NERS User Guide

Introduction

NERS library interface provides 6 public routines. Routine **ners_init** is called before any other NERS functions. It specifies NERS control file and the time range of intended EOP use. Time in NERS library is the interval elapsed since 2000.01.01_00:00:00.0 in TAI time scale. Units of time are seconds. Instead of time, events are often get tags of UTC function. In order to convert UTC tag to TAI time, NERS provides function **ners_get_utcmtai** that returns function UTC minus TAI on the specified timetag UTC. Routine **ners_init** initializes internal data structure of NERS object that is defined in **ners.i** or **ners.h**include block. This object is then passed to any other NERS function. NERS provides either the vector Earth orientation parameters on the specified moment of time or the table of the Earth orientation series for the specified range of time and the specified time step. At the end, function **ners_quit** releases memory acquired by **ners_init**.

The Earth orientation can be described either as a 3×3 rotation matrix that transforms a Cartesian vector from the rotating terrestrial coordinate system to the inertial non-rotating celestial coordinate system or as parameters on which this matrix depends. For practical needs the rotation matrix and its time derivatives are sufficient to perform astronomical reduction for Earth's rotation. However, NERS also provides the parameters that describes the Earth's rotation on which the rotation matrix depends as well. These parameters are empirical corrections to a deterministic model. Since the Earth's rotation depends on motion of the hydrosphere and atmosphere, it cannot be described with a deterministic model with the accuracy comparable with accuracy of observations and should be continuously monitored using space geodesy observations. For historical reasons several alternative Earth orientation parameters were used. For instance. For instance the rate of change of the angular variable along the axis 3, i.e. the axial motion can be described as Euler angle 3, UT1 rate, or the length of day. NERS provides many alternative Earth orientation parameters. These parameters are not independent.

Use cases

- Transform a Carthesian vector of station coordinates in the terrestrial coordinate system to the inertial celestial coordinate system at the specified moment of time. All you need is to call three NERS routines: **ners_init**, **ners_get_eop**, **ners_quit**. First, you run **ners_init** and specify the time range. If you need to compute the rotation matrix only for one epoch, the range may be just ± 10 seconds. If you are going to compute the transformation matrix for a number of epochs, just specify TAI time of the earliest and the lastest epoch. The artgument time is 64-bit float number of seconds elapsed since 2000.01.01_00:00:00.0 TAI. Second, you run **ners_get_eop** and specify the time. The type of the EOP is "mat". A product of this mastrix with the Carthesian vector of station coordinates in the terrestrial coordinate system will transform it to the inertial celestial coordinate system. If you need the transfromation matrix for a number of epochs, you call **ners_get_eop** in a cycle. Finally, you call **ners_quit** to release dynamic memory allocated by *NERS*.

See **example/ners_fortran_example_01.f** and **example/ners_c_example_01.c**.

- Get TAI time for a given UTC timetag. It is rather common to have value of of the non-differntiable function of time UTC(t) for events instead of time TAI. In order to process such data you need to convert UTC timetag to TAI moment of time. All you need is to call three NERS routines: **ners_init**, **ners_get_utcmtai**, **ners_quit**. First, you run **ners_init** and specify the time range. If you are going to compute convert UTC to TAI for a number of epochs, just specify TAI time of the earliest and the lastest epoch. Second, you run **ners_get_utcmtai** and specify the UTC timetag elapsed since epoch 2000.01.01_00:00:00.0 UTC. The routine returns function UTC minus TAI. Substracting this function from UTC you tet TAI time: TAI = UTC - UTC_M_TAI. If you need transfrom UTC time tag to TAI for a number of epochs, you call **ners_get_utcmtai** in a cycle. Finally, you call **ners_quit** to release dynamic memory allocated by *NERS*.

  See **example/ners_fortran_example_02.f** and **example/ners_c_example_02.c**.

- Get the last time epoch of the short-term EOP forecast. You need call three NERS routines: **ners_init**, **ners_inq**, **ners_quit**. First, you run **ners_init** and specify the time range. For this case the range may be just ± 10 seconds of the current moment. Second, you run **ners_inq** wioth request parameter "range". The routine returns three parameters: start time for EOP assimilaiton, stop time of the short-term EOP forecast, and stop time of the long-term prediction. Time is in seconds elapsed since 2000.01.01_00:00:00.0 TAI. Finally, you call **ners_quit** to release dynamic memory allocated by *NERS*.

  See **example/ners_fortran_example_04.f** and **example/ners_c_example_04.c**.

Description of NERS public functions

- **ners_init** — initializes internal NERS data structure, parses the configuration file and reads the leap second file. It also specifues the interval of time for the Earth orientation parameters that will be computed at the next call to NERS. This interval should not exceed 10 days.

```
FORTRAN:  NERS_INIT ( CONFIG_FILE, NERS, TIME_TAI_START, TIME_TAI_STC
C:        cners_init ( char * config_file, struct ners_struct * ners,
                   double time_tai_beg, double time_tai_end, int * iuer )
```

Input parameters:

config_file (CHARACTER*(*)) -- name of the NERS configuration file
NERS__CONFIG defined in ners_local.i

```
                              specifies the default configuration file
ners          ( NERS__TYPE    ) -- NERS internal data structure
time_tai_start ( REAL*8       ) -- Start time of the time range
                              Units: seconds since 2000.01.01_00:00:00.0 TAI
 Value -1.0D0 means the current moment.
time_tai_stop  ( REAL*8       ) -- Stop  time of the time range.
                              Units: seconds since 2000.01.01_00:00:00.0 TAI
 Value -1.0D0 means the current moment.

Input/Output parameter:

iuer          ( INTEGER*4     ) -- Error parameter.
                              On input:  -1 -- to print the error message if
                                               an error is detected;
                                         otherwise, not to print the message:
                              On output:  0 -- normal completion
                                         >0 -- error code during NERS
                                               initialization
```

- **ners_get_eop** — returns array of the Earth Orientation Parameters on the specified moment of time. The type of Earth orientation parameter is defined in variable cpar. If it is called the first time, it communicates with the server, downloads the NERS message and prepares the interpolating polynomials for the specified time range, computes intermediate angles and expand them into B-spline basis over the specified time period. Then it computes the requested Earth orientation parameters. When it is called the next time, it reuses the B-spline coefficients computed previously, unless more than 20 minutes elapsed from the computation of the coefficients.

```
FORTRAN:  NERS_GET_EOP ( NERS, TIME_TAI, CPAR, M_PAR, L_PAR, PARS, IU
C:      cners_get_eop ( struct ners_struct * ners, double time_tai_obs, char * cpar,
                int mpar, int * lpar, double eops[], int * iuer )

 Input parameters:

 ners      ( NERS__TYPE    ) -- NERS internal data structure
 time_tai  ( REAL*8        ) -- Time. Units: seconds since 2000.01.01_00:00:00.0 TAI
 cpar      ( CHARACTER*(*) ) -- Earth orientation parameter name:

           mat     --  3x3 matrix of the transformation from the terrestrial
                       coordinate system to the celestial coordinate system
           matr    --  First time derivative of the 3x3 matrix of the
                       transformation from the terrestrial coordinate system
                       to the celestial coordinate system
           matrr   --  Second time derivative of the 3x3 matrix of the
                       transformation from the terrestrial coordinate system
```

to the celestial coordinate system
matall -- 3x3x3 array of the matrix of the transformation from
the terrestrial coordinate system to the celestial
coordinate system, its first and second time derivative.
[1:3,1:3,1] -- transformation matrix from the terrestrial
coordinate system to the celestial
coordinate system,
[1:3,1:3,2] -- first time derivative of the transformation
matrix above.
[1:3,1:3,3] -- second time derivative of the transformation
matrix above.

utcmtai -- Value of function UTC minus TAI. Units: s.

e1 -- Euler angle around axis one. Units: rad.
e2 -- Euler angle around axis two. Units: rad.
e3 -- Euler angle around axis three. Units: rad.
e1r -- First time derivative of Euler angle around axis one. Units: rad/s.
e2r -- First time derivative of Euler angle around axis two. Units: rad/s.
e3r -- First time derivative of Euler angle around axis three. Units: rad/s.
e1rr -- Second time derivative of Euler angle around axis one. Units: rad/s
e2rr -- Second time derivative of Euler angle around axis two. Units: rad/s.
e3rr -- Second time derivative of Euler angle around axis three. Units: rad/s

ut1mtai -- Angle UT1 minus TAI. Units: s.
ut1rat -- First time derivative of angle UT1 minus TAI. Units s/s.
ut1rr -- Second time derivative of angle UT1 minus TAI. Units s/s^2.
lod -- Length of day. Units: s.
lodr -- Rate of change of the length of day. Units: s/day.
xpol -- X pole coordinate. Unit: arcsec.
ypol -- Y pole coordinate. Unit: arcsec.
xpolr -- First time derivative of X pole coordinate. Unit: arcsec/day.
ypolr -- First time derivative of Y pole coordinate. Unit: arcsec/day.
xpolrr -- Second time derivative of X pole coordinate. Unit: arcsec/day**2.
ypolrr -- Second time derivative of Y pole coordinate. Unit: arcsec/day**2.
eop3 -- Array of three EOP parameters:
1: X pole coordinate. Unit: arcsec.
2: Y pole coordinate. Unit: arcsec.
3: UT1 minus TAI. Unit: s.
eop3r -- Array of six EOP parameters:
1: X pole coordinate. Unit: arcsec.
2: Y pole coordinate. Unit: arcsec.
3: UT1 minus TAI. Unit: s.
4: First time derivative of X pole coordinate. Unit: arcsec/day.
5: Second time derivative of Y pole coordinate. Unit: arcsec/day.

6: First time derivative of UT1 minus TAI coordinate. Unit: s/day.

dpsi   -- Nutation angle in longitude. Units: rad
deps   -- Nutation angle in obliquity. Units: rad
dpsir   -- First time derivative of nutation angle in longitude. Units: rad/s.
depsr   -- First time derivative of nutation angle in obliquity. Units: rad/s.
nut   -- Array of two EOP parameters:
1: Nutation angle in longitude. Units: rad
2: Nutation angle in obliquity. Units: rad
nutr   -- Array of four EOP parameters:
1: Nutation angle in longitude. Units: rad
2: Nutation angle in obliquity. Units: rad
3: First time derivative of nutation angle in longitude. Units: rad/s.
4: First time derivative of nutation angle in obliquity. Units: rad/s.
eops   -- Array of eight EOP parameters:
1: X pole coordinate. Unit: arcsec.
2: Y pole coordinate. Unit: arcsec.
3: UT1 minus TAI.   Unit: s.
4: First time derivative of X pole coordinate. Unit: arcsec/day.
5: Second time derivative of Y pole coordinate. Unit: arcsec/day.
6: First time derivative of UT1 minus TAI coordinate. Unit: s/day.
7: Nutation angle in longitude. Units: arcsec
8: Nutation angle in obliquity. Units: arcsec

h1   -- Contribution of empirical harmonic variations in the EOPs with respe
h2   -- Contribution of empirical harmonic variations in the EOPs with respe
h3   -- Contribution of empirical harmonic variations in the EOPs with respe
h1r   -- First time derivative of the contribution of empirical harmonic variatio
h2r   -- First time derivative of the contribution of empirical harmonic variatio
h3r   -- First time derivative of the contribution of empirical harmonic variatio
h1rr   -- Second time derivative of the contribution of empirical harmonic varia
h2rr   -- Second time derivative of the contribution of empirical harmonic varia
h3rr   -- Second time derivative of the contribution of empirical harmonic varia
heo   -- Array of three components of the contribution of empirical harmonic
1: Contribution of empirical harmonic variations in the EOPs with resp
2: Contribution of empirical harmonic variations in the EOPs with resp
3: Contribution of empirical harmonic variations in the EOPs with resp
heor   -- Array of three components of the first time derivative of the contribu
1: First time derivative of the contribution of empirical harmonic variat
2: First time derivative of the contribution of empirical harmonic variat
3: First time derivative of the contribution of empirical harmonic variat
heorr   -- Array of three components of the first time derivative of the contribu
1: Second time derivative of the contribution of empirical harmonic var
2: Second time derivative of the contribution of empirical harmonic var
3: Second time derivative of the contribution of empirical harmonic var

m_par (INTEGER*4) -- Maximum number of the elements in the output array.

Output parameters:

l_par (INTEGER*4) -- Actual number of the EOPs
pars (REAL*8) -- Array of EOPs. Dimension: m_par.

Input/Output parameter:

iuer (INTEGER*4) -- Error parameter.
On input: -1 -- to print the error message if
an error is detected;
otherwise, not to print the message:
On output: 0 -- normal completion
>0 -- error code during NERS
initialization

- **ners_quit** — releases memory for internal data structures and un-initialize them.

FORTRAN: NERS_QUIT (CODE, NERS)
C: cners_quit (int * quit_code, struct ners_struct * ners)

Input parameter:

code (INTEGER*4) -- Level of re-initialization:
NERS__EXP -- release memory with coefficients of EOP expansion fo
the interval of time specified in previous call of
routine ners_init.
NERS__FCS -- release memory with the NERS forecast message
NERS__ALL -- release memory for all internal data structures

Input/Output parameter:

ners (NERS__TYPE) -- NERS internal data structure

- **ners_inq** — inquirers either the time interval of EOP availability or the time of the EOP forecast generation.

FORTRAN: NERS_INQ (NERS, REQ, M_PAR, L_PAR, PARS, IUER)
C: cners_quit (int * quit_code, struct ners_struct * ners)

Input parameter:

ners (NERS__TYPE) -- NERS internal data structure

```
req    ( CHARACTER*(*) ) -- request type. Supported requests:

            range        -- request of the EOP range. Returns three values:
                            1: start time of EOP availability. Units: seconds since
                               2000.01.01_00:00:00.0 TAI.
                            2: stop  time of the EOP or the EOP forecast availability.
                               Units: seconds since 2000.01.01_00:00:00.0 TAI.
                            3: stop  time of the long-term EOP prediction
            fcs_gen_time -- request of the EOP forecast message generation time.
                            Return one value:
                            1: time of the EOP forecast generation. Units: seconds since
                               2000.01.01_00:00:00.0 TAI.

m_par  ( INTEGER*4     ) -- Maximum number of the elements in the output array.
```

Output parameters:

```
l_par     ( INTEGER*4     ) -- Actual number of the EOPs
pars      ( REAL*8        ) -- Array of EOPs. Dimension: m_par.
```

Input/Output parameter:

```
iuer          ( INTEGER*4     ) -- Error parameter.
                                   On input:  -1 -- to print the error message if
                                                    an error is detected;
                                              otherwise, not to print the message:
                                   On output:  0 -- normal completion
                                              >0 -- error code during NERS
```

- **ners_get_series** — returns a 2D array of the Earth Orientation Parameters time series for the specified time range with the specified time step. The type of Earth orientation parameter(s) is defined in variable cpar.

```
FORTRAN:  NERS_GET_SERIES ( NERS, TIME_TAI_START, TIME_TAI_STOP, T
                            DIM1, DIM2, NS, TIM, SER, IUER )
C:        cners_get_series  ( struct ners_struct * ners, double time_tai_beg, double time_
                       double tim_step, char * cpar, int m_par, int m_ser, int * ns,
                       double tim[], double ser[], int * iuer )
```

Input parameters:

```
ners            ( NERS__TYPE    ) -- NERS internal data structure
time_tai_start ( REAL*8        ) -- Start time of EOP series. Units: seconds since 2000.(
time_tai_stop  ( REAL*8        ) -- Stop  time of EOP series. Units: seconds since 2000
time_step      ( REAL*8        ) -- Time step of the EOP series. Units: seconds since 20
```

cpar (CHARACTER*(*)) -- Earth orientation parameter name.

euler -- Euler angles of the Earth rotation
1: Euler angle around axis one. Units: rad.
2: Euler angle around axis two. Units: rad.
3: Euler angle around axis three. Units: rad.

euler_r -- First time derivative of Euler angles of the Earth rotation
1: First time derivative of Euler angle around axis one. Units: rad/s.
2: First time derivative of Euler angle around axis two. Units: rad/s.
3: First time derivative of Euler angle around axis three. Units: rad/s.

polu -- Array of three EOP parameters:
1: X pole coordinate. Unit: arcsec.
2: Y pole coordinate. Unit: arcsec.
3: UT1 minus TAI. Unit: s.

polur -- Array of six EOP parameters:
1: X pole coordinate. Unit: arcsec.
2: Y pole coordinate. Unit: arcsec.
3: UT1 minus TAI. Unit: s.
4: First time derivative of X pole coordinate. Unit: arcsec/day.
5: Second time derivative of Y pole coordinate. Unit: arcsec/day.
6: First time derivative of UT1 minus TAI coordinate. Unit: s/day.

eops -- Array of eight EOP parameters:
1: X pole coordinate. Unit: arcsec.
2: Y pole coordinate. Unit: arcsec.
3: UT1 minus TAI. Unit: s.
4: First time derivative of X pole coordinate. Unit: arcsec/day.
5: Second time derivative of Y pole coordinate. Unit: arcsec/day.
6: First time derivative of UT1 minus TAI coordinate. Unit: s/day.
7: Nutation angle in longitude. Units: arcsec
8: Nutation angle in obliquity. Units: arcsec

heo -- Array of three components of the contribution of empirical harmonic
1: Contribution of empirical harmonic variations in the EOPs with resp
2: Contribution of empirical harmonic variations in the EOPs with resp
3: Contribution of empirical harmonic variations in the EOPs with resp

heor -- Array of three components of the first time derivative of the contribu
1: First time derivative of the contribution of empirical harmonic variat
2: First time derivative of the contribution of empirical harmonic variat
3: First time derivative of the contribution of empirical harmonic variat

dim1 (INTEGER*4) -- First dimension of the EOP series array. The dimens
dim2 (INTEGER*4) -- Second dimension of the EOP series array. The dime

Output parameters:

ns (INTEGER*4) -- the number of returned EOP values
tim (REAL*8) -- Array of EOP time epochs. Dimension: ns. Units: seconc
ser (REAL*8) -- Two dimensional Array of EOPs. Dimensions: dim1,dim2
filled elements along the first dimension is ns. The number of
filled elements along the second dimension depends on value
parameter cpar.

Input/Output parameter:

iuer (INTEGER*4) -- Error parameter.
On input: -1 -- to print the error message if
an error is detected;
otherwise, not to print the message:
On output: 0 -- normal completion
>0 -- error code during NERS

- **ners_get_utcmtai** — returns value of UTC minus TAI on the specified UTC time tag.

FORTRAN: NERS_GET_UTCMTAI (NERS, UTC, UTC_M_TAI, IUER)
C: cners_get_utcmtai (struct ners_struct * ners, double * utc_obs,
double * utc, int * iuer)

Input parameters:

ners (NERS__TYPE) -- NERS internal data structure
utc (REAL*8) -- UTC time tag. Units: seconds elapsed since 2000.01.01_00

Output:

utc_m_tai (REAL*8) -- UTC minus TAI function. Units: seconds.

Input/Output parameter:

iuer (INTEGER*4) -- Error parameter.
On input: -1 -- to print the error message if
an error is detected;
otherwise, not to print the message:
On output: 0 -- normal completion
>0 -- error code during NERS

This web page was prepared by Leonid Petrov ()
Last update: