

```

0001 // ++++++
0002 funcprot(0);
0003 // ++++++
0004
0005 // Get body position from "cen" in EOD
0006 // cjd: TREF time scale
0007
0008 function [pos_ECI]=getBodyPos(name, cjd)
0009     pos_ICRS=CL_eph_de405(name,cjd,"Earth");
0010
0011     pos_ECI=CL_fr_convert("ICRS","ECI",cjd,pos_ICRS);
0012 endfunction
0013
0014 // Azimuth and elevation of celestial body from location loc at times t
0015
0016 function [azim, elev]=sphPos(t, loc, cbody)
0017     pos_ECI=[];
0018
0019     pos_ECI=getBodyPos(cbody,t);
0020
0021     // Transform position of celestial body in ECF frame
0022     // (Earth Centered Fixed Frame)
0023
0024     pos_ECF=CL_fr_convert("ECI","ECF",t,pos_ECI);
0025
0026     // Transform elliptical coordinates of the position of celestial body
0027     // to cartesian coordinates
0028
0029     loc_car=CL_co_ell2car(loc);
0030
0031     // Get the transformation matrix to the local Topocentric North frame
0032     // The "topocentric North" local frame is defined as follows:
0033     // X: unit vector in the horizontal plane and towards north
0034     // Y: unit vector in the horizontal plane and towards west
0035     // Z: unit vector in direction of the local vertical
0036
0037     M_topoN=CL_fr_topoNMat(loc);
0038
0039     // Transform position of celestial body to the local Topocentric North
0040     // frame in cartesian coordinates
0041
0042     pos_topoN=M_topoN*(pos_ECF-loc_car);
0043
0044     // Transform cartesian coordinates to spherical coordinates
0045
0046     pos_topoN_sph=CL_co_car2sph(pos_topoN);
0047
0048     // Get azimuth and elevation of celestial body
0049
0050     azim=pos_topoN_sph(1);
0051     elev=pos_topoN_sph(2);
0052
0053 endfunction
0054
0055 // #####

```

```

0056 // Main
0057
0058 // longitude/latitude/altitude of location
0059
0060 lon=11.42*%CL_deg2rad;
0061 lat=48.15*%CL_deg2rad;
0062 alt=531;
0063
0064 // scal=reference date (string, calendar)
0065 scal=[];
0066
0067 desc_param=list(..
0068     CL_defParam("Longitude of location",lon,units=["rad","deg"],..
0069         valid='$x>=0 & $x <=360'), ..
0070     CL_defParam("Latitude of location",lat,units=["rad","deg"],..
0071         valid='$x>=-90 & $x <=90'), ..
0072     CL_defParam("Altitude of location",alt,units=["m"],..
0073         valid='$x>=-100 & $x <=3000'), ..
0074     CL_defParam("Reference time (calendar format, TREF)",scal,typ="cal") ..
0075 );
0076
0077 [lon,lat,alt,scal]=CL_inputParam(desc_param);
0078
0079 loc=[lon;lat;alt];
0080
0081 t0=CL_dat_cal2cjd(CL_dat_str2cal(scal));
0082
0083 // ++++++
0084
0085 cal = CL_dat_cjd2cal(t0);
0086
0087 months=["Jan","Feb","Mär","Apr","Mai","Jun","Jul","Aug","Sep",..
0088     "Okt","Nov","Dez"];
0089 str_day=sprintf("%d %s %d", cal(3), months(cal(2)), cal(1));
0090 str_hour=sprintf("%02d:%02d", cal(4), cal(5));
0091 dateText=str_day+" "+str_hour+" (TREF)";
0092
0093 // ++++++
0094
0095 function [razi]=cvAzi(pazi)
0096     razi=360-pazi*%CL_rad2deg;
0097 endfunction
0098
0099 function [rele]=cvEle(pele)
0100     rele=pele*%CL_rad2deg;
0101 endfunction
0102
0103 // ++++++
0104
0105 cBodyNames=["Sun","Moon","Mercury","Venus","Mars",..
0106     "Jupiter","Saturn","Uranus","Neptune","Pluto"];
0107
0108 pNames=["Sonne","Mond","Merkur","Venus","Mars",..
0109     "Jupiter","Saturn","Uranus","Neptun","Pluto"];
0110

```

```

0111 pData=[];
0112
0113 NbBodies=size(cBodyNames,"c");
0114
0115 for Ix=1:NbBodies
0116     [azim,elev]=sphPos(t0,loc,cBodyNames(Ix));
0117
0118     pData(Ix,1:2)=[cvAzi(azim),cvEle(elev)];
0119 end
0120
0121 mprintf('\n\n    Sichtbarkeit von Himmelskörpern:\n');
0122
0123 mprintf('    Zeit - %s\n',dateText);
0124
0125 mprintf('    Standort -\n\tLänge:  %7.2f°\n\tBreite:  %7.2f°\n\tHöhe:\t  %d
m \n',...
0126         lon*%CL_rad2deg,lat*%CL_rad2deg,alt);
0127
0128 mprintf('    Azimuth - Nord: 0° -> Ost: 90°\n\n');
0129
0130 mprintf('    Übern Horizont:\n');
0131
0132 for Ix=1:NbBodies
0133
0134     if (pData(Ix,2)>0) then
0135         mprintf('    %s- Azimuth:  %11.6f° | Elevation:  %11.6f° \n',...
0136                 pNames(Ix),pData(Ix,1),pData(Ix,2));
0137     end
0138 end
0139
0140 mprintf('\n');
0141 mprintf('    Unterm Horizont:\n');
0142
0143 for Ix=1:NbBodies
0144
0145     if (pData(Ix,2)<=0) then
0146         mprintf('    %s- Azimuth:  %11.6f° | Elevation:  %11.6f° \n',...
0147                 pNames(Ix),pData(Ix,1),pData(Ix,2));
0148     end
0149 end
0150
0151 // #####
0152
0153 // name of body (capitalized)
0154 function [name]=body_name(body)
0155     if (part(body,1:1) == 'm')
0156         name = "Moon";
0157     else
0158         name = "Sun";
0159     end
0160 endfunction
0161
0162 // ++++++
0163
0164 // thickness after plot

```

```

0165 function Thickness(th)
0166     e=gce();
0167     e.children.thickness = th;
0168 endfunction
0169
0170 // ++++++
0171
0172 // xtring interface
0173 function String(x, y, text, col, sz, style)
0174     xstring(x,y,text);
0175     e=gce();
0176     e.font_foreground = col;
0177     e.font_size = sz;
0178     e.font_style = style;
0179 endfunction
0180
0181 // ++++++
0182
0183 // marks for sun and moon
0184 function plot_mark(name, azi, ele, col, fg, sz)
0185     plot(azi,ele,"o");
0186     e=gce();
0187     e.children.mark_background=col;
0188     e.children.mark_foreground=fg;
0189     e.children.mark_size=sz;
0190     String(azi,ele," "+name, col,3,5);
0191 endfunction
0192
0193 // ++++++
0194
0195 f=scf();
0196 f.visible="off";
0197 f.immediate_drawing="off";
0198 a=gca();
0199
0200 col1 = color("orange"); // Sun
0201 col2 = color("steelblue"); // Moon
0202 col3 = color("darkseagreen"); // meridians
0203 col4 = color("forestgreen"); // meridians
0204
0205 dbElev=80;
0206 db = [0,-dbElev;360,dbElev]; // data bounds
0207
0208 // ++++++
0209
0210 col1=color("gold"); // Sonne
0211 col2=color("magenta"); // Mond
0212 col3=color("aquamarine"); // Merkur
0213 col4=color("limegreen"); // Venus
0214 col5=color("orangered"); // Mars
0215 col6=color("darkgoldenrod"); // Jupiter
0216 col7=color("royalblue"); // Saturn
0217 col8=color("salmon"); // Uranus
0218 col9=color("mediumorchid"); // Neptun
0219 col10=color("peachpuff4"); // Pluto

```

```

0220
0221 plotNames=["SO","MO","Me","Ve","Ma","Ju","Sa","Ur","Ne","Pl"];
0222
0223 sizes=[15, 14, 9, 12, 10, 11, 8, 7, 6, 5];
0224 //      SO  MO  Me Ve  Ma  Ju Sa  Ur Ne Pl
0225
0226 cols=[col1,col2,col3,col4,col5,col6,col7,col8,col9,col10];
0227 //      SO  MO  Me  Ve  Ma  Ju  Sa  Ur  Ne  Pl
0228
0229 for Ix=1:NbBodies
0230     plot_mark(plotNames(Ix),pData(Ix,1),pData(Ix,2),cols(Ix),1,sizes(Ix));
0231 end
0232
0233 // ++++++
0234
0235 // various settings
0236
0237 a.x_ticks = tlist("ticks", 0:60:360, string(0:60:360));
0238 a.y_ticks = tlist("ticks", -dbElev:10:dbElev, string(-dbElev:10:dbElev));
0239 a.x_label.text = "Azimuth [°]";
0240 a.y_label.text = "Elevation [°]";
0241
0242
0243 months = ["Jan", "Feb", "Mär", "Apr", "Mai", "Jun", "Jul", "Aug", "Sep", "Okt", "Nov", "Dez"];
0244 str_day = sprintf("%d %s %d", cal(3), months(cal(2)), cal(1));
0245 str_hour = sprintf("%02d:%02d", cal(4), cal(5));
0246
0247 a.title.text = "Sonne, Mond und Planeten
- " + str_day + " " + str_hour + " (TREF)";
0248 a.data_bounds = db;
0249
0250 CL_g_stdaxes(a);
0251 a.grid = [0,0];
0252 xgrid(color("grey"));
0253
0254 CL_g_legend(a, str=['Sonne';'Mond';'Merkur';'Venus';'Mars';'Jupiter';..
0255                   'Saturn';'Uranus';'Neptun';'Pluto']);
0256
0257 f.immediate_drawing="on";
0258 f.visible="on";

```