



Byron Informatik AG
Eisengasse 6
CH-4051 Basel
Tel. +41 (0)61 690 96 00
Fax +41 (0)61 690 96 09

byron@byron.ch
www.byron.ch

Byron/BIS – Technical product information Expressions in XDS variables and FilterNavigation

Table of contents

1	Contexts	4
2	Syntax	4
3	Predefined Functions	5
	ARRAYELEMENTOF(array : Variant; index : Integer) : Variant (>= v5.5.2)	5
	ATTRIBUTE(attribute : String; option : String) : Variant	5
	BASE64TOBYTES (base64String : String) : Variant (>= v5.7.5)	5
	BYTESTOBASE64 (bytes : Variant) : String (>= v5.7.5)	5
	BYTESTOGUIDSTR (bytes : Variant) : String (>= v5.7.5)	5
	CLIPBOARD : String	5
	CONVERTIFCGUID (value : String) : String (>= v5.7.5)	6
	COUNT : Integer	6
	DATE(y, m, d : Integer) : DateTime	6
	DATETIMETO(d : DateTime; op : String) : DateTime	7
	DAY(d : DateTime) : Integer	7
	ENUMVALUE(attribute : String; lang : String; index : Integer; option : String) : Variant	8
	ENUMVALUES(attribute : String; lang : String; separator : String) : Variant	8
	EVAL(v : String) : String	8
	EXTRACTNAME(fullFileName : String) : String	8
	FIELD(fullText : String; pos : Integer; [sep : String; quote : String]) : String	9
	FIELDBYNAME(line, header, field, [def, sep, quote : String]) : String	9
	FILEAGE(fullFileName : String) : DateTime	9
	FILECONTENTS(fullFileName : String) : String	9
	FILESIZE(fullFileName : String) : Float	9
	FORMAT(v : Integer/Float/DateTime, format : String, useLocales: Boolean) : String	10
	GUID : String	10
	GUIDSTRTOBYTES (guid : String) : String (>= v5.7.5)	10
	HASLIC(aFeature : String) : Integer	10
	HASRIGHT(class : String; theRight: String) : Integer	10
	IFNULL(v : Variant; def : Variant) : Variant	10
	INT(s : String/DateTime/Float) : Integer	10
	INTERVALTOFLOAT(interv : Interval; baseUnit : Integer) : Float	11
	LANGUAGE(range : String; type : Integer) : String/Integer	11
	MD5(s : String) : String	11
	MONTH(d : DateTime) : Integer	11
	NOW : DateTime	11
	OBFUSCATE(val : Variant; mode : Integer) : Variant	12
	OBJCOUNT(association : String) : Integer	12
	OBJIMAGE : String * (>= v5.1.1)	12
	OSENv(v : String) : String	12
	OVERLAY(background, arg : String; AtPos, align : Integer) : String	13
	PARAM(v : String) : String *	13
	RANDOM(max : Float) : Float	13
	REGEXP(test : String; pattern : String; options : String) : Boolean	13
	STRCASE(str : String; op : Integer) : String	14

	STRLEN(str : String) : Integer.....	14
	STRPOS(sub, str : String; [times : Integer; quote : String]) returns Integer	14
	STRREPL(orig, old, new : String) : String.....	14
	STRTODATE(str : String [format : String]) : DateTime.....	14
	STRTOFLOAT(str : String [local : Integer]) : Float	15
	SUBSTR(s : String; first, len : Integer) : String	15
	TRIM(orig, trimchar : String; OpMode : Integer) : String	15
	USER([attribute : String]) : Variant *	15
	VAR(varName : String) : Variant.....	16
	YEAR(d : DateTime) : Integer.....	16
4	Examples.....	17

1 Contexts

Expressions only make sense, if used in a context. A context may define external values and additional functions. Currently expressions are used in these contexts:

- XDS variables
- FilterNavigation values
- Reporting action output parameter values

The XDS-context and the FilterNavigation-context provide additional functions which can be looked up [here](#) and [here](#).

2 Syntax

expression	= simpleexp [relop simpleexp] condexp.
condexp	= 'IF' BoolExp 'THEN' expression 'ELSE' expression.
simpleexp	= ['+' '-'] term { ('+' '-' 'OR') term }.1)
term	= factor { ('*' '/' '%' 'AND') factor }. 2)
factor	= ['NOT'] (Value Identifier function '(' expression ')').
function	= Functionname ['(' expression { ',' expression } ')'].
relop	= ('<' '<=' '=' '>' '>=' '<>').

BoolExp	Expression that returns a Boolean value.
Identifier	Name that is used as a placeholder for an external value (variable). The external value must be provided by the environment (e.g. in XDS a variable).
Value	Number, TextValue, date, 'TRUE', 'FALSE', 'NULL'
TextValue	String in ' or #n where n is the integer of the char code. #9 = Tab #13 = Cr #10 = Lf #34 = Doublequote codes greater 127: >= v4.11.5 #8364 = € (note U+20AC = 8364)
Functionname	One of the functions described below or a function defined.

Notes

- '+' ignores operands with value NULL
- '%' is the modulo operation see example >= V4.11.6

3 Predefined Functions

ARRAYELEMENTOF(array : Variant; index : Integer) : Variant (>= v5.5.2)

gives the element at the specified index in the array

The first element is at index=0.

If index is greater or equal the count of Element NULL is returned.

If index= -1 the size (count) of array is returned if array is not an array, -1 is returned. In any case an integer is returned.

If index= -2 a text containing information about the type of the array is returned. This can help to debug variables, it works also if array is not an array.

ATTRIBUTE(attribute : String; option : String) : Variant

Returns the value of an attribute of the current object. The current object is set by the VALUE and COMPARE statements prior to evaluating the expression. To filter for coordinates, add the axis to the attribute-identifier with a ".".

If option is set to 'AsText' the attribute is returned as a text representation: the enumeration value is returned not the index, numbers are formatted according to the model definition.

If option is set to 'AsInterval' the interval attribute is returned as interval, this is useful as input for INTERVALTOFLOAT. (>= v5.2.1)

Examples:

```
ATTRIBUTE('Object_Display_Kontext')
ATTRIBUTE('Position.x')
ATTRIBUTE('bisKL_LockState', 'AsText')
INTERVALTOFLOAT(ATTRIBUTE('bisTT_Repetition', 'AsInterval'), 17) (>= v5.2.1)
```

BASE64TOBYTES (base64String : String) : Variant (>= v5.7.5)

Converts the base64String to a byte array in the result.

BYTESTOBASE64 (bytes : Variant) : String (>= v5.7.5)

Converts the bytes to a base64String in the result.

BYTESTOGUIDSTR (bytes : Variant) : String (>= v5.7.5)

Converts the bytes to a guid string representation in the result.

E.g. {00000000-0000-0000-0000-000000000000}.

CLIPBOARD : String

Returns the content of the clipboard as text.

CONVERTIFCGUID (value : String) : String (>= v5.7.5)

Converts an [Ifc Guid](#) to a string that can be uniquely indexed in Byron/BIS (case insensitive) and back to an Ifc Guid. All lower case characters ('a-z') are escaped using the dash '-'.
Examples:

```
CONVERTIFCGUID('2yeSWdixz5hBpMywNJnu8J') => '2-y-eSW-d-i-x-z5-hB-pM-y-wNJ-n-u8J'
```

```
CONVERTIFCGUID('2-y-eSW-d-i-x-z5-hB-pM-y-wNJ-n-u8J') => '2yeSWdixz5hBpMywNJnu8J'
```

Can be used for strings that are not Ifc Guids, as long as they do not contain a dash '-' or lower case umlauts.

COUNT : Integer

Returns the number of objects

Example:

```
IF 'COUNT = 0' (  
  START USER  
ELSE  
  VIA Object_To_Parent  
)
```

DATE(y, m, d : Integer) : DateTime

Takes three integer arguments as year, month, day and returns a date.

Example:

```
DATE(2002, 7, 12) → 12.07.2002
```

DATETIMETO(d : DateTime; op : String) : DateTime

Extracts the date and time parts from the parameter "d" and set them to fixed values.
The parts for overriding are defined in the operation parameter "op".

You can use upper case letters ('YMDHNSX') to override one or more date and/or time parts:

'Y' set the year to 2004
'M' set the month to 7
'D' set the day to 1
'H' set the hour to 12
'N' set the minutes to 30
'S' set the seconds to 5
'X' set the milliseconds to 0 (default)

If you use lower case letters ('ymdhnsx'), the date and/or time part will not be overridden.

'y' let the year as is (default)
'm' let the month as is (default)
'd' let the day as is (default)
'h' let the hour as is (default)
'n' let the minutes as is (default)
's' let the seconds as is (default)
'x' let the milliseconds as is

The default value is 'ymdhnsX'

Examples

```
DATETIMETO("21.01.2009 16:57:33", "YMDHNSX") → "01.01.2004 12:30:05"  
DATETIMETO("21.01.2009 16:57:33", "ymdHNSX") → "21.01.2009 12:30:05"  
DATETIMETO("21.01.2009 16:57:33", "YMDhnsX") → "01.07.2004 16:57:33"
```

DAY(d : DateTime) : Integer

Returns the day number of a date.

Examples:

```
DAY(12.07.2002) → 12  
DAY(NULL) → 0
```

ENUMVALUE(attribute : String; lang : String;
index : Integer; option : String) : Variant

Returns the attribute's enumeration value with index as a text.

If the attribute is not found or not an enumeration, or if the index contains an invalid value, the result is an empty text.

If the language lang is not one of the supported Byron/BIS languages, the result is the integer value of the enumeration literal.

If the parameter option has the value 'isValue', index is interpreted as the enumeration's value instead of the index.

Examples:

```
ENUMVALUE('Offertstatus', 'DE', 1) → 'versendet'
ENUMVALUE('Offertstatus', 'XX', 1) → '20'
ENUMVALUE('Offertstatus', 'DE', 10, 'isValue') → 'erfasst'
ENUMVALUE('Offertstatus', 'XX', 10) → '10'
```

ENUMVALUES(attribute : String; lang : String;
separator : String) : Variant

Returns all enumeration literals of the attribute as a text separated by the separator given in the third parameter. If the attribute is not found or not an enumeration, the result is an empty text. If the language lang is not one of the supported Byron/BIS languages, the result is the list of the integer values of the enumeration literals.

Examples:

```
ENUMVALUES('Offertstatus', 'DE', ';')
→ 'erfasst;versendet;akzeptiert;abgelehnt'
ENUMVALUES('Offertstatus', 'XX', ';') → '10;20;30;40'
```

EVAL(v : String) : String

Returns the recursively evaluated text. Variables must be surrounded by %. The variable is first searched in the global parameter list (PARAM). If it is not found the environment variables (OSENv) are evaluated.

Example:

```
EVAL('%mydir%\list.txt')
Global Parameter: mydir = '%temp%\BIS'
OS-Environment: temp = 'C:\Users\thomas\AppData\Local\Temp'
→ C:\Users\thomas\AppData\Local\Temp\BIS\list.txt
```

EXTRACTNAME(fullFileName : String) : String

Extracts the filename from fullFileName. Strips the file path and the extension.

Example:

```
EXTRACTNAME('c:\temp\Hello.txt') → 'Hello'
```


FIELD(fullText : String; pos : Integer;
[sep : String; quote : String]) : String

Extracts the pos'th substring from fullText using sep as field separator. The first field has pos = 0.

Default: sep = #9 (tab)

Default: quote = #34 (") (>= v4.4.10)

FIELD(rec, pos, sep) is (if pos > 0), a short form of
SUBSTR(rec, STRPOS(sep, rec, pos), STRPOS(sep, rec, pos+1) - STRPOS(sep, rec, pos))

Examples:

```
FIELD('AA_B_CC', 1, '_') → 'B'
FIELD('aa;"bb;b";cc', 1, ';', #34) → bb;b
FIELD('aa;"bb;b";cc', 1, ';') → bb;b
FIELD('aa;"bb;b";cc', 1, ';', '') → "bb
```

FIELDBYNAME(line, header, field, [def, sep, quote : String]) : String

Search field in header.

If found the value of line at this position is returned.

If not found the value of def is returned, if def is undefined (=default=NULL) an exception is raised. Sep is the field separator and quote see function FIELD

Default: def = NULL (an exception is raised if field does not exist)

Default: sep = #9 (tab)

Default: quote = #34 (")

FILEAGE(fullFileName : String) : DateTime

Return the modification date of the fullFileName. If the file does not exist the result is "-1"

Example:

```
FILEAGE('c:\temp\Hello.txt') → 21.1.2005
```

FILECONTENTS(fullFileName : String) : String

Return the the whole contents of a file. Be careful in use it.

Example:

```
FILECONTENTS('c:\temp\Log.txt')
→ "started at 1.1.2007 ...and many line breaks..."
```

FILESIZE(fullFileName : String) : Float

Return the file size in bytes as float. If the file does not exists 0 is returned

FORMAT(v : Integer/Float/DateTime, format : String, useLocales: Boolean) : String
 converts v to the resulting text according the format.
 useLocales is optional (default TRUE).)
 NB: if v = NULL then FORMAT returns NULL

GUID : String
 without arguments, returns a new GUID.

GUIDSTRTOBYTES (guid : String) : String (>= v5.7.5)
 Converts a guid string representation to the bytes in the result.
 E.g. {00000000-0000-0000-0000-000000000000}.

HASLIC(aFeature : String) : Integer
 Returns true if the specified license feature is available.
 Example:
 HASLIC('Tool_MasterDataManagement')

HASRIGHT(class : String; theRight: String) : Integer
 Returns true if the current user has the specified right (obrRead, obrChange, obrCreate, obrDelete) for the objectclass.
 Example:
 HASRIGHT('Space', 'obrCreate')

IFNULL(v : Variant; def : Variant) : Variant
 If v is null then def is returned, else v is returned.
 Defaults: def = '' (Note '' is an empty string, not a NULL).
 IFNULL is a short form of 'IF n = NULL THEN def ELSE v'

INT(s : String/DateTime/Float) : Integer
 converts text to an integer. If the parameter is a DateTime, then the result is the truncated DateTime, not an integer!
 In the following example the number of days is kept in the global parameter in BIS.
 =NOW - INT(PARAM('PeriodOfNotice'))
 converts (truncates) numeric values (integer, dates, floats) to integer. The following example truncates the current date and time to the current date.
 =INT(NOW)

INTERVALTOFLOAT(interv : Interval; baseUnit : Integer) : Float

gives the value of the interval according the baseUnit.

BaseUnit is a number in the range 5..24

see Help: Darstellungsformate für Attribute - Zeit Intervalle.

baseUnit: 9 = hours, 13 = days, 22 = years.

LANGUAGE(range : String; type : Integer) : String/Integer

Returns the current language.

The range controls the possible result; this is defined by one char for each language. The first language is the fallback if none of the defined is current. If the range is empty the current language is returned, this is the default.

Example:

LANGUAGE('EDFI') = 'EN' if the current language is Spanish.

The type controls the return type:

type = 0: returns an integer 0=German, 1=English, 2=French, 3=Italian...

type = 1: returns text like 'D', 'E', 'F', 'I', 'T', 'S', 'R', 'C'

type = 2: returns text like 'DE', 'EN', 'FR', 'IT', 'TR', 'ES', 'RO', 'HR'

Default is type = 2

Example:

VALUE "= 'Name' + LANGUAGE" 'Ba*'

This filters NameDE or NameFR or NameEN, ... depending on the current language

MD5(s : String) : String

returns MD5 code as Hex. The resulting value is the same like MD5SUM.exe and other tools (php) in the internet.

MONTH(d : DateTime) : Integer

returns the month number of a date.

Examples:

MONTH(12.07.2002) → 7

MONTH(NULL) → 0

NOW : DateTime

without arguments, returns the current DateTime.

OBFUSCATE(val : Variant; mode : integer) : Variant

Returns val changed at random.
mode controls how val is changed.

If val is null, null is returned

If val is a string

mode=0: A string with the same length is returned containing any letters, first capital, others lowercase.

mode=1: A string with the same length is returned containing any letters, any case.

If val is a Date/Time

mode=0: result date is in the range +/- 200 days

mode=1: result date is in the range +/- 500 days

mode=2: result date is in the range +/- 50 days

mode=3: result date is in the range +/- 5 days

mode=4: result date is in the range +/- 12 hours

If val is a integer

the result is an integer in the range 0..val

Defaults: mode = 0

OBJCOUNT(association : String) : Integer

Returns the number of objects that are associated via the given association.

Example – get only the objects with children.

poor: COUNT (VIA "Object_To_Children") > 0

better: CONDITION '=OBJCOUNT("Object_To_Children") > 0'

OBJCOUNT does not retrieve the objects and therefore is faster than COUNT(VIA ...).

If the association is calculated with getter method, OBJCOUNT calculates the correct result. The performance in this case is the same as the performance of COUNT(VIA ...).

Defaults: mode = 0

OBJIMAGE : String *

(>= v5.1.1)

Returns the object images of all the objects accessible to the expression. If more than one object is accessible, object images are separated by a comma.

Example:

```
SAVEAS TRANS 'CurrentSelection' '=OBJIMAGE'
```

```
... in another Filternavigation...
```

```
START DATAOF 'CurrentSelection'
```

OSENV(v : String) : String

Operating System Environment returns the value of a operating system environment variable.

OSENV('BIS_ROOT') -> 'C:\programme\Byron\ByronBIS\'

This expression will also resolve any environment variable that is referenced in the value of the initial variable.

OVERLAY(background, arg : String; AtPos, align : Integer) : String

Lays arg over the background at AtPos according the align.

AtPos = 1 is the first character.

Align: 0=Left, 1 = Right, 2 = Centered.

Examples:

```
OVERLAY('123456789', 'AB', 1, 0) → 'AB3456789'
```

```
OVERLAY('123456789', 'AB', 9, 1) → '1234567AB'
```

```
OVERLAY('123456789', 'ABC', 5, 2) → '123ABC789'
```

Defaults: AtPos = 1, Align = 0

Note: Background and arg accept NULL too, which is interpreted as ''.

PARAM(v : String) : String *

Returns the value of a global database parameter.

Example:

```
LOG '=PARAM("ExtDocPath_Reservation")' 1
```

PARAM does not resolve additional environment variables or global parameters that are potentially referenced in the value of the global parameter.

Global parameters are managed in 'Arbeitsumgebung – Anpassen...'.

RANDOM(max : Float) : Float

Returns a value v in the range $0 \leq v < \text{max}$
default for max = 1.0.

Example:

```
INT(RANDOM(count)) = 0 .. count - 1
```

REGEXP(test : String; pattern : String; options : String) : Boolean

returns true if test matches the regular expression pattern. If options contains a 'c' both (test and pattern) are converted to uppercase. More about regular expression:

http://de.wikipedia.org/wiki/Regul%C3%A4rer_Ausdruck (Deutsch)

<http://www.regular-expressions.info> (English)

http://www.regenechsen.de/phpwcms/index.php?regex_allg (Kurs Deutsch)

Example:

```
REGEXP('Meier', '^Me.*') → true
```

STRCASE(str : String; op : Integer) : String

Changes the case of str according the operation code in op:

op = 0 all to lowercase

op = 1 all to uppercase

op = 10 first to upper case, the second to the end to lowercase.

Examples:

STRCASE('aBcD', 0) → 'abcd'

STRCASE('aBcD', 1) → 'ABCD'

STRCASE('aBcD', 10) → 'Abcd'

STRLEN(str : String) : Integer

Returns the length of str.

Examples:

STRLEN('Hello') → 5

STRLEN('') → 0

STRPOS(sub, str : String; [times : Integer; quote : String]) returns Integer

returns the index, where sub starts within str. Times gives the occurrence (1 is default and gives the first occurrence). Times = 0 returns the count of the sub and negative times starts at the right side, that is times -1 is the first occurrence starting at the right side.
quote (default empty) controls that sub within quotes are not counted.

Examples:

STRPOS('_', 'A_B_C_D') → 2

STRPOS('_', 'A_B_C_D', 2) → 4 (second occurrence of '_')

STRPOS('_', 'A_B_C_D', 0) → 3 (there are 3 '_' in the text)

STRPOS('_', 'A_B_C_D', -1) → 6 (there are 3 '_' in the text)

STRPOS(';', 'A;"B;b";C;D', 0, #32) → 3 ("B;b" is quoted by #32 so the contained ';' is not counted)

STRREPL(orig, old, new : String) : String

Returns a string like orig, but all old pattern replaced by new

Example:

STRREPL('abcdcbc', 'bc', '-') → 'a-dc-'

STRTODATE(str : String [format : String]) : DateTime

Converts a text to a date value. Optionally a format can be given.

STRTOFLOAT(str : String [local : Integer]) : Float

Converts a text to a float value. local = 0 (default) uses '.' as the decimal separator. local = 1 uses the actual separator. An empty test returns 0.0

SUBSTR(s : String; first, len : Integer) : String

Returns a substring (first >= 1)

Example:

SUBSTR('HELLO', 3, 2) → 'LL'

TRIM(orig, trimchar : String; OpMode : Integer) : String

Returns orig changed according trimchar and OpMode. Trimchars is a text containing those characters which are extracted or accepted. OpMode defines two things: 1. where is trimmed and 2. whether trimchars is a positive or negative list. Add Op1 and Op2 to get your OpMode:

Op1: Left = 0 Right = 1 LeftRight = 2 WholeText = 3

Op2: ExtractTrimChars = 0 TrimCharsOnly = 10

In Trimchar characters may be defined by their code, using a '#'. Consecutive characters can be shortened by .. ('0123456789' can be written as '0'..'9').

Defaults: trimchar = ' ', OpMode = 0

Examples:

TRIM(' +41 61 6909600 ', ' ', 0) = '+41 61 6909600 '

TRIM(' +41 61 6909600 ', ' ', 1) = ' +41 61 6909600 '

TRIM(' +41 61 6909600 ', ' ', 2) = '+41 61 6909600 '

TRIM(' +41 61 6909600 ', ' ', 3) = '+41616909600 '

TRIM(' +41 61 6909600 ', '0..9', 13) = '41616909600 '

USER([attribute : String]) : Variant *

Returns the value of an attribute of the current user. If no attribute name is given, the user name is returned.

VAR(varName : String) : Variant

Returns the value of a variable. This function is used if the variable name is not-standard. (e.g. contains a period).

normal case:

```
... SAVEAS obj "=ATTRIBUTE('bisB_ObjectImage')" ...  
... CONDITION "=ATTRIBUTE('bisB_ObjectImage') = obj"
```

this is easier and equal to

```
... SAVEAS obj "=ATTRIBUTE('bisB_ObjectImage')" ...  
... CONDITION "=ATTRIBUTE('bisB_ObjectImage') = VAR('obj ')"
```

but if there is a name with a period you need:

```
... SAVEAS "o.bj" "=ATTRIBUTE('bisB_ObjectImage')" ...  
... CONDITION "=ATTRIBUTE('bisB_ObjectImage') = VAR('o.bj ')"
```

because this expression is invalid:

```
... CONDITION "=ATTRIBUTE('bisB_ObjectImage') = o.bj"
```

and this expression compares with the text 'o.bj' not the value of the variable:

```
... CONDITION "=ATTRIBUTE('bisB_ObjectImage') = 'o.bj'"
```

Example in tool configuration:

```
IF "VAR('formNav.VAL_CurrentFunction') = 'anforderer'" (  
  -- something  
)
```

YEAR(d : DateTime) : Integer

returns the year number of a date

Examples:

```
YEAR(12.07.2002) → 2002
```

```
YEAR(NULL) → 0
```

* These expressions are only available in database FilterNavigation. They are usually evaluated only once per execution. If an "ATTRIBUTE" function is used, expression are evaluated for every object.

4 Examples

'line1' + #13 + #10 + 'second line'

OSENV('Temp') + '\xdslist.txt'

FORMAT(VonDatum, 'dddd d.mmmm.yyyy')

FORMAT(NOW, 'dddd d.mmmm.yyyy')

STRREPL(EXTRACTNAME(varFILENAME), '_', '.')

IF NOW > DATE(2005, 12, 31) THEN 'A' ELSE 'B'

For week day use the modulo 7 operation (Monday = 0.. Sunday = 6)

(NOW - 2) % 7