

Navigation

- [index](#)
- [modules](#) |
- [next](#) |
- [previous](#) |
- [Python v2.6.4 documentation](#) »
- [The Python Standard Library](#) »
- [20. Structured Markup Processing Tools](#) »

20.9. `xml.sax` — Support for SAX2 parsers¶

New in version 2.0.

The `xml.sax` package provides a number of modules which implement the Simple API for XML (SAX) interface for Python. The package itself provides the SAX exceptions and the convenience functions which will be most used by users of the SAX API.

The convenience functions are:

```
xml.sax.make_parser([parser_list])¶
```

Create and return a SAX `XMLReader` object. The first parser found will be used. If `parser_list` is provided, it must be a sequence of strings which name modules that have a function named `create_parser()`. Modules listed in `parser_list` will be used before modules in the default list of parsers.

```
xml.sax.parse(filename_or_stream, handler[, error_handler])¶
```

Create a SAX parser and use it to parse a document. The document, passed in as `filename_or_stream`, can be a filename or a file object. The `handler` parameter needs to be a SAX `ContentHandler` instance. If `error_handler` is given, it must be a SAX `ErrorHandler` instance; if omitted, [SAXParseException](#) will be raised on all errors. There is no return value; all work must be done by the `handler` passed in.

```
xml.sax.parseString(string, handler[, error_handler])¶
```

Similar to [parse\(\)](#), but parses from a buffer `string` received as a parameter.

A typical SAX application uses three kinds of objects: readers, handlers and input sources. “Reader” in this context is another term for parser, i.e. some piece of code that reads the bytes or characters from the input source, and produces a sequence of events. The events then get distributed to the handler objects, i.e. the reader invokes a method on the handler. A SAX application must therefore obtain a reader object, create or open the input sources, create the handlers, and connect these objects all together. As the final step of preparation, the reader is called to parse the input. During parsing, methods on the handler objects are called based on structural and syntactic events from the input data.

For these objects, only the interfaces are relevant; they are normally not instantiated by the application itself. Since Python does not have an explicit notion of interface, they are formally introduced as classes, but applications may use implementations which do not inherit from the provided classes. The `InputSource`, `Locator`, `Attributes`, `AttributesNS`, and `XMLReader` interfaces are defined in the module [xml.sax.xmlreader](#). The handler interfaces are defined in [xml.sax.handler](#). For convenience, `InputSource` (which is often instantiated directly) and the handler classes are also available from `xml.sax`. These interfaces are described below.

In addition to these classes, `xml.sax` provides the following exception classes.

```
exception xml.sax.SAXException(msg[, exception])¶
```

Encapsulate an XML error or warning. This class can contain basic error or warning information from either the XML parser or the application: it can be subclassed to provide additional functionality or to add localization. Note that although the handlers defined in the `ErrorHandler` interface receive instances of this exception, it is not required to actually raise the exception — it is also useful as a container for information.

When instantiated, `msg` should be a human-readable description of the error. The optional `exception` parameter, if given, should be `None` or an exception that was caught by the parsing code and is being passed along as information.

This is the base class for the other SAX exception classes.

```
exception xml.sax.SAXParseException(msg, exception, locator)¶
```

Subclass of [SAXException](#) raised on parse errors. Instances of this class are passed to the methods of the SAX `ErrorHandler` interface to provide information about the parse error. This class supports the SAX `Locator` interface as well as the [SAXException](#) interface.

```
exception xml.sax.SAXNotRecognizedException(msg[, exception])¶
```

Subclass of [SAXException](#) raised when a SAX `XMLReader` is confronted with an unrecognized feature or property. SAX applications and extensions may use this class for similar purposes.

```
exception xml.sax.SAXNotSupportedException(msg[, exception])¶
```

Subclass of [SAXException](#) raised when a SAX `XMLReader` is asked to enable a feature that is not supported, or to set a property to a value that the implementation does not support. SAX applications and extensions may use this class for similar purposes.

See also

[SAX: The Simple API for XML](#)

This site is the focal point for the definition of the SAX API. It provides a Java implementation and online documentation. Links to implementations and historical information are also available.

Module [xml.sax.handler](#)

Definitions of the interfaces for application-provided objects.

Module [xml.sax.saxutils](#)

Convenience functions for use in SAX applications.

Module [xml.sax.xmlreader](#)

Definitions of the interfaces for parser-provided objects.

20.9.1. SAXException Objects¶

The [SAXException](#) exception class supports the following methods:

`SAXException.getMessage()`¶

Return a human-readable message describing the error condition.

`SAXException.getException()`¶

Return an encapsulated exception object, or `None`.

[Table Of Contents](#)

[20.9. xml.sax — Support for SAX2 parsers](#)

- [20.9.1. SAXException Objects](#)

Previous topic

[20.8. xml.dom.pulldom — Support for building partial DOM trees](#)

Next topic

[20.10. xml.sax.handler — Base classes for SAX handlers](#)

This Page

- [Show Source](#)

Navigation

- [index](#)
- [modules](#) |
- [next](#) |
- [previous](#) |
- [Python v2.6.4 documentation](#) »
- [The Python Standard Library](#) »
- [20. Structured Markup Processing Tools](#) »

© [Copyright](#) 1990-2010, Python Software Foundation.

The Python Software Foundation is a non-profit corporation. [Please donate.](#)

Last updated on Feb 26, 2010. Created using [Sphinx](#) 0.6.3.