(https://www.nasa.gov/)
(http://www.jpl.nasa.gov/)
(http://www.caltech.edu/)

(/)

Horizons API

Version: 1.2 (2022 September)
change log

This API provides access to JPL's Horizons system (https://ssd.jpl.nasa.gov/horizons/) by specifying Horizons parameters as query parameters in the URL. An alternate file-based Horizons API (/doc/horizons_file.html) is available if you would prefer to submit a Horizons batch input file via HTTP POST.

> Additional information about available query parameters is available in the Horizons batch example/instructions (https://ssd.jpl.nasa.gov/ftp/ssd/horizons_batch.txt) document.

HTTP Request

GET `https://ssd.jpl.nasa.gov/api/horizons.api`

Example Ephemeris Query

Request an observer ephemeris of Mars in plain-text format:

- `https://ssd.jpl.nasa.gov/api/horizons.api?format=text&COMMAND='499'&OBJ_DATA='YES'&MAKE_EPHEM='YES'&EPHEM_TYPE='OBSERVER'&CENTER='500@399'&START_TIME='2006-01-01'&STOP_TIME='2006-01-20'&STEP_SIZE='1%20d'&QUANTITIES='1,9,20,23,24,29'`

Query Parameters

Special/reserved characters require URL encoding.

The following tables show available query parameters for the Horizons API. The first table "Common Parameters" lists query parameters that are independent of the selected `EPHEM_TYPE`. The next table "Ephemeris-Specific Parameters" lists remaining parameters that are specific to `OBSERVER`, `VECTORS`, and `ELEMENTS` ephemeris types and indicates the `EPHEM_TYPE` for which they are available. The last two tables, "SPK File Parameters" and "Close-Approach Table Parameters", list parameters specific to those special ephemeris types.

Details on parameters can be found in the Horizons documentation (https://ssd.jpl.nasa.gov/horizons/manual.html) and also the Horizons batch example (https://ssd.jpl.nasa.gov/ftp/ssd/horizons_batch.txt) text file. In the tables below, direct links to the appropriate section of the Horizons documentation for specific parameters is given in the "Manual" column when available.

Common Parameters

Parameter	Default	Allowable Values/Format	Description	Manual
format	`json`	`json`, `text`	specify output format: `json` for JSON or `text` for plain-text	
COMMAND	*none*	see details below	target search, selection, or enter user-input object mode	link (https://ssd.jpl.nasa.gov/horizons/manual.html#select)
OBJ_DATA	`YES`	`NO`, `YES`	toggles return of object summary data	
MAKE_EPHEM	`YES`	`NO`, `YES`	toggles generation of ephemeris, if possible	
EPHEM_TYPE	`OBSERVER`	`OBSERVER`, `VECTORS`, `ELEMENTS`, `SPK`, `APPROACH`	selects type of ephemeris to generate (see details below)	

Parameter	Default	Allowable Values/Format	Description	Manual
EMAIL_ADDR	*none*	any valid email address	optional; used only in the event of highly unlikely problems needing follow-up	

Note: No output will be generated unless `MAKE_EPHEM='YES'` and/or `OBJ_DATA='YES'`, unless a database-search only.

Ephemeris-Specific Parameters

In the following table, the column heading `O` (OBSERVER), `V` (VECTORS), and `E` (ELEMENTS) indicate the ephemeris type for which the parameter is available.

Symbol `x` means the parameter is fully supported for the indicated ephemeris type. Symbol `-` means the parameter is partially supported and no symbol (blank) means the parameter is not supported.

Parameter	O	V	E	Default	Allowable Values/Format	Description	Manual
CENTER	x	x	x	`Geocentric`	see details below	selects coordinate origin (observing site)	link (https://ssd.jpl.nasa.gov/horizons/manual.html#c
REF_PLANE		x	x	`ECLIPTIC`	`ECLIPTIC`, `FRAME`, `BODY EQUATOR`	Ephemeris reference plane (can be abbreviated `E`, `F`, `B`, respectively))	
COORD_TYPE	x	x	x	`GEODETIC`	`GEODETIC`, `CYLINDRICAL`	selects type of user coordinates	link (https://ssd.jpl.nasa.gov/horizons/manual.html#c
SITE_COORD	x	x	x	`'0,0,0'`		set coordinate triplets for `COORD_TYPE`	link (https://ssd.jpl.nasa.gov/horizons/manual.html#c

Parameter	O	V	E	Default	Allowable Values/Format	Description	Manual
START_TIME	x	x	x	*none*		specifies ephemeris start time	link (https://ssd.jpl.nasa.gov/horizons/manual.htm
STOP_TIME	x	x	x	*none*		specifies ephemeris stop time	link (https://ssd.jpl.nasa.gov/horizons/manual.htm
STEP_SIZE	x	x	x	`'60 min'`	see details below	ephemeris output print step. Can be fixed time, uniform interval (unitless), calendar steps, or plane-of-sky angular change steps. See also `TLIST` alternative.	link (https://ssd.jpl.nasa.gov/horizons/manual.htm
TLIST	x	x	x	*none*	see details below	list of up to 10,000 of discrete output times. Either Julian Day numbers (JD), Modified JD (MJD), or calendar dates	
TLIST_TYPE	x	x	x	*none*	`JD`, `MJD`, `CAL`	optional specification of type of time in TLIST	

Parameter	O	V	E	Default	Allowable Values/Format	Description	Manual
QUANTITIES	x			`'A'`		list of desired output quantity option codes	link (https://ssd.jpl.nasa.gov/horizons/manual.html#o link (https://ssd.jpl.nasa.gov/horizons/manual.html#ob
REF_SYSTEM	x	x	x	`ICRF`	`ICRF`, `B1950`	specifies reference frame for any geometric and astrometric quantities	link (https://ssd.jpl.nasa.gov/horizons/manual.html#f
OUT_UNITS		x	x	`KM-S`	`KM-S`, `AU-D`, `KM-D`	selects output units for distance and time; for example, `AU-D` selects astronomical units (au) and days (d)	
VEC_TABLE		x		`3`	see details below	selects vector table format	link (https://ssd.jpl.nasa.gov/horizons/manual.html#c
VEC_CORR		x		`NONE`	`NONE`, `LT`, `LT+S`	selects level of correction to output vectors; `NONE` (geometric states), `LT` (astrometric light-time corrected states) or `LT+S` (astrometric states corrected for stellar aberration)	

Parameter	O	V	E	Default	Allowable Values/Format	Description	Manual
CAL_FORMAT	x			`CAL`	`CAL`, `JD`, `BOTH`	selects type of date output; `CAL` for calendar date/time, `JD` for Julian Day numbers, or `BOTH` for both CAL and JD	
CAL_TYPE	x	x	x	`MIXED`	`MIXED`, `GREGORIAN`	Selects Gregorian-only calendar input/output, or mixed Julian/Gregorian, switching on 1582-Oct-5. Recognized for close-approach tables also.	
ANG_FORMAT	x			`HMS`	`HMS`, `DEG`	selects RA/DEC output format	
APPARENT	x			`AIRLESS`	`AIRLESS`, `REFRACTED`	toggles refraction correction of apparent coordinates (Earth topocentric only)	
TIME_DIGITS	x	x	x	`MINUTES`	`MINUTES`, `SECONDS`, `FRACSEC`	controls output time precision	

Parameter	O	V	E	Default	Allowable Values/Format	Description	Manual
TIME_ZONE	x			`'+00:00'`		specifies local civil time offset relative to UT	
RANGE_UNITS	x			`AU`	`AU`, `KM`	sets the units on range quantities output	
SUPPRESS_RANGE_RATE	x			`NO`	`NO`, `YES`	turns off output of delta-dot and rdot (range-rate)	
ELEV_CUT	x			`'-90'`	integer [-90:90]	skip output when object elevation is less than specified	
SKIP_DAYLT	x			`NO`	`NO`, `YES`	toggles skipping of print-out when daylight at `CENTER`	
SOLAR_ELONG	x			`'0,180'`		sets bounds on output based on solar elongation angle	

Parameter	O	V	E	Default	Allowable Values/Format	Description	Manual
AIRMASS	x			`38.0`		select airmass cutoff; output is skipped if relative optical airmass is greater than the single decimal value specified. Note than `1.0` =zenith, `38.0` ~= local-horizon. If value is set >= `38.0`, this turns OFF the filtering effect.	
LHA_CUTOFF	x			`0.0`		skip output when local hour angle exceeds a specified value in the domain `0.0` < X < `12.0`. To restore output (turn OFF the cut-off behavior), set X to `0.0` or `12.0`. For example, a cut-off value of `1.5` will output table data only when the LHA is within +/- 1.5 angular hours of zenith meridian.	

Parameter	O	V	E	Default	Allowable Values/Format	Description	Manual
ANG_RATE_CUTOFF	x			`0.0`		skip output when the total plane-of-sky angular rate exceeds a specified value	
EXTRA_PREC	x			`NO`	`NO`, `YES`	toggles additional output digits on some angles such as RA/DEC	
CSV_FORMAT	x	x	x	`NO`	`NO`, `YES`	toggles output of table in comma-separated value format	
VEC_LABELS		x		`YES`	`NO`, `YES`	toggles labeling of each vector component	
VEC_DELTA_T		x		`NO`	`NO`, `YES`	toggles output of the time-varying delta-T difference TDB-UT	
ELM_LABELS			x	`YES`	`NO`, `YES`	toggles labeling of each osculating element	
TP_TYPE			x	`ABSOLUTE`	`ABSOLUTE`, `RELATIVE`	determines what type of periapsis time (Tp) is returned	

Parameter	O	V	E	Default	Allowable Values/Format	Description	Manual
R_T_S_ONLY	x			`NO`	`NO`, `YES`	toggles output only at target rise/transit/set	

SPK File Parameters (`EPHEM_TYPE=SPK`)

The following parameters are available when EPHEM_TYPE is `SPK`. Binary SPK file generation is restricted to small-bodies only (asteroids and comets). See the Horizons documentation (https://ssd.jpl.nasa.gov/horizons/manual.html#spk) for details. See also the section below for details on SPK file output format.

Parameter	Default	Description	Manual
START_TIME	*none*	specifies ephemeris start time	link (https://ssd.jpl.nasa.gov/horizons/manual.html#time)
STOP_TIME	*none*	specifies ephemeris stop time	link (https://ssd.jpl.nasa.gov/horizons/manual.html#time)

Close-Approach Table Parameters (`EPHEM_TYPE=APPROACH`)

The following parameters are available when EPHEM_TYPE is `APPROACH`. See the Horizons documentation (https://ssd.jpl.nasa.gov/horizons/manual.html#ca) on close-approach tables for details.

Parameter	Default	Units	Allowable Values/Format	Description
CA_TABLE_TYPE	`STANDARD`		`STANDARD`, `EXTENDED`	Extended close-approach tables include Julian Day numbers. B-plane information is also output if there is a covariance for the object stored in the system database or specified with user-input elements.
TCA3SG_LIMIT	`14400`	minutes		maximum computed 3-sigma uncertainty in time of Earth close-approach
CALIM_SB	`0.05`	au		sets the spherical radius within which the nominal target must pass one of the perturbing asteroids (Ceres, Pallas, Vesta, etc.) to activate close-approach flagging

Parameter	Default	Units	Allowable Values/Format	Description
CALIM_PL	`.1, .1, .1, .1, 1.0, 1.0, 1.0, 1.0, .1, .003`	au		sets the spherical radius within which the nominal target must pass one of the planets (or the Moon) to activate close-approach flagging, in the order: Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune, Pluto, and Moon

User-specified Heliocentric Ecliptic Osculating Elements

The following parameters are used to define an arbitrary small-body target for numerical integration. See the Horizons documentation (https://ssd.jpl.nasa.gov/horizons/manual.html#user) for a complete description of these parameters.

Heliocentric ecliptic osculating elements may be specified with [`TP` , `QR`], [`MA` , `A`], or [`MA` , `N`]. If elements are not specified with [`TP` , `QR`], they will be computed from the other possible input pairs because Horizons always internally uses [`TP` , `QR`].

Note that other obliquities might be implicit in ecliptic elements from non-JPL sources. For accurate propagation, users should verify which obliquity is relevant, and may have to convert elements obtained from a different source before input here.

Parameter	Units	Description
OBJECT		Name of user input object
EPOCH		Julian Day number (JDTDB) of osculating elements
ECLIP		Reference ecliptic frame of elements: `J2000` or `B1950` . `J2000` assumes the IAU76/80 J2000 obliquity of 84381.448 arcsec relative to the ICRF reference frame. `B1950` assumes FK4/B1950 obliquity of 84404.8362512 arcsec.
EC		Eccentricity
QR	au	Perihelion distance (see note above)
TP		Perihelion Julian Day number (see note above)
OM	deg	Longitude of ascending node wrt ecliptic
W	deg	Argument of perihelion wrt ecliptic
IN	deg	Inclination wrt ecliptic

Parameter	Units	Description
MA	deg	Mean anomaly (see note above)
A	au	Semi-major axis (see note above)
N	deg/d	Mean motion (see note above)

Optional Small-body Parameters

Parameter	Object Type	Default	Units	Description
RAD			km	Object radius
H	asteroid			Absolute magnitude parameter
G	asteroid			Magnitude slope parameter; can be < 0
M1	comet			Total absolute magnitude
M2	comet			Nuclear absolute magnitude
K1	comet			Total magnitude scaling factor
K2	comet			Nuclear magnitude scaling factor
PHCOF	comet			Phase coefficient for `K2` =5
A1			au/d^2	Radial non-gravitational acceleration
A2			au/d^2	Transverse non-gravitational acceleration
A3			au/d^2	Normal non-gravitational acceleration
R0		2.808	au	Non-grav. model constant, normalizing distance
ALN		0.1112620426		Non-grav. model constant, normalizing factor
NM		2.15		Non-grav. model constant, exponent m
NN		5.093		Non-grav. model constant, exponent n
NK		4.6142		Non-grav. model constant, exponent k

Parameter	Object Type	Default	Units	Description
DT			d	Non-grav. lag/delay parameter (comets)
AMRAT			m^2/kg	Solar pressure model, area/mass ratio
SRC				Square-root covariance from JPL (upper-triangular, vector-stored)
EST				Estimated non-grav parameter names `A1`, `A2`, `A3`, `DT` in order matching SRC. If no estimated non-gravs in the solution's SRC, `EST=' '` or unspecified.

Description of Selected Parameters

`EPHEM_TYPE` Parameter

There are five ephemeris types available. These are selected using the `EPHEM_TYPE` parameter. The table below shows the `EPHEM_TYPE` values and corresponding ephemeris type.

EPHEM_TYPE	Description	Typical Usage
OBSERVER	Observables (RA/DEC, Az/El, physical aspect, angles, uncertainties)	telescope observations
ELEMENTS	Osculating orbital elements	instantaneous geometry over time, celestial mechanics
VECTORS	Cartesian state vectors and uncertainties	dynamical studies, propagation, programming
APPROACH	Close approaches to planets (and 16 largest asteroids)	encounter planning & hazards
SPK	SPK binary trajectory files (asteroids and comets only)	time-continuous states, navigation, mission-planning, plug-in for visualization tools

`COMMAND` Parameter

The `COMMAND` parameter looks up the target body, selecting it if a unique match is found, or returning a list of matches if there are multiple matches. It is also used to enter user-defined objects. The assigned value for `COMMAND` should be delimited with single quotes (`'`) and some symbols within the quotes may need to be encoded to be successfully passed as a URL, as described below.

To specify a major-body (a planet, natural-satellite, spacecraft, or special pre-computed cases), specify the unique Horizons ID associated with that body. For example, `COMMAND='499'` selects the planet Mars while `COMMAND='5'` selects the Jupiter system barycenter. `COMMAND='MB'` will return a list of current major-bodies and their ID codes.

For small-bodies (asteroids and comets), a search syntax with numerous parameter keywords separated by semi-colons can be used to find objects with certain combinations of properties. For discussion of small-body look-up searching, see Horizons documentation (https://ssd.jpl.nasa.gov/horizons/manual.html#search).

When you already know what you are looking for, numbered asteroids can be selected with the IAU number followed by a semicolon (`;`). For example, `COMMAND='1;'` selects the small-body "1 Ceres". If the semicolon is not specified, `COMMAND='1'`, this will instead select the Mercury barycenter. The semicolon syntax resolves numbering ambiguities between major bodies and small-bodies.

Small-body designations, such as '1999 AN10' can be specified using the DES keyword `COMMAND='DES=1999 AN10;'`.

SPK IDs, if known, are considered a type of designation. For example, `COMMAND='99942;'` (IAU number) and `COMMAND='DES=2099942;'` (SPK ID), or `COMMAND='Apophis;'` (name) all select the object Apophis, though the numeric forms are guaranteed to be unique matches, while a name search may not be.

To input an arbitrary target for numerical integration, heliocentric ecliptic small-body osculating elements can be specified by users when `COMMAND=';'`. TLEs (two-line elements) may be input for Earth-orbiting objects by setting `COMMAND='TLE'`.

IMPORTANT: API calls must encode semicolons as `%3B` in the URL since the symbol usually cannot be directly sent without premature interpretation. Implementation of the above examples in a URL would be `COMMAND='1%3B'`, `COMMAND='DES=1999%20AN10%3B'`, `COMMAND='Apophis%3B'` and `COMMAND='%3B'`, for example.

Please see the Horizons documentation (https://ssd.jpl.nasa.gov/horizons/manual.html#select) for more details and examples of object selection, including specification of surface targets.

`CENTER` Parameter

This setting can be the observing site name, its unique IAU site code, the string 'coord' (which tells the system to use coordinate information stored in parameters `SITE_COORD` and `COORD_TYPE`) or the string 'geo' (meaning geocenter or body center).

Please see the Horizons documentation (https://ssd.jpl.nasa.gov/horizons/manual.html#center) for more details and examples.

`STEP_SIZE` Parameter

This setting specifies table output print times using the form '**integer {units} {mode}**'.

{units}	Minimum Abbreviation	Description	Example
days	d	fixed interval	`STEP_SIZE='1d'`
hours	h	fixed interval	`STEP_SIZE='3%20h'` (w/URL-encoded space)
minutes	m	fixed interval	`STEP_SIZE='10m'`
years	y	calendar stepping	`STEP_SIZE='1 year'`
months	mo	calendar stepping	`STEP_SIZE='1 mo'`
		unitless fixed intervals	`STEP_SIZE='86400'` (1 second output given 1 day between start/stop)

Calendar stepping

If calendar-stepping units of years or months are specified, output steps will follow the calendar based on the starting date. For example, if the start is 2008-Feb-29, and output is requested at "1 year" steps, output will be returned only for Feb 29 calendar days in those leap years having 29 days in February.

If output is requested at "1 month" intervals, output will occur for every successive month on the 29th of that month. If a start date on the 31st is requested, output will only occur for months having 31 days.

Unitless output stepping

If only the integer component is given in `STEP_SIZE`, the interval between specified `START_TIME` and `STOP_TIME` is divided into that many evenly spaced output steps. For example, if start and stop times are 1 hour (3600 seconds) apart, requesting a unitless output step of '3600' will produce output every second; '240' will output every 15 seconds (3600/15 = 240 intervals).

Unitless `STEP_SIZE` is the only way to obtain output with less than 1 minute spacing, while Horizons output steps must be greater than 0.5 seconds.

For example, given a two-minute span between start/stop (120 seconds), specifying a unitless step-size of 240 would produce output at 0.5 second intervals, the smallest output resolution. Only short periods of time can be covered in a single ephemeris when such fine output resolution is specified.

Rise-Transit-Set event stepping

When `EPHEM_TYPE= 'OBS'` and `CENTER` is a topocentric location, the optional {mode} setting can be used to toggle output at rise-transit-set events only (within integer <= 9 min resolution), where rise and set are relative to one of three reference planes while considering `ELEV_CUT` angle setting:

{mode}	Rise-Transit-Set aberrations relative to reference plane	Example
TVH	True visual horizon (include horizon dip and refraction)	`STEP_SIZE='1m TVH'` (RTS with +/- 1 minute resolution)
GEO	Geometric horizon (include refraction)	`STEP_SIZE='5m GEO'` (RTS with +/- 5 minute resolution)
RAD	Radar horizon (geometric horizon, NO refraction)	`STEP_SIZE='1m RAD'` (RTS with +/-1 minute resolution)

'Geometric horizon' refers to the horizon defined by the plane perpendicular to the local zenith (modified by `ELEV_CUT`).

Time-varying output (angular stepping)

This mode turns on time-varying output, requesting approximate angular motion criteria for output instead of time-based stepping. Angular range can be 60 to 3600 (arcsec).

{mode}	Description	Example
VAR	Output after specified angular change in plane-of-sky position	`STEP_SIZE='VAR 600'` (output after moving 600 arcseconds in plane-of-sky)

There is no {units} specification for angle-stepping (time-varying) mode, it is implicitly arcseconds.

`TLIST` Parameter

The Horizons `TLIST` parameter allows specification of up to 10,000 discrete output times, though URL length limits may impose smaller limits depending on local software. When using TLIST, do not use any of the other time-span parameters: `START_TIME`, `STOP_TIME`, or `STEP_SIZE`.

`TLIST` values may be specified as individually quote-wrapped Julian Day numbers (JD), Modified Julian Day numbers (MJD, where MJD = JD - 2400000.5, the number of days since 1858-Nov-17), or any calendar date form that Horizons recognizes, such as '2035-Jul-12 10:17:19.373'. See system documentation for more examples of supported calendar date forms.

For numeric TLIST values (JD or MJD input), the default behavior is to determine if the chronologically earliest value is JD or MJD based on its magnitude. A small value less than `abs(625360.5)` will be interpreted as toggling MJD input for the rest of the TLIST. If they are in fact ancient Julian Day Numbers (prior to around 3001 BC), set TLIST_TYPE= 'JD' to force interpretation of small numeric TLIST values as JD instead of MJD.

TLIST can contain a mix of JD and calendar dates, but MJD values cannot be included in the mix without resulting in all other dates being assumed to be MJD.

When more than one TLIST value needs to be specified, the list of individually quoted values is further delimited using either comma `(,)` or space ``` characters. An example specification for Horizons is

```
TLIST= '2455339.95748' '2455354.92142' '2033-Jan-17 12:10:25.1'
```

However, to successfully transmit such a list via URL to the API, the spaces (or commas) and single quotes in the TLIST setting should be URL-encoded using `%20` for spaces, `%2C` for commas, and `%27` for single-quotes. The resulting URL for successful API handling of the example would look like this:

```
TLIST=%272455339.95748%27%20%272455354.92142%27%20%272033-Jan-17%2012:10:25.1%27
```

If URL-length limits are encountered when specifying long TLISTS, the file-based Horizons API (/doc/horizons_file.html) provides an alternate input method that supports long lists without the need for URL character-encoding.

`VEC_TABLE` Parameter

The VEC_TABLE parameter is used to set table format details when `EPHEM_TYPE=VECTOR`. Values assigned can be a single integer from 1 to 6, with optionally present modifier symbols allowed for numerically integrated small-bodies

Quantities Output:

Value	Quantities Output
1	Position components {x,y,z} only (with optional statistical request codes)
2	State vector {x,y,z,Vx,Vy,Vz} (with optional statistical request codes)
3	State vector, 1-way light-time, range, and range-rate
4	Position, 1-way light-time, range, and range-rate
5	Velocity components {vx, vy, vz} only
6	1-way light-time, range, and range-rate

Modifier codes are accepted for values `1` and `2` to trigger output of formal statistical uncertainties when available (**for asteroids and comets only**).

Available modifier codes for values `1` and `2` as shown in the following table.

Code	Description
x	XYZ uncertainties (ICRF or FK4/B1950)
a	ACN uncertainties (along-track, cross-track, normal)
r	RTN uncertainties (radial, transverse, normal)
p	POS uncertainties (plane-of-sky; radial, RA, and DEC components)

Examples:

- `VEC_TABLE='1xa'` returns position components along with their uncertainties, and with uncertainties in the ACN system also.

- `VEC_TABLE='2xarp'` returns position and velocity, with uncertainties in all four coordinate systems.

TLEs (Two-Line Elements)

Two-Line Elements (TLEs) may be input to define an artificial Earth-orbiting satellite. They are used only when `COMMAND='TLE'` (input object is the target), or if `CENTER='@TLE'` also (meaning the input object is the coordinate center).

TLEs must be supplied in standard format with starting and ending quote marks enclosing the entire block. Name specification line(s) are optional. Up to 600 pairs (1200 data lines) can be specified, but this limit is subject to change.

Example assignment for object 'SC-1':

```
TLE = '
SC-1
1 87820U 11053A   11273.79990913  .00099611  00000-0  64461-3 0  9991
2 87820 042.7843 189.7738 0014383 039.8647 002.5266 15.74868665   196
1 87820U 11053A   11273.86983630 -.00085102 +00000-0 -55758-3 0  9998
2 87820 042.7804 189.3478 0014258 040.7498 038.5752 15.74826749000204'
```

> IMPORTANT: You must URI-encode such content because a new-line character cannot be directly input via URL. The above example would use `%0A` for the new-line (LF) character as shown below.

```
TLE='SC-1%0A1 87820U 11053A   11273.79990913  .00099611  00000-0  64461-3 0  9991%0A2 87820 042.7843 189.7738 0014383 039.8647
```

Although the above form may work on some systems with space characters contained within the quote characters (`'`), it is best practice to encode all space characters using `%20`. The corresponding content with space-encoding would look like the following.

```
TLE='SC-1%0A1%2087820U%2011053A%20%20%2011273.79990913%20%20.00099611%20%2000000-0%20%2064461-3%200%20%209991%0A2%2087820%2004
```

See the Horizons documentation (https://ssd.jpl.nasa.gov/horizons/manual.html#tles) for more details about TLEs.

Data Output

> Please **always check** the JSON payload "signature" object for the API "version". If the version does not match the version in this document (at the top), there is no guarantee that the format has not changed.
>
> Example "signature" object with "version" value "1.0": `"signature":{"version":"1.0","source":"NASA/JPL ... API"}`

Successful query requests result in a data payload. The format of such a data payload is controlled by the `format` parameter.

Example `text` -format Output

The first two lines are added by the API. It is important to note the `API VERSION` value when programmatically processing results to ensure the API version is what you expect. The number of blank lines following `API SOURCE` may vary but will be at least one. If you request the body information page (`OBJ_DATA='YES'`), there will be an additional block of text output after the blank lines following `API SOURCE`. The remainder of the output is formatted as provided by the Horizons system.

```
API VERSION: 1.0
API SOURCE: NASA/JPL Horizons API

*******************************************************************************
Ephemeris / WWW_USER Tue Sep  6 22:51:14 2022 Pasadena, USA        / Horizons
*******************************************************************************
Target body name: Mars (499)                      {source: mar097}
Center body name: Earth (399)                     {source: mar097}
Center-site name: GEOCENTRIC
*******************************************************************************
Start time      : A.D. 1998-Jan-01 10:00:00.0000 UT
Stop  time      : A.D. 1998-Jan-02 00:00:00.0000 UT
Step-size       : 60 minutes
*******************************************************************************
Target pole/equ : IAU_MARS                        {West-longitude positive}
Target radii    : 3396.19, 3396.19, 3376.2 km     {Equator_a, b, pole_c}
Center geodetic : 0.0, 0.0, -6378.137             {E-lon(deg),Lat(deg),Alt(km)}
Center cylindric: 0.0, 0.0, 0.0                   {E-lon(deg),Dxy(km),Dz(km)}
Center pole/equ : ITRF93                          {East-longitude positive}
Center radii    : 6378.137, 6378.137, 6356.752 km {Equator_a, b, pole_c}
Target primary  : Sun
Vis. interferer : MOON (R_eq= 1737.400) km        {source: mar097}
Rel. light bend : Sun, EARTH                      {source: mar097}
Rel. lght bnd GM: 1.3271E+11, 3.9860E+05 km^3/s^2
Atmos refraction: NO (AIRLESS)
RA format       : HMS
Time format     : CAL
Calendar mode   : Mixed Julian/Gregorian
EOP file        : eop.220904.p221128
EOP coverage    : DATA-BASED 1962-JAN-20 TO 2022-SEP-04. PREDICTS-> 2022-NOV-27
Units conversion: 1 au= 149597870.700 km, c= 299792.458 km/s, 1 day= 86400.0 s
Table cut-offs 1: Elevation (-90.0deg=NO ),Airmass (>38.000=NO), Daylight (NO )
Table cut-offs 2: Solar elongation (  0.0,180.0=NO ),Local Hour Angle( 0.0=NO )
Table cut-offs 3: RA/DEC angular rate (     0.0=NO )
*******************************************************************************
 Date__(UT)__HR:MN     R.A._____(ICRF)_____DEC    APmag   S-brt             delta      deldot
*******************************************************************************
```

```
$$SOE
 1998-Jan-01 10:00     20 55 41.20 -18 33 23.0    1.199   4.107  2.13799045474771   5.6049390
 1998-Jan-01 11:00     20 55 49.17 -18 32 49.7    1.205   4.112  2.13812533287512   5.6044117
 1998-Jan-01 12:00     20 55 57.13 -18 32 16.5    1.221   4.128  2.13826019831403   5.6038846
 1998-Jan-01 13:00     20 56 05.09 -18 31 43.1    1.234   4.141  2.13839505107206   5.6033579
 1998-Jan-01 14:00     20 56 13.05 -18 31 09.8    1.249   4.156  2.13852989115697   5.6028315
 1998-Jan-01 15:00     20 56 21.01 -18 30 36.5    1.229   4.136  2.13866471857652   5.6023055
 1998-Jan-01 16:00     20 56 28.97 -18 30 03.1    1.242   4.149  2.13879953333835   5.6017797
 1998-Jan-01 17:00     20 56 36.93 -18 29 29.7    1.236   4.143  2.13893433545018   5.6012543
 1998-Jan-01 18:00     20 56 44.89 -18 28 56.3    1.225   4.132  2.13906912492008   5.6007292
 1998-Jan-01 19:00     20 56 52.85 -18 28 22.9    1.233   4.139  2.13920390175678   5.6002045
 1998-Jan-01 20:00     20 57 00.81 -18 27 49.4    1.239   4.146  2.13933866596968   5.5996802
 1998-Jan-01 21:00     20 57 08.76 -18 27 15.9    1.246   4.153  2.13947341756857   5.5991564
 1998-Jan-01 22:00     20 57 16.72 -18 26 42.5    1.226   4.133  2.13960815656331   5.5986329
 1998-Jan-01 23:00     20 57 24.67 -18 26 08.9    1.216   4.123  2.13974288296371   5.5981098
 1998-Jan-02 00:00     20 57 32.63 -18 25 35.4    1.205   4.111  2.13987759677950   5.5975872
$$EOE
*******************************************************************************************
Column meaning:

... [some content excluded here for brevity] ...

 Computations by ...
     Solar System Dynamics Group, Horizons On-Line Ephemeris System
     4800 Oak Grove Drive, Jet Propulsion Laboratory
     Pasadena, CA  91109   USA
     Information: http://ssd.jpl.nasa.gov/
     Connect    : telnet://ssd.jpl.nasa.gov:6775  (via browser)
                  telnet ssd.jpl.nasa.gov 6775    (via command-line)
     Author     : Jon.D.Giorgini@jpl.nasa.gov

*******************************************************************************************
```

If there are any errors in your input file (such as requesting ephemeris output times outside the range available), they will be shown at the end of the normal output. Thus, such error detection is the responsibility of the user.

In some cases, error messages will provide suggestions that are only applicable to the command-line interface. For example, when specifying an observatory code that does not exist, the message suggests typing `..., ?! for help`. In such cases, if the error is not obvious, please check the Horizons system documentation (https://ssd.jpl.nasa.gov/horizons/manual.html).

Example `json`-format Output

When `json`-format is requested, the API attempts to detect any Horizons error message(s), capture them, and report them in the `error` field. Thus, if the `error` field is present in the output, an appropriate programmatic response can be issued.

In the following successful request, the content of interest will be in the JSON `result` field.


```
{
  "signature" : {
    "source" : "NASA/JPL Horizons API",
    "version" : "1.0"
  },
  "result" : " \n \n*******************************************************************************\nEphemeris / WWW_USER Tue
}
```


Non-unique Object Specification

If the object specified via the `COMMAND` query parameter is not unique to the Horizons system, you will get a list of matching objects. Below is an example (`text`-format) output where `COMMAND='DES=141P;'`.

```
API VERSION: 1.0
API SOURCE: NASA/JPL Horizons API

*******************************************************************************
JPL/DASTCOM            Small-body Index Search Results     2021-May-11 07:53:26

 Comet AND asteroid index search:

    DES = 141P;

 Matching small-bodies:

    Record #  Epoch-yr  >MATCH DESIG<  Primary Desig  Name
    --------  --------  -------------  -------------  -------------------------
    90001032    2011    141P           141P            Machholz 2
    90001033    2019    141P           141P            Machholz 2
    90001034    1994    141P-A         141P-A          Machholz 2
    90001035    1997    141P-A         141P-A          Machholz 2
    90001036    1999    141P-A         141P-A          Machholz 2
    90001037    1994    141P-D         141P-D          Machholz 2
    90001038    1999    141P-D         141P-D          Machholz 2

 (7 matches. To SELECT, enter record # (integer), followed by semi-colon.)
*******************************************************************************
```

In the above example, there are three bodies represented: the parent comet `141P`, fragment A `(141P-A)`, and fragment D `(141P-D)`. There are also multiple orbits for each of these comets represented by the `Epoch-yr` field in the results table. This is typically because the orbit may change between apparitions due to changed outgassing accelerations.

To select the object and orbit of interest, you must specify the Horizons record number listed in the `Record #` column. For example, to select the orbit for the parent comet with epoch 2019, specify `COMMAND='90001033;'`. However, be warned that for small bodies (comets and asteroids) these **record numbers are subject to change without notice**.

For unique comet selection without using Horizons record numbers, there are additional flags that can be used in the `COMMAND` specification. For example, if you know there are fragments for comet 141P and you want the parent comet, you can add the `NOFRAG` flag to the `COMMAND` specification such as `COMMAND='DES=141P;NOFRAG'`.

Similarly, if you know you always want the orbit with the most recent epoch, you can add the `CAP` flag (closest-apparition) to the `COMMAND` specification such as `COMMAND='DES=141P;CAP'`. In this example, combining `CAP` and `NOFRAG` will result in a unique match for the parent comet 141P as in `COMMAND='DES=141P;CAP;NOFRAG'`. For fragments, the designation is already unique so you'd only need the `CAP` flag as in `COMMAND='DES=141P-A;CAP'`.

See the Horizons documentation (https://ssd.jpl.nasa.gov/horizons/manual.html) for more details.

Small-Body SPK File Generation

The binary SPK file is base-64 encoded on return and, for `text`-format output, begins after the API header text with the base-64 string `REFGL1NQ...` in the following example for asteroid 1 Ceres (SPK-ID 2000001).

```
API VERSION: 1.0
API SOURCE: NASA/JPL Horizons API


ID= '2000001'

SPK Binary Data Follows -- base64 encoded:

REFGL1NQSyACAAAABgAAAFNNQl9TUEtfRklMRSAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICAgICAgICAgICAgID4AAAA+AAAAMLcAAExUTC1JRUVFAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
...
AAAAAAAAAAAAAAAAAAAAAAAA==
```

To convert the base-64 encoded content to a usable binary SPK file, extract the encoded portion to a file and use a base-64 decoder. On most Linux systems this is done with the command `base64 --decode {extracted_file_content}`, where `{extacted_file_content}` is the name of the file containing the encoded content from the API.

Example SPK Horizons Request

For Small-Body in the JPL Database

The following example requests an SPK file spanning 2020-01-01 through 2030-12-31 (TDB) for asteroids 1 Ceres.

```
https://ssd.jpl.nasa.gov/api/horizons.api?COMMAND='1%3B'&EPHEM_TYPE=SPK&START_TIME='2020-01-01'&STOP_TIME='2030-12-31'&OBJ_DAT
```

For Small-Body *not* in the JPL Database

The following example requests an SPK file covering 2030-01-01 through 2031-01-01 (TDB) for the small-body specified with user-supplied osculating elements and non-gravitational parameters.

```
https://ssd.jpl.nasa.gov/api/horizons.api?COMMAND='%3B'&MAKE_EPHEM=YES&EPHEM_TYPE=SPK&OBJ_DATA=NO&START_TIME='2030-Jan-1'&STOP
```

Command-line Examples

An example Linux command-line to extract the binary SPK file content from a successful `text`-format output is shown below. However, be advised that such usage will not easily catch errors returned from the API.

```
curl -s "https://ssd.jpl.nasa.gov/api/horizons.api?format=text&COMMAND='1%3B'&EPHEM_TYPE=SPK&START_TIME='2020-01-01'&STOP_TIME
```

Similarly, for the `json`-format output, the SPK encoded content is contained in the `spk:` object as in the following example:

```
{
  "signature" : {
    "source" : "NASA/JPL Horizons API",
    "version" : "0.2"
  },
  "result" : "\nID= '2000001'\n\n",
  "spk_file_id" : "2000001",
  "spk" : "REFGL1NQSyACAAAABgAAAFNNQl9TUEtfRklMRSAgICAgICAgICAgICAgICAgICAgICAgICAg\nICAgICAgICAgICAgICAgICAgID4AAAA+AAA .
}
```


Extract the content from the JSON `spk` object excluding the surrounding quotes and then decode to produce the corresponding binary SPK file. The JSON `spk_file_id` value suggests an appropriate filename (excluding the `.bsp` extension) for output: `2000001` in this example from which the filename `2000001.bsp` could be formed.

Sample Scripts

The following scripts generate SPK files spanning `2030-01-01` to `2031-01-01` for the small-body specified on the command-line. In this example, the desired small-body is specified using its SPK-ID (e.g., `2000001` for asteroid 1 Ceres).

> IMPORTANT: You must take care to encode special characters in the URL within your scripts for proper interpretation of your query parameters. In the scripts below, the Horizons `COMMAND` parameter contains special characters `=` and `;` which must be converted to `%3D` and `%3B`, respectively. For example, `COMMAND='DES=2000001;'` must be encoded as `COMMAND='DES%3D2000001%3B'`.

Perl

Assuming the following Perl script is saved to a file `horizons_spk.pl`, it could be invoked from the command-line to generate an SPK file for asteroid 1 Ceres (SPKID 2000001) as `perl horizons_spk.pl 2000001`.

```
use strict;
use LWP::UserAgent;
use JSON;
use MIME::Base64;

# Define API URL and SPK filename:
my $url = "https://ssd.jpl.nasa.gov/api/horizons.api";
my $spk_filename = 'spk_file.bsp';

# Define the time span:
my $start_time = '2030-01-01';
my $stop_time = '2031-01-01';

# Get the requested SPK-ID from the command-line:
my $spkid = shift;
unless ( defined $spkid ) { die "please specify SPK-ID on the command-line"; }

# Build the appropriate URL for this API request:
# IMPORTANT: You must encode the "=" as "%3D" and the ";" as "%3B" in the
#            Horizons COMMAND parameter specification.
$url .= "?format=json&EPHEM_TYPE=SPK&OBJ_DATA=NO";
$url .= "&COMMAND='DES%3D$spkid%3B'&START_TIME='$start_time'&STOP_TIME='$stop_time'";

# Build and submit the API request and decode the JSON-response:
my $ua = LWP::UserAgent->new;
my $response = $ua->get($url);
my $data = decode_json($response->content);

# If the request was valid...
if ( $response->code eq '200' ) {
  #
  # If the SPK file was generated, decode it and write it to the output file:
  if ( defined $data->{spk} ) {
    #
    # If a suggested SPK file basename was provided, use it:
    if ( defined $data->{spk_file_id} ) { $spk_filename = "$data->{spk_file_id}.bsp"; }
    open (my $fh, ">", $spk_filename) or die "unable to open $spk_filename: $!\n";
```

```
    #
    # Decode and write the binary SPK file content:
    print $fh decode_base64($data->{spk});
    close $fh;
    print "wrote SPK content to $spk_filename\n";
    exit;
  }
  #
  # Otherwise, the SPK file was not generated so output an error:
  print "ERROR: SPK file not generated\n";
  if ( defined $data->{result} ) { print $data->{result}; }
  else { print $response->content; }
  exit 1;
}

# If the request was invalid, extract error content and display it:
if ( $response->code eq '400' ) {
  print "ERROR in API call: HTTPS response: ", $response->status_line, "\n";
  if ( defined $data->{message} ) { print $data->{message}, "\n"; }
  else { print $response->content; }
  exit 1;
}

# Otherwise, some other error occurred:
print "ERROR in API call: HTTPS response: ", $response->status_line, "\n";
exit 1;
```

Python

Assuming the following Python script is saved to a file `horizons_spk.py`, it could be invoked from the command-line to generate an SPK file for asteroid 1 Ceres (SPKID 2000001) as `python horizons_spk.py 2000001`.

```
import sys
import json
import base64
import requests

# Define API URL and SPK filename:
url = 'https://ssd.jpl.nasa.gov/api/horizons.api'
spk_filename = 'spk_file.bsp'

# Define the time span:
start_time = '2030-01-01'
stop_time = '2031-01-01'

# Get the requested SPK-ID from the command-line:
if (len(sys.argv)) == 1:
  print("please specify SPK-ID on the command-line");
  sys.exit(2)
spkid = sys.argv[1]

# Build the appropriate URL for this API request:
# IMPORTANT: You must encode the "=" as "%3D" and the ";" as "%3B" in the
#            Horizons COMMAND parameter specification.
url += "?format=json&EPHEM_TYPE=SPK&OBJ_DATA=NO"
url += "&COMMAND='DES%3D{}%3B'&START_TIME='{}'&STOP_TIME='{}'".format(spkid, start_time, stop_time)

# Submit the API request and decode the JSON-response:
response = requests.get(url)
try:
  data = json.loads(response.text)
except ValueError:
  print("Unable to decode JSON results")

# If the request was valid...
if (response.status_code == 200):
  #
  # If the SPK file was generated, decode it and write it to the output file:
  if "spk" in data:
```

```
    #
    # If a suggested SPK file basename was provided, use it:
    if "spk_file_id" in data:
      spk_filename = data["spk_file_id"] + ".bsp"
    try:
      f = open(spk_filename, "wb")
    except OSError as err:
      print("Unable to open SPK file '{0}': {1}".format(spk_filename, err))
    #
    # Decode and write the binary SPK file content:
    f.write(base64.b64decode(data["spk"]))
    f.close()
    print("wrote SPK content to {0}".format(spk_filename))
    sys.exit()
  #
  # Otherwise, the SPK file was not generated so output an error:
  print("ERROR: SPK file not generated")
  if "result" in data:
    print(data["result"])
  else:
    print(response.text)
  sys.exit(1)

# If the request was invalid, extract error content and display it:
if (response.status_code == 400):
  data = json.loads(response.text)
  if "message" in data:
    print("MESSAGE: {}".format(data["message"]))
  else:
    print(json.dumps(data, indent=2))

# Otherwise, some other error occurred:
print("response code: {0}".format(response.status_code))
sys.exit(2)
```

URL Encoding

If a query parameter *value* contains any of the following characters, those characters should be URI-encoded before submitting your request. The following table shows these reserved characters and their corresponding encoding. For example, a query string value "My Value" should be encoded as `My%20Value` where the space character is replaced with `%20`.

Char.	Encoding	Name
LF	%0A	line feed (new-line)
``	%20	space
`#`	%23	hash tag
`$`	%24	dollar sign
`&`	%26	ampersand
`+`	%2B	plus sign
`,`	%2C	comma
`/`	%2F	slash
`:`	%3A	colon
`;`	%3B	semicolon
`=`	%3D	equals sign
`?`	%3F	question mark
`@`	%40	at symbol
`[`	%5B	left square bracket
`]`	%5D	right square bracket

More information is available in section 2.3 of RFC3975 (https://datatracker.ietf.org/doc/html/rfc3875#section-2.3).

HTTP Response Codes

Most errors are returned via appropriate HTTP response codes. However, it is possible to submit Horizons input data (e.g., a bad `START_TIME` value) resulting in Horizons-generated errors. In such cases, a non-error code of 200 is returned so the user is responsible for checking the payload if they wish to detect such errors.

HTTP Code	Description	Typical Usage
200	OK	normal successful result
400	Bad Request	the request contained invalid keywords and/or content or used a request-method other than GET or POST (details returned in the JSON or text payload)
405	Method Not Allowed	the request used an incorrect method (see the HTTP Request section)
500	Internal Server Error	the database is not available at the time of the request
503	Service Unavailable	the server is currently unable to handle the request due to a temporary overloading or maintenance of the server, which will likely be alleviated after some delay

Change Log

Version 1.2 (2022 September)

- Update to list new `CAL_TYPE` setting

Version 1.1 (2021 December)

- Update to mention `TLIST` calendar date option, new `TLIST_TYPE` setting, and elaborate on `STEP_SIZE` options

Version 1.0 (2021 July)

- Initial release

Privacy (http://www.jpl.nasa.gov/copyrights.php)
Image Policy (http://www.jpl.nasa.gov/imagepolicy)

Site Manager : Ryan Park
Webmaster : Alan Chamberlin
URS Clearance : CL#16-5242
Contact : contact-ssd-api@jpl.nasa.gov