,b)**

#### 機能

揮発レイヤに描画する図形の塗りつぶし色を指定する .

#### 引数

塗りつぶし色の三原色の値を指定する .

r 赤色の強さ ( 0 ~ 255 の整数値 )

g 緑色の強さ ( 0 ~ 255 の整数値 )

b 青色の強さ ( 0 ~ 255 の整数値 )

#### 戻り値

戻り値はなし .

#### 解説

揮発レイヤに描画する図形の塗りつぶし色を指定する .赤 , 緑 , 青それぞれの値を整数値で指定する .

#### 使用例

```
top.map.draw_color2(0,255,0);
top.map.draw_disc(140.0875,36.1042,300);
// 緑色で塗りつぶした円を描く
```

#### 関数名

**top.map.draw\_circle(cx,cy,r)**

#### 機能

中心座標と半径を指定して揮発レイヤに円を描画する .

#### 引数

円の中心座標の経緯度値と円の半径を指定する .

cx 円中心の座標 ( 経度又は X 値 )

cy 円中心の座標 ( 緯度又は Y 値 )

r 半径 ( メートル又はピクセル単位 )

#### 戻り値

戻り値はなし .

#### 解説

中心座標と半径を指定して揮発レイヤに円を描画する .

中心座標 ( cx,xy ) を指定する際の単位は , draw\_xyunit 関数で「screen」以外を指定した場合 , 半径は常にメートル単位である .

#### 使用例

```
top.map.draw_xyunit('screen');
top.map.draw_circle(250,150,50);
// 円中心座標(250,150)に半径 50 ピクセルの円を描く
```

#### 関数名

**top.map.draw\_disc(cx,cy,r)**

#### 機能

中心座標と半径を指定して揮発レイヤに塗りつぶし円を描画する .

#### 引数

円の中心座標の経緯度値と塗りつぶし円の半径を指定する .

cx 円中心の座標 ( 経度又は X 値 )

cy 円中心の座標 ( 緯度又は Y 値 )

r 半径 ( メートル又はピクセル単位 )

#### 戻り値

戻り値はなし .

#### 解説

中心座標と半径を指定して揮発レイヤに塗りつぶし円を描画する .

中心座標 ( cx,xy ) を指定する際の単位は , draw\_xyunit 関数で「screen」以外を指定した場合 , 半径は常にメートル単位である .

#### 使用例

```
top.map.draw_xyunit('deg');
top.map.draw_disc(140.087,36.104,300);
// 円中心座標東経 140.087 度 , 北緯 36.104 度に , 半径 300 メートルの塗りつぶし円を描く
```

#### 関数名

**top.map.draw\_line([xy 配列])**

#### 機能

折れ線を描画する . 引数には , 2 つ以上の頂点座標を順番に指定する .

#### 引数

線を構成する点座標の配列変数を指定する . 各座標は , と Y が組みにならなければならぬ . 配列を引数に渡した場合 , 添え字の 0 を含む偶数番目が経度座標値 , 奇数番目が緯度座標値として扱われる .

#### 戻り値

戻り値はなし .

#### 解説

折れ線を描画する . 引数には , 2 つ以上の頂点座標を順番に指定する . 頂点座標の単位は , draw\_xyunit 関数で指定した単位である .

#### 使用例

```
top.map.draw_line(["250 250","350 250"]);
// 直線を引く
```

#### 関数名

**top.map.draw\_poly([xy 配列])**

#### 機能

ポリゴンを描画する . 引数には , 3 つ以上の頂点座標を順番に指定する .

#### 引数

ポリゴンを構成する点座標の配列変数を指定する . 各座標は , と Y が組みにならなければならぬ . また , ポリゴンの性質上 , 少なくとも 3 点を指定しなければならない . 配列を引数に渡した場合 , 添え字の 0 を含む偶数番目が経度座標値 , 奇数番目が緯度座標値として扱われる .

#### 戻り値

戻り値はなし .

#### 解説

ポリゴンを描画する . 引数には , 3 つ以上の頂点座標を順番に指定する .

最初と最後の頂点の座標が一致していない場合 , 最初と最後の頂点を結ぶ線が自動的に発生し , 閉領域となる . 頂点座標の単位は , draw\_xyunit 関数で指定した単位である .

内部は塗りつぶし色で塗りつぶされるが , 透過状態になる .

#### 使用例

```
top.map.draw_color(0,0,255);
top.map.draw_color2(0,255,0);
top.map.draw_poly(["250 250","350 250"
,"350 350","250 250"]);
// 縁が青色で 緑で塗りつぶされた四角形を描く
```

#### 関数名

**top.map.draw\_symbol(url,size,x,y)**

#### 機能

シンボルを描画する .

#### 引数

url シンボルデータの URL

size シンボルの画像サイズ ( ピクセル単位 )

x シンボルの描画位置 ( 経度又は X 座標 )

y シンボルの描画位置 ( 緯度又は Y 座標 )

#### 戻り値

戻り値はなし .

#### 解説

シンボルを描画する . シンボルに使用できる画像の形式は , 現時点では bmp のみである . シンボ

ル中の `rgb` 値が(255,255,255)の部分は透過色となり,白色は(254,254,254)で作成する必要がある。

シンボルの画像サイズは,ピクセル単位である。画面の表示縮尺に関係なく同じサイズで描画され,サイズは変わらない。

シンボルの描画位置は,画像の中心座標となり,座標単位は,`draw_xyunit` 関数で指定した単位である。

#### 使用例

```
top.map.draw_xyunit('screen');
top.map.draw_symbol('http://domain/image/
                     シンボル.bmp',50,300,200);
// 座標(300,200)にシンボル.bmpを描く
```

#### 関数名

```
top.map.draw_text(font,style,size,origin,te
xt,x,y)
```

#### 機能

テキストを描画する。

#### 引数

font	フォント名称文字列
style	フォント太さ文字列
size	文字のサイズ(ピクセル)
origin	文字の配置位置を示す2文字からなる文字列(下図参照)
text	描画文字列
x	文字列左下隅の描画位置(経度又はX座標)
y	文字列左下隅の描画位置(緯度又はY座標)

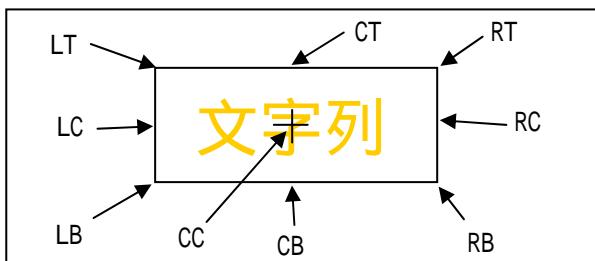
#### 戻り値

戻り値はなし。

#### 解説

テキストを描画する。フォント名称文字列には,使用的フォント名を,例えば「MS Gothic」等で指定する。フォント太さ文字列は、「標準」や「太字」などを指定する。文字のサイズは,画面のピクセル数で指定し,ポイント数ではない。

文字列の描画位置の座標の単位は,`draw_xyunit` 関数で指定した単位である。



#### 使用例

```
top.map.draw_text('MS Gothic','太字',
                  20,'LT','電子国土',150,200);
// 文字の左上座標(150,200)に,書体MS Gothic,
// サイズ20ピクセル,太字の文字を描く
```

#### 関数名

```
top.map.draw_display(group,onoff)
```

#### 機能

グループ毎に表示をON/OFFする。

#### 引数

group `draw_group` 関数で宣言したグループ名

onoff 表示/非表示を表すフラグ

0:非表示

1:表示

#### 戻り値

戻り値はなし。

#### 解説

揮発レイヤのグループ毎に表示をON/OFFする。

#### 使用例

```
top.map.draw_display('kihatsu',0);
```

// グループ名('kihatsu')に描画されている図形を非表示にする

#### 関数名

```
top.map.draw_remove(group)
```

#### 機能

グループ毎に図形を削除する。

#### 引数

group `draw_group` 関数で宣言したグループ名

#### 戻り値

戻り値はなし。

#### 解説

揮発レイヤのグループ毎に図形を削除する。指定された図形グループで描画されているオブジェクトは,すべてクリアされる。

#### 使用例

```
top.map.draw_remove('kihatsu');
```

// グループ名('kihatsu')に描画されている図形を削除する

## 2.10 プラグイン関連関数

プラグイン関連関数は,電子国土Webシステムのプラグインインストール状況,バージョン等を調べる関数である。

#### 一覧

- `pluginVersion()`  
インストールされているプラグインのバージョンを調べる
- `testplugin()`  
プラグインがインストールされているか調べる
- `doInstall()`  
IE用の電子国土プラグインインストーラーを

ダウンロードする

- `checkXML4()`  
MSXML4 がインストールされているか調べる

関数名

`top.map.pluginVersion()`

機能

インストールされているプラグインのバージョンを調べる。

引数

引数はなし。

戻り値

プラグインバージョンを表す文字列で返す。

解説

この関数は、クライアントにインストールされている電子国土 Web システムのプラグインバージョンを調べる。

関数名

`top.map.testplugin()`

機能

プラグインがインストールされているか調べる。

引数

引数はなし。

戻り値

プラグインのインストール状態を返す。

`true` : インストールされている

`false` : インストールされていない

解説

この関数は、クライアントに電子国土 Web システムのプラグインがインストールされているか調べる。

関数名

`top.map.doInstall()`

機能

IE 用の電子国土 Web システムプラグインインストーラをダウンロードする。

引数

引数はなし。

戻り値

戻り値はなし。

解説

ファイルを保存するための確認ダイアログが表示され、IE 用の電子国土 Web システムプラグインインストーラをダウンロードすることができる。ダウンロードされるファイル名は、`webtis_[バージョン].exe` である。

関数名

`top.map.checkXML4()`

機能

MSXML4 がインストールされているか調べる。

引数

引数はなし。

戻り値

`true` : インストールされている

`false` : インストールされていない(上記以外)

解説

電子国土 Web システムで利用する電子国土 XML データを扱うために MSXML4 が必要であり、クライアントに MSXML4 がインストールされているか調べる。

## 2.11 グラフ関連関数

グラフ関連関数は、ポップアップウィンドウへのグラフ描画機能を提供する。

グラフ関連機能は、グラフオブジェクトとして実装されており、そのメソッドを呼び出すことによってグラフを描画する。

一覧

- `graph()`  
グラフオブジェクトを作成する
- `graph.newdata()`  
データオブジェクトを作成する
- `graph.title(title)`  
グラフのタイトルを設定する
- `graph.size(w,h)`  
グラフウィンドウのサイズを設定する
- `graph.print(bool)`  
印刷ボタンを表示するかどうかを設定する
- `graph.xstep(step)`  
X 軸の目盛り刻み値を設定する
- `graph.yscale(min,max,split)`  
Y 軸の範囲と目盛り分割数を設定する
- `graph.y2scale(min,max,split)`  
Y 2 軸の範囲と目盛り分割数を設定する
- `graph.xtitle(title)`  
X 軸タイトルを設定する
- `graph.ytitle(title)`  
Y 軸タイトルを設定する
- `graph.y2title(title)`  
Y 2 軸タイトルを設定する
- `graph.xscalelabel(labelArray)`  
X 軸の目盛りラベルを設定する
- `graph.yscalelabel(labelArray)`  
Y 軸の目盛りラベルを設定する
- `graph.y2scalelabel(labelArray)`  
Y 2 軸の目盛りラベルを設定する
- `graph.create(title,x,y,DataNameArray)`  
グラフウィンドウを表示
- `graph.close()`  
グラフウィンドウを閉じる
- `graph.sheet(n)`

グラフの下に表を表示する

- `data.graph(kind)`  
グラフタイプ(棒, 折れ線, シンボル)を設定する
- `data.value(valueArray)`  
データ配列を設定する
- `data.num()`  
データ数を返す
- `data.range()`  
データの範囲を返す
- `data.color(rgb,rgb2)`  
グラフの表示色を設定する
- `data.linesymbol(type)`  
折れ線グラフのシンボルを設定する
- `data.connect(bool)`  
折れ線グラフの結線指示
- `data.weight(w)`  
棒グラフの幅を設定する
- `data.symbols(symbolArray)`  
シンボルグラフに表示するシンボルを設定する
- `data.symbolvalues(thresholdArray)`  
データに対応するシンボルをシンボル配列から選択するための閾値を設定する
- `data.symbolvaluekind(kind)`  
閾値種別を設定する
- `data.xshift(shift)`  
グラフ描画位置をシフトさせる
- `data.yaxis(axisno)`  
Y軸を選択する

関数名

`top.map.graph()`

機能

グラフオブジェクトを作成する。

引数

引数はなし。

戻り値

グラフオブジェクトを返す。

解説

グラフを表示する際に使用するグラフオブジェクトを作成するコンストラクタである。1つのグラフオブジェクトが1つのポップアップウィンドウに相当し、グラフオブジェクトはメモリが許す限りのデータオブジェクトを持つことができる。  
グラフオブジェクトの削除には`,delete`を用いる。

関数名

`graph.newdata(name)`

機能

データオブジェクトを作成する。

引数

データオブジェクトの名前を文字列で指定する。

戻り値

データオブジェクトへのハンドルを返す。

解説

グラフに表示するデータ系列を表すデータオブジェクトを作成するコンストラクタである。データオブジェクトは、自分自身をどのようにグラフ表示するかを示すプロパティと、データである。

プロパティを設定するためには、データオブジェクトのメソッドを使用する。

関数名

`graph.title(title)`

機能

グラフオブジェクトのタイトルプロパティを設定する。

引数

グラフのタイトルとする文字列を指定する。

戻り値

引数を省略した場合、現在の設定値を返す。

解説

グラフに表示されるタイトルを設定する。

関数名

`graph.size(w,h)`

機能

グラフを表示するポップアップウィンドウのサイズを設定する。

引数

グラフのタイトルとする文字列を指定する。

w ウィンドウの幅を示す整数値

h ウィンドウの高さを示す整数値

戻り値

引数を省略した場合、現在の設定値を返す。

解説

グラフを表示するポップアップウィンドウのサイズを指定する。hに0を設定すると、グラフは表示されない。

関数名

`graph.print(bool)`

機能

グラフウィンドウを印刷するボタンを表示するかどうかを設定する。

引数

印刷ボタンを表示するかどうかを Boolean 値で指定する。

true 表示する

false 表示しない

戻り値

引数を省略した場合、現在の設定値を返す。

解説

印刷ボタンは、デフォルトで true に設定されて

おり、右上隅に印刷ボタンが表示される。引数に `false` を指定すると、表示されなくなる。

印刷は、グラフウィンドウ内のみが出力される。  
プリンタによっては線が極細になることがある。

#### 関数名

`graph.xstep(step)`

#### 機能

軸のメモリの刻み値を設定する。

#### 引数

刻み値を表す 1 以上の整数を指定する。

#### 戻り値

引数を省略した場合、現在の設定値を返す。

#### 解説

データオブジェクトが持つデータ数が多いと、  
X 軸の目盛が混み合って判読不能となる。そのよ  
うな場合に、ある程度の間隔に X 軸の目盛表示を  
省略することによってグラフを見やすくするために  
を使用する。

#### 関数名

`graph.yscale(min,max,split)`

`graph.y2scale(min,max,split)`

#### 機能

Y 及び Y 2 軸の範囲と目盛分割数を設定する。

#### 引数

`min` 最小値

`max` 最大値

`split` メモリ刻み数（1 以上の整数）

#### 戻り値

引数を省略した場合、現在の設定値を返す。

#### 解説

Y 軸（左側の Y 軸）と Y 2 軸（右側の Y 軸）の  
値の範囲と目盛数を指定する。X 軸の目盛数はデ  
ータオブジェクトが持つデータ数によって自動的  
に決まるが、Y 軸の目盛数はそれぞれ指定しなけ  
ればならない。

#### 関数名

`graph.xtitle(title)`

`graph.ytitle(title)`

`graph.y2title(title)`

#### 機能

それぞれ、X 軸、Y 軸、Y 2 軸のタイトルを設  
定する。

#### 引数

タイトルとなる文字列を指定する。

#### 戻り値

引数を省略した場合、現在の設定値を返す。

#### 解説

`graph.xtitle` 関数は X 軸の `graph.ytitle` 関数  
は Y 軸の、`graph.y2title` 関数は Y 2 軸のタイ  
トルを設定する。

Y 軸及び Y 2 軸のタイトルは縦書きされる。

#### 関数名

`graph.xscalelabel(labelArray)`

`graph.yscalelabel(labelArray)`

`graph.y2scalelabel(labelArray)`

#### 機能

それぞれ、X 軸、Y 軸、Y 2 軸の目盛ラベルを  
設定する。

#### 引数

目盛ラベルとなる文字列配列を指定する。

#### 戻り値

引数を省略した場合、現在の設定値を返す。

#### 解説

`graph.xscalelabel` 関数は X 軸の、  
`graph.yscalelabel` 関数は Y 軸の、  
`graph.y2scalelabel` 関数は Y 2 軸の目盛に表示  
するラベル文字を設定する。

X 軸のラベル数は、データオブジェクトのデ  
ータ数で自動的に決まる。引数の配列変数に渡す文  
字列の数は、データ数と一致していなければならない。`graph.xstep` 関数で目盛を間引いている場  
合は、間引いたところにはカンマ「,」を連続して  
指定する。

Y 及び Y 2 軸のラベル数は、`graph.yscale`、  
`graph.y2scale` 関数で指定された数となる。

#### 関数名

`graph.sheet(n)`

#### 機能

グラフウィンドウのグラフ下に表を表示する。

#### 引数

表示する表の行数を整数で指定する。

#### 戻り値

引数を省略した場合、現在の設定値を返す。

#### 解説

グラフ化したデータの値を表示する表をグラフ  
ウィンドウのグラフの下に表示する。表の最も左  
の列に `graph.xscalelabel` 関数で渡した  
labelArray の各要素が表示され、そこから右にデ  
ータオブジェクトのデータ値が表示される。

引数で指定した行数よりもデータの数が多い場  
合は、自動的にスクロールバーが付加される。

#### 関数名

`graph.create(title,x,y,DataNameArray)`

#### 機能

グラフウィンドウを生成し、プロパティに従つ  
てタイトルや目盛軸を表示し、さらに指定するデ  
ータオブジェクトのプロパティに従つてグラフ及  
び表を描画する。

#### 引数

`title` ウィンドウのタイトル

x ウィンドウを表示する画面上の横方向の位置（画面座標）  
y ウィンドウを表示する画面上の縦方向の位置（画面座標）  
DataNameArray データオブジェクト名の配列

**戻り値**

ウィンドウの幅と高さを返す。

**解説**

必要なプロパティを設定後、グラフを表示するときにこの関数を使用する。データオブジェクト名の配列は、graph.newdata 関数で指定した引数の配列である。

データオブジェクトは、メモリが許す限りいくつでも指定できる。

**関数名**

**data.graph(kind)**

**機能**

データオブジェクトをグラフとして表示する種類を設定する。

**引数**

グラフの種類を以下のいずれかで指定する。

bar	棒グラフ
line	折れ線グラフ
symbol	シンボルグラフ

**戻り値**

引数を省略した場合、現在の設定値を返す。

**解説**

データオブジェクトは、それぞれのデータ系列を表す。データオブジェクト毎に、自分自身がどのようなグラフとして表示されるかを示すプロパティを持っている。この関数は、グラフとしての表示タイプを指定する。

**関数名**

**data.value(valueArray)**

**機能**

データオブジェクトが持つデータを設定する。

**引数**

配列変数 数値の配列または文字列の配列

**戻り値**

引数を省略した場合、現在の設定値を返す。

**解説**

実際にグラフとして表示されるデータを配列で指定する。配列の内容は数値または文字列を指定するが、文字列配列の場合はシンボルグラフでしか描画されない。

**関数名**

**data.num()**

**機能**

データオブジェクトが持つデータ数を返す。

**引数**

引数はなし。

**戻り値**

現在のデータ数（配列の要素数）を返す。

**関数名**

**data.range()**

**機能**

データオブジェクトが持つデータの範囲を返す。

**引数**

引数はなし。

**戻り値**

データ配列の中の 最小値と最大値が、"最小値、最大値"の形で返す。

**関数名**

**data.color(rgb,rgb2)**

**機能**

グラフの表示色を設定する。

**引数**

rgb 折れ線グラフでは線の色、棒グラフでは塗りつぶし色を#RRGGBB の形式で指定する。

rgb2折れ線グラフではシンボルの色、棒グラフでは棒の枠色を#RRGGBB の形で指定する。

**戻り値**

引数を省略した場合、現在の設定値を返す。

**解説**

グラフの描画色を設定する。折れ線グラフでシンボルを表示しない場合、rgb2 に指定した値は無視される。

**関数名**

**data.linesymbol(type)**

**機能**

折れ線グラフのシンボルを設定する。

**引数**

折れ線グラフの値の位置に描画するシンボルを以下の文字列で設定する。

circle 円

box 四角形

**戻り値**

引数を省略した場合、現在の設定値を返す。

**解説**

折れ線グラフの値の位置に描画するシンボルの形状を設定する。デフォルトではシンボルは描画されない。

**関数名**

**data.connect(bool)**

**機能**

折れ線グラフの値間を線で結ぶかどうかを設定する。

**引数**

値間を線で結ぶかどうかを Boolean 値で指定する。

<code>true</code>	線で結ぶ	値の種類 (評価方式) を設定する .
<code>false</code>	線で結ばない	
<b>戻り値</b>		
引数を省略した場合 , 現在の設定値を返す .		
<b>解説</b>		
折れ線グラフの値の間を線で結んで描画するかどうかを設定する . デフォルトでは結ぶ設定である .		
線で結ばない設定としたときは , 値にシンボルを設定しないと何も表示されない .		
<b>関数名</b>		
<code>data.weight(w)</code>		
<b>機能</b>		
棒グラフの棒の幅を設定する .		
<b>引数</b>		
目盛間隔に閉める割合を % の数値で指定する . デフォルトは 90 である .		
<b>戻り値</b>		
引数を省略した場合 , 現在の設定値を返す .		
<b>関数名</b>		
<code>data.symbols(symbolArray)</code>		
<b>機能</b>		
シンボルグラフに表示するシンボルを設定する .		
<b>引数</b>		
シンボルファイル (BMP 形式) のある URL (フルパス) の配列を指定する .		
<b>戻り値</b>		
引数を省略した場合 , 現在の設定値を返す .		
<b>解説</b>		
要素数は <code>symbolvalues</code> 関数で設定する閾値配列要素数と一致させる . ただし , 閾値の種別が "range" の場合にはひとつ多く設定する必要がある (いずれの閾値よりも大なるときのシンボル) .		
<b>関数名</b>		
<code>data.symbolvalues(thresholdArray)</code>		
<b>機能</b>		
シンボルグラフでシンボルを選択するための閾値を設定する .		
<b>引数</b>		
閾値の配列を指定する .		
<b>戻り値</b>		
引数を省略した場合 , 現在の設定値を返す .		
<b>解説</b>		
閾値を範囲 (range) とする場合は , 閾値は数値でなければならない .		
閾値は , 「以下」「より大きい」で判断される .		
<b>関数名</b>		
<code>data.symbolvaluekind(kind)</code>		
<b>機能</b>		
シンボルグラフでシンボルを選択するための閾		
<b>引数</b>		
閾値の種類 (評価方式) を設定する .		
<b>戻り値</b>		
引数を省略した場合 , 現在の設定値を返す .		
<b>解説</b>		
閾値の種類を以下の文字列で設定する .		
value 値と完全一致 (文字列として評価)		
range 値以下 (数値として評価)		
<b>戻り値</b>		
引数を省略した場合 , 現在の設定値を返す .		
<b>関数名</b>		
<code>data.xshift(shift)</code>		
<b>機能</b>		
本来の位置からシフトしてグラフを描画する際のシフト量を設定する .		
<b>引数</b>		
右をプラスとし , シフトさせる量を目盛間隔の % 値で指定する .		
<b>戻り値</b>		
引数を省略した場合 , 現在の設定値を返す .		
<b>関数名</b>		
<code>data.yaxis(axisno)</code>		
<b>機能</b>		
データを適用する Y 軸を選択する .		
<b>引数</b>		
左側の Y 軸の場合は 1 , 右側の Y 2 軸の場合は 2 を数値で指定する .		
<b>戻り値</b>		
引数を省略した場合 , 現在の設定値を返す .		
<b>3. おわりに</b>		
現時点の電子国土 Web システムの最新バージョンは , 1.0.0 であるが , 今後バージョン 1.1.0 の公開を予定している . バージョン 1.1.0 では , 位置情報や投影法が埋め込まれた TIFF 画像ファイル (GeoTIFF) を表示する API などが追加される予定である .		
また , 電子国土 Web システム (非プラグイン版) についても , プラグイン版と同等の機能を持たせるよう開発を進めている . プラグイン版と非プラグイン版の機能が同等になれば , 電子国土サイト構築者は , プラグインのインストール状態やブラウザによる違いを意識することなく , 電子国土サイトを構築することができるようになる .		
電子国土 Web システムを利用して情報を発信している電子国土サイト数は , 平成 20 年 2 月 7 日に 1,000 サイトを超えた . 今後もインターネットを利用した地理空間情報の発信は益々増加すると考えられる . 電子国土 Web システムについても機能追加・改良などを進め , 利用しやすい環境を整えて行きたい .		

### 参考文献

- 大野裕幸, 明野和彦, 久松文男, 石関隆幸(2004):電子国土Webシステム,国土地理院時報, 104, 25-33.  
国土地理院(2004):電子国土Webシステム APIリファレンス,国土地理院技術資料E・1- 296.