Documentation et manuel d'utilisation du serveur WWW de l'Atmosphère de Mars

Christophe Hourdin

September 8, 1997

Abstract

Contents

Part I

La documentation du GCM Martien

Part II

La serveur Martien - Base de données

Chapter 1

Généralitées sur le serveur

1.1 Attention chaine de caractère des axes!

Dans *present_data.c*, La chaine de caractère de chacun des axes est initialisée en blanc par:

```
char *axString[4]={"           ", "            ",
             "                  ", "              "};
```

Surtout avec l'altitude donnée en "sigma_level", il est nécessaire d'élargir cette chaîne de caractères.
Si cette chaine est trop courte, le serveur renvoie un message d'erreur indiquant des arguments abhérents pour les GO scripts ferret en ce qui concerne les valeurs de temps, level... des axes.

1.2 Attention dans *initialize.h*!

La première variable dans *initialize.h* (variable 0) doit être doublée. Pour la faire apparaitre sur le menu de la page html, il faut la référencer par **!1** (variable 1) et non par **!0** (variable 0) dans *select-by-dataset.html.template.frames* au risque de ne pas voir apparaitre de page lorsqu'on clique sur le bouton "Change data set" de la main page (et pas de message d'erreur ce qui permet de se poser à peu près 300.000 questions).

1.3 Test Modifications

Lorsqu'on effectue des modifications dans les scripts ferret (.jnl) appellés par le serveur, il faut, pour tester ces changements quitter et relancer netscape. Un reload ne suffit pas. Il faut également, comme à chaque test, aller détruire dans *WWW/ferret/p* l'image correspondante, afin de forcer le serveur à la recréer (avec le nouveau script par exemple).

On peut aussi, quand on ne comprend plus rien! (modifications sans effets) , detruire tous les .o et recompiler.

1.4 Les GO scripts ferret (.jnl)

Les seuls scripts appellés directement par le serveur sont:
"setup_region.jnl"

"std_gif_*.jnl"
"refmap.jnl"
"std_list.jnl"
Ils sont appellés par le programme *live_access.c* (répertoire *WWW/live_access/src*).

Afin de savoir quelles sont les commandes passées par le serveur à ferret pour créer les images et ainsi pouvoir les modifier, on a créé une sortie de toutes ces commandes (fichier *MOMO* dans *WWW/ferret/p*).

1.5 Les cannevas (template et frames)

Le serveur va lire les fichiers .template et .frames stockés dans le répertoire *WWW/ferret/pcgi_htmls_v2.0* et remplace le numéro rencontré après un "**!**" dans ces fichiers par le nom correspondant dans le fichier *initialize.h* (*WWW/live_access/src*).

1.6 Format image gd

Le format *gd* est un format image qui permet d'afficher rapidement une image sous HTML (librairie et démo dans */dg/chourdin/gd1.2*).

Les fonctions correspondantes utilisées par le serveur sont dans le fichier *gd.c* (répertoire *WWW/live_access/src*).

Chapter 2

Pour modifier le serveur

2.1 Changement de l'image en première page

Cette image est stockée en format *gd* dans le rèpèrtoire *WWW/ferret/pzf* sous le nom de *world-small.gd*

Attention: cette image doit etre en coordonnees 0:360 -90:90

Afin de créer cette image, on peut générer par n'importe quel utilitaire une image au format GIF et la convertir ensuite avec *giftogd* (répertoire */dg/chourdin/gd1.2*).

Si on veut créer une image à partir de *ferret*, ferret ne pouvant générer que du metafile (GKSM format), il est nécessaire ensuite de convertir ce format au format gd. Le shell script *Meta2gif* s'en charge (répertoire *WWW/ferret/world-small*). Il passe du format GKSM au format ps par gksm2ps ou par la fonction Fprint de ferret, puis du format ps au format ppm par ghostscript, et enfin du format ppm au format gif par ppmtogif. Il faut ensuite bien sur lancer la commande "giftogd out.gif out.gd".

2.2 3 bugs pour le zoom en (-180:180)

remet la valeur a 180 au lieu de -180 au 2eme appel du zoom.

Dans *pzf.c*, le calcul de la largeur du cadre du zoom donne 0 pour x1 = x2 (cas de -180 a 180 ou -180 est ramene a 180 par addition de 360). Il faut donc rajouter le cas d'egalité dans $if(x1 >= x2)thenwidth = (sx - x1) + x2$ sinon $width = x2 - x1 = 0$, et on obtient alors un trait vertiavl sur la map à 180 degrés.

Quand on demande un zoom sur une région en -180: 180, de retour sur la page principal, le serveur indique 180:180 au lieu de 180e:180w. Le traitement de la chaine de caractère pour la longitude est fait dans *test_axis_entries.c*. Il faut retirer le cas de la fin du traitement qui fixe à 180 tout ce qui est en 180 (180e ou 180w). Il faut donc remplacer:

```
if (rval == 180.0 || rval == 360.0 || rval == 0.0)
```

par

```
if (rval == 360.0 || rval == 0.0)
```

Il faut également remplacer:

```
if (kar == 'w' && rval >= 180.0)
```

par

```
if (kar == 'w' && rval > 180.0)
```

et

```
else if (kar == 'e' && rval >= 180.0)
```

par

```
else if (kar == 'e' && rval > 180.0)
```

2.3 Changement des régions prédéfinies pour le Zoom

Indiquer les coordonnées du zoom dans le fichier *process_xy_zoom_menu.c* ainsi qu'un nom correspondant à cette région.
Reporter ce nom dans les fichiers *zoom.html.template* et *second_zoom.html.template* du répertoire *WWW/ferret/pcgi_htmls_v2.0*.

2.4 Changement de la couleur du cadre pour le zoom

Dans le fichier *pzf.c* (répertoire *WWW/live_access/src*), changer les valeurs pour rouge, vert et bleu de la fonction *gdImageColorAllocate*.

trait_couleur = gdImageColorAllocate(imo, ROUGE, VERT, BLEU);

Chacune des couleur allant de 0 à 256.

exemple pour un trait bleu:

trait_couleur = gdImageColorAllocate(imo, 0, 0, 255);

2.5 Changement des données

Pour donner l'acces à des données, il faut qu'elles soient au format Netcdf et placées dans un des répertoires désignés par la variable FER_DATA (variable définie dans le fichier *site_specific_references.h* du répertoire *WWW/live_access/src*). A priori on les place dans *WWW/mars_data/data*.

On spécifie ensuite les nouvelles données dans le fichier *initialize.h* de *WWW/live_access/src*.

Puis, pour voir ces nouvelles données apparaitre dans le menu "Change Data Set", on indique dans le fichier *select-by-dataset.html.template.frames* (*WWW/ferret/pcgi_htmls_v2.0*), derriere un point d'exclamation "!", l'index correspondant au nouveau jeu de données (index fixé dans *initialize.h*).

2.6 Changement des valeurs par défaut

Pour sélectionner le fichier de données, et la variable choisie par défaut au moment de la connexion, , changer la valeur de "MOMO_default" dans *get_current_status.c* par l'index correspondant à la variable souhaitée (index fixé dans *initialize.h*).

La vue par défault choisie est celle fixée dans la définition de la variable dans *initialize.h*.

2.7 Gestion de l'axe temps

2.7.1 Cas une variable un fichier Netcdf

Le serveur est initialement prévu pour 2 cas:
La date sous forme jour et mois (ex: 16-dec) avec le cas ONE_WIDGET,
La date sous forme jour,mois et année (ex: 15-dec-1988) avec le cas MONTH_YEAR.
Les dates définies à la fin du fichier *initialize.h* correspondent chacune à une valeur, qui correspond dans le fichier Netcdf à une valeur de temps en "hour" ou "day" ou... Ferret prend cette valeur comme un offset par rapport à la date de réference du fichier Netcdf:
exemple d'entête d'un fichier Netcdf:
time:units = "days since 1984-01-01").

2.7.2 Cas une variable plusieurs fichiers Netcdf (stats)

Cohérence en temps des différents fichiers

Lorsqu'on a un fichier Netcdf par jour par exemple, on peut créer un fichier ferret de description (*.des*) qui appelle les différents fichiers Netcdf. L'appel des différent fichiers est basé sur la chronologie de l'axe des temps. Les fichiers peuvent ne pas être régulièrement espacés, (Time INDEPENDANT), mais doivent avoir la même unité de temps et des temps qui se succèdent.
Par exemple:
Le fichier "simulation.des" appelle
fichier "simulation.001" time(in days) 7000 7001 7002
fichier "simulation.002" time(in days) 7053 7055 7089
fichier "simulation.003" time(in days) 8427 8428 8429
On indique dans le fichier .des une date de référence et la valeur du jour qui correspond.
exemple:
D_T0TIME = '14-JAN-1984 00:00:00',
S_START = 7000.0,

Pas de cohérence en temps des différents fichiers

Voir cahier au Mercredi 19 Mars 1997.

Dans le cas des fichiers "stats", il n'y a pas de cohérence en temps, puisque les fichiers contiennent des moyennes diurne sur la période de simulation. On dispose donc de 12 valeurs de temps (toutes les 2 heures) par mois de simulation. On ne peut donc pas créer un fichier .des.
On a modifié le programme, pour avoir une date sous la forme:
"A2m1-h20", ou A2m1 est récupéré et ajouté comme extension au nom du fichier de données, et h20 est identifiée par le 'h' comme l'heure 20, la valeur de time du fichier Netcdf.

Le fichier *present_data.c* récupère les informations pour ferret, les met en forme, puis appelle le programme *live_access* qui se charge de lancer les GO.

3 cas sont possibles pour le temps dans *present_data.c*:
1/ les valeurs de temps ont étées affectées à l'exterieur dans TLO et THI. On détecte donc la première lettre de ces valeurs et on traite en fonction.
2/ une valeur de temps est affectée dans la variable "value" par *validate_custom_location*. On détecte donc la première lettre de value.

3/ une valeur de temps est affectée dans la variable "value" par *get_axis_location*. On détecte également la première lettre de value.
On modifie également la fonction *get_axis_location*. On rajoute le cas LS_STATS (en plus de ONE_WIDGET et MONTH_YEAR).

On récupère dans chacun de ces trois cas, le début (A2m12 par exemple) pour affecter la variable "DSET_ext". Cette variable est définie dans *tmap_cgi.h*. Cette variable est l'extension qu'on concatène au nom du fichier netcdf avant l'appel à *live_acces*.
Il faut également l'intégrer dans le nom du fichier gif créé, sinon un même nom de fichier correspondra à plusieurs cas (ex: *viking_year_xy_ps-180.0:180.0_-90.0:90.0_12.gif*) pour tous les mois. Il faut donc avoir un nom différent par saison (*viking_year_a2m3_xy_ps...* et *viking_year_a3m7_xy_ps...*).

La récupération du temps se fait par la fonction *parse_date* du fichier *test_axis_entries.c*. On a rajouté dans cette fonction *parse_date* les cas HOUR_SOLO et HOUR_STATS.
HOUR_SOLO pour une date en heure de la forme "h01"
HOUR_STATS pour une date de la forme "A2m1-h20".

Il faut également modifier la fonction *express_error_sadness* du fichier *test_axis_entries.c* pour avoir un message d'erreur sur le temps approprié au cas LS_STATS.

2.8 Messages d'erreur

Voir la fonction *express_error_sadness* dans *test_axis_entries.c*. A l'appel de cette fonction, une page HTML est créée avec le message d'erreur.
On peut en créer autant qu'on veut sur le même modèle. (exemple *momo_error_sadness* dans le même fichier.

2.9 Palette fonction de la variable

la subroutine *live_access.c* envoie à ferret la commande "GO std_gif_..." Ce script a pour argument 7 le nom de la palette. *live_access.c* donne comme valeur pour l'argument 7 soit "light_centered" (palettte light_centered.spk) soit " " (la palette par defaut "rainbow.spk?") A la place, on lui passe comme argument 7 "var" le nom de la variable. On place dans *WWW/mars_data/palette* la palette voulue nommée "var.spk". On remplace pour cela:

```
center ? "\"light_centered\"" : "\" \"");
```

par

```
var);
```

2.10 Reglage Timeout pour ferret

Si le temps d'execution des scripts GO de ferret est trop long, passé un certain temps, le serveur stop ferret et redonne la main après un message d'erreur. Ce temps est fixé dans *live_access.c*: **int timeout**. Cet entier est passé en argument à la subroutine *ferret_synch*.

2.11 Fond de carte: Map terrestre - relief martien

Pour les cartes en Longitude-Latitude, au lieu d'utiliser en fond de carte les continents terrestres définis dans etopo30 etopo60... (numero = différentes résolutions), on utilise le relief martien de surface.cdf (variable zcon).
On effectue les remplacements ad'hoc dans les scripts *std_gif_xyref.jnl* et *std_gif_xyref.jnl*.

2.12 Carte de référence: Map terrestre - relief martien

Pour toutes les cartes (scripts *std_gif_**), on souhaite avoir comme carte de référence, si il y a lieu, le relief martien. Cette carte est créée par *refmap.jnl*. On effectue dans *refmap.jnl* les changements necessaires pour remplacer les continents terrestres par le relief martien. Afin que refmap fonctionne comme cela avait été prévu initialement, on a besoin de rendre périodique en longitude le fichier surface.cdf qui contient le relief martien par l'instruction:

```
set axis/modulo longitude
```

Un problème se pose dans la mesure ou en fonction de l'ordre d'ouverture des différents fichiers par ferret, le nom de la longitude est longitude ou longitude1 ou longitude2... Pour régler ce problème, on ouvre en premier dans tous les scripts *std_gif_** le fichier surface.cdf. On sait alors qu'a n'importe quel niveau des scripts, la longitude pour le fichier surface se nomme "longitude".