OPL Open eBook Lib Documentation

Contents

ebook.php

- **Package**
- **Throws** no exceptions thrown

require_once **'ebookData.php'** *[line 14]*

require_once **'ebookWrite.php'** *[line 13]*

require_once **'ebookRead.php'** *[line 12]*

****
Software: eBook Libiary * Version: 0.5 Alpha * Date: 2008-07-21 * Author: Jacob Weigand of RIT's Open Publishing Lab * License: GNU General Public License (GPL) *
* You may use, modify and redistribute this software as you wish. *
****

ebookData.php

- **Package**
- **Throws** no exceptions thrown

ebookRead.php

- **Package**
- **Throws** no exceptions thrown

require_once **'ebookData.php'**[*line 12]*

****
Software: eBook Reader * Version: 0.5 Alpha * Date: 2008-07-21 * Author: Jacob Weigand of RIT's Open Publishing Lab * License: GNU General Public License (GPL) * * You may use, modify and redistribute this software as you wish. *
****

ebookWrite.php

- **Package**
- **Throws** no exceptions thrown

require_once **'ebookData.php'** *[line 12]*

****
Software: eBook Writter * Version: 0.5 Alpha * Date: 2008-07-21 * Author: Jacob Weigand of RIT's Open Publishing Lab * License: GNU General Public License (GPL) * * You may use, modify and redistribute this software as you wish. *
****

Class ebook

[line 21]

This is the primary class intented for general use. This class uses the open ebook reader and the open ebook writter classes. If you wish to only read or write you can use the single libary insted of this combined one. This class is only intended to make it easier to see and use the reader and writter, There should only be calls to functions from reader or writter here.

- **Package**
- **Throws** no exceptions thrown

Constructor *void* function ebook::ebook([$epubFile = null]) *[line 30]*

Function Parameters:

- *$ebookData* **$epubFile** epubFile Optional. Use this if you are editing an existing epub file.

Constructor

- **Throws** no exceptions thrown
- public

Returns function ebook::addContentFile($fileLoc, $name, $mime, [$subDir = null], [$fallBackId = null]) *[line 552]*
Function Parameters:

- *$fileLoc* **$fileLoc** Where is the file currently located.
- *$subDir* **$name** If you want the file in a subdirectory of the content folder.
- *$name* **$mime** Short discriptive name of what the content item is. eg: Chapter 3.
- *$mime* **$subDir** the mime type of the file being added.
- *$fallBackId* **$fallBackId** the id of an item you wish to use as a fallback item.

This is used for adding content files to an epub. Make sure you have write permission to the destination directory.

- **Throws** no exceptions thrown
- public

void function ebook::buildEPUB($name, $dest) *[line 46]*
Function Parameters:

- *$name* **$name** The name you want the epub file to have. Don't include a file extenshion.
- *$dest* **$dest** The location you want the final epub file.

Puts together the eBook and places it in the desired directory.

- **Throws** no exceptions thrown

- public

a function ebook::getContentFile($location) *[line 645]*

Function Parameters:

- *$location* **$location** Where inside of the epub is the file you want

Will give you the requested file out of the epub

Note: When finding files based on the manifest's href attribute, the file locations are relative to where the opf file is all ready located in the file structure. So if the opf file is at /ops/contnet.opf and your image is at /ops/img/wolf.jpg then your manifest href for that image will read /img/wolf.jpg. Use basename(getFilePath("opf")); for the directories leading to the OPF file.

- **Throws** no exceptions thrown
- public

the function ebook::getContentLoc() *[line 519]*

Gets the location of all the content inside of the epub.

- **Throws** no exceptions thrown
- public

returns function ebook::getDcContributor([$optional = null]) *[line 179]*

Function Parameters:

- *$optional* **$optional** optional attributes are, Role, XmlLang, OpfFileAs.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

returns function ebook::getDcCoverage([$optional = null]) *[line 459]*
Function Parameters:

- *$optional* **$optional** optional attributes are, XmlLang.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

returns function ebook::getDcCreator([$optional = null]) *[line 148]*
Function Parameters:

- *$optional* **$optional** optional attributes are, Role, XmlLang, OpfFileAs.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

returns function ebook::getDcDate([$optional = null]) *[line 303]*
Function Parameters:

- *$optional* **$optional** optional attributes are, Event, XsiType.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

returns function ebook::getDcDescription([$optional = null]) *[line 272]*
Function Parameters:

- *$optional* **$optional** optional attributes are, XmlLang.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

returns function ebook::getDcFormat([$optional = null]) *[line 366]*
Function Parameters:

- *$optional* **$optional** optional attributes are, XsiType.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

returns function ebook::getDcIdentifier([$optional = null]) *[line 116]*
Function Parameters:

- *$optional* **$optional** optional attributes are, Scheme, XsiType, Id.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

returns function ebook::getDcLanguage([$optional = null]) *[line 85]*
Function Parameters:

- *$optional* **$optional** optional attributes are, XsiType.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

returns function ebook::getDcPublisher([$optional = null]) *[line 210]*
Function Parameters:

- *$optional* **$optional** optional attributes are, XmlLang.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

returns function ebook::getDcRelation([$optional = null]) *[line 428]*

Function Parameters:

- *$optional* **$optional** optional attributes are, XmlLang.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

returns function ebook::getDcRights([$optional = null]) *[line 490]*

Function Parameters:

- *$optional* **$optional** optional attributes are, XmlLang.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

returns function ebook::getDcSource([$optional = null]) *[line 397]*

Function Parameters:

- *$optional* **$optional** optional attributes are, XmlLang.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

returns function ebook::getDcSubject([$optional = null]) *[line 241]*

Function Parameters:

- *$optional* **$optional** optional attributes are, XmlLang.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

returns function ebook::getDcTitle([$optional = null]) *[line 56]*

Function Parameters:

- *$optional* **$optional** optional attributes are, XmlLang.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

returns function ebook::getDcType([$optional = null]) *[line 335]*

Function Parameters:

- *$optional* **$optional** optional attributes are, XsiType.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

Sends function ebook::getEBookDataObject() *[line 653]*

Get the eBook data object. Holds all the data from the epub.

- **Throws** no exceptions thrown
- public

Null function ebook::getFilePath($ext) *[line 632]*

Function Parameters:

- *$ext* **$ext** The file exenshion of the file your looking for.

Get a file path by extenshion

- **Throws** no exceptions thrown
- public

a function ebook::getGuide($item) *[line 605]*

Function Parameters:

- *$item* **$item** What attribute do you want from the guide (title, href, or type).

Gets the guide item desired

- **Throws** no exceptions thrown
- public

a function ebook::getGuideItem($index, $item) *[line 596]*
Function Parameters:

- *$index* **$index** The index number of the guide item.
- *$item* **$item** What attribute do you want from the guide (title, href, or type).

Gets the guide item desired

- **Throws** no exceptions thrown
- public

a function ebook::getManifest($item) *[line 538]*
Function Parameters:

- *$item* **$item** What attribute do you want from the manifest (id, href, or media-type).

Gets the manifest item desired

- **Throws** no exceptions thrown
- public

a function ebook::getManifestItem($index, $item) *[line 529]*
Function Parameters:

- *$index* **$index** The index number of the manifest item.
- *$item* **$item** What attribute do you want from the manifest (id, href, or media-type).

Gets the manifest item desired

- **Throws** no exceptions thrown
- public

a function ebook::getSpine() *[line 569]*

Gets a desired spine item

- **Throws** no exceptions thrown
- public

a function ebook::getSpineItem($index) *[line 561]*

Function Parameters:

- *$index* **$index** The index of the spine item to return.

Gets a desired spine item

- **Throws** no exceptions thrown
- public

void function ebook::setDcContributor($contributor, [$index = null]) *[line 188]*

Function Parameters:

- *$contributor* **$contributor** The contributor to be set. If you want set more then one contributor send an array into this parameter.
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the contributor. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

true function ebook::setDcContributorAttrib($attrib, $attribValue, [$index = null]) *[line 200]*

Function Parameters:

- *$attrib* **$attrib** The name of the optional attributes to be set. Optional attributes are, Role, XmlLang, OpfFileAs.
- *$attribValue* **$attribValue** The value to be set for the optional attribute. If you want set more then one contributor attribute send an array into this parameter. Make sure the attributes corresponde the appropriate contributor index
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional contributor attributes.

- **Throws** no exceptions thrown
- public

void function ebook::setDcCoverage($coverage, [$index = null]) *[line 468]*

Function Parameters:

- *$coverage* **$coverage** The coverage to be set. If you want set more then one coverage send an array into this parameter.
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the coverage. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

true function ebook::setDcCoverageAttrib($attrib, $attribValue, [$index = null]) *[line 480]*

Function Parameters:

- *$attrib* **$attrib** The name of the optional attributes to be set. Optional attributes are XmlLang.
- *$attribValue* **$attribValue** The value to be set for the optional attribute. If you want set more then one coverage attribute send an array into this parameter. Make sure the attributes corresponde the appropriate coverage index
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Set the optional coverage attributes.

- **Throws** no exceptions thrown
- public

void function ebook::setDcCreator($creator, [$index = null]) *[line 157]*

Function Parameters:

- *$creator* **$creator** The creator to be set. If you want set more then one creator send an array into this parameter.
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the creator. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

true function ebook::setDcCreatorAttrib($attrib, $attribValue, [$index = null]) *[line 169]*

Function Parameters:

- *$attrib* **$attrib** The name of the optional attributes to be set. optional attributes are, Role, XmlLang, OpfFileAs.
- *$attribValue* **$attribValue** The value to be set for the optional attribute. If you want set more then one creator attribute send an array into this parameter. Make sure the attributes corresponde the appropriate creator index
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional creator attributes.

- **Throws** no exceptions thrown
- public

void function ebook::setDcDate($eBookdate, [$index = null]) *[line 312]*

Function Parameters:

- *$date* **$eBookdate** The date to be set. If you want set more then one date send an array into this parameter.
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the date. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

true function ebook::setDcDateAttrib($attrib, $attribValue, [$index = null]) *[line 325]*

Function Parameters:

- *$attrib* **$attrib** The name of the optional attributes to be set. Optional attributes are, Event, XsiType.

- *$attribValue* **$attribValue** The value to be set for the optional attribute. If you want set more then one date attribute send an array into this parameter. Make sure the attributes corresponde the appropriate date index
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional date attributes.

- **Throws** no exceptions thrown
- public

void function ebook::setDcDescription($description, [$index = null]) *[line 281]*

Function Parameters:

- *$description* **$description** The description to be set. If you want set more then one description send an array into this parameter.
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the description. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

true function ebook::setDcDescriptionAttrib($attrib, $attribValue, [$index = null]) *[line 293]*

Function Parameters:

- *$attrib* **$attrib** The name of the optional attributes to be set. Optional attributes are, XmlLang.
- *$attribValue* **$attribValue** The value to be set for the optional attribute. If you want set more then one description attribute send an array into this parameter. Make sure the attributes corresponde the appropriate description index
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional description attributes.

- **Throws** no exceptions thrown
- public

void function ebook::setDcFormat($format, [$index = null]) *[line 375]*
Function Parameters:

- *$format* **$format** The format to be set. If you want set more then one format send an array into this parameter.
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the format. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

true function ebook::setDcFormatAttrib($attrib, $attribValue, [$index = null]) *[line 387]*
Function Parameters:

- *$attrib* **$attrib** The name of the optional attributes to be set. Optional attributes are XsiType.
- *$attribValue* **$attribValue** The value to be set for the optional attribute. If you want set more then one format attribute send an array into this parameter. Make sure the attributes corresponde the appropriate format index
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional format attributes.

- **Throws** no exceptions thrown
- public

void function ebook::setDcIdentifier($identifier, [$index = null]) *[line 125]*

Function Parameters:

- *$identifier* **$identifier** the Identifier to be set. If you want set more then one Identifier send an array into this parameter.
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the Identifier. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

true function ebook::setDcIdentifierAttrib($attrib, $attribValue, [$index = null]) *[line 138]*

Function Parameters:

- *$attrib* **$attrib** Id is a required attribute. The name of the optional attributes to be set. Optional attributes are Scheme, XsiType.
- *$attribValue* **$attribValue** The value to be set for the optional attribute. If you want set more then one identifier attribute send an array into this parameter. Make sure the attributes corresponde the appropriate identifier index
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional identifier attributes.

- **Throws** no exceptions thrown
- public

void function ebook::setDcLanguage($language, [$index = null]) *[line 94]*

Function Parameters:

- *$language* **$language** The language to be set. If you want set more then one language send an array into this parameter.
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the language. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

true function ebook::setDcLanguageAttrib($attrib, $attribValue, [$index = null]) *[line 106]*
Function Parameters:

- *$attrib* **$attrib** The name of the optional attributes to be set. Optional attributes are XsiType.
- *$attribValue* **$attribValue** The value to be set for the optional attribute. If you want set more then one language attribute send an array into this parameter. Make sure the attributes corresponde the appropriate language index
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional language attributes.

- **Throws** no exceptions thrown
- public

void function ebook::setDcPublisher($publisher, [$index = null]) *[line 219]*
Function Parameters:

- *$publisher* **$publisher** The publisher to be set. If you want set more then one publisher send an array into this parameter.
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the publisher. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

true function ebook::setDcPublisherAttrib($attrib, $attribValue, [$index = null]) *[line 231]*

Function Parameters:

- *$attrib* **$attrib** The name of the optional attributes to be set. Optional attributes are, XmlLang.
- *$attribValue* **$attribValue** The value to be set for the optional attribute. If you want set more then one publisher attribute send an array into this parameter. Make sure the attributes corresponde the appropriate publisher index
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional publisher attributes.

- **Throws** no exceptions thrown
- public

void function ebook::setDcRelation($relation, [$index = null]) *[line 437]*

Function Parameters:

- *$relation* **$relation** The relation to be set. If you want set more then one relation send an array into this parameter.
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the relation. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

true function ebook::setDcRelationAttrib($attrib, $attribValue, [$index = null]) *[line 449]*

Function Parameters:

- *$attrib* **$attrib** The name of the optional attributes to be set. Optional attributes are XmlLang.
- *$attribValue* **$attribValue** The value to be set for the optional attribute. If you want set more then one relation attribute send an array into this parameter. Make sure the attributes corresponde the appropriate relation index
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Set the optional relation attributes.

- **Throws** no exceptions thrown
- public

void function ebook::setDcRights($rights, [$index = null]) *[line 499]*

Function Parameters:

- *$rights* **$rights** The rights to be set. If you want set more then one rights send an array into this parameter.
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the rights. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

true function ebook::setDcRightsAttrib($attrib, $attribValue, [$index = null]) *[line 511]*

Function Parameters:

- *$attrib* **$attrib** The name of the optional attributes to be set. Optional attributes are XmlLang.
- *$attribValue* **$attribValue** The value to be set for the optional attribute. If you want set more then one rights attribute send an array into this parameter. Make sure the attributes corresponde the appropriate rights index
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Set the optional rights attributes.

- **Throws** no exceptions thrown
- public

void function ebook::setDcSource($source, [$index = null]) *[line 406]*

Function Parameters:

- *$source* **$source** The source to be set. If you want set more then one source send an array into this parameter.
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the source. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

true function ebook::setDcSourceAttrib($attrib, $attribValue, [$index = null]) *[line 418]*

Function Parameters:

- *$attrib* **$attrib** The name of the optional attributes to be set. Optional attributes are XmlLang.
- *$attribValue* **$attribValue** The value to be set for the optional attribute. If you want set more then one source attribute send an array into this parameter. Make sure the attributes

corresponde the appropriate source index

- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Set the optional source attributes.

- **Throws** no exceptions thrown
- public

void function ebook::setDcSubject($subject, [$index = null]) *[line 250]*

Function Parameters:

- *$subject* **$subject** The subject to be set. If you want set more then one subject send an array into this parameter.
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the subject. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

true function ebook::setDcSubjectAttrib($attrib, $attribValue, [$index = null]) *[line 262]*

Function Parameters:

- *$attrib* **$attrib** The name of the optional attributes to be set. Optional attributes are, XmlLang.
- *$attribValue* **$attribValue** The value to be set for the optional attribute. If you want set more then one subject attribute send an array into this parameter. Make sure the attributes corresponde the appropriate subject index
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional subject attributes.

- **Throws** no exceptions thrown
- public

void function ebook::setDcTitle($title, [$index = null]) *[line 65]*

Function Parameters:

- *$title* **$title** The title to be set. If you want set more then one send an array.
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the title

- **Throws** no exceptions thrown
- public

true function ebook::setDcTitleAttrib($attrib, $attribValue, [$index = null]) *[line 75]*

Function Parameters:

- *$attrib* **$attrib** The name of the optional attributes to be set. Optional attributes are XmlLang.
- *$attribValue* **$attribValue** The value to be set for the optional attribute.
- **$index**

Set the optional title attributes.

- **Throws** no exceptions thrown
- public

void function ebook::setDcType($type, [$index = null]) *[line 344]*

Function Parameters:

- *$type* **$type** The type to be set. If you want set more then one type send an array into this parameter.
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the type. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

true function ebook::setDcTypeAttrib($attrib, $attribValue, [$index = null]) *[line 356]*
Function Parameters:

- *$attrib* **$attrib** The name of the optional attributes to be set. Optional attributes are XsiType.
- *$attribValue* **$attribValue** The value to be set for the optional attribute. If you want set more then one type attribute send an array into this parameter. Make sure the attributes corresponde the appropriate type index
- *$index* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional type attributes.

- **Throws** no exceptions thrown
- public

void function ebook::setGuide($title, $type, $href, [$index = null]) *[line 621]*
Function Parameters:

- *$title* **$title** Name of the guide item.
- *$type* **$type** The required type attribute describes the publication component referenced by the href attribute. Type should be of this list cover, title-page, toc "table of contents", index "back-of-book style index", glossary, acknowledgements, bibliography,

colophon, copyright-page, dedication, epigraph, foreword, loi "list of illustrations", lot "list of tables", notes, and preface.

- *$href* **$href** location of the guide item.
- *$index* **$index** The index number of the guide item to be set.

sets the guide item desired. Arrays can be sent into the attributes to set more then one guide item at a time. The guide element identifies fundamental structural components of the publication, to enable Reading Systems to provide convenient access to them.

- **Throws** no exceptions thrown
- public

void function ebook::setSpine($spineId) *[line 577]*

Function Parameters:

- *$spineId* **$spineId** Send an array of ID names as they appear in the manifest in the order you want them read.

Sets a desired spine order. The spine is used to determine the reading order.

- **Throws** no exceptions thrown
- public

void function ebook::setSpineToc($toc) *[line 586]*

Function Parameters:

- *$toc* **$toc** The manifest item that is the table of contents.

Used to tell the spine what manifest item is the Table Of Contents. This does not need to be set, this is optional.

- **Throws** no exceptions thrown
- public

Class ebookData

[line 11]

****
Software: eBook Data Model * Version: 0.5 Alpha * Date: 2008-07-21 * Author: Jacob Weigand of RIT's Open Publishing Lab * License: GNU General Public License (GPL) * * You may use, modify and redistribute this software as you wish. *
****

- **Package**
- **Throws** no exceptions thrown

ebookData::$contentFolder

mixed = [line 14]

- **Access** public

ebookData::$contributor

mixed = [line 43]

- **Access** public

ebookData::$contributorOpfFileAs

mixed = [line 46]

- **Access** public

ebookData::$contributorRole

mixed = [line 44]

- **Access** public

ebookData::$contributorXmlLang

mixed = [line 45]

- **Access** public

ebookData::$coverage

mixed = [line 64]

- **Access** public

ebookData::$coverageXmlLang

mixed = [line 65]

- **Access** public

ebookData::$creator

mixed = [line 39]

- **Access** public

ebookData::$creatorOpfFileAs

mixed = [line 42]

- **Access** public

ebookData::$creatorRole

mixed = [line 40]

- **Access** public

ebookData::$creatorXmlLang

mixed = [line 41]

- **Access** public

ebookData::$css

mixed = [line 18]

- **Access** public

ebookData::$description

mixed = [line 51]

- **Access** public

ebookData::$descriptionXmlLang

mixed = [line 52]

- **Access** public

ebookData::$eBookdate

mixed = [line 53]

- **Access** public

ebookData::$eBookdateEvent

mixed = [line 54]

- **Access** public

ebookData::$eBookdateXsiType

mixed = [line 55]

- **Access** public

ebookData::$epub

mixed = [line 12]

- **Access** public

ebookData::$format

mixed = [line 58]

- **Access** public

ebookData::$formatXsiType

mixed = [line 59]

- **Access** public

ebookData::$guide

mixed = [line 27]

- **Access** public

ebookData::$guideData

mixed = [line 28]

- **Access** public

ebookData::$identifier

mixed = [line 34]

- **Access** public

ebookData::$identifierId

mixed = [line 35]

- **Access** public

ebookData::$identifierScheme

mixed = [line 36]

- **Access** public

ebookData::$identifierXsiType

mixed = [line 37]

- **Access** public

ebookData::$language

mixed = [line 32]

- **Access** public

ebookData::$languageXsiType

mixed = [line 33]

- **Access** public

ebookData::$manifest

mixed = [line 22]

- **Access** public

ebookData::$manifestData

mixed = [line 23]

- **Access** public

ebookData::$metadata

mixed = [line 21]

- **Access** public

ebookData::$opfPath

mixed = [line 15]

- **Access** public

ebookData::$publisher

mixed = [line 47]

- **Access** public

ebookData::$publisherXmlLang

mixed = [line 48]

- **Access** public

ebookData::$relation

mixed = [line 62]

- **Access** public

ebookData::$relationXmlLang

mixed = [line 63]

- **Access** public

ebookData::$rights

mixed = [line 66]

- **Access** public

ebookData::$rightsXmlLang

mixed = [line 67]

- **Access** public

ebookData::$source

mixed = [line 60]

- **Access** public

ebookData::$sourceXmlLang

mixed = [line 61]

- **Access** public

ebookData::$spine

mixed = [line 24]

- **Access** public

ebookData::$spineData

mixed = [line 25]

- **Access** public

ebookData::$spineToc

mixed = [line 26]

- **Access** public

ebookData::$subject

mixed = [line 49]

- **Access** public

ebookData::$subjectXmlLang

mixed = [line 50]

- **Access** public

ebookData::$tempDir

mixed = [line 13]

- **Access** public

ebookData::$title

mixed = [line 30]

- **Access** public

ebookData::$titleXmlLang

mixed = [line 31]

- **Access** public

ebookData::$toc

mixed = [line 16]

- **Access** public

ebookData::$type

mixed = [line 56]

- **Access** public

ebookData::$typeXsiType

mixed = [line 57]

- **Access** public

ebookData::$xpgt

mixed = [line 17]

- **Access** public

Class ebookRead

[line 13]

- **Package**
- **Throws** no exceptions thrown

ebookRead::$ebookData

mixed = [line 20]

Holds all the ebook data.

- **Access** public

Constructor *void* function ebookRead::ebookRead($epub) *[line 374]*
Function Parameters:

- *$epub* **$epub** Location of an .epub file for processing or an instance of an ebookData object.

Constructor

- **Throws** no exceptions thrown
- public

string function ebookRead::getContentFile($location) *[line 763]*

Function Parameters:

- *string* **$location** Where file is located inside of the content area.

Will give you the requested file out of the epub Note: When finding files based on the manifest's href attribute, the file locations are relative to where the opf file is located in the file structure.

- **Throws** no exceptions thrown
- public

string function ebookRead::getContentLoc() *[line 663]*

Gets the location of all the content inside of the epub.

- **Throws** no exceptions thrown
- public

string function ebookRead::getDcContributor([$optional = null]) *[line 480]*

Function Parameters:

- *string* **$optional** optional attributes are, Role, XmlLang, OpfFileAs.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

string function ebookRead::getDcCoverage([$optional = null]) *[line 633]*
Function Parameters:

- *string* **$optional** optional attributes are, XmlLang.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

string function ebookRead::getDcCreator([$optional = null]) *[line 458]*
Function Parameters:

- *string* **$optional** optional attributes are, Role, XmlLang, OpfFileAs.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

string function ebookRead::getDcDate([$optional = null]) *[line 550]*
Function Parameters:

- *string* **$optional** optional attributes are, Event, XsiType.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

string function ebookRead::getDcDescription([$optional = null]) *[line 534]*

Function Parameters:

- *string* **$optional** optional attributes are, XmlLang.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

string function ebookRead::getDcFormat([$optional = null]) *[line 585]*

Function Parameters:

- *string* **$optional** optional attributes are, XsiType.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

string function ebookRead::getDcIdentifier([$optional = null]) *[line 436]*

Function Parameters:

- *string* **$optional** optional attributes are, Scheme, XsiType, Id.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

string function ebookRead::getDcLanguage([$optional = null]) *[line 420]*
Function Parameters:

- *string* **$optional** optional attributes are, XsiType.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

string function ebookRead::getDcPublisher([$optional = null]) *[line 502]*
Function Parameters:

- *string* **$optional** optional attributes are, XmlLang.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

string function ebookRead::getDcRelation([$optional = null]) *[line 617]*
Function Parameters:

- *string* **$optional** optional attributes are, XmlLang.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

string function ebookRead::getDcRights([$optional = null]) *[line 649]*
Function Parameters:

- *string* **$optional** optional attributes are, XmlLang.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

string function ebookRead::getDcSource([$optional = null]) *[line 601]*
Function Parameters:

- *string* **$optional** optional attributes are, XmlLang.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

string function ebookRead::getDcSubject([$optional = null]) *[line 518]*
Function Parameters:

- *string* **$optional** optional attributes are, XmlLang.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

string function ebookRead::getDcTitle([$optional = null]) *[line 404]*
Function Parameters:

- *string* **$optional** optional attributes are, XmlLang.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

string function ebookRead::getDcType([$optional = null]) *[line 569]*
Function Parameters:

- *string* **$optional** optional attributes are, XsiType.

Gives you the title by default or the attribute specified

- **Throws** no exceptions thrown
- public

eBookData function ebookRead::getEBookDataObject() *[line 777]*

Get the eBook data object. Holds all the data from the epub.

- **Throws** no exceptions thrown
- public

string function ebookRead::getFilePath($ext) *[line 746]*

Function Parameters:

- *string* **$ext** The file exenshion of the file your looking for.

Get a file path by extenshion

- **Throws** no exceptions thrown
- public

string function ebookRead::getGuide($item) *[line 726]*

Function Parameters:

- *string* **$item** What attribute do you want from the guide (title, href, or type).

Gets the guide item desired

- **Throws** no exceptions thrown
- public

string function ebookRead::getGuideItem($index, $item) *[line 717]*
Function Parameters:

- *int* **$index** The index number of the guide item.
- *string* **$item** What attribute do you want from the guide (title, href, or type).

Gets the guide item desired

- **Throws** no exceptions thrown
- public

string function ebookRead::getManifest($item) *[line 684]*
Function Parameters:

- *string* **$item** What attribute do you want from the manifest (id, href, or media-type).

Gets the manifest item desired

- **Throws** no exceptions thrown
- public

string function ebookRead::getManifestItem($index, $item) *[line 673]*
Function Parameters:

- *int* **$index** The index number of the manifest item.

- *string* **$item** What attribute do you want from the manifest (id, href, or media-type).

Gets the manifest item desired

- **Throws** no exceptions thrown
- public

string function ebookRead::getSpine() *[line 707]*

Gets a desired spine item

- **Throws** no exceptions thrown
- public

string function ebookRead::getSpineItem($index) *[line 699]*

Function Parameters:

- *int* **$index** The index of the spine item to return.

Gets a desired spine item

- **Throws** no exceptions thrown
- public

Class ebookWrite

[line 13]

- **Package**
- **Throws** no exceptions thrown

Constructor *void* function ebookWrite::ebookWrite([$ebookData = null]) *[line 619]*

Function Parameters:

- *ebookData* **$ebookData** Optional, give this writter a ebookData object for editing.

Constructor

- **Throws** no exceptions thrown
- public

bool function ebookWrite::addContentFile($fileLoc, $name, $mime, [$subDir = null], [$fallBackId = null]) *[line 1590]*

Function Parameters:

- *string* **$fileLoc** Where is the file currently located.
- *string* **$subDir** If you want the file in a subdirectory of the content folder. Must end with a slash.
- *string* **$name** Short discriptive name of what the content item is. eg: Chapter 3.
- *string* **$mime** the mime type of the file being added.
- *string* **$fallBackId** the id of an item you wish to use as a fallback item.

This is used for adding content files to an epub. Make sure you have write permission to the destination directory.

- **Throws** no exceptions thrown
- public

void function ebookWrite::buildEPUB($name, $dest) *[line 636]*

Function Parameters:

- *string* **$name** The name you want the epub file to have. Don't include a file extenshion.
- *string* **$dest** The location you want the final epub file.

Puts together the eBook and places it in the desired directory.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setDcContributor($contributor, [$index = null]) *[line 910]*

Function Parameters:

- *string* **$contributor** The contributor to be set. If you want set more then one contributor send an array into this parameter.
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the contributor. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

bool function ebookWrite::setDcContributorAttrib($attrib, $attribValue, [$index = null]) *[line 928]*

Function Parameters:

- *string* **$attrib** The name of the optional attributes to be set. Optional attributes are, Role, XmlLang, OpfFileAs.
- *string* **$attribValue** The value to be set for the optional attribute. If you want set more then one contributor attribute send an array into this parameter. Make sure the attributes corresponde the appropriate contributor index
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional contributor attributes.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setDcCoverage($coverage, [$index = null]) *[line 1403]*

Function Parameters:

- *string* **$coverage** The coverage to be set. If you want set more then one coverage send an array into this parameter.
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the coverage. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

bool function ebookWrite::setDcCoverageAttrib($attrib, $attribValue, [$index = null]) *[line 1421]*

Function Parameters:

- *string* **$attrib** The name of the optional attributes to be set. Optional attributes are XmlLang.
- *string* **$attribValue** The value to be set for the optional attribute. If you want set more then one coverage attribute send an array into this parameter. Make sure the attributes corresponde the appropriate coverage index
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Set the optional coverage attributes.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setDcCreator($creator, [$index = null]) *[line 842]*

Function Parameters:

- *string* **$creator** The creator to be set. If you want set more then one creator send an array into this parameter.
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the creator. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

bool function ebookWrite::setDcCreatorAttrib($attrib, $attribValue, [$index = null]) *[line 860]*

Function Parameters:

- *string* **$attrib** The name of the optional attributes to be set. optional attributes are, Role, XmlLang, OpfFileAs.
- *string* **$attribValue** The value to be set for the optional attribute. If you want set more then one creator attribute send an array into this parameter. Make sure the attributes corresponde the appropriate creator index
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional creator attributes.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setDcDate($eBookdate, [$index = null], $date) *[line 1134]*

Function Parameters:

- *string* **$date** The date to be set. If you want set more then one date send an array into this parameter.

- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.
- **$eBookdate**

Sets the date. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

bool function ebookWrite::setDcDateAttrib($attrib, $attribValue, [$index = null]) *[line 1153]*

Function Parameters:

- *string* **$attrib** The name of the optional attributes to be set. Optional attributes are, Event, XsiType.
- *string* **$attribValue** The value to be set for the optional attribute. If you want set more then one date attribute send an array into this parameter. Make sure the attributes corresponde the appropriate date index
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional date attributes.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setDcDescription($description, [$index = null]) *[line 1082]*

Function Parameters:

- *string* **$description** The description to be set. If you want set more then one description send an array into this parameter.
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the description. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

bool function ebookWrite::setDcDescriptionAttrib($attrib, $attribValue, [$index = null]) *[line 1100]*

Function Parameters:

- *string* **$attrib** The name of the optional attributes to be set. Optional attributes are, XmlLang.
- *string* **$attribValue** The value to be set for the optional attribute. If you want set more then one description attribute send an array into this parameter. Make sure the attributes corresponde the appropriate description index
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional description attributes.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setDcFormat($format, [$index = null]) *[line 1247]*

Function Parameters:

- *string* **$format** The format to be set. If you want set more then one format send an array into this parameter.
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the format. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

bool function ebookWrite::setDcFormatAttrib($attrib, $attribValue, [$index = null]) *[line 1265]*

Function Parameters:

- *string* **$attrib** The name of the optional attributes to be set. Optional attributes are XsiType.
- *string* **$attribValue** The value to be set for the optional attribute. If you want set more then one format attribute send an array into this parameter. Make sure the attributes corresponde the appropriate format index
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional format attributes.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setDcIdentifier($identifier, $id, [$index = null]) *[line 773]*

Function Parameters:

- *string* **$identifier** the Identifier to be set. If you want set more then one Identifier send an array into this parameter.
- *string* **$id** discription of what the identifier is.
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the Identifier. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

bool function ebookWrite::setDcIdentifierAttrib($attrib, $attribValue, [$index = null]) *[line 794]*

Function Parameters:

- *string* **$attrib** Id is a required attribute. The name of the optional attributes to be set. Optional attributes are Scheme, XsiType.
- *string* **$attribValue** The value to be set for the optional attribute. If you want set more then one identifier attribute send an array into this parameter. Make sure the attributes corresponde the

appropriate identifier index

- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional identifier attributes.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setDcLanguage($language, [$index = null]) *[line 720]*

Function Parameters:

- *string* **$language** The language to be set. If you want set more then one language send an array into this parameter.
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the language. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

bool function ebookWrite::setDcLanguageAttrib($attrib, $attribValue, [$index = null]) *[line 738]*

Function Parameters:

- *string* **$attrib** The name of the optional attributes to be set. Optional attributes are XsiType.
- *string* **$attribValue** The value to be set for the optional attribute. If you want set more then one language attribute send an array into this parameter. Make sure the attributes corresponde the appropriate language index
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional language attributes.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setDcPublisher($publisher, [$index = null]) *[line 978]*

Function Parameters:

- *string* **$publisher** The publisher to be set. If you want set more then one publisher send an array into this parameter.
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the publisher. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

bool function ebookWrite::setDcPublisherAttrib($attrib, $attribValue, [$index = null]) *[line 996]*

Function Parameters:

- *string* **$attrib** The name of the optional attributes to be set. Optional attributes are, XmlLang.
- *string* **$attribValue** The value to be set for the optional attribute. If you want set more then one publisher attribute send an array into this parameter. Make sure the attributes corresponde the appropriate publisher index
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional publisher attributes.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setDcRelation($relation, [$index = null]) *[line 1351]*

Function Parameters:

- *string* **$relation** The relation to be set. If you want set more then one relation send an array into this parameter.
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the relation. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

bool function ebookWrite::setDcRelationAttrib($attrib, $attribValue, [$index = null]) *[line 1369]*

Function Parameters:

- *string* **$attrib** The name of the optional attributes to be set. Optional attributes are XmlLang.
- *string* **$attribValue** The value to be set for the optional attribute. If you want set more then one relation attribute send an array into this parameter. Make sure the attributes corresponde the appropriate relation index
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Set the optional relation attributes.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setDcRights($rights, [$index = null]) *[line 1455]*

Function Parameters:

- *string* **$rights** The rights to be set. If you want set more then one rights send an array into this parameter.
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the rights. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

bool function ebookWrite::setDcRightsAttrib($attrib, $attribValue, [$index = null]) *[line 1473]*

Function Parameters:

- *string* **$attrib** The name of the optional attributes to be set. Optional attributes are XmlLang.
- *string* **$attribValue** The value to be set for the optional attribute. If you want set more then one rights attribute send an array into this parameter. Make sure the attributes corresponde the appropriate rights index
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Set the optional rights attributes.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setDcSource($source, [$index = null]) *[line 1299]*

Function Parameters:

- *string* **$source** The source to be set. If you want set more then one source send an array into this parameter.
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the source. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

bool function ebookWrite::setDcSourceAttrib($attrib, $attribValue, [$index = null]) *[line 1317]*

Function Parameters:

- *string* **$attrib** The name of the optional attributes to be set. Optional attributes are XmlLang.
- *string* **$attribValue** The value to be set for the optional attribute. If you want set more then one source attribute send an array into this parameter. Make sure the attributes corresponde the appropriate source index
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Set the optional source attributes.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setDcSubject($subject, [$index = null]) *[line 1030]*

Function Parameters:

- *string* **$subject** The subject to be set. If you want set more then one subject send an array into this parameter.
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the subject. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

bool function ebookWrite::setDcSubjectAttrib($attrib, $attribValue, [$index = null]) *[line 1048]*

Function Parameters:

- *string* **$attrib** The name of the optional attributes to be set. Optional attributes are, XmlLang.
- *string* **$attribValue** The value to be set for the optional attribute. If you want set more then one subject attribute send an array into this parameter. Make sure the attributes corresponde the appropriate subject index

- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional subject attributes.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setDcTitle($title, [$index = null]) *[line 668]*

Function Parameters:

- *string* **$title** The title to be set. If you want set more then one title send an array into this parameter.
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the title. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

bool function ebookWrite::setDcTitleAttrib($attrib, $attribValue, [$index = null]) *[line 686]*

Function Parameters:

- *string* **$attrib** The name of the optional attributes to be set. Optional attributes are XmlLang.
- *string* **$attribValue** The value to be set for the optional attribute. If you want set more then one title attribute send an array into this parameter. Make sure the attributes corresponde the appropriate title index
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Set the optional title attributes.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setDcType($type, [$index = null]) *[line 1195]*
Function Parameters:

- *string* **$type** The type to be set. If you want set more then one type send an array into this parameter.
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the type. Multiple instances are permitted.

- **Throws** no exceptions thrown
- public

bool function ebookWrite::setDcTypeAttrib($attrib, $attribValue, [$index = null]) *[line 1213]*
Function Parameters:

- *string* **$attrib** The name of the optional attributes to be set. Optional attributes are XsiType.
- *string* **$attribValue** The value to be set for the optional attribute. If you want set more then one type attribute send an array into this parameter. Make sure the attributes corresponde the appropriate type index
- *int* **$index** if there is more then one instance of this item, specify it and that item will be edited.

Sets the optional type attributes.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setGuide($title, $type, $href, [$index = null]) *[line 1551]*
Function Parameters:

- *string* **$title** Name of the guide item.
- *string* **$type** The required type attribute describes the publication component referenced by the href attribute. Type should be of this list: cover, title-page, toc "table of contents", index "back-of-book style index", glossary, acknowledgements, bibliography, colophon, copyright-page, dedication, epigraph, foreword, loi "list of illustrations", lot "list of tables", notes, and preface.
- *string* **$href** location of the guide item.
- *int* **$index** The index number of the guide item to be set.

sets the guide item desired. Arrays can be sent into the attributes to set more then one guide item at a time. The guide element identifies fundamental structural components of the publication, to enable Reading Systems to provide convenient access to them.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setSpine($spineId, [$index = null]) *[line 1507]*

Function Parameters:

- *string* **$spineId** Send an array of ID names as they appear in the manifest in the order you want them read.
- *int* **$index** If you wish to set or change just one element of the spine then specify what one in the index parameter.

Sets a desired spine order. The spine is used to determine the reading order.

- **Throws** no exceptions thrown
- public

void function ebookWrite::setSpineToc($toc) *[line 1532]*

Function Parameters:

- *$toc* **$toc** The manifest item that is the table of contents.

Used to tell the spine what manifest item is the Table Of Contents. This does not need to be set, this is optional.

- **Throws** no exceptions thrown
- public

Class guide

[line 80]

The guide element identifies fundamental structural components of the publication, to enable Reading Systems to provide convenient access to them.

- **Package**
- **Throws** no exceptions thrown

guide::$href

mixed = [line 83]

- **Access** public

guide::$title

mixed = [line 82]

- **Access** public

guide::$type

mixed = [line 81]

- **Access** public

Class manifest

[line 69]

- **Package**
- **Throws** no exceptions thrown

manifest::$fallback

mixed = [line 73]

- **Access** public

manifest::$href

mixed = [line 71]

- **Access** public

manifest::$id

mixed = [line 70]

- **Access** public

manifest::$type

mixed = [line 72]

- **Access** public

Class tagSet

[line 85]

- **Package**
- **Throws** no exceptions thrown

tagSet::$tag

mixed = [line 86]

- **Access** public

tagSet::$value

mixed = [line 87]

- **Access** public

Appendices

Appendix A - Class Trees

Package

ebook

- ebook

ebookData

- ebookData

ebookRead

- ebookRead

ebookWrite

- ebookWrite

guide

- guide

manifest

- manifest

tagSet

- tagSet

Index