

---

# **opaque\_keys Documentation**

**Author**

**Jul 13, 2021**



# CONTENTS

<b>1</b>	<b>opaque_keys package</b>	<b>3</b>
1.1	Subpackages	3
1.1.1	opaque_keys.edx package	3
1.1.1.1	Submodules	3
1.1.1.2	opaque_keys.edx.keys module	3
1.1.1.3	opaque_keys.edx.locations module	6
1.1.1.4	opaque_keys.edx.locator module	8
1.1.1.5	Module contents	18
1.2	Module contents	18
<b>2</b>	<b>Indices and tables</b>	<b>21</b>
	<b>Python Module Index</b>	<b>23</b>
	<b>Index</b>	<b>25</b>



Contents:



## OPAQUE\_KEYS PACKAGE

### 1.1 Subpackages

#### 1.1.1 opaque\_keys.edx package

##### 1.1.1.1 Submodules

##### 1.1.1.2 opaque\_keys.edx.keys module

; OpaqueKey abstract classes for edx-platform object types (courses, definitions, usages, and assets).

**class** opaque\_keys.edx.keys.AsideDefinitionKey(\*args, \*\*kwargs)

Bases: *opaque\_keys.edx.keys.DefinitionKey*

A definition key for an aside.

**abstract property** aside\_type

Return the type of this aside.

**abstract property** definition\_key

Return the DefinitionKey that this aside is decorating.

**class** opaque\_keys.edx.keys.AsideUsageKey(\*args, \*\*kwargs)

Bases: *opaque\_keys.edx.keys.UsageKey*

A usage key for an aside.

**abstract property** aside\_type

Return the type of this aside.

**abstract property** usage\_key

Return the UsageKey that this aside is decorating.

**class** opaque\_keys.edx.keys.AssetKey(\*args, \*\*kwargs)

Bases: *opaque\_keys.edx.keys.CourseObjectMixin*, *opaque\_keys.OpaqueKey*

An *opaque\_keys.OpaqueKey* identifying a course asset.

**KEY\_TYPE** = 'asset\_key'

**abstract property** asset\_type

Return what type of asset this is.

**abstract property** path

Return the path for this asset.

**class** opaque\_keys.edx.keys.**BlockTypeKey**(\*args, \*\*kwargs)

Bases: [opaque\\_keys.OpaqueKey](#)

A key class that encodes XBlock-family block types, including which family the block was loaded from.

**KEY\_TYPE** = 'block\_type'

**abstract property** block\_family

Return the block-family identifier (the entry-point used to load that block family).

**abstract property** block\_type

Return the block\_type of this block (the key in the entry-point to load the block with).

**class** opaque\_keys.edx.keys.**CourseKey**(\*args, \*\*kwargs)

Bases: [opaque\\_keys.edx.keys.LearningContextKey](#)

An [opaque\\_keys.OpaqueKey](#) identifying a particular Course object.

**abstract property** course

The name for this course.

In old-style IDs, it's the "course" in org/course/run

**is\_course** = True

**abstract make\_asset\_key**(asset\_type, path)

Return an asset key, given the given the specified path.

This function should not actually create any new ids, but should simply return one that already exists.

**abstract make\_usage\_key**(block\_type, block\_id)

Return a usage key, given the given the specified block\_type and block\_id.

This function should not actually create any new ids, but should simply return one that already exists.

**abstract property** org

The organization that this course belongs to.

**abstract property** run

The run for this course.

In old-style IDs, it's the "run" in org/course/run

**class** opaque\_keys.edx.keys.**CourseObjectMixin**

Bases: object

An abstract [opaque\\_keys.OpaqueKey](#) mixin for keys that belong to courses.

**abstract property** course\_key

Return the [CourseKey](#) for the course containing this usage.

**abstract map\_into\_course**(course\_key)

Return a new [UsageKey](#) or [AssetKey](#) representing this usage inside the course identified by the supplied [CourseKey](#). It returns the same type as *self*

**Args:** course\_key ([CourseKey](#)): The course to map this object into.

**Returns:** A new [CourseObjectMixin](#) instance.

**class** opaque\_keys.edx.keys.**DefinitionKey**(\*args, \*\*kwargs)

Bases: [opaque\\_keys.OpaqueKey](#)

An [opaque\\_keys.OpaqueKey](#) identifying an XBlock definition.

**KEY\_TYPE** = 'definition\_key'



**abstract property block\_type**  
The XBlock type of this definition.

**class** opaque\_keys.edx.keys.LearningContextKey(\*args, \*\*kwargs)

Bases: [opaque\\_keys.OpaqueKey](#)

An [opaque\\_keys.OpaqueKey](#) identifying a course, a library, a program, a website, or some other collection of content where learning happens.

This concept is more generic than “course.”

A learning context does not necessarily have an org, course, or, run.

**KEY\_TYPE** = 'context\_key'

**is\_course** = False

**make\_definition\_usage**(definition\_key, usage\_id=None)

Return a usage key, given the given the specified definition key and usage\_id.

**class** opaque\_keys.edx.keys.UsageKey(\*args, \*\*kwargs)

Bases: [opaque\\_keys.edx.keys.CourseObjectMixin](#), [opaque\\_keys.OpaqueKey](#)

An [opaque\\_keys.OpaqueKey](#) identifying an XBlock usage.

**KEY\_TYPE** = 'usage\_key'

**abstract property block\_id**

The name of this usage.

**abstract property block\_type**

The XBlock type of this usage.

**property context\_key**

Get the learning context key ([LearningContextKey](#)) for this XBlock usage.

**abstract property definition\_key**

Return the [DefinitionKey](#) for the XBlock containing this usage.

**class** opaque\_keys.edx.keys.UsageKeyV2(\*args, \*\*kwargs)

Bases: [opaque\\_keys.edx.keys.UsageKey](#)

An [opaque\\_keys.OpaqueKey](#) identifying an XBlock used in a specific learning context (e.g. a course).

Definition + Learning Context = Usage

UsageKeyV2 is just a subclass of UsageKey with slightly different behavior, but not a distinct key type (same KEY\_TYPE). UsageKeyV2 should be used for new usage key types; the main differences between it and UsageKey are:

- the .course\_key property is considered deprecated for the new V2 key types, and they should implement .context\_key instead.
- the .definition\_key property is explicitly disabled for V2 usage keys

**abstract property context\_key**

Get the learning context key ([LearningContextKey](#)) for this XBlock usage. May be a course key, library key, or some other [LearningContextKey](#) type.

**property course\_key**

Return the [CourseKey](#) for the course containing this usage.

**property definition\_key**

Returns the definition key for this usage. For the newer V2 key types, this cannot be done with the key alone, so it’s necessary to ask the key’s learning context to provide the underlying definition key.

`map_into_course(course_key)`

Implement `map_into_course` for API compatibility. Shouldn't be used in new code.

```
class opaque_keys.edx.keys.i4xEncoder(*, skipkeys=False, ensure_ascii=True, check_circular=True,
                                     allow_nan=True, sort_keys=False, indent=None,
                                     separators=None, default=None)
```

Bases: `json.encoder.JSONEncoder`

If provided as the `cls` to `json.dumps`, will serialize and Locations as i4x strings and other keys using the unicode strings.

`default(key)`

Implement this method in a subclass such that it returns a serializable object for `o`, or calls the base implementation (to raise a `TypeError`).

For example, to support arbitrary iterators, you could implement `default` like this:

```
def default(self, o):
    try:
        iterable = iter(o)
    except TypeError:
        pass
    else:
        return list(iterable)
    # Let the base class default method raise the TypeError
    return JSONEncoder.default(self, o)
```

### 1.1.1.3 opaque\_keys.edx.locations module

Deprecated OpaqueKey implementations used by XML and Mongo module stores

```
class opaque_keys.edx.locations.AssetLocation(org, course, run, category, name, revision=None,
                                              **kwargs)
```

Bases: `opaque_keys.edx.locations.LocationBase`, `opaque_keys.edx.locator.AssetLocator`

Deprecated. Use `locator.AssetLocator`

`DEPRECATED_TAG = 'c4x'`

`block_id`

`block_type`

`course_key`

`replace(**kwargs)`

**Return:** a new `AssetLocation` with specific `kwargs` replacing their corresponding values.

Using `AssetLocator`'s `replace` function results in a mismatch of `__init__` args and `kwargs`. Replace tries to instantiate an `AssetLocation` object with `AssetLocator`'s args and `kwargs`.

```
class opaque_keys.edx.locations.DeprecatedLocation(course_key, block_type, block_id)
```

Bases: `opaque_keys.edx.locator.BlockUsageLocator`

The short-lived location:org+course+run+block\_type+block\_id syntax

`CANONICAL_NAMESPACE = 'location'`

```

URL_RE = re.compile('^\\n
(?P<org>[\\w\\-~.:]+)\\+(?P<course>[\\w\\-~.:]+)\\+(?P<run>[\\w\\-~.:]+)\\+\\n
(?P<block_type>[\\w\\-~.:]+)\\+\\n (?P<block_id>[\\w\\-~.:]+)\\n \\Z', re.VERBOSE)

URL_RE_SOURCE = '\\n
(?P<org>[\\w\\-~.:]+)\\+(?P<course>[\\w\\-~.:]+)\\+(?P<run>[\\w\\-~.:]+)\\+\\n
(?P<block_type>[\\w\\-~.:]+)\\+\\n (?P<block_id>[\\w\\-~.:]+)\\n '

```

```

class opaque_keys.edx.locations.Location(org, course, run, category, name, revision=None, **kwargs)
    Bases: opaque_keys.edx.locations.LocationBase, opaque_keys.edx.locator.
    BlockUsageLocator

```

Deprecated. Use `locator.BlockUsageLocator`

```
DEPRECATED_TAG = 'i4x'
```

```
replace(**kwargs)
```

**Return:** a new `Location` with specific `kwargs` replacing their corresponding values.

Using `BlockUsageLocator`'s `replace` function results in a mismatch of `__init__` args and `kwargs`.

Replace tries to instantiate a `Location` object with `BlockUsageLocator`'s args and `kwargs`.

```

class opaque_keys.edx.locations.LocationBase(org, course, run, category, name, revision=None,
                                             **kwargs)

```

Bases: `object`

Deprecated. Base class for `Location` and `AssetLocation`

```
DEPRECATED_TAG = None
```

```
classmethod clean(value)
```

Deprecated. See `BlockUsageLocator.clean`

```
classmethod clean_for_html(value)
```

Deprecated. See `BlockUsageLocator.clean_for_html`

```
classmethod clean_for_url_name(value)
```

Deprecated. See `BlockUsageLocator.clean_for_url_name`

```
classmethod clean_keeping_underscores(value)
```

Deprecated. See `BlockUsageLocator.clean_keeping_underscores`

```
classmethod from_string(serialized)
```

Deprecated. Use `locator.BlockUsageLocator.from_string()`.

```
property tag
```

Deprecated. Returns the deprecated tag for this `Location`.

```

class opaque_keys.edx.locations.SlashSeparatedCourseKey(org, course, run, **kwargs)

```

Bases: `opaque_keys.edx.locator.CourseLocator`

Deprecated. Use `locator.CourseLocator`

```
branch
```

```
course
```

```
classmethod from_string(serialized)
```

Deprecated. Use `locator.CourseLocator.from_string()`.

```
org
```

`replace(**kwargs)`

**Return:** a new *SlashSeparatedCourseKey* with specific `kwargs` replacing their corresponding values.

Using `CourseLocator`'s `replace` function results in a mismatch of `__init__` args and `kwargs`. `replace` tries to instantiate a `SlashSeparatedCourseKey` object with `CourseLocator` args and `kwargs`.

`run`

`version_guid`

`class opaque_keys.edx.locations.i4xEncoder(*args, **kwargs)`

Bases: *opaque\_keys.edx.keys.i4xEncoder*

Deprecated. Use `keys.i4xEncoder`

#### 1.1.1.4 opaque\_keys.edx.locator module

Identifier for course resources.

`class opaque_keys.edx.locator.AssetLocator(course_key, block_type, block_id, **kwargs)`

Bases: *opaque\_keys.edx.locator.BlockUsageLocator*, *opaque\_keys.edx.keys.AssetKey*

An `AssetKey` implementation class.

`ALLOWED_ID_RE = re.compile('^[\w\-\~.:%]+\Z')`

`ASSET_URL_RE = re.compile('\n ^\n /c4x/\n (?P<org>[^/]+)/\n (?P<course>[^/]+)/\n (?P<category>[^/]+)/\n (?P<name>[^@]+)\n (@(?P<revision>[^/]+))?\n \Z\n ', re.VERBOSE)`

`CANONICAL_NAMESPACE = 'asset-v1'`

`DEPRECATED_ALLOWED_ID_RE = re.compile('^[\w\-\~.:%]+\Z')`

`DEPRECATED_TAG = 'c4x'`

`property asset_type`

Return what type of asset this is.

`block_id`

`block_type`

`course_key`

`property path`

Return the path for this asset.

`replace(**kwargs)`

**Return:** a new `OpaqueKey` with `KEY_FIELDS` specified in `kwargs` replaced their corresponding values. Deprecation value is also preserved.

Subclasses should override this if they have required properties that aren't included in their `KEY_FIELDS`.

`property tag`

Returns the deprecated tag for this `Location`.

**to\_deprecated\_list\_repr()**

Thumbnail locations are stored as lists [c4x, org, course, thumbnail, path, None] in contentstore.mongo. That should be the only use of this method, but the method is general enough to provide the pre-opaque Location fields as an array in the old order with the tag.

**class** opaque\_keys.edx.locator.**BlockLocatorBase**(\*args, \*\*kwargs)

Bases: [opaque\\_keys.edx.locator.Locator](#)

Abstract base class for XBlock locators.

See subclasses for more detail, particularly *CourseLocator* and *BlockUsageLocator*.

**ALLOWED\_ID\_RE** = re.compile('^[\w\-\~.:]+\Z')

**BLOCK\_ALLOWED\_ID\_CHARS** = '[\w\-\~.:]'

**BLOCK\_PREFIX** = 'block'

**BRANCH\_PREFIX** = 'branch'

**DEPRECATED\_ALLOWED\_ID\_RE** = re.compile('^[\w\-\~.:]+\Z')

**URL\_RE** = re.compile('^[\n ((?P<org>[\w\-\~.:]+\Z)\+(?P<course>[\w\-\~.:]+\Z)(\+(?P<run>[\w\-\~.:]+\Z)?(\+(?=.)|\Z))??\n (branch@(P<branch>[\w\-\~.:]+\Z)(\+(?=.)|\Z))?\n (version@(P<version\_guid>[, re.VERBOSE)

**URL\_RE\_SOURCE** = '\n ((?P<org>[\w\-\~.:]+\Z)\+(?P<course>[\w\-\~.:]+\Z)(\+(?P<run>[\w\-\~.:]+\Z)?(\+(?=.)|\Z))??\n (branch@(P<branch>[\w\-\~.:]+\Z)(\+(?=.)|\Z))?\n (version@(P<version\_guid>[a-f0-9]+\Z)(\+(?=.)|\Z))?\n (type@(P<block\_type>[\w\-\~.:]+\Z)(\+(?=.)|\Z))?\n (block@(P<block\_id>[\w\-\~.:]+\Z))?\n '

**deprecated**

**classmethod** **parse\_url**(string)

If it can be parsed as a version\_guid with no preceding org + offering, returns a dict with key 'version\_guid' and the value,

If it can be parsed as a org + offering, returns a dict with key 'id' and optional keys 'branch' and 'version\_guid'.

**Raises:** InvalidKeyError: if string cannot be parsed -or- string ends with a newline.

**class** opaque\_keys.edx.locator.**BlockUsageLocator**(course\_key, block\_type, block\_id, \*\*kwargs)

Bases: [opaque\\_keys.edx.locator.BlockLocatorBase](#), [opaque\\_keys.edx.keys.UsageKey](#)

Encodes a location.

Locations address modules (aka blocks) which are definitions situated in a course instance. Thus, a Location must identify the course and the occurrence of the defined element in the course. Courses can be a version of an offering, the current draft head, or the current production version.

Locators can contain both a version and a org + course + run w/ branch. The split mongo functions may raise errors if these conflict w/ the current db state (i.e., the course's branch != the version\_guid)

**Locations can express as urls as well as dictionaries. They consist of** package\_identifier: course\_guid | version\_guid block : guid branch : string

BlockUsageLocators also support deprecated Location-style formatting with the following mapping: Location(org, course, run, category, name, revision) is represented as a BlockUsageLocator with:

- course\_key = a CourseKey comprised of (org, course, run, branch=revision)
- block\_type = category

- `block_id = name`

`CANONICAL_NAMESPACE = 'block-v1'`

`CHECKED_INIT = False`

`DEPRECATED_INVALID_CHARS = re.compile('[^\w.%-]')`

`DEPRECATED_INVALID_CHARS_NAME = re.compile('[^\w.:%-]')`

`DEPRECATED_INVALID_HTML_CHARS = re.compile('[^\w-]')`

`DEPRECATED_TAG = 'i4x'`

`DEPRECATED_URL_RE = re.compile('\n i4x://\n (?P<org>[^/]+)/\n (?P<course>[^/]+)/\n (?P<category>[^/]+)/ # category == block_type\n (?P<name>[^@]+) # name == block_id\n (@(?P<re, re.VERBOSE)`

`KEY_FIELDS = ('course_key', 'block_type', 'block_id')`

`block_id = None`

`block_type = None`

**property branch**

Returns the branch for this object's `course_key`.

**property category**

Deprecated. The ambiguously named field from `Location` which code expects to find. Equivalent to `block_type`.

**classmethod clean(*value*)**

Should only be called on deprecated-style values

Return value, made into a form legal for locations

**classmethod clean\_for\_html(*value*)**

Should only be called on deprecated-style values

Convert a string into a form that's safe for use in html ids, classes, urls, etc. Replaces all `INVALID_HTML_CHARS` with `'_'`, collapses multiple `'_'` chars

**classmethod clean\_for\_url\_name(*value*)**

Should only be called on deprecated-style values

Convert value into a format valid for location names (allows colons).

**classmethod clean\_keeping\_underscores(*value*)**

Should only be called on deprecated-style values

Return value, replacing `INVALID_CHARS`, but not collapsing multiple `'_'` chars. This for cleaning asset names, as the YouTube ID's may have underscores in them, and we need the transcript asset name to match. In the future we may want to change the behavior of `_clean`.

**property course**

Returns the course for this object's `course_key`.

**course\_agnostic()**

We only care about the locator's version not its course. Returns a copy of itself without any course info.

**Raises:** `ValueError` if the block locator has no `version_guid`

`course_key = None`

**property definition\_key**

Returns the definition key for this object. Undefined for `Locators`.

**for\_branch**(*branch*)

Return a UsageLocator for the same block in a different branch of the course.

**for\_version**(*version\_guid*)

Return a UsageLocator for the same block in a different branch of the course.

**html\_id**()

Return an id which can be used on an html page as an id attr of an html element. It is currently also persisted by some clients to identify blocks.

To make compatible with old Location object functionality. I don't believe this behavior fits at this place, but I have no way to override. We should clearly define the purpose and restrictions of this (e.g., I'm assuming periods are fine).

**is\_fully\_specified**()

Returns boolean; whether or not this object's course\_key is fully specified.

**classmethod make\_relative**(*course\_locator, block\_type, block\_id*)

Return a new instance which has the given block\_id in the given course :param course\_locator: may be a BlockUsageLocator in the same snapshot

**map\_into\_course**(*course\_key*)

Return a new instance which has the this block\_id in the given course :param course\_key: a CourseKey object representing the new course to map into

**property name**

Deprecated. The ambiguously named field from Location which code expects to find. Equivalent to block\_id.

**property offering**

Deprecated. Use course and run independently.

**property org**

Returns the org for this object's course\_key.

**replace**(*\*\*kwargs*)

**Return:** a new **OpaqueKey** with **KEY\_FIELDS** specified in **kwargs** replaced their corresponding values. Deprecation value is also preserved.

Subclasses should override this if they have required properties that aren't included in their KEY\_FIELDS.

**property revision**

Deprecated. The ambiguously named field from Location which code expects to find. Equivalent to branch.

**property run**

Returns the run for this object's course\_key.

**to\_deprecated\_son**(*prefix='', tag='i4x'*)

Returns a SON object that represents this location

**property version**

Deprecated. The ambiguously named field from CourseLocation which code expects to find. Equivalent to version\_guid.

**version\_agnostic**()

We don't care if the locator's version is not the current head; so, avoid version conflict by reducing info. Returns a copy of itself without any version info.

**Raises:** ValueError: if the block locator has no org, course, and run

### property `version_guid`

Returns the version guid for this object.

```
class opaque_keys.edx.locator.BundleDefinitionLocator(bundle_uuid, block_type, olx_path,  
bundle_version=None, draft_name=None,  
_version_or_draft=None)
```

Bases: `opaque_keys.edx.locator.CheckFieldMixin`, `opaque_keys.edx.keys.DefinitionKey`

Implementation of the `DefinitionKey` type, for XBlock content stored in Blockstore bundles. This is a low-level identifier used within the Open edX system for identifying and retrieving OLX.

A “Definition” is a specific OLX file in a specific `BundleVersion` (or sometimes rather than a `BundleVersion`, it may point to a named draft.) The OLX file, and thus the definition key, defines `Scope.content` fields as well as defaults for `Scope.settings` and `Scope.children` fields. However the definition has no parent and no position in any particular course or other context - both of which require a *usage key* and not just a definition key. The same block definition (.olx file) can be used in multiple places in a course, each with a different usage key.

Example serialized definition keys follow.

The ‘html’ type OLX file “html/introduction/definition.xml” in bundle 11111111-1111-1111-1111-111111111111, bundle version 5:

```
bundle-olx:11111111-1111-1111-1111-111111111111:5:html:html/introduction/definition.xml
```

The ‘problem’ type OLX file “problem324234.xml” in bundle 22222222-2222-2222-2222-222222222222, draft ‘studio-draft’:

```
bundle-olx:22222222-2222-2222-2222-222222222222:studio-draft:problem:problem/324234.xml
```

(The serialized version is somewhat long and verbose because it should rarely be used except for debugging - the in-memory python key instance will be used most of the time, and users will rarely/never see definition keys.)

User state should never be stored using a `BundleDefinitionLocator` as the key. State should always be stored against a usage locator, which refers to a particular definition being used in a particular context.

### Each `BundleDefinitionLocator` holds the following data

1. Bundle UUID and [bundle version OR draft name]
2. Block type (e.g. ‘html’, ‘problem’, etc.)
3. Path to OLX file

Note that since the data in an .olx file can only ever change in a bundle draft (not in a specific bundle version), an XBlock that is actively making changes to its `Scope.content/Scope.settings` field values must have a `BundleDefinitionLocator` with a draft name (not a bundle version).

```
CANONICAL_NAMESPACE = 'bundle-olx'
```

```
CHECKED_INIT = False
```

```
KEY_FIELDS = ('bundle_uuid', 'block_type', 'olx_path', '_version_or_draft')
```

```
OLX_PATH_REGEX = re.compile('^[\\w\\-\\.]+$')
```

```
block_type
```

```
bundle_uuid
```

```
property bundle_version
```

Get the Blockstore bundle version number, or None if a Blockstore draft name has been specified instead.

```
property draft_name
```

Get the Blockstore draft name, or None if a Blockstore bundle version number has been specified instead.

```
olx_path
```



**class** opaque\_keys.edx.locator.**CheckFieldMixin**

Bases: object

Mixin that provides handy methods for checking field types/values.

**class** opaque\_keys.edx.locator.**CourseLocator**(*org=None, course=None, run=None, branch=None, version\_guid=None, deprecated=False, \*\*kwargs*)

Bases: *opaque\_keys.edx.locator.BlockLocatorBase, opaque\_keys.edx.keys.CourseKey*

**Examples of valid CourseLocator specifications:** CourseLocator(version\_guid=ObjectId('519665f6223ebd6980884f2b'))

```
CourseLocator(org='mit.eecs', course='6.002x', run='T2_2014') CourseLoca-
tor(org='mit.eecs', course='6002x', run='fall_2014' branch = 'published')
CourseLocator.from_string('course-v1:version@519665f6223ebd6980884f2b')
CourseLocator.from_string('course-v1:mit.eecs+6002x') CourseLocator.from_string('course-
v1:mit.eecs+6002x+branch@published') CourseLocator.from_string('course-
v1:mit.eecs+6002x+branch@published+version@519665f6223ebd6980884f2b')
```

Should have at least a specific org, course, and run with optional 'branch', or version\_guid (which points to a specific version). Can contain both in which case the persistence layer may raise exceptions if the given version != the current such version of the course.

**CANONICAL\_NAMESPACE** = 'course-v1'

**CHECKED\_INIT** = False

**INVALID\_CHARS\_DEPRECATED** = re.compile('[^\w.%-]')

**KEY\_FIELDS** = ('org', 'course', 'run', 'branch', 'version\_guid')

**branch**

**course**

**course\_agnostic()**

We only care about the locator's version not its course. Returns a copy of itself without any course info.

**Raises:** ValueError: if the block locator has no version\_guid

**for\_branch(branch)**

Return a new CourseLocator for another branch of the same course (also version agnostic)

**for\_version(version\_guid)**

Return a new CourseLocator for another version of the same course and branch. Usually used when the head is updated (and thus the course x branch now points to this version)

**html\_id()**

Return an id which can be used on an html page as an id attr of an html element.

To make compatible with old Location object functionality. I don't believe this behavior fits at this place, but I have no way to override. We should clearly define the purpose and restrictions of this (e.g., I'm assuming periods are fine).

**make\_asset\_key(asset\_type, path)**

Return an asset key, given the given the specified path.

This function should not actually create any new ids, but should simply return one that already exists.

**make\_usage\_key(block\_type, block\_id)**

Return a usage key, given the given the specified block\_type and block\_id.

This function should not actually create any new ids, but should simply return one that already exists.

**make\_usage\_key\_from\_deprecated\_string(location\_url)**

Deprecated mechanism for creating a UsageKey given a CourseKey and a serialized Location.

NOTE: this prejudicially takes the tag, org, and course from the url not self.

**Raises:** `InvalidKeyError`: if the url does not parse

### **property offering**

Deprecated. Use `course` and `run` independently.

### **org**

### **run**

### **property version**

Deprecated. The ambiguously named field from `CourseLocation` which code expects to find. Equivalent to `version_guid`.

### **version\_agnostic()**

We don't care if the locator's version is not the current head; so, avoid version conflict by reducing info. Returns a copy of itself without any version info.

**Raises:** `ValueError`: if the block locator has no `org` & `course`, run

### **version\_guid**

**class** `opaque_keys.edx.locator.DefinitionLocator`(*block\_type, definition\_id, deprecated=False*)

Bases: `opaque_keys.edx.locator.Locator`, `opaque_keys.edx.keys.DefinitionKey`

Container for how to locate a description (the course-independent content).

**CANONICAL\_NAMESPACE** = 'def-v1'

**CHECKED\_INIT** = False

**KEY\_FIELDS** = ('definition\_id', 'block\_type')

**URL\_RE** =

```
re.compile('^(?P<definition_id>[a-f0-9]+)\\+type@(?P<block_type>[\\w\\-\\.:]+)\\Z',  
re.VERBOSE)
```

**block\_type** = None

**definition\_id** = None

### **property version**

Returns the `ObjectId` referencing this specific location.

**class** `opaque_keys.edx.locator.LibraryLocator`(*org=None, library=None, branch=None, version\_guid=None, \*\*kwargs*)

Bases: `opaque_keys.edx.locator.BlockLocatorBase`, `opaque_keys.edx.keys.CourseKey`

Locates a library. Libraries are `XBlock` structures with a 'library' block at their root.

Libraries are treated analogously to courses for now. Once opaque keys are better supported, they will no longer have the 'run' property, and may no longer conform to `CourseKey` but rather some more general key type.

**Examples of valid LibraryLocator specifications:** `LibraryLocator(version_guid=ObjectId('519665f6223ebd6980884f2b'))`  
`LibraryLocator(org='UniX', library='PhysicsProbs')`      `LibraryLocator.from_string('library-v1:UniX+PhysicsProbs')`

`version_guid` is optional.

The constructor accepts 'course' as a deprecated alias for the 'library' attribute.

`branch` is optional.

**CANONICAL\_NAMESPACE** = 'library-v1'

**CHECKED\_INIT** = False

**KEY\_FIELDS** = ('org', 'library', 'branch', 'version\_guid')

**RUN** = 'library'

**branch**

**property course**

Deprecated. Return a 'course' for compatibility with CourseLocator.

**course\_agnostic()**

We only care about the locator's version not its library. Returns a copy of itself without any library info.

**Raises:** ValueError: if the block locator has no version\_guid

**for\_branch(*branch*)**

Return a new CourseLocator for another branch of the same library (also version agnostic)

**for\_version(*version\_guid*)**

Return a new LibraryLocator for another version of the same library and branch. Usually used when the head is updated (and thus the library x branch now points to this version)

**html\_id()**

Return an id which can be used on an html page as an id attr of an html element.

**is\_course** = False

**library**

**make\_asset\_key(*asset\_type, path*)**

Return an asset key, given the given the specified path.

This function should not actually create any new ids, but should simply return one that already exists.

**make\_usage\_key(*block\_type, block\_id*)**

Return a usage key, given the given the specified block\_type and block\_id.

This function should not actually create any new ids, but should simply return one that already exists.

**org**

**property run**

Deprecated. Return a 'run' for compatibility with CourseLocator.

**property version**

Deprecated. The ambiguously named field from CourseLocation which code expects to find. Equivalent to version\_guid.

**version\_agnostic()**

We don't care if the locator's version is not the current head; so, avoid version conflict by reducing info. Returns a copy of itself without any version info.

**Raises:** ValueError: if the block locator has no org & course, run

**version\_guid**

**class** opaque\_keys.edx.locator.**LibraryLocatorV2**(*org, slug*)

Bases: [opaque\\_keys.edx.locator.CheckFieldMixin](#), [opaque\\_keys.edx.keys.LearningContextKey](#)

A key that represents a Blockstore-based content library.

**When serialized, these keys look like:** lib:MITx:reallyhardproblems lib:hogwarts:p300-potions-exercises

**CANONICAL\_NAMESPACE** = 'lib'

**CHECKED\_INIT** = False

**KEY\_FIELDS** = ('org', 'slug')

```
SLUG_REGEX = re.compile('^[\\w\\-\\.]+$')
```

```
for_branch(branch)
```

Compatibility helper. Some code calls `.for_branch(None)` on course keys. By implementing this, it improves backwards compatibility between library keys and course keys.

```
make_definition_usage(definition_key, usage_id=None)
```

Return a usage key, given the given the specified definition key and usage\_id.

```
org
```

```
slug
```

```
class opaque_keys.edx.locator.LibraryUsageLocator(library_key, block_type, block_id, **kwargs)
```

Bases: `opaque_keys.edx.locator.BlockUsageLocator`

Just like `BlockUsageLocator`, but this points to a block stored in a library, not a course.

```
CANONICAL_NAMESPACE = 'lib-block-v1'
```

```
KEY_FIELDS = ('library_key', 'block_type', 'block_id')
```

```
block_id = None
```

```
block_type = None
```

```
property course_key
```

To enable compatibility with `BlockUsageLocator`, we provide a read-only `course_key` property.

```
for_branch(branch)
```

Return a `UsageLocator` for the same block in a different branch of the library.

```
for_version(version_guid)
```

Return a `UsageLocator` for the same block in a different version of the library.

```
library_key = None
```

```
replace(**kwargs)
```

**Return:** a new `OpaqueKey` with `KEY_FIELDS` specified in `kwargs` replaced their corresponding values. Deprecation value is also preserved.

Subclasses should override this if they have required properties that aren't included in their `KEY_FIELDS`.

```
property run
```

Returns the run for this object's `library_key`.

```
to_deprecated_son(prefix='', tag='i4x')
```

Disable some deprecated methods of our parent class.

```
version_agnostic()
```

We don't care if the locator's version is not the current head; so, avoid version conflict by reducing info. Returns a copy of itself without any version info.

**Raises:** `ValueError`: if the block locator has no org, course, and run

```
class opaque_keys.edx.locator.LibraryUsageLocatorV2(lib_key, block_type, usage_id)
```

Bases: `opaque_keys.edx.locator.CheckFieldMixin`, `opaque_keys.edx.keys.UsageKeyV2`

An `XBlock` in a Blockstore-based content library.

**When serialized, these keys look like:** `lb:MITx:reallyhardproblems:problem:problem1`

```
CANONICAL_NAMESPACE = 'lb'
```

```
CHECKED_INIT = False
```

```

KEY_FIELDS = ('lib_key', 'block_type', 'usage_id')
USAGE_ID_REGEX = re.compile('^[\\w\\-\\.]+$')

property block_id
    Get the 'block ID' which is another name for the usage ID.

block_type

property context_key
    Get the learning context key (LearningContextKey) for this XBlock usage. May be a course key, library key, or some other LearningContextKey type.

html_id()
    Return an id which can be used on an html page as an id attr of an html element. This is only in here for backwards-compatibility with XModules; don't use in new code.

lib_key

usage_id

class opaque_keys.edx.locator.LocalId(block_id=None)
    Bases: object

    Class for local ids for non-persisted xblocks (which can have hardcoded block_ids if necessary)

class opaque_keys.edx.locator.Locator(*args, **kwargs)
    Bases: opaque_keys.OpaqueKey

    A locator identifies a course resource.

    Locator is an abstract base class: do not instantiate

ALLOWED_ID_CHARS = '[\\w\\-~.:]'
BLOCK_TYPE_PREFIX = 'type'
DEPRECATED_ALLOWED_ID_CHARS = '[\\w\\-~.:%]'
VERSION_PREFIX = 'version'

classmethod as_object_id(value)
    Attempts to cast value as a bson.objectid.ObjectId.

    Raises: ValueError: if casting fails

deprecated

property version
    Returns the ObjectId referencing this specific location.

    Raises: InvalidKeyError: if the instance doesn't have a complete enough specification.

class opaque_keys.edx.locator.VersionTree(locator, tree_dict=None)
    Bases: object

    Holds trees of Locators to represent version histories.

```

### 1.1.1.5 Module contents

## 1.2 Module contents

Defines the *OpaqueKey* class, to be used as the base-class for implementing pluggable OpaqueKeys.

These keys are designed to provide a limited, forward-evolveable interface to an application, while concealing the particulars of the serialization formats, and allowing new serialization formats to be installed transparently.

**exception** opaque\_keys.InvalidKeyError(*key\_class*, *serialized*)

Bases: Exception

Raised to indicate that a serialized key isn't valid (wasn't able to be parsed by any available providers).

**class** opaque\_keys.OpaqueKey(\**args*, \*\**kwargs*)

Bases: object

A base-class for implementing pluggable opaque keys. Individual key subclasses identify particular types of resources, without specifying the actual form of the key (or its serialization).

There are two levels of expected subclasses: Key type definitions, and key implementations

```
OpaqueKey
|
Key type
|
Key implementation
```

The key type base class must define the class property `KEY_TYPE`, which identifies which `entry_point` namespace the keys implementations should be registered with.

The KeyImplementation classes must define the following:

**CANONICAL\_NAMESPACE** Identifies the key namespace for the particular key implementation (when serializing).

Key implementations must be registered using the `CANONICAL_NAMESPACE` as their `entry_point` name, but can also be registered with other names for backwards compatibility.

**KEY\_FIELDS** A list of attribute names that will be used to establish object identity. Key implementation instances will compare equal iff all of their `KEY_FIELDS` match, and will not compare equal to instances of different KeyImplementation classes (even if the `KEY_FIELDS` match). These fields must be hashable.

**\_to\_string** Serialize the key into a unicode object. This should not include the namespace prefix (`CANONICAL_NAMESPACE`).

**\_from\_string** Construct an instance of this *OpaqueKey* from a unicode object. The namespace will already have been parsed.

OpaqueKeys will not have optional constructor parameters (due to the implementation of `KEY_FIELDS`), by default. However, an implementation class can provide a default, as long as it passes that default to a call to `super().__init__`. If the KeyImplementation sets the class attribute `CHECKED_INIT` to `False`, then the *OpaqueKey* base class constructor will not validate any of the `KEY_FIELDS` arguments, and will instead just expect all `KEY_FIELDS` to be passed as `kwargs`.

*OpaqueKey* objects are immutable.

Serialization of an *OpaqueKey* is performed by using the `unicode()` builtin. Deserialization is performed by the `from_string()` method.

**CANONICAL\_NAMESPACE** = None

**CHECKED\_INIT** = True

```
KEY_FIELDS = []
```

```
LOADED_DRIVERS = {}
```

```
NAMESPACE_SEPARATOR = ':'
```

**deprecated**

**classmethod** `from_string`(*serialized*)

Return a *OpaqueKey* object deserialized from the *serialized* argument. This object will be an instance of a subclass of the *cls* argument.

**Args:** *serialized*: A stringified form of a *OpaqueKey*

**classmethod** `get_namespace_plugin`(*namespace*)

Return the registered *OpaqueKey* subclass of *cls* for the supplied namespace

**replace**(*\*\*kwargs*)

**Return:** a new *OpaqueKey* with **KEY\_FIELDS** specified in **kwargs** replaced their corresponding values. Deprecation value is also preserved.

Subclasses should override this if they have required properties that aren't included in their **KEY\_FIELDS**.

**classmethod** `set_deprecated_fallback`(*fallback*)

Register a deprecated fallback class for this class to revert to.

**class** `opaque_keys.OpaqueKeyMetaClass`(*name, bases, attrs*)

Bases: `abc.ABCMeta`

Metaclass for *OpaqueKey*. Sets the default value for the values in **KEY\_FIELDS** to `None`.





## INDICES AND TABLES

- genindex
- modindex
- search



## PYTHON MODULE INDEX

### O

`opaque_keys`, 18

`opaque_keys.edx`, 18

`opaque_keys.edx.keys`, 3

`opaque_keys.edx.locations`, 6

`opaque_keys.edx.locator`, 8



## INDEX

### A

ALLOWED\_ID\_CHARS (*opaque\_keys.edx.locator.Locator* attribute), 17

ALLOWED\_ID\_RE (*opaque\_keys.edx.locator.AssetLocator* attribute), 8

ALLOWED\_ID\_RE (*opaque\_keys.edx.locator.BlockLocatorBase* attribute), 9

as\_object\_id() (*opaque\_keys.edx.locator.Locator* class method), 17

aside\_type (*opaque\_keys.edx.keys.AsideDefinitionKey* property), 3

aside\_type (*opaque\_keys.edx.keys.AsideUsageKey* property), 3

AsideDefinitionKey (class in *opaque\_keys.edx.keys*), 3

AsideUsageKey (class in *opaque\_keys.edx.keys*), 3

asset\_type (*opaque\_keys.edx.keys.AssetIdKey* property), 3

asset\_type (*opaque\_keys.edx.locator.AssetLocator* property), 8

ASSET\_URL\_RE (*opaque\_keys.edx.locator.AssetLocator* attribute), 8

AssetKey (class in *opaque\_keys.edx.keys*), 3

AssetLocation (class in *opaque\_keys.edx.locations*), 6

AssetLocator (class in *opaque\_keys.edx.locator*), 8

### B

BLOCK\_ALLOWED\_ID\_CHARS (*opaque\_keys.edx.locator.BlockLocatorBase* attribute), 9

block\_family (*opaque\_keys.edx.keys.BlockTypeKey* property), 4

block\_id (*opaque\_keys.edx.keys.UsageKey* property), 5

block\_id (*opaque\_keys.edx.locations.AssetIdLocation* attribute), 6

block\_id (*opaque\_keys.edx.locator.AssetLocator* attribute), 8

block\_id (*opaque\_keys.edx.locator.BlockUsageLocator* attribute), 10

block\_id (*opaque\_keys.edx.locator.LibraryUsageLocator* attribute), 16

block\_id (*opaque\_keys.edx.locator.LibraryUsageLocatorV2* property), 17

BLOCK\_PREFIX (*opaque\_keys.edx.locator.BlockLocatorBase* attribute), 9

block\_type (*opaque\_keys.edx.keys.BlockTypeKey* property), 4

block\_type (*opaque\_keys.edx.keys.DefinitionKey* property), 4

block\_type (*opaque\_keys.edx.keys.UsageKey* property), 5

block\_type (*opaque\_keys.edx.locations.AssetIdLocation* attribute), 6

block\_type (*opaque\_keys.edx.locator.AssetLocator* attribute), 8

block\_type (*opaque\_keys.edx.locator.BlockUsageLocator* attribute), 10

block\_type (*opaque\_keys.edx.locator.BundleDefinitionLocator* attribute), 12

block\_type (*opaque\_keys.edx.locator.DefinitionLocator* attribute), 14

block\_type (*opaque\_keys.edx.locator.LibraryUsageLocator* attribute), 16

block\_type (*opaque\_keys.edx.locator.LibraryUsageLocatorV2* attribute), 17

BLOCK\_TYPE\_PREFIX (*opaque\_keys.edx.locator.Locator* attribute), 17

BlockLocatorBase (class in *opaque\_keys.edx.locator*), 9

BlockTypeKey (class in *opaque\_keys.edx.keys*), 3

BlockUsageLocator (class in *opaque\_keys.edx.locator*), 9

branch (*opaque\_keys.edx.locations.SlashSeparatedCourseKey* attribute), 7

branch (*opaque\_keys.edx.locator.BlockUsageLocator* property), 10

branch (*opaque\_keys.edx.locator.CourseLocator* attribute), 13

branch (*opaque\_keys.edx.locator.LibraryLocator* attribute), 15

BRANCH\_PREFIX (*opaque\_keys.edx.locator.BlockLocatorBase* attribute), 9

bundle\_uuid (*opaque\_keys.edx.locator.BundleDefinitionLocator*

- attribute), 12
- bundle\_version (*opaque\_keys.edx.locator.BundleDefinitionLocator* attribute), 16
- property), 12
- BundleDefinitionLocator (class in *opaque\_keys.edx.locator*), 12
- ## C
- CANONICAL\_NAMESPACE (*opaque\_keys.edx.locations.DeprecatedLocation* attribute), 6
- CANONICAL\_NAMESPACE (*opaque\_keys.edx.locator.AssetLocator* attribute), 8
- CANONICAL\_NAMESPACE (*opaque\_keys.edx.locator.BlockUsageLocator* attribute), 10
- CANONICAL\_NAMESPACE (*opaque\_keys.edx.locator.BundleDefinitionLocator* attribute), 12
- CANONICAL\_NAMESPACE (*opaque\_keys.edx.locator.CourseLocator* attribute), 13
- CANONICAL\_NAMESPACE (*opaque\_keys.edx.locator.DefinitionLocator* attribute), 14
- CANONICAL\_NAMESPACE (*opaque\_keys.edx.locator.LibraryLocator* attribute), 14
- CANONICAL\_NAMESPACE (*opaque\_keys.edx.locator.LibraryLocatorV2* attribute), 15
- CANONICAL\_NAMESPACE (*opaque\_keys.edx.locator.LibraryUsageLocator* attribute), 16
- CANONICAL\_NAMESPACE (*opaque\_keys.edx.locator.LibraryUsageLocatorV2* attribute), 16
- CANONICAL\_NAMESPACE (*opaque\_keys.OpaqueKey* attribute), 18
- category (*opaque\_keys.edx.locator.BlockUsageLocator* property), 10
- CHECKED\_INIT (*opaque\_keys.edx.locator.BlockUsageLocator* attribute), 10
- CHECKED\_INIT (*opaque\_keys.edx.locator.BundleDefinitionLocator* attribute), 12
- CHECKED\_INIT (*opaque\_keys.edx.locator.CourseLocator* attribute), 13
- CHECKED\_INIT (*opaque\_keys.edx.locator.DefinitionLocator* attribute), 14
- CHECKED\_INIT (*opaque\_keys.edx.locator.LibraryLocator* attribute), 14
- CHECKED\_INIT (*opaque\_keys.edx.locator.LibraryLocatorV2* attribute), 15
- CHECKED\_INIT (*opaque\_keys.edx.locator.LibraryUsageLocatorV2* attribute), 16
- CHECKED\_INIT (*opaque\_keys.OpaqueKey* attribute), 18
- CheckFieldMixin (class in *opaque\_keys.edx.locator*), 12
- clean() (*opaque\_keys.edx.locations.LocationBase* class method), 7
- clean() (*opaque\_keys.edx.locator.BlockUsageLocator* class method), 10
- clean\_for\_html() (*opaque\_keys.edx.locations.LocationBase* class method), 7
- clean\_for\_html() (*opaque\_keys.edx.locator.BlockUsageLocator* class method), 10
- clean\_for\_url\_name() (*opaque\_keys.edx.locations.LocationBase* class method), 7
- clean\_for\_url\_name() (*opaque\_keys.edx.locator.BlockUsageLocator* class method), 10
- clean\_keeping\_underscores() (*opaque\_keys.edx.locations.LocationBase* class method), 7
- clean\_keeping\_underscores() (*opaque\_keys.edx.locator.BlockUsageLocator* class method), 10
- context\_key (*opaque\_keys.edx.keys.UsageKey* property), 5
- context\_key (*opaque\_keys.edx.keys.UsageKeyV2* property), 5
- context\_key (*opaque\_keys.edx.locator.LibraryUsageLocatorV2* property), 17
- course (*opaque\_keys.edx.keys.CourseKey* property), 4
- course (*opaque\_keys.edx.locations.SlashSeparatedCourseKey* attribute), 7
- course (*opaque\_keys.edx.locator.BlockUsageLocator* property), 10
- course (*opaque\_keys.edx.locator.CourseLocator* attribute), 13
- course (*opaque\_keys.edx.locator.LibraryLocator* property), 15
- course\_agnostic() (*opaque\_keys.edx.locator.BlockUsageLocator* method), 10
- course\_agnostic() (*opaque\_keys.edx.locator.CourseLocator* method), 13
- course\_agnostic() (*opaque\_keys.edx.locator.LibraryLocator* method), 15
- course\_key (*opaque\_keys.edx.keys.CourseObjectMixin* property), 4
- course\_key (*opaque\_keys.edx.keys.UsageKeyV2* property), 5
- course\_key (*opaque\_keys.edx.locations.AssetPosition* attribute), 6
- course\_key (*opaque\_keys.edx.locator.AssetLocator* attribute), 8

- course\_key (*opaque\_keys.edx.locator.BlockUsageLocator* DEPRECATED\_TAG (*opaque\_keys.edx.locator.AssetLocator* attribute), 10  
 course\_key (*opaque\_keys.edx.locator.LibraryUsageLocator* DEPRECATED\_TAG (*opaque\_keys.edx.locator.BlockUsageLocator* property), 16  
 CourseKey (class in *opaque\_keys.edx.keys*), 4  
 CourseLocator (class in *opaque\_keys.edx.locator*), 13  
 CourseObjectMixin (class in *opaque\_keys.edx.keys*), 4  
 DEPRECATED\_URL\_RE (*opaque\_keys.edx.locator.BlockUsageLocator* attribute), 10  
 DeprecatedLocation (class in *opaque\_keys.edx.locations*), 6  
 draft\_name (*opaque\_keys.edx.locator.BundleDefinitionLocator* property), 12  
**D**  
 default() (*opaque\_keys.edx.keys.i4xEncoder* method), 6  
 definition\_id (*opaque\_keys.edx.locator.DefinitionLocator* attribute), 14  
 definition\_key (*opaque\_keys.edx.keys.AsideDefinitionKey* property), 3  
 definition\_key (*opaque\_keys.edx.keys.UsageKey* property), 5  
 definition\_key (*opaque\_keys.edx.keys.UsageKeyV2* property), 5  
 definition\_key (*opaque\_keys.edx.locator.BlockUsageLocator* property), 10  
 DefinitionKey (class in *opaque\_keys.edx.keys*), 4  
 DefinitionLocator (class in *opaque\_keys.edx.locator*), 14  
 deprecated (*opaque\_keys.edx.locator.BlockLocatorBase* attribute), 9  
 deprecated (*opaque\_keys.edx.locator.Locator* attribute), 17  
 deprecated (*opaque\_keys.OpaqueKey* attribute), 19  
 DEPRECATED\_ALLOWED\_ID\_CHARS (*opaque\_keys.edx.locator.Locator* attribute), 17  
 DEPRECATED\_ALLOWED\_ID\_RE (*opaque\_keys.edx.locator.AssetLocator* attribute), 8  
 DEPRECATED\_ALLOWED\_ID\_RE (*opaque\_keys.edx.locator.BlockLocatorBase* attribute), 9  
 DEPRECATED\_INVALID\_CHARS (*opaque\_keys.edx.locator.BlockUsageLocator* attribute), 10  
 DEPRECATED\_INVALID\_CHARS\_NAME (*opaque\_keys.edx.locator.BlockUsageLocator* attribute), 10  
 DEPRECATED\_INVALID\_HTML\_CHARS (*opaque\_keys.edx.locator.BlockUsageLocator* attribute), 10  
 DEPRECATED\_TAG (*opaque\_keys.edx.locations.AssetLocation* attribute), 6  
 DEPRECATED\_TAG (*opaque\_keys.edx.locations.Location* attribute), 7  
 DEPRECATED\_TAG (*opaque\_keys.edx.locations.LocationBase* attribute), 7  
**F**  
 for\_branch() (*opaque\_keys.edx.locator.BlockUsageLocator* method), 10  
 for\_branch() (*opaque\_keys.edx.locator.CourseLocator* method), 13  
 for\_branch() (*opaque\_keys.edx.locator.LibraryLocator* method), 15  
 for\_branch() (*opaque\_keys.edx.locator.LibraryLocatorV2* method), 16  
 for\_branch() (*opaque\_keys.edx.locator.LibraryUsageLocator* method), 16  
 for\_version() (*opaque\_keys.edx.locator.BlockUsageLocator* method), 11  
 for\_version() (*opaque\_keys.edx.locator.CourseLocator* method), 13  
 for\_version() (*opaque\_keys.edx.locator.LibraryLocator* method), 15  
 for\_version() (*opaque\_keys.edx.locator.LibraryUsageLocator* method), 16  
 from\_string() (*opaque\_keys.edx.locations.LocationBase* class method), 7  
 from\_string() (*opaque\_keys.edx.locations.SlashSeparatedCourseKey* class method), 7  
 from\_string() (*opaque\_keys.OpaqueKey* class method), 19  
**G**  
 get\_namespace\_plugin() (*opaque\_keys.OpaqueKey* class method), 19  
**H**  
 html\_id() (*opaque\_keys.edx.locator.BlockUsageLocator* method), 11  
 html\_id() (*opaque\_keys.edx.locator.CourseLocator* method), 13  
 html\_id() (*opaque\_keys.edx.locator.LibraryLocator* method), 15  
 html\_id() (*opaque\_keys.edx.locator.LibraryUsageLocatorV2* method), 17  
 i4xEncoder (class in *opaque\_keys.edx.keys*), 6

- `i4xEncoder` (class in `opaque_keys.edx.locations`), 8
- `INVALID_CHARS_DEPRECATED` (`opaque_keys.edx.locator.CourseLocator` attribute), 13
- `InvalidKeyError`, 18
- `is_course` (`opaque_keys.edx.keys.CourseKey` attribute), 4
- `is_course` (`opaque_keys.edx.keys.LearningContextKey` attribute), 5
- `is_course` (`opaque_keys.edx.locator.LibraryLocator` attribute), 15
- `is_fully_specified()` (`opaque_keys.edx.locator.BlockUsageLocator` method), 11
- ## K
- `KEY_FIELDS` (`opaque_keys.edx.locator.BlockUsageLocator` attribute), 10
- `KEY_FIELDS` (`opaque_keys.edx.locator.BundleDefinitionLocator` attribute), 12
- `KEY_FIELDS` (`opaque_keys.edx.locator.CourseLocator` attribute), 13
- `KEY_FIELDS` (`opaque_keys.edx.locator.DefinitionLocator` attribute), 14
- `KEY_FIELDS` (`opaque_keys.edx.locator.LibraryLocator` attribute), 15
- `KEY_FIELDS` (`opaque_keys.edx.locator.LibraryLocatorV2` attribute), 15
- `KEY_FIELDS` (`opaque_keys.edx.locator.LibraryUsageLocator` attribute), 16
- `KEY_FIELDS` (`opaque_keys.edx.locator.LibraryUsageLocatorV2` attribute), 16
- `KEY_FIELDS` (`opaque_keys.OpaqueKey` attribute), 18
- `KEY_TYPE` (`opaque_keys.edx.keys.AssetKey` attribute), 3
- `KEY_TYPE` (`opaque_keys.edx.keys.BlockTypeKey` attribute), 4
- `KEY_TYPE` (`opaque_keys.edx.keys.DefinitionKey` attribute), 4
- `KEY_TYPE` (`opaque_keys.edx.keys.LearningContextKey` attribute), 5
- `KEY_TYPE` (`opaque_keys.edx.keys.UsageKey` attribute), 5
- ## L
- `LearningContextKey` (class in `opaque_keys.edx.keys`), 5
- `lib_key` (`opaque_keys.edx.locator.LibraryUsageLocatorV2` attribute), 17
- `library` (`opaque_keys.edx.locator.LibraryLocator` attribute), 15
- `library_key` (`opaque_keys.edx.locator.LibraryUsageLocator` attribute), 16
- `LibraryLocator` (class in `opaque_keys.edx.locator`), 14
- `LibraryLocatorV2` (class in `opaque_keys.edx.locator`), 15
- `LibraryUsageLocator` (class in `opaque_keys.edx.locator`), 16
- `LibraryUsageLocatorV2` (class in `opaque_keys.edx.locator`), 16
- `LOADED_DRIVERS` (`opaque_keys.OpaqueKey` attribute), 19
- `LocalId` (class in `opaque_keys.edx.locator`), 17
- `Location` (class in `opaque_keys.edx.locations`), 7
- `LocationBase` (class in `opaque_keys.edx.locations`), 7
- `Locator` (class in `opaque_keys.edx.locator`), 17
- ## M
- `make_asset_key()` (`opaque_keys.edx.keys.CourseKey` method), 4
- `make_asset_key()` (`opaque_keys.edx.locator.CourseLocator` method), 13
- `make_asset_key()` (`opaque_keys.edx.locator.LibraryLocator` method), 15
- `make_definition_usage()` (`opaque_keys.edx.keys.LearningContextKey` method), 5
- `make_definition_usage()` (`opaque_keys.edx.locator.LibraryLocatorV2` method), 16
- `make_relative()` (`opaque_keys.edx.locator.BlockUsageLocator` class method), 11
- `make_usage_key()` (`opaque_keys.edx.keys.CourseKey` method), 4
- `make_usage_key()` (`opaque_keys.edx.locator.CourseLocator` method), 13
- `make_usage_key()` (`opaque_keys.edx.locator.LibraryLocator` method), 15
- `make_usage_key_from_deprecated_string()` (`opaque_keys.edx.locator.CourseLocator` method), 13
- `map_into_course()` (`opaque_keys.edx.keys.CourseObjectMixin` method), 4
- `map_into_course()` (`opaque_keys.edx.keys.UsageKeyV2` method), 5
- `map_into_course()` (`opaque_keys.edx.locator.BlockUsageLocator` method), 11
- module
- `opaque_keys`, 18
  - `opaque_keys.edx`, 18
  - `opaque_keys.edx.keys`, 3
  - `opaque_keys.edx.locations`, 6
  - `opaque_keys.edx.locator`, 8
- ## N
- `name` (`opaque_keys.edx.locator.BlockUsageLocator` property), 11
- `NAMESPACE_SEPARATOR` (`opaque_keys.OpaqueKey` attribute), 19



## O

offering (*opaque\_keys.edx.locator.BlockUsageLocator* property), 11  
 offering (*opaque\_keys.edx.locator.CourseLocator* property), 14  
 olx\_path (*opaque\_keys.edx.locator.BundleDefinitionLocator* attribute), 12  
 OLX\_PATH\_REGEX (*opaque\_keys.edx.locator.BundleDefinitionLocator* attribute), 12  
 opaque\_keys  
   module, 18  
 opaque\_keys.edx  
   module, 18  
 opaque\_keys.edx.keys  
   module, 3  
 opaque\_keys.edx.locations  
   module, 6  
 opaque\_keys.edx.locator  
   module, 8  
 OpaqueKey (class in *opaque\_keys*), 18  
 OpaqueKeyMetaclass (class in *opaque\_keys*), 19  
 org (*opaque\_keys.edx.keys.CourseKey* property), 4  
 org (*opaque\_keys.edx.locations.SlashSeparatedCourseKey* attribute), 7  
 org (*opaque\_keys.edx.locator.BlockUsageLocator* property), 11  
 org (*opaque\_keys.edx.locator.CourseLocator* attribute), 14  
 org (*opaque\_keys.edx.locator.LibraryLocator* attribute), 15  
 org (*opaque\_keys.edx.locator.LibraryLocatorV2* attribute), 16

## P

parse\_url() (*opaque\_keys.edx.locator.BlockLocatorBase* class method), 9  
 path (*opaque\_keys.edx.keys.AssetKey* property), 3  
 path (*opaque\_keys.edx.locator.AssetLocator* property), 8

## R

replace() (*opaque\_keys.edx.locations.AssetLocation* method), 6  
 replace() (*opaque\_keys.edx.locations.Location* method), 7  
 replace() (*opaque\_keys.edx.locations.SlashSeparatedCourseKey* method), 7  
 replace() (*opaque\_keys.edx.locator.AssetLocator* method), 8  
 replace() (*opaque\_keys.edx.locator.BlockUsageLocator* method), 11  
 replace() (*opaque\_keys.edx.locator.LibraryUsageLocator* method), 16  
 replace() (*opaque\_keys.OpaqueKey* method), 19

revision (*opaque\_keys.edx.locator.BlockUsageLocator* property), 11  
 run (*opaque\_keys.edx.keys.CourseKey* property), 4  
 run (*opaque\_keys.edx.locations.SlashSeparatedCourseKey* attribute), 8  
 run (*opaque\_keys.edx.locator.BlockUsageLocator* property), 11  
 run (*opaque\_keys.edx.locator.CourseLocator* attribute), 14  
 RUN (*opaque\_keys.edx.locator.LibraryLocator* attribute), 15  
 run (*opaque\_keys.edx.locator.LibraryLocator* property), 15  
 run (*opaque\_keys.edx.locator.LibraryUsageLocator* property), 16

## S

set\_deprecated\_fallback() (*opaque\_keys.OpaqueKey* class method), 19  
 SlashSeparatedCourseKey (class in *opaque\_keys.edx.locations*), 7  
 slug (*opaque\_keys.edx.locator.LibraryLocatorV2* attribute), 16  
 SLUG\_REGEX (*opaque\_keys.edx.locator.LibraryLocatorV2* attribute), 15

## T

tag (*opaque\_keys.edx.locations.LocationBase* property), 7  
 tag (*opaque\_keys.edx.locator.AssetLocator* property), 8  
 to\_deprecated\_list\_repr() (*opaque\_keys.edx.locator.AssetLocator* method), 8  
 to\_deprecated\_son() (*opaque\_keys.edx.locator.BlockUsageLocator* method), 11  
 to\_deprecated\_son() (*opaque\_keys.edx.locator.LibraryUsageLocator* method), 16

## U

URL\_RE (*opaque\_keys.edx.locations.DeprecatedLocation* attribute), 6  
 URL\_RE (*opaque\_keys.edx.locator.BlockLocatorBase* attribute), 9  
 URL\_RE (*opaque\_keys.edx.locator.DefinitionLocator* attribute), 14  
 URL\_RE\_SOURCE (*opaque\_keys.edx.locations.DeprecatedLocation* attribute), 7  
 URL\_RE\_SOURCE (*opaque\_keys.edx.locator.BlockLocatorBase* attribute), 9  
 usage\_id (*opaque\_keys.edx.locator.LibraryUsageLocatorV2* attribute), 17

USAGE\_ID\_REGEX (opaque\_keys.edx.locator.LibraryUsageLocatorV2 attribute), 17

usage\_key (opaque\_keys.edx.keys.AsideUsageKey property), 3

UsageKey (class in opaque\_keys.edx.keys), 5

UsageKeyV2 (class in opaque\_keys.edx.keys), 5

## V

version (opaque\_keys.edx.locator.BlockUsageLocator property), 11

version (opaque\_keys.edx.locator.CourseLocator property), 14

version (opaque\_keys.edx.locator.DefinitionLocator property), 14

version (opaque\_keys.edx.locator.LibraryLocator property), 15

version (opaque\_keys.edx.locator.Locator property), 17

version\_agnostic() (opaque\_keys.edx.locator.BlockUsageLocator method), 11

version\_agnostic() (opaque\_keys.edx.locator.CourseLocator method), 14

version\_agnostic() (opaque\_keys.edx.locator.LibraryLocator method), 15

version\_agnostic() (opaque\_keys.edx.locator.LibraryUsageLocator method), 16

version\_guid (opaque\_keys.edx.locations.SlashSeparatedCourseKey attribute), 8

version\_guid (opaque\_keys.edx.locator.BlockUsageLocator property), 11

version\_guid (opaque\_keys.edx.locator.CourseLocator attribute), 14

version\_guid (opaque\_keys.edx.locator.LibraryLocator attribute), 15

VERSION\_PREFIX (opaque\_keys.edx.locator.Locator attribute), 17

VersionTree (class in opaque\_keys.edx.locator), 17