

The xml.parsers.expat module

(Optional). This is an interface to James Clark's Expat XML parser. This is a full-featured and fast XML parser, and an excellent choice for production use.

The parser uses registered handlers to handle different parts of the XML document.

Example: Using the xml.parsers.expat module

```
# File: xml-parsers-ecat-example-1.py

from xml.parsers import expat

class Parser:

    def __init__(self):
        self._parser = expat.ParserCreate()
        self._parser.StartElementHandler = self.start
        self._parser.EndElementHandler = self.end
        self._parser.CharacterDataHandler = self.data

    def feed(self, data):
        self._parser.Parse(data, 0)

    def close(self):
        self._parser.Parse("", 1) # end of data
        del self._parser # get rid of circular references

    def start(self, tag, attrs):
        print "START", repr(tag), attrs

    def end(self, tag):
        print "END", repr(tag)

    def data(self, data):
        print "DATA", repr(data)

p = Parser()
p.feed("<tag>data</tag>")
p.close()

$ python xml-parsers-ecat-example-1.py
START u'tag' {}
DATA u'data'
END u'tag'
```

Note that the parser returns Unicode strings, even if you pass it ordinary text. By default, the parser interprets the source text as UTF-8 (as per the XML standard). To use other encodings, make sure the XML file contains an *encoding* directive.

Example: Using the xml.parsers.expat module to read ISO Latin-1 text

```
# File: xml-parsers-ecat-example-2.py

from xml.parsers import expat

class Parser:

    def __init__(self):
        self._parser = expat.ParserCreate()
        self._parser.StartElementHandler = self.start
        self._parser.EndElementHandler = self.end
        self._parser.CharacterDataHandler = self.data

    def feed(self, data):
        self._parser.Parse(data, 0)

    def close(self):
        self._parser.Parse("", 1) # end of data
```

```
del self._parser # get rid of circular references


def start(self, tag, attrs):
    print "START", repr(tag), attrs

def end(self, tag):
    print "END", repr(tag)

def data(self, data):
    print "DATA", repr(data)

p = Parser()
p.feed("""\
<?xml version='1.0' encoding='iso-8859-1'?>
<author>
<name>fredrik lundh</name>
<city>linköping</city>
</author>
""")
)
p.close()

$ python xml-parsers-expat-example-2.py
START u'author' {}
DATA u'\012'
START u'name' {}
DATA u'fredrik lundh'
END u'name'
DATA u'\012'
START u'city' {}
DATA u'link\366ping'
END u'city'
DATA u'\012'
END u'author'
```

 rendered by a [django](#) application. hosted by [webfaction](#).