

NAME

XML::LibXML::AttributeHash – tie an XML::LibXML::Element to a hash to access its attributes

SYNOPSIS

```
tie my %hash, 'XML::LibXML::AttributeHash', $element;
$hash{'href'} = 'http://example.com/';
print $element->getAttribute('href') . "\n";
```

DESCRIPTION

This class allows an element’s attributes to be accessed as if they were a plain old Perl hash. Attribute names become hash keys. Namespaced attributes are keyed using Clark notation.

```
my $XLINK = 'http://www.w3.org/1999/xlink';
tie my %hash, 'XML::LibXML::AttributeHash', $element;
$hash{"{$XLINK}href"} = 'http://localhost/';
print $element->getAttributeNS($XLINK, 'href') . "\n";
```

There is rarely any need to use XML::LibXML::AttributeHash directly. In general, it is possible to take advantage of XML::LibXML::Element’s overloading. The example in the SYNOPSIS could have been written:

```
$element->{'href'} = 'http://example.com/';
print $element->getAttribute('href') . "\n";
```

The tie interface allows the passing of additional arguments to XML::LibXML::AttributeHash:

```
tie my %hash, 'XML::LibXML::AttributeHash', $element, %args;
```

Currently only one argument is supported, the boolean “weaken” which (if true) indicates that the tied object’s reference to the element should be a weak reference. This is used by XML::LibXML::Element’s overloading. The “weaken” argument is ignored if you don’t have a working Scalar::Util::weaken.