# :mod:`zoneinfo` --- IANA time zone support

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 1); *backlink***

Unknown interpreted text role "mod".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 4)**

Unknown directive type "module".

```
.. module:: zoneinfo
    :synopsis: IANA time zone support
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 7)**

Unknown directive type "versionadded".

```
.. versionadded:: 3.9
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 9)**

Unknown directive type "moduleauthor".

```
.. moduleauthor:: Paul Ganssle <paul@ganssle.io>
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 10)**

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Paul Ganssle <paul@ganssle.io>
```

---

The :mod:`zoneinfo` module provides a concrete time zone implementation to support the IANA time zone database as originally specified in PEP 615. By default, :mod:`zoneinfo` uses the system's time zone data if available; if no system time zone data is available, the library will fall back to using the first-party tzdata package available on PyPI.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 14); *backlink***

Unknown interpreted text role "mod".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 14); *backlink***

Unknown interpreted text role "mod".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 20)**

Unknown directive type "seealso".

```
.. seealso::

    Module: :mod:`datetime`
        Provides the :class:`~datetime.time` and :class:`~datetime.datetime`
        types with which the :class:`ZoneInfo` class is designed to be used.

    Package `tzdata`_
        First-party package maintained by the CPython core developers to supply
        time zone data via PyPI.
```

## Using `ZoneInfo`

:class:`ZoneInfo` is a concrete implementation of the :class:`datetime.tzinfo` abstract base class, and is intended to be attached to `tzinfo`, either via the constructor, the :meth:`datetime.replace <datetime.datetime.replace>` method or :meth:`datetime.astimezone <datetime.datetime.astimezone>`:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 34); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 34); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 34); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 34); *backlink*

Unknown interpreted text role "meth".

```
>>> from zoneinfo import ZoneInfo
>>> from datetime import datetime, timedelta

>>> dt = datetime(2020, 10, 31, 12, tzinfo=ZoneInfo("America/Los_Angeles"))
>>> print(dt)
2020-10-31 12:00:00-07:00

>>> dt.tzname()
'PDT'
```

Datetimes constructed in this way are compatible with datetime arithmetic and handle daylight saving time transitions with no further intervention:

```
>>> dt_add = dt + timedelta(days=1)

>>> print(dt_add)
2020-11-01 12:00:00-08:00

>>> dt_add.tzname()
'PST'
```

These time zones also support the :attr:`~datetime.datetime.fold` attribute introduced in PEP 495. During offset transitions which induce ambiguous times (such as a daylight saving time to standard time transition), the offset from *before* the transition is used when `fold=0`, and the offset *after* the transition is used when `fold=1`, for example:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 60); *backlink*

Unknown interpreted text role "attr".

```
>>> dt = datetime(2020, 11, 1, 1, tzinfo=ZoneInfo("America/Los_Angeles"))
>>> print(dt)
2020-11-01 01:00:00-07:00

>>> print(dt.replace(fold=1))
2020-11-01 01:00:00-08:00
```

When converting from another time zone, the fold will be set to the correct value:

```
>>> from datetime import timezone
>>> LOS_ANGELES = ZoneInfo("America/Los_Angeles")
>>> dt_utc = datetime(2020, 11, 1, 8, tzinfo=timezone.utc)
```

```
>>> # Before the PDT -> PST transition
>>> print(dt_utc.astimezone(LOS_ANGELES))
2020-11-01 01:00:00-07:00

>>> # After the PDT -> PST transition
>>> print((dt_utc + timedelta(hours=1)).astimezone(LOS_ANGELES))
2020-11-01 01:00:00-08:00
```

## Data sources

The `zoneinfo` module does not directly provide time zone data, and instead pulls time zone information from the system time zone database or the first-party PyPI package tzdata, if available. Some systems, including notably Windows systems, do not have an IANA database available, and so for projects targeting cross-platform compatibility that require time zone data, it is recommended to declare a dependency on tzdata. If neither system data nor tzdata are available, all calls to :class:`ZoneInfo` will raise :exc:`ZoneInfoNotFoundError`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 91); *backlink***

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 91); *backlink***

Unknown interpreted text role "exc".

### Configuring the data sources

When `ZoneInfo(key)` is called, the constructor first searches the directories specified in :data:`TZPATH` for a file matching `key`, and on failure looks for a match in the tzdata package. This behavior can be configured in three ways:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 105); *backlink***

Unknown interpreted text role "data".

1. The default :data:`TZPATH` when not otherwise specified can be configured at :ref:`compile time <zoneinfo_data_compile_time_config>`.

   **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 110); *backlink***

   Unknown interpreted text role "data".

   **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 110); *backlink***

   Unknown interpreted text role "ref".

2. :data:`TZPATH` can be configured using :ref:`an environment variable <zoneinfo_data_environment_var>`.

   **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 112); *backlink***

   Unknown interpreted text role "data".

   **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 112); *backlink***

   Unknown interpreted text role "ref".

3. At :ref:`runtime <zoneinfo_data_runtime_config>`, the search path can be manipulated using the :func:`reset_tzpath` function.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 114); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 114); *backlink*

Unknown interpreted text role "func".

**Compile-time configuration**

The default :data:`TZPATH` includes several common deployment locations for the time zone database (except on Windows, where there are no "well-known" locations for time zone data). On POSIX systems, downstream distributors and those building Python from source who know where their system time zone data is deployed may change the default time zone path by specifying the compile-time option `TZPATH` (or, more likely, the :option:`configure flag --with-tzpath <--with-tzpath>`), which should be a string delimited by :data:`os.pathsep`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 122); *backlink*

Unknown interpreted text role "data".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 122); *backlink*

Unknown interpreted text role "option".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 122); *backlink*

Unknown interpreted text role "data".

On all platforms, the configured value is available as the `TZPATH` key in :func:`sysconfig.get_config_var`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 131); *backlink*

Unknown interpreted text role "func".

**Environment configuration**

When initializing :data:`TZPATH` (either at import time or whenever :func:`reset_tzpath` is called with no arguments), the `zoneinfo` module will use the environment variable `PYTHONTZPATH`, if it exists, to set the search path.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 139); *backlink*

Unknown interpreted text role "data".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 139); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 144)

Unknown directive type "envvar".

```
.. envvar:: PYTHONTZPATH

    This is an :data:`os.pathsep`-separated string containing the time zone
    search path to use. It must consist of only absolute rather than relative
    paths. Relative components specified in ``PYTHONTZPATH`` will not be used,
```

```
but otherwise the behavior when a relative path is specified is
implementation-defined; CPython will raise :exc:`InvalidTZPathWarning`, but
other implementations are free to silently ignore the erroneous component
or raise an exception.
```

To set the system to ignore the system data and use the tzdata package instead, set `PYTHONTZPATH=""`.

**Runtime configuration**

The TZ search path can also be configured at runtime using the :func:`reset_tzpath` function. This is generally not an advisable operation, though it is reasonable to use it in test functions that require the use of a specific time zone path (or require disabling access to the system time zones).

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 162); *backlink***

Unknown interpreted text role "func".

## The `ZoneInfo` class

A concrete :class:`datetime.tzinfo` subclass that represents an IANA time zone specified by the string `key`. Calls to the primary constructor will always return objects that compare identically; put another way, barring cache invalidation via :meth:`ZoneInfo.clear_cache`, for all values of `key`, the following assertion will always be true:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 173); *backlink***

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 173); *backlink***

Unknown interpreted text role "meth".

```
a = ZoneInfo(key)
b = ZoneInfo(key)
assert a is b
```

`key` must be in the form of a relative, normalized POSIX path, with no up-level references. The constructor will raise :exc:`ValueError` if a non-conforming key is passed.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 185); *backlink***

Unknown interpreted text role "exc".

If no file matching `key` is found, the constructor will raise :exc:`ZoneInfoNotFoundError`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 189); *backlink***

Unknown interpreted text role "exc".

The `ZoneInfo` class has two alternate constructors:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 195)**

Unknown directive type "classmethod".

```
.. classmethod:: ZoneInfo.from_file(fobj, /, key=None)

    Constructs a ``ZoneInfo`` object from a file-like object returning bytes
    (e.g. a file opened in binary mode or an :class:`io.BytesIO` object).
    Unlike the primary constructor, this always constructs a new object.

    The ``key`` parameter sets the name of the zone for the purposes of
    :py:meth:`~object.__str__` and :py:meth:`~object.__repr__`.
```

```
Objects created via this constructor cannot be pickled (see `pickling`_).
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 206)**

Unknown directive type "classmethod".

```
.. classmethod:: ZoneInfo.no_cache(key)

    An alternate constructor that bypasses the constructor's cache. It is
    identical to the primary constructor, but returns a new object on each
    call. This is most likely to be useful for testing or demonstration
    purposes, but it can also be used to create a system with a different cache
    invalidation strategy.

    Objects created via this constructor will also bypass the cache of a
    deserializing process when unpickled.

    .. TODO: Add "See `cache_behavior`_" reference when that section is ready.

    .. caution::

        Using this constructor may change the semantics of your datetimes in
        surprising ways, only use it if you know that you need to.
```

The following class methods are also available:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 226)**

Unknown directive type "classmethod".

```
.. classmethod:: ZoneInfo.clear_cache(*, only_keys=None)

    A method for invalidating the cache on the ``ZoneInfo`` class. If no
    arguments are passed, all caches are invalidated and the next call to
    the primary constructor for each key will return a new instance.

    If an iterable of key names is passed to the ``only_keys`` parameter, only
    the specified keys will be removed from the cache. Keys passed to
    ``only_keys`` but not found in the cache are ignored.

    .. TODO: Add "See `cache_behavior`_" reference when that section is ready.

    .. warning::

        Invoking this function may change the semantics of datetimes using
        ``ZoneInfo`` in surprising ways; this modifies process-wide global state
        and thus may have wide-ranging effects. Only use it if you know that you
        need to.
```

The class has one attribute:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 247)**

Unknown directive type "attribute".

```
.. attribute:: ZoneInfo.key

    This is a read-only :term:`attribute` that returns the value of ``key``
    passed to the constructor, which should be a lookup key in the IANA time
    zone database (e.g. ``America/New_York``, ``Europe/Paris`` or
    ``Asia/Tokyo``).

    For zones constructed from file without specifying a ``key`` parameter,
    this will be set to ``None``.

    .. note::

        Although it is a somewhat common practice to expose these to end users,
        these values are designed to be primary keys for representing the
        relevant zones and not necessarily user-facing elements.  Projects like
        CLDR (the Unicode Common Locale Data Repository) can be used to get
        more user-friendly strings from these keys.
```

### String representations

The string representation returned when calling :py:class:`str` on a :class:`ZoneInfo` object defaults to using the :attr:`ZoneInfo.key` attribute (see the note on usage in the attribute documentation):

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 268); *backlink*

Unknown interpreted text role "py:class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 268); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 268); *backlink*

Unknown interpreted text role "attr".

```
>>> zone = ZoneInfo("Pacific/Kwajalein")
>>> str(zone)
'Pacific/Kwajalein'

>>> dt = datetime(2020, 4, 1, 3, 15, tzinfo=zone)
>>> f"{dt.isoformat()} [{dt.tzinfo}]"
'2020-04-01T03:15:00+12:00 [Pacific/Kwajalein]'
```

For objects constructed from a file without specifying a `key` parameter, `str` falls back to calling :func:`repr`. `ZoneInfo`'s `repr` is implementation-defined and not necessarily stable between versions, but it is guaranteed not to be a valid `ZoneInfo` key.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 280); *backlink*

Unknown interpreted text role "func".

### Pickle serialization

Rather than serializing all transition data, `ZoneInfo` objects are serialized by key, and `ZoneInfo` objects constructed from files (even those with a value for `key` specified) cannot be pickled.

The behavior of a `ZoneInfo` file depends on how it was constructed:

1. `ZoneInfo(key)`: When constructed with the primary constructor, a `ZoneInfo` object is serialized by key, and when deserialized, the deserializing process uses the primary and thus it is expected that these are expected to be the same object as other references to the same time zone. For example, if `europe_berlin_pkl` is a string containing a pickle constructed from `ZoneInfo("Europe/Berlin")`, one would expect the following behavior:

   ```
   >>> a = ZoneInfo("Europe/Berlin")
   >>> b = pickle.loads(europe_berlin_pkl)
   >>> a is b
   True
   ```

2. `ZoneInfo.no_cache(key)`: When constructed from the cache-bypassing constructor, the `ZoneInfo` object is also serialized by key, but when deserialized, the deserializing process uses the cache bypassing constructor. If `europe_berlin_pkl_nc` is a string containing a pickle constructed from `ZoneInfo.no_cache("Europe/Berlin")`, one would expect the following behavior:

   ```
   >>> a = ZoneInfo("Europe/Berlin")
   >>> b = pickle.loads(europe_berlin_pkl_nc)
   >>> a is b
   False
   ```

3. `ZoneInfo.from_file(fobj, /, key=None)`: When constructed from a file, the `ZoneInfo` object raises an exception on pickling. If an end user wants to pickle a `ZoneInfo` constructed from a file, it is recommended that they use a wrapper type or a custom serialization function: either serializing by key or storing the contents of the file object and serializing that.

This method of serialization requires that the time zone data for the required key be available on both the serializing and deserializing side, similar to the way that references to classes and functions are expected to exist in both the serializing and deserializing environments. It also means that no guarantees are made about the consistency of results when unpickling a `ZoneInfo` pickled in an

environment with a different version of the time zone data.

## Functions

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 341)

Unknown directive type "function".

```
.. function:: available_timezones()

    Get a set containing all the valid keys for IANA time zones available
    anywhere on the time zone path. This is recalculated on every call to the
    function.

    This function only includes canonical zone names and does not include
    "special" zones such as those under the ``posix/`` and ``right/``
    directories, or the ``posixrules`` zone.

    .. caution::

        This function may open a large number of files, as the best way to
        determine if a file on the time zone path is a valid time zone is to
        read the "magic string" at the beginning.

    .. note::

        These values are not designed to be exposed to end-users; for user
        facing elements, applications should use something like CLDR (the
        Unicode Common Locale Data Repository) to get more user-friendly
        strings. See also the cautionary note on :attr:`ZoneInfo.key`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 364)

Unknown directive type "function".

```
.. function:: reset_tzpath(to=None)

    Sets or resets the time zone search path (:data:`TZPATH`) for the module.
    When called with no arguments, :data:`TZPATH` is set to the default value.

    Calling ``reset_tzpath`` will not invalidate the :class:`ZoneInfo` cache,
    and so calls to the primary ``ZoneInfo`` constructor will only use the new
    ``TZPATH`` in the case of a cache miss.

    The ``to`` parameter must be a :term:`sequence` of strings or
    :class:`os.PathLike` and not a string, all of which must be absolute paths.
    :exc:`ValueError` will be raised if something other than an absolute path
    is passed.
```

## Globals

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 381)

Unknown directive type "data".

```
.. data:: TZPATH

    A read-only sequence representing the time zone search path -- when
    constructing a ``ZoneInfo`` from a key, the key is joined to each entry in
    the ``TZPATH``, and the first file found is used.

    ``TZPATH`` may contain only absolute paths, never relative paths,
    regardless of how it is configured.

    The object that ``zoneinfo.TZPATH`` points to may change in response to a
    call to :func:`reset_tzpath`, so it is recommended to use
    ``zoneinfo.TZPATH`` rather than importing ``TZPATH`` from ``zoneinfo`` or
    assigning a long-lived variable to ``zoneinfo.TZPATH``.

    For more information on configuring the time zone search path, see
    :ref:`zoneinfo_data_configuration`.
```

## Exceptions and warnings

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 401)

Unknown directive type "exception".

```
.. exception:: ZoneInfoNotFoundError

    Raised when construction of a :class:`ZoneInfo` object fails because the
    specified key could not be found on the system. This is a subclass of
    :exc:`KeyError`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]zoneinfo.rst, line 407)

Unknown directive type "exception".

```
.. exception:: InvalidTZPathWarning

    Raised when :envvar:`PYTHONTZPATH` contains an invalid component that will
    be filtered out, such as a relative path.
```