StratoScript™ 23.6 Reference

July 18, 2023

StratoScript is a simple yet powerful scripting language developed by Digitalis Education Solutions, Inc. for automating select planetarium simulation software. Anyone with some astronomy knowledge can be successful with StratoScript scripting. Use it to alleviate tedious manual sequences or provide special effects with image manipulation, audio, video playback (where supported), and more.

This document outlines StratoScript commands supported by: **Nightshade G3 23.6.x**

StratoScript features have been enhanced with Nightshade G3 through the addition of new commands, new arguments, and the much greater capabilities of G3. Scripts designed for the older versions of Nightshade should mostly work with G3, except that a few features are not supported. See the notes below on a blue background and this summary page which is kept up to date:

http://NightshadeSoftware.org/projects/nightshade/wiki/StratoScript_in_NG

The Basics (Don't Skip This!)

StratoScript files end in a ".sts" extension and should be formatted in UTF-8 encoding (ASCII is acceptable as it is a subset of UTF-8).

Command format:

- Each command is a single line (terminated with a newline character). If you need a newline in an argument value you must use "\n".
- Each command consists of a command name followed by an optional list of arguments in the form of name/value pairs:

```
COMMAND ARG_NAME1 ARG_VALUE1 ARG_NAME2 ARG_VALUE2 ...
```

- Whitespace is used as a delimiter.
- Argument values requiring included whitespace must be enclosed in double quotes. If you want a double quote within a value you can precede it with a backslash.

```
"This is a value with spaces."
"What is a \"blue moon?\""
```

- Commands and argument names are not case sensitive.
- Argument values are case sensitive.
- Argument pairs can be arranged in any order.
- When multiple arguments are required to perform a command, they must all be supplied at the same time (in the same single-line command).
- Comments start with a '#' character. Anything to the right of and including a '#' is ignored unless the '#' is preceded by a backslash.

```
# My script about the Sun
select object Earth # this is a comment and will be ignored
text name title action load string "Lesson \#1" alpha 1
```

- If an argument is not defined it will be ignored if possible. Otherwise it will default to zero if a number is required, or an empty string otherwise. Any different default values are documented below in the command argument descriptions.

Example Commands:

```
select object Jupiter
flag atmosphere on
date utc 1999-08-11T12:00:00
wait duration 2.5
moveto lat 45.7 lon -122 duration 5
landscape action load type spherical texture egarden.png
select nebula M31 pointer off
set home_planet "Solar System Observer"
```

Best practice is to use the “require” command as the first command in your script as this alerts Nightshade and users to the requirements for proper playback. For best results you should develop scripts using the same software version and projection mode intended for playback.

Format of this document:

command1

A description of what the command does will follow with some examples of using the command in an example box.

```
command1 argument-1 4
command1 argument-2 “literal-value” argument-3 cat
```

argument-1

`INTEGER`

A description of what the argument value means follows each argument option. If an argument value is all caps on a black background like `INTEGER` it is a standard argument type, and should be substituted with an appropriate value of this type. Consult the “Standard Argument Types” below for definitions of standard types.

argument-2

`literal-value`

Argument values that are literal strings are lower case and are used exactly as listed or can be included in quotes.

Examples (note the box used for examples in this document):

```
literal-value
"literal-value"
```

`STRING`

A standard argument value could be another option for argument-2.

In this example, the argument "argument-2" can have any string value desired, but "literal-value" would have a special meaning, which would have been described here.

argument-3

`cat`

`dog`

This example only allows two values, either “dog” or “cat” for argument-3.

Notes about current or future changes have this blue background.

argument-4

`[INTEGER:]INTEGER`

Optional items are in brackets. Here the first integer and colon group is optional before the required last integer. Examples:

```
5
-5
6:3
207:-3
```

Standard Argument Types

Many arguments take standard types of arguments. These are standardized and defined in this section. If an argument value is ALL CAPS on a black background refer back to this section for its definition. The most fundamental types are listed first, followed by those that build upon the former, with each section in alphabetical order.

Fundamental Argument Types

DATE_TIME

A date and/or time specified in ISO8601 format: year-month-dayThour:minute:second

In StratoScript the date or time portion can be left off, in which case the 'T' separator is not required.

```
-200-03-22T13:33:00
59900-12-31T01:00:00
1918-11-11
20:30:0
```

DECLINATION

An angle in degree, minute, and second notation. The following examples are all equivalent.

```
-9d4m3.2s
-9:4:3.2
“-9 4 3.2”
```

DIRECTORY

A case sensitive directory (folder) relative to directory the running script is located in.

```
MyMedia
"jupiter directory"
moons/irregular/Mars
```

FILENAME

A valid filename which is case sensitive. If you need to specify a path to the file using folders, separate these with forward slashes (/). A path by default is relative to the directory the script is running from. If you want to reference a file higher up the filesystem tree, use “../” notation. Note that the required file type(s) vary by command.

Example filesystem structure:

```
show1/myscript.sts
show1/hamster.jpg
show1/photos/fido.png
media/cat.jpg
media/Dog.png
media/birds/parrot.mp4
```

Example command filename arguments usable in show1/myscript.sts with above filesystem:

```
hamster.jpg
photos/fido.png
../media/cat.jpg
../media/Dog.png
../media/birds/parrot.mp4
```

INTEGER

A whole number. Specific ranges are limited for some arguments, but the internal implementation is a C++ 'integer' with a range of at least –2,147,483,648 to 2,147,483,647.

```
5
-123
0
```

LOCALE

Language locale code to use for translations. Not every locale is supported or has full translations.

```
fr # French
es # Spanish
zh_HK # Chinese, Hong Kong
```

ON_OFF

A value of "on" or "1" is on. Off is "off" or "0".

ON_OFF_TOGGLE

A value of "on" or "1" is on. Off is "off" or "0". "toggle" means turn off if currently on, or vice versa.

PATH

A path to a directory or folder containing files in the file system. The path should be relative to the calling script and use Unix style “/” directory separators. See the `FILENAME` description for more details.

REAL

A real number. Specific ranges are limited for some arguments, but the internal implementation is a C++ 'double' with a range of at least 1.7E +/- 308 (15 digits).

```
1900
2.345
-0.0004
```

RIGHT_ASCENSION

An angle in hour, minute, and second notation. The following examples are all equivalent.

```
12h5m4.5s
12:05:4.5
“12 5 4.5”
```

STRING

A string of characters. If the string contains spaces it needs to be quoted. It is best to assume that strings are case sensitive (capitalization has to match). Default is an empty string.

```
"a string with spaces"
```

TIME_ZONE

Time zone name. See https://en.wikipedia.org/wiki/List_of_tz_database_time_zones for a list of generally usable time zone names. Actual supported time zones can vary with your operating system.

TRUE_FALSE

Value can be “true” or 1 to signify truth, and “false” or 0 for not true.

`URL`

A `FILENAME` or Universal Resource Locator for a supported network resource.

```
jupiter.png
../photos/mars.jpg
http://digitaliseducation.com/images/back/home/mauna-kea.jpg
```

Other Standard Argument Types

`COORDINATE_SYSTEM`

A number of useful coordinate systems are defined for positioning media, such as images, on the celestial sphere. The following are the supported coordinate system names and descriptions.

- **dome** – Altitude is the angular distance from the horizon, positive above and negative below the horizon. Azimuth is the angular distance from the direction opposite the focal point azimuth direction in a counter-clockwise direction if you are facing the zenith of your dome. Dome coordinates are only affected by your focal point and do not move with heading, pitch, etc.
- **equatorial** – Right ascension (**ra**) and declination (**dec**) in equatorial coordinates based on the equinox of date.
- **galactic** – Right ascension (**ra**) and declination (**dec**) in galactic coordinates.
- **geocentric** – Latitude (**lat**) and longitude (**lon**) of the currently anchored body.
- **horizontal** – Altitude is the angular distance from the horizon, positive above and negative below the horizon. Azimuth is the angular distance from North along the horizon, positive in the North to East rotational direction (counter-clockwise if you are facing the local zenith). Use this coordinate system for spherical/360 content so that you can pan around (set your media scale to 360 degrees).
- **j2000** – Right ascension (**ra**) and declination (**dec**) in J2000 equatorial coordinates.
- **viewport** – A 2D coordinate system overlayed on the screen with x (**xpos**) and y (**ypos**) coordinates. The center of the viewport is (0,0). X is positive to the right, Y is positive up. Note: In fisheye mode the viewport is a square just containing a non-truncated fisheye projection circle. In NG the viewport is rotated so that the bottom of the viewport (down direction in the image) is aligned to the focal azimuth direction. Use this coordinate system for “fulldome” shows and similar fisheye content (set your media scale to “fill”).

The following table lists the recommended position argument names for each coordinate system. Note that **xpos** and **ypos** are also supported for all systems but are not intuitive. All position arguments are `REAL` values, using degrees for any angles.

coordinate_system	Argument A (xpos)	Argument B (ypos)
dome	altitude	azimuth
equatorial	dec	ra
galactic	dec	ra
geocentric	lat	lon
horizontal	altitude	azimuth
j2000	dec	ra
viewport	xpos	ypos

DEGREES

Angular degrees as a REAL. Note that these are true decimal degrees; do not confuse with other angle notations like RIGHT_ASCENSION or DECLINATION.

```
35
-540.23
```

FADER

A REAL number between 0 and 1, inclusive. Used to adjust a value between off (0) and fully on (1).

DISTANCE

A REAL combined with an optional supported unit of length (with no white space in between). Units are sometimes case sensitive (Mm vs. mm, for example). Default units are meters. Supported units are:

- nm – nanometer
- mm – millimeter
- cm – centimeter
- m – meter
- km – kilometer
- Mm – Megameter
- AU – Astronomical Unit
- ly – light-year
- pc – parsec
- kly – kilolight-year
- kpc – kiloparsec
- Mly – megalight-year
- Mpc – megaparsec
- % - percentage of current distance
-

```
1au
10km
50%
30000
```

JULIAN_DATE

A **REAL** denoting an astronomical date in Julian days.

MARKUP_STRING

A **STRING** that allows newlines (as "\n" escape sequences) and also supports some formatting markup tags. The following markup tags can be used. Note that tags have a beginning and an ending tag as shown:

<b>bold</b>

<i>italics</i>

<sup>superscript</sup>

<sub>subscript</sub>

```
You\nAre\nHere.
H<sub>2</sub>O
E = MC<sup>2</sup>
This is my <b>favorite</b> star.
```

POSITIVE_REAL

A **REAL**real number greater than or equal to zero.

```
0.0
2.89
```

RELATIVE_SECONDS

A **REAL** preceded by a ‘+’ or ‘-’ and denoting a relative change in seconds (either real time or simulated time).

```
+8.34
-0.2
```

RGB

A color defined by three **FADER** values for red, green, and blue values in that order. Remember to include quotes if you put spaces between the numbers.

```
1,0.5,0
"0.33, 0.2, 0.9"
```

RGBA

A color defined by four **FADER** values for red, green, blue, and alpha values in that order. Remember to include quotes if you put spaces between the numbers. An alpha value of 0 is fully transparent, and 1 is fully opaque.

```
1,0.5,0,1
"0.33, 0.2, 0.9, 0.3"
```

RGBW

A color defined by four **FADER** values for red, green, blue, and white values in that order. Remember to include quotes if you put spaces between the numbers. The white value allows the display of pastel colors on RGBW cove lighting systems.

SECONDS

A **POSITIVE_REAL** denoting seconds of time (either real time or simulated time).

`VECTOR3`

A vector defined by three `REAL` values for x, y, and z in that order, separated by commas. Remember to include quotes if you use any spaces.

```
1,0.5,0
"-0.33, 5.2, 0.9"
```

Cascading Settings and Attributes

Configuration settings are in the process of being migrated to a new cascading architecture. Anything that can be configured will be organized into a single hierarchical taxonomy. This includes everything from celestial objects to user interface components.

Things that can be configured reside at specific locations in the taxonomy. A specific locations is called a `TAXON`. Note that some objects can be accessed via multiple taxons due to groupings (e.g. Pluto is both a dwarf planet and a trans Neptunian object).

An attribute that can be changed is called a `TAXON_ATTRIBUTE`. These attributes cascade down the hierarchy to make it easy to change entire classes of objects at once. If an attribute is set at a higher, more general level of the taxonomy that value will cascade down to lower levels unless already set at a lower level (to the right in the taxonomy below). For example, if you set `planetary.orbit_color` to blue but `planetary.dwarf_planets.orbit_color` to red then all bodies would have blue orbits except for dwarf planets.

Attributes set at the same level but later in the list (lower down, below) will override. Continuing the above example, if you then set `planetary.trans_neptunian_objects.orbit_color` to green, then TNO orbits would turn green, including those that are also dwarf planets like Pluto. If you then wanted just the orbit of Pluto to be purple you could set `planetary.dwarf_planets.pluto.orbit_color` (or `planetary.trans_neptunian_objects.pluto.orbit_color`) to red.

Below is the basic structure of the taxonomy. To print out your loaded taxonomy, see the **print** argument to the **set** command.

```
planetary
     planets
          terrestrial
          gas_giants
          ice_giants
     dwarf_planets
     asteroids
     satellites
          natural
          artificial
     comets
     trans_neptunian_objects
     systems
          emb
          mars
```

```
    stellar
          stars
```

TAXON

A taxon identifies a specific group or individual item in the Nightshade G3 configuration hierarchy. This hierarchy includes attributes for rendered objects, and even general application settings. A taxon is expressed in dotted notation, meaning there is a dot between each child taxon as you go down the hierarchy. Taxons are case-insensitive and use underscores instead of spaces. The “attribute.” prefix is assumed and not required. The taxonomy is dynamic based on what you have loaded, but the basic structure is:

```
planetary.planets
planetary.dwarf_planets.pluto
stellar
```

TAXON_ATTRIBUTE

A settable variable that lives at a specific position in the configuration hierarchy. This consists of a `TAXON` followed by a ‘.’ and then the variable name. Individual attributes accept specific value types (e.g. `RGBA`, `FADER`, `REAL`).

```
planetary.dwarf_planets.pluto.orbit_color
stellar.scale
```

Command Index

audio

Play and control audio tracks. Note that an audio track is stopped when the script that started it is stopped or finishes.

```
audio action play filename test.ogg name introduction
wait duration 20
```

action

`clear`

Drop all audio tracks loaded by a script.

`drop`

Halts playback and drops the track from memory. To resume an audio track after pausing it where you left off, you can call action "play" again, but be sure not to specify a filename or it will treat it as a new track.

`pause`

Pause currently playing script audio track.

`play`

Resume playing the existing script audio track if currently paused or begin playing a new track if a 'filename' argument is defined.

`resume`

Resume playing a paused audio track.

`sync`

Deprecated.

filename

`FILENAME`

Used with "play" action. Format support depends on your binary. Ogg Vorbis format is recommended for best compressed quality and backward compatibility. WAV format is also supported.

loop

`ON_OFF`

Used with "play" action.

name

`STRING`

Unique name to identify the audio track. Required if you want more than one track.

seek

`SECONDS`

`RELATIVE_SECONDS`

Jump to a new position in the audio file. The brackets mean that this value can begin with an optional '+' or '-'. If it does the seek will be made relative to the current position. Otherwise the position is measured from the beginning of the track. Can be used with the 'play' action to start somewhere besides the beginning of the track.

volume

Better results will be obtained by adjusting the volume in the actual audio track itself with an audio editor like the free Audacity project (http://audacity.sourceforge.net/).

`decrement`

Reduce audio volume by a step of 10% of maximum volume.

`increment`

Increase audio volume by a step of 10% of maximum volume.

`FADER`

0 is muted, 1 is maximum volume.

body

Load a solar system body such as an asteroid, comet, or artificial satellite. See the separate document “Adding a new celestial body in Nightshade” for a guide to orbital parameters and where to find them.

albedo

`FADER`

Reflectance of the body. 1.0 = White, 0.0 = Black.

action

`clear`

To remove all script added bodies at once use the "clear" action. This will not perform any action if your anchored body would be dropped.

`drop`

Remove an added solar system body with the "drop" action and "name" parameter. Only bodies loaded from scripts with no currently loaded satellites and that are not the current anchored body can be dropped.

`load`

Add a new solar system body defined by further arguments.

batch

`begin`

When loading a collection of new bodies, this tells Nightshade to begin processing a set of body commands in the most efficient way possible. Use this argument on the first command in the set.

end

Finish processing a set of body commands and make them visible. Use this argument on the last command in the set.

close_orbit

`TRUE_FALSE`

Orbit visualizations may not look very jagged or misleading with, for example, hyperbolic orbits due to the large orbit and small sampling size of the orbit line. A false argument value will prevent the connecting line between the start and end of the visualization period from being connected with a line segment.

color

`RGB`

General color of the body (sets the halo color).

comet_absolute_magnitude

`REAL`

comet_magnitude_slope

`REAL`

coord_func

`ell_orbit`

Elliptical type orbit.

`comet_orbit`

Comet type orbit.

halo

`TRUE_FALSE`

Is it bright enough to have a halo?

lighting

`TRUE_FALSE`

Is the body lit by the Sun?

lighting_model

phong

`oren-nayar`

What lighting model to use for a body. For a spherical body, the default is 'oren-nayar' which is realistic for dusty bodies. The 'phong' option is the default for other models and is realistic for artificial bodies like spacecraft.

model

`spherical`

`FILENAME`

Load a 3D model of the body. The model will be scaled to fit withing a sphere sized to the **radius** argument. The default is a 'spherical' body, which will be scaled for the specified **oblateness**. Otherwise provide a filename for a 3D model if supported.

model_front

`VECTOR3`

If you are loading a model such as an artificial satellite that needs to stay aligned to its orbit, this vector defines which direction is toward the direction of forward motion. This will override other rotational parameters other than model_gravity. If you load your model in osgviewer (part of the OpenSceneGraph project) it starts up with x positive to the right, y positive into the screen, and z positive up. If the front of the model is to the right in osgviewer, the model_front vector would be “1,0,0”.

model_gravity

`VECTOR3`

If you are loading a model such as an artificial satellite that needs to stay aligned to the parent body, this vector defines which direction faces down towards the parent in model coordinates. This will override other rotational parameters other than model_front.

name

`STRING`

Required. The body will not be added if there is already a body with the same name.

name_color

`RGB`

Color of the body label.

oblateness

`REAL`

How 'squashed' is the body? A perfect sphere has an oblateness of zero.

orbit_MeanMotion

`REAL`

orbit_ArgOfPericenter

`DEGREES`

Used to define elliptical or comet orbits.

orbit_AscendingNode

`DEGREES`

orbit_color

`RGB`

Color of the orbit visualization line.

orbit_eccentricity

`REAL`

A circle has ratio of 0.0.

orbit_epoch

`REAL`

A Julian day used to define elliptical orbits.

orbit_inclination

`DEGREES`

orbit_LongOfPericenter

`DEGREES`

Used to define elliptical orbits.

orbit_MeanAnomaly

`DEGREES`

Used to define elliptical orbits.

orbit_MeanLongitude

`DEGREES`

Used to define elliptical orbits.

orbit_PericenterDistance

`REAL`

A value with units of AU, used to define comet orbits.

orbit_period

`REAL`

Value in days used to define elliptical orbits.

orbit_SemiMajorAxis

`REAL`

Value in kilometers used to define elliptical orbits.

orbit_TimeAtPericenter

`JULIAN_DATE`

Used to define comet orbits.

orbit_visualization_period

`REAL`

How many days to use when drawing the orbit line for the body. The visualization is roughly centered on the body at it's current position.

parent

`STRING`

Parent body name in English. Required. Case sensitive.

radius

`REAL`

Radius of the body in kilometers.

rings

`TRUE_FALSE`

Does this body have a ring structure around it?

ring_inner_size

`REAL`

Inner radius of the rings, if any, in kilometers.

ring_outer_size

`REAL`

Outside radius of the rings, if any, in kilometers.

rot_epoch

`JDAY`

Epoch for the rotational parameters. Defaults to J2000.

rot_equator_ascending_node

`DEGREES`

rot_obliquity

`DEGREES`

rot_period

`REAL`

Body rotation period (not orbit period) in hours.

rot_periode

Synonym for "rot_period" which is preferred.

rot_pole_ra

`DEGREES`

`RIGHT_ASCENSION`

North pole right ascension at epoch.

rot_pole_de

Synonym for "rot_pole_dec" which is preferred.

rot_pole_dec

`DECLINATION`

`DEGREES`

North pole declination at epoch.

rot_precession_rate

`REAL`

Precession rate in degrees per 100 years.

rot_rotation_offset

`DEGREES`

Offset of the prime meridian.

tex_map

`FILENAME`

Surface texture image.

tex_halo

`FILENAME`

Halo texture image.

tex_ring

`FILENAME`

Ring texture image. This is a cross section of the ring.

clear

A shortcut to turn off lines and labels easily. If state is natural, ground and atmosphere will be turned on, otherwise these will be turned off.

```
clear state natural
```

state

`natural`

Turn off all labels, lines, and art. Turn planet, star, and nebula rendering on. Deselect any selected objects. Return to initial fov and viewing direction.

color

Specify the color of a drawn element. Note that the **color** command takes an argument also called color. Example which sets body orbits to bright yellow:

```
color property planet_orbits color 1,1,0
```

> The r (red), g (green), and b (blue) arguments are deprecated but still work. Use the color argument instead.

alpha

`FADER`

Opacity value. Defaults to 1 (opaque). Although all properties will accept an alpha value, not all properties will render with transparency. If you specify an alpha value in the **color** argument that will override any value for this **alpha** argument.

color

`RGB`

`RGBA`

If alpha is specified it overrides any separate **alpha** argument value.

property

`azimuthal_grid`

`cardinal_points`

`circumpolar_circle`

`constellation_art`

`constellation_boundaries`

`constellation_lines`

`constellation_names`

`ecliptic_line`

`equator_grid`

> Synonym for **equatorial_grid** which is preferred.

`equatorial_grid`

`equator_line`

`galactic_grid`

`galaxy_points`

`j2000_grid`

`meridian_line`

`navigation_px_line`

`navigation_zx_line`

`nebula_circle`

`nebula_names`

`object_trails`

Deprecated feature.

`planet_names`

`planet_orbits`

`precession_circle`

`satellite_orbits`

`tropic_lines`

Color for orbits of bodies which do not orbit the Sun.

configuration

```
configuration action save name elvis-mars
configuration action load name factory
configuration action load
```

action

`load`

Reload the user's default settings.

`save`

Save your current settings.

inline

`ENCODED STRING`

This is a compressed, non-human readable, ultra compact form of configuration data. (It is a base-64 encoded gzipped JSON string). Note that as of 2023 this mostly contains attribute settings.

name

`default`

This is the default value if not specified. Your default settings are loaded at startup and saved when you save your settings.

`Factory`

Your factory settings are failsafe settings from the factory.

`STRING`

You can also save settings by name and then load these later. Your name must only include alphanumeric or dash characters.

cove_lights

Control certain dome cove light systems. Only supported on Digitarium planetarium systems. The lights will change over the time specified by the **duration** argument to the color specified by the **color** argument. Alternately, use a preset setting.

```
cove_lights color 1,0,1,1 duration 15 # fade to pastel magenta
wait duration 15
```

The r (red), g (green), and b (blue), and w (white) arguments are deprecated but still work.

color

`RGB`

`RGBW`

The white value only has an effect on RGBW cove light systems.

`[CHANNEL_GROUP:]CHANNEL_NAME`

`FADER`

For Digitalis Fluid™ cove lights you can specify the name of the channel you want to control. Channel names are usually “intensity”, “red”, “green”, “blue”, and optionally “white” but can include other names depending on your system configuration. If you have multiple channel groups, prefix the channel name with its group followed by a colon. The default group is the first one in your system. Only channels that are specified will be changed.

```
cove_lights red 0.5 green 0.5 intensity 1 duration 9
cove_lights cove:green 1 cove:intensity 1 reading:red 1 duration 5
```

duration

`SECONDS`

Number of seconds to transition to a new color. If absent, the duration defaults to zero.

preset

`STRING`

The name of the desired preset to load. The **duration** argument will not affect preset loading.

date

Change the simulation date and/or time. To change the rate of time, see the **timerate** command.

```
date load current
date local 23:59:00 duration 20
wait duration 20
date sidereal 7
wait duration 1
date sidereal 7
date load current relative 7 # Go to one week from now
```

duration

`SECONDS`

Number of real world seconds to transition to a new date given by other arguments. Default is zero for an immediate change.

jday

`JULIAN_DATE`

Set date to current Julian date.

load

`current`

Set date to current (real world) date.

`preset`

Set date to preset start up date.

local

`DATE_TIME`

Set time to a specified date and/or time using the current timezone. When no date is specified, the current simulation date is used. When no time is provided, the current simulation time is used.

relative

`REAL`

Change date and time by number of days (can be fractional). This change is relative to either the current date/time or one specified in the same command.

sidereal

`REAL`

Change date and time by number of sidereal days (can be fractional) based on the planet or moon you are on. This change is relative to either the current date/time or one specified in the same command.

utc

`DATE_TIME`

Set time to a specified date and time in the UTC timezone. When no date is specified, the current simulation date is used. When no time is provided, the current simulation time is used.

deselect

With no arguments, deselects current object selection, including any constellation selection. See select command.

```
deselect constellation ORI
```

constellation

`CONSTELLATION_SHORT_NAME`

With a 3 character constellation abbreviation specified, will only deselect that constellation. See: http://nightshadesoftware.org/projects/nightshade/wiki/Constellation_Abbreviations

external_viewer

Will still work, but deprecated. See 'video' command.

flag

Flags are simple on/off type settings.

```
flag cardinal_points on
flag atmosphere 0
```

atmosphere

`ON_OFF_TOGGLE`

Draw atmospheric effects.

azimuthal_grid

`ON_OFF_TOGGLE`

Draw azimuthal grid.

cardinal_points

`ON_OFF_TOGGLE`

Draw cardinal points.

clouds

`ON_OFF_TOGGLE`

Draw clouds when rendering planets such as Earth.

circumpolar_circle

`ON_OFF_TOGGLE`

Draw circumpolar circle showing limit of sky visibility.

constellation_art

`ON_OFF_TOGGLE`

Draw constellation artwork.

constellation_boundaries

`ON_OFF_TOGGLE`

Draw contellation boundaries.

constellation_drawing

See synonym 'constellation_lines'

constellation_lines

`ON_OFF_TOGGLE`

Draw constellation line drawings.

constellation_names

`ON_OFF_TOGGLE`

Draw constellation labels.

constellation_pick

`ON_OFF_TOGGLE`

Select constellation pick mode (whether to only draw selected constellations).

ecliptic_line

`ON_OFF_TOGGLE`

Draw ecliptic line.

equator_line

`ON_OFF_TOGGLE`

Draw equator line.

equatorial_grid

`ON_OFF_TOGGLE`

Draw equatorial grid.

force_land_heading

`ON_OFF_TOGGLE`

If on, you will always land with your default heading value. If off, you will preserve your current heading when you land on an object.

galactic_grid

`ON_OFF_TOGGLE`

Draw galactic grid.

galaxy_points

`ON_OFF_TOGGLE`

Draw SDSS galactic point data.

j2000_grid

`ON_OFF_TOGGLE`

Draw J2000 grid.

landscape

`ON_OFF_TOGGLE`

Draw the landscape.

light_pollution

`ON_OFF_TOGGLE`

Whether light pollution affects the night sky within the atmosphere.

light_travel_time

`ON_OFF_TOGGLE`

Whether to correct for light travel time when rendering planets and moons. For performance reasons this is a close approximation.

manual_zoom

`ON_OFF_TOGGLE`

Select manual zoom mode.

media_captions

`ON_OFF_TOGGLE`

Draw text captions during audio or video playback if available.

meridian_line

`ON_OFF_TOGGLE`

Draw the meridian line.

milky_way

`ON_OFF_TOGGLE`

Draw the sky imagery layers. See the **layer** command **Sky** reference_frame.

Deprecated. See **sky** flag.

moon_scaled

`ON_OFF_TOGGLE`

Draw the moon scaled. See also **set moon_scale** to set the size.

navigation_px_line

`ON_OFF_TOGGLE`

Draw navigation triangle line through pole and selected object.

navigation_zx_line

`ON_OFF_TOGGLE`

Draw navigation triangle line through zenith and selected object.

nebula_names

`ON_OFF_TOGGLE`

Draw nebula labels.

object_trails

Deprecated. See **time_lapse** flag.

planets

`ON_OFF_TOGGLE`

Draw planets, moons, and related items like orbit lines and name labels. The bodies still exist, they just are not drawn.

planet_names

`ON_OFF_TOGGLE`

Draw planet labels.

planet_orbits

`ON_OFF_TOGGLE`

Draw planet orbits.

precession_circle

`ON_OFF_TOGGLE`

Draw Earth precession circle.

record_manual_movement

`ON_OFF_TOGGLE`

Whether to record gamepad type motion while recording a script. Turn off if you only want to use keyframes, for example.

script_gui_debug

`ON_OFF_TOGGLE`

If on, will print script errors to the screen. Good for debugging.

shadow_volumes

`ON_OFF_TOGGLE`

Show volumes to visualize body shadows.

show_framerate

`ON_OFF_TOGGLE`

Shows the current rendering framerate next to the date for debugging purposes.

show_tui_datetime

`ON_OFF_TOGGLE`

Draw the date and time.

show_tui_short_obj_info

`ON_OFF_TOGGLE`

Draw information about the selected object.

sky

`ON_OFF_TOGGLE`

Draw the sky imagery layers.   See the **layer** command **Sky** reference_frame.

star_names

`ON_OFF_TOGGLE`

Draw star labels.

star_twinkle

`ON_OFF_TOGGLE`

Draw stars twinkling.

stars

`ON_OFF_TOGGLE`

Draw stars.

time_lapse

`ON_OFF_TOGGLE`

Make a time lapse exposure of non-synthetic scene elements. Illustrate planetary trails, star trails, analemma, etc.

track_object

`ON_OFF_TOGGLE`

Center view on currently selected object.

translate_constellation_names

`ON_OFF_TOGGLE`

Whether to translate constellation names into the current sky locale. If you do not want to show translated constellation labels, set to false. If false, for example, you will see Latin names for all Western sky culture constellations.

tropic_lines

`ON_OFF_TOGGLE`

Draw tropic line.

flyto

Fly straight to the currently selected object or an object of your choice. If you need to be at a particular latitude/longitude when you arrive, see the **moveto** command in addition or instead.

```
flyto object Mars alt 10km duration 20
wait duration 20
```

alt

`DISTANCE`

Altitude desired above the object at final destination.

anchor

`follow`

`geosync`

With an anchor of 'geosync' you will stay over the same latitude and longitude on your anchored body as time passes. This is the default type of anchor for this command. With an anchor of 'follow' you will follow the anchored body but keep your orientation relative to the stars so you can view the body rotate on its axis.

duration

`SECONDS`

How long to take to effect this change. Defaults to 0.

object

`STRING`

Name of an object to fly to. Defaults to currently selected object, or if none, the currently anchored object.

As a side effect the object flown to becomes selected.

image

Display images. Images are loaded as transparent by default, so be sure to set the alpha value to be visible. The images stack on top of eachother (and other media) as they are loaded. Note that an image is dropped when the script is stopped or finishes unless “persist” is set.

```
image action load name fulldome-cat filename cat.jpg alpha 1 \
coordinate_system viewport scale fill
image action load name abc filename perspective.jpg alpha 1 \
coordinate_system dome altitude 45 azimuth 0 scale 35
image action load name 360 filename example.360.jpg alpha 1 \
coordinate_system horizontal scale 360
wait duration 20
```

action

`clear`

Drop all images loaded by a script. Can be filtered with the **persist** argument.

`drop`

Drop images when no longer needed to improve performance.

`load`

Load a new image to display.

alpha

`FADER`

0 is transparent (default), 1 is opaque. Note that images are drawn in the order they were loaded.

altitude

`DEGREES`

For positioning the center of the image in horizontal/dome coordinates. Zero is at the horizon, 90 is at the zenith.

azimuth

`DEGREES`

For positioning the center of the image in horizontal/dome coordinates. Please see page 6.

clone

`ON_OFF`

Whether to clone an image on opposite sides of the dome when using dome coordinates.

color_filter

`STRING`

GLSL shader code which modifies a vec4 color variable for each pixel. This allows filters like making certain colors transparent. There is a built in function to ease “green screen” erasure:

```
“chromaKey( color, vec4(0.05, 0.63, 0.14, 1), vec2(0.2, 0.25) )”
# first parameter:  leave as ‘color’
# second parameter: vec4 RGBA color of screen in [0, 1] range.
# third parameter:  vec2 YUV color distance for transition area (low, high)
```

coordinate_system

`COORDINATE_SYSTEM`

What coordinate system to use when positioning the image. This can not be changed later.

dec

`DECLINATION`

`DEGREES`

Declination of the image center for "equatorial" and "j2000" coordinate systems.

duration

`SECONDS`

How long to take to complete the command.

filename

`FILENAME`

Path must be relative to script. For the greatest backward compatibility or if you want transparency then PNG format is recommended.

lat

`DEGREES`

Latitude of the image center for "geocentric" coordinate system.

lon

`DEGREES`

Longitude of the image center for "geocentric" coordinate system.

persist

`ON_OFF`

If on, the image will persist even after the script ends. You can continue to control the image through the image command in other scripts by referencing name.

name

`STRING`

Unique name used to refer to the image in later calls to manipulate the image.

ra

`DEGREES`

`RIGHT_ASCENSION`

Right ascension of the image center for "equatorial" and "j2000" coordinate systems.

rotation

`DEGREES`

Absolute rotation about the center of the image, positive is clockwise when looking towards the center of the image.

scale

`fill`

In viewport coordinates, the image is scaled to completely cover the viewport while preserving the image aspect ratio. This means some of the image may be outside of the viewport. This argument is invalid for any other coordinate system.

REAL

How large to draw the image. In viewport coordinates, at 1 the image is scaled to fit maximized in the viewport without extending beyond the viewport edges. In other coordinate systems, this defines the maximum angular dimension of the image in degrees.

xpos

REAL

Position of the image center in "viewport" coordinates. In perspective projection mode the viewport is usually the screen. In fisheye mode the viewport is a square just containing the fisheye projection circle.

Zero is center of viewport, 1 is the right edge of the viewport, -1 is the left edge of the viewport.

ypos

REAL

Position of the image center in "viewport" coordinates. In perspective projection mode the viewport is usually the screen. In fisheye mode the viewport is a square just containing the fisheye projection circle.

Zero is center of viewport, 1 is the top edge of the viewport, -1 is the bottom edge of the viewport.

landscape

A landscape surrounds the viewer and is typically a photographic panorama with the sky areas transparent for sky viewing. Landscapes only draw if the user is "landed" on a body.

```
landscape action load name London body Earth night_texture \
night.png texture day.png top_altitude 20 type spherical
```

action

`load`

Load a landscape.

fov

DEGREES

For fisheye type landscapes, sets the field of view of the texture, typically 180°. Default is 180°.

base_altitude

DEGREES

For spherical type landscapes, sets the altitude angle of the bottom of the texture. Default is -90°.

body

`BODY_NAME (STRING)`

If defined, the landscape will only display when on the named body (for example, 'Mars').

night_texture

`FILENAME`

This image will fade in overlaying the normal texture as it gets dark. This image is optional. The file name needs to be specified in full including the path relative to the script. Must be PNG format with the sky transparent.

mipmap

`ON_OFF`

Whether to use mipmapping. If you have high contrast texture details, your landscape may look better with this option. Default is off.

rotate_z

`DEGREES`

Rotate the landscape around the z (up) axis. Default is 0° with the spherical landscape seam to the East and a fisheye oriented with the texture top at North.

texture

`IMAGE_FILENAME`

The standard landscape image. The file name needs to be specified in full including the path relative to the script. Must be PNG format with the sky transparent.

top_altitude

`DEGREES`

For spherical type landscapes, sets the altitude angle of the top of the texture. Default is 90°.

type

`fisheye`

`spherical`

The "fisheye" type is best is you have only one fisheye image with the zenith in the center. The "spherical" type is recommended for it's simplicity and also can support cylindrical landscapes using base_altitude and top_altitude parameters.

The "old_style" type has been deprecated.

layer

Add data layers such as satellite imagery, topographic height fields, or false color data visualization layers to reference frames (currently select bodies, but soon the sky itself). Data layers can cover the entire object or just be an inset of detail for a particular area. Data layers are implemented through the osgEarth library, so documentation for this library may be helpful for more advanced uses. Layers stay loaded until explicitly dropped or you restart Nightshade. A simple example:

```
layer name sea reference_frame Earth action load url imagery/sea.TIFF
wait duration 30
# Fade out over 3 seconds
layer name sea reference_frame Earth alpha 0 duration 3
wait duration 4
layer name sea reference_frame Earth action drop
```

action

`clear`

Drop all user loaded layers. Can also specify **reference_frame** to limit to layers on that object.

`load`

Load a new layer to display.

`drop`

Drop a layer when no longer needed to improve performance.

alpha

`FADER`

0 is transparent (default), 1 is opaque. Note that layers are drawn in the **z_order** defined, defaulting to newer layers over older layers.

duration

`SECONDS`

How long to take to complete the command.

lighting

`ON_OFF`

If lighting is on (default) then a body is shadowed normally. When off, the entire object is lit for better visibility.

max_level

`INTEGER`

Stop drawing the layer after this level of detail is reached. Level 0 is the least detailed with the level number increasing with detail.

min_level

`INTEGER`

Start drawing the layer when this level of detail is reached. Level 0 is the least detailed with the level number increasing with detail.

name

`STRING`

Unique name used to refer to the layer in later calls to manipulate the layer.

reference_frame

`Mars`

`Mercury`

`Moon`

`Earth`

`Europa`

`Pluto`

`Sky`

`Titan`

`Venus`

Name of the reference frame to load the layer into.

type

`heightfield`

`image`

Type of data layer. The default type is image. A heightfield is used for topography.

url

`URL`

This is the location of the data layer file or network resource. File paths are relative to the script itself.

xmax

`DEGREES`

If using an inset layer this defines the maximum X extent of the layer in degrees of longitude.

xmin

`DEGREES`

If using an inset layer this defines the minimum X extent of the layer in degrees of longitude.

ymax

`DEGREES`

If using an inset layer this defines the maximum Y extent of the layer in degrees of latitude.

ymin

`DEGREES`

If using an inset layer this defines the minimum Y extent of the layer in degrees of latitude.

z_order

`INTEGER`

Data layers stack on top of each other and the **z_order** defines this order. The **z_order** increases from the lowest layer. By default loading a layer places it on top with the highest **z_order**.

meteors

```
meteors zhr 10000 ra 30 dec 110 velocity 15
```

action

`reset`

Reset the radiant location to the apex of the Earth's way in space and the background meteor rate.

dec

`DECLINATION`

`DEGREES`

Declination of the radiant center in "j2000" coordinate system.

ra

`DEGREES`

`RIGHT_ASCENSION`

Right ascension of the radiant center in "j2000" coordinate system.

velocity

`default`

`REAL`

Speed of the meteors relative to the Earth in meters per second.

zhr

`default`

`REAL`

Zenith hourly rate in meteors per hour.

moveto

Move to another location on the same or a different body. Change your landed state. Latitude, longitude, pitch, heading, and altitude all are adjustable independently. If you want to fly directly in a straight line to another body, see the **flyto** command.

```
moveto lat 47.123 lon -122.345 alt 12km duration 30
wait duration 30
```

Currently the different dimensions can not be updated on overlapping moveto commands. For example, if one moveto starts while another is still running, the first will be interrupted by the second moveto command.

acceleration

`FADER`

Adjust the acceleration curve of the moveto altitude/distance animation. The value 0 produces linear motion.

alt

`default`

`DISTANCE`

Altitude or distance above the surface of the destination object. Value of "default" will reload saved value from configuration file.

duration

`default`

`SECONDS`

How long to take to effect this change. Value of "default" will use the value from your configuration file for auto_move_duration.

heading

`default`

`DEGREES`

Heading relative to North. Positive to the East. Value of "default" will reload saved value from configuration file.

This is not backward compatible with Nightshade Legacy **moveto heading** which was relative to screen up direction.

land

`default`

`ON_OFF_TOGGLE`

Whether to be landed on the anchored body at the end of the movement. Altitude must be relatively low for land to be able to take effect. Value of "default" will reload value from configuration file. When you are landed photographic landscapes can draw if enabled and terrain will not be visible.

lat

`default`

`DEGREES`

Latitude. South is negative. Value of "default" will reload value from configuration file.

lon

`default`

`DEGREES`

Longitude. West is negative. Value of "default" will reload value from configuration file.

look_at

`ON_OFF`

If on, you will look at the anchored object over a duration of ¼ your total moveto duration.

object

`default`

`STRING`

You can change your anchored body as part of a moveto. Value of "default" will reload value from configuration file.

pitch

`default`

`DEGREES`

Pitch. Positive up from the horizon. Value of "default" will reload value from configuration file.

qll

qypr

You will see these arguments generated when recording a script with gamepad movement. These are rotation information and not meant for human consumption. “qll” is a quaternion for latitude/longitude rotation, and “qypr” is for yaw/pitch/roll rotation. When present, these override other rotation arguments.

roll

`default`

`DEGREES`

Roll. At zero roll the horizon is level, with positive rotation of the horizon in the counter-clockwise direction about the focal point. Value of "default" will reload value from configuration file. Warning: roll and heading are conflated when tracking a body, which can lead to unpredictable results.

nebula

Not implemented.

point_cloud

Load a point cloud type data set, which at the most basic level is just a set of 3D points. Some examples are the SDSS galaxy data set, or a quasar catalog.

action

`clear`

Remove all loaded datasets.

`drop`

Remove a named dataset.

`load`

Read a dataset file and load it into the simulation.

color

`RGB`

`RGBA`

Define the base color of the dataset.

coordinate_system

`galactic`

`j2000`

This argument defines how the drawing coordinate system will be oriented (rotated) in space for the data set. Also see the related **reference_frame** argument. The default is to add no rotation.

filename

FILENAME

Data file containing point data. These are tab delimited files with the first line being the column headings. Supported headings are:

ra	Right ascension in degrees.
dec	Declination in degrees.
redshift	Redshift (unitless). Calculated distances will be in Mpc units, so be sure to set the value of the units argument to Mpc.
distance	Distance to object if not using redshift.
x	3D cartesian coordinate.
y	3D cartesian coordinate.
z	3D cartesian coordinate.

See the **units** or **scale_power** arguments to define your dataset scale. By default there is no scaling applied.

name

STRING

Unique name for referencing your dataset later.

scale_power

REAL

If the **units** argument is inadequate, you can specify a scale power factor instead. All data point dimensions will be scaled by 10 to the **scale_power** value from your reference frame scale. For example, if your reference frame is using meters and your data is in mega meters, your scale power would need to be 6.

reference_frame

Root

STRING

This defines where in the universe your dataset will be positioned. By default your dataset will be placed in the ‘Root’ reference frame, whose origin is the center of the Sun and whose units are mega meters. Reference frames are listed by Nightshade at startup and depend on what features you have. See also the related **coordinate_system** argument.

units

STRING

A unit of distance abbreviation as shown for **DISTANCE** on page 7 (except for ‘%’). If your reference frame is using standard world units of mega meters, your dataset will be scaled so that the dimensions are in these units. If your reference frame is not in world units, then you need to use **scale_power** instead for correct results.

require

This command is highly recommended to be the first line of every script you create. It identifies what is required for faithful playback of the script. This helps users understand what scripts are compatible with their software version and also affects some StratoScript behaviors for backward compatibility. In the future warnings may be displayed for incompatible scripts.

```
require projection_type fisheye version 15.1.1 release professional
```

projection_type

`perspective`

`fisheye`

Identify what projection type is required, if this matters.

release

`basic`

`community`

`professional`

Identify what software release is required, if this matters. The biggest potential issue is that Nightshade Basic is limited to travel within the solar system.

version

`INTEGER.INTEGER.INTEGER`

Use to define what minimum version of Nightshade is required. If no version is supplied, a script will be assumed to be written for version 11.12.1 (Nightshade Legacy), meaning that you will get some deprecated behavior such as no easing (speed ramp up and down at start and end of motion) on moveto animations.

script

```
# Pause script until user hits play
script action pause
# Exit this script and play a second script
script action play filename second.sts
```

action

end

See 'stop'. This is not required at the end of a script, but is useful if you want to quickly end a script in the middle without deleting the rest of the file.

play

Start playing a new script defined by the "filename" argument. The original script will terminate.

pause

Pause the current script.

stop

Stop playing the current script.

filename

FILENAME

Specify a script file.

select

If no arguments are supplied, deselects current object but leaves constellation selections alone. See "deselect" command.

```
select constellation TAU
select object Mars pointer off
```

constellation

`CONSTELLATION_SHORT_NAME`

Three character abbreviation. For built in sky cultures see: http://nightshadesoftware.org/projects/nightshade/wiki/Constellation_Abbreviations

hp

`INTEGER`

Select a star by its Hipparcos star catalog number.

nebula

`STRING`

Valid names begin with catalog identifiers M, NGC, or IC.

object

`home_planet`

`default`

`STRING`

English name of a currently loaded body or "home_planet" to select the body you are currently anchored to. To select your saved home body use “default”.

planet

Synonym for "object" which is preferred.

pointer

`ON_OFF`

Whether to draw the highlighting pointer around the selected object. Default is on.

star_only

`ON_OFF`

When selecting a star this is OFF by default, meaning that if the star is in a constellation line drawing for the current sky culture, that constellation will also be selected.

taxon

`TAXON`

When selecting a star this is OFF by default, meaning that if the star is in a constellation line

```
select taxon planetary.dwarf_planets.2002ms4
```

set

```
# Shrink labels
set label_scale 0.75
```

TAXON_ATTRIBUTE

Set the value of a setting in the cascading configuration taxonomy. The duration argument is usually supported. If an attribute is not set, it will be inherited from a higher level in the taxonomy. If multiple values could be inherited, the last parent in the list wins. See the **print** argument to be able to print out your current taxonomy. To remove a value, set the argument value to the literal value “unset”.

Current attributes:

- flag_label **ON_OFF** Show/hide body labels.
- flag_orbit **ON_OFF** Show/hide body orbit lines.
- flag_scale **ON_OFF** Enable/disable scaling bodies or 3D stars (not halos).
- flag_shadow_volume **ON_OFF** Show/hide body shadows visualizations.
- label_color **RGBA** Change body label color.
- orbit_color **RGBA** Change body orbit color.
- scale **REAL** Artificially scale bodies or 3D stars by this amount.
- shadow_volume_color **RGBA** Change body shadow volume color.

```
# Make planet orbits red by default
set planetary.planets.orbit_color 1,0,0,1 duration 5
# Allow satellites to scale by default
set planetary.satellites.flag_scale on duration 5
# Set the Moon’s scale to 4.5 times normal
set planetary.satellites.natural.moon.scale 4.5 duration 5
# Remove orbit color specific to Mars
set planetary.planets.mars.orbit_color unset
```

anchor

follow

geosync

With an anchor of 'geosync' you will stay over the same latitude and longitude on your anchored body as time passes. This is the default type of anchor. With an anchor of 'follow' you will follow the anchored body but keep your orientation relative to the stars so you can view the body rotate on its axis.

atmosphere_fade_duration

SECONDS

How long it should take to fade the atmosphere when turning on or off.

atmosphere_rendering_multiplier

REAL

Increase the rendered brightness of the atmosphere. Default is 1.

atmosphere_sun_bloom

REAL

Adjust the intensity of the Sun bloom in the atmosphere.

auto_move_duration

SECONDS

Used for auto zoom feature.

constellation_art_fade_duration

SECONDS

constellation_art_intensity

FADER

Works, but deprecated. Use "color property constellation_art alpha FADER" instead.

core_density

REAL

Change the density of the smaller/brighter star and body halo. Default is 0.45. Range is 0 to 1.

cursor_timeout

SECONDS

Amount of time before cursor will hide. At zero, the cursor never hides.

dome_tilt

DEGREES

Adjust this if you have a tilted dome and want to have the simulated horizon level with true horizontal.

duration

default

SECONDS

Currently only used for setting a duration for a heading or home_planet change (see below). Will use user's default duration settings if "default" is used. If not duration is provided, change is immediate.

NG should eventually support duration argument for all fadable setting changes.

flight_duration

SECONDS

This is the default length of time to use when flying to an object.

focal_alt

DEGREES

For use in a planetarium, this defines the altitude angle of the focal point above the dome springline. The focal point is where a tracked object will go. A user will already have this set correctly for their theater so changing this in a script you intend to share is discouraged.

focal_azi

DEGREES

For use in a planetarium, this defines the azimuth angle of the focal point as measured from the up direction on your video source. The focal point is where a tracked object will go. A user will already have this set correctly for their theater so changing this in a script you intend to share is discouraged.

gamma

REAL

Change the system rendering gamma. In general this should match your display gamma. Range is 0 to 2.8.

halo_density

`REAL`

Change the density of the fainter/larger star and body halo. Default is 1.01 Range is 0 to 10.

home_planet

`STRING`

Change viewing location, case sensitive, English name for the body.

label_scale

`REAL`

Affects body, star, and constellation labels. Range is 0.25 to 2. Default is 1.

landscape_name

`beach`

`forest`

`generic`

`hogerielen`

`hurricane`

`jupiter`

`mars`

`moon`

`mountain`

`neptune`

`saturn`

`uranus`

Set the built-in landscape to use. A landscape specific to Earth, for example, will only draw if you are landed on the Earth. If you land on a new body, a landscape for that body will try to be loaded if the current one is not suitable. For example, an Earth landscape will switch to a Mars landscape if you fly to and land on Mars. To load a custom landscape, see the **landscape** command.

light_exposure

`REAL`

Change the brightness of sunlit planetary terrain. Default is 1. Range is 0 to 30.

light_pollution_limiting_magnitude

`REAL`

Set naked eye limiting magnitude due to light pollution (light_pollution_luminance is now deprecated)

line_width

`REAL`

Pixel line width, can be fractional.

max_mag_planet_name

`REAL`

Only label and show orbit lines for bodies brighter than this magnitude.

max_mag_star_name

`REAL`

Only label stars brighter than this magnitude.

milky_way_intensity

`REAL`

1 is the default

milky_way_texture

See the **layer** command instead.

moon_scale

`REAL`

1 is real size. See **flag moon_scaled** to enable/disable.

print

`TAXON`

`*`

Print out attributes for the specified location in the configuration taxonomy to the log file (viewable in the Script Editor of the Universal Console).   A literal asterisk means to print out the entire taxonomy.

saturation

`REAL`

Change the system rendering color saturation. Default is 1. Range is 0 to 4.

sky_culture

`arab`

`aymara`

`aztec`

`belarusian`

`boorong`

`chinese`

`dakota`

`egyptian`

`hindu`

`inca`

`inuit`

`inuit-color`

`kamilaroi`

`korean`

`lakota`

`lokono`

`macedonian`

`maori`

mongolian

navajo

nazca

norse

ojibwe

polynesian

romanian

sami

sardinian

siberian

tongan

tukano

tupi

western

western-color

western-hevelius

western-mod

western-modest

Change to a different sky culture, which includes constellation data and star names. See abbreviations for selecting constellations.

sky_locale

LOCALE

What locale to use to translate your sky labels.

star_limiting_magnitude

MAGNITUDE

Default is 6.5. Simply does not draw stars dimmer than this value at a full sky view. Might be removed in future releases.

star_twinkle_amount

FADER

0 is no twinkling and 1 is deep twinkling. The amount sets how much the star can dip in brightness at each twinkle sampling.

time_zone

TIME_ZONE

Set the time zone used for your time display.

```
set time_zone America/Louisville
set time_zone Australia/Lord_Howe
```

terrain_detail_multiplier

REAL

Increase or decrease the amount of detail rendered when viewing high resolution layer imagery. Default is 1, and higher numbers decrease rendering performance. Range is 0.3 to 3.

sky_culture

Load a new sky culture, which can include constellation lines, constellation art, constellation boundaries, star names, and planet names. See instructions on creating a sky culture.

```
sky_culture action load path MyCultureDirectory
```

action

`load`

This data is loaded into memory and will be replaced if another sky culture is selected or loaded.

path

PATH

The directory should contain all the files necessary to define a sky culture.

soundscape

Digitarium NG Professional systems support adaptive sound tracks that react to where you are in space. It is also possible to create your own soundscapes, but this needs to be documented.

```
soundscape action play filename /media/Internal/sonic.defs volume 0.5
```

action

`drop`

Halt soundscape playback and unload the soundscape.

`pause`

Pause playback of soundscape.

`play`

Continue playing a paused soundscape or if 'filename' is defined, start a new soundscape.

`resume`

Continue playing a paused soundscape.

filename

FILENAME

OAML definition file for the soundscape.

This currently has to be an absolute path, but will be fixed to allow relative paths.

volume

FADER

Adjust the soundscape playback volume. 0 is muted, 1 is maximum volume.

text

Draw a string of text on screen. Only the position, rotation, and alpha can be changed once loaded.

```
text action load name title string “My Show” font_size 15 \
coordinate_system dome altitude 30 azimuth 180 r 1
text name title alpha 1 duration 5
wait duration 5
```

This is an experimental feature and subject to revisions.

The r (red), g (green), and b (blue) arguments are deprecated but still work. Use the color argument instead.

action

`clear`

Drop all text loaded by a script. Can be filtered with the **persist** argument.

`drop`

Drop text when no longer needed to improve performance.

`load`

Load a new text to display.

alpha

`FADER`

0 is transparent (default), 1 is opaque. Note that texts are drawn in the order they were loaded. If alpha is specified using the **color** argument, that overrides any alpha argument value here.

altitude

`DEGREES`

For positioning the center of the text in horizontal/dome coordinates. Zero is at the horizon, 90 is at the zenith.

azimuth

`DEGREES`

For positioning the center of the text in horizontal/dome coordinates. Please see page 6.

color

`RGB`

`RGBA`

If alpha is specified it overrides any separate **alpha** argument value.

coordinate_system

`COORDINATE_SYSTEM`

What coordinate system to use when positioning the image. This can not be changed later.

dec

`DECLINATOIN`

`DEGREES`

Declination of the text center for "equatorial" and "j2000" coordinate systems.

duration

`SECONDS`

How long to take to complete the command.

face

`sans`

`serif`

font_size

`DEGREES`

Approximate font height in degrees. May only be set at load time and may not exceed 45 degrees.

h_align

`left`

`center`

`right`

Horizontally align text relative to your desired position.

lat

`DEGREES`

Latitude of the image center for "geocentric" coordinate system.

lon

`DEGREES`

Longitude of the image center for "geocentric" coordinate system.

name

`STRING`

Unique name used to refer to the text in later calls to manipulate it.

ra

`DEGREES`

`RIGHT_ASCENSION`

Right ascension of the text center for "equatorial" and "j2000" coordinate systems.

rotation

`DEGREES`

Absolute rotation about the center of the text, positive is clockwise when looking towards the center of the text.

string

`MARKUP_STRING`

Text to draw onscreen.

xpos

`REAL`

Position of the text center in "viewport" coordinates. In perspective projection mode the viewport is usually the screen. In fisheye mode the viewport is a square just containing the fisheye projection circle.

Zero is center of viewport, 1 is the right edge of the viewport, -1 is the left edge of the viewport.

ypos

REAL

Position of the text center in "viewport" coordinates. In perspective projection mode the viewport is usually the screen. In fisheye mode the viewport is a square just containing the fisheye projection circle.

Zero is center of viewport, 1 is the top edge of the viewport, -1 is the bottom edge of the viewport.

v_align

`top`

`center`

`bottom`

`baseline`

Vertically align text relative to your desired position.

timerate

This command is used to adjust the how fast time elapses in the simulation. Units are seconds of simulation time per second of real time. A timerate of 1 is normal time, meaning that the simulation is running just as fast as real time.

```
timerate rate 100
wait duration 60
# Back to normal realtime rate of 1 second per second
timerate rate 1
```

action

`decrement`

`increment`

Decrement and increment generally adjust the simulation time rate by multiples of 10. Example steps: … -1000 -100 -10 -1 0 1 10 100 1000 …

`pause`

The simulation can be paused and resumed with 'pause'. If paused and resumed the simulation time rate will remain the same.

rate

REAL

Set simulation time rate in seconds of simulation time per second of real time.

video

Play and control video playback. Note that a video is stopped when the script that started it is stopped or finishes. Typically you will want the script to start the video, wait for some duration, and then call this command again with an "action stop" to stop the video.

Videos around 2k x 1k or higher in resolution should always be encoded for and played back with the vdec media plugin (see Administration menu item 8.15 and Vdec Release Notes).

```
video action play name fulldome-intro filename intro.nvdec.full.mpg \
alpha 1 coordinate_system viewport scale fill
wait duration 15
video action stop name fulldome-intro
video action play name tv filename ../media/landing.mp4 \
coordinate_system dome altitude 35 azimuth 45 scale 40 clone on
video name tv alpha 1 duration 5
wait duration 50
```

action

`clear`

Drop all videos loaded by a script.

`pause`

Pause playback of a video.

`play`

Continue playing a paused video or if 'filename' is defined, start a new video.

`resume`

Continue playing a paused video.

`stop`

Halt video playback and unload the video.

alpha

`FADER`

0 is transparent (default), 1 is opaque. Note that videos are drawn in the order they were loaded.

altitude

`DEGREES`

For positioning the center of the video in horizontal/dome coordinates. Zero is at the horizon, 90 is at the zenith.

azimuth

`DEGREES`

For positioning the center of the video in horizontal/dome coordinates. Please see page 6.

clone

`ON_OFF`

Whether to clone a video on opposite sides of the dome when using dome coordinates.

color_filter

`STRING`

GLSL shader code which modifies a vec4 color variable for each pixel. This allows filters like making certain colors transparent. There is a built in function to ease “green screen” erasure:

```
“chromaKey( color, vec4(0.05, 0.63, 0.14, 1), vec2(0.2, 0.25) )”
# first parameter:  leave as ‘color’
# second parameter: vec4 RGBA color of screen in [0, 1] range.
# third parameter:  vec2 YUV color distance for transition area (low, high)
```

coordinate_system

`COORDINATE_SYSTEM`

What coordinate system to use when positioning the image. This can not be changed later.

dec

`DECLINATION`

`DEGREES`

Declination of the video center for "equatorial" and "j2000" coordinate systems.

duration

`SECONDS`

How long to take to complete the command.

lat

`DEGREES`

Latitude of the image center for "geocentric" coordinate system.

lon

`DEGREES`

Longitude of the image center for "geocentric" coordinate system.

filename

`FILENAME`

Path must be relative to script.

max_performance

`ON_OFF`

Render only this video for maximum playback performance. Anything else that would be visible will be black.

name

`STRING`

Unique name used to refer to the video in later calls to manipulate the video.

ra

`DEGREES`

`RIGHT_ASCENSION`

Right ascension of the video center for "equatorial" and "j2000" coordinate systems.

rotation

`DEGREES`

Absolute rotation about the center of the video, positive is clockwise when looking towards the center of the video.

scale

`fill`

In viewport coordinates, the video is scaled to completely cover the viewport while preserving the video aspect ratio. This means some of the video may be outside of the viewport. This argument is invalid for any other coordinate system.

`REAL`

How large to draw the video. In viewport coordinates, at 1 the video is scaled to fit maximized in the viewport without extending beyond the viewport edges. In other coordinate systems, this defines the maximum angular dimension of the video in degrees.

seek

`SECONDS`

`RELATIVE_SECONDS`

Jump to a new position in the video file. The brackets mean that this value can begin with an optional '+' or '-'. If it does the seek will be made relative to the current position. Otherwise the position is measured from the beginning of the video. Can be used with the 'play' action to start somewhere besides the beginning of the video.

xpos

`REAL`

Position of the video center in "viewport" coordinates. In perspective projection mode the viewport is usually the screen. In fisheye mode the viewport is a square just containing the fisheye projection circle.

Zero is center of viewport, 1 is the right edge of the viewport, -1 is the left edge of the viewport.

ypos

`REAL`

Position of the video center in "viewport" coordinates. In perspective projection mode the viewport is usually the screen. In fisheye mode the viewport is a square just containing the fisheye projection circle.

Zero is center of viewport, 1 is the top edge of the viewport, -1 is the bottom edge of the viewport.

wait

This is a very important command, because most of the time in a script you will be waiting. Without wait commands everything would happen so quickly that you would not see or hear much of anything. Note that commands can sometimes take a bit of time to complete, such as loading a large image, and this can vary by system. If timing is critical, such as when synchronizing with a soundtrack, load your assets before starting the audio track.

```
wait duration 5  # Wait 5 seconds before executing next command
wait until 60 # Wait until 1 minute after the beginning of the script
```

duration

SECONDS

SECONDS can be fractional. How long to wait after the last command before running the next command.

until

[[INTEGER:]INTEGER:]SECONDS ([[HOURS:]MINUTES:]SECONDS)

Wait to proceed with the next command until the script has run for this much time since being started. **SECONDS** are required, HOURS are not required, and MINUTES are only required if HOURS are used.

```
wait until 3:59 # Wait until three minutes and 59 seconds
wait until 15.6 # Wait until 15.6 seconds into the script
```

zoom

This command allows you to change your field of view (fov). This is not the same thing as moving closer to an object, such as with the **flyto** command. With the zoom command you do not move, but instead enlarge (or shrink) your view like you have a powerful adjustable telescope.

```
select planet Jupiter
zoom auto in
wait duration 15
zoom auto initial
```

auto

in

When using auto zoom to zoom in, the currently selected object will be tracked and the field of view (fov) will be adjusted to show a system view of satellites of the object. If there are no satellites or the fov is smaller than the system view already, the fov will shrink to enlarge a view of just the object itself.

initial

Returns to configured initial fov and if landed, returns to initial view direction.

out

When using auto zoom to zoom out, the field of view will be adjusted to show a body view if zoomed in further already. Otherwise, if zoomed in further than a system view of satellites of the object (if there is one) then that will be shown on a next call, otherwise fov returns to the default. If called when at the satellite view, fov is returned to the default. When returning to default fov and in landed mode then the initial view direction is also returned to the default.

fov

`DEGREES`

Change the current field of view, in degrees

duration

`SECONDS`

manual

`in`

When using manual zoom to zoom in, the currently selected object will be tracked and the field of view (fov) will be reduced by one half. In other words magnification will be doubled.

`out`

When using manual zoom to zoom out, the fov will be doubled. In other words magnification will be reduced by one half. When the base fov is reached and in landed mode, the view direction will be reset to default.