

# Package ‘geoarrow’

May 31, 2024

**Title** Extension Types for Spatial Data for Use with 'Arrow'

**Version** 0.2.0

**Description** Provides extension types and conversions to between R-native object types and 'Arrow' columnar types. This includes integration among the 'arrow', 'nanoarrow', 'sf', and 'wk' packages such that spatial metadata is preserved wherever possible. Extension type implementations ensure first-class geometry data type support in the 'arrow' and 'nanoarrow' packages.

**License** Apache License ( $\geq 2$ )

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Imports** nanoarrow ( $\geq 0.5.0$ ), wk ( $\geq 0.9.0$ )

**LinkingTo** wk

**Config/testthat/edition** 3

**URL** <https://github.com/geoarrow/geoarrow-r>

**BugReports** <https://github.com/geoarrow/geoarrow-r/issues>

**Depends** R ( $\geq 3.6.0$ )

**Suggests** arrow, R6, sf, testthat ( $\geq 3.0.0$ )

**NeedsCompilation** yes

**Author** Dewey Dunnington [aut, cre] (<<https://orcid.org/0000-0002-9415-4582>>),  
Anthony North [ctb],  
Apache Software Foundation [cph],  
Ulf Adams [cph],  
Daniel Lemire [cph],  
Joao Paulo Magalhaes [cph]

**Maintainer** Dewey Dunnington <dewey@dunnington.ca>

**Repository** CRAN

**Date/Publication** 2024-05-31 19:02:57 UTC

## R topics documented:

as_geoarrow_array	2
as_geoarrow_vctr	3
geoarrow_handle	3
geoarrow_schema_parse	4
geoarrow_wkb	5
infer_geoarrow_schema	6
na_extension_wkb	7

<b>Index</b>	<b>9</b>
--------------	----------

---

as_geoarrow_array	<i>Convert an object to a GeoArrow array</i>
-------------------	--

---

### Description

Convert an object to a GeoArrow array

### Usage

```
as_geoarrow_array(x, ..., schema = NULL)
```

```
as_geoarrow_array_stream(x, ..., schema = NULL)
```

### Arguments

x	An object
...	Passed to S3 methods
schema	A geoarrow extension schema to use as the target type

### Value

A [nanoarrow\\_array](#).

### Examples

```
as_geoarrow_array(wk::wkt("POINT (0 1)"))
```

---

as\_geoarrow\_vctr      *GeoArrow encoded arrays as R vectors*

---

**Description**

GeoArrow encoded arrays as R vectors

**Usage**

```
as_geoarrow_vctr(x, ..., schema = NULL)
```

**Arguments**

x	An object that works with <code>as_geoarrow_array_stream()</code> . Most spatial objects in R already work with this method.
...	Passed to <code>as_geoarrow_array_stream()</code>
schema	An optional schema (e.g., <code>na_extension_geoarrow()</code> ).

**Value**

A vctr of class 'geoarrow\_vctr'

**Examples**

```
as_geoarrow_vctr("POINT (0 1)")
```

---

geoarrow\_handle      *Handler/writer interface for GeoArrow arrays*

---

**Description**

Handler/writer interface for GeoArrow arrays

**Usage**

```
geoarrow_handle(x, handler, size = NA_integer_)
```

```
geoarrow_writer(schema)
```

**Arguments**

x	An object implementing <code>as_geoarrow_array_stream()</code>
handler	A <a href="#">wk handler</a>
size	The number of elements in the stream or NA if unknown
schema	A <a href="#">nanoarrow_schema</a>

**Value**

- `geoarrow_handle()`: Returns the result of handler
- `geoarrow_writer()`: Returns a [nanoarrow array](#)

**Examples**

```
geoarrow_handle(wk::xy(1:3, 2:4), wk::wk_debug_filter())
wk::wk_handle(wk::xy(1:3, 2:4), geoarrow_writer(na_extension_wkt()))
```

---

`geoarrow_schema_parse` *Inspect a GeoArrow schema*

---

**Description**

Inspect a GeoArrow schema

**Usage**

```
geoarrow_schema_parse(  
  schema,  
  extension_name = NULL,  
  infer_from_storage = FALSE  
)  
  
is_geoarrow_schema(schema)  
  
as_geoarrow_schema(schema)
```

**Arguments**

`schema` A [nanoarrow\\_schema](#)

`extension_name` An extension name to use if schema is a storage type.

`infer_from_storage` Attempt to guess an extension name if schema is not a geoarrow extension type.

**Value**

A list of parsed properties

**Examples**

```
geoarrow_schema_parse(na_extension_geoarrow("POINT"))
```

---

`geoarrow_wkb`*GeoArrow Types*

---

**Description**

These functions provide GeoArrow type definitions as zero-length vectors.

**Usage**

```
geoarrow_wkb(crs = NULL, edges = "PLANAR")
```

```
geoarrow_wkt(crs = NULL, edges = "PLANAR")
```

```
geoarrow_large_wkb(crs = NULL, edges = "PLANAR")
```

```
geoarrow_large_wkt(crs = NULL, edges = "PLANAR")
```

```
geoarrow_native(  
  geometry_type,  
  dimensions = "XY",  
  coord_type = "SEPARATE",  
  crs = NULL,  
  edges = "PLANAR"  
)
```

```
geoarrow_point(  
  dimensions = "XY",  
  coord_type = "SEPARATE",  
  crs = NULL,  
  edges = "PLANAR"  
)
```

```
geoarrow_linestring(  
  dimensions = "XY",  
  coord_type = "SEPARATE",  
  crs = NULL,  
  edges = "PLANAR"  
)
```

```
geoarrow_polygon(  
  dimensions = "XY",  
  coord_type = "SEPARATE",  
  crs = NULL,  
  edges = "PLANAR"  
)
```

```
geoarrow_multipoint(  
  dimensions = "XY",  
  coord_type = "SEPARATE",  
  crs = NULL,  
  edges = "PLANAR"  
)
```

```

    dimensions = "XY",
    coord_type = "SEPARATE",
    crs = NULL,
    edges = "PLANAR"
)

geoarrow_multilinestring(
  dimensions = "XY",
  coord_type = "SEPARATE",
  crs = NULL,
  edges = "PLANAR"
)

geoarrow_multipolygon(
  dimensions = "XY",
  coord_type = "SEPARATE",
  crs = NULL,
  edges = "PLANAR"
)

```

### Arguments

crs	An object representing a CRS. For maximum portability, it should implement <a href="#">wk::wk_crs_projjson()</a> .
edges	One of "PLANAR" or "SPHERICAL".
geometry_type	One of "POINT", "LINESTRING", "POLYGON", "MULTIPOINT", "MULTILINESTRING", "MULTIPOLYGON".
dimensions	One of "XY", "XYZ", "XYM", or "XYZM"
coord_type	One of "SEPARATE" or "INTERLEAVED"

### Value

A [geoarrow\\_vctr](#)

### Examples

```

geoarrow_wkb()
geoarrow_wkt()
geoarrow_point()

```

---

infer\_geoarrow\_schema *Infer a GeoArrow-native type from a vector*

---

### Description

Infer a GeoArrow-native type from a vector

**Usage**

```
infer_geoarrow_schema(x, ..., promote_multi = TRUE, coord_type = NULL)
```

**Arguments**

x	An object from which to infer a schema.
...	Passed to S3 methods.
promote_multi	Use TRUE to return a MULTI type when both normal and MULTI elements are in the same array.
coord_type	Specify the coordinate type to use if returning

**Value**

A [nanoarrow\\_schema](#)

**Examples**

```
infer_geoarrow_schema(wk::wkt("POINT (0 1)"))
```

---

na_extension_wkb	<i>Extension type definitions for GeoArrow extension types</i>
------------------	--

---

**Description**

Extension type definitions for GeoArrow extension types

**Usage**

```
na_extension_wkb(crs = NULL, edges = "PLANAR")
na_extension_wkt(crs = NULL, edges = "PLANAR")
na_extension_large_wkb(crs = NULL, edges = "PLANAR")
na_extension_large_wkt(crs = NULL, edges = "PLANAR")

na_extension_geoarrow(
  geometry_type,
  dimensions = "XY",
  coord_type = "SEPARATE",
  crs = NULL,
  edges = "PLANAR"
)
```

**Arguments**

crs	An object representing a CRS. For maximum portability, it should implement <a href="#">wk::wk_crs_projjson()</a> .
edges	One of "PLANAR" or "SPHERICAL".
geometry_type	One of "POINT", "LINESTRING", "POLYGON", "MULTIPOINT", "MULTILINESTRING", "MULTIPOLYGON".
dimensions	One of "XY", "XYZ", "XYM", or "XYZM"
coord_type	One of "SEPARATE" or "INTERLEAVED"

**Value**

A [nanoarrow\\_schema](#).

**Examples**

```
na_extension_wkb(crs = "OGC:CRS84")
na_extension_geoarrow("POINT")
```



# Index

as\_geoarrow\_array, 2  
as\_geoarrow\_array\_stream  
    (as\_geoarrow\_array), 2  
as\_geoarrow\_array\_stream(), 3  
as\_geoarrow\_schema  
    (geoarrow\_schema\_parse), 4  
as\_geoarrow\_vctr, 3

geoarrow\_handle, 3  
geoarrow\_large\_wkb (geoarrow\_wkb), 5  
geoarrow\_large\_wkt (geoarrow\_wkb), 5  
geoarrow\_linestring (geoarrow\_wkb), 5  
geoarrow\_multilinestring  
    (geoarrow\_wkb), 5  
geoarrow\_multipoint (geoarrow\_wkb), 5  
geoarrow\_multipolygon (geoarrow\_wkb), 5  
geoarrow\_native (geoarrow\_wkb), 5  
geoarrow\_point (geoarrow\_wkb), 5  
geoarrow\_polygon (geoarrow\_wkb), 5  
geoarrow\_schema\_parse, 4  
geoarrow\_vctr, 6  
geoarrow\_wkb, 5  
geoarrow\_wkt (geoarrow\_wkb), 5  
geoarrow\_writer (geoarrow\_handle), 3

infer\_geoarrow\_schema, 6  
is\_geoarrow\_schema  
    (geoarrow\_schema\_parse), 4

na\_extension\_geoarrow  
    (na\_extension\_wkb), 7  
na\_extension\_geoarrow(), 3  
na\_extension\_large\_wkb  
    (na\_extension\_wkb), 7  
na\_extension\_large\_wkt  
    (na\_extension\_wkb), 7  
na\_extension\_wkb, 7  
na\_extension\_wkt (na\_extension\_wkb), 7  
nanoarrow\_array, 4  
nanoarrow\_array, 2

nanoarrow\_schema, 3, 4, 7, 8  
wk handler, 3  
wk::wk\_crs\_projjson(), 6, 8