

DBMS_XMLPARSER

Using `DBMS_XMLPARSER`, you can access the contents and structure of XML documents. XML describes a class of data XML document objects. It partially describes the behavior of computer programs which process them. By construction, XML documents are conforming SGML documents.

XML documents are made up of storage units called entities, which contain either parsed or unparsed data. Parsed data is made up of characters, some of which form character data, and some of which form markup. Markup encodes a description of the document's storage layout and logical structure. XML provides a mechanism to impose constraints on the storage layout and logical structure.

A software module called an XML processor is used to read XML documents and provide access to their content and structure. It is assumed that an XML processor is doing its work on behalf of another module, called the application. This PL/SQL implementation of the XML processor (or parser) follows the W3C XML specification REC-xml-19980210 and includes the required behavior of an XML processor in terms of how it must read XML data and the information it must provide to the application.

The default behavior for this PL/SQL XML parser is to build a parse tree that can be accessed by DOM APIs, validate it if a DTD is found (otherwise, it is non-validating), and record errors if an error log is specified. If parsing fails, an application error is raised.

This chapter contains the following topics:

- [Security Model](#)
- [Summary of DBMS_XMLPARSER Subprograms](#)



See Also:

Oracle XML DB Developer's Guide

DBMS_XMLPARSER Security Model

Owned by `XDB`, the `DBMS_XMLPARSER` package must be created by `SYS` or `XDB`. The `EXECUTE` privilege is granted to `PUBLIC`.

Subprograms in this package are executed using the privileges of the current user.

Summary of DBMS_XMLPARSER Subprograms

This table lists the `DBMS_XMLPARSER` subprograms and briefly describes them.

Table 236-1 DBMS_XMLPARSER Package Subprograms

Method	Description
FREEPARSER	Frees a parser object.
GETDOCTYPE	Gets parsed DTD.
GETDOCUMENT	Gets DOM document.
GETRELEASEVERSION	Returns the release version of Oracle XML Parser for PL/SQL.
GETVALIDATIONMODE	Returns validation mode.
NEWPARSER	Returns a new parser instance
PARSE	Parses XML stored in the given url/file.
PARSEBUFFER	Parses XML stored in the given buffer
PARSECLOB	Parses XML stored in the given clob
PARSEDTD	Parses DTD stored in the given url/file
PARSEDTCBUFFER	Parses DTD stored in the given buffer
PARSEDTCLOB	Parses DTD stored in the given clob
SETBASEDIR	Sets base directory used to resolve relative URLs.
SETDOCTYPE	Sets DTD.
SETERRORLOG	Sets errors to be sent to the specified file
SETPRESERVEWHITESPACE	Sets white space preserve mode
SETVALIDATIONMODE	Sets validation mode.
SHOWWARNINGS	Turns warnings on or off.

FREEPARSER

This procedure frees a parser object.

Syntax

```
PROCEDURE freeParser(  
    p Parser);
```

Parameters

Table 236-2 FREEPARSER Procedure Parameters

Parameter	IN / OUT	Description
p	(IN)	Parser instance.

GETDOCTYPE

The `GETDOCTYPE` function returns the parsed DTD. This function must be called only after a DTD is parsed.

Syntax

```
FUNCTION getDoctype(  
    p Parser)  
RETURN DOMDocumentType;
```

Parameters

Table 236-3 GETDOCTYPE Function Parameters

Parameter	IN / OUT	Description
p	(IN)	Parser instance.

GETDOCUMENT

`GETDOCUMENT` returns the document node of a DOM tree document built by the parser. This function must be called only after a document is parsed.

Syntax

```
FUNCTION GETDOCUMENT(  
    p Parser)  
RETURN DOMDocument;
```

Parameters

Table 236-4 GETDOCUMENT Function Parameters

Parameter	IN / OUT	Description
p	(IN)	Parser instance.

GETRELEASEVERSION

`GETRELEASEVERSION` returns the release version of the Oracle XML parser for PL/SQL.

Syntax

```
FUNCTION getReleaseVersion  
RETURN VARCHAR2;
```

GETVALIDATIONMODE

The `GETVALIDATIONMODE` function retrieves the validation mode: `TRUE` for validating, `FALSE` otherwise.

Syntax

```
FUNCTION GETVALIDATIONMODE (  
    p Parser)  
RETURN BOOLEAN;
```

Parameters

Table 236-5 GETVALIDATIONMODE Function Parameters

Parameter	IN / OUT	Description
p	(IN)	Parser instance.

NEWPARSER

This function returns a new parser instance.

This function must be called before the default behavior of Parser can be changed and if other parse methods need to be used.

Syntax

```
FUNCTION newParser  
RETURN Parser;
```

PARSE

`PARSE` parses XML stored in the given URL or file. An application error is raised if parsing fails.

There are several versions of this method.

Syntax

Function. Use this when the default parser behavior is acceptable, and only a URL or file needs to be parsed. Returns the built DOM document.

```
FUNCTION parse(url VARCHAR2)  
RETURN DOMDocument;
```

Procedure. Any changes to the default parser behavior should be effected before calling this procedure.

```
PROCEDURE parse(  
    p Parser,  
    url VARCHAR2);
```

Parameters

Table 236-6 PARSE Subprogram Parameters

Parameter	IN / OUT	Description
url	(IN)	Complete path of the url/file to be parsed.
p	(IN)	Parser instance.

PARSEBUFFER

PARSEBUFFER parses XML stored in the given buffer.

Any changes to the default parser behavior should be effected before calling this procedure.
An application error is raised if parsing fails.

Syntax

```
PROCEDURE PARSEBUFFER(  
    p    Parser,  
    doc VARCHAR2);
```

Parameters

Table 236-7 PARSEBUFFER Procedure Parameters

Parameter	IN / OUT	Description
p	(IN)	Parser instance.
doc	(IN)	XML document buffer to parse.

PARSECLOB

PARSECLOB parses XML stored in the given clob.

Any changes to the default parser behavior should be effected before calling this procedure.
An application error is raised if parsing fails.

Syntax

```
PROCEDURE PARSECLOB(  
    p    Parser,  
    doc CLOB);
```

Parameters

Table 236-8 PARSECLOB Procedure Parameters

Parameter	IN / OUT	Description
p	(IN)	Parser instance.

Table 236-8 (Cont.) PARSECLOB Procedure Parameters

Parameter	IN / OUT	Description
doc	(IN)	XML document buffer to parse.

PARSEDTD

PARSEDTD parses the DTD stored in the given URL or file.

Any changes to the default parser behavior should be effected before calling this procedure. An application error is raised if parsing fails.

Syntax

```
PROCEDURE PARSEDTD(  
    p      Parser,  
    url    VARCHAR2,  
    root   VARCHAR2);
```

Parameters

Table 236-9 PARSEDTD Procedure Parameters

Parameter	IN / OUT	Description
p	(IN)	Parser instance.
url	(IN)	Complete path of the URL or file to be parsed.
root	(IN)	Name of the root element.

PARSEDTDBUFFER

PARSEDTDBUFFER parses the DTD stored in the given buffer.

Any changes to the default parser behavior should be effected before calling this procedure. An application error is raised if parsing fails.

Syntax

```
PROCEDURE PARSEDTDBUFFER(  
    p      Parser,  
    dtd    VARCHAR2,  
    root   VARCHAR2);
```

Parameters

Table 236-10 PARSEDTDBUFFER Procedure Parameters

Parameter	IN / OUT	Description
p	(IN)	Parser instance.

Table 236-10 (Cont.) PARSEDTDBUFFER Procedure Parameters

Parameter	IN / OUT	Description
dtd	(IN)	DTD buffer to parse.
root	(IN)	Name of the root element.

PARSEDTDCLOB

PARSEDTDCLOB parses the DTD stored in the given clob.

Any changes to the default parser behavior should be effected before calling this procedure. An application error is raised if parsing fails.

Syntax

```
PROCEDURE PARSEDTDCLOB(  
    p      Parser,  
    dtd    CLOB,  
    root   VARCHAR2);
```

Parameters

Table 236-11 PARSEDTDCLOB Procedure Parameters

Parameter	IN / OUT	Description
p	(IN)	Parser instance.
dtd	(IN)	DTD Clob to parse.
root	(IN)	Name of the root element.

SETBASEDIR

This procedure sets the base directory used to resolve relative URLs. An application error is raised if parsing fails.

Syntax

```
PROCEDURE setBaseDir(  
    p      Parser,  
    dir    VARCHAR2);
```

Parameters

Table 236-12 SETBASEDIR Procedure Parameters

Parameter	IN / OUT	Description
p	(IN)	Parser instance.
dir	(IN)	Directory used as a base directory.

SETDOCTYPE Procedure

This procedure sets a DTD to be used by the parser for validation. This call should be made before the document is parsed.

Syntax

```
PROCEDURE setDoctype(  
    p    Parser,  
    dtd DOMDocumentType);
```

Parameters

Table 236-13 SETDOCTYPE Procedure Parameters

Parameter	IN / OUT	Description
p	(IN)	Parser instance.
dtd	(IN)	DTD to set.

SETERRORLOG Procedure

This procedure sets errors to be sent to the specified file.

Syntax

```
PROCEDURE setErrorLog(  
    p          Parser,  
    fileName VARCHAR2);
```

Parameters

Table 236-14 SETERRORLOG Procedure Parameters

Parameter	IN / OUT	Description
p	(IN)	Parser instance.
fileName	(IN)	Complete path of the file to use as the error log.

SETPRESERVEWHITESPACE

This procedure sets whitespace preserving mode.

Syntax

```
PROCEDURE setPreserveWhitespace(  
    p    Parser,  
    yes BOOLEAN);
```


Parameters

Table 236-15 SETPRESERVEWHITESPACE Procedure Parameters

Parameter	IN / OUT	Description
p	(IN)	Parser instance.
yes	(IN)	Mode to set: TRUE - preserve, FALSE - don't preserve.

SETVALIDATIONMODE

This procedure sets the validation mode.

Syntax

```
PROCEDURE setValidationMode(  
    p    Parser,  
    yes BOOLEAN);
```

Parameters

Table 236-16 SETVALIDATIONMODE Procedure Parameters

Parameter	IN / OUT	Description
p	(IN)	Parser instance.
yes	(IN)	Mode to set: TRUE - validate, FALSE - don't validate.

SHOWWARNINGS

This procedure turns warnings on or off.

Syntax

```
PROCEDURE showWarnings(  
    p    Parser,  
    yes BOOLEAN);
```

Parameters

Table 236-17 SHOWWARNINGS Procedure Parameters

Parameter	IN / OUT	Description
p	(IN)	Parser instance.
yes	(IN)	Mode to set: TRUE - show warnings, FALSE - don't show warnings.