:mod:`xml.parsers.expat` --- Fast XML parsing using Expat

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, line 1`); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, line 4`)

Unknown directive type "module".

```
.. module:: xml.parsers.expat
   :synopsis: An interface to the Expat non-validating XML parser.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, line 7`)

Unknown directive type "moduleauthor".

```
.. moduleauthor:: Paul Prescod <paul@prescod.net>
```

> **Warning**
>
> The :mod:`pyexpat` module is not secure against maliciously constructed data. If you need to parse untrusted or unauthenticated data see :ref:`xml-vulnerabilities`.
>
> > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, line 21`); *backlink***
> >
> > Unknown interpreted text role "mod".
>
> > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, line 21`); *backlink***
> >
> > Unknown interpreted text role "ref".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, line 26`)

Unknown directive type "index".

```
.. index:: single: Expat
```

The :mod:`xml.parsers.expat` module is a Python interface to the Expat non-validating XML parser. The module provides a single extension type, :class:`xmlparser`, that represents the current state of an XML parser. After an :class:`xmlparser` object has been created, various attributes of the object can be set to handler functions. When an XML document is then fed to the parser, the handler functions are called for the character data and markup in the XML document.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, line 28`); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, line 28`); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, line 28`); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, line 36`)

Unknown directive type "index".

```
.. index:: module: pyexpat
```

This module uses the :mod:`pyexpat` module to provide access to the Expat parser. Direct use of the :mod:`pyexpat` module is deprecated.

This module provides one exception and one type object:

```
.. exception:: ExpatError

   The exception raised when Expat reports an error.  See section
   :ref:`expaterror-objects` for more information on interpreting Expat errors.
```

```
.. exception:: error

   Alias for :exc:`ExpatError`.
```

```
.. data:: XMLParserType

   The type of the return values from the :func:`ParserCreate` function.
```

The :mod:`xml.parsers.expat` module contains two functions:

```
.. function:: ErrorString(errno)

   Returns an explanatory string for a given error number *errno*.
```

```
.. function:: ParserCreate(encoding=None, namespace_separator=None)

   Creates and returns a new :class:`xmlparser` object.  *encoding*, if specified,
   must be a string naming the encoding  used by the XML data.  Expat doesn't
   support as many encodings as Python does, and its repertoire of encodings can't
   be extended; it supports UTF-8, UTF-16, ISO-8859-1 (Latin1), and ASCII.  If
   *encoding* [1]_ is given it will override the implicit or explicit encoding of the
   document.

   Expat can optionally do XML namespace processing for you, enabled by providing a
   value for *namespace_separator*.  The value must be a one-character string; a
   :exc:`ValueError` will be raised if the string has an illegal length (``None``
   is considered the same as omission).  When namespace processing is enabled,
   element type names and attribute names that belong to a namespace will be
   expanded.  The element name passed to the element handlers
```

```
:attr:`StartElementHandler` and :attr:`EndElementHandler` will be the
concatenation of the namespace URI, the namespace separator character, and the
local part of the name.  If the namespace separator is a zero byte (``chr(0)``)
then the namespace URI and the local part will be concatenated without any
separator.

For example, if *namespace_separator* is set to a space character (``' '``) and
the following document is parsed:

.. code-block:: xml

   <?xml version="1.0"?>
   <root xmlns    = "http://default-namespace.org/"
         xmlns:py = "http://www.python.org/ns/">
    <py:elem1 />
    <elem2 xmlns="" />
   </root>

:attr:`StartElementHandler` will receive the following strings for each
element::

   http://default-namespace.org/ root
   http://www.python.org/ns/ elem1
   elem2

Due to limitations in the ``Expat`` library used by :mod:`pyexpat`,
the :class:`xmlparser` instance returned can only be used to parse a single
XML document.  Call ``ParserCreate`` for each document to provide unique
parser instances.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 113)

Unknown directive type "seealso".

```
.. seealso::

   `The Expat XML Parser <http://www.libexpat.org/>`_
      Home page of the Expat project.
```

XMLParser Objects

:class:`xmlparser` objects have the following methods:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 124); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 127)

Unknown directive type "method".

```
.. method:: xmlparser.Parse(data[, isfinal])

   Parses the contents of the string *data*, calling the appropriate handler
   functions to process the parsed data.  *isfinal* must be true on the final call
   to this method; it allows the parsing of a single file in fragments,
   not the submission of multiple files.
   *data* can be the empty string at any time.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 136)

Unknown directive type "method".

```
.. method:: xmlparser.ParseFile(file)

   Parse XML data reading from the object *file*.  *file* only needs to provide
   the ``read(nbytes)`` method, returning the empty string when there's no more
   data.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 143)

Unknown directive type "method".

```
.. method:: xmlparser.SetBase(base)

   Sets the base to be used for resolving relative URIs in system identifiers in
   declarations.  Resolving relative identifiers is left to the application: this
   value will be passed through as the *base* argument to the
```

```
    :func:`ExternalEntityRefHandler`, :func:`NotationDeclHandler`, and
    :func:`UnparsedEntityDeclHandler` functions.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, **line 152)**

Unknown directive type "method".

```
    .. method:: xmlparser.GetBase()

       Returns a string containing the base set by a previous call to :meth:`SetBase`,
       or ``None`` if  :meth:`SetBase` hasn't been called.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, **line 158)**

Unknown directive type "method".

```
    .. method:: xmlparser.GetInputContext()

       Returns the input data that generated the current event as a string. The data is
       in the encoding of the entity which contains the text. When called while an
       event handler is not active, the return value is ``None``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, **line 165)**

Unknown directive type "method".

```
    .. method:: xmlparser.ExternalEntityParserCreate(context[, encoding])

       Create a "child" parser which can be used to parse an external parsed entity
       referred to by content parsed by the parent parser.  The *context* parameter
       should be the string passed to the :meth:`ExternalEntityRefHandler` handler
       function, described below. The child parser is created with the
       :attr:`ordered_attributes` and :attr:`specified_attributes` set to the values of
       this parser.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, **line 174)**

Unknown directive type "method".

```
    .. method:: xmlparser.SetParamEntityParsing(flag)

       Control parsing of parameter entities (including the external DTD subset).
       Possible *flag* values are :const:`XML_PARAM_ENTITY_PARSING_NEVER`,
       :const:`XML_PARAM_ENTITY_PARSING_UNLESS_STANDALONE` and
       :const:`XML_PARAM_ENTITY_PARSING_ALWAYS`.  Return true if setting the flag
       was successful.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, **line 182)**

Unknown directive type "method".

```
    .. method:: xmlparser.UseForeignDTD([flag])

       Calling this with a true value for *flag* (the default) will cause Expat to call
       the :attr:`ExternalEntityRefHandler` with :const:`None` for all arguments to
       allow an alternate DTD to be loaded.  If the document does not contain a
       document type declaration, the :attr:`ExternalEntityRefHandler` will still be
       called, but the :attr:`StartDoctypeDeclHandler` and
       :attr:`EndDoctypeDeclHandler` will not be called.

       Passing a false value for *flag* will cancel a previous call that passed a true
       value, but otherwise has no effect.

       This method can only be called before the :meth:`Parse` or :meth:`ParseFile`
       methods are called; calling it after either of those have been called causes
       :exc:`ExpatError` to be raised with the :attr:`code` attribute set to
       ``errors.codes[errors.XML_ERROR_CANT_CHANGE_FEATURE_ONCE_PARSING]``.
```

:class:`xmlparser` objects have the following attributes:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, **line 199);** *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

Unknown directive type "attribute".

```
.. attribute:: xmlparser.buffer_size

   The size of the buffer used when :attr:`buffer_text` is true.
   A new buffer size can be set by assigning a new integer value
   to this attribute.
   When the size is changed, the buffer will be flushed.
```

Unknown directive type "attribute".

```
.. attribute:: xmlparser.buffer_text

   Setting this to true causes the :class:`xmlparser` object to buffer textual
   content returned by Expat to avoid multiple calls to the
   :meth:`CharacterDataHandler` callback whenever possible.  This can improve
   performance substantially since Expat normally breaks character data into chunks
   at every line ending.  This attribute is false by default, and may be changed at
   any time.
```

Unknown directive type "attribute".

```
.. attribute:: xmlparser.buffer_used

   If :attr:`buffer_text` is enabled, the number of bytes stored in the buffer.
   These bytes represent UTF-8 encoded text.  This attribute has no meaningful
   interpretation when :attr:`buffer_text` is false.
```

Unknown directive type "attribute".

```
.. attribute:: xmlparser.ordered_attributes

   Setting this attribute to a non-zero integer causes the attributes to be
   reported as a list rather than a dictionary.  The attributes are presented in
   the order found in the document text.  For each attribute, two list entries are
   presented: the attribute name and the attribute value.  (Older versions of this
   module also used this format.)  By default, this attribute is false; it may be
   changed at any time.
```

Unknown directive type "attribute".

```
.. attribute:: xmlparser.specified_attributes

   If set to a non-zero integer, the parser will report only those attributes which
   were specified in the document instance and not those which were derived from
   attribute declarations.  Applications which set this need to be especially
   careful to use what additional information is available from the declarations as
   needed to comply with the standards for the behavior of XML processors.  By
   default, this attribute is false; it may be changed at any time.
```

The following attributes contain values relating to the most recent error encountered by an :class:`xmlparser` object, and will only have correct values once a call to :meth:`Parse` or :meth:`ParseFile` has raised an :exc:`xml.parsers.expat.ExpatError` exception.

Unknown interpreted text role "class".

Unknown interpreted text role "meth".

Unknown interpreted text role "meth".

```
.. attribute:: xmlparser.ErrorByteIndex

   Byte index at which an error occurred.
```

```
.. attribute:: xmlparser.ErrorCode

   Numeric code specifying the problem.  This value can be passed to the
   :func:`ErrorString` function, or compared to one of the constants defined in the
   ``errors`` object.
```

```
.. attribute:: xmlparser.ErrorColumnNumber

   Column number at which an error occurred.
```

```
.. attribute:: xmlparser.ErrorLineNumber

   Line number at which an error occurred.
```

The following attributes contain values relating to the current parse location in an :class:`xmlparser` object. During a callback reporting a parse event they indicate the location of the first of the sequence of characters that generated the event. When called outside of a callback, the position indicated will be just past the last parse event (regardless of whether there was an associated callback).

```
.. attribute:: xmlparser.CurrentByteIndex

   Current byte index in the parser input.
```

```
.. attribute:: xmlparser.CurrentColumnNumber

   Current column number in the parser input.
```

```
.. attribute:: xmlparser.CurrentLineNumber

   Current line number in the parser input.
```

Here is the list of handlers that can be set. To set a handler on an :class:`xmlparser` object *o*, use `o.handlername = func`. *handlername* must be taken from the following list, and *func* must be a callable object accepting the correct number of arguments. The arguments are all strings, unless otherwise stated.

```
.. method:: xmlparser.XmlDeclHandler(version, encoding, standalone)

   Called when the XML declaration is parsed.  The XML declaration is the
   (optional) declaration of the applicable version of the XML recommendation, the
   encoding of the document text, and an optional "standalone" declaration.
   *version* and *encoding* will be strings, and *standalone* will be ``1`` if the
   document is declared standalone, ``0`` if it is declared not to be standalone,
   or ``-1`` if the standalone clause was omitted. This is only available with
   Expat version 1.95.0 or newer.
```

```
.. method:: xmlparser.StartDoctypeDeclHandler(doctypeName, systemId, publicId, has_internal_subset)

   Called when Expat begins parsing the document type declaration (``<!DOCTYPE
   ...``).  The *doctypeName* is provided exactly as presented.  The *systemId* and
   *publicId* parameters give the system and public identifiers if specified, or
   ``None`` if omitted.  *has_internal_subset* will be true if the document
   contains and internal document declaration subset. This requires Expat version
   1.2 or newer.
```

```
.. method:: xmlparser.EndDoctypeDeclHandler()

   Called when Expat is done parsing the document type declaration. This requires
   Expat version 1.2 or newer.
```

```
.. method:: xmlparser.ElementDeclHandler(name, model)

   Called once for each element type declaration.  *name* is the name of the
   element type, and *model* is a representation of the content model.
```

```
.. method:: xmlparser.AttlistDeclHandler(elname, attname, type, default, required)

   Called for each declared attribute for an element type.  If an attribute list
   declaration declares three attributes, this handler is called three times, once
   for each attribute.  *elname* is the name of the element to which the
   declaration applies and *attname* is the name of the attribute declared.  The
   attribute type is a string passed as *type*; the possible values are
   ``'CDATA'``, ``'ID'``, ``'IDREF'``, ... *default* gives the default value for
   the attribute used when the attribute is not specified by the document instance,
   or ``None`` if there is no default value (``#IMPLIED`` values).  If the
   attribute is required to be given in the document instance, *required* will be
   true. This requires Expat version 1.95.0 or newer.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 350)

Unknown directive type "method".

```
.. method:: xmlparser.StartElementHandler(name, attributes)

   Called for the start of every element.  *name* is a string containing the
   element name, and *attributes* is the element attributes. If
   :attr:`ordered_attributes` is true, this is a list (see
   :attr:`ordered_attributes` for a full description). Otherwise it's a
   dictionary mapping names to values.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 359)

Unknown directive type "method".

```
.. method:: xmlparser.EndElementHandler(name)

   Called for the end of every element.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 364)

Unknown directive type "method".

```
.. method:: xmlparser.ProcessingInstructionHandler(target, data)

   Called for every processing instruction.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 369)

Unknown directive type "method".

```
.. method:: xmlparser.CharacterDataHandler(data)

   Called for character data.  This will be called for normal character data, CDATA
   marked content, and ignorable whitespace.  Applications which must distinguish
   these cases can use the :attr:`StartCdataSectionHandler`,
   :attr:`EndCdataSectionHandler`, and :attr:`ElementDeclHandler` callbacks to
   collect the required information.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 378)

Unknown directive type "method".

```
.. method:: xmlparser.UnparsedEntityDeclHandler(entityName, base, systemId, publicId, notationName)

   Called for unparsed (NDATA) entity declarations.  This is only present for
   version 1.2 of the Expat library; for more recent versions, use
   :attr:`EntityDeclHandler` instead.  (The underlying function in the Expat
   library has been declared obsolete.)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 386)

Unknown directive type "method".

```
.. method:: xmlparser.EntityDeclHandler(entityName, is_parameter_entity, value, base, systemId, publicId, no

   Called for all entity declarations.  For parameter and internal entities,
   *value* will be a string giving the declared contents of the entity; this will
   be ``None`` for external entities.  The *notationName* parameter will be
   ``None`` for parsed entities, and the name of the notation for unparsed
   entities. *is_parameter_entity* will be true if the entity is a parameter entity
   or false for general entities (most applications only need to be concerned with
   general entities). This is only available starting with version 1.95.0 of the
   Expat library.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 398)

Unknown directive type "method".

```
.. method:: xmlparser.NotationDeclHandler(notationName, base, systemId, publicId)
```

```
      Called for notation declarations.  *notationName*, *base*, and *systemId*, and
      *publicId* are strings if given.  If the public identifier is omitted,
      *publicId* will be ``None``.
```

```
   .. method:: xmlparser.StartNamespaceDeclHandler(prefix, uri)

      Called when an element contains a namespace declaration.  Namespace declarations
      are processed before the :attr:`StartElementHandler` is called for the element
      on which declarations are placed.
```

```
   .. method:: xmlparser.EndNamespaceDeclHandler(prefix)

      Called when the closing tag is reached for an element  that contained a
      namespace declaration.  This is called once for each namespace declaration on
      the element in the reverse of the order for which the
      :attr:`StartNamespaceDeclHandler` was called to indicate the start of each
      namespace declaration's scope.  Calls to this handler are made after the
      corresponding :attr:`EndElementHandler` for the end of the element.
```

```
   .. method:: xmlparser.CommentHandler(data)

      Called for comments.  *data* is the text of the comment, excluding the leading
      ``'<!-``\ ``-'`` and trailing ``'-``\ ``->'``.
```

```
   .. method:: xmlparser.StartCdataSectionHandler()

      Called at the start of a CDATA section.  This and :attr:`EndCdataSectionHandler`
      are needed to be able to identify the syntactical start and end for CDATA
      sections.
```

```
   .. method:: xmlparser.EndCdataSectionHandler()

      Called at the end of a CDATA section.
```

```
   .. method:: xmlparser.DefaultHandler(data)

      Called for any characters in the XML document for which no applicable handler
      has been specified.  This means characters that are part of a construct which
      could be reported, but for which no handler has been supplied.
```

```
   .. method:: xmlparser.DefaultHandlerExpand(data)
```

This is the same as the :func:`DefaultHandler`, but doesn't inhibit expansion
of internal entities. The entity reference will not be passed to the default
handler.

```
.. method:: xmlparser.NotStandaloneHandler()

   Called if the XML document hasn't been declared as being a standalone document.
   This happens when there is an external subset or a reference to a parameter
   entity, but the XML declaration does not set standalone to ``yes`` in an XML
   declaration.  If this handler returns ``0``, then the parser will raise an
   :const:`XML_ERROR_NOT_STANDALONE` error.  If this handler is not set, no
   exception is raised by the parser for this condition.
```

```
.. method:: xmlparser.ExternalEntityRefHandler(context, base, systemId, publicId)

   Called for references to external entities.  *base* is the current base, as set
   by a previous call to :meth:`SetBase`.  The public and system identifiers,
   *systemId* and *publicId*, are strings if given; if the public identifier is not
   given, *publicId* will be ``None``.  The *context* value is opaque and should
   only be used as described below.

   For external entities to be parsed, this handler must be implemented. It is
   responsible for creating the sub-parser using
   ``ExternalEntityParserCreate(context)``, initializing it with the appropriate
   callbacks, and parsing the entity.  This handler should return an integer; if it
   returns ``0``, the parser will raise an
   :const:`XML_ERROR_EXTERNAL_ENTITY_HANDLING` error, otherwise parsing will
   continue.

   If this handler is not provided, external entities are reported by the
   :attr:`DefaultHandler` callback, if provided.
```

ExpatError Exceptions

```
.. sectionauthor:: Fred L. Drake, Jr. <fdrake@acm.org>
```

:exc:`ExpatError` exceptions have a number of interesting attributes:

```
.. attribute:: ExpatError.code

   Expat's internal error number for the specific error.  The
   :data:`errors.messages <xml.parsers.expat.errors.messages>` dictionary maps
   these error numbers to Expat's error messages.  For example::

      from xml.parsers.expat import ParserCreate, ExpatError, errors

      p = ParserCreate()
      try:
          p.Parse(some_xml_document)
      except ExpatError as err:
          print("Error:", errors.messages[err.code])

   The :mod:`~xml.parsers.expat.errors` module also provides error message
   constants and a dictionary :data:`~xml.parsers.expat.errors.codes` mapping
   these messages back to the error codes, see below.
```

Example

The following program defines three handlers that just print out their arguments.

```python
import xml.parsers.expat

# 3 handler functions
def start_element(name, attrs):
    print('Start element:', name, attrs)
def end_element(name):
    print('End element:', name)
def char_data(data):
    print('Character data:', repr(data))

p = xml.parsers.expat.ParserCreate()

p.StartElementHandler = start_element
p.EndElementHandler = end_element
p.CharacterDataHandler = char_data

p.Parse("""<?xml version="1.0"?>
<parent id="top"><child1 name="paul">Text goes here</child1>
<child2 name="fred">More text</child2>
</parent>""", 1)
```

The output from this program is:

```
Start element: parent {'id': 'top'}
Start element: child1 {'name': 'paul'}
Character data: 'Text goes here'
End element: child1
Character data: '\n'
Start element: child2 {'name': 'fred'}
Character data: 'More text'
End element: child2
Character data: '\n'
End element: parent
```

Content Model Descriptions

Content models are described using nested tuples. Each tuple contains four values: the type, the quantifier, the name, and a tuple of children. Children are simply additional content model descriptions.

The values of the first two fields are constants defined in the :mod:`xml.parsers.expat.model` module. These constants can be collected in two groups: the model type group and the quantifier group.

The constants in the model type group are:

The constants in the quantifier group are:

Unknown directive type "data".

```
.. data:: XML_CQUANT_OPT
   :noindex:

   The model is optional: it can appear once or not at all, as for ``A?``.
```

```
.. data:: XML_CQUANT_PLUS
   :noindex:

   The model must occur one or more times (like ``A+``).
```

```
.. data:: XML_CQUANT_REP
   :noindex:

   The model must occur zero or more times, as for ``A*``.
```

Expat error constants

```
.. module:: xml.parsers.expat.errors
```

The following constants are provided in the :mod:`xml.parsers.expat.errors` module. These constants are useful in interpreting some of the attributes of the :exc:`ExpatError` exception objects raised when an error has occurred. Since for backwards compatibility reasons, the constants' value is the error *message* and not the numeric error *code*, you do this by comparing its :attr:`code` attribute with :samp:`errors.codes[errors.XML_ERROR_{CONSTANT_NAME}]`.

The `errors` module has the following attributes:

```
.. data:: codes

   A dictionary mapping string descriptions to their error codes.

   .. versionadded:: 3.2
```

Unknown directive type "data".

```
.. data:: messages

   A dictionary mapping numeric error codes to their string descriptions.

   .. versionadded:: 3.2
```

Unknown directive type "data".

```
.. data:: XML_ERROR_ASYNC_ENTITY
```

Unknown directive type "data".

```
.. data:: XML_ERROR_ATTRIBUTE_EXTERNAL_ENTITY_REF

   An entity reference in an attribute value referred to an external entity instead
   of an internal entity.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_BAD_CHAR_REF

   A character reference referred to a character which is illegal in XML (for
   example, character ``0``, or '``&#0;``').
```

Unknown directive type "data".

```
.. data:: XML_ERROR_BINARY_ENTITY_REF

   An entity reference referred to an entity which was declared with a notation, so
   cannot be parsed.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_DUPLICATE_ATTRIBUTE

   An attribute was used more than once in a start tag.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_INCORRECT_ENCODING
```

Unknown directive type "data".

```
.. data:: XML_ERROR_INVALID_TOKEN

   Raised when an input byte could not properly be assigned to a character; for
   example, a NUL byte (value ``0``) in a UTF-8 input stream.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_JUNK_AFTER_DOC_ELEMENT

    Something other than whitespace occurred after the document element.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 720)`**)**

Unknown directive type "data".

```
.. data:: XML_ERROR_MISPLACED_XML_PI

    An XML declaration was found somewhere other than the start of the input data.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 725)`**)**

Unknown directive type "data".

```
.. data:: XML_ERROR_NO_ELEMENTS

    The document contains no elements (XML requires all documents to contain exactly
    one top-level element)..
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 731)`**)**

Unknown directive type "data".

```
.. data:: XML_ERROR_NO_MEMORY

    Expat was not able to allocate memory internally.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 736)`**)**

Unknown directive type "data".

```
.. data:: XML_ERROR_PARAM_ENTITY_REF

    A parameter entity reference was found where it was not allowed.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 741)`**)**

Unknown directive type "data".

```
.. data:: XML_ERROR_PARTIAL_CHAR

    An incomplete character was found in the input.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 746)`**)**

Unknown directive type "data".

```
.. data:: XML_ERROR_RECURSIVE_ENTITY_REF

    An entity reference contained another reference to the same entity; possibly via
    a different name, and possibly indirectly.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)pyexpat.rst, line 752)`**)**

Unknown directive type "data".

```
.. data:: XML_ERROR_SYNTAX

    Some unspecified syntax error was encountered.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-`

Unknown directive type "data".

```
.. data:: XML_ERROR_TAG_MISMATCH

    An end tag did not match the innermost open start tag.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_UNCLOSED_TOKEN

    Some token (such as a start tag) was not closed before the end of the stream or
    the next token was encountered.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_UNDEFINED_ENTITY

    A reference was made to an entity which was not defined.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_UNKNOWN_ENCODING

    The document encoding is not supported by Expat.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_UNCLOSED_CDATA_SECTION

    A CDATA marked section was not closed.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_EXTERNAL_ENTITY_HANDLING
```

Unknown directive type "data".

```
.. data:: XML_ERROR_NOT_STANDALONE

    The parser determined that the document was not "standalone" though it declared
    itself to be in the XML declaration, and the :attr:`NotStandaloneHandler` was
    set and returned ``0``.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_UNEXPECTED_STATE
```

Unknown directive type "data".

```
            .. data:: XML_ERROR_ENTITY_DECLARED_IN_PE
```

```
    .. data:: XML_ERROR_FEATURE_REQUIRES_XML_DTD

        An operation was requested that requires DTD support to be compiled in, but
        Expat was configured without DTD support.  This should never be reported by a
        standard build of the :mod:`xml.parsers.expat` module.
```

```
    .. data:: XML_ERROR_CANT_CHANGE_FEATURE_ONCE_PARSING

        A behavioral change was requested after parsing started that can only be changed
        before parsing has started.  This is (currently) only raised by
        :meth:`UseForeignDTD`.
```

```
    .. data:: XML_ERROR_UNBOUND_PREFIX

        An undeclared prefix was found when namespace processing was enabled.
```

```
    .. data:: XML_ERROR_UNDECLARING_PREFIX

        The document attempted to remove the namespace declaration associated with a
        prefix.
```

```
    .. data:: XML_ERROR_INCOMPLETE_PE

        A parameter entity contained incomplete markup.
```

```
    .. data:: XML_ERROR_XML_DECL

        The document contained no document element at all.
```

```
    .. data:: XML_ERROR_TEXT_DECL

        There was an error parsing a text declaration in an external entity.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_PUBLICID
```

```
   Characters were found in the public id that are not allowed.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_SUSPENDED
```

```
   The requested operation was made on a suspended parser, but isn't allowed.  This
   includes attempts to provide additional input or to stop the parser.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_NOT_SUSPENDED
```

```
   An attempt to resume the parser was made when the parser had not been suspended.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_ABORTED
```

```
   This should not be reported to Python applications.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_FINISHED
```

```
   The requested operation was made on a parser which was finished parsing input,
   but isn't allowed.  This includes attempts to provide additional input or to
   stop the parser.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_SUSPEND_PE
```

Unknown directive type "data".

```
.. data:: XML_ERROR_RESERVED_PREFIX_XML
```

```
   An attempt was made to
   undeclare reserved namespace prefix ``xml``
   or to bind it to another namespace URI.
```

Unknown directive type "data".

```
.. data:: XML_ERROR_RESERVED_PREFIX_XMLNS
```

```
   An attempt was made to declare or undeclare reserved namespace prefix ``xmlns``.
```

main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, line 882)

Unknown directive type "data".

```
.. data:: XML_ERROR_RESERVED_NAMESPACE_URI

   An attempt was made to bind the URI of one the reserved namespace
   prefixes ``xml`` and ``xmlns`` to another namespace prefix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, line 888)

Unknown directive type "data".

```
.. data:: XML_ERROR_INVALID_ARGUMENT

   This should not be reported to Python applications.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, line 893)

Unknown directive type "data".

```
.. data:: XML_ERROR_NO_BUFFER

   This should not be reported to Python applications.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)pyexpat.rst, line 898)

Unknown directive type "data".

```
.. data:: XML_ERROR_AMPLIFICATION_LIMIT_BREACH

   The limit on input amplification factor (from DTD and entities)
   has been breached.
```

Footnotes

[1] The encoding string included in XML output should conform to the appropriate standards. For example, "UTF-8" is valid, but "UTF8" is not. See https://www.w3.org/TR/2006/REC-xml11-20060816/#NT-EncodingDecl and https://www.iana.org/assignments/character-sets/character-sets.xhtml.