

MPAD (Multi-Purpose APRS Daemon) Commands

by DF1JSL / Source:

https://github.com/joergschultzelutter/mpad/blob/master/docs/COMMANDS/ACTION_KEYWORDS.md

Based on the sender's call sign, MPAD automatically provides its content in the 'correct format'. By default, MPAD uses the metric system. MPAD will evaluate the sender's call sign. If a call sign from Liberia, Myanmar and the U.S. has been received, MPAD will automatically switch to the imperial system, thus providing the user wx temperature reports in Fahrenheit, speed in mph, distances in miles and so on. You can override this behavior with specific keywords (can be found at the end of this document)

WX data inquiries

One or multiple spaces between the respective separators are permitted

ALL Commands and keywords are **case insensitive**

For most of the cases, wx inquiries can be specified without any specific keyword; just specify the address. Unless any other keyword has been specified, wx **is always assumed as default keyword**. Action Keyword can be combined with date / daytime keyword parameters: **YES**

City, State and/or Country

Formats

<city>, <state>; <country>

<city>, <state>

<city>; <country>

Country = iso3166-a2 country (de, us, fr, uk, ...)

Example requests

Los Angeles, CA will try to return the Wx data for Los Angeles, CA, United States (Country code is not specified --> will automatically be set to "US")

Mountain View, CA; US will try to return the Wx data for Mountain View, CA, United States

Holzminden; de will try to return the Wx data for Holzminden, Germany

Spaces between the various parts of the message and the separators are permitted, meaning that e.g. both commands `los angeles , ca ; us` and `los angeles,ca;us` return the same result. This assumption is applicable to all keywords that are supported by MPAD.

Example response

(applicable to all wx request types)

A wx response consists of 1..n lines, The actual content as well as the message length is dependent on how much data is actually available. Here is an example for a wx report in metric format:

```
10-Jan-21 San Francisco,CA overcast clouds morn:10c day:13c eve:12c  
nite:11c sunrise/set 16:24/02:09UTC clouds:90% uvi:1.9 hPa:1026  
hum:57% dewpt:4c wndspd:2m/s wnddeg:52
```

Glossary

| | |
|------|-----------------------|
| morn | morning temperature |
| day | daytime temperature |
| eve | evening temperature |
| nite | nighttime temperature |

MPAD (Multi-Purpose APRS Daemon) Commands

sunrise/set sunrise and sunset in UTC time zone format
clouds clouds
uvi UV index
hPa air pressure
hum humidity
dewpt dew point
wndspd wind speed
wnddeg wind degrees

Zip Codes

Formats

zip <3-10 char/digit zipcode>[;iso3166-a2 country code] or **zip** <3-10 char/digit zipcode>[;iso3166-a2 country code] <5-digit code>. Note that a 5-digit zipcode without country code will automatically assume that the given zip code is a U.S. zip code; in this case, the country code will be set implicitly.

Example requests

zip 94043 returns the wx information for Mountain View, CA, United States
zip 85609 returns the wx information for Dragoon, AZ, United States
zip 85609;us returns the wx information for Dragoon, AZ, United States
zip 85609;de returns the wx information for Aschheim, Germany
94043 returns the wx information for Mountain View, CA, United States
37627;d returns wx information for Stadtoldendorf, Germany

A 5-digit zip code **without** iso-3166-a2 qualifier automatically sets the country setting to "US". Zip codes can be of 3..10 characters. If a zipcode consists of multiple parts which are separated by spaces, using such a zipcode is very likely not going to work.

Numeric Coordinates

Format: latitude/longitude (can be positive or negative)

Example requests

51.8458575/8.2997425
37.773972/-122.431297
-33.447487/-70.673676

Whereas possible, the program will try to turn these coordinates into a human readable address

Maidenhead / Grid locator

Formats

grid <4-or 6-character grid locator>
mh <4-or 6-character grid locator>

Example requests

grid jo41du
mh jo41

Example response

jo41du overcast clouds morn:-0c day:1c eve:1c nite:-0c
sunrise/set 08:31/16:36UTC clouds:90% uvi:0.5 hPa:1024 hum:97%
dewpt:0c wndspd:2m/s wnddeg:291

MPAD (Multi-Purpose APRS Daemon) Commands

Note: When a maidenhead locator is specified, the program will *not* try to translate this information to a human readable address, meaning that WX information will reference to the given grid data and not to a human-readable address (city, street and so on)

METAR/TAF Data for airport locations

You have three options:

specify a specific ICAO code

specify a specific IATA code

specify the **metar** or the **taf** keyword, which instructs the program to look for the airport that is close to your position. That 'nearest' airport position can either be based on the user's own call sign or alternatively on a different user's call sign.

If the given airport or the nearest one has been found but does **not** support METAR data, the program will try to provide you with a standard WX report for the airport's coordinates instead. If the airport is capable of providing METAR data but the METAR report cannot be retrieved, an error message is returned to the user.

Action Keyword can be combined with date / daytime keyword parameters: **YES**. If the user specifies the **metar** or the **taf** keyword in combination with the **full** keyword, the returned message will contain both METAR and TAF information. Otherwise, only the data related to the given keyword is returned (e.g. METAR data for **metar**, TAF data for **taf**). If WX data is returned, 'today'/'full' settings will be applied.

ICAO METAR / wx inquiries

Get a METAR/TAF report for a specific ICAO code. If the ICAO code is valid but the airport does not provide METAR data, a default wx report is returned instead.

Formats

`icao <4-character ICAO code>`

`<4-character ICAO code>`

Example requests

`icao eddf`

`eddf`

Example response

(applicable to all METAR options)

```
EDDF 171150Z 02008KT 340V050 5000 -SHSNRA FEW004 SCT011CB BKN019
```

```
03/01 Q1023 NOSIG ## TAF EDDF 171100Z 1712/1818 02008KT 9999
```

```
BKN030 TEMPO 1712/1716 SHRAGS BKN020TCU BECMG 1717/1720 FEW030
```

```
BECMG 1800/1802 02002KT BECMG 1806/1809 30005KT TEMPO 1811/1818
```

```
SHRAGS BKN020TCU SCT030
```

Specifying an ICAO code without keyword may or may not be successful as there is a log of ambiguity.

IATA METAR / wx inquiries

Get a METAR/TAF report for a specific IATA code by retrieving its associated ICAO code (and then performing an ICAO metar inquiry). If the IATA code is valid but the airport does not provide METAR data,

MPAD (Multi-Purpose APRS Daemon) Commands

a default wx report is returned instead. If the airport is capable of providing METAR data but the METAR report cannot be retrieved, an error message is returned to the user.

Formats

iata <3-character IATA code>
<3-character IATA code>

Example requests

iata sea
sea

Specifying an IATA code without keyword may or may not be successful as there is a log of ambiguity.

METAR / TAF keywords

Get a METAR / TAF report for the nearest airport in relation to the user's own call sign or a different call sign

Formats

| | |
|------------------------------|----------------------------------------------------------|
| metar <callsign>[-ssid] | METAR report that is closest to the call sign's position |
| taf <callsign>[-ssid] | TAF report that is closest to the call sign's position |
| metar <callsign>[-ssid] full | METAR report that is closest to the call sign's position |
| taf <callsign>[-ssid] full | TAF report that is closest to the call sign's position |
| metar | METAR report that is closest to the user's own position |
| taf | TAF report that is closest to the user's own position |
| metar full | METAR report that is closest to the user's own position |
| taf full | TAF report that is closest to the user's own position |

If no call sign is specified, then the user's own call sign (the one that he has send us the message with) is used

Example requests

metar ko4jvr-9
taf lb7ji
metar full

Based on the user's lat/lon, the program will then try to find the nearest airport for you. If that airport supports METAR data, the program is going to return METAR data to the user. Otherwise, it will try to pull a standard wx report for the airport's coordinates. If the airport is capable of providing METAR data but the METAR report cannot be retrieved, an error message is returned to the user.

WhereIs

Returns the geocoordinates/address info of the sender's position or a specific call sign. Respose data includes:

Maidenhead locator

MGRS coordinates

DMS coordinates

UTM coordinates

Human-readable address (whereas available)

Distance, direction and bearing to the call sign (if requested position differs from user's position)

Position' age (when was this position transmitted for the last time). Note: this information is only provided for the **whereis** command but not for the **whereami** command

Altitude information whereas present

MPAD (Multi-Purpose APRS Daemon) Commands

If address data is available for the requested coordinates, MPAD tries to honor the respective countries' native street / street number format. If the domestic format for your country is unknown, the address format for the street/street number data will always be `street street_number`, e.g. `Yellow Brick Road 12`. Open a ticket if I have accidentally messed up your country's street/ street number format. Action Keyword can be combined with date / daytime keyword parameters: **NO**

Formats

`whereami` returns my last known coordinates

`whereis <callsign>[-ssid]` returns the position information for this user

Example requests

`whereami`

`whereis df1jsl-1`

Example response

Request `whereis wa1gov-10` in metric format

Result Pos WA1GOV-10 Grid FN41lu95 DMS N41.51'17.4/W71.00'24.0 Dst 5829 km
Brg 50deg NE UTM 19T 333431 4635605 MGRS 19TCG3343135605
LatLon 41.85483/-71.00667 Taunton, 02718, US Seekell Street 329
Last heard 2021-01-25 23:32:42

Glossary

Grid Maidenhead locator

DMS Coordinates in degresses/minutes/seconds

Dst Distance to target in km (or miles)

Brg Bearing

UTM UTM coordinates

MGRS MGRS coordinates

LatLon coordinates in numerical format

street / zip code / country / city, if available

alt altitude in meters or feet, if available.

Sunrise/Sunset and Moonset/Moonrise

Returns the sunrise/sunset and moonset/moonrise info of the sender's position or a specific call sign. Note: values are calculated for the given day. In case the moonSET value overlaps from the previous date, then this is not taken into consideration. Action Keyword can be combined with date / daytime keyword parameters: **YES**

Formats

`riseset` returns the values based on the sender's position

`riseset <callsign>[-ssid]` returns the values based on a different call sign's position

Example requests

`riseset`

`riseset df1jsl-1 wednesday`

`riseset df1jsl-1`

MPAD (Multi-Purpose APRS Daemon) Commands

Example response

RiseSet DF1JSL-4 09-Jan GMT sun_rs 07:31-15:36 mn_sr 12:20-03:19

sun_rs - time settings for sunrise and sunset in GMT

mn_sr - time settings for moonset and moonrise in GMT

CWOP (Customer Weather's Observer Program)

Returns the latest CWOP Wx report of the nearest CWOP station (related to the sender's call sign or a different call sign) OR a specific CWOP station ID. Action Keyword can be combined with date / daytime keyword parameters: **NO**

Formats

`cwop` returns the nearest CWOP report, based on the user's position

`cwop <callsign>[-ssid]` returns the nearest CWOP report, based on the given call sign's position

`cwop <station_id>` returns the weather report for the given CWOP station ID

Example requests

`cwop`

`cwop df1jst-1`

`cwop at166`

Example response

```
CWOP AT166 09-Jan-21 1C Spd 0.0km/h Gust 1.6km/h Hum 95%
Pres 1021.6mb Rain(cm) 1h=0.0, 24h=0.05, mn=0.05
```

Satellite passes

Retrieves the next pass of the given satellite ID for the user's position. Satellites can be specified by their satellite IDs as defined in the respective amateur radio satellite tle file

<http://celestrak.org/NORAD/elements/amateur.txt>. The following rules apply:

If a satellite name contains spaces, then these spaces will be replaced by dashes. As an example, **SAUDISAT 1C** will internally be identified by **SAUDISAT-1C**

For convenience reasons, the ISS can be selected by requesting the satellite pass data for either **ISS** or **ZARYA**.

You can use the top command for telling MPAD to return more than one result (if available)

if a **date** / **daytime** setting is specified, the program will try to honor that setting and use it as the **starting** datetime for its calculations.

Action Keyword can be combined with date / daytime keyword parameters: **YES**

Formats

`satpass <satellite_name>` returns the next satellite pass of the requested satellite back to you.

`vispass <satellite_name>` same as `satpass`, but ensures that the result that is returned to you relates to a **visible** satellite pass.

Program assumptions for returning a result to you:

Elevation / Horizon is > 10 degrees

A **visible** satellite pass requires that between the satellite's rise time and its set time, it needs to be at least briefly in sunlight.

The following rules apply wrt to the **starting** point in time that is used for calculating the next passes:

MPAD (Multi-Purpose APRS Daemon) Commands

If you have specified an hour-based offset for date (e.g. 37h), MPAD uses the current UTC time, adds the hourly offset to it and starts the calculation

If you have specified a day-based offset for date that is at least one day in the future (e.g. `tomorrow`), MPAD calculates that date and sets the time H/M/S to zero. If you've also specified a daytime setting, then this daytime setting will be added on top as follows:

| | |
|----------------------|-----------|
| <code>morning</code> | 03:00 UTC |
| <code>daytime</code> | 12:00 UTC |
| <code>evening</code> | 17:00 UTC |
| <code>night</code> | 22:00 UTC |

if you don't specify any date or daytime settings, the current UTC date is the calculation's starting point. If there is no satellite pass for the day you've specified, the **next available** one will be presented to you. There is no error message if there is no satellite pass on the day that you've specified.

Example requests

```
satpass iss
vispass zarya
satpass saudisat-1c
vispass iss top5 friday noon
```

Satellite frequencies

Retrieves the satellite frequencies (whereas present) from JE9PEL's database. Satellites can be specified by their satellite IDs as defined in the respective amateur radio satellite tle file. The following rules apply:

If a satellite name contains spaces, then these spaces will be replaced by dashes. As an example, **SAUDISAT 1C** will internally be identified by **SAUDISAT-1C**

For convenience reasons, the ISS can be selected by requesting the satellite pass data for either **ISS** or **ZARYA**.

The requested satellite ID name **MUST** exist in the Celestrak list - this is MPAD's master. Those satellite IDs are matched against JE9PEL's satellite frequency data through an automated process. If a match is found on the satellite ID, the available frequencies are returned to the user (can be 0..n).

Action Keyword can be combined with date / daytime keyword parameters: **NO**

Formats

```
satfreq <satellite_name> returns the frequency data for the satellite
```

Example requests

```
satfreq iss
satfreq es'hail-2
```


Repeater data

Retrieves the nearest repeater, based on the user's position. In addition, 'band' and 'mode' filters can be specified.

Action Keyword can be combined with date / daytime keyword parameters: **NO**

Formats

`repeater [band] [mode]`

The positions for both parameters **band** and **mode** are position-interchangeable, meaning that `repeater [band] [mode]` and `repeater [mode] [band]` are both valid.

Band parameter needs to be specified with 'm' or 'cm' unit of measure, e.g. `70cm`, `2m`, `80m` Mode parameter can be one of the following: `fm`, `dstar`, `d-star`, `dmr`, `c4fm`, `ysf`, `tetra`, `atv`.

`d-star` and `dstar` are identical; the two options just exist because of convenience issues. `ysf` will convenience-map to `c4fm`.

Example requests

`repeater` returns the nearest repeater, regardless of its capabilities

`repeater c4fm` returns the nearest c4fm repeater without checking the band requirements

`repeater 70cm` returns the nearest 70cm repeater without checking the mode requirements

`repeater c4fm 70cm` returns the nearest c4fm repeater that runs on the 70cm band

`repeater 70cm c4fm` same command as in the previous example

Note that this keyword can be used in conjunction with the `top_x` keyword. If you e.g. want to get the data for up to 3 repeaters near your location, use

`repeater top3` or

`repeater c4fm 70cm top3`

See the documentation for the `top_x` keyword on how to use it properly.

Example response

Message enumerations are only included if more than one result is available.

```
#1 Bad Iburg / Doerenberg Dst 43 km 333 deg NNW Rx 430.9375  
Tx 438.5375 WIRES-X,Startreflektor DL-Nordwest 70cm JO42AE #2  
Poembsen Dst 53 km 98 deg E Rx 430.5125 Tx 439.9125 70cm JO41MS
```

If you've specified band or mode as a query parameter, that data will not be part of the outgoing message (I'm trying to save some bytes here). So if you've e.g. issued a `repeater c4fm 70cm` command, both `c4fm` and `70cm` references will not be part of the outgoing message - I simply assume that you remember what you've requested. However, if you did not request band and/or mode, the data will be added to the outgoing message.

OpenStreetMap Nearby Category Searches

MPAD allows you to find e.g. a supermarket or abank that the nearest one to your location. OSM offers a couple of classification categories https://wiki.openstreetmap.org/wiki/Nominatim/Special_Phrases/EN Some - but not all - of these categories are currently supported by MPAD. These categories are:

| OSM Category Code | Meaning | OSM Category Code | Meaning |
|--------------------------|-----------------------------|--------------------------|--------------------------|
| aerodrome | Airport | hostel | Hostel |
| alpine_hut | Alpine Hut | hotel | Hotel |
| ambulance_station | Ambulance | information | Information |
| atm | ATM | laundry | Laundry |
| bakery | Bakery | mall | (Shopping) mall |
| bank | Bank | motel | Motel |
| butcher | Butcher | motorcycle | Motorcycle |
| car_rental | Car Rental | optician | Optician |
| car_repair | Car Repair | pharmacy | Pharmacy |
| charging_station | Charging Station | phone | Phone |
| chemist | Chemist / Pharmacy | photographer | Photographer |
| clinic | Clinic / Hospital | police | Police Office / Precinct |
| college | College | post_box | Post Box |
| deli | Deli | post_office | Post Office |
| dentist | Dentist | pub | Pub / Bar |
| department_store | Department Store | shoes | Shoes |
| doctors | Doctors | subway | Subway Station |
| drinking_water | Drinking Water | supermarket | Supermarket |
| dry_cleaning | Dry Cleaning | taxi | Taxi |
| electronics | Electronics Shop | telephone | Telephone (booth) |
| fire_station | Fire Station | tobacco | Tobacco |
| fuel | Fuel / Petrol / Gas Station | toilets | Toilets |
| guest_house | Guest house | train_station | Train Station |
| hairdresser | Hairdresser | veterinary | Veterinary |
| hospital | Hospital | university | University |

The OSM category code can be specified with or without its associated keyword (`osm`). Note that some of the categories with shorter names may be mis-interpreted by the parser as something else if you submit such a category without the `osm` keyword. For example, the category `pub` might be misinterpreted as IATA code `PUB` as the program parser processes the IATA data prior to the OSM category data. First come, first serve. When in doubt: submit the category with a keyword :-)

Formats

```
osm <osm_category_name>
<osm_category_name>
```

Example requests

```
osm police
police
```

Note that this keyword can be used in conjunction with the `top_x` keyword. If you e.g. want to see up to 3 supermarkets near your location, use

MPAD (Multi-Purpose APRS Daemon) Commands

`osm supermarket top3` or
`supermarket top3`

See the documentation for the `top_x` keyword on how to use it properly.

Example responses

Message enumerations are only included if more than one result is available.

#1 Volksbank Weserbergland eG Am Schloßpark 2 Holzminden Dst 8 km
Brg 203 deg SSW #2 Braunschweigische Landessparkasse Am Wildenkiel
Holzminden Dst 8 km Brg 204 deg SSW #3
Braunschweigische Landessparkasse Angerstraße 12 Bevern Dst 6 km
Brg 313 deg NW

Send a message to DAPNET

Sends a message text to a DAPNET user Action Keyword can be combined with date / daytime keyword parameters: **NO**

Formats

`dapnet <user> <text>`

user can be specified with or without SSID (if specified with SSID, the SSID will be removed).

Example requests

`dapnet df1jsl Hello World!` Sends the text `Hello World!` to DAPNET user `df1jsl`

Example response

In case of success, MPAD will return the message

Successfully sent DAPNET message to [user]

IF MPAD had not been configured for DAPNET access or there was an error during sending the message, an error message will be sent to the user.

Send a position report via email

Sends an email to a user on the Internet, containing retailed position information on your call sign's current APRS position. Action Keyword can be combined with date / daytime keyword parameters: **NO**

Formats

`posmsg <mail_address>` or `posrpt <mail_address>` (both keywords do the same thing)

This keyword can be combined with the `lang` keyword (default value: `en`), thus allowing you to get e.g. Russian address data in cyrillic characters.

Example requests

`posmsg test@gmail.com` Sends your call sign's position report to email recipient `test@gmail.com`

Example response

In case of success, MPAD will return the message

The requested position report was emailed to its recipient

IF MPAD has not been configured for DAPNET access or there was an error during sending the message, an error message will be sent to the user.

Fortuneteller

MPAD has the power to predict your future - all you have to do is ask a question (and send an APRS message of course). Action Keyword can be combined with date / daytime keyword parameters: **NO**

Formats

fortuneteller
magic8ball
magic8
m8b

This keyword can be combined with the lang keyword, meaning that for certain languages, you will receive a localised prediction of your future :-). Default language is en. Supported languages so far are en,de,es,it,fr,nl,ru,tr and cn

The purpose for this keyword is mainly for UTF-8 / localised content testing.

Example requests

magic8ball lang de will request a localised Magic 8 Ball prediction in German.

Example response

Well Better not tell you now :-)

Radiosonde landing prediction

Based on the coordinates and altitude settings from aprs.fi, MPAD determines the probe's landing coordinates predictions and returns them to the user. Similar to the **whereis** keyword, this keyword also provides additional direction information (address, bearing etc) to the user. Action Keyword can be combined with date / daytime keyword parameters: **NO**

This keyword uses predict.habhub.org for the whole prediction landing prediction process. Here's an illustration of the input parameters and how they are used on the habhub.org web site:

| | |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| latitude | latitude value from aprs.fi |
| longitude | longitude value from aprs.fi |
| altitude | altitude value from aprs.fi |
| launch time | time stamp from aprs.fi |
| launch date | time stamp from aprs.fi |
| ascent rate | equal to aprs.fi clmb value if that value is > 0. If clmb has a negative value, then ascent rate is set to 0.01. |
| descent rate | set to value 6 if clmb value is > 0. If clmb has a negative value, then descent rate is set to the absolute (sign-less) value of clmb |
| burst altitude | clmb = positive: set to 25000 if aprs.fi altitude is < 25000. Set to 30000 if aprs.fi altitude is > 25000 and < 30000. Set to 35000 if aprs.fi altitude is > 30000 and < 35000. Set to 38000 if aprs.fi altitude is > 35000. clmb = negative: burst altitude = altitude + 1 |

Formats

sonde <callsign>

MPAD (Multi-Purpose APRS Daemon) Commands

callsign can be specified with or without SSID (if specified with SSID, the SSID will be removed). Additionally, radiosonde callsigns which do not follow the standard APRS call sign pattern are also recognised. **sonde** can be combined with the **lang** keyword.

Example requests

sonde S3421116 will request a radiosonde landing prediction for the given probe

Example response

Landing Pred. 'S3421116' Lat/Lon 47.7853/10.6331 02-Apr 15:45UTC Dst 481 km Brg 159deg SSE Grid JN57hs58 Addr: Baerenleitenweg, Marktoberdorf, Landkreis Ostallgaeu, Bavaria, 87616, Germany

General help

Returns general program help to the user. Action Keyword can be combined with date / daytime keyword parameters: **NO**

Formats

info
help

Example requests

info
help

Switching between the metric and imperial system

By default, the program will automatically switch from the metric system (default) to the imperial system if the **sender's** call sign is from Liberia, Myanmar or the United States (per Wikipedia, these are the only three countries which still use the imperial system over the metric system).

metric system (**default**): temperatures in degrees Celsius, speed in km/h, rain levels in cm etc.

imperial system: temperatures in degrees Fahrenheit, speed in mph, rain levels in inch etc.

If you don't want to rely on the automatic mode, you can always override this automated setting by specifying one of the following keywords:

Formats

mtr OR **metric**
imp OR **imperial**

Example requests

metric
imperial

To illustrate how this works, have a look at these two examples:

Request: **san francisco, ca; us** (issued by my German call sign). The result is returned in metric format:
09-Jan-21 San Francisco,CA few clouds morn:9c day:12c eve:11c
nite:10c sunrise/set 16:25/02:08UTC clouds:16% uvi:2.2 hPa:1025
hum:68% dewpt:6c wndspd:1m/s wnddeg:50

MPAD (Multi-Purpose APRS Daemon) Commands

Now let's request the same wx report - but this time, we want it to be delivered in imperial format (this is what an American user would see as default format):

```
09-Jan-21 San Francisco,CA few clouds morn:48f day:53f eve:52f  
nite:50f sunrise/set 16:25/02:08UTC clouds:16% uvi:2.2 hPa:1025  
hum:68% dewpt:43f wndspd:2mph wnddeg:50
```

MPAD does not perform an imperial-to-metric calculation (or vice versa) but requests the desired format as part of its REST requests to e.g. Openweathermap and other services. Apart from rounding these values in order to limit the message length, all data is displayed 'as is'.

Language

Allows you to specify a language in a (somewhat) ISO639-1 format. Default language is 'en'. Action Keyword can be combined with date / daytime keyword parameters: **YES**

Currently, this keyword is **only** used for WX reports from Openweathermap. In addition, it only provides a localised version of the wx free text, e.g. en=**snow**, de=**Schnee**, pl=**Snieg**. If you enable your MPAD instance for UTF-8 support, those parts of the message may contain e.g. cyrillic characters.

Formats

lang [iso639-1 code]

lng [iso639-1 code]

Example requests

lang de

Erding;de lng pl returns a wx report for the city of Erding in Germany where the wx report's free text part will be in Polish.

Supported languages

| | |
|----|----------------------------|
| af | Afrikaans |
| al | Albanian |
| ar | Arabic |
| az | Azerbaijani |
| bg | Bulgarian |
| ca | Catalan |
| cz | Czech |
| da | Danish |
| de | German |
| el | Greek |
| en | English (default) |
| eu | Basque |
| fa | Persian (Farsi) |
| fi | Finnish |
| fr | French |

| | |
|----|------------|
| gl | Galician |
| he | Hebrew |
| hi | Hindi |
| hr | Croatian |
| hu | Hungarian |
| id | Indonesian |
| it | Italian |
| ja | Japanese |
| kr | Korean |
| la | Latvian |
| lt | Lithuanian |
| mk | Macedonian |
| no | Norwegian |
| nl | Dutch |
| pl | Polish |

| | |
|--------|---------------------|
| pt | Portuguese |
| ro | Romanian |
| ru | Russian |
| sv, se | Swedish |
| sk | Slovak |
| sl | Slovenian |
| sp, es | Spanish |
| sr | Serbian |
| th | Thai |
| tr | Turkish |
| ua, uk | Ukrainian |
| vi | Vietnamese |
| cn | Chinese Simplified |
| tw | Chinese Traditional |
| zu | Zulu |

MPAD (Multi-Purpose APRS Daemon) Commands

Enforce outgoing UTF-8 messages

By default, MPAD 'downgrades' all of its content prior to sending its responses to the user. The reason behind this is simple: nearly all transceivers cannot digest unicode characters and display them as e.g. placeholders etc. You can override this behavior in two ways:

change the program configuration of your local instance (`mpad_enforce_unicode_messages`) use the `unicode` keyword

Formats

`unicode`

Example requests

`fortuneteller lang ru unicode`

Allow to receive more than one result

Certain keywords such as the `osm` or the `repeater` keyword allow more than one result. For example, an OSM query for the nearest `supermarket` along with the `top5` command will return up to 5 results to you which are even ordered by distance between your current location and the target location. For the `repeater` keyword, you can run a query such as `repeater c4fm 70cm top3` which will return the nearest 3 repeater results to you.

Default number of results is 1. You can change the number of results with the `top` keywords. A `top2` keyword will return up to 2 results and a `top5` keyword will try to do the same for 5 results. If not enough results are available, MPAD will return a lower number of results to the user.

Note: if you use this keyword, then either place it at the end or the beginning of the keyword that you want to use it for. `top3 repeater c4fm` or `repeater c4fm top3` are both fine. However, a `repeater top3 c4fm` breaks the existing regex and you will receive repeater data which may not be applicable to the `c4fm` filter.

Formats

`top2`

`top3`

`top4`

`top5`

Example requests

see "Formats"

Date settings

MPAD is able to understand the following keywords:

today (Note: if not specified, this is the program default)

tomorrow

monday or **mon**

tuesday or **tue**

wednesday or **wed**

thursday or **thu**

friday or **fri**

saturday or **sat**

sunday or **sun**

tonight,tonite --> results in date today and daytime night value

1d ... 7d results in 1...7 days in the future.

Additionally, you can specify an 'hour' keyword in order to receive hourly wx reports (other action keywords may ignore these values). 1h up until 47h are permitted.

If you request the same weekday that is today, then the program assumes that you refer to that day in the next week.

Whereas noted for the respective action keyword, these date setting keywords (and also the daytime setting keywords) can be combined with the action keywords. Examples:

San Francisco, CA tomorrow full --> returns a full wx report for location 'San Francisco' and date setting 'tomorrow'

riseset thursday --> returns sunrise/sunset and moonrise / moonset values for Thursday.

San Diego, CA 46h --> returns a report for location 'San Diego' for the wx in 46h

Deensen;de 5d --> returns a report for location 'Deensen' in Germany for the wx in 5 days

Daytime settings

MPAD is able to understand the following keywords:

full Returns all data for the given day (**program default**). Note that full can refer to time frames or the amount of data that is returned by the program (dependent on the action keyword)

Morning (Keywords: **morn**, **morning**)

Daytime (Keywords: **day**, **daytime**, **noon**)

Evening (Keywords: **eve**, **evening**)

Night (Keywords: **nite**, **night**, **tonite**, **tonight**).

tonite and **tonight** keywords will set the date keyword to today unless the user has already specified a different day.

All references are to UTC time zone settings.

Whereas noted for the respective action keyword, these daytime setting keywords (and also the date setting keywords) can be combined with the action keywords. Examples:

San Francisco, CA tomorrow full --> returns a full wx report for location 'San Francisco' and date setting 'tomorrow'

satpass iss friday morning --> returns the first pass of the ISS on Friday morning

riseset thursday --> returns sunrise/sunset and moonrise / moonset values for Thursday.

If an action keyword does not support daytime information (e.g. repeater, riseset etc.), the daytime keyword information will be ignored.