| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ContentHandler.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/org/xml/sax/Attributes.html)   [**NEXT CLASS**](http://docs.google.com/org/xml/sax/DocumentHandler.html) | [**FRAMES**](http://docs.google.com/index.html?org/xml/sax/ContentHandler.html)    [**NO FRAMES**](http://docs.google.com/ContentHandler.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

## **org.xml.sax**

Interface ContentHandler

**All Known Subinterfaces:** [TemplatesHandler](http://docs.google.com/javax/xml/transform/sax/TemplatesHandler.html), [TransformerHandler](http://docs.google.com/javax/xml/transform/sax/TransformerHandler.html), [UnmarshallerHandler](http://docs.google.com/javax/xml/bind/UnmarshallerHandler.html) **All Known Implementing Classes:** [DefaultHandler](http://docs.google.com/org/xml/sax/helpers/DefaultHandler.html), [DefaultHandler2](http://docs.google.com/org/xml/sax/ext/DefaultHandler2.html), [ValidatorHandler](http://docs.google.com/javax/xml/validation/ValidatorHandler.html), [XMLFilterImpl](http://docs.google.com/org/xml/sax/helpers/XMLFilterImpl.html), [XMLReaderAdapter](http://docs.google.com/org/xml/sax/helpers/XMLReaderAdapter.html)

public interface **ContentHandler**

Receive notification of the logical content of a document.

*This module, both source code and documentation, is in the Public Domain, and comes with* ***NO WARRANTY****.* See <http://www.saxproject.org> for further information.

This is the main interface that most SAX applications implement: if the application needs to be informed of basic parsing events, it implements this interface and registers an instance with the SAX parser using the [setContentHandler](http://docs.google.com/org/xml/sax/XMLReader.html#setContentHandler(org.xml.sax.ContentHandler)) method. The parser uses the instance to report basic document-related events like the start and end of elements and character data.

The order of events in this interface is very important, and mirrors the order of information in the document itself. For example, all of an element's content (character data, processing instructions, and/or subelements) will appear, in order, between the startElement event and the corresponding endElement event.

This interface is similar to the now-deprecated SAX 1.0 DocumentHandler interface, but it adds support for Namespaces and for reporting skipped entities (in non-validating XML processors).

Implementors should note that there is also a ContentHandler class in the java.net package; that means that it's probably a bad idea to do

import java.net.\*;  
 import org.xml.sax.\*;

In fact, "import ...\*" is usually a sign of sloppy programming anyway, so the user should consider this a feature rather than a bug.

**Since:** SAX 2.0 **See Also:**[XMLReader](http://docs.google.com/org/xml/sax/XMLReader.html), [DTDHandler](http://docs.google.com/org/xml/sax/DTDHandler.html), [ErrorHandler](http://docs.google.com/org/xml/sax/ErrorHandler.html)

| **Method Summary** | |
| --- | --- |
| void | [**characters**](http://docs.google.com/org/xml/sax/ContentHandler.html#characters(char%5B%5D,%20int,%20int))(char[] ch, int start, int length)            Receive notification of character data. |
| void | [**endDocument**](http://docs.google.com/org/xml/sax/ContentHandler.html#endDocument())()            Receive notification of the end of a document. |
| void | [**endElement**](http://docs.google.com/org/xml/sax/ContentHandler.html#endElement(java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) uri, [String](http://docs.google.com/java/lang/String.html) localName, [String](http://docs.google.com/java/lang/String.html) qName)            Receive notification of the end of an element. |
| void | [**endPrefixMapping**](http://docs.google.com/org/xml/sax/ContentHandler.html#endPrefixMapping(java.lang.String))([String](http://docs.google.com/java/lang/String.html) prefix)            End the scope of a prefix-URI mapping. |
| void | [**ignorableWhitespace**](http://docs.google.com/org/xml/sax/ContentHandler.html#ignorableWhitespace(char%5B%5D,%20int,%20int))(char[] ch, int start, int length)            Receive notification of ignorable whitespace in element content. |
| void | [**processingInstruction**](http://docs.google.com/org/xml/sax/ContentHandler.html#processingInstruction(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) target, [String](http://docs.google.com/java/lang/String.html) data)            Receive notification of a processing instruction. |
| void | [**setDocumentLocator**](http://docs.google.com/org/xml/sax/ContentHandler.html#setDocumentLocator(org.xml.sax.Locator))([Locator](http://docs.google.com/org/xml/sax/Locator.html) locator)            Receive an object for locating the origin of SAX document events. |
| void | [**skippedEntity**](http://docs.google.com/org/xml/sax/ContentHandler.html#skippedEntity(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Receive notification of a skipped entity. |
| void | [**startDocument**](http://docs.google.com/org/xml/sax/ContentHandler.html#startDocument())()            Receive notification of the beginning of a document. |
| void | [**startElement**](http://docs.google.com/org/xml/sax/ContentHandler.html#startElement(java.lang.String,%20java.lang.String,%20java.lang.String,%20org.xml.sax.Attributes))([String](http://docs.google.com/java/lang/String.html) uri, [String](http://docs.google.com/java/lang/String.html) localName, [String](http://docs.google.com/java/lang/String.html) qName, [Attributes](http://docs.google.com/org/xml/sax/Attributes.html) atts)            Receive notification of the beginning of an element. |
| void | [**startPrefixMapping**](http://docs.google.com/org/xml/sax/ContentHandler.html#startPrefixMapping(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) prefix, [String](http://docs.google.com/java/lang/String.html) uri)            Begin the scope of a prefix-URI Namespace mapping. |

| **Method Detail** |
| --- |

### setDocumentLocator

void **setDocumentLocator**([Locator](http://docs.google.com/org/xml/sax/Locator.html) locator)

Receive an object for locating the origin of SAX document events.

SAX parsers are strongly encouraged (though not absolutely required) to supply a locator: if it does so, it must supply the locator to the application by invoking this method before invoking any of the other methods in the ContentHandler interface.

The locator allows the application to determine the end position of any document-related event, even if the parser is not reporting an error. Typically, the application will use this information for reporting its own errors (such as character content that does not match an application's business rules). The information returned by the locator is probably not sufficient for use with a search engine.

Note that the locator will return correct information only during the invocation SAX event callbacks after [startDocument](http://docs.google.com/org/xml/sax/ContentHandler.html#startDocument()) returns and before [endDocument](http://docs.google.com/org/xml/sax/ContentHandler.html#endDocument()) is called. The application should not attempt to use it at any other time.

**Parameters:**locator - an object that can return the location of any SAX document event**See Also:**[Locator](http://docs.google.com/org/xml/sax/Locator.html)

### startDocument

void **startDocument**()  
 throws [SAXException](http://docs.google.com/org/xml/sax/SAXException.html)

Receive notification of the beginning of a document.

The SAX parser will invoke this method only once, before any other event callbacks (except for [setDocumentLocator](http://docs.google.com/org/xml/sax/ContentHandler.html#setDocumentLocator(org.xml.sax.Locator))).

**Throws:** [SAXException](http://docs.google.com/org/xml/sax/SAXException.html) - any SAX exception, possibly wrapping another exception**See Also:**[endDocument()](http://docs.google.com/org/xml/sax/ContentHandler.html#endDocument())

### endDocument

void **endDocument**()  
 throws [SAXException](http://docs.google.com/org/xml/sax/SAXException.html)

Receive notification of the end of a document.

**There is an apparent contradiction between the documentation for this method and the documentation for** [**ErrorHandler.fatalError(org.xml.sax.SAXParseException)**](http://docs.google.com/org/xml/sax/ErrorHandler.html#fatalError(org.xml.sax.SAXParseException))**. Until this ambiguity is resolved in a future major release, clients should make no assumptions about whether endDocument() will or will not be invoked when the parser has reported a fatalError() or thrown an exception.**

The SAX parser will invoke this method only once, and it will be the last method invoked during the parse. The parser shall not invoke this method until it has either abandoned parsing (because of an unrecoverable error) or reached the end of input.

**Throws:** [SAXException](http://docs.google.com/org/xml/sax/SAXException.html) - any SAX exception, possibly wrapping another exception**See Also:**[startDocument()](http://docs.google.com/org/xml/sax/ContentHandler.html#startDocument())

### startPrefixMapping

void **startPrefixMapping**([String](http://docs.google.com/java/lang/String.html) prefix,  
 [String](http://docs.google.com/java/lang/String.html) uri)  
 throws [SAXException](http://docs.google.com/org/xml/sax/SAXException.html)

Begin the scope of a prefix-URI Namespace mapping.

The information from this event is not necessary for normal Namespace processing: the SAX XML reader will automatically replace prefixes for element and attribute names when the http://xml.org/sax/features/namespaces feature is true (the default).

There are cases, however, when applications need to use prefixes in character data or in attribute values, where they cannot safely be expanded automatically; the start/endPrefixMapping event supplies the information to the application to expand prefixes in those contexts itself, if necessary.

Note that start/endPrefixMapping events are not guaranteed to be properly nested relative to each other: all startPrefixMapping events will occur immediately before the corresponding [startElement](http://docs.google.com/org/xml/sax/ContentHandler.html#startElement(java.lang.String,%20java.lang.String,%20java.lang.String,%20org.xml.sax.Attributes)) event, and all [endPrefixMapping](http://docs.google.com/org/xml/sax/ContentHandler.html#endPrefixMapping(java.lang.String)) events will occur immediately after the corresponding [endElement](http://docs.google.com/org/xml/sax/ContentHandler.html#endElement(java.lang.String,%20java.lang.String,%20java.lang.String)) event, but their order is not otherwise guaranteed.

There should never be start/endPrefixMapping events for the "xml" prefix, since it is predeclared and immutable.

**Parameters:**prefix - the Namespace prefix being declared. An empty string is used for the default element namespace, which has no prefix.uri - the Namespace URI the prefix is mapped to **Throws:** [SAXException](http://docs.google.com/org/xml/sax/SAXException.html) - the client may throw an exception during processing**See Also:**[endPrefixMapping(java.lang.String)](http://docs.google.com/org/xml/sax/ContentHandler.html#endPrefixMapping(java.lang.String)), [startElement(java.lang.String, java.lang.String, java.lang.String, org.xml.sax.Attributes)](http://docs.google.com/org/xml/sax/ContentHandler.html#startElement(java.lang.String,%20java.lang.String,%20java.lang.String,%20org.xml.sax.Attributes))

### endPrefixMapping

void **endPrefixMapping**([String](http://docs.google.com/java/lang/String.html) prefix)  
 throws [SAXException](http://docs.google.com/org/xml/sax/SAXException.html)

End the scope of a prefix-URI mapping.

See [startPrefixMapping](http://docs.google.com/org/xml/sax/ContentHandler.html#startPrefixMapping(java.lang.String,%20java.lang.String)) for details. These events will always occur immediately after the corresponding [endElement](http://docs.google.com/org/xml/sax/ContentHandler.html#endElement(java.lang.String,%20java.lang.String,%20java.lang.String)) event, but the order of [endPrefixMapping](http://docs.google.com/org/xml/sax/ContentHandler.html#endPrefixMapping(java.lang.String)) events is not otherwise guaranteed.

**Parameters:**prefix - the prefix that was being mapped. This is the empty string when a default mapping scope ends. **Throws:** [SAXException](http://docs.google.com/org/xml/sax/SAXException.html) - the client may throw an exception during processing**See Also:**[startPrefixMapping(java.lang.String, java.lang.String)](http://docs.google.com/org/xml/sax/ContentHandler.html#startPrefixMapping(java.lang.String,%20java.lang.String)), [endElement(java.lang.String, java.lang.String, java.lang.String)](http://docs.google.com/org/xml/sax/ContentHandler.html#endElement(java.lang.String,%20java.lang.String,%20java.lang.String))

### startElement

void **startElement**([String](http://docs.google.com/java/lang/String.html) uri,  
 [String](http://docs.google.com/java/lang/String.html) localName,  
 [String](http://docs.google.com/java/lang/String.html) qName,  
 [Attributes](http://docs.google.com/org/xml/sax/Attributes.html) atts)  
 throws [SAXException](http://docs.google.com/org/xml/sax/SAXException.html)

Receive notification of the beginning of an element.

The Parser will invoke this method at the beginning of every element in the XML document; there will be a corresponding [endElement](http://docs.google.com/org/xml/sax/ContentHandler.html#endElement(java.lang.String,%20java.lang.String,%20java.lang.String)) event for every startElement event (even when the element is empty). All of the element's content will be reported, in order, before the corresponding endElement event.

This event allows up to three name components for each element:

1. the Namespace URI;
2. the local name; and
3. the qualified (prefixed) name.

Any or all of these may be provided, depending on the values of the http://xml.org/sax/features/namespaces and the http://xml.org/sax/features/namespace-prefixes properties:

* the Namespace URI and local name are required when the namespaces property is true (the default), and are optional when the namespaces property is false (if one is specified, both must be);
* the qualified name is required when the namespace-prefixes property is true, and is optional when the namespace-prefixes property is false (the default).

Note that the attribute list provided will contain only attributes with explicit values (specified or defaulted): #IMPLIED attributes will be omitted. The attribute list will contain attributes used for Namespace declarations (xmlns\* attributes) only if the http://xml.org/sax/features/namespace-prefixes property is true (it is false by default, and support for a true value is optional).

Like [characters()](http://docs.google.com/org/xml/sax/ContentHandler.html#characters(char%5B%5D,%20int,%20int)), attribute values may have characters that need more than one char value.

**Parameters:**uri - the Namespace URI, or the empty string if the element has no Namespace URI or if Namespace processing is not being performedlocalName - the local name (without prefix), or the empty string if Namespace processing is not being performedqName - the qualified name (with prefix), or the empty string if qualified names are not availableatts - the attributes attached to the element. If there are no attributes, it shall be an empty Attributes object. The value of this object after startElement returns is undefined **Throws:** [SAXException](http://docs.google.com/org/xml/sax/SAXException.html) - any SAX exception, possibly wrapping another exception**See Also:**[endElement(java.lang.String, java.lang.String, java.lang.String)](http://docs.google.com/org/xml/sax/ContentHandler.html#endElement(java.lang.String,%20java.lang.String,%20java.lang.String)), [Attributes](http://docs.google.com/org/xml/sax/Attributes.html), [AttributesImpl](http://docs.google.com/org/xml/sax/helpers/AttributesImpl.html)

### endElement

void **endElement**([String](http://docs.google.com/java/lang/String.html) uri,  
 [String](http://docs.google.com/java/lang/String.html) localName,  
 [String](http://docs.google.com/java/lang/String.html) qName)  
 throws [SAXException](http://docs.google.com/org/xml/sax/SAXException.html)

Receive notification of the end of an element.

The SAX parser will invoke this method at the end of every element in the XML document; there will be a corresponding [startElement](http://docs.google.com/org/xml/sax/ContentHandler.html#startElement(java.lang.String,%20java.lang.String,%20java.lang.String,%20org.xml.sax.Attributes)) event for every endElement event (even when the element is empty).

For information on the names, see startElement.

**Parameters:**uri - the Namespace URI, or the empty string if the element has no Namespace URI or if Namespace processing is not being performedlocalName - the local name (without prefix), or the empty string if Namespace processing is not being performedqName - the qualified XML name (with prefix), or the empty string if qualified names are not available **Throws:** [SAXException](http://docs.google.com/org/xml/sax/SAXException.html) - any SAX exception, possibly wrapping another exception

### characters

void **characters**(char[] ch,  
 int start,  
 int length)  
 throws [SAXException](http://docs.google.com/org/xml/sax/SAXException.html)

Receive notification of character data.

The Parser will call this method to report each chunk of character data. SAX parsers may return all contiguous character data in a single chunk, or they may split it into several chunks; however, all of the characters in any single event must come from the same external entity so that the Locator provides useful information.

The application must not attempt to read from the array outside of the specified range.

Individual characters may consist of more than one Java char value. There are two important cases where this happens, because characters can't be represented in just sixteen bits. In one case, characters are represented in a *Surrogate Pair*, using two special Unicode values. Such characters are in the so-called "Astral Planes", with a code point above U+FFFF. A second case involves composite characters, such as a base character combining with one or more accent characters.

Your code should not assume that algorithms using char-at-a-time idioms will be working in character units; in some cases they will split characters. This is relevant wherever XML permits arbitrary characters, such as attribute values, processing instruction data, and comments as well as in data reported from this method. It's also generally relevant whenever Java code manipulates internationalized text; the issue isn't unique to XML.

Note that some parsers will report whitespace in element content using the [ignorableWhitespace](http://docs.google.com/org/xml/sax/ContentHandler.html#ignorableWhitespace(char%5B%5D,%20int,%20int)) method rather than this one (validating parsers *must* do so).

**Parameters:**ch - the characters from the XML documentstart - the start position in the arraylength - the number of characters to read from the array **Throws:** [SAXException](http://docs.google.com/org/xml/sax/SAXException.html) - any SAX exception, possibly wrapping another exception**See Also:**[ignorableWhitespace(char[], int, int)](http://docs.google.com/org/xml/sax/ContentHandler.html#ignorableWhitespace(char%5B%5D,%20int,%20int)), [Locator](http://docs.google.com/org/xml/sax/Locator.html)

### ignorableWhitespace

void **ignorableWhitespace**(char[] ch,  
 int start,  
 int length)  
 throws [SAXException](http://docs.google.com/org/xml/sax/SAXException.html)

Receive notification of ignorable whitespace in element content.

Validating Parsers must use this method to report each chunk of whitespace in element content (see the W3C XML 1.0 recommendation, section 2.10): non-validating parsers may also use this method if they are capable of parsing and using content models.

SAX parsers may return all contiguous whitespace in a single chunk, or they may split it into several chunks; however, all of the characters in any single event must come from the same external entity, so that the Locator provides useful information.

The application must not attempt to read from the array outside of the specified range.

**Parameters:**ch - the characters from the XML documentstart - the start position in the arraylength - the number of characters to read from the array **Throws:** [SAXException](http://docs.google.com/org/xml/sax/SAXException.html) - any SAX exception, possibly wrapping another exception**See Also:**[characters(char[], int, int)](http://docs.google.com/org/xml/sax/ContentHandler.html#characters(char%5B%5D,%20int,%20int))

### processingInstruction

void **processingInstruction**([String](http://docs.google.com/java/lang/String.html) target,  
 [String](http://docs.google.com/java/lang/String.html) data)  
 throws [SAXException](http://docs.google.com/org/xml/sax/SAXException.html)

Receive notification of a processing instruction.

The Parser will invoke this method once for each processing instruction found: note that processing instructions may occur before or after the main document element.

A SAX parser must never report an XML declaration (XML 1.0, section 2.8) or a text declaration (XML 1.0, section 4.3.1) using this method.

Like [characters()](http://docs.google.com/org/xml/sax/ContentHandler.html#characters(char%5B%5D,%20int,%20int)), processing instruction data may have characters that need more than one char value.

**Parameters:**target - the processing instruction targetdata - the processing instruction data, or null if none was supplied. The data does not include any whitespace separating it from the target **Throws:** [SAXException](http://docs.google.com/org/xml/sax/SAXException.html) - any SAX exception, possibly wrapping another exception

### skippedEntity

void **skippedEntity**([String](http://docs.google.com/java/lang/String.html) name)  
 throws [SAXException](http://docs.google.com/org/xml/sax/SAXException.html)

Receive notification of a skipped entity. This is not called for entity references within markup constructs such as element start tags or markup declarations. (The XML recommendation requires reporting skipped external entities. SAX also reports internal entity expansion/non-expansion, except within markup constructs.)

The Parser will invoke this method each time the entity is skipped. Non-validating processors may skip entities if they have not seen the declarations (because, for example, the entity was declared in an external DTD subset). All processors may skip external entities, depending on the values of the http://xml.org/sax/features/external-general-entities and the http://xml.org/sax/features/external-parameter-entities properties.

**Parameters:**name - the name of the skipped entity. If it is a parameter entity, the name will begin with '%', and if it is the external DTD subset, it will be the string "[dtd]" **Throws:** [SAXException](http://docs.google.com/org/xml/sax/SAXException.html) - any SAX exception, possibly wrapping another exception

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ContentHandler.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/org/xml/sax/Attributes.html)   [**NEXT CLASS**](http://docs.google.com/org/xml/sax/DocumentHandler.html) | [**FRAMES**](http://docs.google.com/index.html?org/xml/sax/ContentHandler.html)    [**NO FRAMES**](http://docs.google.com/ContentHandler.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).