

org.xml.sax

Interface ErrorHandler

All Known Implementing Classes:
[DefaultHandler](#), [HandlerBase](#), [XMLFilterImpl](#)

public interface **ErrorHandler**

Basic interface for SAX error handlers.

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

If a SAX application needs to implement customized error handling, it must implement this interface and then register an instance with the XML reader using the [setErrorHandler](#) method. The parser will then report all errors and warnings through this interface.

WARNING: If an application does *not* register an ErrorHandler, XML parsing errors will go unreported and bizarre behaviour may result.

For XML processing errors, a SAX driver must use this interface instead of throwing an exception: it is up to the application to decide whether to throw an exception for different types of errors and warnings. Note, however, that there is no requirement that the parser continue to provide useful information after a call to [fatalError](#) (in other words, a SAX driver class could catch an exception and report a fatalError).

Since:
SAX 1.0

See Also:
[Parser.setErrorHandler\(org.xml.sax.ErrorHandler\)](#), [SAXParseException](#)

Method Summary	
void	error (SAXParseException exception) Receive notification of a recoverable error.
void	fatalError (SAXParseException exception) Receive notification of a non-recoverable error.
void	warning (SAXParseException exception) Receive notification of a warning.

Method Detail

warning

public void **warning**([SAXParseException](#) exception)
throws [SAXException](#)

Receive notification of a warning.

SAX parsers will use this method to report conditions that are not errors or fatal errors as defined by the XML 1.0 recommendation. The default behaviour is to take no action.

The SAX parser must continue to provide normal parsing events after invoking this method: it should still be possible for the application to process the document through to the end.

Filters may use this method to report other, non-XML warnings as well.

Parameters:
exception - The warning information encapsulated in a SAX parse exception.

Throws:

[SAXException](#) - Any SAX exception, possibly wrapping another exception.

See Also:

[SAXParseException](#)

error

```
public void error(SAXParseException exception)
    throws SAXException
```

Receive notification of a recoverable error.

This corresponds to the definition of "error" in section 1.2 of the W3C XML 1.0 Recommendation. For example, a validating parser would use this callback to report the violation of a validity constraint. The default behaviour is to take no action.

The SAX parser must continue to provide normal parsing events after invoking this method: it should still be possible for the application to process the document through to the end. If the application cannot do so, then the parser should report a fatal error even if the XML 1.0 recommendation does not require it to do so.

Filters may use this method to report other, non-XML errors as well.

Parameters:

exception - The error information encapsulated in a SAX parse exception.

Throws:

[SAXException](#) - Any SAX exception, possibly wrapping another exception.

See Also:

[SAXParseException](#)

fatalError

```
public void fatalError(SAXParseException exception)
    throws SAXException
```

Receive notification of a non-recoverable error.

This corresponds to the definition of "fatal error" in section 1.2 of the W3C XML 1.0 Recommendation. For example, a parser would use this callback to report the violation of a well-formedness constraint.

The application must assume that the document is unusable after the parser has invoked this method, and should continue (if at all) only for the sake of collecting addition error messages: in fact, SAX parsers are free to stop reporting any other events once this method has been invoked.

Parameters:

exception - The error information encapsulated in a SAX parse exception.

Throws:

[SAXException](#) - Any SAX exception, possibly wrapping another exception.

See Also:

[SAXParseException](#)

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

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY: NESTED | FIELD | CONSTR | \[METHOD\]\(#\)](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL: FIELD | CONSTR | \[METHOD\]\(#\)](#)

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](#).

