



M-UDAS egg manual

for M-UDAS egg 1.00

20 August 2020

IUGONET project team

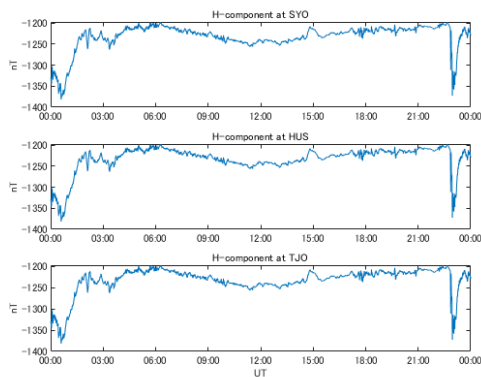
M-UDAS egg is **a template program for M-UDAS**

(MATLAB-based UDAS) to read/analyze scientific data on MATLAB.

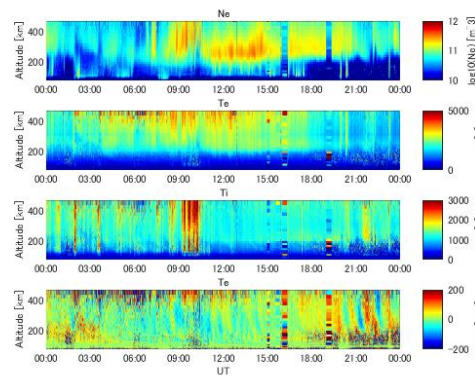
You can easily load/visualize/analyze your data

by modifying some parts marked in the code according to this manual.

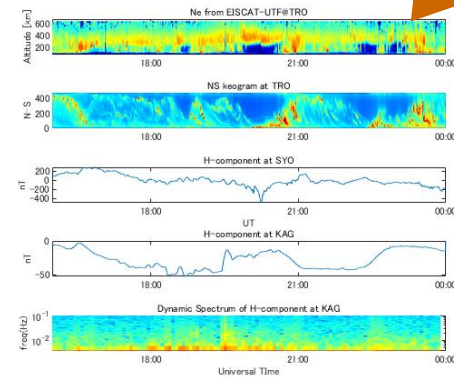
It supports
Windows,
MacOS, Linux



Line plot



Color contour



Combination with other data

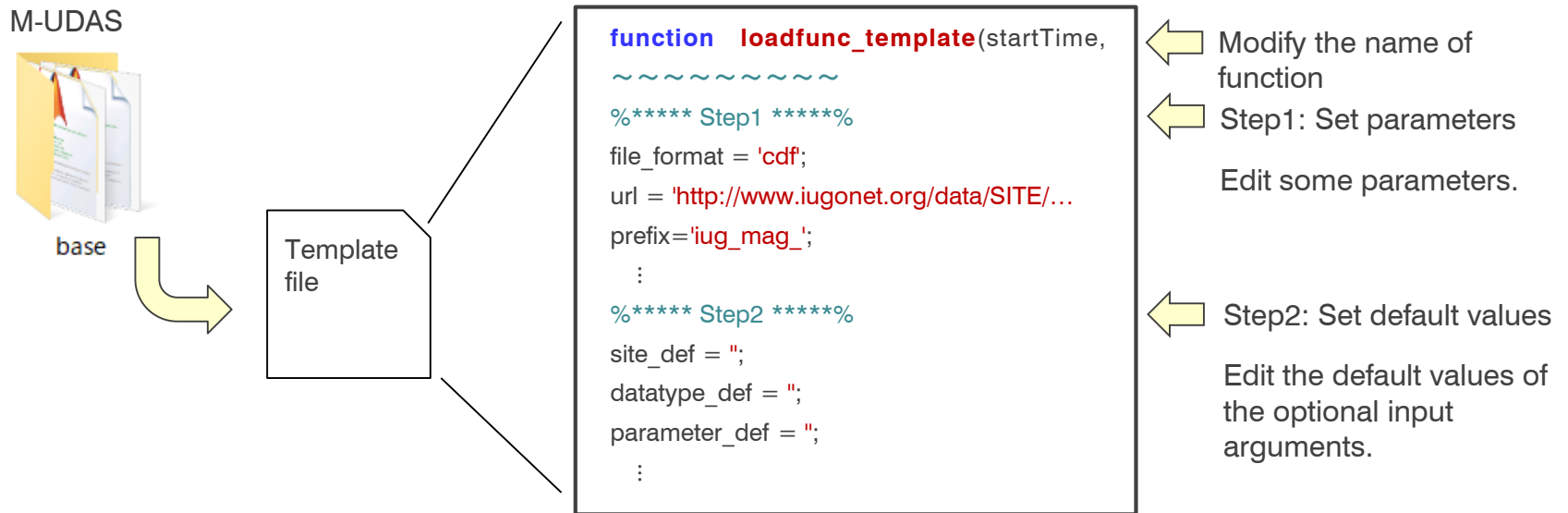
MATLAB is a high-performance language and interactive environment for data analysis and visualization. M-UDAS is a set of MATLAB routines for loading and analyzing various upper atmosphere data released by the IUGONET universities and institutions. The details of these software are described at the websites as follows; M-UDAS: <https://github.com/iugonet/UdasMatlab>

The readable file formats are as follows:

1. CDF (Common Data Format)
2. NetCDF (Network Common Data Form)

* It will support the FITS (Flexible Image Transport System) format and some types of the ASCII formats in future.

Create your own **load functions** by modifying the template file.



Execute the created load function.

```
>> loadfunc_xxx(startTime, endTime, 'option', 'value');
```



Data will be loaded as variables into the MATLAB workspace.

Prepare

1. Install M-UDAS

p.5

Customize

2. Customize the template

p.6

Run

3. Plot your data

p.14

Proceed to advanced analysis

p.16

Appendix 1: Program description

p.19

1. Download and install M-UDAS.

URL: <https://github.com/iugonet/UdasMatlab>

Download the zip file from **[Clone or Download]** button and unzip it in an arbitrary directory.

2. Download and install the latest version of MATLAB-CDF software (V3.8.0).

URL: https://spdf.gsfc.nasa.gov/pub/software/cdf/dist/cdf38_0/matlab/

- Windows: matlab_cdf380_win64_VS2015.exe
- Linux: matlab_cdf380_lin64.tar.gz * Please see HOWTO-install.txt for the installation.
- Mac OS: matlab_cdf380_mac64.tar.gz

3. Set path to M-UDAS and CDF software.

GUI:

1. Start MATLAB and click **[Set Path]** in the **[ENVIRONMENT]** section of the **[HOME]** tab.
2. Click **[Add with Subfolders]** in the **[Set Path]** window and select the M-UDAS directory.
3. Similarly, Click **[Add with Subfolders]** and select the CDF software directory and subdirectories.
4. Click **[Save]** to save path and click **[Close]** to close the window.

CUI:

1. Start MATLAB and run the commands as follows:

```
>> addpath(genpath('Path to the top directory of M-UDAS'))  
>> addpath(genpath('Path to the top directory of CDF software'))
```

* If you want to add these directories permanently to the MATLAB path, create “startup.m” in the MATLAB startup directory or any directory in the MATLAB path and add the commands above to the startup.m.

1. Copy and rename the template file.

There is a template file
in the directory:
iugonet/base/



base



docs



examples



load

[Original file] loadfunc_template.m

As an example, it is assumed that the template file was renamed to "loadproc_mag.m".

2. Edit the new file.

* : required

```
function loadfunc_mag(startTime, endTime, varargin)
%
% loadfunc_template(startTime, endTime, varargin)
%
```

* **function loadxxxxx**: Routine's name

Set the same name as the file name. This routine's name corresponds to the command name.

2. Edit the new file. (Step.1: Settings of general parameters)

* : required

```

%*****%
%***** Step1: Set parameters *****%
%*****%
file_format = 'cdf'; % 'cdf' or 'netcdf'
url = 'http://www.iugonet.org/data/SITE/DATATYPE/YYYY/mag_SITE_DATATYPE_YYYYMMDD_vVERSION.cdf';
prefix = 'iug_';
site_list = {}; % ex. {'sta1', 'sta2', 'sta3'}
datatype_list = {}; % ex. {'1sec', '1min', '1hr'}
parameter_list = {}; % ex. {'par1', 'par2', 'par3'}
version_list = {}; % ex. {'01', '02', '03'}
....
    
```

* file_format

Choose 'cdf' or 'netcdf'.

* url: URL of data files that you want to read

Set the URL of data files that you want to read. The following strings are automatically converted to specific values.

SITE: Site name (corresponds to the elements defined in site_list)

DATATYPE: Data type (corresponds to the elements defined in datatype_list)

PARAMETER: Parameter (corresponds to the elements defined in parameter_list)

VERSION: Version (corresponds to the elements defined in version_list)

Strings used for date and time

Year (4-digit)	Year (2-digit)	Month (01-12)	Day (01-31)	Hour (2-digit)
YYYY	yy	MM	DD	hh

2. Edit the new file. (Step.1: Settings of general parameters)

```

%*****%
%***** Step1: Set parameters *****%
%*****%
file_format = 'cdf'; % 'cdf' or 'netcdf'
url = 'http://www.iugonet.org/data/SITE/DATATYPE/YYYY/mag_SITE_DATATYPE_YYYYMMDD_vVERSION.cdf';
prefix = 'iug_';
site_list = {}; % ex. {'sta1', 'sta2', 'sta3'}
datatype_list = {}; % ex. {'1sec', '1min', '1hr'}
parameter_list = {}; % ex. {'par1', 'par2', 'par3'}
version_list = {}; % ex. {'01', '02', '03'}
....

```

【Example】

If you set the URL to

<http://iugonet0.nipr.ac.jp/data/fmag/SITE/DATATYPE/2018/>

[nipr_DATATYPE_fmagn_SITE_YYYYMMDD_vVERSION.cdf](#)

the URL is automatically converted as follows:

http://iugonet0.nipr.ac.jp/data/fmag/syo/1sec/2018/nipr_1sec_fmagn_syo_20181013_v02.cdf

http://iugonet0.nipr.ac.jp/data/fmag/syo/1sec/2018/nipr_1sec_fmagn_syo_20181014_v02.cdf

.....

where “syo”, “1sec”, and “02” are one of the elements of “site_list”, “datatype_list”, and “version_list”.

The date and time are fixed by the input arguments of the load routine, “startTime” and “endTime”.

2. Edit the new file. (Step.1: Settings of general parameters)

```

%*****%
%***** Step1: Set parameters *****%
%*****%
file_format = 'cdf'; % 'cdf' or 'netcdf'
url = 'http://www.iugonet.org/data/SITE/DATATYPE/YYYY/mag_SITE_DATATYPE_YYYYMMDD_vVERSION.cdf';
prefix = 'iug_';
site_list = {}; % ex. {'sta1', 'sta2', 'sta3'}
datatype_list = {}; % ex. {'1sec', '1min', '1hr'}
parameter_list = {}; % ex. {'par1', 'par2', 'par3'}
version_list = {}; % ex. {'01', '02', '03'}
....

```

prefix: prefix for loaded variables

Specify a prefix that is added to loaded variables. You can choose any prefix which is easy for you to understand. In the example above, “iug_” is prefixed to the variables (ex., iug_???_time, iug_???_info)

site_list: site list

Specify a list of observation sites. If there are more than one site, it is defined by a cell array, e.g., {'syo', 'hus', 'tjo'}. If there is no site, set it to {}. This corresponds to site names, which are available for the 'site' option.

datatype_list: data type list

Specify a list of data types, for example, sampling interval, wave length, observation mode etc. If there are more than one data type, it is defined by a cell array, e.g., {'1sec', '1min', '1hr'}. If there is no data type, set it to {}. This corresponds to data types, which are available for the 'datatype' option.

2. Edit the new file. (Step.1: Settings of general parameters)

```
%*****%
%***** Step1: Set parameters *****%
%*****%
file_format = 'cdf'; % 'cdf' or 'netcdf'
....
parameter_list = {}; % ex. {'par1', 'par2', 'par3'}
version_list = {}; % ex. {'01', '02', '03'}
acknowledgement = sprintf(['You can write the data use policy here.¥n',...
    'This description is displayed when you use this load procedure.']);
rootpath = default_rootpath;
```

parameter_list: parameter list

Specify a list of parameters. If there are more than one parameter, it is defined by a cell array, e.g., {'par1', 'par2', 'par3'}. If there is no parameter, set it to {}. This corresponds to parameter names, which are available for the 'parameter' option.

version_list: version list

Specify a list of versions. If there are more than one version, it is defined by a cell array, e.g., {'01', '02', '03'}. If there is no version, set it to {}. This corresponds to version numbers, which are available for the 'version' option.

acknowledgement: Data usage policy

Describe the data usage policy here. The description will be displayed on the window when you run this routine.

2. Edit the new file. (Step.1: Settings of general parameters)

```

%*****%
%***** Step2: Set default values *****%
%*****%
site_def = "";
datatype_def = "";
parameter_def = "";
version_def = version_list;
downloadonly_def = 0;
no_download_def = 0;
....

```

site_def: default value of site

Specify the default value of site. When the 'site' option is not used for the input argument, this value is used.

If site_list is an empty array, set to site_def="".

datatype_def: default value of datatype

Specify the default value of datatype. When the 'datatype' option is not used for the input argument, this value is used.

If datatype_list is an empty array, set to datatype_def="".

parameter_def: default value of parameter

Specify the default value of parameter. When the 'parameter' option is not used for the input argument, this value is used. If parameter_list is an empty array, set to parameter_def="".

2. Edit the new file. (Step.1: Settings of default values of options)

```
%*****%  
%***** Step2: Set default values *****%  
%*****%  
site_def = "  
datatype_def = "  
parameter_def = "  
version_def = version_list;  
downloadonly_def = 0;  
no_download_def = 0;  
....
```

version_def: default value of version

Specify the default value of version. When the 'version' option is not used for the input argument, this value is used. When version_def=version_list, all values in version_list are used for reading files.

downloadonly_def: default value of downloadonly

Specify the default value of the downloadonly option. When the 'downloadonly' option is not used for the input argument, this value is used. 0: Load data after downloading files. 1: Download only.

no_download_def: default value of no_download

Specify the default value of the no_download option. When the 'no_download' option is not used for the input argument, this value is used. 0: Download data files from the remote servers. 1: Does not download data files from the remote servers and reads the local data files.

2. Edit the new file. (Step.1: Settings of default values of options)

```

%*****%
%***** Step2: Set default values *****%
%*****%
site_def = "";

....
username_def = "";
password_def = "";
time_format='yyyy-MM-dd HH:mm:ss Z'; % Time format string for NetCDF
    
```

username_def : default value of username

Specify the default value of username. If this option is not used, set to username_def="".

password_def: default value of password

Specify the default value of password. If this option is not used, set to password_def="".

time_format: time format string used in the NetCDF files

Specify the format string of time data included in the NetCDF files. Please refer to the Format property of the “datetime” function in MATLAB. Examples: 'yyyy-MM-dd HH:mm:ss', 'yyyy-MM-dd"T"HH:mm:ss'.

3. Plot your data

1. Start MATLAB and run the following command.

```
>> loadfunc_mag('2015-01-01', '2015-01-03', 'site', 'syo', 'datatype', '1sec')
```

```
Data Downloading ... http://iugonet0.nipr.ac.jp/data/fmag/...
Data Saving ... C:\data\iugonet0.nipr.ac.jp\data\fmag\...
....
```

Data policy written in acknowledgement will be displayed on the console.

Run the created routine.
 If you defined site_list, datatype_list, parameter_list, you can set the options, 'site', 'datatype', and 'parameter' here.

```
>> whos
```

Name	Size	Bytes	Class
iug_mag_syo_1sec_epoch_1sec	172800x1	1382400	double
iug_mag_syo_1sec_f_1sec	172800x1	1382400	double
iug_mag_syo_1sec_hdz_1sec	172800x3	4147200	double
iug_mag_syo_1sec_info	1x1	67554	struct
iug_mag_syo_1sec_label_hdz_1sec	3x1	6	char
iug_mag_syo_1sec_label_time_1sec	7x11	154	char
iug_mag_syo_1sec_time_1sec	172800x7	4838400	int32
iug_mag_syo_1sec_time_cal_1sec	172800x1	1382400	double
iug_mag_syo_1sec_unit_hdz_1sec	3x2	12	char
iug_mag_syo_1sec_unit_time_1sec	7x3	42	char

Display the loaded variables.
 The variable names come from “prefix” defined at STEP1 and the input arguments, “site”, “datatype”, and “parameter”.
 In this example, because of prefix = 'iug_mag', site = 'syo', and datatype = '1sec', the loaded variable names become 'iug_mag_syo_1sec_???'.

3. Plot your data

```
>> disp_info(iug_mag_syo_1sec_info)
```

```
1. epoch_1sec
   FIELDNAM: Epoch
   CATDESC: Time, beginning of interval
   VALIDMIN: 01-Jan-1960 00:00:00.000
   VALIDMAX: 31-Dec-2100 23:59:59.999
```

```
....
```

```
2. time_1sec
   FIELDNAM: Time
   CATDESC: Observation date and time
   VALIDMIN: -9999
   VALIDMAX: 9999
```

```
....
```

```
>> plot(iug_mag_syo_1sec_epoch_1sec,...
        iug_mag_syo_1sec_hdz_1sec(:,1))
```

```
>> datetick('x', 'mm/dd HH:MM')
```

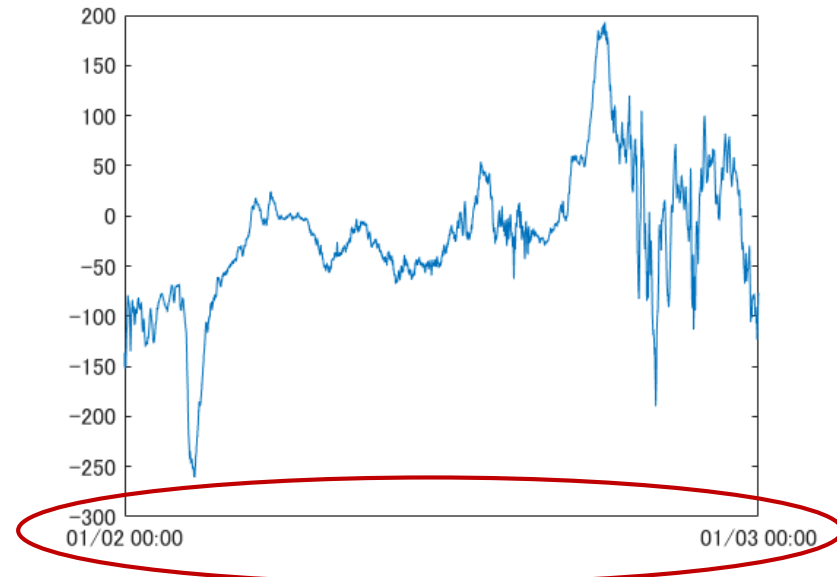
```
>> set(gca, 'xlim', [datenum(2015,1,2), ...
                    datenum(2015,1,3)])
```

Display the metadata of the loaded variables.
The command, `disp_info(???_info)`, displays the metadata of each variable.

The information shown below "1. epoch_1sec" is the metadata of the variable: `iug_mag_syo_1sec_epoch_1sec`.

Plot data.
The command, `plot(time_data, time_series_data)`, creates a line plot.

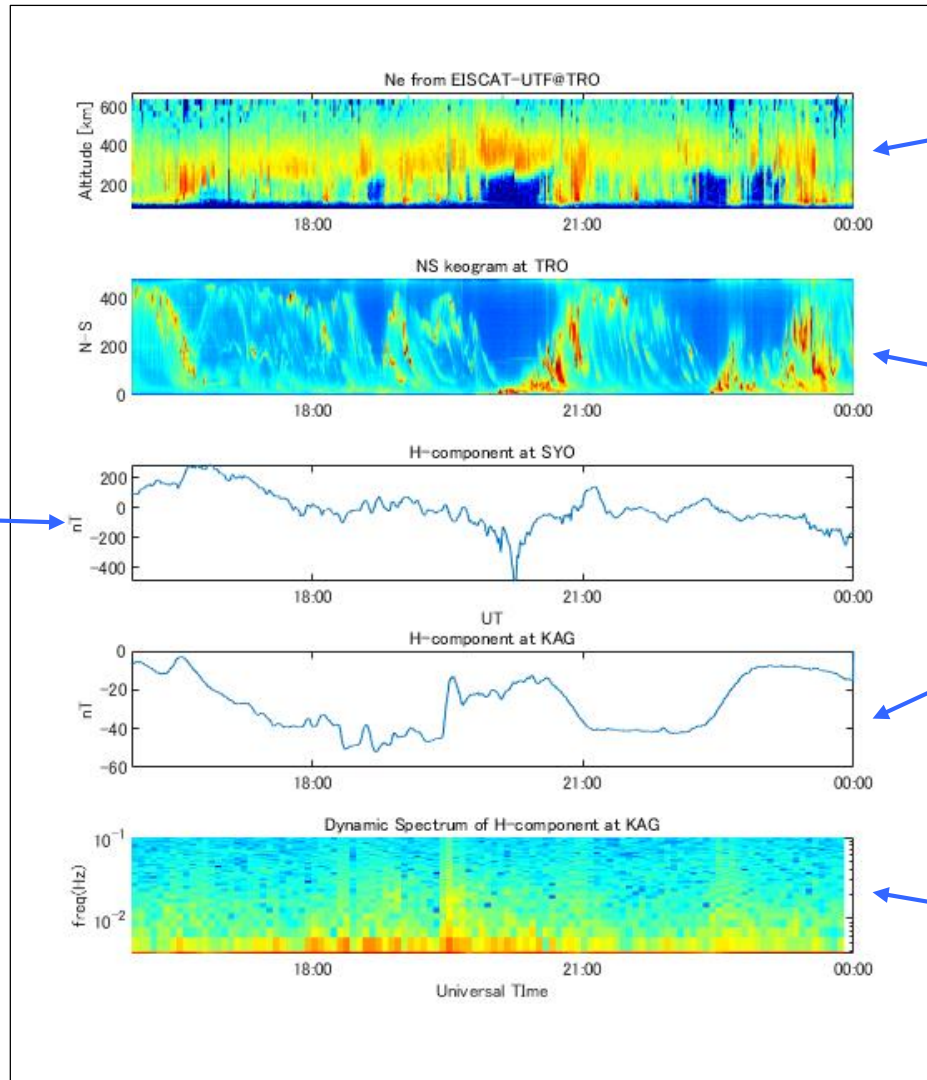
Time is a serial date number from a fixed, preset date (January 0, 0000). The "datetick" function changes the tick labels to date-based labels. The x-axis limit is specified by using the "datenum" function.



Proceed to more advanced analysis

Analyze more according to the sample scripts included in the M-UDAS package.

An example of visualization



Read CDF file of the geomagnetic field data at SYO with M-UDAS egg.

Read the electron density data from the EISCAT-UHF radar with M-UDAS.

Read the North-South keogram data at TRO with M-UDAS.

Read the geomagnetic field data at KAG from 210 MM with M-UDAS.

Calculate and plot the dynamic spectrum of the geomagnetic field data at KAG.

System requirements (As of August 2020)

OS	Windows / MacOS / Linux
MATLAB	8.5 or higher
CDFライブラリ	v3.7.1 or higher (https://cdf.gsfc.nasa.gov/)
対応フォーマット	CDF (Common Data Format) NetCDF (Network Common Data Form)

Notes

1. When you use M-UDAS egg, please refer to the rules of the road of the IUGONET project. Rules of the road: <http://www.iugonet.org/rules/>
2. Please note that we do not take any responsibility or liability for any damage or loss caused by M-UDAS egg.

Contact information

We welcome any questions and comments. Please send any feedback to
Email: iugonet-contact@iugonet.org

We describe the template in detail in the appendix.

```

1 function loadfunc_template(startTime, endTime, varargin)
2 %
3 % loadfun_template(startTime, endTime, varargin)
4 %
5 % A template load function.
6 %
7 % (Input arguments)
8 % startTime:      Start time (datetime or char or datenum)
9 % endTime:       End time (datetime or char or datenum)
10 % (Options)
11 % site:          Site name (ex., 'asb' or {'asb', 'ama', 'kuj'})
12 % datatype:     Data type (ex., '1sec' or {'1sec', '1min', '1hr'})
13 % parameter:    Parameter (ex., 'par1' or {'par1', 'par2', 'par3'})
14 % version:      Version number (ex., '1')
15 % downloadonly: 0:Load data after download, 1:Download only
16 % no_download:  0:Download files, 1:No download before loading data
17 % username:     Username (for https)
18 % password:     Password (for https)
19 %
20 % (Returns)
21 % automatically-named variables
22 %
23 % (Examples)
24 % template_loadfun('2017-1-1', '2017-1-2', 'site', 'asb');
25 % template_loadfun('2017-1-1', '2017-1-2', 'site', {'asb','kuj'});
26 %
27 % Written by Y.-M. Tanaka, April 30, 2020
28 % Modified by Y.-M. Tanaka, July 27, 2020
29 %

```

Line 1-29 header (lines commented out by “%”)

The header describes the summary of program, but it is not required.

If you want to share the program with other people or implement it into a package/large-scale software, the header description is helpful.

```
31 %*****%
32 %***** Step1: Set parameters *****%
33 %*****%
34 file_format = 'cdf'; % 'cdf' or 'netcdf'
35 url = 'http://www.iugonet.org/data/SITE/DATATYPE/YYYY/mag_SITE_DATATYPE_YYYYMMDD_vVERSION.cdf';
36 prefix = 'iug_';
37 site_list = {}; % ex. {'sta1', 'sta2', 'sta3'}
38 datatype_list = {}; % ex. {'1sec', '1min', '1hr'}
39 parameter_list = {}; % ex. {'par1', 'par2', 'par3'}
40 version_list = {}; % ex. {'01', '02', '03'}
41 acknowledgement = sprintf(['You can write the data use policy here.¥n',...
42     'This description is displayed when you use this load procedure.']);
43 rootpath = default_rootpath;
44
45 %*****%
46 %***** Step2: Set default values *****%
47 %*****%
48 site_def = "";
49 datatype_def = "";
50 parameter_def = "";
51 version_def = version_list;
52 downloadonly_def = 0;
53 no_download_def = 0;
54 username_def = "";
55 password_def = "";
56 time_format='yyyy-MM-dd HH:mm:ss Z'; % Time format string for NetCDF
```

--- Line 31~56: Parameters which users should edit in the template ---

The way to edit the “Step1” and “Step2” is described on p7-13 in details.

```

58 %===== Set input arguments =====%
59 p = inputParser;
60
61 %---- Required input arguments ----%
62 validTime = @(x) isdatetime(x) || ischar(x) || isscalar(x);
63 addRequired(p, 'startTime', validTime);
64 addRequired(p, 'endTime', validTime);
65
66 %---- Input arguments as parameters ----%
67 validSite = @(x) iscell(x) || ischar(x);
68 addParameter(p, 'site', site_def, validSite);
69 validDataType = @(x) iscell(x) || ischar(x);
70 addParameter(p, 'datatype', datatype_def, validDataType);
71 validParameters = @(x) iscell(x) || ischar(x);
72 addParameter(p, 'parameter', parameter_def, validParameters);
73 validVersion = @(x) isscalar(x);
74 addParameter(p, 'version', version_def, validVersion);
75 validDownloadOnly = @(x) isscalar(x);
76 addParameter(p, 'downloadonly', downloadonly_def, validDownloadOnly);
77 validNo_Download = @(x) isscalar(x);
78 addParameter(p, 'no_download', no_download_def, validNo_Download);
79 validUserName = @(x) ischar(x);
80 addParameter(p, 'username', username_def, validUserName);
81 validPassWord = @(x) ischar(x);
82 addParameter(p, 'password', password_def, validPassWord);

```

--- Line 58~82: Set the input arguments ---

Line 62-64, addRequired: Add the required input arguments *MATLAB function

Define the required input arguments.

Here, the required input arguments, startTime and endTime, and their types are defined.

Line 67-82, addParameter: Add the optional input arguments *MATLAB function

Define the optional input arguments.

Here, the optional input arguments, site, datatype, parameter, etc., their types and default values are defined.

```

84 parse(p, startTime, endTime, varargin{:});
85 startTime     = p.Results.startTime;
86 endTime      = p.Results.endTime;
87 site         = p.Results.site;
88 datatype     = p.Results.datatype;
89 parameter    = p.Results.parameter;
90 version      = p.Results.version;
91 downloadonly = p.Results.downloadonly;
92 no_download  = p.Results.no_download;
93 username     = p.Results.username;
94 password     = p.Results.password;
95
96 %===== Set local directory for saving data files =====%
97 ipos=strfind(url, '://')+3;
98 relpath = url(ipos:end);

```

--- Line 84~94: Substitute the input arguments for the variables ---

Substitute the input arguments for the MATLAB variables.

--- Line 97~98: Make the local path for saving files by using URL ---

Line 97-98, strfind(url, '://'): Make the local path for saving files by using URL *MATLAB function

Find the string, '://', from URL and use the string below '://' for the local path (relpath).

The 'relpath' is the directory where the downloaded data files are saved.

```

100 %==== Input of 'all'and '*' means all elements ==== %
101 st_vec=cellstr(site); % convert to cell of char
102 dt_vec=cellstr(datatype);
103 pr_vec=cellstr(parameter);
104 if strcmp(lower(st_vec{1}),'all') || strcmp(st_vec{1},'*')
105     st_vec=site_list;
106 end
107 if strcmp(lower(dt_vec{1}),'all') || strcmp(dt_vec{1},'*')
108     dt_vec=datatype_list;
109 end
110 if strcmp(lower(pr_vec{1}),'all') || strcmp(pr_vec{1},'*')
111     pr_vec=parameter_list;
112 end
113 vs=cellstr(version);

```

--- Line. 101-113: Define of 'all' and '*' for the input arguments ---

Line 101-103, cellstr(variable): Convert char or string to cell array *MATLAB function

Convert the input arguments as char and string to cell array.

Line 104-112, Define 'all' and '*' for the input arguments

If the options 'site' (or 'datatype' or 'parameter') is 'all' or '*', a cell array of site_list (or datatype_list or parameter_list) is used for its value.

Line 113, cellstr(variable): Convert char or string to cell array *MATLAB function

The default value of the 'version' option is version_list.

```

115 %===== Loop for site, datatype, and parameter =====%
116 %----- Loop for site -----%
117 for ist=1:length(st_vec)
118     st=st_vec{ist};
119     st=check_valid_name(st, site_list);
120     disp(st);
121     if isempty(st)
122         varname_st=prefix;
123     else
124         varname_st=[prefix, st];
125     end
126
127 %----- Loop for datatype -----%
128 for idt=1:length(dt_vec)
129     dt=dt_vec{idt};
130     dt=check_valid_name(dt, datatype_list);
131     disp(dt);
132     if isempty(dt)
133         varname_st_dt=varname_st;
134     else
135         varname_st_dt=[varname_st, '_', dt];
136     end
137
138 %----- Loop for parameter -----%
139 for ipr=1:length(pr_vec)
140     pr=pr_vec{ipr};
141     pr=check_valid_name(pr, parameter_list);
142     disp(pr);
143     if isempty(pr)
144         varname_st_dt_pr=varname_st_dt;
145     else
146         varname_st_dt_pr=[varname_st_dt, '_', pr];
147     end

```

--- Line. 117-147: Create the variable names for output ---

Line 117, 128, 139, for site_list (datatype_list, parameter_list) *MATLAB statement

Loop for site_list, datatype_list, parameter_list.

Line 119, 130, 141, check_valid_name: Validation for the options *M-UDAS function

Validate if given site (or datatype or parameter) is included in site_list (or datatype_list or parameter_list).

Line 121-125, 132-136, 143-147, Create the variable names for output

Create the output variable names using prefix, site, datatype, and parameter.

```

149 %===== Download files =====%
150 file_url = replace_string(url, startTime, endTime, st, dt, pr, vs);
151 file_relpath = replace_string(relpath, startTime, endTime, st, dt, pr, vs);
152 file_local = replace_string([rootpath, relpath], startTime, endTime, st, dt, pr, vs);
153 if no_download==1,
154     files = file_local;
155 else
156     files = file_download(file_url, 'rootpath', rootpath, 'files', file_relpath,...
157         'username', username, 'password', password);
158 end

```

--- Line. 150-152: Make URLs and local file paths (specialized process in M-UDAS egg) ---

Line 150-152, replace_string: Make URLs and local file paths *M-UDAS function

Make URLs and local file paths that you want to read.

--- Line. 153-158: Download files from remote servers ---

Line 153-158, file_download: Download files from remote servers (URL) *M-UDAS function

Download files from remote servers.

If no_download=1, it does not download files and reads files in the local directory.


```

160 %===== Load data into variables =====%
161 if downloadonly==0,
162     switch file_format
163         case 'cdf'
164             [data, info]=load_cdf(startTime, endTime, files);
165         case 'netcdf'
166             [data, info]=load_netcdf(startTime, endTime, files);
167         otherwise
168             error('Such a file_format is not allowed in this version.');
```

```

169     end
170
171     if ~isempty(data)
172         varname_base=[varname_st_dt_pr, '_'];
173         pretmp='test_';
174         set_varname(info, data, pretmp);
175         vartmp=whos([pretmp, '*']);
176         vartmp={vartmp.name};
177         varpart=strrep(vartmp, pretmp, '');
178
179         for i=1:length(vartmp)
180             eval(['assignin("base",', ''', varname_base, varpart{i}, ''', ', ', vartmp{i}, ');']);
181         end
182         eval(['assignin("base", "", varname_base, 'info', ', ', 'info');']);
183         clear data info;
184     end
185 end
```

--- Line. 161-169: Read files (specialized process in UDAS egg) ---

Line 161-169, load_cdf or load_netcdf: Read CDF or NetCDF files *M-UDAS function

Read CDF or NetCDF files and substitute real data for “data” and metadata for “info”.

--- Line. 171-184: Export data to variables in the workspace (specialized process in UDAS egg) ---

Line 172-177, set_varname: Create output variable names using metadata (info) *M-UDAS function

Create the output variable names using variable names included in the metadata (info).

Line 179-183, Export data to variables in the workspace.

Export data from the load function to variables in the base workspace.