

**NAME**

XML::LibXML::Common – Constants and Character Encoding Routines

**SYNOPSIS**

```
use XML::LibXML::Common;

$encodedstring = encodeToUTF8( $name_of_encoding, $string_to_encode );
$decodedstring = decodeFromUTF8($name_of_encoding, $string_to_decode );
```

**DESCRIPTION**

XML::LibXML::Common defines constants for all node types and provides interface to libxml2 charset conversion functions.

Since XML::LibXML use their own node type definitions, one may want to use XML::LibXML::Common in its compatibility mode:

**Exporter TAGS**

```
use XML::LibXML::Common qw(:libxml);
```

:libxml tag will use the XML::LibXML Compatibility mode, which defines the old 'XML\_' node-type definitions.

```
use XML::LibXML::Common qw(:gdome);
```

:gdome tag will use the XML::GDOME Compatibility mode, which defines the old 'GDOME\_' node-type definitions.

```
use XML::LibXML::Common qw(:w3c);
```

This uses the nodetype definition names as specified for DOM.

```
use XML::LibXML::Common qw(:encoding);
```

This tag can be used to export only the charset encoding functions of XML::LibXML::Common.

**Exports**

By default the W3 definitions as defined in the DOM specifications and the encoding functions are exported by XML::LibXML::Common.

**Encoding functions**

To encode or decode a string to or from UTF-8, XML::LibXML::Common exports two functions, which provide an interface to the encoding support in libxml2. Which encodings are supported by these functions depends on how libxml2 was compiled. UTF-16 is always supported and on most installations, ISO encodings are supported as well.

This interface was useful for older versions of Perl. Since Perl >= 5.8 provides similar functions via the Encode module, it is probably a good idea to use those instead.

**encodeToUTF8**

```
$encodedstring = encodeToUTF8( $name_of_encoding, $string_to_encode );
```

The function will convert a byte string from the specified encoding to an UTF-8 encoded character string.

**decodeToUTF8**

```
$decodedstring = decodeFromUTF8($name_of_encoding, $string_to_decode );
```

This function converts an UTF-8 encoded character string to a specified encoding. Note that the conversion can raise an error if the given string contains characters that cannot be represented in the target encoding.

Both these functions report their errors on the standard error. If an error occurs the function will **croak()**. To catch the error information it is required to call the encoding function from within an eval block in order to prevent the entire script from being stopped on encoding error.

**A note on history**

Before XML::LibXML 1.70, this class was available as a separate CPAN distribution, intended to provide functionality shared between XML::LibXML, XML::GDOME, and possibly other modules. Since there seems to be no progress in this direction, we decided to merge XML::LibXML::Common 0.13 and XML::LibXML 1.70 to one CPAN distribution.

The merge also naturally eliminates a practical and urgent problem experienced by many XML::LibXML users on certain platforms, namely mysterious misbehavior of XML::LibXML occurring if the installed (often pre-packaged) version of XML::LibXML::Common was compiled against an older version of libxml2 than XML::LibXML.

**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.