

# **PowerBuilder Extension Reference**

*Appeon PowerBuilder® 2021*

# Contents

1	PowerBuilder Extensions .....	9
1.1	About PowerBuilder extensions .....	9
1.2	Using PowerBuilder extensions .....	9
1.3	Getting information about PowerBuilder extensions .....	11
2	EJB Client (obsolete) .....	12
3	Web Services Client (Obsolete) .....	13
4	PowerBuilder Document Object Model .....	14
4.1	About PBDOM .....	14
4.1.1	Node trees .....	14
4.1.2	XML parser .....	14
4.1.3	Objects and methods .....	14
4.1.4	PBDOM objects .....	15
5	PBDOM_ATTRIBUTE Class .....	17
5.1	PBDOM_ATTRIBUTE .....	17
5.1.1	AddContent .....	18
5.1.2	Clone .....	19
5.1.3	Detach .....	20
5.1.4	Equals .....	21
5.1.5	GetBooleanValue .....	23
5.1.6	GetContent .....	24
5.1.7	GetDateValue .....	25
5.1.8	GetDateTimeValue .....	26
5.1.9	GetDoubleValue .....	27
5.1.10	GetIntValue .....	27
5.1.11	GetLongValue .....	28
5.1.12	GetName .....	28
5.1.13	GetNamespacePrefix .....	29
5.1.14	GetNamespaceUri .....	30
5.1.15	GetObjectClass .....	30
5.1.16	GetObjectClassString .....	31
5.1.17	GetOwnerDocumentObject .....	32
5.1.18	GetOwnerElementObject .....	33
5.1.19	GetQualifiedName .....	34
5.1.20	GetRealValue .....	35
5.1.21	GetText .....	35
5.1.22	GetTextNormalize .....	37
5.1.23	GetTextTrim .....	39
5.1.24	GetTimeValue .....	40
5.1.25	GetUIntValue .....	41
5.1.26	GetUlongValue .....	41
5.1.27	HasChildren .....	42
5.1.28	InsertContent .....	43
5.1.29	IsAncestorObjectOf .....	45
5.1.30	RemoveContent .....	46
5.1.31	SetBooleanValue .....	48
5.1.32	SetContent .....	48

5.1.33	SetDateValue .....	51
5.1.34	SetDateTimeValue .....	52
5.1.35	SetDoubleValue .....	53
5.1.36	SetIntValue .....	54
5.1.37	SetLongValue .....	54
5.1.38	SetName .....	55
5.1.39	SetNamespace .....	57
5.1.40	SetOwnerElementObject .....	59
5.1.41	SetRealValue .....	61
5.1.42	SetText .....	61
5.1.43	SetTimeValue .....	62
5.1.44	SetUIntValue .....	63
5.1.45	SetUlongValue .....	63
6	PBDOM_BUILDER Class .....	65
6.1	PBDOM_BUILDER .....	65
6.1.1	BuildFromDataStore .....	65
6.1.2	BuildFromFile .....	66
6.1.3	BuildFromString .....	68
6.1.4	GetParseErrors .....	69
7	PBDOM_CDATA Class .....	72
7.1	PBDOM_CDATA .....	72
7.1.1	Append .....	73
7.1.2	Clone .....	74
7.1.3	Detach .....	75
7.1.4	Equals .....	76
7.1.5	GetObjectClass .....	76
7.1.6	GetObjectClassString .....	77
7.1.7	GetOwnerDocumentObject .....	77
7.1.8	GetParentObject .....	78
7.1.9	GetText .....	78
7.1.10	GetTextNormalize .....	79
7.1.11	GetTextTrim .....	79
7.1.12	SetParentObject .....	80
7.1.13	SetText .....	80
8	PBDOM_ENTITYREFERENCE Class .....	82
8.1	PBDOM_ENTITYREFERENCE .....	82
8.1.1	Clone .....	82
8.1.2	Detach .....	84
8.1.3	Equals .....	85
8.1.4	GetName .....	85
8.1.5	GetObjectClass .....	86
8.1.6	GetObjectClassString .....	86
8.1.7	GetOwnerDocumentObject .....	86
8.1.8	GetParentObject .....	87
8.1.9	SetName .....	87
8.1.10	SetParentObject .....	88
9	PBDOM_CHARACTERDATA Class .....	90
9.1	PBDOM_CHARACTERDATA .....	90

9.1.1	Append .....	91
9.1.1.1	Append Syntax 1 .....	91
9.1.1.2	Append Syntax 2 .....	94
9.1.2	Clone .....	94
9.1.3	Detach .....	96
9.1.4	Equals .....	98
9.1.5	GetOwnerDocumentObject .....	98
9.1.6	GetName .....	101
9.1.7	GetObjectClass .....	102
9.1.8	GetObjectClassString .....	102
9.1.9	GetParentObject .....	104
9.1.10	GetText .....	106
9.1.11	GetTextNormalize .....	107
9.1.12	GetTextTrim .....	111
9.1.13	HasChildren .....	114
9.1.14	IsAncestorObjectOf .....	115
9.1.15	SetParentObject .....	116
9.1.16	SetText .....	118
10	PBDOM_COMMENT Class .....	119
10.1	PBDOM_COMMENT .....	119
10.1.1	Append .....	120
10.1.1.1	Append Syntax 1 .....	120
10.1.1.2	Append Syntax 2 .....	121
10.1.2	Clone .....	121
10.1.3	Detach .....	123
10.1.4	Equals .....	124
10.1.5	GetObjectClass .....	124
10.1.6	GetObjectClassString .....	125
10.1.7	GetOwnerDocumentObject .....	125
10.1.8	GetParentObject .....	126
10.1.9	GetText .....	126
10.1.10	GetTextNormalize .....	127
10.1.11	GetTextTrim .....	127
10.1.12	SetParentObject .....	128
10.1.13	SetText .....	129
11	PBDOM_DOCTYPE Class .....	131
11.1	PBDOM_DOCTYPE .....	131
11.1.1	Clone .....	131
11.1.2	Detach .....	132
11.1.3	Equals .....	132
11.1.4	GetInternalSubset .....	133
11.1.5	GetName .....	133
11.1.6	GetObjectClass .....	134
11.1.7	GetObjectClassString .....	134
11.1.8	GetOwnerDocumentObject .....	135
11.1.9	GetParentObject .....	135
11.1.10	GetPublicID .....	135
11.1.11	GetSystemID .....	136

11.1.12	SetDocument .....	137
11.1.13	SetInternalSubset .....	137
11.1.14	SetName .....	138
11.1.15	SetParentObject .....	139
11.1.16	SetPublicID .....	140
11.1.17	SetSystemID .....	141
12	PBDOM_DOCUMENT Class .....	142
12.1	PBDOM_DOCUMENT .....	142
12.1.1	AddContent .....	143
12.1.2	Clone .....	145
12.1.3	DetachRootElement .....	146
12.1.4	Equals .....	147
12.1.5	GetContent .....	147
12.1.6	GetDocType .....	149
12.1.7	GetElementsByTagName .....	149
12.1.8	GetObjectClass .....	150
12.1.9	GetObjectClassString .....	151
12.1.10	GetRootElement .....	151
12.1.11	HasChildren .....	152
12.1.12	HasRootElement .....	152
12.1.13	InsertContent .....	153
12.1.14	IsAncestorObjectOf .....	155
12.1.15	NewDocument .....	155
12.1.15.1	NewDocument Syntax 1 .....	156
12.1.15.2	NewDocument Syntax 2 .....	156
12.1.16	RemoveContent .....	159
12.1.17	SaveDocument .....	160
12.1.18	SaveDocumentIntoString .....	161
12.1.19	SetContent .....	161
12.1.20	SetDocType .....	163
12.1.21	SetRootElement .....	164
13	PBDOM_ELEMENT Class .....	165
13.1	PBDOM_ELEMENT .....	165
13.1.1	AddContent .....	166
13.1.1.1	AddContent Syntax 1 .....	166
13.1.1.2	AddContent Syntax 2 .....	168
13.1.2	AddNamespaceDeclaration .....	169
13.1.3	Clone .....	170
13.1.4	Detach .....	172
13.1.5	Equals .....	172
13.1.6	GetAttribute .....	173
13.1.6.1	GetAttribute Syntax 1 .....	173
13.1.6.2	GetAttribute Syntax 2 .....	174
13.1.7	GetAttributes .....	175
13.1.8	GetAttributeValue .....	176
13.1.8.1	GetAttributeValue Syntax 1 .....	177
13.1.8.2	GetAttributeValue Syntax 2 .....	178
13.1.8.3	GetAttributeValue Syntax 3 .....	179

13.1.8.4 GetAttributeValue Syntax 4 .....	180
13.1.9 GetChildElement .....	180
13.1.9.1 GetChildElement Syntax 1 .....	181
13.1.9.2 GetChildElement Syntax 2 .....	182
13.1.10 GetChildElements .....	182
13.1.10.1 GetChildElements Syntax 1 .....	183
13.1.10.2 GetChildElements Syntax 2 .....	184
13.1.10.3 GetChildElements Syntax 3 .....	185
13.1.11 GetContent .....	186
13.1.12 GetName .....	187
13.1.13 GetNamespacePrefix .....	188
13.1.14 GetNamespaceUri .....	188
13.1.15 GetObjectClass .....	189
13.1.16 GetObjectClassString .....	190
13.1.17 GetOwnerDocumentObject .....	190
13.1.18 GetParentObject .....	191
13.1.19 GetQualifiedName .....	192
13.1.20 GetText .....	192
13.1.21 GetTextNormalize .....	193
13.1.22 GetTextTrim .....	194
13.1.23 HasAttributes .....	194
13.1.24 HasChildElements .....	195
13.1.25 HasChildren .....	196
13.1.26 InsertContent .....	197
13.1.27 IsAncestorObjectOf .....	199
13.1.28 IsRootElement .....	199
13.1.29 RemoveAttribute .....	200
13.1.29.1 RemoveAttribute Syntax 1 .....	200
13.1.29.2 RemoveAttribute Syntax 2 .....	201
13.1.29.3 RemoveAttribute Syntax 3 .....	201
13.1.30 RemoveChildElement .....	202
13.1.30.1 RemoveChildElement Syntax 1 .....	202
13.1.30.2 RemoveChildElement Syntax 2 .....	203
13.1.31 RemoveChildElements .....	204
13.1.31.1 RemoveChildElements Syntax 1 .....	204
13.1.31.2 RemoveChildElements Syntax 2 .....	205
13.1.31.3 RemoveChildElements Syntax 3 .....	206
13.1.32 RemoveContent .....	207
13.1.33 RemoveNamespaceDeclaration .....	208
13.1.34 SetAttribute .....	209
13.1.34.1 SetAttribute Syntax 1 .....	209
13.1.34.2 SetAttribute Syntax 2 .....	211
13.1.34.3 SetAttribute Syntax 3 .....	213
13.1.35 SetAttributes .....	216
13.1.36 SetContent .....	218
13.1.37 SetDocument .....	221
13.1.38 SetName .....	221
13.1.39 SetNamespace .....	222

13.1.40	SetParentObject .....	223
13.1.41	SetText .....	224
14	PBDOM_EXCEPTION Class .....	226
14.1	PBDOM exceptions .....	226
14.1.1	PBDOM exception descriptions .....	226
14.1.1.1	EXCEPTION_USE_OF_UNNAMED_PBDOM_OBJECT .....	226
14.1.1.2	EXCEPTION_WRONG_DOCUMENT_ERROR .....	227
14.1.1.3	EXCEPTION_MULTIPLE_ROOT_ELEMENT .....	227
14.1.1.4	EXCEPTION_INAPPROPRIATE_USE_OF_PBDOM_OBJECT .....	227
14.1.1.5	EXCEPTION_PBDOM_OBJECT_INVALID_FOR_USE .....	227
14.1.1.6	EXCEPTION_PBDOM_OBJECT_ALREADY_HAS_PARENT .....	227
14.1.1.7	EXCEPTION_MULTIPLE_DOCTYPE .....	228
14.1.1.8	EXCEPTION_ILLEGAL_PBOBJECT .....	228
14.1.1.9	EXCEPTION_WRONG_PARENT_ERROR .....	228
14.1.1.10	EXCEPTION_INVALID_ARGUMENT .....	228
14.1.1.11	EXCEPTION_INVALID_NAME .....	228
14.1.1.12	EXCEPTION_DATA_CONVERSION .....	229
14.1.1.13	EXCEPTION_MEMORY_ALLOCATION_FAILURE .....	229
14.1.1.14	EXCEPTION_INTERNAL_XML_ENGINE_ERROR .....	229
14.1.1.15	EXCEPTION_MULTIPLE_XMLDECL .....	229
14.1.1.16	EXCEPTION_INVALID_STRING .....	229
14.1.1.17	EXCEPTION_INVALID_OPERATION .....	229
14.1.1.18	EXCEPTION_HIERARCHY_ERROR .....	230
14.1.1.19	EXCEPTION_PBDOM_OBJECT_ALREADY_HAS_OWNER .....	230
14.1.1.20	EXCEPTION_PBDOM_NOT_INITIALIZED .....	230
14.2	PBDOM_EXCEPTION .....	230
14.2.1	GetExceptionCode .....	230
15	PBDOM_OBJECT Class .....	232
15.1	PBDOM_OBJECT .....	232
15.1.1	AddContent .....	232
15.1.2	Clone .....	233
15.1.3	Detach .....	234
15.1.4	Equals .....	235
15.1.5	GetContent .....	235
15.1.6	GetName .....	236
15.1.7	GetObjectClass .....	237
15.1.8	GetObjectClassString .....	238
15.1.9	GetOwnerDocumentObject .....	239
15.1.10	GetParentObject .....	240
15.1.11	GetText .....	241
15.1.12	GetTextNormalize .....	243
15.1.13	GetTextTrim .....	246
15.1.14	HasChildren .....	248
15.1.15	InsertContent .....	249

15.1.16	IsAncestorObjectOf .....	250
15.1.17	RemoveContent .....	252
15.1.18	SetContent .....	253
15.1.19	SetName .....	254
15.1.20	SetParentObject .....	254
16	PBDOM_PROCESSINGINSTRUCTION Class .....	257
16.1	PBDOM_PROCESSINGINSTRUCTION .....	257
16.1.1	Clone .....	258
16.1.2	Detach .....	259
16.1.3	Equals .....	259
16.1.4	GetData .....	260
16.1.5	GetName .....	260
16.1.6	GetNames .....	261
16.1.7	GetObjectClass .....	261
16.1.8	GetObjectClassString .....	262
16.1.9	GetOwnerDocumentObject .....	262
16.1.10	GetParentObject .....	262
16.1.11	GetTarget .....	263
16.1.12	GetText .....	263
16.1.13	GetTextNormalize .....	264
16.1.14	GetTextTrim .....	265
16.1.15	GetValue .....	265
16.1.16	RemoveValue .....	266
16.1.17	SetData .....	266
16.1.18	SetName .....	268
16.1.19	SetParentObject .....	269
16.1.20	SetValue .....	270
17	PBDOM_TEXT Class .....	272
17.1	PBDOM_TEXT .....	272
17.1.1	Append .....	273
17.1.1.1	Append Syntax 1 .....	273
17.1.1.2	Append Syntax 2 .....	273
17.1.2	Clone .....	274
17.1.3	Detach .....	276
17.1.4	Equals .....	276
17.1.5	GetObjectClass .....	277
17.1.6	GetObjectClassString .....	277
17.1.7	GetOwnerDocumentObject .....	278
17.1.8	GetParentObject .....	278
17.1.9	GetText .....	279
17.1.10	GetTextNormalize .....	279
17.1.11	GetTextTrim .....	280
17.1.12	SetParentObject .....	280
17.1.13	SetText .....	281
18	PBDOM Summary .....	283
18.1	Summary of PBDOM classes and methods .....	283
	Index .....	288



# 1 PowerBuilder Extensions

## About this chapter

This chapter provides a brief introduction to PowerBuilder extensions.

## 1.1 About PowerBuilder extensions

The PowerBuilder Native Interface (PBNI) is a standard programming interface that enables developers to extend the functionality of PowerBuilder. A PowerBuilder extension can be provided by Appeon, by you, or by a third party.

This book provides reference information for extensions provided by Appeon. In PowerBuilder, these extensions are for Enterprise JavaBeans clients, the PowerBuilder Document Object Model (PBDOM), SOAP clients for Web services (obsolete), and the UDDIProxy class. Embedding these features in separate extension files instead of adding them to the core PowerBuilder runtime files helps keep the footprint of deployed applications as small as possible.

For information about building your own extensions, see Part I, “PowerBuilder Native Interface Programmers Guide and Reference”.

To find out about extensions provided by other developers, check the PBNI section of the PowerBuilder Code Samples website at <https://www.appeon.com/developers/library/code-samples-for-pb>.

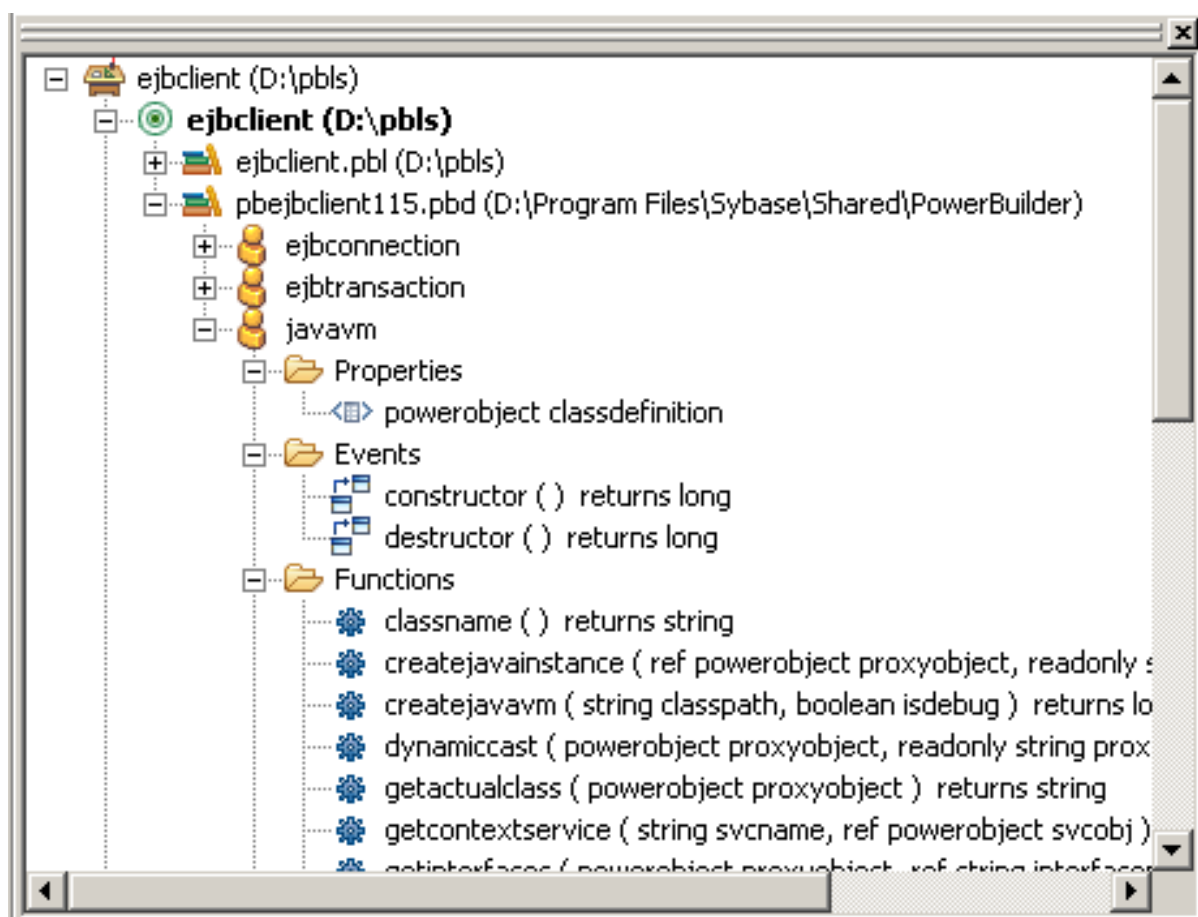
## 1.2 Using PowerBuilder extensions

Every PowerBuilder extension requires a compiled C++ shared library, usually with the extension .pbx (for PowerBuilder eXtension). The C++ shared library file contains classes and methods that you use in your PowerScript target in the same way that you use PowerBuilder system objects or user objects.

To use the shared library in PowerBuilder, you place it in PowerBuilder's search path. In the System Tree, right-click a library in your PowerScript target, select Import PB Extension from the pop-up menu, navigate to the shared library, and select Open. This imports the definitions in the PBX into the library in your target. You can alternatively add the associated PBD file to the target's library search path. The PBD acts as a wrapper for the C++ shared library, enabling PowerBuilder to display the objects and methods it contains.

When you deploy an application that uses an extension, the C++ shared library must be deployed in the application's search path with the other PowerBuilder runtime files.

When you import an extension into a PowerScript target, the classes it contains display in the System Tree as user objects. You can expand the objects to display properties, events, and functions. You can inherit from extension objects and use drag-and-drop programming from the inherited objects in the System Tree as you do for other user objects.



### Using nonvisual classes

In PowerScript, use the classes in a nonvisual extension just as you would a custom class user object: declare an instance of the object, use the CREATE statement to create the instance, invoke the object's functions, and destroy the instance when you have finished with it. You can inherit from the native classes if you want to add functions or events to the class.

At runtime, instances of the native class are created as normal PowerBuilder objects.

### Using visual classes

You do not need to declare an instance of a visual class or use the CREATE statement to create an instance. To use a visual extension, select File>Inherit from the PowerBuilder menu, select the PBL or PBD that contains the visual class in the Libraries list in the Inherit from Object dialog box, select the visual class, and click OK.

In the User Object painter, size the visual object and make any other changes you need, then save the object. You can then drag the new user object from the System Tree directly onto a window or onto another visual control, such as a tab control, and use it like any other visual user object.

### PBXRuntimeError

PowerBuilder extensions can throw a special exception, PBXRuntimeError, that inherits from the PowerBuilder RuntimeError exception. If you use an extension in a PowerBuilder application, you should include try-catch blocks for this exception and report any occurrences to the provider of the extension. This exception is usually caused by programming errors within the extension.

## 1.3 Getting information about PowerBuilder extensions

### Online Help

The classes and methods in the extensions provided by Appeon are described in this book, which is available in the PowerBuilder Help. For PBDOM, each class is described in a separate chapter.

You can open the Help in several ways:

- Select PowerBuilder Extension Reference from the PowerBuilder Help Contents tab page.
- Type a method name in the Script view, then press Shift+F1 to open the PowerBuilder Help Index tab with the focus on the first index entry for that method name. The name of the extension class displays in parentheses after the method name on the Index tab page, and it displays above the name of the method when you open the Help for the method.

If a PowerScript function description displays

If there is a PowerScript function with the same name, the Help opens automatically to display the PowerScript function. Click the Help Topics button in the Help window to display the Index tab so that you can select the extension method.

### HTML books

For information about using the extensions provided by Appeon in your applications, see Part I, “Application Techniques”.

### Third-party extensions

The PowerBuilder Help and documentation do not provide any specific information for extensions developed by third parties. To find out how to use a third-party extension, see the documentation provided with the extension.

## 2 EJB Client (obsolete)

Enterprise JavaBeans components are **obsolete** technology, although the components operate as usual in this release. An obsolete feature is no longer eligible for technical support and will no longer be enhanced, although it is still available.

For users who still want to use the following PowerBuilder extension classes (obsolete) to connect to an application server and employ Enterprise JavaBeans (EJB) components, refer to [this section](#).

- EJBConnection
- EJBTransaction
- JavaVM

## 3 Web Services Client (Obsolete)

Creating Web service proxy for connecting to SOAP server is no longer eligible for technical support. Developers who build Web service client that connects to SOAP server can choose to either continue using the feature without support, or use Section 2.41, “HTTPClient object” to call SOAP Web service. For how to use HTTPClient to call SOAP Web service, refer to this article: [Call SOAP Web Service Using HTTPClient Object](#).

For users who still want to use the following PowerBuilder extension classes (obsolete) to connect to a SOAP server, refer to [this section](#).

- SoapConnection
- SoapException
- SoapPBCookie
- UDDIProxy

## 4 PowerBuilder Document Object Model

### About this chapter

This chapter presents an overview of the PowerBuilder Document Object Model (PBDOM). For more information about using PBDOM, see the chapter on using XML services in Part I, “Application Techniques”.

### 4.1 About PBDOM

PBDOM is the PowerBuilder implementation of the Document Object Model (DOM), a programming interface defining the means by which XML documents can be accessed and manipulated.

Although PBDOM is not an implementation of the World Wide Web Consortium (W3C) DOM API, it is very similar. The PBDOM PowerBuilder API can be used for reading, writing, and manipulating standard-format XML from within PowerScript code. PBDOM portrays an XML document as a collection of interconnected objects and provides intuitive methods indicating the use and functionality of each object.

PBDOM is also similar to JDOM, which is a Java-based document object model for XML files.

For more information about W3C DOM, go to the W3C Document Object Model website at <http://www.w3.org/DOM/>. For more information about JDOM, go to the JDOM website at <http://www.jdom.org>.

#### 4.1.1 Node trees

PBDOM interacts with XML documents according to a tree-view model consisting of parent and child nodes. A document element represents the top-level node of a standalone XML document. This element has one or many child nodes that represent the branches of the tree. You access nodes in the node tree through the appropriate class methods.

#### 4.1.2 XML parser

The PBDOM XML parser is used to load and parse an XML document, and also to generate XML based on user-specified DOM nodes.

PBDOM provides the methods you need to traverse the node tree, access the nodes and attribute values (if any), insert and delete nodes, and serialize the node tree back to XML.

#### 4.1.3 Objects and methods

The PBDOM object hierarchy is described in [PBDOM objects](#). The methods for each object are described in the following chapters. The chapters are arranged in alphabetical order for ease of reference.

[PBDOM Summary](#) provides quick reference tables showing the signatures of the methods defined in each PBDOM object. The tables are arranged in an order that reflects the object hierarchy shown in [Object hierarchy](#).

#### 4.1.4 PBDOM objects

PBDOM\_OBJECT, the base class for PBDOM objects that represent XML nodes, inherits from the PowerBuilder NonVisualObject class. PBDOM represents node types by the following classes:

- [PBDOM\\_ATTRIBUTE](#)
- [PBDOM\\_CDATA](#)
- [PBDOM\\_CHARACTERDATA](#)
- [PBDOM\\_COMMENT](#)
- [PBDOM\\_DOCTYPE](#)
- [PBDOM\\_DOCUMENT](#)
- [PBDOM\\_ELEMENT](#)
- [PBDOM\\_ENTITYREFERENCE](#)
- [PBDOM\\_PROCESSINGINSTRUCTION](#)
- [PBDOM\\_TEXT](#)

You use methods from these classes to access objects in a PBDOM node tree.

The PBDOM\_BUILDER class does not represent DOM nodes but can be used to build a PBDOM object tree from XML. It inherits from the PowerBuilder NonVisualObject class.

The PBDOM\_EXCEPTION class inherits from the PowerBuilder Exception class and provides a method that obtains error codes.

Each of these classes and their methods are described in the chapters that follow.

#### Comparing PBDOM objects with W3C DOM and JDOM objects

The following table shows the W3C DOM and JDOM objects that correspond to each PBDOM object that represents a node in the DOM tree. Note that although these W3C DOM and JDOM objects correspond to PBDOM objects, they are not equivalent to the PBDOM objects.

**Table 4.1: W3C DOM and JDOM objects that correspond to PBDOM objects**

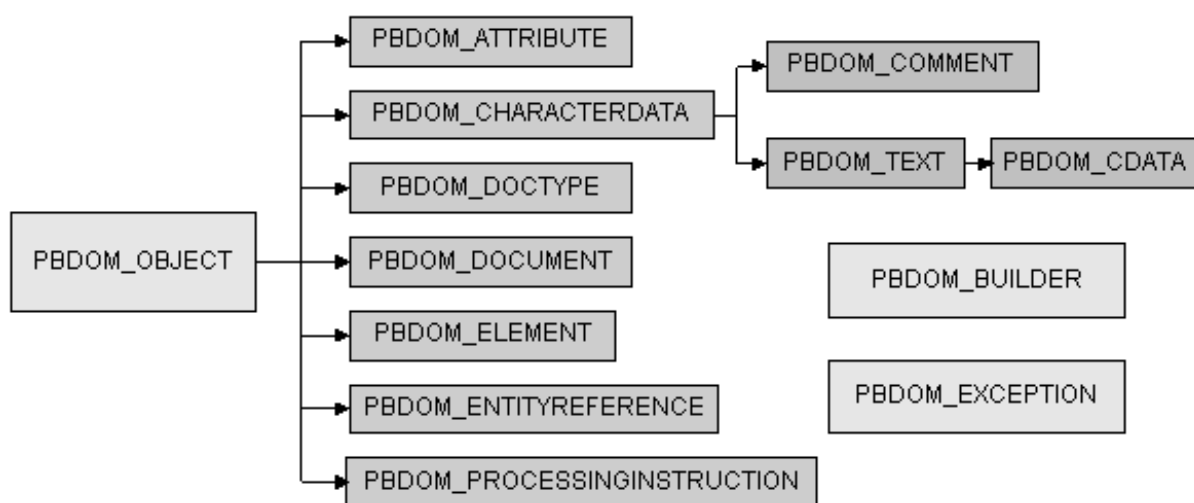
PBDOM	W3C DOM	JDOM
PBDOM_ATTRIBUTE	ATTRIBUTE_NODE	Attribute
PBDOM_BUILDER	None	DOMBuilder
PBDOM_CDATA	CDATA_SECTION_NODE	CDATA
PBDOM_CHARACTERDATA	CHARACTER_DATA_NODE	None
PBDOM_COMMENT	COMMENT_NODE	Comment
PBDOM_DOCUMENT	DOCUMENT_NODE	Document
PBDOM_DOCTYPE	DOCUMENT_TYPE_NODE	DocType

PBDOM	W3C DOM	JDOM
PBDOM_ELEMENT	ELEMENT_NODE	Element
PBDOM_ENTITYREFERENCE	ENTITY_REFERENCE_NODE	EntityRef
PBDOM_OBJECT	NODE	None
PBDOM_PROCESSINGINSTRUCTION	PROCESSING_INSTRUCTION_NODE	ProcessingInstruction
PBDOM_TEXT	TEXT_NODE	Text

### Object hierarchy

The W3C DOM and JDOM object hierarchies also differ from the PBDOM object hierarchy, which is shown in the following illustration.

Figure: The PBDOM object hierarchy



For more information about working with PBDOM, see the chapter on PowerBuilder XML services in Part I, “Application Techniques”.



## 5 PBDOM\_ATTRIBUTE Class

### About this chapter

This chapter describes the PBDOM\_ATTRIBUTE class.

### 5.1 PBDOM\_ATTRIBUTE

#### Description

The PBDOM\_ATTRIBUTE class defines the behavior for an XML attribute, modeled in PowerScript. Its methods allow you to obtain the value of the attribute as well as namespace information.

A PBDOM\_ATTRIBUTE contains a subtree of child PBDOM\_OBJECTS. These children can be a combination of PBDOM\_TEXT and PBDOM\_ENTITYREFERENCE objects.

---

#### **PBDOM\_ATTRIBUTE has no parent.**

A PBDOM\_ATTRIBUTE does not have a parent. However, it does have an owner PBDOM\_ELEMENT. Use the GetOwnerElementObject and SetOwnerElementObject to get and set the owner.

---

For more information about the PBDOM\_ATTRIBUTE object, including its default PBDOM\_TEXT object and its behavior with respect to XML namespaces, see Section 4.3, “Using PowerBuilder XML Services” in *Application Techniques*.

#### Methods

Some of the inherited methods from PBDOM\_OBJECT serve no meaningful objective and only default or trivial functionalities result. These are described in the following table:

**Table 5.1:**

Method	Always returns
GetParentObject	null
SetParentObject	The current PBDOM_ATTRIBUTE, returned unmodified as a PBDOM_OBJECT

PBDOM\_ATTRIBUTE has the following methods:

**Table 5.2:**

<a href="#">AddContent</a>	<a href="#">GetUIntValue</a>
<a href="#">Clone</a>	<a href="#">GetTimeValue</a>
<a href="#">Detach</a>	<a href="#">GetUlongValue</a>
<a href="#">Equals</a>	<a href="#">HasChildren</a>
<a href="#">GetBooleanValue</a>	<a href="#">InsertContent</a>
<a href="#">GetContent</a>	<a href="#">IsAncestorObjectOf</a>
<a href="#">GetDateValue</a>	<a href="#">RemoveContent</a>
<a href="#">GetDateTimeValue</a>	<a href="#">SetBooleanValue</a>

<a href="#">GetDoubleValue</a>	<a href="#">SetContent</a>
<a href="#">GetIntValue</a>	<a href="#">SetDateValue</a>
<a href="#">GetLongValue</a>	<a href="#">SetDateTimeValue</a>
<a href="#">GetName</a>	<a href="#">SetDoubleValue</a>
<a href="#">GetNamespacePrefix</a>	<a href="#">SetIntValue</a>
<a href="#">GetNamespaceUri</a>	<a href="#">SetLongValue</a>
<a href="#">GetObjectClass</a>	<a href="#">SetName</a>
<a href="#">GetObjectClassString</a>	<a href="#">SetNamespace</a>
<a href="#">GetOwnerDocumentObject</a>	<a href="#">SetOwnerElementObject</a>
<a href="#">GetOwnerElementObject</a>	<a href="#">SetRealValue</a>
<a href="#">GetQualifiedName</a>	<a href="#">SetText</a>
<a href="#">GetRealValue</a>	<a href="#">SetTimeValue</a>
<a href="#">GetText</a>	<a href="#">SetUIntValue</a>
<a href="#">GetTextNormalize</a>	<a href="#">SetUlongValue</a>
<a href="#">GetTextTrim</a>	

### 5.1.1 AddContent

#### Description

Adds the input PBDOM\_OBJECT as a child of the PBDOM\_ATTRIBUTE.

#### Syntax

```
pbdom_attribute_name.AddContent( pbdom_object pbdom_object_ref)
```

**Table 5.3:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
pbdom_object_ref	The PBDOM_OBJECT to add

#### Return value

PBDOM\_OBJECT. The PBDOM\_ATTRIBUTE modified.

#### Throws

EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT -- If the input PBDOM\_OBJECT is not a PBDOM\_TEXT or PBDOM\_ENTITYREFERENCE object.

EXCEPTION\_USE\_OF\_UNNAMED\_OBJECT -- If the input PBDOM\_OBJECT has not been given a user-defined name.

#### Usage

pbdom\_object\_ref must be a reference to a PBDOM\_TEXT or PBDOM\_ENTITYREFERENCE object.

**See also**[GetContent](#)[InsertContent](#)[RemoveContent](#)[SetContent](#)**5.1.2 Clone****Description**

Creates a clone of the PBDOM\_ATTRIBUTE object.

**Syntax**

```
pbdom_attribute_name.Clone(boolean bDeep)
```

**Table 5.4:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE.
bDeep	A boolean specifying whether a deep or shallow clone is returned. Values are true for a deep clone and false for a shallow clone.

**Return value**

PBDOM\_OBJECT. A clone of this PBDOM\_ATTRIBUTE returned as a PBDOM\_OBJECT.

**Throws**

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_ATTRIBUTE object's internal implementation is null. The occurrence of this exception is rare but can take place if severe memory corruption occurs.

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- If this PBDOM\_ATTRIBUTE does not have or has not been assigned a user-defined name.

**Examples**

This example creates a PBDOM\_DOCUMENT from the string <abc My\_Attr="An Attribute"/>, gets the attribute from the root element, and creates a shallow clone and a deep clone from it. For the shallow clone, an empty string is returned in the message box. For the deep clone, the string An Attribute is returned:

```
PBDOM_BUILDER      pbdom_buildr
PBDOM_DOCUMENT     pbdom_doc
PBDOM_ATTRIBUTE    pbdom_attr
PBDOM_ATTRIBUTE    pbdom_attr_clone_deep
PBDOM_ATTRIBUTE    pbdom_attr_clone_shallow
string strXML = "<abc My_Attr=~"An Attribute~/>"

TRY
    pbdom_buildr = Create PBDOM_BUILDER
    pbdom_doc = pbdom_buildr.BuildFromString(strXML)
    pbdom_attr = pbdom_doc.GetRootElement(). &
        GetAttribute("My_Attr")
    pbdom_attr_clone_shallow = pbdom_attr.Clone(false)
```

```

    MessageBox ("Shallow Attribute Clone Text", &
        pbdom_attr_clone_shallow.GetText())
    pbdom_attr_clone_deep = pbdom_attr.Clone(true)
    MessageBox ("Deep Attribute Clone Text", &
        pbdom_attr_clone_deep.GetText())

CATCH (PBDOM_EXCEPTION pbdom_except)
    MessageBox ("PBDOM_EXCEPTION", &
        pbdom_except.GetMessage())
END TRY

```

## Usage

The Clone method creates and returns a duplicate of the current PBDOM\_ATTRIBUTE.

If a shallow clone is requested, this method clones the original PBDOM\_ATTRIBUTE together with its namespace information values. The subtree of child PBDOM\_TEXT and/or PBDOM\_ENTITYREFERENCE objects is not cloned.

If a deep clone is requested, this method additionally recursively clones the subtree under the PBDOM\_ATTRIBUTE. This subtree consists of a combination of PBDOM\_TEXT and PBDOM\_ENTITYREFERENCE objects that are the legal children of a PBDOM\_ATTRIBUTE.

A PBDOM\_ATTRIBUTE clone has no parent. However, the clone resides in the same PBDOM\_DOCUMENT as its original, and if the original PBDOM\_ATTRIBUTE is standalone, the clone is standalone.

### 5.1.3 Detach

#### Description

Detaches a PBDOM\_ATTRIBUTE from its owner PBDOM\_OBJECT, a PBDOM\_ELEMENT.

#### Syntax

```
pbdom_attribute_name.Detach()
```

**Table 5.5:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

#### Return value

PBDOM\_OBJECT. The PBDOM\_ATTRIBUTE object detached from its owner object.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_ATTRIBUTE object's internal implementation is null. The occurrence of this exception is rare but can take place if severe memory corruption occurs.

#### Examples

The Detach method can be used to manipulate an XML document as follows:

```

PBDOM_BUILDER          pbdombuilder_new
PBDOM_DOCUMENT         pbdom_doc
PBDOM_ATTRIBUTE       pbdom_attr

```

```

PBDOM_ELEMENT      pbdom_elem
string strXML = "<abc My_Attr=~\"My Attribute Value~\"><data>Data</data></abc>"

TRY
  pbdbuilder_new = Create PBDOM_Builder
  pbdom_doc = pbdbuilder_new.BuildFromString (strXML)

  pbdom_attr = pbdom_doc.GetRootElement(). &
    GetAttribute("My_Attr")
  pbdom_attr.Detach()

  pbdom_elem = pbdom_doc.GetRootElement(). &
    GetChildElement("data")
  pbdom_elem.SetAttribute (pbdom_attr)

  Destroy pbdbuilder_new
  Destroy pbdom_doc

CATCH (PBDOM_Exception except)
  MessageBox ("Exception Occurred", except.Text)
END TRY

```

Here, the PBDOM\_Builder BuildFromString method is used to create the following PBDOM\_DOCUMENT object, pbdom\_doc, using an XML string:

```

<abc My_Attr="My Attribute Value">
  <data>Data </data>
</abc>

```

The GetAttribute method is used to obtain the attribute from the root element of pbdom\_doc. This value is assigned to the PBDOM\_ATTRIBUTE object pbdom\_attr. The pbdom\_attr object is detached from its parent element, and the data element is obtained from pbdom\_doc using the GetChildElement method. The data element is then assigned to the PBDOM\_ELEMENT object pbdom\_elem. The attribute assigned to pbdom\_attr is assigned to pbdom\_elem, yielding the following modified pbdom\_doc:

```

<abc>
  <data My_Attr="My Attribute Value">Data</data>
</abc>

```

## Usage

If the PBDOM\_ATTRIBUTE object has no owner PBDOM\_ELEMENT, the Detach method does nothing.

### 5.1.4 Equals

#### Description

Tests for equality between the supplied PBDOM\_OBJECT and the PBDOM\_ATTRIBUTE from which the method is invoked.

#### Syntax

```
pbdom_attribute_name.Equals(pbdom_object pbdom_object_ref)
```

**Table 5.6:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

Argument	Description
pbdom_object_ref	A PBDOM_OBJECT to be compared

### Return value

Boolean.

Returns true if the current PBDOM\_ATTRIBUTE is equivalent to the input PBDOM\_OBJECT and false otherwise.

### Throws

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- If this PBDOM\_ATTRIBUTE does not have or has not been assigned a user-defined name.

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- if the input PBDOM\_OBJECT is not a reference to an object derived from PBDOM\_OBJECT.

### Examples

1. The following code uses the Equals method to test for equivalence between a referenced PBDOM\_OBJECT and a cloned object.

```
pbdom_attr = Create PBDOM_Attribute
pbdom_attr.SetName("My_Attr")
pbdom_attr_clone = pbdom_attr.Clone(true)

if (pbdom_attr_clone.Equals(pbdom_attr)) then
    MessageBox ("Equals", "Yes")
else
    MessageBox ("Equals", "No")
end if
```

The SetName method names the newly created PBDOM\_ATTRIBUTE, which is subsequently cloned with the Clone method. The Equals method tests for equality between the cloned PBDOM\_ATTRIBUTE pbdom\_attr\_clone and the referenced PBDOM\_OBJECT pbdom\_attr. A message box displays the result returned from the Equals method.

Note here that because a cloned object is never equivalent to the object from which it is cloned, the Equals method returns false.

2. The following code uses the Equals method to test for equivalence between two cloned objects.

```
pbdom_attr = Create PBDOM_Attribute
pbdom_attr.SetName("My_Attr")
pbdom_attr_clone = pbdom_attr.Clone(true)
pbdom_attr_2 = pbdom_attr_clone

if (pbdom_attr_clone.Equals(pbdom_attr_2)) then
    MessageBox ("Equals", "Yes")
else
    MessageBox ("Equals", "No")
end if
```

A newly created PBDOM\_ATTRIBUTE is cloned, and a reference to this clone is assigned to pbdom\_attr\_2. The Equals method tests for equality between the cloned

PBDOM\_ATTRIBUTE pbdom\_attr\_clone and the reference to it, pbdom\_attr\_2. A message box displays the result returned from the Equals method.

Here the Equals method returns true.

### Usage

Note that the clone of a PBDOM\_ATTRIBUTE is not considered equal to itself.

## 5.1.5 GetBooleanValue

### Description

Obtains the value of a PBDOM\_ATTRIBUTE object in boolean form.

### Syntax

```
pbdom_attribute_name.GetBooleanValue()
```

**Table 5.7:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

### Return value

Boolean.

The following table lists the PBDOM\_ATTRIBUTE string values that are accepted as boolean and the corresponding return values from the GetBooleanValue method.

**Table 5.8:**

PBDOM_ATTRIBUTE string value	GetBooleanValue
1	true
0	false
TRUE	true
FALSE	false
ON	true
OFF	false
YES	true
NO	false

Strings are treated without case sensitivity. If no conversion can occur, the GetBooleanValue method throws an exception.

### Throws

EXCEPTION\_DATA\_CONVERSION -- If data conversion fails.

### Examples

The GetBooleanValue can be used to evaluate a PBDOM\_ATTRIBUTE object as follows:

```
PBDOM_BUILDER            pbombuilder_new
```

```

PBDOM_DOCUMENT      pbdom_doc
PBDOM_ATTRIBUTE     pbdom_attr
string strXML = "<abc My_Boolean_Attribute =~"on~"><data An_Attribute=~"Some
Text~">Data</data></abc>"

TRY
  pbdombuilder_new = Create PBDOM_Builder
  pbdom_doc = pbdombuilder_new.BuildFromString (strXML)

  pbdom_attr = pbdom_doc.GetRootElement(). &
    GetAttribute("My_Boolean_Attribute")

  MessageBox ("Boolean Value", &
    string(pbdom_attr.GetBooleanValue()))

  Destroy pbdombuilder_new
  Destroy pbdom_doc
CATCH (PBDOM_Exception except)
  MessageBox ("Exception Occurred", except.Text)
END TRY

```

The BuildFromString method is used to create a PBDOM\_DOCUMENT object, pbdom\_doc, using an XML string. The attribute value of the root element of pbdom\_doc is assigned to the PBDOM\_ATTRIBUTE object pbdom\_attr. The attribute value, on, is evaluated with the GetBooleanValue method. A message box reports the return value of the GetBooleanValue method.

### See also

[SetBooleanValue](#)

## 5.1.6 GetContent

### Description

Returns an array of PBDOM\_OBJECT objects that are the children of the PBDOM\_ATTRIBUTE. The children of a PBDOM\_ATTRIBUTE can be only PBDOM\_TEXT or PBDOM\_ENTITYREFERENCE objects.

### Syntax

```
pbdom_attribute_name.GetContent(ref pbdom_object pbdom_object_array[ ])
```

**Table 5.9:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
pbdom_object_array	The referenced name of an array of PBDOM_OBJECTs that receives PBDOM_OBJECTs

### Return value

Boolean.

This method always returns true.

### See also



[AddContent](#)[InsertContent](#)[RemoveContent](#)[SetContent](#)

### 5.1.7 GetDateValue

#### Description

Returns the value of a PBDOM\_ATTRIBUTE object as type Date.

#### Syntax

```
pbdom_attribute_name.GetDateValue(string strDateFormat)
```

**Table 5.10:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
strDateFormat	The date format for the return value, for example, MM:DD:YYYY

The value of the strDateFormat parameter can use slashes or colons as delimiters. The following table illustrates characters with special meaning in strDateFormat.

**Table 5.11:**

Character	Meaning	Example
D	Day number with no leading zero	5
DD	Day number with leading zero, if applicable	05
M	Month number with no leading zero	5
MM	Month number with leading zero, if applicable	05
YY	Two-digit year number	05
YYYY	Four-digit year number	2005

#### Return value

Date.

#### Throws

EXCEPTION\_DATA\_CONVERSION -- If data conversion fails.

#### See also

[SetDateValue](#)

### 5.1.8 GetDateTimeValue

#### Description

Returns the value of a PBDOM\_ATTRIBUTE object as type DateTime.

#### Syntax

```
pbdom_attribute_name.GetDateTimeValue(string strDateFormat, string strTimeFormat)
```

**Table 5.12:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
strDateFormat	The date format for the return value, for example, MM:DD:YYYY
strTimeFormat	The time format for the return value, for example, HH:MM:SS

The value of the strDateFormat parameter can use slashes or colons as delimiters. The following table illustrates characters that have special meaning in strDateFormat.

**Table 5.13:**

Character	Meaning	Example
D	Day number with no leading zero	5
DD	Day number with leading zero, if applicable	05
M	Month number with no leading zero	5
MM	Month number with leading zero, if applicable	05
YY	Two-digit year number	05
YYYY	Four-digit year number	2005

The value of the strTimeFormat parameter can use slashes or colons as delimiters. The following table illustrates characters that have special meaning in strTimeFormat.

**Table 5.14:**

Character	Meaning	Example
H	Hour number with no leading zero	5
HH	Hour number with leading zero, if applicable	05
M	Minutes number with no leading zero	5

Character	Meaning	Example
MM	Minutes number with leading zero, if applicable	05
S	Seconds number with no leading zero	5
SS	Seconds number with leading zero, if applicable	55

**Return value**

DateTime.

**Throws**

EXCEPTION\_DATA\_CONVERSION -- If data conversion fails.

**See also**

[SetDateTimeValue](#)

**5.1.9 GetDoubleValue****Description**

Returns the value of a PBDOM\_ATTRIBUTE object in double form.

**Syntax**

```
pbdom_attribute_name.GetDoubleValue()
```

**Table 5.15:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

**Return value**

Double.

**Throws**

EXCEPTION\_DATA\_CONVERSION -- If data conversion fails.

**Usage**

Throws exception\_data\_conversion if the method fails to convert data.

**See also**

[SetDoubleValue](#)

**5.1.10 GetIntValue****Description**

Returns the value of a PBDOM\_ATTRIBUTE object as type int.

**Syntax**

```
pbdom_attribute_name.GetIntValue()
```

**Table 5.16:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

**Return value**

Int.

**Throws**

EXCEPTION\_DATA\_CONVERSION -- If data conversion fails.

**See also**[SetIntValue](#)**5.1.11 GetLongValue****Description**

Returns the value of a PBDOM\_ATTRIBUTE object as type long.

**Syntax**

```
pbdom_attribute_name.GetLongValue()
```

**Table 5.17:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

**Return value**

Long.

**Throws**

EXCEPTION\_DATA\_CONVERSION -- If data conversion fails.

**See also**[SetLongValue](#)**5.1.12 GetName****Description**

Retrieves the local name of the PBDOM\_ATTRIBUTE object.

**Syntax**

```
pbdom_attribute_name.GetName()
```

**Table 5.18:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

## Return value

String.

## Throws

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- If this PBDOM\_ATTRIBUTE does not have or has not been assigned a user-defined name.

## Examples

1. When the GetName method is invoked for the attribute name in the following element, it returns the string ATTRIBUTE\_1:

```
<abc ATTRIBUTE_1="My Attribute">
```

2. When the GetName method is invoked for the name of the eMusic:Type attribute in the following element, it returns the string Type:

```
<eMusic:CD xmlns:eMusic="http://www.eMusic_Records.com" eMusic:Type="Jazz" />
```

The namespace prefix is not part of the return string.

## Usage

For an XML attribute that appears in the form [namespace\_prefix]:[attribute\_name], the local attribute name is attribute\_name. Where the XML attribute has no namespace prefix, the local name is simply the attribute name.

Use the GetNamespacePrefix method to obtain the namespace prefix for a PBDOM\_ATTRIBUTE object. Use the GetQualifiedName method to obtain the fully qualified name for a PBDOM\_ATTRIBUTE object.

## See also

[GetNamespacePrefix](#)

[GetNamespaceUri](#)

[GetQualifiedName](#)

[SetName](#)

[SetNamespace](#)

## 5.1.13 GetNamespacePrefix

### Description

Obtains the namespace prefix of a PBDOM\_ATTRIBUTE object. The GetNamespacePrefix method returns an empty string if the PBDOM\_ATTRIBUTE has no namespace.

### Syntax

```
pbdom_attribute_name.GetNamespacePrefix()
```

**Table 5.19:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

**Return value**

String

For a PBDOM\_ATTRIBUTE object that has the form [namespacePrefix]:[attributeName], the namespace prefix is [namespacePrefix].

**See also**[GetNamespaceUri](#)[GetQualifiedName](#)[SetName](#)[SetNamespace](#)**5.1.14 GetNamespaceUri****Description**

Obtains the namespace URI of a PBDOM\_ATTRIBUTE object. The GetNamespaceUri method returns an empty string if the PBDOM\_ATTRIBUTE has no namespace.

**Syntax**

```
pbdom_attribute_name.GetNamespaceUri()
```

**Table 5.20:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

**Return value**

String.

**See also**[GetNamespacePrefix](#)[GetQualifiedName](#)[SetName](#)[SetNamespace](#)**5.1.15 GetObjectClass****Description**

Returns a long integer code that indicates the class of the current PBDOM\_OBJECT.

**Syntax**

```
pbdom_object_name.GetObjectClass()
```

**Table 5.21:**

Argument	Description
pbdom_object_name	The name of a PBDOM_OBJECT

**Return value**

Long.

GetObjectClass returns a long integer code that indicates the class of the current PBDOM\_OBJECT. If pbdom\_object\_name is a PBDOM\_ATTRIBUTE, the returned value is 5.

**Examples**

This example illustrates polymorphism: pbdom\_obj is declared as PBDOM\_OBJECT but instantiated as PBDOM\_ATTRIBUTE. A message box returns the result of the GetObjectClass method invoked for PBDOM\_ATTRIBUTE. Here the result is 5, indicating that pbdom\_obj is a PBDOM\_ATTRIBUTE object.

```
PBDOM_OBJECT pbdom_obj

pbdom_obj = Create PBDOM_ATTRIBUTE
MessageBox ("Class", &
    string(pbdom_obj.GetObjectClass()))
```

**Usage**

This method can be used for diagnostic purposes to dynamically determine the type of a PBDOM\_OBJECT at runtime.

**See also**

[GetObjectClassString](#)

**5.1.16 GetObjectClassString****Description**

Returns a string form of the class of the PBDOM\_OBJECT.

**Syntax**

```
pbdom_object_name.GetObjectClassString()
```

**Table 5.22:**

Argument	Description
pbdom_object_name	The name of a PBDOM_OBJECT

**Return value**

String.

GetObjectClassString returns a string that indicates the class of the current PBDOM\_OBJECT. If pbdom\_object\_name is a PBDOM\_ATTRIBUTE, the returned string is "pbdom\_attribute".

**Examples**

The GetObjectClass method returns a string specific to the class of the object from which the method is invoked.

This example illustrates polymorphism: pbdom\_obj is declared as PBDOM\_OBJECT but instantiated as PBDOM\_ATTRIBUTE. A message box returns the result of the

GetObjectClassString method invoked for PBDOM\_ATTRIBUTE. Here the result is pbdom\_attribute, indicating that pbdom\_obj is a PBDOM\_ATTRIBUTE object.

```
PBDOM_OBJECT pbdom_obj
pbdom_obj = Create PBDOM_ATTRIBUTE
MessageBox ("Class", pbdom_obj.GetObjectClassString())
```

## Usage

This method can be used for diagnostic purposes to dynamically determine the actual type of a PBDOM\_OBJECT at runtime.

## See also

[GetObjectClass](#)

### 5.1.17 GetOwnerDocumentObject

#### Description

Returns the PBDOM\_DOCUMENT object that owns the PBDOM\_ATTRIBUTE.

#### Syntax

```
pbdom_attribute_name.GetOwnerDocumentObject()
```

**Table 5.23:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

#### Return value

PBDOM\_DOCUMENT. The PBDOM\_DOCUMENT that owns the PBDOM\_ATTRIBUTE object from which the GetOwnerDocumentObject method is invoked.

A return value of null indicates the PBDOM\_ATTRIBUTE object is not owned by any PBDOM\_DOCUMENT.

#### Examples

The GetOwnerDocumentObject method can be used to identify the PBDOM\_DOCUMENT object that owns a PBDOM\_ATTRIBUTE object.

Here, the BuildFromString method is used to create the following PBDOM\_DOCUMENT object, pbdom\_doc, using an XML string:

```
<abc My_Attr="My Attribute Value">
  <data>Data </data>
</abc>
```

The GetAttribute method is used to obtain the attribute from the root element of pbdom\_doc. This value is assigned to the PBDOM\_ATTRIBUTE object pbdom\_attr. The GetOwnerDocumentObject method is used to obtain the pbdom\_doc that owns pbdom\_attr. The result of the GetOwnerDocumentObject method is assigned to the PBDOM\_DOCUMENT object pbdom\_doc\_2. Then pbdom\_doc\_2 is compared to pbdom\_doc using the Equals method, and the result is displayed in a message box.



```

PBDOM_Builder pbdombuilder_new
pbdom_document pbdom_doc
pbdom_document pbdom_doc_2
PBDOM_ATTRIBUTE pbdom_attr
string strXML = "<abc My_Attr=~\"My Attribute Value~\"><data>Data </data></abc>"

TRY
    pbdombuilder_new = Create PBDOM_Builder
    pbdom_doc = pbdombuilder_new.BuildFromString (strXML)

    pbdom_attr = pbdom_doc.GetRootElement(). &
        GetAttribute("My_Attr")
    pbdom_doc_2 = pbdom_attr.GetOwnerDocumentObject()

    if (pbdom_doc.Equals(pbdom_doc_2)) then
        MessageBox ("Equals", "pbdom_doc equals " &
            + "pbdom_attr.GetOwnerDocumentObject()")
    end if

    Destroy pbdombuilder_new

CATCH (PBDOM_Exception except)
    MessageBox ("Exception Occurred", except.Text)
END TRY

```

**See also**[GetOwnerElementObject](#)[SetOwnerElementObject](#)**5.1.18 GetOwnerElementObject****Description**

Returns the owner PBDOM\_ELEMENT of this PBDOM\_ATTRIBUTE. If there is no owner element, null is returned.

**Syntax**

```
pbdom_attribute_name.GetOwnerElementObject( )
```

**Table 5.24:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

**Return value**

PBDOM\_ELEMENT. The owner PBDOM\_ELEMENT of this PBDOM\_ATTRIBUTE or null if this PBDOM\_ATTRIBUTE has no owner element.

**Throws**

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_ATTRIBUTE object's internal implementation is null. The occurrence of this exception is rare but can take place if severe memory corruption occurs.

**Examples**

This example creates a PBDOM\_DOCUMENT from a string strXML in which the abc root element contains one attribute, My\_Attr. The code gets this attribute, calls GetOwnerElementObject on it to obtain the owner element, then calls GetName to return the string abc. Finally, it sets My\_Attr as an attribute of the child element Data:

```
PBDOM_BUILDER      pbdombuilder_new
PBDOM_DOCUMENT    pbdom_doc
PBDOM_ATTRIBUTE   pbdom_attr
PBDOM_ELEMENT     pbdom_elem
string strXML = "<abc My_Attr=~\"My Attribute Value~\"><data>Data</data></abc>"

TRY
  pbdombuilder_new = Create PBDOM_Builder
  pbdom_doc = pbdombuilder_new.BuildFromString (strXML)

  // Get the attribute
  pbdom_attr = pbdom_doc.GetRootElement(). &
    GetAttribute("My_Attr")

  MessageBox ("pbdom_attr Owner Element Name", &
    pbdom_attr.GetOwnerElementObject().GetName())

  pbdom_attr.Detach()

  pbdom_elem = pbdom_doc.GetRootElement(). &
    GetChildElement("data")
  pbdom_elem.SetAttribute (pbdom_attr)

  MessageBox ("pbdom_attr Owner Element Name", &
    pbdom_attr.GetOwnerElementObject().GetName())

  Destroy pbdombuilder_new
  Destroy pbdom_doc

CATCH (PBDOM_Exception except)
  MessageBox ("Exception Occurred", except.Text)
END TRY
```

## See also

[SetOwnerElementObject](#)

### 5.1.19 GetQualifiedName

#### Description

Obtains the qualified name of a PBDOM\_ATTRIBUTE. The GetQualifiedName method returns the local name for a PBDOM\_ATTRIBUTE that has no namespace.

#### Syntax

```
pbdom_attribute_name.GetQualifiedName()
```

**Table 5.25:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

#### Return value

String.

## Usage

For a PBDOM\_ATTRIBUTE object that has the form [namespacePrefix]:[attributeName], the qualified name for the PBDOM\_ATTRIBUTE consists of the entire name, [namespacePrefix], and [attributeName].

To obtain the local name of the PBDOM\_ATTRIBUTE, use the GetName method.

To obtain the namespace prefix for the PBDOM\_ATTRIBUTE, use the GetNamespacePrefix method.

## See also

[GetName](#)

[GetNamespacePrefix](#)

[GetNamespaceUri](#)

[SetName](#)

[SetNamespace](#)

## 5.1.20 GetRealValue

### Description

Returns the value of a PBDOM\_ATTRIBUTE object as type real.

### Syntax

```
pbdom_attribute_name.GetRealValue()
```

**Table 5.26:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

### Return value

Real.

### Throws

EXCEPTION\_DATA\_CONVERSION -- If data conversion fails.

### Usage

GetRealValue is the exact counterpart of the JDOM getFloatValue method.

## See also

[SetRealValue](#)

## 5.1.21 GetText

### Description

Returns the text value of the PBDOM\_ATTRIBUTE object.

### Syntax

```
pbdom_attribute_name.GetText()
```

**Table 5.27:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

**Return value**

String.

**Throws**

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- If this PBDOM\_ATTRIBUTE does not have or has not been assigned a user-defined name.

**Examples**

1. The GetText method is invoked for the attribute in the following element:

```
<abc ATTRIBUTE_1="My Attribute">
```

The GetText method returns the following string:

```
My Attribute
```

2. This example sets an attribute called my\_attr for the root element with text value text part. A PBDOM\_ENTITYREFERENCE with the name ent\_ref and a PBDOM\_TEXT with the text value text part again are then added as part of the contents of my\_attr. A call to GetText on my\_attr returns the following text:

```
"text part &ent_ref; text part again."
```

The entity reference &ent\_ref; is not expanded. If an entity reference is included in an input XML document that is parsed, then the entity reference is expanded before the XML document is transformed into a DOM tree in memory.

```
PBDOM_DOCUMENT      pbdom_doc
PBDOM_ATTRIBUTE     pbdom_attr
PBDOM_ENTITYREFERENCE pbdom_entref
PBDOM_TEXT          pbdom_txt

try
    pbdom_doc = Create PBDOM_DOCUMENT
    pbdom_entref = Create PBDOM_ENTITYREFERENCE
    pbdom_txt = Create PBDOM_TEXT

    // Create a new document object.
    pbdom_doc.NewDocument ("root")

    // Set the text of "pbdom_txt".
    pbdom_txt.SetText (" text part again.")

    // Add an attribute "my_attr" to the root element.
    pbdom_doc.GetRootElement().SetAttribute("my_attr", &
        "text part ")

    // Set the name of the PBDOM_ENTITYREFERENCE.
    pbdom_entref.SetName ("ent_ref")

    // Append the entity reference to the root
```

```

// element's "my_attr" attribute.
pbdom_doc.GetRootElement(). &
    GetAttribute("my_attr").AddContent(pbdom_entref)

// Append a new text node to the "my_attr" attribute.
pbdom_doc.GetRootElement(). &
    GetAttribute("my_attr").AddContent (pbdom_txt)

// Now test the text contents of "my_attr "
if pbdom_doc.GetRootElement(). &
    GetAttribute("my_attr").GetText() = &
    "text part &ent_ref; text part again." then
    MessageBox ("Pass", &
        "GetText() on my_attr is correct.")
else
    MessageBox ("Fail", &
        "GetText() on my_attr is incorrect.")
end if

catch (pbdom_exception pbdom_e)
    MessageBox ("PBDOM_EXCEPTION", pbdom_e.GetMessage())
end try

```

## Usage

This method returns the actual textual value of this PBDOM\_ATTRIBUTE, including all text within the quotation marks. If there are any PBDOM\_ENTITYREFERENCE objects included within the PBDOM\_ATTRIBUTE, the PBDOM\_ENTITYREFERENCE object's name is returned together with the leading ampersand ('&') character plus the terminating semicolon character(';').

## See also

[GetTextNormalize](#)

[GetTextTrim](#)

[SetText](#)

## 5.1.22 GetTextNormalize

### Description

Returns the text data contained within a PBDOM\_ATTRIBUTE object with surrounding whitespace characters removed and internal whitespace characters replaced by a single space.

### Syntax

```
pbdom_attribute_name.GetTextNormalize()
```

**Table 5.28:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

### Return value

String.

### Examples

1. The `GetTextNormalize` method is invoked for the `PBDOM_ATTRIBUTE` of the following element:

```
<abc ATTRIBUTE_1=" My Attribute ">
```

The `GetTextNormalize` method returns the following string:

```
My Attribute
```

2. This example creates a `PBDOM_DOCUMENT` based on the following DOM tree, which has a Tab character between the words "My" and "Attribute" in the `My_Attr` attribute, specified by the `&#9;` entity reference. There are also several space characters:

```
<abc My_Attr="My&#9;Attribute Value ">
  <data>Data</data>
</abc>
```

The call to `GetAttribute` stores `My_Attr` in `pbdom_attr`. Calling `GetText` on `pbdom_attr` returns the entire string content of `My_Attr`, including the beginning Tab character. Calling `GetTextNormalize` returns the string with all surrounding whitespace characters removed, and the whitespace characters between the words, including the Tab character, replaced by a single space.

```
PBDOM_BUILDER      pbdombuilder_new
PBDOM_DOCUMENT     pbdom_doc
PBDOM_ATTRIBUTE    pbdom_attr
string strXML = "<abc My_Attr=~"My&#9;Attribute Value ~"><data>Data</data></abc>"

TRY
  pbdombuilder_new = Create PBDOM_Builder
  pbdom_doc = pbdombuilder_new.BuildFromString (strXML)

  pbdom_attr = pbdom_doc.GetRootElement(). &
    GetAttribute("My_Attr")

  MessageBox ("pbdom_attr text", "[" &
    "+ pbdom_attr.GetText() + "]")
  MessageBox ("pbdom_attr text normalize", "[" &
    "+ pbdom_attr.GetTextNormalize() + "]")

  Destroy pbdombuilder_new
  Destroy pbdom_doc

  CATCH (PBDOM_Exception except)
    MessageBox ("Exception Occurred", except.Text)
END TRY
```

## Usage

Surrounding whitespace characters are removed from the returned text data, and internal whitespace characters are normalized to a single space. The `GetTextNormalize` method returns an empty string if no text value exists for the `PBDOM_ATTRIBUTE` or if the text value contains only whitespace characters.

If this `PBDOM_ATTRIBUTE` contains any `PBDOM_ENTITYREFERENCE` objects, the name of the `PBDOM_ENTITYREFERENCE` object is returned as part of the normalized string.

JDOM does not provide a `getTextNormalize` method for its `Attribute` class.

**See also**[GetText](#)[GetTextTrim](#)[SetText](#)**5.1.23 GetTextTrim****Description**

Returns the text data contained within a PBDOM\_ATTRIBUTE object with surrounding spaces removed.

**Syntax**

```
pbdom_attribute_name.GetTextTrim()
```

**Table 5.29:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

**Return value**

String.

**Examples**

1. The GetTextTrim method is invoked for the PBDOM\_ATTRIBUTE of the following element:

```
<abc ATTRIBUTE_1=" My Attribute ">
```

The GetTextNormalize method returns the following string:

```
My Attribute
```

Note that the whitespace characters surrounding the string are removed, but the whitespace characters within the string remain.

2. This example builds a PBDOM\_DOCUMENT based on the following XML tree:

```
<abc My_Attr="&#32;&#32;&#32;My&#9;Attribute Value&#32;&#32;&#32;">
  <data>Data</data>
</abc>
```

The My\_Attr attribute contains an entity reference for a Tab character (&#9;) and several entity references for the space character (&#32;). The message boxes in the following code show that GetText returns the complete text string of the attribute, whereas GetTextTrim returns the string with the surrounding whitespace characters removed. The Tab character between the words is not removed:

```
PBDOM_BUILDER      pbdombuilder_new
PBDOM_DOCUMENT     pbdom_doc
PBDOM_ATTRIBUTE    pbdom_attr
string             strXML
```

```

TRY
  strXML = "<abc  My_Attr=~" &#32;&#32;&#32;My&#9;Attribute
Value&#32;&#32;&#32;~"><data>Data</data></abc>"
  pbdombuilder_new = Create PBDOM_Builder
  pbdom_doc = pbdombuilder_new.BuildFromString (strXML)

  pbdom_attr = pbdom_doc.GetRootElement(). &
  GetAttribute("My_Attr")

  MessageBox ("pbdom_attr text", "[" &
  + "pbdom_attr.GetText() + "]")
  MessageBox ("pbdom_attr text normalize", &
  "[" + pbdom_attr.GetTextTrim() + "]")

Destroy pbdombuilder_new
Destroy pbdom_doc

CATCH (PBDOM_Exception except)
  MessageBox ("Exception Occurred", except.Text)
END TRY

```

## Usage

Surrounding whitespace characters are removed from the returned text data.

The `GetTextTrim` method returns an empty string if no text value exists for the `PBDOM_ATTRIBUTE` or if the text value contains only whitespace characters.

If this `PBDOM_ATTRIBUTE` contains any `PBDOM_ENTITYREFERENCE` objects, the name of the `PBDOM_ENTITYREFERENCE` object is returned as part of the trimmed string.

## See also

[GetText](#)

[GetTextNormalize](#)

[SetText](#)

### 5.1.24 GetTimeValue

#### Description

Returns the value of a `PBDOM_ATTRIBUTE` object as type `Time`.

#### Syntax

```
pbdom_attribute_name.GetTimeValue(string strTimeFormat)
```

**Table 5.30:**

Argument	Description
<code>pbdom_attribute_name</code>	The name of the <code>PBDOM_ATTRIBUTE</code>
<code>strTimeFormat</code>	The time format for the return value, for example, <code>HH:MM:SS</code>

The value of the `strTimeFormat` parameter can use slashes or colons as delimiters. The following table illustrates characters that have special meaning in `strTimeFormat`.



**Table 5.31:**

Character	Meaning	Example
H	Hour number with no leading zero	5
HH	Hour number with leading zero, if applicable	05
M	Minutes number with no leading zero	5
MM	Minutes number with leading zero, if applicable	05
S	Seconds number with no leading zero	5
SS	Seconds number with leading zero, if applicable	55

**Return value**

Time.

**Throws**

EXCEPTION\_DATA\_CONVERSION -- If data conversion fails.

**See also**

[SetTimeValue](#)

**5.1.25 GetUIntValue****Description**

Returns the value of a PBDOM\_ATTRIBUTE object as type UInt.

**Syntax**

```
pbdom_attribute_name.GetUIntValue()
```

**Table 5.32:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

**Return value**

UInt.

**Throws**

EXCEPTION\_DATA\_CONVERSION -- If data conversion fails.

**See also**

[SetUIntValue](#)

**5.1.26 GetUlongValue****Description**

Returns the value of a PBDOM\_ATTRIBUTE object as type Ulong.

### Syntax

```
pbdom_attribute_name.GetUlongValue()
```

**Table 5.33:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

### Return value

Ulong.

### Throws

EXCEPTION\_DATA\_CONVERSION -- If data conversion fails.

### See also

[SetUlongValue](#)

## 5.1.27 HasChildren

### Description

Determines whether this PBDOM\_ATTRIBUTE object contains any child PBDOM\_OBJECTs.

### Syntax

```
pbdom_attribute_name.HasChildren()
```

**Table 5.34:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE

### Return value

Boolean.

Returns true if this PBDOM\_ATTRIBUTE contains child objects and false otherwise.

### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object is not associated with a derived PBDOM\_OBJECT class object.

### Examples

This example creates a PBDOM\_DOCUMENT from a string. The XML document in the string already contains a root element named root that contains an attribute attr that contains an empty string. It then represents attr as a PBDOM\_ATTRIBUTE object and calls its HasChildren method, which returns true because a PBDOM\_ATTRIBUTE always contains at least one child object. After a call to GetContent, the message box shows that attr contains only one child, a PBDOM\_TEXT that represents the empty string:

```
PBDOM_BUILDER pbdom_buildr
PBDOM_DOCUMENT pbdom_doc
```

```

PBDOM_ATTRIBUTE pbdom_attr
string strXML = "<root attr=~\"~\"></root>"

try
  pbdom_buildr = Create PBDOM_BUILDER
  pbdom_doc = pbdom_buildr.BuildFromString(strXML)

  pbdom_attr = pbdom_doc.GetRootElement(). &
    GetAttribute("attr")

  if (pbdom_attr.HasChildren()) then
    PBDOM_OBJECT pbdom_obj_array[]
    long l = 0

    pbdom_attr.GetContent(pbdom_obj_array)

    for l = 1 to UpperBound (pbdom_obj_array)
      MessageBox ("Attr Child Object", &
        pbdom_obj_array[l].GetObjectClassString())
    next
  end if

catch (pbdom_exception pbdom_e)
  MessageBox ("PBDOM_EXCEPTION", pbdom_e.GetMessage())
end try

```

## Usage

This method checks to see if this PBDOM\_ATTRIBUTE object contains any child PBDOM\_OBJECTs and returns true if it does. Note that according to the W3C DOM specification, a DOM Attribute Node can contain only Text and Entity Reference Nodes, therefore a PBDOM\_ATTRIBUTE object can contain only PBDOM\_TEXT and PBDOM\_ENTITYREFERENCE objects.

Even if a PBDOM\_ATTRIBUTE object's text value is an empty string, it always contains at least one PBDOM\_TEXT object that represents the empty string.

### 5.1.28 InsertContent

#### Description

Inserts a PBDOM\_OBJECT as a child of the PBDOM\_ATTRIBUTE at a position specified by a referenced PBDOM\_OBJECT.

#### Syntax

```

pbdom_attribute_name.InsertContent(pbdom_object pbdom_object_new, pbdom_object
pbdom_object_ref)

```

**Table 5.35:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
pbdom_object_new	The PBDOM_OBJECT to be inserted
pbdom_object_ref	A positional reference to a PBDOM_OBJECT before which pbdom_object_new is to be inserted

## Return value

PBDOM\_OBJECT. The PBDOM\_ATTRIBUTE returned as a PBDOM\_OBJECT.

## Throws

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- The PBDOM\_OBJECT to be inserted is nameable and has not been given a user-defined name.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT -- The PBDOM\_OBJECT to be inserted already has a parent.

EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT -- The PBDOM\_OBJECT to be inserted is not valid to be inserted as a child of this PBDOM\_ATTRIBUTE.

EXCEPTION\_WRONG\_PARENT\_ERROR -- The reference PBDOM\_OBJECT is not a child of this PBDOM\_ATTRIBUTE.

## Examples

This example adds an attribute to the root element with the name `my_attr` and text content "attribute text". It then creates a PBDOM\_ENTITYREFERENCE object named `ent_ref` and inserts it before the attribute's current content. Testing the new content of the attribute should return "&ent\_ref;attribute text";

Consider the following code :

```
PBDOM_DOCUMENT          pbdom_doc
PBDOM_ATTRIBUTE         pbdom_attr
PBDOM_ENTITYREFERENCE   pbdom_entref
PBDOM_OBJECT            pbdom_obj_array[]

try
  pbdom_doc = Create PBDOM_DOCUMENT
  pbdom_entref = Create PBDOM_ENTITYREFERENCE

  // Create a new document object.
  pbdom_doc.NewDocument ("root")
  // Add an attribute "my_attr" to the root element.
  pbdom_doc.GetRootElement().SetAttribute("my_attr", &
    "attribute text")
  // Set the name of the PBDOM_ENTITYREFERENCE.
  pbdom_entref.SetName ("ent_ref")

  // Get the existing contents of my_attr
  pbdom_doc.GetRootElement().GetAttribute("my_attr").&
    GetContent(pbdom_obj_array)

// Insert the entity reference to the root element's
// my_attr attribute before the attribute text.
pbdom_doc.GetRootElement().GetAttribute("my_attr").&
  InsertContent(pbdom_entref, pbdom_obj_array[1])

// Test the text contents of "my_attr"
if pbdom_doc.GetRootElement(). &
  GetAttribute("my_attr").GetText() = &
  "&ent_ref;attribute text" then
  MessageBox ("Pass", &
    "GetText() on my_attr is correct.")
else
  MessageBox ("Fail", &
    "GetText() on my_attr is incorrect.")
```

```

end if

catch (pbdom_exception pbdom_except)
    MessageBox ("PBDOM_EXCEPTION", &
        pbdom_except.GetMessage())
end try

```

## Usage

This method inserts the input PBDOM\_OBJECT as a child at a specific position (before the reference PBDOM\_OBJECT). Currently, only a PBDOM\_TEXT and a PBDOM\_ENTITYREFERENCE object can be inserted as a child of a PBDOM\_ATTRIBUTE.

If the reference PBDOM\_OBJECT is null, the PBDOM\_OBJECT to be inserted is inserted at the end of this PBDOM\_ATTRIBUTE object's list of children.

## See also

[AddContent](#)

[GetContent](#)

[RemoveContent](#)

[SetContent](#)

## 5.1.29 IsAncestorObjectOf

### Description

Determines whether the current PBDOM\_ATTRIBUTE object is the ancestor of another PBDOM\_OBJECT.

### Syntax

```
pbdom_attribute_name.IsAncestorObjectOf(pbdom_object pbdom_object_ref)
```

**Table 5.36:**

Argument	Description
pbdom_document_name	The name of a PBDOM_ATTRIBUTE object
pbdom_object_ref	A reference to a PBDOM_OBJECT to check against

### Return value

Boolean.

Returns true if this PBDOM\_ATTRIBUTE is the ancestor of the input PBDOM\_PBOBJECT and false otherwise.

### Throws

EXCEPTION\_INVALID\_ARGUMENT -- The input PBDOM\_OBJECT is invalid. This can happen if it has not been initialized properly or is a null object reference.

### Usage

This method checks to see whether the current PBDOM\_ATTRIBUTE is the ancestor object of the input PBDOM\_OBJECT. According to the W3C DOM specification, only a PBDOM\_TEXT and a PBDOM\_ENTITYREFERENCE object can become a child object of a PBDOM\_ATTRIBUTE, and therefore a PBDOM\_ATTRIBUTE can only be an ancestor of a PBDOM\_TEXT or a PBDOM\_ENTITYREFERENCE object.

### 5.1.30 RemoveContent

#### Description

Removes the input PBDOM\_OBJECT from the PBDOM\_ATTRIBUTE.

#### Syntax

```
pbdom_attribute_name.RemoveContent(pbdom_object pbdom_object_ref)
```

**Table 5.37:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
pbdom_object_ref	The PBDOM_OBJECT child to be removed from this PBDOM_ATTRIBUTE

#### Return value

Boolean.

Returns true if the content has been successfully removed and false otherwise.

#### Throws

EXCEPTION\_INVALID\_ARGUMENT -- The input PBDOM\_OBJECT is invalid. This can happen if it has not been initialized properly or is a null object reference.

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_ATTRIBUTE object or the input PBDOM\_OBJECT is not associated with a derived PBDOM\_OBJECT class object.

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- This PBDOM\_ATTRIBUTE or the PBDOM\_OBJECT to be removed is nameable and has not been given a user-defined name.

EXCEPTION\_WRONG\_DOCUMENT\_ERROR -- The input PBDOM\_OBJECT is not contained within the same PBDOM\_DOCUMENT as this PBDOM\_ATTRIBUTE.

EXCEPTION\_WRONG\_PARENT\_ERROR -- The input PBDOM\_OBJECT is not a child of the current PBDOM\_ATTRIBUTE.

#### Examples

This example adds an attribute to the root element with the name my\_attr and text content "attribute text". It then creates a PBDOM\_ENTITYREFERENCE object named ent\_ref and inserts it before the attribute's current content.

At this point, my\_attr contains two child PBDOM\_OBJECTs: a PBDOM\_TEXT containing "attribute text" and a PBDOM\_ENTITYREFERENCE named ent\_ref. The element looks like this when serialized:

```
<root my_attr="attribute text&ent_ref;">
```

A call to `GetContent` returns an array containing these two `PBDOM_OBJECT`s. `pbdom_obj_array[1]` should point to the `PBDOM_TEXT`. After `pbdom_obj_array[1]` is removed from `my_attr`, the element looks like this when serialized: `<root my_attr="&ent_ref;">`.

```
PBDOM_DOCUMENT      pbdom_doc
PBDOM_ATTRIBUTE     pbdom_attr
PBDOM_ENTITYREFERENCE pbdom_entref
PBDOM_OBJECT        pbdom_obj_array[]

try
  pbdom_doc = Create PBDOM_DOCUMENT
  pbdom_entref = Create PBDOM_ENTITYREFERENCE

  // Create a new document object.
  pbdom_doc.NewDocument ("root")
  // Add an attribute "my_attr" to the root element.
  pbdom_doc.GetRootElement().SetAttribute("my_attr", &
    "attribute text")

  // Set the name of our PBDOM_ENTITYREFERENCE.
  pbdom_entref.SetName ("ent_ref")

  // Add the entity reference to the root
  // element's "my_attr" attribute.
  pbdom_doc.GetRootElement(). &
    GetAttribute("my_attr"). AddContent(pbdom_entref)

  // Get the existing contents of "my_attr"
  pbdom_doc.GetRootElement().GetAttribute("my_attr").&
    GetContent(pbdom_obj_array)

  // Remove PBDOM_TEXT object from "my_attr"
  pbdom_doc.GetRootElement().GetAttribute("my_attr").&
    RemoveContent(pbdom_obj_array[1])

  // Test the text contents of "my_attr "
  if pbdom_doc.GetRootElement(). &
    GetAttribute("my_attr").GetText() = &
    "&ent_ref;" then
    MessageBox ("Pass", &
      "GetText() on my_attr is correct.")
  else
    MessageBox ("Fail",
      "GetText() on my_attr is incorrect.")
  end if

catch (pbdom_exception pbdom_e)
  MessageBox ("PBDOM_EXCEPTION", pbdom_e.GetMessage())
end try
```

## Usage

The `RemoveContent` method removes the input `PBDOM_OBJECT` from this `PBDOM_ATTRIBUTE`. Currently, only a `PBDOM_TEXT` and a `PBDOM_ENTITYREFERENCE` object can be part of the contents of a `PBDOM_ATTRIBUTE`. Therefore, the input `PBDOM_OBJECT` must be either a `PBDOM_TEXT` or a `PBDOM_ENTITYREFERENCE` object.

## See also

[AddContent](#)[GetContent](#)[InsertContent](#)[SetContent](#)

### 5.1.31 SetBooleanValue

#### Description

Sets the text value of a PBDOM\_ATTRIBUTE object. The SetBooleanValue method creates this text value by serializing the provided boolean value into a string.

#### Syntax

```
pbdom_attribute_name.SetBooleanValue(boolean boolValue)
```

**Table 5.38:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
boolValue	A boolean value to be set for the PBDOM_ATTRIBUTE

#### Return value

PBDOM\_ATTRIBUTE. The PBDOM\_ATTRIBUTE from which the SetBooleanValue method was invoked.

#### See also

[GetBooleanValue](#)

### 5.1.32 SetContent

#### Description

Sets the content of this PBDOM\_ATTRIBUTE.

#### Syntax

```
pbdom_attribute_name.SetContent(pbdom_object pbdom_object_array)
```

**Table 5.39:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
pbdom_object_array	An array of PBDOM_OBJECTs

#### Return value

PBDOM\_OBJECT. This PBDOM\_ATTRIBUTE modified.

#### Throws



EXCEPTION\_ILLEGAL\_PBOBJECT -- One of the array items is not a valid PBDOM object. This can happen if the array item has not been initialized properly or is a null object reference. This is similar to EXCEPTION\_INVALID\_ARGUMENT.

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- One of the array items is nameable and has not been given a user-defined name.

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- One of the array items is not associated with a derived PBDOM\_OBJECT.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT -- One of the array items already has a parent.

EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT -- One of the array items is not allowed to be set as part of the contents of a PBDOM\_ATTRIBUTE.

### Examples

This example demonstrates setting the contents of a PBDOM\_ATTRIBUTE object. It creates a PBDOM\_DOCUMENT with root element root and attaches to it a PBDOM\_DOCTYPE with the following internal subset:

```
<!ELEMENT root ANY>
<!ATTLIST root attr CDATA #REQUIRED>
<!ENTITY ent_ref "MY ENTITY REFERENCE">
```

It also creates a PBDOM\_ATTRIBUTE, attr, and sets as its contents an array of three PBDOM\_OBJECTS:

- A PBDOM\_TEXT with the text value "start text "
- A PBDOM\_ENTITYREFERENCE named ent\_ref
- A PBDOM\_TEXT with the text value " end text."

This removes the original contents of attr and sets new contents so that when the document is serialized into an external file, the root element looks like this:

```
<root attr="start text &ent_ref; end text."/>
```

Finally, a user-defined function called GetAttributeText parses the external serialized XML file and retrieves the text value of the attr attribute.

The code for GetAttributeText function is as follows:

```
PBDOM_BUILDER      pbdom_buildr
PBDOM_DOCUMENT     pbdom_doc
string             strReturn

try
  pbdom_buildr = Create PBDOM_BUILDER
  pbdom_doc = pbdom_buildr. &
    BuildFromFile (strXMLFileName)

  strReturn = pbdom_doc.GetRootElement(). &
    GetAttribute(strAttributeName).GetText()
catch (PBDOM_EXCEPTION pbdom_except)
  strReturn = ""
end try
return strReturn
```

This function builds a PBDOM\_DOCUMENT from the external XML file (its first argument) and gets the text value of an attribute (its second argument) from the root element.

The code that sets the content of the PBDOM\_ATTRIBUTE is as follows:

```
PBDOM_DOCUMENT      pbdom_doc
PBDOM_DOCTYPE       pbdom_doctyp
PBDOM_ATTRIBUTE     pbdom_attr
PBDOM_TEXT          pbdom_txt
PBDOM_OBJECT        pbdom_obj_array_set[]
long l = 0

try
  pbdom_doc = Create PBDOM_DOCUMENT
  pbdom_doc.NewDocument ("root")

  pbdom_doctyp = Create PBDOM_DOCTYPE
  pbdom_doctyp.SetName ("root")
  pbdom_doctyp.setinternalsubset("<!ELEMENT root ANY><!ATTLIST root attr CDATA
#REQUIRED><!ENTITY ent_ref ~"MY ENTITY REFERENCE~">")

  pbdom_doc.SetDocType(pbdom_doctyp)

  pbdom_doc.GetRootElement().SetAttribute("attr", "")

  pbdom_obj_array_set[1] = Create PBDOM_TEXT
  pbdom_txt = pbdom_obj_array_set[1]
  pbdom_txt.SetText ("start text ")

  pbdom_obj_array_set[2] = Create PBDOM_ENTITYREFERENCE
  pbdom_obj_array_set[2].SetName("ent_ref")

  pbdom_obj_array_set[3] = Create PBDOM_TEXT
  pbdom_txt = pbdom_obj_array_set[3]
  pbdom_txt.SetText (" end text.")

  pbdom_doc.GetRootElement().GetAttribute("attr"). &
    SetContent(pbdom_obj_array_set)

  pbdom_doc.SaveDocument &
    ("c:\xmltests\attr_set_content.xml")

  MessageBox ("Attribute Text", GetAttributeText &
    ("c:\xmltests\attr_set_content.xml", "attr"))

catch (PBDOM_EXCEPTION pbdom_e)
  MessageBox ("PBDOM_EXCEPTION", pbdom_e.GetMessage())
end try
```

## Usage

This method sets the content of this PBDOM\_ATTRIBUTE. The supplied array should contain only objects of type PBDOM\_TEXT and PBDOM\_ENTITYREFERENCE.

When all objects in the supplied array are legal and before the new content is added, all objects in the old content will have their parentage set to null (no parent) and the old content list will be cleared from this PBDOM\_ATTRIBUTE.

This has the effect that the items of any active array (previously obtained with a call to GetContent) also change to reflect the new condition. In addition, all objects in the supplied array have their parentage set to this PBDOM\_ATTRIBUTE.

Passing a null value or an empty array clears the existing content of this PBDOM\_ATTRIBUTE.

### See also

[AddContent](#)

[GetContent](#)

[RemoveContent](#)

[SetContent](#)

## 5.1.33 SetDateValue

### Description

Sets the text value of a PBDOM\_ATTRIBUTE object. The SetDateValue method creates this text value by serializing the provided date value into a string.

### Syntax

```
pbdom_attribute_name.SetDateValue(date dateValue, strDateFormat)
```

**Table 5.40:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
dateValue	A date value to be set for the PBDOM_ATTRIBUTE
strDateFormat	The format in which the date value is to be set for the PBDOM_ATTRIBUTE, for example, MM:DD:YYYY

The value of the strDateFormat parameter can include slashes or colons as delimiters. The following table illustrates characters having special meaning in strDateFormat.

**Table 5.41:**

Character	Meaning	Example
D	Day number with no leading zero	5
DD	Day number with leading zero, if applicable	05
M	Month number with no leading zero	5
MM	Month number with leading zero, if applicable	05
YY	Two-digit year number	05
YYYY	Four-digit year number	2005

### Return value

PBDOM\_ATTRIBUTE. The PBDOM\_ATTRIBUTE from which the SetDateValue method was invoked.

### See also

[GetDateValue](#)

## 5.1.34 SetDateTimeValue

### Description

Sets the text value of a PBDOM\_ATTRIBUTE object and creates this text value by serializing the provided datetime value into a string.

### Syntax

```
pbdom_attribute_name.SetDateTimeValue(datetime datetimeValue, string strDateFormat, string strTimeFormat)
```

**Table 5.42:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
datetimeValue	A datetime value to be set for the PBDOM_ATTRIBUTE
strDateFormat	The format in which the date part of the datetime value is to be set for the PBDOM_ATTRIBUTE, for example, MM:DD:YYYY
strTimeFormat	The format in which the time part of the datetime value is to be set for the PBDOM_ATTRIBUTE, for example, HH:MM:SS

The value of the strDateFormat parameter can use slashes or colons as delimiters. The following table illustrates characters that have special meaning in strDateFormat.

**Table 5.43:**

Character	Meaning	Example
D	Day number with no leading zero	5
DD	Day number with leading zero, if applicable	05
M	Month number with no leading zero	5
MM	Month number with leading zero, if applicable	05
YY	Two-digit year number	05
YYYY	Four-digit year number	2005

The value of the `strTimeFormat` parameter can include slashes or colons as delimiters. The following table illustrates characters that have special meaning in `strTimeFormat`.

**Table 5.44:**

Character	Meaning	Example
H	Hour number with no leading zero	5
HH	Hour number with leading zero, if applicable	05
M	Minutes number with no leading zero	5
MM	Minutes number with leading zero, if applicable	05
S	Seconds number with no leading zero	5
SS	Seconds number with leading zero, if applicable	55

**Return value**

PBDOM\_ATTRIBUTE. The PBDOM\_ATTRIBUTE from which the `SetDateTimeValue` method was invoked.

**See also**

[GetDateTimeValue](#)

**5.1.35 SetDoubleValue****Description**

Sets the text value of a PBDOM\_ATTRIBUTE object. The `SetDoubleValue` method creates this text value by serializing the provided double value into a string.

**Syntax**

```
pbdom_attribute_name.SetDoubleValue( double doubleValue)
```

**Table 5.45:**

Argument	Description
<code>pbdom_attribute_name</code>	The name of the PBDOM_ATTRIBUTE
<code>doubleValue</code>	A double value to be set for the PBDOM_ATTRIBUTE

**Return value**

PBDOM\_ATTRIBUTE. The PBDOM\_ATTRIBUTE from which the `SetDoubleValue` method was invoked.

**See also**

[GetDoubleValue](#)

### 5.1.36 SetIntValue

#### Description

Sets the text value of a PBDOM\_ATTRIBUTE object. The SetIntValue method creates this text value by serializing the provided int value into a string.

#### Syntax

```
pbdom_attribute_name.SetIntValue(integerintValue)
```

**Table 5.46:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
intValue	An int value to be set for the PBDOM_ATTRIBUTE

#### Return value

PBDOM\_ATTRIBUTE. The PBDOM\_ATTRIBUTE from which the SetIntValue method was invoked.

#### See also

[GetIntValue](#)

### 5.1.37 SetLongValue

#### Description

Sets the text value of a PBDOM\_ATTRIBUTE object. The SetLongValue method creates this text value by serializing the provided long value into a string.

#### Syntax

```
pbdom_attribute_name.SetLongValue( long longValue)
```

**Table 5.47:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
longValue	A long value to be set for the PBDOM_ATTRIBUTE

#### Return value

PBDOM\_ATTRIBUTE. The PBDOM\_ATTRIBUTE from which the SetLongValue method was invoked.

#### See also

[GetLongValue](#)

### 5.1.38 SetName

#### Description

Sets the local name of the PBDOM\_ATTRIBUTE object.

#### Syntax

```
pbdom_attribute_name.SetName(string strName)
```

**Table 5.48:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
strName	The new local name for the PBDOM_ATTRIBUTE

#### Return value

Boolean.

Returns true if the local name of the PBDOM\_ATTRIBUTE has been changed and false otherwise.

#### Throws

EXCEPTION\_INVALID\_NAME -- If the input name is not valid for a local name of a PBDOM\_ATTRIBUTE. This happens if the name is an empty string, if the name contains a namespace prefix, or if the name is already the name of an existing attribute of the owning element.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- Insufficient memory was encountered while executing this method.

#### Examples

This example shows how to set the local name of a PBDOM\_ATTRIBUTE and demonstrates that the namespace information it contains is not affected by a change in name.

The sample code first builds a PBDOM\_DOCUMENT from a string that contains XML that has a single root element with a namespace declaration and an attribute a.

The GetAttribute method obtains the attribute a, which does not belong to a namespace, and the returned PBDOM\_ATTRIBUTE is tested and should be valid. After a call to SetName, the code confirms the name change and tests that the namespace information remains the same (the namespace prefix and URI are both still empty strings):

```
PBDOM_BUILDER      pbdom_buildr
PBDOM_DOCUMENT     pbdom_doc
PBDOM_ATTRIBUTE    pbdom_attr
string strXML = "<root xmlns:n1=~\"http://www.n.com~\" a=~\"123~\"/>"

try
    pbdom_buildr = Create PBDOM_BUILDER
    pbdom_doc = pbdom_buildr.BuildFromString (strXML)

    pbdom_attr = pbdom_doc.GetRootElement(). &
        GetAttribute("a")
```

```
if (IsValid(pbdom_attr)) then
    MessageBox ("Pass", &
        "PBDOM_ATTRIBUTE a is retrieved via the " &
        + "NONAMESPACE GetAttribute() method.")
else
    MessageBox ("Fail", &
        "PBDOM_ATTRIBUTE should have been retrievable.")
end if

pbdom_attr.SetName ("b")

if pbdom_attr.GetName() = "b" then
    MessageBox ("Pass", "Name has been changed to b.")
else
    MessageBox ("Fail", &
        "Name should have been changed to b.")
end if

if pbdom_attr.GetNamespacePrefix() = "" then
    MessageBox ("Pass", &
        "Namespace Prefix is an empty string.")
else
    MessageBox ("Fail", "Namespace Prefix is : " &
        + pbdom_attr.GetNamespacePrefix() &
        + " which is incorrect.")
end if

if pbdom_attr.GetNamespaceURI() = "" then
    MessageBox ("Pass", &
        "Namespace URI is an empty string.")
else
    MessageBox ("Fail", "Namespace URI is : " &
        + pbdom_attr.GetNamespaceURI() &
        + " which is incorrect.")
end if

catch(PBDOM_EXCEPTION pbdom_e)
    MessageBox("PBDOM_EXCEPTION", pbdom_e.GetMessage())
end try
```

## Usage

This method sets the local name of the PBDOM\_ATTRIBUTE. When a PBDOM\_ATTRIBUTE is first created, it has no name and the namespace information is by default set to the NONAMESPACE namespace. (Its NS Prefix and URI are both empty strings.)

The SetName method is used to set the local name of the PBDOM\_ATTRIBUTE. The SetNamespace method is used to set the Namespace Prefix and URI of the PBDOM\_ATTRIBUTE.

If a PBDOM\_ATTRIBUTE is retrieved programmatically from a parsed document, then the name and namespace information of the PBDOM\_ATTRIBUTE are inherited from the referred attribute of the parsed document. The name and namespace information of the PBDOM\_ATTRIBUTE, however, can still be modified using the SetName and SetNamespace methods.

Note that according to the W3C "Namespaces in XML" specification, when the SetName method is invoked on a PBDOM\_ATTRIBUTE, if the PBDOM\_ATTRIBUTE (PBDOM\_ATTRIBUTE 1) has an owner PBDOM\_ELEMENT that contains an existing PBDOM\_ATTRIBUTE (PBDOM\_ATTRIBUTE 2) with the same name (to be set



for PBDOM\_ATTRIBUTE 1) and namespace URI as PBDOM\_ATTRIBUTE 1, the EXCEPTION\_INVALID\_NAME exception will be thrown.

### See also

[GetName](#)

[SetOwnerElementObject](#)

## 5.1.39 SetNamespace

### Description

Sets the namespace for a PBDOM\_ATTRIBUTE object based on the specified namespace prefix and URI.

### Syntax

```
pbdom_attribute_name.SetNamespace(stringstrNamespacePrefix, string  
stringstrNamespaceUri, boolean bVerifyNamespace)
```

**Table 5.49:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
strNamespacePrefix	A string containing the namespace prefix to be set for the PBDOM_ATTRIBUTE
strNamespaceUri	A string containing the namespace URI to be set for the PBDOM_ATTRIBUTE
bVerifyNamespace	A boolean value to indicate whether to search for an in-scope namespace declaration that matches the input namespace prefix and URI

### Return value

Long.

Returns 0 if namespace information was set successfully and -1 if no in-scope namespace matching the input prefix and URI exists.

### Throws

EXCEPTION\_INVALID\_NAME -- If the input namespace prefix or the URI or the combination of prefix and URI is not valid. This occurs if:

- The namespace prefix is an empty string and the URI is not an empty string. If both are empty strings, the NONAMESPACE namespace is being specified and this prefix/URI combination is correct.
- The namespace Prefix is xmlns and the URI is not http://www.w3.org/2000/xmlns/. This namespace prefix/URI pair is unique and exclusive. Its elements cannot be used individually and separately. The use of this pair signifies a namespace declaration.
- The namespace prefix string is invalid. That is, it does not conform to the W3C "Namespaces in XML" specifications for the name of a prefix.

- The namespace URI string is invalid. That is, it does not conform to the W3C specifications for a URI string.
- The owner Element of this PBDOM\_ATTRIBUTE already contains an attribute that has the same name as the current PBDOM\_ATTRIBUTE and belongs to the namespace that is to be set for the current PBDOM\_ATTRIBUTE.

**EXCEPTION\_INVALID\_ARGUMENT** -- If the input namespace prefix string or the URI string has been set to null.

**EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE** -- If there is insufficient memory to allocate for internal strings.

**EXCEPTION\_INTERNAL\_XML\_ENGINE\_ERROR** -- If some internal error occurred in the XML engine.

### Examples

This example demonstrates how to set the namespace prefix and URI for a PBDOM\_ATTRIBUTE. It creates a PBDOM\_DOCUMENT based on the following XML document:

```
<root xmlns:pre1="http://www.pre.com">
  <child1 pre1:a="123" b="456"/>
</root>
```

The namespace `http://www.pre.com`, which has the prefix `pre1`, is defined in the root element. The child element `child1` has an attribute `a` that belongs to the declared namespace and an attribute `b` that does not belong to a namespace.

The example uses `GetAttribute` to get and store the attribute `b` in `pbdom_attr`, then calls `SetNamespace` on `pbdom_attr`, specifying the strings `"pre1"` and `"http://www.pre.com"` as the prefix and URI, and setting the `bVerifyNamespace` parameter to `true`. This tells `SetNamespace` to check first to see if the owner element of `b` or the owner element's ancestor elements contain a namespace declaration for the `pre1/http://www.pre.com` namespace prefix/URI pair.

The search for this prefix/URI pair succeeds because the root element contains such a namespace declaration.

```
PBDOM_BUILDER    pbdom_buildr
PBDOM_DOCUMENT  pbdom_doc
PBDOM_ATTRIBUTE  pbdom_attr
string strXML = "<root xmlns:pre1=~\"http://www.pre.com~\"><child1 pre1:a=~\"123~\"
  b=~\"456~\"/></root>"

try
  pbdom_buildr = Create PBDOM_BUILDER
  pbdom_doc = pbdom_buildr.BuildFromString (strXML)

  pbdom_attr =
  pbdom_doc.GetRootElement().GetChildElement("child1").GetAttribute("b", "", "")

  pbdom_attr.SetNamespace("pre1", "http://www.pre.com", true)

  MessageBox ("NS Prefix", pbdom_attr.GetNamespacePrefix())
  MessageBox ("NS URI", pbdom_attr.GetNamespaceURI())
  MessageBox ("Name", pbdom_attr.getName())
  MessageBox ("Text", pbdom_attr.getText())
```

```

pbdom_doc.SaveDocument ("ns.xml")

catch (PBDOM_EXCEPTION pbdom_except)
    MessageBox ("PBDOM_EXCEPTION", pbdom_except.GetMessage())
end try

```

There is no other attribute inside child1 that has the name b and that also belongs to the http://www.pre.com namespace, so the SetNamespace method succeeds. When serialized, the PBDOM\_DOCUMENT looks like this:

```

<root xmlns:prel="http://www.pre.com">
  <child1 prel:b="456" prel:a="123" />
</root>

```

## Usage

This method sets this PBDOM\_ATTRIBUTE object's namespace based on the input prefix and URI. The input prefix can be an empty string, but the input URI cannot be an empty string unless the prefix is also an empty string.

If the input prefix and URI are both empty strings, the PBDOM\_ATTRIBUTE has no namespace. The bVerifyNamespace parameter tells the method whether to search for an in-scope namespace declaration that matches the input namespace prefix and URI.

As required by the W3C specification on "Namespaces in XML," if the current PBDOM\_ATTRIBUTE has an owner PBDOM\_ELEMENT that contains an existing PBDOM\_ATTRIBUTE that has the same name as the current PBDOM\_ATTRIBUTE and the same namespace URI as is to be set for the current PBDOM\_ATTRIBUTE, the EXCEPTION\_INVALID\_NAME exception is thrown.

## See also

[GetName](#)

[GetNamespacePrefix](#)

[GetNamespaceUri](#)

[GetQualifiedName](#)

[SetName](#)

### 5.1.40 SetOwnerElementObject

#### Description

Sets the input PBDOM\_ELEMENT as the owner of the current PBDOM\_ATTRIBUTE.

#### Syntax

```
pbdom_attribute_name.SetOwnerElementObject(pbdom_element pbdom_element_ref)
```

**Table 5.50:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
pbdom_element_ref	The PBDOM_ELEMENT to be set as the owner of this current PBDOM_ATTRIBUTE

## Return value

PBDOM\_ATTRIBUTE. This PBDOM\_ATTRIBUTE itself modified and returned.

## Throws

EXCEPTION\_INVALID\_ARGUMENT -- The input PBDOM\_ELEMENT is invalid. This can happen if it has not been initialized properly or is a null object reference.

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- The internal implementation of the PBDOM\_ATTRIBUTE object or the input PBDOM\_ELEMENT object is null. The occurrence of this exception is rare but can take place if severe memory corruption occurs.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_OWNER -- This PBDOM\_ATTRIBUTE already has an owner Element.

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- The input PBDOM\_ELEMENT has not been named.

EXCEPTION\_INVALID\_NAME -- The input PBDOM\_ELEMENT already contains an attribute that has the same name and that belongs to the same namespace as this current PBDOM\_ATTRIBUTE.

## Examples

This example moves the positions of two PBDOM\_ATTRIBUTE objects from one element to another.

In the string strXML from which a PBDOM\_DOCUMENT is created, the abc root element contains a namespace declaration and two attributes. My\_Attr belongs to no namespace, and pre:My\_Attr\_NS belongs to the http://www.pre.com namespace.

The example obtains handles for the two attributes and the data element, then detaches both attributes from abc and sets data as their new owner:

```
PBDOM_BUILDER      pbdombuilder_new
PBDOM_DOCUMENT     pbdom_doc
PBDOM_ATTRIBUTE    pbdom_attr
PBDOM_ATTRIBUTE    pbdom_attr_ns
PBDOM_ELEMENT      pbdom_elem_data
string strXML = "<abc My_Attr=~\"Attribute Value~\" pre:My_Attr_NS=~\"Attribute Value
NS~\" xmlns:pre=~\"http://www.pre.com~\"><data>Data</data></abc>"

TRY
pbdombuilder_new = Create PBDOM_Builder
pbdom_doc = pbdombuilder_new.BuildFromString(strXML)

pbdom_attr = pbdom_doc.GetRootElement(). &
    GetAttribute("My_Attr")
pbdom_attr_ns = pbdom_doc.GetRootElement(). &
    GetAttribute("My_Attr_NS", "pre", &
    "http://www.pre.com")
pbdom_elem_data = pbdom_doc.GetRootElement(). &
    GetChildElement("data")

pbdom_attr.Detach()
pbdom_attr.SetOwnerElementObject (pbdom_elem_data)

pbdom_attr_ns.Detach()
pbdom_attr_ns.SetOwnerElementObject (pbdom_elem_data)

pbdom_doc.SaveDocument("setownerelementobject.xml")
```

```

Destroy pbdombuilder_new
Destroy pbdom_doc

CATCH (PBDOM_Exception except)
    MessageBox ("Exception Occurred", except.Text)
END TRY

```

When the document is serialized, the XML looks like this:

```

<abc xmlns:pre="http://www.pre.com">
<data pre:My_Attr_NS="Attribute Value NS" My_Attr="Attribute Value">Data</data>
</abc>

```

## Usage

According to the "Namespace in XML" specifications, an element cannot contain two attributes with the same local name and namespace URI. This is true even if the prefixes of the two attributes are different. An exception is thrown if this rule is violated when `SetOwnerElementObject` is invoked.

## See also

[GetOwnerElementObject](#)

### 5.1.41 SetRealValue

#### Description

Sets the text value of a `PBDOM_ATTRIBUTE` object. The `SetRealValue` method creates this text value by serializing the provided real value into a string.

#### Syntax

```
pbdom_attribute_name.SetRealValue(real realValue)
```

**Table 5.51:**

Argument	Description
<code>pbdom_attribute_name</code>	The name of the <code>PBDOM_ATTRIBUTE</code>
<code>realValue</code>	A real value to be set for the <code>PBDOM_ATTRIBUTE</code>

#### Return value

`PBDOM_ATTRIBUTE`. The `PBDOM_ATTRIBUTE` from which the `SetRealValue` method was invoked.

## See also

[GetRealValue](#)

### 5.1.42 SetText

#### Description

Sets the string value of a `PBDOM_ATTRIBUTE` object.

#### Syntax

```
pbdom_attribute_name.SetText(string strText)
```

**Table 5.52:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
strText	The string value to be set in the PBDOM_ATTRIBUTE

**Return value**

PBDOM\_ATTRIBUTE.

**Usage**

This method returns the current PBDOM\_ATTRIBUTE with the input string value set.

This method is the counterpart of the JDOM setValue method.

**See also**[GetText](#)[GetTextNormalize](#)[GetTextTrim](#)**5.1.43 SetTimeValue****Description**

Sets the text value of a PBDOM\_ATTRIBUTE object. The SetTimeValue method creates this text value by serializing the provided time value into a string.

**Syntax**

```
pbdom_attribute_name.SetTimeValue(time timeValue, string strTimeFormat)
```

**Table 5.53:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
timeValue	A time value to be set for the PBDOM_ATTRIBUTE
strTimeFormat	The format in which the time value is to be set for the PBDOM_ATTRIBUTE, for example, HH:MM:SS

The value of the strTimeFormat parameter can use slashes or colons as delimiters. The following table illustrates characters that have special meaning in strTimeFormat.

**Table 5.54:**

Character	Meaning	Example
H	Hour number with no leading zero	5

Character	Meaning	Example
HH	Hour number with leading zero, if applicable	05
M	Minutes number with no leading zero	5
MM	Minutes number with leading zero, if applicable	05
S	Seconds number with no leading zero	5
SS	Seconds number with leading zero, if applicable	55

**Return value**

PBDOM\_ATTRIBUTE. The PBDOM\_ATTRIBUTE from which the SetTimeValue method was invoked.

**See also**

[GetTimeValue](#)

**5.1.44 SetUIntValue****Description**

Sets the text value of a PBDOM\_ATTRIBUTE object. The SetUIntValue method creates this text value by serializing the provided uint value into a string.

**Syntax**

```
pbdom_attribute_name.SetUIntValue(unsignedinteger uintValue)
```

**Table 5.55:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
uintValue	A uint value to be set for the PBDOM_ATTRIBUTE

**Return value**

PBDOM\_ATTRIBUTE. The PBDOM\_ATTRIBUTE from which the SetUIntValue method was invoked.

**See also**

[GetUIntValue](#)

**5.1.45 SetUlongValue****Description**

Sets the text value of a PBDOM\_ATTRIBUTE object. The SetUlongValue method creates this text value by serializing the provided ulong value into a string.

**Syntax**

```
pbdom_attribute_name.SetUlongValue(unsignedlong ulongValue)
```

**Table 5.56:**

Argument	Description
pbdom_attribute_name	The name of the PBDOM_ATTRIBUTE
ulongValue	A ulong value to be set for the PBDOM_ATTRIBUTE

**Return value**

PBDOM\_ATTRIBUTE. The PBDOM\_ATTRIBUTE from which the SetUlongValue method was invoked.

**See also**

[GetUlongValue](#)



## 6 PBDOM\_BUILDER Class

### About this chapter

This chapter describes the PBDOM\_BUILDER class.

### 6.1 PBDOM\_BUILDER

#### Description

The PBDOM\_BUILDER class serves as a DOM factory that creates a PBDOM\_DOCUMENT from various input sources, such as a string and a DataStore. A PBDOM\_BUILDER class is not a PBDOM\_OBJECT. There are no DOM objects to which you can map a PBDOM\_BUILDER class.

The PBDOM\_BUILDER methods can be contrasted with the PBDOM\_DOCUMENT NewDocument methods (overloaded with several versions) that are intended to be used to build a PBDOM\_DOCUMENT from scratch.

#### Methods

PBDOM\_BUILDER has the following methods:

[BuildFromDataStore](#)

[BuildFromFile](#)

[BuildFromString](#)

[GetParseErrors](#)

#### 6.1.1 BuildFromDataStore

##### Description

Builds a PBDOM\_DOCUMENT from the referenced DataStore object.

##### Syntax

```
pbdom_builder_name.BuildFromDataStore(datastore datastore_ref)
```

**Table 6.1:**

Argument	Description
pbdom_builder_name	The name of a PBDOM_BUILDER object
datastore_ref	A DataStore object

##### Return value

PBDOM\_DOCUMENT.

##### Throws

EXCEPTION\_INVALID\_ARGUMENT -- The input DataStore object is invalid. This can happen if it has not been initialized properly or is a null object reference.

##### Examples

The following PowerScript code fragment demonstrates how to use the `BuildFromDataStore` method with a referenced `DataStore` object.

```
PBDOM_Builder pbdom_bldr
pbdom_document pbdom_doc
datastore ds

ds = Create datastore
ds.DataObject = "d_customer"
ds.SetTransObject (SQLCA)
ds.Retrieve()

pbdom_doc = pbdom_bldr.BuildFromDataStore(ds)
```

In this example, a `DataStore` object `ds` is created and populated with data, and then passed to the `BuildFromDataStore` method. The `BuildFromDataStore` method causes the `DataStore` to export the data to XML, using the most current XML template for the `DataStore`, and then it uses the XML to build a `PBDOM_DOCUMENT`. The `PBDOM_DOCUMENT` object is assigned to `pbdom_doc`.

### Usage

This method creates a temporary file in the directory pointed to by the user's `TMP` environment variable. If this directory is invalid, the temporary file is created in the Windows `\temp` directory.

The encoding specified in the XML export template has no effect on the encoding of the document created using `BuildFromDataStore`. It always has UTF-16LE encoding.

### See also

[BuildFromFile](#)

[BuildFromString](#)

## 6.1.2 BuildFromFile

### Description

Builds a `PBDOM_DOCUMENT` from the file pointed to by the input URL string. The URL can be a local file path.

### Syntax

```
pbdom_builder_name.BuildFromFile (string strURL)
```

**Table 6.2:**

Argument	Description
<code>pbdom_builder_name</code>	The name of a <code>PBDOM_BUILDER</code> object
<code>strURL</code>	A string that indicates the URL of the file from which to build a <code>PBDOM_DOCUMENT</code>

### Return value

`PBDOM_DOCUMENT`.

### Throws

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- If there is insufficient memory to create a PBDOM\_DOCUMENT object.

## Examples

Suppose the file c:\pbdom\_doc\_1.xml contains the following XML string:

```
<!DOCTYPE abc [<!ENTITY text "Some Text" >]>
<abc>
  <data>
    <child_data>Child Data Text</child_data>
    <child_data An_Attribute="Some Attribute Value"/>
    &text;
    <!--Comment String-->
    <![CDATA[Some CDATA String]]>
  </data>
</abc>
```

The file contains a Document Type Declaration that indicates that <abc> is the root element, and a declaration for the text entity that expands to "Some Text":

The root element abc contains a child element data, which contains five child PBDOM\_OBJECTs: two PBDOM\_ELEMENT objects, and PBDOM\_TEXT, PBDOM\_COMMENT, and PBDOM\_CDATA objects.

The first child\_data element contains a PBDOM\_TEXT with the string "Child Data Text". The second child\_data element contains no child PBDOM\_OBJECTs but it does contain a PBDOM\_ATTRIBUTE, An\_Attribute, that contains the value "Some Attribute Value".

This example creates a PBDOM\_DOCUMENT called pbdom\_doc from c:\pbdom\_doc\_1.xml, tests the content of pbdom\_doc, then saves the DOM tree contained within pbdom\_doc into a separate file, c:\pbdom\_doc\_2.xml. The input and output files should be identical.

```
PBDOM_Builder      pbdom_bldr
PBDOM_Document    pbdom_doc
PBDOM_Object       pbdom_obj_array[]
PBDOM_Element      pbdom_elem
integer iFileNum1
long l = 0

// Create a PBDOM_DOCUMENT from the XML file
pbdom_bldr = Create PBDOM_Builder
pbdom_doc = pbdom_bldr.BuildFromFile &
  ("c:\pbdom_doc_1.xml")

// Test the contents of the PBDOM_DOCUMENT
// First test the PBDOM_DOCTYPE in the document
MessageBox ("PBDOM_DOCTYPE GetName()", &
  pbdom_doc.GetDocType().GetName())
MessageBox ("PBDOM_DOCTYPE GetInternalSubset()", &
  pbdom_doc.GetDocType().GetInternalSubset())

// Test the root element
MessageBox ("PBDOM_DOC Root Element Name", &
  pbdom_doc.GetRootElement().GetName())

// test the root element's child element
MessageBox ("PBDOM_DOC <data> Element Name", &
  pbdom_doc.GetRootElement().GetChildElement &
  ("data").GetName())
```

```

// Collect all the child PBDOM_OBJECTS of the
// <data> element
pbdom_doc.GetRootElement().GetChildElement &
    ("data").GetContent(pbdom_obj_array)

// Display the class name, the name and the text contained
// within each PBDOM_OBJECT array item
for l = 1 to UpperBound(pbdom_obj_array)
    MessageBox ("Child Object " + string(l) + " Class",&
        pbdom_obj_array[l].GetObjectClassString())
    MessageBox ("Child Object " + string(l) + " Name",&
        pbdom_obj_array[l].GetName())
    MessageBox ("Child Object " + string(l) + " Text",&
        pbdom_obj_array[l].GetText())
next

// Retrieve and display the name and text value of the
// "An_Attribute" attribute from the <child_data> element
pbdom_elem = pbdom_obj_array[2]
MessageBox ("child_data Attribute name", &
    pbdom_elem.GetAttribute("An_Attribute").GetName())
MessageBox ("child_data Attribute value", &
    pbdom_elem.GetAttribute("An_Attribute").GetText())

// save the DOM Tree contained within pbdom_doc into
// a separate file "c:\pbdom_doc_2.xml"
pbdom_doc.SaveDocument ("c:\pbdom_doc_2.xml")

Destroy pbdom_bldr

CATCH (PBDOM_Exception except)
    MessageBox ("Exception Occurred", except.Text)
END TRY

```

## Usage

The input URL string can be a local file path.

The encoding specified in the XML export template determines the encoding of the document created using BuildFromFile.

## See also

[BuildFromDataStore](#)

[BuildFromString](#)

### 6.1.3 BuildFromString

#### Description

Builds a PBDOM\_DOCUMENT from a string.

#### Syntax

```
pbdom_builder_name.BuildFromString(string strXMLStream)
```

**Table 6.3:**

Argument	Description
pbdom_builder_name	The name of a PBDOM_BUILDER object
strXMLStream	A string containing XML

**Return value**

PBDOM\_DOCUMENT.

**Throws**

EXCEPTION\_INVALID\_ARGUMENT -- The input string is invalid. This can happen if it has not been initialized properly or is a null object reference.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- Insufficient memory was encountered while executing this method.

**Examples**

The following PowerShell code fragment demonstrates how to use the BuildFromString method with an input string. A string containing XML is passed to the BuildFromString method and the return value is assigned to a PBDOM\_DOCUMENT.

```
PBDOM_Builder pbdom_bldr
PBDOM_DOCUMENT pbdom_doc
string strXML

strXML = "<Music:abc xmlns:ZMusic="
strXML += "~\"http://www.ZMusic.com~\">"
strXML += "Root Element Data<data>ABC Data"
strXML += "<inner_data>My Inner Data</inner_data>"

strXML += "My Data</data></abc>"

pbdom_bldr = Create PBDOM_Builder
pbdom_doc = pbdom_bldr.BuildFromString (strXML)
```

**Usage**

The encoding specified in the XML export template determines the encoding of the document created using BuildFromString.

**See also**

[BuildFromDataStore](#)

[BuildFromFile](#)

**6.1.4 GetParseErrors****Description**

Obtains a list of parsing errors detected during document parsing.

**Syntax**

```
pbdom_builder_name.GetParseErrors(ref string strErrorMessageArray[])
```

**Table 6.4:**

Argument	Description
pbdom_builder_name	The name of a PBDOM_BUILDER object
strErrorMessageArray	An unbounded array of strings, each of which will be filled with a formatted string containing a parse error.

## Return value

Boolean.

Returns true if a list of parse errors has been retrieved and false otherwise. Also returns false if there are no parse errors.

## Throws

**EXCEPTION\_INVALID\_ARGUMENT** -- The input string array is invalid. This can happen if it has not been initialised properly or is a null object reference.

**EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE** -- Insufficient memory was encountered while executing this method.

## Examples

The code in this example attempts to create a PBDOM\_DOCUMENT based on the following XML:

```
<!DOCTYPE root
[
<!ELEMENT root ANY>
<!ELEMENT data (#PCDATA)>
<!ENTITY text "Some Text">
]
>
<root><abc/><def/></root>
```

This XML is well formed but is not valid, because the element root contains two child elements abc and def that are not declared in the DOCTYPE. When GetParseErrors is called, it returns the value true, indicating that at least one parse error has occurred, and generates the following list of errors:

```
"1,103,Unknown element 'abc'"
"1,109,Unknown element 'def'"
```

The 1 in both error messages indicates that the error occurred in line 1 of the XML string, and the 103 and 109 indicate columns 103 and 109, respectively.

```
PBDOM_BUILDER pbdom_buildr
PBDOM_DOCUMENT pbdom_doc
long l = 0
string strXML = "<!DOCTYPE root [<!ELEMENT root ANY><!ELEMENT data (#PCDATA)> <!
ENTITY text ~\"Some Text~\">]> <root><abc/><def/></root>"
string strParseErrors[]
BOOLEAN bRetTemp = FALSE

try
    pbdom_buildr = Create PBDOM_BUILDER
    pbdom_doc = pbdom_buildr.BuildFromString (strXML)
    bRetTemp = &
        pbdom_buildr.GetParseErrors(strParseErrors)

    if bRetTemp = true then
        for l = 1 to UpperBound(strParseErrors)
            MessageBox ("Parse Error", strParseErrors[l])
        next
    end if
catch (PBDOM_EXCEPTION pbdom_except)
    MessageBox ("PBDOM_EXCEPTION", &
        pbdom_except.GetMessage())
end try
```

## Usage

This method retrieves a list of errors detected during the last parse operation performed by this PBDOM\_BUILDER. Each string in the array has the following format:

```
[Line Number],[Column Number],[Error Message]
```

where Line Number and Column Number indicate the line number and column number in the XML document where the error was encountered. Error Message is the parse error message.

## 7 PBDOM\_CDATA Class

### About this chapter

This chapter describes the PBDOM\_CDATA class.

### 7.1 PBDOM\_CDATA

#### Description

The PBDOM\_CDATA class represents an XML DOM CDATA section. The PBDOM\_CDATA class is derived from PBDOM\_TEXT, which inherits from the PBDOM\_CHARACTERDATA class.

A PBDOM\_CDATA object is used to hold text that contains characters that are prohibited in text objects, such as "<" and "&", without using entity references. For example, consider the following PBDOM\_CDATA object:

```
<some_text>
  <![CDATA[ (x < y) & (y < z) => x < z ]]>
</some_text>
```

A PBDOM\_TEXT object with the same text content must be written like this:

```
<some_text>
  (x &lt; y) &amp; (y &lt; z) =&gt; x &lt; z
</some_text>
```

However, although the PBDOM\_CDATA class is derived from PBDOM\_TEXT, a PBDOM\_CDATA object cannot always be inserted in the same context as a PBDOM\_TEXT. For example, a PBDOM\_TEXT object can be added as a child of a PBDOM\_ATTRIBUTE, but a PBDOM\_CDATA object cannot.

#### Methods

Some of the inherited methods from PBDOM\_OBJECT serve no meaningful objective, and only default or trivial functionalities result. These are described in the following table:

**Table 7.1:**

Method	Always returns
AddContent	current PBDOM_CDATA
GetContent	false
GetName	a string "#cdata"
HasChildren	false
InsertContent	current PBDOM_CDATA
IsAncestorObjectOf	false
RemoveContent	false
SetContent	current PBDOM_CDATA
SetName	false

PBDOM\_CDATA has the following non-trivial methods:



[Append](#)[Clone](#)[Detach](#)[Equals](#)[GetObjectClass](#)[GetObjectClassString](#)[GetOwnerDocumentObject](#)[GetParentObject](#)[GetText](#)[GetTextNormalize](#)[GetTextTrim](#)[SetParentObject](#)[SetText](#)

### 7.1.1 Append

#### Description

Appends the input string or the input text data of the PBDOM\_CHARACTERDATA object to the text content that already exists within the current PBDOM\_CDATA object.

#### Syntax

```
pbdom_cdata_name.Append(string strAppend)
pbdom_cdata_name.Append(pbdom_characterdata pbdom_characterdata_ref)
```

**Table 7.2:**

Argument	Description
pbdom_cdata_name	The name of a PBDOM_CDATA
strAppend	The string you want appended to the existing text of the current PBDOM_CDATA object
pbdom_characterdata_ref	The referenced PBDOM_CHARACTERDATA object whose text data is to be appended to the existing text of the current PBDOM_CDATA object

#### Return value

PBDOM\_CHARACTERDATA.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the input PBDOM\_CHARACTERDATA is not a reference to an object derived from PBDOM\_CHARACTERDATA (applies to second syntax).

## 7.1.2 Clone

### Description

Creates and returns a clone of the current PBDOM\_CDATA.

### Syntax

```
pbdom_cdata_name.Clone(boolean bDeep)
```

**Table 7.3:**

Argument	Description
pbdom_cdata_name	The name of a PBDOM_CDATA.
bDeep	A boolean specifying whether a deep or shallow clone is returned. Values are true for a deep clone and false for a shallow clone. This argument is currently ignored.

### Return value

PBDOM\_OBJECT. The return value is a clone of the current PBDOM\_CDATA housed in a PBDOM\_OBJECT.

### Examples

This example tests the following characteristics of a cloned PBDOM\_CDATA object:

- The contents of an original and cloned PBDOM\_CDATA object are exactly the same
- A cloned PBDOM\_CDATA initially has no parent object
- A cloned PBDOM\_CDATA is initially contained within the same owner document as the original

```
PBDOM_BUILDER      pbdom_buildr
PBDOM_DOCUMENT    pbdom_doc
PBDOM_CDATA       pbdom_cdat
PBDOM_OBJECT      pbdom_obj_array[]
string strXML = "<!DOCTYPE root [<!ELEMENT root (#PCDATA)>]><root><![CDATA[This
is a CDATA Section.]]></root>"

try
  // Build a PBDOM_DOCUMENT based on strXML.
  pbdom_buildr = Create PBDOM_BUILDER
  pbdom_doc = pbdom_buildr.BuildFromString (strXML)

  // Get the contents of the root element.
  pbdom_doc.GetRootElement().GetContent(pbdom_obj_array)

  // Test if the root element contains only one child object.
  if (UpperBound(pbdom_obj_array) = 1) then
    MessageBox ("Pass", "Root Element has only one child.")
  else
    MessageBox ("Fail", "Root Element must have only one child.")
  end if

  // Make a clone of the only child of the root element.
  pbdom_cdat = pbdom_obj_array[1].Clone(true)
```

```

// Test if the clone is a PBDOM_CDATA object.
if (pbdom_cdat.GetObjectClassString() = "pbdom_cdata") then
  MessageBox ("Pass", &
    "The first child, after being cloned, is indeed a PBDOM_CDATA object.")
else
  MessageBox ("Fail", "The first child, after being cloned, " &
    + "is found to be a " + pbdom_cdat.GetObjectClassString() + " object.")
end if

// Test if the clone is a CDATA section.
if (pbdom_cdat.GetText() = "This is a CDATA Section.") then
  MessageBox ("Pass", "The text contents of the clone is correct.")
else
  MessageBox ("Fail", "The text contents of the clone is : [" &
    + pbdom_cdat.GetText() + "]. This is incorrect.")
end if

// Test that the clone has no parent.
if (Not IsValid(pbdom_cdat.GetParentObject())) then
  MessageBox ("Pass", "The clone has no parent.")
else
  MessageBox ("Fail", "The clone should have no parent.")
end if

// Test that the clone's owner document is the same
// as the original's owner document.
if (pbdom_cdat.GetOwnerDocumentObject() = pbdom_doc) then
  MessageBox ("Pass", "The clone's owner document is correct.")
else
  MessageBox ("Fail", "The clone's owner document is incorrect.")
end if

catch (PBDOM_EXCEPTION pbdom_except)
  MessageBox ("PBDOM_EXCEPTION", pbdom_except.GetMessage())
end try

```

## Usage

The Clone method creates a new PBDOM\_CDATA object that is a duplicate of, and a separate object from, the original. The clone of a PBDOM\_CDATA is always identical to its original whether deep or shallow cloning is invoked, because a PBDOM\_CDATA object does not contain any subtree of child PBDOM\_OBJECTs.

A PBDOM\_CDATA clone has no parent. However, the clone resides in the same PBDOM\_DOCUMENT as its original, and if the original PBDOM\_CDATA is standalone, the clone is standalone.

### 7.1.3 Detach

#### Description

Detaches a PBDOM\_CDATA from its parent PBDOM\_OBJECT.

#### Syntax

```
pbdom_cdata_name.Detach()
```

**Table 7.4:**

Argument	Description
pbdom_cdata_name	The name of a PBDOM_CDATA

**Return value**

PBDOM\_OBJECT. The current PBDOM\_CDATA detached from its parent.

**Usage**

If the current PBDOM\_CDATA object has no parent, no modifications occur.

**7.1.4 Equals****Description**

Tests for the equality of the current PBDOM\_CDATA and a referenced PBDOM\_OBJECT.

**Syntax**

```
pbdom_cdata_name.Equals(pbdom_object pbdom_object_ref)
```

**Table 7.5:**

Argument	Description
pbdom_cdata_name	The name of a PBDOM_CDATA
pbdom_object_ref	A PBDOM_OBJECT to test for equality with the current PBDOM_CDATA

**Return value**

Boolean.

Returns true if the current PBDOM\_CDATA object is equivalent to the referenced PBDOM\_OBJECT and false otherwise.

**Throws**

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the input PBDOM\_OBJECT is not a reference to an object derived from PBDOM\_OBJECT.

**Usage**

True is returned only if the referenced PBDOM\_OBJECT is also a derived PBDOM\_CDATA object and refers to the same DOM object as the current PBDOM\_CDATA. Two separately created PBDOM\_CDATA objects, for example, can contain exactly the same text but not be equal.

**7.1.5 GetObjectClass****Description**

Returns a long integer code that indicates the class of the current PBDOM\_OBJECT.

**Syntax**

```
pbdom_object_name.GetObjectClass()
```

**Table 7.6:**

Argument	Description
pbdom_object_name	The name of a PBDOM_OBJECT

**Return value**

Long.

GetObjectClass returns a long integer code that indicates the class of the current PBDOM\_OBJECT. If pbdom\_object\_name is a PBDOM\_CDATA object, the returned value is 8.

**See also**

[GetObjectClassString](#)

**7.1.6 GetObjectClassString****Description**

Returns a string form of the class of the PBDOM\_OBJECT.

**Syntax**

```
pbdom_object_name.GetObjectClassString()
```

**Table 7.7:**

Argument	Description
pbdom_object_name	The name of a PBDOM_OBJECT

**Return value**

String.

GetObjectClassString returns a string that indicates the class of the current PBDOM\_OBJECT. If pbdom\_object\_name is a PBDOM\_CDATA object, the returned string is "pbdom\_cdata".

**See also**

[GetObjectClass](#)

**7.1.7 GetOwnerDocumentObject****Description**

Returns the owning PBDOM\_DOCUMENT of the current PBDOM\_CDATA.

**Syntax**

```
pbdom_cdata_name.GetOwnerDocumentObject()
```

**Table 7.8:**

Argument	Description
pbdom_cdata_name	The name of a PBDOM_CDATA

**Return value**

PBDOM\_OBJECT.

**Usage**

If there is no owning PBDOM\_DOCUMENT, null is returned.

**See also**

[GetParentObject](#)

[SetParentObject](#)

### 7.1.8 GetParentObject

**Description**

Returns the parent PBDOM\_OBJECT of the PBDOM\_CDATA. If there is no parent, null is returned.

**Syntax**

```
pbdom_cdata_name.GetParentObject()
```

**Table 7.9:**

Argument	Description
pbdom_cdata_name	The name of a PBDOM_CDATA

**Return value**

PBDOM\_OBJECT.

**See also**

[GetOwnerDocumentObject](#)

[SetParentObject](#)

### 7.1.9 GetText

**Description**

Returns the text data that is contained within the current PBDOM\_CDATA object.

**Syntax**

```
pbdom_cdata_name.GetText()
```

**Table 7.10:**

Argument	Description
pbdom_cdata_name	The name of a PBDOM_CDATA

**Return value**

String.

The textual content of the current PBDOM\_CDATA object.

**See also**

[GetTextNormalize](#)

[GetTextTrim](#)

[SetText](#)

### 7.1.10 GetTextNormalize

#### Description

Returns the text data that is contained within the current PBDOM\_CDATA object, with all surrounding whitespace characters removed and internal whitespace characters normalized to a single space.

#### Syntax

```
pbdom_cdata_name.GetTextNormalize()
```

Table 7.11:

Argument	Description
pbdom_cdata_name	The name of a PBDOM_CDATA

#### Return value

String.

#### Usage

If no textual value exists for the current PBDOM\_OBJECT, or if only whitespace characters exist, an empty string is returned.

#### See also

[GetText](#)

[GetTextTrim](#)

[SetText](#)

### 7.1.11 GetTextTrim

#### Description

Returns the textual content of the current PBDOM\_CDATA object with all surrounding whitespace characters removed.

#### Syntax

```
pbdom_cdata_name.GetTextTrim()
```

Table 7.12:

Argument	Description
pbdom_cdata_name	The name of a PBDOM_CDATA

#### Return value

String.

#### Usage

If no textual value exists for the current PBDOM\_CDATA, or if only whitespace characters exist, an empty string is returned.

#### See also

[GetText](#)[GetTextNormalize](#)[SetText](#)

### 7.1.12 SetParentObject

#### Description

Sets the referenced PBDOM\_OBJECT to be the parent of the current PBDOM\_CDATA.

#### Syntax

```
pbdom_cdata_name.SetParentObject(pbdom_object pbdom_object_ref)
```

**Table 7.13:**

Argument	Description
pbdom_cdata_name	The name of a PBDOM_CDATA
pbdom_object_ref	A PBDOM_OBJECT to be set as the parent of this PBDOM_CDATA object

#### Return value

PBDOM\_OBJECT.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the input PBDOM\_OBJECT is not a reference to an object derived from PBDOM\_OBJECT.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT -- If the current PBDOM\_CDATA already has a parent.

EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT -- If the input PBDOM\_OBJECT is of a class that does not have a legal parent-child relationship with the PBDOM\_CDATA class.

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- If the input PBDOM\_OBJECT requires a user-defined name and it has not been named.

#### Usage

The PBDOM\_OBJECT that you set to be the parent of the current PBDOM\_CDATA must have a legal parent-child relationship. If it does not, an exception is thrown. Only a PBDOM\_ELEMENT object can be set as the parent of a PBDOM\_CDATA object.

#### See also

[GetParentObject](#)

### 7.1.13 SetText

#### Description

Sets the input string to be the text content of the current PBDOM\_CDATA object.

#### Syntax



```
pbdom_cdata_name.SetText(string strSet)
```

**Table 7.14:**

Argument	Description
pbdom_cdata_name	The name of a PBDOM_CDATA
strSet	The string you want set as the text of the PBDOM_CDATA

**Return value**

PBDOM\_CHARACTERDATA. This PBDOM\_CDATA modified and returned as a PBDOM\_CHARACTERDATA object.

**See also**

[GetText](#)

[GetTextNormalize](#)

[GetTextTrim](#)

## 8 PBDOM\_ENTITYREFERENCE Class

### About this chapter

This chapter describes the PBDOM\_ENTITYREFERENCE class.

### 8.1 PBDOM\_ENTITYREFERENCE

#### Description

The PBDOM\_ENTITYREFERENCE class defines behavior for an XML Entity reference Node. It allows you to insert entity references within element nodes as well as attribute nodes. The PBDOM\_ENTITYREFERENCE class is derived from PBDOM\_OBJECT.

#### Methods

Some of the inherited methods from PBDOM\_OBJECT currently serve no meaningful objective, and only default or trivial functionalities result. These are described in the following table:

**Table 8.1:**

Method	Always returns
AddContent	current PBDOM_ENTITYREFERENCE
GetContent	false
GetText	an empty string
GetTextNormalize	an empty string
GetTextTrim	an empty string
HasChildren	false
InsertContent	current PBDOM_ENTITYREFERENCE
IsAncestorObjectOf	false
RemoveContent	false
SetContent	current PBDOM_ENTITYREFERENCE

PBDOM\_ENTITYREFERENCE has the following non-trivial methods:

**Table 8.2:**

<a href="#">Clone</a>	<a href="#">GetName</a>	<a href="#">GetParentObject</a>
<a href="#">Detach</a>	<a href="#">GetObjectClass</a>	<a href="#">SetName</a>
<a href="#">Equals</a>	<a href="#">GetObjectClassString</a>	<a href="#">SetParentObject</a>
	<a href="#">GetOwnerDocumentObject</a>	

#### 8.1.1 Clone

##### Description

Creates and returns a clone of the current PBDOM\_ENTITYREFERENCE object.

##### Syntax

```
pbdom_entityref_name.Clone(boolean bDeep)
```

**Table 8.3:**

Argument	Description
pbdom_entityref_name	The name of a PBDOM_ENTITYREFERENCE object.
bDeep	A boolean specifying whether a deep or shallow clone is returned. Values are true for a deep clone and false for a shallow clone. This parameter is currently ignored.

**Return value**

PBDOM\_OBJECT. A clone of the current PBDOM\_ENTITYREFERENCE object housed in a PBDOM\_OBJECT.

**Examples**

This example creates a PBDOM\_DOCUMENT based on a string that contains an XML document, and creates a PBDOM\_ENTITYREFERENCE object to reference the ENTITY my\_er defined in the DOCTYPE. The DOCTYPE also indicates that the root element must contain zero or more child elements named child, and that each child can contain only parsed character data.

The FOR loop creates ten child elements and inserts a new clone of pbdom\_er into each child element. You must use a clone, because the same object cannot be inserted as a child of more than one parent:

```
PBDOM_BUILDER      pbdom_builder
PBDOM_DOCUMENT    pbdom_doc
PBDOM_ENTITYREFERENCE pbdom_er
string strXML = "<!DOCTYPE root [<!ELEMENT root (child)*><!ELEMENT child
(#PCDATA)><!ENTITY my_er ~"MY ENTITY~">]><root/>"
long l = 0

TRY
  pbdom_builder = Create PBDOM_BUILDER
  pbdom_doc = pbdom_builder.BuildFromString(strXML)
  pbdom_er = Create PBDOM_ENTITYREFERENCE
  pbdom_er.SetName("my_er")

// Create 10 child elements for the root element
for l = 1 to 10
  PBDOM_ELEMENT pbdom_elem_child

  pbdom_elem_child = Create PBDOM_ELEMENT
  pbdom_elem_child.SetName("child")
  // Add a clone of pbdom_er as content
  pbdom_elem_child.AddContent(pbdom_er.Clone(true))

  pbdom_doc.GetRootElement(). &
    AddContent(pbdom_elem_child)
next

pbdom_doc.SaveDocument("clone_er.xml")
CATCH(PBDOM_EXCEPTION pbdom_e)
  MessageBox ("PBDOM_EXCEPTION", pbdom_e.GetMessage())
END TRY
```

When the PBDOM\_DOCUMENT object is serialized, it produces the following XML document:

```
<!DOCTYPE root
[
<!ELEMENT root (child)*>
<!ELEMENT child (#PCDATA)*>
<!ENTITY my_er "MY ENTITY">
]
>
<root> <child>MY ENTITY</child>
<child>MY ENTITY</child>
<child>MY ENTITY</child>
<child>MY ENTITY</child>
<child>MY ENTITY</child>
<child>MY ENTITY</child>
<child>MY ENTITY</child>
<child>MY ENTITY</child>
<child>MY ENTITY</child>
<child>MY ENTITY</child>
<child>MY ENTITY</child>
</root>
```

## Usage

The Clone method creates a new PBDOM\_ENTITYREFERENCE object which is a duplicate of the original. A PBDOM\_ENTITYREFERENCE object cannot contain any child PBDOM\_OBJECTs, so there is no subtree beneath a PBDOM\_ENTITYREFERENCE object. A shallow clone is therefore structurally no different than a deep clone of a PBDOM\_ENTITYREFERENCE object.

This method allows you to use an entity reference node more than once. You cannot add a PBDOM\_ENTITYREFERENCE object as the child of more than one PBDOM\_OBJECT, but you can clone it and then add the clone as the child of another PBDOM\_OBJECT.

A PBDOM\_ENTITYREFERENCE clone does not have any parent. However, the clone resides in the same PBDOM\_DOCUMENT as its original. If the original PBDOM\_ENTITYREFERENCE object is standalone, the clone is also standalone.

### 8.1.2 Detach

#### Description

Detaches a PBDOM\_ENTITYREFERENCE object from its parent PBDOM\_OBJECT.

#### Syntax

```
pbdom_entityref_name.Detach()
```

**Table 8.4:**

Argument	Description
pbdom_entityref_name	The name of a PBDOM_ENTITYREFERENCE object

#### Return value

PBDOM\_OBJECT. The current PBDOM\_ENTITYREFERENCE object detached from its parent.

#### Usage

If the current PBDOM\_ENTITYREFERENCE object has no parent, no modifications occur.

### 8.1.3 Equals

#### Description

Tests for the equality of the current PBDOM\_ENTITYREFERENCE object and a referenced PBDOM\_OBJECT.

#### Syntax

```
pbdom_entityref_name.Equals(pbdom_object pbdom_object_ref)
```

Table 8.5:

Argument	Description
pbdom_entityref_name	The name of a PBDOM_ENTITYREFERENCE object

#### Return value

Boolean.

Returns true if the current PBDOM\_ENTITYREFERENCE object is equivalent to the input PBDOM\_OBJECT, and false otherwise.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the input PBDOM\_OBJECT is not an object derived from PBDOM\_OBJECT.

#### Usage

This method returns true only if the referenced PBDOM\_OBJECT is also a derived PBDOM\_ENTITYREFERENCE object and it refers to the same DOM object as the current PBDOM\_ENTITYREFERENCE object. Two separately created PBDOM\_COMMENTS, for example, can contain exactly the same text but not be equal.

### 8.1.4 GetName

#### Description

Obtains the name of the current PBDOM\_ENTITYREFERENCE object.

#### Syntax

```
pbdom_entityref_name.GetName()
```

Table 8.6:

Argument	Description
pbdom_entityref_name	The name of a PBDOM_ENTITYREFERENCE object

#### Return value

String.

#### See also

[SetName](#)

### 8.1.5 GetObjectClass

#### Description

Returns a long integer code that indicates the class of the current PBDOM\_OBJECT.

#### Syntax

```
pbdom_object_name.GetObjectClass()
```

**Table 8.7:**

Argument	Description
pbdom_object_name	The name of a PBDOM_OBJECT

#### Return value

Long.

A code that indicates the class of the current PBDOM\_OBJECT. If pbdom\_object\_name is a PBDOM\_ENTITYREFERENCE object, the returned value is 11.

#### See also

[GetObjectClassString](#)

### 8.1.6 GetObjectClassString

#### Description

Returns a string form of the class of the PBDOM\_OBJECT.

#### Syntax

```
pbdom_object_name.GetObjectClassString()
```

**Table 8.8:**

Argument	Description
pbdom_object_name	The name of a PBDOM_OBJECT

#### Return value

String.

A string that indicates the class of the current PBDOM\_OBJECT. If pbdom\_object\_name is a PBDOM\_ENTITYREFERENCE object, the returned string is "pbdom\_entityreference".

#### See also

[GetObjectClass](#)

### 8.1.7 GetOwnerDocumentObject

#### Description

The GetOwnerDocumentObject method returns the owning PBDOM\_DOCUMENT of the current PBDOM\_ENTITYREFERENCE object.

**Syntax**

```
pbdom_entityref_name.GetOwnerDocumentObject()
```

**Table 8.9:**

Argument	Description
pbdom_entityref_name	The name of a PBDOM_ENTITYREFERENCE object

**Return value**

PBDOM\_DOCUMENT.

**Usage**

If there is no owning PBDOM\_DOCUMENT, null is returned.

**See also**

[GetParentObject](#)

[SetParentObject](#)

**8.1.8 GetParentObject****Description**

The GetParentObject method returns the parent PBDOM\_OBJECT of the current PBDOM\_ENTITYREFERENCE object.

**Syntax**

```
pbdom_entityref_name.GetParentObject()
```

**Table 8.10:**

Argument	Description
pbdom_entityref_name	The name of a PBDOM_ENTITYREFERENCE object

**Return value**

PBDOM\_OBJECT.

**Usage**

The GetParentObject method returns the parent PBDOM\_OBJECT of the current PBDOM\_ENTITYREFERENCE object. If the PBDOM\_ENTITYREFERENCE object has no parent, null is returned.

**See also**

[GetOwnerDocumentObject](#)

[SetParentObject](#)

**8.1.9 SetName****Description**

Changes the name of the PBDOM\_ENTITYREFERENCE object, effectively making it refer to another DOM entity object.

### Syntax

```
pbdom_entityref_name.SetName(string strName)
```

**Table 8.11:**

Argument	Description
pbdom_entityref_name	The name of a PBDOM_ENTITYREFERENCE object
strName	The new name you want to set for the current PBDOM_ENTITYREFERENCE object

### Return value

Boolean.

Returns true if the name of the current PBDOM\_ENTITYREFERENCE object was changed, and false if it was not.

### See also

[GetName](#)

## 8.1.10 SetParentObject

### Description

The SetParentObject method sets the referenced PBDOM\_OBJECT to be the parent of the current PBDOM\_ENTITYREFERENCE object.

### Syntax

```
pbdom_entityref_name.SetParentObject(pbdom_object pbdom_object_ref)
```

**Table 8.12:**

Argument	Description
pbdom_entityref_name	The name of a PBDOM_ENTITYREFERENCE object
pbdom_object_ref	The PBDOM_OBJECT to be set as the parent of the current PBDOM_ENTITYREFERENCE object

### Return value

PBDOM\_OBJECT.

### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the input PBDOM\_OBJECT is not an object derived from PBDOM\_OBJECT.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT -- If the current PBDOM\_ENTITYREFERENCE object already has a parent.



**EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT** -- If the input PBDOM\_OBJECT is of a class that does not have a legal parent-child relationship with the PBDOM\_ENTITYREFERENCE class.

**EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT** -- If the input PBDOM\_OBJECT requires a user-defined name and it has not been named, or the name of the entity reference object has not been set.

### **Usage**

This method sets the input PBDOM\_OBJECT to be the parent of this PBDOM\_ENTITYREFERENCE object. The caller is responsible for ensuring that the current PBDOM\_ENTITYREFERENCE object and the input PBDOM\_OBJECT can have a legal parent-child relationship. Currently only a PBDOM\_ELEMENT or a PBDOM\_ATTRIBUTE can be set as the parent of a PBDOM\_ENTITYREFERENCE object.

### **See also**

[GetOwnerDocumentObject](#)

[GetParentObject](#)

## 9 PBDOM\_CHARACTERDATA Class

### About this document

This chapter describes the PBDOM\_CHARACTERDATA class.

### 9.1 PBDOM\_CHARACTERDATA

#### Description

The PBDOM\_CHARACTERDATA class represents character-based content (not markup) within an XML document. It extends the PBDOM\_OBJECT class with a set of methods specifically intended for manipulating character data in the DOM.

The PBDOM\_CHARACTERDATA class is the parent class of three other PBDOM classes:

- PBDOM\_TEXT
- PBDOM\_CDATA
- PBDOM\_COMMENT

The PBDOM\_CHARACTERDATA class, like its parent class PBDOM\_OBJECT, is a "virtual" class (similar to a virtual C++ class) in that it is not expected to be directly instantiated and used.

For example, in the following code, the attempt to set the text of pbdom\_chrdata raises an exception:

```
PBDOM_CHARACTERDATA pbdom_chrdata
pbdom_chrdata = CREATE PBDOM_CHARACTERDATA
pbdom_chrdata.SetText ("character string");//error
```

In this example, the attempt to set the text of pbdom\_chrdata succeeds because pbdom\_chrdata is declared as a PBDOM\_CHARACTERDATA but instantiated as a PBDOM\_TEXT:

```
PBDOM_CHARACTERDATA pbdom_chrdata
pbdom_chrdata = CREATE PBDOM_TEXT
pbdom_chrdata.SetText ("character string");//success
```

#### Methods

Some of the inherited methods from PBDOM\_OBJECT serve no meaningful objective and only default or trivial functionalities result. These are described in the following table:

**Table 9.1:**

Method	Always returns
AddContent	current PBDOM_CHARACTERDATA
GetContent	false
InsertContent	current PBDOM_CHARACTERDATA
RemoveContent	false

Method	Always returns
SetContent	current PBDOM_CHARACTERDATA
SetName	false

PBDOM\_CHARACTERDATA has the following non-trivial methods:

**Table 9.2:**

<a href="#">Append</a>	<a href="#">GetParentObject</a>
<a href="#">Clone</a>	<a href="#">GetText</a>
<a href="#">Detach</a>	<a href="#">GetTextNormalize</a>
<a href="#">Equals</a>	<a href="#">GetTextTrim</a>
<a href="#">GetName</a>	<a href="#">HasChildren</a>
<a href="#">GetObjectClass</a>	<a href="#">IsAncestorObjectOf</a>
<a href="#">GetObjectClassString</a>	<a href="#">SetParentObject</a>
<a href="#">GetOwnerDocumentObject</a>	<a href="#">SetText</a>

### 9.1.1 Append

#### Description

The Append method is overloaded:

- Syntax 1 appends an input string to the text content that already exists within the current PBDOM\_CHARACTERDATA object.
- Syntax 2 appends the text data of a PBDOM\_CHARACTERDATA object to the text content that already exists within the current PBDOM\_CHARACTERDATA object.

#### Syntax

**Table 9.3:**

For this syntax	See
Append(string strAppend)	<a href="#">Append Syntax 1</a>
Append(pbdom_characterdata pbdom_characterdata_ref)	<a href="#">Append Syntax 2</a>

#### 9.1.1.1 Append Syntax 1

#### Description

Appends an input string to the text content that already exists within the current PBDOM\_CHARACTERDATA object.

#### Syntax

```
pbdom_text_name.Append(string strAppend)
```

**Table 9.4:**

Argument	Description
pbdom_text_name	The name of a PBDOM_CHARACTERDATA object
strAppend	The string you want appended to the existing text of the current PBDOM_CHARACTERDATA object

**Return value**

PBDOM\_CHARACTERDATA. The current PBDOM\_CHARACTERDATA modified and returned as a PBDOM\_CHARACTERDATA object.

**Throws**

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If this PBDOM\_CHARACTERDATA object is not a reference to an object derived from PBDOM\_CHARACTERDATA.

**Examples**

In this example, the PowerScript code builds a PBDOM\_DOCUMENT based on the following DOM Tree:

```
<abc>
  <data>
    <child_1>
      My Text      </child_1>
    <child_2>
      <!--My Comment-->
    </child_2>
    <child_3>
      <![CDATA[My CDATA]]>
    </child_3>
  </data>
</abc>
```

The root element abc has a child element, data, that has three child elements. child\_1 contains a child PBDOM\_TEXT with the string "My Text". child\_2 contains a child PBDOM\_COMMENT with the string "My Comment". child\_3 contains a child PBDOM\_CDATA with the string "My CDATA".

In the following PowerScript code, the single statement that follows the comment // obtain the child PBDOM\_TEXT of child\_1 does the following:

1. Obtains the root element of the PBDOM\_DOCUMENT pbdom\_doc using GetRootElement. A new PBDOM\_ELEMENT representing the root element abc is created in memory and returned.
2. Calls the GetChildElement method on the returned root abc PBDOM\_ELEMENT using data as the parameter to single out the data child element. A PBDOM\_ELEMENT representing the data element is created in memory and returned.
3. Calls the GetChildElement on the returned data PBDOM\_ELEMENT, using child\_1 as the parameter to single out the child\_1 child element. A PBDOM\_ELEMENT representing the child\_1 element is created in memory and returned.

4. Calls the GetContent method on the returned child\_1 PBDOM\_ELEMENT, supplying a reference to the unbounded array pbdom\_chardata\_array.

You can supply PBDOM\_CHARACTERDATA array instead of a PBDOM\_OBJECT array because PBDOM\_CHARACTERDATA is a subclass of PBDOM\_OBJECT.

However, GetContent fails if child\_1 contains any objects other than PBDOM\_CHARACTERDATA objects.

Because child\_1 holds only the PBDOM\_TEXT containing the string "My Text", this statement returns an array that has only one array item. The next statement appends another string to the array item. The example then repeats these steps for child\_2 and child\_3 and saves pbdom\_doc to a file:

```
PBDOM_Builder      pbdombuilder_new
pbdom_document     pbdom_doc
PBDOM_CHARACTERDATA  pbdom_chardata_array[]

string strXML = "<abc><data><child_1>My Text</child_1><child_2><!--My Comment--></child_2><child_3><![CDATA[My CDATA]]></child_3></data></abc>"

TRY
    pbdombuilder_new = Create PBDOM_Builder
    pbdom_doc = pbdombuilder_new.BuildFromString (strXML)

// obtain the child PBDOM_TEXT of child_1
    pbdom_doc.GetRootElement().GetChildElement("data").&
        GetChildElement("child_1"). &
        GetContent(pbdom_chardata_array)

// append the string "Now Appended" to the text
// returned by the call to GetContent
    pbdom_chardata_array[1].Append (" Now Appended")

// repeat for child_2 and child_3
    pbdom_doc.GetRootElement().GetChildElement("data").&
        GetChildElement("child_2"). &
        GetContent(pbdom_chardata_array)
    pbdom_chardata_array[1].Append (" Now Appended")

    pbdom_doc.GetRootElement().GetChildElement("data").&
        GetChildElement("child_3"). &
        GetContent(pbdom_chardata_array)
    pbdom_chardata_array[1].Append (" Now Appended")

// save pbdom_doc to a file
    pbdom_doc.SaveDocument ("c:\pbdom_doc_1.xml")

    Destroy pbdombuilder_new

CATCH (PBDOM_Exception except)
    MessageBox ("Exception Occurred", except.Text)
END TRY
```

The saved file contains the following:

```
<abc>
  <data>
    <child_1>
      My Text Now Appended      </child_1>
    <child_2>
      <!--My Comment Now Appended-->
```

```

</child_2>
<child_3>
  <![CDATA[My CDATA Now Appended]]>
</child_3>
</data>
</abc>

```

### 9.1.1.2 Append Syntax 2

#### Description

Appends the text data of a PBDOM\_CHARACTERDATA object to the text content that already exists within the current PBDOM\_CHARACTERDATA object.

#### Syntax

```
pbdom_text_name.Append(pbdom_characterdata pbdom_characterdata_ref)
```

**Table 9.5:**

Argument	Description
pbdom_text_name	The name of a PBDOM_CHARACTERDATA
pbdom_characterdata_ref	The referenced PBDOM_CHARACTERDATA object whose text data is to be appended to the existing text of the current PBDOM_CHARACTERDATA object

#### Return value

PBDOM\_CHARACTERDATA. The current PBDOM\_CHARACTERDATA modified and returned as a PBDOM\_CHARACTERDATA object.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the current PBDOM\_CHARACTERDATA or the input PBDOM\_CHARACTERDATA is not a reference to an object derived from PBDOM\_CHARACTERDATA.

#### Usage

Note that JDOM does not define an Append method for its CHARACTERDATA class. Because PBDOM implements its Append method in the base PBDOM\_CHARACTERDATA class, a PBDOM\_TEXT object, a PBDOM\_CDATA object, and a PBDOM\_TEXT object can append their internal text data to each other because they are all PBDOM\_CHARACTERDATA-derived objects.

### 9.1.2 Clone

#### Description

Creates and returns a clone of the current PBDOM\_CHARACTERDATA.

#### Syntax

```
pbdom_chardata_name.Clone(boolean bDeep)
```

**Table 9.6:**

Argument	Description
pbdom_chardata_name	The name of a PBDOM_CHARACTERDATA.
bDeep	A boolean specifying whether a deep or shallow clone is returned. Values are true for a deep clone and false for a shallow clone. This argument is currently ignored.

**Return value**

PBDOM\_OBJECT.

**Throws**

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If this PBDOM\_CHARACTERDATA is not a reference to an object derived from PBDOM\_CHARACTERDATA.

**Examples**

This example creates a PBDOM\_DOCUMENT based on the following DOM tree:

```
<abc>
  <data>Data</data>
</abc>
```

The PowerScript code obtains the data element of the root element as a PBDOM\_ELEMENT and obtains an array of its children. The array has only one item, the PBDOM\_TEXT containing the string "data":

```
PBDOM_BUILDER pbdombuilder_new
PBDOM_DOCUMENT pbdom_doc
PBDOM_ELEMENT pbdom_elem
PBDOM_CHARACTERDATA pbdom_chardata_1
PBDOM_CHARACTERDATA pbdom_chardata_2
PBDOM_CHARACTERDATA pbdom_chardata_3
PBDOM_OBJECT pbdom_obj_array[]
string strXML = "<abc><data>Data</data></abc>"

TRY
  pbdombuilder_new = CREATE PBDOM_BUILDER
  pbdom_doc = pbdombuilder_new.BuildFromString (strXML)

// get the data element, store in pbdom_elem,
// and get an array of its children
pbdom_elem = pbdom_doc.GetRootElement(). &
  GetChildElement("data")
pbdom_elem.GetContent(pbdom_obj_array)
```

This PBDOM\_TEXT is assigned into a PBDOM\_CHARACTERDATA object, pbdom\_chardata\_1. Calling GetObjectClassString on pbdom\_chardata\_1 returns the class name of the actual object contained within it, pbdom\_text. Calling GetText on it returns the string Data:

```
pbdom_chardata_1 = pbdom_obj_array[1]
MessageBox ("Class", &
  pbdom_chardata_1.GetObjectClassString())
MessageBox ("Text", pbdom_chardata_1.GetText())
```

Calling Clone on pbdom\_chardata\_1 creates a new PBDOM\_CHARACTERDATA object. However, because the actual object referenced by pbdom\_chardata\_1 is a PBDOM\_TEXT, the clone is a PBDOM\_TEXT object.

Calling GetObjectClassString and GetText on the clone have the same result as for pbdom\_chardata\_1. The clone and the original object are separate objects and a call to Equals returns false:

```
pbdom_chardata_2 = pbdom_chardata_1.Clone(TRUE)
MessageBox ("Class", &
    pbdom_chardata_2.GetObjectClassString())
MessageBox ("Text", pbdom_chardata_2.GetText())
if (pbdom_chardata_1.Equals(pbdom_chardata_2)) then
    MessageBox ("Equals", &
        "pbdom_chardata_1 equals pbdom_chardata_2")
else
    MessageBox ("Equals", &
        "pbdom_chardata_1 NOT equals pbdom_chardata_2")
end if
```

However, a call to Equals returns true if the object being compared to pbdom\_chardata\_1 is a reference to pbdom\_chardata\_1:

```
pbdom_chardata_3 = pbdom_chardata_1
if (pbdom_chardata_1.Equals(pbdom_chardata_3)) then
    MessageBox ("Equals", &
        "pbdom_chardata_1 equals pbdom_chardata_3")
else
    MessageBox ("Equals", &
        "pbdom_chardata_1 NOT equals pbdom_chardata_3")
end if

DESTROY pbdombuilder_new

CATCH (PBDOM_Exception except)
    MessageBox ("Exception Occurred", except.Text)
END TRY
```

## Usage

The Clone method creates a new PBDOM\_CHARACTERDATA object which is a duplicate of, and a separate object from, the original. Calling Equals using these two objects returns false.

The clone of a PBDOM\_CHARACTERDATA object is always identical to its original whether bDeep is true or false, because a PBDOM\_CHARACTERDATA object contains no subtree of child PBDOM\_OBJECTs.

A PBDOM\_CHARACTERDATA clone has no parent, but it resides in the same PBDOM\_DOCUMENT as its original, and if the original PBDOM\_CHARACTERDATA is standalone, the clone is standalone.

### 9.1.3 Detach

#### Description

Detaches a PBDOM\_CHARACTERDATA object from its parent.

#### Syntax

```
pbdom_chardata_name.Detach()
```



**Table 9.7:**

Argument	Description
pbdom_chardata_name	The name of a PBDOM_CHARACTERDATA object

**Return value**

PBDOM\_OBJECT.

**Throws**

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If this PBDOM\_CHARACTERDATA is not a reference to an object derived from PBDOM\_CHARACTERDATA.

**Examples**

This example creates a PBDOM\_DOCUMENT based on the following DOM tree:

```
<abc>
  <data>Data</data>
</abc>
```

The PowerScript code obtains the root element, uses it to obtain the child element, and then obtains an array of the child element's own children. This array has a single item, the PBDOM\_TEXT object with the text Data. The array can be cast to a PBDOM\_CHARACTERDATA object because it does not contain any objects that are not derived from PBDOM\_CHARACTERDATA.

Calling Detach separates the PBDOM\_TEXT object from its parent PBDOM\_OBJECT, data.

```
PBDOM_Builder      pbdombuilder_new
pbdom_document     pbdom_doc
pbdom_document     pbdom_owner_doc
PBDOM_CHARACTERDATA pbdom_chardata
PBDOM_OBJECT       pbdom_obj_array[]
string strXML = "<abc><data>Data</data></abc>"

TRY
  pbdombuilder_new = Create PBDOM_Builder
  pbdom_doc = pbdombuilder_new.BuildFromString (strXML)

  pbdom_doc.GetRootElement(). &
    GetChildElement("data"). &
    GetContent(pbdom_obj_array)

  pbdom_chardata = pbdom_obj_array[1]
  pbdom_chardata.Detach()
  pbdom_doc.SaveDocument("c:\pbdom_doc_1.xml")
  Destroy pbdombuilder_new
CATCH (PBDOM_Exception except)
  MessageBox("Exception Occurred", except.Text)
END TRY
```

When the document is saved to a file, the file's contents are as follows, because the PBDOM\_TEXT object was removed from data:

```
<abc>
  <data/>
</abc>
```

## Usage

Nothing occurs if the PBDOM\_CHARACTERDATA object has no parent.

### 9.1.4 Equals

#### Description

Tests for the equality of the current PBDOM\_CHARACTERDATA and a referenced PBDOM\_OBJECT.

#### Syntax

```
pbdom_chardata_name.Equals(pbdom_object pbdom_object_ref)
```

**Table 9.8:**

Argument	Description
pbdom_chardata_name	The name of a PBDOM_CHARACTERDATA object
pbdom_object_ref	A reference to a PBDOM_OBJECT to test for equality with the current PBDOM_CHARACTERDATA object

#### Return value

Boolean.

Returns true if the current PBDOM\_CHARACTERDATA is equivalent to the input PBDOM\_OBJECT and false otherwise.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If this PBDOM\_CHARACTERDATA is not a reference to an object derived from PBDOM\_CHARACTERDATA.

#### Usage

True is returned only if the referenced PBDOM\_OBJECT is also a derived PBDOM\_CHARACTERDATA object and refers to the same DOM object as the current PBDOM\_CHARACTERDATA. Two separately created PBDOM\_COMMENTS, for example, can contain exactly the same text but are not equal.

#### See also

[Clone](#)

### 9.1.5 GetOwnerDocumentObject

#### Description

The GetOwnerDocumentObject method returns the owning PBDOM\_DOCUMENT of the current PBDOM\_CHARACTERDATA.

#### Syntax

```
pbdom_chardata_name.GetOwnerDocumentObject()
```

**Table 9.9:**

Argument	Description
pbdom_chardata_name	The name of a PBDOM_CHARACTERDATA object

**Return value**

PBDOM\_OBJECT.

**Throws**

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If this PBDOM\_CHARACTERDATA is not associated with a derived PBDOM\_CHARACTERDATA class.

**Examples**

1. This example creates a PBDOM\_DOCUMENT based on the following DOM tree:

```
<abc>
  <data>Data</data>
</abc>
```

The PowerScript code obtains the root element, uses it to obtain the child element, and then obtains an array of the child element's own children. This array has a single item, the PBDOM\_TEXT object with the text Data. The array can be cast to a PBDOM\_CHARACTERDATA object because it does not contain any objects that are not derived from PBDOM\_CHARACTERDATA,

The call to GetOwnerDocumentObject returns a PBDOM\_OBJECT, which is stored in a PBDOM\_DOCUMENT called pbdom\_owner\_doc. The call to Equals tests whether the owner document of the "Data" PBDOM\_TEXT and the main document, referenced using pbdom\_doc, refer to the same document.

```
PBDOM_Builder      pbdombuilder_new
pbdom_document    pbdom_doc
pbdom_document    pbdom_owner_doc
pbdom_element     pbdom_elem
PBDOM_CHARACTERDATA  pbdom_chardata
PBDOM_OBJECT      pbdom_obj_array[]
string strXML = "<abc><data>Data</data></abc>"

TRY
  pbdombuilder_new = Create PBDOM_Builder
  pbdom_doc = pbdombuilder_new.BuildFromString (strXML)

  pbdom_elem = pbdom_doc.GetRootElement(). &
    GetChildElement("data")
  pbdom_elem.GetContent(pbdom_obj_array)

  pbdom_chardata = pbdom_obj_array[1]

  pbdom_owner_doc = &
    pbdom_chardata.GetOwnerDocumentObject()

  if (pbdom_doc.Equals(pbdom_owner_doc)) then
    MessageBox ("Equals", &
      "pbdom_doc Equals pbdom_owner_doc")
  else
```

```

        MessageBox ("Equals", &
            "pbdom_doc Not Equals pbdom_owner_doc")
    end if

    Destroy pbdombuilder_new

CATCH (PBDOM_Exception except)
    MessageBox ("Exception Occurred", except.Text)
END TRY

```

2. This example creates a PBDOM\_DOCUMENT based on the same DOM tree as example 1. It creates a PBDOM\_TEXT, stores it in the PBDOM\_CHARACTERDATA variable pbdom\_chardata, and assigns it some text. Objects created in this way are standalone objects -- they have no owner document or parent. Calling GetOwnerDocumentObject on pbdom\_chardata returns null.

The code then adds pbdom\_chardata as a child to the data element. This implicitly imports pbdom\_chardata into the original document. pbdom\_chardata now has an owner document and a parent (the data element). Calling GetOwnerDocumentObject on pbdom\_chardata returns the original document. When the returned PBDOM\_DOCUMENT has been assigned into pbdom\_owner\_doc, a call to Equals to compare pbdom\_doc with pbdom\_owner\_doc returns true:

```

PBDOM_Builder          pbdombuilder_new
pbdom_document         pbdom_doc
pbdom_document         pbdom_owner_doc
PBDOM_CHARACTERDATA   pbdom_chardata
string strXML = "<abc><data>Data</data></abc>"

TRY
    pbdombuilder_new = Create PBDOM_Builder
    pbdom_doc = pbdombuilder_new.BuildFromString (strXML)

    pbdom_chardata = Create PBDOM_TEXT
    pbdom_chardata.SetText(" Some Text")

    if (IsValid (pbdom_chardata.GetOwnerDocumentObject())) then
        MessageBox ("Owner Document", &
            "PBDOM_TEXT (~'Some Text~') has an owner document.")
    else
        MessageBox ("Owner Document", &
            "PBDOM_TEXT (~'Some Text~') has NO owner document.")
    end if

    pbdom_doc.GetRootElement().GetChildElement("data"). &
        AddContent(pbdom_chardata)

    pbdom_owner_doc = pbdom_chardata.GetOwnerDocumentObject()

    if (pbdom_doc.Equals(pbdom_owner_doc)) then
        MessageBox ("Equals", "pbdom_doc Equals pbdom_owner_doc")
    else
        MessageBox ("Equals", "pbdom_doc Not Equals pbdom_owner_doc")
    end if

    Destroy pbdombuilder_new
    Destroy pbdom_chardata

CATCH (PBDOM_Exception except)
    MessageBox ("Exception Occurred", except.Text)
END TRY

```

## Usage

If there is no owning PBDOM\_DOCUMENT, null is returned.

## See also

[GetParentObject](#)

[SetParentObject](#)

## 9.1.6 GetName

### Description

The GetName method allows you to obtain the name of the current PBDOM\_CHARACTERDATA.

### Syntax

```
pbdom_chardata_name.GetName()
```

**Table 9.10:**

Argument	Description
pbdom_chardata_name	The name of a PBDOM_CHARACTERDATA object

### Return value

String.

### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If this PBDOM\_CHARACTERDATA is not a reference to an object derived from PBDOM\_CHARACTERDATA.

### Usage

The returned string depends on the specific type of DOM object that is contained within PBDOM\_CHARACTERDATA.

---

### Note

A PBDOM\_CHARACTERDATA is abstract and is not to be instantiated into an object of its own. Thus, there is no name returned as "#characterdata".

---

The following table lists the return values based on the type of DOM Object contained within PBDOM\_CHARACTERDATA.

**Table 9.11:**

DOM Object	Return Value
PBDOM_CDATA	"#cdata-section"
PBDOM_COMMENT	"#comment"

DOM Object	Return Value
PBDOM_TEXT	"#text"

### 9.1.7 GetObjectClass

#### Description

The GetObjectClass method returns a long integer code that indicates the class of the current PBDOM\_OBJECT.

#### Syntax

```
pbdom_object_name.GetObjectClass()
```

**Table 9.12:**

Argument	Description
pbdom_object_name	The name of a PBDOM_OBJECT

#### Return value

Long.

GetObjectClass returns a long integer value that indicates the class of the current PBDOM\_OBJECT.

The possible return values for classes inherited from PBDOM\_CHARACTERDATA are:

- 7 for PBDOM\_TEXT
- 8 for PBDOM\_CDATA
- 9 for PBDOM\_COMMENT

The PBDOM\_CHARACTERDATA class itself cannot be instantiated, so the class ID 6, for PBDOM\_CHARACTERDATA, is never returned.

#### See also

[GetObjectClassString](#)

### 9.1.8 GetObjectClassString

#### Description

The GetObjectClassString method returns a string form of the class of the PBDOM\_OBJECT.

#### Syntax

```
pbdom_object_name.GetObjectClassString()
```

**Table 9.13:**

Argument	Description
pbdom_object_name	The name of a PBDOM_OBJECT

## Return value

String.

GetObjectClassString returns a string that indicates the class of the current PBDOM\_OBJECT.

The possible return values for classes inherited from PBDOM\_CHARACTERDATA are:

- pbdom\_text
- pbdom\_cdata
- pbdom\_comment

The PBDOM\_CHARACTERDATA class itself cannot be instantiated, so the string "pbdom\_characterdata" is never returned.

## Examples

This example creates a PBDOM\_DOCUMENT based on the following DOM tree:

```
<abc>
  <data>
    Data with a &lt; character      <!-- Comment with a &lt; character -->
    <![CDATA[ CDATA with an actual > character and
      an entity reference &lt; ]]>
  </data>
</abc>
```

The PowerScript code obtains the root element, uses it to obtain the child element, and then obtains an array of the child element's own children. This is an array of three PBDOM\_OBJECTs, each of which is a child node of data. This array provides the ability to access and manipulate the child nodes, but to illustrate the virtual nature of the PBDOM\_CHARACTERDATA class and the calling of methods of the PBDOM\_CHARACTERDATA class, the example defines an array of PBDOM\_CHARACTERDATA objects.

Each array item of the pbdom\_obj\_array is assigned to the pbdom\_chardata array, so you can call the methods of each array item without needing to know what subclass the item belongs to.

---

### Children must be subclasses of PBDOM\_CHARACTERDATA

If the data element contained a child that was not a subclass of PBDOM\_CHARACTERDATA, the FOR loop to assign each pbdom\_obj\_array item to a corresponding pbdom\_chardata array item would fail when it reached that item.

---

The MessageBox calls illustrate how the entity reference &lt; is handled by the different PBDOM\_CHARACTERDATA subclasses. In the PBDOM\_TEXT object, it is expanded. In the PBDOM\_COMMENT and PBDOM\_CDATA objects, it is not. The character to which the entity reference refers, ">", can also be included in a PBDOM\_CDATA object.

```
PBDOM_Builder      pbdombuilder_new
pbdom_document    pbdom_doc
pbdom_element     pbdom_elem
PBDOM_CHARACTERDATA pbdom_chardata[ ]
```

```

PBDOM_OBJECT      pbdom_obj_array[]
long l = 0
string strXML = "<abc><data>Data with a &lt; character<!-- Comment with a &lt;
character --><![CDATA[ CDATA with an actual > character and an entity reference
&lt; ]]></data></abc>"

TRY
  pbdombuilder_new = Create PBDOM_Builder
  pbdom_doc = pbdombuilder_new.BuildFromString (strXML)

  pbdom_elem = pbdom_doc.GetRootElement(). &
    GetChildElement("data")
  pbdom_elem.GetContent(pbdom_obj_array)

// populate an array of PBDOM_CHARACTERDATA  objects
for l = 1 to UpperBound(pbdom_obj_array)
  pbdom_chardata[l] = pbdom_obj_array[l]
next
for l = 1 to UpperBound(pbdom_chardata)
  MessageBox ("Class", &
    pbdom_chardata[l].GetObjectClassString())
  MessageBox ("Text", pbdom_chardata[l].GetText())
next

  Destroy pbdombuilder_new

CATCH (PBDOM_Exception except)
  MessageBox ("Exception Occurred", except.Text)
END TRY

```

**See also**[GetObjectClass](#)**9.1.9 GetParentObject****Description**

The GetParentObject method returns the parent PBDOM\_OBJECT of the current PBDOM\_CHARACTERDATA.

**Syntax**

```
pbdom_chardata_name.GetParentObject()
```

**Table 9.14:**

Argument	Description
pbdom_chardata_name	The name of a PBDOM_CHARACTERDATA object

**Return value**

PBDOM\_OBJECT.

**Throws**

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If this PBDOM\_CHARACTERDATA is not a reference to an object derived from PBDOM\_CHARACTERDATA.

**Examples**



This example creates a PBDOM\_DOCUMENT based on the following DOM tree and demonstrates how a PBDOM\_CHARACTERDATA INSTANCE can be detached from its parent:

```
<abc>
  <data>Data</data>
</abc>
```

The PowerScript code obtains the root element, uses it to obtain the child element, and then obtains an array of the child element's own children. This array has a single item, the PBDOM\_TEXT object with the text Data. The array can be cast to a PBDOM\_CHARACTERDATA object, because it does not contain any objects that are not derived from PBDOM\_CHARACTERDATA.

The parent of pbdom\_chardata\_1 is the data element. The following steps detach it from its parent:

- Create a PBDOM\_COMMENT in the PBDOM\_CHARACTERDATA object pbdom\_chardata\_2 and assign to it the text "Some Comments".
- Set pbdom\_chardata\_2 as an array item of pbdom\_obj\_array.
- Call SetContent on the parent of pbdom\_chardata\_1 (the data element).

Calling SetContent resets the contents of data, which can cause its original contents (including pbdom\_chardata\_1) to be removed, depending on what is stored inside pbdom\_obj\_array. Because pbdom\_obj\_array contains only the newly created PBDOM\_COMMENT, pbdom\_chardata\_2, data will have only this PBDOM\_COMMENT as its child.

pbdom\_chardata\_1 will have no parent, because it has been silently detached from it. Calling GetParentObject on it will return null:

```
PBDOM_Builder      pbdombuilder_new
pbdom_document    pbdom_doc
pbdom_document    pbdom_owner_doc
PBDOM_CHARACTERDATA  pbdom_chardata_1
PBDOM_CHARACTERDATA  pbdom_chardata_2
PBDOM_OBJECT      pbdom_obj_array[]
string strXML = "<abc><data>Data</data></abc>"

TRY
  pbdombuilder_new = Create PBDOM_Builder
  pbdom_doc = pbdombuilder_new.BuildFromString (strXML)

  pbdom_doc.GetRootElement(). &
    GetChildElement("data"). &
    GetContent(pbdom_obj_array)

  pbdom_chardata_1 = pbdom_obj_array[1]

  pbdom_chardata_2 = Create PBDOM_COMMENT
  pbdom_chardata_2.SetText ("Some Comments")

  pbdom_obj_array[1] = pbdom_chardata_2

  pbdom_chardata_1.GetParentObject(). &
    SetContent(pbdom_obj_array)
```

```

if (IsValid(pbdom_chardata_1.GetParentObject())) then
    MessageBox ("Has Parent Object", &
        "PBDOMTEXT (~'Data~') has a parent")
else
    MessageBox ("Has Parent Object", &
        "PBDOMTEXT (~'Data~') has NO parent")
end if

pbdom_doc.SaveDocument("c:\pbdom_doc_1.xml")

Destroy pbdombuilder_new
Destroy pbdom_chardata_2

CATCH (PBDOM_Exception except)
    MessageBox ("Exception Occurred", except.Text)
END TRY

```

When the resulting PBDOM\_DOCUMENT is saved to a file, it looks like this:

```

<abc>
  <data>
    <!-- Some Comments -->
  </data>
</abc>

```

## Usage

The parent is also an object derived from PBDOM\_CHARACTERDATA. If the PBDOM\_OBJECT has no parent, null is returned.

## See also

[SetParentObject](#)

### 9.1.10 GetText

#### Description

Calling the GetText method allows you to obtain text data that is contained within the current PBDOM\_CHARACTERDATA.

#### Syntax

```
pbdom_chardata_name.GetText()
```

**Table 9.15:**

Argument	Description
pbdom_chardata_name	The name of a PBDOM_CHARACTERDATA object

#### Return value

String.

The text of the current PBDOM\_CHARACTERDATA-derived object.

#### Throws

Throws EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If this PBDOM\_CHARACTERDATA is not a reference to an object derived from PBDOM\_CHARACTERDATA.

## Usage

The following table lists the return values based on the type of DOM Object contained within PBDOM\_CHARACTERDATA.

**Table 9.16:**

DOM Object	Return Value
PBDOM_TEXT	<p>The text data contained within the PBDOM_TEXT object itself.</p> <p>For example, suppose you have the following element:</p> <pre>&lt;abc&gt;MY TEXT&lt;/abc&gt;</pre> <p>If you have a PBDOM_TEXT object to represent the TEXT NODE "MY TEXT", then calling GetText on the PBDOM_TEXT returns the string MY TEXT.</p>
PBDOM_CDATA	<p>The string data that is contained within the CDATA section itself. For example, suppose you have the following CDATA:</p> <pre>&lt;![CDATA[ They're saying "x &lt; y" &amp; that "z &gt; y" so I guess that means that z &gt; x ]]&gt;</pre> <p>If there is a PBDOM_CDATA to represent the above CDATA section, then calling GetText returns the string:</p> <pre>They're saying "x &lt; y" &amp; that "z &gt; y" so I guess that means that z &gt; x</pre>
PBDOM_COMMENT	<p>The comment itself. For example, suppose you have the following comment:</p> <pre>&lt;!--This is a comment. --&gt;</pre> <p>Calling GetText on the comment returns the string:</p> <pre>This is a comment.</pre>

### See also

[GetTextNormalize](#)

[GetTextTrim](#)

[SetText](#)

### 9.1.11 GetTextNormalize

#### Description

The GetTextNormalize method allows you to obtain the text data that is contained within the current PBDOM\_CHARACTERDATA object, with all surrounding whitespace characters removed and internal whitespace characters normalized to a single space.

**Syntax**

```
pbdom_chardata_name.GetTextNormalize()
```

**Table 9.17:**

Argument	Description
pbdom_chardata_name	The name of a PBDOM_CHARACTERDATA object

**Return value**

String.

The following table lists the return values, based on the type of DOM object contained within PBDOM\_CHARACTERDATA.

**Table 9.18:**

DOM Object	Return Value
PBDOM_TEXT	<p>Suppose you have the following element:</p> <pre>&lt;abc&gt; MY TEXT &lt;/abc&gt;</pre> <p>If there is a PBDOM_TEXT object to represent the TEXT NODE "MY TEXT", then calling GetTextNormalize on the PBDOM_TEXT returns the string MY TEXT.</p>
PBDOM_CDATA	<p>Suppose there is the following CDATA:</p> <pre>&lt;![CDATA] They're saying "x &lt; y" &amp; that "z &gt; y" so I guess that means that z &gt; x ]]&gt;</pre> <p>If there is a PBDOM_CDATA to represent the above CDATA section, then calling GetTextNormalize on it returns the string:</p> <pre>They're saying " x &lt; y " &amp; that "z &gt; y" so I guess that means that z &gt; x</pre> <p>Note that the initial spaces before "They're" and the trailing space after the last "x" are removed. Additionally, the spaces between the words "guess" and "that" are reduced to just one space.</p>
PBDOM_COMMENT	<p>Suppose there is the following comment:</p> <pre>&lt;!--This is a comment --&gt;</pre>

DOM Object	Return Value
	Calling GetTextNormalize on this comment returns:  <div style="border: 1px solid gray; padding: 2px; background-color: #f0f0f0;">This is a comment</div>

### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If this PBDOM\_CHARACTERDATA is not a reference to an object derived from PBDOM\_CHARACTERDATA.

### Examples

This example demonstrates:

1. Using an external general parsed entity.
2. Using a single line statement to obtain the children PBDOM\_OBJECTs of an element.
3. Obtaining the text of the three separate types of PBDOM\_CHARACTERDATA objects : PBDOM\_TEXT, PBDOM\_COMMENT, and PBDOM\_CDATA.
4. Obtaining the normalized text of the same three separate types of PBDOM\_CHARACTERDATA objects.
5. The difference between the two types of text retrieved in 3 and 4.

Suppose the file C:\entity\_text.txt contains the following string:

```
&#9;&#32;Some&#32;External&#32;&#32;&#9;&#32;Text&#32;&#9;
```

The example creates a PBDOM\_DOCUMENT pbdom\_doc based on the following DOM tree, which is in the file C:\inputfile.txt:

```
<!DOCTYPE abc [<!ENTITY text1 SYSTEM "c:\entity_text.txt" >]>
<abc>
  <data>
    &text1;
    <!-- &text1;-->
    <![CDATA[&text1;]]>
  </data>
</abc>
```

The Document Type Declaration defines an external general parsed entity text1.

The example obtains the root element, uses it to obtain the data child element, and then obtains an array of the child element's own children. PBDOM collects all the PBDOM\_OBJECTs that are the children of data and stores them in the PBDOM\_OBJECT array pbdom\_obj\_array.

Next, the FOR loop iterates through all the items in pbdom\_obj\_array and stores each item in the PBDOM\_CHARACTERDATA array pbdom\_chardata. This step is not required -- the pbdom\_obj\_array can be used to manipulate the data element's children. It is done to demonstrate that you can cast each item into a PBDOM\_CHARACTERDATA object by assigning it into a PBDOM\_CHARACTERDATA array. This is possible if and only

if each PBDOM\_OBJECT is also derived from PBDOM\_CHARACTERDATA. If a PBDOM\_OBJECT is not derived from PBDOM\_CHARACTERDATA, the PowerBuilder VM throws an exception.

The next FOR loop iterates through all the items of the pbdom\_chardata array and calls the GetText and GetTextNormalize methods on each. Each of the returned strings from GetText and GetTextNormalize is delimited by "[" and "]" characters so that the complete text content displays clearly in the message boxes.

The first child of data is the PBDOM\_TEXT &text1;, which has been declared as an external general parsed entity whose content is the content of the file c:\entity\_text.txt. The &text1; entity reference and the entity references it contains are expanded by the parser. The call to GetTextNormalize strips away the whitespace characters.

The second child of data is the PBDOM\_COMMENT <!-- &text1;--> and the third child is the PBDOM\_CDATA <![CDATA[&text1;]]>. Entity references within comments and CDATA sections are never expanded. Both GetText and GetTextNormalize return &text1;.

```
PBDOM_Builder      pbdombuilder_new
pbdom_document    pbdom_doc
PBDOM_CHARACTERDATA  pbdom_chardata[]
PBDOM_OBJECT      pbdom_obj_array[]
integer           iFileNum1
long              l = 0

TRY
  pbdombuilder_new = Create PBDOM_Builder
  pbdom_doc = pbdombuilder_new.BuildFromFile &
    ("C:\inputfile.txt")

  pbdom_doc.GetRootElement(). &
    GetChildElement("data"). &
    GetContent(pbdom_obj_array)

  for l = 1 to UpperBound(pbdom_obj_array)
    pbdom_chardata[l] = pbdom_obj_array[l]
  next

  for l = 1 to UpperBound(pbdom_chardata)
    MessageBox(pbdom_chardata[l]. &
      GetObjectClassString() + "GetText()", &
      "[" + pbdom_chardata[l].GetText() + "]")
    MessageBox (pbdom_chardata[l]. &
      GetObjectClassString() + " GetTextNormalize()", &
      "[" + pbdom_chardata[l].GetTextNormalize() + "]")
  next

  Destroy pbdombuilder_new

CATCH (PBDOM_Exception except)
  MessageBox ("Exception Occurred", except.Text)
END TRY
```

## Usage

If no textual value exists for the current PBDOM\_OBJECT, or if only whitespace characters exist, an empty string is returned.

## See also

[GetText](#)

[GetTextTrim](#)[SetText](#)

### 9.1.12 GetTextTrim

#### Description

The GetTextTrim method returns the textual content of the current PBDOM\_CHARACTERDATA object with all surrounding whitespace characters removed.

#### Syntax

```
pbdom_chardata_name.GetTextTrim()
```

**Table 9.19:**

Argument	Description
pbdom_chardata_name	The name of a PBDOM_CHARACTERDATA

#### Return value

String.

**Table 9.20:**

DOM Object	Return Value
PBDOM_TEXT	<p>The text data contained within the PBDOM_TEXT object itself with surrounding whitespace characters removed.</p> <p>For example, suppose there is the following element:</p> <pre>&lt;abc&gt; MY TEXT &lt;/abc&gt;</pre> <p>If there is a PBDOM_TEXT object to represent the TEXT NODE "MY TEXT", then calling GetTextTrim on the PBDOM_TEXT returns the string MY TEXT.</p>
PBDOM_CDATA	<p>The string data that is contained within the CDATA section itself with surrounding whitespace characters removed. For example, suppose there is the following CDATA:</p> <pre>&lt;![CDATA[ They're saying "x &lt; y" &amp; that "z &gt; y" so I guess that means that z &gt; x ]]&gt;</pre> <p>If there is a PBDOM_CDATA to represent the above CDATA section, then calling GetTextTrim on it returns the string:</p> <pre>They're saying " x &lt; y " &amp; that "z &gt; y" so I</pre>

DOM Object	Return Value
	<pre>guess      that means that z &gt; x</pre> <p>Note that the initial spaces before "They're" and the trailing space after the last "x" are removed.</p>
PBDOM_COMMENT	<p>Suppose there is the following comment:</p> <pre>&lt;!-- This is a comment --&gt;</pre> <p>Calling GetTextTrim on this comment returns:</p> <pre>This is a comment</pre> <p>Note that the spaces between the individual words in the comment are preserved. Only the surrounding whitespace characters are removed.</p>

### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If this PBDOM\_CHARACTERDATA is not a reference to an object derived from PBDOM\_CHARACTERDATA.

### Examples

This example demonstrates:

1. Using an External DTD.
2. Using a parameter entity.
3. Using a single line statement to obtain the children PBDOM\_OBJECTs of an element.
4. Obtaining the text of the three separate types of PBDOM\_CHARACTERDATA objects : PBDOM\_TEXT, PBDOM\_COMMENT, and PBDOM\_CDATA.
5. Obtaining the trimmed text of the same three separate types of PBDOM\_CHARACTERDATA objects.
6. The difference between the two types of text retrieved in 4 and 5.

The PowerScript code saves a string into an external file, then creates a PBDOM\_DOCUMENT pbdom\_doc based on the following DOM tree:

```
<!DOCTYPE abc SYSTEM "c:\external_entity.dtd">
<abc>
  <data>
    &text1;
    <!-- &text1;-->
    <![CDATA[&text1;]]>
  </data>
</abc>
```



c:\external\_entity.dtd is an external Document Type Definition file. Its contents are the external subset of the Document Type Definition. The first line declares a PARAMETER entity param\_entity\_ref that contains the following replacement text:

```
&#32;&#32;&#32;PARAMETER ENTITY REFERENCE&#9;&#9;&#9;
```

The next line declares a general entity text1 that contains the following replacement text:

```
%param_entity_ref;
```

When the entity text1 is used in an XML document, it is expanded to the contents of the PARAMETER entity param\_entity\_ref.

The PowerScript code then obtains the root element, uses it to obtain the data child element, and then obtains an array of the child element's own children. PBDOM collects all the PBDOM\_OBJECTs that are the children of data and stores them in the PBDOM\_OBJECT array pbdom\_obj\_array.

Next, the FOR loop iterates through all the items in pbdom\_obj\_array and stores each item in the PBDOM\_CHARACTERDATA array pbdom\_chardata. This step is not required -- the pbdom\_obj\_array can be used to manipulate the data element's children. It is done to demonstrate that you can cast each item into a PBDOM\_CHARACTERDATA object by assigning it into a PBDOM\_CHARACTERDATA array.

This is possible if and only if each PBDOM\_OBJECT is also derived from PBDOM\_CHARACTERDATA. If a PBDOM\_OBJECT is not derived from PBDOM\_CHARACTERDATA, the PowerBuilder VM throws an exception.

The next FOR loop iterates through all the items of the pbdom\_chardata array and calls the GetText and GetTextTrim methods on each. Each of the returned strings from GetText and GetTextTrim is delimited by "[" and "]" characters so that the complete text content displays clearly in the message boxes.

The first child of data is the PBDOM\_TEXT &text1;, which expands to the string in param\_entity\_ref. The entity references within this string are also expanded and the Tab and Space characters display when GetText is called. When GetTextTrim is called, PBDOM removes the beginning and trailing whitespace characters and the resulting string is simply PARAMETER ENTITY REFERENCE.

The second child of data is the PBDOM\_COMMENT <!-- &text1;-->., and the third child is the PBDOM\_CDATA <![CDATA[&text1;]]>. The string &text1; is not considered to be an entity reference by PBDOM because W3C DOM comments and CDATA sections cannot hold any entity references. Both GetText and GetTextTrim return the string &text1;. There are no leading or trailing spaces to remove.

```
PBDOM_CHARACTERDATA    pbdom_chardata[]
PBDOM_OBJECT           pbdom_obj_array[]
integer                 iFileNum1
long                    l = 0
string strExternalDTD = "<!ENTITY % param_entity_ref ~"&#32;&#32;&#32;PARAMETER
ENTITY REFERENCE&#9;&#9;&#9;~"><!ENTITY text1 ~"%param_entity_ref;~">"
string strXML = "<!DOCTYPE abc SYSTEM ~"c:
\external_entity.dtd~"><abc><data>&text1;<!-- &text1;--><![CDATA[&text1;]]></
data></abc>"
```

```
TRY
```

```
    iFileNum1 = FileOpen("c:\external_entity.dtd", &
```

```

    StreamMode!, Write!, LockWrite!, Replace!)
FileWrite(iFileNum1, strExternalDTD)
FileClose(iFileNum1)

pbdombuilder_new = Create PBDOM_Builder
pbdom_doc = pbdombuilder_new.BuildFromString (strXML)

pbdom_doc.GetRootElement(). &
    GetChildElement("data"). &
    GetContent(pbdom_obj_array)

for l = 1 to UpperBound(pbdom_obj_array)
    pbdom_chardata[l] = pbdom_obj_array[l]
next

for l = 1 to UpperBound(pbdom_chardata)
    MsgBox (pbdom_chardata[l]. &
        GetObjectClassString() + " GetText()", &
        "[" + pbdom_chardata[l].GetText() + "]")
    MsgBox (pbdom_chardata[l]. &
        GetObjectClassString() + " GetTextTrim()" , &
        "[" + pbdom_chardata[l].GetTextTrim() + "]")
next

Destroy pbdombuilder_new

CATCH (PBDOM_Exception except)
    MsgBox ("Exception Occurred", except.Text)
END TRY

```

## Usage

If no textual value exists for the current PBDOM\_CHARACTERDATA, or if only whitespace characters exist, an empty string is returned.

## See also

[GetText](#)

[GetTextNormalize](#)

[SetText](#)

## 9.1.13 HasChildren

### Description

This method returns true if this PBDOM\_CHARACTERDATA has at least one child PBDOM\_OBJECT. If this PBDOM\_CHARACTERDATA has no children, false is returned.

### Syntax

```
pbdom_chardata_name.HasChildren()
```

**Table 9.21:**

Argument	Description
pbdom_chardata_name	The name of a PBDOM_CHARACTERDATA.

### Return value

Boolean.

**Table 9.22:**

Value	Description
true	The current PBDOM_CHARACTERDATA has at least one child PBDOM_OBJECT
false	The current PBDOM_CHARACTERDATA has no child PBDOM_OBJECTs

### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If this PBDOM\_CHARACTERDATA is not a reference to an object derived from PBDOM\_CHARACTERDATA.

### Usage

If the PBDOM\_CHARACTERDATA has at least one child PBDOM\_OBJECT, true is returned. False is returned if there are no children.

Currently, false is always returned because no subclasses of PBDOM\_CHARACTERDATA contain child nodes.

## 9.1.14 IsAncestorObjectOf

### Description

The IsAncestorObjectOf method determines whether the current PBDOM\_CHARACTERDATA is the ancestor of another PBDOM\_OBJECT.

### Syntax

```
pbdom_chardata_name.IsAncestorObjectOf(pbdom_object pbdom_object_ret)
```

**Table 9.23:**

Argument	Description
pbdom_chardata_name	The name of a PBDOM_CHARACTERDATA
pbdom_object_ref	A PBDOM_OBJECT to check against

### Return value

Boolean.

Returns true if the current PBDOM\_CHARACTERDATA is the ancestor of the referenced PBDOM\_OBJECT, and false otherwise.

### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If this PBDOM\_CHARACTERDATA is not a reference to an object derived from PBDOM\_CHARACTERDATA.

## Usage

Currently, false is always returned because no subclasses of PBDOM\_CHARACTERDATA contain child nodes. Therefore, they cannot be ancestors of a PBDOM\_OBJECT.

### 9.1.15 SetParentObject

#### Description

The SetParentObject method sets the referenced PBDOM\_OBJECT to be the parent of the current PBDOM\_CHARACTERDATA.

#### Syntax

```
pbdom_chardata_name.SetParentObject(pbdom_object pbdom_object_ref)
```

**Table 9.24:**

Argument	Description
pbdom_chardata_name	The name of a PBDOM_CHARACTERDATA
pbdom_object_ref	A PBDOM_OBJECT to be set as the parent of this PBDOM_CHARACTERDATA object

#### Return value

PBDOM\_OBJECT.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If this PBDOM\_CHARACTERDATA is not a reference to an object derived from PBDOM\_CHARACTERDATA. This exception also occurs if the input PBDOM\_OBJECT is not a reference to an object derived from PBDOM\_OBJECT.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT -- If the current PBDOM\_CHARACTERDATA already has a parent.

EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT -- If the input PBDOM\_OBJECT is of a class that does not have a proper parent-child relationship with the class of this PBDOM\_CHARACTERDATA.

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- If the input PBDOM\_OBJECT requires a user-defined name, and it has not been named.

#### Examples

This example creates a PBDOM\_DOCUMENT based on the following DOM tree:

```
<abc>
  <data>
    <child_1/>
    <child_2/>
    <child_3/>
  </data>
</abc>
```

The code creates three separate types of PBDOM\_CHARACTERDATA objects and stores them in the pbdom\_chardata array. It then obtains the root element, uses it to obtain the data child element, and then uses that to obtain the first child element, which it sets as the parent of the first item in the pbdom\_chardata array.

The text of the array item is set to Comment. You can set the string content of any PBDOM\_CHARACTERDATA object after you have set it as the child of a parent.

The same process is repeated for the text and CDATA objects:

```
PBDOM_Builder      pbdombuilder_new
pbdom_document    pbdom_doc
PBDOM_CHARACTERDATA pbdom_chardata[]
PBDOM_ELEMENT     pbdom_elem
long              = 0
string strXML = "<abc><data><child_1/><child_2/><child_3/></data></abc>"

TRY
  pbdombuilder_new = Create PBDOM_Builder
  pbdom_doc = pbdombuilder_new.BuildFromString (strXML)

  pbdom_chardata[1] = Create PBDOM_COMMENT
  pbdom_chardata[2] = Create PBDOM_TEXT
  pbdom_chardata[3] = Create PBDOM_CDATA

  pbdom_elem = pbdom_doc.GetRootElement(). &
    GetChildElement("data").GetChildElement("child_1")
  pbdom_chardata[1].SetParentObject (pbdom_elem)
  pbdom_chardata[1].SetText ("Comment")

  pbdom_elem = pbdom_doc.GetRootElement(). &
    GetChildElement("data").GetChildElement("child_2")
  pbdom_chardata[2].SetParentObject (pbdom_elem)
  pbdom_chardata[2].SetText ("Text")

  pbdom_elem = pbdom_doc.GetRootElement(). &
    GetChildElement("data").GetChildElement("child_3")
  pbdom_chardata[3].SetParentObject (pbdom_elem)
  pbdom_chardata[3].SetText ("CDATA")

  pbdom_doc.SaveDocument ("c:\pbdom_doc_1.xml")

  Destroy pbdombuilder_new

CATCH (PBDOM_Exception except)
  MessageBox ("Exception Occurred", except.Text)
END TRY
```

When the PBDOM\_DOCUMENT is saved to a file, the output DOM tree looks like this:

```
<abc>
  <data>
    <child_1>
      <!--Comment-->
    </child_1>
    <child_2>
      Text
    </child_2>
    <child_3>
      <![CDATA[CDATA]]>
    </child_3>
  </data>
</abc>
```

## Usage

The PBDOM\_OBJECT that you set to be the parent of the current PBDOM\_CHARACTERDATA must have a legal parent-child relationship. If it does not, an exception is thrown.

## See also

[GetParentObject](#)

### 9.1.16 SetText

#### Description

The SetText method sets the input string to be the text content of the current PBDOM\_CHARACTERDATA object.

#### Syntax

```
pbdom_chardata_name.SetText(string strSet)
```

**Table 9.25:**

Argument	Description
pbdom_chardata_name	The name of a PBDOM_CHARACTERDATA
strSet	The string you want set as the text of the PBDOM_CHARACTERDATA

#### Return value

PBDOM\_CHARACTERDATA. The current PBDOM\_CHARACTERDATA object modified.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If this PBDOM\_CHARACTERDATA is not a reference to an object derived from PBDOM\_CHARACTERDATA.

#### Usage

The SetText method sets the input string to be the text content of the current PBDOM\_CHARACTERDATA object.

## See also

[GetText](#)

[GetTextNormalize](#)

[GetTextTrim](#)

## 10 PBDOM\_COMMENT Class

### About this chapter

This chapter describes the PBDOM\_COMMENT class.

### 10.1 PBDOM\_COMMENT

#### Description

The PBDOM\_COMMENT class represents a DOM Comment Node within an XML document. The PBDOM\_COMMENT class is derived from the PBDOM\_CHARACTERDATA class and is intended to extend the PBDOM\_CHARACTERDATA class with a set of methods intended specifically for manipulating DOM comment nodes.

You can use comments to annotate an XML document with user-readable information.

In PBDOM, when a document is parsed, any comments found within the document persist as part of the resultant DOM tree in memory. A PBDOM\_COMMENT created at runtime also becomes part of the DOM tree. However, an XML comment does not usually form part of the content model of a document.

The presence or absence of comments has no bearing on a document's validity. There is no requirement that comments must be predeclared in a DTD.

#### Methods

Some of the inherited methods from PBDOM\_OBJECT serve no meaningful objective, and only default or trivial functionalities result. These are described in the following table:

**Table 10.1:**

Method	Always returns
AddContent	current PBDOM_COMMENT
GetContent	false
GetName	a string "#comment"
HasChildren	false
InsertContent	current PBDOM_COMMENT
IsAncestorObjectOf	false
RemoveContent	false
SetContent	current PBDOM_COMMENT
SetName	false

PBDOM\_COMMENT has the following non-trivial methods:

[Append](#)

[Clone](#)

[Detach](#)

[Equals](#)

[GetObjectClass](#)[GetObjectClassString](#)[GetOwnerDocumentObject](#)[GetParentObject](#)[GetText](#)[GetTextNormalize](#)[GetTextTrim](#)[SetParentObject](#)[SetText](#)

### 10.1.1 Append

#### Description

The Append method is overloaded:

- Syntax 1 appends an input string to the text content that already exists within the current PBDOM\_COMMENT object.
- Syntax 2 appends the text data of a PBDOM\_CHARACTERDATA object to the text content that already exists within the current PBDOM\_COMMENT object.

#### Syntax

**Table 10.2:**

For this syntax	See
<code>Append(string strAppend)</code>	<a href="#">Append Syntax 1</a>
<code>Append(pbdom_characterdata pbdom_characterdata_ref)</code>	<a href="#">Append Syntax 2</a>

#### 10.1.1.1 Append Syntax 1

##### Description

Appends an input string to the text content that already exists within the current PBDOM\_COMMENT object.

##### Syntax

```
pbdom_comment_name.Append(string strAppend)
```

**Table 10.3:**

Argument	Description
<code>pbdom_comment_name</code>	The name of a PBDOM_COMMENT
<code>strAppend</code>	The string you want to append to the existing text of the current PBDOM_COMMENT object

##### Return value



PBDOM\_CHARACTERDATA. The current PBDOM\_COMMENT modified and returned as a PBDOM\_CHARACTERDATA object.

### 10.1.1.2 Append Syntax 2

#### Description

Appends the text data of a PBDOM\_CHARACTERDATA object to the text content that exists within the current PBDOM\_COMMENT object.

#### Syntax

```
pbdom_comment_name.Append(pbdom_characterdata pbdom_characterdata_ref)
```

**Table 10.4:**

Argument	Description
pbdom_comment_name	The name of a PBDOM_COMMENT
pbdom_characterdata_ref	The referenced PBDOM_CHARACTERDATA object whose text data is to be appended to the existing text of the current PBDOM_COMMENT object

#### Return value

PBDOM\_CHARACTERDATA. The current PBDOM\_COMMENT modified and returned as a PBDOM\_CHARACTERDATA object.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the input PBDOM\_CHARACTERDATA is not a reference to a PBDOM\_CHARACTERDATA-derived object.

#### Usage

Note that JDOM does not define an Append method for its COMMENT class. Because PBDOM implements its Append method in the base PBDOM\_CHARACTERDATA class, a PBDOM\_TEXT object, a PBDOM\_CDATA object, and a PBDOM\_COMMENT object can append their internal text data to each other because they are all PBDOM\_CHARACTERDATA-derived objects.

### 10.1.2 Clone

#### Description

Creates and returns a clone of the current PBDOM\_COMMENT.

#### Syntax

```
pbdom_comment_name.Clone(boolean bDeep)
```

**Table 10.5:**

Argument	Description
pbdom_comment_name	The name of a PBDOM_COMMENT

Argument	Description
bDeep	A boolean specifying whether a deep or shallow clone is returned. Values are true for a deep clone and false for a shallow clone

### Return value

PBDOM\_OBJECT.

### Examples

This example creates an XML document that, when serialized, appears as follows:

```
<!DOCTYPE root
[
<!ELEMENT root (level_1)*>
<!ELEMENT level_1 (level_2)*>
<!ELEMENT level_2 (#PCDATA)*>
]>
<root>
  <level_1>
    <!--Element at level : 1-->
    <level_2>
      <!--Element at level : 2-->
    </level_2>
  </level_1>
</root>
```

The definition of the DTD shows that the document is required to have the following composition:

- The document contains a root element with the name root.
- The root element contains zero or more occurrences of level\_1 elements.
- A level\_1 element contains zero or more level\_2 elements.
- A level\_2 element is expected to contain text.

The following PowerShell code supplies annotations within the document by including comments to mark level\_1 and level\_2 elements. The sample code creates a PBDOM\_DOCUMENT from an XML string that contains a DTD and a minimal root element. Then, it creates a comment that serves as a template. The template comment is then cloned, and instance-specific text is added for each element:

```
PBDOM_COMMENT pbdom_comm
PBDOM_COMMENT pbdom_comm_clone
PBDOM_ELEMENT pbdom_elem
PBDOM_DOCUMENT pbdom_doc
PBDOM_BUILDER pbdom_buildr
string strXML = "<!DOCTYPE root [<!ELEMENT root (level_1)*><!ELEMENT level_1
(level_2)*><!ELEMENT level_2 (#PCDATA)>]><root/>"

try
  // Create a PBDOM_DOCUMENT from the XML string that
  // contains a DTD and a minimal root element.
  pbdom_buildr = Create PBDOM_BUILDER
  pbdom_doc = pbdom_buildr.BuildFromString(strXML)

  // Create a template comment that can be reused.
```

```

pbdom_comm = Create PBDOM_COMMENT
pbdom_comm.SetText ("Element at level : ")

// Create a level_1 element.
pbdom_elem = Create PBDOM_ELEMENT
pbdom_elem.SetName("level_1")

// Clone the template comment, append instance-
// specific text, and add it to the level_1 element.
pbdom_comm_clone = pbdom_comm.Clone(true)
pbdom_elem.AddContent(pbdom_comm_clone.Append("1"))

// Add a level_1 element into the root element
// as stipulated by the DTD.
pbdom_doc.GetRootElement().AddContent(pbdom_elem)

// Create a level_2 element.
pbdom_elem = Create PBDOM_ELEMENT
pbdom_elem.SetName("level_2")

// Clone the template comment, append instance-
// specific text, and add it to the level_2 element.
pbdom_comm_clone = pbdom_comm.Clone(true)
pbdom_elem.AddContent(pbdom_comm_clone.Append("2"))

// Add a level_2 element into the level_1 element
// as stipulated by the DTD.
pbdom_doc.GetRootElement().GetChildElement &
("level_1").AddContent(pbdom_elem)

// Finally, serialize the document.
pbdom_doc.SaveDocument("sample.xml")

catch(PBDOM_EXCEPTION pbdom_e)
    MessageBox ("PBDOM_EXCEPTION", pbdom_e.GetMessage())
end try

```

## Usage

The Clone method creates a new PBDOM\_COMMENT object that is a duplicate of, and a separate object from, the original. Whether true or false is supplied, the clone is always identical to its original, because a PBDOM\_COMMENT does not contain a subtree of child PBDOM\_OBJECTs.

A PBDOM\_COMMENT clone has no parent. However, the clone resides in the same PBDOM\_DOCUMENT as its original, and if the original is standalone, the clone is standalone.

### 10.1.3 Detach

#### Description

Detaches a PBDOM\_COMMENT from its parent PBDOM\_OBJECT.

#### Syntax

```
pbdom_comment_name.Detach()
```

**Table 10.6:**

Argument	Description
pbdom_comment_name	The name of a PBDOM_COMMENT

**Return value**

PBDOM\_OBJECT.

The current PBDOM\_COMMENT is detached from its parent.

**Usage**

If the current PBDOM\_COMMENT object has no parent, no modifications occur.

**10.1.4 Equals****Description**

Tests for the equality of the current PBDOM\_COMMENT and a referenced PBDOM\_OBJECT.

**Syntax**

```
pbdom_comment_name.Equals(pbdom_object pbdom_object_ref)
```

**Table 10.7:**

Argument	Description
pbdom_comment_name	The name of a PBDOM_COMMENT.
pbdom_object_ref	A PBDOM_OBJECT to test for equality with the current PBDOM_COMMENT

**Return value**

Boolean.

Returns true if the current PBDOM\_COMMENT is equivalent to the input PBDOM\_OBJECT, and false otherwise.

**Throws**

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the referenced PBDOM\_OBJECT is not a reference to an object derived from a PBDOM\_OBJECT object.

**Usage**

True is returned only if the referenced PBDOM\_OBJECT is also a derived PBDOM\_COMMENT object and refers to the same DOM object as the current PBDOM\_COMMENT. Two separately created PBDOM\_COMMENTS, for example, can contain exactly the same text but are not equal.

**10.1.5 GetObjectClass****Description**

Returns a long integer code that indicates the class of the current PBDOM\_OBJECT.

**Syntax**

```
pbdom_object_name.GetObjectClass()
```

**Table 10.8:**

Argument	Description
pbdom_object_name	The name of a PBDOM_OBJECT

**Return value**

Long.

GetObjectClass returns a long integer code that indicates the class of the current PBDOM\_OBJECT. If pbdom\_object\_name is a PBDOM\_COMMENT, the returned value is 9.

**See also**

[GetObjectClassString](#)

**10.1.6 GetObjectClassString****Description**

Returns a string form of the class of the PBDOM\_OBJECT.

**Syntax**

```
pbdom_object_name.GetObjectClassString()
```

**Table 10.9:**

Argument	Description
pbdom_object_name	The name of a PBDOM_OBJECT

**Return value**

String.

GetObjectClassString returns a string that indicates the class of the current PBDOM\_OBJECT. If pbdom\_object\_name is a PBDOM\_COMMENT, the returned string is "pbdom\_comment".

**See also**

[GetObjectClass](#)

**10.1.7 GetOwnerDocumentObject****Description**

Returns the owning PBDOM\_DOCUMENT of the current PBDOM\_COMMENT.

**Syntax**

```
pbdom_comment_name.GetOwnerDocumentObject()
```

**Table 10.10:**

Argument	Description
pbdom_comment_name	The name of a PBDOM_COMMENT

**Return value**

PBDOM\_OBJECT.

**Usage**

If there is no owning PBDOM\_DOCUMENT, null is returned.

### 10.1.8 GetParentObject

#### Description

Returns the parent PBDOM\_OBJECT of the current PBDOM\_COMMENT.

#### Syntax

```
pbdom_comment_name.GetParentObject()
```

**Table 10.11:**

Argument	Description
pbdom_comment_name	The name of a PBDOM_COMMENT

#### Return value

PBDOM\_OBJECT.

#### Usage

The GetParentObject method returns the parent PBDOM\_OBJECT of the current PBDOM\_COMMENT. If the PBDOM\_COMMENT has no parent, null is returned.

#### See also

[SetParentObject](#)

### 10.1.9 GetText

#### Description

Allows you to obtain the text data that is contained within the current PBDOM\_COMMENT object.

#### Syntax

```
pbdom_comment_name.GetText()
```

**Table 10.12:**

Argument	Description
pbdom_comment_name	The name of a PBDOM_COMMENT

#### Return value

String.

The textual content of the current PBDOM\_COMMENT object.

#### Examples

If you have the comment <!-- A COMMENT-->, the GetText method returns the string A COMMENT.

#### See also

[GetTextNormalize](#)

[GetTextTrim](#)

[SetText](#)

### 10.1.10 GetTextNormalize

#### Description

Allows you to obtain the text data that is contained within the current PBDOM\_COMMENT object, with all surrounding whitespace characters removed and internal whitespace characters normalized to a single space.

#### Syntax

```
pbdom_comment_name.GetTextNormalize()
```

Table 10.13:

Argument	Description
pbdom_comment_name	The name of a PBDOM_COMMENT

#### Return value

String.

#### Examples

If you have the comment `<!-- A COMMENT -->`, which has three spaces before and after the text and between the two words, the `GetTextNormalize` method returns the string `A COMMENT`, which has a single space between the words.

#### Usage

This method allows the caller to obtain the text data that is contained within the current PBDOM\_COMMENT with all surrounding whitespace characters removed and internal whitespace characters normalized to single spaces. If no textual value exists for the current PBDOM\_COMMENT, or if only whitespace characters exist, an empty string is returned.

#### See also

[GetText](#)

[GetTextTrim](#)

[SetText](#)

### 10.1.11 GetTextTrim

#### Description

Returns the textual content of the current PBDOM\_COMMENT object with all surrounding whitespace characters removed.

#### Syntax

```
pbdom_comment_name.GetTextTrim()
```

Table 10.14:

Argument	Description
pbdom_comment_name	The name of a PBDOM_COMMENT

#### Return value

String.

### Examples

If you have the comment `<!-- A COMMENT -->`, which has three spaces before and after the text and between the two words, the `GetTextTrim` method returns the string `A COMMENT`. The whitespace characters between the words are preserved.

### Usage

This method allows the caller to obtain the text data that is contained within the current `PBDOM_COMMENT` with all surrounding whitespace characters removed. Internal whitespace characters are preserved. If no textual value exists for the current `PBDOM_COMMENT`, or if only whitespace characters exist, an empty string is returned.

### See also

[GetText](#)

[GetTextNormalize](#)

[SetText](#)

## 10.1.12 SetParentObject

### Description

Sets the referenced `PBDOM_OBJECT` to be the parent of the current `PBDOM_COMMENT`.

### Syntax

```
pbdom_comment_name.SetParentObject(pbdom_object pbdom_object_ref)
```

**Table 10.15:**

Argument	Description
<code>pbdom_comment_name</code>	The name of a <code>PBDOM_COMMENT</code>
<code>pbdom_object_ref</code>	A <code>PBDOM_OBJECT</code> to be set as the parent of the current <code>PBDOM_COMMENT</code>

### Return value

`PBDOM_OBJECT`.

### Throws

`EXCEPTION_PBDOM_OBJECT_INVALID_FOR_USE` -- If the input `PBDOM_OBJECT` is not a reference to an object derived from `PBDOM_OBJECT`.

`EXCEPTION_PBDOM_OBJECT_ALREADY_HAS_PARENT` -- If the current `PBDOM_COMMENT` already has a parent.

`EXCEPTION_INAPPROPRIATE_USE_OF_PBDOM_OBJECT` -- If the input `PBDOM_OBJECT` is of a class that does not have a proper parent-child relationship with the `PBDOM_COMMENT` class.

`EXCEPTION_USE_OF_UNNAMED_PBDOM_OBJECT` -- If the input `PBDOM_OBJECT` requires a user-defined name, and it has not been named.

### Usage



This method sets the input PBDOM\_OBJECT as the parent of this PBDOM\_COMMENT. The caller is responsible for ensuring that the current PBDOM\_COMMENT and the input PBDOM\_OBJECT can have a legal parent-child relationship. Currently, only a PBDOM\_ELEMENT and a PBDOM\_DOCUMENT can be set as the parent of a PBDOM\_COMMENT.

The PBDOM\_COMMENT SetParentObject method differs from the JDOM Comment setParent method in two ways:

- JDOM defines a setParent method for several specific classes, including Element, Comment, and CDATA. PBDOM implements the SetParentObject method in the base PBDOM\_OBJECT class to allow for polymorphism.
- The JDOM Comment's setParent method takes only an Element class object as a parameter:

```
COMMENT::setParent(Element parent)
```

To set a Document as the parent owner of a Comment using JDOM, you use the setDocument method:

```
COMMENT::setDocument(Document document)
```

In PBDOM, SetParentObject takes a reference to a PBDOM\_OBJECT, so that both a PBDOM\_ELEMENT and a PBDOM\_DOCUMENT can be set as a parent.

#### See also

[GetOwnerDocumentObject](#)

[GetParentObject](#)

### 10.1.13 SetText

#### Description

Sets the input string to be the text content of the current PBDOM\_COMMENT object.

#### Syntax

```
pbdom_comment_name.SetText(string strSet)
```

**Table 10.16:**

Argument	Description
pbdom_comment_name	The name of a PBDOM_COMMENT
strSet	The string you want set as the text of the PBDOM_COMMENT

#### Return value

String.

#### See also

[GetText](#)

[GetTextNormalize](#)

[GetTextTrim](#)

# 11 PBDOM\_DOCTYPE Class

## About this chapter

This chapter describes the PBDOM\_DOCTYPE class.

## 11.1 PBDOM\_DOCTYPE

### Description

The PBDOM\_DOCTYPE class represents the Document Type Declaration Object of an XML DOM Document. The PBDOM\_DOCTYPE class provides access to the name of the root element that is constrained within the DOCTYPE as well as the internal subset, system, and public IDs.

### Methods

Some of the inherited methods from PBDOM\_OBJECT serve no meaningful objective and only default or trivial functionalities result. These are described in the following table:

**Table 11.1:**

Method	Always returns
AddContent	The current PBDOM_DOCTYPE
GetContent	false
GetText	Empty string
GetTextNormalize	Empty string
GetTextTrim	Empty string
HasChildren	false
InsertContent	The current PBDOM_DOCTYPE
IsAncestorObjectOf	false
RemoveContent	false
SetContent	The current PBDOM_DOCTYPE

PBDOM\_DOCTYPE has the following non-trivial methods:

**Table 11.2:**

<a href="#">Clone</a>	<a href="#">GetObjectClassString</a>	<a href="#">SetInternalSubset</a>
<a href="#">Detach</a>	<a href="#">GetOwnerDocumentObject</a>	<a href="#">SetName</a>
<a href="#">Equals</a>	<a href="#">GetParentObject</a>	<a href="#">SetParentObject</a>
<a href="#">GetInternalSubset</a>	<a href="#">GetPublicID</a>	<a href="#">SetPublicID</a>
<a href="#">GetName</a>	<a href="#">GetSystemID</a>	<a href="#">SetSystemID</a>
<a href="#">GetObjectClass</a>	<a href="#">SetDocument</a>	

### 11.1.1 Clone

#### Description

Creates and returns a clone of the current PBDOM\_DOCTYPE.

### Syntax

```
pbdom_doctype_name.Clone(boolean bDeep)
```

**Table 11.3:**

Argument	Description
pbdom_doctype_name	The name of a PBDOM_DOCTYPE object.
bDeep	A boolean specifying whether a deep or shallow clone is returned. Values are TRUE for a deep clone and FALSE for a shallow clone. This argument is currently ignored.

### Return value

PBDOM\_OBJECT. A deep clone of the current PBDOM\_DOCTYPE housed in a PBDOM\_OBJECT.

### Usage

A PBDOM\_DOCTYPE clone (whether shallow or deep) is always an exact copy of its original. This is because a PBDOM\_DOCTYPE does not contain any subtree of child PBDOM\_OBJECTs.

A PBDOM\_DOCTYPE clone has no parent. However, the clone resides in the same PBDOM\_DOCUMENT as its original. If the original PBDOM\_DOCTYPE is standalone, the clone is standalone.

## 11.1.2 Detach

### Description

Detaches a PBDOM\_DOCTYPE object from its parent PBDOM\_DOCUMENT object. The detached PBDOM\_DOCTYPE object is still part of the PBDOM\_DOCUMENT object in which it resided before the Detach method was invoked, but it no longer has a parent PBDOM\_DOCUMENT object.

### Syntax

```
pbdom_doctype_name.Detach()
```

**Table 11.4:**

Argument	Description
pbdom_doctype_name	The name of a PBDOM_DOCTYPE object

### Return value

PBDOM\_OBJECT. The PBDOM\_DOCTYPE object modified and returned as a PBDOM\_OBJECT object.

## 11.1.3 Equals

### Description

Tests for the equality of the current PBDOM\_DOCTYPE and a referenced PBDOM\_OBJECT.

### Syntax

```
pbdom_doctype_name.Equals(pbdom_object_ref)
```

**Table 11.5:**

Argument	Description
pbdom_doctype_name	The name of a PBDOM_DOCTYPE object
pbdom_object_ref	A PBDOM_OBJECT to test for equality with the current PBDOM_DOCTYPE

### Return value

Boolean.

Returns true if the current PBDOM\_DOCTYPE is equivalent to the input PBDOM\_OBJECT, and false otherwise.

### Usage

True is returned only if the referenced PBDOM\_OBJECT is also a PBDOM\_DOCTYPE and refers to the same DOM Doctype object as the current PBDOM\_DOCTYPE.

## 11.1.4 GetInternalSubset

### Description

Returns the internal subset data of the DOCTYPE.

### Syntax

```
pbdom_doctype_name.GetInternalSubset()
```

**Table 11.6:**

Argument	Description
pbdom_doctype_name	The name of a PBDOM_DOCTYPE object

### Return value

String.

### See also

[SetInternalSubset](#)

## 11.1.5 GetName

### Description

Allows you to obtain the name of the root element that is being constrained within the current PBDOM\_DOCTYPE.

### Syntax

```
pbdom_doctype_name.GetName()
```

**Table 11.7:**

Argument	Description
pbdom_doctype_name	The name of a PBDOM_DOCTYPE object

**Return value**

String.

**Examples**

If you have the following DOCTYPE declaration, the GetName method returns abc.

```
<!DOCTYPE abc [<!-- internal subset -->
<!ELEMENT abc (#PCDATA)> <!ELEMENT data (#PCDATA)> <!ELEMENT inner_data
(#PCDATA)>]>
```

**11.1.6 GetObjectClass****Description**

Returns a long integer code that indicates the class of the current PBDOM\_OBJECT.

**Syntax**

```
pbdom_object_name.GetObjectClass()
```

**Table 11.8:**

Argument	Description
pbdom_object_name	The name of a PBDOM_OBJECT

**Return value**

Long.

A long integer code that indicates the class of the current PBDOM\_OBJECT. If pbdom\_object\_name is a PBDOM\_DOCTYPE, the returned value is 4.

**11.1.7 GetObjectClassString****Description**

Returns a string form of the class of the PBDOM\_OBJECT.

**Syntax**

```
pbdom_object_name.GetObjectClassString()
```

**Table 11.9:**

Argument	Description
pbdom_object_name	The name of your PBDOM_OBJECT

**Return value**

String.

A string that indicates the class of the current PBDOM\_OBJECT. If pbdom\_object\_name is a PBDOM\_DOCTYPE, the returned string is "pbdom\_doctype".

### 11.1.8 GetOwnerDocumentObject

#### Description

Returns the owning PBDOM\_DOCUMENT of the current PBDOM\_DOCTYPE.

#### Syntax

```
pbdom_doctype_name.GetOwnerDocumentObject()
```

**Table 11.10:**

Argument	Description
pbdom_doctype_name	The name of a PBDOM_DOCTYPE object

#### Return value

PBDOM\_OBJECT.

#### Usage

If there is no owning PBDOM\_DOCUMENT, null is returned.

### 11.1.9 GetParentObject

#### Description

Returns the parent PBDOM\_OBJECT of the current PBDOM\_DOCTYPE.

#### Syntax

```
pbdom_doctype_name.GetParentObject()
```

**Table 11.11:**

Argument	Description
pbdom_doctype_name	The name of a PBDOM_DOCTYPE object

#### Return value

PBDOM\_OBJECT.

#### Usage

The parent is also a PBDOM\_DOCUMENT object. If the PBDOM\_OBJECT has no parent, null is returned.

### 11.1.10 GetPublicID

#### Description

Retrieves the public ID of an externally reference DTD declared in the DOCTYPE.

#### Syntax

```
pbdom_doctype_name.GetPublicID()
```

**Table 11.12:**

Argument	Description
pbdom_doctype_name	The name of a PBDOM_DOCTYPE object

**Return value**

String.

If no public ID is referenced, an empty string is returned.

**Examples**

Suppose you have the following DTD declaration:

```
<!DOCTYPE Books PUBLIC "-//MyCompany//DTD//EN" "http://mycompany.com/dtd/mydoctype.dtd">
```

The following PowerShell code displays the public and system IDs in message boxes:

```
pbdom_doctype pbdom_doctype_1
pbdom_document pbdom_doc

pbdom_doctype_1 = pbdom_doc.GetDocType()
MessageBox ("DocType Public ID", &
    pbdom_doctype_1.GetPublicID())
MessageBox ("DocType System ID", &
    pbdom_doctype_1.GetSystemID())
```

The returned strings from the calls to GetPublicID and GetSystemID are:

```
"-//MyCompany//DTD//EN"
"http://mycompany.com/dtd/mydoctype.dtd"
```

**See also**

[GetSystemID](#)

[SetPublicID](#)

[SetSystemID](#)

**11.1.11 GetSystemID****Description**

Retrieves the system ID of an externally referenced DTD declared in the DOCTYPE.

**Syntax**

```
pbdom_doctype_name.GetSystemID()
```

**Table 11.13:**

Argument	Description
pbdom_doctype_name	The name of a PBDOM_DOCTYPE object

**Return value**

String.

If no system ID is referenced, an empty string is returned.



## Examples

See [GetPublicID](#).

## See also

[GetPublicID](#)

[SetPublicID](#)

[SetSystemID](#)

### 11.1.12 SetDocument

#### Description

Sets the owning PBDOM\_DOCUMENT of the current PBDOM\_DOCTYPE.

#### Syntax

```
pbdom_doctype_name.SetDocument (pbdom_document pbdom_document_ref)
```

**Table 11.14:**

Argument	Description
pbdom_doctype_name	The name of a PBDOM_DOCTYPE object
pbdom_document_ref	A PBDOM_DOCUMENT object to be set as the owner document of this PBDOM_DOCTYPE object

#### Return value

PBDOM\_DOCTYPE. The current PBDOM\_DOCTYPE modified to be the DOCTYPE of the referenced PBDOM\_DOCUMENT.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- if the input PBDOM\_DOCUMENT object is invalid for use in any way.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT -- if this current PBDOM\_DOCTYPE already has a parent PBDOM\_OBJECT. In this case, this PBDOM\_DOCTYPE is already the DOCTYPE of some document.

#### Usage

A DOM DOCTYPE object can have no owner document, or it can have an owner document but no parent node. A DOCTYPE that has an owner document as well as a parent node is the actual DOCTYPE of the owner document.

## See also

[SetParentObject](#)

### 11.1.13 SetInternalSubset

#### Description

Sets the data for the internal subset of the PBDOM\_DOCTYPE.

## Syntax

```
pbdom_doctype_name.SetInternalSubset()
```

**Table 11.15:**

Argument	Description
pbdom_doctype_name	The name of a PBDOM_DOCTYPE object

## Return value

PBDOM\_DOCTYPE. The current PBDOM\_DOCTYPE with the new internal subset.

## Examples

Suppose you have the following DTD declaration:

```
<!DOCTYPE abc [<!ELEMENT abc (#PCDATA)> <!ELEMENT data (#PCDATA)> <!ELEMENT inner_data (#PCDATA)>]>
```

The following code displays the internal subset in a message box:

```
string strInternalSubset
pbdom_document pbdom_doc

strInternalSubset = pbdom_doc.GetDocType().GetInternalSubset()
strInternalSubset += "<!ELEMENT another_data (#PCDATA)>"
pbdom_doc.GetDocType().SetInternalSubset (strInternalSubset)
MessageBox ("Get Internal Subset", &
    pbdom_doc.GetDocType().GetInternalSubset())
```

The returned string from the call to GetInternalSubset is:

```
"<!-- internal subset --> <!ELEMENT abc (#PCDATA)> <!ELEMENT data (#PCDATA)> <!ELEMENT inner_data (#PCDATA)> <!ELEMENT another_data (#PCDATA)>"
```

The new ELEMENT declaration for "another\_data" is included in the final internal subset.

## See also

[GetInternalSubset](#)

### 11.1.14 SetName

#### Description

The SetName method sets the name of the root element that is declared by this PBDOM\_DOCTYPE.

#### Syntax

```
pbdom_doctype_name.SetName(string strName)
```

**Table 11.16:**

Argument	Description
pbdom_doctype_name	The name of a PBDOM_DOCTYPE object
strName	The new name you want to set for the root element that is declared by the current PBDOM_DOCTYPE

**Return value**

Boolean.

Returns true if the name of the root element was changed and false otherwise.

**11.1.15 SetParentObject****Description**

The SetParentObject method sets the referenced PBDOM\_OBJECT to be the parent of the current PBDOM\_OBJECT and so sets the DOCTYPE represented by this PBDOM\_DOCTYPE to be the DOCTYPE of the referenced PBDOM\_DOCUMENT.

**Syntax**

```
pbdom_doctype_name.SetParentObject(pbdom_object pbdom_object_ref)
```

**Table 11.17:**

Argument	Description
pbdom_doctype_name	The name of a PBDOM_DOCTYPE object
pbdom_object_ref	A PBDOM_OBJECT to be set as the parent of the current PBDOM_DOCTYPE

**Return value**

PBDOM\_OBJECT.

**Throws**

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT -- If this PBDOM\_DOCTYPE already has a parent.

EXCEPTION\_MULTIPLE\_DOCTYPE -- If the input PBDOM\_OBJECT is a PBDOM\_DOCUMENT object and already has a doctype.

EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT -- If the input PBDOM\_OBJECT is not a PBDOM\_DOCUMENT.

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the input PBDOM\_OBJECT is not associated with a derived PBDOM\_OBJECT.

**Usage**

This method sets the input PBDOM\_OBJECT as the parent of the current PBDOM\_OBJECT. The input PBDOM\_OBJECT must be a PBDOM\_DOCUMENT. If it is not, an exception is thrown.

In PBDOM, calling SetParentObject is equivalent to setting the input PBDOM\_DOCUMENT as the owner document and parent node of the current PBDOM\_DOCTYPE. This has the effect of setting the DOCTYPE in PBDOM\_DOCTYPE as the DOCTYPE of the document.

A DOM DOCTYPE object can have no owner document, or it can have an owner document but no parent node. A DOCTYPE that has an owner document as well as a parent node is the actual DOCTYPE of the owner document.

This method is exactly the same as the SetDocument method.

**See also**[SetDocument](#)**11.1.16 SetPublicID****Description**

Sets the public ID of an externally referenced DTD.

**Syntax**

```
pbdom_doctype_name.SetPublicID(string strPublicID)
```

**Table 11.18:**

Argument	Description
pbdom_doctype_name	The name of a PBDOM_DOCTYPE object
strPublicID	A string that specifies the new public ID

**Return value**

PBDOM\_DOCTYPE.

**Examples**

Suppose you have the following DTD declaration:

```
<!DOCTYPE abc [<!ELEMENT abc (#PCDATA)> <!ELEMENT data (#PCDATA)> <!ELEMENT inner_data (#PCDATA)>]>
```

The following PowerScript sets the public ID, and then gets it and displays it in a message box:

```
PBDOM_DOCUMENT pbdom_doc  
  
pbdom_doc.GetDocType().SetPublicID &  
  (" -//MyCompany//DTD//EN" )  
MessageBox ("Get Public ID", &  
  pbdom_doc.GetDocType().GetPublicID())
```

The returned string from the GetPublicID call is:

```
" -//MyCompany//DTD//EN"
```

The final DOCTYPE definition in the document is:

```
<!DOCTYPE abc PUBLIC "-//MyCompany//DTD//EN" [<!ELEMENT abc (#PCDATA)> <!ELEMENT data (#PCDATA)> <!ELEMENT inner_data (#PCDATA)>]>
```

---

**About Public ID**

The PUBLIC ID is usually accompanied by a SYSTEM ID, so the DOCTYPE declaration in this example (with a PUBLIC ID but no SYSTEM ID) might be considered invalid by some parsers.

---

**See also**[GetPublicID](#)[GetSystemID](#)

[SetSystemID](#)**11.1.17 SetSystemID****Description**

Sets the system ID of an externally referenced DTD.

**Syntax**

```
pbdom_doctype_name.SetSystemID(strSystemID)
```

**Table 11.19:**

Argument	Description
pbdom_doctype_name	The name of a PBDOM_DOCTYPE object
strSystemID	A string that specifies the new system ID

**Return value**

PBDOM\_DOCTYPE.

**Examples**

Suppose you have the following DTD declaration:

```
<!DOCTYPE abc [<!ELEMENT abc (#PCDATA)> <!ELEMENT data (#PCDATA)> <!ELEMENT inner_data (#PCDATA)>]>
```

The following PowerScript sets the system ID and then gets it and returns it in a message box:

```
PBDOM_DOCUMENT pbdom_doc
pbdom_doc.GetDocType().SetSystemID &
 ("http://www.appeon.com/dtd/datadef.dtd")
MessageBox ("Get System ID", &
 pbdom_doc.GetDocType().GetSystemID())
```

The returned string from the GetSystemID call is:

```
"http://www.appeon.com/dtd/datadef.dtd"
```

The final DOCTYPE definition in the document is:

```
<!DOCTYPE abc SYSTEM "http://www.appeon.com/dtd/datadef.dtd" [<!ELEMENT abc (#PCDATA)> <!ELEMENT data (#PCDATA)> <!ELEMENT inner_data (#PCDATA)>]>
```

**See also**

[GetPublicID](#)

[GetSystemID](#)

[SetPublicID](#)

## 12 PBDOM\_DOCUMENT Class

### About this chapter

This chapter describes the PBDOM\_DOCUMENT class.

### 12.1 PBDOM\_DOCUMENT

#### Description

The PBDOM\_DOCUMENT class defines behavior for an XML DOM document. Methods allow access to the root element, processing instructions, and other document-level information.

The PBDOM\_DOCUMENT class inherits from a PBDOM\_OBJECT and so provides specialized implementations for most of the PBDOM\_OBJECT class methods.

#### Methods

Some of the inherited methods from PBDOM\_OBJECT serve no meaningful objective and only default or trivial functionalities result. These are described in the following table:

**Table 12.1:**

Method	Always returns
Detach	The current PBDOM_DOCUMENT
GetName	The string "#document"
GetOwnerDocumentObject	null
GetParentObject	null
GetText	An empty string
GetTextNormalize	An empty string
GetTextTrim	An empty string
SetName	false
SetParentObject	The current PBDOM_DOCUMENT

PBDOM\_DOCUMENT has the following non-trivial methods:

**Table 12.2:**

<a href="#">AddContent</a>	<a href="#">HasRootElement</a>
<a href="#">Clone</a>	<a href="#">InsertContent</a>
<a href="#">DetachRootElement</a>	<a href="#">IsAncestorObjectOf</a>
<a href="#">Equals</a>	<a href="#">NewDocument</a>
<a href="#">GetContent</a>	<a href="#">RemoveContent</a>
<a href="#">GetDocType</a>	<a href="#">SaveDocument</a>
<a href="#">GetElementsByTagName</a>	<a href="#">SaveDocumentIntoString</a>
<a href="#">GetObjectClass</a>	<a href="#">SetContent</a>

<a href="#">GetObjectClassString</a>	<a href="#">SetDocType</a>
<a href="#">GetRootElement</a>	<a href="#">SetRootElement</a>
<a href="#">HasChildren</a>	

### 12.1.1 AddContent

#### Description

Allows you to add a new PBDOM\_OBJECT into the current PBDOM\_DOCUMENT object.

#### Syntax

```
pbdom_document_name.AddContent(pbdom_object pbdom_object_ref)
```

**Table 12.3:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object
pbdom_object_ref	The PBDOM_OBJECT to add

#### Return value

PBDOM\_OBJECT. The return value is the newly modified PBDOM\_DOCUMENT object returned as a PBDOM\_OBJECT.

#### Throws

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- The input PBDOM\_OBJECT is nameable, but it currently has no name.

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- The input PBDOM\_OBJECT object is not associated with a derived PBDOM\_OBJECT class object.

EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT -- Adding the input PBDOM\_OBJECT is inappropriate. See description section below on the valid PBDOM\_OBJECTs that can be added to a PBDOM\_DOCUMENT object.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT -- If the PBDOM\_OBJECT to be added already has a parent PBDOM\_OBJECT.

EXCEPTION\_MULTIPLE\_ROOT\_ELEMENT -- If a PBDOM\_ELEMENT is to be added and this document already has a root element.

EXCEPTION\_MULTIPLE\_DOCTYPE -- If a PBDOM\_DOCTYPE is to be added and this document already has a DOCTYPE.

#### Examples

The document pbdom\_doc1 is created with three elements: pbdom\_elem\_1, pbdom\_elem\_2, and pbdom\_elem\_3. pbdom\_elem\_2 and pbdom\_elem\_3 are set as children of pbdom\_element\_1. pbdom\_doc1.GetRootElement().Detach()detaches the root element from pbdom\_doc1. pbdom\_elem\_1 is added as a child of pbdom\_doc1 with pbdom\_doc1.AddContent(pbdom\_elem\_1).

TRY

```
PBDOM_ELEMENT pbdom_elem_1
```

```

PBDOM_ELEMENT pbdom_elem_2
PBDOM_ELEMENT pbdom_elem_3
PBDOM_DOCUMENT pbdom_doc1

pbdom_doc1 = Create PBDOM_DOCUMENT
pbdom_elem_1 = Create PBDOM_ELEMENT
pbdom_elem_2 = Create PBDOM_ELEMENT
pbdom_elem_3 = Create PBDOM_ELEMENT

pbdom_elem_1.SetName("pbdom_elem_1")
pbdom_elem_2.SetName("pbdom_elem_2")
pbdom_elem_3.SetName("pbdom_elem_3")

pbdom_elem_1.AddContent(pbdom_elem_2)
pbdom_elem_1.AddContent(pbdom_elem_3)

pbdom_doc1.NewDocument("", "", "Root_Element", &
    "", "")
pbdom_doc1.GetRootElement().Detach()
pbdom_doc1.AddContent(pbdom_elem_1)
CATCH (pbdom_exception ex)
    MessageBox("Exception", ex.getMessage())
END TRY

```

The original root element <Root\_Element> has been detached and replaced by <pbdom\_elem\_1>. The document is transformed to:

```

<!DOCTYPE Root_Element>
<pbdom_elem_1>
  <pbdom_elem_2/>
  <pbdom_elem_3/>
</pbdom_elem_1>

```

If the following root element detachment statement is omitted, an exception is thrown:

```
pbdom_doc1.GetRootElement().Detach()
```

### Usage

The new PBDOM\_OBJECT becomes a child PBDOM\_OBJECT of the current PBDOM\_DOCUMENT. The following table lists the PBDOM\_OBJECTs that can be added to a PBDOM\_DOCUMENT object and the restrictions for their addition.

**Table 12.4:**

PBDOM_OBJECT	Restrictions
PBDOM_ELEMENT	<p>Allowed to be added only if this document currently does not contain any root element. Otherwise the exception EXCEPTION_MULTIPLE_ROOT_ELEMENT is thrown.</p> <p>The PBDOM_ELEMENT to be added must not already have a parent PBDOM_OBJECT. If it does, the exception EXCEPTION_PBDOM_OBJECT_ALREADY_HAS_PARE is thrown.</p>
PBDOM_COMMENT	Any number of PBDOM_COMMENT objects can be added to a document.



PBDOM_OBJECT	Restrictions
	The only restriction is that the PBDOM_COMMENT must not already have a parent. If so, the exception EXCEPTION_PBDOM_OBJECT_ALREADY_HAS_PARENT is thrown.
PBDOM_PROCESSINGINSTRUCTION	Any number of PBDOM_PROCESSINGINSTRUCTION objects can be added to a document.  The only restriction is that the PBDOM_PROCESSINGINSTRUCTION must not already have a parent. If so, the exception EXCEPTION_PBDOM_OBJECT_ALREADY_HAS_PARENT is thrown.
PBDOM_DOCTYPE	Allowed to be added only if this document currently does not contain any DOCTYPE node. Otherwise the exception EXCEPTION_MULTIPLE_DOCTYPE is thrown.  The PBDOM_DOCTYPE to be added must not already have a parent PBDOM_OBJECT. If it does, the exception EXCEPTION_PBDOM_OBJECT_ALREADY_HAS_PARENT is thrown.

**See also**[GetContent](#)[InsertContent](#)[RemoveContent](#)[SetContent](#)**12.1.2 Clone****Description**

Creates a clone of the current PBDOM\_DOCUMENT object.

**Syntax**

```
pbdom_document_name.Clone(boolean bDeep)
```

**Table 12.5:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object

Argument	Description
bDeep	A boolean specifying whether a deep or shallow clone is returned. Values are true for a deep clone and false for a shallow clone.

**Return value**

PBDOM\_OBJECT.

**Throws**

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- The internal implementation of the PBDOM\_DOCUMENT object is null. The occurrence of this exception is rare but can happen if severe memory corruption occurs.

**Usage**

If you specify a deep clone, the Clone method creates a deep clone of the current PBDOM\_DOCUMENT object as a PBDOM\_OBJECT. The method recursively clones the subtree under the PBDOM\_DOCUMENT object, where the subtree consists of all legal children of the PBDOM\_DOCUMENT object.

If a shallow clone is requested, this method clones only the PBDOM\_DOCUMENT object and returns a completely empty PBDOM\_DOCUMENT object as a PBDOM\_OBJECT.

**12.1.3 DetachRootElement****Description**

Detaches the root element of this document and returns it.

**Syntax**

```
pbdom_document_name.DetachRootElement()
```

**Table 12.6:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object

**Return value**

PBDOM\_ELEMENT.

**Throws**

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- Insufficient memory was encountered while executing this method.

**See also**

[GetRootElement](#)

[HasRootElement](#)

[SetRootElement](#)

### 12.1.4 Equals

#### Description

Tests for the equality of the current PBDOM\_DOCUMENT object and a referenced PBDOM\_OBJECT.

#### Syntax

```
pbdom_document_name.Equals(pbdom_object pbdom_object_ref)
```

**Table 12.7:**

Argument	Description
pbdom_document_name	The name of a PBDOM_OBJECT
pbdom_object_ref	A PBDOM_OBJECT to test for equality with the current PBDOM_OBJECT

#### Return value

Boolean.

Returns true if the current PBDOM\_DOCUMENT object is equivalent to the input PBDOM\_OBJECT, and false otherwise.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- The input PBDOM\_OBJECT is not associated with a derived PBDOM\_OBJECT class object.

EXCEPTION\_INVALID\_ARGUMENT -- The input PBDOM\_OBJECT is invalid. This can happen if the object has not been initialized properly or is a null object reference.

#### Usage

True is returned only if the referenced PBDOM\_OBJECT is also a PBDOM\_DOCUMENT object and refers to the same DOM document as the current PBDOM\_DOCUMENT object.

### 12.1.5 GetContent

#### Description

Returns all child content of the current PBDOM\_DOCUMENT object.

#### Syntax

```
pbdom_document_name.GetContent(ref pbdom_object pbdom_object_array[ ])
```

**Table 12.8:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object
pbdom_object_array	The referenced name of an array of PBDOM_OBJECTs that receives PBDOM_OBJECTs

#### Return value

Boolean.

Returns true for success and false for failure.

### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object is not associated with a derived PBDOM\_OBJECT class object.

### Examples

Assume a PBDOM\_DOCUMENT object called pbdom\_doc contains the following XML document.

```
<Root>
  <Element_1>
    <Element_1_1/>
    <Element_1_2/>
    <Element_1_3/>
  </Element_1>
  <Element_2/>
  <Element_3/>
</Root>
```

In the following PowerShell code fragment, the array pbdom\_obj\_array contains just one PBDOM\_ELEMENT which represents the element Root: pbdom\_obj\_array[1] - <Root>:

```
PBDOM_DOCUMENT pbdom_doc
PBDOM_OBJECT pbdom_obj_array[]
...
pbdom_doc.GetContent(pbdom_obj_array)
pbdom_doc.GetRootElement().GetContent(pbdom_obj_array)
```

The call to GetRootElement in the last line of the previous code fragment yields an array that contains:

```
pbdom_obj_array[1] - <Element_1>
pbdom_obj_array[2] - <Element_2>
pbdom_obj_array[3] - <Element_3>
```

The returned PBDOM\_OBJECT array can be manipulated. For example, the following statement causes Element\_2 to contain the Text node "Element 2 Text":

```
pbdom_obj_array[2].AddContent ("Element 2 Text")
```

After this call, the tree is as follows:

```
<Root>
  Element_1>
    Element_1_1/>
    Element_1_2/>
    Element_1_3/>
  /Element_1>
  Element_2>Element 2 Text<Element_2/>
  Element_3/>
</Root>
```

### Usage

The returned array is passed by reference, with items in the same order in which they appear in the PBDOM\_DOCUMENT object. Any changes to any item of the array affect the actual item to which it refers.

**See also**[AddContent](#)[InsertContent](#)[RemoveContent](#)[SetContent](#)**12.1.6 GetDocType****Description**

Allows you to retrieve the DOCTYPE declaration of the current XML DOM document.

**Syntax**

```
pbdom_document_name.GetDocType()
```

**Table 12.9:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object

**Return value**

PBDOM\_DOCTYPE.

**Throws**

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- Insufficient memory was encountered while executing this method.

**Usage**

The DOCTYPE declaration is housed in a PBDOM\_OBJECT.

**12.1.7 GetElementsByTagName****Description**

Retrieves all the elements in the XML document that have the specified TagName.

**Syntax**

```
pbdom_object_name.GetElementsByTagName(string strTagName, ref pbdom_element  
pbdom_element_array[])
```

**Table 12.10:**

Argument	Description
strTagName	The TagName of the elements to be searched for
pbdom_element_array[]	A reference to a PBDOM_ELEMENT object array that has the specified TagName

**Return value**

Boolean.

GetElementsByTagName returns true for success and false if an exception occurs.

## Examples

Assume a PBDOM\_DOCUMENT contains the following XML fragment:

```
<book>
  <title>The Winter's Tale</title>
  <author>William Shakespeare</author>
  <price>7.95</price>
  <quantity>1</quantity>
</book>
<book>
  <title>Le Lecon</title>
  <author>Eugene Ionesco</author>
  <price>10.95</price>
  <quantity>1</quantity>
</book>
<book>
  <title>Deutsches Tempo</title>
  <author>Kurt Tucholsky</author>
  <price>13.95</price>
  <quantity>1</quantity>
</book>
```

The following statements extract the list of titles from the document and display it in a multilineedit control:

```
pbdom_document doc
pbdom_element element[]

// doc contains role elements
boolean bb_bool

bb_bool = doc.getelementsbytagname("title",element[])

integer ii_bound, i

ii_bound = upperbound(element)
for i = 1 to ii_bound
  mle_1.text += element[i].gettext() + "~r~n"
next
```

### 12.1.8 GetObjectClass

#### Description

Returns a long integer code that indicates the class of the current PBDOM\_OBJECT.

#### Syntax

```
pbdom_object_name.GetObjectClass()
```

**Table 12.11:**

Argument	Description
pbdom_object_name	The name of a PBDOM_OBJECT

#### Return value

Long.

GetObjectClass returns a long integer code that indicates the class of the current PBDOM\_OBJECT. If pbdom\_object\_name is a PBDOM\_DOCUMENT object, the returned value is 2.

### 12.1.9 GetObjectClassString

#### Description

Returns a string form of the class of the PBDOM\_OBJECT.

#### Syntax

```
pbdom_object_name.GetObjectClassString()
```

**Table 12.12:**

Argument	Description
pbdom_object_name	The name of a PBDOM_OBJECT

#### Return value

String.

GetObjectClassString returns a string that indicates the class of the current PBDOM\_OBJECT. If pbdom\_object\_name is a PBDOM\_DOCUMENT object, the returned string is "pbdom\_document".

### 12.1.10 GetRootElement

#### Description

Retrieves the root element of the current XML DOM document.

#### Syntax

```
pbdom_document_name.GetRootElement()
```

**Table 12.13:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object

#### Return value

PBDOM\_ELEMENT. The root element of the PBDOM\_DOCUMENT object housed in a PBDOM\_ELEMENT object.

#### Throws

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- Insufficient memory was encountered while executing this method.

#### Usage

The return value is the root element encapsulated in a PBDOM\_ELEMENT object.

**See also**

[DetachRootElement](#)

[HasRootElement](#)

[SetRootElement](#)

### 12.1.11 HasChildren

**Description**

Returns true if the current PBDOM\_DOCUMENT object has at least one child PBDOM\_OBJECT, and false if it has none.

**Syntax**

```
pbdom_document_name.HasChildren()
```

**Table 12.14:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object

**Return value**

Boolean.

Returns true if the current PBDOM\_DOCUMENT object has at least one child PBDOM\_OBJECT, and false otherwise.

### 12.1.12 HasRootElement

**Description**

Returns true if this document has a root element.

**Syntax**

```
pbdom_document_name.HasRootElement()
```

**Table 12.15:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object

**Return value**

Boolean.

Returns true if the current PBDOM\_DOCUMENT object has a root element, and false otherwise.

**See also**



[DetachRootElement](#)[GetRootElement](#)[SetRootElement](#)

### 12.1.13 InsertContent

#### Description

Inserts a new PBDOM\_OBJECT into the current PBDOM\_DOCUMENT object.

#### Syntax

```
pbdom_document_name.InsertContent(pbdom_object pbdom_object_new, pbdom_object
pbdom_object_ref)
```

**Table 12.16:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object
pbdom_object_new	The PBDOM_OBJECT to insert
pbdom_object_ref	The PBDOM_OBJECT in front of which the new PBDOM_OBJECT will be inserted

#### Return value

PBDOM\_OBJECT. The modified PBDOM\_DOCUMENT object returned as a PBDOM\_OBJECT.

#### Throws

EXCEPTION\_INVALID\_ARGUMENT -- The input PBDOM\_OBJECT to insert is invalid. This can happen if it has not been initialized properly or is a null object reference.

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- The input PBDOM\_OBJECT to insert has not been given a user-defined name. The same exception is thrown if the reference PBDOM\_OBJECT is also not given a user-defined name, unless the reference PBDOM\_OBJECT is specifically set to null.

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- The input PBDOM\_OBJECT to insert is not associated with a derived PBDOM\_OBJECT. The same exception is thrown if the reference PBDOM\_OBJECT is also not associated with a derived PBDOM\_OBJECT, unless the reference PBDOM\_OBJECT is specifically set to null.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT -- The input PBDOM\_OBJECT to insert already as a parent.

EXCEPTION\_MULTIPLE\_ROOT\_ELEMENT -- A PBDOM\_ELEMENT is to be inserted, but this document already has a root element.

EXCEPTION\_MULTIPLE\_DOCTYPE -- A PBDOM\_DOCTYPE is to be inserted, but this document already has a DOCTYPE.

EXCEPTION\_HIERARCHY\_ERROR -- Inserting the PBDOM\_OBJECT adversely affects how well-formed the document is.

**EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT** -- An invalid PBDOM\_OBJECT is to be inserted. See [AddContent](#) for information on the valid PBDOM\_OBJECTs that can be added to a PBDOM\_DOCUMENT object.

**EXCEPTION\_WRONG\_PARENT\_ERROR** -- The reference PBDOM\_OBJECT is not a child of this PBDOM\_DOCUMENT object.

### Examples

A PBDOM\_DOCUMENT object is created from an XML string. The PBDOM\_ELEMENT pbdom\_elem\_1 is also created and set as Elem\_1. The PBDOM\_DOCTYPE pbdom\_doctype\_1 and the root element pbdom\_root\_elem are set.

The root element is detached from its parent, which is also the PBDOM\_DOCUMENT object itself. This makes it possible to insert pbdom\_elem\_1 into the document specifically before pbdom\_doctype\_1.

```
pbdom_builder pbdom_builder_1
pbdom_document pbdom_doc
pbdom_doctype pbdom_doctype_1
pbdom_element pbdom_elem_1
pbdom_element pbdom_elem_root
string strXML

strXML = "<!DOCTYPE abc [<!-- internal subset -->"
strXML += "<!ELEMENT abc (#PCDATA)> "
strXML += "<!ELEMENT data&(#PCDATA)> "
strXML += "<!ELEMENT inner_data (#PCDATA)>]><abc>"
strXML += "Root Element Data<data>ABC Data<inner_data>"
strXML += "My Inner Data</inner_data>My Data</data>"
strXML += " now with extra& info</abc>"

pbdom_builder_1 = Create PBDOM_Builder
pbdom_elem_1 = Create PBDOM_Element

pbdom_doc = pbdom_builder_1.BuildFromString (strXML)
pbdom_elem_1.SetName ("Elem_1")
pbdom_doctype_1 = pbdom_doc.GetDocType()
pbdom_elem_root = pbdom_doc.GetRootElement()

pbdom_elem_root.Detach()
pbdom_doc.InsertContent(pbdom_elem_1, pbdom_doctype_1
```

The result is the following document, which is not well-formed:

```
<Elem_1/>
<!DOCTYPE abc[<!-- internal subset -->
<!ELEMENT abc (#PCDATA)*> <!ELEMENT data (#PCDATA)*> <!ELEMENT inner_data
(#PCDATA)*>]>
```

### Usage

When a new PBDOM\_OBJECT is inserted into the current PBDOM\_DOCUMENT object, the new PBDOM\_OBJECT becomes a child node of the current PBDOM\_DOCUMENT object. Also, the new PBDOM\_OBJECT is to be positioned specifically before another PBDOM\_OBJECT, denoted using the second parameter.

If the second PBDOM\_OBJECT is specified as null, then the new PBDOM\_OBJECT is to be inserted at the end of the list of children of the current PBDOM\_DOCUMENT object.

### See also

[AddContent](#)[GetContent](#)[RemoveContent](#)[SetContent](#)

### 12.1.14 IsAncestorObjectOf

#### Description

The IsAncestorObjectOf method determines whether the current PBDOM\_DOCUMENT object is the ancestor of another PBDOM\_OBJECT.

#### Syntax

```
pbdom_document_name.IsAncestorObjectOf(pbdom_object pbdom_object_ret)
```

**Table 12.17:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object
pbdom_object_ref	The PBDOM_OBJECT to check against

#### Return value

Boolean.

Returns true if the current PBDOM\_DOCUMENT object is the ancestor of the referenced PBDOM\_OBJECT, and false otherwise.

#### Throws

EXCEPTION\_INVALID\_ARGUMENT -- The input PBDOM\_OBJECT is invalid. This can happen if it has not been initialized properly or is a null object reference.

### 12.1.15 NewDocument

#### Description

The NewDocument method is overloaded:

- Syntax 1 creates a new XML DOM document using the name of the root element to be contained within the new DOM document.
- Syntax 2 creates a new XML DOM document using the name and namespace URI of the root element to be contained in the new DOM document, and also the external subset public and system identifiers.

#### Syntax

**Table 12.18:**

For this syntax	See
<code>NewDocument(string strRootElementName)</code>	<a href="#">NewDocument Syntax 1</a>

For this syntax	See
<pre>NewDocument( string strRootElementNamespacePrefix, string strRootElementNamespaceURI, string strRootElementName, string strDocTypePublicId, string strDocTypeSystemId)</pre>	<a href="#">NewDocument Syntax 2</a>

### 12.1.15.1 NewDocument Syntax 1

#### Description

Creates a new XML DOM document from scratch.

#### Syntax

```
pbdom_document_name.NewDocument(strRootElementName)
```

**Table 12.19:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object
strRootElementName	The name of the root element to be contained in the DOM document

#### Return value

Boolean.

Returns true if a new document is successfully created and false otherwise.

#### Throws

EXCEPTION\_INVALID\_ARGUMENT -- The input string is invalid, which can occur if the string was set to null by means of the PowerScript SetNull method.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- Insufficient memory was encountered while executing this method.

#### Usage

The parameter strRootElementName becomes the name of the root element.

#### See also

[SaveDocument](#)

### 12.1.15.2 NewDocument Syntax 2

#### Description

Creates a new XML DOM document from scratch.

#### Syntax

```
pbdom_document_name.NewDocument(string strRootElementNamespacePrefix, string
strRootElementNamespaceURI, string strRootElementName, string strDocTypePublicId,
string strDocTypeSystemId)
```

**Table 12.20:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object.
strRootElementNamespacePrefix	The namespace prefix of the root element to be contained in the DOM document. This can be an empty string.
strRootElementNamespaceURI	The namespace URI of the root element to be contained in the DOM document. This can be an empty string.
strRootElementName	The name of the root element to be contained in the DOM document.
strDocTypePublicId	The external subset public identifier.
strDocTypeSystemId	The external subset system identifier.

**Return value**

Boolean.

Returns true if a new document is successfully created, and false otherwise.

**Throws**

EXCEPTION\_INVALID\_ARGUMENT -- One of the input strings is invalid. This can happen if the string has been set to null using the PowerScript SetNull method.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- Insufficient memory was encountered while executing this method.

EXCEPTION\_INVALID\_NAME -- The root element name, or the root element namespace prefix or URI, is invalid.

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_DOCUMENT object's internal implementation is NULL. The occurrence of this exception is rare but can take place if severe memory corruption occurs.

**Examples**

1. This example attempts to create a PBDOM\_DOCUMENT object in which the root element belongs to no namespace, as indicated by the empty strings for the namespace prefix and URI arguments to NewDocument:

```
PBDOM_DOCUMENT pbdom_doc

try
    pbdom_doc = Create PBDOM_DOCUMENT
    pbdom_doc.NewDocument ("", "", "root", "public_id", &
        "system_id.dtd")

    pbdom_doc.SaveDocument &
        ("new_document_no_namespace.xml")

catch (PBDOM_EXCEPTION except)
    MessageBox ("PBDOM_EXCEPTION", except.GetMessage())
```

```
end try
```

When serialized, the XML document looks like the following:

```
<!DOCTYPE root PUBLIC "public_id" "system_id.dtd">
<root xmlns="" />
```

The namespace declaration attribute (`xmlns=""`) present in the root element indicates that the root element belongs to no namespace.

2. This example attempts to create a `PBDOM_DOCUMENT` object in which the root element belongs to a default namespace. The URI is `http://www.pre.com`, which means that the root element belongs to the namespace `http://www.pre.com`. The prefix is an empty string, which means that the root element belongs to the `http://www.pre.com` namespace by default:

```
PBDOM_DOCUMENT pbdom_doc

try
    pbdom_doc = Create PBDOM_DOCUMENT
    pbdom_doc.NewDocument ("", "http://www.pre.com", &
        "root", "public_id", "system_id.dtd")

    pbdom_doc.SaveDocument &
        ("new_document_default_namespace.xml")

catch (PBDOM_EXCEPTION except)
    MessageBox ("PBDOM_EXCEPTION", except.GetMessage())
end try
```

When serialized, the XML document looks like the following:

```
<!DOCTYPE root PUBLIC "public_id" "system_id.dtd">
<root xmlns="http://www.pre.com" />
```

The namespace declaration attribute (`xmlns="http://www.pre.com"`) present in the root element indicates that the root element belongs to the default namespace `http://www.pre.com`. All child elements of root belong to this same namespace unless another in-scope namespace declaration is present and is used.

3. This example attempts to create a `PBDOM_DOCUMENT` object in which the root element belong to a prefixed namespace. The namespace prefix is `pre` and the URI is `http://www.pre.com`. This means that the root element will belong to the namespace `http://www.pre.com`, and that the root element will have a namespace prefix of `pre`:

```
PBDOM_DOCUMENT pbdom_doc

try
    pbdom_doc = Create PBDOM_DOCUMENT
    pbdom_doc.NewDocument ("pre", "http://www.pre.com", &
        "root", "public_id", "system_id.dtd")

    pbdom_doc.SaveDocument &
        ("new_document_namespace.xml")

catch (PBDOM_EXCEPTION except)
    MessageBox ("PBDOM_EXCEPTION", except.GetMessage())
end try
```

When serialized, the XML document looks like the following:

```
<!DOCTYPE pre:root PUBLIC "public_id" "system_id.dtd">
<pre:root xmlns:pre="http://www.pre.com"/>
```

A namespace declaration attribute (`xmlns:pre="http://www.pre.com"`) is present in the root element. The root element also contains a `pre` prefix. This indicates that the root element belongs to the namespace `http://www.pre.com`.

However, the fact that the `http://www.pre.com` namespace is prefixed by `pre` indicates that the child elements of root belong to this same namespace only if their qualified names also contain the `pre` prefix and there is an in-scope namespace declaration for `http://www.pre.com` that is prefixed by `pre`.

## Usage

Using the five parameters available with this syntax provides more control over the DOCTYPE definition of the document.

## See also

[SaveDocument](#)

### 12.1.16 RemoveContent

#### Description

Removes a child PBDOM\_OBJECT from the current PBDOM\_DOCUMENT object.

#### Syntax

```
pbdom_document_name.RemoveContent(pbdom_object pbdom_object_ref)
```

**Table 12.21:**

Argument	Description
<code>pbdom_document_name</code>	The name of a PBDOM_DOCUMENT object
<code>pbdom_object_ref</code>	The PBDOM_OBJECT to remove

#### Return value

Boolean.

Returns true if the content was removed, and false otherwise.

#### Throws

**EXCEPTION\_INVALID\_ARGUMENT** -- The input PBDOM\_OBJECT to remove is invalid. This can happen if it has not been initialized properly or is a null object reference.

**EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT** -- The input PBDOM\_OBJECT is nameable, but it has not been assigned a name.

**EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE** -- The input PBDOM\_OBJECT is not associated with a derived PBDOM\_OBJECT class object.

**EXCEPTION\_WRONG\_DOCUMENT\_ERROR** -- The input PBDOM\_OBJECT is not contained within the current PBDOM\_DOCUMENT object.

EXCEPTION\_WRONG\_PARENT\_ERROR -- The input PBDOM\_OBJECT is not a child of the current PBDOM\_DOCUMENT object.

### Usage

When a PBDOM\_OBJECT is removed from the current PBDOM\_DOCUMENT object, all children under the removed PBDOM\_OBJECT are also removed.

### See also

[AddContent](#)

[GetContent](#)

[InsertContent](#)

[SetContent](#)

## 12.1.17 SaveDocument

### Description

Saves the serialized XML string of the DOM tree contained within the PBDOM\_DOCUMENT object into a disk file.

### Syntax

```
pbdom_document_name.SaveDocument(string strFileName)
```

**Table 12.22:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object
strFileName	The name of the disk file to which the contents of the current PBDOM_DOCUMENT object is to be serialized

### Return value

Boolean.

Returns true if a new document was successfully saved to a disk file, and false otherwise.

### Throws

EXCEPTION\_INVALID\_ARGUMENT -- The input string specifying the file name is invalid. This can happen if the string has been set to null using the PowerShell SetNull method.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- Insufficient memory was encountered while executing this method.

### Usage

A PBDOM\_DOCUMENT object that has been created from an existing XML document or string can differ from its original after it has been converted back to an XML



string or document. This can occur even if no modifications have been made to the PBDOM\_DOCUMENT object using PowerScript.

This can occur if the original XML document or string referred to an external DTD that mandates the inclusion of default attributes. In this case, PBDOM complies with the rules of the DTD and inserts these required attributes into the relevant elements while building up the in-memory DOM tree.

When the PBDOM\_DOCUMENT object is saved and converted back to an XML document, these default attributes are saved in the document.

### See also

[NewDocument](#)

## 12.1.18 SaveDocumentIntoString

### Description

Saves the serialized XML string of the DOM tree contained within the PBDOM\_DOCUMENT object into a string.

### Syntax

```
pbdom_document_name.SaveDocumentIntoString( )
```

**Table 12.23:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object

### Return value

String.

Returns a string containing the XML string of the PBDOM\_DOCUMENT.

### Examples

This code creates a new PBDOM\_DOCUMENT and saves it to the string ls\_xml:

```
PBDOM_DOCUMENT pbdom_doc
string ls_xml

try
  pbdom_doc = Create PBDOM_DOCUMENT
  pbdom_doc.NewDocument ("pre", "http://www.pre.com", &
    "root", "public_id", "system_id.dtd")
  ls_xml = pbdom_doc.SaveDocumentIntoString
catch (PBDOM_EXCEPTION except)
  MessageBox ("PBDOM_EXCEPTION", except.GetMessage())
end try
```

### See also

[SaveDocument](#)

## 12.1.19 SetContent

### Description

Sets the entire content of the PBDOM\_DOCUMENT object, removing pre-existing children first.

### Syntax

```
pbdom_document_name.SetContent(pbdom_object pbdom_object_array)
```

**Table 12.24:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object
pbdom_object_array	An array of PBDOM_OBJECTs set as the contents of the PBDOM_DOCUMENT object

pbdom\_object\_array must contain only PBDOM\_OBJECT objects that can legally be set as the contents of a PBDOM\_DOCUMENT object. The SetContent method restricts the array to one PBDOM\_ELEMENT object to set as the root element of the PBDOM\_DOCUMENT object from which the method is invoked. The SetContent method also restricts the array to one PBDOM\_DOCTYPE object to set as the DOCTYPE of the PBDOM\_DOCUMENT object.

### Return value

PBDOM\_OBJECT. The modified PBDOM\_DOCUMENT object returned as a PBDOM\_OBJECT.

### Throws

EXCEPTION\_ILLEGAL\_PBOBJECT -- An array item is not a valid PBDOM object. This can happen if the array item has not been initialized properly or is a null object reference.

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- An array item is nameable and has not been given a user-defined name.

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- An array item is not associated with a derived PBDOM\_OBJECT.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT -- An array item already has a parent PBDOM\_OBJECT.

EXCEPTION\_MULTIPLE\_ROOT\_ELEMENT -- The array contains more than one PBDOM\_ELEMENT. The array must contain at most one PBDOM\_ELEMENT that is set as the root element of this document.

EXCEPTION\_MULTIPLE\_DOCTYPE -- The array contains more than one PBDOM\_DOCTYPE. The array must contain at most one PBDOM\_DOCTYPE that is set as the DOCTYPE of this document.

EXCEPTION\_MULTIPLE\_XMLDECL -- The array contains more than one PBDOM\_PROCESSINGINSTRUCTION that has been constructed into an XML Declaration.

EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT -- An array item is not allowed to be set as a document-level content.

## Usage

The supplied array contains PBDOM\_OBJECTs that can legally be set as the content of a PBDOM\_DOCUMENT object.

For example, a PBDOM\_DOCUMENT object accepts only an array that contains PBDOM\_ELEMENT, PBDOM\_COMMENT, PBDOM\_DOCTYPE, or PBDOM\_PROCESSINGINSTRUCTION objects. In addition, the array can contain at most one PBDOM\_ELEMENT object that it sets as its root element, at most one PBDOM\_DOCTYPE object that it sets as its DOCTYPE, and at most one XML declaration .PBDOM\_PROCESSINGINSTRUCTION.

In the event of an exception, the original contents of this PBDOM\_DOCUMENT object are unchanged, and the PBDOM\_OBJECTs contained in the supplied array are unaltered.

## See also

[AddContent](#)

[GetContent](#)

[InsertContent](#)

[RemoveContent](#)

### 12.1.20 SetDocType

#### Description

Sets the DOCTYPE declaration of this document.

#### Syntax

```
pbdom_document_name.SetDocType(pbdom_doctype pbdom_doctype_ref)
```

**Table 12.25:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object
pbdom_doctype_ref	A PBDOM_DOCTYPE object to be set as the DOCTYPE of this document

#### Return value

PBDOM\_DOCUMENT. The same PBDOM\_DOCUMENT object with a modified DOCTYPE declaration.

#### Throws

EXCEPTION\_INVALID\_ARGUMENT -- The input PBDOM\_DOCTYPE is invalid. This can happen if it has not been initialized properly or is a null object reference.

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- The input PBDOM\_DOCTYPE is nameable and has not been given a user-defined name.

EXCEPTION\_WRONG\_DOCUMENT\_ERROR -- The input PBDOM\_DOCTYPE already has an owner document.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT -- The input PBDOM\_DOCTYPE is already the DOCTYPE of another document.

### Usage

If this document already contains a DOCTYPE declaration, the new PBDOM\_DOCTYPE replaces it. The DOCTYPE of a PBDOM\_DOCUMENT object can be changed multiple times, and it is legal for a user to call the SetDocType method multiple times.

A DOM DOCTYPE object can have no owner document, or it can have an owner document but no parent node. A DOCTYPE that has an owner document as well as a parent node is the actual DOCTYPE of the owner document.

## 12.1.21 SetRootElement

### Description

Sets the root element for this document.

### Syntax

```
pbdom_document_name.SetRootElement(pbdom_element pbdom_element_ref)
```

**Table 12.26:**

Argument	Description
pbdom_document_name	The name of a PBDOM_DOCUMENT object
pbdom_element_ref	A PBDOM_ELEMENT object to be set as the root element for this document

### Return value

PBDOM\_DOCUMENT. The PBDOM\_DOCUMENT object with a modified root element.

### Throws

EXCEPTION\_INVALID\_ARGUMENT -- The input PBDOM\_ELEMENT is invalid. This can happen if it has not been initialized properly or is a null object reference.

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- The input PBDOM\_ELEMENT is nameable and it has not been given a user-defined name.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT -- The input PBDOM\_ELEMENT already has a parent PBDOM\_OBJECT.

### Usage

If this document already has a root element, the existing root element is replaced. The root element of a PBDOM\_DOCUMENT object can be changed multiple times, and it is legal for a user to call the SetRootElement method multiple times.

### See also

[DetachRootElement](#)

[GetRootElement](#)

[HasRootElement](#)

## 13 PBDOM\_ELEMENT Class

### About this chapter

This chapter describes the PBDOM\_ELEMENT class.

### 13.1 PBDOM\_ELEMENT

#### Description

The PBDOM\_ELEMENT class defines the behavior for an XML element modeled in PowerScript. Methods allow the user to obtain the text content of an element, the attributes of an element, and the children of an element.

In PBDOM, an XML element's attributes are not its children. Attributes are properties of elements rather than having a separate identity from the elements with which they are associated. An element's PBDOM\_ATTRIBUTE objects do not have sibling relationships with each other in the same way as the element's children.

For more information on the relationships among PBDOM\_ELEMENT and PBDOM\_ATTRIBUTE objects, see the chapter on XML services in Application Techniques.

#### Methods

PBDOM\_ELEMENT has the following methods:

**Table 13.1:**

<a href="#">AddContent</a>	<a href="#">GetTextTrim</a>
<a href="#">AddNamespaceDeclaration</a>	<a href="#">HasAttributes</a>
<a href="#">Clone</a>	<a href="#">HasChildElements</a>
<a href="#">Detach</a>	<a href="#">HasChildren</a>
<a href="#">Equals</a>	<a href="#">InsertContent</a>
<a href="#">GetAttribute</a>	<a href="#">IsAncestorObjectOf</a>
<a href="#">GetAttributes</a>	<a href="#">IsRootElement</a>
<a href="#">GetAttributeValue</a>	<a href="#">RemoveAttribute</a>
<a href="#">GetChildElement</a>	<a href="#">RemoveChildElement</a>
<a href="#">GetChildElements</a>	<a href="#">RemoveChildElements</a>
<a href="#">GetContent</a>	<a href="#">RemoveContent</a>
<a href="#">GetName</a>	<a href="#">RemoveNamespaceDeclaration</a>
<a href="#">GetNamespacePrefix</a>	<a href="#">SetAttribute</a>
<a href="#">GetNamespaceUri</a>	<a href="#">SetAttributes</a>
<a href="#">GetObjectClass</a>	<a href="#">SetContent</a>
<a href="#">GetObjectClassString</a>	<a href="#">SetDocument</a>
<a href="#">GetOwnerDocumentObject</a>	<a href="#">SetName</a>
<a href="#">GetParentObject</a>	<a href="#">SetNamespace</a>

<a href="#">GetQualifiedName</a>	<a href="#">SetParentObject</a>
<a href="#">GetText</a>	<a href="#">SetText</a>
<a href="#">GetTextNormalize</a>	

### 13.1.1 AddContent

#### Description

The AddContent method is overloaded:

- Syntax 1 adds a new PBDOM\_OBJECT into a PBDOM\_ELEMENT object.
- Syntax 2 adds a new text string to the PBDOM\_ELEMENT object from which the method is invoked.

#### Syntax

**Table 13.2:**

For this syntax	See
AddContent(pbdom_object pbdom_object_ref)	<a href="#">AddContent Syntax 1</a>
AddContent(string strText)	<a href="#">AddContent Syntax 2</a>

#### 13.1.1.1 AddContent Syntax 1

#### Description

Adds a new PBDOM\_OBJECT into a PBDOM\_ELEMENT object. The added PBDOM\_OBJECT becomes a child of the PBDOM\_ELEMENT object.

#### Syntax

```
pbdom_element_name.AddContent(pbdom_object pbdom_object_ref)
```

**Table 13.3:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
pbdom_object_ref	The PBDOM_OBJECT to add

#### Return value

PBDOM\_OBJECT. The PBDOM\_ELEMENT object modified and returned as a PBDOM\_OBJECT.

#### Throws

EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT -- If an invalid PBDOM\_OBJECT is added. See description section below on the valid PBDOM\_OBJECTs that can be added to a PBDOM\_ELEMENT object. This exception is also thrown if the input PBDOM\_OBJECT is this PBDOM\_ELEMENT object itself.

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- If the input PBDOM\_OBJECT has not been given a user-defined name.

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the input PBDOM\_OBJECT is not associated with a derived PBDOM\_OBJECT.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT -- If the input PBDOM\_OBJECT already has a parent PBDOM\_OBJECT.

EXCEPTION\_HIERARCHY\_ERROR -- If adding the input PBDOM\_OBJECT will cause the current PBDOM\_ELEMENT object to be no longer well-formed.

### Examples

The AddContent method is invoked for the Element\_2 PBDOM\_ELEMENT object in the following XML fragment:

```
<Element_1>
  <Element_1_1/>
  <Element_1_2/>
  <Element_1_3/>
</Element_1>
<Element_2>Element 2 Text</Element_2>
<Element_3/>
```

The AddContent is invoked from the following PowerScript code, where pbdom\_elem\_2 represents the Element\_2 object:

```
PBDOM_ELEMENT pbdom_elem
pbdom_elem = Create PBDOM_ELEMENT
pbdom_elem.SetName("Sub_Element")
pbdom_elem.AddContent("Sub Element Text")
pbdom_elem_2.AddContent (pbdom_elem)
```

The following XML fragment results:

```
<Element_1>
  <Element_1_1/>
  <Element_1_2/>
  <Element_1_3/>
</Element_1>
<Element_2>
  Element 2 Text
  <Sub_Element>
    Sub Element Text
  </Sub_Element>
</Element_2/>
<Element_3/>
```

### Usage

Only the following PBDOM\_OBJECT types can be validly added to a PBDOM\_ELEMENT object:

- PBDOM\_ELEMENT
- PBDOM\_CDATA
- PBDOM\_COMMENT
- PBDOM\_ENTITYREFERENCE
- PBDOM\_PROCESSINGINSTRUCTION

- PBDOM\_TEXT

### See also

[AddContent Syntax 2](#)

[GetContent](#)

[InsertContent](#)

[RemoveContent](#)

[SetContent](#)

### 13.1.1.2 AddContent Syntax 2

#### Description

Adds a new text string to the PBDOM\_ELEMENT object from which the method is invoked.

#### Syntax

```
pbdom_element_name.AddContent(string strText)
```

**Table 13.4:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strText	A string to be added to the PBDOM_ELEMENT object as new text content

#### Return value

PBDOM\_OBJECT. The PBDOM\_ELEMENT object modified and returned as a PBDOM\_OBJECT.

#### Examples

The AddContent method is invoked for the abc element of the following XML document:

```
<abc>
  Root Element Data
  <data>
    ABC Data
    <inner_data>My Inner Data</inner_data>
  </data>
</abc>
```

The AddContent method is invoked from the following PowerShell statement:

```
pbdom_doc.GetRootElement().AddContent(" And More !")
```

The following XML results:

```
<abc>
  Root Element Data
  <data>
    ABC Data
    <inner_data>My Inner Data</inner_data>
  </data>
```



```
And More !
</abc>
```

**See also**[AddContent Syntax 1](#)[GetContent](#)[InsertContent](#)[RemoveContent](#)[SetContent](#)**13.1.2 AddNamespaceDeclaration****Description**

Adds a new namespace declaration to this PBDOM\_ELEMENT object. The new namespace can apply to the PBDOM\_ELEMENT object itself if the namespace becomes the default namespace in the PBDOM\_ELEMENT object.

**Syntax**

```
pbdom_element_name.AddNamespaceDeclaration(string strNamespacePrefix, string
strNamespaceUri)
```

**Table 13.5:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strNamespacePrefix	The prefix of the new namespace to be declared
strNamespaceUri	The URI of the new namespace to be declared

**Return value**

PBDOM\_ELEMENT. The modified PBDOM\_ELEMENT object.

**Throws**

EXCEPTION\_INVALID\_ARGUMENT -- If any of the input parameters is invalid (null).

EXCEPTION\_INVALID\_NAME -- If the input Prefix is invalid, as, for example, if it contains a colon.

EXCEPTION\_INVALID\_STRING -- If the input URI is invalid.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- If memory allocation failure occurred in this method.

**Examples**

Consider the following element:

```
<Vehicle>
  <seats>4</seats>
  <color>Red</color>
```

```
<engine>
  <capacity units="cc">1600</capacity>
</engine>
</Vehicle>
```

Given a PBDOM\_ELEMENT object `elem_vehicle` that represents the `Vehicle` element, consider the following statement:

```
elem_vehicle.AddNamespaceDeclaration("vehicle_specs", &
  "http://www.vehicle.com/specs")
```

It transforms the `Vehicle` element as follows:

```
<Vehicle xmlns:vehicle_specs="http://www.vehicle.com/specs">
  <seats>4</seats>
  <color>Red</color>
  <engine>
    <capacity units="cc">1600</capacity>
  </engine>
</Vehicle>
```

`Vehicle`, `seats`, `color`, `engine`, and `capacity` are all unqualified (that is, they have no namespace prefix). Therefore, the `vehicle_specs` namespace does not apply to any of them or their attributes or subelements.

However, consider the following statement:

```
elem_vehicle.AddNamespaceDeclaration("", &
  "http://www.vehicle.com/specs")
```

It transforms the `Vehicle` element as follows:

```
<Vehicle xmlns:"http://www.vehicle.com/specs">
  <seats>4</seats>
  <color>Red</color>
  <engine>
    <capacity units="cc">1600</capacity>
  </engine>
</Vehicle>
```

`http://www.vehicle.com/specs` is the default namespace and so `Vehicle`, `seats`, `color`, `engine`, and `capacity` are all part of this namespace. Note that the default namespace does not apply to the `units` attribute.

### See also

[GetNamespacePrefix](#)

[GetNamespaceUri](#)

[GetQualifiedName](#)

[RemoveNamespaceDeclaration](#)

[SetNamespace](#)

## 13.1.3 Clone

### Description

Creates a clone of a PBDOM\_ELEMENT object.

### Syntax

```
pbdom_element_name.Clone(boolean bDeep)
```

**Table 13.6:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object.
bDeep	A boolean specifying whether a deep or shallow clone is returned. Values are true for a deep clone and false for a shallow clone.

**Return value**

PBDOM\_OBJECT. A clone of this PBDOM\_ELEMENT object returned as a PBDOM\_OBJECT.

**Examples**

The Clone method is used to alter the following XML:

```
<Telephone_Book>
  <Entry>
    <Particulars>
      <Name>John Doe</Name>
      <Age>21</Age>
      <Phone_Number>1234567</Phone_Number>
    </Particulars>
  </Entry>
</Telephone_Book>
```

The Clone method is invoked from the following PowerShell code, where entry represents the Entry> element in the preceding XML:

```
PBDOM_ELEMENT elem_clone

elem_clone = entry.Clone(true)
pbdom_doc.AddContent(elem_clone)
```

The resulting XML contains two identical Entry> elements:

```
<Telephone_Book>
  <Entry>
    <Particulars>
      <Name>John Doe</Name>
      <Age>21</Age>
      <Phone_Number>1234567</Phone_Number>
    </Particulars>
  </Entry>
  <Entry>
    <Particulars>
      <Name>John Doe</Name>
      <Age>21</Age>
      <Phone_Number>1234567</Phone_Number>
    </Particulars>
  </Entry>
</Telephone_Book>
```

**Usage**

This method creates and returns a duplicate of the current PBDOM\_ELEMENT object. If a shallow clone is requested, this method clones the PBDOM\_ELEMENT object together

with its namespace information values and its PBDOM\_ATTRIBUTES and their subtrees. If a deep clone is requested, this method additionally recursively clones the subtree under the PBDOM\_ELEMENT object.

A PBDOM\_ELEMENT clone has no parent. However, the clone resides in the same PBDOM\_DOCUMENT as its original, and if the original PBDOM\_ELEMENT object is standalone, the clone is standalone.

### 13.1.4 Detach

#### Description

Detaches a PBDOM\_ELEMENT object from its parent PBDOM\_OBJECT.

#### Syntax

```
pbdom_element_name.Detach()
```

**Table 13.7:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object

#### Return value

PBDOM\_OBJECT. The PBDOM\_ELEMENT object detached from its parent object and returned as a PBDOM\_OBJECT. If the PBDOM\_ELEMENT object has no parent, the Detach method does nothing.

### 13.1.5 Equals

#### Description

Tests for equality between the PBDOM\_ELEMENT object from which the method is invoked and a PBDOM\_OBJECT indicated by the method parameter.

#### Syntax

```
pbdom_element_name.Equals(pbdom_object pbdom_object_ref)
```

**Table 13.8:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
pbdom_object_ref	A PBDOM_OBJECT to be tested for equality with this PBDOM_ELEMENT object

#### Return value

Boolean.

Returns true if the PBDOM\_ELEMENT object is equivalent to the referenced PBDOM\_OBJECT and false otherwise.

#### Examples

The Equals method is invoked from the following PowerScript code, in which pbdom\_doc represents a PBDOM\_DOCUMENT object containing a root element:

```
PBDOM_ELEMENT pbdom_elem_1
PBDOM_ELEMENT pbdom_elem_2
PBDOM_OBJECT pbdom_obj
PBDOM_DOCUMENT pbdom_doc

pbdom_elem_1 = pbdom_doc.GetRootElement()
pbdom_elem_2 = pbdom_doc.GetRootElement()

IF pbdom_elem_1.Equals(pbdom_elem_2) THEN
    MessageBox ("Equals", "The objects are equal")
ELSE
    MessageBox ("Equals", "The objects are NOT equal")
END IF

pbdom_obj = Create PBDOM_ELEMENT
pbdom_obj.SetName("An_Element")

IF pbdom_elem_1.Equals(pbdom_obj) THEN
    MessageBox ("Equals", "The objects are equal")
ELSE
    MessageBox ("Equals", "The objects are NOT equal")
END IF
```

Because pbdom\_elem\_1 and pbdom\_elem\_2 refer to the same root element, a message box reports that the objects are equal.

### 13.1.6 GetAttribute

#### Description

The GetAttribute method is overloaded:

- Syntax 1 returns the PBDOM\_ATTRIBUTE object for a PBDOM\_ELEMENT object using the name of the PBDOM\_ATTRIBUTE.
- Syntax 2 returns the PBDOM\_ATTRIBUTE object for a PBDOM\_ELEMENT object with the name provided and within the namespace specified by the prefix and URI provided.

#### Syntax

**Table 13.9:**

For this syntax	See
GetAttribute(string strName)	<a href="#">GetAttribute Syntax 1</a>
GetAttribute(string strName, string strNamespacePrefix, string strNamespaceUri)	<a href="#">GetAttribute Syntax 2</a>

#### 13.1.6.1 GetAttribute Syntax 1

#### Description

Returns the PBDOM\_ATTRIBUTE object for a PBDOM\_ELEMENT object.

#### Syntax

```
pbdom_element_name.GetAttribute(string strName)
```

**Table 13.10:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strName	The name of the PBDOM_ATTRIBUTE to be returned

**Return value**

PBDOM\_ATTRIBUTE. The PBDOM\_ATTRIBUTE object matching the name specified in the method parameter. If no such PBDOM\_ATTRIBUTE object exists, the GetAttribute method returns a value of null.

**Throws**

EXCEPTION\_INVALID\_NAME -- If the supplied name is a qualified name that contains a namespace prefix.

**Examples**

The GetAttribute method is invoked for the following XML document:

```
<MyMusic:abc xmlns:MyMusic="http://www.MyMusic_records.com" My_Attr="My MyMusic Attribute">Root Element Data</MyMusic:abc>
```

The GetAttribute method is invoked from the following PowerShell statement:

```
pbdom_attr = &
    pbdom_doc.GetRootElement().GetAttribute("My_Attr")
```

The GetAttribute method returns the PBDOM\_ATTRIBUTE object My\_Attr.

**Usage**

If the PBDOM\_ATTRIBUTE name specified in the method parameter is a qualified name, an exception is thrown. A qualified name appears in the following form: [namespace\_prefix]:[local\_name].

**See also**

[GetAttribute Syntax 2](#)

[GetAttributes](#)

[GetAttributeValue](#)

[HasAttributes](#)

[SetAttribute](#)

[SetAttributes](#)

**13.1.6.2 GetAttribute Syntax 2****Description**

Returns the PBDOM\_ATTRIBUTE object for a PBDOM\_ELEMENT object with the name provided and within the namespace specified by the prefix and URI provided.

## Syntax

```
pbdom_element_name.GetAttribute(string strName, string strNamespacePrefix, string strNamespaceUri)
```

**Table 13.11:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strName	The name of the PBDOM_ATTRIBUTE to be returned
strNamespacePrefix	The prefix of the namespace of the PBDOM_ATTRIBUTE to return
strNamespaceUri	The URI of the namespace of the PBDOM_ATTRIBUTE to return

## Return value

PBDOM\_ATTRIBUTE. The PBDOM\_ATTRIBUTE object matching the name, namespace prefix, and URI specified in the method parameters. If no such PBDOM\_ATTRIBUTE object exists, the GetAttribute method returns a value of null.

## Throws

EXCEPTION\_INVALID\_ARGUMENT -- If any of the arguments is invalid, for example, null.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- If there was any memory allocation failure during the running of this method.

## See also

[GetAttribute Syntax 1](#)

[GetAttributes](#)

[GetAttributeValue](#)

[HasAttributes](#)

[SetAttribute](#)

[SetAttributes](#)

## 13.1.7 GetAttributes

### Description

Returns the complete set of PBDOM\_ATTRIBUTE objects for a PBDOM\_ELEMENT object.

If there are no PBDOM\_ATTRIBUTE objects for the PBDOM\_ELEMENT object, the GetAttributes method returns an empty array.

### Syntax

```
pbdom_element_name.GetAttributes(ref pbdom_attribute pbdom_attribute_array)
```

**Table 13.12:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
pbdom_attribute_array	An empty and unbounded array to be filled with references to the PBDOM_ATTRIBUTE objects contained in the PBDOM_ELEMENT object

**Return value**

Boolean.

Returns true if an array of PBDOM\_ATTRIBUTE objects for the PBDOM\_ELEMENT object has been retrieved, and false otherwise.

**Usage**

GetAttributes returns the complete set of PBDOM\_ATTRIBUTE objects for a PBDOM\_ELEMENT object as an array of PBDOM\_ATTRIBUTE objects, or as an empty list (empty array) if there are none. The returned array items are "live" and changes to any item affect the referenced PBDOM\_ATTRIBUTE.

**See also**

[GetAttribute](#)

[GetAttributeValue](#)

[HasAttributes](#)

[SetAttribute](#)

[SetAttributes](#)

**13.1.8 GetAttributeValue****Description**

The GetAttributeValue method is overloaded:

- Syntax 1 returns the string value of a PBDOM\_ATTRIBUTE object with the specified name.
- Syntax 2 returns the string value of a PBDOM\_ATTRIBUTE object with the specified name, using the prefix and URI of the namespace of the PBDOM\_ATTRIBUTE.
- Syntax 3 returns the string value of a PBDOM\_ATTRIBUTE object with the specified name, using the prefix and URI of the namespace of the PBDOM\_ATTRIBUTE. Syntax 3 also provides a default string value to return if the attribute does not exist.
- Syntax 4 returns the string value of a PBDOM\_ATTRIBUTE object with the specified name. Syntax 4 also provides a default string value to return if the attribute does not exist.

**Syntax**



**Table 13.13:**

For this syntax	See
<code>GetAttributeValue(string strAttributeName)</code>	<a href="#">GetAttributeValue Syntax 1</a>
<code>GetAttributeValue(string strAttributeName, string strNamespacePrefix, string strNamespaceUri)</code>	<a href="#">GetAttributeValue Syntax 2</a>
<code>GetAttributeValue(string strAttributeName, string strNamespacePrefix, string strNamespaceUri, string strDefaultValue)</code>	<a href="#">GetAttributeValue Syntax 3</a>
<code>GetAttributeValue(string strAttributeName, string strDefaultValue)</code>	<a href="#">GetAttributeValue Syntax 4</a>

**13.1.8.1 GetAttributeValue Syntax 1****Description**

Returns the string value of the PBDOM\_ATTRIBUTE object (within a PBDOM\_ELEMENT object) with the specified name and within no namespace.

**Syntax**

```
pbdom_element_name.GetAttributeValue(string strAttributeName)
```

**Table 13.14:**

Argument	Description
<code>pbdom_element_name</code>	The name of a PBDOM_ELEMENT object
<code>strAttributeName</code>	The name of the attribute whose value is to be returned

**Return value**

String.

The string value of the PBDOM\_ATTRIBUTE object specified in `strAttributeName`. If no such object exists, the `GetAttributeValue` method returns null.

**Usage**

If the text value of the PBDOM\_ATTRIBUTE object is empty, the `GetAttributeValue` method returns an empty string.

**See also**

[GetAttribute](#)

[GetAttributeValue Syntax 2](#)

[GetAttributeValue Syntax 3](#)

[GetAttributeValue Syntax 4](#)

[HasAttributes](#)

[SetAttribute](#)[SetAttributes](#)

### 13.1.8.2 GetAttributeValue Syntax 2

#### Description

Returns the string value of the PBDOM\_ATTRIBUTE object (within a PBDOM\_ELEMENT object) with the specified name and within the specified namespace.

#### Syntax

```
pbdom_element_name.GetAttributeValue( string strAttributeName, string  
strNamespacePrefix, string strNamespaceUri)
```

**Table 13.15:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strAttributeName	The name of the attribute whose value is to be returned
strNamespacePrefix	The prefix of the namespace of the PBDOM_ATTRIBUTE whose value is to be returned
strNamespaceUri	The URI of the namespace of the PBDOM_ATTRIBUTE whose value is to be returned

#### Return value

String.

The string value of the PBDOM\_ATTRIBUTE object specified in strAttributeName. If no such object exists, the GetAttributeValue method returns an empty string.

#### Throws

EXCEPTION\_INVALID\_ARGUMENT -- If any of the input arguments is invalid, for example, null.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- If there was any memory allocation failure during the execution of this method.

EXCEPTION\_INVALID\_NAME -- If the input attribute name or namespace prefix or namespace URI is invalid.

#### See also

[GetAttribute](#)[GetAttributeValue Syntax 1](#)[GetAttributeValue Syntax 3](#)[GetAttributeValue Syntax 4](#)[HasAttributes](#)

[SetAttribute](#)[SetAttributes](#)

### 13.1.8.3 GetAttributeValue Syntax 3

#### Description

Returns the string value of the PBDOM\_ATTRIBUTE object (within a PBDOM\_ELEMENT object) with the specified name and within the specified namespace. If no such PBDOM\_ATTRIBUTE exists, the default value is returned.

#### Syntax

```
pbdom_element_name.GetAttributeValue( string strAttributeName, string
strNamespacePrefix, string strNamespaceUri, string strDefaultValue)
```

**Table 13.16:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strAttributeName	The name of the attribute whose value is to be returned
strNamespacePrefix	The prefix of the namespace of the PBDOM_ATTRIBUTE whose value is to be returned
strNamespaceUri	The URI of the namespace of the PBDOM_ATTRIBUTE whose value is to be returned
strDefaultValue	Default string value to return if the attribute does not exist

#### Return value

String.

The string value of the PBDOM\_ATTRIBUTE object specified in strAttributeName. If no such object exists, the GetAttributeValue method returns the string provided in strDefaultValue.

#### Throws

EXCEPTION\_INVALID\_ARGUMENT -- If any of the input arguments is invalid, for example, null.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- If there was any memory allocation failure during the execution of this method.

EXCEPTION\_INVALID\_NAME -- If the input attribute name or namespace prefix or namespace URI is invalid.

#### See also

[GetAttribute](#)[GetAttributeValue Syntax 1](#)

[GetAttributeValue Syntax 2](#)[GetAttributeValue Syntax 4](#)[HasAttributes](#)[SetAttribute](#)[SetAttributes](#)

#### 13.1.8.4 GetAttributeValue Syntax 4

##### Description

Returns the string value of the PBDOM\_ATTRIBUTE object (within a PBDOM\_ELEMENT object) with the specified name. If no such PBDOM\_ATTRIBUTE exists, the default value is returned.

##### Syntax

```
pbdom_element_name.GetAttributeValue(string strAttributeName, string strDefaultValue)
```

**Table 13.17:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strAttributeName	The name of the attribute whose value is to be returned
strDefaultValue	Default string value to return if the attribute does not exist

##### Return value

String.

The string value of the PBDOM\_ATTRIBUTE object specified in strAttributeName. If no such object exists, the GetAttributeValue method returns the string provided in strDefaultValue.

##### See also

[GetAttribute](#)[GetAttributeValue Syntax 1](#)[GetAttributeValue Syntax 2](#)[GetAttributeValue Syntax 3](#)[HasAttributes](#)[SetAttribute](#)[SetAttributes](#)

#### 13.1.9 GetChildElement

##### Description

The GetChildElement method is overloaded:

- Syntax 1 returns the first child PBDOM\_ELEMENT object that matches the name indicated by the method parameter.
- Syntax 2 returns the first child PBDOM\_ELEMENT object that matches the name and namespace indicated by the method parameter.

## Syntax

**Table 13.18:**

For this syntax	See
<code>GetChildElement(string strElementName)</code>	<a href="#">GetChildElement Syntax 1</a>
<code>GetChildElement(string strElementName, string strNamespacePrefix, string strNamespaceUri)</code>	<a href="#">GetChildElement Syntax 2</a>

### 13.1.9.1 GetChildElement Syntax 1

#### Description

Returns the first child PBDOM\_ELEMENT object, matching the name indicated by the method parameter that is contained in the PBDOM\_ELEMENT object from which the method is invoked.

#### Syntax

```
pbdom_element_name.GetChildElement(string strElementName)
```

**Table 13.19:**

Argument	Description
<code>pbdom_element_name</code>	The name of a PBDOM_ELEMENT object
<code>strElementName</code>	The local name of the child PBDOM_ELEMENT object to be returned

#### Return value

PBDOM\_ELEMENT. The first child PBDOM\_ELEMENT object whose name matches the value of the method parameter. If no PBDOM\_ELEMENT object exists for the specified name, the GetChildElement method returns a value of null.

#### See also

[GetChildElement Syntax 2](#)

[GetChildElements](#)

[HasChildElements](#)

[HasChildren](#)

[IsRootElement](#)

[RemoveChildElement](#)

[RemoveChildElements](#)

### 13.1.9.2 GetChildElement Syntax 2

#### Description

Returns the first child PBDOM\_ELEMENT object, matching the name and namespace indicated by the method parameter contained in the PBDOM\_ELEMENT object from which the method is invoked.

#### Syntax

```
pbdom_element_name.GetChildElement(string strElementName, string
  strNamespacePrefix, string strNamespaceUri)
```

**Table 13.20:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strElementName	The local name of the child PBDOM_ELEMENT object to be returned
strNamespacePrefix	The prefix of the namespace of the child PBDOM_ELEMENT object to be returned
strNamespaceUri	The URI of the namespace of the child PBDOM_ELEMENT object to be returned

#### Return value

PBDOM\_ELEMENT. The first child PBDOM\_ELEMENT object whose name and namespace information match the values of the method parameters. If no PBDOM\_ELEMENT object exists for the specified name and namespace information, the GetChildElement method returns a value of null.

#### Throws

EXCEPTION\_INVALID\_ARGUMENT -- If any of the input arguments is invalid, for example, null.

EXCEPTION\_INVALID\_NAME -- If the input Element Name or input namespace prefix or namespace URI is invalid.

#### See also

[GetChildElement Syntax 1](#)

[GetChildElements](#)

[HasChildElements](#)

[HasChildren](#)

[IsRootElement](#)

[RemoveChildElement](#)

[RemoveChildElements](#)

### 13.1.10 GetChildElements

#### Description

The `GetChildElements` method is overloaded:

- Syntax 1 retrieves a list of all child `PBDOM_ELEMENT` objects nested one level deep within a `PBDOM_ELEMENT` object. The list is stored in the array specified when the method is invoked.
- Syntax 2 retrieves a list of all child `PBDOM_ELEMENT` objects nested one level deep within a `PBDOM_ELEMENT` object specified by the name provided and belonging to no namespace. The list is stored in the array specified when the method is invoked.
- Syntax 3 retrieves a list of all child `PBDOM_ELEMENT` objects nested one level deep within a `PBDOM_ELEMENT` object specified by the local name and namespace provided.

## Syntax

**Table 13.21:**

For this syntax	See
<code>GetChildElements(ref pbdom_element pbdom_element_array[])</code>	<a href="#">GetChildElements Syntax 1</a>
<code>GetChildElements(string strElementName, ref pbdom_element pbdom_element_array[])</code>	<a href="#">GetChildElements Syntax 2</a>
<code>GetChildElements(string strElementName, string strNamespacePrefix, string strNamespaceUri, ref pbdom_element pbdom_element_array[])</code>	<a href="#">GetChildElements Syntax 3</a>

### 13.1.10.1 GetChildElements Syntax 1

#### Description

Retrieves a list of all child `PBDOM_ELEMENT` objects nested one level deep within a `PBDOM_ELEMENT` object. The list is stored in the array specified when the method is invoked.

#### Syntax

```
pbdom_element_name.GetChildElements(ref pbdom_element pbdom_element_array)
```

**Table 13.22:**

Argument	Description
<code>pbdom_element_name</code>	The name of a <code>PBDOM_ELEMENT</code> object
<code>pbdom_element_array</code>	The array that stores the child <code>PBDOM_ELEMENT</code> objects

#### Return value

Boolean.

Returns true if child `PBDOM_ELEMENT` objects have been collected, and false otherwise.

#### Usage

If the `PBDOM_ELEMENT` object has no nested elements, `GetChildElements` returns an empty array.

**See also**[GetChildElement](#)[GetChildElements Syntax 2](#)[GetChildElements Syntax 3](#)[HasChildElements](#)[HasChildren](#)[IsRootElement](#)[RemoveChildElement](#)[RemoveChildElements](#)**13.1.10.2 GetChildElements Syntax 2****Description**

Retrieves a list of all child PBDOM\_ELEMENT objects nested one level deep within a PBDOM\_ELEMENT object specified by the name provided and belonging to no namespace. The list is stored in the array specified when the method is invoked.

**Syntax**

```
pbdom_element_name.GetChildElements(string strElementName, ref pbdom_element  
pbdom_element_array[])
```

**Table 13.23:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strElementName	The name of the PBDOM_ELEMENT object for which to find children
pbdom_element_array	The array that stores the child PBDOM_ELEMENT objects

**Return value**

Boolean.

Returns true if child PBDOM\_ELEMENT objects have been collected, and false otherwise.

**Usage**

If the PBDOM\_ELEMENT object has no nested elements, GetChildElements returns an empty array.

**See also**[GetChildElement](#)[GetChildElements Syntax 1](#)[GetChildElements Syntax 3](#)[HasChildElements](#)



[HasChildren](#)[IsRootElement](#)[RemoveChildElement](#)[RemoveChildElements](#)

### 13.1.10.3 GetChildElements Syntax 3

#### Description

Retrieves a list of all child PBDOM\_ELEMENT objects nested one level deep within a PBDOM\_ELEMENT object specified by the local name and namespace provided.

#### Syntax

```
pbdom_element_name.GetChildElements(string strElementName, string
  strNamespacePrefix, string strNamespaceUri, ref pbdom_element
  pbdom_element_array[])
```

**Table 13.24:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strElementName	The name of a PBDOM_ELEMENT object for which to find children
strNamespacePrefix	The prefix of the namespace of the child PBDOM_ELEMENT objects to match
strNamespaceUri	The URI of the namespace of the child PBDOM_ELEMENT objects to match
pbdom_element_array[]	The array that stores the child PBDOM_ELEMENT objects

#### Return value

Boolean.

Returns true if child PBDOM\_ELEMENT objects have been collected, and false otherwise.

#### Throws

EXCEPTION\_INVALID\_ARGUMENT -- If any of the parameters is invalid.

EXCEPTION\_INVALID\_NAME -- If the input element name or namespace prefix or namespace URI is invalid. The only exception is if the input element name is an empty string.

#### Usage

If the PBDOM\_ELEMENT object has no nested elements, GetChildElements returns an empty array.

If the value of strElementName is an empty string, then all child elements match.

#### See also

[GetChildElement](#)[GetChildElements Syntax 1](#)

[GetChildElements Syntax 2](#)[HasChildElements](#)[HasChildren](#)[IsRootElement](#)[RemoveChildElement](#)[RemoveChildElements](#)

### 13.1.11 GetContent

#### Description

Obtains an array of PBDOM\_OBJECT objects, each of which is a child node of the PBDOM\_ELEMENT object from which the method is invoked. The returned array is "live" in that changes to any item of the array affect the actual item to which the array refers.

#### Syntax

```
pbdom_element_name.GetContent(ref pbdom_object pbdom_object_array[ ])
```

**Table 13.25:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
pbdom_object_array	The name of an array of PBDOM_OBJECT objects that receive references to the PBDOM_OBJECT objects contained within the PBDOM_ELEMENT object

#### Return value

Boolean.

Returns true for success and false otherwise.

#### Throws

EXCEPTION\_INVALID\_ARGUMENT -- If the input array is null.

#### Examples

The GetContent method is invoked for the Root> PBDOM\_ELEMENT object in the following XML DOM document:

```
<Root>
  <Element_1>
    <Element_1_1/>
    <Element_1_2/>
    <Element_1_3/>
  </Element_1>
  <Element_2/>
  <Element_3/>
</Root>
```

The GetContent method is invoked from the following PowerShell code:

```
PBDOM_DOCUMENT pbdom_doc
```

```
PBDOM_ELEMENT pbdom_elem_root
PBDOM_OBJECT pbdom_obj_array[]

pbdom_elem_root = pbdom_doc.GetRootElement()
pbdom_elem_root.GetContent(pbdom_obj_array)
```

If the GetContent method returns the value true, the PBDOM\_OBJECT object pbdom\_obj\_array then contains the following content:

**Table 13.26:**

Array element	Value
1	<Element_1>
2	<Element_2>
3	<Element_3>

### See also

[AddContent Syntax 1](#)

[AddContent Syntax 2](#)

[InsertContent](#)

[RemoveContent](#)

[SetContent](#)

### 13.1.12 GetName

#### Description

Retrieves the local name of a PBDOM\_ELEMENT object.

#### Syntax

```
pbdom_element_name.GetName()
```

**Table 13.27:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object

#### Return value

String.

The name of the element as it appears in the XML document but without any namespace prefix.

#### Examples

The GetName method returns the string abc when it is invoked for the name of the following element:

```
<ns:abc>My Element</ns:abc>
```

#### Usage

For an XML element that appears in the form [namespace\_prefix]:[element\_name], the local element name is element\_name. When the XML element has no namespace prefix, the local name is simply the element name.

Use the GetQualifiedName method to obtain the fully qualified name of an element (with the namespace prefix).

**See also**

[GetNamespacePrefix](#)

[GetNamespaceUri](#)

[RemoveNamespaceDeclaration](#)

[SetName](#)

### 13.1.13 GetNamespacePrefix

**Description**

Returns the namespace prefix for a PBDOM\_ELEMENT object. If no namespace prefix exists for the PBDOM\_ELEMENT object, GetNamespacePrefix returns an empty string.

**Syntax**

```
pbdom_element_name.GetNamespacePrefix()
```

**Table 13.28:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object

**Return value**

String.

The namespace prefix for the PBDOM\_ELEMENT object.

**See also**

[AddNamespaceDeclaration](#)

[GetNamespaceUri](#)

[GetQualifiedName](#)

[RemoveNamespaceDeclaration](#)

[SetNamespace](#)

### 13.1.14 GetNamespaceUri

**Description**

Returns the URI that is mapped to a PBDOM\_ELEMENT object prefix or, if there is no prefix, to the PBDOM\_ELEMENT object default namespace. If no URI is mapped to the PBDOM\_ELEMENT object, GetNameSpaceUri returns an empty string.

**Syntax**

```
pbdom_element_name.GetNamespaceUri()
```

**Table 13.29:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object

**Return value**

String.

The namespace URI for the PBDOM\_ELEMENT object.

**See also**

[AddNamespaceDeclaration](#)

[GetNamespacePrefix](#)

[GetQualifiedName](#)

[RemoveNamespaceDeclaration](#)

[SetNamespace](#)

**13.1.15 GetObjectClass****Description**

Returns a long integer code that indicates the class of the current PBDOM\_OBJECT.

**Syntax**

```
pbdom_object_name.GetObjectClass()
```

**Table 13.30:**

Argument	Description
pbdom_object_name	The name of a PBDOM_OBJECT object

**Return value**

Long.

A code that indicates the class of the current PBDOM\_OBJECT. If pbdom\_object\_name is a PBDOM\_ELEMENT object, the returned value is 3.

**Examples**

The GetObjectClass method returns a value specific to the class of the object from which the method is invoked.

```
PBDOM_OBJECT pbdom_obj
pbdom_obj = Create PBDOM_ELEMENT
MessageBox ("Class", &
string(pbdom_obj.GetObjectClass()))
```

This example illustrates polymorphisms: pbdom\_obj is declared as PBDOM\_OBJECT but instantiated as PBDOM\_ELEMENT. A message box returns the result of the GetObjectClass

method invoked for PBDOM\_ELEMENT object. Here the result is 3, indicating that pbdom\_obj is a PBDOM\_ELEMENT object.

### Usage

This method can be used for diagnostic purposes to dynamically determine the type of a PBDOM\_OBJECT at runtime.

## 13.1.16 GetObjectClassString

### Description

Returns a string form of the class of the PBDOM\_OBJECT.

### Syntax

```
pbdom_object_name.GetObjectClassString()
```

**Table 13.31:**

Argument	Description
pbdom_object_name	The name of your PBDOM_OBJECT object

### Return value

String.

A string that indicates the class of the current PBDOM\_OBJECT. If pbdom\_object\_name is a PBDOM\_ELEMENT object, the returned string is "pbdom\_element".

### Examples

The GetObjectClass method returns a string specific to the class of the object from which the method is invoked.

```
PBDOM_OBJECT pbdom_obj
pbdom_obj = Create PBDOM_ELEMENT
MessageBox ("Class", pbdom_obj.GetObjectClassString())
```

This example illustrates polymorphisms: pbdom\_obj is declared as PBDOM\_OBJECT but instantiated as PBDOM\_ELEMENT object. A message box returns the result of the GetObjectClassString method invoked for PBDOM\_ELEMENT object. Here the result is pbdom\_element, indicating that pbdom\_obj is a PBDOM\_ELEMENT object.

### Usage

This method can be used for diagnostic purposes to dynamically determine the actual type of a PBDOM\_OBJECT at runtime.

## 13.1.17 GetOwnerDocumentObject

### Description

Returns the PBDOM\_DOCUMENT object that owns the PBDOM\_ELEMENT object.

### Syntax

```
pbdom_element_name.GetOwnerDocumentObject()
```

**Table 13.32:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object

**Return value**

PBDOM\_DOCUMENT. The PBDOM\_DOCUMENT that owns the PBDOM\_ELEMENT object from which the GetOwnerDocumentObject method is invoked. A return value of null indicates that the PBDOM\_ELEMENT object is not owned by any PBDOM\_DOCUMENT.

**Examples**

The GetOwnerDocumentObject method is invoked from the following PowerShell code, where pbdom\_root\_elem refers to the root element of the PBDOM\_DOCUMENT object pbdom\_doc:

```
PBDOM_DOCUMENT pbdom_doc
PBDOM_ELEMENT pbdom_root_elem

pbdom_root_elem = pbdom_doc.GetRootElement()

IF
    pbdom_doc.Equals &
    (pbdom_root_elem.GetOwnerDocumentObject())
THEN
    MessageBox ("Equals", "The objects are equal")
END IF
```

The Equals method tests for equality between pbdom\_doc and the PBDOM\_DOCUMENT object returned from the GetOwnerDocumentObject method. A message box reports that the objects are equal.

**See also**

[GetParentObject](#)

[SetParentObject](#)

**13.1.18 GetParentObject****Description**

Returns the parent object for the PBDOM\_ELEMENT object.

**Syntax**

```
pbdom_element_name.GetParentObject()
```

**Table 13.33:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object

**Return value**

PBDOM\_OBJECT. The parent object of the PBDOM\_ELEMENT object from which the GetParentObject method is invoked. A return value of null indicates the PBDOM\_ELEMENT object has no parent.

**See also**[GetOwnerDocumentObject](#)[SetParentObject](#)**13.1.19 GetQualifiedName****Description**

Returns the full name of a PBDOM\_ELEMENT object in the form [namespace\_prefix]: [local\_name]. If there is no namespace prefix for the PBDOM\_ELEMENT object, the GetQualifiedName method returns the local name.

**Syntax**

```
pbdom_element_name.GetQualifiedName()
```

**Table 13.34:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object

**Return value**

String.

The full name of the PBDOM\_ELEMENT object. The full name consists of both a namespace prefix and a local name.

**See also**[AddNamespaceDeclaration](#)[GetNamespacePrefix](#)[GetNamespaceUri](#)[RemoveNamespaceDeclaration](#)[SetNamespace](#)**13.1.20 GetText****Description**

Obtains a concatenation of the text values of all the PBDOM\_TEXT and PBDOM\_CDATA nodes contained within the PBDOM\_ELEMENT object from which the method is invoked.

**Syntax**

```
pbdom_element_name.GetText()
```

**Table 13.35:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object

**Return value**

String



## Examples

The GetText method is invoked for the abc PBDOM\_ELEMENT object:

```
<abc>Root Element Data<data>ABC Data </data> now with extra info</abc>
```

The GetText method returns the following string:

```
Root Element Data now with extra info
```

The text "ABC Data" is excluded because it is not contained within the PBDOM\_ELEMENT abc.

## See also

[GetTextNormalize](#)

[GetTextTrim](#)

[SetText](#)

### 13.1.21 GetTextNormalize

#### Description

Returns the normalized text data contained in a PBDOM\_ELEMENT object.

#### Syntax

```
pbdom_element_name.GetTextNormalize()
```

**Table 13.36:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object

#### Return value

String

#### Examples

The GetTextNormalize method is invoked for the abc element of the following XML:

```
<abc> Root Element Data <data>ABC Data </data> now with extra info </abc>
```

The GetTextNormalize method returns the following string:

```
Root Element Data now with extra info
```

#### Usage

The text data returned includes any text data contained in PBDOM\_CDATA objects. All surrounding whitespace characters are removed. Internal whitespace characters are normalized to a single space. The GetTextNormalize method returns an empty string if no text values exist for the PBDOM\_ELEMENT object or if there are only whitespace characters.

## See also

[GetText](#)

[GetTextTrim](#)[SetText](#)

### 13.1.22 GetTextTrim

#### Description

Returns the text data contained within a PBDOM\_ELEMENT object with any leading and trailing whitespace characters removed.

#### Syntax

```
pbdom_element_name.GetTextTrim()
```

**Table 13.37:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object

#### Return value

String

#### Examples

The GetTextTrim method is invoked for the abc element of the following XML:

```
<abc> Root Element Data  with some cdata text ]]&gt;&lt;/abc&gt;</pre></div><div data-bbox="115 481 561 500" data-label="Text"><p>The GetTextTrim method returns the following string:</p></div><div data-bbox="115 509 467 522" data-label="Text"><pre>Root Element Data with some cdata text</pre></div><div data-bbox="115 531 174 549" data-label="Section-Header"><h4>Usage</h4></div><div data-bbox="115 556 733 573" data-label="Text"><p>Surrounding whitespace characters are removed from the returned text data.</p></div><div data-bbox="115 572 804 607" data-label="Text"><p>The GetTextTrim method returns an empty string if no text value exists for the PBDOM_ELEMENT object or if the text value contains only whitespace characters.</p></div><div data-bbox="115 613 190 630" data-label="Section-Header"><h4>See also</h4></div><div data-bbox="115 638 189 655" data-label="Text"><a href="#">GetText</a></div><div data-bbox="115 662 273 679" data-label="Text"><a href="#">GetTextNormalize</a></div><div data-bbox="115 686 186 703" data-label="Text"><a href="#">SetText</a></div><div data-bbox="115 723 327 740" data-label="Section-Header"><h3>13.1.23 HasAttributes</h3></div><div data-bbox="115 747 222 765" data-label="Section-Header"><h4>Description</h4></div><div data-bbox="115 771 733 789" data-label="Text"><p>Indicates whether a PBDOM_ELEMENT object has one or more attributes.</p></div><div data-bbox="115 795 183 813" data-label="Section-Header"><h4>Syntax</h4></div><div data-bbox="115 822 428 837" data-label="Text"><pre>pbdom_element_name.HasAttributes()</pre></div><div data-bbox="115 847 219 863" data-label="Caption"><p><b>Table 13.38:</b></p></div><div data-bbox="115 864 870 910" data-label="Table"><table border="1"><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td>pbdom_element_name</td><td>The name of a PBDOM_ELEMENT object</td></tr></tbody></table></div><div data-bbox="805 938 880 955" data-label="Page-Footer">Page 194</div>
```

**Return value**

Boolean.

Returns true if this PBDOM\_ELEMENT object has at least one attribute and false if this PBDOM\_ELEMENT object has no attributes.

**Examples**

In the following document fragment, only the element site has an attribute (href):

```
<books>
  <title>Inside Wizardry</title>
  <author>Ron Potter</author>
  <site href="http://www.mybooks.com/press" />
</books>
```

If the PBDOM\_ELEMENT object pbdom\_elem\_site represents the element site, the following call returns true:

```
pbdom_elem_site.HasAttributes()
```

**See also**

[GetAttribute](#)

[GetAttributes](#)

[GetAttributeValue](#)

[SetAttribute](#)

[SetAttributes](#)

**13.1.24 HasChildElements****Description**

Indicates whether a PBDOM\_ELEMENT object has one or more child PBDOM\_ELEMENT objects.

**Syntax**

```
pbdom_element_name.HasChildElements()
```

**Table 13.39:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object

**Return value**

Boolean.

Returns true if this PBDOM\_ELEMENT object has at least one child PBDOM\_ELEMENT object and false if this PBDOM\_ELEMENT object has no child PBDOM\_ELEMENT objects.

**Examples**

The HasChildElements method is invoked for the books PBDOM\_ELEMENT object in the following XML fragment:

```
<books>
  <title>Inside OLE</title>
  <author>Kraig Brockschmidt</author>
  <site href="http://www.microsoft.com/press" />
</books>
```

The books object has three child PBDOM\_ELEMENT objects: title, author, and site. The HasChildElements method returns true.

### See also

[GetChildElement](#)

[GetChildElements](#)

[HasChildren](#)

[IsRootElement](#)

[RemoveChildElement](#)

[RemoveChildElements](#)

## 13.1.25 HasChildren

### Description

Indicates whether a PBDOM\_ELEMENT object has one or more child objects.

### Syntax

```
pbdom_element_name.HasChildren()
```

**Table 13.40:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object

### Return value

Boolean.

Returns true if this PBDOM\_ELEMENT object has at least one child object and false if this PBDOM\_ELEMENT object has no child objects.

### Examples

The HasChildren method is invoked for elements in the following XML fragment:

```
<books>
  <title>Inside OLE</title>
  <author>Kraig Brockschmidt</author>
  <site href="http://www.microsoft.com/press" />
</books>
```

The books element has three child elements: title, author, and site. The title and author elements each have a child PBDOM\_TEXT object. The HasChildren method returns a value of true when invoked for these elements.

In contrast, the site element has a PBDOM\_ATTRIBUTE href, which is not considered a child PBDOM\_OBJECT. The HasChildren method returns a value of False when invoked for the site element.

## Usage

PBDOM's implementation of the HasChildren method differs from JDOM's implementation in that the JDOM HasChildren method returns true only if an Element contains child Elements. Text and other types of objects do not count.

PBDOM provides an alternative method, HasChildElements, to specifically detect whether a PBDOM\_ELEMENT object has at least one child PBDOM\_ELEMENT object.

## See also

[HasChildElements](#)

[IsRootElement](#)

## 13.1.26 InsertContent

### Description

Inserts a new PBDOM\_OBJECT into a PBDOM\_ELEMENT object.

### Syntax

```
pbdom_element_name.InsertContent(pbdom_object pbdom_object_new, pbdom_object
pbdom_object_ref)
```

**Table 13.41:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
pbdom_object_new	The PBDOM_OBJECT to insert
pbdom_object_ref	A positional reference PBDOM_OBJECT in front of which the new PBDOM_OBJECT is to be inserted

### Return value

PBDOM\_OBJECT. The PBDOM\_ELEMENT object modified and returned as a PBDOM\_OBJECT.

### Throws

EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT -- If an invalid PBDOM\_OBJECT is added. See [AddContent](#) for the valid PBDOM\_OBJECT objects that can be added to a PBDOM\_ELEMENT object. This exception is also thrown if the input PBDOM\_OBJECT or the reference PBDOM\_OBJECT is this PBDOM\_ELEMENT object itself.

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- If the input PBDOM\_OBJECT to insert has not been given a user-defined name. The same exception is also thrown if the reference PBDOM\_OBJECT is also not given a user-defined name, unless the reference PBDOM\_OBJECT is specifically set to null.

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the input PBDOM\_OBJECT to insert is not associated with a derived PBDOM\_OBJECT. The same exception is also thrown if the reference PBDOM\_OBJECT is also not associated with a derived PBDOM\_OBJECT unless the reference PBDOM\_OBJECT is specifically set to null.

**EXCEPTION\_INVALID\_ARGUMENT** -- If the reference PBDOM\_OBJECT (second parameter) is intended to be null but is not specifically set to null using the SetNull method.

**EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT** -- If the input PBDOM\_OBJECT to insert already has a parent.

**EXCEPTION\_WRONG\_PARENT\_ERROR** -- If the reference PBDOM\_OBJECT is not a child of this PBDOM\_ELEMENT object.

**EXCEPTION\_HIERARCHY\_ERROR** -- If inserting the input PBDOM\_OBJECT will cause the current PBDOM\_ELEMENT object to be no longer well formed.

## Examples

The following PowerShell code is used to create an XML document:

```
pbdom_doc1 = Create PBDOM_DOCUMENT
pbdom_elem_1 = Create PBDOM_ELEMENT
pbdom_elem_2 = Create PBDOM_ELEMENT
pbdom_elem_3 = Create PBDOM_ELEMENT

pbdom_elem_1.SetName ("pbdom_elem_1")
pbdom_elem_2.SetName ("pbdom_elem_2")
pbdom_elem_3.SetName ("pbdom_elem_3")

pbdom_doc1.NewDocument ("", "", "Root_Element", "", "")
pbdom_elem_root = pbdom_doc1.GetRootElement()
pbdom_elem_root.AddContent (pbdom_elem_1)
pbdom_elem_root.AddContent (pbdom_elem_3)
```

The following XML results:

```
!DOCTYPE Root_Element>
<Root_Element>
  <pbdom_elem_1 />
  <pbdom_elem_3 />
</Root_Element>
```

The InsertContent method is used to add an element between pbdom\_elem\_1 and pbdom\_elem\_3:

```
pbdom_elem_root.InsertContent(pbdom_elem_2, &
    pbdom_elem_3)
```

The following XML results:

```
<!DOCTYPE Root_Element>
<Root_Element>
  <pbdom_elem_1 />
  <pbdom_elem_2 />
  <pbdom_elem_3 />
</Root_Element>
```

## Usage

The inserted object becomes a child of the PBDOM\_ELEMENT object. The new PBDOM\_OBJECT is positioned before another PBDOM\_OBJECT, which is specified in the second of two parameters.

## See also

[AddContent Syntax 1](#)

[AddContent Syntax 2](#)[GetContent](#)[RemoveContent](#)[SetContent](#)

### 13.1.27 IsAncestorObjectOf

#### Description

Determines whether a PBDOM\_ELEMENT object is the ancestor of the PBDOM\_OBJECT indicated by the method parameter.

#### Syntax

```
pbdom_element_name.IsAncestorObjectOf(pbdom_object pbdom_object_ref)
```

Table 13.42:

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
pbdom_object_ref	The PBDOM_OBJECT to be tested for equality with this PBDOM_ELEMENT object

#### Return value

Boolean.

Returns true if this PBDOM\_ELEMENT object is the ancestor of the specified PBDOM\_OBJECT, and false otherwise.

### 13.1.28 IsRootElement

#### Description

Indicates whether a PBDOM\_ELEMENT object is the root element of a PBDOM\_DOCUMENT object.

#### Syntax

```
pbdom_element_name.IsRootElement()
```

Table 13.43:

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object

#### Return value

Boolean.

Returns true if this PBDOM\_ELEMENT object is the root element of a PBDOM\_DOCUMENT, and false otherwise.

#### See also

[GetChildElement](#)[GetChildElements](#)[HasChildElements](#)[HasChildren](#)[RemoveChildElement](#)[RemoveChildElements](#)

### 13.1.29 RemoveAttribute

#### Description

The RemoveAttribute method is overloaded:

- Syntax 1 removes a PBDOM\_ATTRIBUTE from its owner PBDOM\_ELEMENT object using a reference to the PBDOM\_ATTRIBUTE.
- Syntax 2 removes a PBDOM\_ATTRIBUTE from its owner PBDOM\_ELEMENT object using the name of the PBDOM\_ATTRIBUTE.
- Syntax 3 removes a PBDOM\_ATTRIBUTE from its owner PBDOM\_ELEMENT object using the name and namespace of the PBDOM\_ATTRIBUTE.

#### Syntax

**Table 13.44:**

For this syntax	See
<code>RemoveAttribute(pbdom_attribute pbdom_attribute_ref)</code>	<a href="#">RemoveAttribute Syntax 1</a>
<code>RemoveAttribute(string strAttributeName)</code>	<a href="#">RemoveAttribute Syntax 2</a>
<code>RemoveAttribute(string strAttributeName, string strNamespacePrefix, string strNamespaceUri)</code>	<a href="#">RemoveAttribute Syntax 3</a>

#### 13.1.29.1 RemoveAttribute Syntax 1

#### Description

Removes a PBDOM\_ATTRIBUTE from its owner PBDOM\_ELEMENT object.

#### Syntax

```
pbdom_element_name.RemoveAttribute(pbdom_attribute pbdom_attribute_ref)
```

**Table 13.45:**

Argument	Description
<code>pbdom_element_name</code>	The name of a PBDOM_ELEMENT object
<code>pbdom_attribute_ref</code>	The PBDOM_ATTRIBUTE object to remove from this PBDOM_ELEMENT object



**Return value**

Boolean.

Returns true if the specified PBDOM\_ATTRIBUTE was removed, and false otherwise.

**13.1.29.2 RemoveAttribute Syntax 2****Description**

Removes a PBDOM\_ATTRIBUTE specified by the name provided that is not contained in a namespace. If no such PBDOM\_ATTRIBUTE exists, RemoveAttribute does nothing.

**Syntax**

```
pbdom_element_name.RemoveAttribute(string strAttributeName)
```

**Table 13.46:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strAttributeName	The name of the PBDOM_ATTRIBUTE object to remove

**Return value**

Boolean.

Returns true if the specified PBDOM\_ATTRIBUTE was removed, and false otherwise.

**13.1.29.3 RemoveAttribute Syntax 3****Description**

Removes a PBDOM\_ATTRIBUTE specified by the name and namespace provided. If no such PBDOM\_ATTRIBUTE exists, RemoveAttribute does nothing.

**Syntax**

```
pbdom_element_name.RemoveAttribute(string strAttributeName, string  
strNamespacePrefix, string strNamespaceUri)
```

**Table 13.47:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strAttributeName	The name of the PBDOM_ATTRIBUTE object to remove
strNamespacePrefix	Prefix of the namespace of the PBDOM_ATTRIBUTE to remove
strNamespaceUri	URI of the namespace of the PBDOM_ATTRIBUTE to remove

**Return value**

Boolean.

Returns true if the specified PBDOM\_ATTRIBUTE was removed, and false otherwise.

### Throws

EXCEPTION\_INVALID\_ARGUMENT -- If any of the input parameters is invalid, for example, null.

EXCEPTION\_INVALID\_STRING -- If the input Attribute Name is invalid (for example, contains a colon), or if the namespace prefix or URI is invalid.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- If a memory allocation failure occurred during the execution of this method.

## 13.1.30 RemoveChildElement

### Description

The RemoveChildElement method is overloaded:

- Syntax 1 removes the first child PBDOM\_ELEMENT object (one level deep) that has the local name provided and belongs to no namespace.
- Syntax 2 removes the first child PBDOM\_ELEMENT object (one level deep) that has the local name provided and belongs to the specified namespace.

### Syntax

**Table 13.48:**

For this syntax	See
<code>RemoveChildElement(string strElementName)</code>	<a href="#">RemoveChildElement Syntax 1</a>
<code>RemoveChildElement(string strElementName, string strNamespacePrefix, string strNamespaceUri)</code>	<a href="#">RemoveChildElement Syntax 2</a>

### 13.1.30.1 RemoveChildElement Syntax 1

#### Description

Removes the first child PBDOM\_ELEMENT object (one level deep) that has the local name provided and belongs to no namespace.

#### Syntax

```
pbdom_element_name.RemoveChildElement(string strElementName)
```

**Table 13.49:**

Argument	Description
<code>pbdom_element_name</code>	The name of a PBDOM_ELEMENT object
<code>strElementName</code>	The name of the child PBDOM_ELEMENT object to remove

#### Return value

Boolean.

Returns true if the specified PBDOM\_ELEMENT object was removed, and false otherwise.

### Throws

EXCEPTION\_INVALID\_ARGUMENT -- If the input parameter is invalid, for example, null.

EXCEPTION\_INVALID\_STRING -- If the input element name is invalid.

### See also

[GetChildElement](#)

[GetChildElements](#)

[HasChildElements](#)

[HasChildren](#)

[IsRootElement](#)

[RemoveChildElement Syntax 2](#)

[RemoveChildElements](#)

#### 13.1.30.2 RemoveChildElement Syntax 2

### Description

Removes the first child PBDOM\_ELEMENT object (one level deep) that has the local name provided and belongs to the specified namespace.

### Syntax

```
pbdom_element_name.RemoveChildElement(string strElementName, string
    strNamespacePrefix, string strNamespaceUri)
```

**Table 13.50:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strElementName	The name of the PBDOM_ELEMENT object to remove
strNamespacePrefix	Prefix of the namespace of the PBDOM_ELEMENT object to remove
strNamespaceUri	URI of the namespace of the PBDOM_ATTRIBUTE to remove

### Return value

Boolean.

Returns true if the specified PBDOM\_ELEMENT object was removed and false otherwise.

### Throws

EXCEPTION\_INVALID\_ARGUMENT -- If the input parameter is invalid, for example, null.

EXCEPTION\_INVALID\_STRING -- If the input element name is invalid or the input namespace prefix or URI is invalid.

### See also

[GetChildElement](#)

[GetChildElements](#)

[HasChildElements](#)

[HasChildren](#)

[IsRootElement](#)

[RemoveChildElement Syntax 1](#)

[RemoveChildElements](#)

## 13.1.31 RemoveChildElements

### Description

The RemoveChildElements method is overloaded:

- Syntax 1 method removes from the current PBDOM\_ELEMENT object all child PBDOM\_ELEMENT objects. It uses no parameters.
- Syntax 2 method removes from the current PBDOM\_ELEMENT object all child PBDOM\_ELEMENT objects that have the specified local name and belong to no namespace.
- Syntax 3 removes from the current PBDOM\_ELEMENT object all child PBDOM\_ELEMENT objects (one level deep) that have the specified local name and belong to the specified namespace.

### Syntax

**Table 13.51:**

For this syntax	See
<code>RemoveChildElements()</code>	<a href="#">RemoveChildElements Syntax 1</a>
<code>RemoveChildElements(string strElementName)</code>	<a href="#">RemoveChildElements Syntax 2</a>
<code>RemoveChildElements(string strElementName, string strNamespacePrefix, string strNamespaceUri)</code>	<a href="#">RemoveChildElements Syntax 3</a>

#### 13.1.31.1 RemoveChildElements Syntax 1

### Description

Removes from the current PBDOM\_ELEMENT object all child PBDOM\_ELEMENT objects. It uses no parameters.

### Syntax

```
pbdom_element_name.RemoveChildElements()
```

**Table 13.52:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object

**Return value**

Boolean.

Returns true if any child PBDOM\_ELEMENT object was removed and false otherwise.

**See also**

[GetChildElement](#)

[GetChildElements](#)

[HasChildElements](#)

[HasChildren](#)

[IsRootElement](#)

[RemoveChildElement](#)

[RemoveChildElements Syntax 2](#)

[RemoveChildElements Syntax 3](#)

**13.1.31.2 RemoveChildElements Syntax 2****Description**

Removes from the current PBDOM\_ELEMENT object all child PBDOM\_ELEMENT objects that have the specified local name and belong to no namespace.

**Syntax**

```
pbdom_element_name.RemoveChildElements(string strElementName)
```

**Table 13.53:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strElementName	The name of the child PBDOM_ELEMENT objects to remove

**Return value**

Boolean.

Returns true if any child PBDOM\_ELEMENT object was removed, and false otherwise.

**See also**

[GetChildElement](#)

[GetChildElements](#)

[HasChildElements](#)

[HasChildren](#)

[IsRootElement](#)[RemoveChildElement](#)[RemoveChildElements Syntax 1](#)[RemoveChildElements Syntax 3](#)

### 13.1.31.3 RemoveChildElements Syntax 3

#### Description

Removes from the current PBDOM\_ELEMENT object all child PBDOM\_ELEMENT objects (one level deep) that have the specified local name and belong to the specified namespace.

#### Syntax

```
pbdom_element_name.RemoveChildElements(string strElementName, string
strNamespacePrefix, string strNamespaceUri)
```

**Table 13.54:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strElementName	The name of the child PBDOM_ELEMENT objects to remove
strNamespacePrefix	Prefix of the namespace of the child PBDOM_ELEMENT objects to remove
strNamespaceUri	URI of the namespace of the child PBDOM_ATTRIBUTE objects to remove

#### Return value

Boolean.

Returns true if any child PBDOM\_ELEMENT object was removed and false otherwise.

#### Throws

EXCEPTION\_INVALID\_ARGUMENT -- If any of the input parameters is invalid, for example, null.

EXCEPTION\_INVALID\_NAME -- If the input element name or namespace prefix or URI is invalid. The only exception is if the input element name is an empty string, in which case all element names match.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- If there was any memory allocation failure during the execution of this method.

#### See also

[GetChildElement](#)[GetChildElements](#)[HasChildElements](#)[HasChildren](#)

[IsRootElement](#)[RemoveChildElement](#)[RemoveChildElements Syntax 1](#)[RemoveChildElements Syntax 2](#)

### 13.1.32 RemoveContent

#### Description

Removes a child PBDOM\_OBJECT from a PBDOM\_ELEMENT object. All children of the removed PBDOM\_OBJECT are also removed.

#### Syntax

```
pbdom_element_name.RemoveContent(pbdom_object pbdom_object_ref)
```

**Table 13.55:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
pbdom_object_ref	The PBDOM_OBJECT to remove

#### Return value

Boolean.

Returns true if the specified content was removed and false otherwise.

#### Throws

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- If the input PBDOM\_OBJECT has not been given a user-defined name.

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the input PBDOM\_OBJECT is not associated with a derived PBDOM\_OBJECT.

EXCEPTION\_WRONG\_DOCUMENT\_ERROR -- If the input PBDOM\_OBJECT is not from the same document as this PBDOM\_ELEMENT object.

EXCEPTION\_WRONG\_PARENT\_ERROR -- If the input PBDOM\_OBJECT is not a child of the current PBDOM\_ELEMENT object.

#### Examples

The RemoveContent method is used to modify the following XML fragment:

```
<Telephone_Book>
  <Entry>
    <Particulars>
      <Name>John Doe</Name>
      <Age>21</Age>
      <Phone_Number>1234567</Phone_Number>
    </Particulars>
  </Entry>
</Telephone_Book>
```

The RemoveContent method is invoked from the following PowerShell code:

```
PBDOM_DOCUMENT pbdom_doc
```

```
PBDOM_ELEMENT pbdom_entry
```

```
pbdom_doc.GetRootElement().RemoveContent(pbdom_entry)
```

The following XML results:

```
<Telephone_Book></Telephone_Book>
```

### See also

[AddContent Syntax 1](#)

[AddContent Syntax 2](#)

[GetContent](#)

[InsertContent](#)

[SetContent](#)

## 13.1.33 RemoveNamespaceDeclaration

### Description

Removes the specified PBDOM\_NAMESPACE declaration for a PBDOM\_ELEMENT object. If the namespace prefix is an empty string, RemoveNamespaceDeclaration removes a default namespace declaration.

### Syntax

```
pbdom_element_name.RemoveNamespaceDeclaration(string strNamespacePrefix, string strNamespaceUri)
```

**Table 13.56:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strNamespacePrefix	Prefix of the namespace declaration to remove
strNamespaceUri	URI of the namespace declaration to remove

### Return value

Boolean.

Returns true if the namespace has been removed from the PBDOM\_ELEMENT object, and false otherwise.

### Throws

EXCEPTION\_INVALID\_ARGUMENT -- If any of the input parameters is invalid, for example, null.

EXCEPTION\_INVALID\_NAME -- If the namespace prefix or URI is invalid, or both the namespace prefix and URI are invalid as a pair.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- If any memory allocation failure occurred during the execution of this method.

### See also



[AddNamespaceDeclaration](#)[GetNamespacePrefix](#)[GetNamespaceUri](#)[GetQualifiedName](#)[SetNamespace](#)

### 13.1.34 SetAttribute

#### Description

The SetAttribute method is overloaded:

- Syntax 1 adds a predefined PBDOM\_ATTRIBUTE object to a PBDOM\_ELEMENT object.
- Syntax 2 adds a PBDOM\_ATTRIBUTE object and its value to a PBDOM\_ELEMENT object using strings for the name and value of the PBDOM\_ATTRIBUTE.
- Syntax 3 adds an attribute/value pair to a PBDOM\_ELEMENT object using strings for the name and value of the PBDOM\_ATTRIBUTE, and the prefix and URI of the namespace to which the PBDOM\_ATTRIBUTE belongs.

#### Syntax

**Table 13.57:**

For this syntax	See
<code>SetAttribute(pbdom_attribute pbdom_attribute_ref)</code>	<a href="#">SetAttribute Syntax 1</a>
<code>SetAttribute(string strName, string strValue)</code>	<a href="#">SetAttribute Syntax 2</a>
<code>SetAttribute(string strName, string strValue, string strNamespacePrefix, string strNamespaceUri, boolean bVerifyNamespace)</code>	<a href="#">SetAttribute Syntax 3</a>

#### 13.1.34.1 SetAttribute Syntax 1

#### Description

Adds a predefined PBDOM\_ATTRIBUTE object to a PBDOM\_ELEMENT object. Any existing attribute with the same name and namespace URI is overwritten.

#### Syntax

```
pbdom_element_name.SetAttribute(pbdom_attribute pbdom_attribute_ref)
```

**Table 13.58:**

Argument	Description
<code>pbdom_element_name</code>	The name of a PBDOM_ELEMENT object
<code>pbdom_attribute_ref</code>	The PBDOM_ATTRIBUTE object to be set for this PBDOM_ELEMENT object

## Return value

PBDOM\_ELEMENT. The PBDOM\_ELEMENT object modified to contain the specified PBDOM\_ATTRIBUTE.

## Throws

EXCEPTION\_INVALID\_ARGUMENT -- The input PBDOM\_ATTRIBUTE is invalid. This can happen if it has not been initialized properly or it is a null object reference.

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- The input PBDOM\_ATTRIBUTE has not been given a user-defined name.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_OWNER -- The input PBDOM\_ATTRIBUTE already has an owner element.

## Examples

1. The SetAttribute method is invoked for the following element:

```
<image></image>
```

The SetAttribute method is invoked from the following PowerShell code, where elem\_image represents the image element from the preceding XML:

```
attr_src.SetName("src")
attr_src.SetValue("logo.gif")
elem_image.SetAttribute(attr_src)
```

The following XML results:

```
<image src="logo.gif"></image>
```

2. The following example demonstrates the impact of setting a PBDOM\_ATTRIBUTE for a PBDOM\_ELEMENT object where the PBDOM\_ELEMENT object already contains an attribute of the same name and namespace URI as the input PBDOM\_ATTRIBUTE.

The example creates a PBDOM\_DOCUMENT based on the following document:

```
<root xmlns:pre1="http://www.pre.com" xmlns:pre2="http://www.pre.com">
  <child1 pre1:a="123"/>
</root>
```

Then it creates a PBDOM\_ATTRIBUTE object and sets its name to a and its prefix and URI to pre2 and http://www.pre.com. The bVerifyNamespace argument is set to false because this PBDOM\_ATTRIBUTE has not been assigned an owner PBDOM\_ELEMENT object yet, so that the verification for a predeclared namespace would fail. The text value is set to 456.

The child1 element already contains an attribute named a that belongs to the namespace http://www.pre.com, as indicated by the prefix pre1. The new PBDOM\_ATTRIBUTE uses the prefix pre2, but it represents the same namespace URI, so setting the new PBDOM\_ATTRIBUTE to child1 successfully replaces the existing pre1:a with the new PBDOM\_ATTRIBUTE pre2:a.

```
PBDOM_BUILDER pbdom_builder
PBDOM_DOCUMENT pbdom_doc
PBDOM_ATTRIBUTE pbdom_attr
string strXML = "<root xmlns:pre1=~\"http://www.pre.com~\" xmlns:pre2=~\"http://www.pre.com~\"><child1 pre1:a=~\"123~\"/></root>"
```

```
try
  pbdom_buildr = Create PBDOM_BUILDER
  pbdom_doc = pbdom_buildr.BuildFromString (strXML)

  // Create a PBDOM_ATTRIBUTE and set its properties
  pbdom_attr = Create PBDOM_ATTRIBUTE
  pbdom_attr.SetName ("a")
  pbdom_attr.SetNamespace ("pre2", &
    "http://www.pre.com", false)
  pbdom_attr.SetText("456")

  // Attempt to obtain the child1 element and
  // set the new attribute to it
  pbdom_doc.GetRootElement(). &
    GetChildElement("child1").SetAttribute(pbdom_attr)

  pbdom_doc.SaveDocument &
    ("pbdom_elem_set_attribute_1.xml")
catch (PBDOM_EXCEPTION except)
  MessageBox ("PBDOM_EXCEPTION", except.GetMessage())
end try
```

When saved and converted to an XML document, the document looks like the following:

```
<root xmlns:pre1="http://www.pre.com" xmlns:pre2="http://www.pre.com"
  <child1 pre2:a="456"/
</root
```

## Usage

This method allows the caller to add a predefined PBDOM\_ATTRIBUTE object to a PBDOM\_ELEMENT object. If this PBDOM\_ELEMENT object already contains an existing attribute with the same name and namespace URI as the input PBDOM\_ATTRIBUTE, the existing attribute is replaced by the input PBDOM\_ATTRIBUTE.

If a PBDOM\_ATTRIBUTE has been created to represent the original attribute, it is still valid after the call, but the attribute that it represents has been detached from the original owner element. Calling GetOwnerElementObject on this PBDOM\_ATTRIBUTE returns a null value.

### See also

[GetAttribute](#)

[GetAttributes](#)

[GetAttributeValue](#)

[HasAttributes](#)

[SetAttribute Syntax 2](#)

[SetAttribute Syntax 3](#)

[SetAttributes](#)

### 13.1.34.2 SetAttribute Syntax 2

#### Description

Adds a PBDOM\_ATTRIBUTE object and its value to a PBDOM\_ELEMENT object. Any existing attribute with the same name and namespace URI is overwritten.

## Syntax

```
pbdom_element_name.SetAttribute(string strName, string strValue)
```

**Table 13.59:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strName	The name of the PBDOM_ATTRIBUTE to be added
strValue	The value of the PBDOM_ATTRIBUTE to be added

## Return value

PBDOM\_ELEMENT. The PBDOM\_ELEMENT object modified to contain the specified PBDOM\_ATTRIBUTE with the specified value.

## Throws

EXCEPTION\_INVALID\_ARGUMENT -- One or both of the input strings are invalid. This can happen if either or both strings have not been initialized properly or are null.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- Insufficient memory was encountered while executing this method.

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_ELEMENT object's internal implementation is null. The occurrence of this exception is rare but can take place if severe memory corruption occurs.

EXCEPTION\_INVALID\_NAME -- An invalid name for the attribute is supplied.

EXCEPTION\_INVALID\_STRING -- An invalid string for the attribute value is supplied.

## Examples

1. The SetAttribute method is invoked for the following XML element:

```
<code0789725045</code>
```

The SetAttribute method is invoked from the following PowerShell statement, where elem\_code represents the code element:

```
elem_code.SetAttribute("type", "ISBN")
```

The following XML element results:

```
<code type="ISBN">0789725045</code>
```

2. The following example demonstrates the effect of setting an attribute for a PBDOM\_ELEMENT object when the PBDOM\_ELEMENT object already contains an attribute of the same name. The example creates a PBDOM\_DOCUMENT based on the following document:

```
<root xmlns:pre1="http://www.pre.com">
  <child1 pre1:a="123" b="456"/>
</root>
```

The child1 element already contains an attribute named b with value 456. Calling the SetAttribute method with name b and value 789 creates a new attribute for child1 that replaces the original b attribute.

```
PBDOM_BUILDER      pbdom_buildr
PBDOM_DOCUMENT     pbdom_doc
string strXML = "<root xmlns:prel=~\"http://www.pre.com~\" ><child1 prel:a=~\"123~\"
  b=~\"456~\"/></root>"

try
  pbdom_buildr = Create PBDOM_BUILDER
  pbdom_doc = pbdom_buildr.BuildFromString (strXML)
  pbdom_doc.GetRootElement(). &
    GetChildElement("child1").SetAttribute("b", "789")
catch (PBDOM_EXCEPTION except)
  MessageBox ("PBDOM_EXCEPTION", except.GetMessage())
end try
```

After the PBDOM\_DOCUMENT object is saved and converted to XML, the XML document looks like the following:

```
<root xmlns:prel="http://www.pre.com">
  <child1 prel:a="123" b="789"/>
</root>
```

## Usage

This method allows the caller to add an attribute/value pair to a PBDOM\_ELEMENT object. If the PBDOM\_ELEMENT object already contains an existing attribute that has the same name as the input name and that belongs to no namespace, the original attribute is removed from this PBDOM\_ELEMENT object and a new one (corresponding to the specified attribute name and value) is created and set in its place.

If a PBDOM\_ATTRIBUTE has been created to represent the original attribute, it is still valid, but the attribute that it represents has been detached from the original owner element. Calling GetOwnerElementObject on this PBDOM\_ATTRIBUTE returns a null value.

## See also

[GetAttribute](#)

[GetAttributes](#)

[GetAttributeValue](#)

[HasAttributes](#)

[SetAttribute Syntax 1](#)

[SetAttribute Syntax 3](#)

[SetAttributes](#)

### 13.1.34.3 SetAttribute Syntax 3

#### Description

Adds an attribute/value pair to a PBDOM\_ELEMENT object. The attribute namespace is specified, and any existing attribute of the same name and namespace URI is removed.

#### Syntax

```
pbdom_element_name.SetAttribute(string strName, string strValue, string
strNamespacePrefix, string strNamespaceUri, boolean bVerifyNamespace)
```

**Table 13.60:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strName	The name of the PBDOM_ATTRIBUTE to be added
strValue	The value of the PBDOM_ATTRIBUTE to be added
strNamespacePrefix	The prefix of the namespace to which the PBDOM_ATTRIBUTE belongs
strNamespaceUri	The URI of the namespace to which the PBDOM_ATTRIBUTE belongs
bVerifyNamespace	Specifies whether or not the method should verify the existence of an in-scope namespace declaration for the given prefix and URI

**Return value**

Long.

Returns 0 if no namespace verification error occurs and -1 if no in-scope namespace declaration exists for the given prefix and URI settings.

**Throws**

EXCEPTION\_INVALID\_ARGUMENT -- If any of the arguments is invalid. This can happen if any of the input strings has been set to null using the PowerScript SetNull function.

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_ELEMENT object's internal implementation is null. The occurrence of this exception is rare but can take place if severe memory corruption occurs.

EXCEPTION\_INVALID\_NAME -- The input namespace prefix or the URI, or their combination, is not valid. This will happen if:

- The namespace prefix is an empty string and the URI is not an empty string. If both are empty strings, the NONNAMESPACE namespace is being specified and this prefix/URI combination is correct.
- The namespace prefix is xmlns and the URI is not http://www.w3.org/2000/xmlns/. This namespace prefix/URI pair is unique and exclusive and cannot be used separately. The use of this pair signifies a namespace declaration.
- The namespace prefix string is invalid. That is, it does not conform to the W3C "Namespaces in XML" specifications for the name of a prefix.
- The namespace URI string is invalid. That is, it does not conform to the W3C specifications for a URI string.

**EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE** -- If there has been any memory allocation failure during this method call.

## Examples

1. The SetAttribute method is invoked for the following XML element:

```
<code>0789725045</code>
```

The SetAttribute method is invoked from the following PowerShell statement, where elem\_code represents the code element:

```
elem_code.SetAttribute("type", "ISBN", "ns", & "http://www.books.com/codes",  
false)
```

The following XML element results:

```
<code ns:type="ISBN">0789725045</code>
```

2. The following example demonstrates the effect of setting an attribute with a particular name and namespace URI for an element that already contains an existing attribute with the same name and namespace URI. It creates a PBDOM\_DOCUMENT based on the following XML:

```
<root xmlns:pre1="http://www.pre.com" xmlns:pre2="http://www.pre.com">  
  <child1 pre1:a="123"/>  
</root>
```

The child1 element already contains an attribute named a that belongs to the namespace http://www.pre.com, as indicated by the pre1 prefix. The call to SetAttribute attempts to set an attribute for child1 with the same name, a, but with the namespace prefix pre2.

The last parameter, bVerifyNamespace, is set to true. This tells the SetAttribute method to check first to see if an in-scope namespace declaration for pre2 and http://www.pre.com exists. An in-scope declaration for this namespace prefix/URI pair does exist, and so the verification succeeds.

The original pre1:a attribute is removed from the child1 element and a new attribute pre2:a, belonging to the same namespace and with the value 456, is created and set in its place. The new attribute replaces the original attribute, instead of being set as an additional attribute, because both attributes have the same URI.

```
PBDOM_BUILDER      pbdom_buildr  
PBDOM_DOCUMENT    pbdom_doc  
string strXML = "<root xmlns:pre1=~\"http://www.pre.com~\" xmlns:pre2=~\"http://  
www.pre.com~\"><child1 pre1:a=~\"123~\"/></root>\"  
  
try  
  pbdom_buildr = Create PBDOM_BUILDER  
  pbdom_doc = pbdom_buildr.BuildFromString (strXML)  
  
pbdom_doc.GetRootElement().GetChildElement("child1").SetAttribute("a", "456",  
  "pre2", "http://www.pre.com", true)  
  
catch (PBDOM_EXCEPTION pbdom_except)  
  MessageBox ("PBDOM_EXCEPTION", pbdom_except.GetMessage())  
end try
```

## Usage

This method allows the caller to add an attribute/value pair to a PBDOM\_ELEMENT object.

The parameter `bVerifyNamespace`, when set to true, instructs the method to perform a thorough search up the DOM node tree, starting at the current PBDOM\_ELEMENT object, to check for an in-scope namespace declaration for the given prefix and URI. If a namespace declaration is not found, no attribute is created. If a namespace declaration is found, an attribute is created.

If the `bVerifyNamespace` parameter is set to false, no verification search is performed, and the method always returns 0.

If the PBDOM\_ELEMENT object already contains an existing attribute that has the same name as the input name and the same namespace URI as the input namespace URI, the original attribute is replaced with a new one with the same name and URI.

If a PBDOM\_ATTRIBUTE has been created to represent the original attribute, it is still valid, but the attribute that it represents has been detached from the original owner element. Calling `GetOwnerElementObject` on this PBDOM\_ATTRIBUTE returns a null value.

#### See also

[GetAttribute](#)

[GetAttributes](#)

[GetAttributeValue](#)

[HasAttributes](#)

[SetAttribute Syntax 1](#)

[SetAttribute Syntax 2](#)

[SetAttributes](#)

### 13.1.35 SetAttributes

#### Description

Sets the attributes for the DOM element represented by the current PBDOM\_ELEMENT object.

#### Syntax

```
pbdom_element_name.SetAttributes(pbdom_attribute pbdom_attribute_array[])
```

**Table 13.61:**

Argument	Description
<code>pbdom_element_name</code>	The name of a PBDOM_ELEMENT object
<code>pbdom_attribute_array</code>	An array of PBDOM_ATTRIBUTE objects

#### Return value

PBDOM\_ELEMENT. The PBDOM\_ELEMENT object modified.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- The internal implementation of this PBDOM\_ELEMENT object or one of the PBDOM\_ATTRIBUTE array items is null. This exception is rare but can take place if severe memory corruption occurs.



EXCEPTION\_INVALID\_ARGUMENT -- One of the PBDOM\_ATTRIBUTE array items is null.

EXCEPTION\_INVALID\_NAME -- If two or more PBDOM\_ATTRIBUTES in the array contain the same name and namespace URI.

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- One of the PBDOM\_ATTRIBUTE array items has not been named.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_OWNER -- One of the PBDOM\_ATTRIBUTE array items already has an owner PBDOM\_ELEMENT object.

### Examples

This example demonstrates setting the attributes of a PBDOM\_ELEMENT object using an array of PBDOM\_ATTRIBUTE objects. It builds a PBDOM\_DOCUMENT based on the following XML:

```
<root xmlns:pre1="http://www.pre.com">
  <child1 pre1:a="123"/>
</root>
```

The code creates an array of three PBDOM\_ATTRIBUTE objects with names a, b, and c, and sets their namespace prefixes and URIs to pre1 and http://www.pre.com. The call to SetAttributes attempts to set the attributes of child1 using the PBDOM\_ATTRIBUTES of this array. When you save PBDOM\_DOCUMENT and convert it to an XML document, the result is:

```
<root xmlns:pre1="http://www.pre.com">
  <child1 pre1:a="456" pre1:b="456" pre1:c="456" />
</root>
```

Although child1 originally contained the pre1:a attribute, and the PBDOM\_ATTRIBUTE array also contained an item with name a within the namespace URI http://www.pre.com, no exception is thrown. The original pre1:a attribute is replaced by the PBDOM\_ATTRIBUTE array item with name a within the namespace URI http://www.pre.com.

```
PBDOM_BUILDER      pbdom_buildr
PBDOM_DOCUMENT     pbdom_doc
PBDOM_ATTRIBUTE    pbdom_attr_array[]
string             Name[]
long              l = 0
string strXML = "<root xmlns:pre1=~"http://www.pre.com~"><child1 pre1:a=~"123~"/></root>"

try
  pbdom_buildr = Create PBDOM_BUILDER
  pbdom_doc = pbdom_buildr.BuildFromString (strXML)

  Name[1] = "a"
  Name[2] = "b"
  Name[3] = "c"

  for l = 1 to 3
    pbdom_attr_array[l] = Create PBDOM_ATTRIBUTE
    pbdom_attr_array[l].SetName (Name[l])
    pbdom_attr_array[l].SetNamespace ("pre1", &
      "http://www.pre.com", false)
    pbdom_attr_array[l].SetText("456")
  next
```

```
pbdom_doc.GetRootElement().GetChildElement &
    ("child1").SetAttributes(pbdom_attr_array)
pbdom_doc.SaveDocument ("set_attributes.xml")

catch (PBDOM_EXCEPTION except)
    MessageBox ("PBDOM_EXCEPTION", except.GetMessage())
end try
```

## Usage

This method sets the attributes of the DOM element represented by this PBDOM\_ELEMENT object. The supplied array should contain only objects of type PBDOM\_ATTRIBUTE.

When all objects in the supplied array are legal and before the new attributes are added, all old attributes have their parentage set to null (no parent) and the old attribute list is cleared from this PBDOM\_ELEMENT object. This has the effect that any active attribute list (previously obtained with a call to GetAttributes) also changes to reflect the new situation with the old attributes. In addition, all PBDOM\_ATTRIBUTES in the supplied array have their parentage set to this current PBDOM\_ELEMENT object.

Passing an empty array clears the existing attributes of this PBDOM\_ELEMENT object.

This method fails and an exception is thrown if the PBDOM\_ATTRIBUTE array contains two or more PBDOM\_ATTRIBUTES with the same name and namespace URI.

No exception is thrown if this PBDOM\_ELEMENT object contains an existing attribute whose name and namespace URI matches one of the PBDOM\_ATTRIBUTE array items. All the existing attributes of this PBDOM\_ELEMENT object are removed, so it does not matter whether any existing attribute matches any of the PBDOM\_ATTRIBUTE items in the array in terms of name and namespace URI.

In the event of an exception, the original attributes of the PBDOM\_ELEMENT object remain unchanged, and the PBDOM\_ATTRIBUTES in the supplied array are not altered.

If any PBDOM\_ATTRIBUTE has been created to represent any original attribute, it is still valid, but the attribute it represents has been detached from the original owner element. Calling GetOwnerElementObject on this PBDOM\_ATTRIBUTE returns a null value.

## See also

[GetAttribute](#)

[GetAttributes](#)

[GetAttributeValue](#)

[HasAttributes](#)

[SetAttribute](#)

## 13.1.36 SetContent

### Description

Sets the content of the PBDOM\_ELEMENT object using an array containing PBDOM\_OBJECT objects legal for a PBDOM\_ELEMENT object. Any existing children of the PBDOM\_ELEMENT object are removed when the SetContent method is invoked.

If the input array reference is null, all contents of the PBDOM\_ELEMENT object are removed. If the array contains illegal objects, an exception is thrown, and nothing is altered.

## Syntax

```
pbdom_element_name.SetContent(pbdom_object pbdom_object_array[])
```

**Table 13.62:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
pbdom_object_array	An array of PBDOM_OBJECTS to form the contents the PBDOM_ELEMENT object

## Return value

PBDOM\_OBJECT. The PBDOM\_ELEMENT object modified and returned as a PBDOM\_OBJECT.

## Throws

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- If an input PBDOM\_OBJECT array item has not been given a user-defined name.

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If an input PBDOM\_OBJECT array item is not associated with a derived PBDOM\_OBJECT.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT -- If an input PBDOM\_OBJECT array item already has a parent PBDOM\_OBJECT.

EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT -- If an inappropriate PBDOM\_OBJECT array item is found. This happens if the PBDOM\_OBJECT array item is not allowed to be added as a child of a PBDOM\_ELEMENT object (for example, a PBDOM\_DOCUMENT).

EXCEPTION\_HIERARCHY\_ERROR -- If one of the PBDOM\_OBJECT array items, if set as part of the contents of this PBDOM\_ELEMENT object, will cause the current PBDOM\_ELEMENT object to be no longer well formed.

## Examples

The SetContent method is invoked on the following XML fragment:

```
<Telephone_Book>
  <Entry>
    <Particulars>
      <Name>John Doe</Name>
      <Age>21</Age>
      <Phone_Number>1234567</Phone_Number>
    </Particulars>
  </Entry>
</Telephone_Book>
```

The SetContent method is invoked from the following PowerShell code:

```
PBDOM_OBJECT pbdom_obj_array[]

pbdom_obj_array[1] = entry_1
pbdom_obj_array[2] = entry_2

pbdom_doc.GetRootElement().SetContent(pbdom_obj_array)
```

The entry\_1 PBDOM\_ELEMENT object contains the following:

```
<Entry>
  <Particulars>
    <Name>James Gomez</Name>
    <Age>25</Age>
    <Phone_Number>1111111</Phone_Number>
  </Particulars>
</Entry>
```

The entry\_2 PBDOM\_ELEMENT object contains the following:

```
<Entry>
  <Particulars>
    <Name>Mary Jones</Name>
    <Age>22</Age>
    <Phone_Number>2222222</Phone_Number>
  </Particulars>
</Entry>
```

The SetContent method returns the following:

```
<Telephone_Book>
  <Entry>
    <Particulars>
      <Name>James Gomez</Name>
      <Age>25</Age>
      <Phone_Number>1111111</Phone_Number>
    </Particulars>
  </Entry>
  <Entry>
    <Particulars>
      <Name>Mary Jones</Name>
      <Age>22</Age>
      <Phone_Number>2222222</Phone_Number>
    </Particulars>
  </Entry>
</Telephone_Book>
```

## Usage

Only the following PBDOM\_OBJECT types can be validly added to a PBDOM\_ELEMENT object:

- PBDOM\_ELEMENT
- PBDOM\_CDATA
- PBDOM\_COMMENT
- PBDOM\_ENTITYREFERENCE
- PBDOM\_PROCESSINGINSTRUCTION
- PBDOM\_TEXT

## See also

[AddContent Syntax 1](#)

[AddContent Syntax 2](#)

[GetContent](#)

[InsertContent](#)[RemoveContent](#)

### 13.1.37 SetDocument

#### Description

Sets a PBDOM\_DOCUMENT as parent of a PBDOM\_ELEMENT object, making the PBDOM\_ELEMENT object the root element.

#### Syntax

```
pbdom_element_name.SetDocument(pbdom_document pbdom_document_ref)
```

**Table 13.63:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
pbdom_document_ref	The PBDOM_DOCUMENT to be set as the owner document and parent of this PBDOM_ELEMENT object

#### Return value

PBDOM\_ELEMENT. The modified PBDOM\_ELEMENT object.

#### Usage

The PBDOM\_OBJECT referenced must be a PBDOM\_DOCUMENT object. The PBDOM\_ELEMENT object must not already have a parent object. If the target PBDOM\_DOCUMENT already has a root element, the existing root element is replaced by the new PBDOM\_ELEMENT object.

### 13.1.38 SetName

#### Description

Sets the local name of a PBDOM\_ELEMENT object. This name refers to the local portion of the element tag name.

#### Syntax

```
pbdom_element_name.SetName(string strName)
```

**Table 13.64:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strName	The new local name for the PBDOM_ELEMENT object

#### Return value

Boolean.

Returns true if the local name of the PBDOM\_ELEMENT object has been changed, and false otherwise.

## Examples

The SetName method is invoked for the abc element of the following XML fragment:

```
<abc>My Data</abc>
```

The SetName method is invoked in the following PowerShell code, in which the PBDOM\_ELEMENT object elem represents the abc element.

```
elem.SetName("def")
```

The following XML results:

```
<def>My Data</def>
```

Since the elem object still represents the same element, calling the SetName method changes the def element.

## See also

[GetName](#)

### 13.1.39 SetNamespace

#### Description

Sets the namespace for a PBDOM\_ELEMENT object. If the namespace prefix and URI provided are empty strings, SetNamespace assigns no namespace to the PBDOM\_ELEMENT object.

#### Syntax

```
pbdom_element_name.SetNamespace(string strNamespacePrefix, string strNamespaceUri,
boolean bVerifyNamespace)
```

**Table 13.65:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strNamespacePrefix	Prefix of the namespace to be set for the PBDOM_ELEMENT object
strNamespaceUri	URI of the namespace to be set for the PBDOM_ELEMENT object
bVerifyNamespace	A boolean value indicating whether verification should be performed to ensure that the provided namespace prefix and URI have been declared either within this PBDOM_ELEMENT object or in an ancestor PBDOM_ELEMENT object

#### Return value

Long.

Returns 0 for success and -1 if no in-scope namespace declaration matching the input prefix and URI exists.

**Throws**

EXCEPTION\_INVALID\_ARGUMENT -- If any of the input arguments is invalid, for example, null.

EXCEPTION\_INVALID\_NAME -- If the input namespace prefix or URI is invalid.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- If a memory allocation failure occurred during the execution of this method.

EXCEPTION\_INTERNAL\_XML\_ENGINE\_ERROR -- If an internal XML engine failure occurred during the execution of this method.

**Usage**

If `bVerifyNamespace` is set to true and the namespace prefix and URI have not been declared, `SetNamespace` returns a value of -1 and fails.

If `bVerifyNamespace` is set to false, `SetNamespace` sets the namespace of the PBDOM\_ELEMENT object to the specified prefix and URI. It is the responsibility of the PBDOM user to ensure that such a namespace is declared and is in scope for this PBDOM\_ELEMENT object before the document is saved and converted to an XML document.

**See also**

[AddNamespaceDeclaration](#)

[GetNamespacePrefix](#)

[GetNamespaceUri](#)

[GetQualifiedName](#)

[RemoveNamespaceDeclaration](#)

**13.1.40 SetParentObject****Description**

Sets the referenced PBDOM\_OBJECT as the parent of the PBDOM\_ELEMENT object from which the method is invoked.

**Syntax**

```
pbdom_element_name.SetParentObject(pbdom_object pbdom_object_ref)
```

**Table 13.66:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
pbdom_object_ref	The PBDOM_OBJECT to be set as the parent of this PBDOM_ELEMENT object

**Return value**

PBDOM\_OBJECT. The PBDOM\_ELEMENT object modified and returned as a PBDOM\_OBJECT.

**Throws**

**EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE** -- If the input PBDOM\_OBJECT is not associated with a derived PBDOM\_OBJECT.

**EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT** -- The input PBDOM\_OBJECT already has a parent.

**EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT** -- If the input PBDOM\_OBJECT is not allowed to be the parent of a PBDOM\_ELEMENT object.

**EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT** -- If the input PBDOM\_OBJECT is nameable and has not been named.

### Usage

If the class of the referenced PBDOM\_OBJECT is PBDOM\_DOCUMENT, then the behavior of SetParentObject is identical to that of the SetDocument method. If the class of the referenced PBDOM\_OBJECT is PBDOM\_ELEMENT, SetParentObject sets the referenced object as the parent of the PBDOM\_ELEMENT object from which the method is invoked. If the referenced PBDOM\_OBJECT is of any other class, an exception is thrown.

### See also

[GetOwnerDocumentObject](#)

[GetParentObject](#)

## 13.1.41 SetText

### Description

Sets the content of a PBDOM\_ELEMENT object to the text provided.

### Syntax

```
pbdom_element_name.SetText(string strText)
```

**Table 13.67:**

Argument	Description
pbdom_element_name	The name of a PBDOM_ELEMENT object
strText	String to be set as the content of the PBDOM_ELEMENT object

### Return value

PBDOM\_OBJECT. The PBDOM\_ELEMENT object modified and returned as a PBDOM\_OBJECT.

### Usage

Existing text content and non-text content are replaced by the text provided in strText. A value of null for strText is equivalent to an empty string value. If the PBDOM\_ELEMENT is to have both text content and nested elements, use the SetContent method instead of SetText.

### See also

[GetText](#)

[GetTextNormalize](#)



[GetTextTrim](#)

## 14 PBDOM\_EXCEPTION Class

### About this chapter

This chapter lists PBDOM exception codes and describes the PBDOM\_EXCEPTION class.

### 14.1 PBDOM exceptions

PBDOM defines an exception class derived from the standard PowerBuilder Exception class. This class extends the Exception class with a method, GetExceptionCode, that returns the unique code that identifies the exception being thrown.

The following table lists PBDOM exceptions and their code values. The circumstances in which each exception is thrown are described after the table.

**Table 14.1: PBDOM exceptions and code values**

Exception	Value
<a href="#">EXCEPTION USE OF UNNAMED PBDOM OBJECT</a>	1
<a href="#">EXCEPTION WRONG DOCUMENT ERROR</a>	2
<a href="#">EXCEPTION MULTIPLE ROOT ELEMENTS</a>	3
<a href="#">EXCEPTION INAPPROPRIATE USE OF PBDOM OBJECT</a>	4
<a href="#">EXCEPTION PBDOM OBJECT INVALID FOR USE</a>	5
<a href="#">EXCEPTION PBDOM OBJECT ALREADY HAS PARENT</a>	6
<a href="#">EXCEPTION MULTIPLE DOCTYPE</a>	7
<a href="#">EXCEPTION ILLEGAL PBOBJECT</a>	8
<a href="#">EXCEPTION WRONG PARENT ERROR</a>	9
<a href="#">EXCEPTION INVALID ARGUMENT</a>	10
<a href="#">EXCEPTION INVALID NAME</a>	11
<a href="#">EXCEPTION DATA CONVERSION</a>	12
<a href="#">EXCEPTION MEMORY ALLOCATION FAILURE</a>	13
<a href="#">EXCEPTION INTERNAL XML ENGINE ERROR</a>	14
<a href="#">EXCEPTION MULTIPLE XMLDECL</a>	15
<a href="#">EXCEPTION INVALID STRING</a>	16
<a href="#">EXCEPTION INVALID OPERATION</a>	17
<a href="#">EXCEPTION HIERARCHY ERROR</a>	18
<a href="#">EXCEPTION PBDOM OBJECT ALREADY HAS OWNER</a>	19
<a href="#">EXCEPTION PBDOM NOT INITIALIZED</a>	20

#### 14.1.1 PBDOM exception descriptions

##### 14.1.1.1 EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT

Code Value: 1

This exception is thrown when you use a nameable PBDOM\_OBJECT -- for example, to invoke a method or serve as a parameter -- without first being given a user-defined name.

#### **14.1.1.2 EXCEPTION\_WRONG\_DOCUMENT\_ERROR**

Code Value: 2

This exception is thrown when you use incorrect PBDOM\_DOCUMENT objects when performing a PBDOM operation. For example, in a RemoveContent method call, if the PBDOM\_OBJECT you want to remove is not from the same document as the active PBDOM\_DOCUMENT whose RemoveContent method is being invoked, this exception is thrown.

#### **14.1.1.3 EXCEPTION\_MULTIPLE\_ROOT\_ELEMENT**

Code Value: 3

This exception is thrown when a PBDOM method call causes a PBDOM\_DOCUMENT to contain more than one root element.

For example, in an AddContent method call, if the input PBDOM\_OBJECT to add is a PBDOM\_ELEMENT and the active PBDOM\_DOCUMENT already contains a root element, this exception is thrown.

#### **14.1.1.4 EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT**

Code Value: 4

This exception is thrown when a PBDOM\_OBJECT is used in an inappropriate manner. A typical scenario is one in which a PBDOM method call results in the violation of the well-formedness of a PBDOM\_DOCUMENT.

For example, in an AddContent method invoked on a PBDOM\_DOCUMENT object, only PBDOM\_OBJECTs of class PBDOM\_ELEMENT, PBDOM\_COMMENT, PBDOM\_PROCESSINGINSTRUCTION, and PBDOM\_DOCTYPE can be added. The inclusion of PBDOM\_OBJECTs of any other class results in this exception being thrown.

#### **14.1.1.5 EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE**

Code Value: 5

This exception is thrown when an invalid PBDOM\_OBJECT is used, either directly to invoke a method, or as a parameter.

Situations where a PBDOM\_OBJECT is deemed invalid include those where a PBDOM\_OBJECT is instantiated as a PBDOM\_OBJECT and not as a derived class object. They also include the situation where a PBDOM\_CHARACTERDATA object is instantiated directly as a PBDOM\_CHARACTERDATA object.

#### **14.1.1.6 EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT**

Code Value: 6

This exception occurs when a PBDOM\_OBJECT is set to be the child of another PBDOM\_OBJECT, but the prospective child already has a parent PBDOM\_OBJECT.

Examples of such method calls include the AddContent method and the SetParentObject, SetContent, and InsertContent methods of all classes derived from PBDOM\_OBJECT classes.

#### **14.1.1.7 EXCEPTION\_MULTIPLE\_DOCTYPE**

Code Value: 7

This exception is thrown when a PBDOM method call causes a PBDOM\_DOCUMENT to contain more than one DOCTYPE.

For example, in an AddContent method call, if the input PBDOM\_OBJECT to add is a PBDOM\_DOCTYPE and the active PBDOM\_DOCUMENT already contains a DOCTYPE DOM Node, this exception is thrown.

#### **14.1.1.8 EXCEPTION\_ILLEGAL\_PBOBJECT**

Code Value: 8

This exception is thrown in method calls that take an array of PBDOM\_OBJECTs in which one of the array items is invalid. A PBDOM\_OBJECT array item is deemed to be invalid when it has been specifically set to null or has not been initialized properly.

#### **14.1.1.9 EXCEPTION\_WRONG\_PARENT\_ERROR**

Code Value: 9

This exception is thrown when an incorrect parent/child relationship error is encountered during a PBDOM operation.

Method calls in which this exception might be thrown include InsertContent and RemoveContent. These methods involve at least one PBDOM\_OBJECT parameter that is assumed to be a child of the PBDOM\_OBJECT to which the method is applied. If this parameter is not a child of the current PBDOM\_OBJECT, this exception is thrown.

#### **14.1.1.10 EXCEPTION\_INVALID\_ARGUMENT**

Code Value: 10

This exception is thrown when an input PBDOM\_OBJECT parameter to a method is invalid. This can happen if it has not been initialized properly, or if it is a null object reference.

This exception might also be thrown when an input string parameter to a method is invalid. This can happen if the string has been set to null using the PowerScript SetNull function.

#### **14.1.1.11 EXCEPTION\_INVALID\_NAME**

Code Value: 11

This exception is thrown when a name is supplied as a parameter and the name does not conform to the W3C specifications for an XML name or namespace prefix or namespace URI.

Methods in which this exception might be thrown include the SetName, SetNamespace, and SetNamespace methods.

**14.1.1.12 EXCEPTION\_DATA\_CONVERSION**

Code Value: 12

This exception is thrown when you attempt to perform a data conversion operation and the conversion fails. This exception is thrown only in the PBDOM\_ATTRIBUTE object's Get methods, for example, GetDateValue in PBDOM\_ATTRIBUTE.

**14.1.1.13 EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE**

Code Value: 13

This exception is thrown when insufficient memory is encountered while executing a method. PBDOM internally allocates, frees, and reallocates memory for storing strings, structures, and so on. Each memory allocation might fail, and if this occurs, this exception is thrown.

**14.1.1.14 EXCEPTION\_INTERNAL\_XML\_ENGINE\_ERROR**

Code Value: 14

This exception is thrown when an internal error occurs that involves the XML engine used by PBDOM. PBDOM currently uses the Xerces XML parser as the underlying device for processing XML documents and for building up and sustaining the DOM tree.

There may be problems in the low-level XML parser engine, and if one is encountered, this exception, which is rare, might be thrown.

**14.1.1.15 EXCEPTION\_MULTIPLE\_XMLDECL**

Code Value: 15

This exception is thrown when a PBDOM method call causes a PBDOM\_DOCUMENT to contain more than one XML declaration.

For example, in a SetContent method call invoked on a PBDOM\_DOCUMENT object, if the input PBDOM\_OBJECT array contains more than one PBDOM\_PROCESSINGINSTRUCTION that is constructed as an XML declaration, this exception is thrown.

**14.1.1.16 EXCEPTION\_INVALID\_STRING**

Code Value: 16

This exception is thrown when a string is supplied as a parameter to a method that sets a text or attribute value, and the string contains characters that do not conform to the W3C specifications for acceptable XML characters.

Methods in which this exception might be thrown include SetText in PBDOM\_ATTRIBUTE and SetAttribute in PBDOM\_ELEMENT.

**14.1.1.17 EXCEPTION\_INVALID\_OPERATION**

Code Value: 17

This exception is thrown when a method call could potentially cause severe and unexpected problems to the currently running PowerBuilder application.

#### 14.1.1.18 EXCEPTION\_HIERARCHY\_ERROR

Code Value: 18

This exception is thrown when a method call violates the well-formedness or validity of a PBDOM\_DOCUMENT.

#### 14.1.1.19 EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_OWNER

Code Value: 19

This exception is thrown when a PBDOM\_ELEMENT is set as the owner of a PBDOM\_ATTRIBUTE when the specified PBDOM\_ATTRIBUTE already has an owner PBDOM\_ELEMENT.

#### 14.1.1.20 EXCEPTION\_PBDOM\_NOT\_INITIALIZED

Code Value: 20

This exception is thrown in rare circumstances in which the PBDOM engine has failed to be initialized or has been uninitialized prematurely. In such situations, an exception is thrown to prevent a crash.

## 14.2 PBDOM\_EXCEPTION

### Description

The PBDOM\_EXCEPTION class is derived from the PowerBuilder Exception class.

### Methods

This class extends the Exception class with one method that returns the unique code that identifies the exception being thrown:

[GetExceptionCode](#)

#### 14.2.1 GetExceptionCode

### Description

Returns the code of the exception being thrown.

### Syntax

```
pbdom_exception.GetExceptionCode()
```

Table 14.2:

Argument	Description
pbdom_exception	The name of a PBDOM_EXCEPTION object

### Return value

Long.

The code value associated with the exception being thrown.

### Examples

In this example, an attempt to call the PBDOM\_ELEMENT GetAttribute method on the root element of a PBDOM\_DOCUMENT with the parameter xmlns:nuskin causes an exception to be thrown, because the name is not a valid NCName (no-colon-name). The correct way to get an attribute that belongs to a namespace is to use the namespace version of the PBDOM\_ELEMENT [GetAttribute](#) method.

The EXCEPTION\_INVALID\_NAME (code value 11) exception is thrown and is displayed in a message box:

```
PBDOM_DOCUMENT    pbdom_doc1
PBDOM_DOCUMENT    pbdom_get_doc
PBDOM_ELEMENT     pbdom_elem_root
PBDOM_ATTRIBUTE   pbdom_attr
PBDOM_OBJECT      pbdom_obj

try
    pbdom_doc1 = Create PBDOM_DOCUMENT

    pbdom_doc1.NewDocument("nuskin", &
        "http://www.nuskin.com", "nuskin:root", "", "")
    pbdom_elem_root = pbdom_doc1.GetRootElement()
    pbdom_attr = &
        pbdom_elem_root.GetAttribute("xmlns:nuskin")

catch (PBDOM_EXCEPTION pbdom_except)
    MessageBox ("Exception", "Code : " &
        + string(pbdom_except.GetExceptionCode()) &
        + "~r~nText : " + pbdom_except.Text)
end try
```

## Usage

For a list of exception codes, see [PBDOM exceptions](#). For a description of the conditions under which each exception can occur, see [PBDOM exception descriptions](#).

## See also

[GetAttribute Syntax 2](#) (PBDOM\_ELEMENT)

GetMessage and SetMessage in Section 10.291, “GetMessage” in *PowerScript Reference* and Section 10.729, “SetMessage” in *PowerScript Reference*.

# 15 PBDOM\_OBJECT Class

## About this chapter

This chapter describes the PBDOM\_OBJECT class.

## 15.1 PBDOM\_OBJECT

### Description

A PBDOM\_OBJECT serves as the base class for all the PBDOM classes. It contains all the basic methods required by derived classes. The derived classes of a PBDOM\_OBJECT each inherit the base methods of a PBDOM\_OBJECT, and additionally contain their own specialized methods.

### Methods

PBDOM\_OBJECT has the following methods:

[AddContent](#)

[Clone](#)

[Detach](#)

[Equals](#)

[GetContent](#)

[GetOwnerDocumentObject](#)

[GetName](#)

[GetObjectClass](#)

[GetObjectClassString](#)

[GetParentObject](#)

[GetText](#)

[GetTextNormalize](#)

[GetTextTrim](#)

[HasChildren](#)

[InsertContent](#)

[IsAncestorObjectOf](#)

[RemoveContent](#)

[SetContent](#)

[SetName](#)

[SetParentObject](#)

### 15.1.1 AddContent

#### Description

Adds a new PBDOM\_OBJECT into the current PBDOM\_OBJECT.



## Syntax

```
pbdom_object_name.AddContent(pbdom_object pbdom_object_ref)
```

**Table 15.1:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT
pbdom_object_ref	The PBDOM_OBJECT to add

### Return value

PBDOM\_OBJECT.

The return value is the newly modified PBDOM\_OBJECT.

### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object or the input PBDOM\_OBJECT is not associated with a derived PBDOM\_OBJECT class object.

EXCEPTION\_INVALID\_ARGUMENT -- Input argument is invalid.

### Usage

When a new PBDOM\_OBJECT is added to the current one, the new PBDOM\_OBJECT becomes a child node of the current PBDOM\_OBJECT.

### See also

[GetContent](#)

[InsertContent](#)

[RemoveContent](#)

[SetContent](#)

## 15.1.2 Clone

### Description

Creates a general duplicate of the current PBDOM\_OBJECT.

### Syntax

```
pbdom_object_name.Clone(boolean bDeep)
```

**Table 15.2:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT.
bDeep	A boolean specifying whether a deep or shallow clone is returned. Values are true for a deep clone and false for a shallow clone.

### Return value

PBDOM\_OBJECT. The return value is the clone of the PBDOM\_OBJECT.

### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object is not associated with a derived PBDOM\_OBJECT class object.

### Usage

The Clone method creates a general duplicate of the current PBDOM\_OBJECT. If the bDeep parameter is true, a deep clone is returned; otherwise, a shallow clone is returned.

A PBDOM\_OBJECT clone does not have a parent; however, it resides in the same PBDOM\_DOCUMENT as its original. If the original PBDOM\_OBJECT is standalone, the clone is also standalone.

If general, if bDeep is true, the Clone method recursively clones the subtree under the PBDOM\_OBJECT. If bDeep is false, the Clone method clones only the PBDOM\_OBJECT itself, together with as much information as possible.

---

### Cloning is class specific

Cloning is not uniform across all PBDOM\_OBJECT classes. See the documentation for each class for specific information.

---

## 15.1.3 Detach

### Description

Detaches a PBDOM\_OBJECT from its parent.

### Syntax

```
pbdom_object_name.Detach()
```

**Table 15.3:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT

### Return value

PBDOM\_OBJECT.

### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object is not associated with a derived PBDOM\_OBJECT class object.

### Examples

This example detaches the root element of a PBDOM\_DOCUMENT called pbdom\_doc from its parent object -- that is, from the PBDOM\_DOCUMENT itself. Then, it attempts to obtain the parent PBDOM\_OBJECT and tests whether it is null using the IsValid method:

```
pbdom_obj = pbdom_doc.GetRootElement()
pbdom_obj.Detach()
pbdom_parent_obj = pbdom_obj.GetParentObject()
if (not IsValid(pbdom_parent_obj)) then
    MessageBox ("Invalid", "Root Element has no Parent")
end if
```

### Usage

If the PBDOM\_OBJECT has no parent, this method does nothing.

### 15.1.4 Equals

#### Description

Tests for the equality of a referenced PBDOM\_OBJECT.

#### Syntax

```
pbdom_object_name.Equals(pbdom_object pbdom_object_ref)
```

**Table 15.4:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT
pbdom_object_ref	The PBDOM_OBJECT to test for equality with the current PBDOM_OBJECT

#### Return value

Boolean.

Returns true if the current PBDOM\_OBJECT is equivalent to the input PBDOM\_OBJECT, and false otherwise.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object or the input PBDOM\_OBJECT is not associated with a derived PBDOM\_OBJECT class object.

EXCEPTION\_INVALID\_ARGUMENT -- The input PBDOM\_OBJECT is invalid. This can happen if the object has not been initialized properly or is a null object reference.

### 15.1.5 GetContent

#### Description

Obtains an array of PBDOM\_OBJECT objects, each of which is a child node of the called PBDOM\_OBJECT.

#### Syntax

```
pbdom_object_name.GetContent(ref pbdom_object pbdom_object_array[ ])
```

**Table 15.5:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT
pbdom_object_array	A reference to an array of PBDOM_OBJECT objects that will receive the PBDOM_OBJECT objects

#### Return value

Boolean.

Returns true for success, and false otherwise.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object is not associated with a derived PBDOM\_OBJECT class object.

## Usage

The returned array is passed by reference, with items in the same order in which they appear in the PBDOM\_OBJECT. Any changes to any item of the array affect the actual item to which it refers.

### See also

[AddContent](#)

[InsertContent](#)

[RemoveContent](#)

[SetContent](#)

## 15.1.6 GetName

### Description

Obtains the name of the current PBDOM\_OBJECT. The returned string depends on the type of DOM Object that is contained within a PBDOM\_OBJECT.

Table 15.6:

DOM Object Type	Return Value
PBDOM_DOCTYPE	"#document"
PBDOM_ELEMENT	<p>The local tag name of the element, without any namespace prefixes.</p> <p>For example, if the element is: <code>&lt;abc&gt;Value&lt;/abc&gt;</code>, then the string returned from <code>GetName</code> is "abc".</p> <p>Also, if the tag name of the element contains a namespace prefix, the prefix is not included in the returned string.</p> <p>For example, if the element is:  <code>&lt;MyMusic:CD xmlns:MyMusic=</code>, then the string returned from <code>GetName</code> is "CD".  <code>"http://www.MyMusicDiscs.com"/&gt;</code></p>
PBDOM_ATTRIBUTE	<p>The local name of the attribute itself, without a namespace.</p> <p>For example, if the element with the attribute is: <code>&lt;abc ATTRIBUTE_1="My Attribute"&gt;</code>, then <code>GetName</code> returns "ATTRIBUTE_1".</p> <p>If the name of the attribute contains a namespace prefix, then the prefix is not included in the returned string.</p> <p>For example, if the element with an attribute is: <code>&lt;MyMusic:CD xmlns:MyMusic=</code>, then <code>GetName</code> returns the string "Type".</p>

DOM Object Type	Return Value
	"http://www.MyMusicDiscs.com" MyMusic:Type="Jazz"/>
PBDOM_CDATA	"#cdata-section"
PBDOM_COMMENT	"#comment"
PBDOM_DOCTYPE	The name that was given to the doctype object itself.  For example, if the DOCTYPE declaration is: <!DOCTYPE d_grid_object >, then GetName returns "d_grid_object".
PBDOM_PROCESSINGINSTRUCTION	The name that was given to the processing instruction itself.  For example, if the processing instruction definition is: <?works document="hello.doc" data="hello.wks" ?>, then GetName returns "works".
PBDOM_TEXT	"#text"

## Syntax

```
pbdom_object_name.GetName()
```

**Table 15.7:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT

## Return value

The following table lists the return values, based on the type of DOM Object contained within the PBDOM\_OBJECT:

## Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If this PBDOM\_OBJECT is not a reference to an object derived from PBDOM\_OBJECT.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- Insufficient memory was encountered while executing this method.

## Usage

A PBDOM\_OBJECT cannot be instantiated directly.

## See also

[SetName](#)

### 15.1.7 GetObjectClass

#### Description

Returns a long integer code that indicates the class of this PBDOM\_OBJECT.

## Syntax

```
pbdom_object_name.GetObjectClass()
```

**Table 15.8:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT

## Return value

Long.

A code that indicates the class of the current PBDOM\_OBJECT.

## Usage

This method returns the following possible values:

**Table 15.9:**

Class	Long integer value
UNKNOWN (indicates an error)	0
PBDOM_OBJECT (the base class)	1
PBDOM_DOCUMENT	2
PBDOM_ELEMENT	3
PBDOM_DOCTYPE	4
PBDOM_ATTRIBUTE	5
PBDOM_CHARACTERDATA	6
PBDOM_TEXT	7
PBDOM_CDATA	8
PBDOM_COMMENT	9
PBDOM_PROCESSINGINSTRUCTION	10
PBDOM_ENTITYREFERENCE	11

## See also

[GetObjectClassString](#)

## 15.1.8 GetObjectClassString

### Description

Returns a string form of the class of the PBDOM\_OBJECT.

### Syntax

```
pbdom_object_name.GetObjectClassString()
```

**Table 15.10:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT

## Return value

String.

A string that indicates the class of the current PBDOM\_OBJECT.

### Usage

This method returns the following possible values:

**Table 15.11:**

Class	String returned
PBDOM_OBJECT	pbdom_object
PBDOM_DOCUMENT	pbdom_document
PBDOM_ELEMENT	pbdom_element
PBDOM_ENTITYREFERENCE	pbdom_entityreference
PBDOM_DOCTYPE	pbdom_doctype
PBDOM_ATTRIBUTE	pbdom_attribute
PBDOM_CHARACTERDATA	pbdom_characterdata
PBDOM_TEXT	pbdom_text
PBDOM_CDATA	pbdom_cdata
PBDOM_COMMENT	pbdom_comment
PBDOM_PROCESSINGINSTRUCTION	pbdom_processinginstruction

### See also

[GetObjectClass](#)

## 15.1.9 GetOwnerDocumentObject

### Description

Returns the owning PBDOM\_DOCUMENT of the current PBDOM\_OBJECT.

### Syntax

```
pbdom_object_name.GetOwnerDocumentObject()
```

**Table 15.12:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT

### Return value

PBDOM\_DOCUMENT.

### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object is not associated with a derived PBDOM\_OBJECT class object.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- Insufficient memory was encountered while executing this method.

### Usage

The owning PBDOM\_DOCUMENT of the current PBDOM\_OBJECT is null if PBDOM\_OBJECT is not owned by any PBDOM\_DOCUMENT, or if the current PBDOM\_OBJECT is itself a PBDOM\_DOCUMENT object.

### See also

[GetParentObject](#)

[SetParentObject](#)

## 15.1.10 GetParentObject

### Description

Returns the parent PBDOM\_OBJECT of the current PBDOM\_OBJECT.

### Syntax

```
pbdom_object_name.GetParentObject()
```

**Table 15.13:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT

### Return value

PBDOM\_OBJECT.

### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object is not associated with a derived PBDOM\_OBJECT class object.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- Insufficient memory was encountered while executing this method.

### Examples

Using the GetRootElement method, the root element of a PBDOM\_DOCUMENT called pbdom\_doc is returned into a PBDOM\_OBJECT called pbdom\_obj. The GetParentObject method returns the parent of the root element, which is the PBDOM\_DOCUMENT itself, and stores it in pbdom\_parent\_obj.

The GetObjectClassString method returns the class name of pbdom\_parent\_obj as a string that is displayed in a message box:

```
pbdom_document pbdom_doc
pbdom_object pbdom_obj
pbdom_object pbdom_parent_obj
string strClassName
// code omitted
...
pbdom_doc = pbdombuilder_new.BuildFromString (strXML)
pbdom_obj = pbdom_doc.GetRootElement()
pbdom_parent_obj = pbdom_obj.GetParentObject()
strClassName = pbdom_parent_obj.GetObjectClassString()
MessageBox ("Parent Class Name", strClassName)
```

### Usage

If the PBDOM\_OBJECT has no parent, null is returned.



**See also**[GetOwnerDocumentObject](#)[SetParentObject](#)**15.1.11 GetText****Description**

Obtains the text data that is contained within the current PBDOM\_OBJECT.

**Syntax**

```
pbdom_object_name.GetText ( )
```

**Table 15.14:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT

**Return value**

String.

The following table lists the return values, based on the type of DOM Object contained within a PBDOM\_OBJECT:

**Table 15.15:**

DOM Object Type	Return Value
PBDOM_ELEMENT	<p>The concatenation of the text values of all the TEXT nodes contained within the PBDOM_ELEMENT.</p> <p>If the PBDOM_ELEMENT definition is <code>&lt;abc&gt;Root Element Data&lt;data&gt;ABC Data&lt;/data&gt; now with extra info &lt;/abc&gt;</code>, then <code>GetText</code> returns "Root Element Data now with extra info".</p> <p>Extra Spaces</p> <p>There are extra spaces between the word "Data" and "now" and again after the word "info". They are there because they originally exist in the text.</p> <p>If the PBDOM_ELEMENT definition is: <code>&lt;abc&gt;Root Element Data&lt;/abc&gt;</code>, then <code>GetText</code> returns "Root Element Data".</p>
PBDOM_ATTRIBUTE	<p>The text data contained within the PBDOM_ATTRIBUTE object.</p> <p>If the element with an attribute is <code>&lt;abc ATTRIBUTE_1="My Attribute"&gt;</code>, then <code>GetText</code> returns "My Attribute".</p>

DOM Object Type	Return Value
PBDOM_TEXT	<p>The text data contained within the PBDOM_TEXT object itself.</p> <p>For example, suppose there is the following element:</p> <pre data-bbox="807 443 1393 472">&lt;abc&gt;MY TEXT&lt;/abc&gt;</pre> <p>If there is a PBDOM_TEXT object to represent the text node "MY TEXT", then calling GetText on the PBDOM_TEXT returns the string "MY TEXT"</p>
PBDOM_CDATA	<p>The string data that is contained within the CDATA section itself. For example, suppose there is the following CDATA:</p> <pre data-bbox="807 801 1393 913">&lt;![CDATA[ They're saying "x &lt; y" &amp; that "z &gt; y" so I guess that means that z &gt; x ]]&gt;</pre> <p>If there is a PBDOM_CDATA to represent the above CDATA section, then calling GetText on it returns the following string:</p> <pre data-bbox="807 1081 1393 1160">They're saying "x &lt; y" &amp; that "z &gt; y" so I guess that means that z &gt; x</pre>
PBDOM_COMMENT	<p>The string data that is contained within the COMMENT itself. For example, suppose there is the following COMMENT:</p> <pre data-bbox="807 1317 1393 1346">&lt;!--This is some comment. --&gt;</pre> <p>If there is a PBDOM_COMMENT to represent the above COMMENT, then calling GetText on it returns the following string:</p> <pre data-bbox="807 1507 1393 1536">This is some comment.</pre>

### Throws

**EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE** -- This PBDOM\_OBJECT object is not associated with a derived PBDOM\_OBJECT class object.

**EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE** -- Insufficient memory was encountered while executing this method.

### Usage

This method returns meaningful data only if the PBDOM\_OBJECT is of a type that can contain text nodes, CDATA sections, or basic text. These include:

- PBDOM\_ELEMENT
- PBDOM\_ATTRIBUTE

- PBDOM\_TEXT
- PBDOM\_CDATA
- PBDOM\_COMMENT

The PBDOM\_TEXT, PBDOM\_CDATA, and PBDOM\_COMMENT objects are special cases that cause the GetText method to return the text data that is intrinsically contained within the objects. A PBDOM\_TEXT object is basically a DOM text node and therefore does not hold any child text nodes. A PBDOM\_CDATA object represents a DOM CDATA object, and therefore does not hold any child DOM nodes. The same rule applies to a PBDOM\_COMMENT object.

**See also**

[GetTextNormalize](#)

[GetTextTrim](#)

### 15.1.12 GetTextNormalize

**Description**

Gets the text data that is contained in the current PBDOM\_OBJECT with all surrounding whitespace characters removed and internal whitespace characters normalized to a single space.

**Syntax**

```
pbdom_object_name.GetTextNormalize()
```

**Table 15.16:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT

**Return value**

String.

The normalized text content of the current PBDOM\_OBJECT, or an empty string if there is no text content.

**Throws**

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object is not associated with a derived PBDOM\_OBJECT class object.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- Insufficient memory was encountered while executing this method.

**Usage**

This method returns meaningful data only if the PBDOM\_OBJECT is of a type that can contain text nodes or CDATA sections, or of a type that intrinsically contains basic text. These types are:

- PBDOM\_ELEMENT
- PBDOM\_ATTRIBUTE

- PBDOM\_TEXT
- PBDOM\_CDATA
- PBDOM\_COMMENT

The PBDOM\_TEXT, PBDOM\_CDATA, and PBDOM\_COMMENT classes are special cases that cause the GetTextNormalize method to return the intrinsic text data contained within their instances. A PBDOM\_TEXT object represents a DOM text node, therefore it does not hold any child DOM Nodes. PBDOM\_CDATA object is a representation of a DOM CDATA object and does not hold any child DOM Nodes. Nor does PBDOM\_COMMENT contain any child DOM Nodes.

The following table lists the return values based on the type of actual DOM Object contained within PBDOM\_OBJECT:

**Table 15.17:**

DOM Object Type	Return Value
PBDOM_ELEMENT	<p>The normalized text of the concatenation of the text values of all the TEXT Nodes and CDATA Sections contained within the PBDOM_ELEMENT.</p> <p>Suppose there is a PBDOM_ELEMENT defined as follows:</p> <pre data-bbox="807 1131 1391 1216">&lt;abc&gt; Root Element Data &lt;data&gt;ABC Data &lt;/data&gt; now with extra info &lt;/abc&gt;</pre> <p>GetTextNormalize returns Root Element Data now with extra info.</p> <p>Suppose there is a PBDOM_ELEMENT defined as follows:</p> <pre data-bbox="807 1435 1391 1462">&lt;abc&gt; Root Element Data &lt;/abc&gt;</pre> <p>GetTextNormalize returns Root Element Data.</p> <p>Suppose there is a PBDOM_ELEMENT defined as follows:</p> <pre data-bbox="807 1682 1391 1767">&lt;abc&gt; Root Element Data &lt;![CDATA [ with some cdata text]]&gt;&lt;/abc&gt;</pre> <p>GetTextNormalize returns "Root Element Data with some cdata text".</p>
PBDOM_ATTRIBUTE	<p>The normalized text data contained within the PBDOM_ATTRIBUTE object.</p> <p>Suppose there is an element with an attribute as follows:</p>

DOM Object Type	Return Value
	<pre data-bbox="804 241 1388 300">&lt;abc ATTRIBUTE_1=" My Attribute "&gt;</pre> <p data-bbox="804 322 1331 358">GetTextNormalize returns My Attribute.</p>
PBDOM_TEXT	<p data-bbox="804 376 1353 448">The normalized text data contained within the PBDOM_TEXT object itself.</p> <p data-bbox="804 465 1369 537">For example, suppose there is the following element:</p> <pre data-bbox="804 564 1123 591">&lt;abc&gt; MY TEXT &lt;/abc&gt;</pre> <p data-bbox="804 618 1305 797">If there is a PBDOM_TEXT object to represent the text node "MY TEXT", then calling GetTextNormalize on the PBDOM_TEXT returns the string MY TEXT.</p>
PBDOM_CDATA	<p data-bbox="804 824 1372 967">The normalized string data that is contained within the CDATA section itself. For example, suppose there is the following CDATA:</p> <pre data-bbox="804 1003 1372 1111">&lt;![CDATA[ They're saying "x &lt; y" &amp; that "z &gt; y" so I guess that means that z &gt; x ]]&gt;</pre> <p data-bbox="804 1137 1359 1245">If there is a PBDOM_CDATA to represent the above CDATA section, then calling GetTextNormalize on it returns the string:</p> <pre data-bbox="804 1281 1369 1361">They're saying " x &lt; y " &amp; that "z &gt; y" so I guess that means that z &gt; x</pre> <p data-bbox="804 1388 1369 1568">Note that the initial spaces before "They're" and the trailing space after the last "x" have been removed. Additionally, the spaces between the word "guess" and "that" have been reduced to just one space.</p>
PBDOM_COMMENT	<p data-bbox="804 1594 1372 1693">The normalized string data that is contained within the COMMENT itself. For example, suppose there is the following COMMENT:</p> <pre data-bbox="804 1729 1197 1756">&lt;!-- Comment Here !--&gt;</pre> <p data-bbox="804 1783 1334 1881">Calling GetTextNormalize on the COMMENT returns the string Comment Here !</p>

**See also**[GetText](#)

[GetTextTrim](#)

### 15.1.13 GetTextTrim

#### Description

Gets the text data that is contained in the current PBDOM\_OBJECT with all surrounding whitespace characters removed.

#### Syntax

```
pbdom_object_name.GetTextTrim()
```

**Table 15.18:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT

#### Return value

String.

The trimmed text content of the current PBDOM\_OBJECT, or an empty string if there is no text content or only whitespace characters.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object is not associated with a derived PBDOM\_OBJECT class object.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- Insufficient memory was encountered while executing this method.

#### Usage

This method returns meaningful data only if the PBDOM\_OBJECT is of a type that can contain TEXT NODEs or CDATA Sections, or of a type that intrinsically contains basic text. These types are:

- PBDOM\_ELEMENT
- PBDOM\_ATTRIBUTE
- PBDOM\_TEXT
- PBDOM\_CDATA
- PBDOM\_COMMENT

The PBDOM\_TEXT, PBDOM\_CDATA, and PBDOM\_COMMENT classes are special cases that cause the GetTextTrim method to return the intrinsic text data contained within their instances. A PBDOM\_TEXT object represents a DOM text node, so it does not hold any child DOM Nodes. PBDOM\_CDATA object is a representation of a DOM CDATA object and does not hold any child DOM Nodes, nor does PBDOM\_COMMENT contain any child DOM Nodes.

The following table lists the return values based on the type of actual DOM Object contained within PBDOM\_OBJECT:

**Table 15.19:**

DOM Object Type	Return Value
PBDOM_ELEMENT	<p>The trimmed concatenation of the text values of all the TEXT Nodes and CDATA Sections contained within the PBDOM_ELEMENT. Surrounding whitespace characters are removed.</p> <p>Suppose there is a PBDOM_ELEMENT defined as follows:</p> <pre data-bbox="804 734 1394 824">&lt;abc&gt;   Root Element Data&lt;data&gt;ABC Data &lt;/data&gt; now with extra info   &lt;/abc&gt;</pre> <p>GetTextTrim returns Root Element Data now with extra info.</p> <p>Suppose there is a PBDOM_ELEMENT defined as follows:</p> <pre data-bbox="804 1039 1394 1070">&lt;abc&gt;   Root Element Data   &lt;/abc&gt;</pre> <p>GetTextTrim returns Root Element Data.</p> <p>Suppose there is a PBDOM_ELEMENT defined as follows:</p> <pre data-bbox="804 1249 1394 1308">&lt;abc&gt;Root Element Data &lt;![CDATA[ with some cdata text]]&gt;&lt;/abc&gt;</pre> <p>GetTextTrim returns Root Element Data with some cdata text.</p>
PBDOM_ATTRIBUTE	<p>The trimmed text data contained within the PBDOM_ATTRIBUTE object with surrounding whitespace characters removed.</p> <p>Suppose there is an element with an attribute as follows:</p> <pre data-bbox="804 1653 1394 1711">&lt;abc ATTRIBUTE_1="My   Attribute "&gt;</pre> <p>GetTextTrim returns:</p> <pre data-bbox="804 1792 1394 1823">My   Attribute</pre> <p>Note, however, that the spaces between "My" and "Attribute" are still present.</p>
PBDOM_TEXT	<p>The trimmed text data contained within the PBDOM_TEXT object itself with surrounding whitespace characters removed.</p>

DOM Object Type	Return Value
	<p>For example, suppose there is the following element:</p> <pre data-bbox="804 344 1398 371">&lt;abc&gt; MY TEXT &lt;/abc&gt;</pre> <p>If there is a PBDOM_TEXT object to represent the text node "MY TEXT ", then calling GetTextTrim on the PBDOM_TEXT returns the string MY TEXT.</p>
PBDOM_CDATA	<p>The trimmed string data that is contained within the CDATA section itself with surrounding whitespace characters removed. For example, suppose there is the following CDATA:</p> <pre data-bbox="804 779 1398 864">&lt;![CDATA[  They're saying "x &lt; y" &amp; that "z &gt; y" so I guess  that means that z &gt; x  ]]&gt;</pre> <p>If there is a PBDOM_CDATA to represent the above CDATA section, then calling GetTextTrim on it returns the string:</p> <pre data-bbox="804 1025 1398 1111">They're saying " x &lt; y " &amp; that "z &gt; y" so I guess  that means that z &gt; x</pre> <p>Note that the initial spaces before "They're" and the trailing space after the last "x" have been removed.</p>
PBDOM_COMMENT	<p>The trimmed string data that is contained within the COMMENT itself. For example, suppose there is the following COMMENT:</p> <pre data-bbox="804 1402 1398 1429">&lt;!--  Comment  Here !  --&gt;</pre> <p>Note the spaces before the word "Comment" and after the exclamation mark "!".</p> <p>Calling GetTextTrim on the COMMENT returns the string Comment Here !</p>

**See also**[GetText](#)[GetTextNormalize](#)**15.1.14 HasChildren****Description**

Determines whether the PBDOM\_OBJECT has any child objects.

**Syntax**



```
pbdom_object_name.HasChildren()
```

**Table 15.20:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT

**Return value**

Boolean.

Returns true if the current PBDOM\_OBJECT has at least one child PBDOM\_OBJECT, and false if it has none.

**Throws**

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object is not associated with a derived PBDOM\_OBJECT class object.

**Examples**

In the following example, a PBDOM\_DOCUMENT is created from a simple XML string. The root element abc has a child text node that encapsulates the text "abc data". Calling HasChildren on the root element returns true. The message box displays Has Children. If the method returns false, the message box displays Has No Children

```
PBDOM_Builder pbdombuilder_new
pbdom_document pbdom_doc
pbdom_object pbdom_root_element
string strXML = "<abc>abc data</abc>"

pbdombuilder_new = Create PBDOM_Builder
pbdom_doc = pbdombuilder_new.BuildFromString (strXML)
pbdom_root_element = pbdom_doc.GetRootElement()
if (pbdom_root_element.HasChildren()) then
    MessageBox ("pbdom_root_element", "Has Children")
else
    MessageBox ("pbdom_root_element", "Has No Children")
end if
Destroy pbdombuilder_new
```

**Usage**

True is returned if the PBDOM\_OBJECT has at least one child, and false if there are no children.

**15.1.15 InsertContent****Description**

Inserts a new PBDOM\_OBJECT into the current PBDOM\_OBJECT.

**Syntax**

```
pbdom_object_name.InsertContent(pbdom_object_new, pbdom_object_ref)
```

**Table 15.21:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT

---

Argument	Description
<code>pbdom_object_new</code>	The referenced name of a PBDOM_OBJECT you want to insert
<code>pbdom_object_ref</code>	The name of the PBDOM_OBJECT in front of which you want to insert the new PBDOM_OBJECT

### Return value

PBDOM\_OBJECT. The return value is the newly modified PBDOM\_OBJECT.

### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object or the new PBDOM\_OBJECT or the reference PBDOM\_OBJECT is not associated with a derived PBDOM\_OBJECT class object.

EXCEPTION\_INVALID\_ARGUMENT -- One of the input arguments is invalid. This can happen if the input argument has not been initialized properly or is a null object reference.

### Usage

When a new PBDOM\_OBJECT is inserted into the current PBDOM\_OBJECT, the new PBDOM\_OBJECT becomes a child node of the current PBDOM\_OBJECT. Also, the new PBDOM\_OBJECT is to be positioned specifically before another PBDOM\_OBJECT, designated using the second parameter.

If the second PBDOM\_OBJECT is specified as null, then the new PBDOM\_OBJECT is to be inserted at the end of the list of children of the current PBDOM\_OBJECT.

---

### Derived Classes

Methods of classes that derive from the PBDOM\_OBJECT class return trivial results when the derived classes can have no child objects and when the methods concern manipulation of child-node content.

---

### See also

[AddContent](#)

[GetContent](#)

[RemoveContent](#)

[SetContent](#)

### 15.1.16 IsAncestorObjectOf

#### Description

Determines whether the current PBDOM\_OBJECT is the ancestor of another PBDOM\_OBJECT.

#### Syntax

```
pbdom_object_name.IsAncestorObjectOf(pbdom_object_ret)
```

**Table 15.22:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT
pbdom_object_ref	The PBDOM_OBJECT to check against

**Return value**

Boolean.

Returns true if the current PBDOM\_OBJECT is the ancestor of the referenced PBDOM\_OBJECT, and false otherwise.

**Throws**

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object is not associated with a derived PBDOM\_OBJECT class object.

EXCEPTION\_INVALID\_ARGUMENT -- The input PBDOM\_OBJECT is invalid. This can happen if it has not been initialized properly or it is a null object reference.

**Examples**

The following code fragment uses the IsAncestorObjectOf method and creates a structured document. In the fragment, pbdom\_elem\_1 represents the pbdom\_elem\_1 element. Because it is an ancestor of pbdom\_elem\_3, which represents the pbdom\_elem\_ element, the call to IsAncestorObjectOf returns true.

```
PBDOM_ELEMENT pbdom_elem_1
PBDOM_ELEMENT pbdom_elem_2
PBDOM_ELEMENT pbdom_elem_3
PBDOM_ELEMENT pbdom_elem_root
PBDOM_DOCUMENT pbdom_doc1

pbdom_doc1 = Create PBDOM_DOCUMENT
pbdom_elem_1 = Create PBDOM_ELEMENT
pbdom_elem_2 = Create PBDOM_ELEMENT
pbdom_elem_3 = Create PBDOM_ELEMENT

pbdom_elem_1.SetName("pbdom_elem_1")
pbdom_elem_2.SetName("pbdom_elem_2")
pbdom_elem_3.SetName("pbdom_elem_3")

pbdom_elem_1.AddContent(pbdom_elem_2)
pbdom_elem_2.AddContent(pbdom_elem_3)

pbdom_doc1.NewDocument("", "", &
    "Root_Element_From_Doc_1" , "", "")
pbdom_elem_root = pbdom_doc1.GetRootElement()
pbdom_elem_root.AddContent(pbdom_elem_1)

IF (pbdom_elem_1.IsAncestorObjectOf(pbdom_elem_3)) THEN
    MessageBox ("Ancestry", &
        "pbdom_elem_1 Is The Ancestor Of pbdom_elem_3")
ELSE
    MessageBox ("Ancestry", &
        "pbdom_elem_1 Is NOT The Ancestor Of pbdom_elem_3")
END IF

destroy pbdom_elem_1
```

```
destroy pbdom_elem_2
destroy pbdom_elem_3
destroy pbdom_elem_root
destroy pbdom_doc1
```

The preceding code fragment creates the following document:

```
<!DOCTYPE Root_Element_From_Doc_1>
<Root_Element_From_Doc_1>
  <pbdom_elem_1>
    <pbdom_elem_2>
      <pbdom_elem_3 />
    </pbdom_elem_2>
  </pbdom_elem_1>
</Root_Element_From_Doc_1>
```

## Usage

The `IsAncestorObjectOf` method determines whether the current `PBDOM_OBJECT` is the ancestor of another `PBDOM_OBJECT`.

### 15.1.17 RemoveContent

#### Description

Removes a child `PBDOM_OBJECT` from the current `PBDOM_OBJECT`.

#### Syntax

```
pbdom_object_name.RemoveContent(pbdom_object_ref)
```

**Table 15.23:**

Argument	Description
<code>pbdom_object_name</code>	The name of the <code>PBDOM_OBJECT</code>
<code>pbdom_object_ref</code>	The <code>PBDOM_OBJECT</code> to remove

#### Return value

Boolean.

Returns true if the content was removed, and false otherwise.

#### Throws

`EXCEPTION_PBDOM_OBJECT_INVALID_FOR_USE` -- This `PBDOM_OBJECT` object or the input `PBDOM_OBJECT` is not associated with a derived `PBDOM_OBJECT` class object.

`EXCEPTION_INVALID_ARGUMENT` -- The input `PBDOM_OBJECT` to be removed is invalid. This can happen if this object has not been initialized properly or is a null object reference.

#### Usage

When a new `PBDOM_OBJECT` is removed from the current one, all children under the removed `PBDOM_OBJECT` are also removed.

#### See also

[AddContent](#)

[GetContent](#)[InsertContent](#)[SetContent](#)

### 15.1.18 SetContent

#### Description

Sets the entire content of the PBDOM\_OBJECT.

#### Syntax

```
pbdom_object_name.SetContent(pbdom_object_array)
```

**Table 15.24:**

Argument	Description
pbdom_object_name	The name of the PBDOM object
pbdom_object_array	An array of PBDOM_OBJECT objects to be set as the contents of the PBDOM_OBJECT

#### Return value

PBDOM\_OBJECT. Returns the newly modified PBDOM\_OBJECT.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object is not associated with a derived PBDOM\_OBJECT class object.

#### Usage

The supplied array contains PBDOM\_OBJECT objects that are legal for the particular derived PBDOM\_OBJECT that is associated with this PBDOM\_OBJECT.

For example, a PBDOM\_DOCUMENT accepts only an array that contains PBDOM\_ELEMENT, PBDOM\_COMMENT, PBDOM\_DOCTYPE, or PBDOM\_PROCESSINGINSTRUCTION objects. In addition, the array can contain only one PBDOM\_ELEMENT object that it sets as its root element, and only one PBDOM\_DOCTYPE object that is set as its DOCTYPE.

If illegal objects are included in the array, exceptions (specific to the particular derived PBDOM\_OBJECT) are thrown. For more details, please refer to the SetContent method of the objects derived from PBDOM\_OBJECT.

In the event of an exception, the original contents of this PBDOM\_OBJECT are unchanged, and the PBDOM\_OBJECT objects contained in the supplied array are unaltered.

#### See also

[AddContent](#)[GetContent](#)[InsertContent](#)[RemoveContent](#)

### 15.1.19 SetName

#### Description

Sets the name of the PBDOM\_OBJECT.

#### Syntax

```
pbdom_object_name.SetName(stringstrName)
```

Table 15.25:

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT
strName	The new name you want to set for PBDOM_OBJECT

#### Return value

Boolean.

Returns true if the name of the PBDOM\_OBJECT was changed, and false otherwise.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object is not associated with a derived PBDOM\_OBJECT class object.

EXCEPTION\_INVALID\_ARGUMENT -- Input name string is invalid. This can happen if the string has been specifically set to null.

EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE -- Insufficient memory was encountered while executing this method.

EXCEPTION\_INVALID\_NAME -- The input name string does not conform to the W3C standards for XML names.

#### Usage

This name refers to the name of the particular derived PBDOM\_OBJECT to which this PBDOM\_OBJECT refers. Certain types of PBDOM\_OBJECT do not have any name associated with them. See the description of [GetName](#).

For example, PBDOM\_DOCUMENT does not have any name, so calling the SetName method returns false.

#### See also

[GetName](#)

### 15.1.20 SetParentObject

#### Description

Sets the referenced PBDOM\_OBJECT as the parent of the current PBDOM\_OBJECT.

#### Syntax

```
pbdom_object_name.SetParentObject(pbdom_object pbdom_object_ref)
```

**Table 15.26:**

Argument	Description
pbdom_object_name	The name of the PBDOM_OBJECT
pbdom_object_ref	The PBDOM_OBJECT to be set as the parent of the current PBDOM_OBJECT

**Return value**

PBDOM\_OBJECT. The current PBDOM\_OBJECT is appended as a child node of the referenced parent.

**Throws**

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- This PBDOM\_OBJECT object or the input PBDOM\_OBJECT is not associated with a derived PBDOM\_OBJECT class object.

EXCEPTION\_INVALID\_ARGUMENT -- The input PBDOM\_OBJECT is invalid. This can happen if it has not been initialized properly, or if it is a null object reference.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT -- The current PBDOM\_OBJECT already has a parent.

EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT -- If the input PBDOM\_OBJECT is of a class that cannot have a legal parent-child relationship with this PBDOM\_OBJECT.

**Examples**

In the following code example, a PBDOM\_ELEMENT object is created and called pbdom\_elem\_1. Its parent is set to be the root element of the PBDOM\_DOCUMENT called pbdom\_doc. Once this is done, pbdom\_elem\_1 is immediately transferred to the pbdom\_doc document and pbdom\_elem\_1 is immediately appended as a child node of the root element of pbdom\_doc.

The following method call returns the string "pbdom\_element", because the root element is a PBDOM\_ELEMENT:

```
pbdom_elem_1.GetParentObject().GetObjectClassString()
```

The following method call returns the string "Root\_Element", which is the name of the root element:

```
pbdom_elem_1.GetParentObject().GetName()
```

Here is the complete example:

```
PBDOM_ELEMENT pbdom_elem_1
PBDOM_ELEMENT pbdom_elem_root
PBDOM_DOCUMENT pbdom_doc1

pbdom_doc1 = Create PBDOM_DOCUMENT
pbdom_elem_1 = Create PBDOM_ELEMENT
pbdom_elem_1.SetName ("pbdom_elem_1")

pbdom_doc1.NewDocument ("", "", "Root_Element", "", "")
pbdom_elem_root = pbdom_doc1.GetRootElement()
pbdom_elem_1.SetParentObject(pbdom_elem_root)
```

```
MessageBox ("Parent Class", &
  pbdom_elem_1.GetParentObject(). &
  GetObjectClassString())
MessageBox ("Parent Name", &
  pbdom_elem_1.GetParentObject().GetName())

destroy pbdom_elem_1
destroy pbdom_elem_root
destroy pbdom_doc1
```

### Usage

The caller is responsible for ensuring that the current PBDOM\_OBJECT and the referenced PBDOM\_OBJECT can have a legal parent-child relationship. The caller is also responsible for making sure pre-existing parentage is legal.

The PBDOM SetParentObject method differs from the JDOM SetParent method in that JDOM defines a setParent method for several specific classes, including Element, Comment, and CDATA. PBDOM implements the SetParentObject method in the base PBDOM\_OBJECT class to allow polymorphism.

See the SetParentObject documentation of derived PBDOM\_OBJECT classes for more details on implementation of specific classes.

### See also

[GetOwnerDocumentObject](#)

[GetParentObject](#)



## 16 PBDOM\_PROCESSINGINSTRUCTION Class

### About this chapter

This chapter describes the PBDOM\_PROCESSINGINSTRUCTION class.

### 16.1 PBDOM\_PROCESSINGINSTRUCTION

#### Description

The PBDOM\_PROCESSINGINSTRUCTION class defines behavior for an XML processing instruction. Methods allow you to obtain the target of the processing instruction object as well as its data. You can always access the data as a string, and, where appropriate, as name/value pairs.

Note that the actual processing instruction of a processing instruction object is a string, even if the instruction is divided into separate name="value" pairs. PBDOM does support such a processing instruction object format. If the processing instruction object data does contain pairs, as is commonly the case, then PBDOM\_PROCESSINGINSTRUCTION parses them into an internal list of name/value pairs.

#### Methods

Some of the inherited methods from PBDOM\_OBJECT serve no meaningful objective, and only default or trivial functionalities result. These are described in the following table:

**Table 16.1:**

Method	Always returns
AddContent	Current PBDOM_PROCESSINGINSTRUCTION. Use AddValue instead.
GetContent	false. Use GetName and GetValue instead.
HasChildren	false.
InsertContent	Current PBDOM_PROCESSINGINSTRUCTION.
IsAncestorObjectOf	false.
RemoveContent	false. Use RemoveValue instead.
SetContent	Current PBDOM_PROCESSINGINSTRUCTION. Use SetData instead.

PBDOM\_PROCESSINGINSTRUCTION has the following methods:

**Table 16.2:**

<a href="#">Clone</a>	<a href="#">GetTarget</a>
<a href="#">Detach</a>	<a href="#">GetText</a>
<a href="#">Equals</a>	<a href="#">GetTextNormalize</a>

<a href="#">GetData</a>	<a href="#">GetTextTrim</a>
<a href="#">GetName</a>	<a href="#">GetValue</a>
<a href="#">GetNames</a>	<a href="#">RemoveValue</a>
<a href="#">GetObjectClass</a>	<a href="#">SetData</a>
<a href="#">GetObjectClassString</a>	<a href="#">SetName</a>
<a href="#">GetOwnerDocumentObject</a>	<a href="#">SetParentObject</a>
<a href="#">GetParentObject</a>	<a href="#">SetValue</a>

### 16.1.1 Clone

#### Description

Creates and returns a clone of the current PBDOM\_PROCESSINGINSTRUCTION object.

#### Syntax

```
pbdom_pi_name.Clone(boolean bDeep)
```

**Table 16.3:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object.
bDeep	A boolean specifying whether a deep or shallow clone is returned. Values are true for a deep clone and false for a shallow clone. This argument is currently ignored.

#### Return value

PBDOM\_OBJECT. A clone of the current PBDOM\_PROCESSINGINSTRUCTION object returned as a PBDOM\_OBJECT.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the internal implementation of this PBDOM\_PROCESSINGINSTRUCTION object is null. The occurrence of this exception is rare, but it can take place if severe memory corruption occurs.

#### Usage

The Clone method creates a new PBDOM\_PROCESSINGINSTRUCTION object that is a duplicate of, and a separate object from, the original. The clone of a PBDOM\_PROCESSINGINSTRUCTION object is always identical to its original whether bDeep is true or false, because a PBDOM\_PROCESSINGINSTRUCTION object contains no subtree of child PBDOM\_OBJECTs.

A PBDOM\_PROCESSINGINSTRUCTION clone has no parent, but it resides in the same PBDOM\_DOCUMENT as its original, and if the original PBDOM\_PROCESSINGINSTRUCTION object is standalone, so is the clone.

### 16.1.2 Detach

#### Description

Detaches a PBDOM\_PROCESSINGINSTRUCTION object from its parent PBDOM\_OBJECT.

#### Syntax

```
pbdom_pi_name.Detach()
```

**Table 16.4:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object

#### Return value

PBDOM\_OBJECT. This PBDOM\_PROCESSINGINSTRUCTION object detached from its parent object. This method does nothing if this PBDOM\_PROCESSINGINSTRUCTION object has no parent.

### 16.1.3 Equals

#### Description

Tests for the equality of the current PBDOM\_PROCESSINGINSTRUCTION object with the supplied PBDOM\_OBJECT.

#### Syntax

```
pbdom_pi_name.Equals(pbdom_object_ref)
```

**Table 16.5:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object
pbdom_object_ref	A PBDOM_OBJECT for testing for equality with the current PBDOM_PROCESSINGINSTRUCTION object

#### Return value

Boolean.

Returns true if the current PBDOM\_PROCESSINGINSTRUCTION object is equivalent to the input PBDOM\_OBJECT, and false otherwise.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the input PBDOM\_OBJECT is not a reference to an object derived from PBDOM\_OBJECT.

### 16.1.4 GetData

#### Description

Returns the raw data of the PBDOM\_PROCESSINGINSTRUCTION object.

#### Syntax

```
pbdom_pi_name.GetData()
```

**Table 16.6:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object

#### Return value

String.

The data of the PBDOM\_PROCESSINGINSTRUCTION object.

#### Usage

The processing instruction data is fundamentally a string and not a set of name="value" pairs.

### 16.1.5 GetName

#### Description

Obtains the name of the current PBDOM\_PROCESSINGINSTRUCTION object.

#### Syntax

```
pbdom_pi_name.GetName()
```

**Table 16.7:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object

#### Return value

String.

#### Examples

Calling the GetName method on the following processing instruction returns works:

```
<?works document="hello.doc" data="hello.wks" ?>
```

#### Usage

This method is similar to the GetTarget method. To PBDOM, the processing instruction target is synonymous with its name.

### 16.1.6 GetNames

#### Description

Retrieves a list of names taken from the part of the PBDOM\_PROCESSINGINSTRUCTION object's data that is factored into name="value" pairs. This method can be used in conjunction with the GetValue method.

#### Syntax

```
pbdom_pi_name.GetNames(string name_array[ ])
```

**Table 16.8:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object
name_array	An unbounded string array filled with names

#### Return value

Boolean.

Returns true if a list of names is retrieved, and false otherwise. If there are no name/value pairs, this method returns false.

#### Examples

Given the following PBDOM\_PROCESSINGINSTRUCTION object, GetNames returns three strings, a, b, and c, even though a occurs more than once:

```
<? dw-set_values a="1" b="2" c="3" a="4" ?>
```

When the GetValue method is called on a, the value 4 is returned, because it is the last value set for a.

#### Usage

If a name is used more than once as the name of a name/value pair in a PBDOM\_PROCESSINGINSTRUCTION object, then the value set in the last occurrence of the name is used, and values declared in all previous occurrences of the name are discarded.

### 16.1.7 GetObjectClass

#### Description

Returns a long integer code that indicates the class of the current PBDOM\_PROCESSINGINSTRUCTION object.

#### Syntax

```
pbdom_pi_name.GetObjectClass()
```

**Table 16.9:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_OBJECT

**Return value**

Long.

GetObjectClass returns a long integer code that indicates the class of the current PBDOM\_OBJECT. If pbdom\_pi\_name is a PBDOM\_PROCESSINGINSTRUCTION object, the returned value is 10.

**16.1.8 GetObjectClassString****Description**

Returns a string form of the class of the PBDOM\_PROCESSINGINSTRUCTION object.

**Syntax**

```
pbdom_pi_name.GetObjectClassString()
```

**Table 16.10:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_OBJECT

**Return value**

String.

GetObjectClassString returns a string that indicates the class of the current PBDOM\_OBJECT. If pbdom\_pi\_name is a PBDOM\_PROCESSINGINSTRUCTION, the returned string is "pbdom\_processinginstruction".

**16.1.9 GetOwnerDocumentObject****Description**

Returns the owning PBDOM\_DOCUMENT of the current PBDOM\_PROCESSINGINSTRUCTION object.

**Syntax**

```
pbdom_pi_name.GetOwnerDocumentObject()
```

**Table 16.11:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object

**Return value**

PBDOM\_DOCUMENT. If there is no owning PBDOM\_DOCUMENT, null is returned.

**16.1.10 GetParentObject****Description**

Returns the parent PBDOM\_OBJECT of the current PBDOM\_PROCESSINGINSTRUCTION object.

**Syntax**

```
pbdom_pi_name.GetParentObject()
```

**Table 16.12:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object

**Return value**

PBDOM\_OBJECT. The parent of the PBDOM\_PROCESSINGINSTRUCTION object. If there is no parent, null is returned.

**16.1.11 GetTarget****Description**

Returns the target of the PBDOM\_PROCESSINGINSTRUCTION object.

**Syntax**

```
pbdom_pi_name.GetTarget()
```

**Table 16.13:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object

**Return value**

String.

The target of the PBDOM\_PROCESSINGINSTRUCTION object.

**Examples**

Given the following PBDOM\_PROCESSINGINSTRUCTION object, calling the GetTarget method returns the string "xml-stylesheet":

```
<?xml-stylesheet href="simple-ie5.xsl" type="text/xsl" ?>
```

Calling the GetName method returns the same string.

**See also**

[GetName](#)

**16.1.12 GetText****Description**

Obtains text data that is contained within the current PBDOM\_PROCESSINGINSTRUCTION object.

**Syntax**

```
pbdom_pi_name.GetText()
```

**Table 16.14:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object

**Return value**

String.

**Usage**

The GetText method returns the text data of the current PBDOM\_PROCESSINGINSTRUCTION object. GetText is similar to GetData. However, the textual content of a processing instruction object is not a text node.

**See also**

[GetData](#)

[GetTextNormalize](#)

[GetTextTrim](#)

[SetData](#)

**16.1.13 GetTextNormalize****Description**

Obtains the text data that is contained within the current PBDOM\_PROCESSINGINSTRUCTION object with all surrounding whitespace characters removed and internal whitespace characters normalized to a single space.

**Syntax**

```
pbdom_pi_name.GetTextNormalize()
```

**Table 16.15:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object

**Return value**

String.

The normalized text content of the PBDOM\_PROCESSINGINSTRUCTION object. If no textual value exists for the current PBDOM\_OBJECT, or if only whitespace characters exist, an empty string is returned.

**See also**

[GetData](#)

[GetText](#)



[GetTextTrim](#)[SetData](#)

### 16.1.14 GetTextTrim

#### Description

Obtains the text data that is contained within the current PBDOM\_PROCESSINGINSTRUCTION object with all surrounding whitespaces removed.

#### Syntax

```
pbdom_pi_name.GetTextTrim()
```

**Table 16.16:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object

#### Return value

String.

The trimmed text content of the PBDOM\_PROCESSINGINSTRUCTION object. If no textual value exists for the current PBDOM\_PROCESSINGINSTRUCTION object, or if only whitespace characters exist, an empty string is returned.

#### See also

[GetData](#)[GetText](#)[GetTextNormalize](#)[SetData](#)

### 16.1.15 GetValue

#### Description

Returns the value for a specific name/value pair on the PBDOM\_PROCESSINGINSTRUCTION object. If no such pair is found for the PBDOM\_PROCESSINGINSTRUCTION object, an empty string is returned.

#### Syntax

```
pbdom_pi_name.GetValue(string strName)
```

**Table 16.17:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object
strName	String name of name/value pair

**Return value**

String.

String name of the name/value pair to search for value.

**Examples**

Given the following PBDOM\_PROCESSINGINSTRUCTION object, GetValue("href") returns the string "simple-ie5.xsl":

```
<?xml-stylesheet href="simple-ie5.xsl" type="text/xsl" ?>
```

**See also**

[GetData](#)

[GetText](#), [SetValue](#)

**16.1.16 RemoveValue****Description**

Removes the specified name/value pair.

**Syntax**

```
pbdom_pi_name.RemoveValue(string strName)
```

**Table 16.18:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object
strName	String name of name/value pair to be removed

**Return value**

Boolean.

Returns true if the requested name/value pair is removed and false otherwise.

**Examples**

Suppose the following PBDOM\_PROCESSINGINSTRUCTION object is given:

```
<?xml-stylesheet href="simple-ie5.xsl" type="text/xsl" ?>
```

Then, RemoveValue("href") results in the PBDOM\_PROCESSINGINSTRUCTION object being transformed into the following:

```
<?xml-stylesheet type="text/xsl" ?>
```

**16.1.17 SetData****Description**

Sets the raw data for the PBDOM\_PROCESSINGINSTRUCTION object.

## Syntax

```
pbdom_pi_name.SetData(string strData)
```

**Table 16.19:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object
strData	New data for the PBDOM_PROCESSINGINSTRUCTION object

## Return value

PBDOM\_PROCESSINGINSTRUCTION. The PBDOM\_PROCESSINGINSTRUCTION object modified with the new data.

## Throws

EXCEPTION\_INVALID\_STRING -- The input data is invalid. This can happen in the following circumstances:

1. The input data contains the sub-string "?>". This violates the requirements for the data of a processing instruction.
2. If the processing instruction target name is xml, making this PBDOM\_PROCESSINGINSTRUCTION object an XML declaration processing instruction, this exception is thrown if the input data string does not conform to the following criteria:
  - The data must contain a name/value pair for the name version.
  - The data can contain a name/value pair for the name encoding.
  - The data can contain a name/value pair for the name standalone. If it does, the value for standalone must either be yes or no.
  - The data must not contain any other data in the form of name/value pairs or in any other form.

---

## Lowercase

The strings xml, version, encoding, standalone, yes, and no are all case sensitive and must be in lowercase.

---

## Examples

Suppose there is a PBDOM\_PROCESSINGINSTRUCTION object as follows:

```
<?xml-stylesheet href="simple-ie5.xsl" type="text/xsl" ?>
```

Then, SetData("href=new.xsl") results in the PBDOM\_PROCESSINGINSTRUCTION object being transformed into the following:

```
<?xml-stylesheet href=new.xsl" ?>
```

The entire data for the PBDOM\_PROCESSINGINSTRUCTION object is now reset.

## Usage

Special processing is performed when the name of the processing instruction's target is xml, which indicates that it is an XML declaration. The valid instructions allowed in the input Data as part of the name in the name/value pairs are version, encoding, and standalone. The version instruction is mandatory before the processing instruction can be added to a document.

The XML specification expects the instructions to be in the specific order version , encoding , standalone. This function reorders the input data to conform to the specification, for example:

```
<? xml version="1.0" encoding="utf-8" standalone="yes"?>
```

### 16.1.18 SetName

#### Description

Sets the name of the current PBDOM\_PROCESSINGINSTRUCTION object.

#### Syntax

```
pbdom_pi_name.SetName(string strName)
```

**Table 16.20:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object
strName	The new name you want to set for the current PBDOM_PROCESSINGINSTRUCTION object

#### Return value

Boolean.

Returns true if the name of the current PBDOM\_PROCESSINGINSTRUCTION object was changed, and false otherwise.

#### Throws

EXCEPTION\_INVALID\_NAME -- This exception is thrown if the name is invalid. The name can be xml, making this PBDOM\_PROCESSINGINSTRUCTION object an XML declaration processing instruction. However, in this case, the name xml must be in lowercase, or the EXCEPTION\_INVALID\_NAME exception will be thrown.

EXCEPTION\_INVALID\_STRING -- This exception is thrown if the name is xml and the current data of this PBDOM\_PROCESSINGINSTRUCTION object is not valid. The data is valid only under the following circumstances:

- It is an empty string.

- If it is not an empty string, it must contain a name/value pair for the name version.
- If it is not an empty string and it contains a name/value pair for the name version, it can also contain a name/value pair for the name encoding.
- If it is not an empty string and it contains a name/value pair for the name version, it can also contain a name/value pair for the name standalone. If it does, the value for standalone must be either yes or no (both are case sensitive).
- If it is not an empty string and it contains a name/value pair for the name version, it must not contain any other data (in name/value pair format or otherwise) except for encoding and standalone.

### Usage

This method is equivalent to setting the target of the processing instruction object. See the list of exceptions for information about the restrictions on the use of xml as the target.

### 16.1.19 SetParentObject

#### Description

Sets the referenced PBDOM\_OBJECT to be the parent of the current PBDOM\_PROCESSINGINSTRUCTION object.

#### Syntax

```
pbdom_pi_name.SetParentObject(pbdom_object pbdom_object_ref)
```

**Table 16.21:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object
pbdom_object_ref	A PBDOM_OBJECT to be set as the parent of the current PBDOM_PROCESSINGINSTRUCTION object

#### Return value

PBDOM\_OBJECT. This PBDOM\_PROCESSINGINSTRUCTION object modified.

#### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the input PBDOM\_OBJECT is not a reference to an object derived from PBDOM\_OBJECT.

EXCEPTION\_HIERARCHY\_ERROR -- If setting the input PBDOM\_OBJECT to be the parent of this PBDOM\_PROCESSINGINSTRUCTION object will cause the parent PBDOM\_OBJECT to be no longer well formed. For example, if this PBDOM\_PROCESSINGINSTRUCTION object is an XML declaration and the parent to be set is a PBDOM\_ELEMENT.

## Usage

The PBDOM\_OBJECT that you set as the parent and the current PBDOM\_PROCESSINGINSTRUCTION object must have a legal parent-child relationship. Currently, only a PBDOM\_ELEMENT and a PBDOM\_DOCUMENT can be set as the parent of a PBDOM\_PROCESSINGINSTRUCTION object.

### 16.1.20 SetValue

#### Description

Sets the value for the specified name/value pair.

#### Syntax

```
pbdom_pi_name.SetValue(string strName, stringstrValue)
```

**Table 16.22:**

Argument	Description
pbdom_pi_name	The name of a PBDOM_PROCESSINGINSTRUCTION object
strName	String name of a name/value pair
strValue	String value of a name/value pair

#### Return value

PBDOM\_PROCESSINGINSTRUCTION.

#### Throws

EXCEPTION\_INVALID\_STRING -- The input strName/strValue is invalid. This can happen in the following circumstances:

- The input strName/strValue data contains the sub-string ?>. This violates the requirements for the data of a processing instruction.
- If the target name is xml, making this PBDOM\_PROCESSINGINSTRUCTION object an XML declaration processing instruction, this exception is thrown if the input data string does not conform to the following criterion: the data can contain a name/value pair for the name standalone. If it does, the value for standalone must either be yes or no. The strings xml, standalone, yes, and no are case sensitive and must be lowercase.

EXCEPTION\_INVALID\_NAME -- The input strName is invalid. This can happen if the target name is xml, making this PBDOM\_PROCESSINGINSTRUCTION object an XML declaration processing instruction, and either of the following is true:

- The strName value is other than version, standalone or encoding.
- Either standalone or encoding is set without the version first being set.

#### Examples

Consider the following PBDOM\_PROCESSINGINSTRUCTION object:

```
<?xml-stylesheet href="simple-ie5.xml" type="text/xsl" ?>
```

SetValue("href","new.xml")

transforms this processing instruction to the following, modifying the value for href:

```
<?xml-stylesheet href="new.xml" type="text/xsl"?>
```

SetValue("extra\_info","xalan")

transforms the processing instruction to the following, adding a new name/value pair for extra\_info:

```
<?xml-stylesheet href=new.xml" type="text/xsl" extra_info "xalan" ?>
```

Then SetValue("extra\_info\_2","") transforms the processing instruction to the following, adding a new name/value pair for extra\_info\_2 with an empty string as the value:

```
<?xml-stylesheet href=new.xml" type="text/xsl" extra_info="xalan" extra_info_2="" ?>
```

### Usage

If no value is found, the supplied pair is added to the processing instruction data.

The appearance of name/value pairs in a PBDOM\_PROCESSINGINSTRUCTION object is not subject to any order. In this way, name/value pairs in a PBDOM\_PROCESSINGINSTRUCTION object are similar to attributes in an element. Attributes are specifically not ordered.

Special processing is performed when the name of the processing instruction's target is xml, which indicates that it is an XML declaration. The valid instructions allowed in the input Data as part of the name in the name/value pairs are version, encoding, and standalone. The version instruction is mandatory before the processing instruction can be added to a document.

The XML specification expects the instructions to be in this specific order: version , encoding , standalone. This function reorders the input data to conform to the specification, for example:

```
<? xml version="1.0" encoding="utf-8" standalone="yes"?>
```

## 17 PBDOM\_TEXT Class

### About this chapter

This chapter describes the PBDOM\_TEXT class.

### 17.1 PBDOM\_TEXT

#### Description

The PBDOM\_TEXT class represents a DOM Text Node within an XML document. It extends the PBDOM\_CHARACTERDATA class with a set of methods specifically intended for manipulating DOM text nodes.

The PBDOM\_TEXT class is derived from the PBDOM\_CHARACTERDATA class. PBDOM\_TEXT objects are commonly used to represent the textual content of a PBDOM\_ELEMENT or PBDOM\_ATTRIBUTE.

---

#### Whitespace characters

The text in a PBDOM\_TEXT object can include whitespace characters such as carriage returns, linefeeds, tabs, and spacebar spaces.

---

#### Methods

Some of the inherited methods from PBDOM\_OBJECT serve no meaningful objective, and only default or trivial functionalities result. These are described in the following table:

**Table 17.1:**

Method	Always returns
AddContent	current PBDOM_TEXT
GetContent	false
GetName	a string "#text"
HasChildren	false
InsertContent	current PBDOM_TEXT
IsAncestorObjectOf	false
RemoveContent	false
SetContent	current PBDOM_TEXT
SetName	false

PBDOM\_TEXT has the following non-trivial methods:

**Table 17.2:**

<a href="#">Append</a>	<a href="#">GetParentObject</a>
<a href="#">Clone</a>	<a href="#">GetText</a>
<a href="#">Detach</a>	<a href="#">GetTextNormalize</a>
<a href="#">Equals</a>	<a href="#">GetTextTrim</a>



<a href="#">GetObjectClass</a>	<a href="#">SetParentObject</a>
<a href="#">GetObjectClassString</a>	<a href="#">SetText</a>
<a href="#">GetOwnerDocumentObject</a>	

### 17.1.1 Append

#### Description

The Append method is overloaded:

- Syntax 1 appends an input string to the text content that already exists within the current PBDOM\_TEXT object.
- Syntax 2 appends the text data of a PBDOM\_CHARACTERDATA object to the text content that already exists within the current PBDOM\_TEXT object.

#### Syntax

**Table 17.3:**

For this syntax	See
<code>Append(string strAppend)</code>	<a href="#">Append Syntax 1</a>
<code>Append(pbdom_characterdata pbdom_characterdata_ref)</code>	<a href="#">Append Syntax 2</a>

#### 17.1.1.1 Append Syntax 1

##### Description

Appends an input string to the text content that already exists within the current PBDOM\_TEXT object.

##### Syntax

```
pbdom_text_name.Append(string strAppend)
```

**Table 17.4:**

Argument	Description
<code>pbdom_text_name</code>	The name of a PBDOM_TEXT object
<code>strAppend</code>	The string you want appended to the existing text of the current PBDOM_TEXT object

##### Return value

PBDOM\_CHARACTERDATA. The current PBDOM\_TEXT object modified and returned as a PBDOM\_CHARACTERDATA object.

#### 17.1.1.2 Append Syntax 2

##### Description

Appends the text data of a PBDOM\_CHARACTERDATA object to the text content that already exists within the current PBDOM\_TEXT object.

## Syntax

```
pbdom_text_name.Append(pbdom_characterdata pbdom_characterdata_ref)
```

**Table 17.5:**

Argument	Description
pbdom_text_name	The name of a PBDOM_TEXT object
pbdom_characterdata_ref	The referenced PBDOM_CHARACTERDATA object whose text data is to be appended to the existing text of the current PBDOM_TEXT object

## Return value

PBDOM\_CHARACTERDATA. The current PBDOM\_TEXT object modified and returned as a PBDOM\_CHARACTERDATA object.

## Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the input PBDOM\_CHARACTERDATA is not a reference to an object inherited from PBDOM\_CHARACTERDATA.

## Usage

Note that JDOM does not define an Append method for its TEXT class. Because PBDOM implements its Append method in the base PBDOM\_CHARACTERDATA class, a PBDOM\_COMMENT object, a PBDOM\_CDATA object, and a PBDOM\_TEXT object can append their internal text data to each other, because they are all objects inherited from PBDOM\_CHARACTERDATA.

## 17.1.2 Clone

### Description

Creates and returns a clone of the current PBDOM\_TEXT object.

### Syntax

```
pbdom_text_name.Clone(boolean bDeep)
```

**Table 17.6:**

Argument	Description
pbdom_text_name	The name of a PBDOM_TEXT object.
bDeep	A boolean specifying whether a deep or shallow clone is returned. Values are true for a deep clone and false for a shallow clone. This parameter is ignored.

## Return value

PBDOM\_OBJECT. The return value is a clone of the current PBDOM\_TEXT object returned as a PBDOM\_OBJECT.

## Examples

This example creates an XML document that, when serialized, appears as follows:

```
<!DOCTYPE root
[
<!ELEMENT root (child_1, child_2)>
<!ELEMENT child_1 (#PCDATA)*>
<!ELEMENT child_2 (#PCDATA)*>
]>
<root>
  <child_1>text for child.</child_1>
  <child_2>text for child.</child_2>
</root>
```

The definition of the DTD shows that the document is required to have the following composition:

- The document contains a root element with the name root.
- The root element contains a sequence of two child elements named child\_1 and child\_2.
- Both child\_1 and child\_2 contain only text.

The following PowerShell code creates a PBDOM\_TEXT object and assigns it a text value. It then creates a child\_1 element, adds the PBDOM\_TEXT object to it, creates a shallow clone of child\_1, and names the clone child\_2. After adding a clone of the text object to child\_2, the code adds both child objects to the root element:

```
PBDOM_BUILDER      pbdom_buildr
PBDOM_DOCUMENT     pbdom_doc
PBDOM_ELEMENT      pbdom_elem_child_1
PBDOM_ELEMENT      pbdom_elem_child_2
PBDOM_TEXT         pbdom_txt
string strXML = "<!DOCTYPE root [<!ELEMENT root (child_1, child_2)><!ELEMENT
child_1 (#PCDATA)><!ELEMENT child_2 (#PCDATA)>]><root/>"

try
    pbdom_buildr = Create PBDOM_BUILDER
    pbdom_doc = pbdom_buildr.BuildFromString (strXML)

    pbdom_txt = Create PBDOM_TEXT
    pbdom_txt.SetText ("text for child.")

    pbdom_elem_child_1 = Create PBDOM_ELEMENT
    pbdom_elem_child_1.SetName ("child_1")
    pbdom_elem_child_1.AddContent (pbdom_txt)

    pbdom_elem_child_2 = pbdom_elem_child_1.Clone(false)
    pbdom_elem_child_2.SetName("child_2")
    pbdom_elem_child_2.AddContent (pbdom_txt.Clone(false))

    pbdom_doc.GetRootElement().AddContent(pbdom_elem_child_1)
    pbdom_doc.GetRootElement().AddContent(pbdom_elem_child_2)

    pbdom_doc.SaveDocument ("sample.xml")

catch (PBDOM_EXCEPTION pbdom_except)
```

```

    MessageBox ("PBDOM_EXCEPTION", pbdom_except.GetMessage())
end try

```

## Usage

The Clone method creates a new PBDOM\_TEXT object that is a duplicate of, and a separate object from, the original. Whether true or false is supplied as the parameter to this function, a PBDOM\_TEXT clone is always identical to its original. This is because a PBDOM\_TEXT does not contain any subtree of children PBDOM\_OBJECTs.

A PBDOM\_TEXT clone has no parent. However, the clone resides in the same PBDOM\_DOCUMENT as its original, and if the original PBDOM\_TEXT object is standalone, the clone is standalone

### 17.1.3 Detach

#### Description

Detaches a PBDOM\_TEXT object from its parent PBDOM\_OBJECT.

#### Syntax

```
pbdom_text_name.Detach()
```

**Table 17.7:**

Argument	Description
pbdom_text_name	The name of a PBDOM_TEXT object

#### Return value

PBDOM\_OBJECT. The current PBDOM\_TEXT object is detached from its parent.

#### Usage

If the current PBDOM\_TEXT object has no parent, nothing happens.

### 17.1.4 Equals

#### Description

Tests for the equality of the current PBDOM\_TEXT object and a referenced PBDOM\_OBJECT.

#### Syntax

```
pbdom_text_name.Equals(pbdom_object pbdom_object_ref)
```

**Table 17.8:**

Argument	Description
pbdom_text_name	The name of a PBDOM_TEXT object
pbdom_object_ref	A reference to a PBDOM_OBJECT to test for equality with the current PBDOM_TEXT object

#### Return value

Boolean.

Returns true if the current PBDOM\_TEXT object is equivalent to the input PBDOM\_OBJECT, and false otherwise.

### Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the input PBDOM\_OBJECT is not a reference to an object derived from PBDOM\_OBJECT.

### Usage

True is returned only if the referenced PBDOM\_OBJECT is also a derived PBDOM\_TEXT object and refers to the same DOM object as the current PBDOM\_TEXT object. Two separately created PBDOM\_TEXT objects, for example, can contain exactly the same text but not be equal.

## 17.1.5 GetObjectClass

### Description

Returns a long integer code that indicates the class of the current PBDOM\_OBJECT.

### Syntax

```
pbdom_object_name.GetObjectClass()
```

**Table 17.9:**

Argument	Description
pbdom_object_name	The name of a PBDOM_OBJECT

### Return value

Long.

GetObjectClass returns a long integer code that indicates the class of the current PBDOM\_OBJECT. If pbdom\_object\_name is a PBDOM\_TEXT object, the returned value is 7.

### See also

[GetObjectClassString](#)

## 17.1.6 GetObjectClassString

### Description

Returns a string form of the class of the PBDOM\_OBJECT.

### Syntax

```
pbdom_object_name.GetObjectClassString()
```

**Table 17.10:**

Argument	Description
pbdom_object_name	The name of a PBDOM_OBJECT

### Return value

String.

GetObjectClassString returns a string that indicates the class of the current PBDOM\_OBJECT. If pbdom\_object\_name is a PBDOM\_TEXT object, the returned string is "pbdom\_text".

**See also**

[GetObjectClass](#)

### 17.1.7 GetOwnerDocumentObject

#### Description

Returns the owning PBDOM\_DOCUMENT of the current PBDOM\_TEXT object.

#### Syntax

```
pbdom_text_name.GetOwnerDocumentObject()
```

**Table 17.11:**

Argument	Description
pbdom_text_name	The name of a PBDOM_TEXT object

#### Return value

PBDOM\_OBJECT.

#### Usage

If there is no owning PBDOM\_DOCUMENT, null is returned.

### 17.1.8 GetParentObject

#### Description

Returns the parent PBDOM\_OBJECT of the current PBDOM\_TEXT object.

#### Syntax

```
pbdom_text_name.GetParentObject()
```

**Table 17.12:**

Argument	Description
pbdom_text_name	The name of a PBDOM_TEXT object

#### Return value

PBDOM\_OBJECT.

#### Usage

The parent is also an object inherited from PBDOM\_TEXT object. If the PBDOM\_TEXT object has no parent, null is returned.

**See also**

[SetParentObject](#)

### 17.1.9 GetText

#### Description

Obtains the text data that is contained within the current PBDOM\_TEXT object.

#### Syntax

```
pbdom_text_name.GetText()
```

Table 17.13:

Argument	Description
pbdom_text_name	The name of a PBDOM_TEXT object

#### Return value

String.

The GetText method returns the textual content of the current PBDOM\_TEXT object.

#### Examples

If you have the element `<abc>MY TEXT</abc>`, and you have a PBDOM\_TEXT object to represent the text node "MY TEXT", then calling GetText on the PBDOM\_TEXT object returns the string "MY TEXT".

#### See also

[GetTextNormalize](#)

[GetTextTrim](#)

[SetText](#)

### 17.1.10 GetTextNormalize

#### Description

Obtains the text data that is contained within the current PBDOM\_TEXT object, with all surrounding whitespace characters removed and internal whitespace characters normalized to a single space.

#### Syntax

```
pbdom_text_name.GetTextNormalize()
```

Table 17.14:

Argument	Description
pbdom_text_name	The name of a PBDOM_TEXT object

#### Return value

String.

#### Examples

If you have a PBDOM\_TEXT object that represents the text node " MY TEXT ", calling GetTextNormalize returns the string "MY TEXT". All surrounding whitespaces are removed, and the whitespaces between the words "MY" and "TEXT" are reduced to a single space.

## Usage

This method allows the caller to obtain the text data that is contained within the current PBDOM\_TEXT object with all surrounding whitespaces removed and internal whitespaces normalized to single spaces. If no textual value exists for the current PBDOM\_TEXT object, or if only whitespaces exist, an empty string is returned.

### See also

[GetText](#)

[GetTextTrim](#), [SetText](#)

## 17.1.11 GetTextTrim

### Description

Returns the textual content of the current PBDOM\_TEXT object with all surrounding whitespace characters removed.

### Syntax

```
pbdom_text_name.GetTextTrim()
```

**Table 17.15:**

Argument	Description
pbdom_text_name	The name of a PBDOM_TEXT object

### Return value

String.

### Examples

If you have a PBDOM\_TEXT object that represents the text node " MY TEXT ", calling GetTextNormalize returns the string "MY TEXT". All surrounding white spaces are removed. The whitespaces between the words "MY" and "TEXT" are preserved.

### Usage

This method allows the caller to obtain the text data that is contained within the current PBDOM\_TEXT object with all surrounding whitespaces removed. Internal whitespaces are preserved. If no textual value exists for the current PBDOM\_TEXT object, or if only whitespaces exist, an empty string is returned.

### See also

[GetText](#)

[GetTextNormalize](#)

[SetText](#)

## 17.1.12 SetParentObject

### Description

Sets the referenced PBDOM\_OBJECT to be the parent of the current PBDOM\_TEXT object.



## Syntax

```
pbdom_text_name.SetParentObject(pbdom_object pbdom_object_ref)
```

**Table 17.16:**

Argument	Description
pbdom_text_name	The name of a PBDOM_TEXT object
pbdom_object_ref	A PBDOM_OBJECT to be set as the parent of the current PBDOM_TEXT object

## Return value

PBDOM\_OBJECT.

## Throws

EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_USE -- If the input PBDOM\_OBJECT is not referenced to an object derived from PBDOM\_OBJECT.

EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_HAS\_PARENT -- If the current PBDOM\_TEXT object already has a parent.

EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJECT -- If the input PBDOM\_OBJECT is of a class that does not have a proper parent-child relationship with the PBDOM\_TEXT class.

EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_OBJECT -- If the input PBDOM\_OBJECT requires a user-defined name and it has not been named.

## Usage

The PBDOM\_OBJECT that you set to be the parent of the current PBDOM\_TEXT object must have a legal parent-child relationship with the current object. If it does not, an exception is thrown. Only a PBDOM\_ELEMENT is allowed to be set as the parent of a PBDOM\_TEXT object.

## See also

[GetParentObject](#)

### 17.1.13 SetText

#### Description

Sets the input string to be the text content of the current PBDOM\_TEXT object.

#### Syntax

```
pbdom_text_name.SetText(strSet)
```

**Table 17.17:**

Argument	Description
pbdom_text_name	The name of a PBDOM_TEXT object
strSet	The string you want set as the text of the PBDOM_TEXT object

**Return value**

String.

If no DTD is referenced, an empty string is returned.

**See also**

[GetText](#)

[GetTextNormalize](#)

[GetTextTrim](#)

## 18 PBDOM Summary

### About this chapter

This chapter provides a quick reference to the methods of PBDOM base classes and additional methods provided by inherited classes.

### 18.1 Summary of PBDOM classes and methods

**Table 18.1:**

<b>PBDOM_OBJECT inherited from PowerBuilder NonVisualObject</b>
addcontent ( pbdom_object pbdom_object_ref ) returns pbdom_object
clone ( boolean bdeep ) returns pbdom_object
detach ( ) returns pbdom_object
equals ( pbdom_object pbdom_object_ref ) returns boolean
getcontent ( ref pbdom_object pbdom_object_array[] ) returns boolean
getname ( ) returns string
getobjectclass ( ) returns long
getobjectclasstring ( ) returns string
getownerdocumentobject ( ) returns pbdom_document
getparentobject ( ) returns pbdom_object
gettext ( ) returns string
gettextnormalize ( ) returns string
gettexttrim ( ) returns string
haschildren ( ) returns boolean
insertcontent ( pbdom_object pbdom_object_new, pbdom_object pbdom_object_ref ) returns pbdom_object
isancestorobjectof ( pbdom_object pbdom_object_ref ) returns boolean
removecontent ( pbdom_object pbdom_object_ref ) returns boolean
setcontent ( pbdom_object pbdom_object_array[] ) returns pbdom_object
setname ( string strname ) returns boolean
setparentobject ( pbdom_object pbdom_object_ref ) returns pbdom_object

**Table 18.2:**

<b>PBDOM_ELEMENT inherited from PBDOM_OBJECT</b>
addcontent ( string strtext ) returns pbdom_element
addnamespacedeclaration ( string strnamespaceprefix, string strnamespaceuri ) returns pbdom_element
getattribute ( string strname ) returns pbdom_attribute

getAttribute ( string strname, string strnamespaceprefix, string strnamespaceuri ) returns pbdom\_attribute

getAttributes ( ref pbdom\_attribute pbdom\_attribute\_array[] ) returns boolean

getAttributevalue ( string strattributename ) returns string

getAttributevalue ( string strattributename, string strdefaultvalue ) returns string

getAttributevalue ( string strattributename, string strnamespaceprefix, string strnamespaceuri ) returns string

getAttributevalue ( string strattributename, string strnamespaceprefix, string strnamespaceuri, string strdefaultvalue ) returns string

getChildElement ( string strelementname ) returns pbdom\_element

getChildElement ( string strelementname, string strnamespaceprefix, string strnamespaceuri ) returns pbdom\_element

getChildElements ( ref pbdom\_element pbdom\_element\_array[] ) returns boolean

getChildElements ( string strelementname, ref pbdom\_element pbdom\_element\_array[] ) returns boolean

getChildElements ( string strelementname, string strnamespaceprefix, string strnamespaceuri, ref pbdom\_element pbdom\_element\_array[] ) returns boolean

getNamespacePrefix ( ) returns string

getNamespaceUri ( ) returns string

getQualifiedName ( ) returns string

hasAttributes ( ) returns boolean

hasChildElements ( ) returns boolean

isRootElement ( ) returns boolean

removeAttribute ( pbdom\_attribute pbdom\_attribute\_ref ) returns boolean

removeAttribute ( string strattributename ) returns boolean

removeAttribute ( string strattributename, string strnamespaceprefix, string strnamespaceuri ) returns boolean

removeChildElement ( string strelementname ) returns boolean

removeChildElement ( string strelementname, string strnamespaceprefix, string strnamespaceuri ) returns boolean

removeChildElements ( ) returns boolean

removeChildElements ( string strelementname ) returns boolean

removeChildElements ( string strelementname, string strnamespaceprefix, string strnamespaceuri ) returns boolean

removenamespaceprefix ( string strnamespaceprefix, string strnamespaceuri ) returns boolean

setattribute ( pbdom\_attribute pbdom\_attribute\_ref ) returns pbdom\_element

setattribute ( string strname, string strvalue ) returns pbdom\_element

setattribute ( string strname, string strvalue, string strnamespaceprefix, string strnamespaceuri, boolean bverifynamespace ) returns long

setattributes ( pbdom\_attribute pbdom\_attribute\_array[] ) returns pbdom\_element

setdocument ( pbdom\_object pbdom\_document\_ref ) returns pbdom\_element

setnamespace ( string strnamespaceprefix, string strnamespaceuri, boolean bverifynamespace ) returns long

settext ( string strtext ) returns pbdom\_element

**Table 18.3:**

<b>PBDOM_ATTRIBUTE inherited from PBDOM_OBJECT</b>
getbooleanvalue ( ) returns boolean
getdatetimevalue ( string strdateformat, string strtimeformat ) returns datetime
getdatevalue ( string strdateformat ) returns date
getdoublevalue ( ) returns double
getintvalue ( ) returns integer
getlongvalue ( ) returns long
getnamespaceprefix ( ) returns string
getnamespaceuri ( ) returns string
getownerelementobject ( ) returns pbdom_element
getqualifiedname ( ) returns string
getrealvalue ( ) returns real
gettimevalue ( string strtimeformat ) returns time
getuintvalue ( ) returns unsignedinteger
getulongvalue ( ) returns unsignedlong
setbooleanvalue ( boolean boolvalue ) returns pbdom_attribute
setdatetimevalue ( datetime datetimevalue, string strdateformat, string strtimeformat ) returns pbdom_attribute
setdatevalue ( date datevalue, string strdateformat ) returns pbdom_attribute
setdoublevalue ( double doublevalue ) returns pbdom_attribute
setintvalue ( integer intvalue ) returns pbdom_attribute
setlongvalue ( long longvalue ) returns pbdom_attribute

<p>setnamespace ( string strnamespaceprefix, string strnamespaceuri, boolean bverifynamespace ) returns long</p> <p>setownerelementobject( pbdom_element pbdom_element_ref ) returns pbdom_attribute</p> <p>setrealvalue ( real realvalue ) returns pbdom_attribute</p> <p>settext ( string strtext ) returns pbdom_attribute</p> <p>settimevalue ( time timevalue, string strtimeformat ) returns pbdom_attribute</p> <p>setuintvalue ( unsignedinteger uintvalue ) returns pbdom_attribute</p> <p>setulongvalue ( unsignedlong ulongvalue ) returns pbdom_attribute</p>
---

**Table 18.4:**

<b>PBDOM_CHARACTERDATA inherited from PBDOM_OBJECT</b>
<p>append ( pbdom_characterdata pbdom_characterdata_ref ) returns pbdom_characterdata</p> <p>append ( string strappend ) returns pbdom_characterdata</p> <p>settext ( string strtext ) returns pbdom_characterdata</p>

**Table 18.5:**

<b>PBDOM_COMMENT inherited from PBDOM_CHARACTERDATA</b>
No added methods.

**Table 18.6:**

<b>PBDOM_TEXT inherited from PBDOM_CHARACTERDATA</b>
No added methods.

**Table 18.7:**

<b>PBDOM_CDATA inherited from PBDOM_TEXT</b>
No added methods.

**Table 18.8:**

<b>PBDOM_DOCTYPE inherited from PBDOM_OBJECT</b>
<p>getinternalsubset ( ) returns string</p> <p>getpublicid ( ) returns string</p> <p>getsystemid ( ) returns string</p> <p>setdocument ( pbdom_document pbdom_document_ref ) returns pbdom_doctype</p> <p>setinternalsubset ( string strinternalsubset ) returns pbdom_doctype</p> <p>setpublicid ( string strpublicid ) returns pbdom_doctype</p> <p>setsystemid ( string strsystemid ) returns pbdom_doctype</p>

**Table 18.9:**

<b>PBDOM_DOCUMENT inherited from PBDOM_OBJECT</b>
---

detachrootelement ( ) returns pbdom_element
getdoctype ( ) returns pbdom_doctype
getrootelement ( ) returns pbdom_element
hasrootelement ( ) returns boolean
newdocument ( string strrootelementname ) returns boolean
newdocument ( string strrootelementnamespaceprefix, string strrootelementnamespaceuri, string strrootelementname, string strdoctypepublicid, string strdoctypepublicid ) returns boolean
savedocument ( string strfilename ) returns boolean
setdoctype ( pbdom_doctype pbdom_doctype_ref ) returns pbdom_document
setrootelement ( pbdom_element pbdom_element_ref ) returns pbdom_document

**Table 18.10:**

<b>PBDOM_ENTITYREFERENCE inherited from PBDOM_OBJECT</b>
No added methods.

**Table 18.11:**

<b>PBDOM_PROCESSINGINSTRUCTION inherited from PBDOM_OBJECT</b>
getdata ( ) returns string
getnames ( ref string name_array[] ) returns boolean
gettarget ( ) returns string
getvalue ( string strname ) returns string
removevalue ( string strname ) returns boolean
setdata ( string strdata ) returns pbdom_processinginstruction
setvalue ( string strname, string strvalue ) returns pbdom_processinginstruction

**Table 18.12:**

<b>PBDOM_BUILDER inherited from PowerBuilder NonVisualObject</b>
buildfromdatastore ( datastore datastore_ref ) returns pbdom_document
buildfromfile ( string strurl ) returns pbdom_document
buildfromstring ( string strxmlstream ) returns pbdom_document
getparseerrors( ref string strErrorMessageArray[]) returns boolean

**Table 18.13:**

<b>PBDOM_EXCEPTION inherited from PowerBuilder Exception</b>
getexceptioncode ( ) returns long

# Index

## A

AddContent method  
(PBDOM\_ATTRIBUTE), [18](#)  
 AddContent method  
(PBDOM\_DOCUMENT), [143](#)  
 AddContent method (PBDOM\_ELEMENT),  
[166](#)  
 AddContent method (PBDOM\_OBJECT),  
[232](#)  
 AddNamespaceDeclaration method  
(PBDOM\_ELEMENT), [169](#)  
 Append method (PBDOM\_CDATA), [73](#)  
 Append method  
(PBDOM\_CHARACTERDATA), [91](#)  
 Append method (PBDOM\_COMMENT),  
[120](#)  
 Append method (PBDOM\_TEXT), [273](#)

## B

BuildFromDataStore method  
(PBDOM\_BUILDER), [65](#)  
 BuildFromFile method  
(PBDOM\_BUILDER), [66](#)  
 BuildFromString method  
(PBDOM\_BUILDER), [68](#)

## C

Clone method (PBDOM\_ATTRIBUTE), [19](#)  
 Clone method (PBDOM\_CDATA), [74](#)  
 Clone method  
(PBDOM\_CHARACTERDATA), [94](#)  
 Clone method (PBDOM\_COMMENT), [121](#)  
 Clone method (PBDOM\_DOCTYPE), [131](#)  
 Clone method (PBDOM\_DOCUMENT), [145](#)  
 Clone method (PBDOM\_ELEMENT), [170](#)  
 Clone method  
(PBDOM\_ENTITYREFERENCE), [82](#)  
 Clone method (PBDOM\_OBJECT), [233](#)  
 Clone method  
(PBDOM\_PROCESSINGINSTRUCTION),  
[258](#)  
 Clone method (PBDOM\_TEXT), [274](#)

## D

Detach method (PBDOM\_ATTRIBUTE), [20](#)  
 Detach method (PBDOM\_CDATA), [75](#)

Detach method  
(PBDOM\_CHARACTERDATA), [96](#)  
 Detach method (PBDOM\_COMMENT), [123](#)  
 Detach method (PBDOM\_DOCTYPE), [132](#)  
 Detach method (PBDOM\_ELEMENT), [172](#)  
 Detach method  
(PBDOM\_ENTITYREFERENCE), [84](#)  
 Detach method (PBDOM\_OBJECT), [234](#)  
 Detach method  
(PBDOM\_PROCESSINGINSTRUCTION),  
[259](#)  
 Detach method (PBDOM\_TEXT), [276](#)  
 DetachRootElement method  
(PBDOM\_DOCUMENT), [146](#)

## E

EJB Client, [12](#)  
 Equals method (PBDOM\_ATTRIBUTE), [21](#)  
 Equals method (PBDOM\_CDATA), [76](#)  
 Equals method  
(PBDOM\_CHARACTERDATA), [98](#)  
 Equals method (PBDOM\_COMMENT), [124](#)  
 Equals method (PBDOM\_DOCTYPE), [132](#)  
 Equals method (PBDOM\_DOCUMENT),  
[147](#)  
 Equals method (PBDOM\_ELEMENT), [172](#)  
 Equals method  
(PBDOM\_ENTITYREFERENCE), [85](#)  
 Equals method (PBDOM\_OBJECT), [235](#)  
 Equals method  
(PBDOM\_PROCESSINGINSTRUCTION),  
[259](#)  
 Equals method (PBDOM\_TEXT), [276](#)  
 EXCEPTION\_DATA\_CONVERSION, [229](#)  
 EXCEPTION\_HIERARCHY\_ERROR, [230](#)  
 EXCEPTION\_ILLEGAL\_PBOBJECT, [228](#)  
 EXCEPTION\_INAPPROPRIATE\_USE\_OF\_PBDOM\_OBJ  
[227](#)  
 EXCEPTION\_INTERNAL\_XML\_ENGINE\_ERROR,  
[229](#)  
 EXCEPTION\_INVALID\_ARGUMENT,  
[228](#)  
 EXCEPTION\_INVALID\_NAME, [228](#)  
 EXCEPTION\_INVALID\_OPERATION,  
[229](#)  
 EXCEPTION\_INVALID\_STRING, [229](#)  
 EXCEPTION\_MEMORY\_ALLOCATION\_FAILURE,  
[229](#)



- EXCEPTION\_MULTIPLE\_DOCTYPE, [228](#) GetDocType method
- EXCEPTION\_MULTIPLE\_ROOT\_ELEMENT(PBDOM\_DOCUMENT), [149](#)  
[227](#) GetDoubleValue method
- EXCEPTION\_MULTIPLE\_XMLDECL, (PBDOM\_ATTRIBUTE), [27](#)  
[229](#) GetElementsByTagName method
- EXCEPTION\_PBDOM\_NOT\_INITIALIZED, (PBDOM\_DOCUMENT), [149](#)  
[230](#) GetExceptionCode method
- EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_EXISTS(PBDOM\_DOCUMENT), [230](#)  
[230](#) GetInternalSubset method
- EXCEPTION\_PBDOM\_OBJECT\_ALREADY\_EXISTS(PBDOM\_DOCUMENT), [133](#)  
[227](#) GetIntValue method
- EXCEPTION\_PBDOM\_OBJECT\_INVALID\_FOR\_DOCUMENT(PBDOM\_ATTRIBUTE), [27](#)  
[227](#) GetLongValue method
- EXCEPTION\_USE\_OF\_UNNAMED\_PBDOM\_DOCUMENT(PBDOM\_ATTRIBUTE), [28](#)  
[226](#) GetName method (PBDOM\_ATTRIBUTE),
- EXCEPTION\_WRONG\_DOCUMENT\_ERROR, [28](#)  
[227](#) GetName method
- EXCEPTION\_WRONG\_PARENT\_ERROR, (PBDOM\_CHARACTERDATA), [101](#)  
[228](#) GetName method (PBDOM\_DOCTYPE),  
[133](#)
- G**
- GetAttribute method (PBDOM\_ELEMENT),  
[173](#)
- GetAttributes method  
(PBDOM\_ELEMENT), [175](#)
- GetAttributeValue method  
(PBDOM\_ELEMENT), [176](#)
- GetBooleanValue method  
(PBDOM\_ATTRIBUTE), [23](#)
- GetChildElement method  
(PBDOM\_ELEMENT), [180](#)
- GetChildElements method  
(PBDOM\_ELEMENT), [182](#)
- GetContent method  
(PBDOM\_ATTRIBUTE), [24](#)
- GetContent method  
(PBDOM\_DOCUMENT), [147](#)
- GetContent method (PBDOM\_ELEMENT),  
[186](#)
- GetContent method (PBDOM\_OBJECT),  
[235](#)
- GetData method  
(PBDOM\_PROCESSINGINSTRUCTION),  
[260](#)
- GetDateTimeValue method  
(PBDOM\_ATTRIBUTE), [26](#)
- GetDateValue method  
(PBDOM\_ATTRIBUTE), [25](#)
- GetName method (PBDOM\_ELEMENT),  
[187](#)
- GetName method  
(PBDOM\_ENTITYREFERENCE), [85](#)
- GetName method (PBDOM\_OBJECT), [236](#)
- GetName method  
(PBDOM\_PROCESSINGINSTRUCTION),  
[260](#)
- GetNames method  
(PBDOM\_PROCESSINGINSTRUCTION),  
[261](#)
- GetNamespacePrefix method  
(PBDOM\_ATTRIBUTE), [29](#)
- GetNamespacePrefix method  
(PBDOM\_ELEMENT), [188](#)
- GetNamespaceUri method  
(PBDOM\_ATTRIBUTE), [30](#)
- GetNamespaceUri method  
(PBDOM\_ELEMENT), [188](#)
- GetObjectClass method  
(PBDOM\_ATTRIBUTE), [30](#)
- GetObjectClass method (PBDOM\_CDATA),  
[76](#)
- GetObjectClass method  
(PBDOM\_CHARACTERDATA), [102](#)
- GetObjectClass method  
(PBDOM\_COMMENT), [124](#)

- GetObjectClass method (PBDOM\_DOCTYPE), [134](#)
- GetObjectClass method (PBDOM\_DOCUMENT), [150](#)
- GetObjectClass method (PBDOM\_ELEMENT), [189](#)
- GetObjectClass method (PBDOM\_ENTITYREFERENCE), [86](#)
- GetObjectClass method (PBDOM\_OBJECT), [237](#)
- GetObjectClass method (PBDOM\_PROCESSINGINSTRUCTION), [261](#)
- GetObjectClass method (PBDOM\_TEXT), [277](#)
- GetObjectClassString method (PBDOM\_ATTRIBUTE), [31](#)
- GetObjectClassString method (PBDOM\_CDATA), [77](#)
- GetObjectClassString method (PBDOM\_CHARACTERDATA), [102](#)
- GetObjectClassString method (PBDOM\_COMMENT), [125](#)
- GetObjectClassString method (PBDOM\_DOCTYPE), [134](#)
- GetObjectClassString method (PBDOM\_DOCUMENT), [151](#)
- GetObjectClassString method (PBDOM\_ELEMENT), [190](#)
- GetObjectClassString method (PBDOM\_ENTITYREFERENCE), [86](#)
- GetObjectClassString method (PBDOM\_OBJECT), [238](#)
- GetObjectClassString method (PBDOM\_PROCESSINGINSTRUCTION), [262](#)
- GetObjectClassString method (PBDOM\_TEXT), [277](#)
- GetOwnerDocumentObject method (PBDOM\_ATTRIBUTE), [32](#)
- GetOwnerDocumentObject method (PBDOM\_CDATA), [77](#)
- GetOwnerDocumentObject method (PBDOM\_CHARACTERDATA), [98](#)
- GetOwnerDocumentObject method (PBDOM\_COMMENT), [125](#)
- GetOwnerDocumentObject method (PBDOM\_DOCTYPE), [135](#)
- GetOwnerDocumentObject method (PBDOM\_ELEMENT), [190](#)
- GetOwnerDocumentObject method (PBDOM\_ENTITYREFERENCE), [86](#)
- GetOwnerDocumentObject method (PBDOM\_OBJECT), [239](#)
- GetOwnerDocumentObject method (PBDOM\_PROCESSINGINSTRUCTION), [262](#)
- GetOwnerDocumentObject method (PBDOM\_TEXT), [278](#)
- GetOwnerElementObject method (PBDOM\_ATTRIBUTE), [33](#)
- GetParentObject method (PBDOM\_CDATA), [78](#)
- GetParentObject method (PBDOM\_CHARACTERDATA), [104](#)
- GetParentObject method (PBDOM\_COMMENT), [126](#)
- GetParentObject method (PBDOM\_DOCTYPE), [135](#)
- GetParentObject method (PBDOM\_ELEMENT), [191](#)
- GetParentObject method (PBDOM\_ENTITYREFERENCE), [87](#)
- GetParentObject method (PBDOM\_OBJECT), [240](#)
- GetParentObject method (PBDOM\_PROCESSINGINSTRUCTION), [262](#)
- GetParentObject method (PBDOM\_TEXT), [278](#)
- GetParseErrors method (PBDOM\_BUILDER), [69](#)
- GetPublicID method (PBDOM\_DOCTYPE), [135](#)
- GetQualifiedName method (PBDOM\_ATTRIBUTE), [34](#)
- GetQualifiedName method (PBDOM\_ELEMENT), [192](#)
- GetRealValue method (PBDOM\_ATTRIBUTE), [35](#)
- GetRootElement method (PBDOM\_DOCUMENT), [151](#)
- GetSystemID method (PBDOM\_DOCTYPE), [136](#)

GetTarget method  
(PBDOM\_PROCESSINGINSTRUCTION),  
[263](#)

GetText method (PBDOM\_ATTRIBUTE),  
[35](#)

GetText method (PBDOM\_CDATA), [78](#)

GetText method  
(PBDOM\_CHARACTERDATA), [106](#)

GetText method (PBDOM\_COMMENT),  
[126](#)

GetText method (PBDOM\_ELEMENT), [192](#)

GetText method (PBDOM\_OBJECT), [241](#)

GetText method  
(PBDOM\_PROCESSINGINSTRUCTION),  
[263](#)

GetText method (PBDOM\_TEXT), [279](#)

GetTextNormalize method  
(PBDOM\_ATTRIBUTE), [37](#)

GetTextNormalize method  
(PBDOM\_CDATA), [79](#)

GetTextNormalize method  
(PBDOM\_CHARACTERDATA), [107](#)

GetTextNormalize method  
(PBDOM\_COMMENT), [127](#)

GetTextNormalize method  
(PBDOM\_ELEMENT), [193](#)

GetTextNormalize method  
(PBDOM\_OBJECT), [243](#)

GetTextNormalize method  
(PBDOM\_PROCESSINGINSTRUCTION),  
[264](#)

GetTextNormalize method  
(PBDOM\_TEXT), [279](#)

GetTextTrim method  
(PBDOM\_ATTRIBUTE), [39](#)

GetTextTrim method (PBDOM\_CDATA),  
[79](#)

GetTextTrim method  
(PBDOM\_CHARACTERDATA), [111](#)

GetTextTrim method  
(PBDOM\_COMMENT), [127](#)

GetTextTrim method  
(PBDOM\_ELEMENT), [194](#)

GetTextTrim method (PBDOM\_OBJECT),  
[246](#)

GetTextTrim method  
(PBDOM\_PROCESSINGINSTRUCTION),  
[265](#)

GetTextTrim method (PBDOM\_TEXT), [280](#)

GetTimeValue method  
(PBDOM\_ATTRIBUTE), [40](#)

GetUIntValue method  
(PBDOM\_ATTRIBUTE), [41](#)

GetUlongValue method  
(PBDOM\_ATTRIBUTE), [41](#)

GetValue method  
(PBDOM\_PROCESSINGINSTRUCTION),  
[265](#)

## H

HasAttributes method  
(PBDOM\_ELEMENT), [194](#)

HasChildElements method  
(PBDOM\_ELEMENT), [195](#)

HasChildren method  
(PBDOM\_ATTRIBUTE), [42](#)

HasChildren method  
(PBDOM\_CHARACTERDATA), [114](#)

HasChildren method  
(PBDOM\_DOCUMENT), [152](#)

HasChildren method (PBDOM\_ELEMENT),  
[196](#)

HasChildren method (PBDOM\_OBJECT),  
[248](#)

HasRootElement method  
(PBDOM\_DOCUMENT), [152](#)

## I

InsertContent method  
(PBDOM\_ATTRIBUTE), [43](#)

InsertContent method  
(PBDOM\_DOCUMENT), [153](#)

InsertContent method  
(PBDOM\_ELEMENT), [197](#)

InsertContent method (PBDOM\_OBJECT),  
[249](#)

IsAncestorObjectOf method  
(PBDOM\_ATTRIBUTE), [45](#)

IsAncestorObjectOf method  
(PBDOM\_CHARACTERDATA), [115](#)

IsAncestorObjectOf method  
(PBDOM\_DOCUMENT), [155](#)

IsAncestorObjectOf method  
(PBDOM\_ELEMENT), [199](#)

IsAncestorObjectOf method  
(PBDOM\_OBJECT), [250](#)

IsRootElement method  
(PBDOM\_ELEMENT), [199](#)

## L

Lowercase, [\\_\\_\\_\\_\\_](#)

## N

NewDocument method  
(PBDOM\_DOCUMENT), [155](#)

## P

PBDOM

overview, [14](#)

PBDOM objects, [15](#)

PBDOM\_ATTRIBUTE Class, [17](#)

PBDOM\_BUILDER class, [65](#)

PBDOM\_CDATA class, [72](#)

PBDOM\_CHARACTERDATA class, [90](#)

PBDOM\_COMMENT class, [119](#)

PBDOM\_DOCTYPE class, [131](#)

PBDOM\_DOCUMENT class, [142](#)

PBDOM\_ELEMENT class, [165](#)

PBDOM\_ENTITYREFERENCE class, [82](#)

PBDOM\_EXCEPTION class, [226](#)

PBDOM\_OBJECT class, [232](#)

PBDOM\_PROCESSINGINSTRUCTION  
class, [257](#)

PBDOM\_TEXT class, [272](#)

PowerBuilder extensions

about, [9](#)

get information about, [11](#)

use, [9](#)

## R

RemoveAttribute method  
(PBDOM\_ELEMENT), [200](#)

RemoveChildElement method  
(PBDOM\_ELEMENT), [202](#)

RemoveChildElements method  
(PBDOM\_ELEMENT), [204](#)

RemoveContent method  
(PBDOM\_ATTRIBUTE), [46](#)

RemoveContent method  
(PBDOM\_DOCUMENT), [159](#)

RemoveContent method  
(PBDOM\_ELEMENT), [207](#)

RemoveContent method  
(PBDOM\_OBJECT), [252](#)

RemoveNamespaceDeclaration method  
(PBDOM\_ELEMENT), [208](#)

RemoveValue method  
(PBDOM\_PROCESSINGINSTRUCTION),  
[266](#)

## S

SaveDocument method  
(PBDOM\_DOCUMENT), [160](#)

SaveDocumentIntoString method  
(PBDOM\_DOCUMENT), [161](#)

SetAttribute method (PBDOM\_ELEMENT),  
[209](#)

SetAttributes method  
(PBDOM\_ELEMENT), [216](#)

SetBooleanValue method  
(PBDOM\_ATTRIBUTE), [48](#)

SetContent method  
(PBDOM\_ATTRIBUTE), [48](#)

SetContent method  
(PBDOM\_DOCUMENT), [161](#)

SetContent method (PBDOM\_ELEMENT),  
[218](#)

SetContent method (PBDOM\_OBJECT),  
[253](#)

SetData method  
(PBDOM\_PROCESSINGINSTRUCTION),  
[266](#)

SetDateTimeValue method  
(PBDOM\_ATTRIBUTE), [52](#)

SetDateValue method  
(PBDOM\_ATTRIBUTE), [51](#)

SetDocType method  
(PBDOM\_DOCUMENT), [163](#)

SetDocument method  
(PBDOM\_DOCTYPE), [137](#)

SetDocument method  
(PBDOM\_ELEMENT), [221](#)

SetDoubleValue method  
(PBDOM\_ATTRIBUTE), [53](#)

SetInternalSubset method  
(PBDOM\_DOCTYPE), [137](#)

SetIntValue method  
(PBDOM\_ATTRIBUTE), [54](#)

SetLongValue method  
(PBDOM\_ATTRIBUTE), [54](#)

SetName method (PBDOM\_ATTRIBUTE),  
[55](#)

SetName method (PBDOM\_DOCTYPE), [138](#)  
SetName method (PBDOM\_ELEMENT), [221](#)  
SetName method  
(PBDOM\_ENTITYREFERENCE), [87](#)  
SetName method (PBDOM\_OBJECT), [254](#)  
SetName method  
(PBDOM\_PROCESSINGINSTRUCTION), [268](#)  
SetNamespace method  
(PBDOM\_ATTRIBUTE), [57](#)  
SetNamespace method  
(PBDOM\_ELEMENT), [222](#)  
SetOwnerElementObject method  
(PBDOM\_ATTRIBUTE), [59](#)  
SetParentObject method  
(PBDOM\_CDATA), [80](#)  
SetParentObject method  
(PBDOM\_CHARACTERDATA), [116](#)  
SetParentObject method  
(PBDOM\_COMMENT), [128](#)  
SetParentObject method  
(PBDOM\_DOCTYPE), [139](#)  
SetParentObject method  
(PBDOM\_ELEMENT), [223](#)  
SetParentObject method  
(PBDOM\_ENTITYREFERENCE), [88](#)  
SetParentObject method  
(PBDOM\_OBJECT), [254](#)  
SetParentObject method  
(PBDOM\_PROCESSINGINSTRUCTION), [269](#)  
SetParentObject method (PBDOM\_TEXT), [280](#)  
SetPublicID method (PBDOM\_DOCTYPE), [140](#)  
SetRealValue method  
(PBDOM\_ATTRIBUTE), [61](#)  
SetRootElement method  
(PBDOM\_DOCUMENT), [164](#)  
SetSystemID method  
(PBDOM\_DOCTYPE), [141](#)  
SetText method (PBDOM\_ATTRIBUTE), [61](#)  
SetText method (PBDOM\_CDATA), [80](#)  
SetText method  
(PBDOM\_CHARACTERDATA), [118](#)

SetText method (PBDOM\_COMMENT), [129](#)  
SetText method (PBDOM\_ELEMENT), [224](#)  
SetText method (PBDOM\_TEXT), [281](#)  
SetTimeValue method  
(PBDOM\_ATTRIBUTE), [62](#)  
SetUIntValue method  
(PBDOM\_ATTRIBUTE), [63](#)  
SetUlongValue method  
(PBDOM\_ATTRIBUTE), [63](#)  
SetValue method  
(PBDOM\_PROCESSINGINSTRUCTION), [270](#)

## W

Web Services Client (Obsolete), [13](#)