| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/EntityResolver2.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/ext/DefaultHandler2.html)   [**NEXT CLASS**](http://docs.google.com/org/xml/sax/ext/LexicalHandler.html) | [**FRAMES**](http://docs.google.com/index.html?org/xml/sax/ext/EntityResolver2.html)    [**NO FRAMES**](http://docs.google.com/EntityResolver2.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

## **org.xml.sax.ext**

Interface EntityResolver2

**All Superinterfaces:** [EntityResolver](http://docs.google.com/org/xml/sax/EntityResolver.html) **All Known Implementing Classes:** [DefaultHandler2](http://docs.google.com/org/xml/sax/ext/DefaultHandler2.html)

public interface **EntityResolver2**extends [EntityResolver](http://docs.google.com/org/xml/sax/EntityResolver.html)

Extended interface for mapping external entity references to input sources, or providing a missing external subset. The [XMLReader.setEntityResolver()](http://docs.google.com/org/xml/sax/XMLReader.html#setEntityResolver(org.xml.sax.EntityResolver)) method is used to provide implementations of this interface to parsers. When a parser uses the methods in this interface, the [EntityResolver2.resolveEntity()](http://docs.google.com/org/xml/sax/ext/EntityResolver2.html#resolveEntity(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String)) method (in this interface) is used *instead of* the older (SAX 1.0) [EntityResolver.resolveEntity()](http://docs.google.com/org/xml/sax/EntityResolver.html#resolveEntity(java.lang.String,%20java.lang.String)) method.

*This module, both source code and documentation, is in the Public Domain, and comes with* ***NO WARRANTY****.*

If a SAX application requires the customized handling which this interface defines for external entities, it must ensure that it uses an XMLReader with the *http://xml.org/sax/features/use-entity-resolver2* feature flag set to *true* (which is its default value when the feature is recognized). If that flag is unrecognized, or its value is false, or the resolver does not implement this interface, then only the [EntityResolver](http://docs.google.com/org/xml/sax/EntityResolver.html) method will be used.

That supports three categories of application that modify entity resolution. *Old Style* applications won't know about this interface; they will provide an EntityResolver. *Transitional Mode* provide an EntityResolver2 and automatically get the benefit of its methods in any systems (parsers or other tools) supporting it, due to polymorphism. Both *Old Style* and *Transitional Mode* applications will work with any SAX2 parser. *New style* applications will fail to run except on SAX2 parsers that support this particular feature. They will insist that feature flag have a value of "true", and the EntityResolver2 implementation they provide might throw an exception if the original SAX 1.0 style entity resolution method is invoked.

**Since:** SAX 2.0 (extensions 1.1 alpha) **See Also:**[XMLReader.setEntityResolver(org.xml.sax.EntityResolver)](http://docs.google.com/org/xml/sax/XMLReader.html#setEntityResolver(org.xml.sax.EntityResolver))

| **Method Summary** | |
| --- | --- |
| [InputSource](http://docs.google.com/org/xml/sax/InputSource.html) | [**getExternalSubset**](http://docs.google.com/org/xml/sax/ext/EntityResolver2.html#getExternalSubset(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) name, [String](http://docs.google.com/java/lang/String.html) baseURI)            Allows applications to provide an external subset for documents that don't explicitly define one. |
| [InputSource](http://docs.google.com/org/xml/sax/InputSource.html) | [**resolveEntity**](http://docs.google.com/org/xml/sax/ext/EntityResolver2.html#resolveEntity(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) name, [String](http://docs.google.com/java/lang/String.html) publicId, [String](http://docs.google.com/java/lang/String.html) baseURI, [String](http://docs.google.com/java/lang/String.html) systemId)            Allows applications to map references to external entities into input sources, or tell the parser it should use conventional URI resolution. |

| **Methods inherited from interface org.xml.sax.**[**EntityResolver**](http://docs.google.com/org/xml/sax/EntityResolver.html) |
| --- |
| [resolveEntity](http://docs.google.com/org/xml/sax/EntityResolver.html#resolveEntity(java.lang.String,%20java.lang.String)) |

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

### getExternalSubset

[InputSource](http://docs.google.com/org/xml/sax/InputSource.html) **getExternalSubset**([String](http://docs.google.com/java/lang/String.html) name,  
 [String](http://docs.google.com/java/lang/String.html) baseURI)  
 throws [SAXException](http://docs.google.com/org/xml/sax/SAXException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Allows applications to provide an external subset for documents that don't explicitly define one. Documents with DOCTYPE declarations that omit an external subset can thus augment the declarations available for validation, entity processing, and attribute processing (normalization, defaulting, and reporting types including ID). This augmentation is reported through the [startDTD()](http://docs.google.com/org/xml/sax/ext/LexicalHandler.html#startDTD(java.lang.String,%20java.lang.String,%20java.lang.String)) method as if the document text had originally included the external subset; this callback is made before any internal subset data or errors are reported.

This method can also be used with documents that have no DOCTYPE declaration. When the root element is encountered, but no DOCTYPE declaration has been seen, this method is invoked. If it returns a value for the external subset, that root element is declared to be the root element, giving the effect of splicing a DOCTYPE declaration at the end the prolog of a document that could not otherwise be valid. The sequence of parser callbacks in that case logically resembles this:

... comments and PIs from the prolog (as usual)  
 startDTD ("rootName", source.getPublicId (), source.getSystemId ());  
 startEntity ("[dtd]");  
 ... declarations, comments, and PIs from the external subset  
 endEntity ("[dtd]");  
 endDTD ();  
 ... then the rest of the document (as usual)  
 startElement (..., "rootName", ...);

Note that the InputSource gets no further resolution. Implementations of this method may wish to invoke [resolveEntity()](http://docs.google.com/org/xml/sax/ext/EntityResolver2.html#resolveEntity(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String)) to gain benefits such as use of local caches of DTD entities. Also, this method will never be used by a (non-validating) processor that is not including external parameter entities.

Uses for this method include facilitating data validation when interoperating with XML processors that would always require undesirable network accesses for external entities, or which for other reasons adopt a "no DTDs" policy. Non-validation motives include forcing documents to include DTDs so that attributes are handled consistently. For example, an XPath processor needs to know which attibutes have type "ID" before it can process a widely used type of reference.

**Warning:** Returning an external subset modifies the input document. By providing definitions for general entities, it can make a malformed document appear to be well formed.

**Parameters:**name - Identifies the document root element. This name comes from a DOCTYPE declaration (where available) or from the actual root element.baseURI - The document's base URI, serving as an additional hint for selecting the external subset. This is always an absolute URI, unless it is null because the XMLReader was given an InputSource without one. **Returns:**An InputSource object describing the new external subset to be used by the parser, or null to indicate that no external subset is provided. **Throws:** [SAXException](http://docs.google.com/org/xml/sax/SAXException.html) - Any SAX exception, possibly wrapping another exception. [IOException](http://docs.google.com/java/io/IOException.html) - Probably indicating a failure to create a new InputStream or Reader, or an illegal URL.

### resolveEntity

[InputSource](http://docs.google.com/org/xml/sax/InputSource.html) **resolveEntity**([String](http://docs.google.com/java/lang/String.html) name,  
 [String](http://docs.google.com/java/lang/String.html) publicId,  
 [String](http://docs.google.com/java/lang/String.html) baseURI,  
 [String](http://docs.google.com/java/lang/String.html) systemId)  
 throws [SAXException](http://docs.google.com/org/xml/sax/SAXException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Allows applications to map references to external entities into input sources, or tell the parser it should use conventional URI resolution. This method is only called for external entities which have been properly declared. This method provides more flexibility than the [EntityResolver](http://docs.google.com/org/xml/sax/EntityResolver.html) interface, supporting implementations of more complex catalogue schemes such as the one defined by the [OASIS XML Catalogs](http://www.oasis-open.org/committees/entity/spec-2001-08-06.html) specification.

Parsers configured to use this resolver method will call it to determine the input source to use for any external entity being included because of a reference in the XML text. That excludes the document entity, and any external entity returned by [getExternalSubset()](http://docs.google.com/org/xml/sax/ext/EntityResolver2.html#getExternalSubset(java.lang.String,%20java.lang.String)). When a (non-validating) processor is configured not to include a class of entities (parameter or general) through use of feature flags, this method is not invoked for such entities.

Note that the entity naming scheme used here is the same one used in the [LexicalHandler](http://docs.google.com/org/xml/sax/ext/LexicalHandler.html), or in the [ContentHandler.skippedEntity()](http://docs.google.com/org/xml/sax/ContentHandler.html#skippedEntity(java.lang.String)) method.

**Parameters:**name - Identifies the external entity being resolved. Either "[dtd]" for the external subset, or a name starting with "%" to indicate a parameter entity, or else the name of a general entity. This is never null when invoked by a SAX2 parser.publicId - The public identifier of the external entity being referenced (normalized as required by the XML specification), or null if none was supplied.baseURI - The URI with respect to which relative systemIDs are interpreted. This is always an absolute URI, unless it is null (likely because the XMLReader was given an InputSource without one). This URI is defined by the XML specification to be the one associated with the "<" starting the relevant declaration.systemId - The system identifier of the external entity being referenced; either a relative or absolute URI. This is never null when invoked by a SAX2 parser; only declared entities, and any external subset, are resolved by such parsers. **Returns:**An InputSource object describing the new input source to be used by the parser. Returning null directs the parser to resolve the system ID against the base URI and open a connection to resulting URI. **Throws:** [SAXException](http://docs.google.com/org/xml/sax/SAXException.html) - Any SAX exception, possibly wrapping another exception. [IOException](http://docs.google.com/java/io/IOException.html) - Probably indicating a failure to create a new InputStream or Reader, or an illegal URL.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/EntityResolver2.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/ext/DefaultHandler2.html)   [**NEXT CLASS**](http://docs.google.com/org/xml/sax/ext/LexicalHandler.html) | [**FRAMES**](http://docs.google.com/index.html?org/xml/sax/ext/EntityResolver2.html)    [**NO FRAMES**](http://docs.google.com/EntityResolver2.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

[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).