

[Overview](#) **Package** [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[PREV PACKAGE](#) [NEXT PACKAGE](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)*Java™ 2 Platform  
Std. Ed. v1.4.2*

## Package org.w3c.dom

Provides the interfaces for the Document Object Model (DOM) which is a component API of the [Java API for XML Processing](#).

See:

[Description](#)

### Interface Summary

|                                   |                                                                                                                                                                                                           |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">Attr</a>              | The <code>Attr</code> interface represents an attribute in an <code>Element</code> object.                                                                                                                |
| <a href="#">CDATASection</a>      | CDATA sections are used to escape blocks of text containing characters that would otherwise be regarded as markup.                                                                                        |
| <a href="#">CharacterData</a>     | The <code>CharacterData</code> interface extends <code>Node</code> with a set of attributes and methods for accessing character data in the DOM.                                                          |
| <a href="#">Comment</a>           | This interface inherits from <code>CharacterData</code> and represents the content of a comment, i.e., all the characters between the starting ' <code>&lt;!--</code> ' and ending ' <code>&gt;</code> '. |
| <a href="#">Document</a>          | The <code>Document</code> interface represents the entire HTML or XML document.                                                                                                                           |
| <a href="#">DocumentFragment</a>  | <code>DocumentFragment</code> is a "lightweight" or "minimal" <code>Document</code> object.                                                                                                               |
| <a href="#">DocumentType</a>      | Each <code>Document</code> has a <code>doctype</code> attribute whose value is either <code>null</code> or a <code>DocumentType</code> object.                                                            |
| <a href="#">DOMImplementation</a> | The <code>DOMImplementation</code> interface provides a number of methods for performing operations that are independent of any particular instance of the document object model.                         |
| <a href="#">Element</a>           | The <code>Element</code> interface represents an element in an HTML or XML document.                                                                                                                      |
| <a href="#">Entity</a>            | This interface represents an entity, either parsed or unparsed, in an XML document.                                                                                                                       |
| <a href="#">EntityReference</a>   | <code>EntityReference</code> objects may be inserted into the structure model when an entity reference is in the source document, or when the user wishes to insert an entity reference.                  |
| <a href="#">NamedNodeMap</a>      | Objects implementing the <code>NamedNodeMap</code> interface are used to represent collections of nodes that can be accessed by name.                                                                     |
| <a href="#">Node</a>              | The <code>Node</code> interface is the primary datatype for the entire Document Object Model.                                                                                                             |
| <a href="#">NodeList</a>          | The <code>NodeList</code> interface provides the abstraction of an ordered collection of nodes, without defining or constraining how this collection is implemented.                                      |
| <a href="#">Notation</a>          | This interface represents a notation declared in the DTD.                                                                                                                                                 |

|                                       |                                                                                                                                                                     |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">ProcessingInstruction</a> | The ProcessingInstruction interface represents a "processing instruction", used in XML as a way to keep processor-specific information in the text of the document. |
| <a href="#">Text</a>                  | The Text interface inherits from CharacterData and represents the textual content (termed character data in XML) of an Element or Attr.                             |

## Exception Summary

|                              |                                                                                                                                                                                                                              |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">DOMException</a> | DOM operations only raise exceptions in "exceptional" circumstances, i.e., when an operation is impossible to perform (either for logical reasons, because data is lost, or because the implementation has become unstable). |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## Package org.w3c.dom Description

Provides the interfaces for the Document Object Model (DOM) which is a component API of the [Java API for XML Processing](#). The Document Object Model Level 2 Core API allows programs to dynamically access and update the content and structure of documents. See the [specification](#) for more information.

Since:

JDK1.4

---

**[Overview](#)** **[Package](#)** [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV PACKAGE](#) [NEXT PACKAGE](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

*Java™ 2 Platform  
Std. Ed. v1.4.2*

[Submit a bug or feature](#)

For further API reference and developer documentation, see [Java 2 SDK SE Developer Documentation](#). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright © 2003, 2010 Oracle and/or its affiliates. All rights reserved. Use is subject to [license terms](#). Also see the [documentation redistribution policy](#).