NAME

XML::LibXML::Namespace – XML::LibXML Namespace Implementation

SYNOPSIS

```
use XML::LibXML;
# Only methods specific to Namespace nodes are listed here,
# see the XML::LibXML::Node manpage for other methods

my $ns = XML::LibXML::Namespace->new($nsURI);
print $ns->nodeName();
print $ns->name();
$localname = $ns->getLocalName();
print $ns->getData();
print $ns->getValue();
print $ns->value();
$known_uri = $ns->getNamespaceURI();
$known_prefix = $ns->getPrefix();
$key = $ns->unique_key();
```

DESCRIPTION

Namespace nodes are returned by both `$element`–>findnodes('namespace::foo') or by `$node`–>**getNamespaces**().

The namespace node API is not part of any current DOM API, and so it is quite minimal. It should be noted that namespace nodes are *not* a sub class of XML::LibXML::Node, however Namespace nodes act a lot like attribute nodes, and similarly named methods will return what you would expect if you treated the namespace node as an attribute. Note that in order to fix several inconsistencies between the API and the documentation, the behavior of some functions have been changed in 1.64.

METHODS

new

```
my $ns = XML::LibXML::Namespace->new($nsURI);
```

Creates a new Namespace node. Note that this is not a 'node' as an attribute or an element node. Therefore you can't do call all XML::LibXML::Node Functions. All functions available for this node are listed below.

Optionally you can pass the prefix to the namespace constructor. If this second parameter is omitted you will create a so called default namespace. Note, the newly created namespace is not bound to any document or node, therefore you should not expect it to be available in an existing document.

declaredURI

Returns the URI for this namespace.

declaredPrefix

Returns the prefix for this namespace.

nodeName

```
print $ns->nodeName();
```

Returns "xmlns:prefix", where prefix is the prefix for this namespace.

name

```
print $ns->name();
```

Alias for **nodeName()**

getLocalName

```
$localname = $ns->getLocalName();
```

Returns the local name of this node as if it were an attribute, that is, the prefix associated with the namespace.

getData

```
print $ns->getData();
```

Returns the URI of the namespace, i.e. the value of this node as if it were an attribute.

getValue

```
print $ns->getValue();
```

Alias for **getData()**

value

```
print $ns->value();
```

Alias for **getData()**

getNamespaceURI

```
$known_uri = $ns->getNamespaceURI();
```

Returns the string “http://www.w3.org/2000/xmlns/”

getPrefix

```
$known_prefix = $ns->getPrefix();
```

Returns the string “xmlns”

unique_key

```
$key = $ns->unique_key();
```

This method returns a key guaranteed to be unique for this namespace, and to always be the same value for this namespace. Two namespace objects return the same key if and only if they have the same prefix and the same URI. The returned key value is useful as a key in hashes.

AUTHORS

Matt Sergeant, Christian Glahn, Petr Pajas

VERSION

2.0134

COPYRIGHT

2001−2007, AxKit.com Ltd.

2002−2006, Christian Glahn.

2006−2009, Petr Pajas.

LICENSE

This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.