Macro expressions allow you to use a large variety of methods. The following categories list the methods according to the type of provided functionality:

- [Data manipulation](#)
- [Data conversion](#)
- [Text manipulation](#)
- [Advanced text processing](#)
- [Date and time modification](#)
- [Mathematical operations](#)
- [E-commerce](#)
  - [Basic product methods](#)
  - [Related methods](#)
  - [URLs](#)
  - [Google tag manager e-commerce data](#)
- [User membership and permissions](#)
- [Transformations](#)

> ℹ️ Keep in mind that the recommended macro syntax is to use infix notation for the first parameter of methods.
>
> For example: *{% "word".ToUpper() %}* instead of *{% ToUpper("word") %}*

Data manipulation

Method	Return type	Parameters	Description
GetValue	object	- ISimpleDataContainer container - string column - *object defaultValue*	Gets the value of the specified data column of an object that implements the ISimpleDataContainer interface. You can specify a default value used if the requested data is null. Example: *{% CurrentDocument.GetValue("NewsTitle") %}*
GetProperty	object	- IHierarchicalObject object - string property - *object defaultValue*	Gets the value of the specified property of an IHierarchicalObject object. You can specify a default value used if the requested property is null. Example: *{% CurrentDocument.GetProperty("NewsTitle") %}*
GetItem	object	- IEnumerable collection - int index - *object defaultValue*	Gets the object at the specified index of a collection. You can specify a default value used if the requested object is null. Example: *{% GlobalObjects.Users.GetItem(0).UserName %}*

OrderBy	IInfoObject Collection	• IInfoObjectCollection collection • string orderBy	Returns the collection of objects, with the order defined by the specified SQL ORDER BY clause. If the macro is saved by a user without the Administrator privilege level, the method only allows the following SQL syntax: • one or more column names (separated by commas) • the ASC and DESC keywords Example: *{% GlobalObjects.Users.OrderBy("UserCreated DESC").FirstItem.UserName %}*
Where	IInfoObject Collection	• IInfoObjectCollection collection • string where	Filters a collection of objects according to the specified SQL Where condition. If the macro is saved by a user without the Administrator privilege level, the method only allows the following SQL syntax in the where condition: • column names, values and basic operators: =, !=, >, < • AND & OR operators, parentheses • *column* BETWEEN *value* AND *value* • *column* LIKE *value* • *column* IN *(values)* • *column* IS NULL • NOT keyword for the above expressions (NOT BETWEEN, NOT LIKE, NOT IN, IS NOT NULL) Other expressions and SQL functions are not supported. Example: *{% GlobalObjects.Users.Where("Email LIKE '%@localhost.local'").FirstItem.UserName %}*
TopN	IInfoObject Collection	• IInfoObjectCollection collection • int topn	Returns only the specified number of objects in the collection. Example: *{% GlobalObjects.Users.TopN(1).FirstItem.UserName %}*
Columns	IInfoObject Collection	• IInfoObjectCollection collection • string columns	Returns the collection of objects containing only the specified data columns. If the macro is saved by a user without the Administrator privilege level, the method only allows the following SQL syntax: • one or more column names (separated by commas) • SQL aliases for columns defined using the AS keyword (must be upper case) Example: *{% GlobalObjects.Users.Columns("UserName, Email").FirstItem.UserName %}*
Filter	IList	• IEnumerable collection • MacroExpression condition	Filters a collection of objects according to the specified macro condition. Example: *{% GlobalObjects.Users.Filter(UserID == 53).FirstItem.UserName %}*
ClassNames	TreeNodeCollection	• TreeNodeCollection collection • string classNames	Filters a TreeNodeCollection of pages according to the specified page types. The second parameter must contain a list of allowed page type code names, separated by semicolons. Example: *{% Documents.ClassNames("cms.menuitem;cms.news").Count %}*

InList	bool	• object object • IEnumerable collection	Returns a true value if the object exists within the specified collection. Example: *{% CurrentDocument.ClassName.InList("cms.menuitem,cms.root".Split(",")) %}*
All	bool	• IEnumerable collection • MacroExpression condition	Returns a true value if all of the objects in the collection match the given condition. Specify the condition as a macro expression. Example: *{% GlobalObjects.Users.All(UserEnabled == true) %}*
Any	bool	• IEnumerable collection • *MacroExpression condition*	Returns a true value if at least one object in the collection matches the given condition. Specify the condition as a macro expression. If you leave out the condition parameter, the method returns true if the collection contains at least one object. Throws an exception if the collection is null. Example: *{% GlobalObjects.Users.Any(UserEnabled == false) %}*
Exists	bool	• IEnumerable collection • *MacroExpression condition*	Returns a true value if at least one object in the collection matches the given condition. Specify the condition as a macro expression. If you leave out the condition parameter, the method returns true if the collection contains at least one object. Returns false if the collection is null. Example: *{% GlobalObjects.Users.Exists(UserEnabled == false) %}*
RandomSelection	IList	• IEnumerable items • *int numberOfItems*	Returns randomly selected objects from the collection. You can optionally specify the number of items. Example: *{% GlobalObjects.Users.RandomSelection().UserName %}*
SelectInterval	IList	• IEnumerable items • int lowerBound • int upperBound	Returns the specified interval of objects from the collection. Example: *{% Documents.SelectInterval(0,9) %}*
Cache	object	• object expression • *int cacheMinutes* • *bool condition* • *string cacheItemName* • *string cacheItemNameParts* • *CMSCacheDependency cacheDependency*	Evaluates the specified macro expressions and stores the result in the server-side application cache. The system only evaluates the expression if the result is not found in the cache. See also: Caching the results of macros

| GetCache Dependency | CMSCache Dependency | • string[] dependencies | Returns a CMSCacheDependency object based on the specified dummy cache keys (one or more string parameters).

See also:

• [Setting cache dependencies](#)
• [Caching the results of macros](#) |

[> Back to the list of macro method categories](#)

Data conversion

Method	Return type	Parameters	Description
ToString	string	• object value • *string defaultValue* • *string culture* • *string format*	Converts an object to a string. Returns the optional default value if the conversion is not possible. You can also specify the culture context and a formatting string used for the conversion. Examples: • *{% DocumentName.ToString() %}* • *{% DocumentName.ToString("defaultValue") %}* • *{% DocumentName.ToString("defaultValue", "en-US") %}* • *{% DocumentName.ToString("defaultValue", DocumentCulture, "Document name is: {0}") %}*
ToInt	int	• object value • *int defaultValue*	Converts an object to an integer number. Returns the optional default value if the conversion is not possible. Example: *{% DocumentID.ToInt() %}*
ToBool	bool	• object value • *bool defaultValue*	Converts an object to a boolean value. Returns the optional default value if the conversion is not possible. Example: *{% DocumentShowInSiteMap.ToBool() %}*
ToDecimal	decimal	• object value • *decimal defaultValue* • *string culture*	Converts an object to a decimal number. Returns the optional default value if the conversion is not possible. You can also specify a culture code to determine the formatting of the decimal number. Examples: • *{% DocumentID.ToDecimal() %}* • *{% DocumentID.ToDecimal("0.0", DocumentCulture) %}*

ToDouble	double	• object value • *double defaultValue* • *string culture*	Converts an object to a double number. Returns the optional default value if the conversion is not possible. You can also specify a culture code to determine the formatting of the double number. Examples: • *{% DocumentID.ToDouble() %}* • *{% DocumentID.ToDouble("0.0", DocumentCulture) %}*
ToGuid	guid	• object value • *Guid defaultValue*	Converts an object to a [GUID value](#). Returns the optional default value if the conversion is not possible. Example: *{% DocumentGUID.ToGuid() %}*
ToDateTime	DateTime	• object value • *DateTime defaultValue* • *string culture*	Converts an object to a DateTime value. Returns the optional default value if the conversion is not possible. You can also specify a culture code to determine the date format. Example: *{% ToDateTime("12/31/2017 11:59 PM") %}*
FromOADate	DateTime	• double value	Converts a double representation of a date and time value (OLE Automation Date) to a DateTime object. Example: *{% FromOADate(43100.999305556) %}*
ToTimeSpan	TimeSpan	• object value	Converts an object to a TimeSpan value. Returns null if the conversion is not possible. Example: *{% ToTimeSpan("1:00:00") %}*
ToBaseInfo	BaseInfo	• object value • *BaseInfo defaultValue*	Converts an object to a BaseInfo — the general type for Kentico system objects and pages. Returns the optional default value if the conversion is not possible. Example: *{% CurrentUser.ToBaseInfo() %}*
List	ArrayList	• object items	Converts a list of objects to an ArrayList. Example: *{% List("Apple", "Orange", "Banana") %}*

> Back to the list of macro method categories

Text manipulation

Method	Return type	Parameters	Description

Contains	bool	• string text • string search	Returns a true value if the string specified by the second parameter occurs within the first string.
EndsWith	bool	• string text • string findText	Indicates whether the string specified by the second parameter occurs at the end of the first string.
Format	string	• string value • string format	Replaces all formatting expressions in a string using the text equivalents specified in the second parameter. Based on composite formatting.
FormatNotEmpty	string	• string value • string format • *string emptyResult*	If the first parameter is not empty or null, replaces all formatting expressions in the string using the text equivalents specified in the second parameter. Returns the optional third parameter if the value is null or empty. Based on composite formatting.
GetMatch	string	• string text • string regex	Matches the string value to the specified regular expression and returns the match.
IndexOf	int	• string text • string searchFor	Returns the index of the first occurrence of the second string within the first string.
LastIndexOf	int	• string text • string searchFor	Returns the index of the last occurrence of the second string within the first string.

LimitLength	string	- string text - int length *string padString* *bool wholeWords*	Limits the length of the string to the specified number of characters. The *padString* parameter allows you to set a string that the method appends to the end of the limited result. The length of the parameter is included in the maximum length. If you set the *wholeWords* parameter to true, the method preserves the last word in the string.
LoremIpsum	string	*int length*	Generates [lorem Ipsum](#) text. You can specify the number of generated characters (1000 by default).
Matches	bool	- string text - string regex	Indicates whether the string matches the specified regular expression.
NotContains	bool	- string text - string search	Returns a true value if the string specified by the second parameter does NOT occur within the first string.
PadLeft	string	- string text - int length *string paddingString*	Adds leading characters to the string until the total length matches the second parameter. By default, the method uses spaces as the padding character, but you can optionally specify a different character.
PadRight	string	- string text - int length *string paddingString*	Adds trailing characters to the string until the total length matches the second parameter. By default, the method uses spaces as the padding character, but you can optionally specify a different character.
RegexReplace	string	- string text - string regex - string replacement	Replaces strings that match the regular expression pattern within the first string using the replacement string.

Remove	string	• string text • int position • *int length*	Deletes characters from the string, starting at the specified index. By default, the method removes all remaining characters from the specified position. You can optionally specify the number of characters to be removed.
Replace	string	• string text • string replace • string replacement	Replaces all occurrences of the second string within the first string using the replacement string.
Split	string[]	• string text • string delimiters • *bool removeEmpty*	Separates the string into an array of substrings according to the specified delimiting characters. If the optional third parameter is true, the method removes empty substrings from the result.
StartsWith	bool	• string text • string findText	Indicates whether the string specified by the second parameter occurs at the start of the first string.
Substring	string	• string text • int index • *int length*	Returns a substring starting at the specified index. By default, the method returns all remaining characters from the specified index. You can optionally specify the number of characters to be returned.
ToLower	string	• string text	Converts the entire string to lower case.
ToUpper	string	• string text	Converts the entire string to upper case.
Trim	string	• string text • *string charsToTrim*	Removes all leading and trailing occurrences of characters from the string. By default, the method trims white space characters, but you can optionally specify a different set of characters (as a string).

TrimEnd	string	• string text • *string charsToTrim*	Removes all trailing occurrences of characters from the string. By default, the method trims white space characters, but you can optionally specify a different set of characters (as a string).
TrimStart	string	• string text • *string charsToTrim*	Removes all leading occurrences of characters from the string. By default, the method trims white space characters, but you can optionally specify a different set of characters (as a string).

> ✓ **Tips**: To perform lexicographical comparison of strings, use the basic comparison operators (==, <, >, <=, >=). Add [macro parameters](#) to specify case sensitivity and the culture context of text operations.

[> Back to the list of macro method categories](#)

Advanced text processing

Method	Return type	Parameters	Description
GetResourceString	string	• string resourceStringKey • *string culture*	Translates the specified [resource (localization) string](#). By default, the target language of the localization depends on the current culture. You can specify the target language using the optional parameter.
JSEscape	string	• string text	Escapes the string for safe usage in JavaScript. Adds the \ escape character before special characters used by JavaScript (quotes, etc.). Note: The *JSEscape* method is not available when resolving macros in an external application using the *Kentico.Libraries* NuGet package.
Localize	string	• string inputText • *string culture*	Resolves [localization expressions](#) within the specified text. By default, the target language of the localization depends on the current culture. You can specify the target language using the optional parameter.
MapPath	string	• string path	Returns the physical file path that matches the specified virtual path.
ResolveBBCode	string	• string text	Resolves BBCode tags in the specified text, for example: *{% ResolveBBCode("[quote]Sample text[/quote]") %}*.
ResolveMacroExpressions	string	• string expression	Resolves the specified macro expression (without the *{% %}* parentheses).

ResolveMacros	string	• string inputText	Resolves all macros within the specified text.
ResolveUrl	string	• string url	Converts relative URLs starting with ~/ to relative URLs containing the current application path (so they can be resolved by clients). Absolute URLs remain unchanged.
StripTags	string	• string text	Removes all HTML tags from the specified text.
SQLEscape	string	• string text	Escapes the string for safe usage in SQL commands (to avoid SQL injection).
UnresolveUrl	string	• string url	Converts relative URLs containing the application path to relative URLs starting with ~/. Absolute URLs remain unchanged.
UrlEncode	string	• string url	Applies URL encoding to the specified string.

> Back to the list of macro method categories

Date and time modification

Method	Return type	Parameters	Description
AddMilliseconds	DateTime	• DateTime datetime • int milliseconds	Adds the specified number of milliseconds to a DateTime value.
AddSeconds	DateTime	• DateTime datetime • int seconds	Adds the specified number of seconds to a DateTime value.
AddMinutes	DateTime	• DateTime datetime • int minutes	Adds the specified number of minutes to a DateTime value.
AddHours	DateTime	• DateTime datetime • int hours	Adds the specified number of hours to a DateTime value.
AddDays	DateTime	• DateTime datetime • int days	Adds the specified number of days to a DateTime value.

AddWeeks	DateTime	• DateTime datetime • int weeks	Adds the specified number of weeks to a DateTime value.
AddMonths	DateTime	• DateTime datetime • int months	Adds the specified number of months to a DateTime value.
AddYears	DateTime	• DateTime datetime • int years	Adds the specified number of years to a DateTime value.
ToShortDateString	string	• DateTime datetime	Converts the value of the DateTime parameter to an equivalent short date string representation.
ToShortTimeString	string	• DateTime datetime	Converts the value of the DateTime parameter to an equivalent short time string representation.

> Back to the list of macro method categories

Mathematical operations

> The macro autocomplete help only shows the mathematical methods as members of the **Math** namespace, for example *{% Math.Abs(-2) %}*. However, the system resolves the methods even without the namespace.

Method	Return type	Parameters	Description
Abs	double	• double number	The absolute value of the specified number.
Acos	double	• double number	The angle whose cosine is the specified number.
Asin	double	• double number	The angle whose sine is the specified number.
Atan	double	• double number	The angle whose tangent is the specified number.

Average	double	• InfoObjectCollection collection • string columnName	The average of all numbers in the specified data column of the objects in the collection.
Ceiling	double	• double number	The smallest whole number greater than or equal to the specified number.
Cos	double	• double number	The cosine of the specified angle.
Cosh	double	• double number	The hyperbolic cosine of the specified angle.
Exp	double	• double number	**e** raised to the specified power.
Floor	double	• double number	The largest whole number lesser than or equal to the specified number.
GetRandomInt	int	• *int minValue* • *int maxValue* • *int seed*	Generates a random positive integer. You can use the optional parameters to specify the range of possible numbers, and the seed number for the generator.
GetRandomDouble	double	• *int minValue* • *int maxValue* • *int seed*	Generates a random positive decimal number. You can use the optional parameters to specify the range of possible numbers, and the seed number for the generator.
IsOdd	bool	• int number	Returns a true value if the specified number is odd.
IsEven	bool	• int number	Returns a true value if the specified number is even.
Log	double	• double number	The base **e** logarithm of a specified number.
Log10	double	• double number	The base 10 logarithm of a specified number.
Max	double	• double parameters	The maximum from the given list of numbers.

Maximum	double	• InfoObjectCollection collection • string columnName	The maximum of all numbers in the specified data column of the objects in the collection.
Min	double	• double parameters	The minimum from the given list of numbers.
Minimum		• InfoObjectCollection collection • string columnName	The minimum of all numbers in the specified data column of the objects in the collection.
Modulo	int	• int left • int right	The modulo of two integer numbers.
Pow	double	• double base • double exp	The number raised to the specified power.
Percent	double	• double percent	Multiples the specified number by 0.01.
Round	double	• double number • *int digits* • *string mode*	The number nearest to the specified value. The optional second parameter sets the number of fractional digits in the rounded value. You can use the third parameter to set the rounding mode for numbers half-way between two other numbers. Supported variants are: • AwayFromZero - rounds to the nearest number away from zero • ToEven - rounds toward the nearest even number
Sign	double	• double number	Returns a value indicating the sign of a number: • -1 (the number is less than zero) • 0 (the number is zero) • 1 (the number is greater than zero)
Sin	double	• double number	The sine of the specified angle.
Sinh	double	• double number	The hyperbolic sine of the specified angle.

Sqrt	double	• double number	The square root of a specified number.
Sum	double	• InfoObjectCollection collection • string columnName	The sum of all numbers in the specified data column of the objects in the collection.
Tan	double	• double number	The tangent of the specified angle.
Tanh	double	• double number	The hyperbolic tangent of the specified angle.

> Back to the list of macro method categories

E-commerce

Basic product methods

Method	Return type	Parameters	Description
GetSKUPrice	string	• object (int) skuId	Returns the price of the specified product (*object skuId*). The price includes any catalog discounts that apply. Example: *{% FormatPrice(GetSKUPrice(SKUID), true) %}*
GetSKUOriginalPrice	string	• object (int) skuId	Returns the *greater* of the following price options for the specified product (*object skuId*): • The product's list price (from the *SKURetailPrice* column) – OR – • The product's price (from the *SKUPrice* column) without any discounts applied If both prices are equal to the standard product price with discounts (i.e. the product's list price is equal to the price and no discounts apply), the method returns *0*. The price is in the current user's shopping cart currency. Example: *{% FormatPrice(GetSKUOriginalPrice(SKUID), true) %}*
GetSKUPriceSaving	string	• object (int) skuId	Returns the amount of money saved on the price of the specified product (*object skuId*). The savings are calculated based on the difference between the product's price (from the *SKUPrice* column, including any catalog discounts that apply) and the *greater* of the following options: • The product's list price (from the *SKURetailPrice* column) – OR – • The product's price (from the *SKUPrice* column) without any discounts applied If the product's price is greater than the list price and no discounts apply, the method returns *0*. The result is in the current user's shopping cart currency. Example: *{% GetSKUPriceSaving(SKUID) %}*

| GetSKUTax | string | • object (int) skuId | Depending on the site's **Prices include tax** setting, returns either:

• The tax value that will be added to the price of the specified product (*object skuId*) during checkout (if prices do not include tax)
– OR –
• The tax portion of the specified product's price (if prices already include tax)

For example:

• Prices do not include tax, the product price is 50 and the applied tax rate is 5% – returns **2.5**
• Prices include tax, the product price is 55 and the applied tax rate is 10% – returns **5**

Example: *{% FormatPrice(GetSKUTax(SKUID), false) %}* |
| IsOption UsedInVa riants | bool | • int skuId | Returns true if the product option with the specified ID (*int skuID*) is used in any existing product variant. |

Related methods

Method	Return type	Parameters	Description
GetForma ttedWeight	string	• double weight	Returns formatted weight (*double weight*) based on the mass unit format configuration in the settings.
GetManuf acturer	object	• int id • string column	Returns an object from the specified column (*string column*) of the manufacturer with the specified ID (*int id*).
GetDepart ment	object	• int id • string column	Returns an object from the specified column (*string column*) of the department with the specified ID (*int id*).
GetSuppli er	object	• int id • string column	Returns an object from the specified column (*string column*) of the supplier with the specified ID (*int id*).
GetIntern alStatus	object	• int id • string column	Returns an object from the specified column (*string column*) of the internal status with the specified ID (*int id*).
GetPublic Status	object	• int id • string column	Returns an object from the specified column (*string column*) of the public status with the specified ID (*int id*).
GetMainSi teCurrency	Currency Info	• int siteID	Returns the currency used on the site with the specified ID (*int siteID*).
GetShopp ingCartIte msCount	int		Returns the number of products in the current shopping cart.

GetCoupo nsMessage	string	• object (DiscountInfo) Discount	Returns an information message about the redemption count of coupon codes belonging to the specified discount (*object Discount*).
GetMultiB uyCoupon sMessage	string	• object (MultiBuyDisc ountInfo) MultiBuyDisco unt	Returns an information message about the redemption count of coupon codes belonging to the specified Buy X Get Y discount or Product coupon (*object MultiBuyDiscount*).
GetGiftCa rdCoupon sMessage	string	• object (GiftCardInfo) GiftCard	Returns an information message containing the total redeemed and remaining amount for all coupon codes belonging to the specified Gift card (*object GiftCard*). The amount values are in the site's main currency.
GetMassU nit	string		Returns the mass unit of the Kentico instance.
FormatPri ce	string	• decimal price • *bool round*	Returns the price (*decimal price*) formatted according to the currency used by the current shopping cart. The optional parameter determines whether the price is rounded based on the current site's settings (*bool round*).
FormatPri ceForCart	string	• decimal price • int cartID • *bool round*	Returns the price (*decimal price*) formatted according to the currency used by the specified shopping cart (*int cartID*). The optional parameter determines whether the price is rounded based on the settings of the shopping cart's site (*bool round*).
FormatPri ceInMain Currency	string	• decimal price • *bool round*	Returns the price (*decimal price*) formatted according to the current site's main currency. The optional parameter determines whether the price is rounded based on the current site's settings (*bool round*).
Shopping CartProdu cts	IEnumer able <Shoppi ngCartIte mInfo>	• object (Guid) cartGUID	Returns the products (shopping cart items) in the specified shopping cart (*object cartGUID*). Bundle items and product options are excluded from the list.

URLs

Method	Return type	Parameters	Description
GetAddT oWishList Link	string	• object (int) productId	Returns a HTML link (<a> tag) with a URL that adds the specified product (*object productId*) to the current user's wishlist. Additionally, an image is rendered before the link tag (using the *~/CMSModules/CMS_Ecommerce /addtowishlist.png* file).

GetRemoveFromWishListLink	string	• object (string) productId	Returns the URL for removing the specified product (*object productId*) from the current user's wishlist.
ShoppingCartURL	string	• string siteName • *object (string) cartGUID*	Returns the URL of the current user's shopping cart set up in the [checkout process settings](link) on the specified site (*string siteName*). You can specify a shopping cart's GUID (object cartGUID), which is added to the URL's query string, to restore products in the shopping cart (for example, when sending an email reminder of an abandoned shopping cart).
WishlistURL	string	• string siteName	Returns the URL of the current user's wishlist on the specified site (*string siteName*).
GetProductUrl	string	• object (int, string) skuId	Returns the URL of the product specified by the identifier (*object SKUID*), which is the primary key.
GetProductUrlForFeed	string	• object (Guid) skuGUID • object (string) skuName • object (string) siteName	Returns the URL of the specified product (*object skuGUID*) with the specified name (*object skuName*) with the feed query string parameter in the URL for the feed on the specified site (*object siteName*).
GetEproductUrl	string	• Guid fileToken • string fileName • int siteId	Returns the URL for downloading an [e-product](link) specified with a unique download URL (*Guid fileToken*), a file name (*string fileName*), and an ID of the site on which the order with the e-product took place (*int siteId*).
GetProductUrlByGUID	string	• object (Guid) skuGuid • object (string) skuName • *object (string) siteName*	Returns the user-friendly URL of the product specified by the [GUID](link) (*object skuGuid*) and name (*object skuName*) of the product. You can specify the site name of the product's site (*object siteName*).

GetProductImage	string	- object (string) imageUrl - object (string) alt - *object (int) maxSideSize* - *object (int) width* - *object (int) height*	Returns the complete HTML code of the URL of the specified product image (*object imageURL*) on the current site with the alternative text (*object alt*). You can specify the maximum width (*object maxSideSize*), or the required width (*object width*) and height (*object height*).
GetAddToShoppingCartLink (obsolete)	string	- object (int) productId - *object (bool) enabled*	Note: This method is obsolete, and only works if the site's shopping cart page (specified by the *Shopping cart URL* setting) contains the *Checkout process (obsolete)* web part. Returns a HTML link (<a> tag) with a URL that adds the specified product (*object productId*) to the current user's shopping cart. Use the optional *enabled* parameter if you wish to dynamically set whether the method returns the link (based on the status of individual products). Additionally, an image is rendered before the link tag (using the *~/CMSModules/CMS_Ecommerce/addorder.png* file).

Google tag manager e-commerce data

Method	Return type	Parameters	Description
GetGtmProductJson	string	- int skuId - *string additionalDataJson* - *string purpose*	Returns a JSON string containing the data of a specified product (*int skuId*), in a format suitable for use with the Google Tag Manager and Universal Analytics Enhanced Ecommerce features. For more information, see Integrating Google Analytics Enhanced Ecommerce. The optional *additionalDataJson* parameter allows you to extend the returned JSON string (custom fields and values). The optional *purpose* parameter can be processed in code customizations of the product data creation process, if you need to return different JSON data for various scenarios. Example: *{% GetGtmProductJson(SKUID), "{ 'list' : 'Search Results'}", "searchImpressions") %}*
GetGtmPurchaseJson	string	- *string additionalDataJson* - *string purpose*	Returns a JSON string containing the data of a completed order (purchase), in a format suitable for use with the Google Tag Manager and Universal Analytics Enhanced Ecommerce features. Only works on the page that you display to customers after they go through your website's checkout process (the page specified by the **Final step URL** property of your site's **Page wizard manager** web part, typically a payment page or "thank you" page). For more information, see Integrating Google Analytics Enhanced Ecommerce. The optional *additionalDataJson* parameter allows you to extend the returned JSON string (custom fields and values). The optional *purpose* parameter can be processed in code customizations of the purchase data creation process, if you need to return different JSON data for various scenarios.

GetGtmShoppingCartItemsJson	string	• *string additionalDataJson* • *string purpose*	Returns a JSON string containing the data of all products in the current shopping cart, in a format suitable for use with the Google Tag Manager and Universal Analytics Enhanced Ecommerce features. For more information, see [Integrating Google Analytics Enhanced Ecommerce](#). The optional *additionalDataJson* parameter allows you to extend the returned JSON string (custom fields and values). The optional *purpose* parameter can be processed in [code customizations of the product data creation process](#), if you need to return different JSON data for various scenarios.

[> Back to the list of macro method categories](#)

User membership and permissions

Method	Return type	Parameters	Description
CheckPrivilegeLevel	bool	• object user (UserInfo) • UserPrivilegeLevelEnum privilegeLevel	Checks the user's [privilege level](#). Returns a true value if the user's privilege level matches the required *UserPrivilegeLevelEnum* value or is higher. You can check the following levels: • *{% CurrentUser.CheckPrivilegeLevel(UserPrivilegeLevelEnum.Editor) %}* • *{% CurrentUser.CheckPrivilegeLevel(UserPrivilegeLevelEnum.Admin) %}* • *{% CurrentUser.CheckPrivilegeLevel(UserPrivilegeLevelEnum.GlobalAdmin) %}*
GetFormattedUserName	string	• object user (UserInfo) • *bool isLiveSite*	Returns the formatted username of the specified user object. The default format is *<full name> (<nickname>)* if the user has a nickname, otherwise *<full name> (<username>)*. The **isLiveSite** parameter determines whether the macro displays the username on the live website.
HasAnyMembership	bool	• object user (UserInfo)	Checks whether the user belongs to any [membership](#) in the system.
HasMembership	bool	• object user (UserInfo) • string userMemberships • *bool allMemberships*	Checks if a user belongs to the specified [memberships](#). Enter the code names of memberships through the **userMemberships** parameter. When checking multiple memberships, use semicolons (;) to separate the code names. If the **allMemberships** parameter is true, the user must belong to all specified memberships for the check to succeed. To check whether the user belongs to a membership on a global level, add the period character (.) prefix before the membership code name.

IsAuthorizedPerResource	bool	- object user (UserInfo) - string resource - string permission	Evaluates whether a user has a specific [permission](#) for a resource (module). Use code names to identify the resource and permission.
IsAuthorizedPerUIElement	bool	- object user (UserInfo) - string resource - string elementName	Evaluates whether a user is allowed to access a specific UI element of a resource (module). Use code names to identify the resource and element.
IsInGroup	bool	- object user (UserInfo) - string userGroup	Checks if a user belongs to a [group](#) (community or workgroup).
IsInRole	bool	- object user (UserInfo) - string userRole *bool allRoles*	Checks if a user belongs to the specified roles. Enter the code names of roles through the **userRole** parameter. When checking multiple roles, use semicolons (;) to separate the code names. If the **allRoles** parameter is true, the user must belong to all specified roles for the check to succeed. To check whether the user belongs to a role on a global level, add the period character (.) prefix before the role code name.

[> Back to the list of macro method categories](#)

Transformations

Method	Return type	Parameters	Description

ApplyTransformation	string	- IEnumerable collection - string transformationName *string contentBeforeTransformationName* *string contentAfterTransformationName*	Applies **Text / XML** transformations to a collection of items. See also: Using transformations in macro expressions
Transform	string	- IEnumerable collection - string transformationText	Applies ad-hoc transformation code (Text / XML type) to a collection of items, or a single item. Note: To keep the macro syntax valid, you need to break all *%}* closing macro tags within the ad-hoc transformation code into separate strings. Example: *{% CurrentUser.Transform("{%FullName%" + "}") %}*

ℹ️ When writing **Text / XML** transformations, you can use most ASCX transformation methods inside macro expressions.

Outside of transformations, the macro autocomplete only offers the methods under the **Transformation** namespace, for example *Transformation.GetDocumentUrl().* However, you can manually enter the methods even without the namespace.

> Back to the list of macro method categories