

Docs » Data Handling Category » XML Project » XML Document

XML Document

XmlDocument Class

Anonymous · 09/09/2026

XmlDocument Class

The `XmlDocument` class extends PHP's native `DOMDocument` with additional functionality for XML manipulation, transformation, and querying. This class is the core component for working with XML documents in the Derafu XML library.

Key Features

- Extended XML loading with character encoding handling.
- Simplified access to XML metadata (namespace, schema).
- Canonicalization support with encoding options.
- XPath query integration.
- Array conversion capabilities.
- Signature node handling for XML digital signatures.

Basic Usage

Creating a New Document

```
use Derafu\Xml\XmlDocument;

// Create with default version (1.0) and encoding (ISO-8859-1).
$document = new XmlDocument();

// Or specify version and encoding.
$document = new XmlDocument('1.0', 'UTF-8');
```

Loading XML Content

```
// Load from a string.
$xmlString = '<root><element>value</element></root>';
$document->loadXml($xmlString);

// The method handles encoding conversion automatically.
$utf8Xml = '<?xml version="1.0"
```

```
encoding="UTF-8"?><root><element>Árbol</element></root>';  
$document->loadXml($utf8Xml); // Will convert to ISO-8859-1 if needed.
```

Accessing Document Information

```
// Get the root element name.  
$rootName = $document->getName(); // e.g., "root"  
  
// Get the XML namespace (if any).  
$namespace = $document->getNamespace(); // e.g., "http://example.com"  
  
// Get the schema location (if any).  
$schema = $document->getSchema(); // e.g., "schema.xsd"
```

Saving and Serializing

```
// Get the complete XML document with declaration.  
$xmlString = $document->saveXml();  
  
// Get only the XML content without the declaration.  
$xmlContent = $document->getXml();  
  
// Get the canonicalized (C14N) version.  
$canonXml = $document->C14N();  
  
// Get canonicalized version with ISO-8859-1 encoding.  
$isoCanonXml = $document->C14NWithIso88591Encoding();  
  
// Get flattened canonicalized version (whitespace removed between tags).  
$flatXml = $document->C14NWithIso88591EncodingFlattened();
```

XPath Querying

The XmlDocument class has integrated XPath querying capabilities:

```
// Execute an XPath query and get result as string or array.  
$result = $document->query('/root/element');  
  
// Get DOMNodeList from an XPath query.  
$nodes = $document->getNodes('/root/element');  
  
// Use parameters in XPath queries.  
$params = ['id' => '123'];  
$result = $document->query('/root/element[@id=:id]', $params);
```

Array Conversion

```
// Convert the entire document to an array.
$array = $document->toArray();

// Access a specific part using dot notation.
$value = $document->get('root.element');

// With a default value if not found.
$value = $document->get('root.missing', 'default value');
```

Working with XML Digital Signatures

If the document contains an XML digital signature, you can extract it:

```
// Get the signature node as XML.
$signatureXml = $document->getSignatureNodeXml();

// Returns null if no signature is present.
if ($signatureXml !== null) {
    // Process signature...
}
```

Handling Special Characters and Entities

The `XmlDocument` class works with the `XmlHelper` to properly handle special characters and entities:

```
// When saving XML, entities are handled correctly.
$document->loadXml('<root><element>Text with & < > " \'</element></root>');
$xml = $document->saveXml();
// Produces: <root><element>Text with &amp; &lt; &gt; &quot; &apos;</element></root>
```

Advanced Canonicalization

Working with Specific Nodes

You can apply canonicalization to specific parts of the document:

```
// Canonicalize only a portion of the document.
$canonXml = $document->C14NWithIso88591Encoding('/root/section');
```

Differences Between Canonicalization Methods

The class offers several canonicalization methods:

1. **C14N()**: Standard canonicalization, always outputs UTF-8.
2. **C14NWithIso88591Encoding()**: Canonicalization with conversion to ISO-8859-1.
3. **C14NWithIso88591EncodingFlattened()**: ISO-8859-1 canonicalization with whitespace removal.

These methods are particularly useful when working with digital signatures or when XML needs to be processed by systems with specific encoding requirements.

Error Handling

The `loadXml()` method throws descriptive exceptions when it encounters errors:

```
use Derafu\Xml\Exception\XmlException;

try {
    $document->loadXml($potentiallyInvalidXml);
} catch (XmlException $e) {
    echo "Error loading XML: " . $e->getMessage();
    $errors = $e->getErrors(); // Get detailed error information if available.
}
```

Performance Considerations

- When working with large XML documents, prefer using XPath queries to extract specific sections rather than converting the entire document to an array.
- The canonicalization methods perform more processing, so use them only when needed.
- The `getXml()` method is more efficient than `saveXml()` when you only need the XML content without the declaration.

Last updated on 09/09/2026