

XML exceptions that do not allow continuation

With these XML exceptions, no further events are returned from the parser, even if you set `XML-CODE` to zero and return control to the parser after processing the exception. Control is passed to the statement that you specify on your `NOT ON EXCEPTION` phrase or to the end of the parse statement if you have not coded a `NOT ON EXCEPTION` phrase.

Table 32. XML exceptions that do not allow continuation

Code	Description
100	The parser reached the end of the document while scanning the start of the XML declaration.
101	The parser reached the end of the document while looking for the end of the XML declaration.
102	The parser reached the end of the document while looking for the root element.
103	The parser reached the end of the document while looking for the version information in the XML declaration.
104	The parser reached the end of the document while looking for the version information value in the XML declaration.
106	The parser reached the end of the document while looking for the encoding declaration value in the XML declaration.
108	The parser reached the end of the document while looking for the standalone declaration value in the XML declaration.
109	The parser reached the end of the document while scanning an attribute name.
110	The parser reached the end of the document while scanning an attribute value.
111	The parser reached the end of the document while scanning a character reference or entity reference in an attribute value.
112	The parser reached the end of the document while scanning an empty element tag.
113	The parser reached the end of the document while scanning the root element name.
114	The parser reached the end of the document while scanning an element name.
115	The parser reached the end of the document while scanning character data in element content.
116	The parser reached the end of the document while scanning a processing instruction in element content.
117	The parser reached the end of the document while scanning a comment or CDATA section in element content.
118	The parser reached the end of the document while scanning a comment in element content.
119	The parser reached the end of the document while scanning a CDATA section in element content.
120	The parser reached the end of the document while scanning a character reference or entity reference in element content.
121	The parser reached the end of the document while scanning after the close of the root element.
122	The parser found a possible invalid start of a document type declaration.
123	The parser found a second document type declaration.
124	The first character of the root element name was not a letter, '_', or ':'.
125	The first character of the first attribute name of an element was not a letter, '_', or ':'.
126	The parser found an invalid character either in or following an element name.
127	The parser found a character other than '=' following an attribute name.
128	The parser found an invalid attribute value delimiter.
130	The first character of an attribute name was not a letter, '_', or ':'.
131	The parser found an invalid character either in or following an attribute name.
132	An empty element tag was not terminated by a '>' following the '/'.
133	The first character of an element end tag name was not a letter, '_', or ':'.
134	An element end tag name was not terminated by a '>'.
135	The first character of an element name was not a letter, '_', or ':'.
136	The parser found an invalid start of a comment or CDATA section in element content.
137	The parser found an invalid start of a comment.
138	The first character of a processing instruction target name was not a letter, '_', or ':'.
139	The parser found an invalid character in or following a processing instruction target name.
140	A processing instruction was not terminated by the closing character sequence '?>'.
141	The parser found an invalid character following '&' in a character reference or entity reference.
142	The version information was not present in the XML declaration.
143	'version' in the XML declaration was not followed by '='.
144	The version declaration value in the XML declaration is either missing or improperly delimited.
145	The version information value in the XML declaration specified a bad character, or the start and end delimiters did not match.
146	The parser found an invalid character following the version information value closing delimiter in the XML declaration.
147	The parser found an invalid attribute instead of the optional encoding declaration in the XML declaration.
148	'encoding' in the XML declaration was not followed by '='.
149	The encoding declaration value in the XML declaration is either missing or improperly delimited.
150	The encoding declaration value in the XML declaration specified a bad character, or the start and end delimiters did not match.
151	The parser found an invalid character following the encoding declaration value closing delimiter in the XML declaration.
152	The parser found an invalid attribute instead of the optional standalone declaration in the XML declaration.
153	'standalone' in the XML declaration was not followed by '='.
154	The standalone declaration value in the XML declaration is either missing or improperly delimited.
155	The standalone declaration value was neither 'yes' nor 'no' only.
156	The standalone declaration value in the XML declaration specified a bad character, or the start and end delimiters did not match.

Code	Description
157	The parser found an invalid character following the standalone declaration value closing delimiter in the XML declaration.
158	The XML declaration was not terminated by the proper character sequence '?>', or contained an invalid attribute.
159	The parser found the start of a document type declaration after the end of the root element.
160	The parser found the start of an element after the end of the root element.
300	The document was encoded in EBCDIC, but the CCSID of the COBOL source member is a supported ASCII CCSID.
301	The document was encoded in EBCDIC, but the CCSID of the COBOL source member is Unicode.
302	The document was encoded in EBCDIC, but the CCSID of the COBOL source member is an unsupported CCSID.
303	The document was encoded in EBCDIC, but CCSID of the COBOL source member is unsupported and the document encoding declaration was either empty or contained an unsupported alphabetic encoding alias.
304	The document was encoded in EBCDIC, but the CCSID of the COBOL source member is unsupported and the document did not contain an encoding declaration.
305	The document was encoded in EBCDIC, but the CCSID of the COBOL source member is unsupported and the document encoding declaration did not specify a supported EBCDIC encoding.
306	The document was encoded in ASCII, but the CCSID of the COBOL source member is a supported EBCDIC CCSID.
307	The document was encoded in ASCII, but the CCSID of the COBOL source member is Unicode.
308	The document was encoded in ASCII, but the CCSID of the COBOL source member is unsupported and the document did not contain an encoding declaration.
309	The CCSID of the COBOL source member is a supported ASCII CCSID, but the document was encoded in Unicode.
310	The CCSID of the COBOL source member specified a supported EBCDIC CCSID, but the document was encoded in Unicode.
311	The CCSID of the COBOL source member specified an unsupported CCSID and the document was encoded in Unicode.
312	The document was encoded in ASCII, but the CCSID of the COBOL source member is unsupported and the document encoding declaration was either empty or contained an unsupported alphabetic encoding alias.
313	The document was encoded in ASCII, but the CCSID of the COBOL source member is unsupported and the document did not contain an encoding declaration.
314	The document was encoded in ASCII, but the CCSID of the COBOL source member is unsupported and the document encoding declaration did not specify a supported ASCII encoding.
315	The document was encoded in UTF-16 Little Endian, which the parser does not support on this platform.
316	The document was encoded in UCS4, which the parser does not support.
317	The parser cannot determine the document encoding. The document may be damaged.
318	The document was encoded in UTF-8, which the parser does not support.
319	The document was encoded in UTF-16 Big Endian, which the parser does not support on this platform.
500-999	Internal error. Please report the error to your service representative.

RELATED TASKS

[Handling errors in XML documents](#)

(C) Copyright IBM Corporation 1992, 2006. All Rights Reserved.