



Focus Forward 2020

2020 Harris Govern
Training Conference

CEU Registration

- CEU Credits for:
 - IAAO
 - TAAD
 - TACA
- In order to receive CEU Credit, you must sign in for the session:
 - Attendee may use Whoova App to sign-in; OR
 - Sign Attendee Sign-in Sheet in class

Diamond Sponsors

 InvoiceCloud® CERTIFIED
PAYMENTS

Platinum Sponsors



Gold Sponsors

 EAGLEVIEW® point
& pay

Expressions in OpenForms

Eric Martineau

What are the Expressions?

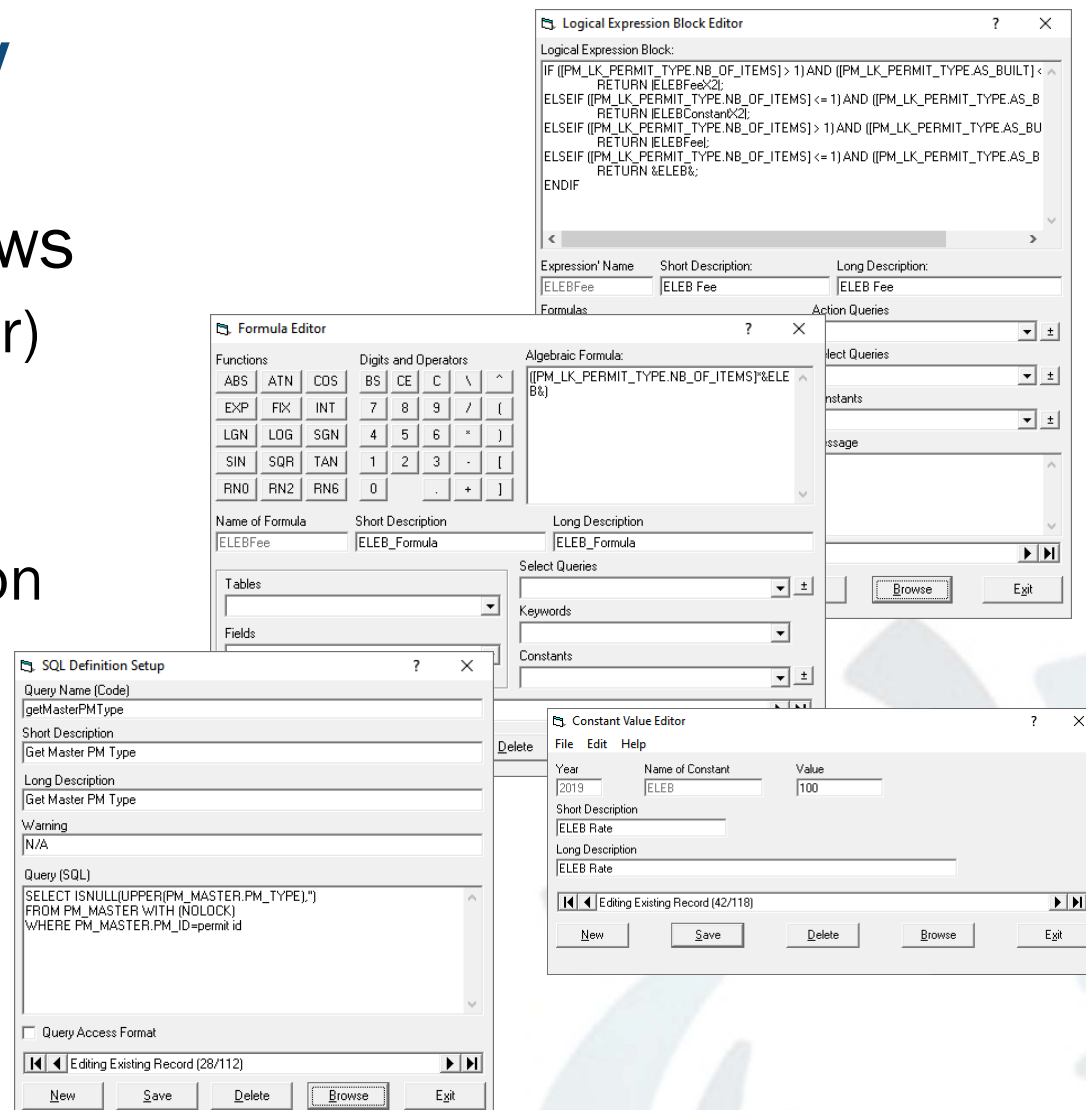
The Govern Expressions is a scripting language that let you use a series of functions and operators to create custom value calculations or application behaviors.

Where can we use Expressions?

- Govern New Admin
 - Ribbon Tile Properties
 - Email & Message Template
 - Module Configuration
 - Ex. : Levies & Exemptions, Workflow Actions, etc.
- Business Entity Designer
 - Calculated Fields
 - Default Values
 - Validation Rules
- OpenForms Designer
 - Tab Properties
 - Ex.: Allow Insert, Allow Update, Allow Delete, Record Description
 - Object Properties
 - Ex. : Is Enabled, Is Visible, Text Style, Value

A Bit of History

- Govern for Windows
 - Constant (By Year)
 - Queries
 - Formulas
 - Logical Expression



Now and the Future...

- Govern.NET
 - Constant (By Year and/or Jurisdiction)
 - Queries (By Year and/or Jurisdiction)
 - Expressions (By Year and/or Jurisdiction)
 - **Formulas ***
 - **Logical Expression ***

Constants Editor

User Constant    

▼ BONDVAL - Bond Value

☒ By Year ☐ By Jurisdiction

^ Values By Year/Jurisdiction   Year: '0'

Drag a column header here to group by that column.

Year ID	Value
0	0.08
2016	0.08
2017	0.08
2018	0.08
2019	0.08

Value

^ Used In

Drag a column header here to group by that column.

Group	Description
-------	-------------

SQL Queries Editor

Select Queries

arAcList - A/R - Get list of all AR_ID related to Aircraft for a year

☐ By Year ☐ By Jurisdiction

Context Alternate Connection Key ☒ Is System

Query

```
SELECT
AR_ID as Code,
CAST(AR_ID as Varchar(10))+ ' (' + CYCLE_CODE + ')' as SHORT_DESC_EN
FROM AR_MASTER
WHERE AC_ID =@idac_id
AND YEAR_ID = @idyear_id
AND SUB_SYSTEM = 'AC'
```

Evaluate

Result

Execute

System ID test values

ac_id

year_id

Used In

Drag a column header here to group by that column.

Group	Description
-------	-------------

Shared Expressions Editor

Shared Expression

GetTifLevyAmt - Get TIF levy amount.

☐ By Year ☐ By Jurisdiction

Context ☒ Is System

Shared Expression

☒ Convert Null to Default

```
{percentage = IsNull(Query('GetTifPercent'),0)}  
{netDistributed = IsNull(Query('GetTifNetDistr'),0)}  
{levyAmount = netDistributed * percentage / 100 }  
  
if(levyAmount < 0, 0, levyAmount)}
```

Evaluate

Result

Execute

year_id

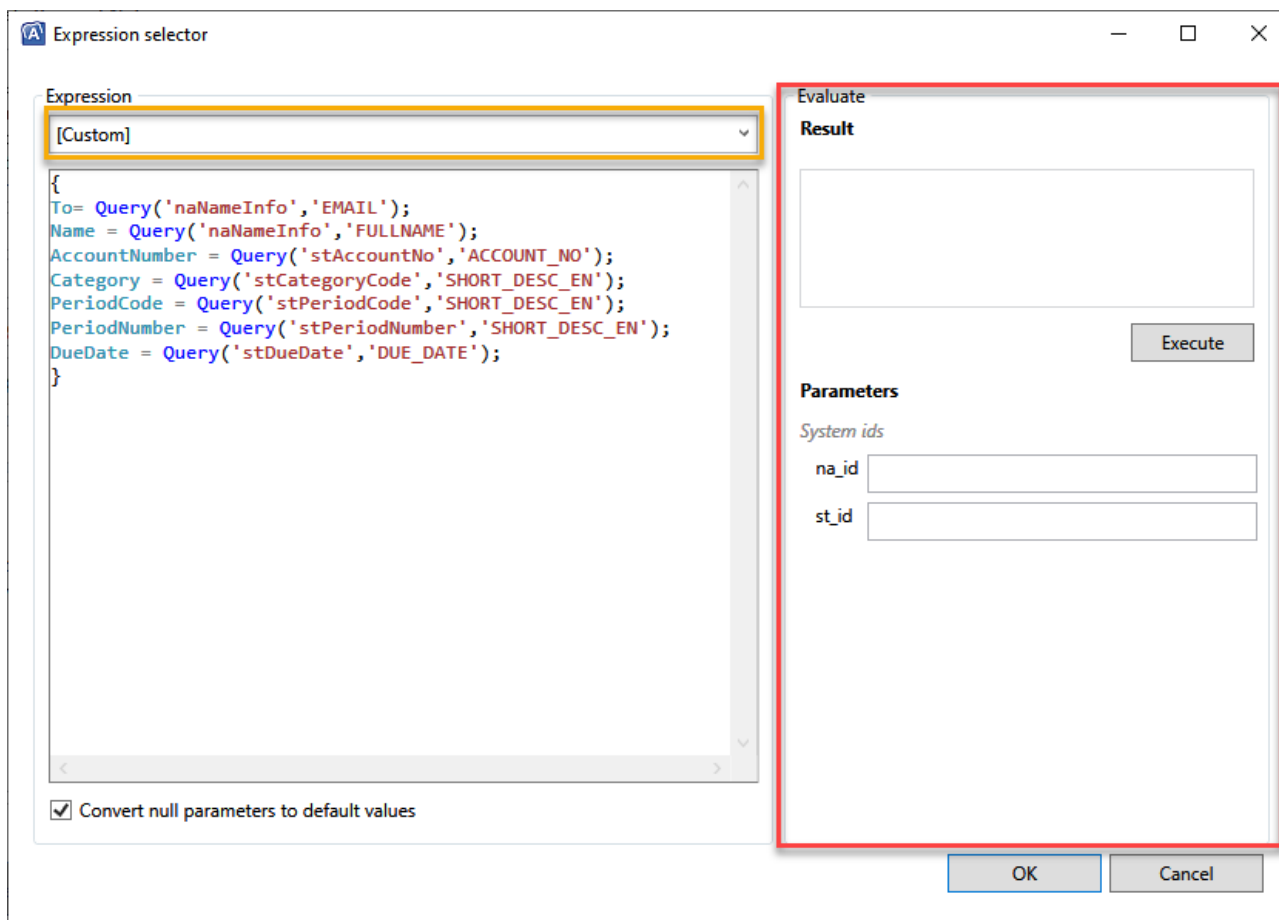
p_id

Used In

Drag a column header here to group by that column.

Group	Description
-------	-------------

Expression Editor



The screenshot shows the 'Expression selector' window. The 'Expression' dropdown is set to '[Custom]'. The main text area contains a JavaScript-like expression: `{ To= Query('naNameInfo','EMAIL'); Name = Query('naNameInfo','FULLNAME'); AccountNumber = Query('stAccountNo','ACCOUNT_NO'); Category = Query('stCategoryCode','SHORT_DESC_EN'); PeriodCode = Query('stPeriodCode','SHORT_DESC_EN'); PeriodNumber = Query('stPeriodNumber','SHORT_DESC_EN'); DueDate = Query('stDueDate','DUE_DATE'); }`. A checkbox at the bottom left is checked and labeled 'Convert null parameters to default values'. On the right, the 'Evaluate' panel is highlighted with a red border. It contains a 'Result' field, an 'Execute' button, and a 'Parameters' section with 'System ids' and input fields for 'na_id' and 'st_id'. At the bottom of the window are 'OK' and 'Cancel' buttons.

Expression selector

Expression

[Custom]

```
{  
To= Query('naNameInfo','EMAIL');  
Name = Query('naNameInfo','FULLNAME');  
AccountNumber = Query('stAccountNo','ACCOUNT_NO');  
Category = Query('stCategoryCode','SHORT_DESC_EN');  
PeriodCode = Query('stPeriodCode','SHORT_DESC_EN');  
PeriodNumber = Query('stPeriodNumber','SHORT_DESC_EN');  
DueDate = Query('stDueDate','DUE_DATE');  
}
```

☒ Convert null parameters to default values

Evaluate

Result

Execute

Parameters

System ids

na_id

st_id

OK Cancel

Expression Basic Syntax

Element

Comments

Variables

String Value

Numeric Value

Boolean Value

Attribute Value

Constant Value

Govern ID Value

Syntax

-- user comment

{variablename = value}

'Text'

10 / 3.141592

True / False

@attrAttributeName

@ctConstantCode

@idKeyName

```
-- Comments
{variable0 = 'Text Value'}
{variable1 = 10}
{variable2 = @attrNA_ID}
{variable3 = @ctBONDVAL}
{variable4 = @idp_id}
{variable5 = True}
```

Expression Basic Functions

Function

Case

Syntax

`Case(expression, case1, value1, ..., case#, value#, else)`

Shared Expression

☒ Convert Null to Default

```
--Case Syntax
{Number1 = 3}

Case(Number1,
  1, 'Value 1',
  2, 'Value 2',
  3, 'Value 3',
  4, 'Value 4',
  'Else')
```

Evaluate

Result

Value 3

Execute

Expression Basic Functions

Function

If

Syntax

If(condition, resultTrue, resultFalse)

Shared Expression

☒ Convert Null to Default

```
--If Syntax
{Output = 'Text'}
{Number1 = 10}
{Number2 = 11}

If (Output = 'Text',
  If (Number1=Number2,'Value are Equal','Value are not Equal'),
  If (Number1=Number2,True,False)
)
```

Evaluate

Result

Value are not Equal

Execute

Expression Comparison Operator

Function

=

< / <=

> / >=

<> / !=

In

Syntax

expression1 = expression2

expression1 < expression2 / expression1 <= expression2

expression1 > expression2 / expression1 >= expression2

expression1 <> expression2 / expression1 != expression2

In(expression, value1, ...,value#)

Shared Expression

☒ Convert Null to Default

```
-- Comparison Operator
-- = / < / <= / > / >= / <> / !=
{
variable1 = 12;
variable2 = 25;

result1 = If (variable1 = variable2, '1 is True', '1 is False');
result2 = If (variable1 > variable2, '2 is True', '2 is False');
result3 = If (variable1 >= variable2, '3 is True', '3 is False');
result4 = If (variable1 < variable2, '4 is True', '4 is False');
result5 = If (variable1 <= variable2, '5 is True', '5 is False');
result6 = If (variable1 <> variable2, '6 is True', '6 is False');
result7 = If (variable1 != variable2, '7 is True', '7 is False');
result8 = If (In(variable1,1,2,3,4,5), '8 is True', '9 is False');
}

result1 + ' / ' + result2 + ' / ' + result3 + ' / ' + result4 + '\n' +
result5 + ' / ' + result6 + ' / ' + result7 + ' / ' + result8
```

Evaluate

Result

1 is False / 2 is False / 3 is False / 4 is True
5 is True / 6 is True / 7 is True / 9 is False

Execute

Expression Logical Operator

Function

Not

And

OR

Syntax

NOT condition

condition1 AND condition2

condition1 OR condition2

Shared Expression

☒ Convert Null to Default

```
-- Logical Operator
-- NOT / AND / OR
{
variable0 = 10;
variable1 = 11;
variable2 = 12;

result1 = If ((variable0 < variable1) AND (variable1 > variable2),'1 is True','1 is False');
result2 = If ((variable0 < variable1) OR (variable1 > variable2),'2 is True','2 is False');
result3 = If (NOT((variable0 = variable1) AND (variable1 = variable2)),'3 is True','3 is False');
}

result1 + '\n' + result2 + '\n' + result3
```

Evaluate

Result

Execute

Expression NULL Value Functions

Function

Syntax

HasValue

`HasValue(expression)`

IsNull

`IsNull(expression, value)`

IsNullOrEmpty

`IsNullOrEmpty(expression, value)`

IsNullOrWhiteSpace

`IsNullOrWhiteSpace(expression, value)`

Shared Expression

☐ Convert Null to Default 

```
--Null Management
-- IsNull / IsNullOrEmpty / IsNull
{
  variable1 = '  ';
  variable2 = '';

  result1 = IsNullOrWhiteSpace(variable1,'Contain WhiteSpace');
  result2 = IsNullOrEmpty(variable2,'Is Empty');
  result3 = IsNull(@idp_id,-1);
  result4 = HasValue(@idp_id);
}

result1 + ' / ' + result2 + ' / ' + result3 + ' / ' + result4
```

Evaluate

Result

Contain WhiteSpace / Is Empty / -1 / False

Execute

Parameters

System ids

p_id

Expression Basic Math Operator

Function

+ (Addition)

- (Subtraction)

* (Multiplication)

/ (Division)

% (Modulo)

Syntax

expression1 + expression2

expression1 - expression2

expression1 * expression2

expression1 / expression2

expression1 % expression2

Shared Expression

☒ Convert Null to Default

```
--Basic Math
{
Number1=27;
Number2=8;

result1 = Number1 + Number2;
result2 = Number1 - Number2;
result3 = Number1 * Number2;
result4 = Number1 / Number2;
result5 = Number1 % Number2;
}

Str(result1) + ' / ' + Str(result2) + ' / ' + Str(result3) + ' / ' + Str(result4) + ' / ' + Str(result5)
```

Evaluate

Result

35 / 19 / 216 / 3.375 / 3

Execute

Expression Math Functions

<u>Function</u>	<u>Syntax</u>
Abs	Abs(number)
Acos / Asin / Atan	Acos(number) / Asin(number) / Atan(number)
Cos / Sin / Tan	Cos(number) / Sin(number) / Tan(number)
Ceiling / Floor	Ceiling(number) / Floor(number)
Exp	Exp(number)
IEEERemainder	IEEERemainder(number1,number2)
Log / Log10 / Ln	Log(number, log) / Log10(number) / Ln(number)
Min / Max	Min(number1, number2) / Max(number1, number2)
Pow	Pow(number, power)
Round / Truncate	Round(number, #digit) / Truncate(number)
Sqrt	Sqrt(number)
Sign	Sign(number)

Expression String Functions

Function

Syntax

Len

Len('string')

Lower

Lower('string')

Str

Str(number)

SubStr

SubStr('string', position, length)

Upper

Upper('string')

+ (Concatenate)

'string1' + ... + 'string#'

Shared Expression

☒ Convert Null to Default

```
--String Function
-- Len / Lower / Str / Substr / Upper
{
variable1 = 'Govern.NET';
variable2 = 5;

result1 = Lower(variable1);
result2 = Upper(variable1);
result3 = Substr(variable1,0,6);
result4 = Len(variable1);
result5 = Str(variable2);
}

result1 + ' / ' + result2 + ' / ' + result3 + ' / ' + result4 + ' / ' + result5
```

Evaluate

Result

govern.net / GOVERN.NET / Govern / 10 / 5

Execute

Expression Date Functions

<u>Function</u>	<u>Syntax</u>
Date	<code>Date(date)</code>
Now	<code>Now()</code>
DateAdd	<code>DateAdd(part, value, date)</code>
DateDiff	<code>DateDiff(part, date1, date2)</code>
Day / Month / Year	<code>Day(date) / Month(date) / Year(date)</code>

Shared Expression
☒ Convert Null to Default

```
--Date Function
-- Date / DateAdd / DateDiff / Day / Month / Now / Year
-- Format
{
variable1 = Date(@idsystem_date);
variable2 = Now();

result1 = DateAdd('year',1,variable1);
result2 = DateAdd('month',2,result1);
result3 = DateAdd('day',-7,result2);
result4 = DateDiff('day',variable1,result3);
result5 = Str(Year(result3)) + '-' + Str(Month(result3)) + '-' + Str(Day(result3));
}
Format(result1,'D') + '\n' + Format(result2,'D') + '\n' + Format(result3,'D') + '\n' +
Format(result4,'D') + '\n' + result5
```

Evaluate

Result

Friday, January 1, 2021
Monday, March 1, 2021
Monday, February 22, 2021
418
2021-2-22

Execute

Parameters

System ids

system_date 1/1/2020 12:00:00 AM

Expression Special Functions

<u>Function</u>	<u>Syntax</u>
Display	<code>@attrAttributName.Display</code>
Double	<code>Double('number')</code>
Format	<code>Format(expression, 'code') /</code> <code>Format(expression, 'format')</code>
Query	<code>Query('QueryCode') /</code> <code>Query('QueryCode', #minutecache) /</code> <code>Query('QueryCode', 'ColumnName') /</code> <code>Query('QueryCode', 'ColumnName', #minutecache)</code>

Conversion Example

From Govern for Windows...

Logical Expression Block Editor

Logical Expression Block:

```
IF [PM_LK_PERMIT_TYPE.NB_OF_ITEMS] > 1 THEN
  RETURN [BLDB];
ELSE
  RETURN &BLDB&;
ENDIF
```

Expression Name: [BLDB] Short Description: [BLDB] Long Description: [BLDB]

Formulas: [] Action Queries: []

Keywords: [] Select Queries: []

Constants: []

Tables: [] Fields: []

Message: []

Editing Existing Record (5/205)

New Save Delete Browse Exit

Formula Editor

Functions: ABS ATN COS EXP FIX INT LGN LOG SGN SIN SQR TAN RN0 RN2 RN6

Digits and Operators: BS CE C \ ^ 7 8 9 / (4 5 6 *) 1 2 3 - [0 . +]

Algebraic Formula: ([PM_LK_PERMIT_TYPE.NB_OF_ITEMS]&BLDB&)

Name of Formula: [BLDB] Short Description: [BLDB_Formula] Long Description: [BLDB_Formula]

Tables: [] Fields: []

Select Queries: [] Keywords: []

Constants: [] BLDB Rate: []

Consider Null as 0: ☒

Editing Existing Record (32/665)

New Save Delete Browse Exit

Constant Value Editor

File Edit Help

Year: [2019] Name of Constant: [BLDB] Value: [150]

Short Description: [BLDB Rate]

Long Description: [BLDB Rate]

Editing Existing Record (9/118)

New Save Delete Browse Exit

...to Govern.NET

User Constant

BLDB - BLD8 Rate

☒ By Year ☐ By Jurisdiction

Values By Year/Jurisdiction Year: '2020'

Drag a column header here to group by that column.	
Year ID	Value
2020	175
2019	150
2018	125

Shared Expression

BLDB - BLD8 Fee

☐ By Year ☐ By Jurisdiction

Context: PM_PERMIT - Building Permit ☐ Is System

Shared Expression

☒ Convert Null to Default

```
-- BLD8 Fee
If (@attrNB_OF_ITEMS > 1,
    @attrNB_OF_ITEMS * @ctBLDB,
    @ctBLDB)
```

Evaluate

Result

150

Parameters

System ids

year_id 2019

Attributes

nb_of_items 1

User constants

bldb 150

From Govern for Windows...

CONSTANT

BLDB 150

FORMULA

BLDB ([PM_LK_PERMIT_TYPE.NB_OF_ITEMS]*&BLDB&)

LOGICAL EXPRESSION

BLDB IF [PM_LK_PERMIT_TYPE.NB_OF_ITEMS] > 1 THEN
RETURN |BLDB|;
ELSE
RETURN &BLDB&;
ENDIF

...to Govern.NET

CONSTANT

BLDB 150

EXPRESSION

BLDB

```
-- BLDB Fee  
If (@attrNB_OF_ITEMS > 1,  
    @attrNB_OF_ITEMS * @ctBLDB,  
    @ctBLDB)
```

Questions

References

- Microsoft: Numeric Format Codes
 - [https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2008/y006s0cz\(v=vs.90\)](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2008/y006s0cz(v=vs.90))
- Microsoft: Date Format Codes
 - [https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2008/362bt8f\(v=vs.90\)](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2008/362bt8f(v=vs.90))

References

- Harris Govern KB: Expressions
 - <https://kb.harrisgovern.com/100-expressions/>

Thank you