

QPR Suite Accelerators Expression Language

Version 1.6.0 (28.2.2016)

Contents

1 Overview	1
2 Arithmetic functions	1
3 String functions.....	2
4 Datetime functions	3
5 Array functions.....	4
6 Conversion functions	5
7 QPR Web Services functions	6
8 Dataset function.....	8
9 XML functions	11
10 Binary data functions.....	12
11 Other functions	13
12 Appendix 1: ExecuteSearch function configuration.....	15

1 Overview

This document describes an expression language used in several QPR Suite accelerators and extensions. The expression language is based on NCalc expression language (ncalc.codeplex.com). Expressions (formulas) make it possible to define values dynamically, i.e. values are calculated at runtime.

The evaluation process is case sensitive meaning parameter and function names must be written with right case letter. (Actually all function names are not case sensitive, but it's easiest to consider they are).

Expressions can be combined using **operators**. Expression's priority order starting from largest priority is: primary, unary, power, multiplicative, additive, relational, logical **and**, logical **or**. Following operators can be used:

- Logical or: **or**, **||** (these are equivalent)
- Logical and: **and**, **&&** (these are equivalent)
- Relational: **=**, **!=**, **<>**, **<**, **<=**, **>**, **>=** (**!=** and **<>** are equivalent)
- Basic calculations: **+**, **-**, *****, **/**, **%**
- Bitwise: **&** (bitwise and), **|** (bitwise or), **^** (bitwise xor), **<<** (left shift), **>>** (right shift)
- Unary: **!**, **not**, **-**, **~** (bitwise not)
- functions: **Abs(1)**, **doSomething(1, 'dummy')**

In expressions, characters **** and **'** need to be written as **** and **\'** (i.e. escaped).

The expression language supports all .Net datatypes. Following table contains examples, how to write literals of basic datatypes.

Data type	Example
integer	123456
double	123.456, 0.123
datetime	#2015/01/31#
boolean	true, false
string	'Hello world!'

When operating with datetime datatype, note that all data returned by QPR Web Service are strings formatted in XML date format (yyyy-MM-ddTHH:mm:ss.fffzzz). They can be converted to datetimes using function **StringToDate**. To use literal datetimes, the syntax is to encase the datevalue with hash signs (#), e.g. #2015/01/31#. This supports many date formattings, but incompatibilities in some formats may exists.

In the expressions any unicode characters can be defined using syntax **\uxxxx**. For example, new line is **\u000D\u000A** (carriage return and line feed characters used in Windows systems).

2 Arithmetic functions

Function	Description	Example expression	Result
Abs	Returns the absolute value of a specified number.	Abs(-1)	1
Acos	Returns the angle whose cosine is the specified number.	Acos(1)	0 (double)

Asin	Returns the angle whose sine is the specified number.	Asin(0)	0 (double)
Atan	Returns the angle whose tangent is the specified number.	Atan(0)	0 (double)
Ceiling	Returns the smallest integer greater than or equal to the specified number.	Ceiling(1.5)	2 (double)
Cos	Returns the cosine of the specified angle.	Cos(0)	1 (double)
Exp	Returns e raised to the specified power.	Exp(0)	1 (double)
Floor	Returns the largest integer less than or equal to the specified number.	Floor(1.5)	1 (double)
IEEERem ainder	Returns the remainder resulting from the division of a specified number by another specified number.	IEEERemainder(3, 2)	-1 (double)
Log	Returns the logarithm of a specified number.	Log(1, 10)	0 (double)
Log10	Returns the base 10 logarithm of a specified number.	Log10(1)	0 (double)
Max	Returns the larger of two specified numbers.	Max(1, 2)	2
Min	Returns the smaller of two numbers.	Min(1, 2)	1
Pow	Returns a specified number raised to the specified power.	Pow(3, 2)	9 (double)
Round	Rounds a value to the nearest integer or specified number of decimal places.	Round(3.222, 2)	3.22 (double)
Sign	Returns a value indicating the sign of a number.	Sign(-10)	-1
Sin	Returns the sine of the specified angle.	Sin(0)	0 (double)
Sqrt	Returns the square root of a specified number.	Sqrt(4)	2 (double)
Tan	Returns the tangent of the specified angle.	Tan(0)	0 (double)
Truncate	Calculates the integral part of a number.	Truncate(1.7)	1
in	Returns whether an element is in a set of values.	in(1 + 1, 1, 2, 3)	true
Random	Returns a random value between 0 and 1 (the value can be 0 but cannot be 1).	Random()	0,358024762

3 String functions

All string function are identical of string methods in .Net Framework 4.5. See following link for their documentation: [https://msdn.microsoft.com/en-us/library/system.string.methods\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.string.methods(v=vs.110).aspx). The first parameter is the string to which the operation is done.

Function	Parameters
Substring (string)	string, int, (int)
IndexOf (int)	string, string, (int)
LastIndexOf (int)	string, string, (int)
CharAt (string)	string, int

Replace (string)	string, string, string
Trim (string)	string
ToLower (string)	string
ToUpper (string)	string
Length (int)	string
Contains (boolean)	string, string
StartsWith (boolean)	string, string
EndsWith (boolean)	string, string

4 Datetime functions

Custom function	Parameters	Description
AddMonths (datetime)	datetime, int	see (1) below
AddMilliseconds (datetime)	datetime, numeric	see (1) below
CompareDates (int)	datetime, datetime	see (1) below
CurrentDateTime (datetime)	[none]	Returns a datetime representing the current date and time. As there are no parameters, use syntax CurrentDateTime()
GetTicks (int)	datetime	see (2) below (property: Ticks)
GetDate (int)	datetime	see (2) below (property: Date)
GetYear (int)	datetime	see (2) below (property: Year)
GetMonth (int)	datetime	see (2) below (property: Month)
GetDay (int)	datetime	see (2) below (property: Day)
ToMetricsDateFormat (double)	datetime	Returns number of days between 1.1.1900 00:00 UTC and the provided datetime. QPR Metrics handles date values using this format. Also the function is needed for criteria in Web Service's QueryObjects, e.g. 'createddate>' + ToMetricsDateFormat(CurrentDateTime())
FromMetricsDateFormat (datetime)	numerical value (string, double or integer)	Returns a date that this the provided number of days from 1.1.1900 00:00 UTC. QPR Metrics and QPR Web Services using this date format.

- (1) DateTime methods in .Net Framework 4.5: [https://msdn.microsoft.com/en-us/library/system.datetime.methods\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.datetime.methods(v=vs.110).aspx)
- (2) DateTime properties in .Net Framework 4.5: [https://msdn.microsoft.com/en-us/library/system.datetime.properties\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.datetime.properties(v=vs.110).aspx)

5 Array functions

An array is a list of objects of any type, such as strings or integers. Arrays may contains same valued items multiple times.

Function	Parameters	Description
Sum (double)	- array	Calculates a sum of array items. Array must contains numerical data.
Average (double)	- array	Calculates an average of array items. Array must contains numerical data.
ArraySize (int)	- array	Calculates number of items in an array.
Concatenate (string)	- array - separator (string)	Concatenates items of an array into a string separated by a provided separator. Array must contain items that can be converted into string.
ArrayUnique (array)	- array	Returns an array where duplicate values are removed. Order of items is preserved.
Transform (array)	- array - expression (string) - argument1 (object) - argument2 (object), ...	Transforms every item of an array to another array using the provided expression. The expression has following arguments: <ul style="list-style-type: none"> - value which gets the value of the item. - argument1, argument2, ... which are the additional function parameters starting from the 3. parameter - index: iteration order number starting from 0 E.g. transform([inputarray], 'substring([value], 5)') returns first five characters from every string of the array.
IndexOfInArray (int)	- array - item to search (object)	Returns an index number of an object in the array. First object array starting from index 0. If object is not found, -1 is returned.
IndexOfAnyInArray (int)	- array - array	Returns an index number of the any of objects of the second array in the first array. The second array objects are searched from left to right. First object array starting from index 0. If the first array doesn't contain any of objects in the second array, -1 is returned.
ItemAt (object)	- array - int	Returns an item in the provided index number.
ArrayUnion (array)	- array - array	Joins two arrays. Duplicates are not removed. If only one parameter is provided, that parameter is expected to contain an array, item of which are arrays. In that case all the items of all arrays are joined.
ArrayIntersect (array)	- array - array	Returns an array which contains items that are in both provided arrays.
ArrayExcept (array)	- array - array	Returns an array which contains items that are in the first array but not in the second array.
ArrayWhere (array)	- array - expression (string)	Filters out all items of an array, where the expression is evaluated as false. The expression must evaluate as boolean. Available arguments in the expression:

	<ul style="list-style-type: none"> - argument1 (object) - argument2 (object), ... 	<ul style="list-style-type: none"> - item (item in the array) - index (iteration order number starting from 0) - argument1, argument2, ... (additional arguments starting from the 3. parameter)
ArraySort (array)	- array	Sorts an array
ArrayReverse (array)	- array	Reverses an array, i.e. first element becomes last and so on.

6 Conversion functions

Function	Parameters	Description
ToArray (array)	<ul style="list-style-type: none"> - string to convert - separator (string) - remove empties (bool) 	Converts a string to an array using provided separator string. If remove empties is true, there are no empty strings in the output array.
StringToDate (datetime)	<ul style="list-style-type: none"> - string (string to convert) - string (dateformat) 	Converts string to datetime. If the second parameter is omitted, the string is interpreted as XML date (format is yyyy-MM-ddTHH:mm:ss.fffzzz)
DateToString (string)	<ul style="list-style-type: none"> - date to convert (datetime) - dateformat (string) 	Converts datetime to string. More information how to construct the dateformat: https://msdn.microsoft.com/en-us/library/8kb3ddd4(v=vs.110).aspx
ArrayToDataset (dataset)	<ul style="list-style-type: none"> - array - column name (string) 	Converts an array to a dataset. The dataset has one column which name is given as a second parameter.
DatasetColumnToArray (array)	<ul style="list-style-type: none"> - dataset - column name (string) 	Converts a column of a dataset to an array. The column is referenced using the column name.
DatasetRowToArray (array)	<ul style="list-style-type: none"> - dataset - row number (int) 	Converts a row of a dataset to an array. The row is referenced using the row number (the first is 1).
ConvertToInt (int)	<ul style="list-style-type: none"> - input value (string) - value in error (integer) 	Converts a string value to an integer value. The input value can also be in other formats that can be converted to integer. The second parameter is voluntary, and it's returned if the conversion fails. If it's not defined and the conversion fails, an error is thrown.
ConvertToDouble (double)	<ul style="list-style-type: none"> - input value (string) - value in error (double) 	Converts a string value to a double value. The input value can also be in other formats that can be converted to double. The second parameter is optional, and it's returned if the conversion fails. If it's not defined and the conversion fails, an error is thrown.

ConvertToString	- input value (object) - value in error (double)	Converts an object to string. The second parameter is optional, and it's returned if the conversion fails. If it's not defined and the conversion fails, an error is thrown.
ModelIdFromFullId (string)	- element full id (string)	Extracts a model id from a full id. The function returns an empty string, if the model id cannot be extracted. Example: ModelIdFromFullId('SC.6346904.5290187') returns 6346904
ObjectIdFromFullId (string)	- element full id (string)	Extracts an object id from a full id. The function returns an empty string, if the object id cannot be extracted. Example: ObjectIdFromFullId('SC.6346904.5290187') returns 5290187
InstanceIdFromFullId (string)	- element full id (string)	Extracts an instance id from a full id. The function returns an empty string, if the instance id cannot be extracted. Example: InstanceIdFromFullId('SC.6346904.5290187.4353690') returns 4353690
HtmlEncode (string)	- string	Performs HTML encoding. More information: https://msdn.microsoft.com/en-us/library/system.net.webutility.urlencode(v=vs.110).aspx
UrlEncode (string)	- string	Performs URL encoding. More information: https://msdn.microsoft.com/en-us/library/4fkewx0t(v=vs.110).aspx

7 QPR Web Services functions

Function	Parameters	Description
GetPortalUrl (string)	- objectid (string) - view (string) - options (string)	QPR Web Service's GetPortalUrl operation, see http://kb.qpr.com/qpr2015-1/index.html?getportalurl.htm .
GetAttribute (string)	- objectid (string) - attribute (string) - options (string)	QPR Web Service's GetAttributeAsString operation, see http://kb.qpr.com/qpr2015-1/index.html?getattributeasstring.htm .
GetAttributes (string array)	- objectid (string) - attributes (string) - options (string)	Returns multiple attributes of a single object as an array. Attributes are defined as comma separated. Based on QPR Web Service's QueryObjects operation.
QueryObjects (string array)	- query (string) - filter (string) - sortby (string) - attributes (string) - options (string)	Executes QPR Web Service's QueryObjects and returns results as a string array. Only one attribute is defined for "attributes" parameter.

QueryObjectsCount (int)	<ul style="list-style-type: none"> - query (string) - filter (string) - options (string) 	Returns number of objects returned by QueryObjects. "orderby" and "attributes" cannot be defined as they don't affect the result.
QueryObjectsSum (double)	<ul style="list-style-type: none"> - query (string) - filter (string) - attributes (string) - options (string) 	Sum of queried objects of numerical attributes. Only one attribute is defined in "attributes". Error is thrown if attribute values are not numerical.
QueryObjectsAverage (double)	<ul style="list-style-type: none"> - query (string) - filter (string) - attributes (string) - options (string) 	Average of queried objects numerical attributes. Only one attribute is defined in "attributes". Error is thrown if attribute values are not numerical.
QueryObjectsFirstAttribute (double)	<ul style="list-style-type: none"> - query (string) - filter (string) - orderby (string) - attributes (string) - options (string) 	Attribute value of the first object of queried objects. Only one attribute is defined in "attributes".
QueryObjectsConcatenate (string)	<ul style="list-style-type: none"> - query (string) - filter (string) - orderby (string) - attributes (string) - options (string) - separator (string) 	Concatenates attribute values of all queried objects using defined separator. Only one attribute is defined in "attributes".
QueryObjectsUnique (string)	<ul style="list-style-type: none"> - query (string) - filter (string) - orderby (string) - attributes (string) - options (string) - separator (string) 	Concatenates unique attribute values of all queried objects using defined separator. Only one attribute is defined in "attributes".
LatestValuePeriod (string)	<ul style="list-style-type: none"> - objectid (string) - series (string) 	Returns the id of the period that contains the latest (newest) measure value. This information can be used e.g. in web service attribute prettyvalue(period="[periodid]") .
SubAttributesToArray (array)	<ul style="list-style-type: none"> - objectid (string) - attribute name (string) - subattribute name (string) - filter expression (string) - argument 1 (object) - argument 2 (object), ... 	Returns attribute values (as an array) which are presented in more complex structures. This type of attributes are e.g. <i>graphicalproperties</i> , <i>customattributetypes</i> and <i>properties</i> . In the filter expression is for selecting the desired rows. In the expression there are following arguments_ <ul style="list-style-type: none"> - all sub tag attributes with their tag names - argument1, argument2, ... for all the function's additional parameters starting from 5. parameter - argument ordernumber having values 0, 1, ... for xml tag order number. <p>Example, following graphicalproperties attribute:</p>

		<pre><graphicalproperties> <symbol id="920256947" x="790" y="190" width="400" height="30" instanceid="2029031131" /> <symbol id="2009602777" x="230" y="640" width="200" height="40" instanceid="0" /> <symbol id="368985118" x="730" y="40" width="100" height="60" instanceid="369716419" /> </graphicalproperties></pre> <p>Calling: SubAttributesAsArray([instanceid], 'graphicalproperties', 'width', '[instanceid]=0')</p> <p>returns 200.</p>
SubAttributesAsDataset (dataset)	<ul style="list-style-type: none"> - objectid (string) - attribute name (string) - columns (array) 	Returns attributes which are presented in more complex structures as a dataset. This type of attributes are e.g. graphicalproperties, customattributetypes, properties and hotspots. Dataset columns (names of the sub attributes) are listed as an array (in lower case).

8 Dataset function

Dataset is a table like object containing multiple named columns and multiple rows. Dataset can contains zero rows, but at least one column must exist. Dataset may contain any type of data. Dataset is like a table in relational database or a worksheet in Excel.

Usually dataset functions don't change the input dataset, but create new dataset (exceptions are mentioned). Thus the input datasets may also be used in other functions.

Dataset functions get their idea from SQL language.

Function	Parameters	Description
From (dataset)	<ul style="list-style-type: none"> - query (string) - filter (string) - sortby (string) - attributes (string) - options (string) 	Returns QPR Web Service's QueryObjects result as a dataset. Columns get their names from the result data. Column names can be changed with QueryObjects syntax attribute(as="ColumnName") .
InnerJoin (dataset)	<ul style="list-style-type: none"> - left dataset (dataset) - right dataset (dataset) - matching expression (string) 	Inner join of two datasets. Expression have columns names as available arguments. The dataset cannot have columns with same names.
LeftJoin (dataset)	<ul style="list-style-type: none"> - left dataset (dataset) - right dataset (dataset) - matching expression (string) 	Left join of two datasets. Expression have columns names as available arguments. The dataset cannot have columns with same names.
RightJoin (dataset)	<ul style="list-style-type: none"> - left dataset (dataset) - right dataset (dataset) 	Right join of two datasets. Expression have columns names as available arguments. The dataset cannot have columns with same names.

	- matching expression (string)	
FullOuterJoin (dataset)	- left dataset (dataset) - right dataset (dataset) - matching expression (string)	Full outer join of two datasets. Expression have columns names as available arguments. The dataset cannot have columns with same names.
AddColumn (dataset)	- dataset - columnName (string) - expression (string) - compare mode (boolean) - argument1 (object) - argument2 (object), ...	Adds a new column to the dataset. All other column values are available as arguments in the expression. In addition there are possibly provided additional arguments argument1 , argument2 , Also there is an argument rowordernumber , which is the row order number starting from 0. In the <u>compare mode</u> (true), there are also values of the previous and next rows available as arguments with suffixes _previous , _current and _next . For example if there are columns column1 and column2 , and adding a new column column3 , there are arguments column1_previous, column2_previous, column3_previous, column1_current, column2_current, column1_next and column2_next.
RemoveColumn	- dataset - columnName (string)	Removes a columns from a dataset.
Where (dataset)	- dataset - expression (string) - argument1 (object) - argument2 (object), ...	Filters out all rows in the dataset where expression is evaluated false. Expression has all columns as arguments.
Union (dataset)	- dataset1 - dataset2	Union of two datasets. Results dataset contains all rows of the input datasets. The datasets must have same columns.
Intersect (dataset)	- dataset1 - dataset2	Intersect of two datasets. Result dataset contains only those rows that are in both datasets. The datasets must have same columns.
Except (dataset)	- dataset1 - dataset2	Result dataset contains rows that are in the first dataset but not in the second dataset. The datasets must have same columns.
Distinct (dataset)	- dataset	Removes duplicate rows, i.e. rows that contains identical data.
GroupBy (dataset)	- dataset - list of grouped columns (string) - list of combined column names (string) - combine expressions (string)	Groups the dataset. Parameters: - 1. parameter is the dataset to group. - 2. parameter is a comma separated list of columns to group - 3. parameter is a comma separated list of combine column names - Rest of parameters are combine expression (e.g. count sum, average or number of rows). All

		<p>expression have as arguments an array of objects to combine. Parameter names equal to column names.</p> <p>Example: <code>GroupBy([dataset1], 'modelname,typename', 'count,objectnames', ArraySize([name]), 'Concatenate([name])')</code></p>
SortBy (dataset)	<ul style="list-style-type: none"> - dataset - sorting definition (string) 	<p>Sorts the dataset. Sorting is defined as comma separating the sorted columns. Optionally asc (default) or desc can be added after the column name. E.g. attribute1 asc,attribute2 desc</p>
Transpose(dataset)	<ul style="list-style-type: none"> - dataset 	<p>Transposes a dataset, i.e. changes its rows to columns and columns to rows. In the transposed dataset, first column name is headers and it contains the header names of the original dataset. Rest of the header names are column_1, columns_2, ...</p>
Split (dataset)	<ul style="list-style-type: none"> - dataset - split column name (string) - split expression (string) 	<p>Splits every row of a dataset into multiple rows based on the split expression. The split expression must return an array containing splitted items for a single row. Thus, the number of items in the array determines, to how many rows a single row is splitted. The splitted items are stored in a new column.</p> <p>The split expression has all dataset column values as arguments.</p> <p>Note that the splitting may also result to a single row or even zero rows, if the split expression returns an array of one or zero items.</p> <p>Example: <code>Split([dataset], 'splitted', 'ToArray([column_1], '\,')</code></p>
Matrix (dataset)	<ul style="list-style-type: none"> - dataset - row expression (string) - column expression (string) - value expression (string) - grouping expression (string) - row column name (string) 	<p>Builds a matrix based on the provided dataset. Row and column expressions are executed for each row of the dataset, and rows are positioned in the matrix based on the row and column expression values.</p> <p>The grouping expression has an input argument "value", containing an array of all calculated value expressions for the matrix cell.</p> <p>"Row column name" defines the column containing matrix row names.</p>
DatasetCell (object)	<ul style="list-style-type: none"> - dataset - column name (string) - row number (integer) 	<p>Gets an item from a dataset from the named column and from the named index (first row is number 1). Returned type is the type of the cell.</p>
DatasetSize (int)	<ul style="list-style-type: none"> - dataset 	<p>Number of rows in a dataset. If dataset is null, -1 is returned.</p>
CreateDataset (dataset)	<ul style="list-style-type: none"> - column names (string array) 	<p>Creates a new dataset with provided column names. The created dataset contains no rows.</p>
AddDatasetRow (dataset)	<ul style="list-style-type: none"> - column 1 value (object) 	<p>Adds a new row to a dataset with the provided values. A row is added to the provided dataset, i.e. no new dataset is created.</p>

	<ul style="list-style-type: none"> - column 2 value (object) - ... 	
MetricsMeasureValues (dataset)	<ul style="list-style-type: none"> - source dataset - element id column (string) - series symbol column (string) - attribute name (string) - start period (string) - end period (string) 	<p>Gets multiple Metrics measure values and adds to a dataset. The source dataset contains Metrics element ids and series symbols in own columns. Periods are added as new columns between the defined start and end periods. Period names are the column names. Start and end periods can be defined using string value period name (with \$ prefix, such as \$1/2016) or date values.</p> <p>There can be measures using different period levels. Periods are matched using period names, i.e. values of measured with different period levels will be in same columns if the period names of the different period levels match.</p> <p>Valid attribute names and value and prettyvalue (these are from QPR Web Service's <i>values</i> attribute. The result Metrics values are strings, and in case of numerical values, they can be converted to integer or double.</p>
BuildHierarchy (dataset)	<ul style="list-style-type: none"> - dataset - instance id column name (string) - object id column name (string) - parent object id column name (string) - Hierarchy id column name (string) - Hierarchy parent id column name (string) - level column name (string) - filtering formula (string) - sorting formula (string) 	<p>Builds an element hierarchy based on data containing element instances. The data is provided in a dataset, where there are following columns: instance id, object id and parent object id. The object id can be derived from instance id.</p> <p>The function constructs a new dataset where lines are in the hierarchy order. The function can handle situations where also other than the bottom level objects have been instantiated; it means that the result hierarchy will contain same instances multiple times. Following new columns are added:</p> <ul style="list-style-type: none"> - hierarchy id (explained below) - hierarchy parent id (referring to the hierarchy id) - level (top level is 0, the level directly below top is 1, ...) <p>The hierarchy id means an id that is unique for the whole hierarchy. To enable this following technique is used: If there is an instance id PG.x.y.z which appears multiple times, the first hierarchy id is PG.x.y.z, the second is PG.x.y.z.1, the third is PG.x.y.z.2, and so on.</p>

9 XML functions

Function	Parameters	Description
XmlDocument (XDocument)	<ul style="list-style-type: none"> - XML data (string) - XML schema (string) - custom error message (string) 	Constructs an XML document and validates it. Type of the returned object is XDocument . The XML data is provided as the first parameter and an XML schema as a second parameter. If the XML

		document is not compatible with the schema (i.e. valid), an error is thrown.
SelectXMLElement (XElement)	<ul style="list-style-type: none"> - XDocument or XElement - XPath expression (string) - namespace prefix (string) - namespace URI (string) 	Selects an element from an XML document using XPath expression. The namespace definition is for the XPath expression. Null value is returned if no element matches.
SelectXMLElements (XElement array)	<ul style="list-style-type: none"> - XDocument or XElement - XPath expression (string) - namespace prefix (string) - namespace URI (string) 	Selects an array of elements from an XML document using XPath expression. The namespace definition is for the XPath expression. An empty set is returned if no element matches.
XmlAttribute (string)	<ul style="list-style-type: none"> - XElement - attribute name (string) - default value (string) 	Gets an XML attribute value from an XML element. Empty value (or optional default value) is returned if the attribute is not found.
SelectXmlAttribute (string)	<ul style="list-style-type: none"> - XDocument or XElement - XPath expression (string) - namespace prefix (string) - namespace URI (string) - attribute name (string) 	Gets an XML attribute value from an XML element that is selected from an XML document using an XPath expression. An empty string is returned if no element matches or if the attribute is not found.
SelectXMLAttributes (string array)	<ul style="list-style-type: none"> - XDocument or XElement - XPath expression (string) - namespace prefix (string) - namespace URI (string) - attribute name (string) 	Gets a list of XML attribute values from a list of XML elements that are selected from an XML document using an XPath expression. An empty set is returned if no element matches.
XPathEncode (string)	<ul style="list-style-type: none"> - string 	Encodes a string to be suitable for an XPath expression. The function adds quotes (") if needed.

10 Binary data functions

File data functions are for getting file contents as a byte array from different sources. There are also functions for getting media types of files.

Function	Parameters	Description
HttpFileData (byte array)	<ul style="list-style-type: none"> - url (string) - alternateUrl (string) 	Fetches a file as byte array through an http(s) using the provided url address. The address must point directly to the file resource. If no resource is found from url, an alternateUrl is tried.
HttpFileMediaType (string)	<ul style="list-style-type: none"> - url (string) - alternateUrl (string) 	Media type of the file fetched using HttpFileData function with a same parameter. If no resource is found from url, an alternateUrl is tried.
QprEmbeddedFileData (byte array)	<ul style="list-style-type: none"> - object (string) - attribute (string) - options (string) 	Gets an embedded file from QPR system as byte array using QPR Web Service's GetBinaryData operation with provided parameters. More information http://kb.qpr.com/qpr2015-1/index.html?getbinarydata.htm .

QprGraphData (byte array)	- object (string) - options (string)	Gets an image file as byte array using QPR Web Service's GetGraph operation with provided parameters.
QprGraphMediaType (string)	- object (string) - options(string)	Media type of the file got using qprEmbeddedFileData function with same parameters.

11 Other functions

Function	Parameters	Description
If	- condition (boolean) - true value (object) - false value (object)	Usual programming conditional statement. The condition is evaluated, and if the condition is true, the true value is returned; otherwise it returns the false value . If the condition is evaluated to a null value, the false value is returned. The false value is optional, and in that case null is returned if condition is false. If the condition is true the false value (expression) is not evaluated and vice versa. Example: if([variable1] = 2, 'value is two', 'value is something else')
SwitchCase	- control expression (object) - condition1 (object) - value1 (object) - condition2 (object) - value2 (object) - default value (object)	Conventional programming "switch" statement. If control expression equals condition1 , value1 is returned and so on. Control expression may be string, integer, double or date. If no condition matches, the default value is returned. Note that the number of parameters must be an equal number (4, 6, 8, ...). Example: SwitchCase([variable1], 1, 'value is one', 2, 'value is two', 'value is something else')
Loop (object)	- array - expression (string) - argument1 (object) - argument2 (object), ...	Loops through an array and calculates an expression for every iteration. The function gives a value of the last iteration's expression as a result. If array length is zero, a null value is returned. Following arguments are available in the expression: <ul style="list-style-type: none"> - value: Item in the array. - previousresult: Result of previous iteration's calculated expression. For the first iteration, this value is null. - index: Iteration order number starting from 0. - argument1, argument2, ...: additional arguments provided starting from the 3. parameter Example: Loop(ToArray('4,2,3', ','), 'coalesce([previousresult], 0) + [value]') (gives 4+2+3=9)

ApplicationVersion (string)	[none]	Returns a version number of the application using the expression language, e.g. DWR or DWV.
ExpressionLanguageVersion (string)	[none]	Returns NCalc Expression Language version number, i.e. version of QPR Suite Accelerator .Net Tools. QPR Suite Accelerator .Net Tools can be upgraded by replacing the QPRSuiteAcceleratorTools.dll in the QPR Web Services Extensions folder under IIS with a new one.
RaiseError	- error message (string)	Raises (throws) an error and shows an error message that is passed as a parameter. This can be used if there is a more complex logic for parameter validation.
WsSessionId (string)	[none]	Returns QPR Web Service session id, if there is a valid session. If not, returns an empty string.
ApplicationSetting (string)	- setting name (string)	Returns application setting value by its name. Settings depend on the application where the NCalc library is used. E.g. application settings for applications, that are part of QPR Web Services Extensions, are defined in web.config of QPR Web Services Extensions.
Coalesce (object)	- object - object, ...	Returns first of the objects (counting from left to right) which value is <ul style="list-style-type: none"> - for strings, first string that is not null or empty string, or - for other than strings, first value that is not null. <p>Minimum of two parameters are needed. If all parameters are nulls or empties, the first parameter is returned.</p>
Eval (object)	- expression (string) - argument1 (object) - argument2 (object), ...	Evaluates an expression. The expression has arguments argument1 , argument2 , ... for each additional parameter for the Evaluate function starting from the second parameter. Number of arguments is not restricted.
EvalWithNamedArguments	- expression (string) - argument1 name (string) - argument1 value (object) - argument2 name (string) - argument3 value (object), ...	Evaluates an expression. The expression has arguments with provided names and values. Note that the function must have odd number of arguments. Number of arguments is not restricted.
IsNull (boolean)	- object	Return true if the value is null, otherwise false.
NullValue	- [none]	Return null value.

IsNumeric (boolean)	- string	Returns true if the input string can be converted to a numerical value, otherwise false.
DataType (string)	- object	Return the type of provided parameter, e.g. int, string, datetime.
DiagramPath (string array)	<ul style="list-style-type: none"> - object id (string) - attribute name (string) - parent relation (string) - separator (string) - options (string) 	<p>Returns an array of EA/PD diagram paths for an object or an object instance. An array is returned as there can be several diagram paths if the object has several instances or if diagrams have been instantiated. Value of the provided attribute is used to identify an object in the diagram path (usually object "name" is used).</p> <p>Parent relation is the relation attribute name for getting parent objects (usually "parentobjects" is used). Also a "separator" character must be provided (usually "/"). "Options" is for QueryObjects operation for getting parent objects.</p>
ExecuteSearch (string array)	- search configuration (XDocument)	Executes a search for QPR objects. The search is based on a search configuration provided as an xml document. The schema of the XML document is described in Appendix 1.
ExecuteRecursion	<ul style="list-style-type: none"> - return expression (string) - recursion expression (string) - recursion initial value (object) - exclude traversed (boolean) - argument1 (object) - argument2 (object), ... 	<p>Executes a recursion based on provided expressions. There are two expressions:</p> <ul style="list-style-type: none"> - return expression: Determines the value a recursion step returns. The expression contains an argument recursionresult which is an array containing the return expressions of all the one level below recursion steps. - recursion expression: Determines the next level recursions item. Must return an array. The value of this array item will be given to the next step of the recursion. The expression contains an argument currentrecursionstep which is the recursion value of the current step. <p>The recursion initial value is the value where the recursion starts.</p> <p>Exclude traversed means that items that have already been encountered during the recursion are excluded from next level recursion items.</p> <p>Example:</p> <pre>ExecuteRecursion('1 + Sum([value]), 'QueryObjects(\'[\ + [value] + \].childobjects\', \'\', \'\', \'id\', \'\'', 'PG.123.456')</pre>

12 Appendix 1: ExecuteSearch function configuration

The search is based in a defined **scope**, which is a set of objects where the search is targeted. A scope can be e.g. all published models, all certain types of elements. The scope also includes element attributes where the search is targeted. A scope consist of multiple **scope parts**, which are QPR Web Services queries.

The objects defined by the scope are filtered using a **criteria** (the search results are the matching objects). Criteria can be any expression containing and, or, not and parenthesis.

XML element	Description	Attributes
executesearch (1)	parent: none	scopecombining: and, or
scopepart (1...n)	Defines a scope part, which contains a web service query, transformation and attributes. Possible parent element: executesearch	query: QPR Web Services query transformation: transformation relation name
criteria (1...n)	Defines criteria for search. There can be criteria elements inside other criteria elements to form a hierarchical structure representing expression calculation order. Criteria can be defined under executesearch for a criteria for all criteria parts, or the criteria can be defined under a certain criteria part. Possible parent elements: executesearch, scopepart, criteria	type: <ul style="list-style-type: none"> • and (one to many sub criteria) • or (one to many sub criteria) • not (one sub criteria) • text (no sub criteria) searchtext: search text. Used when type=text attribute: searched QPR Web Service attribute. Used when type=text. compariontype: contain, begin, is. Used when type=text matchcase: true or false
attribute (0..n)	Defines searched attributes related to a scope part. Parent element is scopepart	name: name of the attribute

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<executesearch xmlns="http://www.qpr.com/QPRSuite/ExecuteSearch" scopecombining="and">
  <scopepart query="[PG.1221241820].Ominaisuus" transformation="">
    <criteria type="and">
      <criteria type="text" searchtext="haku12" attribute="name" compariontype="begin" matchcase="true"/>
      <criteria type="text" searchtext="haku3" attribute="sanastontermi.value" compariontype="is" matchcase="false"/>
    </criteria>
  </scopepart>
  <scopepart query="[PG.1221241820].Käsite" transformation="käsitteenominaisuus">
    <criteria type="and">
      <criteria type="text" searchtext="haku1" attribute="name" compariontype="contain" matchcase="false"/>
    </criteria>
  </scopepart>
</executesearch>
```

Schema:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.qpr.com/QPRSuite/ExecuteSearch"
xmlns:search="http://www.qpr.com/QPRSuite/ExecuteSearch" elementFormDefault="qualified">
  <xs:complexType name="criteria">
    <xs:sequence>
      <xs:element name="criteria" type="search:criteria" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="type" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="text"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
```

```

        <xs:enumeration value="and"/>
        <xs:enumeration value="or"/>
        <xs:enumeration value="not"/>
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="searchtext">
<xs:simpleType>
    <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="attribute">
<xs:simpleType>
    <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="compariontype">
<xs:simpleType>
    <xs:restriction base="xs:string">
        <xs:enumeration value="contain"/>
        <xs:enumeration value="is"/>
        <xs:enumeration value="begin"/>
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="matchcase" type="xs:boolean"/>
</xs:complexType>

<xs:element name="executesearch">
<xs:complexType>
<xs:sequence>
    <xs:element name="scopepart" maxOccurs="unbounded">
        <xs:complexType>
            <xs:sequence>
<xs:element name="criteria" type="search:criteria" minOccurs="0" maxOccurs="1"/>
        <xs:element name="attribute" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
                <xs:attribute name="name" type="xs:string"/>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="query" type="xs:string" use="required"/>
    <xs:attribute name="transformation" type="xs:string"/>
</xs:complexType>
</xs:element>
<xs:element name="criteria" type="search:criteria" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
<xs:attribute name="scopecombining">
<xs:simpleType>
<xs:restriction base="xs:string">
    <xs:enumeration value="and"/>
    <xs:enumeration value="or"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:schema>

```