

# **Geodetic Data Modelling System: Honours Thesis**

## **An online interface**

**Michael Still**  
964076

**Geodetic Data Modelling System: Honours Thesis: An online interface**  
by Michael Still

**Abstract**

TODO

# Table of Contents

<b>1. Introduction .....</b>	<b>1</b>
Motivation .....	1
.....	1
<b>2. Implementation rationale.....</b>	<b>3</b>
Presentation of the user interface.....	3
Selection of an image format for graph display.....	3
TIFF.....	3
<b>3. Appendix One: GDMS Internet interface scripting elements.....</b>	<b>5</b>
Introduction .....	5
commands.....	5
dataset .....	5
datasets.....	5
eastplot .....	6
motd.....	6
northplot .....	6
upplot .....	6
<b>4. Appendix Two: Source code .....</b>	<b>7</b>
<b>5. Appendix Three: A sample configuration file .....</b>	<b>11</b>
<b>6. Appendix Four: References.....</b>	<b>13</b>
References.....	13



# Chapter 1. Introduction

## Motivation

At the time that the **Geodetic Data Modeling System** (GDMS) implementation project was initiated, several of the potential users expressed a desire for the application to be available over the Internet. There are a variety of reasons that this type of functionality is enticing -- the main ones for GDMS are that it allows casual users to ability to analyse data, whilst not having to maintain their own copies of the datasets, and it allows users who would normally use the X windows interface to the application to access data whilst "in the field", or otherwise physically separated from their normal research location.

Initially, it was though that a Internet interface to GDMS was outside the achievable scope of the project for 2002. However, as the year progressed, it became clear to me that implementing a Internet interface was indeed achievable, and would add genuinely useful functionality to the application. Hence this honours extension was undertaken.

...

TODO



## Chapter 2. Implementation rationale

### Presentation of the user interface

In accordance with Internet user interface design best practise, the GDMS Internet interface presents a fully configurable user interface. This is implemented by providing extension tags to those which are available in the standard HTML specification (Raggett, Le Hors, & Jacobs 1999).

The GDMS Internet interface follows the recommendations of the most recent HTML specification at the time of development for the insertion of scripting tags into HTML source files (Raggett, Le Hors, & Jacobs 1999). Further documentation on the GDMS Internet interface tags available can be found in Appendix One of this document.

### Selection of an image format for graph display

Much thought was expended on which image format to use for the plots displayed in the GDMS Internet interface. There are advantages and disadvantages to all the formats available at the time of writing this document. The following formats were considered for the GDMS Internet interface:

#### TIFF

TIFF (Tagged Image File Format) is a raster (bitmap) image format which was originally produced by Aldus and Microsoft. Aldus was later acquired by Adobe, who manage the TIFF specification to this day. At the time of writing, the current version of the TIFF specification is TIFF version 6.0 (Adobe 1992).

The main advantage of the TIFF image format is that it is extremely mature. It has a large selection of options to ease the development of imaging software, for example the developer can use whichever endian representation is convenient for them, as well as such fundamental parameters as photometric interpretation (whether a high value is lighter or darker than a low value for a given colour sample within a pixel) (Adobe 1992). There is also a stable, well supported Open Source application programmers interface available to manage creation and interpretation of images within the TIFF format (Warmerdam & Welles 2002). The interface to this library does not frequently change, which is also an advantage for future portability of the GDMS Internet interface.

However, some of these very advantages lead to the disadvantages that the TIFF format suffers from. Because the content of the file is so loosely specified, it is quite hard to write an application which can correctly decode all possible TIFF format images. This has resulted in there being only a few high quality TIFF viewing applications being available. The relative difficulty of implementing a TIFF decoder has also resulted in the web browser support for TIFF images being very poor (Bither 2002)





## Chapter 3. Appendix One: GDMS Internet interface scripting elements

### Introduction

This appendix documents the various tags which are implemented by the GDMS Internet interface in addition to the standard HTML tags normally available. All of these tags are within the scripting namespace provided by HTML 4.01 (Raggett, Le Hors, & Jacobs 1999).

#### commands

This tag will display a menu of the commands available in the current context -- for example, when the Internet interface first starts, it will list the open command. Commands which are not available, but are normally available will appear in a disabled style. This results in a list of commands which is consistent, and therefore reduces the potential for user confusion.

An example use is as follows:

```
&{commands};
```

The display of the command menu is altered by the following configuration file entries:

- *commandentry*: this HTML snippet is used for enabled commands
- *discommandentry*: this HTML snippet is used for disabled commands

For example, these configuration entries ship by default with the GDMS Internet interface:

```
# This line is simple HTML used for format the command entries,
# %s is the name of the command (including link HTML)
$commandentry = "<tr><td bgcolor=\"AAAAAA\">%s</td><tr>";

# This line is used for commands which aren't available...
$discommandentry = "<tr><td bgcolor=\"EEEEEE\">
    <font color=\"777777\">%s</font></td></tr>";
```

#### dataset

This tag will display the name of the current dataset, if one is open.

An example usage is:

```
&{dataset};
```

The dataset name is the portion of the filename before the extensions .dat1, .dat2, and .dat3 are applied. For example *mb\_AUCK\_GPS* is a dataset name.

#### datasets

This tag will display a list of the datasets available in the dataset directory.

An example usage is:

```
&{datasets};
```

The dataset directory is configured with the following configuration file entry:

```
# This line defines where the datasets are stored
$datasets = "/home/httpd/gdms-datasets/";
```

There are also several configuration options which alter the appearance of the list of datasets available. These are:

- *selectstart*: this is used for any output which should appear at the beginning of the list. This could include HTML tags for the creation of the required list markup (for example tables).
- *selectentry*: this configuration item is used for each entry in the list. The special text %s is replaced by the HTML for the entry itself.
- *selectend*: this is used for any HTML required to finalize the list.

```
# This is used for selection lists (for instance datasets)
$selectstart = "List start";
$selectentry = "<LI>%s";
$selectend = "List end";
```

### **eastplot**

This tag will insert the HTML and required URL for the East direction plot from the current dataset. The output of the tag will be empty if there is no currently selected dataset. The plots are 24 bit colour PNG images, as discussed in the implementation methodology section of this document.

An example usage is:

```
&{eastplot};
```

### **motd**

This tag will display a welcome message to the user -- the Message Of The Day (MOTD). This can be used to inform users of new datasets now being available, planned system outages, and other such informational messages.

An example usage is:

```
&{motd};
```

### **northplot**

This tag will insert the HTML and required URL for the North direction plot from the current dataset. The output of the tag will be empty if there is no currently selected dataset. The plots are 24 bit colour PNG images, as discussed in the implementation methodology section of this document.

An example usage is:

```
&{northplot};
```

### **upplot**

This tag will insert the HTML and required URL for the Up direction plot from the current dataset. The output of the tag will be empty if there is no currently selected dataset. The plots are 24 bit colour PNG images, as discussed in the implementation methodology section of this document.

An example usage is:

```
&{upplot};
```

## Chapter 4. Appendix Two: Source code

```
#!/usr/bin/perl

# GDMS web broker
# Copyright (C) Michael Still          2002
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

# This is the broker between the GDMS code and the CGI server. It's role is to
# parse templates, execute requests, and cache results for speedy processing
# next time that request is seen...
use strict;
use CGI;

my($result, $command, $TEMP);

# Variables set by the config file
my($templates, $datasets, $commandentry, $discommandentry, $select-
start, $selectentry, $selectend, $plotcache, $tmpdir, $rooturl, $plo-
turl, $gdms, $temp);

# Setup the CGI module
$result = new CGI();
print $result->header;

# Read in the config file
eval 'cat gdms.config' or die "GDMS web could not read it's config file: $@";
print STDERR "Working directory is: ".$pwd';
$temp = $gdms;
$temp =~ s/\./.*$//;
print STDERR "Changing to: $temp\n";

# Determine what page we are accessing
$command = $result->param('command');
if($command eq ""){
    $command = "main";
}

# Determine if the GDMS script for this page exists already (

print processTemplate("$templates/$command.html");
exit;

#####
# Find the template file, and then parse it
sub processTemplate(){
    my($file) = @_;
    my($pre, $post, $cmd, $output, $len, $line);

    # This local usage is used to make the TEMPLATE filehandle lo-
    cal to this
    # subroutine...
    local *TEMPLATE;
    print STDERR "Processing template file: $file\n";

    $output = "";
```

```

    open TEMPLATE, "< $file" or
die "GMDS web could not open the template file $file";
    while(<TEMPLATE>){
# Repeatedly process a line until there are not more template commands
$line = $_;
$len = -1;

while($len != length($line)){
    $len = length($_);

    # todo: there must be a better way of doing this...
    $_ = $line;
    if(/(.*)&{([^\%]*)};(.*)/){
$pre = $1;
$cmd = $2;
$post = $3;

if($cmd eq "commands"){
    # List the available commands
    $line = $pre.getCommands().$post;
}
elseif($cmd eq "dataset"){
    # The name of the current dataset
    $line = $pre.$result->param('dataset').$post;
}
elseif($cmd eq "datasets"){
    # List the datasets in the dataset directory
    my($temptotal, $tempfile);
    $temptotal = "";

    print STDERR "Getting datasets from $datasets\n";
    open TEMP, "find $datasets -type f -name \"*.dat1\" |";
    while(<TEMP>){
my($linecount);
$tempfile = $_;
$tempfile =~ s/$datasets\/.*//;
$tempfile =~ s/.dat1\n$//;

$linecount = `cat $datasets/$tempfile.dat1 | wc -l`;
$temptotal = $temptotal.
    substHTML($selectentry,
        "<a href=\"$rooturl?command=main&dataset=$tempfile\">$tempfile<\a> ".
        "($linecount lines)");
    }
    close TEMP;

    $line = $pre.$temptotal.$post;
}
elseif($cmd eq "motd"){
    # Output a message of the day
    $line = $pre.processTemplate("$templates/motd.html").$post;
}
elseif($cmd eq "northplot"){
    # A plot in the X direction
    print STDERR "Plotting in x direction\n";
    $line = $pre.generateAndLink("x").$post;
}
elseif($cmd eq "eastplot"){
    # A plot in the Y direction
    $line = $pre.generateAndLink("y").$post;
}
elseif($cmd eq "upplot"){
    # A plot in the Z direction
    $line = $pre.generateAndLink("z").$post;
}
    }
}

# And now we can print out the resultant line
$output = $output.$line;
}

```

```

        return $output;
    }

# Determine what commands should be available at this time
sub getCommands(){
    my($output, $temp);

    $output = "";
    if($result->param('dataset') eq ""){
        # Dataset open / close
        $output = $output.substHTML($commandentry, "<a href=\"$rooturl?command=open\">Open</a>");
        $output = $output.substHTML($discommandentry, "Close");

        # Plotting
        $output = $output.substHTML($discommandentry, "Plot");
    }
    else{
        # Dataset open / close
        $output = $output.substHTML($discommandentry, "Open");
        $output = $output.substHTML($commandentry, "<a href=\"$rooturl?command=main\">Close</a>");

        # Plotting
        if($result->param('command') eq "plot"){
            $output = $output.substHTML($discommandentry, "Plot");
        }
        else{
            $output = $output.substHTML($commandentry, "<a href=\"$rooturl?command=plot&dataset=$result->param('dataset')\">Plot</a>");
        }
    }

    return $output;
}

# Substitute into the HTML stub from the config file
sub substHTML(){
    my($html, $insert) = @_;
    my($temp);

    $temp = $html;
    $temp =~ s/%s/$insert/;
    return $temp;
}

# This subroutine deals with generating plots as required and then outputs
# the HTML needed to link to that image
sub generateAndLink(){
    my($dir) = @_;
    my($file, $unique);
    local *COMMANDS;

    print STDERR "Started generateAndLink()\n";

    # Generate the filename
    $unique = "$$-".time()."-".rand();
    $file = "$plotcache/". $result->param('dataset')."-$dir.png";

    print STDERR "Filename is: $file\n";

    if(! -f $file){
        # We need to generate the image
        print STDERR "Plot cache miss for ".$result->param('dataset')." ($dir)\n";
        open COMMANDS, "> $tmpdir/gdms-$unique.cmd" or
            die "Could not open temporary file $tmpdir/gdms-$unique.cmd\n";
        print COMMANDS "open $datasets/". $result->param('dataset')." \n";
        print COMMANDS "plot $dir $file\n";
        close COMMANDS;

        # Execute the gdms main program with this command script
        '$gdms -b $tmpdir/gdms-$unique.cmd' or

```

```
        die "GDMS execution error for: $gdms -b $tmpdir/gdms-$unique.cmd";
    print STDERR "Return code as $?\n";
}

    # Now link to that image
    return "<img src=\"$ploturl/\".${result->param('dataset')}.\"-$dir.png\">";
}
```

*Code: eps*

## Chapter 5. Appendix Three: A sample configuration file

```
#####
# Directories
#####

# This line defines where the templates are stored
$templates = "/home/httpd/gdms-templates/";

# This line defines where the datasets are stored
$ddatasets = "/home/httpd/gdms-datasets/";

#####
# HTML configuration
#####

# This line is simple HTML used for format the command entries, %s is the name
# of the command (including link HTML)
$commandentry = "<tr><td bgcolor=\"AAAAAA\">%s</td><tr>";

# This line is used for commands which aren't available...
$discommandentry = "<tr><td bgcolor=\"EEEEEE\"><font color=\"777777\">%s</font></td>";

# This is used for selection lists (for instance datasets)
$selectstart = "List start";
$selectentry = "<LI>%s";
$selectend = "List end";

#####
# Caching configuration
#####

# This is the location of the plot cache -- it can get quite big
$plotcache = "/home/httpd/html/gdms-plots/";
$ploturl = "/gdms-plots/";

# Location for temporary files
$tmpdir = "/home/httpd/gdms-temp/";

#####
# Other
#####

# This is the root URL
$rooturl = "/cgi-bin/gdms.pl";

# This is the full path to the GDMS application
$gdms = "/home/httpd/gdms";
```

*Code: eps*





## Chapter 6. Appendix Four: References

### References

Adobe, 3 June 1992, *TIFF Revision 6.0*, [Online]  
<http://partners.adobe.com/asn/developer/pdfs/tn/TIFF6.pdf>

Raggett, D., Le Hors, A., Jacobs, I., 24 December 1999 [last update], *HTML 4.01 Specification*, [Online] Available: <http://www.w3.org/TR/html4/>

Bither, B., [Last accessed] November 2002, *Benefits of the PNG Image Format*, [Online] Available: <http://www.atalasoft.com/png.asp>

Warmerdam, F. and Welles, M., 24 January 2002 [last update], *The libtiff home page*, [Online] <http://www.libtiff.org/>

