Application Data (bpy.app)

This module contains application values that remain unchanged during runtime.

SUBMODULES

bpy.app.**autoexec_fail**

Undocumented, consider contributing.

bpy.app.**autoexec_fail_message**

Undocumented, consider contributing.

bpy.app.**autoexec_fail_quiet**

Undocumented, consider contributing.

bpy.app.**binary_path**

The location of Blender's executable, useful for utilities that open new instances. Read-only unless Blender is built as a Python module - in this case the value is an empty string which script authors may point to a Blender binary.

bpy.app.**debug**

Boolean, for debug info (started with `--debug` / `--debug-*` matching this attribute name)

bpy.app.**debug_depsgraph**

Boolean, for debug info (started with `--debug` / `--debug-*` matching this attribute name)

bpy.app.**debug_depsgraph_build**

Boolean, for debug info (started with `--debug` / `--debug-*` matching this attribute name)

bpy.app.**debug_depsgraph_eval**

Boolean, for debug info (started with `--debug` / `--debug-*` matching this attribute name)

bpy.app.**debug_depsgraph_pretty**

Boolean, for debug info (started with `--debug` / `--debug-*` matching this attribute name)

bpy.app.**debug_depsgraph_tag**

Boolean, for debug info (started with `--debug` / `--debug-*` matching this attribute name)

bpy.app.**debug_depsgraph_time**

Boolean, for debug info (started with `--debug` / `--debug-*` matching this attribute name)

bpy.app.**debug_events**

Boolean, for debug info (started with `--debug` / `--debug-*` matching this attribute name)

bpy.app.**debug_ffmpeg**

Boolean, for debug info (started with `--debug` / `--debug-*` matching this attribute name)

bpy.app.**debug_freestyle**

Boolean, for debug info (started with `--debug` / `--debug-*` matching this attribute name)

bpy.app.**debug_handlers**

Boolean, for debug info (started with `--debug` / `--debug-*` matching this attribute name)

bpy.app.**debug_io**

Boolean, for debug info (started with `--debug` / `--debug-*` matching this attribute name)

bpy.app.**debug_python**

Boolean, for debug info (started with `--debug` / `--debug-*` matching this attribute name)

bpy.app.**debug_simdata**

Boolean, for debug info (started with `--debug` / `--debug-*` matching this attribute name)

bpy.app.**debug_value**

Short, number which can be set to non-zero values for testing purposes

bpy.app.**debug_wm**

Boolean, for debug info (started with `--debug` / `--debug-*` matching this attribute name)

bpy.app.**driver_namespace**

Dictionary for drivers namespace, editable in-place, reset on file load (read-only)

File Loading & Order of Initialization

Since drivers may be evaluated immediately after loading a blend-file it is necessary to ensure the driver name-space is initialized beforehand.

This can be done by registering text data-blocks to execute on startup, which executes the scripts before drivers are evaluated. See *Text -> Register* from Blender's text editor.

Hint

You may prefer to use external files instead of Blender's text-blocks. This can be done using a text-block which executes an external file.

This example runs `driver_namespace.py` located in the same directory as the text-blocks blend-file:

```
import os
import bpy
blend_dir = os.path.normalize(os.path.join(__file__, "..", ".."))
bpy.utils.execfile(os.path.join(blend_dir, "driver_namespace.py"))
```

Using `__file__` ensures the text resolves to the expected path even when library-linked from another file.

Other methods of populating the drivers name-space can be made to work but tend to be error prone:

Using The `--python` command line argument to populate name-space often fails to achieve the desired goal because the initial evaluation will lookup a function that doesn't exist yet, marking the driver as invalid - preventing further evaluation.

Populating the driver name-space before the blend-file loads also doesn't work since opening a file clears the name-space.

It is possible to run a script via the `--python` command line argument, before the blend file. This can register a load-post handler (`bpy.app.handlers.load_post`) that initialized the name-space. While this works for background tasks it has the downside that opening the file from the file selector won't setup the name-space.

bpy.app.**online_access**

Boolean, true when internet access is allowed by Blender & 3rd party scripts (read-only)

bpy.app.**online_access_override**

Boolean, true when internet access preference is overridden by the command line (read-only)

bpy.app.**python_args**

Leading arguments to use when calling Python directly (via `sys.executable`). These arguments match settings Blender uses to ensure Pytho runs with a compatible environment (read-only).

bpy.app.**render_icon_size**

Reference size for icon/preview renders (read-only)

bpy.app.**render_preview_size**

Reference size for icon/preview renders (read-only)

bpy.app.**tempdir**

String, the temp directory used by blender (read-only)

bpy.app.**use_event_simulate**

Boolean, for application behavior (started with `--enable-*` matching this attribute name)

bpy.app.**use_userpref_skip_save_on_exit**

Boolean, for application behavior (started with `--enable-*` matching this attribute name)

bpy.app.**background**

Boolean, True when blender is running without a user interface (started with -b)

bpy.app.**factory_startup**

Boolean, True when blender is running with –factory-startup)

bpy.app.**module**

Boolean, True when running Blender as a python module

bpy.app.**portable**

Boolean, True unless blender was built to reference absolute paths (on UNIX).

bpy.app.**build_branch**

The branch this blender instance was built from

bpy.app.**build_cflags**

C compiler flags

bpy.app.**build_commit_date**

The date of commit this blender instance was built

bpy.app.**build_commit_time**

The time of commit this blender instance was built

bpy.app.**build_cxxflags**

C++ compiler flags

bpy.app.**build_date**

The date this blender instance was built

bpy.app.**build_hash**

The commit hash this blender instance was built with

bpy.app.**build_linkflags**

Binary linking flags

bpy.app.**build_platform**

The platform this blender instance was built for bpy.app.**build_system**

Build system used

bpy.app.**build_time**

The time this blender instance was built

bpy.app.**build_type**

The type of build (Release, Debug)

bpy.app.**build_commit_timestamp**

The unix timestamp of commit this blender instance was built

bpy.app.**version_cycle**

The release status of this build alpha/beta/rc/release

bpy.app.**version_string**

The Blender version formatted as a string

bpy.app.**version**

The Blender version as a tuple of 3 numbers (major, minor, micro). eg. (4, 3, 1)

bpy.app.**version_file**

The Blender File version, as a tuple of 3 numbers (major, minor, file sub-version), that will be used to save a .blend file. The last item in this tuple indicates the file sub-version, which is different from the release micro version (the last item of the *bpy.app.version* tuple). The file sub-version can incremented multiple times while a Blender version is under development. This value is, and should be, used for handling compatibility changes between Blender versions

bpy.app.**alembic**

Constant value bpy.app.alembic(supported=True, version=(1, 8, 3), version_string=' 1, 8, 3')

bpy.app.**build_options**

Constant value bpy.app.build_options(bullet=True, codec_avi=False, codec_ffmpeg=True, codec_sndfile=True, compositor_cpu=True, cycles=True, cycles_osl=True, freestyle=True, image_cineon=True, image_dds=True, image_hdr=True, image_openexr=True, image_openjpeg=True, image_tiff=True, input_ndof=True, audaspace=True, international=True, openal=True, opensubdiv=True, sdl=False, coreaudio=False, jack=True, pulseaudio=True, wasapi=False, libmv=True, mod_oceansim=True, mod_remesh=True, collada=True, io_wavefront_obj=True, io_ply=True, io_stl=True, io_gpencil=True, opencolorio=True, openmp=True, openvdb=True, alembic=True, usd=True, fluid=True, xr_openxr=True, potrace=True, pugixml=True, haru=True)

bpy.app.**ffmpeg**

Constant value bpy.app.ffmpeg(supported=True, avcodec_version=(60, 31, 102), avcodec_version_string='60, 31, 102', avdevice_version=(60,
100), avdevice_version_string='60, 3, 100', avformat_version=(60, 16, 100), avformat_version_string='60, 16, 100', avutil_version=(58, 29, 10
avutil_version_string='58, 29, 100', swscale_version=(7, 5, 100), swscale_version_string=' 7, 5, 100')

bpy.app.**ocio**

Constant value bpy.app.ocio(supported=True, version=(2, 4, 1), version_string=' 2, 4, 1')

bpy.app.**oiio**

Constant value bpy.app.oiio(supported=True, version=(3, 0, 3), version_string=' 3, 0, 3')

bpy.app.**opensubdiv**

Constant value bpy.app.opensubdiv(supported=True, version=(3, 6, 0), version_string=' 3, 6, 0')

bpy.app.**openvdb**

Constant value bpy.app.openvdb(supported=True, version=(12, 0, 0), version_string='12, 0, 0') bpy.app.**sdl**

Constant value bpy.app.sdl(supported=False, version=(0, 0, 0), version_string='Unknown')

bpy.app.**usd**

Constant value bpy.app.usd(supported=True, version=(0, 25, 2), version_string=' 0, 25, 2')

static bpy.app.**help_text(all=False)**

Return the help text as a string.

PARAMETERS:

all (*bool*) – Return all arguments, even those which aren't available for the current platform.

static bpy.app.**is_job_running(job_type)**

Check whether a job of the given type is running.

PARAMETERS:

job_type (*str*) – job type in Wm Job Type Items.

RETURNS:

Whether a job of the given type is currently running.

RETURN TYPE:

bool.

Copyright © Blender Authors
Made with Furo