.NET API Reference

QP7.Framework

Table of Contents

API

Invoking Objects

void ShowObject(string [template_name.]object_name[.format_name]) –
The specified object is added to the end of the Controls Collection of the current object. This method can be called from any object or template. Parameters [template_name] and [format_name] are optional. This method can be called only from Code Behind.

void ShowObject(string [template_name.]object_name[.format_name], System.Object sender) – The specified object (first argument) is added to the end of the Controls Collection of the control passed in as the second parameter. The second argument has to be belong to the Control class.

Example:
C#:
ShowObject("Main.Menu", this);

VB.NET:
ShowObject("Menu", Me)

string GetInternalCall(string [template_name.]object_name[.format_name]) – Returns relative path to control which can be used for control loading.

<qp:placeholder calls="[template_name.]object_name[.format_name]" runat="server" /> - Invokes object from Presentation Layer.

The following methods and calls are out-of-date and not recommended to use:

- GetObjectFullName (Use GetInternalCall instead)
- ShowControl (use ShowObject instead. Now it supports dynamic calls)
- ShowObjectNS (OnScreen for objects is not supported any longer)
- qp:placeholderns (OnScreen for objects is not supported any longer)

Publishing Container

Invoking Fields

string Field(DataRowView pDataItem, string key) – Returns value for field key from the current DataRowView.

string Field(DataRowView pDataItem, string key, string defaultvalue) - Returns value for field key from the current DataRowView. If returned value is not set or null returns the defaultvalue (third parameter)

string Field(DataRow pDataItem, string key)
and
string Field(DataRow pDataItem, string key, string defaultvalue) – Analogous to the previous methods with an exception of using DataRow instead of DataRowView.

Example for the Presentation Layer:
C#:
<%# Field(((DataRowView)(Container.DataItem)), "Title")%>
VB.NET:
<%# Field(CType(Container.DataItem, DataRowView), "Title")%>

Example for Code Behind:
C#:
Field(Data.Rows[e.Item.ItemIndex], "Title");
VB.NET:
Field(Data.Rows(e.Item.ItemIndex), "Title", "DefaultTitle")

string FieldNS(DataRowView pDataItem, string key)
string FieldNS(DataRowView pDataItem, string key, string defaultvalue)
string FieldNS(DataRow pDataItem, string key)
string FieldNS(DataRow pDataItem, string key, string defaultvalue)

- Analogous to Field methods with disabled "OnScreen" support while in "OnScreen" mode.

NOTE: The methods above are also accessible from object of "generic" type.

Additional Properties and Methods

NOTE: The following Methods are only accessible from inside of objects of type "Publishing Container"

DataTable Data – Returns DataTable used by the object.

long TotalRecords – Returns total number of articles from the current content table specified by the paging options (minus the ones eliminated by the filter).

long AbsoluteTotalRecords – Returns total number articles from the current content table (minus the ones eliminated by the filter).

long ContentID – Returns id of the current content table.

string ContentName – Returns the name of the current content table.

string ContentUploadURL – Returns the path to the files belonging to the current content.

Example:
<img src="<%#ContentUploadURL%>/<%#Field(CType(Container.DataItem, DataRowView), "Product Picture")%>">

long RecordsPerPage - Returns total number of articles specified by the paging options.

string GetFieldUploadUrl(string fieldName) – For virtual content returns ContentUploadURL by utilizing the name of the field.

Page Properties

int page_id – Returns page id

int page_template_id – Returns page_template_id

int site_id – Returns site id

string upload_url – Returns URL to the library files/folders without the trailing slash. The Property is accessible within any page or object.

Example:
<img src="<%# upload_url%>/my_photo.jpg">

string site_url – Returns site main folder URL, that is relative to the site root (This method produces different results for live and stage assembling mode). For getting absolute URL use GetSiteUrl method)

bool IsStage – Indicates page assembling mode (live or stage).

DBConnector Cnn – Returns a reference to the QP7 database connection object. The object is recommended to be used by only very experienced programmers. Usually, it's enough to use standard functionality of Publishing Container and Publishing Form objects.

Methods:
DataTable GetData(string strQuery) – Returns object of type DataTable for a given sql query. Usually used for complex queries.
void ProcessData(string strQuery) – Executes sql command, usually used for Insert, Update, Delete.

Format Properties (of an object)

Note: All methods and properties described above are also accessible from all object types.

int format_id – Returns unique format id.

Form Methods

void RemoveContentItem(int content_item_id) - Deletes article for a given article id specified by the content_item_id argument.

void DeleteContentItem() - Deletes article for a given article id, which needs to be specified by creating "content_item_id" key/value pair in the global Values collection beforehand. (AddValue method is used to create key/value pairs)

string FieldName(string content_name, string field_name) – Returns internal field name as used by the QP7 engine. Usually, the method is used for naming HTML fields in conjunction with AddFormToContent or UpdateContentItem.

int FieldID(string content_name, string field_name) – Returns the field id for a field specified by the field_name argument.

int AddFormToContent(string content_name, string status_name) – Adds a new article with a given status specified by the second parameter status_name for a given content specified by the first parameter content_name. Also, creates a global collection item new_content_item_id with a value equaled to the unique id of the newly created article. The new id can be read by using - Value("new_content_item_id").

Article field values can be specified via an HTML form, query string or with a help of AddValue method. AddValue method has to create keys with the following structure (used by QP7 engine internally) - field_<field_id>). In turn, field_<field_id> can be specified by using method FieldName.

Files for fields of type image or file are uploaded into Content folder (determined by content_name). If file or image with a given name already exists then a number is appended to the end of the file name, e.g. myFile[1].jpg.

Example:
AddValue(FieldName("Users", "Login"), Login);
AddValue(FieldName("Users", "Password"), Password);
AddFormToContent("Users", "Published");

int AddFormToContent(string content_name, string status_name, int content_item_id) – Similar to the previous method AddFormToContent(string content_name, string status_name), but in this case an article identified by content_item_id is modified rather than being newly created.

int AddFormToContentWithoutNotification(string content_name, string status_name) - Similar to AddFormToContent(string content_name, string status_name) except that existing notifications are not executed when article is created.

int AddFormToContentWithoutNotification(string content_name, string status_name, int content_item_id) – Similar to AddFormToContent(string content_name, string status_name, int content_item_id) except that existing notifications are not executed when article is created.

void UpdateContentItemField(string content_name, string field_name, int content_item_id, bool with_notification) – Updates an article field. If with_notification is true then notifications are processed. Before calling the method the field value must be entered into the global collection Values via AddValue, query string or form submit.

Files for fields of type image or file are uploaded into Content folder (determined by content_name). If file or image with a given name already exists then a number is appended to the end of the file name, e.g. myFile[1].jpg.

Example:
AddValue(FieldName("Users", "Login"), "newLogin");
UpdateContentItemField("Users", "Login", Value("cid"), false);

void UpdateContentItemField(string content_name, string field_name, int content_item_id) – Similar to UpdateContentItemField(string content_name, string field_name, int content_item_id, bool with_notification) except for the last parameter which is assumed to be false (with_notification = false).

void UpdateContentItem() – Works similar to AddFormToContent(string content_name, string status_name), but updates an article. The method is not using any parameters but expects to use Value("content_item_id") which must be added to the Values collection before calling the method. The method also requires the field values to be present in the Values global collection (via form/url submission or by calling AddValue).

void UpdateContentItemWithoutNotification() - Similar to UpdateContentItem() except that existing notifications are not executed when article is updated.

void UpdateContentItem(bool updateEmpty, string statusName) – This overloaded version of the method. updateEmpty parameter switches article updating mode. When updateEmpty = true all the fields of the content will be nullified if no appropriate data has been provided. This is the behavior by default. To the contrary when updateEmpty = false, only non-empty fields will be updated. StatusName parameter allows changing the status of the article. If you don't want to change the status just pass empty string.

Example:
AddValue("content_item_id", 12345);
AddValue(FieldName("Users", "Login"), Login);
AddValue(FieldName("Users", "Password"), Password);
UpdateContentItem();

int AddUpdateContentItemLink(string LinkFieldName, int ItemID, string LinkItems, string TargetLinkItems) – Adds or updates many-to-many relation fields for a given article id. Returns 1 if successful, -1 if failed.

LinkFieldName – name of many-to-many relation field for which update should be performed.
ItemID – article id for which relation links will be created / updated.
LinkItems – list of article id's from the related content that would be associated with the article specified by ItemID.
TargetLinkItems – list of article id's from the related content for which the update will be performed. If this parameter is empty then all of associations to related articles are removed and new ones are created by using LinkItems parameter. If it is specified then only associations contained within TargetLinkItems are removed and new ones are created from LinkItems.

Example:
AddUpdateContentItemLink("Books", NumValue("AuthorID"), Value("12345,45678,7890, 23415, 8907"), "")

Values (global collection) Methods

void AddValue(string key, Object value) – Creates a key/value pair in the global Values collection. Used only in Code Behind. The key/value pairs can be retrieved via Value(key), StrValue(key), NumValue(key), DirtyValue(key).
Example:
AddValue("Age", 35);

<qp:AddValue key="key" value="value" runat="server" /> - Creates a key/value pair in the global Values collection (HashTable) in the Presentation Layer. It's not recommended to use this technique inside Publishing Container objects.

Example:
<qp:AddValue key="Age" value="35" runat="server" />

string Value(string key) - Returns the value of a global collection item for a given key. This method replaces the use of the ASP Request object (Request.Form, Request.QueryString). The items in the global collection are added by reading values posted to the server via GET or POST. The items could also be added via AddValue method or through specifying default values in an "Object Parameters" section. For security purposes (sql injection attacks) the returned value is stripped out of single quotes. Use DirtyValue to read original values (single quotes intact).

string DirtyValue(string key) – Similar to Value(string key). Returns the value of a global collection item for a given key leaving single quotes intact.

long NumValue(string key) - Similar to Value(string key). Returns the value of a global collection item for a given key converted to a number. If conversion to a number fails, 0 is returned. It's recommended to use this method when specifying filters that contain numeric Value inside "Object Parameters" of Publishing Container objects.

Example:
"[content_item_id] = " & NumValue("id")

string StrValue(string key) - Similar to Value(string key). Returns the value of a global collection item for a given key converted to a string. Single quotes are replaced by double quotes. It's recommended to use this method when specifying filters inside "Object Parameters" of Publishing Container objects.

Example:
"[text] = '" & StrValue("text") & "'"

string Value(string key, string defaultValue) - Similar to Value(string key). Returns the value of a global collection item for a given key. If no value is found for the key argument, defaultValue is returned.

Hashtable Values – Returns the global collection hashtable for the current page.

Working with Contents

int GetContentID(string content_name) – Returns the unique id for a given content specified by the content_name argument.

string GetContentUploadUrl(string content_name) – Returns upload url to content files for a given content by content name.

string GetContentUploadUrlByID(int content_id) - Returns upload url to content files for a given content by content id.

string GetContentItemLinkIDs(string linkFieldName, long itemID) – Returns a comma-delimited list of article id's associated to a given article id (itemID) for a given many-to-many relation field (LinkFieldName)

Example:
GetContentItemLinkIDs("Authors", Field(Data.Rows[e.Item.ItemIndex], "content_item_id"));

string GetContentItemLinkIDs(string linkFieldName, string itemID) – Same as the previous method but itemID could be a comma-delimited list of article id's

string GetContentItemLinkQuery(string linkFieldName, string itemID) – Similar to GetContentItemLinkIDs. The difference is that current method doesn't execute the resulting SQL-query, but returns it as text.

string GetLinkIDs(string LinkFieldName) – Only for backward compatibility with "ASP.NET like ASP" type of assembly. It's not recommended to use this method. Works only in objects of type "Publishing Container", same as GetContentItemLinkIDs(string linkFieldName, string itemID) but assumes a presence of a field collection with "content_item_id".

string GetContentItemLinksFilter(int itemID, string contentName) – Deprecated. GetContentItemLinkIDs replaced this method.

DataTable GetContentData(
string siteName,
string contentName,
string whereExpression,
string orderExpression,
long startRow,
long pageSize,
ref long totalRecords,
byte useSchedule,
string statusName,
byte showSplittedArticle,
byte includeArchive)

- Returns DataTable with a list of articles from a given content using parameters from "Object Parameters" of Publishing Container. Enables non-QP7 components to receive data from QP7 CMS.

siteName – site name, contentName – content name, whereExpression – filter / where clause, orderExpression - order by expression, startRow – specifies which row will become the first row in DataTable (all rows before it are not returned, used for paging), pageSize number or records to be returned, totalRecords - returns total number of records that would be returned by applying only filter (other parameters are not used), useSchedule – specifies whether or not to return currently invisible articles, statusName – specifies article status, showSplittedArticle – whether or not to show "split" articles (version that is currently in a workflow in the backend), includeArchive – specifies whether or not to show archived articles.

OnScreen Methods

string OnFly(DataRowView pDataItem, string key)
string OnFly(DataRowView pDataItem, string key, string defaultvalue)
string OnFly(DataRow pDataItem, string key)
string OnFly(DataRow pDataItem, string key, string defaultvalue)

Deprecated. It is recommended to use analogous Field methods.

string OnScreenFlyEdit(string Value, Int32 ItemID, string FieldName) – This method can be used to organize work with OnScreen from the outside of the Publishing Containers. In the other case use the standard Field method. The parameters include: text to be OnScreen enabled (Value), article id (ItemID) and field name (FieldName).

string OnScreenFlyEdit(string Value, string ItemID, string FieldName) - Same as OnScreenFlyEdit(string Value, Int32 ItemID, string FieldName), second parameter is a string.

string OnScreen(string Value, Int32 ItemID) – The simplified version of OnScreenFlyEdit method. With this method user can edit the article only in Form View, but not in OnFly mode. It's only recommended to use this method with QP7-managed articles.

string OnScreen(string Value, string ItemID) - Same as OnScreen(string Value, Int32 ItemID), second parameter is a string.

Page Initialize Methods

Sometimes it might be necessary to inherit from QP7 page class QPage (or QMobilePage). For those cases QP7 page needs to be initialized in order for it to work correctly (especially if you'll be using QP7-generated controls). There are three overloaded versions of the single method that are provided in the framework now.

void Initialize(int siteId)
This overloaded version gives a base access to the QPage (QMobilePage) instance methods. QP7 generated controls are not supported.

void Initialize(int siteId, int pageId, int pageTemplateId, string pageFileName, string pageFolder)
This is the most common version of the Initialize method. QP7 generated controls are fully supported.

void Initialize(int siteId, int pageId, int pageTemplateId, string uploadUrl, string uploadUrlPrefix, string siteUrl, string pageFileName, string pageFolder)
This version is similar to the previous one, but allows redefining the global URLs.

Parameters:

siteId – id of QP7 website. It is used to get specific data from database

pageId – id of QP7 page. It is used to get specific data from database and support QP7 generated controls

pageTemplateId – id of QP7 page template. It is used to get specific data from database and support QP7 generated controls

uploadUrl – upload URL used by QP7 website (see "Site Properties" in the backend)

uploadUrlPrefix – absolute upload URL prefix used by QP7 website (see "Site Properties" in the backend)

siteUrl –website URL, will be used to calculate paths to QP7 controls and also for QP7-managed links.

pageFileName – name of the page file (e.g. default.aspx). This will be used to calculate the name of the folder where QP7 controls are located. A file name default.aspx will be converted to "page_controls__default_aspx".

pageFolder – parameter is used when current page and its controls are located in their own directory.

The following overloaded versions of Initialize method are deprecated:

void Initialize(int siteId, string uploadUrl, string siteUrl, string pageFileName, string templateNetName)

void Initialize(int siteId, string uploadUrl, string siteUrl, string pageFileName, string templateNetName, Hashtable pageObjects)

void Initialize(int siteId, string uploadUrl, string siteUrl, string pageFileName, string templateNetName, Hashtable pageObjects, Hashtable templates)

Permissions Class

The following methods pertain to Quantumart.QPublishing.Permissions class. Since the methods are almost self-explanatory, minimal definition is given.

int AuthenticateUser(string username, string password) – returns user_id or 0 (zero) if user is not found or incorrect credentials.

DataTable GetUserInfo(int user_id) – returns DataTable with fields: user_id, login, password, disabled, first_name, last_name, email, created, modified.

DataTable GetUserInfo(string login) – returns DataTable with fields: user_id, login, password, disabled, first_name, last_name, email, created, modified.

int AddUser(string username, string password, string First_Name, string Last_Name, string Email) – adds new user and returns new user_id.

void UpdateUser(int userId,
string newUserName,
string newPassword,
string newFirst_Name,
string newLast_Name,
string newEmail
) – updates information for a given user_id.

bool RemoveUser(int userId) – deletes user from the system.

DataTable GetGroupInfo(int group_id) – returns information for a particular group; the returned fieds are: group_id, group_name, created, modified**.**

DataTable GetGroupInfo(string group_name) – returns information for a particular group; the returned fieds are: group_id, group_name, created, modified.

int AddGroup(string name) – creates new group and returns new group_id.

int AddGroup(string name, bool AllowSharedOwnershipOfItem) – creates new group and returns new group_id; AllowSharedOwnershipOfItem – refers to the backend only option that allows users that belong to the same group to share user's permissions when a new article is created.

void UpdateGroup(int groupId, string newName) – updates group name for a given groupId.

void UpdateGroup(int groupId, string newName, bool AllowSharedOwnershipOfItems) – updates group name and AllowSharedOwnershipOfItems for a given groupId; AllowSharedOwnershipOfItem – refers to the backend only option that allows users that belong to the same group to share user's permissions when a new article is created.

void RemoveGroup(int groupId) – removes group from the system.

DataTable GetAllGroups()– returns a list of all groups; returned fields: group_id, group_name, created, modified.

DataTable GetChildParentGroups() – returns a list of groups with parent-child associations; returned fields: child_group_id, child_group_name, parent_group_id, parent_group_name.
void AddChildGroupToParentGroup(int parent_group_id, int child_group_id) – adds a specified group as a child (child_group_id) of another group (parent_group_id).

void RemoveChildGroupFromParentGroup(int parent_group_id, int child_group_id) – removes a specified group as a child (child_group_id) of another group (parent_group_id)

DataTable GetRootGroupsForUser(int user_id) – returns a list of group_id's for a specified user (returns only groups to which user is associated directly).

DataTable GetUsersForGroup(int group_id) – returns a list of users for a specified group; returned fields: user_id, login, password, disabled, first_name, last_name, email, created, modified.

void AddUserToGroup(int userId, int groupId) – adds user to a group.

void RemoveUserFromGroup(int userId, int groupId) – removes user from a group

void MoveUsersFromGroupToGroup(int fromGroupId, int toGroupId) – moves user from one group to another.

void CopyUsersFromGroupToGroup(int fromGroupId, int toGroupId) – copies users from group to another.

DataTable GetAllGroupsForItemPermission(int itemId) – returns a list of groups directly listed on the permissions of a given content item (article); returned fields: group_id, group_name, created, modified.

DataTable GetAllUsersForItemPermission(int itemId) – returns a list of users directly listed on the permissions of a given content item (article); returned fields: user_id, login, password, disabled, first_name, last_name, email, created, modified.

void RemoveAllUsersFromItemPermission(int itemId) – removes all users directly listed on the permissions of a given content item (article).

void RemoveAllGroupsFromItemPermission(int itemId) – removes all users directly listed on the permissions of a given content item (article).

void RemoveAllEntitiesFromItemPermission(int itemId) – removes all users and all groups directly listed on the permissions of a given content item (article).

void AddUserToItemPermission(
int userId,
int itemId,
int permissionId) – adds user to the permissions of a given content item (article).

void AddGroupToItemPermission(
int groupId,

int itemId,
int permissionId) – adds group to the permissions of a given content item (article).

void RemoveUserFromItemPermission(int userId, int itemId) – removes a given user who is directly listed on the permissions of a given content item (article).

void UpdateUserItemPermission(
int userId,
int itemId,
int permissionId) – updates permissions for a given user who is directly listed on the permissions of a given content item (article).
void RemoveGroupFromItemPermission(int groupId, int itemId) – removes a given group that is directly listed on the permissions of a given content item (article).

void UpdateGroupItemPermission(
int groupId,
int itemId,
int permissionId) – updates permissions for a given group that is directly listed on the permissions of a given content item (article).

DataTable GetAllGroupsForContentPermission(int contentId) – returns a list of groups directly listed on the permissions of a given content area (content); returned fields: group_id, group_name, created, modified.

DataTable GetAllUsersForContentPermission(int contentId) – returns a list of users directly listed on the permissions of a given content area (content); returned fields: user_id, login, password, disabled, first_name, last_name, email, created, modified.

void RemoveAllUsersFromContentPermission(int contentId) – removes all users directly listed on the permissions of a given content area (content).

void RemoveAllGroupsFromContentPermission(int contentId) – removes all groups directly listed on the permissions of a given content area (content).

void RemoveAllEntitiesFromContentPermission(int contentId) – removes all users and groups directly listed on the permissions of a given content area (content).

void AddUserToContentPermission(
int userId,
int contentId,
int permissionId
) – adds user to the permissions of a given content area (content).

void AddUserToContentPermission(
int userId,
int contentId,
int permissionId,
bool propagateToItems
) – adds user to the permissions of a given content area (content) with backend-only option – propagateToItems, which propagates permissions to items belonging to the specified content area when new items are added.

void AddGroupToContentPermission(
int groupId,
int contentId,

int permissionId
) – adds group to the permissions of a given content area (content).

void AddGroupToContentPermission(
int groupId,
int contentId,
int permissionId,
bool propagateToItems) – adds group to the permissions of a given content area (content) with backend-only option propagateToItems - propagates permissions to items belonging to the specified content area when new items are added.
.

void RemoveUserFromContentPermission(int userId, int contentId) – removes a given user who is directly listed on the permissions of a given content area (content).

void UpdateUserContentPermission(
int userId,
int contentId,
int permissionId,
bool propagateToItems) – updates permissions for a given user who is directly listed on the permissions of a given content area (content); propagateToItems - propagates permissions to items belonging to the specified content area when new items are added.

void RemoveGroupFromContentPermission(int groupId, int contentId) – removes a given group that is directly listed on the permissions of a given content area (content).

void UpdateGroupContentPermission(
int groupId,
int contentId,
int permissionId
) – updates permissions for a given group that is directly listed on the permissions of a given content area (content).

void UpdateGroupContentPermission(
int groupId,
int contentId,
int permissionId,
bool propagateToItems) – updates permissions for a given group that is directly listed on the permissions of a given content area (content); propagateToItems - propagates permissions to items belonging to the specified content area when new items are added.

DataTable GetPermissionLevels() – returns a DataTable with a list of permission levels and permission ids. Permission_level_id is needed to be used for methods like AddUserToItemPermission() and many others; returned fields: permission_level_id, permission_level, permission_level_name

Miscellaneous Methods

string ReplaceHTML(string str) – Replaces "<" and ">" to "<" and ">"

void SendNotification(string notification_on, int content_item_id, string notification_email) – Sends email notification for a given content by article id. Email Notifications have to be already created in the "Notifications" section of the backend. The parameters include article id which may be used to create email body, type of event and an optional email address. If notification_email is not specified then the system would use the email addresses specified in the Notificaitons.

Types of notification events (notification_on):
for_create – when new article is created
for_modify – when article is edited
for_remove – when article is deleted.

Example:
SendNotification("for_remove", content_item_id, "");

string GetSiteUrl() – Returns absolute URL for site main folder (This method produces different results depending of page assembling mode).