

SAFOX - simple API for XML Handling in object oriented PHP



Contents

Package SAFOX Procedural Elements	2
safox.cls.php	2
Define SAFOXVERSION	4
Package SAFOX Classes	5
Class SAFOX	5
Constructor SAFOX	5
Method createRSSDoc	5
Method createRSSParser	5
Method createXMLDoc	5
Method createXMLParser	6
Method getVersion	6
Method ver	6
safox_g.cls.php	7
Define SAFOXDOC	8
Define SAFOX_G	8
Class xmlDoc	9
Var \$EntityConversion	9
Constructor xmlDoc	9
Method addNode	9
Method addNodeAfter	9
Method addNodeBefore	10
Method cleanUp	10
Method destroy	10
Method getChildNodeByTagName	11
Method getId	11
Method getNextNode	11
Method getNodeById	11
Method setEncoding	11
Method setEntityConversion	12
Method setStandAlone	12
Method setUida	12
Method setVersion	12
Method writeToFile	13
Method writeXML	13
Class xmlNode	13
Var \$EntityConversion	14
Constructor xmlNode	14
Method addNode	14
Method addNodeAfter	14
Method addNodeBefore	15
Method cleanUp	15
Method convertString2Entities	15
Method delete	15

Method destroy	16
Method getAttribute	16
Method getCData	16
Method getChildNodeByTagName	16
Method getChildNodeByType	17
Method getId	17
Method getNextNode	17
Method getParent	17
Method getType	17
Method hasChildren	17
Method hasParent	17
Method remove	17
Method setAttribute	18
Method setCData	18
Method setEntityConversion	18
Method setId	19
Method setParent	19
Method setType	19
Method writeXML	19
safox_p.cls.php	20
Define SAFOX_P	22
Define XML_MAX_LINE_SIZE	22
Define XPE_ON_PARSE_ATT	22
Define XPE_ON_PARSE_CDT	22
Define XPE_ON_PARSE_CMT	22
Define XPE_ON_PARSE_DEF	22
Define XPE_ON_PARSE_DTD	22
Define XPE_ON_PARSE_NOD	22
Define XPE_ON_PARSE_TAG	22
Define XPE_ON_PARSE_XSP	22
Function onParseCDATA	22
Class xmlParser	23
Constructor xmlParser	23
Method destroy	23
Method getError	23
Method getErrors	23
Method getXmlDoc	24
Method loadFile	24
Method parse	24
Method registerEvent	24
Method setString	25
Method setUida	25
safox_g_rss.cls.php	27
Define SAFOXVERSIONRSS	28
Define SAFOX_G_RSS	28
safox_p_rss.cls.php	29
Define SAFOX_P_RSS	30
Class RSSDoc	30
Constructor RSSDoc	30

Method addItem	31
Method getAllItems	31
Method getChannelObject	31
Method getFirstItem	31
Method getLastItem	31
Method getNextItem	31
Method getPrevItem	31
Method printRSS	32
Method setChannelCategory	32
Method setChannelCopyright	32
Method setChannelData	32
Method setChannelDescription	32
Method setChannelDocs	33
Method setChannelGenerator	33
Method setChannelImage	33
Method setChannelLanguage	33
Method setChannelLastBuildDate	34
Method setChannelLink	34
Method setChannelManagingEditor	34
Method setChannelPubDate	34
Method setChannelRating	34
Method setChannelTitle	35
Method setChannelTTL	35
Method setChannelWebMaster	35
Method takeOverXML	35
Method writeRSS	36
Method writeToFile	36
Class RSSItem	36
Var \$rssDoc	37
Var \$self	37
Constructor RSSItem	37
Method getAuthor	37
Method getCategory	37
Method getComments	37
Method getDescription	37
Method getLink	37
Method getPubDate	37
Method getSource	38
Method getTitle	38
Method setAuthor	38
Method setCategory	38
Method setComments	38
Method setDescription	38
Method setEnclosure	38
Method setGuid	38
Method setLink	38
Method setPubDate	38
Method setSource	38
Method setTitle	39

Class RSSParser	39
Constructor RSSParser	39
Method getRSSDoc	39
Method loadFile	39
Method parse	40
Method parseRSS	40
Method setString	40
Appendices	41
Appendix A - Class Trees	42
SAFOX	42

Package SAFOX Procedural Elements

safox.cls.php

This is the SAFOX API Project Wrapper. The SAFOX package is a collection of light-weight API for object-oriented PHP to handle XML files.

As of version 0.5 the SAFOX wrapper is the starting point for all SAFOX operations. It is an extreme light weight class that allows to create any of the SAFOX objects (XMLDoc, XMLParser, RSSDoc, RSSParser) without further include or require commands. All necessary libraries of the SAFOX API will be loaded only when/if the file is required, thus making scripts lighter and faster.

Providing three base classes, XMLDoc, XMLNode, XMLParser, and extended RSSDoc and RSSParser Classes for RSS 2.0 Handling, the SAFOX package allows an easy and clean way to handle all kind of XML files. The SAFOX package is currently developed in PHP4.X and copyright by Christian Hansel, cpi service, Leipzig. The code is provided under the GNU Public License Version 2 and may freely be used, modified, and distributed with original copyright notices to be maintained.

If no copy of the license is provided check <http://www.gnu.org/licenses/gpl2.txt>

The SAFOX Package consists of five files:

- safox.cls.php -- A OOP wrapper for all SAFOX objects which loads one or more of the following files only when required
- safox_g.cls.php -- Provides the XMLDOc & XMLNODE Classes
- safox_p.cls.php -- Provides a simple, light-weight, but strong Parser for XML Documents that returns a XMLDOC Object

The SAFOX_RSS sub package consists of two files but requires the SAFOX Base classes:

- safox_g_rss.cls.php -- Provides the RSSDOc & RSSNODE Classes
- safox_p_rss.cls.php -- Provides a simple, light-weight, but strong Parser for XML Documents that returns a XMLDOC Object

This is the wrapper/main SAFOX library providing access to all SAFOX objects Changes History : version 0.5

- The SAFOX library has been reorganised and now requires only one wrapper file (safox.cls.php)
- added subpackage SAFOX_RSS provides two classes RSSDoc and RSSParser for the generation and parsing of RSS 2.0
- added Feature: setEntityConversion is a function that specifies whether or not special

characters in CDATA

should be converted into HT/XML-Entities (e.g. & into &, ' into ' ...)

(true) or enclosed into CDATA tags (false)

version 0.42

- BUGFIX 20051130-2 fixes a problem where comments containing tags caused the the parser to break
 - version 0.41 of SAFOX package includes changes
- BUGFIX 20051130 Code Cleanup provided by Cristiano Degiorgis [cri@webprofessor.it]
Changes History : version 0.4 of SAFOX package includes changes
- BUGFIX 20051128 that fixes problems with removing nodes from the document
- remove, and delete methods added as aliases to destroy in XMLNode class
- method cleanUP added to XMLDoc class
- method cleanUP added to XMLNode class - see destription
Changes History : version 0.3 of SAFOX package includes changes
- BUGFIX 20051123 corrects a problem that was caused when a repeated call to setld of child Nodes was made
- addNodeAfter Method added to XMLDoc Class
- addNodeAfter Method added to XMLNode Class
- addNodeBefore Method added to XMLDoc Class
- addNodeBefore Method added to XMLNode Class
 - version 0.2 of SAFOX package includes changes
- getChildNodeByTagName() Method added to XMLDOC
- writeToFile Method added to XMLDOC
- BUGFIX 20051117 in xmlParser - corrects a mishandling of <![CDATA[..]]> enclosed tag content that
 - caused the parser to break when dealing with tags within the CDATA

- **Package** SAFOX
- **Copyright** (c) 2003-2006 CVH, cpi-service
- **File** safox.cls.php
- **Date** 2006-02-07
- **Version** 0.5
- **Link** <http://www.cpi-service.com>
- **License** GPL2
- **Author** CVH, cpi-service ; cvh@cpi-service.com

SAFOXVERSION = 0.5 [*line 74*]

Package SAFOX Classes

Class SAFOX

[line 83]

SAFOX a wrapper class for the SAFOX API

The SAFOX Wrapper allows to create all SAFOX objects but only binds/loads source file when considered necessary. A single require / include line of code (include "safox.cls.php";) gives access to all source files and classes

- **Package** SAFOX
- **Author** CVH
- **Version** 0.5

Constructor function SAFOX::SAFOX() *[line 87]*

SAFOX constructor

RSSDoc function SAFOX::createRSSDoc(\$title, \$link, \$description) *[line 125]*

creates a RSSDoc (RSS 2.0) Generation object and returns a reference to this. automatically loads necessary files

RSSParser function SAFOX::createRSSParser() *[line 140]*

creates a RSSParser (RSS 2.0) object and returns a reference to this. automatically loads necessary files

XMLDoc function SAFOX::createXMLDoc() *[line 95]*

creates a XMLDoc Generation object and returns a reference to this. automatically

loads necessary files

@ function SAFOX::createXMLParser() [*line 110*]

creates a XMLParser object and returns a reference to this. automatically loads necessary files

float function SAFOX::getVersion() [*line 154*]

returns the current version of the SAFOX API

float function SAFOX::ver() [*line 161*]

alias to getVersion() - returns the current version of the SAFOX API

safox_g.cls.php

The SAFOX package is a collection of light-weight API for object-oriented PHP to handle XML files.

As of version 0.5 the SAFOX wrapper is the starting point for all SAFOX operations. It is an extreme light weight class that allows to create any of the SAFOX objects (XMLDoc, XMLParser, RSSDoc, RSSParser) without further include or require commands. All necessary libraries of the SAFOX API will be loaded only when/if the file is required, thus making scripts lighter and faster.

Providing three base classes, XMLDoc, XMLNode, XMLParser, and extended RSSDoc and RSSParser Classes for RSS 2.0 Handling, the SAFOX package allows an easy and clean way to handle all kind of XML files. The SAFOX package is currently developed in PHP4.X and copyright by Christian Hansel, cpi service, Leipzig. The code is provided under the GNU Public License Version 2 and may freely be used, modified, and distributed with original copyright notices to be maintained.

If no copy of the license is provided check <http://www.gnu.org/licenses/gpl2.txt>

The SAFOX Package consists of five files:

- safox.cls.php -- A OOP wrapper for all SAFOX objects which loads one or more of the following files only when required
- safox_g.cls.php -- Provides the XMLDOC & XMLNODE Classes
- safox_p.cls.php -- Provides a simple, light-weight, but strong Parser for XML Documents that returns a XMLDOC Object

The SAFOX_RSS sub package consists of two files but requires the SAFOX Base classes:

- safox_g_rss.cls.php -- Provides the RSSDOC & RSSNODE Classes
- safox_p_rss.cls.php -- Provides a simple, light-weight, but strong Parser for XML Documents that returns a XMLDOC Object

full api documentation and examples available at <http://www.cpi-service.com/safox/>

This is the generation library for the object-oriented creation of xml files

Changes History : version 0.5

- The SAFOX library has been reorganised and now requires only one wrapper file (safox.cls.php)
- added subpackage SAFOX_RSS provides two classes RSSDoc and RSSParser for the generation and parsing of RSS 2.0
- added Feature: setEntityConversion is a function that specifies whether or not special characters in CDATA

should be converted into HT/XML-Entities (e.g. & into &, ' into ' ...)

(true) or enclosed into CDATA tags (false)

version 0.42

- BUGFIX 20051130-2 fixes a problem where comments containing tags caused the the parser to break
version 0.41 of SAFOX package includes changes
- BUGFIX 20051130 Code Cleanup provided by Cristiano Degiorgis [cri@webprofessor.it]
version 0.4 of SAFOX package includes changes
- BUGFIX 20051128 that fixes problems with removing nodes from the document includes complete rework of destroy method
- remove, and delete methods added as aliases to destroy in XMLNode class
- method cleanUP added to XMLDoc class
- method cleanUP added to XMLNode class - see destription
version 0.3 of SAFOX package includes changes
- BUGFIX 20051123 corrects a problem that was caused when a repeated call to setld of child Nodes was made
- addNodeAfter Method added to XMLDoc Class
- addNodeAfter Method added to XMLNode Class
- addNodeBefore Method added to XMLDoc Class
- addNodeBefore Method added to XMLNode Class
version 0.2 of SAFOX package includes changes
- getChildNodeByTagName() Method added to XMLDOC
- writeToFile Method added to XMLDOC
- BUGFIX 20051117 in xmlParser - corrects a mishandling of <![CDATA[..]]> enclosed tag content that
caused the parser to break when dealing with tags within the CDATA

- **Package** SAFOX
- **Sub-Package** SAFOX_G
- **Copyright** (c) 2003-2006 CVH, cpi-service
- **File** safox_g.cls.php
- **Date** 2006-02-07
- **Version** 0.5
- **Link** <http://www.cpi-service.com>
- **License** GPL2
- **Author** CVH, cpi-service ; cvh@cpi-service.com

SAFOXDOC = -1 *[line 79]*

SAFOX_G = true *[line 77]*

Class xmlDoc

[line 95]

class xmlDoc

Changes History : See File Information for changes *

- **Package** SAFOX
- **Sub-Package** SAFOX_G
- **License** GNU
- **Version** 0.5
- **Author** CVH, cpi-service; cvh@cpi-service.com

xmlDoc::\$EntityConversion

mixed = true [line 138]

Constructor

Constructor function xmlDoc::xmlDoc() [line 140]

xmlnode function xmlDoc::addNode([\$nodeType = "DATA"], [\$sid = false]) [line 222]

Function Parameters:

- *\$nodeType* **\$nodeType** typ or TagName of of node to be created e.g. "product" for <product>
- *\$id* **\$id** sets a value for the uida (unique ID attribute)

addNode Adds a Node to the Main part of the xmlDoc;

xmlnode function xmlDoc::addNodeAfter(&\$nodeBefore, [\$nodeType = "DATA"], [\$sid = false]) [line 248]

Function Parameters:

- *\$nodeBefore* &**\$nodeBefore**
 - xmlreference to the Node
- *\$nodeType* **\$nodeType** or TagName of of node to be created e.g. "product" for <product>
- *\$id* **\$id** sets a value for the uida (unique ID attribute)

Adds a new Node to the Document directly **after a specified node, whereas `addNode()` method adds a node to the end of the Document**

xmlNode function `xmlDoc::addNodeBefore(&$nodeAfter, [$nodeType = "DATA"], [$id = false])` [*line 301*]
Function Parameters:

- *\$nodeBefore* &**\$nodeAfter**
 - xmlreference to the Node
- *\$nodeType* **\$nodeType** or TagName of of node to be created e.g. "product" for <product>
- *\$id* **\$id** sets a value for the uida (unique ID attribute)

Adds a new Node to the Document directly **before a specified node, whereas `addNode()` method adds a node to the end of the Document**

void function `xmlDoc::cleanUp()` [*line 190*]

Cleans up internal data arrays from removed nodes and rebuilds the ID Referencing Array

This method should be called directly after removing nodes from the document, need not be called if `node->destroy(clean)` is called with optional parameter `clean = true` (default) When removing more than one node it is faster to call `node->destroy(clean)` with parameter `clean` set to false and calling `xmlDoc->cleanUp()` after making all removals

void function `xmlDoc::destroy([$clean = false])` [*line 160*]

Function Parameters:

- *clean* **\$clean**
 - boolean if set true all nodes will be destroyed recursively

Destructor- empties internal data

mixed function xmlDoc::getChildNodeByTagName(\$tagName) [line 456]

Function Parameters:

- *string \$tagName* -- node's tagname to return

getChildNodeByTagName - direct reference to `xmlNode` Object if only one tag found, array of `xmlNode` Object references if more found, or boolean false -- suggested by Cristiano

SAFOXDOC function xmlDoc::getId() [line 410]

return Id (Constant SAFOXDOC) This function returns the mixed returns the UID (Unique ID Attribute) Value of current node, for the XMLDOC returns SAFOXDOC (constant)

mixed function xmlDoc::getNextNode([\$step = 1]) [line 429]

Function Parameters:

- *\$step \$step* -- Int Step & Direction : -1 previous;0-current; 1-next; 2-next after next etc.

Returns next Reference to Child Node or False;

mixed function xmlDoc::getNodeById(\$id) [line 416]

Function Parameters:

- *\$id \$id*
 - mixed registered ID value (UID)

Return a xmlNodeObject or False

void function xmlDoc::setEncoding([\$encoding = "utf-8"]) [line 354]

Function Parameters:

- *\$encoding \$encoding*

- string by default "utf-8"

Set the XML-Document Encoding Attribute

void function xmlDoc::setEntityConversion([\$bool = true]) [line 152]

Function Parameters:

- *bool \$bool* boolean

setEntityConversion is a function that specifies whether or not special characters in **CDATA** should be converted into HT/XML-Entities (e.g.

& into &, ' into ' ...) (true) or enclosed into CDATA tags (false)
added 20060125

void function xmlDoc::setStandAlone([\$standalone = true]) [line 370]

Function Parameters:

- *\$standalone \$standalone*
 - boolean sets the standalone attribute of xml document

Sets the XML Document StandAlone Attribute

void function xmlDoc::setUida([\$uida = "id"]) [line 177]

Function Parameters:

- *\$uida \$uida*
 - the name of the unique ID attribute

Set the "Unique ID Attribute"-name (usually ID)

void function xmlDoc::setVersion([\$version = "1.0"]) [line 362]

Function Parameters:

- ***\$version* \$version**
 - string by default "1.0"

Sets the XML-Document Version-Attribute

boolean function xmlDoc::writeToFile(\$filepath) [*line 520*]

Function Parameters:

- ***\$filepath* \$filepath**
 - specified filepath

Writing an XML Document to File

string function xmlDoc::writeXML() [*line 477*]

returns a string of the XML Document

Class xmlNode

[*line 545*]

class xmlNode

Changes History : See File Information for changes

- **Package** SAFOX
- **Sub-Package** SAFOX_G
- **License** GNU
- **Version** 0.5
- **Author** CVH, cpi-service; cvh@cpi-service.com

xmlNode::\$EntityConversion

mixed = true [line 600]

EntityConversion - flag used to determine whether or not special characters in CDATA should be converted into HT/XML-Entities (e.g.

& into &, ' into ' ...) (true) or enclosed into CDATA tags (false)
added 20060125

Constructor *void* function xmlNode::xmlNode([\$type = "DATA"], &\$doc) *[line 612]*

Function Parameters:

- *\$type* **\$type**
- *\$doc* **&\$doc**

Constructor

reference function xmlNode::addNode(\$type, [\$sid = "false"]) *[line 765]*

Function Parameters:

- *\$type* **\$type** -- Type of tag/Node

addNode - Adds a child node to the current Object (Node) and returns a reference to childNode

xmlnode function xmlNode::addNodeAfter(&\$nodeBefore, [\$nodeType = "DATA"], [\$sid = false]) *[line 786]*

Function Parameters:

- *\$nodeBefore* **&\$nodeBefore**
 - xmlreference to the Node
- *\$nodeType* **\$nodeType** or TagName of of node to be created e.g. "product" for <product>
- *\$sid* **\$sid** sets a value for the uida (unique ID attribute)

Adds a new Node to the Node directly *after* a specified ChildNode, whereas

addNode() method adds a node to the end of the Node

xmlNode function `xmlNode::addNodeBefore(&$nodeAfter, [$nodeType = "DATA"], [$sid = false])` *[line 839]*

Function Parameters:

- ***\$nodeBefore* &*\$nodeAfter***
 - xmlreference to the Node
- ***\$nodeType* *\$nodeType*** or TagName of of node to be created e.g. "product" for <product>
- ***\$sid* *\$id*** sets a value for the uida (unique ID attribute)

Adds a new Node to the Node directly **before a specified ChildNode, whereas `addNode()` method adds a node to the end of the Node**

void function `xmlNode::cleanUp()` *[line 915]*

Cleans up internal data arrays from removed nodes and rebuilds the ID Referencing Array

This method should not be called directly. Instead call the CleanUp method of the parenting XMLDOC Object after removing nodes from the document. This function need not be called if `node->destroy(clean)` is called with optional parameter `clean = true` (default). When removing more than one node it is faster to call `node->destroy(clean)` with parameter `clean` set to false and calling `xmlDoc->cleanUp()` after making all removals.

string function `xmlNode::convertString2Entities($string)` *[line 980]*

Function Parameters:

- *string* ***\$string***

`convertString2Entities` - returns a string that represents a well formed xml-string to be used as CDATA based on `setEntityConversion()` converts the parameter into HT/XML-Entities or encapsulates in `<![CDATA[]]>`

void function `xmlNode::delete([$clean = true])` *[line 936]*

Function Parameters:

- ***\$clean* *\$clean***
 - boolean if set true deletes child-nodes recursively and resets the internal arrays if this

Alias to destroy - Deletes a node and all childnodes, if the optional parameter clean is set true (default), a call to the cleanUp Method of the XMLDOC Object is made automatically.

To enhance performance you may want to set clean to false if you are removing more than one node and call xmlDoc->cleanUp() after these operations manually. Otherwise the cleanUp operation is called automatically everytime a call to destroy is made

void function XmlNode::destroy([\$clean = true]) [*line 960*]

Function Parameters:

- *\$clean* **\$clean**
 - boolean if set true deletes child-nodes recursively and resets the internal arrays if this

Deletes a node and all childnodes, if the optional parameter clean is set true (default), a call to the cleanUp Method of the XMLDOC Object is made automatically.

To enhance performance you may want to set clean to false if you are removing more than one node and call xmlDoc->cleanUp() after these operations manually. Otherwise the cleanUp operation is called automatically everytime a call to destroy is made

mixed function XmlNode::getAttribute(\$attribute) [*line 662*]

Function Parameters:

- *\$attribute* **\$attribute**
 - string name of attribute

getAttribute returns the value of an attribute

unknown function XmlNode::getCdata() [*line 677*]

setCdata - returns the CData content of a node

direct function XmlNode::getChildNodeByTagName(\$tagName) [*line 729*]

Function Parameters:

- *string* **\$tagName** -- node's tagname to return

getChildNodeByTagName - returns a reference to specific **ChildNode** of current **Node**
-- suggested by Cristiano

direct function `xmlNode::getChildNodeByType($type, $tagName)` [*line 720*]

Function Parameters:

- *string* **\$tagName** -- node's tagname to return

alias to getChildNodeByTagName - returns a reference to specific **ChildNode** of current **Node** -- suggested by Cristiano

mixed function `xmlNode::getIId()` [*line 684*]

getIId -- return the Unique ID attribute value

reference function `xmlNode::getNextNode([$step = 1])` [*line 692*]

Function Parameters:

- *optional* **\$step** -- defines step and direction (+1 = next, -1 = previos etc.)

getNextNode - returns a reference to the Next **ChildNode** of current **Node**

reference function `xmlNode::getParent()` [*line 755*]

getParent - Returns a reference to the Parent **NodeObject** or null

string function `xmlNode::getType()` [*line 630*]

getType returns the type/TagName of a xml node

boolean function `xmlNode::hasChildren()` [*line 901*]

hasChildren - Returns true if childNodes exist or false

boolean function `xmlNode::hasParent()` [*line 748*]

hasParent - Returns true if a reference to a parent node exists

void function `xmlNode::remove([$clean = true])` [*line 948*]

Function Parameters:

- ***\$clean* \$clean**
 - boolean if set true deletes child-nodes recursively and resets the internal arrays if this

Alias to destroy - Deletes a node and all childnodes, if the optional parameter clean is set true (default), a call to the cleanUp Method of the XMLDOC Object is made automatically.

To enhance performance you may want to set clean to false if you are removing more than one node and call xmlDoc->cleanUp() after these operations manually. Otherwise the cleanUp operation is called automatically everytime a call to destroy is made

void function xmlNode::setAttribute(\$attribute, \$value) [line 654]

Function Parameters:

- ***\$attribute* \$attribute**
 - name of an attribute
- ***\$value* \$value**
 - string or int - Value to be set

setAttribute sets an Attribute

void function xmlNode::setCDATA(\$string) [line 670]

Function Parameters:

- ***\$string* \$string**
 - string value of content to be set

setCDATA - sets the CDATA content of a node

- function xmlNode::setEntityConversion([\$bool = true]) [line 1000]

Function Parameters:

- ***bool* \$bool** boolean

setEntityConversion is a function that specifies whether or not special characters in CDATA should be converted into HT/XML-Entities (e.g. & into &, ' into ' ...) (true) or enclosed into CDATA tags (false)
added 20060125

void function xmlNode::setId(\$id) [line 638]

Function Parameters:

- **\$id \$id**
 - mixed string or int that uniquely identifies the node

setId sets the Unique ID Attribute defined by xmlDoc->setUida or "ID"

void function xmlNode::setParent(&\$node) [line 892]

Function Parameters:

- **\$node &\$node**
 - reference to parentNode

setParent - Sets the parentNode (usually used internally or if moved)

void function xmlNode::setType(\$type) [line 623]

Function Parameters:

- **\$type \$type** string of tagname/typ of node e.g. "product" for <product>

setType Sets the type of a xml node/tag

string function xmlNode::writeXML() [line 1007]

returns a string containing the wellformed xml of current Node/tag

safox_p.cls.php

The SAFOX package is a collection of light-weight API for object-oriented PHP to handle XML files.

As of version 0.5 the SAFOX wrapper is the starting point for all SAFOX operations. It is an extreme light weight class that allows to create any of the SAFOX objects (XMLDoc, XMLParser, RSSDoc, RSSParser) without further include or require commands. All necessary libraries of the SAFOX API will be loaded only when/if the file is required, thus making scripts lighter and faster.

Providing three base classes, XMLDoc, XMLNode, XMLParser, and extended RSSDoc and RSSParser Classes for RSS 2.0 Handling, the SAFOX package allows an easy and clean way to handle all kind of XML files. The SAFOX package is currently developed in PHP4.X and copyright by Christian Hansel, cpi service, Leipzig. The code is provided under the GNU Public License Version 2 and may freely be used, modified, and distributed with original copyright notices to be maintained.

If no copy of the license is provided check <http://www.gnu.org/licenses/gpl2.txt>

The SAFOX Package consists of five files:

- safox.cls.php -- A OOP wrapper for all SAFOX objects which loads one or more of the following files only when required
- safox_g.cls.php -- Provides the XMLDOc & XMLNODE Classes
- safox_p.cls.php -- Provides a simple, light-weight, but strong Parser for XML Documents that returns a XMLDOC Object

The SAFOX_RSS sub package consists of two files but requires the SAFOX Base classes:

- safox_g_rss.cls.php -- Provides the RSSDOc & RSSNODE Classes
- safox_p_rss.cls.php -- Provides a simple, light-weight, but strong Parser for XML Documents that returns a XMLDOC Object

This is the Parsing Base Library for OOP-parsing of xml files to retrieve OOP object (XMLDoc)

Events fired during Parsing are: XPE_ON_PARSE_DEF - Event On Parsing <?xml .. ?>

XPE_ON_PARSE_DTD - On Parsing DTD

XPE_ON_PARSE_TAG - On Parsing TagName/Type of Node

XPE_ON_PARSE_ATT - On Parsing Attribute String

XPE_ON_PARSE_CDT - On Parsing CDATA/content of node

XPE_ON_PARSE_NOD - On Node Parsing Completed Object

XPE_ON_PARSE_CMT - On Parsing a Comment

XPE_ON_PARSE_XSP - On Parsing XML Specific Tags <? ?> * see the documented onParseCDATA function that is by default registered as eventhandler for XPE_ON_PARSE_CDT for further information and an explanatory example full api documentation and examples available at <http://www.cpi-service.com/safox/>

Changes History : version 0.5

- The SAFOX library has been reorganised and now requires only one wrapper file (safox.cls.php)
- added subpackage SAFOX_RSS provides two classes RSSDoc and RSSParser for the generation and parsing of RSS 2.0
- added Feature: setEntityConversion is a function that specifies whether or not special characters in CDATA should be converted into HT/XML-Entities (e.g. & into &, ' into ' ...)
(true) or enclosed into CDATA tags (false)
version 0.42
- BUGFIX 20051130-2 fixes a problem where comments containing tags caused the the parser to break
version 0.41 of SAFOX package includes changes
- BUGFIX 20051130 Code Cleanup provided by Cristiano Degiorgis [cri@webprofessor.it]
Changes History : version 0.4 of SAFOX package includes changes
- BUGFIX 20051128 that fixes problems with removing nodes from the document
- remove, and delete methods added as aliases to destroy in XMLNode class
- method cleanUP added to XMLDoc class
- method cleanUP added to XMLNode class - see destription
Changes History : version 0.3 of SAFOX package includes changes
- BUGFIX 20051123 corrects a problem that was caused when a repeated call to setld of child Nodes was made
- addNodeAfter Method added to XMLDoc Class
- addNodeAfter Method added to XMLNode Class
- addNodeBefore Method added to XMLDoc Class
- addNodeBefore Method added to XMLNode Class
version 0.2 of SAFOX package includes changes
- getChildNodeByTagName() Method added to XMLDOC
- writeToFile Method added to XMLDOC
- BUGFIX 20051117 in xmlParser - corrects a mishandling of <![CDATA[..]]> enclosed tag content that
caused the parser to break when dealing with tags within the CDATA

- **Package** SAFOX
- **Sub-Package** SAFOX_P
- **Copyright** (c) 2003-2005 CVH, cpi-service
- **File** safox_p.cls.php
- **Date** 2005-11-28
- **Version** 0.5
- **Link** <http://www.cpi-service.com>

- **License** GPL2
- **Author** CVH, cpi-service ; cvh@cpi-service.com

SAFOX_P = true *[line 100]*

Event identifiers & constants

Constants defining types of events fired during parsing

XML_MAX_LINE_SIZE = 4096 *[line 105]*

XPE_ON_PARSE_ATT = 3 *[line 109]*

XPE_ON_PARSE_CDT = 4 *[line 110]*

XPE_ON_PARSE_CMT = 6 *[line 112]*

XPE_ON_PARSE_DEF = 0 *[line 106]*

XPE_ON_PARSE_DTD = 1 *[line 107]*

XPE_ON_PARSE_NOD = 5 *[line 111]*

XPE_ON_PARSE_TAG = 2 *[line 108]*

XPE_ON_PARSE_XSP = 7 *[line 113]*

void function onParseCDATA(&\$cdata, [\$node = null], [\$parser = null]) *[line 148]*

Function Parameters:

- **\$cdata &\$cdata**
 - referenced variable containing the content /CDATA that was detected during parsing
- **\$node \$node**
 - referenced node object to which the content will be added during parsing
- **\$parser \$parser**
 - reference to parser object

OnParseCDATA - is registered in Class XMLPARSER as a PreProcessing EventHandler for CDATA-Strings called when the XPE_ON_PARSE_CDT event is fired

Listed here to demonstrate how events should be handled

function onParseCDATA(&\$cdata,\$node = null,\$parser = null) {

if (\$cdata) {

\$search = array ("\\[CDATA\[\'i\",\'i\](\[r\n\s]*\)\]",

""&(quot|#34);\'i\", ""&(amp|#38);\'i\",

""&(lt|#60);\'i\", ""&(gt|#62);\'i\", ""&(apos|#39);\'i");

\$replace = array("", "", "", "\'", "&", "<", ">", "");

\$cdata = preg_replace (\$search, \$replace, \$cdata);

}} **Events fired during Parsing are:** XPE_ON_PARSE_DEF - Event On Parsing <?xml ..

?>

XPE_ON_PARSE_DTD - On Parsing DTD

XPE_ON_PARSE_TAG - On Parsing TagName/Type of Node
XPE_ON_PARSE_ATT - On Parsing Attribute String
XPE_ON_PARSE_CDT - On Parsing CDATA/content of node
XPE_ON_PARSE_NOD - On Node Parsing Completed Object
XPE_ON_PARSE_CMT - On Parsing a Comment
XPE_ON_PARSE_XSP - On Parsing XML Specific Tags <? ?> *

require_once dirname(__FILE__)."/". "safox_g.cls.php" [line 102]

Class xmlParser

[line 192]

class xmlParser

Changes History : See File Information for changes

- **Package** SAFOX
- **Sub-Package** SAFOX_P
- **License** GNU
- **Version** 0.5
- **Author** CVH, cpi-service; cvh@cpi-service.com

Constructor *void* function xmlParser::xmlParser() [line 237]

Constructor

void function xmlParser::destroy() [line 246]

Destructor

string function xmlParser::getError() [line 607]

returns the last occurred error or false when all errors have been returned

-- function xmlParser::getErrors() [line 599]

Returns Array of Errors occurred or false

(xmlDoc) function xmlParser::getXmlDoc() [*line 268*]

return the XMLDOC Object if successfully parsed otherwise false

boolean function xmlParser::loadFile(\$filename) [*line 289*]

Function Parameters:

- **\$filename \$filename**
 - string filename to be set

Load a String to be parsed from a xml Document File

boolean function xmlParser::parse() [*line 311*]

Parse the String and create an xmlDoc Object

- **Access public**

void function xmlParser::registerEvent(\$event, \$functionname) [*line 670*]

Function Parameters:

- **\$event \$event**
- **\$functionname \$functionname**

registerEvent - registers a functionname to be executed when event is fired

currently supported events: XPE_ON_PARSE_DEF (0); // Event On Parsing <? xml ..
String XPE_ON_PARSE_DTD (1); // On Parsing DTD XPE_ON_PARSE_TAG (2); // On
Parsing TagName XPE_ON_PARSE_ATT (3); // On Parsing Attribute String
XPE_ON_PARSE_CDT (4); // On Parsing CDATA XPE_ON_PARSE_NOD (5); // On Node
Parsing Completed Object XPE_ON_PARSE_CMT (6); // On Parsing a Comment
XPE_ON_PARSE_XSP (7); // On Parsing XML Specific Tags <?;

User-Defined Eventhandlers will be called with two parameters :

- **&\$data** - relevant information for event (e.g. the unparsed DOCTYPE string in the XPE_ON_PARSE_DTD event)

- `&$node` - reference to Node Object

EventHandlers should be considered for "PreProcessing Data" if Events are handled by XMLPARSER themselves FullProcessing - The Following Events are currently not handled by the XMLPARSER

- XPE_ON_PARSE_DTD (an event is fired providing the DOCTYPE string that otherwise is ignored by the Parser and the reference to the xmlDoc-object and the parser object-ref)
- XPE_ON_PARSE_CMT (an event is fired providing the comment-tag string that otherwise is ignored by the Parser and the reference to the xmlDoc (or Node)-object and the parser object-ref)
- XPE_ON_PARSE_XSP (an event is fired providing the special XML-tag string that otherwise is ignored by the Parser and the reference to the xmlDoc-object and the parser object-ref)

Post-Processing Events :

- XPE_PN_PARSE_NOD (an event is fired providing an empty string (no unparsed stringdata left) and the reference to the node-object and the parser object-ref)

All Other Events are to be used for PreProcessing Purposes and will be provided the relevant string-data, a null var , and the parser reference

- **Access public**

`void function xmlParser::setString($string) [line 256]`

Function Parameters:

- `$string` **\$string**

Set String to be parsed

`function xmlParser::setUida($uida, $node-) [line 281]`

Function Parameters:

- `$uida` **\$node-**
 - set the "unique ID Attribute"-name to which the parser should listen to call `>setId()`

setUida - sets the "unique ID Attribute"-name

that is used for GetNodeById, SetId etc of XMLNode/Doc Objects a uida usually is "id" but sometimes in xml may be something like "recordno", "uid" ...

safox_g_rss.cls.php

The SAFOX package is a collection of light-weight API for object-oriented PHP to handle XML files.

As of version 0.5 the SAFOX wrapper is the starting point for all SAFOX operations. It is an extreme light weight class that allows to create any of the SAFOX objects (XMLDoc, XMLParser, RSSDoc, RSSParser) without further include or require commands. All necessary libraries of the SAFOX API will be loaded only when/if the file is required, thus making scripts lighter and faster.

Providing three base classes, XMLDoc, XMLNode, XMLParser, and extended RSSDoc and RSSParser Classes for RSS 2.0 Handling, the SAFOX package allows an easy and clean way to handle all kind of XML files. The SAFOX package is currently developed in PHP4.X and copyright by Christian Hansel, cpi service, Leipzig. The code is provided under the GNU Public License Version 2 and may freely be used, modified, and distributed with original copyright notices to be maintained.

If no copy of the license is provided check <http://www.gnu.org/licenses/gpl2.txt> The SAFOX Package consists of five files:

- safox.cls.php -- A OOP wrapper for all SAFOX objects which loads one or more of the following files only when required
- safox_g.cls.php -- Provides the XMLDOC & XMLNODE Classes
- safox_p.cls.php -- Provides a simple, light-weight, but strong Parser for XML Documents that returns a XMLDOC Object

The SAFOX_RSS sub package consists of two files but requires the SAFOX Base classes:

- safox_g_rss.cls.php -- Provides the RSSDOC & RSSNODE Classes
- safox_p_rss.cls.php -- Provides a simple, light-weight, but strong Parser for XML Documents that returns a XMLDOC Object

- **Package** SAFOX
- **Sub-Package** SAFOX_RSS
- **Copyright** (c) 2003-2006 CVH, cpi-service
- **File** safox_g_rss.cls.php
- **Date** 2006-02-07
- **Version** 0.1
- **Link** <http://www.cpi-service.com>

- **License** GPL2
- **Author** CVH, cpi-service ; cvh@cpi-service.com

SAFOXVERSIONRSS = 0.1 *[line 41]*
SAFOX_G_RSS = 1 *[line 40]*

safox_p_rss.cls.php

The SAFOX package is a collection of light-weight API for object-oriented PHP to handle XML files.

As of version 0.5 the SAFOX wrapper is the starting point for all SAFOX operations. It is an extreme light weight class that allows to create any of the SAFOX objects (XMLDoc, XMLParser, RSSDoc, RSSParser) without further include or require commands. All necessary libraries of the SAFOX API will be loaded only when/if the file is required, thus making scripts lighter and faster.

Providing three base classes, XMLDoc, XMLNode, XMLParser, and extended RSSDoc and RSSParser Classes for RSS 2.0 Handling, the SAFOX package allows an easy and clean way to handle all kind of XML files. The SAFOX package is currently developed in PHP4.X and copyright by Christian Hansel, cpi service, Leipzig. The code is provided under the GNU Public License Version 2 and may freely be used, modified, and distributed with original copyright notices to be maintained.

If no copy of the license is provided check <http://www.gnu.org/licenses/gpl2.txt>

The SAFOX Package consists of five files:

- safox.cls.php -- A OOP wrapper for all SAFOX objects which loads one or more of the following files only when required
- safox_g.cls.php -- Provides the XMLDOc & XMLNODE Classes
- safox_p.cls.php -- Provides a simple, light-weight, but strong Parser for XML Documents that returns a XMLDOC Object

The SAFOX_RSS sub package consists of two files but requires the SAFOX Base classes:

- safox_g_rss.cls.php -- Provides the RSSDOc & RSSNODE Classes
- safox_p_rss.cls.php -- Provides a simple, light-weight, but strong Parser for XML Documents that returns a XMLDOC Object

full api documentation and examples available at <http://www.cpi-service.com/safox/>

This is the RSS2.0 Parser library for the object-oriented parsing of rss 2.0 files

- **Package** SAFOX
- **Sub-Package** SAFOX_RSS
- **Copyright** (c) 2003-2006 CVH, cpi-service
- **File** safox_p_rss.cls.php

- **Date** 2006-02-07
- **Version** 0.5
- **Link** <http://www.cpi-service.com>
- **License** GPL2
- **Author** CVH, cpi-service ; cvh@cpi-service.com

SAFOX_P_RSS = true *[line 44]*

Class RSSDoc

[line 59]

class RSSDoc

The current version of the SAFOX RSS Generation API allows generation of valid RSS 2.0 code. However, it does not yet support the optional channel properties (sub-elements): cloud, skipHours, skipDays, textInput. Also, the current version does not check for standard compliancy, that is it does not check if CData provided for subelements is valid according to the specifications (e.g. if a required email is valid or an url provided etc) Changes History : See File Information for changes

- **Package** SAFOX
- **Sub-Package** SAFOX_RSS
- **License** GNU
- **Version** 0.1
- **Author** CVH, cpi-service; cvh@cpi-service.com

Constructor function RSSDoc::RSSDoc(\$title, \$link, \$description) *[line 98]*

Function Parameters:

- *title* **\$title** string - required title tag (sub-element of channel)
- *link* **\$link** string(url) - required link tag (sub-element of channel)

- *description* **\$description** string - required description tag (sub-element of channel)

RSSDOC constructor

RSSItem function `RSSDoc::addItem($title, $link, $description)` [*line 391*]

Function Parameters:

- *string* **\$title** title
- *string* **\$link** link (url)
- *string* **\$description** description

The addItem method is used to add items to the channel.

Currently all items are included when called, meaning that items should be added only after all other channel information has been set.

array function `RSSDoc::getAllItems()` [*line 442*]

getAllItems returns a reference to an array containing the references to all *RSSItem* Objects of the current document

XMLNode function `RSSDoc::getChannelObject()` [*line 197*]

getChannelObject returns a reference to the Channel node of the RSS document

RSSItem function `RSSDoc::getFirstItem()` [*line 475*]

getFirstItem returns a reference to the first Item.

RSSItem function `RSSDoc::getLastItem()` [*line 486*]

getLastItem returns a reference to the last Item.

RSSItem function `RSSDoc::getNextItem()` [*line 450*]

getNextItem returns a reference to the current(first) (when not called before) or next Item.

Thus by repeated calls you can iterate through all items

RSSItem function `RSSDoc::getPrevItem()` [*line 466*]

getPrevItem returns a reference to previous Item.

Thus by repeated calls you can iterate through all items

```
void function RSSDoc::printRSS() [line 415]
```

prints RSS Documents to output

```
function RSSDoc::setChannelCategory($strValue) [line 307]
```

Function Parameters:

- *string* **\$strValue** Value

This method allows to set/change the value of the category tag (subelement to channel)

```
function RSSDoc::setChannelCopyright($strValue) [line 258]
```

Function Parameters:

- *string* **\$strValue** Value

This method allows to set/change the value of the copyright tag (subelement to channel)

```
function RSSDoc::setChannelData($arrData) [line 206]
```

setChannelData used to set optional/required sub elements of the channel tag.

However, it does not yet support the optional channel properties (sub-elements): cloud, skipHours, skipDays, textInput Also, the current version does not check for standard compliancy, that is it does not check if CData provided for subelements is valid according to the specifications (e.g. if a required email is valid or an url provided etc)

```
function RSSDoc::setChannelDescription($strValue) [line 234]
```

Function Parameters:

- *string* **\$strValue** Value

This method allows to set/change the value of the Description tag (subelement to channel)

function RSSDoc::setChannelDocs(\$strValue) [*line 266*]

Function Parameters:

- *string* **\$strValue** Value

This method allows to set/change the value of the docs tag (subelement to channel)

function RSSDoc::setChannelGenerator(\$strValue) [*line 315*]

Function Parameters:

- *string* **\$strValue** Value

This method allows to set/change the value of the generator tag (subelement to channel)

function RSSDoc::setChannelImage(\$url, \$link, \$title, \$description, [\$width = ""0], [\$height = 0]) [*line 344*]

Function Parameters:

- *string* **\$url** url of image
- *string* **\$link** url link to link to
- *string* **\$title** title of img
- *string* **\$description** optional description
- *integer* **\$width** optional width information
- *integer* **\$height** optional height information

This method allows to set/change the values of the optional channel image tag (subelement to channel)

function RSSDoc::setChannelLanguage(\$strValue) [*line 250*]

Function Parameters:

- *string* **\$strValue** Value

This method allows to set/change the value of the Language tag (subelement to channel)

```
function RSSDoc::setChannelLastBuildDate($iTimestamp, [$format = "r"]) [line 274]
```

Function Parameters:

- *int* **\$iTimestamp** Timestamp

This method allows to set/change the value of the lastbuilddate tag (subelement to channel)

```
function RSSDoc::setChannelLink($strValue) [line 242]
```

Function Parameters:

- *string* **\$strValue** Value

This method allows to set/change the value of the Link tag (subelement to channel)

```
function RSSDoc::setChannelManagingEditor($strValue) [line 282]
```

Function Parameters:

- *string* **\$strValue** Value

This method allows to set/change the value of the managingEditor tag (subelement to channel)

```
function RSSDoc::setChannelPubDate($iTimestamp, [$format = "r"]) [line 290]
```

Function Parameters:

- *int* **\$iTimestamp** Timestamp

This method allows to set/change the value of the pubDate tag (subelement to channel)

```
function RSSDoc::setChannelRating($strValue) [line 299]
```

Function Parameters:

- *string* **\$strValue** Value

This method allows to set/change the value of the rating tag (subelement to channel)

function RSSDoc::setChannelTitle(\$strValue) [*line 225*]

Function Parameters:

- *string* **\$strValue** Value

setChannelTitle allows to set/change the value of the title tag (subelement to channel)

function RSSDoc::setChannelTTL(\$iValue) [*line 331*]

Function Parameters:

- *integer* **\$iValue** minutes to live (10 minimum)

This method allows to set/change the value of the ttl tag (subelement to channel)

function RSSDoc::setChannelWebMaster(\$strValue) [*line 323*]

Function Parameters:

- *string* **\$strValue** Value

This method allows to set/change the value of the webMaster tag (subelement to channel)

boolean function RSSDoc::takeOverXML(&\$xmlDoc) [*line 123*]

Function Parameters:

- [*xmlDoc*](#) **&\$xmlDoc**
 - an xmlDoc Instance

The takeOver XML is semi-public.

It is not intended for public calls yet needs to be public for the easy and fast parsing by RSSParser.

This method may be used to "takeover" an XMLDoc object which contains RSS 2.0

string function RSSDoc::writeRSS() [*line 400*]

method returns the xml string of the RSS Document

boolean function RSSDoc::writeToFile(\$filepath) [*line 408*]

Function Parameters:

- *string* **\$filepath** fullfilename

Writes the xml string of the RSS Document to a file

Class RSSItem

[*line 512*]

RSSItem is class representing the item nodes of an RSS channel

The current version of the SAFOX RSS Generation API allows generation of valid RSS 2.0 code. However, it does not yet support the optional channel properties (sub-elements): cloud, skipHours, skipDays, textInput. Also, the current version does not check for standard compliancy, that is it does not check if CData provided for subelements is valid according to the specifications (e.g. if a required email is valid or an url provided etc). The RSSItem also does not yet check for validity of supplied information. Instances of the RSSItem class are returned by the getItem-methods of the RSSDoc class (getNextItem(), getLastItem() ...) and may be used to set and edit the item properties/subtags. Changes History : See File Information for changes

- **Package** SAFOX
- **Sub-Package** SAFOX_RSS
- **License** GNU
- **Version** 0.1

- **Author** CVH, cpi-service; cvh@cpi-service.com

RSSItem::\$rssDoc

mixed = null *[line 513]*

RSSItem::\$self

mixed = null *[line 514]*

Constructor function RSSItem::RSSItem(&\$rssDoc, \$title, \$link, \$description) *[line 522]*

Function Parameters:

- [RSSDoc](#) &\$rssDoc Instance of parent RSS document
- *string* \$title title
- *string* \$link link
- *string* \$description description

constructor of the RSSItem Class

string function RSSItem::getAuthor() *[line 663]*

method returns the value of the author tag of the Item

string function RSSItem::getCategory() *[line 673]*

method returns the value of the category tag of the Item

string function RSSItem::getComments() *[line 683]*

method returns the value of the comments tag of the Item

string function RSSItem::getDescription() *[line 653]*

method returns the value of the description tag of the Item

string function RSSItem::getLink() *[line 643]*

method returns the value of the link tag of the Item

timestamp function RSSItem::getPubDate([\$asTimeStamp = true]) *[line 694]*

Function Parameters:

- *boolean* **\$asTimeStamp** returnAsTimeStamp

method returns the value of the pubDate tag of the Item

string function RSSItem::getSource() [*line 709*]

method returns the value of the source tag of the Item

string function RSSItem::getTitle() [*line 633*]

method returns the value of the title tag of the Item

function RSSItem::setAuthor(\$string) [*line 571*]

method sets/changes the optional author tag of the Item

function RSSItem::setCategory(\$string) [*line 603*]

method sets/changes the optional category tag of the Item

function RSSItem::setComments(\$stringURI) [*line 622*]

method sets/changes the optional comments tag of the Item

function RSSItem::setDescription(\$strValue) [*line 553*]

method sets/changes the description tag of the Item

function RSSItem::setEnclosure(\$url, \$bytelength, \$mimetype) [*line 580*]

method sets/changes the optional enclosure tag of the Item

function RSSItem::setGuid(\$string, [\$isPermaLink = false]) [*line 591*]

method sets/changes the optional guid tag of the Item

function RSSItem::setLink(\$strValue) [*line 544*]

method sets/changes the link tag of the Item

function RSSItem::setPubDate(\$timestamp, [\$format = "r"]) [*line 562*]

method sets/changes the optional PubDate tag of the Item

function RSSItem::setSource(\$string, \$url) [*line 612*]

method sets/changes the optional source tag of the Item

function RSSItem::setTitle(\$strValue) [*line 535*]
method sets/changes the title tag of the Item

Class RSSParser

[*line 57*]

class RSSParser

Changes History : See File Information for changes *

- **Package** SAFOX
- **Sub-Package** SAFOX_RSS
- **License** GNU
- **Version** 0.1
- **Author** CVH, cpi-service; cvh@cpi-service.com

Constructor function RSSParser::RSSParser() [*line 77*]

RSS Parser Constructor

- **Access** public

RSSDoc function RSSParser::getRSSDoc() [*line 180*]

getRSSDoc returns a reference to the parsed RSS Document object

boolean function RSSParser::loadFile(\$file) [*line 102*]

Function Parameters:

- *string* **\$file** rss file location

loadFile return true if remote/local rss-file has been loaded (was found)

- **Access** public

boolean function RSSParser::parse() [*line 125*]

parse method alias to **parseRSS** which makes necessary calls to the SAFOX Parser and returns boolean success message

boolean function RSSParser::parseRSS() [*line 132*]

parseRSS makes necessary calls to the SAFOX Parser and returns boolean success message

boolean function RSSParser::setString(\$string) [*line 112*]

Function Parameters:

- *string* **\$string** rss document

setString may be used to set a rss-file manually without loading a file.
the parameter represent the full xml-string document of the Rss file

Appendices

Appendix A - Class Trees

Package SAFOX

RSSDoc

- [RSSDoc](#)

RSSItem

- [RSSItem](#)

RSSParser

- [RSSParser](#)

SAFOX

- [SAFOX](#)

xmlDoc

- [xmlDoc](#)

xmlNode

- [xmlNode](#)

xmlParser

- [xmlParser](#)

Index

C

constructor RSSItem::RSSItem()	37
<i>constructor of the RSSItem Class</i>	
constructor RSSParser::RSSParser()	39
<i>RSS Parser Constructor</i>	
constructor RSSDoc::RSSDoc()	30
<i>RSSDOC constructor</i>	
constructor xmlParser::xmlParser()	23
<i>Constructor</i>	
constructor xmlDoc::xmlDoc()	9
constructor xmlNode::xmlNode()	14
<i>Constructor</i>	
constructor SAFOX::SAFOX()	5
<i>SAFOX constructor</i>	

O

onParseCDATA()	22
<i>OnParseCDATA - is registered in Class XMLPARSER as a PreProcessing EventHandler for CDATA-Strings called when the XPE_ON_PARSE_CDT event is fired</i>	

R

RSSItem::getSource()	38
<i>method returns the value of the source tag of the Item</i>	
RSSItem::getPubDate()	37
<i>method returns the value of the pubDate tag of the Item</i>	
RSSItem::getTitle()	38
<i>method returns the value of the title tag of the Item</i>	
RSSItem::setAuthor()	38
<i>method sets/changes the optional author tag of the Item</i>	
RSSItem::setCategory()	38
<i>method sets/changes the optional category tag of the Item</i>	
RSSItem::getLink()	37
<i>method returns the value of the link tag of the Item</i>	
RSSItem::getDescription()	37
<i>method returns the value of the description tag of the Item</i>	
RSSItem::\$self	37
RSSItem::\$rssDoc	37
RSSItem::getAuthor()	37
<i>method returns the value of the author tag of the Item</i>	
RSSItem::getCategory()	37
<i>method returns the value of the category tag of the Item</i>	

RSSItem::getComments()	37
<i>method returns the value of the comments tag of the Item</i>	
RSSItem::setComments()	38
<i>method sets/changes the optional comments tag of the Item</i>	
RSSItem::setDescription()	38
<i>method sets/changes the description tag of the Item</i>	
RSSParser::loadFile()	39
<i>loadFile return true if remote/local rss-file has been loaded (was found)</i>	
RSSParser::getRSSDoc()	39
<i>getRSSDoc returns a reference to the parsed RSS Document object</i>	
RSSParser::parse()	40
<i>parse method alias to parseRSS which makes necessary calls to the SAFOX Parser and returns boolean success message</i>	
RSSParser::parseRSS()	40
<i>parseRSS makes necessary calls to the SAFOX Parser and returns boolean success message</i>	
RSSParser::setString()	40
<i>setString may be used to set a rss-file manually without loading a file.</i>	
RSSParser	39
<i>class RSSParser</i>	
RSSItem::setTitle()	39
<i>method sets/changes the title tag of the Item</i>	
RSSItem::setGuid()	38
<i>method sets/changes the optional guid tag of the Item</i>	
RSSItem::setEnclosure()	38
<i>method sets/changes the optional enclosure tag of the Item</i>	
RSSItem::setLink()	38
<i>method sets/changes the link tag of the Item</i>	
RSSItem::setPubDate()	38
<i>method sets/changes the optional PubDate tag of the Item</i>	
RSSItem::setSource()	38
<i>method sets/changes the optional source tag of the Item</i>	
RSSItem	36
<i>RSSItem is class representing the item nodes of an RSS channel</i>	
RSSDoc::writeToFile()	36
<i>Writes the xml string of the RSS Document to a file</i>	
RSSDoc::setChannelCategory()	32
<i>This method allows to set/change the value of the category tag (subelement to channel)</i>	
RSSDoc::printRSS()	32
<i>prints RSS Documents to output</i>	
RSSDoc::setChannelCopyright()	32
<i>This method allows to set/change the value of the copyright tag (subelement to channel)</i>	
RSSDoc::setChannelData()	32
<i>setChannelData used to set optional/required sub elements of the channel tag.</i>	
RSSDoc::setChannelDescription()	32
<i>This method allows to set/change the value of the Description tag (subelement to channel)</i>	
RSSDoc::getPrevItem()	31
<i>getPrevItem returns a reference to previous Item.</i>	
RSSDoc::getNextItem()	31
<i>getNextItem returns a reference to the current(first) (when not called before) or next Item.</i>	
RSSDoc::getAllItems()	31
<i>getAllItems returns a reference to an array containing the references to all RSSItem Objects of the current document</i>	
RSSDoc::addItem()	31

The `addItem` method is used to add items to the channel.

RSSDoc::getChannelObject()	31
<i>getChannelObject</i> returns a reference to the Channel node of the RSS document	
RSSDoc::getFirstItem()	31
<i>getFirstItem</i> returns a reference to the first Item.	
RSSDoc::getLastItem()	31
<i>getLastItem</i> returns a reference to the last Item.	
RSSDoc::setChannelDocs()	33
<i>This method allows to set/change the value of the docs tag (subelement to channel)</i>	
RSSDoc::setChannelGenerator()	33
<i>This method allows to set/change the value of the generator tag (subelement to channel)</i>	
RSSDoc::setChannelTTL()	35
<i>This method allows to set/change the value of the ttl tag (subelement to channel)</i>	
RSSDoc::setChannelTitle()	35
<i>setChannelTitle</i> allows to set/change the value of the title tag (subelement to channel)	
RSSDoc::setChannelWebMaster()	35
<i>This method allows to set/change the value of the webMaster tag (subelement to channel)</i>	
RSSDoc::takeOverXML()	35
<i>The takeOver XML is semi-public.</i>	
RSSDoc::writeRSS()	36
<i>method returns the xml string of the RSS Document</i>	
RSSDoc::setChannelRating()	34
<i>This method allows to set/change the value of the rating tag (subelement to channel)</i>	
RSSDoc::setChannelPubDate()	34
<i>This method allows to set/change the value of the pubDate tag (subelement to channel)</i>	
RSSDoc::setChannelLanguage()	33
<i>This method allows to set/change the value of the Language tag (subelement to channel)</i>	
RSSDoc::setChannelImage()	33
<i>This method allows to set/change the values of the optional channel image tag (subelement to channel)</i>	
RSSDoc::setChannelLastBuildDate()	34
<i>This method allows to set/change the value of the lastbuilddate tag (subelement to channel)</i>	
RSSDoc::setChannelLink()	34
<i>This method allows to set/change the value of the Link tag (subelement to channel)</i>	
RSSDoc::setChannelManagingEditor()	34
<i>This method allows to set/change the value of the managingEditor tag (subelement to channel)</i>	
RSSDoc	30
<i>class RSSDoc</i>	

S

SAFOX_P	22
<i>Event identifiers & constants</i>	
safox_p.cls.php	20
<i>The SAFOX package is a collection of light-weight API for object-oriented PHP to handle XML files.</i>	
SAFOX_G	8
safox_g_rss.cls.php	27
<i>The SAFOX package is a collection of light-weight API for object-oriented PHP to handle XML files.</i>	
SAFOXVERSIONRSS	28
SAFOX_P_RSS	30

safox_p_rss.cls.php	29
<i>The SAFOX package is a collection of light-weight API for object-oriented PHP to handle XML files.</i>	
SAFOX_G_RSS	28
SAFOXDOC	8
safox_g.cls.php	7
<i>The SAFOX package is a collection of light-weight API for object-oriented PHP to handle XML files.</i>	
SAFOX::createRSSDoc()	5
<i>creates a RSSDoc (RSS 2.0) Generation object and returns a reference to this. automatically loads necessary files</i>	
SAFOX	5
<i>SAFOX a wrapper class for the SAFOX API</i>	
SAFOXVERSION	4
SAFOX::createRSSParser()	5
<i>creates a RSSParser (RSS 2.0) object and returns a reference to this. automatically loads necessary files</i>	
SAFOX::createXMLDoc()	5
<i>creates a XMLDoc Generation object and returns a reference to this. automatically loads necessary files</i>	
SAFOX::ver()	6
<i>alias to getVersion() - returns the current version of the SAFOX API</i>	
SAFOX::getVersion()	6
<i>returns the current version of the SAFOX API</i>	
SAFOX::createXMLParser()	6
<i>creates a XMLParser object and returns a reference to this. automatically loads necessary files</i>	
safox.cls.php	2
<i>This is the SAFOX API Project Wrapper. The SAFOX package is a collection of light-weight API for object-oriented PHP to handle XML files.</i>	

X

xmlNode::setType()	19
<i>setType Sets the type of a xml node/tag</i>	
xmlNode::setParent()	19
<i>setParent - Sets the ParentNode (usually used internally or if moved)</i>	
xmlNode::setId()	19
<i>setId sets the Unique ID Attribute defined by xmlDoc->setUida or "ID"</i>	
xmlNode::writeXML()	19
<i>returns a string containing the wellformed xml of current Node/tag</i>	
XML_MAX_LINE_SIZE	22
XPE_ON_PARSE_CDT	22
XPE_ON_PARSE_ATT	22
xmlNode::setEntityConversion()	18
<i>setEntityConversion is a function that specifies whether or not special characters in CDATA should be converted into HT/XML-Entities (e.g.</i>	
xmlNode::setCDATA()	18
<i>setCDATA - sets the CDATA content of a node</i>	
xmlNode::getType()	17
<i>getType returns the type/TagName of a xml node</i>	
xmlNode::getParent()	17

<i>getParent</i> - Returns a reference to the Parent NodeObject or null	
xmlNode::hasChildren()	17
<i>hasChildren</i> - Returns true if childNodes exist or false	
xmlNode::hasParent()	17
<i>hasParent</i> - Returns true if a reference to a parent node exists	
xmlNode::setAttribute()	18
<i>setAttribute</i> sets an Attribute	
xmlNode::remove()	17
<i>Alias to destroy</i> - Deletes a node and all childnodes, if the optional parameter <i>clean</i> is set true (default), a call to the <i>cleanUp</i> Method of the XMLDOC Object is made automatically.	
XPE ON PARSE CMT	22
XPE ON PARSE DEF	22
xmlParser::loadFile()	24
<i>Load a String to be parsed from a xml Document File</i>	
xmlParser::getXmlDoc()	24
<i>return the XMLDOC Object if succesfully parsed otherwise false</i>	
xmlParser::parse()	24
<i>Parse the String and create an xmlDoc Object</i>	
xmlParser::registerEvent()	24
<i>registerEvent</i> - registers a functionname to be exucted when event is fired	
xmlParser::setUida()	25
<i>setUida</i> - sets the "unique ID Attribute"-name	
xmlParser::setString()	25
<i>Set String to be parsed</i>	
xmlParser::getErrors()	23
<i>Returns Array of Errors occured or false</i>	
xmlParser::getError()	23
<i>returns the last occured error or false when all errors have been returned</i>	
XPE ON PARSE NOD	22
XPE ON PARSE DTD	22
XPE ON PARSE TAG	22
XPE ON PARSE XSP	22
xmlParser::destroy()	23
<i>Destructor</i>	
xmlParser	23
<i>class xmlParser</i>	
xmlNode::getNextNode()	17
<i>getNextNode</i> - returns a reference to the Next ChildNode of current Node	
xmlNode::getId()	17
<i>getId</i> -- return the Unique ID attribute value	
xmlDoc::getNodeById()	11
<i>Return a xmlNodeObject or False</i>	
xmlDoc::getNextNode()	11
<i>Returns next Reference to Child Node or False;</i>	
xmlDoc::setEncoding()	11
<i>Set the XML-Document Encoding Attribute</i>	
xmlDoc::setEntityConversion()	12
<i>setEntityConversion</i> is a function that specifies whether or not special characters in CDATA should be converted into HT/XML-Entities (e.g.	
xmlDoc::setUida()	12
<i>Set the "Unique ID Attribute"-name (usually ID)</i>	
xmlDoc::setStandAlone()	12

	<i>Sets the XML Document StandAlone Attribute</i>	
xmlDoc::getId()	<i>return Id (Constant SAFOXDOC) This function returns the mixed returns the UIDA (Unique ID Attribute) Value of current node, for the XMLDOC returns SAFOXDOC (constant)</i>	11
xmlDoc::getChildNodeByTagName()	<i>getChildNodeByTagName - direct reference to xmlNode Object if only one tag found, array of xmlNode Object references if more found, or boolean false -- suggested by Cristiano</i>	11
xmlDoc::addNode()	<i>addNode Adds a Node to the Main part of the xmlDoc;</i>	9
xmlDoc::\$EntityConversion	<i>Constructor</i>	9
xmlDoc::addNodeAfter()	<i>Adds a new Node to the Document directly *after* a specified node, whereas addNode() method adds a node to the end of the Document</i>	9
xmlDoc::addNodeBefore()	<i>Adds a new Node to the Document directly *before* a specified node, whereas addNode() method adds a node to the end of the Document</i>	10
xmlDoc::destroy()	<i>Destructor- empties internal data</i>	10
xmlDoc::cleanUp()	<i>Cleans up internal data arrays from removed nodes and rebuilds the ID Referencing Array</i>	10
xmlDoc::setVersion()	<i>Sets the XML-Document Version-Attribute</i>	12
xmlDoc::writeToFile()	<i>Writing an XML Document to File</i>	13
xmlNode::destroy()	<i>Deletes a node and all childnodes, if the optional parameter clean is set true (default), a call to the cleanUp Method of the XMLDOC Object is made automatically.</i>	16
xmlNode::delete()	<i>Alias to destroy - Deletes a node and all childnodes, if the optional parameter clean is set true (default), a call to the cleanUp Method of the XMLDOC Object is made automatically.</i>	15
xmlNode::getAttribute()	<i>getAttribute returns the value of an attribute</i>	16
xmlNode::getCdata()	<i>setCdata - returns the CData content of a node</i>	16
xmlNode::getChildNodeByType()	<i>alias to getChildNodeByTagName - returns a reference to specific ChildNode of current Node -- suggested by Cristiano</i>	17
xmlNode::getChildNodeByTagName()	<i>getChildNodeByTagName - returns a reference to specific ChildNode of current Node -- suggested by Cristiano</i>	16
xmlNode::convertString2Entities()	<i>convertString2Entities - returns a string that represents a well formed xml-string to be used as CDATA based on setEntityConversion() converts the parameter into into HT/XML-Entities or encapsulates in <![CDATA[]]></i>	15
xmlNode::cleanUp()	<i>Cleans up internal data arrays from removed nodes and rebuilds the ID Referencing Array</i>	15

xmlNode	class xmlNode	13
xmlDoc::writeXML()	returns a string of the XML Document	13
xmlNode::\$EntityConversion	EntityConversion - flag used to determine whether or not special characters in CDATA should be converted into HT/XML-Entities (e.g.	14
xmlNode::addNode()	addNode - Adds a child node to the current Object (Node) and returns a reference to childNode	14
xmlNode::addNodeBefore()	Adds a new Node to the Node directly <i>*before*</i> a specified ChildNode, whereas addNode() method adds a node to the end of the Node	15
xmlNode::addNodeAfter()	Adds a new Node to the Node directly <i>*after*</i> a specified ChildNode, whereas addNode() method adds a node to the end of the Node	14
xmlDoc	class xmlDoc	9