
eQuercus v1.20

Library

Reference Manual

Contents

Introduction.....	4
The Site object.....	5
Proprieties.....	5
Methods.....	5
GetRequestedNode method.....	6
GetTreeNodesById method.....	6
GetTreeNodesByName method.....	7
GetActiveLanguages method.....	7
GetRequestedLanguageId method.....	7
GetNodeLink method.....	8
GetNodeContentLink method.....	10
GetChangeLanguageLink method.....	11
GetContent method.....	11
GetListOfContents method.....	12
GetContentsCount method.....	13
GetRequestedContentId.....	13
GetLastUpdate method.....	13
The Node object	14
GetNextSibling method.....	14
GetPrevSibling method.....	14
GetParent method.....	15
GetChildByOrder method.....	15
HasChilds method.....	15
HasContents method.....	15
IsLink method.....	16
IsExternalLink method.....	16
IsAncestorOf method.....	16
IsEqualOf method.....	17
GetImageMenuNormal method.....	18
GetImageMenuActive method.....	18
GetImageMenuMouseOver method.....	18
The ContentText object.....	19
EOF property.....	19
BOF property.....	19
RecordCount property.....	19
RecordPos property.....	19
MoveFirst method.....	20
MoveLast method.....	20
Move method.....	20
MoveNext method.....	20
MovePrev method.....	20
GetId method.....	20
GetTitle method.....	21
GetSubTitle method.....	21
GetPicture1 method.....	21
GetPicture2 method.....	21
GetLink method.....	21
GetLinkDescription method.....	21
GetBody method.....	21
GetCreationDate method.....	22
GetModificationDate method.....	22
The ContentHTML object	23
EOF property.....	23
BOF property.....	23

RecordCount property	23
RecordPos property	23
MoveFirst method	24
MoveLast method.....	24
Move method.....	24
MoveNext method	24
MovePrev method	24
GetId method.....	24
GetTitle method.....	24
GetBody method.....	25
GetCreationDate method	25
GetModificationDate method	25
The Language object	26
Id property	26
Name property	26
Short property.....	26

CHAPTER 1

Introduction

This manual describes the properties and methods “eQuercus - Library” objects. In particular, the methods and properties of the following objects are described:

- Site
- Node
- ContentText
- ContentHTML
- Language

Note

To better understand operation of “eQuercus - Library” it is necessary to have some background knowledge of Jscript language for the server side (ASP pages).

To use the library properties and methods, include the equercus.inc file on all web site dynamic pages.

The first lines of every site ASP page must be:

```
<%@ LANGUAGE=JScript %>
<!--#include file="./library/equercus.inc"-->
```

CHAPTER 2

The Site object

The *Site* object is the main object of the “eQuercus - Library” and contains the main properties and methods for reading database data related to the site structure and contents. The object also has the job, via the constructor, of reading eQuercus private parameters sent to the page via the GET command, as listed below:

Parameter	Description
<i>nid</i>	Specifies the node identifier for the contents to be represented.
<i>nname</i>	Specifies the node name for the contents to be represented.
<i>lid</i>	Specifies the language identifier for displaying the menu and contents.
<i>lname</i>	Specifies the language abbreviation (E, S, F, D or I) for displaying the menu and contents.
<i>cid</i>	Specifies the content identifier, related to the content list connected to a node.
<i>prv</i>	Specifies the mode for representing a content (preview or normal). This parameter is only used by “eQuercus - Website and content management”.

Proprieties

The various object properties are described in the following table.

Propriety	Description
<i>m_cn</i>	The property of a Connection type object and is used to execute internal operations on the database.
<i>m_nodeId</i>	The property contains the node id for displaying the content of the active page. The value is copied by the <i>nid</i> parameter. If there is no <i>nid</i> parameter, the system searches for the node identifier using the <i>nname</i> parameter. Even if there is no <i>nname</i> , the default value is taken corresponding to the id of the first node of the structure that uses the active page.
<i>m_langId</i>	The property contains the language id for representing the menus and contents. The value is copied by the <i>lid</i> parameter. If there is no <i>lid</i> parameter, the system searches for the language using the <i>lname</i> parameter. Even if there is no <i>lname</i> , the default language id default is assigned as the value.
<i>m_contentId</i>	The property contains the id of the content to be displayed. The value is copied by the <i>cid</i> parameter, and if there is no parameter, the id of the first active language content is searched for, in order of priority, connected to the <i>m_nodeId</i> node.
<i>m_bPreview</i>	The property specifies whether to display the content in preview or normal mode. The value is copied by the <i>prv</i> parameter. This property is used inside the <i>Site</i> object.

Methods

Site object methods are listed below broken down into categories:

Methods related to site structure:

GetRequestedNode

GetTreeNodeById

GetTreeNodeByName

Methods related to links:*GetNodeLink**GetNodeContentLink**GetChangeLanguageLink***Methods related to contents:***GetContent**GetListOfContents**GetContentsCount**GetRequestedContentId***Methods related to languages:***GetActiveLanguages**GetRequestedLanguageId***Methods related to general information:***GetLastUpdate*

GetRequestedNode method

Description

Returns the *Node* type object corresponding node requested by the page.

Syntax

```
oSite.GetRequestedNode( )
```

Remarks

This method internally requests the *Site.GetTreeNodesById(Site.m_nodeId , 0)* method.
The following example illustrates the use of the method:

```
var oSite = new Site();
var oMyRequestedNode = oSite.GetRequestedNode( );
```

GetTreeNodesById method

Description

Returns the *Node* type root object for the requested tree structure.

Syntax

```
oSite.GetTreeNodesById(nodeId, deep)
```

Parameter	Description
<i>nodeId</i>	Request - Type: numeric. Specifies the starting node id of the tree structure. -1: returns the entire structure for the number of levels specified in <i>deep</i> .
<i>deep</i>	Request - Type: numeric. Represents the number of sublevels starting from the requested node, to be considered in constructing the tree. 0: only the requested node is present in the structure. (a single node). 1: the structure is composed of the requested node and its directly connected subnodes. >1: indicates the number of sublevels to be considered.

Remarks

The method uploads the site logical structure for construction of the navigation menu.

Use the *Node* object methods to navigate in the structure.
The following example illustrates the use of the method:

```
var oSite = new Site();
var oRootNode = oSite.GetTreeNodesById(-1, 3);
// Gets the entire site structure for the first three levels.
```

GetTreeNodesByName method

Description

This method is identical to the *Site.GetTreeNodesById()* method, the only difference is that the node identifier is the name and not the id.

Syntax

```
oSite.GetTreeNodesByName(nodeName, deep)
```

Parameter	Description
<i>nodeName</i>	Request - Type: string. Represents the starting node name of the tree structure.
<i>deep</i>	Request - Type: numeric. See the <i>oSite.GetTreeNodesById</i> method for a description.

Remarks

This method searches for the id of the requested node and returns the *Site.GetTreeNodesById()* method.
The following example illustrates the use of the method:

```
var oSite = new Site();
var oRootNode = oSite.GetTreeNodesByName ("Products", 1);
// Gets the "Products" node and all its directly connected subnodes.
```

GetActiveLanguages method

Description

Returns an *Array* containing *Language* type objects, which correspond to the active languages in eQuercus.

Syntax

```
oSite.GetActiveLanguages()
```

Remarks

The following example illustrates the use of the method:

```
var oSite = new Site();
var oLanguages = oSite.GetActiveLanguages();
for (var i=0; i < oLanguages.length ; i++)
    Response.Write(oLanguages[i].name);
```

GetRequestedLanguageId method

Description

Returns the value of the *Site.m_langId* property.

Syntax

```
oSite.GetRequestedLanguageId()
```

GetNodeLink method

Description

Returns the string containing the link applied to the node. If there is not a valid link, the “#” string is returned.

Syntax

```
oSite.GetNodeLink(node, bIncludeTarget, extra_params)
```

Parameter	Description
<i>node</i>	Request – Type: <i>Node</i> . Specifies the node for executing the request.
<i>bIncludeTarget</i>	Request – Type: Boolean true: indicates that the result must also contain the “target” HTML tag. The link must be entered between double quotes false: indicates that the result must only contain the link. The link is not entered between double quotes.
<i>extra_params</i>	Optional – Type: Array type object of strings. Contains additional parameters for the link. The string format is <i>myParameter=myValue</i> .

Remarks

The method returns the link specified in the “Create or Edit node” pages of “eQuercus - Website and content management”. Based on the node configurations, the method can return one of the following values:

- If the node is a valid type and can contain only one content, the specific value is returned in the “*Page for detail*” field.
- If the node is a valid type and can contain more than one content, the specific value is returned in the “*Page for list*” field. To learn the link to the detail page use the *oSite.GetNodeContentLink* method.
- If the node is an “- *external link* -” type, the specific value is returned in the “*External link*” field.
- If the node does not have a connected content (“- *no content* -”), it is assigned the link in the first descendent node of the substructure that is valid (“Text document” or “HTML document”). The rules which determine which link should be considered the descendent node are in points a) and b).

If the parameter *bIncludeTarget* is equal to *true*, the “target” HTML tag is also returned with the value in the “*Frame target*” field.

The following structure is an example which corresponds to the logical structure of the entire site:

Root

<i>Node01</i>	(id=1, content type = “- no content -”)
<i>Node02</i>	(id=2, content type = “- no content -”)
<i>Node03</i>	(id=3, content type = “Text document”, number of contents = “one”, Page for detail = “mydetail.asp”)
<i>Node04</i>	(id=4, content type = “Text document”, number of contents = “many”, Page for detail = “mydetail.asp”, Page for list = “mylist.asp”)
<i>Node05</i>	(id=5, content type = “- external link -”, link = “http://externallink”)

The following example illustrates the use of the **GetNodeLink** method:

```
var oSite = new Site();
var oNode01 = oSite.GetTreeNodesByName("Node01", 0);
var oNode02 = oSite.GetTreeNodesByName("Node02", 0);
var oNode03 = oSite.GetTreeNodesByName("Node03", 0);
var oNode04 = oSite.GetTreeNodesByName("Node04", 0);
var oNode05 = oSite.GetTreeNodesByName("Node05", 0);

var oExtra_parameters = new Array();
oExtra_parameters [0] = "user=test";

var strLink1a = oSite.GetNodeLink (oNode01, true, oExtra_parameters);
// strLink1a: "mydetail.asp?nid=3&lid=1&user=test" target="_self"

var strLink1b = oSite.GetNodeLink (oNode01, false);
// strLink1b: mydetail.asp?nid=3&lid=1

var strLink2 = oSite.GetNodeLink (oNode02, false);
// strLink2: mydetail.asp?nid=3&lid=1

var strLink3 = oSite.GetNodeLink (oNode03, false);
// strLink3: mydetail.asp?nid=3&lid=1

var strLink4 = oSite.GetNodeLink (oNode04, false);
// strLink4: mylist.asp?nid=4&lid=1

var strLink5 = oSite.GetNodeLink (oNode05, false);
// strLink5: http://externallink
```

GetNodeContentLink method

Description

Returns the string contained in the link specified in the “*Page for detail*” field. If a valid link does not exist, the “#” string is returned.

Syntax

```
oSite.GetNodeContentLink(node, bIncludeTarget, contentId, extra_params)
```

Parameter	Description
<i>Node</i>	Request - Type: <i>Node</i> . Specifies the node for executing the request.
<i>bIncludeTarget</i>	Request – Type: Boolean. true: indicates that the result must also contain the “target” HTML tag. The link must be entered between double quotes false: indicates that the result must only contain the link. The link is not entered between double quotes.
<i>contentId</i>	Optional Default: id of the first active language content, in order of priority, connected to the node. Type: numeric. Indicates the content identifier within the content list connected to the node.
<i>extra_params</i>	Optional – Type: Array type object of strings. Contains additional parameters for the link. The string format is <i>myParameter=myValue</i> .

Remarks

The method returns the link specified in the “*Page for detail*” field in the “Create or Edit node” pages of “eQuercus - Website and content management”. This method must be used in place of the *Site.GetNodeLink* method to learn the link to the page for detail, if both the “*Page for detail*” and “*Page for list*” fields have been set.

If the node does not have a connected content type (“- no content -”), the link set in the “*Page for detail*” of the first available descendent node is used.

If the parameter *bIncludeTarget* is equal to *true*, the “target” HTML tag is also returned with the value in the “*Frame target*” field.

The following structure is an example which corresponds to the logical structure of the entire site:

Root

```

Node01          (id=1, content type = “- no content -”)

Node02          (id=2, content type = “Text document”, number of contents = “many”,
                 Page for detail = “mydetail.asp”, Page for list = “mylist.asp”)
  
```

The following example illustrates the use of the **GetNodeContentLink** method:

```

var oSite = new Site();
var oNode01 = oSite.GetTreeNodeByName("Node01", 0);
var oNode02 = oSite.GetTreeNodeByName("Node02", 0);
var strLink1 = oSite.GetNodeContentLink(oNode01, true);
// strLink1: "mydetail.asp?nid=2&lid=1&cid=1" target="_self"

var strLink2 = oSite.GetNodeContentLink(oNode02, false);
// strLink2: mydetail.asp?nid=2&lid=1&cid=1
  
```

GetChangeLanguageLink method

Description

Returns a string containing the link used for displaying the active page in another language.

Syntax

```
oSite.GetChangeLanguageLink(newLangId, extra_params)
```

Parameter	Description
<i>newLangId</i>	Request - Type: numeric. Specifies the new language identifier for displaying the active page.
<i>extra_params</i>	Optional – Type: Array type object of strings. Contains additional parameters for the link. The string format is <i>myParameter=myValue</i> .

Remarks

The following example illustrates the use of the **GetChangeLanguageLink** method:

```
var oSite = new Site();
var oLanguages = oSite.GetActiveLanguages();
var oExtra_parameters = new Array();
oExtra_parameters [0] = "user=myuser";
oExtra_parameters [1] = "color=1";

for (var i=0; i < oLanguages.length ; i++)
    Response.Write(oSite.GetChangeLanguageLink(oLanguages[i].id, oExtra_parameters));

// mypage.asp?nid=10&lid=1&cid=5&user=myuser&color=1
// mypage.asp?nid=10&lid=2&cid=5&user=myuser&color=1
// mypage.asp?nid=10&lid=3&cid=5&user=myuser&color=1
```

GetContent method

Description

Returns the requested content type object.

Syntax

```
oSite.GetContent(type, nodeId, langId, contentId)
```

Parameter	Description
<i>type</i>	Request - Type: numeric. The value specifies the object type to return which is equal to the content type connected to the link. 0: returns <i>ContentText</i> type object. 1: returns <i>ContentHTML</i> type object.
<i>nodeId</i>	Optional - Default: <i>Site.m_nodeId</i> - Type: numeric. Specifies the node id for requesting the content.
<i>langId</i>	Optional - Default: <i>Site.m_langId</i> - Type: numeric. Specifies the language id of the requested content.
<i>contentId</i>	Optional Default: id of the first specified language content, in order of priority, connected to the node. Type: numeric. Specifies the content id.

Remarks

The *type* parameter is used to limit the number of queries to be executed. Since it already returns the content type connected to a node it is not necessary to execute a query to search for the type id.

The following example illustrates the use of the **GetContent** method:

```
var oSite = new Site();
var oContentHTML = oSite.GetContent(1);
Response.Write(oContentHTML.GetBody());
```

GetListOfContents method**Description**

Returns a requested content type object, which contains a *Recordset* type object for reading the information of the requested contents.

Syntax

GetListOfContents (type, orderBy, bAsc, from, to, nodeId, langId)

Parameter	Description
<i>type</i>	Request - Type: numeric. The value specifies the content type to return. 0: returns <i>ContentText</i> type object. 1: returns <i>ContentHTML</i> type object.
<i>orderBy</i>	Optional - Default: 0 - Type: numeric. Specifies the field for ordering the list. 0 (by Order): the list order matches the order on the “Content list” page of “eQuercus - Website and content management”. 1 (by Title): orders the list by the content <i>Title</i> field. 2 (by Last modification date): orders the list by <i>Last modification date</i> . 3 (by Creation date): orders the list by <i>Creation date</i> .
<i>bAsc</i>	Optional - Default: true - Type: Boolean. true: ascending order. false: descending order.
<i>from</i>	Optional - Default: 0 - Type: numeric. Specifies the first record position (of recordset) to consider; zero based index. Use 0 value to request all records up to the record specified in the <i>to</i> parameter.
<i>to</i>	Optional - Default: 1 - Type: numeric. Specifies the last record position (of recordset) to consider; zero based index. Use -1 value to request all records starting from the record specified in the <i>from</i> parameter.
<i>nodeId</i>	Optional - Default: <i>Site.m_nodeId</i> - Type: numeric. Specifies the node id for requesting the content list.
<i>langId</i>	Optional - Default: <i>Site.m_langId</i> - Type: numeric. Specifies the language id of the requested content.

Remarks

The *type* parameter is used to limit the number of queries to be executed. Since it already returns the content type connected to a node it is not necessary to execute a query to search for the type id.

Only use this method for nodes that can contain more than one content (“*Number of contents: Many*”).

The following example illustrates the use of the **GetListOfContents** method:

```
var oSite = new Site();
var oListOfContentsHTML = oSite.GetListOfContents(1);
```

```
while (!oListOfContentsHTML.EOF)
{
    Response.Write(oListOfContentsHTML.GetBody());
    oListOfContentsHTML.MoveNext();
}
```

GetContentsCount method

Description

Returns the number of contents for a certain language in a node.

Syntax

```
GetContentsCount(nodeId, langId)
```

Parameter	Description
<i>nodeId</i>	Optional - Default: <i>Site.m_nodeId</i> - Type: numeric. Specifies the node id for the request.
<i>langId</i>	Optional - Default: <i>Site.m_langId</i> - Type: numeric. Specifies the language id for the request.

The following example illustrates the use of the **GetContentsCount** method:

```
var oSite = new Site();
var count = oSite.GetContentsCount();
```

GetRequestedContentId

Description

Returns the *Site.m_contentId* property value.

Syntax

```
oSite.GetRequestedContentId()
```

GetLastUpdate method

Description

Returns a string containing the date of the last update of a content.

Syntax

```
oSite.GetLastUpdate()
```

Remarks

The date format depends on the configuration in the global.asa file.

CHAPTER 3

The Node object

The *Node* object contains the properties and methods for connecting to other nodes in order to form the structure and tree corresponding to the site logical structure. In addition to this information, it contains the properties and methods used for displaying the navigation menu. The object constructor is not documented because it is used by the *Site* object. The object does not have public properties.

The following methods are used to navigate in the tree structure:

GetNextSibling

GetPrevSibling

GetParent

GetChildByOrder

The following methods are used for executing various controls:

HasChilds

HasContents

IsLink

IsExternalLink

IsAncestorOf

IsEqualOf

The following methods return information connected to the node:

GetId

GetLevel

GetTextMenu

GetImageMenuNormal

GetImageMenuActive

GetImageMenuMouseOver

GetNextSibling method

Description

Returns a *Node* type object corresponding to the sibling node subsequent to this node. Sibling nodes have a common parent node. *Null* is returned if there are no subsequent nodes.

Syntax

```
oNode.GetNextSibling( )
```

GetPrevSibling method

Description

Returns a *Node* type object corresponding to the sibling node previous to this node. Sibling nodes have a common parent node. *Null* is returned if there are no previous nodes.

Syntax

```
oNode.GetPrevSibling( )
```

GetParent method

Description

Returns a *Node* type object corresponding to the parent node of this node. *Null* is returned if the node is not connected to a parent node.

Syntax

```
oNode.GetParent()
```

GetChildByOrder method

Description

Returns a *Node* type object corresponding to the specified child node. *Null* is returned if the node does not have a child node connected to the specified position.

Syntax

```
oNode.GetChildByOrder(order)
```

Parameter	Description
<i>order</i>	Request - Type: numeric. Specifies the position of the child node. The first child node is in position 0.

Remarks

The following example illustrates the use of the **GetChildByOrder** method:

```
var oSite = new Site();
var oRootNode = oSite.GetTreeNodesById(-1, 1);
var oNode = null;
var i = 0;
while ((oNode = oRootNode.GetChildByOrder(i++)) != null)
{
    Response.Write(oNode.GetTextMenu());
}
```

HasChilds method

Description

Returns *true* if the node has one or more child nodes, otherwise *false*.

Syntax

```
oNode.HasChilds()
```

HasContents method

Description

Returns *true* if the node is a valid content type ("Text document", "HTML document", etc..) otherwise *false* ("- no content -" o "- external link -").

Syntax

```
oNode.HasContents()
```

IsLink method

Description

Returns *true* if the node contains a link, otherwise *false*.

Syntax

```
oNode.IsLink()
```

Remarks

A node contains a link if:

- the node is a valid content type Text document”, “ HTML document”, etc..);
- the node is an external link type (“- external link -”);
- the node has at least one descendent link, in any sublevel, that is a valid content type.

IsExternalLink method

Description

Returns *true* if the node is an “-external link-“ type, otherwise *false*.

Syntax

```
oNode.IsExternalLink()
```

IsAncestorOf method

Description

Returns *true* if the node is an ancestor of the to compare node, otherwise *false*.

Syntax

```
oNode.IsAncestorOf (nodeToCompare)
```

Parameter	Description
<i>nodeToCompare</i>	Request - Type: <i>Node</i> . Specifies the node to compare.

Remarks

The comparison is executed on the position of the nodes within the site logical structure. The comparison is valid even if the nodes to be compared do not belong to the same tree structure.

The following example structure corresponds to the entire site logical structure:

Root

MyNode

MySubNode

MySubSubNode1

MySubSubNode2

The following example illustrates the use of the **IsAncestorOf** method:

```
var oSite = new Site();
var oMyNode = oSite.GetTreeNodeByNames("MyNode", 2);
var oMySubNode = oMyNode.GetChildByOrder(0);
var oMySubSubNode1 = oMySubNode.GetChildByOrder(0);
var oMySubSubNode2 = oSite.GetTreeNodeByNames("MySubSubNode2", 0);

Response.Write(oMyNode.IsAncestorOf(oMySubNode));
```



```
// true

Response.Write(oMyNode.IsAncestorOf(oMySubSubNode1));
// true

Response.Write(oMySubNode.IsAncestorOf(oMySubSubNode2));
// true

Response.Write(oMySubNode.IsAncestorOf(oMySubNode));
// false
```

IsEqualOf method

Description

Returns *true* if the node has the same *id* as the to compare node, otherwise false.

Syntax

`oNode.IsEqualOf(nodeToCompare)`

Parameter	Description
nodeToCompare	Request - Type: <i>Node</i> . Specifies the node to compare.

Remarks

The comparison is executed between node ids.

The following example structure corresponds to the entire site logical structure:

Root

MyNode

MySubNode

MySubSubNode1

MySubSubNode2

The following example illustrates the use of the **IsEqualOf** method:

```
var oSite = new Site();
var oMyNode = oSite.GetTreeNodesByName("MyNode", 2);
var oMySubNode = oMyNode.GetChildByOrder(0);
var oMySubSubNode1 = oMySubNode.GetChildByOrder(0);
var oN = oSite.GetTreeNodesByName("MySubSubNode1", 0);

Response.Write(oMySubSubNode1.IsEqualOf(oN));
// true
```

GetImageMenuNormal method

Description

Returns the string containing the address of the image used to represent the menu in normal status.

Syntax

```
oNode.GetImageMenuNormal ( )
```

Remarks

The return value corresponds to the value set in the “*Normal menu image*”, for the active node language, in the “Create or Edit node” pages of “eQuercus - Website and content management”.

GetImageMenuActive method

Description

Returns the string containing the address of the image used to represent the menu in active status.

Syntax

```
oNode.GetImageMenuActive ( )
```

Remarks

The return value corresponds to the value set in the “*Active menu image*”, for the active node language, in the “Create or Edit node” pages of “eQuercus - Website and content management”.

If the field does not contain a value, the address in the “*MouseOver menu image*” field will be returned. And if this field does not contain a value, the value in the “*Normal menu image*” will be returned.

GetImageMenuMouseOver method

Description

Returns the string containing the address of the image used to represent the menu in MouseOver status.

Syntax

```
oNode.GetImageMenuMouseOver ( )
```

Remarks

The return value corresponds to the value set in the “*MouseOver menu image*”, for the active node language, in the “Create or Edit node” pages of “eQuercus - Website and content management”.

If the field does not contain a value, the address in the “*Active menu image*” field will be returned. And if this field does not contain a value, the value in the “*Normal menu image*” will be returned.

CHAPTER 4

The ContentText object

The *ContentText* object is an interface to the *ADO Recordset* object related to records selected by the *Site.GetContent()* and *Site.GetListOfContents* methods. The property and methods of this object are used to navigate in the recordset and read “Text document” content type information.

From and *to* constructor parameters specify a *Recordset* subset, where:

- *from* indicates the first record of the subset, 0 based index.
- *to* indicates the last record of the subset, 0 based index.

All the methods and properties of this object are related to the subset and not the *ADO Recordset* type object sent to the constructor.

If the object is returned by the *Site.GetContent()* or *Site.GetListOfContents* methods, the above is not relevant for the user.

EOF property

Description

Returns *true* if the position of the active record is higher than the last record of the recordset subset, otherwise *false*.

Syntax

`oContentText.EOF`

BOF property

Description

Returns *true* if the position of the active record is lower than the first record of the recordset subset, otherwise *false*.

Syntax

`oContentText.BOF`

RecordCount property

Description

Returns the numeric value corresponding to the record number of the recordset subset.

Syntax

`oContentText.RecordCount ()`

RecordPos property

Description

Returns the numeric value corresponding to position of the active record in relation to the recordset subset. The first record is in position 0.

Syntax

`oContentText.RecordPos ()`

MoveFirst method

Description

Moves the active record to the first position of the recordset subset.

Syntax

`oContentText.MoveFirst ()`

MoveLast method

Description

Moves the active record to the last position of the recordset subset.

Syntax

`oContentText.MoveLast ()`

Move method

Description

Moves the active record to the specified position of the recordset subset. 0 based index.

Syntax

`oContentText.Move (pos)`

MoveNext method

Description

Moves the active record to the next position.

Syntax

`oContentText.MoveNext ()`

MovePrev method

Description

Moves the active record to the previous position.

Syntax

`oContentText.MovePrev ()`

GetId method

Description

Returns a numeric value corresponding to the content id.

Syntax

`oContentText.GetId ()`

GetTitle method

Description

Returns a string containing the text entered in the content “*Title*” field.

Syntax

```
oContentText.GetTitle()
```

GetSubTitle method

Description

Returns a string containing the text entered in the content “*Subtitle*” field.

Syntax

```
oContentText.GetSubTitle()
```

GetPicture1 method

Description

Returns a string containing the text entered in the content “*Image 1*” field.

Syntax

```
oContentText.GetPicture1()
```

GetPicture2 method

Description

Returns a string containing the text entered in the content “*Image 2*” field.

Syntax

```
oContentText.GetPicture2()
```

GetLink method

Description

Returns a string containing the text entered in the content “*Link*” field.

Syntax

```
oContentText.GetLink()
```

GetLinkDescription method

Description

Returns a string containing the text entered in the content “*Link description*” field.

Syntax

```
oContentText.GetLinkDescription()
```

GetBody method

Description

Returns a string containing the text entered in the content “*Text*” field.

Syntax

```
oContentText.GetBody( )
```

GetCreationDate method

Description

Returns a string containing the content creation date.

Syntax

```
oContentText.GetCreationDate( )
```

Remarks

The date format depends on the configuration in the global.asa file.

GetModificationDate method

Description

Returns a string containing the date of the last content modification.

Syntax

```
oContentText.GetModificationDate( )
```

Remarks

The date format depends on the configuration in the global.asa file.

CHAPTER 5

The ContentHTML object

The *ContentHTML* is the interface to the *ADO Recordset* related to records selected by the *Site.GetContent()* and *Site.GetListOfContents* methods. The property and methods of this object are used to navigate in the recordset and read “HTML document” content type information.

From and *to* constructor parameters specify a *Recordset* subset, where:

- *from* indicates the first record of the subset, 0 based index.
- *to* indicates the last record of the subset, 0 based index.

All the methods and properties of this object are related to the subset and not the *ADO Recordset* type object sent to the constructor.

If the object is returned by the *Site.GetContent()* or *Site.GetListOfContents* methods, the above is not relevant for the user.

EOF property

Description

Returns *true* if the position of the active record is higher than the last record of the recordset subset, otherwise *false*.

Syntax

`oContentText.EOF`

BOF property

Description

Returns *true* if the position of the active record is lower than the first record of the recordset subset, otherwise *false*.

Syntax

`oContentText.BOF`

RecordCount property

Description

Returns the numeric value corresponding to the record number of the recordset subset.

Syntax

`oContentText.RecordCount()`

RecordPos property

Description

Returns the numeric value corresponding to position of the active record in relation to the recordset subset. The first record is in position 0.

Syntax

`oContentText.RecordPos()`

MoveFirst method

Description

Moves the active record to the first position of the recordset subset.

Syntax

```
oContentText.MoveFirst()
```

MoveLast method

Description

Moves the active record to the last position of the recordset subset.

Syntax

```
oContentText.MoveLast()
```

Move method

Description

Moves the active record to the specified position of the recordset subset. 0 based index.

Syntax

```
oContentText.Move(pos)
```

MoveNext method

Description

Moves the active record to the next position.

Syntax

```
oContentText.MoveNext()
```

MovePrev method

Description

Moves the active record to the previous position.

Syntax

```
oContentText.MovePrev()
```

GetId method

Description

Returns a numeric value corresponding to the content id.

Syntax

```
oContentHTML.GetId()
```

GetTitle method

Description

Returns a string containing the text entered in the content “*Title*” field.

Syntax

```
oContentHTML.GetTitle()
```

Remarks

In eQuercus the “Title” field can only be changed if the content is connected to a node with more than one content.

GetBody method

Description

Returns a string containing the text entered in the content HTML editor field.

Syntax

```
oContentHTML.GetBody()
```

GetCreationDate method

Description

Returns a string containing the content creation date.

Syntax

```
oContentHTML.GetCreationDate()
```

Remarks

The date format depends on the configuration in the global.asa file.

GetModificationDate method

Description

Returns a string containing the date of the last content modification.

Syntax

```
oContentHTML.GetModificationDate()
```

Remarks

The date format depends on the configuration in the global.asa file.

CHAPTER 6

The Language object

The *Language* object contains the properties which contain information related to a language.

Id property

Description

The numeric type property corresponding to the language id.

Syntax

`oLanguage.id`

Name property

Description

The string type property corresponding to the language name (English, Español, Français, Deutsch or Italiano).

Syntax

`oLanguage.name`

Short property

Description

The string type property corresponding to the abbreviated language name (E, S, F, D or I).

Syntax

`oLanguage.short`