

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

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

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

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	
Attr	The Attr interface represents an attribute in an Element object.
CDATASection	CDATA sections are used to escape blocks of text containing characters that would otherwise be regarded as markup.
CharacterData	The CharacterData interface extends Node with a set of attributes and methods for accessing character data in the DOM.
Comment	This interface inherits from CharacterData and represents the content of a comment, i.e., all the characters between the starting '<!--' and ending '-->'.
Document	The Document interface represents the entire HTML or XML document.
DocumentFragment	DocumentFragment is a "lightweight" or "minimal" Document object.
DocumentType	Each Document has a doctype attribute whose value is either null or a DocumentType object.
DOMConfiguration	The DOMConfiguration interface represents the configuration of a document and maintains a table of recognized parameters.
DOMError	DOMError is an interface that describes an error.
DOMErrorHandler	DOMErrorHandler is a callback interface that the DOM implementation can call when reporting errors that happens while processing XML data, or when doing some other processing (e.g.
DOMImplementation	The DOMImplementation interface provides a number of methods for performing operations that are independent of any particular instance of the document object model.
DOMImplementationList	The DOMImplementationList interface provides the abstraction of an ordered collection of DOM implementations, without defining or constraining how this collection is implemented.
DOMImplementationSource	This interface permits a DOM implementer to supply one or more implementations, based upon requested features and versions, as specified in .
DOMLocator	DOMLocator is an interface that describes a location (e.g.
DOMStringList	The DOMStringList interface provides the abstraction of an ordered collection of DOMString values, without defining or constraining how this collection is implemented.
Element	The Element interface represents an element in an HTML or XML document.
Entity	This interface represents a known entity, either parsed or unparsed, in an XML document.
EntityReference	EntityReference nodes may be used to represent an entity reference in the tree.
NamedNodeMap	Objects implementing the NamedNodeMap interface are used to represent collections of nodes that can be accessed by name.
NameList	The NameList interface provides the abstraction of an ordered collection of parallel pairs of name and namespace values (which could be null values), without defining or constraining how this collection is implemented.
Node	The Node interface is the primary datatype for the entire Document Object Model.
NodeList	The NodeList interface provides the abstraction of an ordered collection of nodes, without defining or constraining how this collection is implemented.
Notation	This interface represents a notation declared in the DTD.
ProcessingInstruction	The ProcessingInstruction interface represents a "processing instruction", used in XML as a way to keep processor-specific information in the text of the document.
Text	The Text interface inherits from CharacterData and represents the textual content (termed character data in XML) of an Element or Attr.
TypeInfo	The TypeInfo interface represents a type referenced from Element or Attr nodes, specified in the schemas associated with the document.
UserDataHandler	When associating an object to a key on a node using Node.setUserData() the application can provide a handler that gets called when the node the object is associated to is being cloned, imported, or renamed.

Exception Summary

DOMException	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
Standard Ed. 5.0

[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 2004 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#). Also see the [documentation redistribution policy](#).