NAME

XML::LibXML::Dtd – XML::LibXML DTD Handling

SYNOPSIS

```
  use XML::LibXML;

  $dtd = XML::LibXML::Dtd->new($public_id, $system_id);
  $dtd = XML::LibXML::Dtd->parse_string($dtd_str);
  $publicId = $dtd->getName();
  $publicId = $dtd->publicId();
  $systemId = $dtd->systemId();
```

DESCRIPTION

This class holds a DTD. You may parse a DTD from either a string, or from an external SYSTEM identifier.

No support is available as yet for parsing from a filehandle.

XML::LibXML::Dtd is a sub-class of XML::LibXML::Node, so all the methods available to nodes (particularly **toString**()) are available to Dtd objects.

METHODS

new

```
  $dtd = XML::LibXML::Dtd->new($public_id, $system_id);
```

Parse a DTD from the system identifier, and return a DTD object that you can pass to `$doc->`**is_valid()** or `$doc->`**validate()**.

```
  my $dtd = XML::LibXML::Dtd->new(
                        "SOME // Public / ID / 1.0",
                        "test.dtd"
                                  );
   my $doc = XML::LibXML->new->parse_file("test.xml");
   $doc->validate($dtd);
```

parse_string

```
  $dtd = XML::LibXML::Dtd->parse_string($dtd_str);
```

The same as **new()** above, except you can parse a DTD from a string. Note that parsing from string may fail if the DTD contains external parametric-entity references with relative URLs.

getName

```
  $publicId = $dtd->getName();
```

Returns the name of DTD; i.e., the name immediately following the DOCTYPE keyword.

publicId

```
  $publicId = $dtd->publicId();
```

Returns the public identifier of the external subset.

systemId

```
  $systemId = $dtd->systemId();
```

Returns the system identifier of the external subset.

AUTHORS

Matt Sergeant, Christian Glahn, Petr Pajas

VERSION

2.0134

COPYRIGHT

2001–2007, AxKit.com Ltd.

2002–2006, Christian Glahn.

LICENSE