# DOCUMENT

## General description

This document describes ArchestrA templates developed for the Process Data Management of RX59 project. It does not include ArchestrA Callisto library objects.  
Some templates have been adapted/modified from the ArchestrA Callisto Library, others have been totally developed. These new objects are in conformance with the Callisto methodology.

## User Defined

### User defined templates

#### Template ArchestrA $AADWGenBoutonsBoolTest

##### Description

Not Applicable

##### Derived from

$dwGenObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bPlcStatus\_FB | boolean |  |  |  |  |  |
| bStatus | boolean |  |  |  |  |  |
| guy | boolean |  |  |  |  |  |
| iTypeAnimationAlternative | integer |  |  |  |  |  |
| iTypeCommande | integer |  |  |  |  |  |
| iValCommande | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |
| teller | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenBoutonsBool  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
logmessage("Instance : " + Me.Tagname + " : Script ObjectDeploy is being executed") ;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Logmessage("Instance : " + Me.Tagname + " : AffectInputSource is activated in script ObjectDeploy") ;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenBoutonsBool  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.bPlcStatus.Input.InputSource = "Me.bPlcStatus\_FB";  
  
logmessage("instance : " + Me.Tagname + " : Script AffectInputSource executed") ;

dcPlcStatus

|  |  |
| --- | --- |
| Name | dcPlcStatus |
| Description |  |
| Trigger | DataChange of Me.bPlcStatus + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 16 Avril 2008  
' Ce script traitera les status  
  
logmessage("Script : dcPlcStatus Test");  
logmessage("Instance : " + Me.TagName);  
logmessage("Input source of bPLCStatus : " + Me.bPlcStatus.Input.InputSource) ;  
  
if Me.bPLCStatus == 1 then  
 logmessage("bPlcStatus : " + "True");  
else  
 logmessage("bPlcStatus : " + "False");  
endif ;  
  
  
logmessage("iTypeAnimation : " + StringFromIntg( Me.iTypeAnimation, 10 ));  
  
  
IF Me.iTypeAnimation == 10 OR Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 60 THEN ' PFEIFFER - BR8 OR BR6 OR CAC/FMT  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 0; ' vert  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 1; ' rouge  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 0 and me.koekebakkevlaaie == 0 THEN ' BR7  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 0; ' vert  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 1; ' rouge  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 1 THEN ' BR7\_CHOIX\_TEMP  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 1; ' rouge  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 0; ' vert  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 0 THEN ' HGS  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 0; ' vert  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 1; ' rouge  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 1 THEN ' HGS\_VIS7CM  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 1; ' rouge  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 0; ' vert  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 40 AND Me.iTypeAnimationAlternative == 0 THEN ' BRO  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 0; ' vert  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 1; ' rouge  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 0) ' CAC/FMT  
OR (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 0)   
OR (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 0)   
THEN ' TSFx  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 0; ' vert  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 1; ' rouge  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 1)  
OR (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 1)   
OR (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 1)   
THEN   
' FMTx\_GAZ\_MANU, FMTx\_REG\_GAZ\_MANU, FMTx\_AIR\_PRIM\_MANU, FMTx\_AIR\_SEC\_MANU, FMTx\_ALV\_CENT\_MANU, FMTx\_ALV\_EXT1\_MANU, FMTx\_ALV\_EXT2\_MANU  
' TSFS\_FR4\_CHAUX, TSFS\_S11\_CHAUX  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 1; ' rouge  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 0; ' vert  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 40 AND Me.iTypeAnimationAlternative == 1 THEN ' BRO\_DECHARG\_AUTORISE\_FILLER  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 1; ' rouge  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 0; ' vert  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN   
  
logmessage("Script dPLCStatus executed") ;

dcRAZCmd

|  |  |
| --- | --- |
| Name | dcRAZCmd |
| Description |  |
| Trigger | DataChange of Me.bPlcStatus |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Juillet 2008  
' Ce script remettra le bit de commande à zéro pour certaines applications  
' iTypeAnimation : 50 = BR6  
  
IF (Me.bPlcCommande == 1 OR Me.iPlcCommande <> 0) AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70) THEN  
 Me.bPlcCommande = 0;  
 Me.iPlcCommande = 0;  
ENDIF;  
  
Me.iTempo = 0;  
  
logmessage("Instance : " + Me.Tagname + " : Script dcRAZCmd executed") ;

RAZ\_Tempo

|  |  |
| --- | --- |
| Name | RAZ\_Tempo |
| Description |  |
| Trigger | WhileTrue of Me.iTempo > 15 |

**Declarations :**

Not Applicable

**Script :**

IF (Me.bPlcCommande == 1 OR Me.iPlcCommande <> 0) AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70) THEN  
 Me.bPlcCommande = 0;  
 Me.iPlcCommande = 0;  
ENDIF;  
  
Me.iTempo = 0;  
  
logmessage("Instance : " + Me.Tagname + " : Script RAZ\_Tempo executed") ;

test

|  |  |
| --- | --- |
| Name | test |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

me.guy = not me.guy ;

wtCommande

|  |  |
| --- | --- |
| Name | wtCommande |
| Description |  |
| Trigger | WhileTrue of Me.bPlcCommande |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommande == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

wtCommandeInt

|  |  |
| --- | --- |
| Name | wtCommandeInt |
| Description |  |
| Trigger | WhileTrue of Me.iPlcCommande <> 0 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommande == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

teller

|  |  |
| --- | --- |
| Name | teller |
| Description |  |
| Trigger | DataChange of me.guy |

**Declarations :**

Not Applicable

**Script :**

me.teller = me.teller + 1 ;

#### Template ArchestrA $aINDEFF\_UserDefined\_V\_1\_0

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$mINDEFF\_UserDefined\_V\_1\_0

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $Alimentateur

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$Moteur1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $Alveolaire

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$Moteur1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Assign\_IO

|  |  |
| --- | --- |
| Name | Assign\_IO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $Bandeau

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

Not Applicable

##### Field Attributes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Attribute name | Data Type | IO Scaling | History | Limit Alarms | Rate Of Change Alarms | Target Deviation Alarms | Bad Value Alarm | Statistics |
| Anim | boolean |  |  |  |  |  |  |  |
| ACK | boolean |  |  |  |  |  |  |  |
| SynchDateHeure | boolean |  |  |  |  |  |  |  |
| FinDelestage | boolean |  |  |  |  |  |  |  |
| ACK\_GEN | boolean |  |  |  |  |  |  |  |
| AlarmeHeureHorsLimite | boolean |  |  |  |  |  |  |  |
| AlarmeFinPasIndique | boolean |  |  |  |  |  |  |  |
| MoisAutomate | integer |  |  |  |  |  |  |  |
| AnneeAutomate | integer |  |  |  |  |  |  |  |
| JourAutomate | integer |  |  |  |  |  |  |  |
| HeureAutomate | integer |  |  |  |  |  |  |  |
| MinuteAutomate | integer |  |  |  |  |  |  |  |
| SecondeAutomate | integer |  |  |  |  |  |  |  |
| TpsDelestage\_Heure | integer |  |  |  |  |  |  |  |
| TpsDelestage\_Minute | integer |  |  |  |  |  |  |  |
| TpsDelestage\_Seconde | integer |  |  |  |  |  |  |  |

##### Scripts

Not Applicable

#### Template ArchestrA $Boolean

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$FieldReference

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $caIDF\_OEESettings

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$cmIDF\_Config

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| P.TimeToLogAvailability | integer |  |  |  |  |  |
| P.TimeToLogDownTime | integer |  |  |  |  |  |
| P.TimeToLogProdCount | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $caIDF\_SQLConnectionString

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$cmIDF\_Config

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| cfg\_ConnectionString | string |  |  |  |  |  |
| obp\_ConnectionState | string |  |  |  |  |  |
| obp\_Database | string |  |  |  |  |  |
| obp\_DisplaySettings | string |  |  |  |  |  |
| obp\_Environment | string |  |  |  |  |  |
| obp\_ErrorMessage | string |  |  |  |  |  |
| obp\_ErrorState | boolean |  |  |  |  |  |
| obp\_ErrorTimeSec | integer |  |  |  |  |  |
| obp\_Server | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

TestConnection

|  |  |
| --- | --- |
| Name | TestConnection |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Dim spProcessor AS Indeff\_MES.DatabaseAccess.SpProcessor;  
Dim con AS System.Data.SqlClient.SqlConnection;  
Dim counter AS Integer;

**Script :**

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
-- Purpose : Tests the connectionstring.  
-- Author : Stijn Van Uytfanghe (Indeff)  
-- Version : 0.1  
-- Date : 2008-11-05  
-- History : 0.1 SUYT - INDEFF 2008-11-05 Original Version  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}  
  
spProcessor = new Indeff\_MES.DatabaseAccess.SpProcessor("DUMMY", me.Cfg\_ConnectionString);  
  
IF spProcessor.CheckConnection() == false THEN  
 'no connection  
 me.obp\_ConnectionState = "ERROR";  
 me.obp\_ErrorState = True;  
 me.obp\_ErrorMessage = spProcessor.Error.ErrorMessage;  
 me.obp\_ErrorTimeSec = me.obp\_ErrorTimeSec + 1;  
 LogMessage("DATABASE CONNECTION ERROR: " + me.obp\_ErrorMessage);  
  
ELSE  
 'connection ok  
 me.obp\_ConnectionState = "OK";  
 me.obp\_ErrorMessage = "Database connection OK";  
 me.obp\_ErrorState = False;  
 me.obp\_ErrorTimeSec = 0;  
  
 'use counter to log a message eacht 60 seconds  
 counter = counter + 1;  
 IF counter == 60 THEN  
 counter = 0;  
 LogMessage("DATABASE CONNECTION OK");  
 ENDIF;  
ENDIF;  
  
spProcessor.Dispose();  
  
  
'copy connectionstring settings to UDA's  
con = new System.Data.SqlClient.SqlConnection(me.Cfg\_ConnectionString);  
me.obp\_Server = con.DataSource;  
me.obp\_Database = con.Database;  
  
IF me.obp\_Server == "NLABLNAB005.NAB.local" OR me.obp\_Server == "10.60.109.5" THEN  
 me.obp\_Environment = "LIVE";  
ELSEIF me.obp\_Server == "NLABLNAB006.NAB.local" OR me.obp\_Server == "10.60.109.6" THEN  
 me.obp\_environment = "DEVELOPMENT";  
ELSE  
 me.obp\_environment = "UNKNOWN";  
ENDIF;

#### Template ArchestrA $Chaux\_10\_50

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwProduits

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| AL2O3 | float |  |  |  |  | X |
| CAO | float |  |  |  |  | X |
| CO2 | float |  |  |  |  | X |
| FE2O3 | float |  |  |  |  | X |
| MGO | float |  |  |  |  | X |
| R\_R | float |  |  |  |  | X |
| S | float |  |  |  |  | X |
| SIO2 | float |  |  |  |  | X |
| T60 | float |  |  |  |  | X |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $Chaux\_2\_10

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwProduits

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| CO2\_2 | float |  |  |  |  | X |
| S | float |  |  |  |  | X |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $Chaux\_4\_14

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwProduits

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| AL2O3 | float |  |  |  |  | X |
| CAO | float |  |  |  |  | X |
| CO2 | float |  |  |  |  | X |
| FE2O3 | float |  |  |  |  | X |
| MGO | float |  |  |  |  | X |
| S | float |  |  |  |  | X |
| SIO2 | float |  |  |  |  | X |
| T60 | float |  |  |  |  | X |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $Chaux\_40\_80

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwProduits

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| AL2O3 | float |  |  |  |  | X |
| CAO | float |  |  |  |  | X |
| CO2 | float |  |  |  |  | X |
| FE2O3 | float |  |  |  |  | X |
| MGO | float |  |  |  |  | X |
| R\_R | float |  |  |  |  | X |
| S | float |  |  |  |  | X |
| SIO2 | float |  |  |  |  | X |
| T60 | float |  |  |  |  | X |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $cmIDF\_Config

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined\_MES

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $colorfond

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $Crible

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$Moteur1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $Dolo\_20\_40

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwProduits

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| CO2 | float |  |  |  |  | X |
| S | float |  |  |  |  | X |
| T70 | float |  |  |  |  | X |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $Dolo\_3\_35

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwProduits

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| AL2O3 | float |  |  |  |  | X |
| CAO | float |  |  |  |  | X |
| CO2 | float |  |  |  |  | X |
| DT05 | float |  |  |  |  | X |
| FE2O3 | float |  |  |  |  | X |
| MGO | float |  |  |  |  | X |
| S | float |  |  |  |  | X |
| SIO2 | float |  |  |  |  | X |
| T65 | float |  |  |  |  | X |
| T70 | float |  |  |  |  | X |
| TTM | float |  |  |  |  | X |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $Dolo\_40\_80

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwProduits

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| CO2 | float |  |  |  |  | X |
| S | float |  |  |  |  | X |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $Dolo\_5\_13

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwProduits

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| CO2 | float |  |  |  |  | X |
| S | float |  |  |  |  | X |
| T70 | float |  |  |  |  | X |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $Dolo\_ULC

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwProduits

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| CO2 | float |  |  |  |  | X |
| S | float |  |  |  |  | X |
| T70 | float |  |  |  |  | X |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $Double

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$FieldReference

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwAlarmes

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| NomLogix | string |  |  |  |  |  |

##### Field Attributes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Attribute name | Data Type | IO Scaling | History | Limit Alarms | Rate Of Change Alarms | Target Deviation Alarms | Bad Value Alarm | Statistics |
| Alarme | boolean |  |  |  |  |  |  |  |

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $dwAlimentateur\_FMT

##### Description

Not Applicable

##### Derived from

$dwMoteurDirect1S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwAlveolaire\_BR8

##### Description

Not Applicable

##### Derived from

$dwMoteurDirect1S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwAlveolaire2S\_BR8

##### Description

Not Applicable

##### Derived from

$dwMoteurDirect2S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwBroyeur\_BR8

##### Description

Not Applicable

##### Derived from

$dwMoteurDirect1S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwClapet2Motorise2S3P

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwClapetMotorise1S3P

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmeDefMTH\_Mot2 | boolean |  | X |  | X |  |
| bAlarmeDefRKMAR\_Mot1 | boolean |  | X |  | X |  |
| bAlarmeDefRKMAR\_Mot2 | boolean |  | X |  | X |  |
| bAlarmeDefRKMAV\_Mot1 | boolean |  | X |  | X |  |
| bAlarmeDefRKMAV\_Mot2 | boolean |  | X |  | X |  |
| bAlarmeDefTemp\_Mot2 | boolean |  | X |  | X |  |
| bCommandeInhibitionDefMTH\_Mot2 | boolean |  |  |  |  |  |
| bCommandeInhibitionDefRKM\_Mot2 | boolean |  |  |  |  |  |
| bCommandeInhibitionDefTemp\_Mot2 | boolean |  |  |  |  |  |
| bStatusPosi\_cl\_mot1\_droite | boolean |  | X |  |  |  |
| bStatusPosi\_cl\_mot1\_gauche | boolean |  | X |  |  |  |
| bStatusPosi\_cl\_mot2\_droite | boolean |  | X |  |  |  |
| bStatusPosi\_cl\_mot2\_gauche | boolean |  | X |  |  |  |
| iCommandeTempsDefRKM\_Mot2 | boolean |  |  |  |  |  |
| sAlarmeDefMTH\_Mot2 | string |  |  |  |  |  |
| sAlarmeDefRKMAr\_Mot1 | string |  |  |  |  |  |
| sAlarmeDefRKMAr\_Mot2 | string |  |  |  |  |  |
| sAlarmeDefRKMAv\_Mot1 | string |  |  |  |  |  |
| sAlarmeDefRKMAv\_Mot2 | string |  |  |  |  |  |
| sAlarmeDefTemp\_Mot2 | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwClapetMotorise1S3P

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwActionneur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmeDefAU | boolean |  | X |  | X |  |
| bAlarmeDefIS | boolean |  | X |  | X |  |
| bAlarmeDefMTH | boolean |  | X |  | X |  |
| bAlarmeDefRKM | boolean |  | X |  | X |  |
| bAlarmeDefTemp | boolean |  | X |  | X |  |
| bAlarmeDisc2Fdc | boolean |  | X |  | X |  |
| bAlarmePosition1 | boolean |  | X |  | X |  |
| bAlarmePosition2 | boolean |  | X |  | X |  |
| bAlarmePosition3 | boolean |  | X |  | X |  |
| bCommandeFermetureManu | boolean | X |  |  |  |  |
| bCommandeInhibitionDefIS | boolean | X |  |  |  |  |
| bCommandeInhibitionDefMTH | boolean | X |  |  |  |  |
| bCommandeInhibitionDefRKM | boolean | X |  |  |  |  |
| bCommandeInhibitionDefTemp | boolean | X |  |  |  |  |
| bCommandeInhibitionDisc2FDC | boolean | X |  |  |  |  |
| bCommandeInhibitionDiscPosition1 | boolean | X |  |  |  |  |
| bCommandeInhibitionDiscPosition2 | boolean | X |  |  |  |  |
| bCommandeInhibitionDiscPosition3 | boolean | X |  |  |  |  |
| bCommandeOuvertureManu | boolean | X |  |  |  |  |
| bStatusAutorisationMarche | boolean |  | X |  |  |  |
| bStatusMarche | boolean |  | X |  |  |  |
| bStatusOrdreMarche | boolean |  | X |  |  |  |
| bStatusPosition1 | boolean |  | X |  |  |  |
| bStatusPosition2 | boolean |  | X |  |  |  |
| bStatusPosition3 | boolean |  | X |  |  |  |
| iCommandeTempsDefPosition1 | integer | X |  |  |  |  |
| iCommandeTempsDefPosition2 | integer | X |  |  |  |  |
| iCommandeTempsDefPosition3 | integer | X |  |  |  |  |
| iCommandeTempsDefRKM | integer | X |  |  |  |  |
| sAlarmeDefAU | string |  |  |  |  |  |
| sAlarmeDefIS | string |  |  |  |  |  |
| sAlarmeDefMTH | string |  |  |  |  |  |
| sAlarmeDefRKM | string |  |  |  |  |  |
| sAlarmeDefTemp | string |  |  |  |  |  |
| sAlarmeDisc2Fdc | string |  |  |  |  |  |
| sAlarmePosition1 | string |  |  |  |  |  |
| sAlarmePosition2 | string |  |  |  |  |  |
| sAlarmePosition3 | string |  |  |  |  |  |
| sTexteFerme | string |  |  |  |  |  |
| sTexteMilieu | string |  |  |  |  |  |
| sTexteOuvert | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

pdicNbForcage

|  |  |
| --- | --- |
| Name | pdicNbForcage |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

Dim nb as Integer;  
  
nb=0;  
  
IF Me.bCommandeInhibitionDefIS == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefMTH == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefRKM == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDiscPosition1 == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDiscPosition2 == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDiscPosition3 == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDisc2Fdc == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefTemp == false THEN  
 nb = nb + 1;  
ENDIF;  
  
  
Me.iNbForcage = nb;

#### Template ArchestrA $dwDecolmateur\_BR8

##### Description

Not Applicable

##### Derived from

$dwMoteurDirect1S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwDefautInhibable

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwDefaut

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bCmdInhibitionDefaut | boolean | X |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwElevateur\_BR8

##### Description

Not Applicable

##### Derived from

$dwMoteurDirect1S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwFLEX

##### Description

Not Applicable

##### Derived from

$dwVanneMotorisee2SG4.dwVariateur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmeCodeDefaut | boolean |  |  |  |  |  |
| bAlarmeCodeDefautTemporise | boolean |  |  |  | X |  |
| iCodeDefaut | integer |  | X |  |  |  |
| sAdresseIP | string |  |  |  |  |  |
| sAlarmeCodeDefautTemporise | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

alarmeCodeDefaut

|  |  |
| --- | --- |
| Name | alarmeCodeDefaut |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

dim generationDefaut as discrete;

**Script :**

if generationDefaut == true then  
 'logmessage("if generationDefaut == true then");  
 Me.sAlarmeCodeDefautTemporise = Me.sAlarmeDefDRIVE+" code défaut="+Me.iCodeDefaut;  
 Me.bAlarmeCodeDefautTemporise = true;  
 generationDefaut = false;'défaut déja généré  
else  
 'logmessage("if generationDefaut == true else");  
 Me.bAlarmeCodeDefautTemporise = false;  
endif;  
if Me.bAlarmeCodeDefaut == true then  
 'logmessage("if Me.bAlarmeCodeDefaut == true then");  
 generationDefaut = true;'mémorisation du défaut  
 Me.bAlarmeCodeDefaut = false;  
endif;

alarmeCodeDefautMemorise

|  |  |
| --- | --- |
| Name | alarmeCodeDefautMemorise |
| Description |  |
| Trigger | OnTrue of Me.bAlarmeDefDrive |

**Declarations :**

Not Applicable

**Script :**

Me.bAlarmeCodeDefaut=true;  
logmessage("Me.bAlarmeCodeDefaut=true");

#### Template ArchestrA $dwGen\_Bouton

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

Not Applicable

##### Field Attributes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Attribute name | Data Type | IO Scaling | History | Limit Alarms | Rate Of Change Alarms | Target Deviation Alarms | Bad Value Alarm | Statistics |
| Bp\_Gen | boolean |  | X |  |  |  |  |  |
| Anim | integer |  |  |  |  |  |  |  |

##### Scripts

Not Applicable

#### Template ArchestrA $dwGen\_Reg\_2S\_Positionne

##### Description

Not Applicable

##### Derived from

$dwGenClapet2S

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bStatusCdeDroite | boolean |  |  |  |  |  |
| bStatusCdeGauche | boolean |  |  |  |  |  |
| bStatusDefautDroite | boolean |  |  |  |  |  |
| bStatusDefautGauche | boolean |  |  |  |  |  |
| bStatusTransitionDroite | boolean |  |  |  |  |  |
| bStatusTransitionGauche | boolean |  |  |  |  |  |
| iPositionRegistre | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcStatucCdeDroite

|  |  |
| --- | --- |
| Name | dcPlcStatucCdeDroite |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusCdeDroite |

**Declarations :**

Not Applicable

**Script :**

Me.bStatusCdeDroite = 0;  
Me.bStatusDefautDroite = 0;  
Me.bStatusTransitionDroite = 0;  
  
IF Me.iTypeAnimation == 40 THEN ' BRO  
  
 IF (Me.iPlcStatusCdeDroite == 1) THEN  
 ' Arrêt sans défaut  
 Me.bStatusCdeDroite = 0;  
 Me.bStatusDefautDroite = 0;  
 Me.bStatusTransitionDroite = 0;  
  
 ELSEIF (Me.iPlcStatusCdeDroite == 2) THEN  
 ' Arrêt avec défaut  
 Me.bStatusCdeDroite = 1;  
  
 ELSEIF (Me.iPlcStatusCdeDroite == 8) THEN  
 ' Marche sans défaut  
 Me.bStatusDefautDroite = 1;  
  
   
 ELSEIF (Me.iPlcStatusCdeDroite == 16) THEN  
 ' Marche avec défaut  
 Me.bStatusCdeDroite = 1;  
 Me.bStatusDefautDroite = 1;  
   
 ELSEIF (Me.iPlcStatusCdeDroite == 4) THEN  
 ' Démarrage  
 Me.bStatusTransitionDroite = 0;  
   
   
 ENDIF;  
endif;

dcPlcStatucCdeGauche

|  |  |
| --- | --- |
| Name | dcPlcStatucCdeGauche |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusCdeGauche |

**Declarations :**

Not Applicable

**Script :**

Me.bStatusCdeGauche = 0;  
Me.bStatusDefautGauche = 0;  
Me.bStatusTransitionGauche = 0;  
  
IF Me.iTypeAnimation == 40 THEN ' BRO  
  
 IF (Me.iPlcStatusCdeGauche == 1) THEN  
 ' Arrêt sans défaut  
 Me.bStatusCdeGauche = 0;  
 Me.bStatusDefautGauche = 0;  
 Me.bStatusTransitionGauche = 0;  
  
 ELSEIF (Me.iPlcStatusCdeGauche == 2) THEN  
 ' Arrêt avec défaut  
 Me.bStatusCdeGauche = 1;  
  
 ELSEIF (Me.iPlcStatusCdeGauche == 8) THEN  
 ' Marche sans défaut  
 Me.bStatusDefautGauche= 1;  
  
   
 ELSEIF (Me.iPlcStatusCdeGauche == 16) THEN  
 ' Marche avec défaut  
 Me.bStatusCdeGauche = 1;  
 Me.bStatusDefautGauche = 1;  
   
 ELSEIF (Me.iPlcStatusCdeGauche == 4) THEN  
 ' Démarrage  
 Me.bStatusTransitionGauche = 0;  
   
   
 ENDIF;  
ENDIF;

dcPlcPositionRegistre

|  |  |
| --- | --- |
| Name | dcPlcPositionRegistre |
| Description |  |
| Trigger | DataChange of Me.iPlcPosition\_Registre |

**Declarations :**

Not Applicable

**Script :**

Me.iPositionregistre = Me.iPlcPosition\_Registre;

#### Template ArchestrA $dwGenACT

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogInOut

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenActionneur

##### Description

Not Applicable

##### Derived from

$dwGenObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bStatusEnDefaut | boolean |  |  |  |  |  |
| bZoneImbriquee | boolean |  |  |  |  |  |
| iTypeAnimationAlternative | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcStatusAuto\_01

|  |  |
| --- | --- |
| Name | dcStatusAuto\_01 |
| Description |  |
| Trigger | DataChange of MyArea.iStatusAuto\_01 |

**Declarations :**

Not Applicable

**Script :**

IF Me.bZoneImbriquee == false THEN  
 Me.iStatusAuto.Input.InputSource = "MyArea.iStatusAuto\_01";  
ENDIF;

dcStatusAuto\_02

|  |  |
| --- | --- |
| Name | dcStatusAuto\_02 |
| Description |  |
| Trigger | DataChange of MyArea.iStatusAuto\_02 |

**Declarations :**

Not Applicable

**Script :**

IF Me.bZoneImbriquee == false THEN  
 Me.iStatusAuto.Input.InputSource = "MyArea.iStatusAuto\_02";  
ENDIF;

dcStatusAuto\_03

|  |  |
| --- | --- |
| Name | dcStatusAuto\_03 |
| Description |  |
| Trigger | DataChange of MyArea.iStatusAuto\_03 |

**Declarations :**

Not Applicable

**Script :**

IF Me.bZoneImbriquee == false THEN  
 Me.iStatusAuto.Input.InputSource = "MyArea.iStatusAuto\_03";  
ENDIF;

dcStatusAuto\_04

|  |  |
| --- | --- |
| Name | dcStatusAuto\_04 |
| Description |  |
| Trigger | DataChange of MyArea.iStatusAuto\_04 |

**Declarations :**

Not Applicable

**Script :**

IF Me.bZoneImbriquee == false THEN  
 Me.iStatusAuto.Input.InputSource = "MyArea.iStatusAuto\_04";  
ENDIF;

dcStatusAuto\_05

|  |  |
| --- | --- |
| Name | dcStatusAuto\_05 |
| Description |  |
| Trigger | DataChange of MyArea.iStatusAuto\_05 |

**Declarations :**

Not Applicable

**Script :**

IF Me.bZoneImbriquee == false THEN  
 Me.iStatusAuto.Input.InputSource = "MyArea.iStatusAuto\_05";  
ENDIF;

dcStatusAuto\_06

|  |  |
| --- | --- |
| Name | dcStatusAuto\_06 |
| Description |  |
| Trigger | DataChange of MyArea.iStatusAuto\_06 |

**Declarations :**

Not Applicable

**Script :**

IF Me.bZoneImbriquee == false THEN  
 Me.iStatusAuto.Input.InputSource = "MyArea.iStatusAuto\_06";  
ENDIF;

dcStatusAuto\_07

|  |  |
| --- | --- |
| Name | dcStatusAuto\_07 |
| Description |  |
| Trigger | DataChange of MyArea.iStatusAuto\_07 |

**Declarations :**

Not Applicable

**Script :**

IF Me.bZoneImbriquee == false THEN  
 Me.iStatusAuto.Input.InputSource = "MyArea.iStatusAuto\_07";  
ENDIF;

dcStatusAuto\_08

|  |  |
| --- | --- |
| Name | dcStatusAuto\_08 |
| Description |  |
| Trigger | DataChange of MyArea.iStatusAuto\_08 |

**Declarations :**

Not Applicable

**Script :**

IF Me.bZoneImbriquee == false THEN  
 Me.iStatusAuto.Input.InputSource = "MyArea.iStatusAuto\_08";  
ENDIF;

#### Template ArchestrA $dwGenAgitateur

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenAlarme

##### Description

Not Applicable

##### Derived from

$Boolean

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenAlimentateur

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenAlveolaire

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenAlveolaire2S

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect2S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenAMP

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenAnimationsBool

##### Description

Not Applicable

##### Derived from

$dwGenObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bPlcStatus\_FB | boolean |  |  |  |  |  |
| bStatus | boolean |  |  |  |  |  |
| iTypeAnimationAlternative | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcStatus

|  |  |
| --- | --- |
| Name | dcPlcStatus |
| Description |  |
| Trigger | DataChange of Me.bPlcStatus + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 16 Avril 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatus");  
 logmessage("bPlcStatus : " + Me.bPlcStatus);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
IF Me.iTypeAnimation == 10 OR Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 40 OR Me.iTypeAnimation == 70   
OR Me.iTypeANimation == 80 THEN ' PFEIFFER - BR8 OR BR6 OR BRO or CAC/FMT or TSFx  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 0; ' vert  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 1; ' rouge  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 0 THEN ' BR7  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 0; ' vert  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 1; ' rouge  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 1 THEN ' BR7\_BROB\_DM60\_1  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 1; ' rouge  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 0; ' vert  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 30 THEN ' HGS  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 0; ' vert  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 1; ' rouge  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 0)  
OR (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 0)  
THEN ' COM  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 0; ' vert  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 1; ' rouge  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 1   
OR (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 1)  
THEN ' COM\_DM  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 1; ' rouge  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 0; ' vert  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 2 THEN ' COM\_SIR  
 IF (Me.bPlcStatus <= 1) THEN  
 Me.bStatus = 0; ' vert  
 ELSEIF (Me.bPlcStatus >= 2) THEN  
 Me.bStatus = 1; ' rouge  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 100 THEN ' EAU  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 0; ' vert  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 1; ' rouge  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenAnimationsBool  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenAnimationsBool  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.bPlcStatus.Input.InputSource = "Me.bPlcStatus\_FB";

#### Template ArchestrA $dwGenAnimationsInt

##### Description

Not Applicable

##### Derived from

$dwGenObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| iPlcStatus\_FB | integer |  |  |  |  |  |
| iStatus | integer |  |  |  |  |  |
| iTypeAnimationAlternative | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenAnimationsInt  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenAnimationsInt  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatus.Input.InputSource = "Me.iPlcStatus\_FB";

dcPlcStatus

|  |  |
| --- | --- |
| Name | dcPlcStatus |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 06 novembre 2008  
' Ce script traitera les status  
' status = 0 - vert  
' status = 1 - rouge  
' status = 2 - jaune  
' status = 3 - vert clignotant  
' status = 4 - rouge clignotant  
' status = 5 - jaune clignotant  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatus");  
 logmessage("iPlcStatus : " + Me.iPlcStatus);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
IF Me.iTypeAnimation == 70 and Me.iTypeAnimationAlternative == 0 THEN 'CAC  
 IF Me.iPlcStatus <= 1 THEN  
 ' VERT  
 Me.iStatus = 0;  
 ELSEIF Me.iPlcStatus == 2 OR Me.iPlcStatus == 3 THEN  
 ' ROUGE  
 Me.iStatus = 1;  
 ELSEIF Me.iPlcStatus >= 4 AND Me.iPlcStatus < 8 THEN  
 ' JAUNE  
 Me.iStatus = 2;  
 ELSEIF Me.iPlcStatus >= 8 AND Me.iPlcStatus < 16 THEN  
 ' VERT CLIGNOTANT  
 Me.iStatus = 3;  
 ELSEIF Me.iPlcStatus >= 16 AND Me.iPlcStatus < 32 THEN  
 ' ROUGE CLIGNOTANT  
 Me.iStatus = 4;  
 ELSEIF Me.iPlcStatus >= 32 THEN  
 ' JAUNE CLIGNOTANT  
 Me.iStatus = 5;  
 ENDIF;  
ENDIF;  
IF Me.iTypeAnimation == 70 and Me.iTypeAnimationAlternative == 1 THEN 'CAC  
 IF Me.iPlcStatus == 0 THEN  
 ' VERT  
 Me.iStatus = 0;  
 ELSEIF Me.iPlcStatus == 1 THEN  
 ' ROUGE  
 Me.iStatus = 1;  
 ENDIF;  
ENDIF;

#### Template ArchestrA $dwGenAutres

##### Description

Not Applicable

##### Derived from

$dwGenActionneur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bStatusDemarrage | boolean |  |  |  |  |  |
| bStatusMarche | boolean |  |  |  |  |  |
| bStatusVidange | boolean |  |  |  |  |  |
| iPlcStatus\_FB | integer |  |  |  |  |  |
| iTypeCommandeArret | integer |  |  |  |  |  |
| iTypeCommandeMarche | integer |  |  |  |  |  |
| iValCommandeArret | integer |  |  |  |  |  |
| iValCommandeMarche | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcStatus

|  |  |
| --- | --- |
| Name | dcPlcStatus |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 16 Avril 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatus");  
 logmessage("iPlcStatus : " + Me.iPlcStatus);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
' Réinitialisation des variables  
Me.bStatusMarche = 0;  
Me.bStatusEnDefaut = 0;  
Me.bStatusVidange = 0;  
Me.bStatusDemarrage = 0;  
  
IF Me.iTypeAnimation == 10 AND Me.iTypeAnimationAlternative == 0 THEN ' PFEIFFER - BR8  
 IF (Me.iPlcStatus == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ELSEIF (Me.iPlcStatus == 3) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus == 4) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 ELSEIF (Me.iPlcStatus == 5) THEN  
 ' Vidange - jaune clignotant  
 Me.bStatusVidange = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 10 AND Me.iTypeAnimationAlternative == 1 THEN ' BR8 - CHAUF1  
 IF (Me.iPlcStatus == 0) THEN  
 Me.bStatusMarche = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.bStatusMarche = 1;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatus >= 2) THEN  
 Me.bStatusMarche = 0;  
 Me.bStatusDemarrage = 1;  
 ENDIF;  
  
ELSEIF Me.iTypeAnimation == 20 THEN ' BR7  
 IF (Me.iPlcStatus == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 ELSEIF (Me.iPlcStatus == 2) OR (Me.iPlcStatus == 3) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ELSEIF (Me.iPlcStatus == 4) OR (Me.iPlcStatus == 5) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus >= 6) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 30 OR Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 80 THEN ' HGS, BR6, TSFM  
 IF (Me.iPlcStatus == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ENDIF;  
  
ELSEIF Me.iTypeAnimation == 90 THEN ' FR  
 IF (Me.iPlcStatus == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 ELSEIF (Me.iPlcStatus == 3) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus == 4) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus >= 5) THEN  
 ' Vidange - jaune clignotant  
 Me.bStatusVidange = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN

dcRAZCmd

|  |  |
| --- | --- |
| Name | dcRAZCmd |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Juillet 2008  
' Ce script remettra le bit de commande à zéro pour certaines applications  
' iTypeAnimation : 50 = BR6  
  
IF (Me.bPlcCommandeArret == 1 OR Me.bPlcCommandeMarche == 1 OR Me.iPlcCommandeMarche <> 0 OR Me.iPlcCommandeArret <> 0)  
AND (Me.iTypeAnimation == 50 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeArret == 3 AND Me.iTypeCommandeMarche == 3)) THEN  
 Me.bPlcCommandeArret = 0;  
 Me.bPlcCommandeMarche = 0;  
 Me.iPlcCommandeArret = 0;  
 Me.iPlcCommandeMarche = 0;  
ENDIF;  
  
Me.iTempo = 0;

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenAutres  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenAutres  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatus.Input.InputSource = "Me.iPlcStatus\_FB";

wtCommandeArret

|  |  |
| --- | --- |
| Name | wtCommandeArret |
| Description |  |
| Trigger | WhileTrue of Me.bPlcCommandeArret == 1 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeArret == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

wtCommandeMarche

|  |  |
| --- | --- |
| Name | wtCommandeMarche |
| Description |  |
| Trigger | WhileTrue of Me.bPlcCommandeMarche == 1 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeMarche == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

RAZ\_Tempo

|  |  |
| --- | --- |
| Name | RAZ\_Tempo |
| Description |  |
| Trigger | WhileTrue of Me.iTempo > 15 |

**Declarations :**

Not Applicable

**Script :**

IF (Me.bPlcCommandeArret == 1 OR Me.bPlcCommandeMarche == 1 OR Me.iPlcCommandeMarche <> 0 OR Me.iPlcCommandeArret <> 0)  
AND (Me.iTypeAnimation == 50 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeArret == 3 AND Me.iTypeCommandeMarche == 3)) THEN  
 Me.bPlcCommandeArret = 0;  
 Me.bPlcCommandeMarche = 0;  
 Me.iPlcCommandeArret = 0;  
 Me.iPlcCommandeMarche = 0;  
ENDIF;  
  
Me.iTempo = 0;

wtCommandeArretInt

|  |  |
| --- | --- |
| Name | wtCommandeArretInt |
| Description |  |
| Trigger | WhileTrue of Me.iPlcCommandeArret <> 0 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeArret == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

wtCommandeMarcheInt

|  |  |
| --- | --- |
| Name | wtCommandeMarcheInt |
| Description |  |
| Trigger | WhileTrue of Me.iPlcCommandeMarche <> 0 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeMarche == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

#### Template ArchestrA $dwGenBaladeur2S

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect2S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenBalance2S

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect2S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenBattage

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenBoutonCom

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| DefautGenAU | boolean |  |  |  |  |  |
| DefautGenAuto | boolean |  |  |  |  |  |
| DefautGenLocal | boolean |  |  |  |  |  |
| DefautGenManu | boolean |  |  |  |  |  |
| DefautGenStop | boolean |  |  |  |  |  |
| NomLogix | string |  |  |  |  |  |

##### Field Attributes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Attribute name | Data Type | IO Scaling | History | Limit Alarms | Rate Of Change Alarms | Target Deviation Alarms | Bad Value Alarm | Statistics |
| BP\_STOP | boolean |  | X |  |  |  |  |  |
| BP\_AUTO | boolean |  | X |  |  |  |  |  |
| BP\_MANU | boolean |  | X |  |  |  |  |  |
| BP\_DEM | boolean |  | X |  |  |  |  |  |
| BP\_AU | boolean |  | X |  |  |  |  |  |
| BP\_Gen | boolean |  | X |  |  |  |  |  |
| BP\_Auto\_Anim | integer |  |  |  |  |  |  |  |
| BP\_Manu\_Anim | integer |  |  |  |  |  |  |  |
| BP\_Stop\_Anim | integer |  |  |  |  |  |  |  |
| BP\_Local\_Anim | integer |  |  |  |  |  |  |  |
| BP\_Dem\_Anim | integer |  |  |  |  |  |  |  |
| Time\_Dem | integer |  |  |  |  |  |  |  |
| BP\_AU\_Anim | integer |  |  |  |  |  |  |  |
| BP\_Gen\_ANIM | integer |  |  |  |  |  |  |  |

##### Scripts

CheckDefautGen

|  |  |
| --- | --- |
| Name | CheckDefautGen |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

if Me.BP\_Stop\_Anim == 16 or Me.BP\_Stop\_Anim == 32 or Me.BP\_Stop\_Anim == 8 then  
Me.DefautGenStop = 1;  
else  
Me.DefautGenStop = 0;  
endif;  
  
if Me.BP\_Auto\_Anim == 16 or Me.BP\_Auto\_Anim == 32 or Me.BP\_Auto\_Anim == 8 then  
Me.DefautGenAuto = 1;  
else  
Me.DefautGenAuto = 0;  
endif;  
  
  
if Me.BP\_Manu\_Anim == 16 or Me.BP\_Manu\_Anim == 32 or Me.BP\_Manu\_Anim == 8 then  
Me.DefautGenManu = 1;  
else  
Me.DefautGenManu = 0;  
endif;  
  
  
if Me.BP\_AU\_Anim == 16 or Me.BP\_AU\_Anim == 32 or Me.BP\_AU\_Anim == 8 then  
Me.DefautGenAU = 1;  
else  
Me.DefautGenAU = 0;  
endif;

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $dwGenBoutonsBool

##### Description

Not Applicable

##### Derived from

$dwGenObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bPlcStatus\_FB | boolean |  |  |  |  |  |
| bStatus | boolean |  |  |  |  |  |
| guy | boolean |  |  |  |  |  |
| iTypeAnimationAlternative | integer |  |  |  |  |  |
| iTypeCommande | integer |  |  |  |  |  |
| iValCommande | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |
| teller | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcStatus

|  |  |
| --- | --- |
| Name | dcPlcStatus |
| Description |  |
| Trigger | DataChange of Me.bPlcStatus + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 16 Avril 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatus");  
 logmessage("bPlcStatus : " + Me.bPlcStatus);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
'logmessage("Script : dcPlcStatus Test");  
'logmessage("Instance : " + Me.TagName);  
'logmessage("Input source of bPLCStatus : " + Me.bPlcStatus.Input.InputSource) ;  
  
'if Me.bPLCStatus == 1 then  
' logmessage("bPlcStatus : " + "True");  
'else  
' logmessage("bPlcStatus : " + "False");  
'endif ;  
  
'logmessage("iTypeAnimation : " + StringFromIntg( Me.iTypeAnimation, 10 ));  
  
IF Me.iTypeAnimation == 10 OR Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 60 THEN ' PFEIFFER - BR8 OR BR6 OR CAC/FMT  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 0; ' vert  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 1; ' rouge  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 0 THEN ' BR7  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 0; ' vert  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 1; ' rouge  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 1 THEN ' BR7\_CHOIX\_TEMP  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 1; ' rouge  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 0; ' vert  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 0 THEN ' HGS  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 0; ' vert  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 1; ' rouge  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 1 THEN ' HGS\_VIS7CM  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 1; ' rouge  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 0; ' vert  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 40 AND Me.iTypeAnimationAlternative == 0 THEN ' BRO  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 0; ' vert  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 1; ' rouge  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 0) ' CAC/FMT  
OR (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 0)   
OR (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 0)   
THEN ' TSFx  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 0; ' vert  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 1; ' rouge  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 1)  
OR (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 1)   
OR (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 1)   
THEN   
' FMTx\_GAZ\_MANU, FMTx\_REG\_GAZ\_MANU, FMTx\_AIR\_PRIM\_MANU, FMTx\_AIR\_SEC\_MANU, FMTx\_ALV\_CENT\_MANU, FMTx\_ALV\_EXT1\_MANU, FMTx\_ALV\_EXT2\_MANU  
' TSFS\_FR4\_CHAUX, TSFS\_S11\_CHAUX  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 1; ' rouge  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 0; ' vert  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 40 AND Me.iTypeAnimationAlternative == 1 THEN ' BRO\_DECHARG\_AUTORISE\_FILLER  
 IF (Me.bPlcStatus == 0) THEN  
 Me.bStatus = 1; ' rouge  
 ELSEIF (Me.bPlcStatus == 1) THEN  
 Me.bStatus = 0; ' vert  
 ENDIF; ' IF (Me.bPlcStatus == )  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN   
  
'logmessage("Script dPLCStatus executed");

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenBoutonsBool  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
'logmessage("Instance : " + Me.Tagname + " : Script ObjectDeploy is being executed");  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 'Logmessage("Instance : " + Me.Tagname + " : AffectInputSource is activated in script ObjectDeploy") ;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenBoutonsBool  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.bPlcStatus.Input.InputSource = "Me.bPlcStatus\_FB";  
  
'logmessage("instance : " + Me.Tagname + " : Script AffectInputSource executed") ;

dcRAZCmd

|  |  |
| --- | --- |
| Name | dcRAZCmd |
| Description |  |
| Trigger | DataChange of Me.bPlcStatus |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Juillet 2008  
' Ce script remettra le bit de commande à zéro pour certaines applications  
' iTypeAnimation : 50 = BR6  
  
IF (Me.bPlcCommande == 1 OR Me.iPlcCommande <> 0) AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70) THEN  
 Me.bPlcCommande = 0;  
 Me.iPlcCommande = 0;  
ENDIF;  
  
Me.iTempo = 0;  
  
'logmessage("Instance : " + Me.Tagname + " : Script dcRAZCmd executed");

wtCommande

|  |  |
| --- | --- |
| Name | wtCommande |
| Description |  |
| Trigger | WhileTrue of Me.bPlcCommande |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommande == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

RAZ\_Tempo

|  |  |
| --- | --- |
| Name | RAZ\_Tempo |
| Description |  |
| Trigger | WhileTrue of Me.iTempo > 15 |

**Declarations :**

Not Applicable

**Script :**

IF (Me.bPlcCommande == 1 OR Me.iPlcCommande <> 0) AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70) THEN  
 Me.bPlcCommande = 0;  
 Me.iPlcCommande = 0;  
ENDIF;  
  
Me.iTempo = 0;  
  
'logmessage("Instance : " + Me.Tagname + " : Script RAZ\_Tempo executed") ;

wtCommandeInt

|  |  |
| --- | --- |
| Name | wtCommandeInt |
| Description |  |
| Trigger | WhileTrue of Me.iPlcCommande <> 0 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommande == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

#### Template ArchestrA $dwGenBoutonsInt

##### Description

Not Applicable

##### Derived from

$dwGenObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bRAZException | boolean |  |  |  |  |  |
| iPlcStatus\_FB | integer |  |  |  |  |  |
| iStatus | integer |  |  |  |  |  |
| iTypeAnimationAlternative | integer |  |  |  |  |  |
| iTypeCommande | integer |  |  |  |  |  |
| iValCommande | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcStatus

|  |  |
| --- | --- |
| Name | dcPlcStatus |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Avril 2008  
' Ce script traitera les status  
' status = 0 - vert  
' status = 1 - rouge  
' status = 2 - jaune  
' status = 3 - vert clignotant  
' status = 4 - rouge clignotant  
' status = 5 - jaune clignotant  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatus");  
 logmessage("iPlcStatus : " + Me.iPlcStatus);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
IF Me.iTypeAnimation == 10 THEN ' PFEIFFER - BR8  
  
 IF (Me.iPlcStatus == 0) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus == 3) THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ELSEIF (Me.iPlcStatus == 4) THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF (Me.iPlcStatus == 5) THEN  
 Me.iStatus = 5; ' jaune clignotant  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 0 THEN ' BR7\_GROUPE\_ALIM, BR7\_GROUPE\_LB7, BR7\_GROUPE\_MC7, BR7\_GROUPE\_PM7  
 IF (Me.iPlcStatus == 0) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus == 2) OR (Me.iPlcStatus == 3) THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF (Me.iPlcStatus == 4) THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF (Me.iPlcStatus == 5) THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ELSEIF (Me.iPlcStatus == 6) THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF (Me.iPlcStatus >= 7) THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 1 THEN ' BR7\_REGUL\_AUTO\_PAP71, BR7\_REGUL\_AUTO\_PAP72, BR7\_REGUL\_AUTO\_PAP91  
 IF (Me.iPlcStatus == 0) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF (Me.iPlcStatus >= 2) THEN  
 Me.iStatus = 1; ' rouge  
 ENDIF;  
  
ELSEIF Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 2 THEN ' BR7\_GR\_LB7\_STOP, BR7\_GR\_MC7\_STOP, BR7\_SEP7\_JAQUET\_N2\_SEP7  
 IF (Me.iPlcStatus == 0) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 5; ' jaune clignotant  
 ELSEIF (Me.iPlcStatus >= 2) THEN  
 Me.iStatus = 1; ' rouge  
 ENDIF;  
  
ELSEIF Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 3 THEN ' COMP1\_CALC, COMP3\_CALC  
 IF (Me.iPlcStatus == 0) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF (Me.iPlcStatus == 2) OR (Me.iPlcStatus == 3) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus == 4) OR (Me.iPlcStatus == 5) THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF (Me.iPlcStatus >= 6) THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 4 THEN ' PAVE\_PERM  
 IF (Me.iPlcStatus == 0) OR (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 2) OR (Me.iPlcStatus == 3) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus >= 4) THEN  
 Me.iStatus = 2; ' jaune  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 5 THEN ' Boutons AUTO GR  
 IF (Me.iPlcStatus == 0) OR (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 2) OR (Me.iPlcStatus == 3) THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF (Me.iPlcStatus >= 4) AND (Me.iPlcStatus < 8) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus >= 8) AND (Me.iPlcStatus < 12) THEN  
 Me.iStatus = 3; ' vert clign  
 ELSEIF (Me.iPlcStatus >= 12) THEN  
 Me.iStatus = 4; ' rouge clig  
 ENDIF; ' IF (Me.iPlcStatusAuto\_01 == )  
  
ELSEIF Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 0 THEN ' HGS  
 IF (Me.iPlcStatus == 0) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus == 3) THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ELSEIF (Me.iPlcStatus == 4) THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF (Me.iPlcStatus == 5) THEN  
 Me.iStatus = 5; ' jaune clignotant  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 1 THEN ' HGS\_ANIMPERM  
 IF (Me.iPlcStatus == 0) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus == 4) THEN  
 Me.iStatus = 2; ' jaune  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 4 THEN ' BR7\_GR\_LB7\_AUTO, BR7\_GR\_MC7\_AUTO  
 IF (Me.iPlcStatus == 0) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF (Me.iPlcStatus == 4) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus == 8) THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF (Me.iPlcStatus == 16) THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 50 AND Me.iTypeAnimationAlternative == 0) OR (Me.iTypeAnimation == 40)  
OR (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 0)  
OR (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 1)  
THEN ' BR6 OR BRO OR CAC/FMT  
  
 IF (Me.iPlcStatus <= 1) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 2) OR (Me.iPlcStatus == 3) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus >= 4) AND (Me.iPlcStatus < 8) THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF (Me.iPlcStatus >= 8) AND (Me.iPlcStatus < 16) THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF (Me.iPlcStatus >= 16) AND (Me.iPlcStatus < 32) THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ELSEIF (Me.iPlcStatus >= 32) THEN  
 Me.iStatus = 5; ' jaune clignotant  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 50 AND Me.iTypeAnimationAlternative == 1 THEN ' BR6\_DP\_BROYEUR - BR6\_VERROUILLAGE\_M13  
  
 IF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.iStatus = 1; ' rouge  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 0 THEN ' COM\_VALIDATION\_PESEE  
 IF (Me.iPlcStatus == 1) OR (Me.iPlcStatus == 2) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 3) THEN  
 Me.iStatus = 1; ' rouge  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 1 THEN ' COM\_VERIF\_SAISIE\_PESEE  
 IF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus == 3) THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 2 THEN ' COM\_AUTO\_PRCPT, COM\_AUTO\_V7V8, COM\_DEMARRAGE\_BYPASS\_COMPACTAGE, COM\_STOP\_ALTER\_V7V8  
 IF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.iStatus = 0; ' vert  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 3 THEN ' COM\_DEMARRAGE\_VARIATEUR\_CPT, COM\_MANU\_PRCPT, COM\_MANU\_V7V8, COM\_MARCHE\_ALTER\_V7V8  
 IF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.iStatus = 1; ' rouge  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 4 )' COM\_GRAISSAGE\_COMPACTAGE, COM\_PULVERISATION\_COMPACTAGE  
 OR Me.iTypeAnimation == 40 'BRO   
 THEN   
 IF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus == 4) THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF (Me.iPlcStatus == 8) THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF (Me.iPlcStatus == 16) THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ELSEIF (Me.iPlcStatus == 32) THEN  
 Me.iStatus = 5; ' jaune clignotant  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 1) THEN ' FMTx\_VEN5\_MANU  
 IF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus >= 2) THEN  
 Me.iStatus = 0; ' vert  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 2) THEN ' FMTx\_FORC\_FERM  
 IF (Me.iPlcStatus == 0) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 1) OR (Me.iPlcStatus == 2) OR (Me.iPlcStatus == 3) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus >= 4) AND (Me.iPlcStatus < 8) THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF (Me.iPlcStatus >= 8) AND (Me.iPlcStatus < 16) THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF (Me.iPlcStatus >= 16) AND (Me.iPlcStatus < 32) THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ELSEIF (Me.iPlcStatus >= 32) THEN  
 Me.iStatus = 5; ' jaune clignotant  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 0) THEN ' TSFx  
 IF Me.iPlcStatus == 0 THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF Me.iPlcStatus == 1 THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF Me.iPlcStatus == 2 OR Me.iPlcStatus == 3 THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF Me.iPlcStatus >= 4 AND Me.iPlcStatus < 8 THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF Me.iPlcStatus >= 8 AND Me.iPlcStatus < 16 THEN  
 Me.iStatus = 5; ' jaune clignotant  
 ELSEIF Me.iPlcStatus >= 16 THEN  
 Me.iStatus = 4; ' rouge clignotant   
 ENDIF;  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 1) THEN ' TSFQ\_FLTx, TSFM\_AU\_xxx, TSFM\_DEPxx  
 IF Me.iPlcStatus == 0 THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF Me.iPlcStatus == 1 THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF Me.iPlcStatus == 2 OR Me.iPlcStatus == 3 THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF Me.iPlcStatus == 4 OR Me.iPlcStatus == 5 THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF Me.iPlcStatus >= 6 THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ENDIF;  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 2) THEN ' TSFQ\_MAT\_CHAUX  
 IF Me.iPlcStatus == 0 THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF Me.iPlcStatus == 1 THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF Me.iPlcStatus == 2 OR Me.iPlcStatus == 3 THEN  
 Me.iStatus = 0; ' vert  
 ENDIF;  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 3) THEN ' TSFQ\_MAT\_DOLO  
 IF Me.iPlcStatus == 0 OR Me.iPlcStatus == 1 THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF Me.iPlcStatus == 2 THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF Me.iPlcStatus == 3 THEN  
 Me.iStatus = 0; ' vert  
 ENDIF;  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 4) THEN ' TSFQ\_MAT\_xx\_MAGNESIA  
 IF Me.iPlcStatus <= 2 THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF Me.iPlcStatus == 3 THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF Me.iPlcStatus > 3 THEN  
 Me.iStatus = 0; ' vert  
 ENDIF;  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 5) THEN ' TSFM\_FOUR\_VERSE  
 IF Me.iPlcStatus == 9 THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF Me.iPlcStatus == 10 THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF Me.iPlcStatus == 11 THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF Me.iPlcStatus >= 12 THEN  
 Me.iStatus = 0; ' vert  
 ENDIF;  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 6) THEN ' SELECTION\_RECETTES  
 IF Me.iPlcStatus == 0 THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF Me.iPlcStatus == 1 THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF Me.iPlcStatus == 2 THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF Me.iPlcStatus == 3 THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF Me.iPlcStatus == 4 THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF Me.iPlcStatus == 5 THEN  
 Me.iStatus = 5; ' jaune clignotant  
 ELSEIF Me.iPlcStatus >= 6 THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ENDIF;  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 7) THEN ' TSFM\_ECHANT  
 IF Me.iPlcStatus == 0 THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF Me.iPlcStatus == 1 THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF Me.iPlcStatus == 2 OR Me.iPlcStatus == 3 THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ELSEIF Me.iPlcStatus >= 4 AND Me.iPlcStatus < 8 THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF Me.iPlcStatus >= 8 AND Me.iPlcStatus < 16 THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF Me.iPlcStatus >= 16 AND Me.iPlcStatus < 32 THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF Me.iPlcStatus >= 32 THEN  
 Me.iStatus = 5; ' jaune clignotant  
 ENDIF;  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 8) THEN 'TSFM\_FMx\_ARRET  
 IF Me.iPlcStatus == 0 THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF Me.iPlcStatus == 1 THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF Me.iPlcStatus >= 2 THEN  
 Me.iStatus = 2; ' jaune   
 ENDIF;  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 9) THEN 'TSFM\_FMx\_MARCHE  
 IF Me.iPlcStatus == 0 THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF Me.iPlcStatus == 1 THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF Me.iPlcStatus >= 2 THEN  
 Me.iStatus = 2; ' jaune   
 ENDIF;  
  
ELSEIF Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 0 THEN ' FR  
 IF (Me.iPlcStatus == 0) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF (Me.iPlcStatus == 3) THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF (Me.iPlcStatus == 4) THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ELSEIF (Me.iPlcStatus >= 5) THEN  
 Me.iStatus = 5; ' jaune clignotant  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 2 THEN ' FR  
 IF (Me.iPlcStatus == 0) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF (Me.iPlcStatus == 3) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus >= 4) THEN  
 Me.iStatus = 2; ' jaune  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenBoutonsInt  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenBoutonsInt  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatus.Input.InputSource = "Me.iPlcStatus\_FB";

dcRAZCmd

|  |  |
| --- | --- |
| Name | dcRAZCmd |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Juillet 2008  
' Ce script remettra le bit de commande à zéro pour certaines applications  
' iTypeAnimation : 50 = BR6  
  
IF (Me.bPlcCommande == 1 OR Me.iPlcCommande <> 0)   
AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70  
OR (Me.iTypeAnimation == 90 AND Me.iTypeCommande == 3))   
AND Me.bRAZException == 0 THEN  
 Me.bPlcCommande = 0;  
 Me.iPlcCommande = 0;  
ENDIF;  
  
Me.iTempo = 0;

wtCommande

|  |  |
| --- | --- |
| Name | wtCommande |
| Description |  |
| Trigger | WhileTrue of Me.bPlcCommande |

**Declarations :**

Not Applicable

**Script :**

IF (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommande == 3)) AND Me.bRAZException == 0 THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

RAZ\_Tempo

|  |  |
| --- | --- |
| Name | RAZ\_Tempo |
| Description |  |
| Trigger | WhileTrue of Me.iTempo > 15 |

**Declarations :**

Not Applicable

**Script :**

IF (Me.bPlcCommande == 1 OR Me.iPlcCommande <> 0)   
AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70  
OR (Me.iTypeAnimation == 90 AND Me.iTypeCommande == 3))  
AND Me.bRAZException == 0 THEN  
 Me.bPlcCommande = 0;  
 Me.iPlcCommande = 0;  
ENDIF;  
  
Me.iTempo = 0;

wtCommandeInt

|  |  |
| --- | --- |
| Name | wtCommandeInt |
| Description |  |
| Trigger | WhileTrue of Me.iPlcCommande <> 0 |

**Declarations :**

Not Applicable

**Script :**

IF (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommande == 3)) AND Me.bRAZException == 0 THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

#### Template ArchestrA $dwGenBrosse

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenBroyeur

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenChaudiere

##### Description

Not Applicable

##### Derived from

$dwGenAutres

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenChauffage

##### Description

Not Applicable

##### Derived from

$dwGenAutres

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenClapet2S

##### Description

Not Applicable

##### Derived from

$dwGenVanne3Input

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bForceCommandeActive | boolean |  |  |  |  |  |
| bStatusEnDefautDroite | boolean |  |  |  |  |  |
| bStatusEnDefautGauche | boolean |  |  |  |  |  |
| bStatusFCDroite | boolean |  |  |  |  |  |
| bStatusFCGauche | boolean |  |  |  |  |  |
| bStatusMouvementDroite | boolean |  |  |  |  |  |
| bStatusMouvementGauche | boolean |  |  |  |  |  |
| iPlcStatusFCDroite\_FB | integer |  |  |  |  |  |
| iPlcStatusFCGauche\_FB | integer |  |  |  |  |  |
| iStatusLocal\_FB | integer |  |  |  |  |  |
| iTypeCommandeDroite | integer |  |  |  |  |  |
| iTypeCommandeGauche | integer |  |  |  |  |  |
| iValCommandeDroite | integer |  |  |  |  |  |
| iValCommandeGauche | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |
| sTexteDroite | string |  |  |  |  |  |
| sTexteGauche | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcStatusFCDroite

|  |  |
| --- | --- |
| Name | dcPlcStatusFCDroite |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusFCDroite + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 16 Avril 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusDroite");  
 logmessage("iPlcStatusFCDroite : " + Me.iPlcStatusFCDroite);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusFCDroite = 0;  
Me.bStatusEnDefautDroite = 0;  
Me.bStatusMouvementDroite = 0;  
  
IF Me.iTypeAnimation == 10 OR Me.iTypeAnimation == 50   
OR (Me.iTypeAnimation == 60 AND (Me.iTypeAnimationAlternative == 1 OR Me.iTypeAnimationAlternative == 2))  
OR (Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 0)  
OR (Me.iTypeAnimation == 40 AND Me.iTypeAnimationAlternative == 0)  
OR Me.iTypeAnimation == 70 ' CAC/FMT  
OR Me.iTypeAnimation == 90 ' FR  
THEN ' PFEIFFER - BR8, BR6, COM, HGS  
 IF (Me.iPlcStatusFCDroite == 0) THEN  
 Me.bStatusFCDroite = 0; ' vert  
 ELSEIF (Me.iPlcStatusFCDroite == 1) THEN  
 Me.bStatusFCDroite = 1; ' rouge  
 ENDIF; ' IF (Me.iPlcStatusFCDroite == )  
  
ELSEIF Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 0 THEN ' COM  
 ' Jaune non-traité  
 IF (Me.iPlcStatusFCDroite == 1) THEN  
 Me.bStatusFCDroite = 0; ' vert  
 ELSEIF (Me.iPlcStatusFCDroite == 2) THEN  
 Me.bStatusFCDroite = 1; ' rouge  
 ELSEIF (Me.iPlcStatusFCDroite == 8) THEN  
 Me.bStatusFCDroite = 0; ' vert cligno  
 Me.bStatusEnDefautDroite = 1;  
 ELSEIF (Me.iPlcStatusFCDroite == 16) THEN  
 Me.bStatusFCDroite = 1; ' rouge cligno  
 Me.bStatusEnDefautDroite = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCDroite == )  
  
ELSEIF Me.iTypeAnimation == 20 THEN ' BR7  
 IF (Me.iPlcStatusFCDroite == 0) THEN  
 Me.bStatusFCDroite = 0; ' vert  
 ELSEIF (Me.iPlcStatusFCDroite == 1) THEN  
 Me.bStatusFCDroite = 1; ' rouge  
 ELSEIF (Me.iPlcStatusFCDroite == 2) THEN  
 Me.bStatusFCDroite = 0; ' vert cligno  
 Me.bStatusEnDefautDroite = 1;  
 ELSEIF (Me.iPlcStatusFCDroite == 3) THEN  
 Me.bStatusFCDroite = 1; ' rouge cligno  
 Me.bStatusEnDefautDroite = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCDroite == )  
  
ELSEIF Me.iTypeAnimation == 40 AND Me.iTypeAnimationAlternative == 1 THEN  
 ' Jaune non-traité  
 IF (Me.iPlcStatusFCDroite == 1) THEN  
 ' Vert  
 Me.bStatusFCDroite = 0;  
 ELSEIF (Me.iPlcStatusFCDroite == 2) OR (Me.iPlcStatusFCDroite == 3) THEN  
 ' Rouge  
 Me.bStatusFCDroite = 1;  
 ELSEIF (Me.iPlcStatusFCDroite >= 8) AND (Me.iPlcStatusFCDroite < 16) THEN  
 Me.bStatusFCDroite = 0; ' vert cligno  
 Me.bStatusEnDefautDroite = 1;  
 ELSEIF (Me.iPlcStatusFCDroite >= 16) AND (Me.iPlcStatusFCDroite < 32) THEN  
 Me.bStatusFCDroite = 1; ' rouge cligno  
 Me.bStatusEnDefautDroite = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 1 THEN 'HGS\_V44A  
 IF (Me.iPlcStatusFCDroite == 0) THEN  
 Me.bStatusFCDroite = 1; ' rouge cligno  
 Me.bStatusEnDefautDroite = 1;  
 ELSEIF (Me.iPlcStatusFCDroite == 1) THEN  
 ' Rouge  
 Me.bStatusFCDroite = 1;  
 ELSEIF (Me.iPlcStatusFCDroite == 2) THEN  
 ' Vert  
 Me.bStatusFCDroite = 0;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 2 THEN 'HGS\_SPSAIGUIL  
 IF (Me.iPlcStatusFCDroite == 0) THEN  
 ' Vert  
 Me.bStatusFCDroite = 0;  
 ELSEIF (Me.iPlcStatusFCDroite == 1) THEN  
 ' Rouge  
 Me.bStatusFCDroite = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 0) THEN 'TSFx  
 IF (Me.iPlcStatusFCDroite == 0) THEN  
 ' Vert  
 Me.bStatusFCDroite = 0;  
 ELSEIF Me.iPlcStatusFCDroite == 1 THEN  
 ' Jaune  
 Me.bStatusMouvementDroite = 1;  
 ELSEIF (Me.iPlcStatusFCDroite == 2) OR (Me.iPlcStatusFCDroite == 3) THEN  
 ' Rouge  
 Me.bStatusFCDroite = 1;  
 ELSEIF (Me.iPlcStatusFCDroite == 4) OR (Me.iPlcStatusFCDroite == 5) THEN  
 Me.bStatusFCDroite = 0; ' vert cligno  
 Me.bStatusEnDefautDroite = 1;  
 ELSEIF (Me.iPlcStatusFCDroite >= 6) THEN  
 Me.bStatusFCDroite = 1; ' rouge cligno  
 Me.bStatusEnDefautDroite = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 1) THEN 'TSFS\_CL4  
 IF (Me.iPlcStatusFCDroite == 0) THEN  
 ' Vert  
 Me.bStatusFCDroite = 0;  
 ELSEIF (Me.iPlcStatusFCDroite == 1) THEN  
 ' Rouge  
 Me.bStatusFCDroite = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN

dcPlcStatusFCGauche

|  |  |
| --- | --- |
| Name | dcPlcStatusFCGauche |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusFCGauche + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 16 Avril 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusGauche");  
 logmessage("iPlcStatusFCGauche: " + Me.iPlcStatusFCGauche);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusFCGauche = 0;  
Me.bStatusEnDefautGauche = 0;  
Me.bStatusMouvementGauche = 0;  
  
IF Me.iTypeAnimation == 10 OR Me.iTypeAnimation == 50  
OR (Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 1)   
OR (Me.iTypeAnimation == 40 AND Me.iTypeAnimationAlternative == 0)  
OR (Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 0)  
OR Me.iTypeAnimation == 70 ' CAC/FMT  
OR Me.iTypeAnimation == 90 ' FR  
THEN ' PFEIFFER - BR8, BR6, COM, HGS\_V16  
 IF (Me.iPlcStatusFCGauche == 0) THEN  
 Me.bStatusFCGauche = 0; ' vert  
 ELSEIF (Me.iPlcStatusFCGauche == 1) THEN  
 Me.bStatusFCGauche = 1; ' rouge  
 ENDIF; ' IF (Me.iPlcStatusFCGauche == )  
  
ELSEIF (Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 2)THEN ' COM\_CL13  
 IF (Me.iPlcStatusFCGauche == 0) THEN  
 Me.bStatusFCGauche = 1; ' rouge  
 ELSEIF (Me.iPlcStatusFCGauche == 1) THEN  
 Me.bStatusFCGauche = 0; ' vert  
 ENDIF; ' IF (Me.iPlcStatusFCGauche == )  
  
ELSEIF Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 0 THEN  
 ' Jaune non-traité  
 IF (Me.iPlcStatusFCGauche == 1) THEN  
 Me.bStatusFCGauche = 1; ' rouge  
 ELSEIF (Me.iPlcStatusFCGauche == 2) THEN  
 Me.bStatusFCGauche = 0; ' vert  
 ELSEIF (Me.iPlcStatusFCGauche == 8) THEN  
 Me.bStatusFCGauche = 1; ' rouge cligno  
 Me.bStatusEnDefautGauche = 1;  
 ELSEIF (Me.iPlcStatusFCGauche == 16) THEN  
 Me.bStatusFCGauche = 0; ' vert cligno  
 Me.bStatusEnDefautGauche = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCGauche == )  
  
ELSEIF Me.iTypeAnimation == 20 THEN  
 IF (Me.iPlcStatusFCGauche == 0) THEN  
 Me.bStatusFCGauche = 0; ' vert  
 ELSEIF (Me.iPlcStatusFCGauche == 1) THEN  
 Me.bStatusFCGauche = 1; ' rouge  
 ELSEIF (Me.iPlcStatusFCGauche == 2) THEN  
 Me.bStatusFCGauche = 0; ' vert cligno  
 Me.bStatusEnDefautGauche = 1;  
 ELSEIF (Me.iPlcStatusFCGauche == 3) THEN  
 Me.bStatusFCGauche = 1; ' rouge cligno  
 Me.bStatusEnDefautGauche = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCGauche == )  
  
ELSEIF Me.iTypeAnimation == 40 AND Me.iTypeAnimationAlternative == 1 THEN  
 ' Jaune non-traité  
 IF (Me.iPlcStatusFCGauche == 0) THEN  
 ' Vert  
 Me.bStatusFCGauche = 0;  
 ELSEIF (Me.iPlcStatusFCGauche == 2) OR (Me.iPlcStatusFCGauche == 3) THEN  
 ' Rouge  
 Me.bStatusFCGauche = 1;  
 ELSEIF (Me.iPlcStatusFCGauche >= 8) AND (Me.iPlcStatusFCGauche < 16) THEN  
 Me.bStatusFCGauche = 0; ' vert cligno  
 Me.bStatusEnDefautGauche = 1;  
 ELSEIF (Me.iPlcStatusFCGauche >= 16) AND (Me.iPlcStatusFCGauche < 32) THEN  
 Me.bStatusFCGauche = 1; ' rouge cligno  
 Me.bStatusEnDefautGauche = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 1 THEN ' HGS\_V44A  
 IF (Me.iPlcStatusFCGauche == 0) THEN  
 Me.bStatusFCGauche = 1; ' rouge cligno  
 Me.bStatusEnDefautGauche = 1;  
 ELSEIF (Me.iPlcStatusFCGauche == 1) THEN  
 ' Vert  
 Me.bStatusFCGauche = 0;  
 ELSEIF (Me.iPlcStatusFCGauche == 2) THEN  
 ' Rouge  
 Me.bStatusFCGauche = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
  
ELSEIF Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 2 THEN ' HGS\_SPSAIGUIL  
 IF (Me.iPlcStatusFCGauche == 0) THEN  
 ' Rouge  
 Me.bStatusFCGauche = 1;  
 ELSEIF (Me.iPlcStatusFCGauche == 1) THEN  
 ' Vert  
 Me.bStatusFCGauche = 0;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 0) THEN 'TSFx  
 IF (Me.iPlcStatusFCGauche == 0) THEN  
 ' Vert  
 Me.bStatusFCGauche = 0;  
 ELSEIF (Me.iPlcStatusFCGauche == 1) THEN  
 ' Jaune  
 Me.bStatusMouvementGauche = 1;  
 ELSEIF (Me.iPlcStatusFCGauche == 2) OR (Me.iPlcStatusFCGauche == 3) THEN  
 ' Rouge  
 Me.bStatusFCGauche = 1;  
 ELSEIF (Me.iPlcStatusFCGauche == 4) OR (Me.iPlcStatusFCGauche == 5) THEN  
 Me.bStatusFCGauche = 0; ' vert cligno  
 Me.bStatusEnDefautGauche = 1;  
 ELSEIF (Me.iPlcStatusFCGauche >= 6) THEN  
 Me.bStatusFCGauche = 1; ' rouge cligno  
 Me.bStatusEnDefautGauche = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 1) THEN 'TSFS\_CL4  
 IF (Me.iPlcStatusFCGauche == 0) THEN  
 ' Vert  
 Me.bStatusFCGauche = 0;  
 ELSEIF (Me.iPlcStatusFCGauche == 1) THEN  
 ' Rouge  
 Me.bStatusFCGauche = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN

dcRAZCmdDroite

|  |  |
| --- | --- |
| Name | dcRAZCmdDroite |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusFCDroite |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Juillet 2008  
' Ce script remettra le bit de commande à zéro pour certaines applications  
' iTypeAnimation : 50 = BR6  
  
IF (Me.bPlcCommandeDroite == 1 OR Me.iPlcCommandeDroite <> 0) AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70) THEN  
 Me.bPlcCommandeDroite = 0;  
 Me.iPlcCommandeDroite = 0;  
ENDIF;  
  
Me.iTempo = 0;

dcRAZCmdGauche

|  |  |
| --- | --- |
| Name | dcRAZCmdGauche |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusFCGauche |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Juillet 2008  
' Ce script remettra le bit de commande à zéro pour certaines applications  
' iTypeAnimation : 50 = BR6  
  
IF (Me.bPlcCommandeGauche == 1 OR Me.iPlcCommandeGauche <> 0) AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70) THEN  
 Me.bPlcCommandeGauche = 0;  
 Me.iPlcCommandeGauche = 0;  
ENDIF;  
  
Me.iTempo = 0;

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenClapet2S  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenClapet2S  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatusFCDroite.Input.InputSource = "Me.iPlcStatusFCDroite\_FB";  
Me.iPlcStatusFCGauche.Input.InputSource = "Me.iPlcStatusFCGauche\_FB";  
Me.iStatusLocal.Input.InputSource = "Me.iStatusLocal\_FB";

RAZ\_Tempo

|  |  |
| --- | --- |
| Name | RAZ\_Tempo |
| Description |  |
| Trigger | WhileTrue of Me.iTempo > 15 |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Juillet 2008  
' Ce script remettra le bit de commande à zéro pour certaines applications  
' iTypeAnimation : 50 = BR6  
  
IF (Me.bPlcCommandeGauche == 1 OR Me.iPlcCommandeGauche <> 0) AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70) THEN  
 Me.bPlcCommandeGauche = 0;  
 Me.iPlcCommandeGauche = 0;  
ENDIF;  
  
IF (Me.bPlcCommandeDroite == 1 OR Me.iPlcCommandeDroite <> 0) AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70) THEN  
 Me.bPlcCommandeDroite = 0;  
 Me.iPlcCommandeDroite = 0;  
ENDIF;  
  
Me.iTempo = 0;

wtCommandeDroite

|  |  |
| --- | --- |
| Name | wtCommandeDroite |
| Description |  |
| Trigger | WhileTrue of Me.bPlcCommandeDroite == 1 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

wtCommandeGauche

|  |  |
| --- | --- |
| Name | wtCommandeGauche |
| Description |  |
| Trigger | WhileTrue of Me.bPlcCommandeGauche == 1 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

#### Template ArchestrA $dwGenClapet2SMotorise

##### Description

Not Applicable

##### Derived from

$dwGenClapet2S

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bStatusArret | boolean |  |  |  |  |  |
| bStatusMarche | boolean |  |  |  |  |  |
| bStatusMouvement | boolean |  |  |  |  |  |
| bStatusNoPosition | boolean |  |  |  |  |  |
| iPlcStatus\_FB | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcStatus

|  |  |
| --- | --- |
| Name | dcPlcStatus |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 16 Avril 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatus");  
 logmessage("iPlcStatus : " + Me.iPlcStatus);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusMarche = 0;  
Me.bStatusArret = 0;  
Me.bStatusEnDefaut = 0;  
Me.bStatusMouvement = 0;  
  
IF Me.iTypeAnimation == 10 THEN ' PFEIFFER - BR8  
 IF (Me.iPlcStatus == 0) THEN  
 Me.bStatusMarche = 0;  
 Me.bStatusArret = 1;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.bStatusMarche = 1;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.bStatusMarche = 0;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 1;   
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 40 AND Me.iTypeAnimationAlternative == 0)   
OR (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 0)  
THEN 'BRO, FR  
 IF (Me.iPlcStatus == 0) THEN  
 Me.bStatusMarche = 0;  
 Me.bStatusArret = 1;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.bStatusMarche = 1;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 1;   
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 40 AND Me.iTypeAnimationAlternative == 1)  
OR Me.iTypeAnimation == 70 'BRO\_BR4CL9  
OR (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 1) ' FR1\_CL1, FR2\_CL2  
THEN  
 IF (Me.iPlcStatus == 1) THEN  
 ' Vert  
 Me.bStatusMarche = 0;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusMouvement = 0;  
 ELSEIF (Me.iPlcStatus == 2) OR (Me.iPlcStatus == 3) THEN  
 ' Rouge  
 Me.bStatusMarche = 1;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusMouvement = 0;  
 ELSEIF (Me.iPlcStatus >= 4) AND (Me.iPlcStatus < 8) THEN  
 ' Jaune  
 Me.bStatusMarche = 0;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusMouvement = 1;  
 ELSEIF (Me.iPlcStatus >= 8) AND (Me.iPlcStatus < 16) THEN  
 ' Vert clignotant  
 Me.bStatusMarche = 0;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus >= 16) AND (Me.iPlcStatus < 32) THEN  
 ' Rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus >= 32) THEN  
 ' Jaune clignotant  
 Me.bStatusEnDefaut = 1;  
 Me.bStatusMouvement = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 0 THEN ' HGS\_V16  
 IF (Me.iPlcStatus == 0) THEN  
 ' Vert  
 Me.bStatusMarche = 0;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusMouvement = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' Rouge  
 Me.bStatusMarche = 1;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusMouvement = 0;  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 ' Rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 1 THEN ' HGS\_V44A  
 IF (Me.iPlcStatus == 0) THEN  
 ' Rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' Vert  
 Me.bStatusMarche = 0;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusMouvement = 0;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 80 THEN 'TSFx  
 IF Me.iPlcStatus == 0 THEN  
 ' Vert  
 Me.bStatusMarche = 0;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusMouvement = 0;  
 ELSEIF Me.iPlcStatus == 1 THEN  
 ' Rouge  
 Me.bStatusMarche = 1;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusMouvement = 0;  
 ELSEIF Me.iPlcStatus == 2 THEN  
 ' Vert clignotant  
 Me.bStatusMarche = 0;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF Me.iPlcStatus == 3 THEN  
 ' Rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusArret = 0;  
 Me.bStatusEnDefaut = 1;  
 ENDIF;  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN

#### Template ArchestrA $dwGenClapet3S

##### Description

Not Applicable

##### Derived from

$dwGenVanne5input

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bStatusDefautMilieu | boolean |  |  |  |  |  |
| bStatusEnDefautDroite | boolean |  |  |  |  |  |
| bStatusEnDefautGauche | boolean |  |  |  |  |  |
| bStatusFdcDrDroite | boolean |  |  |  |  |  |
| bStatusFdcDrGauche | boolean |  |  |  |  |  |
| bStatusFdcGDroite | boolean |  |  |  |  |  |
| bStatusFdcGGauche | boolean |  |  |  |  |  |
| bStatusMoteurArret | boolean |  |  |  |  |  |
| bStatusMoteurDefaut | boolean |  |  |  |  |  |
| bStatusMoteurMarche | boolean |  |  |  |  |  |
| bStatusMoteurMouvement | boolean |  |  |  |  |  |
| bStatusMouvementDroite | boolean |  |  |  |  |  |
| bStatusMouvementGauche | boolean |  |  |  |  |  |
| bStatusMouvementMilieu | boolean |  |  |  |  |  |
| bStatusTrDroite | boolean |  |  |  |  |  |
| bStatusTrGauche | boolean |  |  |  |  |  |
| bStatusTrMilieu | boolean |  |  |  |  |  |
| iTypeCommandeDroite | integer |  |  |  |  |  |
| iTypeCommandeGauche | integer |  |  |  |  |  |
| iTypeCommandeMilieu | integer |  |  |  |  |  |
| iValCommandeDroite | integer |  |  |  |  |  |
| iValCommandeGauche | integer |  |  |  |  |  |
| iValCommandeMilieu | integer |  |  |  |  |  |
| sTexteDroite | string |  |  |  |  |  |
| sTexteGauche | string |  |  |  |  |  |
| sTexteMilieu | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dPlcStatusTrDroite

|  |  |
| --- | --- |
| Name | dPlcStatusTrDroite |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusTrDroite + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

Me.bStatusTrDroite = 0;  
Me.bStatusEnDefautDroite = 0;  
Me.bStatusMouvementDroite = 0;  
Me.bStatusFdcDrDroite = 0;  
Me.bStatusFdcDrGauche = 0;  
  
  
IF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 0) THEN 'TSFx  
 IF (Me.iPlcStatusTrDroite == 0) THEN  
 ' Vert  
 Me.bStatusTrDroite = 0;  
 Me.bStatusFdcDrDroite = 1;  
 Me.bStatusFdcDrGauche = 0;   
  
 ELSEIF Me.iPlcStatusTrDroite == 1 THEN  
 ' Jaune  
 Me.bStatusMouvementDroite = 1;  
 Me.bStatusFdcDrDroite = 0;  
 Me.bStatusFdcDrGauche = 0;  
  
 ELSEIF (Me.iPlcStatusTrDroite == 2) OR (Me.iPlcStatusTrDroite == 3) THEN  
 ' Rouge  
 Me.bStatusTrDroite = 1;  
 Me.bStatusFdcDrDroite = 0;  
 Me.bStatusFdcDrGauche = 1;  
  
 ELSEIF (Me.iPlcStatusTrDroite == 4) OR (Me.iPlcStatusTrDroite == 5) THEN  
 Me.bStatusTrDroite = 0; ' vert cligno  
 Me.bStatusEnDefautDroite = 1;  
 Me.bStatusFdcDrDroite = 1;  
 Me.bStatusFdcDrGauche = 0;  
  
 ELSEIF (Me.iPlcStatusTrDroite >= 6) THEN  
 Me.bStatusTrDroite = 1; ' rouge cligno  
 Me.bStatusEnDefautDroite = 1;  
 Me.bStatusFdcDrDroite = 0;  
 Me.bStatusFdcDrGauche = 1;  
  
 ENDIF; ' IF (Me.iPlcStatus == )  
ENDIF;

dPlcStatusTrGauche

|  |  |
| --- | --- |
| Name | dPlcStatusTrGauche |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusTrGauche + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

Me.bStatusTrGauche = 0;  
Me.bStatusEnDefautGauche = 0;  
Me.bStatusMouvementGauche = 0;  
Me.bStatusFdcGDroite = 0;  
Me.bStatusFdcGGauche = 0;  
  
IF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 0) THEN 'TSFx  
 IF (Me.iPlcStatusTrGauche == 0) THEN  
 ' Vert  
 Me.bStatusTrGauche = 0;  
 Me.bStatusFdcGDroite = 0;  
 Me.bStatusFdcGGauche = 1;  
  
 ELSEIF (Me.iPlcStatusTrGauche == 1) THEN  
 ' Jaune  
 Me.bStatusMouvementGauche = 1;  
 Me.bStatusFdcGDroite = 0;  
 Me.bStatusFdcGGauche = 0;  
  
 ELSEIF (Me.iPlcStatusTrGauche == 2) OR (Me.iPlcStatusTrGauche == 3) THEN  
 ' Rouge  
 Me.bStatusTrGauche = 1;  
 Me.bStatusFdcGDroite = 1;  
 Me.bStatusFdcGGauche = 0;  
 ELSEIF (Me.iPlcStatusTrGauche == 4) OR (Me.iPlcStatusTrGauche == 5) THEN  
 Me.bStatusTrGauche = 0; ' vert cligno  
 Me.bStatusEnDefautGauche = 1;  
 Me.bStatusFdcGDroite = 0;  
 Me.bStatusFdcGGauche = 1;  
 ELSEIF (Me.iPlcStatusTrGauche >= 6) THEN  
 Me.bStatusTrGauche = 1; ' rouge cligno  
 Me.bStatusEnDefautGauche = 1;  
 Me.bStatusFdcGDroite = 1;  
 Me.bStatusFdcGGauche = 0;  
 ENDIF; ' IF (Me.iPlcStatus == )  
ENDIF;

dPlcStatusTrMilieu

|  |  |
| --- | --- |
| Name | dPlcStatusTrMilieu |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusTrMilieu + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

Me.bStatusTrMilieu = 0;  
Me.bStatusDefautMilieu = 0;  
Me.bStatusMouvementMilieu = 0;  
Me.bStatusFdcDrDroite = 0;  
Me.bStatusFdcDrGauche = 0;  
Me.bStatusFdcGDroite = 0;  
Me.bStatusFdcGGauche = 0;  
  
IF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 0) THEN 'TSFx  
 IF (Me.iPlcStatusTrMilieu == 0) THEN  
 ' Vert  
 Me.bStatusTrMilieu = 0;  
  
 ELSEIF (Me.iPlcStatusTrMilieu == 1) THEN  
 ' Jaune  
 Me.bStatusMouvementMilieu = 1;  
  
 ELSEIF (Me.iPlcStatusTrMilieu == 2) OR (Me.iPlcStatusTrMilieu == 3) THEN  
 ' Rouge  
 Me.bStatusTrMilieu = 1;  
  
  
 ELSEIF (Me.iPlcStatusTrMilieu == 4) OR (Me.iPlcStatusTrMilieu == 5) THEN  
 Me.bStatusTrMilieu = 0; ' vert cligno  
 Me.bStatusDefautMilieu = 1;  
  
  
 ELSEIF (Me.iPlcStatusTrMilieu >= 6) THEN  
 Me.bStatusTrMilieu = 1; ' rouge cligno  
 Me.bStatusDefautMilieu = 1;  
  
 ENDIF; ' IF (Me.iPlcStatus == )  
ENDIF;

dPlcstatusMoteur

|  |  |
| --- | --- |
| Name | dPlcstatusMoteur |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusMoteur + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

Me.bStatusMoteurMarche = 0;  
 Me.bStatusMoteurArret = 0;  
 Me.bStatusMoteurDefaut = 0;  
 Me.bStatusMoteurMouvement = 0;  
  
IF Me.iTypeAnimation == 80 THEN 'TSFx  
 IF Me.iPlcStatusMoteur == 0 THEN  
 ' Vert  
 Me.bStatusMoteurMarche = 0;  
 Me.bStatusMoteurArret = 1;  
 Me.bStatusMoteurDefaut = 0;  
 Me.bStatusMoteurMouvement = 0;  
 ELSEIF Me.iPlcStatusMoteur == 1 THEN  
 ' Rouge  
 Me.bStatusMoteurMarche = 1;  
 Me.bStatusMoteurArret = 0;  
 Me.bStatusMoteurDefaut = 0;  
 Me.bStatusMoteurMouvement = 0;  
 ELSEIF Me.iPlcStatusMoteur == 2 THEN  
 ' Vert clignotant  
 Me.bStatusMoteurMarche = 0;  
 Me.bStatusMoteurArret = 1;  
 Me.bStatusMoteurDefaut = 1;  
 ELSEIF Me.iPlcStatusMoteur == 3 THEN  
 ' Rouge clignotant  
 Me.bStatusMoteurMarche = 1;  
 Me.bStatusMoteurArret = 0;  
 Me.bStatusMoteurDefaut = 1;  
 ENDIF;  
  
ENDIF;

#### Template ArchestrA $dwGenClapetMotorise

##### Description

Not Applicable

##### Derived from

$dwGenVanne1Input

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bTjsCommandable | boolean |  |  |  |  |  |
| iPlcStatus\_FB | integer |  |  |  |  |  |
| iStatusLocal\_FB | integer |  |  |  |  |  |
| iTypeCommandeFermeture | integer |  |  |  |  |  |
| iTypeCommandeOuverture | integer |  |  |  |  |  |
| iValCommandeFermeture | integer |  |  |  |  |  |
| iValCommandeOuverture | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |
| sTexteFerme | string |  |  |  |  |  |
| sTexteOuvert | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcStatus

|  |  |
| --- | --- |
| Name | dcPlcStatus |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 16 Avril 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatus");  
 logmessage("iPlcStatus : " + Me.iPlcStatus);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
IF Me.iTypeAnimation == 10 THEN ' PFEIFFER - BR8  
 IF (Me.iPlcStatus == 0) THEN  
 ' Vert  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' Vert clignotant  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 ' Rouge  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatus == 3) THEN  
 ' Rouge clignotant  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus == 4) THEN  
 ' Jaune  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 1;  
 Me.bStatusEnDefaut = 0;   
 ELSEIF (Me.iPlcStatus == 5) THEN  
 ' Jaune clignotant  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 1;  
 Me.bStatusEnDefaut = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 0 THEN ' BR7  
 IF (Me.iPlcStatus == 0) THEN ' vert = Ferme  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN ' rouge = Ouvert  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 1 THEN ' BR7\_PM7\_EV  
 IF (Me.iPlcStatus == 0) THEN   
 'vert = Ferme  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN   
 ' Jaune  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 1;  
 Me.bStatusEnDefaut = 0;   
 ELSEIF (Me.iPlcStatus == 2) OR (Me.iPlcStatus == 3) THEN   
 'rouge = Ouvert  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatus == 4) OR (Me.iPlcStatus == 5) THEN   
 'vert blink  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus >= 6) THEN   
 'rouge blink  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 0 THEN ' HGS  
 IF (Me.iPlcStatus == 0) THEN   
 'vert = Ferme  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN   
 'rouge = Ouvert  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 1 THEN ' HGS\_V14  
 IF (Me.iPlcStatus == 0) THEN   
 'vert = Ferme  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN   
 'rouge ouvert  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatus == 2) THEN   
 'rouge blink  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus == 4) THEN   
 ' Jaune  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 1;  
 Me.bStatusEnDefaut = 0;   
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 2 THEN ' HGS\_V13  
 IF (Me.iPlcStatus == 0) THEN   
 'vert = Ferme  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN   
 ' vert blink  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus == 2) THEN   
 'rouge = Ouvert  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatus == 3) THEN   
 'rouge blink  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus == 4) THEN   
 ' Jaune  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 1;  
 Me.bStatusEnDefaut = 0;   
 ELSEIF (Me.iPlcStatus == 5) THEN   
 ' Jaune blink  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 1;  
 Me.bStatusEnDefaut = 1;   
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 40 ' BRO  
 OR (Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 0) ' COM  
 OR (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 2) ' colire   
 THEN ' colire  
 IF (Me.iPlcStatus == 1) THEN  
 ' Vert  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
  
 ELSEIF (Me.iPlcStatus == 2) OR (Me.iPlcStatus == 3) THEN  
 ' Rouge  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
  
 ELSEIF (Me.iPlcStatus >= 4) AND (Me.iPlcStatus < 8) THEN  
 ' Jaune  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 1;  
 Me.bStatusEnDefaut = 0;   
  
 ELSEIF (Me.iPlcStatus >= 8) AND (Me.iPlcStatus < 16) THEN  
 ' Vert clignotant  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
  
 ELSEIF (Me.iPlcStatus >= 16) AND (Me.iPlcStatus < 32) THEN  
 ' Rouge clignotant  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
  
 ELSEIF (Me.iPlcStatus >= 32) THEN  
 ' Jaune clignotant  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 1;  
 Me.bStatusEnDefaut = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 1 THEN ' COM\_VM1  
  
 IF (Me.iPlcStatus == 0) THEN  
 ' Vert  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' Rouge  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ENDIF;  
  
ELSEIF Me.iTypeAnimation == 90 THEN ' FR  
 IF (Me.iPlcStatus == 0) THEN  
 ' Vert  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' Rouge  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 ' Jaune  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 1;  
 Me.bStatusEnDefaut = 0;   
 ELSEIF (Me.iPlcStatus == 3) THEN  
 ' Vert clignotant  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus >= 4) THEN  
 ' Rouge clignotant  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenClapetMotorise  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenClapetMotorise  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatus.Input.InputSource = "Me.iPlcStatus\_FB";  
Me.iStatusLocal.Input.InputSource = "Me.iStatusLocal\_FB";

#### Template ArchestrA $dwGenClaxon

##### Description

Not Applicable

##### Derived from

$dwGenAnimationsBool

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenCO

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenCO\_seuil

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| SeuilBas | boolean |  | X |  |  |  |
| SeuilHaut | integer |  | X |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenCO2

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenCollecteur

##### Description

Not Applicable

##### Derived from

$dwGenAutres

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenCompacteur

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenCompacteur2S

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect2S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenCompresseur

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenConcasseur

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenCONDUC

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenCrible

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenCSG

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogInOut

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenCSG\_Parametres

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenCSG

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bChgParam | boolean |  |  |  |  |  |
| bChgPV | boolean |  |  |  |  |  |
| bChgStatus | boolean |  |  |  |  |  |
| fParam | float | X |  |  |  |  |
| iStatusAuto | integer |  | X |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPV

|  |  |
| --- | --- |
| Name | dcPV |
| Description |  |
| Trigger | DataChange of Me.PV |

**Declarations :**

Not Applicable

**Script :**

IF (Me.iStatusAuto == 1 OR Me.iStatusAuto == 4) AND Me.bChgParam == 0 THEN  
 Me.bChgPV = 1;  
 Me.fParam = Me.PV;  
ENDIF;  
  
Me.bChgParam = 0;

dcParam

|  |  |
| --- | --- |
| Name | dcParam |
| Description |  |
| Trigger | DataChange of Me.fParam |

**Declarations :**

Not Applicable

**Script :**

IF (Me.iStatusAuto == 1 OR Me.iStatusAuto == 4) AND Me.bChgStatus == 0 AND Me.bChgPV == 0 THEN  
 Me.bChgParam = 1;  
 Me.PV = Me.fParam;  
ENDIF;  
  
Me.bChgStatus = 0;  
Me.bChgPV = 0;

dcStatusAuto

|  |  |
| --- | --- |
| Name | dcStatusAuto |
| Description |  |
| Trigger | DataChange of Me.iStatusAuto |

**Declarations :**

Not Applicable

**Script :**

IF Me.iStatusAuto == 1 OR Me.iStatusAuto == 4 THEN  
 Me.bChgStatus = 1;  
 Me.fParam = Me.PV;  
ENDIF;

#### Template ArchestrA $dwGenCuves

##### Description

Not Applicable

##### Derived from

$dwGenObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| iPlcStatus\_FB | integer |  |  |  |  |  |
| iStatus | integer |  |  |  |  |  |
| iTypeAnimationAlternative | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcStatus

|  |  |
| --- | --- |
| Name | dcPlcStatus |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Avril 2008  
' Ce script traitera les status  
' status = 0 - vert  
' status = 1 - rouge  
' status = 2 - jaune  
' status = 3 - vert clignotant  
' status = 4 - rouge clignotant  
' status = 5 - jaune clignotant  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatus");  
 logmessage("iPlcStatus : " + Me.iPlcStatus);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
IF Me.iTypeAnimation == 30 THEN ' HGS  
 IF (Me.iPlcStatus == 0) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus == 3) THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ELSEIF (Me.iPlcStatus == 4) THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF (Me.iPlcStatus == 5) THEN  
 Me.iStatus = 5; ' jaune clignotant  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 50 THEN ' BR6  
 IF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF (Me.iPlcStatus == 4) THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF (Me.iPlcStatus == 8) THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF (Me.iPlcStatus == 16) THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ELSEIF (Me.iPlcStatus == 32) THEN  
 Me.iStatus = 5; ' jaune clignotant  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 70 THEN 'CAC/FMT  
 IF (Me.iPlcStatus == 0) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 1; ' rouge  
 ENDIF;  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 0) THEN ' TSFx  
 IF (Me.iPlcStatus == 0) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 2; ' jaune  
 ELSEIF Me.iPlcStatus == 2 OR Me.iPlcStatus == 3 THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF Me.iPlcStatus == 4 OR Me.iPlcStatus == 5 THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF (Me.iPlcStatus >= 6) THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 1) THEN ' TSFM\_SILO\_M2, TSFM\_SILO\_M2\_2  
 IF (Me.iPlcStatus == 0) THEN  
 Me.iStatus = 0; ' vert  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.iStatus = 1; ' rouge  
 ELSEIF Me.iPlcStatus == 2 THEN  
 Me.iStatus = 3; ' vert clignotant  
 ELSEIF Me.iPlcStatus == 3 THEN  
 Me.iStatus = 4; ' rouge clignotant  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenCuves  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenCuves  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatus.Input.InputSource = "Me.iPlcStatus\_FB";

#### Template ArchestrA $dwGenCuvesAvecCommande

##### Description

Not Applicable

##### Derived from

$dwGenCuves

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| iTypeCommande | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcRAZCmd

|  |  |
| --- | --- |
| Name | dcRAZCmd |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Juillet 2008  
' Ce script remettra le bit de commande à zéro pour certaines applications  
' iTypeAnimation : 50 = BR6  
  
IF Me.bPlcCommandeManu == 1 AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70) THEN  
 Me.bPlcCommandeManu = 0;  
ENDIF;  
  
Me.iTempo = 0;

RAZ\_Tempo

|  |  |
| --- | --- |
| Name | RAZ\_Tempo |
| Description |  |
| Trigger | WhileTrue of Me.iTempo > 15 |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Juillet 2008  
' Ce script remettra le bit de commande à zéro pour certaines applications  
' iTypeAnimation : 50 = BR6  
  
IF Me.bPlcCommandeManu == 1 AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70) THEN  
 Me.bPlcCommandeManu = 0;  
ENDIF;  
  
Me.iTempo = 0;

wtCommande

|  |  |
| --- | --- |
| Name | wtCommande |
| Description |  |
| Trigger | WhileTrue of Me.bPlcCommandeManu |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

#### Template ArchestrA $dwGenDEB

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenDEC\_DEM

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenDecolmateur

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenDetecteurMetaux

##### Description

Not Applicable

##### Derived from

$dwGenAnimationsBool

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenDoseur

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenDP

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenElectroVanne

##### Description

Not Applicable

##### Derived from

$dwGenVanne1Input

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| iPlcStatus\_FB | integer |  |  |  |  |  |
| iStatusLocal\_FB | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcStatus

|  |  |
| --- | --- |
| Name | dcPlcStatus |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 16 Avril 2008  
' Ce script traitera les status  
' Ferme : vert  
' Ouvert : rouge  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatus");  
 logmessage("iPlcStatus : " + Me.iPlcStatus);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusFinCourseFerme = 0;  
Me.bStatusFinCourseOuvert = 0;  
Me.bStatusMouvement = 0;  
Me.bStatusEnDefaut = 0;  
  
IF Me.iTypeAnimation == 10 OR Me.iTypeAnimation == 20 OR Me.iTypeAnimation == 30 OR Me.iTypeAnimation == 50 '' PFEIFFER - BR8, BR7, BR6, HGS  
OR (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 1) ' CAC\_VLV1\_SILO3  
OR (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 0) ' FR  
THEN   
 IF (Me.iPlcStatus == 0) THEN  
 ' VERT  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' ROUGE  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
ELSEIF Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 2 THEN 'FMT2 Clapets combustibles  
 IF (Me.iPlcStatus == 1) THEN  
 ' VERT  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 ELSEIF (Me.iPlcStatus == 0) THEN  
 ' ROUGE  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
ELSEIF Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 1 THEN ' HGS\_SPSVANNE\_TOR  
' Couleur jaune pas gérée  
 IF (Me.iPlcStatus == 0) THEN   
 ' vert  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' vert clig  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 ' rouge  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 ELSEIF (Me.iPlcStatus == 3) THEN  
 ' rouge clig  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusEnDefaut = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 0)  
OR Me.iTypeAnimation == 40 ' COM OR BRO  
OR (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 0) ' CAC/FMT  
THEN   
' Couleur jaune pas gérée  
 IF (Me.iPlcStatus == 1) THEN  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 ELSEIF (Me.iPlcStatus == 2) OR (Me.iPlcStatus == 3) THEN  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 ELSEIF (Me.iPlcStatus >= 8) AND (Me.iPlcStatus < 16) THEN  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus >= 16) AND (Me.iPlcStatus < 32) THEN  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusEnDefaut = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 1 THEN ' COM\_VM1  
 IF (Me.iPlcStatus == 0) THEN  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 80 THEN ' TSF  
 IF (Me.iPlcStatus == 0) THEN  
 ' Vert  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' Jaune  
 Me.bStatusMouvement = 1;  
 ELSEIF (Me.iPlcStatus == 2) OR (Me.iPlcStatus == 3) THEN  
 ' Rouge  
 Me.bStatusFinCourseOuvert = 1;  
 ELSEIF (Me.iPlcStatus >= 4) AND (Me.iPlcStatus < 6) THEN  
 ' Vert clignotant  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus >= 6) THEN  
 ' Rouge clignotant  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusEnDefaut = 1;   
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 1) THEN ' FR  
 IF (Me.iPlcStatus == 0) THEN  
 ' Vert  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' Rouge  
 Me.bStatusFinCourseOuvert = 1;  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 ' Jaune  
 Me.bStatusMouvement = 1;  
 ELSEIF (Me.iPlcStatus == 3) THEN  
 ' Vert clignotant  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus >= 4) THEN  
 ' Rouge clignotant  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusEnDefaut = 1;   
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 2) THEN ' FR  
 IF (Me.iPlcStatus == 1) THEN  
 ' Vert  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 ELSEIF (Me.iPlcStatus == 2) OR (Me.iPlcStatus == 3) THEN  
 ' Rouge  
 Me.bStatusFinCourseOuvert = 1;  
 ELSEIF (Me.iPlcStatus >= 4) AND (Me.iPlcStatus < 8) THEN  
 ' Jaune  
 Me.bStatusMouvement = 1;  
 ELSEIF (Me.iPlcStatus >= 8) AND (Me.iPlcStatus < 16) THEN  
 ' Vert clignotant  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus >= 16) AND (Me.iPlcStatus < 32) THEN  
 ' Rouge clignotant  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusEnDefaut = 1;   
 ELSEIF (Me.iPlcStatus >= 32) THEN  
 ' Jaune clignotant  
 Me.bStatusMouvement = 1;  
 Me.bStatusEnDefaut = 1;   
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 100 THEN 'EAU  
 IF (Me.iPlcStatus == 0) THEN  
 ' Vert  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' Rouge  
 Me.bStatusFinCourseOuvert = 1;  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 ' Vert clignotant  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusEnDefaut = 1;  
 ENDIF;  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenElectroVanne  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenElectroVanne  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatus.Input.InputSource = "Me.iPlcStatus\_FB";  
Me.iStatusLocal.Input.InputSource = "Me.iStatusLocal\_FB";

#### Template ArchestrA $dwGenElevateur

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenericMoteurDirect1S

##### Description

Not Applicable

##### Derived from

$dwGenMoteur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bCommandeInhibitionDefRot\_FB | boolean |  |  |  |  |  |
| bCommandeInhibitionFB | boolean |  |  |  |  |  |
| iPlcStatusMarche\_FB | integer |  |  |  |  |  |
| iStatusLocal\_FB | integer |  |  |  |  |  |
| iTypeCommandeArret | integer |  |  |  |  |  |
| iTypeCommandeMarche | integer |  |  |  |  |  |
| iValCommandeArret | integer |  |  |  |  |  |
| iValCommandeMarche | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenMoteurDirect1S  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenMoteurDirect1S  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatusMarche.Input.InputSource = "Me.iPlcStatusMarche\_FB";  
Me.iStatusLocal.Input.InputSource = "Me.iStatusLocal\_FB";  
Me.bCommandeInhibitionDefRot.Input.InputSource = "Me.bCommandeInhibitionDefRot\_FB";

#### Template ArchestrA $dwGenericMoteurDirect2S

##### Description

Not Applicable

##### Derived from

$dwGenMoteur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bCommandeInhibitionDefRot\_FB | boolean |  |  |  |  |  |
| bEnMarche | boolean |  |  |  |  |  |
| bInitDefRotation2 | boolean |  |  |  |  |  |
| bInversion | boolean |  |  |  |  |  |
| bStatusDemarrageArriere | boolean |  |  |  |  |  |
| bStatusDemarrageAvant | boolean |  |  |  |  |  |
| bStatusEnDefautArriere | boolean |  |  |  |  |  |
| bStatusEnDefautAvant | boolean |  |  |  |  |  |
| bStatusMarcheArriere | boolean |  |  |  |  |  |
| bStatusMarcheAvant | boolean |  |  |  |  |  |
| bStatusVidangeArriere | boolean |  |  |  |  |  |
| bStatusVidangeAvant | boolean |  |  |  |  |  |
| iPlcStatusMarche\_FB | integer |  |  |  |  |  |
| iPlcStatusMarcheArriere\_FB | integer |  |  |  |  |  |
| iPlcStatusMarcheAvant\_FB | integer |  |  |  |  |  |
| iStatusLocal\_FB | integer |  |  |  |  |  |
| iTypeCommandeArret | integer |  |  |  |  |  |
| iTypeCommandeDroite | integer |  |  |  |  |  |
| iTypeCommandeGauche | integer |  |  |  |  |  |
| iTypeCommandeMarche | integer |  |  |  |  |  |
| iTypeCommandeMarcheAr | integer |  |  |  |  |  |
| iTypeCommandeMarcheAv | integer |  |  |  |  |  |
| iValCommandeArret | integer |  |  |  |  |  |
| iValCommandeDroite | integer |  |  |  |  |  |
| iValCommandeGauche | integer |  |  |  |  |  |
| iValCommandeMarche | integer |  |  |  |  |  |
| iValCommandeMarcheAr | integer |  |  |  |  |  |
| iValCommandeMarcheAv | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |
| sTexteDroite | string |  |  |  |  |  |
| sTexteGauche | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

CPTForcages\_2

|  |  |
| --- | --- |
| Name | CPTForcages\_2 |
| Description |  |
| Trigger | DataChange of Me.bCommandeInhibitionDefRot\_02 |

**Declarations :**

Not Applicable

**Script :**

Dim indVal as indirect;  
Dim sInputSource as string;  
  
sInputSource = "MyArea.Area";  
indVal.BindTo(sInputSource);  
  
sInputSource = indVal + ".iNbForcages";  
indVal.BindTo(sInputSource);  
  
IF Me.bInitDefRotation2 == 0 THEN  
 If Me.bCommandeInhibitionDefRot\_02 == 1 THEN  
 indVal = indVal + 1;  
 ENDIF;  
 Me.bInitDefRotation = 1;  
  
ELSEIF Me.bInitDefRotation2 == 1 THEN  
 IF Me.bCommandeInhibitionDefRot\_02 == 1 THEN  
 indVal = indVal + 1;  
 ELSEIF Me.bCommandeInhibitionDefRot\_02 == 0 THEN  
 indVal = indVal - 1;  
 ENDIF;  
  
ENDIF;

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenMoteurDirect2S  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenMoteurDirect2S  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatusMarche.Input.InputSource = "Me.iPlcStatusMarche\_FB";  
Me.iPlcStatusMarcheAvant.Input.InputSource = "Me.iPlcStatusMarcheAvant\_FB";  
Me.iPlcStatusMarcheArriere.Input.InputSource = "Me.iPlcStatusMarcheArriere\_FB";  
Me.iStatusLocal.Input.InputSource = "Me.iStatusLocal\_FB";  
Me.bCommandeInhibitionDefRot.Input.InputSource = "Me.bCommandeInhibitionDefRot\_FB";

#### Template ArchestrA $dwGenFiltre

##### Description

Not Applicable

##### Derived from

$dwGenActionneur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bStatusBattage | boolean |  |  |  |  |  |
| bStatusDemarrage | boolean |  |  |  |  |  |
| bStatusMarche | boolean |  |  |  |  |  |
| iPlcStatus\_FB | integer |  |  |  |  |  |
| iTypeCommandeArret | integer |  |  |  |  |  |
| iTypeCommandeMarche | integer |  |  |  |  |  |
| iValCommandeArret | integer |  |  |  |  |  |
| iValCommandeMarche | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcStatus

|  |  |
| --- | --- |
| Name | dcPlcStatus |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 16 Avril 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatus");  
 logmessage("iPlcStatus : " + Me.iPlcStatus);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusMarche = 0;  
Me.bStatusEnDefaut = 0;  
Me.bStatusBattage = 0;  
Me.bStatusDemarrage = 0;  
  
IF Me.iTypeAnimation == 10 THEN ' PFEIFFER - BR8  
 IF (Me.iPlcStatus == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusBattage = 0;  
 Me.bStatusDemarrage = 0;  
  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
  
 ELSEIF (Me.iPlcStatus == 3) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
  
 ELSEIF (Me.iPlcStatus == 4) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
  
 ELSEIF (Me.iPlcStatus == 5) THEN  
 ' Battage - jaune clignotant  
 Me.bStatusBattage = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 50) OR (Me.iTypeAnimation == 40) OR (Me.iTypeAnimation == 60)   
OR (Me.iTypeAnimation == 70) OR (Me.iTypeAnimation == 30) OR (Me.iTypeAnimation == 20) THEN ' BR6 OR BRO or COM or CAC/FMT hgs br7  
 IF (Me.iPlcStatus == 1) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusBattage = 0;  
 Me.bStatusDemarrage = 0;  
  
 ELSEIF (Me.iPlcStatus == 2) OR (Me.iPlcStatus == 3) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
  
 ELSEIF (Me.iPlcStatus >= 4) AND (Me.iPlcStatus < 8) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
  
 ELSEIF (Me.iPlcStatus >= 8) AND (Me.iPlcStatus < 16) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
  
 ELSEIF (Me.iPlcStatus >= 16) AND (Me.iPlcStatus < 32) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
  
 ELSEIF (Me.iPlcStatus >= 32) THEN  
 ' Battage - jaune clignotant  
 Me.bStatusBattage = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF Me.iTypeAnimation == 90 THEN ' FR  
 IF (Me.iPlcStatus == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusBattage = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 ELSEIF (Me.iPlcStatus == 3) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus == 4) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus >= 5) THEN  
 ' Battage - jaune clignotant  
 Me.bStatusBattage = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ENDIF;

dcRAZCmd

|  |  |
| --- | --- |
| Name | dcRAZCmd |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Juillet 2008  
' Ce script remettra le bit de commande à zéro pour certaines applications  
' iTypeAnimation : 50 = BR6  
  
IF (Me.bCommandeManuArret == 1 OR Me.bCommandeManuMarche == 1 OR Me.iCommandeManuArret <> 0 OR Me.iCommandeManuMarche <> 0)  
AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeArret == 3 AND Me.iTypeCommandeMarche == 3)) THEN  
 Me.bCommandeManuArret = 0;  
 Me.bCommandeManuMarche = 0;  
 Me.iCommandeManuArret = 0;  
 Me.iCommandeManuMarche = 0;  
ENDIF;  
  
Me.iTempo = 0;

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenFiltre  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenFiltre  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatus.Input.InputSource = "Me.iPlcStatus\_FB";

RAZ\_Tempo

|  |  |
| --- | --- |
| Name | RAZ\_Tempo |
| Description |  |
| Trigger | WhileTrue of Me.iTempo > 15 |

**Declarations :**

Not Applicable

**Script :**

IF (Me.bCommandeManuArret == 1 OR Me.bCommandeManuMarche == 1 OR Me.iCommandeManuArret <> 0 OR Me.iCommandeManuMarche <> 0)  
AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeArret == 3 AND Me.iTypeCommandeMarche == 3)) THEN  
 Me.bCommandeManuArret = 0;  
 Me.bCommandeManuMarche = 0;  
 Me.iCommandeManuArret = 0;  
 Me.iCommandeManuMarche = 0;  
ENDIF;  
  
Me.iTempo = 0;

wtCommandeMarche

|  |  |
| --- | --- |
| Name | wtCommandeMarche |
| Description |  |
| Trigger | WhileTrue of Me.bCommandeManuMarche == 1 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeMarche == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

wtCommandeArret

|  |  |
| --- | --- |
| Name | wtCommandeArret |
| Description |  |
| Trigger | WhileTrue of Me.bCommandeManuArret == 1 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeArret == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

wtCommandeArretInt

|  |  |
| --- | --- |
| Name | wtCommandeArretInt |
| Description |  |
| Trigger | WhileTrue of Me.iCommandeManuArret <> 0 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeArret == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

wtCommandeMarcheInt

|  |  |
| --- | --- |
| Name | wtCommandeMarcheInt |
| Description |  |
| Trigger | WhileTrue of Me.iCommandeManuMarche <> 0 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeMarche == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

#### Template ArchestrA $dwGenFiltre\_FMT2

##### Description

Not Applicable

##### Derived from

$dwGenActionneur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bStatusBattage | boolean |  |  |  |  |  |
| bStatusDemarrage | boolean |  |  |  |  |  |
| bStatusMarche | boolean |  |  |  |  |  |
| iPlcStatus\_FB | integer |  |  |  |  |  |
| iTypeCommandeArret | integer |  |  |  |  |  |
| iTypeCommandeMarche | integer |  |  |  |  |  |
| iValCommandeArret | integer |  |  |  |  |  |
| iValCommandeMarche | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenFiltre  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : CPO  
' Date : 25/09/2013  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUndeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUndeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenFiltre  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : CPO  
' Date : 25/09/2013  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatus.Input.InputSource = "Me.iPlcStatus\_FB";

dcPlcStatus

|  |  |
| --- | --- |
| Name | dcPlcStatus |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Creator : CPO  
' Date : 25/09/2013  
' Date : 16 Avril 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatus");  
 logmessage("iPlcStatus : " + Me.iPlcStatus);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusMarche = 0;  
Me.bStatusEnDefaut = 0;  
Me.bStatusBattage = 0;  
Me.bStatusDemarrage = 0;  
  
IF Me.iTypeAnimation == 70 THEN ' FMT  
 IF (Me.iPlcStatus == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusBattage = 0;  
 Me.bStatusDemarrage = 0;  
  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
  
 ELSEIF (Me.iPlcStatus == 3) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
  
 ELSEIF (Me.iPlcStatus == 4) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
  
 ELSEIF (Me.iPlcStatus == 5) THEN  
 ' Battage - jaune clignotant  
 Me.bStatusBattage = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ENDIF;

dcRAZCmd

|  |  |
| --- | --- |
| Name | dcRAZCmd |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus |

**Declarations :**

Not Applicable

**Script :**

' Creator : CPO  
' Date : 25/09/2013  
' Date : 30 Juillet 2008  
' Ce script remettra le bit de commande à zéro pour certaines applications  
' iTypeAnimation : 50 = BR6  
  
IF (Me.bCommandeManuArret == 1 OR Me.bCommandeManuMarche == 1 OR Me.iCommandeManuArret <> 0 OR Me.iCommandeManuMarche <> 0)  
AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeArret == 3 AND Me.iTypeCommandeMarche == 3)) THEN  
 Me.bCommandeManuArret = 0;  
 Me.bCommandeManuMarche = 0;  
 Me.iCommandeManuArret = 0;  
 Me.iCommandeManuMarche = 0;  
ENDIF;  
  
Me.iTempo = 0;

RAZ\_Tempo

|  |  |
| --- | --- |
| Name | RAZ\_Tempo |
| Description |  |
| Trigger | WhileTrue of Me.iTempo > 15 |

**Declarations :**

Not Applicable

**Script :**

IF (Me.bCommandeManuArret == 1 OR Me.bCommandeManuMarche == 1 OR Me.iCommandeManuArret <> 0 OR Me.iCommandeManuMarche <> 0)  
AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeArret == 3 AND Me.iTypeCommandeMarche == 3)) THEN  
 Me.bCommandeManuArret = 0;  
 Me.bCommandeManuMarche = 0;  
 Me.iCommandeManuArret = 0;  
 Me.iCommandeManuMarche = 0;  
ENDIF;  
  
Me.iTempo = 0;

wtCommandeArret

|  |  |
| --- | --- |
| Name | wtCommandeArret |
| Description |  |
| Trigger | WhileTrue of Me.bCommandeManuArret == 1 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeArret == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

wtCommandeArretInt

|  |  |
| --- | --- |
| Name | wtCommandeArretInt |
| Description |  |
| Trigger | WhileTrue of Me.iCommandeManuArret <> 0 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeArret == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

wtCommandeMarche

|  |  |
| --- | --- |
| Name | wtCommandeMarche |
| Description |  |
| Trigger | WhileTrue of Me.bCommandeManuMarche == 1 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeMarche == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

wtCommandeMarcheInt

|  |  |
| --- | --- |
| Name | wtCommandeMarcheInt |
| Description |  |
| Trigger | WhileTrue of Me.iCommandeManuMarche <> 0 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeMarche == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

#### Template ArchestrA $dwGenFlamme

##### Description

Not Applicable

##### Derived from

$dwGenAnimationsBool

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenFlecheBool

##### Description

Not Applicable

##### Derived from

$dwGenAnimationsBool

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenFondVibrant

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenGroupeElectrogene

##### Description

Not Applicable

##### Derived from

$dwGenAnimationsBool

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenHUM

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenHydratation

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenKlaxonInt

##### Description

Not Applicable

##### Derived from

$dwGenAnimationsInt

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenManchette

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenMelangeur

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenMES

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenMessages

##### Description

Not Applicable

##### Derived from

$dwGenAnimationsBool

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| strInfo | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenMoteur

##### Description

Not Applicable

##### Derived from

$dwGenActionneur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bInitDefRotation | boolean |  |  |  |  |  |
| bInitDoneRot | boolean |  |  |  |  |  |
| bReInitRot | boolean |  |  |  |  |  |
| bStatusDemarrage | boolean |  |  |  |  |  |
| bStatusMarche | boolean |  |  |  |  |  |
| bStatusVidange | boolean |  |  |  |  |  |
| iTempoRot | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

CPTForcages

|  |  |
| --- | --- |
| Name | CPTForcages |
| Description |  |
| Trigger | DataChange of Me.bCommandeInhibitionDefRot |

**Declarations :**

Not Applicable

**Script :**

Dim indVal as indirect;  
Dim sInputSource as string;  
  
sInputSource = "MyArea.Area";  
indVal.BindTo(sInputSource);  
  
sInputSource = indVal + ".iNbForcages";  
indVal.BindTo(sInputSource);  
  
IF Me.bInitDefRotation == 0 THEN  
 If Me.bCommandeInhibitionDefRot == 1 THEN  
 indVal = indVal + 1;  
 ENDIF;  
 Me.bInitDefRotation = 1;  
  
ELSEIF Me.bInitDefRotation == 1 THEN  
 IF Me.bCommandeInhibitionDefRot == 1 THEN  
 indVal = indVal + 1;  
 ELSEIF Me.bCommandeInhibitionDefRot == 0 THEN  
 indVal = indVal - 1;  
 ENDIF;  
  
ENDIF;

#### Template ArchestrA $dwGenMoteurDirect1S

##### Description

Not Applicable

##### Derived from

$dwGenMoteur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bCommandeInhibitionDefRot\_FB | boolean |  |  |  |  |  |
| bCommandeInhibitionFB | boolean |  |  |  |  |  |
| iPlcStatusMarche\_FB | integer |  |  |  |  |  |
| iStatusLocal\_FB | integer |  |  |  |  |  |
| iTypeCommandeArret | integer |  |  |  |  |  |
| iTypeCommandeMarche | integer |  |  |  |  |  |
| iValCommandeArret | integer |  |  |  |  |  |
| iValCommandeMarche | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcStatusMarche

|  |  |
| --- | --- |
| Name | dcStatusMarche |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusMarche + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 16 Avril 2008  
' Ce script traitera les status groupés pour marche/defaut  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusMarche");  
 logmessage("iPlcStatusMarche : " + Me.iPlcStatusMarche);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusMarche = 0;  
Me.bStatusEnDefaut = 0;  
Me.bStatusVidange = 0;  
Me.bStatusDemarrage = 0;  
  
IF Me.iTypeAnimation == 10 AND Me.iTypeAnimationAlternative == 0 THEN ' PFEIFFER - BR8  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche == 2) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ELSEIF (Me.iPlcStatusMarche == 3) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche == 4) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 ELSEIF (Me.iPlcStatusMarche == 5) THEN  
 ' Vidange - jaune clignotant  
 Me.bStatusVidange = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarche == )  
  
ELSEIF Me.iTypeAnimation == 10 AND Me.iTypeAnimationAlternative == 1 THEN ' VEN4 & VEN5  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ENDIF;  
  
ELSEIF (Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 0) ' BR7  
OR (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 0) ' TSFx  
THEN  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 ELSEIF (Me.iPlcStatusMarche == 2) OR (Me.iPlcStatusMarche == 3) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ELSEIF (Me.iPlcStatusMarche == 4) OR (Me.iPlcStatusMarche == 5) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche >= 6) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarche == )  
  
ELSEIF Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 1 THEN ' BR7\_VEN1  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche == 2) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ELSEIF (Me.iPlcStatusMarche == 3) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche >= 4) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarche == )  
  
ELSEIF Me.iTypeAnimation == 30 THEN ' HGS  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche == 2) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ELSEIF (Me.iPlcStatusMarche == 3) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche == 4) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 ELSEIF (Me.iPlcStatusMarche == 5) THEN  
 ' Vidange - jaune clignotant  
 Me.bStatusVidange = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarche == )  
  
ELSEIF (Me.iTypeAnimation == 50) OR (Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 0)  
OR (Me.iTypeAnimation == 40 AND Me.iTypeAnimationAlternative == 0) ' BR6 OR COM OR BRO  
OR (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 0) ' CAC/FMT  
OR (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 1) ' FR  
THEN   
 IF (Me.iPlcStatusMarche == 1) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 2) OR (Me.iPlcStatusMarche == 3) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ELSEIF (Me.iPlcStatusMarche >= 4) AND (Me.iPlcStatusMarche < 8) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 ELSEIF (Me.iPlcStatusMarche >= 8) AND (Me.iPlcStatusMarche < 16) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche >= 16) AND (Me.iPlcStatusMarche < 32) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche >= 32) THEN  
 ' Vidange - jaune clignotant  
 Me.bStatusVidange = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarche == )  
  
ELSEIF (Me.iTypeAnimation == 60 AND Me.iTypeAnimationAlternative == 1) ' BRO\_BROYEUR6   
OR (Me.iTypeAnimation == 40 AND Me.iTypeAnimationAlternative == 1) ' COM\_ALIM\_C7, COM\_ALIM\_M3  
OR (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 1) ' CAC\_CRIBLE2, CAC\_CRIBLE3, CAC\_CRIBLE5, CAC\_CRIBLE6   
OR (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 2) ' TSFS\_ALx  
THEN  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarche == )  
  
ELSEIF (Me.iTypeAnimation == 80 AND Me.iTypeAnimationAlternative == 1) THEN ' TSFx\_ECL\_x, TSFx\_VEN\_x, TSFx\_VIS\_x  
 IF Me.iPlcStatusMarche == 0 THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF Me.iPlcStatusMarche == 1 THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ELSEIF Me.iPlcStatusMarche == 2 THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ELSEIF Me.iPlcStatusMarche >= 3 THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 ENDIF;  
  
ELSEIF Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 0 THEN ' FR  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ELSEIF (Me.iPlcStatusMarche == 2) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 ELSEIF (Me.iPlcStatusMarche == 3) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche == 4) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche >= 5) THEN  
 ' Vidange - jaune clignotant  
 Me.bStatusVidange = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarche == )  
  
ELSEIF Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 2 THEN ' FR  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 ENDIF;  
  
ELSEIF Me.iTypeAnimation == 100 THEN ' EAU  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ELSEIF (Me.iPlcStatusMarche == 2) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ENDIF;  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN

dcRAZCmd

|  |  |
| --- | --- |
| Name | dcRAZCmd |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusMarche |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Juillet 2008  
' Ce script remettra le bit de commande à zéro pour certaines applications  
' iTypeAnimation : 50 = BR6  
  
IF (Me.bCommandeArretManu == 1 OR Me.bCommandeMarcheManu == 1 OR Me.iCommandeArretManu <> 0 OR Me.iCommandeMarcheManu <> 0)   
AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeArret == 3 AND Me.iTypeCommandeMarche == 3)) THEN  
 Me.bCommandeArretManu = 0;  
 Me.bCommandeMarcheManu = 0;  
 Me.iCommandeArretManu = 0;  
 Me.iCommandeMarcheManu = 0;  
ENDIF;  
  
Me.iTempo = 0;

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenMoteurDirect1S  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenMoteurDirect1S  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatusMarche.Input.InputSource = "Me.iPlcStatusMarche\_FB";  
Me.iStatusLocal.Input.InputSource = "Me.iStatusLocal\_FB";  
Me.bCommandeInhibitionDefRot.Input.InputSource = "Me.bCommandeInhibitionDefRot\_FB";

RAZ\_Tempo

|  |  |
| --- | --- |
| Name | RAZ\_Tempo |
| Description |  |
| Trigger | WhileTrue of Me.iTempo > 15 |

**Declarations :**

Not Applicable

**Script :**

IF (Me.bCommandeArretManu == 1 OR Me.bCommandeMarcheManu == 1 OR Me.iCommandeArretManu <> 0 OR Me.iCommandeMarcheManu <> 0)   
AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeArret == 3 AND Me.iTypeCommandeMarche == 3)) THEN  
 Me.bCommandeArretManu = 0;  
 Me.bCommandeMarcheManu = 0;  
 Me.iCommandeArretManu = 0;  
 Me.iCommandeMarcheManu = 0;  
ENDIF;  
  
Me.iTempo = 0;

wtCommandeArret

|  |  |
| --- | --- |
| Name | wtCommandeArret |
| Description |  |
| Trigger | WhileTrue of Me.bCommandeArretManu == 1 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeArret == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

wtCommandeMarche

|  |  |
| --- | --- |
| Name | wtCommandeMarche |
| Description |  |
| Trigger | WhileTrue of Me.bCommandeMarcheManu == 1 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeMarche == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

wtCommandeArretInt

|  |  |
| --- | --- |
| Name | wtCommandeArretInt |
| Description |  |
| Trigger | WhileTrue of Me.iCommandeArretManu <> 0 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeArret == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

wtCommandeMarcheInt

|  |  |
| --- | --- |
| Name | wtCommandeMarcheInt |
| Description |  |
| Trigger | WhileTrue of Me.iCommandeMarcheManu <> 0 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 OR (Me.iTypeAnimation == 90 AND Me.iTypeCommandeMarche == 3) THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

#### Template ArchestrA $dwGenMoteurDirect2S

##### Description

Not Applicable

##### Derived from

$dwGenMoteur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bCommandeInhibitionDefRot\_FB | boolean |  |  |  |  |  |
| bEnMarche | boolean |  |  |  |  |  |
| bInitDefRotation2 | boolean |  |  |  |  |  |
| bInversion | boolean |  |  |  |  |  |
| bStatusDemarrageArriere | boolean |  |  |  |  |  |
| bStatusDemarrageAvant | boolean |  |  |  |  |  |
| bStatusEnDefautArriere | boolean |  |  |  |  |  |
| bStatusEnDefautAvant | boolean |  |  |  |  |  |
| bStatusMarcheArriere | boolean |  |  |  |  |  |
| bStatusMarcheAvant | boolean |  |  |  |  |  |
| bStatusVidangeArriere | boolean |  |  |  |  |  |
| bStatusVidangeAvant | boolean |  |  |  |  |  |
| iPlcStatusMarche\_FB | integer |  |  |  |  |  |
| iPlcStatusMarcheArriere\_FB | integer |  |  |  |  |  |
| iPlcStatusMarcheAvant\_FB | integer |  |  |  |  |  |
| iStatusLocal\_FB | integer |  |  |  |  |  |
| iTypeCommandeArret | integer |  |  |  |  |  |
| iTypeCommandeDroite | integer |  |  |  |  |  |
| iTypeCommandeGauche | integer |  |  |  |  |  |
| iTypeCommandeMarche | integer |  |  |  |  |  |
| iTypeCommandeMarcheAr | integer |  |  |  |  |  |
| iTypeCommandeMarcheAv | integer |  |  |  |  |  |
| iValCommandeArret | integer |  |  |  |  |  |
| iValCommandeDroite | integer |  |  |  |  |  |
| iValCommandeGauche | integer |  |  |  |  |  |
| iValCommandeMarche | integer |  |  |  |  |  |
| iValCommandeMarcheAr | integer |  |  |  |  |  |
| iValCommandeMarcheAv | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |
| sTexteDroite | string |  |  |  |  |  |
| sTexteGauche | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcStatusMarcheAvant

|  |  |
| --- | --- |
| Name | dcStatusMarcheAvant |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusMarcheAvant + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 17 Mars 2008  
' Ce script traitera les status groupés pour marche Avant/defaut  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusMarcheAvant");  
 logmessage("iPlcStatusMarcheAvant : " + Me.iPlcStatusMarcheAvant);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusMarcheAvant = 0;  
Me.bStatusEnDefautAvant = 0;  
Me.bStatusVidangeAvant = 0;  
Me.bStatusDemarrageAvant = 0;  
Me.bEnMarche = 0;  
  
IF Me.iTypeAnimation == 10 THEN ' PFEIFFER - BR8  
 IF (Me.iPlcStatusMarcheAvant == 0) THEN  
 ' Arrêt sans défaut  
 Me.bStatusMarcheAvant = 0;  
 Me.bStatusEnDefautAvant = 0;  
 Me.bStatusVidangeAvant = 0;  
 Me.bStatusDemarrageAvant = 0;  
 ELSEIF (Me.iPlcStatusMarcheAvant == 1) THEN  
 ' Arrêt avec défaut  
 Me.bStatusEnDefautAvant = 1;  
 ELSEIF (Me.iPlcStatusMarcheAvant == 2) THEN  
 ' Marche sans défaut  
 Me.bStatusMarcheAvant = 1;  
 Me.bEnMarche = 1;  
 ELSEIF (Me.iPlcStatusMarcheAvant == 3) THEN  
 ' Marche avec défaut  
 Me.bStatusMarcheAvant = 1;  
 Me.bStatusEnDefautAvant = 1;  
 Me.bEnMarche = 1;  
 ELSEIF (Me.iPlcStatusMarcheAvant == 4) THEN  
 ' Démarrage  
 Me.bStatusDemarrageAvant = 1;  
 Me.bEnMarche = 1;  
 ELSEIF (Me.iPlcStatusMarcheAvant == 5) THEN  
 ' Vidange  
 Me.bStatusVidangeAvant = 1;  
 Me.bEnMarche = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarcheAvant == )  
  
ELSEIF Me.iTypeAnimation == 40 OR Me.iTypeAnimation == 60 and Me.iTypeAnimationAlternative== 0 THEN ' BRO OR COM  
 IF (Me.iPlcStatusMarcheAvant == 0) THEN  
 ' Arrêt sans défaut  
 Me.bStatusMarcheAvant = 0;  
 Me.bStatusEnDefautAvant = 0;  
 Me.bStatusVidangeAvant = 0;  
 Me.bStatusDemarrageAvant = 0;  
 Me.bEnMarche = 0;  
 ELSEIF (Me.iPlcStatusMarcheAvant == 1) THEN  
 ' Marche sans défaut  
 Me.bStatusMarcheAvant = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarcheAvant == )  
  
ELSEIF Me.iTypeAnimation == 20 THEN ' BR7  
 IF (Me.iPlcStatusMarcheAvant == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarcheAvant = 0;  
 Me.bStatusEnDefautAvant = 0;  
 Me.bStatusVidangeAvant = 0;  
 Me.bStatusDemarrageAvant = 0;  
  
 ELSEIF (Me.iPlcStatusMarcheAvant == 1) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrageAvant = 1;  
 Me.bEnMarche = 1;  
  
 ELSEIF (Me.iPlcStatusMarcheAvant == 2) OR (Me.iPlcStatusMarcheAvant == 3) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarcheAvant = 1;  
 Me.bEnMarche = 1;  
  
 ELSEIF (Me.iPlcStatusMarcheAvant == 4) OR (Me.iPlcStatusMarcheAvant == 5) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefautAvant = 1;  
  
 ELSEIF (Me.iPlcStatusMarcheAvant >= 6) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarcheAvant = 1;  
 Me.bStatusEnDefautAvant = 1;  
 Me.bEnMarche = 1;  
 ENDIF;  
  
ELSEIF Me.iTypeAnimation == 30 OR Me.iTypeAnimation == 70 THEN ' HGS - CAC/FMT  
 IF (Me.iPlcStatusMarcheAvant == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarcheAvant = 0;  
 Me.bStatusEnDefautAvant = 0;  
 Me.bStatusVidangeAvant = 0;  
 Me.bStatusDemarrageAvant = 0;  
 Me.bEnMarche = 0;  
  
 ELSEIF (Me.iPlcStatusMarcheAvant == 1) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarcheAvant = 1;  
 ENDIF;  
  
ELSEIF ( Me.iTypeAnimation == 80 and Me.iTypeAnimationAlternative == 0 ) THEN 'TSFx  
 IF (Me.iPlcStatusMarcheAvant == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarcheAvant = 0;  
 Me.bStatusEnDefautAvant = 0;  
 Me.bStatusVidangeAvant = 0;  
 Me.bStatusDemarrageAvant = 0;  
 Me.bEnMarche = 0;  
 ELSEIF (Me.iPlcStatusMarcheAvant == 1) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarcheAvant = 1;  
 Me.bEnMarche = 0;  
 ELSEIF (Me.iPlcStatusMarcheAvant == 2) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrageAvant = 1;  
 Me.bEnMarche = 1;  
 ENDIF;  
  
  
ELSEIF( Me.iTypeAnimation == 80 and Me.iTypeAnimationAlternative == 1 ) THEN 'TSFx  
 IF (Me.iPlcStatusMarcheAvant == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarcheAvant = 0;  
 Me.bStatusEnDefautAvant = 0;  
 Me.bStatusVidangeAvant = 0;  
 Me.bStatusDemarrageAvant = 0;  
 Me.bEnMarche = 0;  
 ELSEIF (Me.iPlcStatusMarcheAvant == 2) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarcheAvant = 1;  
 Me.bEnMarche = 0;  
 ELSEIF (Me.iPlcStatusMarcheAvant == 1) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrageAvant = 1;  
 Me.bEnMarche = 1;  
 ENDIF;  
  
ELSEIF Me.iTypeAnimation == 90 THEN  
 IF (Me.iPlcStatusMarcheAvant == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarcheAvant = 0;  
 Me.bStatusEnDefautAvant = 0;  
 Me.bStatusVidangeAvant = 0;  
 Me.bStatusDemarrageAvant = 0;  
 ELSEIF (Me.iPlcStatusMarcheAvant == 1) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarcheAvant = 1;  
 Me.bEnMarche = 1;  
 ELSEIF (Me.iPlcStatusMarcheAvant == 2) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrageAvant = 1;  
 Me.bEnMarche = 1;  
 ELSEIF (Me.iPlcStatusMarcheAvant == 3) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefautAvant = 1;  
 ELSEIF (Me.iPlcStatusMarcheAvant == 4) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarcheAvant = 1;  
 Me.bStatusEnDefautAvant = 1;  
 Me.bEnMarche = 1;  
 ENDIF;  
  
  
ELSEIF Me.iTypeAnimation == 60 and Me.iTypeAnimationAlternative== 1 THEN 'COM2  
 IF (Me.iPlcStatusMarcheAvant == 0) THEN  
 ' Arrêt sans défaut  
 Me.bStatusMarcheAvant = 0;  
 Me.bStatusEnDefautAvant = 0;  
 Me.bStatusVidangeAvant = 0;  
 Me.bStatusDemarrageAvant = 0;  
 Me.bEnMarche = 0;  
 ELSEIF (Me.iPlcStatusMarcheAvant == 1) THEN  
 ' Marche sans défaut  
 Me.bStatusMarcheAvant = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarcheAvant == )  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN  
  
  
  
  
  
IF Me.iTypeAnimation == 10 OR Me.iTypeAnimation == 20 OR Me.iTypeAnimation == 80 THEN ' BR8  
' Quand ce script est déclenché, cela signifie que le bit d'arrêt doit être redirigé vers l'avant  
 IF Me.bStatusMarcheAvant == 1 OR Me.bStatusVidangeAvant == 1 OR Me.bStatusDemarrageAvant == 1 THEN  
 IF Me.bInversion == 1 THEN  
 Me.bCommandeArretManu.Input.InputSource = Me.bCommandeMarcheArManu.Input.InputSource;  
 Me.bCommandeArretManu.Output.OutputDest = Me.bCommandeMarcheArManu.Output.OutputDest;   
 ELSE  
 Me.bCommandeArretManu.Input.InputSource = Me.bCommandeMarcheAvManu.Input.InputSource;  
 Me.bCommandeArretManu.Output.OutputDest = Me.bCommandeMarcheAvManu.Output.OutputDest;  
 ENDIF;  
 ENDIF;  
ENDIF;

dcStatusMarcheArriere

|  |  |
| --- | --- |
| Name | dcStatusMarcheArriere |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusMarcheArriere + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 17 Mars 2008  
' Ce script traitera les status groupés pour marche arriere/defaut  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusMarcheArriere");  
 logmessage("iPlcStatusMarcheArriere : " + Me.iPlcStatusMarcheArriere);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusMarcheArriere = 0;  
Me.bStatusEnDefautArriere = 0;  
Me.bStatusVidangeArriere = 0;  
Me.bStatusDemarrageArriere = 0;  
Me.bEnMarche = 0;  
  
IF Me.iTypeAnimation == 10 THEN ' PFEIFFER - BR8  
 IF (Me.iPlcStatusMarcheArriere == 0) THEN  
 ' Arrêt sans défaut  
 Me.bStatusMarcheArriere = 0;  
 Me.bStatusEnDefautArriere = 0;  
 Me.bStatusVidangeArriere = 0;  
 Me.bStatusDemarrageArriere = 0;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 1) THEN  
 ' Arrêt avec défaut  
 Me.bStatusEnDefautArriere = 1;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 2) THEN  
 ' Marche sans défaut  
 Me.bStatusMarcheArriere = 1;  
 Me.bEnMarche = 1;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 3) THEN  
 ' Marche avec défaut  
 Me.bStatusMarcheArriere = 1;  
 Me.bStatusEnDefautArriere = 1;  
 Me.bEnMarche = 1;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 4) THEN  
 ' Démarrage  
 Me.bStatusDemarrageArriere = 1;  
 Me.bEnMarche = 1;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 5) THEN  
 ' Vidange  
 Me.bStatusVidangeArriere = 1;  
 Me.bEnMarche = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarcheArriere == )  
  
ELSEIF Me.iTypeAnimation == 40 THEN ' BRO  
 IF (Me.iPlcStatusMarcheArriere == 0) THEN  
 ' Arrêt sans défaut  
 Me.bStatusMarcheArriere = 0;  
 Me.bStatusEnDefautArriere = 0;  
 Me.bStatusVidangeArriere = 0;  
 Me.bStatusDemarrageArriere = 0;  
 Me.bEnMarche = 0;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 1) THEN  
 ' Marche sans défaut  
 Me.bStatusMarcheArriere = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarcheArriere == )  
  
ELSEIF Me.iTypeAnimation == 60 and Me.iTypeAnimationAlternative== 0 THEN ' COM  
 IF (Me.iPlcStatusMarcheArriere == 1) THEN  
 ' Arrêt sans défaut  
 Me.bStatusMarcheArriere = 0;  
 Me.bStatusEnDefautArriere = 0;  
 Me.bStatusVidangeArriere = 0;  
 Me.bStatusDemarrageArriere = 0;  
 Me.bEnMarche = 0;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 0) THEN  
 ' Marche sans défaut  
 Me.bStatusMarcheArriere = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarcheArriere == )  
  
ELSEIF Me.iTypeAnimation == 20 THEN ' BR7  
 IF (Me.iPlcStatusMarcheArriere == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarcheArriere = 0;  
 Me.bStatusEnDefautArriere = 0;  
 Me.bStatusVidangeArriere = 0;  
 Me.bStatusDemarrageArriere = 0;  
  
 ELSEIF (Me.iPlcStatusMarcheArriere == 1) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrageArriere = 1;  
 Me.bEnMarche = 1;  
  
 ELSEIF (Me.iPlcStatusMarcheArriere == 2) OR (Me.iPlcStatusMarcheArriere == 3) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarcheArriere = 1;  
 Me.bEnMarche = 1;  
  
 ELSEIF (Me.iPlcStatusMarcheArriere == 4) OR (Me.iPlcStatusMarcheArriere == 5) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefautArriere = 1;  
  
 ELSEIF (Me.iPlcStatusMarcheArriere >= 6) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarcheArriere = 1;  
 Me.bStatusEnDefautArriere = 1;  
 Me.bEnMarche = 1;  
 ENDIF;  
  
ELSEIF Me.iTypeAnimation == 30 OR Me.iTypeAnimation == 70 THEN ' HGS - CAC/FMT  
 IF (Me.iPlcStatusMarcheArriere == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarcheArriere = 0;  
 Me.bStatusEnDefautArriere = 0;  
 Me.bStatusVidangeArriere = 0;  
 Me.bStatusDemarrageArriere = 0;  
 Me.bEnMarche = 0;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 1) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarcheArriere = 1;  
 ENDIF;  
  
ELSEIF ( Me.iTypeAnimation == 80 and Me.iTypeAnimationAlternative == 0 ) THEN 'TSFx  
 IF (Me.iPlcStatusMarcheArriere == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarcheArriere = 0;  
 Me.bStatusEnDefautArriere = 0;  
 Me.bStatusVidangeArriere = 0;  
 Me.bStatusDemarrageArriere = 0;  
 Me.bEnMarche = 0;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 1) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarcheArriere = 1;  
 Me.bEnMarche = 0;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 2) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrageArriere = 1;  
 Me.bEnMarche = 1;  
 ENDIF;  
  
   
ELSEIF (Me.iTypeAnimation == 80 and Me.iTypeAnimationAlternative== 1 ) THEN 'TSFx  
 IF (Me.iPlcStatusMarcheArriere == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarcheArriere = 0;  
 Me.bStatusEnDefautArriere = 0;  
 Me.bStatusVidangeArriere = 0;  
 Me.bStatusDemarrageArriere = 0;  
 Me.bEnMarche = 0;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 2) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarcheArriere = 1;  
 Me.bEnMarche = 0;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 1) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrageArriere = 1;  
 Me.bEnMarche = 1;  
 ENDIF;  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
ELSEIF Me.iTypeAnimation == 90 THEN  
 IF (Me.iPlcStatusMarcheArriere == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarcheArriere = 0;  
 Me.bStatusEnDefautArriere = 0;  
 Me.bStatusVidangeArriere = 0;  
 Me.bStatusDemarrageArriere = 0;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 1) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarcheArriere = 1;  
 Me.bEnMarche = 1;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 2) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrageArriere = 1;  
 Me.bEnMarche = 1;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 3) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefautArriere = 1;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 4) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarcheArriere = 1;  
 Me.bStatusEnDefautArriere = 1;  
 Me.bEnMarche = 1;  
 ENDIF;   
  
  
ELSEIF Me.iTypeAnimation == 60 and Me.iTypeAnimationAlternative== 1 THEN 'COM2  
 IF (Me.iPlcStatusMarcheArriere == 0) THEN  
 ' Arrêt sans défaut  
 Me.bStatusMarcheArriere = 0;  
 Me.bStatusEnDefautArriere = 0;  
 Me.bStatusVidangeArriere = 0;  
 Me.bStatusDemarrageArriere = 0;  
 Me.bEnMarche = 0;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 1) THEN  
 ' Marche sans défaut  
 Me.bStatusMarcheArriere = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarcheArriere == )  
  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN  
  
  
  
  
IF Me.iTypeAnimation == 10 OR Me.iTypeAnimation == 20 OR Me.iTypeAnimation == 80 THEN ' BR8  
' Quand ce script est déclenché, cela signifie que le bit d'arrêt doit être redirigé vers l'arrière  
 IF Me.bStatusMarcheArriere == 1 OR Me.bStatusVidangeArriere == 1 OR Me.bStatusDemarrageArriere == 1 THEN  
 IF Me.bInversion == 1 THEN  
 Me.bCommandeArretManu.Input.InputSource = Me.bCommandeMarcheAvManu.Input.InputSource;  
 Me.bCommandeArretManu.Output.OutputDest = Me.bCommandeMarcheAvManu.Output.OutputDest;  
 ELSE  
 Me.bCommandeArretManu.Input.InputSource = Me.bCommandeMarcheArManu.Input.InputSource;  
 Me.bCommandeArretManu.Output.OutputDest = Me.bCommandeMarcheArManu.Output.OutputDest;  
 ENDIF;  
 ENDIF;  
ENDIF;

dcStatusMarche

|  |  |
| --- | --- |
| Name | dcStatusMarche |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusMarche + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 6 Juillet 2008  
' Ce script traitera les status groupés pour marche (en cas de moteur 2s avec animation unique)  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusMarche");  
 logmessage("iPlcStatusMarche : " + Me.iPlcStatusMarche);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusMarche = 0;  
Me.bStatusEnDefaut = 0;  
Me.bStatusVidange = 0;  
Me.bStatusDemarrage = 0;  
Me.bEnMarche = 0;  
  
IF Me.iTypeAnimation == 40 OR Me.iTypeAnimation == 60   
OR (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 1)  
THEN ' BRO OR COM  
 IF (Me.iPlcStatusMarche == 1) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
   
 ELSEIF (Me.iPlcStatusMarche == 2) OR (Me.iPlcStatusMarche == 3) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 Me.bEnMarche = 1;  
  
 ELSEIF (Me.iPlcStatusMarche >= 4) AND (Me.iPlcStatusMarche < 8) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 Me.bEnMarche = 1;  
  
 ELSEIF (Me.iPlcStatusMarche >= 8) AND (Me.iPlcStatusMarche < 16) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
  
 ELSEIF (Me.iPlcStatusMarche >= 16) AND (Me.iPlcStatusMarche < 32) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 Me.bEnMarche = 1;  
  
 ELSEIF (Me.iPlcStatusMarche >= 32) THEN  
 ' Vidange - jaune clignotant  
 Me.bStatusVidange = 1;  
 Me.bEnMarche = 1;  
  
 ENDIF; ' IF (Me.iPlcStatusMarche == )  
  
ELSEIF Me.iTypeAnimation == 20 THEN ' BR7  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 Me.bEnMarche = 1;  
  
 ELSEIF (Me.iPlcStatusMarche == 2) OR (Me.iPlcStatusMarche == 3) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 Me.bEnMarche = 1;  
  
 ELSEIF (Me.iPlcStatusMarche == 4) OR (Me.iPlcStatusMarche == 5) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
  
 ELSEIF (Me.iPlcStatusMarche >= 6) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 Me.bEnMarche = 1;  
 ENDIF;  
  
ELSEIF Me.iTypeAnimation == 30 THEN ' HGS  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
  
 ELSEIF (Me.iPlcStatusMarche == 2) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 Me.bEnMarche = 1;  
  
 ELSEIF (Me.iPlcStatusMarche == 3) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 Me.bEnMarche = 1;  
  
 ELSEIF (Me.iPlcStatusMarche == 4) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 Me.bEnMarche = 1;  
  
 ELSEIF (Me.iPlcStatusMarche == 5) THEN  
 ' Vidange - jaune clignotant  
 Me.bStatusVidange = 1;  
 Me.bEnMarche = 1;  
 ENDIF;  
  
ELSEIF Me.iTypeAnimation == 70 THEN 'CAC/FMT  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 Me.bEnMarche = 1;  
 ENDIF;  
  
ELSEIF Me.iTypeAnimation == 80 THEN 'TSFx  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 Me.bEnMarche = 1;  
 ELSEIF (Me.iPlcStatusMarche == 2) OR (Me.iPlcStatusMarche == 3) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 Me.bEnMarche = 1;  
 ELSEIF (Me.iPlcStatusMarche == 4) OR (Me.iPlcStatusMarche == 5) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche >= 6) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 Me.bEnMarche = 1;  
 ENDIF;   
  
ELSEIF Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative <> 1 THEN  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 Me.bEnMarche = 1;  
 ELSEIF (Me.iPlcStatusMarche == 2) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 Me.bEnMarche = 1;  
 ELSEIF (Me.iPlcStatusMarche == 3) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche == 4) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 Me.bEnMarche = 1;  
 ENDIF;   
  
ENDIF;

CPTForcages\_2

|  |  |
| --- | --- |
| Name | CPTForcages\_2 |
| Description |  |
| Trigger | DataChange of Me.bCommandeInhibitionDefRot\_02 |

**Declarations :**

Not Applicable

**Script :**

Dim indVal as indirect;  
Dim sInputSource as string;  
  
sInputSource = "MyArea.Area";  
indVal.BindTo(sInputSource);  
  
sInputSource = indVal + ".iNbForcages";  
indVal.BindTo(sInputSource);  
  
IF Me.bInitDefRotation2 == 0 THEN  
 If Me.bCommandeInhibitionDefRot\_02 == 1 THEN  
 indVal = indVal + 1;  
 ENDIF;  
 Me.bInitDefRotation = 1;  
  
ELSEIF Me.bInitDefRotation2 == 1 THEN  
 IF Me.bCommandeInhibitionDefRot\_02 == 1 THEN  
 indVal = indVal + 1;  
 ELSEIF Me.bCommandeInhibitionDefRot\_02 == 0 THEN  
 indVal = indVal - 1;  
 ENDIF;  
  
ENDIF;

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenMoteurDirect2S  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenMoteurDirect2S  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatusMarche.Input.InputSource = "Me.iPlcStatusMarche\_FB";  
Me.iPlcStatusMarcheAvant.Input.InputSource = "Me.iPlcStatusMarcheAvant\_FB";  
Me.iPlcStatusMarcheArriere.Input.InputSource = "Me.iPlcStatusMarcheArriere\_FB";  
Me.iStatusLocal.Input.InputSource = "Me.iStatusLocal\_FB";  
Me.bCommandeInhibitionDefRot.Input.InputSource = "Me.bCommandeInhibitionDefRot\_FB";

wtCommandeMarche

|  |  |
| --- | --- |
| Name | wtCommandeMarche |
| Description |  |
| Trigger | WhileTrue of Me.bCommandeMarche == 1 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

wtCommandeArret

|  |  |
| --- | --- |
| Name | wtCommandeArret |
| Description |  |
| Trigger | WhileTrue of Me.bCommandeArretManu == 1 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

dcRAZCmd

|  |  |
| --- | --- |
| Name | dcRAZCmd |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusMarche |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Juillet 2008  
' Ce script remettra le bit de commande à zéro pour certaines applications  
' iTypeAnimation : 50 = BR6  
  
IF (Me.bCommandeArretManu == 1 OR Me.bCommandeMarche == 1)   
AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70) THEN  
 Me.bCommandeArretManu = 0;  
 Me.bCommandeMarche = 0;  
ENDIF;  
  
Me.iTempo = 0;

wtCommandeDroite

|  |  |
| --- | --- |
| Name | wtCommandeDroite |
| Description |  |
| Trigger | WhileTrue of Me.bCommandeDroite == 1 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

wtCommandeGauche

|  |  |
| --- | --- |
| Name | wtCommandeGauche |
| Description |  |
| Trigger | WhileTrue of Me.bCommandeGauche == 1 |

**Declarations :**

Not Applicable

**Script :**

IF Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70 THEN  
 Me.iTempo = Me.iTempo + 1;  
ENDIF;

dcRAZCmdArriere

|  |  |
| --- | --- |
| Name | dcRAZCmdArriere |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusMarcheArriere |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Juillet 2008  
' Ce script remettra le bit de commande à zéro pour certaines applications  
' iTypeAnimation : 50 = BR6  
  
IF (Me.bCommandeGauche == 1 OR Me.bCommandeDroite == 1)   
AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70) THEN  
 Me.bCommandeGauche = 0;  
 Me.bCommandeDroite = 0;  
ENDIF;  
  
Me.iTempo = 0;

dcRAZCmdAvant

|  |  |
| --- | --- |
| Name | dcRAZCmdAvant |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusMarcheAvant |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Juillet 2008  
' Ce script remettra le bit de commande à zéro pour certaines applications  
' iTypeAnimation : 50 = BR6  
  
IF (Me.bCommandeGauche == 1 OR Me.bCommandeDroite == 1)   
AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70) THEN  
 Me.bCommandeGauche = 0;  
 Me.bCommandeDroite = 0;  
ENDIF;  
  
Me.iTempo = 0;

RAZ\_Tempo

|  |  |
| --- | --- |
| Name | RAZ\_Tempo |
| Description |  |
| Trigger | WhileTrue of Me.iTempo > 15 |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 30 Juillet 2008  
' Ce script remettra le bit de commande à zéro pour certaines applications  
' iTypeAnimation : 50 = BR6  
  
IF (Me.bCommandeArretManu == 1 OR Me.bCommandeMarche == 1)   
AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70) THEN  
 Me.bCommandeArretManu = 0;  
 Me.bCommandeMarche = 0;  
ENDIF;  
  
IF (Me.bCommandeGauche == 1 OR Me.bCommandeDroite == 1)   
AND (Me.iTypeAnimation == 50 OR Me.iTypeAnimation == 70) THEN  
 Me.bCommandeGauche = 0;  
 Me.bCommandeDroite = 0;  
ENDIF;  
  
Me.iTempo = 0;

#### Template ArchestrA $dwGenMouCable

##### Description

Not Applicable

##### Derived from

$dwGenAnimationsBool

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenNBR

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenNIV

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenObjet

##### Description

Not Applicable

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| iTempo | integer |  |  |  |  |  |
| Ref\_Done | boolean |  |  |  |  |  |
| sDescription | string |  |  |  |  |  |
| strArea | string |  |  |  |  |  |
| strName | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

osSecurity

|  |  |
| --- | --- |
| Name | osSecurity |
| Description |  |
| Trigger | WhileTrue of Me.osSecurity.ExecutionCnt <= 2 And Me.Ref\_Done == False |

**Declarations :**

Not Applicable

**Script :**

If Me.osSecurity.ExecutionCnt == 2 Then  
  
 Dim result as integer;  
  
 Me.strName = Me.Tagname;  
  
 result = StringInString(Me.strName, "\_", 1, 0) - 1;  
 Me.strName = StringLeft(Me.strName,result);  
  
 IF stringLeft(Me.strName,2) == "BR" OR Me.strName == "COM" OR Me.strName == "HGS" or Me.strName == "MA01" THEN  
 Me.strArea = "BROYAGE";  
 ELSEIF stringLeft(Me.strName,2) == "CF" OR stringLeft(Me.strName,3) == "FMT" OR stringLeft(Me.strName,2) == "FR" OR stringLeft(Me.strName,3) == "TSF" OR stringLeft(Me.strName,3) == "CAC" THEN  
 Me.strArea = "FOURS";  
 ENDIF;  
  
 Me.Ref\_Done = True;   
Endif;

#### Template ArchestrA $dwGenOverBand

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenPfister

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenPlaqueMagnetique

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenPOIDS

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenPompe

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenPOS

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenPOUS

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenPR

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenPressionAir

##### Description

Not Applicable

##### Derived from

$dwGenAnimationsBool

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenPuissance

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenQTE

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenQTE\_SEUIL

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| SeuilBas | float |  |  |  |  |  |
| SeuilHaut | float |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenRapportHgs

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogInOut

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| fJour | integer |  | X |  |  |  |
| fJour\_Coef | float |  |  |  |  |  |
| fJour\_Prec | integer |  | X |  |  |  |
| fJour\_Prec\_Coef | float |  |  |  |  |  |
| fJour\_Prec\_R | float |  |  |  |  |  |
| fJour\_R | float |  |  |  |  |  |
| fMois | integer |  | X |  |  |  |
| fMois\_Coef | float |  |  |  |  |  |
| fMois\_Prec | integer |  | X |  |  |  |
| fMois\_Prec\_Coef | float |  |  |  |  |  |
| fMois\_Prec\_R | float |  |  |  |  |  |
| fMois\_R | float |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

JOUR

|  |  |
| --- | --- |
| Name | JOUR |
| Description |  |
| Trigger | DataChange of Me.fjour |

**Declarations :**

Not Applicable

**Script :**

Me.fjour\_R = Me.fjour \* Me.fjour\_Coef;

JOUR\_PREC

|  |  |
| --- | --- |
| Name | JOUR\_PREC |
| Description |  |
| Trigger | DataChange of Me.fjour\_Prec |

**Declarations :**

Not Applicable

**Script :**

Me.fjour\_Prec\_R = Me.fjour\_Prec \* Me.fjour\_Prec\_Coef;

MOIS

|  |  |
| --- | --- |
| Name | MOIS |
| Description |  |
| Trigger | DataChange of Me.fMois |

**Declarations :**

Not Applicable

**Script :**

Me.fMois\_R = Me.fMois \* Me.fMois\_Coef;

MOIS\_PREC

|  |  |
| --- | --- |
| Name | MOIS\_PREC |
| Description |  |
| Trigger | DataChange of Me.fMois\_Prec |

**Declarations :**

Not Applicable

**Script :**

Me.fMois\_Prec\_R = Me.fMois\_Prec \* Me.fMois\_Prec\_Coef;

#### Template ArchestrA $dwGenRapports

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogInOut

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bMoisChange | boolean |  |  |  |  |  |
| fJourCourant | float |  |  |  |  | X |
| fJourCourant6AM | float |  |  |  |  | X |
| fJourPrecedent | float |  |  |  |  | X |
| fJourPrecedent6AM | float |  |  |  |  | X |
| fJourPrecTemp | float |  |  |  |  | X |
| fMoisCourant | float |  |  |  |  | X |
| fMoisCourant6AM | float |  |  |  |  | X |
| fMoisPrecedent | float |  |  |  |  | X |
| fMoisPrecedent6AM | float |  |  |  |  | X |
| fTotalMois | float |  |  |  |  | X |
| iIndexMois | integer |  |  |  |  |  |
| iTypeAnimation | integer |  | X |  |  |  |
| iTypeAnimationAlternative | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

SetJourPrec

|  |  |
| --- | --- |
| Name | SetJourPrec |
| Description |  |
| Trigger | OnTrue of Now().Hour == 6 AND Now().Minute == 0 |

**Declarations :**

Not Applicable

**Script :**

'script that calculates the values for Jour Precedent et Mois Precedent  
  
IF Me.iTypeAnimation == 30 THEN ' HGS  
  
 Me.fJourPrecedent6AM = Me.fJourPrecedent;  
  
 Me.fJourCourant6AM = Me.fJourCourant;   
 Me.fJourPrecTemp = Me.fJourCourant6AM;  
  
 Me.fTotalMois = Me.fTotalMois + Me.fJourCourant6AM;  
  
 Me.fMoisCourant = Me.fTotalMois;  
 Me.fMoisCourant6AM = Me.fMoisCourant;  
  
 'reset values  
 MyArea.bCommandeRAZRapports = MyArea.bCommandeRAZRapports + 1;  
  
 if Me.bMoisChange == 1 then  
 'reset variable bMoisChange   
   
 logmessage("execute remise zero");  
  
 Me.bMoisChange = 0;  
  
 ' Total du mois qui se termine dans le mois précedent  
 Me.fMoisPrecedent = Me.fTotalMois;  
 Me.fMoisPrecedent6AM = Me.fMoisPrecedent;  
  
 Me.fTotalMois = 0;  
 Me.fMoisCourant = 0;  
   
 endif;  
  
 Me.fJourPrecedent = Me.fJourPrecTemp;  
  
ENDIF;

dcMonth

|  |  |
| --- | --- |
| Name | dcMonth |
| Description |  |
| Trigger | DataChange of Now().Month |

**Declarations :**

Not Applicable

**Script :**

'the month has changed  
  
logmessage("execution dcmonth");  
  
Me.bMoisChange = 1;

dcMinute

|  |  |
| --- | --- |
| Name | dcMinute |
| Description |  |
| Trigger | DataChange of Now().Minute |

**Declarations :**

Not Applicable

**Script :**

'script that calculates the values for Jour Courant et Mois Courant  
  
IF Me.iTypeAnimation == 30 THEN ' HGS  
 Me.fJourCourant = Me.PV;  
 Me.fMoisCourant = Me.fTotalMois + Me.fJourCourant;  
  
  
ENDIF;

#### Template ArchestrA $dwGenReducteur

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenRefrigerant

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenRegistre\_Coulissant

##### Description

Not Applicable

##### Derived from

$dwGenVanne

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bStatusFinCourseFerme | boolean |  |  |  |  |  |
| bStatusFinCourseOuvert | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcplcStatusFCFerme

|  |  |
| --- | --- |
| Name | dcplcStatusFCFerme |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusFCFerme |

**Declarations :**

Not Applicable

**Script :**

IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusFCFerme");  
 logmessage("iPlcStatusFCFerme : " + Me.iPlcStatusFCFerme);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
IF Me.iTypeAnimation == 10 OR Me.iTypeAnimation == 20 OR Me.iTypeAnimation == 30   
OR Me.iTypeAnimation == 50 OR (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 0)  
OR (Me.iTypeAnimation == 90 AND (Me.iTypeAnimationAlternative == 0 OR Me.iTypeAnimationAlternative == 2))  
THEN ' PFEIFFER - BR8, BR7, BR6, HGS, CAC/FMT, FR  
 IF (Me.iPlcStatusFCFerme == 0) THEN  
 Me.bStatusFinCourseFerme = 0;  
 ELSEIF (Me.iPlcStatusFCFerme == 1) THEN  
 Me.bStatusFinCourseFerme = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCFerme == )  
ENDIF;

dcPlcStatusFCOuvert

|  |  |
| --- | --- |
| Name | dcPlcStatusFCOuvert |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusFCOuvert |

**Declarations :**

Not Applicable

**Script :**

IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusFCOuvert");  
 logmessage("iPlcStatusFCOuvert : " + Me.iPlcStatusFCOuvert);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
IF Me.iTypeAnimation == 10 OR Me.iTypeAnimation == 20 OR Me.iTypeAnimation == 30 OR Me.iTypeAnimation == 50   
OR (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 0)   
OR (Me.iTypeAnimation == 90 AND (Me.iTypeAnimationAlternative == 0 OR Me.iTypeAnimationAlternative == 1))  
THEN ' PFEIFFER - BR8, BR7, BR6, HGS, CAC/FMT  
 IF (Me.iPlcStatusFCOuvert == 0) THEN  
 Me.bStatusFinCourseOuvert = 0;  
 ELSEIF (Me.iPlcStatusFCOuvert == 1) THEN  
 Me.bStatusFinCourseOuvert = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCOuvert == )  
  
ENDIF;

dcPlcStatuspositionRegistre

|  |  |
| --- | --- |
| Name | dcPlcStatuspositionRegistre |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusPosition |

**Declarations :**

Not Applicable

**Script :**

Me.iPositionRegistre = Me.iPlcStatusPosition;

#### Template ArchestrA $dwGenRoueASable

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenSas

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenSeparateur

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenSondeGoulotte

##### Description

Not Applicable

##### Derived from

$dwGenAnimationsBool

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenSondeNiveau

##### Description

Not Applicable

##### Derived from

$dwGenAnimationsBool

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenSondeNiveau\_BB

##### Description

Not Applicable

##### Derived from

$dwGenSondeNiveau

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenSondeNiveauInt

##### Description

Not Applicable

##### Derived from

$dwGenAnimationsInt

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenTEMPS

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenTextes

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$String

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| sDescription | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenTransporteur1S

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenTransporteur1SAvecDebit

##### Description

Not Applicable

##### Derived from

$dwGenTransporteur1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenTransporteur2S

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect2S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenTT

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenVanne

##### Description

Not Applicable

##### Derived from

$dwGenActionneur

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenVanne1Input

##### Description

Not Applicable

##### Derived from

$dwGenVanne

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bStatusFinCourseFerme | boolean |  |  |  |  |  |
| bStatusFinCourseOuvert | boolean |  |  |  |  |  |
| bStatusMouvement | boolean |  |  |  |  |  |
| bStatusNoPosition | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenVanne2Input

##### Description

Not Applicable

##### Derived from

$dwGenVanne

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bStatusFinCourseFerme | boolean |  |  |  |  |  |
| bStatusFinCourseOuvert | boolean |  |  |  |  |  |
| iPlcStatusFCFerme\_FB | integer |  |  |  |  |  |
| iPlcStatusFCOuvert\_FB | integer |  |  |  |  |  |
| iStatusLocal\_FB | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |
| sTexteFerme | string |  |  |  |  |  |
| sTexteOuvert | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcStatusFCOuvert

|  |  |
| --- | --- |
| Name | dcPlcStatusFCOuvert |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusFCOuvert + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 16 Avril 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusFCOuvert");  
 logmessage("iPlcStatusFCOuvert : " + Me.iPlcStatusFCOuvert);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
IF Me.iTypeAnimation == 10 OR Me.iTypeAnimation == 20   
OR Me.iTypeAnimation == 30 OR Me.iTypeAnimation == 50   
OR Me.iTypeAnimation == 60   
OR (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 0)   
OR (Me.iTypeAnimation == 90 AND (Me.iTypeAnimationAlternative == 0 OR Me.iTypeAnimationAlternative == 1))  
THEN ' PFEIFFER - BR8, BR7, BR6, HGS, CAC/FMT  
 IF (Me.iPlcStatusFCOuvert == 0) THEN  
 Me.bStatusFinCourseOuvert = 0;  
 ELSEIF (Me.iPlcStatusFCOuvert == 1) THEN  
 Me.bStatusFinCourseOuvert = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCOuvert == )  
  
ELSEIF (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 1) THEN  
 IF Me.iPlcStatusFCOuvert <= 1 THEN  
 Me.bStatusFinCourseOuvert = 0;   
 ELSEIF Me.iPlcStatusFCOuvert >= 2 THEN   
 Me.bStatusFinCourseOuvert = 1;  
 ENDIF;  
  
ELSEIF (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 2) THEN  
 IF (Me.iPlcStatusFCOuvert == 0) THEN  
 Me.bStatusFinCourseOuvert = 1;  
 ELSEIF (Me.iPlcStatusFCOuvert == 1) THEN  
 Me.bStatusFinCourseOuvert = 0;  
 ENDIF; ' IF (Me.iPlcStatusFCOuvert == )  
  
ELSEIF (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 3) THEN  
 IF (Me.iPlcStatusFCOuvert == 1) THEN  
 Me.bStatusFinCourseOuvert = 1;  
 ELSE  
 Me.bStatusFinCourseOuvert = 0;  
 ENDIF; ' IF (Me.iPlcStatusFCOuvert == )  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN

dcPlcStatusFCFerme

|  |  |
| --- | --- |
| Name | dcPlcStatusFCFerme |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusFCFerme + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 16 Avril 2008  
' Ce script traitera les status  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusFCFerme");  
 logmessage("iPlcStatusFCFerme : " + Me.iPlcStatusFCFerme);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
IF Me.iTypeAnimation == 10 OR Me.iTypeAnimation == 20   
OR Me.iTypeAnimation == 30 OR Me.iTypeAnimation == 50   
OR Me.iTypeAnimation == 60   
OR (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 0)  
OR (Me.iTypeAnimation == 90 AND (Me.iTypeAnimationAlternative == 0 OR Me.iTypeAnimationAlternative == 2))  
THEN ' PFEIFFER - BR8, BR7, BR6, HGS, CAC/FMT, FR  
 IF (Me.iPlcStatusFCFerme == 0) THEN  
 Me.bStatusFinCourseFerme = 0;  
 ELSEIF (Me.iPlcStatusFCFerme == 1) THEN  
 Me.bStatusFinCourseFerme = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCFerme == )  
  
ELSEIF (Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 1) THEN  
  
 IF Me.iPlcStatusFCFerme == 0 OR Me.iPlcStatusFCFerme == 2 THEN  
 Me.bStatusFinCourseFerme = 0;   
 ELSEIF Me.iPlcStatusFCFerme == 1 OR Me.iPlcStatusFCFerme >= 3 THEN   
 Me.bStatusFinCourseFerme = 1;  
 ENDIF;  
  
ELSEIF (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 1) THEN  
 IF (Me.iPlcStatusFCFerme == 0) THEN  
 Me.bStatusFinCourseFerme = 1;  
 ELSEIF (Me.iPlcStatusFCFerme == 1) THEN  
 Me.bStatusFinCourseFerme = 0;  
 ENDIF; ' IF (Me.iPlcStatusFCFerme == )  
  
ELSEIF (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 3) THEN  
 IF (Me.iPlcStatusFCFerme == 2) THEN  
 Me.bStatusFinCourseFerme = 1;  
 ELSE  
 Me.bStatusFinCourseFerme = 0;  
 ENDIF; ' IF (Me.iPlcStatusFCFerme == )  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenVanne  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenVanne  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatusFCFerme.Input.InputSource = "Me.iPlcStatusFCFerme\_FB";  
Me.iPlcStatusFCOuvert.Input.InputSource = "Me.iPlcStatusFCOuvert\_FB";  
Me.iStatusLocal.Input.InputSource = "Me.iStatusLocal\_FB";

#### Template ArchestrA $dwGenVanne3Input

##### Description

Not Applicable

##### Derived from

$dwGenVanne

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenVanne4Input

##### Description

Not Applicable

##### Derived from

$dwGenVanne

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenVanne5input

##### Description

Not Applicable

##### Derived from

$dwGenVanne

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenVanneAiguillage

##### Description

Not Applicable

##### Derived from

$dwGenVanne3Input

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bStatusDir | boolean |  |  |  |  |  |
| bStatusIsole | boolean |  |  |  |  |  |
| bStatusOuvert | boolean |  |  |  |  |  |
| iPlcStatusDir\_FB | integer |  |  |  |  |  |
| iPlcStatusIsole\_FB | integer |  |  |  |  |  |
| iPlcStatusOuvert\_FB | integer |  |  |  |  |  |
| iStatusLocal\_FB | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |
| sTexteDirection | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcStatusDir

|  |  |
| --- | --- |
| Name | dcPlcStatusDir |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusDir + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 01 Aout 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusDir");  
 logmessage("iPlcStatusDir: " + Me.iPlcStatusDir);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusDir = 0;  
  
IF Me.iTypeAnimation == 50 THEN ' BR6  
 IF (Me.iPlcStatusDir == 0) THEN  
 Me.bStatusDir = 0; ' vert  
 ELSEIF (Me.iPlcStatusDir == 1) THEN  
 Me.bStatusDir = 1; ' rouge  
 ENDIF; ' IF (Me.iPlcStatusDir == )  
ENDIF;

dcPlcStatusOuvert

|  |  |
| --- | --- |
| Name | dcPlcStatusOuvert |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusOuvert + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 01 Aout 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusOuvert");  
 logmessage("iPlcStatusOuvert: " + Me.iPlcStatusOuvert);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusOuvert = 0;  
  
IF Me.iTypeAnimation == 50 THEN ' BR6  
 IF (Me.iPlcStatusOuvert == 0) THEN  
 Me.bStatusOuvert = 0; ' vert  
 ELSEIF (Me.iPlcStatusOuvert == 1) THEN  
 Me.bStatusOuvert = 1; ' rouge  
 ENDIF; ' IF (Me.iPlcStatusOuvert == )  
ENDIF;

dcPlcStatusIsole

|  |  |
| --- | --- |
| Name | dcPlcStatusIsole |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusIsole + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 01 Aout 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusIsole");  
 logmessage("iPlcStatusIsole: " + Me.iPlcStatusIsole);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusIsole = 0;  
  
IF Me.iTypeAnimation == 50 THEN ' BR6  
 IF (Me.iPlcStatusIsole == 0) THEN  
 Me.bStatusIsole = 1; ' rouge  
 ELSEIF (Me.iPlcStatusIsole == 1) THEN  
 Me.bStatusIsole = 0; ' vert  
 ENDIF; ' IF (Me.iPlcStatusIsole == )  
ENDIF;

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenVanneAiguillage  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenVanneAiguillage  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatusDir.Input.InputSource = "Me.iPlcStatusDir\_FB";  
Me.iPlcStatusIsole.Input.InputSource = "Me.iPlcStatusIsole\_FB";  
Me.iPlcStatusOuvert.Input.InputSource = "Me.iPlcStatusOuvert\_FB";  
Me.iStatusLocal.Input.InputSource = "Me.iStatusLocal\_FB";

#### Template ArchestrA $dwGenVanneGUILLOTINE2S

##### Description

Not Applicable

##### Derived from

$dwGenVanne2Input

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bForceCommandeActive | boolean |  |  |  |  |  |
| bStatusFCDroite | boolean |  |  |  |  |  |
| bStatusFCGauche | boolean |  |  |  |  |  |
| iTypeCommandeDroite | integer |  |  |  |  |  |
| iTypeCommandeGauche | integer |  |  |  |  |  |
| iValCommandeDroite | integer |  |  |  |  |  |
| iValCommandeGauche | integer |  |  |  |  |  |
| sTexteDroite | string |  |  |  |  |  |
| sTexteGauche | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcstatusfcDroite

|  |  |
| --- | --- |
| Name | dcPlcstatusfcDroite |
| Description |  |
| Trigger | WhileTrue of Me.iPlcStatusFCDroite + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

IF (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 0) or (Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 0)   
THEN  
  
 IF (Me.iPlcStatusFCDroite == 0) THEN  
 Me.bStatusFCDroite = 0;  
 ELSEIF (Me.iPlcStatusFCDroite == 1) THEN  
 Me.bStatusFCDroite = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCDroite == )  
  
ENDIF;

dcPlcstatusfcGauche

|  |  |
| --- | --- |
| Name | dcPlcstatusfcGauche |
| Description |  |
| Trigger | WhileTrue of Me.iPlcStatusFCGauche + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

IF (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 0) or (Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 0)   
THEN  
  
 IF (Me.iPlcStatusFCGauche == 0) THEN  
 Me.bStatusFCGauche = 0;  
 ELSEIF (Me.iPlcStatusFCGauche == 1) THEN  
 Me.bStatusFCGauche = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCGauche == )  
  
ENDIF;

#### Template ArchestrA $dwGenVannePapillon

##### Description

Not Applicable

##### Derived from

$dwGenVanne1Input

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bTjsCommandable | boolean |  |  |  |  |  |
| iPlcStatus\_FB | integer |  |  |  |  |  |
| iStatusLocal\_FB | integer |  |  |  |  |  |
| iTypeCommandeFermeture | integer |  |  |  |  |  |
| iTypeCommandeOuverture | integer |  |  |  |  |  |
| iValCommandeFermeture | integer |  |  |  |  |  |
| iValCommandeOuverture | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |
| sTexteFerme | string |  |  |  |  |  |
| sTexteOuvert | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcStatus

|  |  |
| --- | --- |
| Name | dcPlcStatus |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 16 Avril 2008  
' Ce script traitera les status  
' Ferme : vert  
' Ouvert : rouge  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatus");  
 logmessage("iPlcStatus : " + Me.iPlcStatus);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusFinCourseFerme = 0;  
Me.bStatusFinCourseOuvert = 0;  
Me.bStatusEnDefaut = 0;  
  
IF (Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 0) THEN ' BR7  
 IF (Me.iPlcStatus == 0) THEN  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 20 AND Me.iTypeAnimationAlternative == 1) THEN  
 IF (Me.iPlcStatus == 0) THEN  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusEnDefaut = 1;   
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF (Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 1) THEN 'HGS\_V44B  
 IF (Me.iPlcStatus == 0) THEN  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
ELSEIF(Me.iTypeAnimation == 30 AND Me.iTypeAnimationAlternative == 0) THEN ' HGS\_CLAPF  
 IF (Me.iPlcStatus == 0) THEN  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 ENDIF; ' IF (Me.iPlcStatus == )  
  
'bw FR4   
ELSEIF (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 0) THEN  
 IF (Me.iPlcStatus == 0) THEN  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusEnDefaut = 0;   
 ELSEIF (Me.iPlcStatus == 1) THEN  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatus == 2) THEN  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusEnDefaut = 0;   
 ELSEIF (Me.iPlcStatus == 3) THEN  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus == 4) THEN  
 Me.bStatusFinCourseFerme = 0;  
 Me.bStatusFinCourseOuvert = 1;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatus == 5) THEN  
 Me.bStatusFinCourseFerme = 1;  
 Me.bStatusFinCourseOuvert = 0;  
 Me.bStatusEnDefaut = 1;   
 ENDIF; ' IF (Me.iPlcStatus == )  
  
  
  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenVannePapillon  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenVannePapillon  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatus.Input.InputSource = "Me.iPlcStatus\_FB";  
Me.iStatusLocal.Input.InputSource = "Me.iStatusLocal\_FB";

#### Template ArchestrA $dwGenVannePapillon2S

##### Description

Not Applicable

##### Derived from

$dwGenVanne2Input

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenVannePapillon3P

##### Description

Not Applicable

##### Derived from

$dwGenVanne3Input

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bStatusFCFerme | boolean |  |  |  |  |  |
| bStatusFCInterm | boolean |  |  |  |  |  |
| bStatusFCOuvert | boolean |  |  |  |  |  |
| bStatusMouvement | boolean |  |  |  |  |  |
| iPlcStatusFCFerme\_FB | integer |  |  |  |  |  |
| iPlcStatusFCInterm\_FB | integer |  |  |  |  |  |
| iPlcStatusFCOuvert\_FB | boolean |  |  |  |  |  |
| iStatusLocal\_FB | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |
| sTexteFerme | string |  |  |  |  |  |
| sTexteInterm | string |  |  |  |  |  |
| sTexteOuvert | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcStatusFCFerme

|  |  |
| --- | --- |
| Name | dcPlcStatusFCFerme |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusFCFerme + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : DIC  
' Date : 13 Juin 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusFCFerme");  
 logmessage("iPlcStatusFCFerme : " + Me.iPlcStatusFCFerme);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusFCFerme = 0;  
Me.bStatusMouvement = 0;  
Me.bStatusEnDefaut = 0;  
  
IF Me.iTypeAnimation == 20 THEN ' BR7   
 IF (Me.iPlcStatusFCFerme == 0) THEN 'gris = Ferme  
 Me.bStatusFCFerme = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCFerme == 1) THEN 'jaune = Mouvement   
 Me.bStatusFCFerme = 0;  
 Me.bStatusMouvement = 1;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCFerme == 2) THEN 'rouge = Ouvert  
 Me.bStatusFCFerme = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCFerme == 4) THEN 'vert blink  
 Me.bStatusFCFerme = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusFCFerme == 6) THEN 'rouge blink  
 Me.bStatusFCFerme = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCFerme == )  
  
ELSEIF Me.iTypeAnimation == 40 OR Me.iTypeAnimation == 50 ' BRO, BR6  
OR (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative <> 2)  
THEN  
 IF (Me.iPlcStatusFCFerme == 0) THEN 'gris  
 Me.bStatusFCFerme = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCFerme == 1) THEN 'rouge   
 Me.bStatusFCFerme = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ENDIF; ' IF (Me.iPlcStatusFCFerme == )  
  
ELSEIF Me.iTypeAnimation == 80 THEN ' TSFx  
 IF (Me.iPlcStatusFCFerme == 0) THEN 'gris  
 Me.bStatusFCFerme = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCFerme == 1) THEN 'rouge   
 Me.bStatusFCFerme = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCFerme == 2) THEN ' gris clig  
 Me.bStatusFCFerme = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusFCFerme == 3) THEN ' rouge clig   
 Me.bStatusFCFerme = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCFerme == )  
  
ELSEIF Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 2 THEN 'FR  
 IF (Me.iPlcStatusFCFerme == 0) THEN 'rouge  
 Me.bStatusFCFerme = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ENDIF;  
  
ENDIF; ' IF Me.iTypeAnimation == 20

dcPlcStatusFCInterm

|  |  |
| --- | --- |
| Name | dcPlcStatusFCInterm |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusFCInterm + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : DIC  
' Date : 16 Juin 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusFCInterm");  
 logmessage("iPlcStatusFCInterm : " + Me.iPlcStatusFCInterm);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusFCInterm = 0;  
Me.bStatusMouvement = 0;  
Me.bStatusEnDefaut = 0;  
  
IF Me.iTypeAnimation == 20 THEN ' BR7   
 IF (Me.iPlcStatusFCInterm == 0) THEN 'gris = Interm  
 Me.bStatusFCInterm = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCInterm == 1) THEN 'jaune = Mouvement   
 Me.bStatusFCInterm = 0;  
 Me.bStatusMouvement = 1;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCInterm == 2) THEN 'rouge = Ouvert  
 Me.bStatusFCInterm = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCInterm == 4) THEN 'vert blink  
 Me.bStatusFCInterm = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusFCInterm == 6) THEN 'rouge blink  
 Me.bStatusFCInterm = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCInterm == )  
  
ELSEIF Me.iTypeAnimation == 40 OR Me.iTypeAnimation == 50   
OR (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 0)  
THEN ' BRO, BR6, FR  
 IF (Me.iPlcStatusFCInterm == 0) THEN 'gris  
 Me.bStatusFCInterm = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCInterm == 1) THEN 'rouge   
 Me.bStatusFCInterm = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ENDIF; ' IF (Me.iPlcStatusFCInterm == )  
  
ELSEIF Me.iTypeAnimation == 80 THEN ' TSFx  
 IF (Me.iPlcStatusFCInterm == 0) THEN 'gris  
 Me.bStatusFCInterm = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCInterm == 1) THEN 'rouge   
 Me.bStatusFCInterm = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCInterm == 2) THEN ' gris clig  
 Me.bStatusFCInterm = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusFCInterm == 3) THEN ' rouge clig   
 Me.bStatusFCInterm = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCInterm == )  
  
ELSEIF (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 1)  
THEN ' FR  
 IF (Me.iPlcStatusFCInterm == 0) THEN 'rouge  
 Me.bStatusFCInterm = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCInterm == 1) THEN 'gris  
 Me.bStatusFCInterm = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ENDIF; ' IF (Me.iPlcStatusFCInterm == )  
  
ELSEIF Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 2 THEN 'FR  
 IF (Me.iPlcStatusFCInterm == 2) THEN ' jaune  
 Me.bStatusFCInterm = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCInterm == 3) THEN 'vert cl  
 Me.bStatusFCInterm = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusFCInterm == 4) THEN 'rouge cl  
 Me.bStatusFCInterm = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCInterm == )  
  
ENDIF; ' IF Me.iTypeAnimation == 20

dcPlcStatusFCOuvert

|  |  |
| --- | --- |
| Name | dcPlcStatusFCOuvert |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusFCOuvert + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : DIC  
' Date : 16 Juin 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusFCOuvert");  
 logmessage("iPlcStatusFCOuvert : " + Me.iPlcStatusFCOuvert);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusFCOuvert = 0;  
Me.bStatusMouvement = 0;  
Me.bStatusEnDefaut = 0;  
  
IF Me.iTypeAnimation == 20 THEN ' BR7   
 IF (Me.iPlcStatusFCOuvert == 0) THEN 'gris = Ouvert  
 Me.bStatusFCOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCOuvert == 1) THEN 'jaune = Mouvement   
 Me.bStatusFCOuvert = 0;  
 Me.bStatusMouvement = 1;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCOuvert == 2) THEN 'rouge = Ouvert  
 Me.bStatusFCOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCOuvert == 4) THEN 'vert blink  
 Me.bStatusFCOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusFCOuvert == 6) THEN 'rouge blink  
 Me.bStatusFCOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCOuvert == )  
  
ELSEIF Me.iTypeAnimation == 40 OR Me.iTypeAnimation == 50 OR (Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative <> 2) THEN ' BRO, BR6, FR  
 IF (Me.iPlcStatusFCOuvert == 0) THEN 'gris  
 Me.bStatusFCOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCOuvert == 1) THEN 'rouge   
 Me.bStatusFCOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ENDIF; ' IF (Me.iPlcStatusFCOuvert == )  
  
ELSEIF Me.iTypeAnimation == 80 THEN ' TSFx  
 IF (Me.iPlcStatusFCOuvert == 0) THEN 'gris  
 Me.bStatusFCOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCOuvert == 1) THEN 'rouge   
 Me.bStatusFCOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ELSEIF (Me.iPlcStatusFCOuvert == 2) THEN ' gris clig  
 Me.bStatusFCOuvert = 0;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusFCOuvert == 3) THEN ' rouge clig   
 Me.bStatusFCOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 1;  
 ENDIF; ' IF (Me.iPlcStatusFCOuvert == )  
  
ELSEIF Me.iTypeAnimation == 90 AND Me.iTypeAnimationAlternative == 2 THEN 'FR  
 IF (Me.iPlcStatusFCOuvert == 1) THEN 'rouge   
 Me.bStatusFCOuvert = 1;  
 Me.bStatusMouvement = 0;  
 Me.bStatusEnDefaut = 0;  
 ENDIF;  
  
ENDIF; ' IF Me.iTypeAnimation == 20

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenVannePapillon3P  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenVannePapillon3P  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatusFCFerme.Input.InputSource = "Me.iPlcStatusFCFerme\_FB";  
Me.iPlcStatusFCInterm.Input.InputSource = "Me.iPlcStatusFCInterm\_FB";  
Me.iPlcStatusFCOuvert.Input.InputSource = "Me.iPlcStatusFCOuvert\_FB";  
Me.iStatusLocal.Input.InputSource = "Me.iStatusLocal\_FB";

#### Template ArchestrA $dwGenVanneRectangle2S

##### Description

Not Applicable

##### Derived from

$dwGenVanne2Input

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenVanneRegistre

##### Description

Not Applicable

##### Derived from

$dwGenVanne4Input

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bStatus\_POS0 | boolean |  |  |  |  |  |
| bStatus\_POS1 | boolean |  |  |  |  |  |
| bStatus\_POS2 | boolean |  |  |  |  |  |
| bStatus\_POSF | boolean |  |  |  |  |  |
| iPlcStatus\_POS0\_FB | integer |  |  |  |  |  |
| iPlcStatus\_POS1\_FB | integer |  |  |  |  |  |
| iPlcStatus\_POS2\_FB | integer |  |  |  |  |  |
| iPlcStatus\_POSF\_FB | integer |  |  |  |  |  |
| iStatusLocal\_FB | integer |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPlcStatus\_POS0

|  |  |
| --- | --- |
| Name | dcPlcStatus\_POS0 |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus\_POS0 + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 14 Juillet 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatus\_POS0");  
 logmessage("iPlcStatus\_POS0 : " + Me.iPlcStatus\_POS0);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatus\_POS0 = 0;  
  
IF Me.iTypeAnimation == 60 THEN ' PFEIFFER - BR8  
 IF Me.iPlcStatus\_POS0 == 0 THEN  
 Me.bStatus\_POS0 = 0;  
 ELSEIF Me.iPlcStatus\_POS0 == 1 THEN  
 Me.bStatus\_POS0 = 1;  
 ENDIF;  
ENDIF;

dcPlcStatus\_POS1

|  |  |
| --- | --- |
| Name | dcPlcStatus\_POS1 |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus\_POS1 + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 14 Juillet 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatus\_POS1");  
 logmessage("iPlcStatus\_POS1 : " + Me.iPlcStatus\_POS1);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatus\_POS1 = 0;  
  
IF Me.iTypeAnimation == 60 THEN ' PFEIFFER - BR8  
 IF Me.iPlcStatus\_POS1 == 0 THEN  
 Me.bStatus\_POS1 = 0;  
 ELSEIF Me.iPlcStatus\_POS1 == 1 THEN  
 Me.bStatus\_POS1 = 1;  
 ENDIF;  
ENDIF;

dcPlcStatus\_POS2

|  |  |
| --- | --- |
| Name | dcPlcStatus\_POS2 |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus\_POS2 + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 14 Juillet 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatus\_POS2");  
 logmessage("iPlcStatus\_POS2 : " + Me.iPlcStatus\_POS2);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatus\_POS2 = 0;  
  
IF Me.iTypeAnimation == 60 THEN ' PFEIFFER - BR8  
 IF Me.iPlcStatus\_POS2 == 0 THEN  
 Me.bStatus\_POS2 = 0;  
 ELSEIF Me.iPlcStatus\_POS2 == 1 THEN  
 Me.bStatus\_POS2 = 1;  
 ENDIF;  
ENDIF;

dcPlcStatus\_POSF

|  |  |
| --- | --- |
| Name | dcPlcStatus\_POSF |
| Description |  |
| Trigger | DataChange of Me.iPlcStatus\_POSF + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 14 Juillet 2008  
' Ce script traitera les status  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatus\_POSF");  
 logmessage("iPlcStatus\_POSF : " + Me.iPlcStatus\_POSF);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatus\_POSF = 0;  
  
IF Me.iTypeAnimation == 60 THEN ' PFEIFFER - BR8  
 IF Me.iPlcStatus\_POSF == 0 THEN  
 Me.bStatus\_POSF = 0;  
 ELSEIF Me.iPlcStatus\_POSF == 1 THEN  
 Me.bStatus\_POSF = 1;  
 ENDIF;  
ENDIF;

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenVanneRegistre  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 ' Affect Input source  
 If MyArea.\_Config.Activate.FB Then  
 Me.Object.Script.AffectInputSource = True;  
 Endif;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenVanneRegistre  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.iPlcStatus\_POS0.Input.InputSource = "Me.iPlcStatus\_POS0\_FB";  
Me.iPlcStatus\_POS1.Input.InputSource = "Me.iPlcStatus\_POS1\_FB";  
Me.iPlcStatus\_POS2.Input.InputSource = "Me.iPlcStatus\_POS2\_FB";  
Me.iPlcStatus\_POSF.Input.InputSource = "Me.iPlcStatus\_POS3\_FB";  
Me.iStatusLocal.Input.InputSource = "Me.iStatusLocal\_FB";

#### Template ArchestrA $dwGenVanneRegistre2S

##### Description

Not Applicable

##### Derived from

$dwGenVanne2Input

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bTjsCommandable | boolean |  |  |  |  |  |
| iTypeCommandeFermeture | integer |  |  |  |  |  |
| iTypeCommandeOuverture | integer |  |  |  |  |  |
| iValCommandeFermeture | integer |  |  |  |  |  |
| iValCommandeOuverture | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenVentilateur

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenVIB

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenVibreur

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenVireur

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenVis1S

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenVis2S

##### Description

Not Applicable

##### Derived from

$dwGenMoteurDirect2S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenVIT

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenAnalogIn

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwIntSplitI

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bBools | boolean |  |  |  |  |  |
| iBools | integer |  | X |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

CalcBools

|  |  |
| --- | --- |
| Name | CalcBools |
| Description |  |
| Trigger | DataChange of Me.iBools |

**Declarations :**

Not Applicable

**Script :**

dim i as integer;  
dim masque as integer;  
masque=1;  
for i=1 to 16  
 Me.bBools[i]=Me.iBools & masque;  
 masque=masque\*2;  
next;

#### Template ArchestrA $dwIntSplitIO

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bBools | boolean |  |  |  |  |  |
| iBools | integer | X |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

CalcbBools

|  |  |
| --- | --- |
| Name | CalcbBools |
| Description |  |
| Trigger | DataChange of false |

**Declarations :**

Not Applicable

**Script :**

{Me.iBools}  
  
  
'logmessage("iBools vers bBools : Valeur " + stringfromintg(Me.iBools ,10));   
dim i as integer;  
dim masque as integer;  
masque=1;  
for i=1 to 16  
 Me.bBools[i]=Me.iBools & masque;  
 'logmessage("iBools vers bBools : masque " + stringfromintg(masque,10) + " ; Valeur : " + stringfromintg(Me.bBools[i],10) );  
 masque=masque\*2;  
next;

CalciBools1

|  |  |
| --- | --- |
| Name | CalciBools1 |
| Description |  |
| Trigger | WhileTrue of Me.bBools[1] |

**Declarations :**

Not Applicable

**Script :**

'logmessage("bBools vers iBools : Bools1");  
Me.iBools = 1;   
Me.bBools[1]=false;

CalciBools2

|  |  |
| --- | --- |
| Name | CalciBools2 |
| Description |  |
| Trigger | WhileTrue of Me.bBools[2] |

**Declarations :**

Not Applicable

**Script :**

'logmessage("bBools vers iBools : Bools2");  
Me.iBools = 2;   
Me.bBools[2]=false;

CalciBools3

|  |  |
| --- | --- |
| Name | CalciBools3 |
| Description |  |
| Trigger | WhileTrue of Me.bBools[3] |

**Declarations :**

Not Applicable

**Script :**

'logmessage("bBools vers iBools : Bools3");  
Me.iBools = 4;   
Me.bBools[3]=false;

CalciBools4

|  |  |
| --- | --- |
| Name | CalciBools4 |
| Description |  |
| Trigger | WhileTrue of Me.bBools[4] |

**Declarations :**

Not Applicable

**Script :**

'logmessage("bBools vers iBools : Bools4");  
Me.iBools = 8;   
Me.bBools[4]=false;

CalciBools5

|  |  |
| --- | --- |
| Name | CalciBools5 |
| Description |  |
| Trigger | WhileTrue of Me.bBools[5] |

**Declarations :**

Not Applicable

**Script :**

'logmessage("bBools vers iBools : Bools5");  
Me.iBools = 16;   
Me.bBools[5]=false;

CalciBools6

|  |  |
| --- | --- |
| Name | CalciBools6 |
| Description |  |
| Trigger | WhileTrue of Me.bBools[6] |

**Declarations :**

Not Applicable

**Script :**

'logmessage("bBools vers iBools : Bools6");  
Me.iBools = 32;   
Me.bBools[6]=false;

CalciBools7

|  |  |
| --- | --- |
| Name | CalciBools7 |
| Description |  |
| Trigger | WhileTrue of Me.bBools[7] |

**Declarations :**

Not Applicable

**Script :**

'logmessage("bBools vers iBools : Bools7");  
Me.iBools = 64;   
Me.bBools[7]=false;

CalciBools8

|  |  |
| --- | --- |
| Name | CalciBools8 |
| Description |  |
| Trigger | WhileTrue of Me.bBools[8] |

**Declarations :**

Not Applicable

**Script :**

'logmessage("bBools vers iBools : Bools8");  
Me.iBools = 128;   
Me.bBools[8]=false;

CalciBools9

|  |  |
| --- | --- |
| Name | CalciBools9 |
| Description |  |
| Trigger | WhileTrue of Me.bBools[9] |

**Declarations :**

Not Applicable

**Script :**

'logmessage("bBools vers iBools : Bools9");  
Me.iBools = 256;   
Me.bBools[9]=false;

CalciBools10

|  |  |
| --- | --- |
| Name | CalciBools10 |
| Description |  |
| Trigger | WhileTrue of Me.bBools[10] |

**Declarations :**

Not Applicable

**Script :**

'logmessage("bBools vers iBools : Bools10");  
Me.iBools = 512;   
Me.bBools[10]=false;

CalciBools11

|  |  |
| --- | --- |
| Name | CalciBools11 |
| Description |  |
| Trigger | WhileTrue of Me.bBools[11] |

**Declarations :**

Not Applicable

**Script :**

'logmessage("bBools vers iBools : Bools11");  
Me.iBools = 1024;   
Me.bBools[11]=false;

CalciBools12

|  |  |
| --- | --- |
| Name | CalciBools12 |
| Description |  |
| Trigger | WhileTrue of Me.bBools[12] |

**Declarations :**

Not Applicable

**Script :**

'logmessage("bBools vers iBools : Bools12");  
Me.iBools = 2048;   
Me.bBools[12]=false;

CalciBools13

|  |  |
| --- | --- |
| Name | CalciBools13 |
| Description |  |
| Trigger | WhileTrue of Me.bBools[13] |

**Declarations :**

Not Applicable

**Script :**

'logmessage("bBools vers iBools : Bools13");  
Me.iBools = 4096;   
Me.bBools[13]=false;

CalciBools14

|  |  |
| --- | --- |
| Name | CalciBools14 |
| Description |  |
| Trigger | WhileTrue of Me.bBools[14] |

**Declarations :**

Not Applicable

**Script :**

'logmessage("bBools vers iBools : Bools14");  
Me.iBools = 8192;   
Me.bBools[14]=false;

CalciBools15

|  |  |
| --- | --- |
| Name | CalciBools15 |
| Description |  |
| Trigger | WhileTrue of Me.bBools[15] |

**Declarations :**

Not Applicable

**Script :**

'logmessage("bBools vers iBools : Bools15");  
Me.iBools = 16384;   
Me.bBools[15]=false;

CalciBools16

|  |  |
| --- | --- |
| Name | CalciBools16 |
| Description |  |
| Trigger | WhileTrue of Me.bBools[16] |

**Declarations :**

Not Applicable

**Script :**

'logmessage("bBools vers iBools : Bools1");  
Me.iBools = 32768;   
Me.bBools[16]=false;

#### Template ArchestrA $dwIntSplitIO.dwListeObjets

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| sTabRecettes | string |  |  |  |  |  |
| sTabTagElements | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwIntSplitIO.dwListeObjets.dwConnectionParams

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| strConnectionStringFB | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwKlaxon

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| Nom | string |  |  |  |  |  |

##### Field Attributes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Attribute name | Data Type | IO Scaling | History | Limit Alarms | Rate Of Change Alarms | Target Deviation Alarms | Bad Value Alarm | Statistics |
| Anim | integer |  |  |  |  |  |  |  |

##### Scripts

Not Applicable

#### Template ArchestrA $dwLabo

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwMelangeur\_FMT

##### Description

Not Applicable

##### Derived from

$dwMoteurDirect1S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwMesureAnalogiqueAvecForcage

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwMesureAnalogique

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bActivationForcage | boolean | X |  |  |  |  |
| rValForcage | float | X |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwMoteur

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwActionneur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmeDefAU | boolean |  | X |  | X |  |
| bAlarmeDefIS | boolean |  | X |  | X |  |
| bAlarmeDefMTH | boolean |  | X |  | X |  |
| bAlarmeDefRot | boolean |  | X |  | X |  |
| bAlarmeDefTemp | boolean |  | X |  | X |  |
| bCommandeArretManu | boolean | X |  |  |  |  |
| bCommandeInhibitionDefIS | boolean | X |  |  |  |  |
| bCommandeInhibitionDefMTH | boolean | X |  |  |  |  |
| bCommandeInhibitionDefRot | boolean | X |  |  |  |  |
| bCommandeInhibitionDefTemp | boolean | X |  |  |  |  |
| bCommandeMarcheManu | boolean | X |  |  |  |  |
| iCommandeTempsDefRKM | integer | X |  |  |  |  |
| iCommandeTempsDefRot | integer | X |  |  |  |  |
| sAlarmeDefAU | string |  |  |  |  |  |
| sAlarmeDefIS | string |  |  |  |  |  |
| sAlarmeDefMTH | string |  |  |  |  |  |
| sAlarmeDefRot | string |  |  |  |  |  |
| sAlarmeDefTemp | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

pdicNbForcage

|  |  |
| --- | --- |
| Name | pdicNbForcage |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

Dim nb as Integer;  
  
nb=0;  
  
IF Me.bCommandeInhibitionDefIS == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefMTH == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefRot == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefTemp == false THEN  
 nb = nb + 1;  
ENDIF;  
  
Me.iNbForcage = nb;

#### Template ArchestrA $dwMoteurDirect1S

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwMoteur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmeDefRKM | boolean |  | X |  | X |  |
| bCommandeInhibitionDefRKM | boolean | X |  |  |  |  |
| bStatusAutorisationMarche | boolean |  | X |  |  |  |
| bStatusMarche | boolean |  | X |  |  |  |
| bStatusOrdreMarche | boolean |  | X |  |  |  |
| sAlarmeDefRKM | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwMoteurDirect1S\_2M

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwMoteurDirect1S

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmeDefMTH\_2 | boolean |  | X |  | X |  |
| bAlarmeDefRKM\_2 | boolean |  | X |  | X |  |
| bCommandeInhibitionDefMTH\_2 | boolean | X |  |  |  |  |
| bCommandeInhibitionDefRKM\_2 | boolean | X |  |  |  |  |
| sAlarmeDefMTH\_2 | string |  |  |  |  |  |
| sAlarmeDefRKM\_2 | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwMoteurDirect1S\_BR8

##### Description

Not Applicable

##### Derived from

$dwGenericMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

dcStatusMarche

|  |  |
| --- | --- |
| Name | dcStatusMarche |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusMarche + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 16 Avril 2008  
' Ce script traitera les status groupés pour marche/defaut  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusMarche");  
 logmessage("iPlcStatusMarche : " + Me.iPlcStatusMarche);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusMarche = 0;  
Me.bStatusEnDefaut = 0;  
Me.bStatusVidange = 0;  
Me.bStatusDemarrage = 0;  
  
IF Me.iTypeAnimation == 10 AND Me.iTypeAnimationAlternative == 0 THEN ' PFEIFFER - BR8  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche == 2) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ELSEIF (Me.iPlcStatusMarche == 3) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche == 4) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 ELSEIF (Me.iPlcStatusMarche == 5) THEN  
 ' Vidange - jaune clignotant  
 Me.bStatusVidange = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarche == )  
  
ELSEIF Me.iTypeAnimation == 10 AND Me.iTypeAnimationAlternative == 1 THEN ' VEN4 & VEN5  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ENDIF;  
ELSEIF Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 0 THEN ' FMT  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche == 2) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ELSEIF (Me.iPlcStatusMarche == 3) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche == 4) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 ELSEIF (Me.iPlcStatusMarche == 5) THEN  
 ' Vidange - jaune clignotant  
 Me.bStatusVidange = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarche == )  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN

#### Template ArchestrA $dwMoteurDirect1S\_FMT

##### Description

Not Applicable

##### Derived from

$dwGenericMoteurDirect1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

dcStatusMarche

|  |  |
| --- | --- |
| Name | dcStatusMarche |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusMarche + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CPO  
' Date : 25/09/2013  
' Ce script traitera les status groupés pour marche/defaut  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusMarche");  
 logmessage("iPlcStatusMarche : " + Me.iPlcStatusMarche);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusMarche = 0;  
Me.bStatusEnDefaut = 0;  
Me.bStatusVidange = 0;  
Me.bStatusDemarrage = 0;  
  
IF Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 0 THEN ' FMT  
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche == 2) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 ELSEIF (Me.iPlcStatusMarche == 3) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 ELSEIF (Me.iPlcStatusMarche == 4) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 ELSEIF (Me.iPlcStatusMarche == 5) THEN  
 ' Vidange - jaune clignotant  
 Me.bStatusVidange = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarche == )  
  
ELSEIF Me.iTypeAnimation == 70 AND Me.iTypeAnimationAlternative == 1 THEN   
 IF (Me.iPlcStatusMarche == 0) THEN  
 ' Arrêt sans défaut  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ELSEIF (Me.iPlcStatusMarche == 1) THEN  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
 ENDIF;  
  
ENDIF; ' IF Me.iTypeAnimation == 10 THEN

#### Template ArchestrA $dwMoteurDirect2S

##### Description

Not Applicable

##### Derived from

$dwMoteur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmeDefRKMAr | boolean |  | X |  | X |  |
| bAlarmeDefRKMAv | boolean |  | X |  | X |  |
| bCommandeInhibitionDefRKMAr | boolean | X |  |  |  |  |
| bCommandeInhibitionDefRKMAv | boolean | X |  |  |  |  |
| bCommandeMarcheArManu | boolean | X |  |  |  |  |
| bCommandeMarcheAvManu | boolean | X |  |  |  |  |
| bStatusAutorisationMarcheAr | boolean |  | X |  |  |  |
| bStatusAutorisationMarcheAv | boolean |  | X |  |  |  |
| bStatusMarcheArriere | boolean |  | X |  |  |  |
| bStatusMarcheAvant | boolean |  | X |  |  |  |
| bStatusOrdreMarcheAr | boolean |  | X |  |  |  |
| bStatusOrdreMarcheAv | boolean |  | X |  |  |  |
| sAlarmeDefRKMAr | string |  |  |  |  |  |
| sAlarmeDefRKMAv | string |  |  |  |  |  |
| sTexteMarcheAr | string |  |  |  |  |  |
| sTexteMarcheAv | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwMoteurDirect2S\_BR8

##### Description

Not Applicable

##### Derived from

$dwGenericMoteurDirect2S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

dcStatusMarcheAvant

|  |  |
| --- | --- |
| Name | dcStatusMarcheAvant |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusMarcheAvant + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 17 Mars 2008  
' Ce script traitera les status groupés pour marche Avant/defaut  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusMarcheAvant");  
 logmessage("iPlcStatusMarcheAvant : " + Me.iPlcStatusMarcheAvant);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusMarcheAvant = 0;  
Me.bStatusEnDefautAvant = 0;  
Me.bStatusVidangeAvant = 0;  
Me.bStatusDemarrageAvant = 0;  
Me.bEnMarche = 0;  
  
 IF (Me.iPlcStatusMarcheAvant == 0) THEN  
 ' Arrêt sans défaut  
 Me.bStatusMarcheAvant = 0;  
 Me.bStatusEnDefautAvant = 0;  
 Me.bStatusVidangeAvant = 0;  
 Me.bStatusDemarrageAvant = 0;  
 Me.bEnMarche = 0;  
 ELSEIF (Me.iPlcStatusMarcheAvant == 1) THEN  
 ' Marche sans défaut  
 Me.bStatusMarcheAvant = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarcheAvant == )

dcStatusMarcheArriere

|  |  |
| --- | --- |
| Name | dcStatusMarcheArriere |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusMarcheArriere + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 17 Mars 2008  
' Ce script traitera les status groupés pour marche arriere/defaut  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusMarcheArriere");  
 logmessage("iPlcStatusMarcheArriere : " + Me.iPlcStatusMarcheArriere);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusMarcheArriere = 0;  
Me.bStatusEnDefautArriere = 0;  
Me.bStatusVidangeArriere = 0;  
Me.bStatusDemarrageArriere = 0;  
Me.bEnMarche = 0;  
  
 IF (Me.iPlcStatusMarcheArriere == 1) THEN  
 ' Arrêt sans défaut  
 Me.bStatusMarcheArriere = 0;  
 Me.bStatusEnDefautArriere = 0;  
 Me.bStatusVidangeArriere = 0;  
 Me.bStatusDemarrageArriere = 0;  
 Me.bEnMarche = 0;  
 ELSEIF (Me.iPlcStatusMarcheArriere == 0) THEN  
 ' Marche sans défaut  
 Me.bStatusMarcheArriere = 1;  
 ENDIF; ' IF (Me.iPlcStatusMarcheArriere == )

dcStatusMarche

|  |  |
| --- | --- |
| Name | dcStatusMarche |
| Description |  |
| Trigger | DataChange of Me.iPlcStatusMarche + Me.iTypeAnimation |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 6 Juillet 2008  
' Ce script traitera les status groupés pour marche (en cas de moteur 2s avec animation unique)  
  
IF MyArea.bDebug == true THEN  
 logmessage("Instance : " + Me.TagName);  
 logmessage("Script : dcPlcStatusMarche");  
 logmessage("iPlcStatusMarche : " + Me.iPlcStatusMarche);  
 logmessage("iTypeAnimation : " + Me.iTypeAnimation);  
ENDIF;  
  
Me.bStatusMarche = 0;  
Me.bStatusEnDefaut = 0;  
Me.bStatusVidange = 0;  
Me.bStatusDemarrage = 0;  
Me.bEnMarche = 0;  
  
  
 IF (Me.iPlcStatusMarche == 1) THEN  
 ' Arrêt sans défaut - vert  
 Me.bStatusMarche = 0;  
 Me.bStatusEnDefaut = 0;  
 Me.bStatusVidange = 0;  
 Me.bStatusDemarrage = 0;  
   
 ELSEIF (Me.iPlcStatusMarche == 2) OR (Me.iPlcStatusMarche == 3) THEN  
 ' Marche sans défaut - rouge  
 Me.bStatusMarche = 1;  
 Me.bEnMarche = 1;  
  
 ELSEIF (Me.iPlcStatusMarche >= 4) AND (Me.iPlcStatusMarche < 8) THEN  
 ' Démarrage - jaune  
 Me.bStatusDemarrage = 1;  
 Me.bEnMarche = 1;  
  
 ELSEIF (Me.iPlcStatusMarche >= 8) AND (Me.iPlcStatusMarche < 16) THEN  
 ' Arrêt avec défaut - vert clignotant  
 Me.bStatusEnDefaut = 1;  
  
 ELSEIF (Me.iPlcStatusMarche >= 16) AND (Me.iPlcStatusMarche < 32) THEN  
 ' Marche avec défaut - rouge clignotant  
 Me.bStatusMarche = 1;  
 Me.bStatusEnDefaut = 1;  
 Me.bEnMarche = 1;  
  
 ELSEIF (Me.iPlcStatusMarche >= 32) THEN  
 ' Vidange - jaune clignotant  
 Me.bStatusVidange = 1;  
 Me.bEnMarche = 1;  
  
 ENDIF; ' IF (Me.iPlcStatusMarche == )

#### Template ArchestrA $dwPlaqueMagnetique\_BR8

##### Description

Not Applicable

##### Derived from

$dwMoteurDirect1S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwPompe\_BR8

##### Description

Not Applicable

##### Derived from

$dwMoteurDirect1S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwProduits

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwLabo

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwSeparateur\_BR8

##### Description

Not Applicable

##### Derived from

$dwMoteurDirect1S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwSignalSonore

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| iNombreAlarmes | integer |  |  | X |  |  |

##### Field Attributes

Not Applicable

##### Scripts

pcCntAlarms

|  |  |
| --- | --- |
| Name | pcCntAlarms |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

' CCO  
' 15 Janvier 2009  
  
  
' Me.iNombreAlarmes = TSFS.AlarmOnCntTotal;  
' Me.iNombreAlarmes = TSFM.AlarmOnCntTotal;  
' Me.iNombreAlarmes = TSFQ.AlarmOnCntTotal;

#### Template ArchestrA $dwStationsClients

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| sAteliers | string |  |  |  |  |  |
| sNoms | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwSuggesteurConsigneFR

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bCalcChaux | boolean |  |  |  |  |  |
| bCalcDolo | boolean |  |  |  |  |  |
| bCalcMagnesia | boolean |  |  |  |  |  |
| bChauxFR123 | boolean |  |  |  |  |  |
| bCorrection | boolean |  |  |  |  |  |
| bDebug | boolean |  |  |  |  |  |
| bFRB\_DS17\_RC | boolean |  |  |  |  |  |
| bFRB\_DS17\_WD | boolean |  |  |  |  |  |
| Const.iNBRE\_MES | integer |  |  |  |  |  |
| fBuf | float |  |  |  |  |  |
| fCO21040FR | float |  |  |  |  |  |
| fCO21040FR\_OLD | float |  |  |  |  |  |
| fCO2210FR | float |  |  |  |  |  |
| fCO2210FR\_OLD | float |  |  |  |  |  |
| fCORCOKFR | float |  |  |  |  |  |
| fCORGAZFR | float |  |  |  |  |  |
| fCORHASFR | float |  |  |  |  |  |
| fCORLIGFR | float |  |  |  |  |  |
| fEVOL1040FR | float |  |  |  |  |  |
| fEVOL210FR | float |  |  |  |  |  |
| fEVOLRMAGFR | float |  |  |  |  |  |
| fFR\_QP | float |  |  |  |  |  |
| fFR\_r | float |  |  |  |  |  |
| fFR\_rTPB | float |  |  |  |  |  |
| fFR\_T | float |  |  |  |  |  |
| fFR\_TIME | double |  |  |  |  |  |
| fFR\_TIME\_TPB | float |  |  |  |  |  |
| fFR\_TLC\_CONS | float |  |  |  |  |  |
| fFR\_TPB | float |  |  |  |  |  |
| fFR\_TPB1 | float |  |  |  |  |  |
| fFR\_Y | float |  |  |  |  |  |
| fFRF | float |  |  |  |  |  |
| fFRGTLC | float |  |  |  |  |  |
| fFRGTPB | float |  |  |  |  |  |
| fFRI | float |  |  |  |  |  |
| fFRJ | float |  |  |  |  |  |
| fFRK | float |  |  |  |  |  |
| fFRL | float |  |  |  |  |  |
| fFRM | float |  |  |  |  |  |
| fFRN | float |  |  |  |  |  |
| fFRO | float |  |  |  |  |  |
| fFRP | float |  |  |  |  |  |
| fFRQ | float |  |  |  |  |  |
| fFRR | float |  |  |  |  |  |
| fPERIODE | float |  |  |  |  |  |
| fPERIODE\_TPB | float |  |  |  |  |  |
| fRCACFR | float |  |  |  |  |  |
| fRMAGFR | float |  |  |  |  |  |
| fRMAGFR\_OLD | float |  |  |  |  |  |
| fSECTIME | double |  |  |  |  |  |
| iIndice | integer |  |  |  |  |  |
| iIndiceTpb | integer |  |  |  |  |  |
| iSecMes | integer |  |  |  |  |  |
| iSecMesTpb | integer |  |  |  |  |  |
| sMAIN\_LIBRE | string |  |  |  |  |  |

##### Field Attributes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Attribute name | Data Type | IO Scaling | History | Limit Alarms | Rate Of Change Alarms | Target Deviation Alarms | Bad Value Alarm | Statistics |
| FR\_QP | float |  |  |  |  |  |  |  |
| FR\_PROD\_CONS | integer |  |  |  |  |  |  |  |
| FR\_TLC\_CONS | float |  |  |  |  |  |  |  |
| FR4\_TLCL | integer |  |  |  |  |  |  |  |
| FR4\_QPL | integer |  |  |  |  |  |  |  |
| FR4\_TPBL | integer |  |  |  |  |  |  |  |
| FR\_TPB | integer |  |  |  |  |  |  |  |

##### Scripts

CALCUL

|  |  |
| --- | --- |
| Name | CALCUL |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

'script that makes the calculations for expert system  
  
Dim fBuf as float;  
  
'calcul for the coefficients  
  
'special case for FR4  
if StringToIntg(StringRight(Me.Tagname,1)) == 4 then  
 fBuf = Me.FR4\_QPL;  
 Me.fFR\_QP = fBuf / 10;  
 fBuf = 0;  
else  
 Me.fFR\_QP = Me.FR\_QP;  
endif;  
  
if Me.fFR\_QP < 10 then  
 Me.fFR\_QP = 10;  
else   
 if Me.fFR\_QP > 40 and Me.FR\_PROD\_CONS <> 1 then  
 Me.fFR\_QP = 40;  
 else  
 if Me.fFR\_QP > 80 and Me.FR\_PROD\_CONS == 1 then  
 Me.fFR\_QP = 80;  
 endif;  
 endif;  
endif;  
  
if Me.FR\_PROD\_CONS == 4 then  
'magnesia  
 Me.bCalcMagnesia = true;  
else  
 if Me.FR\_PROD\_CONS == 2 then  
 'dolo  
 Me.bCalcDolo = true;  
 else  
 if Me.FR\_PROD\_CONS == 1 then  
 'chaux  
 Me.bCalcChaux = true;  
 endif;  
 endif;  
endif;

calcGTLC

|  |  |
| --- | --- |
| Name | calcGTLC |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

' register the temperatures for expert system  
' script that calculates gradient TLC  
  
Dim i as integer;  
'Dim fX as float;  
'Dim fXX as float;  
Dim fX as double;  
Dim fXX as double;  
Dim fFR as float;  
Dim fFR\_XY as double;  
Dim fFR\_YY as float;  
Dim fBuf as float;  
  
Me.fPERIODE = SE\_FR\_Coefficients.iTPSREG \* 60/36;  
  
'Me.fSECTIME = DateTimeGMT() ;  
'Me.fSECTIME = Int(Me.fSECTIME \* 86400 - (10\*365\*24\*3600) - (47\*3600) + (27 \* 60));  
'LogMessage("sectime2 = " + Me.fSECTIME);  
  
' read the datas  
if (Me.iSecMes >= Me.fPERIODE) then   
 Me.iSecMes = 0;  
 'transfer of the values  
 if StringToIntg(StringRight(Me.Tagname,1)) == 4 then  
 Me.fFR\_TLC\_CONS = Me.FR4\_TLCL;  
 else  
 Me.fFR\_TLC\_CONS = Me.FR\_TLC\_CONS;   
 endif;  
  
 Me.fFR\_T[Me.iIndice] = Me.fFR\_TLC\_CONS;  
  
 'fSECTIME = number of seconds since 1980  
 Me.fSECTIME = DateTimeGMT() ;  
 Me.fSECTIME = Int(Me.fSECTIME \* 86400 - (10\*365\*24\*3600) - (47\*3600) + (27 \* 60));  
 fBuf = Me.fSECTIME;  
  
 if Me.bDebug == true then  
 LogMessage("SECTIME = " + fBuf);  
 endif;  
  
 Me.fFR\_TIME[Me.iIndice] = fBuf;  
  
 Me.iIndice = Me.iIndice + 1;  
   
 if (Me.iIndice > Me.Const.iNBRE\_MES) then  
 Me.iIndice = 1;  
 endif;  
  
 'Calculation for regressions  
  
 'calcul of the sums  
 i = 1;  
 fX = 0;  
 fFR = 0;  
  
 while(i <= Me.Const.iNBRE\_MES)  
 fX = fX + Me.fFR\_TIME[i];  
 fFR = fFR + Me.fFR\_T[i];  
 i = i + 1;  
 endwhile;  
  
 fX = fX / Me.Const.iNBRE\_MES;  
 fFR = fFR / Me.Const.iNBRE\_MES;  
  
 i = 1;  
 fFR\_XY = 0;  
 fFR\_YY = 0;  
 fXX = 0;  
  
 while(i <= Me.Const.iNBRE\_MES)  
 fXX = fXX + ((Me.fFR\_TIME[i] - fX) \* (Me.fFR\_TIME[i] - fX));  
 fFR\_XY = fFR\_XY + ((Me.fFR\_TIME[i] - fX) \* (Me.fFR\_T[i] - fFR));  
 fFR\_YY = fFR\_YY + ((Me.fFR\_T[i] - fFR) \* (Me.fFR\_T[i] - fFR));  
 i = i + 1;  
 endwhile;  
   
 ' averages  
 Me.fFR\_Y = fFR;  
   
 fX = Sqrt(fXX);   
 fFR = Sqrt(fFR\_YY);  
  
 ' coefficient of correlation  
  
 if (fFR <> 0) then  
 Me.fFR\_r = fFR\_XY / (fFR \* fX);  
 Me.fFRGTLC = Me.fFR\_r \* fFR \* 3600 / fX;  
 else  
 Me.fFR\_r = 1;  
 Me.fFRGTLC = 0;  
 endif;  
  
 if Me.bDebug == true then  
 LogMessage("fFR = " + fFR);  
 LogMessage("fFR\_r = " + Me.fFR\_r);  
 LogMessage("fFRGTLC = " + Me.fFRGTLC);  
 endif;  
  
else  
 Me.iSecMes = Me.iSecMes + 1;  
endif;

calcGTPB

|  |  |
| --- | --- |
| Name | calcGTPB |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

' register the temperatures for expert system  
' script that calculates gradient TPB - Gradient Temperature Pyrometre à Bague  
  
Dim i as integer;  
'Dim fX as float;  
'Dim fXX as float;  
Dim fX as double;  
Dim fXX as double;  
Dim fFR as float;  
Dim fFR\_XY as double;  
Dim fFR\_YY as float;  
Dim fBuf as float;  
  
Me.fPERIODE\_TPB = SE\_FR\_Coefficients.iTPSREG\_TPB \* 60 / 36;  
  
' read the datas  
if (Me.iSecMesTpb >= Me.fPERIODE\_TPB) then   
 Me.iSecMesTpb = 0;  
  
 'transfer of the values  
 if StringToIntg(StringRight(Me.Tagname,1)) == 4 then  
 Me.fFR\_TPB = Me.FR4\_TPBL;  
 else  
 Me.fFR\_TPB = Me.FR\_TPB;  
 endif;  
  
 Me.fFR\_TPB1[Me.iIndiceTpb] = Me.fFR\_TPB;  
  
 Me.fSECTIME = DateTimeGMT() ;  
 Me.fSECTIME = Int(Me.fSECTIME \* 86400 - (10\*365\*24\*3600) - (47\*3600) + (27 \* 60));  
 fBuf = Me.fSECTIME;  
  
 Me.fFR\_TIME\_TPB[Me.iIndiceTpb] = fBuf;  
  
 Me.iIndiceTpb = Me.iIndiceTpb + 1;  
   
 if (Me.iIndiceTpb > Me.Const.iNBRE\_MES) then  
 Me.iIndiceTpb = 1;  
 endif;  
  
 'Calculation for regressions  
  
 'calcul of the sums  
 i = 1;  
 fX = 0;  
 fFR = 0;  
 while(i <= Me.Const.iNBRE\_MES)  
 fX = fX + Me.fFR\_TIME\_TPB[i];  
 fFR = fFR + Me.fFR\_TPB1[i];  
 i = i + 1;  
 endwhile;  
  
 fX = fX / Me.Const.iNBRE\_MES;  
 fFR = fFR / Me.Const.iNBRE\_MES;  
  
 i = 1;  
 fFR\_XY = 0;  
 fFR\_YY = 0;  
 fXX = 0;  
 while(i <= Me.Const.iNBRE\_MES)  
 fXX = fXX + ((Me.fFR\_TIME\_TPB[i] - fX) \* (Me.fFR\_TIME\_TPB[i] - fX));  
 fFR\_XY = fFR\_XY + ((Me.fFR\_TIME\_TPB[i] - fX) \* (Me.fFR\_TPB1[i] - fFR));  
 fFR\_YY = fFR\_YY + ((Me.fFR\_TPB1[i] - fFR) \* (Me.fFR\_TPB1[i] - fFR));  
 i = i + 1;  
 endwhile;  
   
 fX = Sqrt(fXX);   
 fFR = Sqrt(fFR\_YY);  
   
 ' coefficient of correlation  
   
 if (fFR <> 0) then  
 Me.fFR\_rTPB = fFR\_XY / (fFR \* fX);  
 Me.fFRGTPB = Me.fFR\_rTPB \* fFR \* 3600 / fX;  
 else  
 Me.fFR\_rTPB = 1;  
 Me.fFRGTPB = 0;  
 endif;  
  
else  
 Me.iSecMesTpb = Me.iSecMesTpb + 1;  
endif;  
  
if StringToIntg(StringRight(Me.Tagname,1)) == 4 then  
 if Me.bFRB\_DS17\_RC == 1 then  
 Me.bFRB\_DS17\_WD = 0;  
 endif;  
endif;

calcMagnesia

|  |  |
| --- | --- |
| Name | calcMagnesia |
| Description |  |
| Trigger | WhileTrue of Me.bCalcMagnesia |

**Declarations :**

Not Applicable

**Script :**

'script that does the calculations for magnesia  
  
'reset trigger  
Me.bCalcMagnesia = false;  
  
if Me.bDebug == True then  
 LogMessage("Start calcMagnesia");  
endif;  
  
'FR MAGNESIA  
'if Me.FR\_PROD\_CONS == 4 then  
if StringToIntg(StringRight(Me.Tagname,1)) <> 4 then  
 Me.fFRI = ((SE\_FR\_Coefficients.fA[1] \* Me.fRMAGFR) + SE\_FR\_Coefficients.fB[1]);  
 Me.fFRJ = ((SE\_FR\_Coefficients.fA[2] \* Me.fRMAGFR) + SE\_FR\_Coefficients.fB[2]);  
 if ((Me.fFRGTLC < (0 - SE\_FR\_Coefficients.fK[23])) OR ((Me.fFRGTLC >= (0 - SE\_FR\_Coefficients.fK[22])) AND  
 (Me.fFRGTLC < SE\_FR\_Coefficients.fK[22])) OR (Me.fFRGTLC >= SE\_FR\_Coefficients.fK[23])) then  
 'coefficient K  
 Me.fFRK = 1;  
 else  
 if Me.fFRGTLC >= (0 - SE\_FR\_Coefficients.fK[23]) AND   
 Me.fFRGTLC < (0 - SE\_FR\_Coefficients.fK[22])then  
 Me.fFRK = ((1 - SE\_FR\_Coefficients.fK[2])\*(Me.fFRGTLC + SE\_FR\_Coefficients.fK[22])/(SE\_FR\_Coefficients.fK[23] - SE\_FR\_Coefficients.fK[22])) + 1;  
 else  
 Me.fFRK = ((SE\_FR\_Coefficients.fK[1] - 1)\*(Me.fFRGTLC - SE\_FR\_Coefficients.fK[22])/(SE\_FR\_Coefficients.fK[23] - SE\_FR\_Coefficients.fK[22])) + 1;  
 endif;  
 endif;  
 'end coefficient K  
  
 if ((Me.fFRGTPB < (0 - SE\_FR\_Coefficients.fK[31])) OR ((Me.fFRGTPB >= (0 - SE\_FR\_Coefficients.fK[32])) AND  
 (Me.fFRGTPB < SE\_FR\_Coefficients.fK[32])) OR (Me.fFRGTPB >= SE\_FR\_Coefficients.fK[31])) then  
 'coefficient O  
 Me.fFRO = 1;  
 else  
 if Me.fFRGTPB >= (0 - SE\_FR\_Coefficients.fK[31]) AND   
 Me.fFRGTPB < (0 - SE\_FR\_Coefficients.fK[32])then  
 Me.fFRO = ((1 - SE\_FR\_Coefficients.fK[30])\*(Me.fFRGTPB + SE\_FR\_Coefficients.fK[32])/(SE\_FR\_Coefficients.fK[31] - SE\_FR\_Coefficients.fK[32])) + 1;  
 else  
 Me.fFRO = ((SE\_FR\_Coefficients.fK[29] - 1)\*(Me.fFRGTPB - SE\_FR\_Coefficients.fK[32])/(SE\_FR\_Coefficients.fK[31] - SE\_FR\_Coefficients.fK[32])) + 1;  
 endif;  
 endif;  
 'end coefficient O  
  
 Me.fEVOLRMAGFR = Me.fRMAGFR - Me.fRMAGFR\_OLD;  
  
 'coefficient Q  
 if Me.fEVOLRMAGFR < (0 - SE\_FR\_Coefficients.fK[51]) then  
 Me.fFRQ = SE\_FR\_Coefficients.fK[50];  
 else  
 if (Me.fEVOLRMAGFR >= (0 - SE\_FR\_Coefficients.fK[51])) AND (Me.fEVOLRMAGFR < (0 - SE\_FR\_Coefficients.fK[52])) then  
 Me.fFRQ = ((Me.fEVOLRMAGFR + SE\_FR\_Coefficients.fK[52])\* (SE\_FR\_Coefficients.fK[50] - 1)/(SE\_FR\_Coefficients.fK[52] - SE\_FR\_Coefficients.fK[51])) + 1;  
 else  
 if (Me.fEVOLRMAGFR >= (0 - SE\_FR\_Coefficients.fK[52])) AND (Me.fEVOLRMAGFR < SE\_FR\_Coefficients.fK[52]) then  
 Me.fFRQ = 1;  
 else  
 if (Me.fEVOLRMAGFR >= SE\_FR\_Coefficients.fK[52]) AND (Me.fEVOLRMAGFR < SE\_FR\_Coefficients.fK[53]) then  
 Me.fFRQ = ((Me.fEVOLRMAGFR - SE\_FR\_Coefficients.fK[52]) \* (1 - SE\_FR\_Coefficients.fK[49])/  
 (SE\_FR\_Coefficients.fK[52] - SE\_FR\_Coefficients.fK[53])) + 1;  
 else  
 Me.fFRQ = SE\_FR\_Coefficients.fK[49];  
 endif;  
 endif;  
 endif;  
 endif;   
 'coefficient Q  
  
 if (Me.fRMAGFR >= SE\_FR\_Coefficients.fZ[1]) AND (Me.fRMAGFR <= SE\_FR\_Coefficients.fZ[2]) then  
 Me.fBuf = ((Me.fFRK \* Me.fFRO \* Me.fFRQ \* Me.fFRJ) - 1) \* 901600 \* Me.fFR\_QP;  
 else  
 if (Me.fRMAGFR > SE\_FR\_Coefficients.fZ[2]) AND (Me.fRMAGFR <= SE\_FR\_Coefficients.fZ[3]) then  
 Me.fBuf = ((Me.fFRK \* Me.fFRO \* Me.fFRQ \* Me.fFRI) - 1) \* 901600 \* Me.fFR\_QP;  
 endif;  
 endif;  
 'end FR MAGNESIA  
endif;  
  
'endif;  
  
'trigger the corrections  
Me.bCorrection = True;  
  
if Me.bDebug == True then  
 LogMessage("End calcMagnesiaDolo");  
endif;

calcChaux

|  |  |
| --- | --- |
| Name | calcChaux |
| Description |  |
| Trigger | WhileTrue of Me.bCalcChaux |

**Declarations :**

Not Applicable

**Script :**

'script that does the calculations for chaux  
  
'reset trigger  
Me.bCalcChaux = false;  
  
if Me.bDebug == True then  
 LogMessage("Start calcChaux");  
endif;  
  
'FR CHAUX  
'if Me.FR\_PROD\_CONS == 1 then  
 'coefficient F1  
 if Me.fCO21040FR < 0.1 then  
 Me.fFRF[1] = SE\_FR\_Coefficients.fK[5];  
 else  
 if Me.fCO21040FR >= 0.1 AND Me.fCO21040FR < (SE\_FR\_Coefficients.fK[11] - SE\_FR\_Coefficients.fK[12]) then  
 Me.fFRF[1] = ((1 - SE\_FR\_Coefficients.fK[13] - SE\_FR\_Coefficients.fK[5])\*  
 (Me.fCO21040FR - 0.1)/(SE\_FR\_Coefficients.fK[11] - SE\_FR\_Coefficients.fK[12] - 0.1)) + SE\_FR\_Coefficients.fK[5];  
 else  
 if ((Me.fCO21040FR >= (SE\_FR\_Coefficients.fK[11] - SE\_FR\_Coefficients.fK[12])) AND   
 (Me.fCO21040FR < (SE\_FR\_Coefficients.fK[11] + SE\_FR\_Coefficients.fK[12]))) then  
 Me.fFRF[1] = (SE\_FR\_Coefficients.fK[13] \*  
 (Me.fCO21040FR - SE\_FR\_Coefficients.fK[11])/SE\_FR\_Coefficients.fK[12]) + 1;  
 else  
 if ((Me.fCO21040FR >= (SE\_FR\_Coefficients.fK[11] + SE\_FR\_Coefficients.fK[12])) AND (Me.fCO21040FR < 5)) then  
 Me.fFRF[1] = ((SE\_FR\_Coefficients.fK[6] - SE\_FR\_Coefficients.fK[13] - 1) \*  
 (Me.fCO21040FR - 5) / (5 - SE\_FR\_Coefficients.fK[11] - SE\_FR\_Coefficients.fK[12])) +   
 SE\_FR\_Coefficients.fK[6];  
 else  
 Me.fFRF[1] = SE\_FR\_Coefficients.fK[6];  
 endif;  
 endif;   
 endif;  
 endif;  
 'end coefficient F1  
  
 'coefficient F2  
 if Me.fRCACFR < 30 then  
 Me.fFRF[2] = SE\_FR\_Coefficients.fK[7];  
 else  
 if (Me.fRCACFR >= 30) AND (Me.fRCACFR < SE\_FR\_Coefficients.fK[21]) then  
 Me.fFRF[2] = ((1 - SE\_FR\_Coefficients.fK[7]) \* (Me.fRCACFR - SE\_FR\_Coefficients.fK[21]) / (SE\_FR\_Coefficients.fK[21] - 30)) + 1;  
 else  
 if ((Me.fRCACFR >= SE\_FR\_Coefficients.fK[21]) AND (Me.fRCACFR < 90)) then  
 Me.fFRF[2] = ((SE\_FR\_Coefficients.fK[8] - 1) \* (Me.fRCACFR - SE\_FR\_Coefficients.fK[21]) / (90 - SE\_FR\_Coefficients.fK[21])) + 1;  
 else  
 Me.fFRF[2] = SE\_FR\_Coefficients.fK[8];  
 endif;  
 endif;  
 endif;  
 'end coefficient F2  
  
 'coefficient F3  
 if ((Me.fFRGTLC < (0 - SE\_FR\_Coefficients.fK[20])) OR   
 (Me.fFRGTLC >= (0 - SE\_FR\_Coefficients.fK[19]) AND (Me.fFRGTLC < SE\_FR\_Coefficients.fK[19])) OR  
 (Me.fFRGTLC >= SE\_FR\_Coefficients.fK[20])) then  
 Me.fFRF[3] = 1;  
 else  
 if ((Me.fFRGTLC >= (0 - SE\_FR\_Coefficients.fK[20])) AND (Me.fFRGTLC < ( 0 - SE\_FR\_Coefficients.fK[19]))) then  
 Me.fFRF[3] = ((1 - SE\_FR\_Coefficients.fK[10]) \* (Me.fFRGTLC + SE\_FR\_Coefficients.fK[19]) /   
 (SE\_FR\_Coefficients.fK[20] - SE\_FR\_Coefficients.fK[19])) + 1;  
 else  
 Me.fFRF[3] = ((SE\_FR\_Coefficients.fK[9] - 1) \* (Me.fFRGTLC - SE\_FR\_Coefficients.fK[19]) /   
 (SE\_FR\_Coefficients.fK[20] - SE\_FR\_Coefficients.fK[19])) + 1;  
 endif;  
 endif;  
 'end coefficient F3  
  
 'coefficient F4  
 if ((Me.fFRGTPB < (0 - SE\_FR\_Coefficients.fK[27])) OR   
 (Me.fFRGTPB >= (0 - SE\_FR\_Coefficients.fK[28]) AND (Me.fFRGTPB < SE\_FR\_Coefficients.fK[28])) OR  
 (Me.fFRGTPB >= SE\_FR\_Coefficients.fK[27])) then  
 Me.fFRF[4] = 1;  
 else  
 if ((Me.fFRGTPB >= (0 - SE\_FR\_Coefficients.fK[27])) AND (Me.fFRGTPB < ( 0 - SE\_FR\_Coefficients.fK[28]))) then  
 Me.fFRF[4] = ((1 - SE\_FR\_Coefficients.fK[26]) \* (Me.fFRGTPB + SE\_FR\_Coefficients.fK[28]) /   
 (SE\_FR\_Coefficients.fK[27] - SE\_FR\_Coefficients.fK[28])) + 1;  
 else  
 Me.fFRF[4] = ((SE\_FR\_Coefficients.fK[25] - 1) \* (Me.fFRGTPB - SE\_FR\_Coefficients.fK[28]) /   
 (SE\_FR\_Coefficients.fK[27] - SE\_FR\_Coefficients.fK[28])) + 1;  
 endif;  
 endif;  
 'end coefficient F4  
  
 'coefficient F5  
 if Me.fCO2210FR < 0.2 then  
 Me.fFRF[5] = SE\_FR\_Coefficients.fK[17];  
 else  
 if Me.fCO2210FR >= 0.2 AND Me.fCO2210FR < (SE\_FR\_Coefficients.fK[14] - SE\_FR\_Coefficients.fK[15]) then  
 Me.fFRF[5] = ((1 - SE\_FR\_Coefficients.fK[16] - SE\_FR\_Coefficients.fK[17])\*  
 (Me.fCO2210FR - 0.2)/(SE\_FR\_Coefficients.fK[14] - SE\_FR\_Coefficients.fK[15] - 0.2)) + SE\_FR\_Coefficients.fK[17];  
 else  
 if ((Me.fCO2210FR >= (SE\_FR\_Coefficients.fK[14] - SE\_FR\_Coefficients.fK[15])) AND   
 (Me.fCO2210FR < (SE\_FR\_Coefficients.fK[14] + SE\_FR\_Coefficients.fK[15]))) then  
 Me.fFRF[5] = (SE\_FR\_Coefficients.fK[16] \*  
 (Me.fCO2210FR - SE\_FR\_Coefficients.fK[14])/SE\_FR\_Coefficients.fK[15]) + 1;  
 else  
 if ((Me.fCO2210FR >= (SE\_FR\_Coefficients.fK[14] + SE\_FR\_Coefficients.fK[15])) AND (Me.fCO2210FR < 20)) then  
 Me.fFRF[5] = ((SE\_FR\_Coefficients.fK[18] - SE\_FR\_Coefficients.fK[16] - 1) \*  
 (Me.fCO2210FR - 20) / (20 - SE\_FR\_Coefficients.fK[14] - SE\_FR\_Coefficients.fK[15])) +   
 SE\_FR\_Coefficients.fK[18];  
 else  
 Me.fFRF[5] = SE\_FR\_Coefficients.fK[18];  
 endif;  
 endif;   
 endif;  
 endif;  
 'end coefficient F5  
  
 Me.fEVOL210FR = Me.fCO2210FR - Me.fCO2210FR\_OLD;  
  
 'coefficient F6  
 if Me.fEVOL210FR < (0 - SE\_FR\_Coefficients.fK[41]) then  
 Me.fFRF[6] = SE\_FR\_Coefficients.fK[39];  
 else  
 if (Me.fEVOL210FR >= (0 - SE\_FR\_Coefficients.fK[41])) AND (Me.fEVOL210FR < (0 - SE\_FR\_Coefficients.fK[42])) then  
 Me.fFRF[6] = ((Me.fEVOL210FR + SE\_FR\_Coefficients.fK[42]) \* (1 - SE\_FR\_Coefficients.fK[39])/(SE\_FR\_Coefficients.fK[41] - SE\_FR\_Coefficients.fK[42])) + 1;  
 else  
 if (Me.fEVOL210FR >= (0 - SE\_FR\_Coefficients.fK[42])) AND (Me.fEVOL210FR < SE\_FR\_Coefficients.fK[42]) then  
 Me.fFRF[6] = 1;  
 else  
 if (Me.fEVOL210FR >= SE\_FR\_Coefficients.fK[42]) AND (Me.fEVOL210FR < SE\_FR\_Coefficients.fK[43]) then  
 Me.fFRF[6] = ((Me.fEVOL210FR - SE\_FR\_Coefficients.fK[42]) \* (SE\_FR\_Coefficients.fK[40] - 1)/  
 (SE\_FR\_Coefficients.fK[43] - SE\_FR\_Coefficients.fK[42])) + 1;  
 else  
 Me.fFRF[6] = SE\_FR\_Coefficients.fK[40];  
 endif;  
 endif;  
 endif;  
 endif;   
 'coefficient F6  
  
 Me.fEVOL1040FR = Me.fCO21040FR - Me.fCO21040FR\_OLD;  
  
 'coefficient F6  
 if Me.fEVOL1040FR < (0 - SE\_FR\_Coefficients.fK[46]) then  
 Me.fFRF[7] = SE\_FR\_Coefficients.fK[44];  
 else  
 if (Me.fEVOL1040FR >= (0 - SE\_FR\_Coefficients.fK[46])) AND (Me.fEVOL1040FR < (0 - SE\_FR\_Coefficients.fK[47])) then  
 Me.fFRF[7] = ((Me.fEVOL1040FR + SE\_FR\_Coefficients.fK[47]) \* (1 - SE\_FR\_Coefficients.fK[44])/(SE\_FR\_Coefficients.fK[46] - SE\_FR\_Coefficients.fK[47])) + 1;  
 else  
 if (Me.fEVOL1040FR >= (0 - SE\_FR\_Coefficients.fK[47])) AND (Me.fEVOL1040FR < SE\_FR\_Coefficients.fK[47]) then  
 Me.fFRF[7] = 1;  
 else  
 if (Me.fEVOL1040FR >= SE\_FR\_Coefficients.fK[47]) AND (Me.fEVOL1040FR < SE\_FR\_Coefficients.fK[48]) then  
 Me.fFRF[7] = ((Me.fEVOL1040FR - SE\_FR\_Coefficients.fK[47]) \* (SE\_FR\_Coefficients.fK[45] - 1)/  
 (SE\_FR\_Coefficients.fK[48] - SE\_FR\_Coefficients.fK[47])) + 1;  
 else  
 Me.fFRF[7] = SE\_FR\_Coefficients.fK[45];  
 endif;  
 endif;  
 endif;  
 endif;   
 'coefficient F7  
  
 'FR4 - Utilisation de la reactivité, pour moduler les fonctions calculées  
 if StringToIntg(StringRight(Me.Tagname,1)) == 4 then  
 if Me.fRCACFR >= 90 then  
 Me.fFRF[5] = 1;  
 if Me.fFRF[3] < 1 then  
 Me.fFRF[3] = 1;  
 endif;  
 if Me.fFRF[4] < 1 then  
 Me.fFRF[4] = 1;  
 endif;  
 else  
 if Me.fRCACFR >= 70 then  
 Me.fFRF[5] = 1;  
 if Me.fFRF[3] < 1 then  
 Me.fFRF[3] = 0.5 + (0.5 \* Me.fFRF[3]);  
 endif;  
 if Me.fFRF[4] < 1 then  
 Me.fFRF[4] = 0.5 + (0.5 \* Me.fFRF[4]);  
 endif;  
 else  
 if Me.fRCACFR >= 50 then  
 Me.fFRF[5] = 1;  
 endif;  
 endif;  
 endif;   
 endif;  
 'FR4 - Fin de modulation des fonctions  
  
 Me.fBuf = 856000 \* Me.fFR\_QP \* ((Me.fFRF[1] \* Me.fFRF[2] \* Me.fFRF[3] \* Me.fFRF[4] \* Me.fFRF[5] \* Me.fFRF[6] \* Me.fFRF[7]) - 1);  
  
'end FR CHAUX  
'endif; ' if Me.FR\_PROD\_CONS == 1 (CHAUX)  
  
'trigger the corrections  
Me.bCorrection = True;  
  
if Me.bDebug == True then  
 LogMessage("End calcChaux");  
endif;

calcCorrections

|  |  |
| --- | --- |
| Name | calcCorrections |
| Description |  |
| Trigger | WhileTrue of Me.bCorrection |

**Declarations :**

Not Applicable

**Script :**

'script that calculates the corrections for Lignite, Gaz, Coke and Hasler  
  
'reset trigger  
Me.bCorrection = false;  
  
if Me.bDebug == True then  
 LogMessage("Start calcCorrections");  
endif;  
  
Me.bChauxFR123 = False;  
  
Me.fCORLIGFR = Me.fBuf/5150;  
Me.fCORCOKFR = Me.fBuf/8150;  
Me.fCORGAZFR = Me.fBuf/8800;  
  
'FR DOLIME or DOLO  
if Me.FR\_PROD\_CONS == 4 OR Me.FR\_PROD\_CONS == 2 then   
 'DOLIME or DOLO  
 Me.fCORHASFR = 0 - (Me.fBuf/901600);  
else  
 'FR CHAUX  
 if Me.FR\_PROD\_CONS == 1 then  
 'CHAUX  
 Me.fCORHASFR = 0 - (Me.fBuf/856000);  
 if StringToIntg(StringRight(Me.Tagname,1)) <> 4 then  
 Me.bChauxFR123 = True;  
 endif;  
 endif;  
endif;  
  
if Me.bChauxFR123 == False then  
 if ((Me.fCORHASFR > (0 - 5.5)) AND (Me.fCORHASFR <= (0 - 4.5))) then  
 Me.fCORHASFR = 0 - 5;  
 else  
 if ((Me.fCORHASFR > (0 - 4.5)) AND (Me.fCORHASFR <= (0 - 3.5))) then  
 Me.fCORHASFR = 0 - 4;  
 else  
 if ((Me.fCORHASFR > (0 - 3.5)) AND (Me.fCORHASFR <= (0 - 2.5))) then  
 Me.fCORHASFR = 0 - 3;  
 else  
 if ((Me.fCORHASFR > (0 - 2.5)) AND (Me.fCORHASFR <= (0 - 1.5))) then  
 Me.fCORHASFR = 0 - 2;  
 else  
 if ((Me.fCORHASFR > (0 - 1.5)) AND (Me.fCORHASFR <= (0 - 0.5))) then  
 Me.fCORHASFR = 0 - 1;  
 else  
 if ((Me.fCORHASFR > (0 - 0.5)) AND (Me.fCORHASFR <= 0.5)) then  
 Me.fCORHASFR = 0;  
 else  
 if ((Me.fCORHASFR > 0.5) AND (Me.fCORHASFR <= 1.5)) then  
 Me.fCORHASFR = 1;  
 else  
 if ((Me.fCORHASFR > 1.5) AND (Me.fCORHASFR <= 2.5)) then  
 Me.fCORHASFR = 2;  
 else  
 if ((Me.fCORHASFR > 2.5) AND (Me.fCORHASFR <= 3.5)) then  
 Me.fCORHASFR = 3;  
 else  
 if ((Me.fCORHASFR > 3.5) AND (Me.fCORHASFR <= 4.5)) then  
 Me.fCORHASFR = 4;  
 else  
 if ((Me.fCORHASFR > 4.5) AND (Me.fCORHASFR <= 5.5)) then  
 Me.fCORHASFR = 5;  
 endif;  
 endif;  
 endif;  
 endif;  
 endif;  
 endif;  
 endif;  
 endif;  
 endif;  
 endif;  
 endif;   
endif;  
  
if Me.bDebug == True then  
 LogMessage("End calcCorrections");  
endif;

calcDolo

|  |  |
| --- | --- |
| Name | calcDolo |
| Description |  |
| Trigger | WhileTrue of Me.bCalcDolo |

**Declarations :**

Not Applicable

**Script :**

'script that does the calculations for dolo  
  
'reset trigger  
Me.bCalcDolo = false;  
  
if Me.bDebug == True then  
 LogMessage("Start calcDolo");  
endif;  
  
'FR DOLO  
'if Me.FR\_PROD\_CONS == 2 then  
if StringToIntg(StringRight(Me.Tagname,1)) <> 4 then  
 'FR DOLO 1, 2, 3  
 if (Me.fFRGTLC < (0 - SE\_FR\_Coefficients.fK[34])) OR ((Me.fFRGTLC >= (0 - SE\_FR\_Coefficients.fK[33])) AND  
 (Me.fFRGTLC < SE\_FR\_Coefficients.fK[33])) OR (Me.fFRGTLC >= SE\_FR\_Coefficients.fK[34]) then  
 'coefficient L  
 Me.fFRL = 1;  
 else  
 if Me.fFRGTLC >= (0 - SE\_FR\_Coefficients.fK[34]) AND   
 Me.fFRGTLC < (0 - SE\_FR\_Coefficients.fK[33])then  
 Me.fFRL = ((1 - SE\_FR\_Coefficients.fK[4])\*(Me.fFRGTLC + SE\_FR\_Coefficients.fK[33])/(SE\_FR\_Coefficients.fK[34] - SE\_FR\_Coefficients.fK[33])) + 1;  
 else  
 Me.fFRL = ((SE\_FR\_Coefficients.fK[3] - 1)\*(Me.fFRGTLC - SE\_FR\_Coefficients.fK[33])/(SE\_FR\_Coefficients.fK[34] - SE\_FR\_Coefficients.fK[33])) + 1;  
 endif;  
 endif;  
 'end coefficient L  
  
 Me.fEVOLRMAGFR = Me.fRMAGFR - Me.fRMAGFR\_OLD;  
   
 'coefficient R  
 if Me.fEVOLRMAGFR < (0 - SE\_FR\_Coefficients.fK[56]) then  
 Me.fFRR = SE\_FR\_Coefficients.fK[55];  
 else  
 if (Me.fEVOLRMAGFR >= (0 - SE\_FR\_Coefficients.fK[56])) AND (Me.fEVOLRMAGFR < (0 - SE\_FR\_Coefficients.fK[57])) then  
 Me.fFRR = ((Me.fEVOLRMAGFR + SE\_FR\_Coefficients.fK[57]) \* (SE\_FR\_Coefficients.fK[55] - 1)/(SE\_FR\_Coefficients.fK[57] - SE\_FR\_Coefficients.fK[56])) + 1;  
 else  
 if (Me.fEVOLRMAGFR >= (0 - SE\_FR\_Coefficients.fK[57])) AND (Me.fEVOLRMAGFR < SE\_FR\_Coefficients.fK[57]) then  
 Me.fFRR = 1;  
 else  
 if (Me.fEVOLRMAGFR >= SE\_FR\_Coefficients.fK[57]) AND (Me.fEVOLRMAGFR < SE\_FR\_Coefficients.fK[58]) then  
 Me.fFRR = ((Me.fEVOLRMAGFR - SE\_FR\_Coefficients.fK[57]) \* (1 - SE\_FR\_Coefficients.fK[54])/  
 (SE\_FR\_Coefficients.fK[57] - SE\_FR\_Coefficients.fK[58])) + 1;  
 else  
 Me.fFRR = SE\_FR\_Coefficients.fK[54];  
 endif;  
 endif;  
 endif;  
 endif;   
 'coefficient R  
  
 Me.fFRN = ((SE\_FR\_Coefficients.fA[4] \* Me.fRMAGFR) + SE\_FR\_Coefficients.fB[4]);  
 Me.fFRM = ((SE\_FR\_Coefficients.fA[5] \* Me.fRMAGFR) + SE\_FR\_Coefficients.fB[5]);  
  
 if ((Me.fFRGTPB < (0 - SE\_FR\_Coefficients.fK[38])) OR ((Me.fFRGTPB >= (0 - SE\_FR\_Coefficients.fK[37])) AND  
 (Me.fFRGTPB < SE\_FR\_Coefficients.fK[37])) OR (Me.fFRGTPB >= SE\_FR\_Coefficients.fK[38])) then  
 'coefficient P  
 Me.fFRP = 1;  
 else  
 if Me.fFRGTPB >= (0 - SE\_FR\_Coefficients.fK[38]) AND   
 Me.fFRGTPB < (0 - SE\_FR\_Coefficients.fK[37])then  
 Me.fFRP = ((1 - SE\_FR\_Coefficients.fK[36])\*(Me.fFRGTPB + SE\_FR\_Coefficients.fK[37])/(SE\_FR\_Coefficients.fK[38] - SE\_FR\_Coefficients.fK[37])) + 1;  
 else  
 Me.fFRP = ((SE\_FR\_Coefficients.fK[35] - 1)\*(Me.fFRGTPB - SE\_FR\_Coefficients.fK[37])/(SE\_FR\_Coefficients.fK[38] - SE\_FR\_Coefficients.fK[37])) + 1;  
 endif;  
 endif;  
  
 if (Me.fRMAGFR >= SE\_FR\_Coefficients.fZ[4]) AND (Me.fRMAGFR <= SE\_FR\_Coefficients.fZ[5]) then  
 Me.fBuf = ((Me.fFRL \* Me.fFRM \* Me.fFRP \* Me.fFRR) - 1) \* 901600 \* Me.fFR\_QP;  
 else  
 if (Me.fRMAGFR > SE\_FR\_Coefficients.fZ[5]) AND (Me.fRMAGFR <= SE\_FR\_Coefficients.fZ[6]) then  
 Me.fBuf = ((Me.fFRL \* Me.fFRN \* Me.fFRP \* Me.fFRR) - 1) \* 901600 \* Me.fFR\_QP;  
 endif;  
 endif;  
 'end FR DOLO 1, 2, 3  
 else  
 'FR DOLO 4  
 if (Me.fFRGTLC < (0 - SE\_FR\_Coefficients.fF4K34)) OR ((Me.fFRGTLC >= (0 - SE\_FR\_Coefficients.fF4K33)) AND  
 (Me.fFRGTLC < SE\_FR\_Coefficients.fF4K33)) OR (Me.fFRGTLC >= SE\_FR\_Coefficients.fF4K34) then  
 'coefficient L  
 Me.fFRL = 1;  
 else  
 if Me.fFRGTLC >= (0 - SE\_FR\_Coefficients.fF4K34) AND   
 Me.fFRGTLC < (0 - SE\_FR\_Coefficients.fF4K33)then  
 Me.fFRL = ((1 - SE\_FR\_Coefficients.fF4K4)\*(Me.fFRGTLC + SE\_FR\_Coefficients.fF4K33)/(SE\_FR\_Coefficients.fF4K34 - SE\_FR\_Coefficients.fF4K33)) + 1;  
 else  
 Me.fFRL = ((SE\_FR\_Coefficients.fF4K3 - 1)\*(Me.fFRGTLC - SE\_FR\_Coefficients.fF4K33)/(SE\_FR\_Coefficients.fF4K34 - SE\_FR\_Coefficients.fF4K33)) + 1;  
 endif;  
 endif;  
 'end coefficient L  
  
 Me.fEVOLRMAGFR = Me.fRMAGFR - Me.fRMAGFR\_OLD;  
   
 'coefficient R  
 if Me.fEVOLRMAGFR < (0 - SE\_FR\_Coefficients.fF4K56) then  
 Me.fFRR = SE\_FR\_Coefficients.fF4K55;  
 else  
 if (Me.fEVOLRMAGFR >= (0 - SE\_FR\_Coefficients.fF4K56)) AND (Me.fEVOLRMAGFR < (0 - SE\_FR\_Coefficients.fF4K57)) then  
 Me.fFRR = ((Me.fEVOLRMAGFR + SE\_FR\_Coefficients.fF4K57) \* (SE\_FR\_Coefficients.fF4K55 - 1)/(SE\_FR\_Coefficients.fF4K57 - SE\_FR\_Coefficients.fF4K56)) + 1;  
 else  
 if (Me.fEVOLRMAGFR >= (0 - SE\_FR\_Coefficients.fF4K57)) AND (Me.fEVOLRMAGFR < SE\_FR\_Coefficients.fF4K57) then  
 Me.fFRR = 1;  
 else  
 if (Me.fEVOLRMAGFR >= SE\_FR\_Coefficients.fF4K57) AND (Me.fEVOLRMAGFR < SE\_FR\_Coefficients.fF4K58) then  
 Me.fFRR = ((Me.fEVOLRMAGFR - SE\_FR\_Coefficients.fF4K57) \* (1 - SE\_FR\_Coefficients.fF4K54)/  
 (SE\_FR\_Coefficients.fF4K57 - SE\_FR\_Coefficients.fF4K58)) + 1;  
 else  
 Me.fFRR = SE\_FR\_Coefficients.fF4K54;  
 endif;  
 endif;  
 endif;  
 endif;   
 'coefficient R  
  
 Me.fFRN = ((SE\_FR\_Coefficients.fF4A4 \* Me.fRMAGFR) + SE\_FR\_Coefficients.fF4B4);  
 Me.fFRM = ((SE\_FR\_Coefficients.fF4A5 \* Me.fRMAGFR) + SE\_FR\_Coefficients.fF4B5);  
  
 if ((Me.fFRGTPB < (0 - SE\_FR\_Coefficients.fF4K38)) OR ((Me.fFRGTPB >= (0 - SE\_FR\_Coefficients.fF4K37)) AND  
 (Me.fFRGTPB < SE\_FR\_Coefficients.fF4K37)) OR (Me.fFRGTPB >= SE\_FR\_Coefficients.fF4K38)) then  
 'coefficient P  
 Me.fFRP = 1;  
 else  
 if Me.fFRGTPB >= (0 - SE\_FR\_Coefficients.fF4K38) AND   
 Me.fFRGTPB < (0 - SE\_FR\_Coefficients.fF4K37)then  
 Me.fFRP = ((1 - SE\_FR\_Coefficients.fF4K36)\*(Me.fFRGTPB + SE\_FR\_Coefficients.fF4K37)/(SE\_FR\_Coefficients.fF4K38 - SE\_FR\_Coefficients.fF4K37)) + 1;  
 else  
 Me.fFRP = ((SE\_FR\_Coefficients.fF4K35 - 1)\*(Me.fFRGTPB - SE\_FR\_Coefficients.fF4K37)/(SE\_FR\_Coefficients.fF4K38 - SE\_FR\_Coefficients.fF4K37)) + 1;  
 endif;  
 endif;  
  
 if (Me.fRMAGFR >= SE\_FR\_Coefficients.fF4Z4) AND (Me.fRMAGFR <= SE\_FR\_Coefficients.fF4Z5) then  
 Me.fBuf = ((Me.fFRL \* Me.fFRM \* Me.fFRP \* Me.fFRR) - 1) \* 901600 \* Me.fFR\_QP;  
 else  
 if (Me.fRMAGFR >= SE\_FR\_Coefficients.fF4Z5) AND (Me.fRMAGFR <= SE\_FR\_Coefficients.fF4Z6) then  
 Me.fBuf = ((Me.fFRL \* Me.fFRN \* Me.fFRP \* Me.fFRR) - 1) \* 901600 \* Me.fFR\_QP;  
 endif;  
 endif;  
 'end FR DOLO 4  
endif;  
'endif;  
  
  
'trigger the corrections  
Me.bCorrection = True;  
  
if Me.bDebug == True then  
 LogMessage("End calcMagnesiaDolo");  
endif;

#### Template ArchestrA $dwSuggesteurConsigneFR\_Coef

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| fA | float |  |  |  |  |  |
| fB | float |  |  |  |  |  |
| fF4A4 | float |  |  |  |  |  |
| fF4A5 | float |  |  |  |  |  |
| fF4B4 | float |  |  |  |  |  |
| fF4B5 | float |  |  |  |  |  |
| fF4K3 | float |  |  |  |  |  |
| fF4K33 | float |  |  |  |  |  |
| fF4K34 | float |  |  |  |  |  |
| fF4K35 | float |  |  |  |  |  |
| fF4K36 | float |  |  |  |  |  |
| fF4K37 | float |  |  |  |  |  |
| fF4K38 | float |  |  |  |  |  |
| fF4K4 | float |  |  |  |  |  |
| fF4K54 | float |  |  |  |  |  |
| fF4K55 | float |  |  |  |  |  |
| fF4K56 | float |  |  |  |  |  |
| fF4K57 | float |  |  |  |  |  |
| fF4K58 | float |  |  |  |  |  |
| fF4Z4 | float |  |  |  |  |  |
| fF4Z5 | float |  |  |  |  |  |
| fF4Z6 | float |  |  |  |  |  |
| fK | float |  |  |  |  |  |
| fZ | float |  |  |  |  |  |
| iTPSREG | integer |  |  |  |  |  |
| iTPSREG\_TPB | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwSynchroPlc

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| iHeure | integer |  |  | X |  |  |
| iMinute | integer |  |  | X |  |  |
| iSeconde | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcHeure

|  |  |
| --- | --- |
| Name | dcHeure |
| Description |  |
| Trigger | DataChange of Now().Hour |

**Declarations :**

Not Applicable

**Script :**

Me.iHeure = Now().Hour;

dcMinute

|  |  |
| --- | --- |
| Name | dcMinute |
| Description |  |
| Trigger | DataChange of Now().Minute |

**Declarations :**

Not Applicable

**Script :**

Me.iMinute = Now().Minute;

dcsecond

|  |  |
| --- | --- |
| Name | dcsecond |
| Description |  |
| Trigger | DataChange of Now().second |

**Declarations :**

Not Applicable

**Script :**

Me.iSeconde=Now().second;

#### Template ArchestrA $dwSystemExpert

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| \_Config.SQL.Database | string |  |  |  |  |  |
| \_Config.SQL.Host | string |  |  |  |  |  |
| \_Config.SQL.Pass | string |  |  |  |  |  |
| \_Config.SQL.User | string |  |  |  |  |  |
| \_Config.XML.Directory | string |  |  |  |  |  |
| bCreateDataSets | boolean |  |  |  |  |  |
| bDebug | boolean |  |  |  |  |  |
| bDeploy | boolean |  |  |  |  |  |
| bInitDone | boolean |  |  |  |  |  |
| bMatrElemExist | boolean |  |  |  |  |  |
| bModeExpert | boolean |  | X |  |  |  |
| bNouveauCycle | boolean |  |  |  |  |  |
| bPC\_EXPERT | boolean |  |  |  |  |  |
| bTest | boolean |  |  |  |  |  |
| bTestMatrice | boolean |  |  |  |  |  |
| bXMLReadConnexionLost | boolean |  |  |  | X |  |
| bXMLWriteConnexionLost | boolean |  |  |  | X |  |
| Const.iAD\_ETAT\_EXPERT | integer |  |  |  |  |  |
| Const.iCS\_MAX\_TO\_MAT | integer |  |  |  |  |  |
| Const.iCS\_MIN\_TO\_MAT | integer |  |  |  |  |  |
| Const.iEXPERT\_ON | integer |  |  |  |  |  |
| Const.iHeureConsigne | integer |  |  |  |  |  |
| Const.iMinConsigne | integer |  |  |  |  |  |
| Const.iNBRE\_COND | integer |  |  |  |  |  |
| Const.iNBRE\_LIGNE | integer |  |  |  |  |  |
| fCoeffDecarbo | float |  | X |  |  |  |
| fConsignePierre\_C1 | float |  | X |  |  |  |
| fConsignePierre\_C2 | float |  | X |  |  |  |
| fConsignePierre\_C3 | float |  | X |  |  |  |
| fConsignePierreMoyenne | float |  |  |  |  |  |
| fCS\_Max | float |  |  |  |  |  |
| fCS\_Min | float |  |  |  |  |  |
| fkIP | double |  |  |  |  |  |
| fKIP\_1 | double |  | X |  |  | X |
| fKIP\_10 | double |  | X |  |  | X |
| fKIP\_11 | double |  | X |  |  | X |
| fKIP\_12 | double |  | X |  |  | X |
| fKIP\_13 | double |  | X |  |  | X |
| fKIP\_14 | double |  | X |  |  | X |
| fKIP\_15 | double |  | X |  |  | X |
| fKIP\_16 | double |  | X |  |  | X |
| fKIP\_17 | double |  | X |  |  | X |
| fKIP\_18 | double |  | X |  |  | X |
| fKIP\_19 | double |  | X |  |  | X |
| fKIP\_20 | double |  | X |  |  | X |
| fKIP\_21 | double |  | X |  |  | X |
| fKIP\_22 | double |  | X |  |  | X |
| fKIP\_23 | double |  | X |  |  | X |
| fKIP\_24 | double |  | X |  |  | X |
| fKIP\_25 | double |  | X |  |  | X |
| fKIP\_26 | double |  | X |  |  | X |
| fKIP\_27 | double |  | X |  |  | X |
| fKIP\_28 | double |  | X |  |  | X |
| fKIP\_29 | double |  | X |  |  | X |
| fKIP\_3 | double |  | X |  |  | X |
| fKIP\_30 | double |  | X |  |  | X |
| fKIP\_4 | double |  | X |  |  | X |
| fKIP\_5 | double |  | X |  |  | X |
| fKIP\_6 | double |  | X |  |  | X |
| fKIP\_7 | double |  | X |  |  | X |
| fKIP\_8 | double |  | X |  |  | X |
| fKIP\_9 | double |  | X |  |  | X |
| fKIP\_90 | double |  | X |  |  | X |
| fLIM\_B | double |  |  |  |  |  |
| fLIM\_H | double |  |  |  |  |  |
| fPCIGaz | float |  | X |  |  |  |
| fPCILignite | float |  | X |  |  |  |
| fPoidsMaxGaz | float |  |  |  |  | X |
| fPoidsMaxLignite | float |  |  |  |  | X |
| fPoidsMinGaz | float |  |  |  |  | X |
| fPoidsMinLignite | float |  |  |  |  | X |
| fTCR | double |  |  |  |  | X |
| iConsigneTpsCycle | integer |  | X |  |  |  |
| iDConsSpecGaz | float |  |  |  |  |  |
| iDConsSpecLign | float |  |  |  |  |  |
| iFM\_V | integer |  |  |  |  |  |
| iIndiceActuel | integer |  |  |  |  |  |
| iInitCnt | integer |  |  |  |  |  |
| iNUM\_FOUR | integer |  |  |  |  |  |
| iSTATUS | integer |  |  |  |  |  |
| Object.Script.XMLDSMatriceRead | boolean |  |  |  |  |  |
| Object.Script.XMLDSMatriceWrite | boolean |  |  |  |  |  |
| Object.Script.XMLDSParamRead | boolean |  |  |  |  |  |
| Object.Script.XMLDSParamWrite | boolean |  |  |  |  |  |
| sDate | string |  |  |  |  |  |
| sHour | string |  |  |  |  |  |
| SQL.Query.ConnectString | string |  |  |  |  |  |
| strReportCreationDate | string |  |  |  |  |  |

##### Field Attributes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Attribute name | Data Type | IO Scaling | History | Limit Alarms | Rate Of Change Alarms | Target Deviation Alarms | Bad Value Alarm | Statistics |
| Combustible\_Selected | boolean |  |  |  |  |  |  |  |
| STATUS | float |  | X |  |  |  |  |  |
| TCR | float |  | X |  |  |  |  |  |
| LIGNITE | integer |  | X |  |  |  |  |  |
| GAZ | integer |  | X |  |  |  |  |  |
| ALVE | float |  | X |  |  |  |  |  |
| FM\_V1000 | integer |  |  |  |  |  |  |  |
| FM\_V1001 | integer |  | X |  |  |  |  |  |
| FM\_V1002 | integer |  |  |  |  |  |  |  |
| FM\_V1003 | integer |  |  |  |  |  |  |  |
| FM\_V1004 | integer |  |  |  |  |  |  |  |
| FM\_V1005 | integer |  |  |  |  |  |  |  |
| FM\_V1006 | integer |  |  |  |  |  |  |  |
| FM\_V1007 | integer |  |  |  |  |  |  |  |
| FM\_V1013 | integer |  |  |  |  |  |  |  |
| FM\_V1014 | integer |  |  |  |  |  |  |  |
| FM\_V1015 | integer |  |  |  |  |  |  |  |
| FM\_V1016 | integer |  |  |  |  |  |  |  |
| FM\_V1017 | integer |  |  |  |  |  |  |  |
| FM\_V1018 | integer |  |  |  |  |  |  |  |
| FM\_V1019 | integer |  |  |  |  |  |  |  |
| FM\_V1008 | integer |  |  |  |  |  |  |  |
| FM\_V1009 | integer |  |  |  |  |  |  |  |
| FM\_V1010 | integer |  |  |  |  |  |  |  |
| FM\_V1011 | integer |  |  |  |  |  |  |  |
| FM\_V1012 | integer |  |  |  |  |  |  |  |

##### Scripts

dcCycle

|  |  |
| --- | --- |
| Name | dcCycle |
| Description |  |
| Trigger | DataChange of Me.FM\_V1000 |

**Declarations :**

Not Applicable

**Script :**

'script that executes when the cycle number changed  
  
'if initialisation not done  
if Me.bInitDone == True then  
  
'copy locally all PLC values for the matrix (FM\_V1000 -> FM\_V1019)  
Me.iFM\_V[1] = Me.FM\_V1000;  
'Me.iFM\_V[2] = Me.FM\_V1001;  
'Me.iFM\_V[3] = Me.FM\_V1002;  
'Me.iFM\_V[4] = Me.FM\_V1003;  
'Me.iFM\_V[5] = Me.FM\_V1004;  
'Me.iFM\_V[6] = Me.FM\_V1005;  
'Me.iFM\_V[7] = Me.FM\_V1006;  
'Me.iFM\_V[8] = Me.FM\_V1007;  
'Me.iFM\_V[9] = Me.FM\_V1008;  
'Me.iFM\_V[10] = Me.FM\_V1009;  
'Me.iFM\_V[11] = Me.FM\_V1010;  
'Me.iFM\_V[12] = Me.FM\_V1011;  
'Me.iFM\_V[13] = Me.FM\_V1012;  
'Me.iFM\_V[14] = Me.FM\_V1013;  
'Me.iFM\_V[15] = Me.FM\_V1014;  
'Me.iFM\_V[16] = Me.FM\_V1015;  
'Me.iFM\_V[17] = Me.FM\_V1016;  
'Me.iFM\_V[18] = Me.FM\_V1017;  
'Me.iFM\_V[19] = Me.FM\_V1018;  
'Me.iFM\_V[20] = Me.FM\_V1019;  
  
'if Me.iFM\_V[6] >= Me.Const.iCS\_MIN\_TO\_MAT AND Me.iFM\_V[6] <= Me.Const.iCS\_MAX\_TO\_MAT then  
if Me.FM\_V1005 >= Me.Const.iCS\_MIN\_TO\_MAT AND Me.FM\_V1005 <= Me.Const.iCS\_MAX\_TO\_MAT then  
 'new cycle  
 Me.bNouveauCycle = 1;  
 'calculate the new set points  
 Me.dw\_SE\_Conduite.bCalcConduite = 1;  
endif;  
  
endif;

ObjectDeploy

|  |  |
| --- | --- |
| Name | ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of me.bDeploy |

**Declarations :**

Not Applicable

**Script :**

' create data set for reading the kIP coefficients from XML file  
' create data set for reading the matrix from XML file  
  
if me.iInitCnt < 5 then  
 me.iInitCnt = me.iInitCnt + 1;  
else  
 me.bDeploy = false;  
 'initializion not done  
 Me.bInitDone = False;  
 if Me.bDebug == True then  
 LogMessage("Initialisation NOT DONE");  
 endif;  
   
 'trigger the creation of datasets  
 Me.bCreateDataSets = True;  
endif;

XMLDSParamRead

|  |  |
| --- | --- |
| Name | XMLDSParamRead |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.XMLDSParamRead |

**Declarations :**

Not Applicable

**Script :**

' script that read the kIP coefficients XML file to DataSet  
' read kIP values  
  
Dim MyDataSet as System.Data.DataSet;  
Dim Parameters as System.Data.DataTable;  
Dim FileName as String;  
Dim i as integer;  
Dim foundRows[1] as System.Data.DataRow;  
Dim MyDataRow as System.Data.DataRow;  
Dim NumRecords as integer;  
  
if Me.bDebug == True then  
 LogMessage("Read kIP xml file");  
endif;  
  
' reset trigger  
Me.Object.Script.XMLDSParamRead = False;  
  
if Me.bInitDone == False then  
 Me.bInitDone = True;  
 if Me.bDebug == True then  
 LogMessage("Initialisation DONE");  
 endif;  
endif;  
  
' retreive DataSet for kIP parameters from AppDomain  
FileName = Me.\_Config.XML.Directory + Me.TagName + "\_kIP\_DS.xml";  
if System.IO.File.Exists(FileName) == True then  
 Me.bXMLReadConnexionLost = False;  
  
 MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_Param\_" + Me.Tagname);  
 if MyDataset <> null then  
 ' add data to MyDataSet  
 Parameters = MyDataSet.Tables("Parameters");  
 ' clear result Parameters  
 Parameters.Clear();  
 ' read XML file  
 MyDataSet.ReadXml(FileName, System.Data.XmlReadMode.IgnoreSchema);  
   
 ' retreive the read data  
 foundRows[] = Parameters.Select();  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
   
 ' read the kIP values  
 i = 1;  
 for i = 1 to NumRecords  
 MyDataRow = foundRows[i];  
 Me.fkIP[i] = MyDataRow("Param\_Value");  
 next;  
 endif;   
else  
 Me.bXMLReadConnexionLost = True;  
 if Me.bDebug == True then  
 LogMessage("XML Parameters File does not exist");  
 endif;  
endif;

XMLDSParamWrite

|  |  |
| --- | --- |
| Name | XMLDSParamWrite |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.XMLDSParamWrite |

**Declarations :**

Not Applicable

**Script :**

' script that write kIP coefficients XML file from DataSet  
  
Dim MyDataSet as System.Data.DataSet;  
Dim FileName as string;  
  
if Me.bDebug == True then  
 LogMessage("Write kIP xml file");  
endif;  
  
' reset trigger  
Me.Object.Script.XMLDSParamWrite = False;  
  
' retreive MyDataSet from AppDomain  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_Param\_" + Me.Tagname);  
if MyDataset <> null then  
 ' write file  
 FileName = Me.\_Config.XML.Directory + Me.TagName + "\_kIP\_DS.xml";  
   
 if System.IO.File.Exists(FileName) == True then  
 MyDataSet.WriteXml(FileName, System.Data.XmlWriteMode.IgnoreSchema);  
 Me.bXMLWriteConnexionLost = False;  
 else  
 Me.bXMLWriteConnexionLost = True;  
 if Me.bDebug == True then  
 LogMessage("Parameters XML File does not exist");  
 endif;  
 endif;  
endif;

EnregistreIndiceActuel

|  |  |
| --- | --- |
| Name | EnregistreIndiceActuel |
| Description |  |
| Trigger | WhileTrue of Me.bNouveauCycle == 1 |

**Declarations :**

Not Applicable

**Script :**

'script that read the current index from XML file  
'then calculate the new index and add a new record in the XMl file  
  
Dim MyDataSet as System.Data.DataSet;  
Dim Matrice as System.Data.DataTable;  
Dim MyDataRow as System.Data.DataRow;  
Dim foundRows[1] as System.Data.DataRow;  
Dim NumRecords as integer;  
Dim Expression as string;  
Dim Sort as string;  
Dim i as integer;  
Dim fSECONDS as double;  
Dim sTemp as string;  
Dim day as string;  
Dim month as string;  
Dim hour as string;  
Dim min as string;  
Dim sec as string;  
dim inbrindex as integer;  
  
'reset trigger  
Me.bNouveauCycle = 0;  
  
' retreive MyDataSet from AppDomain  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_Matrice\_" + Me.Tagname);  
  
if MyDataset <> null then  
 Matrice = MyDataSet.Tables("Matrice");  
  
 'if xml file doesn'n exist => matrix dataset is empty  
 if (Me.iIndiceActuel == -1) OR (Me.iIndiceActuel == 0) then  
 Me.iIndiceActuel = 1;  
 else  
 'get the last added index  
 ' init expression  
 Expression = "";  
 ' query  
 Expression = "Indice > 0";  
 ' query - order by  
 Sort = "DateTime\_SEC Desc";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression,Sort);  
   
 ' MODIF EXPERT AND 18/01/2012  
 if (foundRows[].GetUpperBound(0) + 1) > 0 then  
 MyDataRow = foundRows[1];  
   
 Me.iIndiceActuel = MydataRow("Indice");  
 if Me.bDebug == True then  
 LogMessage("IndiceActuel = " + Text(Me.iIndiceActuel, "#"));  
 endif;  
   
 'calculate the new index  
 Me.iIndiceActuel = Me.iIndiceActuel + 1;  
 if Me.iIndiceActuel > Me.Const.iNBRE\_LIGNE then  
 Me.iIndiceActuel = 1;  
 endif;  
 if Me.bDebug == True then  
 LogMessage("Nouveau IndiceActuel = " + Text(Me.iIndiceActuel, "#"));  
 endif;  
   
 'if the index 400 exists in the table then delete the row corresponding to the new index   
 '(the matrix contains maximum 400 rows)  
   
 Expression = "";  
 Expression = "Indice = 400";  
 foundRows[] = Matrice.Select(Expression);  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
 if NumRecords > 0 then  
 if Me.bDebug == True then  
 LogMessage("Index 400 exists");  
 endif;  
 Expression = "";  
 Expression = "Indice = " + Me.iIndiceActuel;  
 foundRows[] = Matrice.Select(Expression);  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
'GPR 19/12/2012 : ajout de la boucle qui permet d'effacer plus d'un record existant dans les fichiers xml  
 if Me.bDebug == True then  
 logmessage("nbr records avec même index dans fichier XML : " + NumRecords );  
 endif;  
  
 if NumRecords > 0 then  
 inbrindex =0;  
 while inbrindex <= NumRecords -1  
  
 Expression = "";  
 Expression = "Indice = " + Me.iIndiceActuel;  
 foundRows[] = Matrice.Select(Expression);  
 foundRows[1].Delete();   
  
 if Me.bDebug == True then  
  
 LogMessage("Index old Row deleted");  
 endif;  
  
 inbrindex = inbrindex +1;  
 endwhile;  
   
   
 endif;  
 endif;  
 endif;  
 endif;  
  
 'add to DataSet the record for the new index  
   
 ' add new row  
 MyDataRow = Matrice.NewRow();  
   
 ' add a line to Matrice for the new index  
 MyDataRow("Indice") = Me.iIndiceActuel;  
 MyDataRow("FM\_V1000") = Me.FM\_V1000;  
 MyDataRow("FM\_V1001") = Me.FM\_V1001;  
 MyDataRow("FM\_V1002") = Me.FM\_V1002;  
 MyDataRow("FM\_V1003") = Me.FM\_V1003;  
 MyDataRow("FM\_V1004") = Me.FM\_V1004;  
 MyDataRow("FM\_V1005") = Me.FM\_V1005;  
 MyDataRow("FM\_V1006") = Me.FM\_V1006;  
 MyDataRow("FM\_V1007") = Me.FM\_V1007;  
 MyDataRow("FM\_V1008") = Me.FM\_V1008;  
 MyDataRow("FM\_V1009") = Me.FM\_V1009;  
 MyDataRow("FM\_V1010") = Me.FM\_V1010;  
 MyDataRow("FM\_V1011") = Me.FM\_V1011;  
 MyDataRow("FM\_V1012") = Me.FM\_V1012;  
 MyDataRow("FM\_V1013") = Me.FM\_V1013;  
 MyDataRow("FM\_V1014") = Me.FM\_V1014;  
 MyDataRow("FM\_V1015") = Me.FM\_V1015;  
 MyDataRow("FM\_V1016") = Me.FM\_V1016;  
 MyDataRow("FM\_V1017") = Me.FM\_V1017;  
 MyDataRow("FM\_V1018") = Me.FM\_V1018;  
 MyDataRow("FM\_V1019") = Me.FM\_V1019;  
   
 'Format DateTime  
 Me.sDate = "";  
 Me.sHour = "";  
   
 if StringToIntg( Now().Day ) <= 9 then  
 day = "0" + Now().Day;  
 else  
 day = Now().Day;  
 endif;  
 if StringToIntg( Now().Month ) <= 9 then  
 month = "0" + Now().Month;  
 else  
 month = Now().Month;  
 endif;  
 if StringToIntg( Now().Hour ) <= 9 then  
 hour = "0" + Now().Hour;  
 else  
 hour = Now().Hour;  
 endif;  
 if StringToIntg( Now().Minute ) <= 9 then  
 min = "0" + Now().Minute;  
 else  
 min = Now().Minute;  
 endif;  
 if StringToIntg( Now().Second ) <= 9 then  
 sec = "0" + Now().Second;  
 else  
 sec = Now().Second;  
 endif;  
   
 Me.sDate = day + "/" + month + "/" + Now().Year;  
 Me.sHour = hour + ":" + min + ":" + sec;  
 MyDataRow("DateTime") = Me.sDate + " " + Me.sHour;  
   
 'calculate the DateTime in seconds  
 fSECONDS = DateTimeGMT() ;  
 fSECONDS = fSECONDS \* 86400 + (2\*3600);  
 sTemp = StringFromTime( fSECONDS, 2 );  
 MyDataRow("DateTime\_SEC") = fSECONDS;  
   
 if Me.bDebug == True then  
 LogMessage("sTemp = " + sTemp + " fSECONDS = " + Text(fSECONDS,"#.#"));  
 endif;  
   
 ' add & accept new row  
 Matrice.Rows.Add(MyDataRow);  
 Matrice.AcceptChanges();  
   
 ' MODIF EXPERT AND 18/01/2012  
 ' aucune raison de ne pas enregistrer le dataset si on est sensé écrire l'indice actuel  
 System.AppDomain.CurrentDomain.SetData("AppDomMyDataSet\_Matrice\_" + Me.Tagname, MyDataSet);  
 ' FIN MODIF AND  
  
 'write matrix DataSet to xml file  
 Me.Object.Script.XMLDSMatriceWrite = True;  
endif;

XMLDSMatriceRead

|  |  |
| --- | --- |
| Name | XMLDSMatriceRead |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.XMLDSMatriceRead |

**Declarations :**

Not Applicable

**Script :**

' script that read the Matrix XML file to DataSet  
' read Matrix values  
  
Dim MyDataSet as System.Data.DataSet;  
Dim Matrice as System.Data.DataTable;  
Dim FileName as String;  
Dim i as integer;  
Dim NumRecords as integer;  
Dim MyDataRow as System.Data.DataRow;  
Dim foundRows[1] as System.Data.DataRow;  
Dim Expression as string;  
Dim Sort as string;  
Dim bGetMatrLOCAL as boolean;  
  
if Me.bDebug == True then  
 LogMessage("Read Matrix xml file");  
endif;  
  
bGetMatrLOCAL = false;  
  
' reset trigger  
Me.Object.Script.XMLDSMatriceRead = False;  
  
if Me.bInitDone == False then  
 Me.bInitDone = True;  
 if Me.bDebug == True then  
 LogMessage("Initialisation DONE");  
 endif;  
endif;  
  
' retreive DataSet for Matrix from AppDomain  
FileName = Me.\_Config.XML.Directory + Me.TagName + "\_Matrice\_DS.xml";  
if System.IO.File.Exists(FileName) == True then  
 Me.bXMLReadConnexionLost = False;  
  
 MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_Matrice\_" + Me.Tagname);  
 ' add data to MyDataSet  
 if MyDataset <> null then  
 if Me.bDebug == 1 then logmessage("Read dataset from appdomain OK");endif;  
 Matrice = MyDataSet.Tables("Matrice");  
 ' clear result Matrice  
 Matrice.Clear();  
 ' read XML file  
 MyDataSet.ReadXml(FileName, System.Data.XmlReadMode.IgnoreSchema);  
   
 'get the last added index  
 ' init expression  
 Expression = "";  
 ' query  
 Expression = "Indice > 0";  
 ' query - order by  
 Sort = "DateTime\_SEC Desc";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression,Sort);  
 if Me.bDebug == 1 then logmessage("Query dataset");endif;  
   
 if foundRows.Length > 0 then  
   
 MyDataRow = foundRows[1];  
 'read the cycle number  
 if Me.bDebug == 1 then logmessage("Read cycle number");endif;  
 Me.iFM\_V[1] = MydataRow("FM\_V1000");  
 'test if a new cycle is started  
 if Me.bDebug == 1 then logmessage("Test if a new cycle started");endif;  
 if Me.FM\_V1000 <> Me.iFM\_V[1] then  
 bGetMatrLOCAL = true;  
 Me.bNouveauCycle = 1;  
 if Me.bDebug == True then  
 LogMessage("New cycle started !!!!");  
 endif;  
 else  
 'not a new cycle => read internally all values from file  
 Me.iIndiceActuel = MydataRow("Indice");  
 Me.iFM\_V[1] = MydataRow("FM\_V1000");  
 Me.iFM\_V[2] = MydataRow("FM\_V1001");  
 Me.iFM\_V[3] = MydataRow("FM\_V1002");  
 Me.iFM\_V[4] = MydataRow("FM\_V1003");  
 Me.iFM\_V[5] = MydataRow("FM\_V1004");  
 Me.iFM\_V[6] = MydataRow("FM\_V1005");  
 Me.iFM\_V[7] = MydataRow("FM\_V1006");  
 Me.iFM\_V[8] = MydataRow("FM\_V1007");  
 Me.iFM\_V[9] = MydataRow("FM\_V1008");  
 Me.iFM\_V[10] = MydataRow("FM\_V1009");  
 Me.iFM\_V[11] = MydataRow("FM\_V1010");  
 Me.iFM\_V[12] = MydataRow("FM\_V1011");  
 Me.iFM\_V[13] = MydataRow("FM\_V1012");  
 Me.iFM\_V[14] = MydataRow("FM\_V1013");  
 Me.iFM\_V[15] = MydataRow("FM\_V1014");  
 Me.iFM\_V[16] = MydataRow("FM\_V1015");  
 Me.iFM\_V[17] = MydataRow("FM\_V1016");  
 Me.iFM\_V[18] = MydataRow("FM\_V1017");  
 Me.iFM\_V[19] = MydataRow("FM\_V1018");  
 Me.iFM\_V[20] = MydataRow("FM\_V1019");  
 if Me.bDebug == True then  
 LogMessage("New cycle not started! ");  
 endif;  
 endif;  
 endif;  
 endif;  
else  
 Me.bXMLReadConnexionLost = True;  
 Me.iIndiceActuel = -1;  
 bGetMatrLOCAL = true;  
 if Me.bDebug == True then  
 LogMessage("Matrice XML File does not exist");  
 endif;  
endif;  
  
if bGetMatrLOCAL == true then  
 if Me.bDebug == 1 then logmessage("New cycle started, copy plc values to FM\_V");endif;  
 'new cycle started => copy all PLC values internally and trigger EnregistreIndiceActuel  
 Me.iFM\_V[1] = Me.FM\_V1000;  
 Me.iFM\_V[2] = Me.FM\_V1001;  
 Me.iFM\_V[3] = Me.FM\_V1002;  
 Me.iFM\_V[4] = Me.FM\_V1003;  
 Me.iFM\_V[5] = Me.FM\_V1004;  
 Me.iFM\_V[6] = Me.FM\_V1005;  
 Me.iFM\_V[7] = Me.FM\_V1006;  
 Me.iFM\_V[8] = Me.FM\_V1007;  
 Me.iFM\_V[9] = Me.FM\_V1008;  
 Me.iFM\_V[10] = Me.FM\_V1009;  
 Me.iFM\_V[11] = Me.FM\_V1010;  
 Me.iFM\_V[12] = Me.FM\_V1011;  
 Me.iFM\_V[13] = Me.FM\_V1012;  
 Me.iFM\_V[14] = Me.FM\_V1013;  
 Me.iFM\_V[15] = Me.FM\_V1014;  
 Me.iFM\_V[16] = Me.FM\_V1015;  
 Me.iFM\_V[17] = Me.FM\_V1016;  
 Me.iFM\_V[18] = Me.FM\_V1017;  
 Me.iFM\_V[19] = Me.FM\_V1018;  
 Me.iFM\_V[20] = Me.FM\_V1019;  
 if Me.bDebug == 1 then logmessage("New cycle started, copy plc values to FM\_V -- OK");endif;  
endif;

XMLDSMatriceWrite

|  |  |
| --- | --- |
| Name | XMLDSMatriceWrite |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.XMLDSMatriceWrite |

**Declarations :**

Not Applicable

**Script :**

' script that write Matrix XML file from DataSet  
  
Dim MyDataSet as System.Data.DataSet;  
Dim FileName as string;  
  
if Me.bDebug == True then  
 LogMessage("Write Matrix xml file");  
endif;  
  
' reset trigger  
Me.Object.Script.XMLDSMatriceWrite = False;  
  
' retreive MyDataSet from AppDomain  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_Matrice\_" + Me.Tagname);  
' write file  
if MyDataset <> null then  
 FileName = Me.\_Config.XML.Directory + Me.TagName + "\_Matrice\_DS.xml";  
   
 if System.IO.File.Exists(FileName) == True then  
 MyDataSet.WriteXml(FileName, System.Data.XmlWriteMode.IgnoreSchema);  
 Me.bXMLWriteConnexionLost = False;  
 else  
 Me.bXMLWriteConnexionLost = True;  
 if Me.bDebug == True then  
 LogMessage("Matrice XML File does not exist");  
 endif;  
 endif;  
endif;

InitTCR

|  |  |
| --- | --- |
| Name | InitTCR |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

'script that initialize the TCR  
'script executed each second  
  
  
'initialize temperature carneau  
if Me.dw\_SE\_RappAct.FM\_C551 <> 1 then  
 Me.fTCR = Me.FM\_V1017;  
endif;  
  
'calculate the limits for temperature carneau   
Me.fLIM\_B = Me.FM\_V1017 - Me.fkIP[24];  
Me.fLIM\_H = Me.FM\_V1017 + Me.fkIP[24];

TriggerCalcConsigne

|  |  |
| --- | --- |
| Name | TriggerCalcConsigne |
| Description |  |
| Trigger | OnTrue of Now().Hour == Me.Const.iHeureConsigne AND Now().Minute == Me.Const.iMinConsigne |

**Declarations :**

Not Applicable

**Script :**

'script that triggers the calculation of the new set point;  
'trigger every day at 12:00  
  
Me.dw\_SE\_RappAct.bCalcConsigne = True;  
   
Me.dw\_SE\_Conduite.bCreateRapportConduite = true;

Every10SEC

|  |  |
| --- | --- |
| Name | Every10SEC |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

'script that is executed every 10 seconds  
'I changed the trigger to 1 minute because 10 seconds is too often   
  
Dim MyDataSet as System.Data.DataSet;  
Dim Parameters as System.Data.DataTable;  
Dim MyDataRow as System.Data.DataRow;  
Dim foundRows[1] as System.Data.DataRow;  
Dim NumRecords as integer;  
Dim Expression as string;  
Dim Sort as string;  
Dim i as integer;  
Dim iCountkIP as integer;  
  
' display the percent in expert mode  
' Me.bAfficheTempsExpert = True;  
  
iCountkIP = 0;  
  
' check if all kIP are zero -> necessary for object deploy/undeploy  
for i = 1 to Me.Const.iNBRE\_COND  
 if Me.fkIP[i] == 0 then  
 iCountkIP = iCountkIP + 1;  
 endif;  
next;  
  
if iCountkIP <> Me.Const.iNBRE\_COND then  
  
' update the kIP XML file  
  
 ' retreive MyDataSet from AppDomain  
 MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_Param\_" + Me.Tagname);  
 if MyDataset <> null then  
 Parameters = MyDataSet.Tables("Parameters");  
  
 ' clear result Parameters  
 Parameters.Clear();  
  
 for i = 1 to Me.Const.iNBRE\_COND  
 ' add new row  
 MyDataRow = Parameters.NewRow();  
 ' add a line to Parameters  
 MyDataRow("Param\_Name") = "kIP" + i;  
 MyDataRow("Param\_Value") = Me.fkIP[i];  
 ' add & accept new row  
 Parameters.Rows.Add(MyDataRow);  
 Parameters.AcceptChanges();  
 next;  
  
 'save the kIP coefficients XML file  
 Me.Object.Script.XMLDSParamWrite = True;  
  
 'if the connection with XML file is lost, then write the Matrix XML file  
 if Me.bXMLWriteConnexionLost == True then  
 Me.bXMLWriteConnexionLost = False;  
 Me.Object.Script.XMLDSMatriceWrite = True;  
 endif;  
 endif;  
 else  
 if Me.bDebug == True then  
 LogMessage("All kIP = 0");  
 endif;  
   
endif;

CreateDataSets

|  |  |
| --- | --- |
| Name | CreateDataSets |
| Description |  |
| Trigger | WhileTrue of Me.bCreateDataSets |

**Declarations :**

Not Applicable

**Script :**

Dim MyDataSetParam as System.Data.DataSet;  
Dim MyDataSetMatrice as System.Data.DataSet;  
Dim Parameters as System.Data.DataTable;  
Dim Matrice as System.Data.DataTable;  
' AND MODIF EXPERT 18/01/2012  
'Dim MyDataRow as System.Data.DataRow;  
'Dim i as integer;  
' FIN MODIF AND  
  
if Me.bDebug == True then  
 LogMessage("Create DataSets: Matrice and kIP Parameters");  
endif;  
  
'reset the trigger  
Me.bCreateDataSets = False;  
  
'set the current four number  
Me.iNUM\_FOUR = StringToIntg(StringRight(Me.TagName,1));  
  
' create New DataSet for Parameters  
MyDataSetParam = New System.Data.DataSet;  
' add a table to our new DataSet  
MyDataSetParam.Tables.Add("Parameters");  
  
' set the table Parameters to point to the new Table in MyDataSet  
Parameters = MyDataSetParam.Tables("Parameters");  
  
'add Parameter Name as column  
Parameters.Columns.Add("Param\_Name", System.Type.GetType("System.String"));  
'add Parameter Value as column  
Parameters.Columns.Add("Param\_Value", System.Type.GetType("System.Double"));  
  
' store DataSet in AppDom (meaning, in the cache memory)  
System.AppDomain.CurrentDomain.SetData("AppDomMyDataSet\_Param\_" + Me.Tagname, MyDataSetParam);  
  
' AND MODIF EXPERT 18/01/2012  
if System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_Param\_" + Me.Tagname) <> null then  
 ' read XML file for Parameters data set  
 Me.Object.Script.XMLDSParamRead = True;  
endif;  
' FIN MODIF AND  
  
  
' create New DataSet for Matrice  
MyDataSetMatrice = New System.Data.DataSet;  
' add a table to our new DataSet  
MyDataSetMatrice.Tables.Add("Matrice");  
  
' set the table Matrice to point to the new Table in MyDataSet  
Matrice = MyDataSetMatrice.Tables("Matrice");  
  
'add Indice as column  
Matrice.Columns.Add("Indice", System.Type.GetType("System.Int32"));  
'add FM\_V1000 ... FM\_V1019 as columns  
Matrice.Columns.Add("FM\_V1000", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1001", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1002", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1003", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1004", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1005", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1006", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1007", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1008", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1009", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1010", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1011", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1012", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1013", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1014", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1015", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1016", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1017", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1018", System.Type.GetType("System.Int32"));  
Matrice.Columns.Add("FM\_V1019", System.Type.GetType("System.Int32"));  
  
'add DateTime as column  
Matrice.Columns.Add("DateTime\_SEC", System.Type.GetType("System.Int32"));  
'Matrice.Columns.Add("DateTime", System.Type.GetType("System.DateTime"));  
Matrice.Columns.Add("DateTime", System.Type.GetType("System.String"));  
  
' store DataSet in AppDom (meaning, in the cache memory)  
System.AppDomain.CurrentDomain.SetData("AppDomMyDataSet\_Matrice\_" + Me.Tagname, MyDataSetMatrice);  
  
' AND MODIF EXPERT 18/01/2012  
if System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_Matrice\_" + Me.Tagname) <> null then  
 ' read XML file for Parameters data set  
 Me.Object.Script.XMLDSMatriceRead = True;  
endif;  
  
' FIN MODIF AND

pStatus

|  |  |
| --- | --- |
| Name | pStatus |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

'script which calculate the status and write it in PLC  
'the variable STATUS contains a value that is always incremented  
'it is written in the PLC, in order to verify that the sation is well functioning  
  
if Me.iSTATUS > 32000 then  
 Me.iSTATUS = 0;  
endif;  
  
Me.iSTATUS = Me.iSTATUS + 1;  
  
' write the set points   
  
Me.STATUS = Me.iSTATUS;  
  
'write in PLC (if not zero -> test necessary for object deploy, and if PC is EXPERT)  
if Me.fTCR <> 0 AND Me.bPC\_EXPERT == 1 then  
 Me.TCR = Me.fTCR;  
endif;  
  
' AND 21/06/2010 - verification de la valeur lignite envoyee par rapport aux bornes 28/29  
' Calcul des paramètres consigne pierre moyenne / Calcul des poids Max & Min Lignite & Max & Min Gaz  
Me.fConsignePierreMoyenne = (Me.fConsignePierre\_C1+Me.fConsignePierre\_C2+Me.fConsignePierre\_C3)/3;  
  
' Avoid division by zero  
' changed by wwBenelux feb 09 2011  
if Me.fPCILignite>0 AND Me.fPCIGAZ>0 AND Me.iConsigneTpsCycle>0 then  
 ' calcul poids max & min lignite  
 Me.fPoidsMaxLignite = Me.fkIP[28]\*Me.fConsignePierreMoyenne\*(5/32)\*Me.fCoeffDecarbo/(Me.fPCILignite\*1000);  
 Me.fPoidsMinLignite = Me.fkIP[29]\*Me.fConsignePierreMoyenne\*(5/32)\*Me.fCoeffDecarbo/(Me.fPCILignite\*1000);  
 ' calcul poids max & min GAZ  
 Me.fPoidsMaxGAZ = Me.fkIP[28]\*Me.fConsignePierreMoyenne\*(5/32)\*Me.fCoeffDecarbo/(Me.fPCIGAZ\*1000)\*(36000/Me.iConsigneTpsCycle);  
 Me.fPoidsMinGAZ = Me.fkIP[29]\*Me.fConsignePierreMoyenne\*(5/32)\*Me.fCoeffDecarbo/(Me.fPCIGAZ\*1000)\*(36000/Me.iConsigneTpsCycle);  
  
 'ENVOIE VALEUR LIGNITE  
 'write in PLC (if not zero -> test necessary for object deploy, and if PC is EXPERT)  
'MODIF LPO 26/07/2012: envoi en permanence pour corriger probleme d'envoi ancienne consigne lors du basculement en mode expert  
 If Me.bModeExpert then  
 if Me.dw\_SE\_Conduite.iLIGNITE <> 0 AND Me.bPC\_EXPERT == 1 then  
   
  
 ' AND 21/06/2010 - verification de la valeur lignite envoyee par rapport aux bornes 28/29  
 if Me.dw\_SE\_Conduite.iLIGNITE <= Me.fPoidsMaxLignite and Me.dw\_SE\_Conduite.iLIGNITE >= Me.fPoidsMinLignite then  
 Me.LIGNITE = Me.dw\_SE\_Conduite.iLIGNITE;  
 endif;  
 if Me.fPoidsMaxLignite < Me.dw\_SE\_Conduite.iLIGNITE then  
 Me.LIGNITE = Me.fPoidsMaxLignite;  
 endif;  
 if Me.fPoidsMinLignite > Me.dw\_SE\_Conduite.iLIGNITE then  
 Me.LIGNITE = Me.fPoidsMinLignite;  
 endif;  
 endif;  
 'ENVOIE VALEUR GAZ  
  
 if Me.dw\_SE\_Conduite.iGAZ <> 0 AND Me.bPC\_EXPERT == 1 then  
   
   
 ' AND 21/06/2010 - verification de la valeur GAZ envoyee par rapport aux bornes 28/29  
 if Me.dw\_SE\_Conduite.iGAZ <= Me.fPoidsMaxGAZ and Me.dw\_SE\_Conduite.iGAZ >= Me.fPoidsMinGAZ then  
 Me.GAZ = Me.dw\_SE\_Conduite.iGAZ;  
 endif;  
 if Me.fPoidsMaxGAZ < Me.dw\_SE\_Conduite.iGAZ then  
 Me.GAZ = Me.fPoidsMaxGAZ;  
 endif;  
 if Me.fPoidsMinGAZ > Me.dw\_SE\_Conduite.iGAZ then  
 Me.GAZ = Me.fPoidsMinGAZ;  
 endif;  
  
 endif;  
  
 Else  
 Me.LIGNITE=Me.FM\_V1003;  
 Me.Gaz=Me.FM\_V1004;  
 ENDIF;  
endif;

dcMoyenneTemp

|  |  |
| --- | --- |
| Name | dcMoyenneTemp |
| Description |  |
| Trigger | DataChange of Me.FM\_V1001 |

**Declarations :**

Not Applicable

**Script :**

'script that executes when the "moyenne temperature" changed  
  
'copy locally the PLC value  
Me.iFM\_V[2] = Me.FM\_V1001;

dcMaxTemp

|  |  |
| --- | --- |
| Name | dcMaxTemp |
| Description |  |
| Trigger | DataChange of Me.FM\_V1002 |

**Declarations :**

Not Applicable

**Script :**

'script that executes when the "maximum temperature" changed  
  
'copy locally the PLC value  
Me.iFM\_V[3] = Me.FM\_V1002;

dcConsLignite

|  |  |
| --- | --- |
| Name | dcConsLignite |
| Description |  |
| Trigger | DataChange of Me.FM\_V1003 |

**Declarations :**

Not Applicable

**Script :**

'script that executes when the "consigne lignite" changed  
  
'copy locally the PLC value  
Me.iFM\_V[4] = Me.FM\_V1003;

dcConsigneGaz

|  |  |
| --- | --- |
| Name | dcConsigneGaz |
| Description |  |
| Trigger | DataChange of Me.FM\_V1004 |

**Declarations :**

Not Applicable

**Script :**

'script that executes when the "consigne lignite" changed  
  
'copy locally the PLC value  
Me.iFM\_V[5] = Me.FM\_V1004;

dcConsomSpecifique

|  |  |
| --- | --- |
| Name | dcConsomSpecifique |
| Description |  |
| Trigger | WhileTrue of Me.FM\_V1005 |

**Declarations :**

Not Applicable

**Script :**

'script that executes when the "consomation specifique cuve" changed  
  
'copy locally the PLC value  
Me.iFM\_V[6] = Me.FM\_V1005;

dcMinTemp

|  |  |
| --- | --- |
| Name | dcMinTemp |
| Description |  |
| Trigger | DataChange of Me.FM\_V1006 |

**Declarations :**

Not Applicable

**Script :**

'script that executes when the "minimum temperature" changed  
  
'copy locally the PLC value  
Me.iFM\_V[7] = Me.FM\_V1006;

dcConsTempCarneau

|  |  |
| --- | --- |
| Name | dcConsTempCarneau |
| Description |  |
| Trigger | DataChange of Me.FM\_V1007 |

**Declarations :**

Not Applicable

**Script :**

'script that executes when the "consigne temperature carneau" changed  
  
'copy locally the PLC value  
Me.iFM\_V[8] = Me.FM\_V1007;

dcTempsInversion

|  |  |
| --- | --- |
| Name | dcTempsInversion |
| Description |  |
| Trigger | DataChange of Me.FM\_V1013 |

**Declarations :**

Not Applicable

**Script :**

'script that executes when the "temps d'inversion moyen sur 100 cycles" changed  
  
'copy locally the PLC value  
Me.iFM\_V[14] = Me.FM\_V1013;

dcConsTempsCycle

|  |  |
| --- | --- |
| Name | dcConsTempsCycle |
| Description |  |
| Trigger | DataChange of Me.FM\_V1014 |

**Declarations :**

Not Applicable

**Script :**

'script that executes when the "consigne temps cycle" changed  
  
'copy locally the PLC value  
Me.iFM\_V[15] = Me.FM\_V1014;

dcValidOUI\_NON

|  |  |
| --- | --- |
| Name | dcValidOUI\_NON |
| Description |  |
| Trigger | DataChange of Me.FM\_V1010 |

**Declarations :**

Not Applicable

**Script :**

'script that executes when the "validation touche OUI = 1 / NON = 2" changed  
  
'copy locally the PLC value  
Me.iFM\_V[16] = Me.FM\_V1015;

dcTempActCarneau

|  |  |
| --- | --- |
| Name | dcTempActCarneau |
| Description |  |
| Trigger | DataChange of Me.FM\_V1016 |

**Declarations :**

Not Applicable

**Script :**

'script that executes when the "temperature actuelle carneau" changed  
  
'copy locally the PLC value  
Me.iFM\_V[17] = Me.FM\_V1016;

dcTempRefCarneau

|  |  |
| --- | --- |
| Name | dcTempRefCarneau |
| Description |  |
| Trigger | DataChange of Me.FM\_V1017 |

**Declarations :**

Not Applicable

**Script :**

'script that executes when the "temperature reference carneau" changed  
  
'copy locally the PLC value  
Me.iFM\_V[18] = Me.FM\_V1017;

dcModeExpert

|  |  |
| --- | --- |
| Name | dcModeExpert |
| Description |  |
| Trigger | DataChange of Me.FM\_V1018 |

**Declarations :**

Not Applicable

**Script :**

'script that executes when the "Mode System EXPERT = 2 / MANU = 1" changed  
  
'copy locally the PLC value  
Me.iFM\_V[19] = Me.FM\_V1018;

InitDeploy

|  |  |
| --- | --- |
| Name | InitDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $dwSystemExpert.dw\_SE\_Conduite

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAfficheAction | boolean |  |  |  |  |  |
| bAlm\_SystExpert | boolean |  |  |  | X |  |
| bCalcConditions | boolean |  |  |  |  |  |
| bCalcConduite | boolean |  |  |  |  |  |
| bCalcTendanceCombustible | boolean |  |  |  |  |  |
| bCalcTendanceCombustibleHF | boolean |  |  |  |  |  |
| bCalcTendanceTC | boolean |  |  |  |  |  |
| bCalcTendanceTCHF | boolean |  |  |  |  |  |
| bCreateRapportConduite | boolean |  |  |  |  |  |
| bDataRapportOK | boolean |  |  |  |  |  |
| Const.d100\_TOUR | double |  |  |  |  |  |
| Const.dCOMB | double |  |  |  |  |  |
| Const.iAJOUT\_COMB | integer |  |  |  |  |  |
| Const.iAJOUT\_COMB\_HF | integer |  |  |  |  |  |
| Const.iCOMB\_STABLE | integer |  |  |  |  |  |
| Const.iCOMB\_STABLE\_HF | integer |  |  |  |  |  |
| Const.iCS | integer |  |  |  |  |  |
| Const.ID\_100\_TOUR | double |  |  |  |  |  |
| Const.iGAZ | integer |  |  |  |  |  |
| Const.iLIGNITE | integer |  |  |  |  |  |
| Const.iNBRE\_AFFICHE\_ACTION | integer |  |  |  |  |  |
| Const.iRETRAIT\_COMB | integer |  |  |  |  |  |
| Const.iRETRAIT\_COMB\_HF | integer |  |  |  |  |  |
| Const.iTEMPE\_CARNEAU | integer |  |  |  |  |  |
| Const.sCALCUL\_R1 | string |  |  |  |  |  |
| Const.sRAJOUT | string |  |  |  |  |  |
| Const.sRAJOUT\_FOURCHETTE | string |  |  |  |  |  |
| Const.sRETRAIT | string |  |  |  |  |  |
| Const.sRETRAIT\_FOURCHETTE | string |  |  |  |  |  |
| Const.sTEXT\_RIEN | string |  |  |  |  |  |
| fDAlveol | double |  |  |  |  | X |
| fEVOL\_TC | double |  |  |  |  | X |
| fMEAN\_CS | double |  |  |  |  |  |
| fMEAN\_CS\_100 | double |  |  |  |  |  |
| fMEAN\_HCS | double |  |  |  |  |  |
| fMEAN\_LCS | double |  |  |  |  |  |
| fMEAN\_TC | double |  |  |  |  | X |
| fR1\_ACT | double |  |  |  |  |  |
| iCOND | integer |  |  |  |  |  |
| iDGAZ | integer |  |  |  |  | X |
| iDGAZ\_ACTION | integer |  |  |  |  |  |
| iDLIGNITE | integer |  |  |  |  | X |
| iDLIGNITE\_ACTION | integer |  |  |  |  |  |
| iGAZ | integer |  |  |  |  | X |
| iGAZ1 | integer |  |  |  |  | X |
| iLIGNITE | integer |  |  |  |  | X |
| iLIGNITE1 | integer |  |  |  |  | X |
| iTempRef | integer |  |  |  |  |  |
| iTempsCycle | integer |  |  |  |  |  |
| iTempsInversion | integer |  |  |  |  |  |
| iTEND | integer |  |  |  |  |  |
| iTendance | integer |  |  |  |  |  |
| iTendanceHF | integer |  |  |  |  |  |
| iTimeX | integer |  |  |  |  |  |
| iTimeY | integer |  |  |  |  |  |
| iValX | integer |  |  |  |  |  |
| iValY | integer |  |  |  |  |  |
| sACTION | string |  |  |  |  |  |
| sACTIONCOND | string |  |  |  |  |  |
| sDATE\_ACTION | string |  |  |  |  |  |
| sHEURE\_ACTION | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

CalcConduite

|  |  |
| --- | --- |
| Name | CalcConduite |
| Description |  |
| Trigger | WhileTrue of Me.bCalcConduite |

**Declarations :**

Not Applicable

**Script :**

'script that calculate the new set points (all displayed on the CONDUITE view)  
  
Dim MyDataSet as System.Data.DataSet;  
Dim Matrice as System.Data.DataTable;  
Dim MyDataRow as System.Data.DataRow;  
Dim foundRows[1] as System.Data.DataRow;  
Dim Mean\_TC\_Temp as double;  
Dim Mean\_CS\_Temp as Double;  
Dim MEAN\_CS\_100\_Temp as Double;  
Dim NumRecords as integer;  
Dim Expression as string;  
Dim iIndice as integer;  
Dim iNbreIndice as integer;  
Dim iError as integer;  
Dim i as integer;  
  
if MyContainer.bDebug == True then  
 LogMessage("Start CalcConduite");  
endif;  
  
me.bAlm\_SystExpert = False;  
  
'reset trigger  
Me.bCalcConduite = False;  
iError = 0;  
  
' retreive MyDataSet from AppDomain  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_Matrice\_" + MyContainer.Tagname);  
  
' MODIF EXPERT AND 18/01/2012  
' BLOC IF..ELSE..ENDIF de sécurité  
if MyDataset <> null then  
  
  
  
Matrice = MyDataSet.Tables("Matrice");  
  
'read the temperature ref. Carneau = address V1017  
Me.iTempRef = MyContainer.FM\_V1017;  
  
'read the cycle time = address V1014  
Me.iTempsCycle = MyContainer.FM\_V1014;  
  
'read the inversion time = address V1013  
Me.iTempsInversion = MyContainer.FM\_V1013;  
  
if MyContainer.bDebug == True then  
 LogMessage("Me.iTempRef = " + Text(Me.iTempRef,"#.#"));  
 LogMessage("Me.iTempsCycle = " + Text(Me.iTempsCycle,"#.#"));  
 LogMessage("Me.iTempsInversion = " + Text(Me.iTempsInversion,"#.#"));  
endif;  
  
if Me.iTempsCycle <= 0 OR Me.iTempsInversion <=0 then  
 iError = -1;  
else  
 iError = 0;  
endif;  
  
'calculate MEAN\_TC AND MEAN\_CS  
if iERROR == 0 then  
 iIndice = MyContainer.iIndiceActuel - MyContainer.fkIP[17] + 1;  
 if iIndice < 1 then  
 iIndice = iIndice + MyContainer.Const.iNBRE\_LIGNE;  
 endif;  
 if iIndice > MyContainer.Const.iNBRE\_LIGNE then  
 iError = -1;  
 endif;  
 if MyContainer.bDebug == True then  
 LogMessage("MEAN: iIndice = " + Text(iIndice,"#"));  
 endif;  
 iNbreIndice = MyContainer.iIndiceActuel - iIndice;  
 if iNbreIndice < 0 then  
 iNbreIndice = iNbreIndice + MyContainer.Const.iNBRE\_LIGNE;  
 endif;  
 if iNbreIndice == -1 then  
 iError = -1;  
 endif;  
  
 MEAN\_TC\_TEMP = 0;  
 MEAN\_CS\_TEMP = 0;  
 MEAN\_CS\_100\_Temp = 0;  
  
 i = iIndice;  
 while i <> MyContainer.iIndiceActuel + 1  
 'read the record   
 Expression = "";  
 ' query  
 Expression = "Indice = '" + i + "'";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression);  
  
 'test if the index exists  
 if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
 exit while;  
 endif;  
 MyDataRow = foundRows[1];  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
 'if matrix row found, read matrix values for index i  
 if NumRecords == 1 then  
 Me.iValX[Me.Const.iTEMPE\_CARNEAU] = MyDataRow("FM\_V1001");  
 Me.iValX[Me.Const.iCS] = MyDataRow("FM\_V1005");  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iValX[" + Me.Const.iTEMPE\_CARNEAU + "] = " + Text(Me.iValX[Me.Const.iTEMPE\_CARNEAU],"#.#"));  
 LogMessage("Me.iValX[" + Me.Const.iCS + "] = " + Text(Me.iValX[Me.Const.iTEMPE\_CARNEAU],"#.#"));  
 endif;  
 else  
 iError = -1;  
 endif;  
 if iError == 0 then  
 MEAN\_TC\_TEMP = MEAN\_TC\_TEMP + Me.iValX[Me.Const.iTEMPE\_CARNEAU];  
 MEAN\_CS\_TEMP = MEAN\_CS\_TEMP + Me.iValX[Me.Const.iCS];  
 endif;  
 if i == MyContainer.Const.iNBRE\_LIGNE then   
 i = 0;  
 endif;  
 'get the DateTime for iIndiceActuel  
 if i == MyContainer.iIndiceActuel then  
 Me.iTimeY = MyDataRow("DateTime\_SEC");  
 endif;  
 i = i + 1;  
 endwhile;  
  
 if iError == 0 then  
 'temperature moyenne  
 Me.fMEAN\_TC = MEAN\_TC\_TEMP /(iNbreIndice + 1);  
 'consomation specifique  
 Me.fMEAN\_CS = MEAN\_CS\_TEMP /(iNbreIndice + 1);  
  
 if MyContainer.bDebug == True then  
 LogMessage("Me.fMEAN\_TC = " + Text(Me.fMEAN\_TC,"#.#"));  
 LogMessage("Me.fMEAN\_CS = " + Text(Me.fMEAN\_CS,"#.#"));  
 endif;  
 endif; 'fin calcul MEAN\_TC and MEAN\_CS  
  
  
 iIndice = MyContainer.iIndiceActuel - Me.Const.ID\_100\_TOUR + 1;  
 if iIndice < 1 then  
 iIndice = iIndice + MyContainer.Const.iNBRE\_LIGNE;  
 endif;  
 if iIndice > MyContainer.Const.iNBRE\_LIGNE then  
 iError = -1;  
 endif;  
 if MyContainer.bDebug == True then  
 LogMessage("CS: iIndice = " + Text(iIndice,"#"));  
 endif;  
 iNbreIndice = MyContainer.iIndiceActuel - iIndice;  
 if iNbreIndice < 0 then  
 iNbreIndice = iNbreIndice + MyContainer.Const.iNBRE\_LIGNE;  
 endif;  
 if iNbreIndice == -1 then  
 iError = -1;  
 endif;  
  
 i = iIndice;  
 while i <> MyContainer.iIndiceActuel + 1  
 'read the record   
 Expression = "";  
 ' query  
 Expression = "Indice = '" + i + "'";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression);  
  
 'test if the index exists  
 if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
 exit while;  
 endif;  
  
 MyDataRow = foundRows[1];  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
  
 if MyContainer.bDebug == True then  
'GPR 19/12/2012 : ajout logmessage  
 logmessage ( "Nombre record = "+ NumRecords );   
 endif;  
  
 'if matrix row found, read matrix values for index i  
 if NumRecords == 1 then  
 Me.iValX[Me.Const.iCS] = MyDataRow("FM\_V1005");  
 if MyContainer.bDebug == True then  
 LogMessage(" i = " + i + "; Me.iValX[" + Me.Const.iCS + "] = " + Text(Me.iValX[Me.Const.iCS],"#.#"));  
 endif;  
 else  
 iError = -1;  
 endif;  
  
 if iError == 0 then  
 MEAN\_CS\_100\_TEMP = MEAN\_CS\_100\_TEMP + Me.iValX[Me.Const.iCS];  
 endif;  
 if i == MyContainer.Const.iNBRE\_LIGNE then   
 i = 0;  
 endif;  
 i = i + 1;  
 endwhile;  
  
 if iError == 0 then  
 'consomation specifique  
 Me.fMEAN\_CS\_100 = MEAN\_CS\_100\_TEMP /(iNbreIndice + 1);  
 Me.fMEAN\_HCS = Me.fMEAN\_CS\_100 + MyContainer.fkIP[27];  
 Me.fMEAN\_LCS = Me.fMEAN\_CS\_100 - MyContainer.fkIP[27];  
   
 if MyContainer.bDebug == True then  
 LogMessage("Me.fMEAN\_CS\_100 = " + Text(Me.fMEAN\_CS\_100,"#.#"));  
 LogMessage("Me.fMEAN\_LCS = " + Text(Me.fMEAN\_LCS,"#.#"));  
 LogMessage("Me.fMEAN\_HCS = " + Text(Me.fMEAN\_HCS,"#.#"));  
 endif;  
 endif; 'fin calcul MEAN\_CS\_100  
  
 'calculate R1\_ACT  
 if iError == 0 then  
 iIndice = MyContainer.iIndiceActuel - MyContainer.fkIP[21];  
 if iIndice < 1 then  
 iIndice = iIndice + MyContainer.Const.iNBRE\_LIGNE;  
 endif;  
 if iIndice > MyContainer.Const.iNBRE\_LIGNE then  
 iError = -1;  
 endif;  
 if MyContainer.bDebug == True then  
 LogMessage("R1\_ACT: iIndice = " + Text(iIndice,"#"));  
 endif;  
 iNbreIndice = MyContainer.iIndiceActuel - iIndice;  
 if iNbreIndice < 0 then  
 iNbreIndice = iNbreIndice + MyContainer.Const.iNBRE\_LIGNE;  
 endif;  
 if iNbreIndice < 0 then  
 iError = -1;  
 endif;  
  
 'read the record for iIndice  
 Expression = "";  
 ' query  
 Expression = "Indice = '" + iIndice + "'";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression);  
  
 'test if the index exists  
 if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
 endif;  
 if iError == 0 then  
  
 MyDataRow = foundRows[1];  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
 'if matrix row found, read DateTime value for index = iIndice  
 if NumRecords == 1 then  
 Me.iTimeX = MyDataRow("DateTime\_SEC");  
 else  
 iError = -1;  
 endif;  
  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iTimeX = " + Text(Me.iTimeX,"#"));  
 LogMessage("Me.iTimeY = " + Text(Me.iTimeY,"#"));  
 endif;  
  
 if (Me.iTempsCycle + Me.iTempsInversion) <> 0 AND (Me.iTimeX <> Me.iTimeY) then  
 Me.fR1\_ACT = iNbreIndice \* 100/((Me.iTimeY - Me.iTimeX)/(Me.iTempsCycle + Me.iTempsInversion));  
 else  
 iError = -1;  
 Me.sACTION = Me.Const.sCALCUL\_R1;  
 endif;  
  
 if MyContainer.bDebug == True then  
 LogMessage("Me.fR1\_ACT = " + Text(Me.fR1\_ACT,"#.#"));  
 LogMessage("Me.sACTION = " + Me.sACTION);  
 endif;  
  
 endif;  
  
 endif; 'fin calcul R1\_ACT  
  
 if iError == 0 then  
 Me.bCalcTendanceTC = True;  
 endif;  
endif;  
  
' MODIF EXPERT AND 18/01/2012  
' BLOC IF..ELSE..ENDIF de sécurité  
else  
 logmessage("SYSTEME EXPERT - impossible de calculer, matrice non mémorisée");  
 logmessage("SYSTEME EXPERT - Le Dataset de travail n'est pas en mémoire");  
 logmessage("SYSTEME EXPERT - Vérifier les fichiers XML dans beshdat2");  
 me.bAlm\_SystExpert = True;  
endif;

CalcTendanceTC

|  |  |
| --- | --- |
| Name | CalcTendanceTC |
| Description |  |
| Trigger | WhileTrue of Me.bCalcTendanceTC |

**Declarations :**

Not Applicable

**Script :**

'script that calculate the tendance for TC  
  
Dim MyDataSet as System.Data.DataSet;  
Dim Matrice as System.Data.DataTable;  
Dim MyDataRow as System.Data.DataRow;  
Dim foundRows[1] as System.Data.DataRow;  
Dim NumRecords as integer;  
Dim Expression as string;  
Dim iIndice as integer;  
Dim iNbreIndice as integer;  
Dim iError as integer;  
Dim i as integer;  
Dim fX as double;  
Dim fY as double;  
Dim fSXY as double;  
Dim fSXX as double;  
  
if MyContainer.bDebug == True then  
 LogMessage("Start CalcTendanceTC");  
endif;  
  
'reset trigger  
Me.bCalcTendanceTC = False;  
iError = 0;  
  
' retreive MyDataSet from AppDomain  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_Matrice\_" + MyContainer.Tagname);  
' MODIF EXPERT AND 18/01/2012  
' BLOC IF..ELSE..ENDIF de sécurité  
if MyDataset <> null then  
Matrice = MyDataSet.Tables("Matrice");  
  
'calculate the evolution of TC in tendance with kIP18  
iIndice = MyContainer.iIndiceActuel - MyContainer.fkIP[18] + 1;  
if iIndice < 1 then  
 iIndice = iIndice + MyContainer.Const.iNBRE\_LIGNE;  
endif;  
if iIndice > MyContainer.Const.iNBRE\_LIGNE then  
 iError = -1;  
endif;  
if MyContainer.bDebug == True then  
 LogMessage("evol TC: iIndice = " + Text(iIndice,"#"));  
endif;  
  
iNbreIndice = MyContainer.iIndiceActuel - iIndice + 1;  
if iNbreIndice < 0 then  
 iNbreIndice = iNbreIndice + MyContainer.Const.iNBRE\_LIGNE;  
endif;  
if iNbreIndice <= 0 then  
 iError = -1;  
endif;  
  
'linear regression over temperature carneau: y = ax + b  
'=> a = sum((X - xi) \* (Y - yi)) / sum((X - xi) \* (X - xi))  
' b = Y - aX  
'where X = average of xi AND Y = average of yi  
'calculate the average of TC and average of time  
fX = 0;  
fY = 0;  
   
i = iIndice;  
while i <> MyContainer.iIndiceActuel + 1  
 'read the record   
 Expression = "";  
 ' query  
 Expression = "Indice = '" + i + "'";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression);  
  
 'test if the index exists  
 if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
 exit while;  
 endif;  
  
 MyDataRow = foundRows[1];  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
 'if matrix row found, read matrix values for index i  
 if NumRecords == 1 then  
 Me.iValX[Me.Const.iTEMPE\_CARNEAU] = MyDataRow("FM\_V1001");  
 Me.iTimeX = MyDataRow("DateTime\_SEC");  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iValX[" + Me.Const.iTEMPE\_CARNEAU + "] = " + Text(Me.iValX[Me.Const.iTEMPE\_CARNEAU],"#.#"));  
 endif;  
 else  
 iError = -1;  
 endif;  
 if iError == 0 then  
 fX = fX + Me.iTimeX;  
 fY = fY + Me.iValX[Me.Const.iTEMPE\_CARNEAU];  
 endif;  
 if i == MyContainer.Const.iNBRE\_LIGNE then   
 i = 0;  
 endif;  
 'calculate average when i = iIndiceActuel  
 if i == MyContainer.iIndiceActuel then  
 fX = fX/iNbreIndice;  
 fY = fY/iNbreIndice;  
 endif;  
 if MyContainer.bDebug == True then  
 LogMessage("fX = " + Text(fX,"#.#") + " fY = " + Text(fY,"#.#") );  
 endif;  
 i = i + 1;  
endwhile;  
  
if iError == 0 then  
  
 'calculate the sums  
 fSXX = 0;  
 fSXY = 0;  
  
 i = iIndice;  
 while i <> MyContainer.iIndiceActuel + 1  
 'read the record   
 Expression = "";  
 ' query  
 Expression = "Indice = '" + i + "'";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression);  
  
 'test if the index exists  
 if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
 exit while;  
 endif;  
  
 MyDataRow = foundRows[1];  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
 'if matrix row found, read matrix values for index i  
 if NumRecords == 1 then  
 Me.iValX[Me.Const.iTEMPE\_CARNEAU] = MyDataRow("FM\_V1001");  
 Me.iTimeX = MyDataRow("DateTime\_SEC");  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iValX[" + Me.Const.iTEMPE\_CARNEAU + "] = " + Text(Me.iValX[Me.Const.iTEMPE\_CARNEAU],"#.#"));  
 endif;  
 else  
 iError = -1;  
 endif;  
 if iError == 0 then  
 fSXY = fSXY + (fY - Me.iValX[Me.Const.iTEMPE\_CARNEAU]) \* (fX - Me.iTimeX);  
 fSXX = fSXX + (fX - Me.iTimeX) \* (fX - Me.iTimeX);  
 if MyContainer.bDebug == True then  
 LogMessage("fSXX = " + Text(fSXX,"#.#") + " fSXY = " + Text(fSXY,"#.#") );  
 endif;  
 endif;  
 if i == MyContainer.Const.iNBRE\_LIGNE then   
 i = 0;   
 endif;  
 i = i + 1;  
 endwhile;  
  
 if iError == 0 then   
   
 Me.fEVOL\_TC = 0;  
  
 'calculate a  
 if iError == 0 then  
 if fSXX <> 0 then  
 Me.fEVOL\_TC = (fSXY/fSXX)\*3600;  
 else  
 iError = -1;  
 endif;  
 endif;  
  
 if MyContainer.bDebug == True then  
 LogMessage("Me.fEVOL\_TC = " + Text(Me.fEVOL\_TC,"#.#"));  
 endif;  
  
 'display the tendance -> iTEND  
 Me.iTEND = 0;  
 if (Me.fMEAN\_TC >= Me.iTempRef - MyContainer.fkIP[24]) OR   
 (Me.fMEAN\_TC <= Me.iTempRef + MyContainer.fkIP[24]) then  
 if Me.fEVOL\_TC >= abs(MyContainer.fkIP[12]) then  
 Me.iTEND = 3;  
 endif;  
 if Me.fEVOL\_TC <= -abs(MyContainer.fkIP[12]) then  
 Me.iTEND = 4;  
 endif;  
 endif;  
  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iTEND = " + Text(Me.iTEND,"#.#"));  
 endif;  
  
 endif;  
  
endif;  
  
if iError == 0 then  
 'calculate the combustible tendance  
 Me.bCalcTendanceCombustible = True;  
endif;  
  
' MODIF EXPERT AND 18/01/2012  
' BLOC IF..ELSE..ENDIF de sécurité  
else  
 logmessage("SYSTEME EXPERT - impossible de calculer, matrice non mémorisée");  
endif;

CalcTendanceCombustible

|  |  |
| --- | --- |
| Name | CalcTendanceCombustible |
| Description |  |
| Trigger | WhileTrue of Me.bCalcTendanceCombustible |

**Declarations :**

Not Applicable

**Script :**

'script that calculate the tendance for combustibles (Lignite and Gaz)  
  
Dim MyDataSet as System.Data.DataSet;  
Dim Matrice as System.Data.DataTable;  
Dim MyDataRow as System.Data.DataRow;  
Dim foundRows[1] as System.Data.DataRow;  
Dim NumRecords as integer;  
Dim Expression as string;  
Dim iIndice as integer;  
Dim iNbreIndice as integer;  
Dim iError as integer;  
Dim i as integer;  
  
if MyContainer.bDebug == True then  
 LogMessage("Start CalcTendanceCombustible");  
endif;  
  
'reset trigger  
Me.bCalcTendanceCombustible = False;  
iError = 0;  
  
' retreive MyDataSet from AppDomain  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_Matrice\_" + MyContainer.Tagname);  
' MODIF EXPERT AND 18/01/2012  
' BLOC IF..ELSE..ENDIF de sécurité  
if MyDataset <> null then  
Matrice = MyDataSet.Tables("Matrice");  
  
'read the combustible set points  
  
'read the record   
Expression = "";  
' query  
Expression = "Indice = '" + MyContainer.iIndiceActuel + "'";  
'retreive data  
foundRows[] = Matrice.Select(Expression);  
  
'test if the index exists  
if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
endif;  
  
if iError == 0 then  
  
MyDataRow = foundRows[1];  
NumRecords = foundRows[].GetUpperBound(0) + 1;  
'if matrix row found, read matrix values for index i  
if NumRecords == 1 then  
 Me.iValX[Me.Const.iLIGNITE] = MyDataRow("FM\_V1003");  
 Me.iValX[Me.Const.iGAZ] = MyDataRow("FM\_V1004");  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iValX[" + Me.Const.iLIGNITE + "] = " + Text(Me.iValX[Me.Const.iLIGNITE],"#.#"));  
 LogMessage("Me.iValX[" + Me.Const.iGAZ + "] = " + Text(Me.iValX[Me.Const.iGAZ],"#.#"));  
 endif;  
else  
 iError = -1;  
endif;  
  
Me.iLIGNITE1 = Me.iValX[Me.Const.iLIGNITE];  
Me.iGAZ1 = Me.iValX[Me.Const.iGAZ];  
Me.iDLIGNITE = 0;  
Me.iDGAZ = 0;  
Me.sACTION = Me.Const.sTEXT\_RIEN;  
  
if MyContainer.bDebug == True then  
 LogMessage("Me.iLIGNITE1 = " + Text(Me.iLIGNITE1,"#.#"));  
 LogMessage("Me.iGAZ1 = " + Text(Me.iGAZ1,"#.#"));  
endif;  
  
'check the combustible tendance following kIP 19  
iIndice = MyContainer.iIndiceActuel - MyContainer.fkIP[19] + 1;  
if iIndice < 1 then  
 iIndice = iIndice + MyContainer.Const.iNBRE\_LIGNE;  
endif;  
if iIndice > MyContainer.Const.iNBRE\_LIGNE then  
 iError = -1;  
endif;  
if MyContainer.bDebug == True then  
 LogMessage("Tendance combustible: iIndice = " + Text(iIndice,"#"));  
endif;  
  
Me.iTendance = Me.Const.iCOMB\_STABLE;  
  
i = iIndice;  
while i <> MyContainer.iIndiceActuel - 1  
 'read the record   
 Expression = "";  
 ' query  
 Expression = "Indice = '" + i + "'";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression);  
  
 'test if the index exists  
 if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
 exit while;  
 endif;  
  
 MyDataRow = foundRows[1];  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
 'if matrix row found, read matrix values for index i  
 if NumRecords == 1 then  
 Me.iValX[Me.Const.iLIGNITE] = MyDataRow("FM\_V1003");  
 Me.iValX[Me.Const.iGAZ] = MyDataRow("FM\_V1004");  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iValX[" + Me.Const.iLIGNITE + "] = " + Text(Me.iValX[Me.Const.iLIGNITE],"#.#"));  
 LogMessage("Me.iValX[" + Me.Const.iGAZ + "] = " + Text(Me.iValX[Me.Const.iGAZ],"#.#"));  
 endif;  
 else  
 iError = -1;  
 endif;  
 if (Me.iLIGNITE1 > Me.iValX[Me.Const.iLIGNITE]) OR (Me.iGAZ1 > Me.iValX[Me.Const.iGAZ]) then  
 Me.iTendance = Me.Const.iAJOUT\_COMB;  
 else  
 if (Me.iLIGNITE1 < Me.iValX[Me.Const.iLIGNITE]) OR (Me.iGAZ1 < Me.iValX[Me.Const.iGAZ]) then  
 Me.iTendance = Me.Const.iRETRAIT\_COMB;  
 endif;  
 endif;  
  
 if i == MyContainer.Const.iNBRE\_LIGNE then   
 i = 0;  
 endif;  
  
 i = i+ 1;  
endwhile;  
  
if iError == 0 and MyContainer.bModeExpert then   
  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iTendance = " + Text(Me.iTendance,"#"));  
 endif;  
  
 'check if "rajout de combustible en tendance"  
 if (Me.fR1\_ACT >= MyContainer.fkIP[20]) AND (Me.fMEAN\_TC < Me.iTempRef) AND  
 (Me.fMEAN\_TC > (Me.iTempRef - MyContainer.fkIP[24])) AND   
 (Me.fEVOL\_TC <= -abs(MyContainer.fkIP[12])) AND   
 (Me.iTendance <> Me.Const.iAJOUT\_COMB) AND  
 (Me.fMEAN\_CS <= (MyContainer.fkIP[28] \* (1 - Me.Const.dCOMB \* MyContainer.fkIP[14]))) AND   
 (Me.fMEAN\_CS <= ((Me.fMEAN\_CS\_100 + MyContainer.fkIP[27]) \* (1 - Me.Const.dCOMB \* MyContainer.fkIP[14]))) then  
 Me.iDLIGNITE = Int( Me.iLIGNITE1 \* Me.Const.dCOMB \* MyContainer.fkIP[14] );  
 Me.iDGAZ = Int( Me.iGAZ1 \* Me.Const.dCOMB \* MyContainer.fkIP[14] );  
 Me.sACTION = Me.Const.sRAJOUT;  
 endif;   
  
 'check if "retrait de combustible en tendance"  
 if (Me.fMEAN\_TC > Me.iTempRef) AND  
 (Me.fMEAN\_TC < (Me.iTempRef + MyContainer.fkIP[24])) AND   
 (Me.fEVOL\_TC >= abs(MyContainer.fkIP[12])) AND   
 (Me.iTendance <> Me.Const.iRETRAIT\_COMB) AND  
 (Me.fMEAN\_CS >= (MyContainer.fkIP[29] \* (1 + Me.Const.dCOMB \* MyContainer.fkIP[14]))) AND   
 (Me.fMEAN\_CS >= ((Me.fMEAN\_CS\_100 - MyContainer.fkIP[27]) \* (1 + Me.Const.dCOMB \* MyContainer.fkIP[14]))) then  
 Me.iDLIGNITE = 0 - Int( Me.iLIGNITE1 \* Me.Const.dCOMB \* MyContainer.fkIP[14] );  
 Me.iDGAZ = 0 - Int( Me.iGAZ1 \* Me.Const.dCOMB \* MyContainer.fkIP[14] );  
 Me.sACTION = Me.Const.sRETRAIT;  
 endif;   
  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iDLIGNITE = " + Text(Me.iDLIGNITE,"#.#"));  
 LogMessage("Me.iDGAZ = " + Text(Me.iDGAZ,"#.#"));  
 LogMessage("Me.sACTION = " + Me.sACTION);  
 endif;  
  
endif;  
  
endif;  
  
if iError == 0 then  
 'calculate the evolution of TC H.F.  
 Me.bCalcTendanceTCHF = True;  
endif;  
  
' MODIF EXPERT AND 18/01/2012  
' BLOC IF..ELSE..ENDIF de sécurité  
else  
 logmessage("SYSTEME EXPERT - impossible de calculer, matrice non mémorisée");  
endif;

CalcTendanceTCHF

|  |  |
| --- | --- |
| Name | CalcTendanceTCHF |
| Description |  |
| Trigger | WhileTrue of Me.bCalcTendanceTCHF |

**Declarations :**

Not Applicable

**Script :**

'script that calculate the evolution of TC H.F. with kIP22  
  
Dim MyDataSet as System.Data.DataSet;  
Dim Matrice as System.Data.DataTable;  
Dim MyDataRow as System.Data.DataRow;  
Dim foundRows[1] as System.Data.DataRow;  
Dim NumRecords as integer;  
Dim Expression as string;  
Dim iIndice as integer;  
Dim iNbreIndice as integer;  
Dim iError as integer;  
Dim i as integer;  
Dim fX as double;  
Dim fY as double;  
Dim fSXY as double;  
Dim fSXX as double;  
  
if MyContainer.bDebug == True then  
 LogMessage("Start CalcTendanceTCHF");  
endif;  
  
' retreive MyDataSet from AppDomain  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_Matrice\_" + MyContainer.Tagname);  
' MODIF EXPERT AND 18/01/2012  
' BLOC IF..ELSE..ENDIF de sécurité  
if MyDataset <> null then  
Matrice = MyDataSet.Tables("Matrice");  
  
'reset trigger  
Me.bCalcTendanceTCHF = False;  
iError = 0;  
  
'calculate the evolution of TC H.F. with kIP22  
iIndice = MyContainer.iIndiceActuel - MyContainer.fkIP[22] + 1;  
if iIndice < 1 then  
 iIndice = iIndice + MyContainer.Const.iNBRE\_LIGNE;  
endif;  
if iIndice > MyContainer.Const.iNBRE\_LIGNE then  
 iError = -1;  
endif;  
if MyContainer.bDebug == True then  
 LogMessage("evol TC HF: iIndice = " + Text(iIndice,"#"));  
endif;  
  
iNbreIndice = MyContainer.iIndiceActuel - iIndice + 1;  
if iNbreIndice < 0 then  
 iNbreIndice = iNbreIndice + MyContainer.Const.iNBRE\_LIGNE;  
endif;  
if iNbreIndice <= 0 then  
 iError = -1;  
endif;  
  
'linear regression over temperature carneau: y = ax + b  
'=> a = sum((X - xi) \* (Y - yi)) / sum((X - xi) \* (X - xi))  
' b = Y - aX  
'where X = average of xi AND Y = average of yi  
'calculate the average of TC and average of time  
fX = 0;  
fY = 0;  
  
i = iIndice;  
while i <> MyContainer.iIndiceActuel + 1  
 'read the record   
 Expression = "";  
 ' query  
 Expression = "Indice = '" + i + "'";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression);  
  
 'test if the index exists  
 if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
 exit while;  
 endif;  
  
 MyDataRow = foundRows[1];  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
 'if matrix row found, read matrix values for index i  
 if NumRecords == 1 then  
 Me.iValX[Me.Const.iTEMPE\_CARNEAU] = MyDataRow("FM\_V1001");  
 Me.iTimeX = MyDataRow("DateTime\_SEC");  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iValX[" + Me.Const.iTEMPE\_CARNEAU + "] = " + Text(Me.iValX[Me.Const.iTEMPE\_CARNEAU],"#.#"));  
 endif;  
 else  
 iError = -1;  
 endif;  
 if iError == 0 then  
 fX = fX + Me.iTimeX;  
 fY = fY + Me.iValX[Me.Const.iTEMPE\_CARNEAU];  
 endif;  
 if i == MyContainer.Const.iNBRE\_LIGNE then   
 i = 0;  
 endif;  
 'calculate average when i = iIndiceActuel  
 if i == MyContainer.iIndiceActuel then  
 fX = fX/iNbreIndice;  
 fY = fY/iNbreIndice;  
 endif;  
 if MyContainer.bDebug == True then  
 LogMessage("fX = " + Text(fX,"#.#") + " fY = " + Text(fY,"#.#") );  
 endif;  
 i = i + 1;  
endwhile;  
  
if iError == 0 then   
  
 'calculate the sums  
 fSXX = 0;  
 fSXY = 0;  
  
 Me.iTimeX = 0;  
 Me.iValX[Me.Const.iTEMPE\_CARNEAU] = 0;  
  
 i = iIndice;  
 while i <> MyContainer.iIndiceActuel + 1  
 'read the record   
 Expression = "";  
 ' query  
 Expression = "Indice = '" + i + "'";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression);  
  
 'test if the index exists  
 if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
 exit while;  
 endif;  
  
 MyDataRow = foundRows[1];  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
 'if matrix row found, read matrix values for index i  
 if NumRecords == 1 then  
 Me.iValX[Me.Const.iTEMPE\_CARNEAU] = MyDataRow("FM\_V1001");  
 Me.iTimeX = MyDataRow("DateTime\_SEC");  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iValX[" + Me.Const.iTEMPE\_CARNEAU + "] = " + Text(Me.iValX[Me.Const.iTEMPE\_CARNEAU],"#.#"));  
 endif;  
 else  
 iError = -1;  
 endif;  
 if iError == 0 then  
 fSXY = fSXY + (fY - Me.iValX[Me.Const.iTEMPE\_CARNEAU]) \* (fX - Me.iTimeX);  
 fSXX = fSXX + (fX - Me.iTimeX) \* (fX - Me.iTimeX);  
 if MyContainer.bDebug == True then  
 LogMessage("fSXX = " + Text(fSXX,"#.#") + " fSXY = " + Text(fSXY,"#.#") );  
 endif;  
 endif;  
 if i == MyContainer.Const.iNBRE\_LIGNE then   
 i = 0;  
 endif;  
 i = i + 1;  
 endwhile;  
  
  
 if iError == 0 then  
  
 Me.fEVOL\_TC = 0;  
  
 'calculate a  
 if iError == 0 then  
 if fSXX <> 0 then  
 Me.fEVOL\_TC = (fSXY/fSXX) \* 3600;  
 else  
 iError = -1;  
 endif;  
 endif;  
  
 if MyContainer.bDebug == True then  
 LogMessage("Me.fEVOL\_TC = " + Text(Me.fEVOL\_TC,"#.#"));  
 endif;  
  
 'display the tendance -> iTEND  
 Me.iTEND = 0;  
 if (Me.fMEAN\_TC > Me.iTempRef + MyContainer.fkIP[24]) OR   
 (Me.fMEAN\_TC < Me.iTempRef - MyContainer.fkIP[24]) then  
 if Me.fEVOL\_TC >= abs(MyContainer.fkIP[13]) then  
 Me.iTEND = 1;  
 endif;  
 if Me.fEVOL\_TC <= -abs(MyContainer.fkIP[13]) then  
 Me.iTEND = 2;  
 endif;  
 endif;  
  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iTEND = " + Text(Me.iTEND,"#.#"));  
 endif;  
  
 endif;  
endif;  
  
if iError == 0 then  
 'calculate the combustible tendance HF  
 Me.bCalcTendanceCombustibleHF = True;  
endif;  
  
' MODIF EXPERT AND 18/01/2012  
' BLOC IF..ELSE..ENDIF de sécurité  
else  
 logmessage("SYSTEME EXPERT - impossible de calculer, matrice non mémorisée");  
endif;

CalcTendanceCombustibleHF

|  |  |
| --- | --- |
| Name | CalcTendanceCombustibleHF |
| Description |  |
| Trigger | WhileTrue of Me.bCalcTendanceCombustibleHF |

**Declarations :**

Not Applicable

**Script :**

'script that calculate the tendance for combustibles (Lignite and Gaz)  
'following kIP23  
  
Dim MyDataSet as System.Data.DataSet;  
Dim Matrice as System.Data.DataTable;  
Dim MyDataRow as System.Data.DataRow;  
Dim foundRows[1] as System.Data.DataRow;  
Dim NumRecords as integer;  
Dim Expression as string;  
Dim iIndice as integer;  
Dim iNbreIndice as integer;  
Dim iError as integer;  
Dim i as integer;  
Dim fFourchette as double;  
  
if MyContainer.bDebug == True then  
 LogMessage("Start CalcTendanceCombustibleHF");  
endif;  
  
'reset trigger  
Me.bCalcTendanceCombustibleHF = False;  
iError = 0;  
  
' retreive MyDataSet from AppDomain  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_Matrice\_" + MyContainer.Tagname);  
' MODIF EXPERT AND 18/01/2012  
' BLOC IF..ELSE..ENDIF de sécurité  
if MyDataset <> null then  
Matrice = MyDataSet.Tables("Matrice");  
  
iIndice = MyContainer.iIndiceActuel - MyContainer.fkIP[23] + 1;  
if iIndice < 1 then  
 iIndice = iIndice + MyContainer.Const.iNBRE\_LIGNE;  
endif;  
if iIndice > MyContainer.Const.iNBRE\_LIGNE then  
 iError = -1;  
endif;  
if MyContainer.bDebug == True then  
 LogMessage("Tendance Combustible HF: iIndice = " + Text(iIndice,"#"));  
endif;  
  
Me.iTendanceHF = Me.Const.iCOMB\_STABLE\_HF;  
  
'read the record   
Expression = "";  
' query  
Expression = "Indice = '" + MyContainer.iIndiceActuel + "'";  
'retreive data  
foundRows[] = Matrice.Select(Expression);  
  
'test if the index exists  
if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
endif;  
  
if iError == 0 then  
  
MyDataRow = foundRows[1];  
NumRecords = foundRows[].GetUpperBound(0) + 1;  
'if matrix row found, read matrix values for index i  
if NumRecords == 1 then  
 Me.iValX[Me.Const.iLIGNITE] = MyDataRow("FM\_V1003");  
 Me.iValX[Me.Const.iGAZ] = MyDataRow("FM\_V1004");  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iValX[" + Me.Const.iLIGNITE + "] = " + Text(Me.iValX[Me.Const.iLIGNITE],"#.#"));  
 LogMessage("Me.iValX[" + Me.Const.iGAZ + "] = " + Text(Me.iValX[Me.Const.iGAZ],"#.#"));  
 endif;  
else  
 iError = -1;  
endif;  
  
Me.iLIGNITE1 = Me.iValX[Me.Const.iLIGNITE];  
Me.iGAZ1 = Me.iValX[Me.Const.iGAZ];  
  
'calculate fourchette (kIP14 + kIP15)/2  
fFourchette = (Mycontainer.fkIP[14] + Mycontainer.fkIP[15])/200 \* Me.iLIGNITE1;  
  
if MyContainer.bDebug == True then  
 LogMessage("Me.iLIGNITE1 = " + Text(Me.iLIGNITE1,"#.#"));  
 LogMessage("Me.iGAZ1 = " + Text(Me.iGAZ1,"#.#"));  
 LogMessage("fFourchette = " + Text(fFourchette,"#.#"));  
endif;  
  
'check the combustible tendance  
i = iIndice;  
while i <> MyContainer.iIndiceActuel  
 'read the record   
 Expression = "";  
 ' query  
 Expression = "Indice = '" + i + "'";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression);  
  
 'test if the index exists  
 if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
 exit while;  
 endif;  
  
 MyDataRow = foundRows[1];  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
 'if matrix row found, read matrix values for index i  
 if NumRecords == 1 then  
 Me.iValX[Me.Const.iLIGNITE] = MyDataRow("FM\_V1003");  
 Me.iValX[Me.Const.iGAZ] = MyDataRow("FM\_V1004");  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iValX[" + Me.Const.iLIGNITE + "] = " + Text(Me.iValX[Me.Const.iLIGNITE],"#.#"));  
 LogMessage("Me.iValX[" + Me.Const.iGAZ + "] = " + Text(Me.iValX[Me.Const.iGAZ],"#.#"));  
 endif;  
 else  
 iError = -1;  
 endif;  
 if (Me.iLIGNITE1 > (Me.iValX[Me.Const.iLIGNITE] + fFourchette)) OR   
 (Me.iGAZ1 > (Me.iValX[Me.Const.iGAZ] + fFourchette)) then  
 Me.iTendanceHF = Me.Const.iAJOUT\_COMB\_HF;  
 else  
 if (Me.iLIGNITE1 < Me.iValX[Me.Const.iLIGNITE] - fFourchette) OR   
 (Me.iGAZ1 < Me.iValX[Me.Const.iGAZ] - fFourchette) then  
 Me.iTendanceHF = Me.Const.iRETRAIT\_COMB\_HF;  
 endif;  
 endif;  
  
 if i == MyContainer.Const.iNBRE\_LIGNE then   
 i = 0;   
 endif;  
 i = i + 1;  
endwhile;  
  
if iError == 0 then  
  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iTendanceHF = " + Text(Me.iTendanceHF,"#"));  
 endif;  
  
 if iError == 0 then  
  
 'check if "retrait hors fourchette haut"  
 if (Me.fMEAN\_TC > (Me.iTempRef + MyContainer.fkIP[24])) AND   
 (Me.fEVOL\_TC >= -abs(MyContainer.fkIP[13])) AND   
 (Me.iTendanceHF <> Me.Const.iRETRAIT\_COMB\_HF) then  
 Me.iDLIGNITE = 0 - Int(Me.iLIGNITE1 \* Me.Const.dCOMB \* MyContainer.fkIP[15]);  
 Me.iDGAZ = 0 - Int(Me.iGAZ1 \* Me.Const.dCOMB \* MyContainer.fkIP[15]);  
 Me.fDAlveol = - MyContainer.fkIP[8];  
 Me.sACTION = Me.Const.sRETRAIT\_FOURCHETTE;  
 'write ALVE in PLC  
 if MyContainer.bPC\_EXPERT == 1 then  
 MyContainer.ALVE = Me.fDAlveol;   
 'FVA - Remise à zéro du fDAlveol après écriture  
 if MyContainer.bDebug == True then  
 LogMessage("Me.fDAlveol = " + Text(Me.fDAlveol,"#.###"));  
 LogMessage("MyContainer.ALVE = " + Text(MyContainer.ALVE,"#.###"));  
 endif;  
 Me.fDAlveol = 0;  
 endif;  
 endif;  
  
 'check if "rajout hors fourchette"  
 if (Me.fR1\_ACT >= MyContainer.fkIP[20]) AND   
 (Me.fMEAN\_TC < (Me.iTempRef - MyContainer.fkIP[24])) AND   
 (Me.fEVOL\_TC <= MyContainer.fkIP[13]) AND   
 (Me.iTendanceHF <> Me.Const.iAJOUT\_COMB\_HF) AND  
 (Me.fMEAN\_CS <= (MyContainer.fkIP[28] \* (1 - Me.Const.dCOMB \* MyContainer.fkIP[15]))) AND   
 (Me.fMEAN\_CS <= ((Me.fMEAN\_CS\_100 + MyContainer.fkIP[27]) \* (1 - Me.Const.dCOMB \* MyContainer.fkIP[15]))) then  
 Me.iDLIGNITE = Int(Me.iLIGNITE1 \* Me.Const.dCOMB \* MyContainer.fkIP[15]);  
 Me.iDGAZ = Int(Me.iGAZ1 \* Me.Const.dCOMB \* MyContainer.fkIP[15]);  
 Me.sACTION = Me.Const.sRAJOUT\_FOURCHETTE;  
 endif;  
  
 endif;  
  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iDLIGNITE = " + Text(Me.iDLIGNITE,"#.#"));  
 LogMessage("Me.iDGAZ = " + Text(Me.iDGAZ,"#.#"));  
 LogMessage("Me.sACTION = " + Me.sACTION);  
 endif;  
  
 endif;  
endif;  
  
if iError == 0 then  
 'display the conditions  
 Me.bCalcConditions = True;  
endif;  
  
' MODIF EXPERT AND 18/01/2012  
' BLOC IF..ELSE..ENDIF de sécurité  
else  
 logmessage("SYSTEME EXPERT - impossible de calculer, matrice non mémorisée");  
endif;

CalcConditions

|  |  |
| --- | --- |
| Name | CalcConditions |
| Description |  |
| Trigger | WhileTrue of Me.bCalcConditions |

**Declarations :**

Not Applicable

**Script :**

'script that make the verifications and display the conditions -> iCOND[1 -- 17]  
  
Dim i as integer;  
  
'reset trigger  
Me.bCalcConditions = False;  
  
for i = 1 to 17  
 Me.iCOND[i] = 1;  
 if Me.fR1\_ACT >= MyContainer.fkIP[20] then  
 Me.iCOND[1] = 2;  
 Me.iCOND[2] = 2;  
 endif;   
 if Me.fMEAN\_TC > (Me.iTempRef + MyContainer.fkIP[24]) then  
 Me.iCOND[3] = 2;  
 endif;  
 if Me.fMEAN\_TC > Me.iTempRef AND  
 Me.fMEAN\_TC < (Me.iTempRef + MyContainer.fkIP[24])then  
 Me.iCOND[4] = 2;  
 endif;  
 if Me.fMEAN\_TC < Me.iTempRef AND  
 Me.fMEAN\_TC > (Me.iTempRef - MyContainer.fkIP[24])then  
 Me.iCOND[5] = 2;  
 endif;  
 if Me.fMEAN\_TC < (Me.iTempRef - MyContainer.fkIP[24])then  
 Me.iCOND[6] = 2;  
 endif;  
 if Me.fEVOL\_TC >= abs(MyContainer.fkIP[12])then  
 Me.iCOND[7] = 2;  
 endif;  
 if Me.fEVOL\_TC <= -abs(MyContainer.fkIP[12])then  
 Me.iCOND[8] = 2;  
 endif;  
 if Me.fEVOL\_TC >= -abs(MyContainer.fkIP[13])then  
 Me.iCOND[9] = 2;  
 endif;  
 if Me.fEVOL\_TC <= MyContainer.fkIP[13] then  
 Me.iCOND[10] = 2;  
 endif;  
 if Me.iTendance <> Me.Const.iRETRAIT\_COMB then  
 Me.iCOND[11] = 2;  
 endif;  
 if Me.iTendance <> Me.Const.iAJOUT\_COMB then  
 Me.iCOND[12] = 2;  
 endif;  
 if Me.iTendanceHF <> Me.Const.iRETRAIT\_COMB\_HF then  
 Me.iCOND[13] = 2;  
 endif;  
 if Me.iTendanceHF <> Me.Const.iAJOUT\_COMB\_HF then  
 Me.iCOND[14] = 2;  
 endif;  
 if Me.fMEAN\_CS >= (MyContainer.fkIP[29] \* (1 + Me.Const.dCOMB \* MyContainer.fkIP[14])) AND  
 Me.fMEAN\_CS >= ((Me.fMEAN\_CS\_100 - MyContainer.fkIP[27]) \* (1 + Me.Const.dCOMB \* MyContainer.fkIP[14])) then  
 Me.iCOND[15] = 2;  
 endif;  
 if Me.fMEAN\_CS <= (MyContainer.fkIP[28] \* (1 - Me.Const.dCOMB \* MyContainer.fkIP[14])) AND  
 Me.fMEAN\_CS <= ((Me.fMEAN\_CS\_100 + MyContainer.fkIP[27]) \* (1 - Me.Const.dCOMB \* MyContainer.fkIP[14])) then  
 Me.iCOND[16] = 2;  
 endif;  
 if Me.fMEAN\_CS <= (MyContainer.fkIP[28] \* (1 - Me.Const.dCOMB \* MyContainer.fkIP[15])) AND  
 Me.fMEAN\_CS <= ((Me.fMEAN\_CS\_100 + MyContainer.fkIP[27]) \* (1 - Me.Const.dCOMB \* MyContainer.fkIP[15])) then  
 Me.iCOND[17] = 2;  
 endif;  
next;  
  
'calculate new combustible set points  
Me.iLIGNITE = Me.iLIGNITE1 + Me.iDLIGNITE;  
Me.iGAZ = Me.iGAZ1 + Me.iDGAZ;  
  
if MyContainer.bDebug == True then  
 LogMessage("Me.iLIGNITE = " + Text(Me.iLIGNITE,"#.#"));  
 LogMessage("Me.iGAZ = " + Text(Me.iGAZ,"#.#"));  
endif;  
  
'display the action  
if Me.sACTION <> Me.Const.sTEXT\_RIEN AND MyContainer.FM\_V1018 == 2 then  
 Me.bAfficheAction = True;  
endif;

AfficheAction

|  |  |
| --- | --- |
| Name | AfficheAction |
| Description |  |
| Trigger | WhileTrue of Me.bAfficheAction |

**Declarations :**

Not Applicable

**Script :**

'script that display the last 5 modifications of combustible in assisted behaviour  
  
Dim i as integer;  
Dim j as integer;  
Dim iIndex as integer;  
Dim day as string;  
Dim month as string;  
Dim hour as string;  
Dim min as string;  
Dim sec as string;  
  
if MyContainer.bDebug == True then  
 LogMessage("Start AfficheAction");  
endif;  
  
'reset trigger  
Me.bAfficheAction = False;  
  
'shift the texts  
for i = Me.Const.iNBRE\_AFFICHE\_ACTION - 1 to 1  
 j = i + 1;  
 'shift the date  
 Me.sDATE\_ACTION[j] = Me.sDATE\_ACTION[i];  
 'shift the hour  
 Me.sHEURE\_ACTION[j] = Me.sHEURE\_ACTION[i];  
 'shift the GAZ  
 Me.iDGAZ\_ACTION[j] = Me.iDGAZ\_ACTION[i];  
 'shift the LIGNITE  
 Me.iDLIGNITE\_ACTION[j] = Me.iDLIGNITE\_ACTION[i];  
 'shift the ACTION  
 Me.sACTIONCOND[j] = Me.sACTIONCOND[i];  
next;  
  
iIndex = 1;  
  
'Format DateTime  
if StringToIntg( Now().Day ) <= 9 then  
 day = "0" + Now().Day;  
else  
 day = Now().Day;  
endif;  
if StringToIntg( Now().Month ) <= 9 then  
 month = "0" + Now().Month;  
else  
 month = Now().Month;  
endif;  
if StringToIntg( Now().Hour ) <= 9 then  
 hour = "0" + Now().Hour;  
else  
 hour = Now().Hour;  
endif;  
if StringToIntg( Now().Minute ) <= 9 then  
 min = "0" + Now().Minute;  
else  
 min = Now().Minute;  
endif;  
if StringToIntg( Now().Second ) <= 9 then  
 sec = "0" + Now().Second;  
else  
 sec = Now().Second;  
endif;  
  
'display the date  
Me.sDATE\_ACTION[iIndex] = day + "/" + month + "/" + Now().Year;  
'display the hour  
Me.sHEURE\_ACTION[iIndex] = hour + ":" + min + ":" + sec;  
'display the gaz  
Me.iDGAZ\_ACTION[iIndex] = Me.iDGAZ;  
'display the lignite  
Me.iDLIGNITE\_ACTION[iIndex] = Me.iDLIGNITE;  
'display the action  
Me.sACTIONCOND[iIndex] = Me.sACTION;

CreateRapportConduite

|  |  |
| --- | --- |
| Name | CreateRapportConduite |
| Description |  |
| Trigger | WhileTrue of Me.bCreateRapportConduite |

**Declarations :**

Not Applicable

**Script :**

'script that insert the necessary data for Conduite report  
  
dim connection as System.Data.SqlClient.SqlConnection;   
dim command as System.Data.SqlClient.SqlCommand;   
dim TypeConduite as System.Data.SqlClient.SqlParameter;   
dim TCActuelle as System.Data.SqlClient.SqlParameter;   
dim TCReference as System.Data.SqlClient.SqlParameter;   
dim CSActuelle as System.Data.SqlClient.SqlParameter;   
dim CSReference as System.Data.SqlClient.SqlParameter;   
dim ExpertLignite as System.Data.SqlClient.SqlParameter;   
dim ExpertGaz as System.Data.SqlClient.SqlParameter;   
dim DateAction1 as System.Data.SqlClient.SqlParameter;   
dim HeureAction1 as System.Data.SqlClient.SqlParameter;   
dim LigniteAction1 as System.Data.SqlClient.SqlParameter;   
dim GazAction1 as System.Data.SqlClient.SqlParameter;   
dim ActionCond1 as System.Data.SqlClient.SqlParameter;   
dim DateAction2 as System.Data.SqlClient.SqlParameter;   
dim HeureAction2 as System.Data.SqlClient.SqlParameter;   
dim LigniteAction2 as System.Data.SqlClient.SqlParameter;   
dim GazAction2 as System.Data.SqlClient.SqlParameter;   
dim ActionCond2 as System.Data.SqlClient.SqlParameter;   
dim DateAction3 as System.Data.SqlClient.SqlParameter;   
dim HeureAction3 as System.Data.SqlClient.SqlParameter;   
dim LigniteAction3 as System.Data.SqlClient.SqlParameter;   
dim GazAction3 as System.Data.SqlClient.SqlParameter;   
dim ActionCond3 as System.Data.SqlClient.SqlParameter;   
dim DateAction4 as System.Data.SqlClient.SqlParameter;   
dim HeureAction4 as System.Data.SqlClient.SqlParameter;   
dim LigniteAction4 as System.Data.SqlClient.SqlParameter;   
dim GazAction4 as System.Data.SqlClient.SqlParameter;   
dim ActionCond4 as System.Data.SqlClient.SqlParameter;   
dim DateAction5 as System.Data.SqlClient.SqlParameter;   
dim HeureAction5 as System.Data.SqlClient.SqlParameter;   
dim LigniteAction5 as System.Data.SqlClient.SqlParameter;   
dim GazAction5 as System.Data.SqlClient.SqlParameter;   
dim ActionCond5 as System.Data.SqlClient.SqlParameter;   
dim NumFour as System.Data.SqlClient.SqlParameter;   
dim query as string;   
  
'reset trigger  
Me.bCreateRapportConduite = False;  
  
MyContainer.SQL.Query.ConnectString = "server=" + MyContainer.\_Config.SQL.Host +  
 ";uid=" + MyContainer.\_Config.SQL.User + ";pwd=" + MyContainer.\_Config.SQL.Pass +  
 ";database="+ MyContainer.\_Config.SQL.Database;  
connection = new System.Data.SqlClient.SqlConnection(MyContainer.SQL.Query.ConnectString);   
connection.Open();   
  
' AND 24/11/2011 vidange table avant mise à jour  
query = "DELETE FROM DWH\_Rapport\_SE\_CONDUITE WHERE NUM\_FOUR = '" + Text(MyContainer.iNUM\_FOUR,"#")+"'";  
command = new System.Data.SqlClient.SqlCommand(query, connection);  
command.executenonquery();  
  
query = "INSERT INTO DWH\_Rapport\_SE\_CONDUITE " +  
 "(TYPE\_CONDUITE, TC\_ACTUELLE, TC\_REFERENCE, CS\_ACTUELLE, CS\_REFERENCE, EXPERT\_LIGNITE, EXPERT\_GAZ, " +  
 "DATE\_ACTION1, HEURE\_ACTION1, LIGNITE\_ACTION1, GAZ\_ACTION1, ACTION\_COND1, " +  
 "DATE\_ACTION2, HEURE\_ACTION2, LIGNITE\_ACTION2, GAZ\_ACTION2, ACTION\_COND2, " +  
 "DATE\_ACTION3, HEURE\_ACTION3, LIGNITE\_ACTION3, GAZ\_ACTION3, ACTION\_COND3, " +  
 "DATE\_ACTION4, HEURE\_ACTION4, LIGNITE\_ACTION4, GAZ\_ACTION4, ACTION\_COND4, " +  
 "DATE\_ACTION5, HEURE\_ACTION5, LIGNITE\_ACTION5, GAZ\_ACTION5, ACTION\_COND5, NUM\_FOUR )" +  
 "VALUES (@typeConduite, @tcActuelle, @tcReference, @csActuelle, @csReference, @expertLignite, @expertGaz, " +  
 "@dateAction1, @heureAction1, @ligniteAction1, @gazAction1, @actionCond1, " +  
 "@dateAction2, @heureAction2, @ligniteAction2, @gazAction2, @actionCond2, " +  
 "@dateAction3, @heureAction3, @ligniteAction3, @gazAction3, @actionCond3, " +  
 "@dateAction4, @heureAction4, @ligniteAction4, @gazAction4, @actionCond4, " +  
 "@dateAction5, @heureAction5, @ligniteAction5, @gazAction5, @actionCond5, @numFour )";  
command = new System.Data.SqlClient.SqlCommand(query, connection);  
  
if MyContainer.bDebug == True then  
 LogMessage(query);  
endif;  
  
TypeConduite = command.Parameters.Add("@typeConduite", System.Data.SqlDbType.NVarChar, 50);  
TCActuelle = command.Parameters.Add("@tcActuelle", System.Data.SqlDbType.NVarChar, 50);  
TCReference = command.Parameters.Add("@tcReference", System.Data.SqlDbType.NVarChar, 50);  
CSActuelle = command.Parameters.Add("@csActuelle", System.Data.SqlDbType.NVarChar, 50);  
CSReference = command.Parameters.Add("@csReference", System.Data.SqlDbType.NVarChar, 50);  
ExpertLignite = command.Parameters.Add("@expertLignite", System.Data.SqlDbType.NVarChar, 50);  
ExpertGaz = command.Parameters.Add("@expertGaz", System.Data.SqlDbType.NVarChar, 50);  
DateAction1 = command.Parameters.Add("@dateAction1", System.Data.SqlDbType.NVarChar, 50);  
HeureAction1 = command.Parameters.Add("@heureAction1", System.Data.SqlDbType.NVarChar, 50);  
LigniteAction1 = command.Parameters.Add("@ligniteAction1", System.Data.SqlDbType.NVarChar, 50);  
GazAction1 = command.Parameters.Add("@gazAction1", System.Data.SqlDbType.NVarChar, 50);  
ActionCond1 = command.Parameters.Add("@actionCond1", System.Data.SqlDbType.NVarChar, 50);  
DateAction2 = command.Parameters.Add("@dateAction2", System.Data.SqlDbType.NVarChar, 50);  
HeureAction2 = command.Parameters.Add("@heureAction2", System.Data.SqlDbType.NVarChar, 50);  
LigniteAction2 = command.Parameters.Add("@ligniteAction2", System.Data.SqlDbType.NVarChar, 50);  
GazAction2 = command.Parameters.Add("@gazAction2", System.Data.SqlDbType.NVarChar, 50);  
ActionCond2 = command.Parameters.Add("@actionCond2", System.Data.SqlDbType.NVarChar, 50);  
DateAction3 = command.Parameters.Add("@dateAction3", System.Data.SqlDbType.NVarChar, 50);  
HeureAction3 = command.Parameters.Add("@heureAction3", System.Data.SqlDbType.NVarChar, 50);  
LigniteAction3 = command.Parameters.Add("@ligniteAction3", System.Data.SqlDbType.NVarChar, 50);  
GazAction3 = command.Parameters.Add("@gazAction3", System.Data.SqlDbType.NVarChar, 50);  
ActionCond3 = command.Parameters.Add("@actionCond3", System.Data.SqlDbType.NVarChar, 50);  
DateAction4 = command.Parameters.Add("@dateAction4", System.Data.SqlDbType.NVarChar, 50);  
HeureAction4 = command.Parameters.Add("@heureAction4", System.Data.SqlDbType.NVarChar, 50);  
LigniteAction4 = command.Parameters.Add("@ligniteAction4", System.Data.SqlDbType.NVarChar, 50);  
GazAction4 = command.Parameters.Add("@gazAction4", System.Data.SqlDbType.NVarChar, 50);  
ActionCond4 = command.Parameters.Add("@actionCond4", System.Data.SqlDbType.NVarChar, 50);  
DateAction5 = command.Parameters.Add("@dateAction5", System.Data.SqlDbType.NVarChar, 50);  
HeureAction5 = command.Parameters.Add("@heureAction5", System.Data.SqlDbType.NVarChar, 50);  
LigniteAction5 = command.Parameters.Add("@ligniteAction5", System.Data.SqlDbType.NVarChar, 50);  
GazAction5 = command.Parameters.Add("@gazAction5", System.Data.SqlDbType.NVarChar, 50);  
ActionCond5 = command.Parameters.Add("@actionCond5", System.Data.SqlDbType.NVarChar, 50);  
NumFour = command.Parameters.Add("@numFour", System.Data.SqlDbType.NVarChar, 50);  
  
'Prepare query  
command.Prepare();  
  
if MyContainer.FM\_V1018 == 1 then  
 TypeConduite.Value = "MANU";  
else  
if MyContainer.FM\_V1018 == 2 then  
 TypeConduite.Value = "ASSISTEE";  
else  
 TypeConduite.Value = "\*\*\*\*\*\*\*\*";  
endif;  
endif;  
  
TCActuelle.Value = Int( Me.fMEAN\_TC );  
TCReference.Value = Int( MyContainer.FM\_V1017 );  
CSActuelle.Value = Int( Me.fMEAN\_CS );  
CSReference.Value = Int( MyContainer.fkIP[27] );  
ExpertLignite.Value = Me.iDLIGNITE;  
ExpertGaz.Value = Me.iDGAZ;  
DateAction1.Value = Me.sDATE\_ACTION[1];  
HeureAction1.Value = Me.sHEURE\_ACTION[1];  
LigniteAction1.Value = Me.iDLIGNITE\_ACTION[1];  
GazAction1.Value = Me.iDGAZ\_ACTION[1];  
ActionCond1.Value = Me.sACTIONCOND[1];  
DateAction2.Value = Me.sDATE\_ACTION[2];  
HeureAction2.Value = Me.sHEURE\_ACTION[2];  
LigniteAction2.Value = Me.iDLIGNITE\_ACTION[2];  
GazAction2.Value = Me.iDGAZ\_ACTION[2];  
ActionCond2.Value = Me.sACTIONCOND[2];  
DateAction3.Value = Me.sDATE\_ACTION[3];  
HeureAction3.Value = Me.sHEURE\_ACTION[3];  
LigniteAction3.Value = Me.iDLIGNITE\_ACTION[3];  
GazAction3.Value = Me.iDGAZ\_ACTION[3];  
ActionCond3.Value = Me.sACTIONCOND[3];  
DateAction4.Value = Me.sDATE\_ACTION[4];  
HeureAction4.Value = Me.sHEURE\_ACTION[4];  
LigniteAction4.Value = Me.iDLIGNITE\_ACTION[4];  
GazAction4.Value = Me.iDGAZ\_ACTION[4];  
ActionCond4.Value = Me.sACTIONCOND[4];  
DateAction5.Value = Me.sDATE\_ACTION[5];  
HeureAction5.Value = Me.sHEURE\_ACTION[5];  
LigniteAction5.Value = Me.iDLIGNITE\_ACTION[5];  
GazAction5.Value = Me.iDGAZ\_ACTION[5];  
ActionCond5.Value = Me.sACTIONCOND[5];  
NumFour.Value = MyContainer.iNUM\_FOUR;  
  
command.ExecuteNonQuery();  
  
'close connection  
connection.Close();   
  
'finish to insert the data for the report  
'Me.bDataRapportOK = True;  
  
if MyContainer.bDebug == True then  
 LogMessage("Data OK for printing Conduite report");  
endif;

#### Template ArchestrA $dwSystemExpert.dw\_SE\_RappAct

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bCalcConsigne | boolean |  |  |  |  |  |
| bCreateRapportActivite | boolean |  |  |  |  |  |
| bNON | boolean |  |  |  |  |  |
| bOUI | boolean |  |  |  |  |  |
| bPrintRequest | boolean |  |  |  |  |  |
| bReset | boolean |  |  |  |  |  |
| bToucheValid | boolean |  |  |  |  |  |
| Const.sACTIVITE\_FAIBLE | string |  |  |  |  |  |
| Const.sPROCESS\_INSTABLE | string |  |  |  |  |  |
| Const.sRIEN | string |  |  |  |  |  |
| Const.sTEMP\_LIM\_ATTEINTE | string |  |  |  |  |  |
| Const.sTEMP\_REF\_ATTEINTE | string |  |  |  |  |  |
| fCO2 | float |  |  |  |  |  |
| fCO2\_1 | float |  | X |  |  |  |
| fCO2\_10 | float |  | X |  |  |  |
| fCO2\_11 | float |  | X |  |  |  |
| fCO2\_12 | float |  | X |  |  |  |
| fCO2\_2 | float |  | X |  |  |  |
| fCO2\_3 | float |  | X |  |  |  |
| fCO2\_4 | float |  | X |  |  |  |
| fCO2\_5 | float |  | X |  |  |  |
| fCO2\_6 | float |  | X |  |  |  |
| fCO2\_7 | float |  | X |  |  |  |
| fCO2\_8 | float |  | X |  |  |  |
| fCO2\_9 | float |  | X |  |  |  |
| fR1ab | double |  |  |  |  |  |
| fR2ab | double |  |  |  |  |  |
| fR3ab | double |  |  |  |  |  |
| fR4ab | double |  |  |  |  |  |
| iPasChangekIP4 | integer |  |  |  |  |  |
| sACTION | string |  |  |  |  |  |
| sBP\_DATE | string |  |  |  |  |  |
| sBP\_HEURE | string |  |  |  |  |  |

##### Field Attributes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Attribute name | Data Type | IO Scaling | History | Limit Alarms | Rate Of Change Alarms | Target Deviation Alarms | Bad Value Alarm | Statistics |
| FM\_C550 | boolean |  | X |  |  |  |  |  |
| FM\_C551 | boolean |  | X |  |  |  |  |  |
| BP\_F | integer |  | X |  |  |  |  |  |

##### Scripts

dcToucheValid

|  |  |
| --- | --- |
| Name | dcToucheValid |
| Description |  |
| Trigger | WhileTrue of Me.bToucheValid |

**Declarations :**

Not Applicable

**Script :**

' script that read the status of the validity button  
' button "NON" => FM\_C551 = 0  
' button "OUI" => FM\_C551 = 1  
  
Dim day as string;  
Dim month as string;  
Dim hour as string;  
Dim min as string;  
Dim sec as string;  
  
'reset trigger  
Me.bToucheValid = 0;  
  
'Format DateTime  
  
if StringToIntg( Now().Day ) <= 9 then  
 day = "0" + Now().Day;  
else  
 day = Now().Day;  
endif;  
if StringToIntg( Now().Month ) <= 9 then  
 month = "0" + Now().Month;  
else  
 month = Now().Month;  
endif;  
if StringToIntg( Now().Hour ) <= 9 then  
 hour = "0" + Now().Hour;  
else  
 hour = Now().Hour;  
endif;  
if StringToIntg( Now().Minute ) <= 9 then  
 min = "0" + Now().Minute;  
else  
 min = Now().Minute;  
endif;  
if StringToIntg( Now().Second ) <= 9 then  
 sec = "0" + Now().Second;  
else  
 sec = Now().Second;  
endif;  
  
'set the current date and time to display  
Me.sBP\_DATE = day + "/" + month + "/" + Now().Year;  
Me.sBP\_HEURE = hour + ":" + min + ":" + sec;

SetOUI

|  |  |
| --- | --- |
| Name | SetOUI |
| Description |  |
| Trigger | WhileTrue of Me.bOUI |

**Declarations :**

Not Applicable

**Script :**

'reset trigger  
Me.bOUI = False;  
  
'set FM\_C551 to 1  
Me.FM\_C551 = 1;  
  
'set bToucheValid to true   
Me.bToucheValid = 1;

SetNON

|  |  |
| --- | --- |
| Name | SetNON |
| Description |  |
| Trigger | WhileTrue of Me.bNON |

**Declarations :**

Not Applicable

**Script :**

'reset trigger  
Me.bNON = False;  
  
'set FM\_C551 to 0  
Me.FM\_C551 = 0;  
  
'set bToucheValid to true   
Me.bToucheValid = 1;

CreateRapportActivite

|  |  |
| --- | --- |
| Name | CreateRapportActivite |
| Description |  |
| Trigger | WhileTrue of Me.bCreateRapportActivite == True |

**Declarations :**

Not Applicable

**Script :**

dim connection as System.Data.SqlClient.SqlConnection;   
dim command as System.Data.SqlClient.SqlCommand;   
dim MoyenneCO2 as System.Data.SqlClient.SqlParameter;   
dim TCMoyenneCO2 as System.Data.SqlClient.SqlParameter;   
dim PercentActivite as System.Data.SqlClient.SqlParameter;   
dim TCReferenceCO2 as System.Data.SqlClient.SqlParameter;   
dim TCSystemExpert as System.Data.SqlClient.SqlParameter;   
dim Action as System.Data.SqlClient.SqlParameter;   
dim ProcessValidite as System.Data.SqlClient.SqlParameter;   
dim TCReferenceRapport as System.Data.SqlClient.SqlParameter;   
dim TCChangement as System.Data.SqlClient.SqlParameter;   
dim ResultatCO2 as System.Data.SqlClient.SqlParameter;   
dim DateTime as System.Data.SqlClient.SqlParameter;   
dim TCMoyenne as System.Data.SqlClient.SqlParameter;   
dim PercentFonctionnement as System.Data.SqlClient.SqlParameter;   
dim NumCO2 as System.Data.SqlClient.SqlParameter;   
dim NumFour as System.Data.SqlClient.SqlParameter;   
dim query as string;   
dim i as integer;  
  
'reset trigger  
Me.bCreateRapportActivite = False;  
  
MyContainer.SQL.Query.ConnectString = "server=" + MyContainer.\_Config.SQL.Host +  
 ";uid=" + MyContainer.\_Config.SQL.User + ";pwd=" + MyContainer.\_Config.SQL.Pass +  
 ";database="+ MyContainer.\_Config.SQL.Database;  
connection = new System.Data.SqlClient.SqlConnection(MyContainer.SQL.Query.ConnectString);   
  
connection.Open();   
  
' AND 24/11/2011 vidange table avant mise à jour  
query = "DELETE FROM DWH\_Rapport\_SE\_ACTIVITE WHERE NUM\_FOUR = '" + Text(MyContainer.iNUM\_FOUR,"#") + "'";  
command = new System.Data.SqlClient.SqlCommand(query, connection);  
command.executenonquery();  
  
'insert the data in DWH\_Rapport\_SE\_ACTIVITE table  
query = "INSERT INTO DWH\_Rapport\_SE\_ACTIVITE " +  
 "(MOYENNE\_CO2, TC\_MOYENNE\_CO2, PERCENT\_ACTIVITE, TC\_REFERENCE\_CO2, TC\_SYSTEM\_EXPERT, ACTION, " +  
 "PROCESS\_VALIDITE, TC\_REFERENCE\_RAPPORT, TC\_CHANGEMENT, NUM\_FOUR )" +  
 "VALUES (@moyenneCO2, @tcMoyenneCO2, @percentActivite, @tcReferenceCO2, @tcSystemExpert, @action, " +  
 "@processValidite, @tcReferenceRapport, @tcChangement, @numFour )";  
command = new System.Data.SqlClient.SqlCommand(query, connection);  
  
if MyContainer.bDebug == True then  
 LogMessage(query);  
endif;  
  
MoyenneCO2 = command.Parameters.Add("@moyenneCO2", System.Data.SqlDbType.NVarChar, 50);  
TCMoyenneCO2 = command.Parameters.Add("@tcMoyenneCO2", System.Data.SqlDbType.NVarChar, 50);  
PercentActivite = command.Parameters.Add("@percentActivite", System.Data.SqlDbType.NVarChar, 50);  
TCReferenceCO2 = command.Parameters.Add("@tcReferenceCO2", System.Data.SqlDbType.NVarChar, 50);  
TCSystemExpert = command.Parameters.Add("@tcSystemExpert", System.Data.SqlDbType.NVarChar, 50);  
Action = command.Parameters.Add("@action", System.Data.SqlDbType.NVarChar, 50);  
ProcessValidite = command.Parameters.Add("@processValidite", System.Data.SqlDbType.NVarChar, 50);  
TCReferenceRapport = command.Parameters.Add("@tcReferenceRapport", System.Data.SqlDbType.NVarChar, 50);  
TCChangement = command.Parameters.Add("@tcChangement", System.Data.SqlDbType.NVarChar, 50);  
NumFour = command.Parameters.Add("@numFour", System.Data.SqlDbType.NVarChar, 50);  
  
'Prepare query  
command.Prepare();  
  
MoyenneCO2.Value = Text(MyContainer.dw\_SE\_ResLab.fR3ab,"#.##");  
TCMoyenneCO2.Value = Int( MyContainer.dw\_SE\_ResLab.fR2ab );  
PercentActivite.Value = Int( MyContainer.dw\_SE\_ResLab.fR1ab );  
TCReferenceCO2.Value = Int( MyContainer.dw\_SE\_ResLab.fR4ab );  
TCSystemExpert.Value = Int( MyContainer.fkIP[4] );  
Action.Value = Me.sACTION;  
TCReferenceRapport.Value = MyContainer.FM\_V1017;  
NumFour.Value = MyContainer.iNUM\_FOUR;  
  
if Me.BP\_F == 1 then  
 ProcessValidite.Value = "Process Valid, le " + Me.sBP\_DATE + " " + Me.sBP\_HEURE;  
else  
 ProcessValidite.Value = "Process non Valid";  
endif;  
  
if Me.iPasChangekIP4 == 1 then  
 TCChangement.Value = 0;  
else  
 TCChangement.Value = 1;  
endif;  
  
command.ExecuteNonQuery();  
  
query = "DELETE FROM DWH\_Rapport\_SE\_RESCO2 where NUM\_FOUR = '" + Text(MyContainer.iNUM\_FOUR,"#") + "'";  
command = new System.Data.SqlClient.SqlCommand(query, connection);  
command.ExecuteNonQuery();  
  
  
'insert the data in DWH\_Rapport\_SE\_RESCO2 table  
for i = 1 to 12  
 query = "";  
 if Me.fCO2[i] == 0 then  
 exit for;  
 else  
 query = "INSERT INTO DWH\_Rapport\_SE\_RESCO2 " +  
 "(RESULTAT\_CO2, DATETIME, TC\_MOYENNE, PERCENT\_FONCTIONNEMENT, NUM\_CO2, NUM\_FOUR )" +  
 "VALUES (@resultatCO2, @dateTime, @tcMoyenne, @percentFonctionnement, @numCO2, @numFour )";  
 command = new System.Data.SqlClient.SqlCommand(query, connection);  
  
 if MyContainer.bDebug == True then  
 LogMessage(query);  
 endif;  
  
 ResultatCO2 = command.Parameters.Add("@resultatCO2", System.Data.SqlDbType.NVarChar, 50);  
 DateTime = command.Parameters.Add("@dateTime", System.Data.SqlDbType.NVarChar, 50);  
 TCMoyenne = command.Parameters.Add("@tcMoyenne", System.Data.SqlDbType.NVarChar, 50);  
 PercentFonctionnement = command.Parameters.Add("@percentFonctionnement", System.Data.SqlDbType.NVarChar, 50);  
 NumCO2 = command.Parameters.Add("@numCO2", System.Data.SqlDbType.NVarChar, 50);  
 NumFour = command.Parameters.Add("@numFour", System.Data.SqlDbType.NVarChar, 50);  
  
 'Prepare query  
 command.Prepare();  
  
 ResultatCO2.Value = Text(Me.fCO2[i],"#.##");  
 DateTime.Value = MyContainer.dw\_SE\_ResLab.sDATE\_CO2[i] + " " + MyContainer.dw\_SE\_ResLab.sHEURE\_CO2[i];  
 TCMoyenne.Value = Int( MyContainer.dw\_SE\_ResLab.fR2[i] );  
 PercentFonctionnement.Value = Int( MyContainer.dw\_SE\_ResLab.fR1[i] );  
 NumCO2.Value = i;  
 NumFour.Value = MyContainer.iNUM\_FOUR;  
  
 command.ExecuteNonQuery();  
 endif;  
next;  
  
'close connection  
connection.Close();   
  
if MyContainer.bDebug == True then  
 LogMessage("Data OK for printing Activite report");  
endif;  
'ajout Me.bReset = True le 02/02/2012 BW   
Me.bReset = True;  
  
'trigger the creation of a printing request for "rapport activite"  
'Me.bPrintRequest = True;

Reset

|  |  |
| --- | --- |
| Name | Reset |
| Description |  |
| Trigger | WhileTrue of Me.bReset == True |

**Declarations :**

Not Applicable

**Script :**

'script that triggers the reset for CO2 values and for validation OUI/NON  
  
'reset trigger  
Me.bReset = False;  
  
'reset CO2 coefficients  
MyContainer.dw\_SE\_ResLab.bResetCO2 = True;  
  
  
if MyContainer.bDebug == True then  
 LogMessage("EXECUTION SCRIPT RESET");  
endif;  
  
'reset the validation  
Me.FM\_C551 = 0;

CalcConsigne

|  |  |
| --- | --- |
| Name | CalcConsigne |
| Description |  |
| Trigger | WhileTrue of Me.bCalcConsigne |

**Declarations :**

Not Applicable

**Script :**

' script that calculates the new "temperature carneau"  
' when the process conditions are validated (button "OUI")then write in PLC  
'  
  
Dim iTempAct as integer;  
Dim iModif as integer;  
  
if MyContainer.bDebug == True then  
 LogMessage("Start CalcConsigne");  
endif;  
  
'reset trigger  
Me.bCalcConsigne = False;  
  
'it will be set to 1 when a kIP4 modification happens  
Me.iPasChangekIP4 = 0;  
imodif = 0;  
  
'read R1ab, R2ab, R3ab, R4ab  
Me.fR1ab = MyContainer.dw\_SE\_ResLab.fR1ab;  
Me.fR2ab = MyContainer.dw\_SE\_ResLab.fR2ab;  
Me.fR3ab = MyContainer.dw\_SE\_ResLab.fR3ab;  
Me.fR4ab = MyContainer.dw\_SE\_ResLab.fR4ab;  
  
if MyContainer.bDebug == True then  
 LogMessage("fR1ab = " + Text(Me.fR1ab,"#.#") + "; fR2ab = " + Text(Me.fR2ab,"#.#") + "; fR3ab = " +   
 Text(Me.fR3ab,"#.#") + "; fR4ab = " + Text(Me.fR4ab,"#.#"));  
endif;  
  
'read the actual temperature carneau = address V1017  
iTempAct = MyContainer.FM\_V1017;  
if MyContainer.bDebug == True then  
 LogMessage("iTempAct = " + Text(iTempAct,"#.#"));  
endif;  
  
'calculate the new temperature carneau (proposed by expert system)  
if Me.fR1ab >= MyContainer.fkIP[1] then  
 if (Me.fR3ab > MyContainer.fkIP[6]) AND (Me.fR2ab >= (Me.fR4ab - MyContainer.fkIP[7])) then  
 MyContainer.fkIP[4] = Me.fR4ab + (MyContainer.fkIP[9] \* MyContainer.fkIP[2]);  
 iModif = 1;  
 if MyContainer.bDebug == True then  
 LogMessage("1) iModif = 1; MyContainer.fkIP[4] = " + Text(MyContainer.fkIP[4],"#"));  
 endif;  
 else  
 if (Me.fR3ab < MyContainer.fkIP[5]) AND (Me.fR2ab <= (Me.fR4ab + MyContainer.fkIP[7])) then  
 MyContainer.fkIP[4] = Me.fR4ab - (MyContainer.fkIP[9] \* MyContainer.fkIP[2]);  
 iModif = 1;  
 if MyContainer.bDebug == True then  
 LogMessage("2) iModif = 1; MyContainer.fkIP[4] = " + Text(MyContainer.fkIP[4],"#"));  
 endif;  
 else  
 if (MyContainer.fkIP[5] <= Me.fR3ab) AND (Me.fR3ab <= MyContainer.fkIP[6] AND  
 (Abs(Me.fR2ab - Me.fR4ab) <= MyContainer.fkIP[7])) then  
 MyContainer.fkIP[4] = Me.fR4ab;  
 iModif = 1;   
 if MyContainer.bDebug == True then  
 LogMessage("3) iModif = 1; MyContainer.fkIP[4] = " + Text(MyContainer.fkIP[4],"#"));  
 endif;  
 else  
 if (MyContainer.fkIP[5] <= Me.fR3ab) AND (Me.fR3ab <= MyContainer.fkIP[6] AND  
 ( (Me.fR2ab - Me.fR4ab) < (0 -MyContainer.fkIP[7]) ) ) then  
 MyContainer.fkIP[4] = Me.fR4ab - MyContainer.fkIP[9];  
 iModif = 1;  
 if MyContainer.bDebug == True then   
 LogMessage("4) iModif = 1; MyContainer.fkIP[4] = " + Text(MyContainer.fkIP[4],"#"));  
 endif;  
 else  
 if (MyContainer.fkIP[5] <= Me.fR3ab) AND (Me.fR3ab <= MyContainer.fkIP[6] AND  
 ( (Me.fR2ab - Me.fR4ab) > MyContainer.fkIP[7] )) then  
 MyContainer.fkIP[4] = Me.fR4ab + MyContainer.fkIP[9];  
 iModif = 1;   
 if MyContainer.bDebug == True then  
 LogMessage("5) iModif = 1; MyContainer.fkIP[4] = " + Text(MyContainer.fkIP[4],"#"));  
 endif;  
 endif;  
 endif;  
 endif;  
 endif;  
 endif;  
  
 if MyContainer.fkIP[4] >= MyContainer.fkIP[11] then  
 MyContainer.fkIP[4] = MyContainer.fkIP[11];  
 iModif = 1;   
 Me.sACTION = Me.Const.sTEMP\_LIM\_ATTEINTE;  
 if MyContainer.bDebug == True then  
 LogMessage("6) iModif = 1; MyContainer.fkIP[4] = " + Text(MyContainer.fkIP[4],"#"));  
 endif;  
 else  
 if MyContainer.fkIP[4] <= MyContainer.fkIP[10] then  
 MyContainer.fkIP[4] = MyContainer.fkIP[10];  
 iModif = 1;   
 Me.sACTION = Me.Const.sTEMP\_REF\_ATTEINTE;  
 if MyContainer.bDebug == True then  
 LogMessage("6) iModif = 1; MyContainer.fkIP[4] = " + Text(MyContainer.fkIP[4],"#"));  
 endif;  
 endif;  
 endif;  
  
 if iModif == 0 then  
 MyContainer.fkIP[4] = System.Convert.ToDouble( iTempAct );  
 Me.iPasChangekIP4 = 1;  
 Me.sACTION = Me.Const.sRIEN;  
 if MyContainer.bDebug == True then  
 LogMessage("7) iModif = 0; MyContainer.fkIP[4] = " + Text(MyContainer.fkIP[4],"#"));  
 endif;  
 endif;  
  
 'write TCR in PLC if process valid  
 if Me.BP\_F == 1 then  
 LogMessage("Retour Mode validité = 1 (BP\_F = 1)");  
 MyContainer.fTCR = MyContainer.fkIP[4];  
 endif;  
  
'end R1ab >= kIP1  
  
else   
 if Me.BP\_F == 2 then  
 Me.sACTION = Me.Const.sPROCESS\_INSTABLE;  
 if MyContainer.bDebug == True then  
 LogMessage("Retour Mode validité = 2 (BP\_F = 2)");  
 endif;  
 else  
 Me.sACTION = Me.Const.sACTIVITE\_FAIBLE;  
 if MyContainer.bDebug == True then  
 LogMessage("BP\_F <> 1; BP\_F <> 2)");  
 endif;  
 endif;  
endif;  
  
' MODIF AND plus besoin c'est reporting services qui s'en charge  
'print the conduite report if pc expert  
' Pour éviter que 4 rapports ne sortent, si un des fours est en mode expert ça imprime  
'IF SE\_FOUR1.bPC\_EXPERT == 1 OR SE\_FOUR2.bPC\_EXPERT == 1 OR SE\_FOUR3.bPC\_EXPERT == 1 OR SE\_FOUR4.bPC\_EXPERT == 1 THEN  
 ' Il n'y a que le four1 qui gère l'impression du rapport conduites  
' IF Me.TagName == "SE\_RappAct\_FOUR1" THEN  
' MyContainer.dw\_SE\_Conduite.bPRN\_CONDUITE = true;  
' LogMessage("Print the conduite report");  
' ENDIF;  
'ENDIF;  
  
IF MyContainer.bPC\_EXPERT == 1 THEN  
 Me.bCreateRapportActivite = True;  
ENDIF;  
'mis en remarque le 2/2/2012 pour deplacer la ligne createrapportactivte  
'Me.bReset = True;  
  
if MyContainer.bDebug == True then  
 LogMessage("End CalcConsigne");  
endif;

#### Template ArchestrA $dwSystemExpert.dw\_SE\_ResLab

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bCalcIndicesXYAB | boolean |  |  |  |  |  |
| bCalcR1R2R3 | boolean |  |  |  |  |  |
| bResetCO2 | boolean |  |  |  |  |  |
| Const.iCYCLE\_DEF | integer |  |  |  |  |  |
| Const.iTEMPE\_CARNEAU | integer |  |  |  |  |  |
| Const.iTEMPE\_REF | integer |  |  |  |  |  |
| Const.iTEMPS\_ECHANTILLON | integer |  |  |  |  |  |
| fCO2\_B | float |  |  |  |  |  |
| fR1 | double |  |  |  |  |  |
| fR1ab | double |  |  |  |  |  |
| fR2 | double |  |  |  |  |  |
| fR2ab | double |  |  |  |  |  |
| fR3ab | double |  |  |  |  |  |
| fR4ab | double |  |  |  |  |  |
| iA | integer |  |  |  |  |  |
| iB | integer |  |  |  |  |  |
| iIndicePremierCO2 | integer |  |  |  |  |  |
| iLatLabo | integer |  |  |  |  |  |
| iNUM\_CO2 | integer |  |  |  |  |  |
| iTempsCycle | integer |  |  |  |  |  |
| iTempsInversion | integer |  |  |  |  |  |
| iTimeA | integer |  |  |  |  |  |
| iTimeB | integer |  |  |  |  |  |
| iTimeX | integer |  |  |  |  |  |
| iTimeY | integer |  |  |  |  |  |
| iValA | integer |  |  |  |  |  |
| iValB | integer |  |  |  |  |  |
| iValX | integer |  |  |  |  |  |
| iValY | integer |  |  |  |  |  |
| iX | integer |  |  |  |  |  |
| iY | integer |  |  |  |  |  |
| sDATE\_AU | string |  |  |  |  |  |
| sDATE\_CO2 | string |  |  |  |  |  |
| sDATE\_DU | string |  |  |  |  |  |
| sDER\_DATE | string |  |  |  |  |  |
| sDER\_HEURE | string |  |  |  |  |  |
| sHEURE\_AU | string |  |  |  |  |  |
| sHEURE\_CO2 | string |  |  |  |  |  |
| sHEURE\_DU | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcCO2\_1

|  |  |
| --- | --- |
| Name | dcCO2\_1 |
| Description |  |
| Trigger | DataChange of Me.fCO2\_B[1] |

**Declarations :**

Not Applicable

**Script :**

'script that executes when it is entered a new value for the first CO2  
  
Dim iNextCO2 as integer;  
Dim day as string;  
Dim month as string;  
Dim hour as string;  
Dim min as string;  
Dim sec as string;  
  
'set the current CO2 index  
Me.iNUM\_CO2 = 1;  
  
iNextCO2 = Me.iNUM\_CO2 + 1;  
  
'check the entered value  
'if all values reseted  
if Me.fCO2\_B[Me.iNUM\_CO2] == -1 then  
 Me.iNUM\_CO2 = 0;  
else  
  
if Me.fCO2\_B[Me.iNUM\_CO2] > 10 OR Me.fR1[iNextCO2] <> 0 then  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 NOT OK");  
 endif;  
else  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 OK");  
 endif;  
  
 MyContainer.dw\_SE\_RappAct.fCO2[Me.iNUM\_CO2] = Me.fCO2\_B[Me.iNUM\_CO2];  
  
 'Format DateTime  
  
 if StringToIntg( Now().Day ) <= 9 then  
 day = "0" + Now().Day;  
 else  
 day = Now().Day;  
 endif;  
 if StringToIntg( Now().Month ) <= 9 then  
 month = "0" + Now().Month;  
 else  
 month = Now().Month;  
 endif;  
 if StringToIntg( Now().Hour ) <= 9 then  
 hour = "0" + Now().Hour;  
 else  
 hour = Now().Hour;  
 endif;  
 if StringToIntg( Now().Minute ) <= 9 then  
 min = "0" + Now().Minute;  
 else  
 min = Now().Minute;  
 endif;  
 if StringToIntg( Now().Second ) <= 9 then  
 sec = "0" + Now().Second;  
 else  
 sec = Now().Second;  
 endif;  
  
 Me.sDATE\_CO2[Me.iNUM\_CO2] = day + "/" + month + "/" + Now().Year;  
 Me.sHEURE\_CO2[Me.iNUM\_CO2] = hour + ":" + min + ":" + sec;  
  
 'set the last date and time to display (= date and time for the last entered CO2)  
 Me.sDER\_DATE = Me.sDATE\_CO2[Me.iNUM\_CO2];  
 Me.sDER\_HEURE = Me.sHEURE\_CO2[Me.iNUM\_CO2];  
  
 'register the actual index for the first introducion of CO2  
 Me.iIndicePremierCO2 = MyContainer.iIndiceActuel;  
   
 'trigger the calculation of indexes X, Y, A, B  
 Me.bCalcIndicesXYAB = True;  
endif;  
  
endif;

CalcR1R2R3

|  |  |
| --- | --- |
| Name | CalcR1R2R3 |
| Description |  |
| Trigger | WhileTrue of Me.bCalcR1R2R3 |

**Declarations :**

Not Applicable

**Script :**

'script that calculates the values for R1, R2 and R3 for the current entered C02  
  
Dim MyDataSet as System.Data.DataSet;  
Dim Matrice as System.Data.DataTable;  
Dim MyDataRow as System.Data.DataRow;  
Dim foundRows[1] as System.Data.DataRow;  
Dim NumRecords as integer;  
Dim Expression as string;  
Dim iError as integer;  
Dim iNbreIndice as integer;  
Dim i as integer;  
  
if MyContainer.bDebug == True then  
 LogMessage("Start CalcR1R2R3");  
endif;  
  
'reset trigger  
Me.bCalcR1R2R3 = 0;  
  
' retreive MyDataSet from AppDomain  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_Matrice\_" + MyContainer.Tagname);  
' MODIF EXPERT AND 18/01/2012  
' BLOC IF..ELSE..ENDIF de sécurité  
if MyDataset <> null then  
Matrice = MyDataSet.Tables("Matrice");  
  
'calculate R1  
iNbreIndice = Me.iY - Me.iX;  
if iNbreIndice < 0 then  
 iNbreIndice = iNbreIndice + MyContainer.Const.iNBRE\_LIGNE;  
endif;  
if iNbreIndice == 0 then  
 iError = -1;  
endif;  
  
if MyContainer.bDebug == True then  
 LogMessage("iNbreIndice X-Y = " + Text(iNbreIndice,"#"));  
endif;  
  
if Me.iTimeY <> Me.iTimeX then  
 Me.fR1[Me.iNUM\_CO2] = iNbreIndice \* 100/((Me.iTimeY - Me.iTimeX)/(Me.iTempsCycle + Me.iTempsInversion));  
else  
 Me.fR1[Me.iNUM\_CO2] = 0;  
endif;  
if Me.fR1[Me.iNUM\_CO2] > 100 then Me.fR1[Me.iNUM\_CO2] = 100; endif;  
  
if MyContainer.bDebug == True then  
 LogMessage("fR1[" + Me.iNUM\_CO2 + "] = " + Text(Me.fR1[Me.iNUM\_CO2],"#"));  
endif;  
  
  
'calculate R2  
  
Me.fR2[Me.iNUM\_CO2] = Me.iValY[Me.Const.iTEMPE\_CARNEAU];  
  
if MyContainer.bDebug == True then  
 LogMessage("initial fR2[" + Me.iNUM\_CO2 + "] = " + Text(Me.fR2[Me.iNUM\_CO2],"#"));  
endif;  
  
i = Me.iX;  
while i <> Me.iY  
  
''read the record   
 Expression = "";  
 ' query  
 Expression = "Indice = '" + i + "'";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression);  
  
 'test if the index exists  
 if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
 exit while;  
 endif;  
  
 MyDataRow = foundRows[1];  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
 'if matrix row found, read matrix values for index i  
 if NumRecords == 1 then  
 Me.iValX[Me.Const.iTEMPE\_CARNEAU] = MyDataRow("FM\_V1001");  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iValX[" + Me.Const.iTEMPE\_CARNEAU + "] = " + Text(Me.iValX[Me.Const.iTEMPE\_CARNEAU],"#.#"));  
 endif;  
 else  
 iError = -1;  
 endif;  
 if iError == 0 then  
 Me.fR2[Me.iNUM\_CO2] = Me.fR2[Me.iNUM\_CO2] + Me.iValX[Me.Const.iTEMPE\_CARNEAU];  
 endif;  
  
 if i == MyContainer.Const.iNBRE\_LIGNE then i = 0; endif;  
 i = i + 1;  
endwhile;  
  
if iError == 0 then  
  
 Me.fR2[Me.iNUM\_CO2] = Me.fR2[Me.iNUM\_CO2] /(iNbreIndice + 1);  
  
 if MyContainer.bDebug == True then  
 LogMessage("fR2[" + Me.iNUM\_CO2 + "] = " + Text(Me.fR2[Me.iNUM\_CO2],"#"));  
 endif;  
  
 'calculate R3ab  
 Me.fR3ab = 0;  
 for i = 1 to Me.iNUM\_CO2  
 Me.fR3ab = Me.fR3ab + MyContainer.dw\_SE\_RappAct.fCO2[i];   
 next;  
 Me.fR3ab = Me.fR3ab/Me.iNUM\_CO2;   
  
 if MyContainer.bDebug == True then  
 LogMessage("Me.fR3ab = " + Text(Me.fR3ab,"#.##"));  
 endif;  
  
 'calculate R1ab  
 iNbreIndice = Me.iB - Me.iA;  
 if iNbreIndice < 0 then  
 iNbreIndice = iNbreIndice + MyContainer.Const.iNBRE\_LIGNE;  
 endif;  
 if iNbreIndice == 0 then  
 iError = -1;  
 endif;  
  
 if MyContainer.bDebug == True then  
 LogMessage("iNbreIndice A-B = " + Text(iNbreIndice,"#"));  
 endif;  
  
 if ((Me.iTempsCycle + Me.iTempsInversion) <> 0) AND (Me.iTimeA <> Me.iTimeB) then  
 Me.fR1ab = iNbreIndice \* 100/((Me.iTimeB - Me.iTimeA)/(Me.iTempsCycle + Me.iTempsInversion));  
 else  
 iError = -1;  
 endif;  
 if Me.fR1ab > 100 then Me.fR1ab = 100; endif;  
  
 if MyContainer.bDebug == True then  
 LogMessage("fR1ab = " + Text(Me.fR1ab,"#"));  
 endif;  
  
 'calculate R2  
  
 Me.fR2ab = Me.iValB[Me.Const.iTEMPE\_CARNEAU];  
 Me.fR4ab = Me.iValB[Me.Const.iTEMPE\_REF];  
  
 i = Me.iA;  
 while i <> Me.iB  
 'read the record   
 Expression = "";  
 ' query  
 Expression = "Indice = '" + i + "'";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression);  
  
 'test if the index exists  
 if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
 exit while;  
 endif;  
  
 MyDataRow = foundRows[1];  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
 'if matrix row found, read matrix values for index i  
 if NumRecords == 1 then  
 Me.iValA[Me.Const.iTEMPE\_CARNEAU] = MyDataRow("FM\_V1001");  
 Me.iValA[Me.Const.iTEMPE\_REF] = MyDataRow("FM\_V1007");  
 else  
 iError = -1;  
 endif;  
 if iError == 0 then  
 Me.fR2ab = Me.fR2ab + Me.iValA[Me.Const.iTEMPE\_CARNEAU];  
 Me.fR4ab = Me.fR4ab + Me.iValA[Me.Const.iTEMPE\_REF];  
 endif;  
  
 if i == MyContainer.Const.iNBRE\_LIGNE then i = 0; endif;  
 i = i + 1;  
 endwhile;  
  
 if iError == 0 then  
  
 Me.fR2ab = Me.fR2ab /(iNbreIndice + 1);  
 Me.fR4ab = Me.fR4ab /(iNbreIndice + 1);  
  
 if MyContainer.bDebug == True then  
 LogMessage("fR2ab = " + Text(Me.fR2ab,"#"));  
 LogMessage("fR4ab = " + Text(Me.fR4ab,"#"));  
 endif;  
  
 endif;  
  
endif;  
  
if MyContainer.bDebug == True then  
 LogMessage("End CalcR1R2R3. Error = " + Text(iError, "#"));  
endif;  
' MODIF EXPERT AND 18/01/2012  
' BLOC IF..ELSE..ENDIF de sécurité  
else  
 logmessage("SYSTEME EXPERT - impossible de calculer, matrice non mémorisée");  
endif;

dcCO2\_2

|  |  |
| --- | --- |
| Name | dcCO2\_2 |
| Description |  |
| Trigger | DataChange of Me.fCO2\_B[2] |

**Declarations :**

Not Applicable

**Script :**

'script that executes when it is entered a new value for the second CO2  
  
Dim iNextCO2 as integer;  
Dim iPrevCO2 as integer;  
Dim day as string;  
Dim month as string;  
Dim hour as string;  
Dim min as string;  
Dim sec as string;  
  
'set the current CO2 index  
Me.iNUM\_CO2 = 2;  
  
iNextCO2 = Me.iNUM\_CO2 + 1;  
iPrevCO2 = Me.iNUM\_CO2 - 1;  
  
'check the entered value  
'if all values reseted  
if Me.fCO2\_B[Me.iNUM\_CO2] == -1 then  
 Me.iNUM\_CO2 = 0;  
else  
  
if Me.fCO2\_B[Me.iNUM\_CO2] > 10 OR Me.fR1[iNextCO2] <> 0 OR Me.fR1[iPrevCO2] == 0 then  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 NOT OK");  
 endif;  
else  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 OK");  
 endif;  
 MyContainer.dw\_SE\_RappAct.fCO2[Me.iNUM\_CO2] = Me.fCO2\_B[Me.iNUM\_CO2];  
  
 'Format DateTime  
  
 if StringToIntg( Now().Day ) <= 9 then  
 day = "0" + Now().Day;  
 else  
 day = Now().Day;  
 endif;  
 if StringToIntg( Now().Month ) <= 9 then  
 month = "0" + Now().Month;  
 else  
 month = Now().Month;  
 endif;  
 if StringToIntg( Now().Hour ) <= 9 then  
 hour = "0" + Now().Hour;  
 else  
 hour = Now().Hour;  
 endif;  
 if StringToIntg( Now().Minute ) <= 9 then  
 min = "0" + Now().Minute;  
 else  
 min = Now().Minute;  
 endif;  
 if StringToIntg( Now().Second ) <= 9 then  
 sec = "0" + Now().Second;  
 else  
 sec = Now().Second;  
 endif;  
  
 Me.sDATE\_CO2[Me.iNUM\_CO2] = day + "/" + month + "/" + Now().Year;  
 Me.sHEURE\_CO2[Me.iNUM\_CO2] = hour + ":" + min + ":" + sec;  
  
 'set the last date and time to display (= date and time for the last entered CO2)  
 Me.sDER\_DATE = Me.sDATE\_CO2[Me.iNUM\_CO2];  
 Me.sDER\_HEURE = Me.sHEURE\_CO2[Me.iNUM\_CO2];  
   
 'trigger the calculation of indexes X, Y, A, B  
 Me.bCalcIndicesXYAB = 1;  
endif;  
  
endif;

dcCO2\_3

|  |  |
| --- | --- |
| Name | dcCO2\_3 |
| Description |  |
| Trigger | DataChange of Me.fCO2\_B[3] |

**Declarations :**

Not Applicable

**Script :**

'script that executes when it is entered a new value for 3rd CO2  
  
Dim iNextCO2 as integer;  
Dim iPrevCO2 as integer;  
Dim day as string;  
Dim month as string;  
Dim hour as string;  
Dim min as string;  
Dim sec as string;  
  
'set the current CO2 index  
Me.iNUM\_CO2 = 3;  
  
iNextCO2 = Me.iNUM\_CO2 + 1;  
iPrevCO2 = Me.iNUM\_CO2 - 1;  
  
'check the entered value  
'if all values reseted  
if Me.fCO2\_B[Me.iNUM\_CO2] == -1 then  
 Me.iNUM\_CO2 = 0;  
else  
  
if Me.fCO2\_B[Me.iNUM\_CO2] > 10 OR Me.fR1[iNextCO2] <> 0 OR Me.fR1[iPrevCO2] == 0 then  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 NOT OK");  
 endif;  
else  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 OK");  
 endif;  
 MyContainer.dw\_SE\_RappAct.fCO2[Me.iNUM\_CO2] = Me.fCO2\_B[Me.iNUM\_CO2];  
  
 'Format DateTime  
  
 if StringToIntg( Now().Day ) <= 9 then  
 day = "0" + Now().Day;  
 else  
 day = Now().Day;  
 endif;   
 if StringToIntg( Now().Month ) <= 9 then  
 month = "0" + Now().Month;  
 else  
 month = Now().Month;  
 endif;  
 if StringToIntg( Now().Hour ) <= 9 then  
 hour = "0" + Now().Hour;  
 else  
 hour = Now().Hour;  
 endif;  
 if StringToIntg( Now().Minute ) <= 9 then  
 min = "0" + Now().Minute;  
 else  
 min = Now().Minute;  
 endif;  
 if StringToIntg( Now().Second ) <= 9 then  
 sec = "0" + Now().Second;  
 else  
 sec = Now().Second;  
 endif;  
  
 Me.sDATE\_CO2[Me.iNUM\_CO2] = day + "/" + month + "/" + Now().Year;  
 Me.sHEURE\_CO2[Me.iNUM\_CO2] = hour + ":" + min + ":" + sec;  
  
 'set the last date and time to display (= date and time for the last entered CO2)  
 Me.sDER\_DATE = Me.sDATE\_CO2[Me.iNUM\_CO2];  
 Me.sDER\_HEURE = Me.sHEURE\_CO2[Me.iNUM\_CO2];  
   
 'trigger the calculation of indexes X, Y, A, B  
 Me.bCalcIndicesXYAB = 1;  
endif;  
  
endif;

dcCO2\_4

|  |  |
| --- | --- |
| Name | dcCO2\_4 |
| Description |  |
| Trigger | DataChange of Me.fCO2\_B[4] |

**Declarations :**

Not Applicable

**Script :**

'script that executes when it is entered a new value for the 4th CO2  
  
Dim iNextCO2 as integer;  
Dim iPrevCO2 as integer;  
Dim day as string;  
Dim month as string;  
Dim hour as string;  
Dim min as string;  
Dim sec as string;  
  
'set the current CO2 index  
Me.iNUM\_CO2 = 4;  
  
iNextCO2 = Me.iNUM\_CO2 + 1;  
iPrevCO2 = Me.iNUM\_CO2 - 1;  
  
'check the entered value  
'if all values reseted  
if Me.fCO2\_B[Me.iNUM\_CO2] == -1 then  
 Me.iNUM\_CO2 = 0;  
else  
  
if Me.fCO2\_B[Me.iNUM\_CO2] > 10 OR Me.fR1[iNextCO2] <> 0 OR Me.fR1[iPrevCO2] == 0 then  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 NOT OK");  
 endif;  
else  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 OK");  
 endif;  
 MyContainer.dw\_SE\_RappAct.fCO2[Me.iNUM\_CO2] = Me.fCO2\_B[Me.iNUM\_CO2];  
  
 'Format DateTime  
  
 if StringToIntg( Now().Day ) <= 9 then  
 day = "0" + Now().Day;  
 else  
 day = Now().Day;  
 endif;  
 if StringToIntg( Now().Month ) <= 9 then  
 month = "0" + Now().Month;  
 else  
 month = Now().Month;  
 endif;  
 if StringToIntg( Now().Hour ) <= 9 then  
 hour = "0" + Now().Hour;  
 else  
 hour = Now().Hour;  
 endif;  
 if StringToIntg( Now().Minute ) <= 9 then  
 min = "0" + Now().Minute;  
 else  
 min = Now().Minute;  
 endif;  
 if StringToIntg( Now().Second ) <= 9 then  
 sec = "0" + Now().Second;  
 else  
 sec = Now().Second;  
 endif;  
  
 Me.sDATE\_CO2[Me.iNUM\_CO2] = day + "/" + month + "/" + Now().Year;  
 Me.sHEURE\_CO2[Me.iNUM\_CO2] = hour + ":" + min + ":" + sec;  
  
 'set the last date and time to display (= date and time for the last entered CO2)  
 Me.sDER\_DATE = Me.sDATE\_CO2[Me.iNUM\_CO2];  
 Me.sDER\_HEURE = Me.sHEURE\_CO2[Me.iNUM\_CO2];  
   
 'trigger the calculation of indexes X, Y, A, B  
 Me.bCalcIndicesXYAB = 1;  
endif;  
  
endif;

dcCO2\_5

|  |  |
| --- | --- |
| Name | dcCO2\_5 |
| Description |  |
| Trigger | DataChange of Me.fCO2\_B[5] |

**Declarations :**

Not Applicable

**Script :**

'script that executes when it is entered a new value for the 5th CO2  
  
Dim iNextCO2 as integer;  
Dim iPrevCO2 as integer;  
Dim day as string;  
Dim month as string;  
Dim hour as string;  
Dim min as string;  
Dim sec as string;  
  
'set the current CO2 index  
Me.iNUM\_CO2 = 5;  
  
iNextCO2 = Me.iNUM\_CO2 + 1;  
iPrevCO2 = Me.iNUM\_CO2 - 1;  
  
'check the entered value  
'if all values reseted  
if Me.fCO2\_B[Me.iNUM\_CO2] == -1 then  
 Me.iNUM\_CO2 = 0;  
else  
  
if Me.fCO2\_B[Me.iNUM\_CO2] > 10 OR Me.fR1[iNextCO2] <> 0 OR Me.fR1[iPrevCO2] == 0 then  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 NOT OK");  
 endif;  
else  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 OK");  
 endif;  
 MyContainer.dw\_SE\_RappAct.fCO2[Me.iNUM\_CO2] = Me.fCO2\_B[Me.iNUM\_CO2];  
  
 'Format DateTime  
  
 if StringToIntg( Now().Day ) <= 9 then  
 day = "0" + Now().Day;  
 else  
 day = Now().Day;  
 endif;  
 if StringToIntg( Now().Month ) <= 9 then  
 month = "0" + Now().Month;  
 else  
 month = Now().Month;  
 endif;  
 if StringToIntg( Now().Hour ) <= 9 then  
 hour = "0" + Now().Hour;  
 else  
 hour = Now().Hour;  
 endif;  
 if StringToIntg( Now().Minute ) <= 9 then  
 min = "0" + Now().Minute;  
 else  
 min = Now().Minute;  
 endif;  
 if StringToIntg( Now().Second ) <= 9 then  
 sec = "0" + Now().Second;  
 else  
 sec = Now().Second;  
 endif;  
  
 Me.sDATE\_CO2[Me.iNUM\_CO2] = day + "/" + month + "/" + Now().Year;  
 Me.sHEURE\_CO2[Me.iNUM\_CO2] = hour + ":" + min + ":" + sec;  
  
 'set the last date and time to display (= date and time for the last entered CO2)  
 Me.sDER\_DATE = Me.sDATE\_CO2[Me.iNUM\_CO2];  
 Me.sDER\_HEURE = Me.sHEURE\_CO2[Me.iNUM\_CO2];  
   
 'trigger the calculation of indexes X, Y, A, B  
 Me.bCalcIndicesXYAB = 1;  
endif;  
  
endif;

dcCO2\_6

|  |  |
| --- | --- |
| Name | dcCO2\_6 |
| Description |  |
| Trigger | DataChange of Me.fCO2\_B[6] |

**Declarations :**

Not Applicable

**Script :**

'script that executes when it is entered a new value for the 6th CO2  
  
Dim iNextCO2 as integer;  
Dim iPrevCO2 as integer;  
Dim day as string;  
Dim month as string;  
Dim hour as string;  
Dim min as string;  
Dim sec as string;  
  
'set the current CO2 index  
Me.iNUM\_CO2 = 6;  
  
iNextCO2 = Me.iNUM\_CO2 + 1;  
iPrevCO2 = Me.iNUM\_CO2 - 1;  
  
'check the entered value  
'if all values reseted  
if Me.fCO2\_B[Me.iNUM\_CO2] == -1 then  
 Me.iNUM\_CO2 = 0;  
else  
  
if Me.fCO2\_B[Me.iNUM\_CO2] > 10 OR Me.fR1[iNextCO2] <> 0 OR Me.fR1[iPrevCO2] == 0 then  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 NOT OK");  
 endif;  
else  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 OK");  
 endif;  
 MyContainer.dw\_SE\_RappAct.fCO2[Me.iNUM\_CO2] = Me.fCO2\_B[Me.iNUM\_CO2];  
  
 'Format DateTime  
  
 if StringToIntg( Now().Day ) <= 9 then  
 day = "0" + Now().Day;  
 else  
 day = Now().Day;  
 endif;  
 if StringToIntg( Now().Month ) <= 9 then  
 month = "0" + Now().Month;  
 else  
 month = Now().Month;  
 endif;  
 if StringToIntg( Now().Hour ) <= 9 then  
 hour = "0" + Now().Hour;  
 else  
 hour = Now().Hour;  
 endif;  
 if StringToIntg( Now().Minute ) <= 9 then  
 min = "0" + Now().Minute;  
 else  
 min = Now().Minute;  
 endif;  
 if StringToIntg( Now().Second ) <= 9 then  
 sec = "0" + Now().Second;  
 else  
 sec = Now().Second;  
 endif;  
  
 Me.sDATE\_CO2[Me.iNUM\_CO2] = day + "/" + month + "/" + Now().Year;  
 Me.sHEURE\_CO2[Me.iNUM\_CO2] = hour + ":" + min + ":" + sec;  
  
 'set the last date and time to display (= date and time for the last entered CO2)  
 Me.sDER\_DATE = Me.sDATE\_CO2[Me.iNUM\_CO2];  
 Me.sDER\_HEURE = Me.sHEURE\_CO2[Me.iNUM\_CO2];  
   
 'trigger the calculation of indexes X, Y, A, B  
 Me.bCalcIndicesXYAB = 1;  
endif;  
  
endif;

dcCO2\_7

|  |  |
| --- | --- |
| Name | dcCO2\_7 |
| Description |  |
| Trigger | DataChange of Me.fCO2\_B[7] |

**Declarations :**

Not Applicable

**Script :**

'script that executes when it is entered a new value for the 7th CO2  
  
dim iNextCO2 as integer;  
dim iPrevCO2 as integer;  
Dim day as string;  
Dim month as string;  
Dim hour as string;  
Dim min as string;  
Dim sec as string;  
  
'set the current CO2 index  
Me.iNUM\_CO2 = 7;  
  
iNextCO2 = Me.iNUM\_CO2 + 1;  
iPrevCO2 = Me.iNUM\_CO2 - 1;  
  
'check the entered value  
'if all values reseted  
if Me.fCO2\_B[Me.iNUM\_CO2] == -1 then  
 Me.iNUM\_CO2 = 0;  
else  
  
if Me.fCO2\_B[Me.iNUM\_CO2] > 10 OR Me.fR1[iNextCO2] <> 0 OR Me.fR1[iPrevCO2] == 0 then  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 NOT OK");  
 endif;  
else  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 OK");  
 endif;  
 MyContainer.dw\_SE\_RappAct.fCO2[Me.iNUM\_CO2] = Me.fCO2\_B[Me.iNUM\_CO2];  
  
 'Format DateTime  
  
 if StringToIntg( Now().Day ) <= 9 then  
 day = "0" + Now().Day;  
 else  
 day = Now().Day;  
 endif;  
 if StringToIntg( Now().Month ) <= 9 then  
 month = "0" + Now().Month;  
 else  
 month = Now().Month;  
 endif;  
 if StringToIntg( Now().Hour ) <= 9 then  
 hour = "0" + Now().Hour;  
 else  
 hour = Now().Hour;  
 endif;  
 if StringToIntg( Now().Minute ) <= 9 then  
 min = "0" + Now().Minute;  
 else  
 min = Now().Minute;  
 endif;  
 if StringToIntg( Now().Second ) <= 9 then  
 sec = "0" + Now().Second;  
 else  
 sec = Now().Second;  
 endif;  
  
 Me.sDATE\_CO2[Me.iNUM\_CO2] = day + "/" + month + "/" + Now().Year;  
 Me.sHEURE\_CO2[Me.iNUM\_CO2] = hour + ":" + min + ":" + sec;  
  
 'set the last date and time to display (= date and time for the last entered CO2)  
 Me.sDER\_DATE = Me.sDATE\_CO2[Me.iNUM\_CO2];  
 Me.sDER\_HEURE = Me.sHEURE\_CO2[Me.iNUM\_CO2];  
   
 'trigger the calculation of indexes X, Y, A, B  
 Me.bCalcIndicesXYAB = 1;  
endif;  
  
endif;

dcCO2\_12

|  |  |
| --- | --- |
| Name | dcCO2\_12 |
| Description |  |
| Trigger | DataChange of Me.fCO2\_B[12] |

**Declarations :**

Not Applicable

**Script :**

'script that executes when it is entered a new value for the 12th CO2  
  
Dim iPrevCO2 as integer;  
Dim day as string;  
Dim month as string;  
Dim hour as string;  
Dim min as string;  
Dim sec as string;  
  
'set the current CO2 index  
Me.iNUM\_CO2 = 12;  
  
iPrevCO2 = Me.iNUM\_CO2 - 1;  
  
'check the entered value  
'if all values reseted  
if Me.fCO2\_B[Me.iNUM\_CO2] == -1 then  
 Me.iNUM\_CO2 = 0;  
else  
  
if Me.fCO2\_B[Me.iNUM\_CO2] > 10 OR Me.fR1[iPrevCO2] == 0 then  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 NOT OK");  
 endif;  
else  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 OK");  
 endif;  
 MyContainer.dw\_SE\_RappAct.fCO2[Me.iNUM\_CO2] = Me.fCO2\_B[Me.iNUM\_CO2];  
  
 'Format DateTime  
  
 if StringToIntg( Now().Day ) <= 9 then  
 day = "0" + Now().Day;  
 else  
 day = Now().Day;  
 endif;  
 if StringToIntg( Now().Month ) <= 9 then  
 month = "0" + Now().Month;  
 else  
 month = Now().Month;  
 endif;  
 if StringToIntg( Now().Hour ) <= 9 then  
 hour = "0" + Now().Hour;  
 else  
 hour = Now().Hour;  
 endif;  
 if StringToIntg( Now().Minute ) <= 9 then  
 min = "0" + Now().Minute;  
 else  
 min = Now().Minute;  
 endif;  
 if StringToIntg( Now().Second ) <= 9 then  
 sec = "0" + Now().Second;  
 else  
 sec = Now().Second;  
 endif;  
  
 Me.sDATE\_CO2[Me.iNUM\_CO2] = day + "/" + month + "/" + Now().Year;  
 Me.sHEURE\_CO2[Me.iNUM\_CO2] = hour + ":" + min + ":" + sec;  
  
 'set the last date and time to display (= date and time for the last entered CO2)  
 Me.sDER\_DATE = Me.sDATE\_CO2[Me.iNUM\_CO2];  
 Me.sDER\_HEURE = Me.sHEURE\_CO2[Me.iNUM\_CO2];  
   
 'trigger the calculation of indexes X, Y, A, B  
 Me.bCalcIndicesXYAB = 1;  
endif;  
  
endif;

dcCO2\_8

|  |  |
| --- | --- |
| Name | dcCO2\_8 |
| Description |  |
| Trigger | DataChange of Me.fCO2\_B[8] |

**Declarations :**

Not Applicable

**Script :**

'script that executes when it is entered a new value for the 7th CO2  
  
Dim iNextCO2 as integer;  
Dim iPrevCO2 as integer;  
Dim day as string;  
Dim month as string;  
Dim hour as string;  
Dim min as string;  
Dim sec as string;  
  
'set the current CO2 index  
Me.iNUM\_CO2 = 8;  
  
iNextCO2 = Me.iNUM\_CO2 + 1;  
iPrevCO2 = Me.iNUM\_CO2 - 1;  
  
'check the entered value  
'if all values reseted  
if Me.fCO2\_B[Me.iNUM\_CO2] == -1 then  
 Me.iNUM\_CO2 = 0;  
else  
  
if Me.fCO2\_B[Me.iNUM\_CO2] > 10 OR Me.fR1[iNextCO2] <> 0 OR Me.fR1[iPrevCO2] == 0 then  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 NOT OK");  
 endif;  
else  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 OK");  
 endif;  
 MyContainer.dw\_SE\_RappAct.fCO2[Me.iNUM\_CO2] = Me.fCO2\_B[Me.iNUM\_CO2];  
  
 'Format DateTime  
  
 if StringToIntg( Now().Day ) <= 9 then  
 day = "0" + Now().Day;  
 else  
 day = Now().Day;  
 endif;  
 if StringToIntg( Now().Month ) <= 9 then  
 month = "0" + Now().Month;  
 else  
 month = Now().Month;  
 endif;  
 if StringToIntg( Now().Hour ) <= 9 then  
 hour = "0" + Now().Hour;  
 else  
 hour = Now().Hour;  
 endif;  
 if StringToIntg( Now().Minute ) <= 9 then  
 min = "0" + Now().Minute;  
 else  
 min = Now().Minute;  
 endif;  
 if StringToIntg( Now().Second ) <= 9 then  
 sec = "0" + Now().Second;  
 else  
 sec = Now().Second;  
 endif;  
  
 Me.sDATE\_CO2[Me.iNUM\_CO2] = day + "/" + month + "/" + Now().Year;  
 Me.sHEURE\_CO2[Me.iNUM\_CO2] = hour + ":" + min + ":" + sec;  
  
 'set the last date and time to display (= date and time for the last entered CO2)  
 Me.sDER\_DATE = Me.sDATE\_CO2[Me.iNUM\_CO2];  
 Me.sDER\_HEURE = Me.sHEURE\_CO2[Me.iNUM\_CO2];  
   
 'trigger the calculation of indexes X, Y, A, B  
 Me.bCalcIndicesXYAB = 1;  
endif;  
  
endif;

dcCO2\_9

|  |  |
| --- | --- |
| Name | dcCO2\_9 |
| Description |  |
| Trigger | DataChange of Me.fCO2\_B[9] |

**Declarations :**

Not Applicable

**Script :**

'script that executes when it is entered a new value for the 7th CO2  
  
Dim iNextCO2 as integer;  
Dim iPrevCO2 as integer;  
Dim day as string;  
Dim month as string;  
Dim hour as string;  
Dim min as string;  
Dim sec as string;  
  
'set the current CO2 index  
Me.iNUM\_CO2 = 9;  
  
iNextCO2 = Me.iNUM\_CO2 + 1;  
iPrevCO2 = Me.iNUM\_CO2 - 1;  
  
'check the entered value  
'if all values reseted  
if Me.fCO2\_B[Me.iNUM\_CO2] == -1 then  
 Me.iNUM\_CO2 = 0;  
else  
  
if Me.fCO2\_B[Me.iNUM\_CO2] > 10 OR Me.fR1[iNextCO2] <> 0 OR Me.fR1[iPrevCO2] == 0 then  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 NOT OK");  
 endif;  
else  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 OK");  
 endif;  
 MyContainer.dw\_SE\_RappAct.fCO2[Me.iNUM\_CO2] = Me.fCO2\_B[Me.iNUM\_CO2];  
  
 'Format DateTime  
  
 if StringToIntg( Now().Day ) <= 9 then  
 day = "0" + Now().Day;  
 else  
 day = Now().Day;  
 endif;  
 if StringToIntg( Now().Month ) <= 9 then  
 month = "0" + Now().Month;  
 else  
 month = Now().Month;  
 endif;  
 if StringToIntg( Now().Hour ) <= 9 then  
 hour = "0" + Now().Hour;  
 else  
 hour = Now().Hour;  
 endif;  
 if StringToIntg( Now().Minute ) <= 9 then  
 min = "0" + Now().Minute;  
 else  
 min = Now().Minute;  
 endif;  
 if StringToIntg( Now().Second ) <= 9 then  
 sec = "0" + Now().Second;  
 else  
 sec = Now().Second;  
 endif;  
  
 Me.sDATE\_CO2[Me.iNUM\_CO2] = day + "/" + month + "/" + Now().Year;  
 Me.sHEURE\_CO2[Me.iNUM\_CO2] = hour + ":" + min + ":" + sec;  
  
 'set the last date and time to display (= date and time for the last entered CO2)  
 Me.sDER\_DATE = Me.sDATE\_CO2[Me.iNUM\_CO2];  
 Me.sDER\_HEURE = Me.sHEURE\_CO2[Me.iNUM\_CO2];  
   
 'trigger the calculation of indexes X, Y, A, B  
 Me.bCalcIndicesXYAB = 1;  
endif;  
  
endif;

dcCO2\_10

|  |  |
| --- | --- |
| Name | dcCO2\_10 |
| Description |  |
| Trigger | DataChange of Me.fCO2\_B[10] |

**Declarations :**

Not Applicable

**Script :**

'script that executes when it is entered a new value for the 7th CO2  
  
Dim iNextCO2 as integer;  
Dim iPrevCO2 as integer;  
Dim day as string;  
Dim month as string;  
Dim hour as string;  
Dim min as string;  
Dim sec as string;  
  
'set the current CO2 index  
Me.iNUM\_CO2 = 10;  
  
iNextCO2 = Me.iNUM\_CO2 + 1;  
iPrevCO2 = Me.iNUM\_CO2 - 1;  
  
'check the entered value  
'if all values reseted  
if Me.fCO2\_B[Me.iNUM\_CO2] == -1 then  
 Me.iNUM\_CO2 = 0;  
else  
  
if Me.fCO2\_B[Me.iNUM\_CO2] > 10 OR Me.fR1[iNextCO2] <> 0 OR Me.fR1[iPrevCO2] == 0 then  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 NOT OK");  
 endif;  
else  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 OK");  
 endif;  
 MyContainer.dw\_SE\_RappAct.fCO2[Me.iNUM\_CO2] = Me.fCO2\_B[Me.iNUM\_CO2];  
  
 'Format DateTime  
  
 if StringToIntg( Now().Day ) <= 9 then  
 day = "0" + Now().Day;  
 else  
 day = Now().Day;  
 endif;  
 if StringToIntg( Now().Month ) <= 9 then  
 month = "0" + Now().Month;  
 else  
 month = Now().Month;  
 endif;  
 if StringToIntg( Now().Hour ) <= 9 then  
 hour = "0" + Now().Hour;  
 else  
 hour = Now().Hour;  
 endif;  
 if StringToIntg( Now().Minute ) <= 9 then  
 min = "0" + Now().Minute;  
 else  
 min = Now().Minute;  
 endif;  
 if StringToIntg( Now().Second ) <= 9 then  
 sec = "0" + Now().Second;  
 else  
 sec = Now().Second;  
 endif;  
  
 Me.sDATE\_CO2[Me.iNUM\_CO2] = day + "/" + month + "/" + Now().Year;  
 Me.sHEURE\_CO2[Me.iNUM\_CO2] = hour + ":" + min + ":" + sec;  
  
 'set the last date and time to display (= date and time for the last entered CO2)  
 Me.sDER\_DATE = Me.sDATE\_CO2[Me.iNUM\_CO2];  
 Me.sDER\_HEURE = Me.sHEURE\_CO2[Me.iNUM\_CO2];  
   
 'trigger the calculation of indexes X, Y, A, B  
 Me.bCalcIndicesXYAB = 1;  
endif;  
  
endif;

dcCO2\_11

|  |  |
| --- | --- |
| Name | dcCO2\_11 |
| Description |  |
| Trigger | DataChange of Me.fCO2\_B[11] |

**Declarations :**

Not Applicable

**Script :**

'script that executes when it is entered a new value for the 7th CO2  
  
Dim iNextCO2 as integer;  
Dim iPrevCO2 as integer;  
Dim day as string;  
Dim month as string;  
Dim hour as string;  
Dim min as string;  
Dim sec as string;  
  
'set the current CO2 index  
Me.iNUM\_CO2 = 11;  
  
iNextCO2 = Me.iNUM\_CO2 + 1;  
iPrevCO2 = Me.iNUM\_CO2 - 1;  
  
'check the entered value  
'if all values reseted  
if Me.fCO2\_B[Me.iNUM\_CO2] == -1 then  
 Me.iNUM\_CO2 = 0;  
else  
  
if Me.fCO2\_B[Me.iNUM\_CO2] > 10 OR Me.fR1[iNextCO2] <> 0 OR Me.fR1[iPrevCO2] == 0 then  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 NOT OK");  
 endif;  
else  
 if MyContainer.bDebug == True then  
 LogMessage("Value CO2 OK");  
 endif;  
 MyContainer.dw\_SE\_RappAct.fCO2[Me.iNUM\_CO2] = Me.fCO2\_B[Me.iNUM\_CO2];  
  
 'Format DateTime  
  
 if StringToIntg( Now().Day ) <= 9 then  
 day = "0" + Now().Day;  
 else  
 day = Now().Day;  
 endif;  
 if StringToIntg( Now().Month ) <= 9 then  
 month = "0" + Now().Month;  
 else  
 month = Now().Month;  
 endif;  
 if StringToIntg( Now().Hour ) <= 9 then  
 hour = "0" + Now().Hour;  
 else  
 hour = Now().Hour;  
 endif;  
 if StringToIntg( Now().Minute ) <= 9 then  
 min = "0" + Now().Minute;  
 else  
 min = Now().Minute;  
 endif;  
 if StringToIntg( Now().Second ) <= 9 then  
 sec = "0" + Now().Second;  
 else  
 sec = Now().Second;  
 endif;  
  
 Me.sDATE\_CO2[Me.iNUM\_CO2] = day + "/" + month + "/" + Now().Year;  
 Me.sHEURE\_CO2[Me.iNUM\_CO2] = hour + ":" + min + ":" + sec;  
  
 'set the last date and time to display (= date and time for the last entered CO2)  
 Me.sDER\_DATE = Me.sDATE\_CO2[Me.iNUM\_CO2];  
 Me.sDER\_HEURE = Me.sHEURE\_CO2[Me.iNUM\_CO2];  
   
 'trigger the calculation of indexes X, Y, A, B  
 Me.bCalcIndicesXYAB = 1;  
endif;  
  
endif;

CalcIndicesXYAB

|  |  |
| --- | --- |
| Name | CalcIndicesXYAB |
| Description |  |
| Trigger | WhileTrue of Me.bCalcIndicesXYAB |

**Declarations :**

Not Applicable

**Script :**

'script that calculates the values for indexes X, Y, A, B corresponding to the current entered C02  
'and reads the matrix values for each of these indexes  
  
Dim iError as integer;  
Dim MyDataSet as System.Data.DataSet;  
Dim Matrice as System.Data.DataTable;  
Dim MyDataRow as System.Data.DataRow;  
Dim foundRows[1] as System.Data.DataRow;  
Dim NumRecords as integer;  
Dim Expression as string;  
Dim iIndexSpace as integer;  
Dim sTimeX as string;  
Dim sTimeY as string;  
Dim sTimeA as string;  
Dim sTimeB as string;  
  
if MyContainer.bDebug == True then  
 LogMessage("Start CalcIndicesXYAB");  
endif;  
  
'reset trigger  
Me.bCalcIndicesXYAB = False;  
  
'reset indexes X, Y, A, B  
Me.iX = 0;  
Me.iY = 0;  
Me.iA = 0;  
Me.iB = 0;  
  
' retreive MyDataSet from AppDomain  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_Matrice\_" + MyContainer.Tagname);  
' MODIF EXPERT AND 18/01/2012  
' BLOC IF..ELSE..ENDIF de sécurité  
if MyDataset <> null then  
Matrice = MyDataSet.Tables("Matrice");  
  
'read the cycle time = address V1014  
Me.iTempsCycle = MyContainer.FM\_V1014;  
  
'read the inversion time = address V1013  
Me.iTempsInversion = MyContainer.FM\_V1013;  
  
'read the latence labo time = kIP16  
Me.iLatLabo = MyContainer.fkIP[16];  
  
if MyContainer.bDebug == True then  
 LogMessage("iFM\_V[15] = Me.iTempsCycle = " + Text(Me.iTempsCycle,"#.#"));  
 LogMessage("iFM\_V[14] = Me.iTempsInversion = " + Text(Me.iTempsInversion,"#.#"));  
 LogMessage("fkIP[16] = Me.iLatLabo = " + Text(Me.iLatLabo,"#.#"));  
endif;  
  
if Me.iTempsCycle <= 0 OR Me.iTempsInversion <=0 OR Me.iLatLabo <= 0 then  
 iError = -1;  
else  
 iError = 0;  
endif;  
  
'calculate the X index  
if iError == 0 then  
 if (Me.iTempsCycle + Me.iTempsInversion) <> 0 then  
 Me.iX = MyContainer.iIndiceActuel - (Me.iLatLabo/(Me.iTempsCycle + Me.iTempsInversion)) -   
 (Me.Const.iTEMPS\_ECHANTILLON/(Me.iTempsCycle + Me.iTempsInversion)) - Me.Const.iCYCLE\_DEF;  
 else  
 Me.iX = 0;  
 endif;  
  
 'check the indexes values  
 if Me.iX < 1 then  
 Me.iX = Me.iX + MyContainer.Const.iNBRE\_LIGNE;  
 endif;  
 if Me.iX > MyContainer.Const.iNBRE\_LIGNE then  
 iError = -1;  
 endif;  
  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iX = " + Text(Me.iX,"#"));  
 endif;  
  
endif;  
  
'calculate the Y index  
if iError == 0 then  
 if (Me.iTempsCycle + Me.iTempsInversion) <> 0 then  
 Me.iY = MyContainer.iIndiceActuel - (Me.iLatLabo/(Me.iTempsCycle + Me.iTempsInversion)) - Me.Const.iCYCLE\_DEF;  
 else  
 Me.iY = 0;  
 endif;  
  
 'check the indexes values  
 if Me.iY < 1 then  
 Me.iY = Me.iY + MyContainer.Const.iNBRE\_LIGNE;  
 endif;  
 if Me.iY > MyContainer.Const.iNBRE\_LIGNE then  
 iError = -1;  
 endif;  
  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iY = " + Text(Me.iY,"#"));  
 endif;  
endif;  
  
'read the record corresponding to index X  
if iError == 0 then  
 ' init expression  
 Expression = "";  
 ' query  
 Expression = "Indice = '" + Me.iX + "'";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression);  
  
 'test if the index exists  
 if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
 endif;  
  
 if iError == 0 then  
  
 MyDataRow = foundRows[1];  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
 'if matrix row found, read matrix values for index X  
 if NumRecords == 1 then  
 Me.iValX[2] = MyDataRow("FM\_V1001");  
 Me.iValX[8] = MyDataRow("FM\_V1007");  
 Me.iTimeX = MyDataRow("DateTime\_SEC");  
 sTimeX = MyDataRow("DateTime");  
 'split the DateTime string in Date and Time  
 iIndexSpace = sTimeX.IndexOf(" ", 0);  
 Me.sDATE\_DU[Me.iNUM\_CO2] = StringMid( sTimeX, 0, iIndexSpace );  
 Me.sHEURE\_DU[Me.iNUM\_CO2] = StringMid( sTimeX, iIndexSpace + 2, (sTimeX.Length - iIndexSpace) );  
 if MyContainer.bDebug == True then  
 LogMessage("Me.sTimeX = " + sTimeX);  
 LogMessage("Date\_DU\_X = " + Me.sDATE\_DU[Me.iNUM\_CO2]);  
 LogMessage("Heure\_DU\_X = " + Me.sHEURE\_DU[Me.iNUM\_CO2]);  
 endif;  
 else  
 iError = -1;  
 endif;  
   
 endif;  
  
endif;  
  
'read the record corresponding to index Y  
if iError == 0 then  
 ' init expression  
 Expression = "";  
 ' query  
 Expression = "Indice = '" + Me.iY + "'";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression);  
  
 'test if the index exists  
 if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
 endif;  
  
 if iError == 0 then   
  
 MyDataRow = foundRows[1];  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
 'if matrix row found, read matrix values for index Y  
 if NumRecords == 1 then  
 Me.iValY[2] = MyDataRow("FM\_V1001");  
 Me.iValY[8] = MyDataRow("FM\_V1007");  
 Me.iTimeY = MyDataRow("DateTime\_SEC");  
 sTimeY = MyDataRow("DateTime");  
 'split the DateTime string in Date and Time  
 iIndexSpace = sTimeY.IndexOf(" ", 0);  
 Me.sDATE\_AU[Me.iNUM\_CO2] = StringMid( sTimeY, 0, iIndexSpace );  
 Me.sHEURE\_AU[Me.iNUM\_CO2] = StringMid( sTimeY, iIndexSpace + 2, (sTimeY.Length - iIndexSpace) );  
 if MyContainer.bDebug == True then  
 LogMessage("Me.sTimeY = " + sTimeY);  
 LogMessage("Date\_AU\_Y = " + Me.sDATE\_AU[Me.iNUM\_CO2]);  
 LogMessage("Heure\_AU\_Y = " + Me.sHEURE\_AU[Me.iNUM\_CO2]);  
 endif;  
 else  
 iError = -1;  
 endif;  
  
 endif;  
  
endif;  
  
'calculate the A index  
if iError == 0 then  
 if (Me.iTempsCycle + Me.iTempsInversion) <> 0 then  
 Me.iA = Me.iIndicePremierCO2 - (Me.iLatLabo/(Me.iTempsCycle + Me.iTempsInversion)) -   
 (Me.Const.iTEMPS\_ECHANTILLON/(Me.iTempsCycle + Me.iTempsInversion)) - Me.Const.iCYCLE\_DEF;  
 endif;  
  
 'check the indexes values  
 if Me.iA < 1 then  
 Me.iA = Me.iA + MyContainer.Const.iNBRE\_LIGNE;  
 endif;  
 if Me.iA > MyContainer.Const.iNBRE\_LIGNE then  
 iError = -1;  
 endif;  
  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iA = " + Text(Me.iA,"#"));  
 endif;  
endif;  
  
'calculate the B index  
if iError == 0 then  
 if (Me.iTempsCycle + Me.iTempsInversion) <> 0 then  
 Me.iB = MyContainer.iIndiceActuel - (Me.iLatLabo/(Me.iTempsCycle + Me.iTempsInversion)) - Me.Const.iCYCLE\_DEF;  
 else  
 Me.iB = 0;  
 endif;  
  
 'check the indexes values  
 if Me.iB < 1 then  
 Me.iB = Me.iB + MyContainer.Const.iNBRE\_LIGNE;  
 endif;  
 if Me.iB > MyContainer.Const.iNBRE\_LIGNE then  
 iError = -1;  
 endif;  
  
 if MyContainer.bDebug == True then  
 LogMessage("Me.iB = " + Text(Me.iB,"#"));  
 endif;  
endif;  
  
'read the record corresponding to index A  
if iError == 0 then  
 ' init expression  
 Expression = "";  
 ' query  
 Expression = "Indice = '" + Me.iA + "'";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression);  
  
 'test if the index exists  
 if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
 endif;  
  
 if iError == 0 then  
  
 MyDataRow = foundRows[1];  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
 'if matrix row found, read matrix values for index A  
 if NumRecords == 1 then  
 Me.iValA[2] = MyDataRow("FM\_V1001");  
 Me.iValA[8] = MyDataRow("FM\_V1007");  
 'get the DateTime for index A  
 Me.iTimeA = MyDataRow("DateTime\_SEC");  
 else  
 iError = -1;  
 endif;  
  
 endif;  
  
endif;  
  
'read the record corresponding to index B  
if iError == 0 then  
 ' init expression  
 Expression = "";  
 ' query  
 Expression = "Indice = '" + Me.iY + "'";  
 'retreive data  
 foundRows[] = Matrice.Select(Expression);  
  
 'test if the index exists  
 if foundRows[].GetUpperBound(0) == -1 then  
 if MyContainer.bDebug == True then  
 LogMessage("Index not found");  
 endif;  
 iError = -1;  
 endif;  
  
 if iError == 0 then  
  
 MyDataRow = foundRows[1];  
 NumRecords = foundRows[].GetUpperBound(0) + 1;  
 'if matrix row found, read matrix values for index B  
 if NumRecords == 1 then  
 Me.iValB[2] = MyDataRow("FM\_V1001");  
 Me.iValB[8] = MyDataRow("FM\_V1007");  
 'get the DateTime for index B  
 Me.iTimeB = MyDataRow("DateTime\_SEC");  
 else  
 iError = -1;  
 endif;  
   
 endif;  
  
endif;  
  
if iError == 0 then  
 'trigger the calculation of R1, R2, R3  
 Me.bCalcR1R2R3 = 1;  
else  
 if MyContainer.bDebug == True then  
 LogMessage("Error in calculating indexes X, Y, A, B");  
 endif;  
endif;  
  
if MyContainer.bDebug == True then  
 LogMessage("End CalcIndicesXYAB. Error = " + Text(iError, "#"));  
endif;  
  
' MODIF EXPERT AND 18/01/2012  
' BLOC IF..ELSE..ENDIF de sécurité  
else  
 logmessage("SYSTEME EXPERT - impossible de calculer, matrice non mémorisée");  
endif;

ResetCO2

|  |  |
| --- | --- |
| Name | ResetCO2 |
| Description |  |
| Trigger | WhileTrue of Me.bResetCO2 |

**Declarations :**

Not Applicable

**Script :**

'script that reset the display page for CO2 labo  
  
Dim i as integer;  
  
if MyContainer.bDebug == True then  
 LogMessage("Reset CO2");  
endif;  
  
'reset trigger  
Me.bResetCO2 = False;  
  
For i = 1 to 12  
 'reset CO2 values  
 Me.fCO2\_B[i] = -1;  
 MyContainer.dw\_SE\_RappAct.fCO2[i] = 0;  
 'reset date time  
 Me.sDATE\_CO2[i] = "";  
 Me.sDATE\_AU[i] = "";  
 Me.sDATE\_DU[i] = "";  
 Me.sHEURE\_CO2[i] = "";  
 Me.sHEURE\_AU[i] = "";  
 Me.sHEURE\_DU[i] = "";  
 Me.fR1[i] = 0;  
 Me.fR2[i] = 0;  
 Me.fR1ab = 0;  
 Me.fR2ab = 0;  
 Me.fR3ab = 0;  
 Me.fR4ab = 0;  
Next;  
  
Me.iNUM\_CO2 = 0;  
Me.sDER\_DATE = "";  
Me.sDER\_HEURE = "";  
MyContainer.dw\_SE\_RappAct.sBP\_DATE = "";  
MyContainer.dw\_SE\_RappAct.sBP\_HEURE = "";

#### Template ArchestrA $dwTransporteur1S

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwMoteurDirect1S

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmeBourrage | boolean |  | X |  | X |  |
| bCommandeInhibitionBourrage | boolean | X |  |  |  |  |
| fCourant | float |  | X |  |  | X |
| iTempsVidange | integer | X |  |  |  |  |
| sAlarmeBourrage | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwTransporteur1S\_BR8

##### Description

Not Applicable

##### Derived from

$dwMoteurDirect1S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwTransporteur1SAvecDebit

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTransporteur1S

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bCmdResetTotaliseur | boolean | X |  |  |  |  |
| fDebit | float |  | X |  |  | X |
| fTotaliseur | float |  | X |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwTransporteur1SDebit\_BR8

##### Description

Not Applicable

##### Derived from

$dwTransporteur1S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwTransporteur2S

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwMoteurDirect2S

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| iTempsVidange | integer | X |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwTransporteur2S\_BR8

##### Description

Not Applicable

##### Derived from

$dwMoteurDirect2S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwTSFCObjet

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bSelection | boolean |  | X |  |  |  |
| iNbForcage | integer |  |  |  |  |  |
| iPriorite | integer |  |  |  |  |  |
| sDescription | string |  |  |  |  |  |
| sItem | string |  |  |  |  |  |
| sTabTagForcage | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

scSelection

|  |  |
| --- | --- |
| Name | scSelection |
| Description |  |
| Trigger | DataChange of Me.bSelection |

**Declarations :**

Not Applicable

**Script :**

if Me.bSelection then  
 Me.iPriorite.value=100;  
else  
 Me.iPriorite.value=200;  
endif;

#### Template ArchestrA $dwVanne

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwActionneur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmeDisc2Fdc | boolean |  | X |  | X |  |
| bAlarmeFermeture | boolean |  | X |  | X |  |
| bAlarmeOuverture | boolean |  | X |  | X |  |
| bCommandeFermetureManu | boolean | X |  |  |  |  |
| bCommandeInhibDisc2Fdc | boolean | X |  |  |  |  |
| bCommandeInhibDiscFermeture | boolean | X |  |  |  |  |
| bCommandeInhibDiscOuverture | boolean | X |  |  |  |  |
| bCommandeOuvertureManu | boolean | X |  |  |  |  |
| bStatusAutorisationFermeture | boolean |  | X |  |  |  |
| bStatusAutorisationOuverture | boolean |  | X |  |  |  |
| bStatusFerme | boolean |  | X |  |  |  |
| bStatusOuvert | boolean |  | X |  |  |  |
| iTempsDef | integer | X |  |  |  |  |
| sAlarmeDisc2Fdc | string |  |  |  |  |  |
| sAlarmeFermeture | string |  |  |  |  |  |
| sAlarmeOuverture | string |  |  |  |  |  |
| sTexteFerme | string |  |  |  |  |  |
| sTexteOuvert | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

pdicNbForcage

|  |  |
| --- | --- |
| Name | pdicNbForcage |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

Dim nb as Integer;  
  
nb=0;  
  
IF Me.bCommandeInhibDisc2Fdc == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibDiscFermeture == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibDiscOuverture == false THEN  
 nb = nb + 1;  
ENDIF;  
  
Me.iNbForcage = nb;

#### Template ArchestrA $dwVanne1S

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwVanne

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwVanne2S

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwVanne

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwVanneMotorisee2S

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwVanne

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmeDefAU | boolean |  | X |  | X |  |
| bAlarmeDefIS | boolean |  | X |  | X |  |
| bAlarmeDefMTH | boolean |  | X |  | X |  |
| bAlarmeDefRKMAr | boolean |  | X |  | X |  |
| bAlarmeDefRKMAv | boolean |  | X |  | X |  |
| bCommandeInhibitionDefIS | boolean | X |  |  |  |  |
| bCommandeInhibitionDefMTH | boolean | X |  |  |  |  |
| bCommandeInhibitionDefRKMAr | boolean | X |  |  |  |  |
| bCommandeInhibitionDefRKMAv | boolean | X |  |  |  |  |
| iCommandeTempsDefRKM | integer | X |  |  |  |  |
| sAlarmeDefAU | string |  |  |  |  |  |
| sAlarmeDefIS | string |  |  |  |  |  |
| sAlarmeDefMTH | string |  |  |  |  |  |
| sAlarmeDefTKMAr | string |  |  |  |  |  |
| sAlarmeDefTKMAv | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwVanneMotorisee2SG4

##### Description

Not Applicable

##### Derived from

$dwActionneur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bCommandeFermetureManu | boolean |  |  |  |  |  |
| bCommandeFermetureManu\_1 | boolean | X |  |  |  |  |
| bCommandeFermetureManu\_2 | boolean | X |  |  |  |  |
| bCommandeFermetureManu\_3 | boolean | X |  |  |  |  |
| bCommandeFermetureManu\_4 | boolean | X |  |  |  |  |
| bCommandeOuvertureManu | boolean |  |  |  |  |  |
| bCommandeOuvertureManu\_1 | boolean | X |  |  |  |  |
| bCommandeOuvertureManu\_2 | boolean | X |  |  |  |  |
| bCommandeOuvertureManu\_3 | boolean | X |  |  |  |  |
| bCommandeOuvertureManu\_4 | boolean | X |  |  |  |  |
| bSelection\_1 | boolean |  | X |  |  |  |
| bSelection\_2 | boolean |  | X |  |  |  |
| bSelection\_3 | boolean |  | X |  |  |  |
| bSelection\_4 | boolean |  | X |  |  |  |
| bSelectionCalc | boolean |  |  |  |  |  |
| bStatusAutorisationFermeture | boolean |  |  |  |  |  |
| bStatusAutorisationFermeture\_1 | boolean |  | X |  |  |  |
| bStatusAutorisationFermeture\_2 | boolean |  | X |  |  |  |
| bStatusAutorisationFermeture\_3 | boolean |  | X |  |  |  |
| bStatusAutorisationFermeture\_4 | boolean |  | X |  |  |  |
| bStatusAutorisationOuverture | boolean |  |  |  |  |  |
| bStatusAutorisationOuverture\_1 | boolean |  | X |  |  |  |
| bStatusAutorisationOuverture\_2 | boolean |  | X |  |  |  |
| bStatusAutorisationOuverture\_3 | boolean |  | X |  |  |  |
| bStatusAutorisationOuverture\_4 | boolean |  | X |  |  |  |
| bStatusDisponible\_1 | boolean |  | X |  |  |  |
| bStatusDisponible\_2 | boolean |  | X |  |  |  |
| bStatusDisponible\_3 | boolean |  | X |  |  |  |
| bStatusDisponible\_4 | boolean |  | X |  |  |  |
| bStatusDisponibleCalc | boolean |  |  |  |  |  |
| bStatusEnDefaut\_1 | boolean |  | X |  |  |  |
| bStatusEnDefaut\_2 | boolean |  | X |  |  |  |
| bStatusEnDefaut\_3 | boolean |  | X |  |  |  |
| bStatusEnDefaut\_4 | boolean |  | X |  |  |  |
| bStatusEnDefautCalc | integer |  |  |  |  |  |
| bStatusFerme | boolean |  |  |  |  |  |
| bStatusFerme\_1 | boolean |  | X |  |  |  |
| bStatusFerme\_2 | boolean |  | X |  |  |  |
| bStatusFerme\_3 | boolean |  | X |  |  |  |
| bStatusFerme\_4 | boolean |  | X |  |  |  |
| bStatusModeAuto\_1 | boolean |  | X |  |  |  |
| bStatusModeAuto\_2 | boolean |  | X |  |  |  |
| bStatusModeAuto\_3 | boolean |  | X |  |  |  |
| bStatusModeAuto\_4 | boolean |  | X |  |  |  |
| bStatusModeAutoCalc | boolean |  |  |  |  |  |
| bStatusModeLocal\_1 | boolean |  | X |  |  |  |
| bStatusModeLocal\_2 | boolean |  | X |  |  |  |
| bStatusModeLocal\_3 | boolean |  | X |  |  |  |
| bStatusModeLocal\_4 | boolean |  | X |  |  |  |
| bStatusModeLocalCalc | boolean |  |  |  |  |  |
| bStatusOuvert | boolean |  |  |  |  |  |
| bStatusOuvert\_1 | boolean |  | X |  |  |  |
| bStatusOuvert\_2 | boolean |  | X |  |  |  |
| bStatusOuvert\_3 | boolean |  | X |  |  |  |
| bStatusOuvert\_4 | boolean |  | X |  |  |  |
| sTabTagnames | string |  |  |  |  |  |
| sTexteFerme | string |  |  |  |  |  |
| sTexteOuvert | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

bCommandeFermetureManu2Vannes

|  |  |
| --- | --- |
| Name | bCommandeFermetureManu2Vannes |
| Description |  |
| Trigger | OnTrue of Me.bCommandeFermetureManu |

**Declarations :**

Not Applicable

**Script :**

Me.bCommandeFermetureManu\_1=true;  
Me.bCommandeFermetureManu\_2=true;  
Me.bCommandeFermetureManu\_3=true;  
Me.bCommandeFermetureManu\_4=true;

bCommandeOuvertureManu2Vannes

|  |  |
| --- | --- |
| Name | bCommandeOuvertureManu2Vannes |
| Description |  |
| Trigger | OnTrue of Me.bCommandeOuvertureManu |

**Declarations :**

Not Applicable

**Script :**

Me.bCommandeOuvertureManu\_1=true;  
Me.bCommandeOuvertureManu\_2=true;  
Me.bCommandeOuvertureManu\_3=true;  
Me.bCommandeOuvertureManu\_4=true;

Vannes2bCommandeOuvertureManu

|  |  |
| --- | --- |
| Name | Vannes2bCommandeOuvertureManu |
| Description |  |
| Trigger | WhileFalse of Me.bCommandeOuvertureManu\_1 or Me.bCommandeOuvertureManu\_2 or Me.bCommandeOuvertureManu\_3 or Me.bCommandeOuvertureManu\_4 |

**Declarations :**

Not Applicable

**Script :**

'logmessage("Me.bCommandeOuvertureManu=false;");  
Me.bCommandeOuvertureManu=false;

Vannes2bCommandeFermetureManu

|  |  |
| --- | --- |
| Name | Vannes2bCommandeFermetureManu |
| Description |  |
| Trigger | WhileFalse of Me.bCommandeFermetureManu\_1 or Me.bCommandeFermetureManu\_2 or Me.bCommandeFermetureManu\_3 or Me.bCommandeFermetureManu\_4 |

**Declarations :**

Not Applicable

**Script :**

'logmessage("Me.bCommandeFermetureManu=False;");  
Me.bCommandeFermetureManu=False;

V2bStatusAutorisationFermeture

|  |  |
| --- | --- |
| Name | V2bStatusAutorisationFermeture |
| Description |  |
| Trigger | DataChange of Me.bStatusAutorisationFermeture\_1 and Me.bStatusAutorisationFermeture\_2 and Me.bStatusAutorisationFermeture\_3 and Me.bStatusAutorisationFermeture\_4 |

**Declarations :**

Not Applicable

**Script :**

Me.bStatusAutorisationFermeture = Me.bStatusAutorisationFermeture\_1   
 and Me.bStatusAutorisationFermeture\_2   
 and Me.bStatusAutorisationFermeture\_3   
 and Me.bStatusAutorisationFermeture\_4;

V2bStatusAutorisationOuverture

|  |  |
| --- | --- |
| Name | V2bStatusAutorisationOuverture |
| Description |  |
| Trigger | DataChange of Me.bStatusAutorisationOuverture\_1 and Me.bStatusAutorisationOuverture\_2 and Me.bStatusAutorisationOuverture\_3 and Me.bStatusAutorisationOuverture\_4 |

**Declarations :**

Not Applicable

**Script :**

Me.bStatusAutorisationOuverture = Me.bStatusAutorisationOuverture\_1   
 and Me.bStatusAutorisationOuverture\_2   
 and Me.bStatusAutorisationOuverture\_3   
 and Me.bStatusAutorisationOuverture\_4;

V2bStatusEnDefautCalc

|  |  |
| --- | --- |
| Name | V2bStatusEnDefautCalc |
| Description |  |
| Trigger | DataChange of Me.bStatusEnDefaut\_1 or Me.bStatusEnDefaut\_2 or Me.bStatusEnDefaut\_3 or Me.bStatusEnDefaut\_4 |

**Declarations :**

Not Applicable

**Script :**

Me.bStatusEnDefautCalc = Me.bStatusEnDefaut\_1   
 or Me.bStatusEnDefaut\_2   
 or Me.bStatusEnDefaut\_3   
 or Me.bStatusEnDefaut\_4;

V2bStatusFerme

|  |  |
| --- | --- |
| Name | V2bStatusFerme |
| Description |  |
| Trigger | DataChange of Me.bStatusFerme\_1 and Me.bStatusFerme\_2 and Me.bStatusFerme\_3 and Me.bStatusFerme\_4 |

**Declarations :**

Not Applicable

**Script :**

Me.bStatusFerme = Me.bStatusFerme\_1   
 and Me.bStatusFerme\_2   
 and Me.bStatusFerme\_3   
 and Me.bStatusFerme\_4;

V2bStatusOuvert

|  |  |
| --- | --- |
| Name | V2bStatusOuvert |
| Description |  |
| Trigger | DataChange of Me.bStatusOuvert\_1 and Me.bStatusOuvert\_2 and Me.bStatusOuvert\_3 and Me.bStatusOuvert\_4 |

**Declarations :**

Not Applicable

**Script :**

Me.bStatusOuvert = Me.bStatusOuvert\_1   
 and Me.bStatusOuvert\_2   
 and Me.bStatusOuvert\_3   
 and Me.bStatusOuvert\_4;

V2bStatusModeAutoCalc

|  |  |
| --- | --- |
| Name | V2bStatusModeAutoCalc |
| Description |  |
| Trigger | DataChange of Me.bStatusModeAuto\_1 or Me.bStatusModeAuto\_2 or Me.bStatusModeAuto\_3 or Me.bStatusModeAuto\_4 |

**Declarations :**

Not Applicable

**Script :**

Me.bStatusModeAutoCalc = Me.bStatusModeAuto\_1   
 or Me.bStatusModeAuto\_2   
 or Me.bStatusModeAuto\_3   
 or Me.bStatusModeAuto\_4;

V2bStatusModeLocalCalc

|  |  |
| --- | --- |
| Name | V2bStatusModeLocalCalc |
| Description |  |
| Trigger | DataChange of Me.bStatusModeLocal\_1 or Me.bStatusModeLocal\_2 or Me.bStatusModeLocal\_3 or Me.bStatusModeLocal\_4 |

**Declarations :**

Not Applicable

**Script :**

Me.bStatusModeLocalCalc = Me.bStatusModeLocal\_1   
 or Me.bStatusModeLocal\_2   
 or Me.bStatusModeLocal\_3   
 or Me.bStatusModeLocal\_4;

V2bSelectionCalc

|  |  |
| --- | --- |
| Name | V2bSelectionCalc |
| Description |  |
| Trigger | DataChange of Me.bSelection\_1 or Me.bSelection\_2 or Me.bSelection\_3 or Me.bSelection\_4 |

**Declarations :**

Not Applicable

**Script :**

Me.bSelectionCalc = Me.bSelection\_1   
 or Me.bSelection\_2   
 or Me.bSelection\_3   
 or Me.bSelection\_4;

V2bStatusDisponibleCalc

|  |  |
| --- | --- |
| Name | V2bStatusDisponibleCalc |
| Description |  |
| Trigger | DataChange of Me.bStatusDisponible\_1 and Me.bStatusDisponible\_2 and Me.bStatusDisponible\_3 and Me.bStatusDisponible\_4 |

**Declarations :**

Not Applicable

**Script :**

Me.bStatusDisponibleCalc = Me.bStatusDisponible\_1   
 and Me.bStatusDisponible\_2   
 and Me.bStatusDisponible\_3   
 and Me.bStatusDisponible\_4;

#### Template ArchestrA $dwVanneMotorisee2SG4.dwVariateur

##### Description

Not Applicable

##### Derived from

$dwActionneur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmeDefAU | boolean |  | X |  | X |  |
| bAlarmeDefDRIVE | boolean |  | X |  | X |  |
| bAlarmeDefIS | boolean |  | X |  | X |  |
| bAlarmeDefMTH | boolean |  | X |  | X |  |
| bAlarmeDefRKM | boolean |  | X |  | X |  |
| bAlarmeDefRot | boolean |  | X |  | X |  |
| bAlarmeDefTemp | boolean |  | X |  | X |  |
| bCommandeArretManu | boolean | X |  |  |  |  |
| bCommandeInhibitionDefDRIVE | boolean | X |  |  |  |  |
| bCommandeInhibitionDefIS | boolean | X |  |  |  |  |
| bCommandeInhibitionDefMTH | boolean | X |  |  |  |  |
| bCommandeInhibitionDefRKM | boolean | X |  |  |  |  |
| bCommandeInhibitionDefRot | boolean | X |  |  |  |  |
| bCommandeInhibitionDefTemp | boolean | X |  |  |  |  |
| bCommandeMarcheManu | boolean | X |  |  |  |  |
| bStatusAutorisationMarche | boolean |  | X |  |  |  |
| bStatusMarche | boolean |  | X |  |  |  |
| bStatusOrdreMarche | boolean |  | X |  |  |  |
| fPVCourant | float |  | X |  |  |  |
| fPVVitesse | float |  | X |  |  |  |
| fSPVitesse | float | X |  |  |  |  |
| fSPVitesseMaxEU | float |  |  |  |  |  |
| fSPVitesseMinEU | float |  |  |  |  |  |
| sAlarmeDefAU | string |  |  |  |  |  |
| sAlarmeDefDRIVE | string |  |  |  |  |  |
| sAlarmeDefIS | string |  |  |  |  |  |
| sAlarmeDefMTH | string |  |  |  |  |  |
| sAlarmeDefRKM | string |  |  |  |  |  |
| sAlarmeDefRot | string |  |  |  |  |  |
| sAlarmeDefTemp | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

pdicNbForcage

|  |  |
| --- | --- |
| Name | pdicNbForcage |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

Dim nb as Integer;  
  
nb=0;  
  
IF Me.bCommandeInhibitionDefDRIVE == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefIS == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefMTH == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefRKM == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefRot == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefTemp == false THEN  
 nb = nb + 1;  
ENDIF;  
  
Me.iNbForcage = nb;

#### Template ArchestrA $dwVanneMotorisee3P

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwVanneMotorisee2S

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bStatusMilieu | boolean |  | X |  |  |  |
| sTexteMilieu | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwVariateur

##### Description

Not Applicable

##### Derived from

$dwActionneur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmeDefAU | boolean |  | X |  | X |  |
| bAlarmeDefDRIVE | boolean |  | X |  | X |  |
| bAlarmeDefIS | boolean |  | X |  | X |  |
| bAlarmeDefMTH | boolean |  | X |  | X |  |
| bAlarmeDefRKM | boolean |  | X |  | X |  |
| bAlarmeDefRot | boolean |  | X |  | X |  |
| bAlarmeDefTemp | boolean |  | X |  | X |  |
| bCommandeArretManu | boolean | X |  |  |  |  |
| bCommandeInhibitionDefDRIVE | boolean | X |  |  |  |  |
| bCommandeInhibitionDefIS | boolean | X |  |  |  |  |
| bCommandeInhibitionDefMTH | boolean | X |  |  |  |  |
| bCommandeInhibitionDefRKM | boolean | X |  |  |  |  |
| bCommandeInhibitionDefRot | boolean | X |  |  |  |  |
| bCommandeInhibitionDefTemp | boolean | X |  |  |  |  |
| bCommandeMarcheManu | boolean | X |  |  |  |  |
| bStatusAutorisationMarche | boolean |  | X |  |  |  |
| bStatusMarche | boolean |  | X |  |  |  |
| bStatusOrdreMarche | boolean |  | X |  |  |  |
| fPVCourant | float |  | X |  |  |  |
| fPVVitesse | float |  | X |  |  |  |
| fSPVitesse | float | X |  |  |  |  |
| fSPVitesseMaxEU | float |  |  |  |  |  |
| fSPVitesseMinEU | float |  |  |  |  |  |
| sAlarmeDefAU | string |  |  |  |  |  |
| sAlarmeDefDRIVE | string |  |  |  |  |  |
| sAlarmeDefIS | string |  |  |  |  |  |
| sAlarmeDefMTH | string |  |  |  |  |  |
| sAlarmeDefRKM | string |  |  |  |  |  |
| sAlarmeDefRot | string |  |  |  |  |  |
| sAlarmeDefTemp | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

pdicNbForcage

|  |  |
| --- | --- |
| Name | pdicNbForcage |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

Dim nb as Integer;  
  
nb=0;  
  
IF Me.bCommandeInhibitionDefDRIVE == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefIS == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefMTH == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefRKM == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefRot == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefTemp == false THEN  
 nb = nb + 1;  
ENDIF;  
  
Me.iNbForcage = nb;

#### Template ArchestrA $dwVariateur2S

##### Description

Not Applicable

##### Derived from

$dwActionneur

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmeDefAU | boolean |  | X |  | X |  |
| bAlarmeDefDRIVE | boolean |  | X |  | X |  |
| bAlarmeDefIS | boolean |  | X |  | X |  |
| bAlarmeDefMTH | boolean |  | X |  | X |  |
| bAlarmeDefRKM | boolean |  | X |  | X |  |
| bAlarmeDefRot | boolean |  | X |  | X |  |
| bAlarmeDefTemp | boolean |  | X |  | X |  |
| bCommandeArretManu | boolean | X |  |  |  |  |
| bCommandeInhibitionDefDRIVE | boolean | X |  |  |  |  |
| bCommandeInhibitionDefIS | boolean | X |  |  |  |  |
| bCommandeInhibitionDefMTH | boolean | X |  |  |  |  |
| bCommandeInhibitionDefRKM | boolean | X |  |  |  |  |
| bCommandeInhibitionDefRot | boolean | X |  |  |  |  |
| bCommandeInhibitionDefTemp | boolean | X |  |  |  |  |
| bCommandeMarcheArManu | boolean | X |  |  |  |  |
| bCommandeMarcheAvManu | boolean | X |  |  |  |  |
| bStatusAutorisationMarcheAr | boolean |  | X |  |  |  |
| bStatusAutorisationMarcheAv | boolean |  | X |  |  |  |
| bStatusMarcheArriere | boolean |  | X |  |  |  |
| bStatusMarcheAvant | boolean |  | X |  |  |  |
| bStatusOrdreMarcheAr | boolean |  | X |  |  |  |
| bStatusOrdreMarcheAv | boolean |  | X |  |  |  |
| fPVCourant | float |  | X |  |  |  |
| fPVVitesse | float |  | X |  |  |  |
| fSPVitesse | float | X |  |  |  |  |
| fSPVitesseMaxEU | float |  |  |  |  |  |
| fSPVitesseMinEU | float |  |  |  |  |  |
| iCommandeTempsDefRKM | integer | X |  |  |  |  |
| iCommandeTempsDefRot | integer | X |  |  |  |  |
| sAlarmeDefAU | string |  |  |  |  |  |
| sAlarmeDefDRIVE | string |  |  |  |  |  |
| sAlarmeDefIS | string |  |  |  |  |  |
| sAlarmeDefMTH | string |  |  |  |  |  |
| sAlarmeDefRKM | string |  |  |  |  |  |
| sAlarmeDefRot | string |  |  |  |  |  |
| sAlarmeDefTemp | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

pdicNbForcage

|  |  |
| --- | --- |
| Name | pdicNbForcage |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

Dim nb as Integer;  
  
nb=0;  
  
IF Me.bCommandeInhibitionDefDRIVE == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefIS == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefMTH == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefRKM == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefRot == false THEN  
 nb = nb + 1;  
ENDIF;  
  
IF Me.bCommandeInhibitionDefTemp == false THEN  
 nb = nb + 1;  
ENDIF;  
  
Me.iNbForcage = nb;

#### Template ArchestrA $dwVariateurCable

##### Description

Not Applicable

##### Derived from

$dwVanneMotorisee2SG4.dwVariateur

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwVariateurCable2S

##### Description

Not Applicable

##### Derived from

$dwVariateur2S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwVentilateur\_BR8

##### Description

Not Applicable

##### Derived from

$dwMoteurDirect1S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwVireur\_BR8

##### Description

Not Applicable

##### Derived from

$dwMoteurDirect1S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwVis1S\_BR8

##### Description

Not Applicable

##### Derived from

$dwMoteurDirect1S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwVis2S\_BR8

##### Description

Not Applicable

##### Derived from

$dwMoteurDirect2S\_BR8

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwWatchDog

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarme | boolean |  |  |  | X |  |
| bAlerte | boolean |  |  |  | X |  |
| iMotViePrec | integer |  |  |  |  |  |
| iNbErreurs | integer |  |  |  |  |  |

##### Field Attributes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Attribute name | Data Type | IO Scaling | History | Limit Alarms | Rate Of Change Alarms | Target Deviation Alarms | Bad Value Alarm | Statistics |
| iPlcLifeWord | integer |  |  |  |  |  |  |  |

##### Scripts

pdicWatchDog

|  |  |
| --- | --- |
| Name | pdicWatchDog |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

' Author : CCO  
' Date : 22 Avril 2008  
' Script : pdicWatchDog  
' Description : Ce script va vérifier toutes les 2 secondes si la valeur de l'entier de vie bouge.  
' Si la valeur de l'entier de vie stockée est la même que celle lue, un entier s'incrémente.  
' Après 3 erreurs, cet entier produit une ALERTE, après 5 erreurs, une ALARME.  
' L'entier est remis à zéro lorsqu'une valeur est correctement lue.  
  
IF (Me.iPlcLifeWord == Me.iMotViePrec) OR Not IsGood(Me.iPlcLifeWord) THEN  
 ' Erreur : le mot de vie n'a pas bougé : problème de communication  
 ' AJOUT CCO : Vérification de la qualité du Watchdog : Bad = Alarme  
 Me.iNbErreurs = Me.iNbErreurs + 1;  
ELSE  
 ' OK  
 Me.iMotViePrec = Me.iPlcLifeWord;  
 Me.iNbErreurs = 0;  
ENDIF;  
  
IF Me.iNbErreurs < 3 THEN  
 ' OK  
 Me.bAlerte = 0;  
 Me.bAlarme = 0;  
ELSEIF Me.iNbErreurs >=3 AND Me.iNbErreurs < 5 THEN  
 ' ALERTE  
 Me.bAlerte = 1;  
ELSEIF Me.iNbErreurs >= 5 THEN  
 ' ALARME  
 Me.bAlarme = 1;  
ENDIF;

#### Template ArchestrA $Filtre

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$Moteur1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $Float

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$FieldReference

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $FR4\_CONSO\_LIG\_CHAUX\_10\_50\_S5\_TR

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwGenQTE

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $GroupeHydraulique

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$Moteur1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $IDF\_aOEE

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$IDF\_mOEE

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $IDF\_Installation

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$IDF\_ProductionInstallation

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $IDF\_mOEE

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined\_MES

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| CFG.ConnectionString | string |  | X |  |  |  |
| init | boolean |  |  |  |  |  |
| P.DeletePreviousManualDownTimes | boolean |  |  |  |  |  |
| P.TimeToLogAvailability | integer | X |  |  |  |  |
| P.TimeToLogDownTime | integer | X |  |  |  |  |
| P.TimeToLogProdCount | integer | X |  |  |  |  |
| P.TimeToSetDowntimePulse | integer |  |  |  |  |  |
| S.ActualWorkorder | string |  |  |  |  |  |
| S.CheckRT | boolean |  |  |  |  |  |
| S.DownTime | boolean |  |  |  |  |  |
| S.DownTimePulse | boolean |  |  |  |  |  |
| S.LastDowntimeChangeUTC | string |  |  |  |  |  |
| S.LastDowntimeSetUTC | time |  |  |  |  |  |
| S.LastTimeAvailabilityLogged | time |  |  |  |  |  |
| S.LastTimeQtyChangedUTC | string |  |  |  |  |  |
| S.LastTimeRecalc | time |  |  |  |  |  |
| S.ProductQty | float |  |  |  |  |  |
| S.RunSleeper | integer |  |  |  |  |  |
| S.TimeOfRunTime | time |  |  |  |  |  |
| S.TotalProductQty | float |  |  |  |  |  |
| S.TotalProductQtySaved | float |  |  |  |  |  |
| S.TotalUtilizationQty | float |  |  |  |  |  |
| S.WorkCenter | string |  | X |  |  |  |
| S.WorkOrder | string |  | X |  |  |  |
| S.WorkorderId | integer |  | X |  |  |  |
| Simulate.C | boolean |  |  |  |  |  |
| Simulate.S.Downtime | boolean |  |  |  |  |  |
| Simulate.S.LastDowntimeChangeUTC | string |  |  |  |  |  |
| TRG.LOG | boolean |  |  |  |  |  |
| TRG.LOGAvailability | boolean |  |  |  |  |  |
| TRG.Reset | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

scr\_CheckProduction

|  |  |
| --- | --- |
| Name | scr\_CheckProduction |
| Description |  |
| Trigger | DataChange of ME.S.TotalUtilizationQty |

**Declarations :**

Not Applicable

**Script :**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
' Purpose: Check DownTime  
' set time to now()   
' Version: 0.1  
' Author : SBLE - INDEFF  
' Date : 01/12/2010  
' History: 0.1 01/12/2010 SBLE - INDEFF Preliminary Version  
'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
 'logmessage ("Reset DownTime");  
  
 ME.S.LastTimeQtyChangedUTC = Now().ToUniversalTime().ToString("yyyy-MM-dd HH:mm:ss");

scr\_CheckTimerToLogProdCnt

|  |  |
| --- | --- |
| Name | scr\_CheckTimerToLogProdCnt |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

DIM dtTime AS TIME;

**Script :**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
' Purpose: Check Timer  
' Script: CheckTimer  
' Each x minutes data has to be written to Database  
' x is a paramater (P.RecalculateTime)  
' Er moet enkel gecontroleerd worden wanneer er wel degelijk  
' een workorder is  
' Version: 0.1  
' Author : SBLE - INDEFF  
' Date : 01/12/2010  
' History: 0.1 01/12/2010 SBLE - INDEFF Preliminary Version  
'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
IF False THEN  
  
 IF ME.S.WorkOrder <> "" THEN  
  
   
 dtTime = ME.S.LastTimeRecalc;  
  
 IF ME.\_Debug == 1 THEN  
  
 LogMessage("Now() >= CheckTime: " + Now() + " >= " + dtTime.Addminutes(ME.P.TimeToLogProdCount));  
  
 ENDIF;  
  
 IF Now() >= dtTime.Addminutes(ME.P.TimeToLogProdCount) And ME.TRG.LOG == 0 THEN  
  
 ME.TRG.LOG = 1;  
  
 ENDIF;  
  
 ENDIF;  
  
ENDIF;

scr\_LOG\_ProdCount

|  |  |
| --- | --- |
| Name | scr\_LOG\_ProdCount |
| Description |  |
| Trigger | WhileTrue of ME.TRG.LOG AND FALSE |

**Declarations :**

DIM spProcessor AS INDEFF\_MES.DatabaseAccess.SpProcessor;

**Script :**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
' Purpose: LOG  
' LOG production Qty to database  
' substract previous logged version from total Qty  
' Version: 0.1  
' Author : SBLE - INDEFF  
' Date : 01/12/2010  
' History: 0.1 01/12/2010 SBLE - INDEFF Preliminary Version  
'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
  
 ME.S.ProductQty = (ME.S.TotalProductQty - ME.S.TotalProductQtySaved);  
  
 ME.S.TotalProductQtySaved= ME.S.TotalProductQty;  
  
 IF ME.S.WorkOrder <> "" THEN  
  
 spProcessor = new INDEFF\_MES.DatabaseAccess.SpProcessor( "spFRONT\_IAS\_LogProductionCounter", ME.CFG.ConnectionString);  
 spProcessor.AddInputParameter("@p\_Equipment" ,INDEFF\_MES.DatabaseAccess.Tools.GetStringValue( ME.S.WorkOrder ));  
 spProcessor.AddInputParameter("@p\_WorkOrder" ,INDEFF\_MES.DatabaseAccess.Tools.GetStringValue( ME.S.WorkOrder ));  
 spProcessor.AddInputParameter("@p\_Qty" ,INDEFF\_MES.DatabaseAccess.Tools.GetDoubleValue( ME.S.ProductQty ));  
  
 spProcessor.Execute( );  
  
 IF spProcessor.ErrorOccured THEN  
 LogMessage("DB\_ERROR: " + spProcessor.Error.ErrorMessage);  
 ENDIF;  
  
 spProcessor.Dispose();  
 ENDIF;  
  
 'Reset timer & trigger  
 ME.S.LastTimeRecalc = Now();  
 ME.TRG.LOG = 0;

scr\_CheckTimerToLogAvailability

|  |  |
| --- | --- |
| Name | scr\_CheckTimerToLogAvailability |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

DIM dtTime AS TIME;

**Script :**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
' Purpose: Check Timer  
' Script: CheckTimer  
' Each x minutes data has to be written to Database  
' x is a paramater (P.TimeToLogAvailability)  
  
' Version: 0.1  
' Author : SBLE - INDEFF  
' Date : 15/12/2010  
' History: 0.1 15/12/2010 SBLE - INDEFF Preliminary Version  
'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
   
 dtTime = ME.S.LastTimeAvailabilityLogged;  
  
 IF ME.\_Debug == 1 THEN  
  
 LogMessage("Now() >= CheckTime: " + Now() + " >= " + dtTime.Addminutes(ME.P.TimeToLogAvailability));  
  
 ENDIF;  
  
 IF Now() >= dtTime.Addminutes(ME.P.TimeToLogAvailability) And ME.TRG.LOGAvailability == 0 THEN  
  
 ME.TRG.LOGAvailability = 1;  
  
 ENDIF;

scr\_LOG\_Availability

|  |  |
| --- | --- |
| Name | scr\_LOG\_Availability |
| Description |  |
| Trigger | WhileTrue of ME.TRG.LOGAvailability |

**Declarations :**

DIM spProcessor AS INDEFF\_MES.DatabaseAccess.SpProcessor;

**Script :**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
' Purpose: LOG  
' LOG Availability  
' Version: 0.1  
' Author : SBLE - INDEFF  
' Date : 15/12/2010  
' History: 0.1 15/12/2010 SBLE - INDEFF Preliminary Version  
'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
 'Reset timer & trigger  
 ME.S.LastTimeAvailabilityLogged = Now();  
 ME.TRG.LOGAvailability = 0;  
  
 IF ME.\_Debug == 1 THEN  
  
 LogMessage("LOG Availability");  
  
 ENDIF;  
IF ME.S.WorkorderId <> 0 Then  
 spProcessor = new INDEFF\_MES.DatabaseAccess.SpProcessor( "spFRONT\_IAS\_LogUtilization", ME.CFG.ConnectionString);  
 spProcessor.AddInputParameter("@p\_WorkCenter",INDEFF\_MES.DatabaseAccess.Tools.GetStringValue(ME.S.WorkOrder));  
 spProcessor.AddInputParameter("@p\_Equipment",INDEFF\_MES.DatabaseAccess.Tools.GetStringValue(ME.S.WorkOrder));  
 spProcessor.AddInputParameter("@p\_ProductionRequestId",INDEFF\_MES.DatabaseAccess.Tools.GetIntValue(ME.S.WorkorderId));  
 IF ME.Simulate.C == 0 Then  
 spProcessor.AddInputParameter("@p\_DownTime",INDEFF\_MES.DatabaseAccess.Tools.GetBoolValue(ME.S.DownTime));  
 spProcessor.AddInputParameter("@p\_LastStateChangeUTC",INDEFF\_MES.DatabaseAccess.Tools.GetStringValue(ME.S.LastDowntimeChangeUTC));  
 ELSE  
 spProcessor.AddInputParameter("@p\_DownTime",INDEFF\_MES.DatabaseAccess.Tools.GetBoolValue(ME.Simulate.S.DownTime));   
 spProcessor.AddInputParameter("@p\_LastStateChangeUTC",INDEFF\_MES.DatabaseAccess.Tools.GetStringValue(ME.Simulate.S.LastDowntimeChangeUTC));   
 ENDIF;  
 spProcessor.AddInputParameter("@p\_DeletePreviousManualDownTimes",INDEFF\_MES.DatabaseAccess.Tools.GetBoolValue(ME.P.DeletePreviousManualDownTimes));  
  
 spProcessor.Execute( );  
  
 IF spProcessor.ErrorOccured THEN  
   
 LogMessage("DB\_ERROR: " + spProcessor.Error.ErrorMessage);  
 LogMessage("ME.S.WorkOrder: " + ME.S.WorkOrder);  
 LogMessage("ME.S.WorkorderId: " + ME.S.WorkorderId);  
 LogMessage("ME.S.DownTime: " + ME.S.DownTime);  
 LogMessage("ME.S.LastDowntimeChangeUTC: " + ME.S.LastDowntimeChangeUTC);  
 ELSE  
 ME.P.DeletePreviousManualDownTimes = False;  
 ENDIF;  
  
 spProcessor.Dispose();  
ENDIF;

scr\_init

|  |  |
| --- | --- |
| Name | scr\_init |
| Description |  |
| Trigger | WhileTrue of ME.init |

**Declarations :**

DIM spProcessor AS INDEFF\_MES.DatabaseAccess.SpProcessor;  
DIM dt AS System.Data.DataTable;

**Script :**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
' Purpose: Init  
' Selects the init values from the database  
' Version: 0.1  
' Author : SBLE - INDEFF  
' Date : 01/12/2010  
' History: 0.1 01/12/2010 SBLE - INDEFF Preliminary Version  
'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
 spProcessor = new INDEFF\_MES.DatabaseAccess.SpProcessor( "spFRONT\_IAS\_GetProperties", DB\_INDEFF\_MES.cfg\_ConnectionString);  
 spProcessor.AddInputParameter("@p\_WorkCenter" ,INDEFF\_MES.DatabaseAccess.Tools.GetStringValue( ME.S.WorkCenter ));  
  
 dt = spProcessor.GetList();  
  
 IF spProcessor.ErrorOccured THEN  
 LogMessage("DB\_ERROR: " + spProcessor.Error.ErrorMessage);  
 ELSE  
  
 IF dt.Rows.Count > 0 THEN  
 'copy the results to the uda's  
   
 ME.P.TimeToLogAvailability = INDEFF\_MES.DatabaseAccess.Tools.GetIntValue( dt.Rows[0][0]);  
 ME.P.TimeToLogProdCount = INDEFF\_MES.DatabaseAccess.Tools.GetIntValue( dt.Rows[0][1]);  
 ME.P.TimeToLogDownTime = INDEFF\_MES.DatabaseAccess.Tools.GetIntValue( dt.Rows[0][2]);  
   
 ENDIF;  
  
 ENDIF;  
  
 spProcessor.Dispose();  
  
 ME.init = 0;

scr\_WorkorderChanged

|  |  |
| --- | --- |
| Name | scr\_WorkorderChanged |
| Description |  |
| Trigger | DataChange of ME.S.WorkorderId |

**Declarations :**

DIM spProcessor AS INDEFF\_MES.DatabaseAccess.SpProcessor;

**Script :**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
' Purpose: Logs the last produces of a workorder when when it's stopped  
' Version: 0.1  
' Author : SUYT - INDEFF  
' Date : 2010-02-04  
' History: 0.1 2010-02-04 SUYT - INDEFF   
' Initial version  
'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
 'log availability  
 ME.S.LastTimeAvailabilityLogged = Now();  
  
 spProcessor = new INDEFF\_MES.DatabaseAccess.SpProcessor( "spFRONT\_IAS\_LogUtilization", ME.CFG.ConnectionString);  
 spProcessor.AddInputParameter("@p\_WorkCenter" ,INDEFF\_MES.DatabaseAccess.Tools.GetStringValue( ME.S.WorkOrder ));  
 spProcessor.AddInputParameter("@p\_Equipment" ,INDEFF\_MES.DatabaseAccess.Tools.GetStringValue( ME.S.WorkOrder ));  
 spProcessor.AddInputParameter("@p\_ProductionRequestId" ,INDEFF\_MES.DatabaseAccess.Tools.GetIntValue( ME.S.WorkorderId ));  
 spProcessor.AddInputParameter("@p\_DownTime" ,INDEFF\_MES.DatabaseAccess.Tools.GetBoolValue( ME.S.DownTime ));  
 spProcessor.AddInputParameter("@p\_LastStateChangeUTC" ,INDEFF\_MES.DatabaseAccess.Tools.GetStringValue( ME.S.LastDowntimeChangeUTC ));   
  
 spProcessor.Execute( );  
  
 IF spProcessor.ErrorOccured THEN  
   
 LogMessage("DB\_ERROR: " + spProcessor.Error.ErrorMessage);  
 LogMessage("ME.S.WorkOrder: " + ME.S.WorkOrder);  
 LogMessage("ME.S.WorkorderId: " + ME.S.WorkorderId);  
 LogMessage("ME.S.DownTime: " + ME.S.DownTime);  
 LogMessage("ME.S.LastDowntimeChangeUTC: " + ME.S.LastDowntimeChangeUTC);  
  
 ENDIF;  
  
 spProcessor.Dispose();  
  
 IF NOT System.String.IsNullOrEmpty(ME.S.Workorder ) THEN  
  
 ME.S.ActualWorkorder = ME.S.Workorder;  
 LogMessage("Running order: " + ME.S.ActualWorkorder );  
  
 ELSE  
  
 IF NOT System.String.IsNullOrEmpty(ME.S.ActualWorkorder ) THEN  
  
 'log last produces  
 ME.S.ProductQty = (ME.S.TotalProductQty - ME.S.TotalProductQtySaved);  
  
 ME.S.TotalProductQtySaved= ME.S.TotalProductQty;  
  
 LogMessage("Logging last produce: " + ME.S.ProductQty );  
  
 'LOG ProductQty to DB when it isn't zero  
 IF ME.S.ProductQty > 0 THEN  
  
 spProcessor = new INDEFF\_MES.DatabaseAccess.SpProcessor( "spFRONT\_IAS\_LogProductionCounter", ME.CFG.ConnectionString);  
 spProcessor.AddInputParameter("@p\_Equipment" ,INDEFF\_MES.DatabaseAccess.Tools.GetStringValue( MyContainer.TagName ));  
 spProcessor.AddInputParameter("@p\_WorkOrder" ,INDEFF\_MES.DatabaseAccess.Tools.GetStringValue( ME.S.WorkOrder ));  
 spProcessor.AddInputParameter("@p\_Qty" ,INDEFF\_MES.DatabaseAccess.Tools.GetDoubleValue( ME.S.ProductQty ));  
  
 spProcessor.Execute( );  
  
 IF spProcessor.ErrorOccured THEN  
 LogMessage("DB\_ERROR: " + spProcessor.Error.ErrorMessage);  
 ENDIF;  
  
 spProcessor.Dispose();  
 ENDIF;  
  
 ME.TRG.Reset = TRUE;  
  
 ENDIF;  
  
  
 ENDIF;

scr\_Reset

|  |  |
| --- | --- |
| Name | scr\_Reset |
| Description |  |
| Trigger | WhileTrue of ME.TRG.Reset |

**Declarations :**

Not Applicable

**Script :**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
' Purpose: Resets the object  
' Version: 0.1  
' Author : SUYT - INDEFF  
' Date : 2010-02-04  
' History: 0.1 2010-02-04 SUYT - INDEFF   
' Initial version  
'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
 ME.TRG.Reset = FALSE;  
  
 ME.S.ProductQty = 0;   
 ME.S.TotalProductQty = 0;  
 ME.S.TotalProductQtySaved = 0;  
 ME.S.RunSleeper = 0;  
  
 ME.S.ActualWorkorder = "";

scr\_CheckDownTime

|  |  |
| --- | --- |
| Name | scr\_CheckDownTime |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

DIM lastProduced AS System.DateTime;  
DIM now AS System.DateTime;  
DIM lastDowntime AS System.DateTime;

**Script :**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
' Purpose: Checks the status of the machine: runtime/downtime  
' Version: 0.1  
' Author : SUYT - INDEFF  
' Date : 2010-02-04  
' History: 0.1 2010-02-04 SUYT - INDEFF   
' Initial version  
'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
 lastProduced = ME.S.LastTimeQtyChangedUTC;  
 IF ME.\_Debug == 1 THEN  
 LogMessage(System.DateTime.Now.ToUniversalTime());  
 LogMessage(lastProduced);  
 LogMEssage(lastProduced.Addminutes(ME.P.TimeToLogDownTime));  
 ENDIF;  
  
 IF System.DateTime.Now.ToUniversalTime()>= lastProduced.Addminutes(ME.P.TimeToLogDownTime) THEN  
  
 IF ME.S.DownTime == 0 THEN  
 ME.S.DownTime = 1;   
   
 ENDIF;  
 ME.S.RunSleeper = 60000 \* ME.P.TimeToLogDownTime;  
 ELSE  
 ME.S.RunSleeper = ME.S.RunSleeper - 1000;  
  
 IF ME.S.RunSleeper <= 0 AND ME.S.DownTime == 1 THEN  
   
 ME.S.DownTime = 0;  
  
 ENDIF;  
   
 ENDIF;

scr\_DowntimeChange

|  |  |
| --- | --- |
| Name | scr\_DowntimeChange |
| Description |  |
| Trigger | DataChange of ME.S.DownTime |

**Declarations :**

Not Applicable

**Script :**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
' Purpose: Executed when the downtimen state is changing  
' Version: 0.1  
' Author : SUYT - INDEFF  
' Date : 2010-02-04  
' History: 0.1 2010-02-04 SUYT - INDEFF   
' Initial version  
'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
 ME.S.LastDowntimeChangeUTC = Now().ToUniversalTime().ToString("yyyy-MM-dd HH:mm:ss");  
 ME.TRG.LOGAvailability = TRUE; 'log status change emmidiataly

scr\_Simulate\_Downtime

|  |  |
| --- | --- |
| Name | scr\_Simulate\_Downtime |
| Description |  |
| Trigger | DataChange of ME.Simulate.S.Downtime |

**Declarations :**

Not Applicable

**Script :**

ME.Simulate.S.LastDowntimeChangeUTC = Now().ToUniversalTime().ToString("yyyy-MM-dd HH:mm:ss");  
IF ME.Simulate.C == 1 THEN  
 ME.TRG.LOGAvailability = TRUE; 'log status change immediately  
ENDIF;

#### Template ArchestrA $IDF\_mWorkCenter

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined\_MES

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| S.WorkCenter | string | X |  |  |  |  |
| S.WorkOrder | string |  |  |  |  |  |
| S.WorkOrderId | integer |  |  |  |  |  |
| TRG.GetWorkOrder | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

scr\_GetWorkOrder

|  |  |
| --- | --- |
| Name | scr\_GetWorkOrder |
| Description |  |
| Trigger | WhileTrue of ME.TRG.GetWorkOrder |

**Declarations :**

DIM spProcessor AS Indeff\_MES.DatabaseAccess.SpProcessor;  
Dim dt AS System.Data.DataTable;

**Script :**

DIM WorkOrder as String;  
  
WorkOrder = StringLeft(ME.Tagname, StringInString( ME.Tagname, "\_", 1, 0 ) - 1 );  
  
 ME.TRG.GetWorkOrder = 0;  
  
 spProcessor = new INDEFF\_MES.DatabaseAccess.SpProcessor( "spFRONT\_IAS\_WorkOrder\_Get", DB\_INDEFF\_MES.cfg\_ConnectionString);  
  
 spProcessor.TransactionIsolationLevel = System.Data.IsolationLevel.ReadUncommitted;  
  
   
 spProcessor.AddInputParameter("@p\_WorkCenter" ,INDEFF\_MES.DatabaseAccess.Tools.GetStringValue( WorkOrder ));  
   
 dt = spProcessor.GetList();  
  
 IF spProcessor.ErrorOccured THEN  
 LogMessage("DB\_ERROR: " + spProcessor.Error.ErrorMessage);  
 ELSE  
   
 IF dt.Rows.Count > 0 THEN  
   
 'copy the results to the uda's  
 ME.S.WorkOrder = INDEFF\_MES.DatabaseAccess.Tools.GetStringValue( dt.Rows[0][0] );  
 ME.S.WorkOrderId = INDEFF\_MES.DatabaseAccess.Tools.GetIntValue( dt.Rows[0][1] );  
  
 ELSE  
  
 ME.S.WorkOrder = "";  
 ME.S.WorkOrderId = 0;  
  
 ENDIF;  
  
 ENDIF;  
  
 spProcessor.Dispose();

scr\_Refresh

|  |  |
| --- | --- |
| Name | scr\_Refresh |
| Description |  |
| Trigger | DataChange of Refreshes.WorkOrders |

**Declarations :**

Not Applicable

**Script :**

ME.TRG.GetWorkOrder = 1;

scr\_Init

|  |  |
| --- | --- |
| Name | scr\_Init |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $IDF\_ProductionInstallation

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$IDF\_mWorkCenter

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $IDF\_ResLabFromSAP

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$IDF\_SAP

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| \_CFG.DB.ConnectionString | string |  |  |  |  |  |
| \_FL.Busy | boolean |  |  |  |  |  |
| \_FL.HistorianUpdated | boolean |  |  |  |  |  |
| \_FL.SystemExpertUpdated | boolean |  |  |  |  |  |
| AT.Buffer | string |  |  |  |  |  |
| AT.Buffer.RowsIndex | integer |  |  |  |  |  |
| CMD.UpdateSystemExpert | boolean |  |  |  |  |  |
| DATA.A2\_Processed | boolean |  |  |  |  |  |
| DATA.Datetime | time |  |  |  |  |  |
| DATA.DigitNr | integer |  |  |  |  |  |
| DATA.Id | integer |  |  |  |  |  |
| DATA.TagName | string |  |  |  |  |  |
| DATA.ToUpdate | boolean |  |  |  |  |  |
| DATA.UoM | string |  |  |  |  |  |
| DATA.Value | float |  |  |  |  |  |
| ENA.SystemExpertCopy | boolean |  |  |  |  |  |
| SE.FL.Busy | boolean |  |  |  |  |  |
| SE.FMT1.fCO2\_B | float |  |  |  |  |  |
| SE.FMT2.fCO2\_B | float |  |  |  |  |  |
| SE.FMT3.fCO2\_B | float |  |  |  |  |  |
| SE.FMT4.fCO2\_B | float |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

scr\_GetDataToUpdate

|  |  |
| --- | --- |
| Name | scr\_GetDataToUpdate |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
' Purpose : Gets data from SAP in order to update Historian and System Expert (in some case)  
' Author : PBEA - INDEFF  
' Version : 0.1  
' Status : DRAFT  
' Date : 2014-11-27  
' History : 2014-11-27 PBEA - INDEFF Initial Version  
'   
'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
  
  
DIM spProcessor AS Indeff\_MES.DatabaseAccess.SpProcessor;  
DIM dt AS System.Data.DataTable;  
  
spProcessor = new Indeff\_MES.DatabaseAccess.SpProcessor("spFRONT\_A2\_GetData", me.\_CFG.DB.ConnectionString);  
  
IF spProcessor.ErrorOccured THEN  
 LogMessage(spProcessor.Error.ErrorMessage);  
ELSE  
 dt = spProcessor.GetTable();  
 IF dt.Rows.Count > 0 THEN   
 me.DATA.TagName = Indeff\_MES.DatabaseAccess.Tools.GetStringValue(dt.Rows[0]["TagName"]);  
 me.DATA.Value = Indeff\_MES.DatabaseAccess.Tools.GetDoubleValue(dt.Rows[0]["Value"]);  
 me.DATA.DigitNr = Indeff\_MES.DatabaseAccess.Tools.GetIntValue(dt.Rows[0]["DigitNr"]);  
 me.DATA.UoM = Indeff\_MES.DatabaseAccess.Tools.GetStringValue(dt.Rows[0]["UoM"]);  
 me.DATA.Datetime = Indeff\_MES.DatabaseAccess.Tools.GetDateTimeValue(dt.Rows[0]["SamplingDateTime"]);  
 me.DATA.ToUpdate = Indeff\_MES.DatabaseAccess.Tools.GetBoolValue(dt.Rows[0]["FL\_ToUpdate"]);  
 me.DATA.A2\_Processed = Indeff\_MES.DatabaseAccess.Tools.GetBoolValue(dt.Rows[0]["FL\_A2\_Processed"]);  
 me.DATA.Id = Indeff\_MES.DatabaseAccess.Tools.GetIntValue(dt.Rows[0]["Id"]); 'Keep Id at the end of the data copy  
 ELSE  
 me.DATA.A2\_Processed = True;  
 ENDIF;  
 spProcessor.Dispose();  
ENDIF;

scr\_DataManagement

|  |  |
| --- | --- |
| Name | scr\_DataManagement |
| Description |  |
| Trigger | WhileTrue of me.DATA.Id <> Id\_CPY OR me.\_FL.Busy OR NOT me.DATA.A2\_Processed |

**Declarations :**

DIM Id\_CPY AS INTEGER;  
DIM x AS INDIRECT;  
DIM ScriptExecutionCnt\_CPY AS INTEGER;

**Script :**

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
' Purpose : Gets data from SAP in order to update System Expert (in some case)  
' Author : PBEA - INDEFF  
' Version : 0.1  
' Status : DRAFT  
' Date : 2014-11-27  
' History : 2014-11-27 PBEA - INDEFF Initial Version  
'   
'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
DIM spProcessor AS Indeff\_MES.DatabaseAccess.SpProcessor;  
DIM dt AS System.Data.DataTable;  
DIM i AS INTEGER;  
  
IF Id\_CPY <> 0 AND me.\_FL.Busy == False THEN  
 x.BindTo(me.DATA.TagName + ".ValueDeadBand");  
 ScriptExecutionCnt\_CPY = me.scr\_DataManagement.ExecutionCnt;  
 me.\_FL.Busy = True;  
 'Audit Trails - Define AT.Buffer row usable  
 FOR i = 1 TO 100 STEP 1  
 IF StringLen(me.AT.Buffer[i]) > 0 THEN  
 me.AT.Buffer.RowsIndex = i + 1;  
 ENDIF;  
 NEXT;  
 IF me.AT.Buffer.RowsIndex >= 100 OR me.AT.Buffer.RowsIndex == 0 THEN  
 me.AT.Buffer.RowsIndex = 1;  
 ENDIF;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.DATA.TagName + "|Id: " + me.DATA.Id + "|Historization check";   
ENDIF;  
  
IF me.\_FL.Busy AND me.scr\_DataManagement.ExecutionCnt >= ScriptExecutionCnt\_CPY + 1 THEN 'Jump 1 engine cycle to allow multi Engine/platform binding  
 IF isgood(x) THEN 'Data exist for historization  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Historization possible";  
  
 spProcessor = new Indeff\_MES.DatabaseAccess.SpProcessor("spFRONT\_A2\_DataProcessed", me.\_CFG.DB.ConnectionString);  
 spProcessor.AddInputParameter("@p\_Id",Indeff\_MES.DatabaseAccess.Tools.GetIntValue(me.DATA.Id));  
   
 dt = spProcessor.GetList();  
  
 IF spProcessor.ErrorOccured THEN  
 LogMessage(spProcessor.Error.ErrorMessage);  
 me.\_FL.HistorianUpdated = False;  
 ELSE   
 me.\_FL.HistorianUpdated = True;  
 ENDIF;  
 IF me.DATA.ToUpdate THEN  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Value to update";  
 ELSE  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Value not to update";  
 ENDIF;  
  
 me.\_FL.Busy = False;  
 IF me.ENA.SystemExpertCopy THEN  
 me.CMD.UpdateSystemExpert = True;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Copy Attempt to SE";  
 ENDIF;  
 ELSE  
 me.\_FL.Busy = False;  
 me.CMD.UpdateSystemExpert = False;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Historization not possible";  
 ENDIF;  
  
ENDIF;  
  
Id\_CPY = me.DATA.Id;

scr\_DataToSystemExpert

|  |  |
| --- | --- |
| Name | scr\_DataToSystemExpert |
| Description |  |
| Trigger | WhileTrue of me.CMD.UpdateSystemExpert OR me.SE.FL.Busy |

**Declarations :**

DIM x AS INDIRECT;  
DIM fCO2\_B\_1 AS INDIRECT;  
DIM fCO2\_B\_2 AS INDIRECT;  
DIM fCO2\_B\_3 AS INDIRECT;  
DIM fCO2\_B\_4 AS INDIRECT;  
DIM fCO2\_B\_5 AS INDIRECT;  
DIM fCO2\_B\_6 AS INDIRECT;  
DIM ScriptExecutionCnt\_CPY AS INTEGER;

**Script :**

DIM dtNow AS System.DateTime;  
DIM dtValue AS System.DateTime;  
DIM FM\_Nr AS STRING;  
DIM FM\_Material AS STRING;  
DIM CPY\_Index AS INTEGER;  
  
DIM DateTimeMin AS System.DateTime;  
DIM DateTimeMax AS System.DateTime;  
  
dtNow = System.DateTime.Now();  
dtValue = me.DATA.DateTime;  
   
FM\_Nr = StringRight(StringLeft(me.DATA.TagName, 4), 1);  
  
IF StringLeft(me.DATA.TagName, 3) == "FMT" AND StringRight(me.DATA.TagName, 3) == "CO2" THEN  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Copy to SE possible";  
   
 'We have to check if the data sampling time is corresponding to the actual time  
 IF dtNow.Year == dtValue.Year AND dtNow.Month == dtValue.Month AND dtNow.Day == dtValue.Day THEN 'Value from today   
 fCO2\_B\_1.BindTo("SE\_ResLab\_FOUR" + FM\_Nr + ".fCO2\_B[1]" );  
 fCO2\_B\_2.BindTo("SE\_ResLab\_FOUR" + FM\_Nr + ".fCO2\_B[2]" );  
 fCO2\_B\_3.BindTo("SE\_ResLab\_FOUR" + FM\_Nr + ".fCO2\_B[3]" );  
 fCO2\_B\_4.BindTo("SE\_ResLab\_FOUR" + FM\_Nr + ".fCO2\_B[4]" );  
 fCO2\_B\_5.BindTo("SE\_ResLab\_FOUR" + FM\_Nr + ".fCO2\_B[5]" );  
 fCO2\_B\_6.BindTo("SE\_ResLab\_FOUR" + FM\_Nr + ".fCO2\_B[6]" );  
 ScriptExecutionCnt\_CPY = me.scr\_DataToSystemExpert.ExecutionCnt;  
 me.SE.FL.Busy = True;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Sampling time corresponding to actual day";   
 ELSE  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Sampling time not corresponding to actual day";   
 ENDIF;  
 IF me.SE.FL.Busy AND me.scr\_DataToSystemExpert.ExecutionCnt >= ScriptExecutionCnt\_CPY + 1 THEN 'Jump 1 engine cycle to allow multi Engine/platform binding  
 IF isgood(fCO2\_B\_1) AND isgood(fCO2\_B\_2) AND isgood(fCO2\_B\_3) AND isgood(fCO2\_B\_4) AND isgood(fCO2\_B\_5) AND isgood(fCO2\_B\_6) THEN 'UDA exists  
 IF fCO2\_B\_1 == -1.0 THEN  
 fCO2\_B\_1 = me.DATA.Value;  
 me.\_FL.SystemExpertUpdated = True;   
 ELSE  
 IF fCO2\_B\_2 == -1.0 THEN  
 fCO2\_B\_2 = me.DATA.Value;  
 me.\_FL.SystemExpertUpdated = True;  
 ELSE  
 IF fCO2\_B\_3 == -1.0 THEN  
 fCO2\_B\_3 = me.DATA.Value;  
 me.\_FL.SystemExpertUpdated = True;  
 ELSE  
 IF fCO2\_B\_4 == -1.0 THEN  
 fCO2\_B\_4 = me.DATA.Value;  
 me.\_FL.SystemExpertUpdated = True;  
 ELSE  
 IF fCO2\_B\_5 == -1.0 THEN  
 fCO2\_B\_5 = me.DATA.Value;  
 me.\_FL.SystemExpertUpdated = True;  
 ELSE  
 IF fCO2\_B\_6 == -1.0 THEN  
 fCO2\_B\_6 = me.DATA.Value;  
 me.\_FL.SystemExpertUpdated = True;  
 ELSE  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|SE full of values. Copy not possible automatically";   
 ENDIF;  
 ENDIF;  
 ENDIF;   
 ENDIF;  
 ENDIF;  
 ENDIF;  
 ELSE  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|SE objects not reachable";   
 ENDIF;   
 me.SE.FL.Busy = False;   
 ENDIF;  
   
ELSE  
 me.\_FL.SystemExpertUpdated = False;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Tag not corresponding to FMT SE CO2";  
ENDIF;  
  
IF StringLeft(me.DATA.TagName, 2) == "FR" AND (StringRight(me.DATA.TagName, 3) == "CO2" OR StringRight(me.DATA.TagName, 3) == "T70") THEN  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Copy to SE possible";  
 'We have to check if the data sampling time is corresponding to the actual time  
 IF dtNow.Year == dtValue.Year AND dtNow.Month == dtValue.Month AND dtNow.Day == dtValue.Day AND dtValue.Hour >= dtNow.Hour - 1 AND dtValue.Hour <= dtNow.Hour + 1 THEN 'Value from today   
 IF StringLeft(me.DATA.TagName, 3) == "FR1" THEN  
 IF StringRight(me.DATA.TagName, 3) == "T70" THEN  
 IF isgood(SE\_FR\_FOUR1.fRMAGFR) AND isgood(SE\_FR\_FOUR1.fRMAGFR\_OLD) THEN  
 SE\_FR\_FOUR1.fRMAGFR\_OLD = SE\_FR\_FOUR1.fRMAGFR;  
 SE\_FR\_FOUR1.fRMAGFR = me.DATA.Value;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Copy to SE SE\_FR\_FOUR1.fRMAGFR";  
 ENDIF;  
 ENDIF;  
 IF me.DATA.TagName == "FR1\_CHAUX\_10\_50.CO2" THEN  
 IF isgood(SE\_FR\_FOUR1.fCO21040FR) AND isgood(SE\_FR\_FOUR1.fCO21040FR\_OLD) THEN  
 SE\_FR\_FOUR1.fCO21040FR\_OLD = SE\_FR\_FOUR1.fCO21040FR;  
 SE\_FR\_FOUR1.fCO21040FR = me.DATA.Value;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Copy to SE SE\_FR\_FOUR1.fCO21040FR";  
 ENDIF;  
 ENDIF;  
 IF me.DATA.TagName == "FR1\_CHAUX\_2\_10.CO2\_2" THEN  
 IF isgood(SE\_FR\_FOUR1.fCO2210FR) AND isgood(SE\_FR\_FOUR1.fCO2210FR\_OLD) THEN  
 SE\_FR\_FOUR1.fCO2210FR\_OLD = SE\_FR\_FOUR1.fCO2210FR;  
 SE\_FR\_FOUR1.fCO2210FR = me.DATA.Value;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Copy to SE SE\_FR\_FOUR1.fCO2210FR";  
 ENDIF;  
 ENDIF;  
 ENDIF;  
 IF StringLeft(me.DATA.TagName, 3) == "FR2" THEN  
 IF StringRight(me.DATA.TagName, 3) == "T70" THEN  
 IF isgood(SE\_FR\_FOUR2.fRMAGFR) AND isgood(SE\_FR\_FOUR2.fRMAGFR\_OLD) THEN  
 SE\_FR\_FOUR2.fRMAGFR\_OLD = SE\_FR\_FOUR2.fRMAGFR;  
 SE\_FR\_FOUR2.fRMAGFR = me.DATA.Value;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Copy to SE SE\_FR\_FOUR2.fRMAGFR";  
 ENDIF;  
 ENDIF;  
 IF me.DATA.TagName == "FR2\_CHAUX\_10\_50.CO2" THEN  
 IF isgood(SE\_FR\_FOUR2.fCO21040FR) AND isgood(SE\_FR\_FOUR2.fCO21040FR\_OLD) THEN  
 SE\_FR\_FOUR2.fCO21040FR\_OLD = SE\_FR\_FOUR2.fCO21040FR;  
 SE\_FR\_FOUR2.fCO21040FR = me.DATA.Value;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Copy to SE SE\_FR\_FOUR2.fCO21040FR";  
 ENDIF;  
 ENDIF;  
 IF me.DATA.TagName == "FR2\_CHAUX\_2\_10.CO2\_2" THEN  
 IF isgood(SE\_FR\_FOUR2.fCO2210FR) AND isgood(SE\_FR\_FOUR2.fCO2210FR\_OLD) THEN  
 SE\_FR\_FOUR2.fCO2210FR\_OLD = SE\_FR\_FOUR2.fCO2210FR;  
 SE\_FR\_FOUR2.fCO2210FR = me.DATA.Value;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Copy to SE SE\_FR\_FOUR2.fCO2210FR";  
 ENDIF;  
 ENDIF;  
 ENDIF;  
 IF StringLeft(me.DATA.TagName, 3) == "FR3" THEN  
 IF StringRight(me.DATA.TagName, 3) == "T70" THEN  
 IF isgood(SE\_FR\_FOUR3.fRMAGFR) AND isgood(SE\_FR\_FOUR3.fRMAGFR\_OLD) THEN  
 SE\_FR\_FOUR3.fRMAGFR\_OLD = SE\_FR\_FOUR3.fRMAGFR;  
 SE\_FR\_FOUR3.fRMAGFR = me.DATA.Value;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Copy to SE SE\_FR\_FOUR3.fRMAGFR";  
 ENDIF;  
 ENDIF;  
 IF me.DATA.TagName == "FR3\_CHAUX\_10\_50.CO2" THEN  
 IF isgood(SE\_FR\_FOUR3.fCO21040FR) AND isgood(SE\_FR\_FOUR3.fCO21040FR\_OLD) THEN  
 SE\_FR\_FOUR3.fCO21040FR\_OLD = SE\_FR\_FOUR3.fCO21040FR;  
 SE\_FR\_FOUR3.fCO21040FR = me.DATA.Value;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Copy to SE SE\_FR\_FOUR3.fCO21040FR";  
 ENDIF;  
 ENDIF;  
 IF me.DATA.TagName == "FR3\_CHAUX\_2\_10.CO2\_2" THEN  
 IF isgood(SE\_FR\_FOUR3.fCO2210FR) AND isgood(SE\_FR\_FOUR3.fCO2210FR\_OLD) THEN  
 SE\_FR\_FOUR3.fCO2210FR\_OLD = SE\_FR\_FOUR3.fCO2210FR;  
 SE\_FR\_FOUR3.fCO2210FR = me.DATA.Value;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Copy to SE SE\_FR\_FOUR3.fCO2210FR";  
 ENDIF;  
 ENDIF;  
 ENDIF;  
 IF StringLeft(me.DATA.TagName, 3) == "FR4" THEN  
 IF StringRight(me.DATA.TagName, 3) == "T70" THEN  
 IF isgood(SE\_FR\_FOUR4.fRMAGFR) AND isgood(SE\_FR\_FOUR4.fRMAGFR\_OLD) THEN  
 SE\_FR\_FOUR4.fRMAGFR\_OLD = SE\_FR\_FOUR4.fRMAGFR;  
 SE\_FR\_FOUR4.fRMAGFR = me.DATA.Value;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Copy to SE SE\_FR\_FOUR4.fRMAGFR";  
 ENDIF;  
 ENDIF;  
 IF me.DATA.TagName == "FR4\_CHAUX\_10\_50.CO2" THEN  
 IF isgood(SE\_FR\_FOUR4.fCO21040FR) AND isgood(SE\_FR\_FOUR4.fCO21040FR\_OLD) THEN  
 SE\_FR\_FOUR4.fCO21040FR\_OLD = SE\_FR\_FOUR4.fCO21040FR;  
 SE\_FR\_FOUR4.fCO21040FR = me.DATA.Value;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Copy to SE SE\_FR\_FOUR4.fCO21040FR";  
 ENDIF;  
 ENDIF;  
 IF me.DATA.TagName == "FR4\_CHAUX\_2\_10.CO2\_2" THEN  
 IF isgood(SE\_FR\_FOUR4.fCO2210FR) AND isgood(SE\_FR\_FOUR4.fCO2210FR\_OLD) THEN  
 SE\_FR\_FOUR4.fCO2210FR\_OLD = SE\_FR\_FOUR4.fCO2210FR;  
 SE\_FR\_FOUR4.fCO2210FR = me.DATA.Value;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Copy to SE SE\_FR\_FOUR4.fCO2210FR";  
 ENDIF;  
 ENDIF;  
 ENDIF;  
 ENDIF;  
ELSE  
 me.\_FL.SystemExpertUpdated = False;  
 'Audit Trails information  
 me.AT.Buffer[me.AT.Buffer.RowsIndex] = me.AT.Buffer[me.AT.Buffer.RowsIndex] + "|Tag not corresponding to FR SE T70 CO2";  
ENDIF;  
me.CMD.UpdateSystemExpert = False;

#### Template ArchestrA $IDF\_SAP

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $Integer

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$FieldReference

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $LocalVerin

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

Not Applicable

##### Field Attributes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Attribute name | Data Type | IO Scaling | History | Limit Alarms | Rate Of Change Alarms | Target Deviation Alarms | Bad Value Alarm | Statistics |
| ANIM | boolean |  |  |  |  |  |  |  |

##### Scripts

Not Applicable

#### Template ArchestrA $Mélangeur

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$Moteur1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $mINDEFF\_UserDefined\_V\_1\_0

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| \_Initialized | boolean |  |  |  |  |  |
| \_Initializing | boolean |  |  |  |  |  |
| \_InitLoopCount | integer |  |  |  |  |  |
| \_InitScanCycles | integer |  |  |  |  |  |
| \_OverrideConnectionName | string |  |  |  |  |  |
| ConnectionName | string |  |  |  |  |  |
| InitializeObjectRq | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

INDEFF\_TriggerInitObject

|  |  |
| --- | --- |
| Name | INDEFF\_TriggerInitObject |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

' Description   
' This script Controls the activation of initializing  
  
' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
' History   
' 05/12/2006 Creation  
' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
 'Wait with actual execution of logic until a few scancycles have passed.  
' IF me.INDEFF\_TriggerInitObject.ExecutionCnt > me.\_InitScanCycles THEN  
  
 'If there's an initialize request, reset the InitLoopCount  
' IF Me.InitializeObjectRq == True OR  
' MyArea.ReinitializeObjects == True OR  
' Me.\_InitLoopCount == 0  
' THEN  
' Me.\_InitLoopCount = 0;  
' Me.\_Initializing = True;  
' Me.\_Initialized = False;  
' Me.InitializeObjectRq = False;  
' ENDIF;  
  
 'If the loopcount exceeds the initscancycles, the object is initialized  
' IF Me.\_InitLoopCount > Me.\_InitScanCycles THEN  
' Me.\_Initializing = False;  
' Me.\_Initialized = True;  
' ENDIF;  
  
 'Add a Loop Count  
' Me.\_InitLoopCount = Me.\_InitLoopCount + 1;  
' ENDIF;

#### Template ArchestrA $Moteur1S

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| BTN\_TexteStatus | string |  |  |  |  |  |
| IsVariFreq | boolean |  |  |  |  |  |
| Nom | string |  |  |  |  |  |
| NomLogix | string |  |  |  |  |  |

##### Field Attributes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Attribute name | Data Type | IO Scaling | History | Limit Alarms | Rate Of Change Alarms | Target Deviation Alarms | Bad Value Alarm | Statistics |
| STATUTS\_MODE\_AUTO | boolean |  |  |  |  |  |  |  |
| CHECK\_DETECTEUR\_ROTATION | boolean |  |  |  |  |  |  |  |
| CHECK\_DETECTEUR\_TEMPERATURE | boolean |  |  |  |  |  |  |  |
| CHECK\_DETECTEUR\_BOURRAGE | boolean |  |  |  |  |  |  |  |
| CHECK\_DETECTEUR\_AU | boolean |  |  |  |  |  |  |  |
| CHECK\_DETECTEUR\_DEPORT\_BANDE | boolean |  |  |  |  |  |  |  |
| ALARM\_RKM\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_MTH\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_IS\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_ROT\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_LOCAL\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_TEMP\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_AU\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_SB\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_DDB\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_DDB\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_RKM\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_MTH\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_IS\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_ROT\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_LOCAL\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_TEMP\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_AU\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_SB\_A2 | boolean |  |  |  |  |  |  |  |
| STATUT\_DEFAUT\_GENERAL | boolean |  |  |  |  |  |  |  |
| STATUT\_MODE\_LOCAL | boolean |  |  |  |  |  |  |  |
| ALARM\_DRIVE\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_DRIVE\_A2 | boolean |  |  |  |  |  |  |  |
| Marche | boolean |  |  |  |  |  |  |  |
| Arret | boolean |  |  |  |  |  |  |  |
| ANIM | integer |  |  |  |  |  |  |  |
| TEMPS\_DEFAUT\_DDB | integer |  |  |  |  |  |  |  |
| TEMPS\_DEFAUT\_SB | integer |  |  |  |  |  |  |  |
| TEMPS\_DEFAUT\_RI | integer |  |  |  |  |  |  |  |
| TEMPS\_DEFAUT\_ROT | integer |  |  |  |  |  |  |  |

##### Scripts

AssigneTexteBTN

|  |  |
| --- | --- |
| Name | AssigneTexteBTN |
| Description |  |
| Trigger | DataChange of Me.Anim |

**Declarations :**

Not Applicable

**Script :**

IF Me.ANIM == 0 OR Me.ANIM == 1 then   
Me.BTN\_TexteStatus = "Arrêt";  
endif;  
  
IF Me.ANIM == 2 OR Me.ANIM == 3 then   
Me.BTN\_TexteStatus = "Marche";  
endif;  
  
IF Me.ANIM > 4 then   
Me.BTN\_TexteStatus = "Transition";  
endif;

AssignNom

|  |  |
| --- | --- |
| Name | AssignNom |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $Moteur2S

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| BTN\_TexteStatus | string |  |  |  |  |  |
| IsVariFreq | boolean |  |  |  |  |  |
| Nom | string |  |  |  |  |  |
| NomLogix | string |  |  |  |  |  |
| sTexteDroite | string |  |  |  |  |  |
| sTexteGauche | string |  |  |  |  |  |

##### Field Attributes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Attribute name | Data Type | IO Scaling | History | Limit Alarms | Rate Of Change Alarms | Target Deviation Alarms | Bad Value Alarm | Statistics |
| ALARM\_AU\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_AU\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_DDB\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_DDB\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_DRIVE\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_DRIVE\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_IS\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_IS\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_LOCAL\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_LOCAL\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_MTH\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_MTH\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_RKM\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_RKM\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_ROT\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_ROT\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_SB\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_SB\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_TEMP\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_TEMP\_A2 | boolean |  |  |  |  |  |  |  |
| Arret | boolean |  |  |  |  |  |  |  |
| CHECK\_DETECTEUR\_AU | boolean |  |  |  |  |  |  |  |
| CHECK\_DETECTEUR\_BOURRAGE | boolean |  |  |  |  |  |  |  |
| CHECK\_DETECTEUR\_DEPORT\_BANDE | boolean |  |  |  |  |  |  |  |
| CHECK\_DETECTEUR\_ROTATION | boolean |  |  |  |  |  |  |  |
| CHECK\_DETECTEUR\_TEMPERATURE | boolean |  |  |  |  |  |  |  |
| STATUT\_DEFAUT\_GENERAL | boolean |  |  |  |  |  |  |  |
| STATUT\_MODE\_LOCAL | boolean |  |  |  |  |  |  |  |
| STATUTS\_MODE\_AUTO | boolean |  |  |  |  |  |  |  |
| Marche\_AV | boolean |  |  |  |  |  |  |  |
| Marche\_AR | boolean |  |  |  |  |  |  |  |
| STATUTS\_MARCHE\_AR | boolean |  |  |  |  |  |  |  |
| STATUTS\_MARCHE\_AV | boolean |  |  |  |  |  |  |  |
| ANIM | integer |  |  |  |  |  |  |  |
| TEMPS\_DEFAUT\_DDB | integer |  |  |  |  |  |  |  |
| TEMPS\_DEFAUT\_RI | integer |  |  |  |  |  |  |  |
| TEMPS\_DEFAUT\_ROT | integer |  |  |  |  |  |  |  |
| TEMPS\_DEFAUT\_SB | integer |  |  |  |  |  |  |  |

##### Scripts

AssignTexte

|  |  |
| --- | --- |
| Name | AssignTexte |
| Description |  |
| Trigger | DataChange of Me.Anim |

**Declarations :**

Not Applicable

**Script :**

IF Me.ANIM == 1 OR Me.ANIM == 8 then   
Me.BTN\_TexteStatus = "Arrêt";  
endif;  
  
IF Me.ANIM == 2 OR Me.ANIM == 16 then   
Me.BTN\_TexteStatus = "Marche";  
endif;  
  
IF Me.ANIM == 4 then   
Me.BTN\_TexteStatus = "Transition";  
endif;

AssignNom

|  |  |
| --- | --- |
| Name | AssignNom |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $MyPID

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| Nom | string |  |  |  |  |  |
| NomLogix | string |  |  |  |  |  |

##### Field Attributes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Attribute name | Data Type | IO Scaling | History | Limit Alarms | Rate Of Change Alarms | Target Deviation Alarms | Bad Value Alarm | Statistics |
| BP\_EXPERT | boolean |  |  |  |  |  |  |  |
| BP\_AUTO | boolean |  |  |  |  |  |  |  |
| BP\_MANU | boolean |  |  |  |  |  |  |  |
| Alarme\_SHH | boolean |  |  |  |  |  |  |  |
| Alarme\_SH | boolean |  |  |  |  |  |  |  |
| Alarme\_SB | boolean |  |  |  |  |  |  |  |
| Alarme\_SBB | boolean |  |  |  |  |  |  |  |
| SP\_EXPERT | float |  |  |  |  |  |  |  |
| SP\_AUTO | float |  |  |  |  |  |  |  |
| PV | float |  |  |  |  |  |  |  |
| PV\_MAX | float |  |  |  |  |  |  |  |
| PV\_MIN | float |  |  |  |  |  |  |  |
| OUT | float |  |  |  |  |  |  |  |
| OUT\_MAX | float |  |  |  |  |  |  |  |
| OUT\_MIN | float |  |  |  |  |  |  |  |
| CST\_P | float |  |  |  |  |  |  |  |
| CST\_I | float |  |  |  |  |  |  |  |
| CST\_D | float |  |  |  |  |  |  |  |
| DEV\_SHH | float |  |  |  |  |  |  |  |
| DEV\_SH | float |  |  |  |  |  |  |  |
| DEV\_SB | float |  |  |  |  |  |  |  |
| DEV\_SBB | float |  |  |  |  |  |  |  |
| MODE | integer |  |  |  |  |  |  |  |
| DEV\_HYS | float |  |  |  |  |  |  |  |

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

AssignNom

|  |  |
| --- | --- |
| Name | AssignNom |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $NAVI\_BEWHCLI

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| CurrentIsView | boolean |  |  |  |  |  |
| CurrentView | string |  |  |  |  |  |
| CurrentViewExpand | string |  |  |  |  |  |
| CurrentViewID | integer |  |  |  |  |  |
| CurrentViewPere | string |  |  |  |  |  |
| CurrentViewPereID | integer |  |  |  |  |  |
| CurrentViewTxt | string |  |  |  |  |  |
| FileName | string |  |  |  |  |  |
| Fils\_IsView | boolean |  |  |  |  |  |
| Fils\_View | string |  |  |  |  |  |
| FilsCount | integer |  |  |  |  |  |
| Frere\_IsView | boolean |  |  |  |  |  |
| Frere\_View | string |  |  |  |  |  |
| FreresCount | integer |  |  |  |  |  |
| MenuRefresh | boolean |  |  |  |  |  |
| MenuRefreshExpand | boolean |  |  |  |  |  |
| NavigationArea | string |  |  |  |  |  |
| PetitFils\_View | string |  |  |  |  |  |
| PetitFilsCount | integer |  |  |  |  |  |
| QuerySelectedView | boolean |  |  |  |  |  |
| Reset | boolean |  |  |  |  |  |
| SelectedIsView | boolean |  |  |  |  |  |
| SelectedRecipe | string |  |  |  |  |  |
| SelectedViewFileTxt | string |  |  |  |  |  |
| SelectedViewTxt | string |  |  |  |  |  |
| Startup | boolean |  |  |  |  |  |
| XMLDSRead | boolean |  |  |  |  |  |
| XMLOK | boolean |  |  |  |  |  |
| XMLPath | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

MenuRefresh

|  |  |
| --- | --- |
| Name | MenuRefresh |
| Description |  |
| Trigger | WhileTrue of Me.menurefresh and Me.xmlok |

**Declarations :**

Dim MyDataSet as System.Data.DataSet;  
Dim DataTable as System.Data.DataTable;  
Dim foundRows[1] as System.Data.DataRow;  
Dim Sort as string;  
Dim i as Integer;  
Dim j as Integer;  
Dim k as Integer;  
Dim RecordCount as integer;

**Script :**

'logmessage("A² Graphic Menu - MenuRefresh");  
  
'Reset Petit-Fils  
For j = 1 to 15  
 Me.PetitFils\_View[j] = "";  
Next;  
  
' Retrieve MyDataSet from AppDomain  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_"+ me.Tagname);  
  
' Add Data to MyDataSet  
DataTable = MyDataSet.Tables("NavMenu");  
  
' Retrieve data from Array  
foundRows[] = DataTable.Select("CurrentViewTxt like '"+Me.NavigationArea+"'","");  
RecordCount = foundRows[].GetUpperBound(0) + 1;  
  
if (RecordCount > 0) then  
 'logmessage("A² Graphic Menu - MenuRefresh - Vue courante : [ "+foundRows[1].ItemArray(2)+" ]");  
 Me.CurrentViewTxt = foundRows[1].ItemArray(1);  
 Me.CurrentView = foundRows[1].ItemArray(2);  
 Me.CurrentIsView = foundRows[1].ItemArray(5);  
 Me.CurrentViewID = foundRows[1].ItemArray(7);  
 Me.CurrentViewPereID = foundRows[1].ItemArray(3);  
' if(CurrentIsView == false) then  
  
 ' Retrieve data from Array  
 foundRows[] = DataTable.Select("CurrentViewID = "+Me.CurrentViewPereID,"");  
 RecordCount = foundRows[].GetUpperBound(0) + 1;  
   
 if (RecordCount > 0) then  
 Me.CurrentViewPere = foundRows[1].ItemArray(2);   
 endif;  
   
 'logmessage("A² Graphic Menu - MenuRefresh - Lancement Query sur table NavMenu");  
 'logmessage("A² Graphic Menu - MenuRefresh - QueryString : [ FatherViewID like "+Me.CurrentViewPereID+" ]");  
   
 ' Retrieve data from Array  
 foundRows[] = DataTable.Select("FatherViewID = "+Me.CurrentViewPereID,"CurrentViewID ASC");  
 RecordCount = foundRows[].GetUpperBound(0) + 1;  
   
 Me.FreresCount = RecordCount;  
   
 'logmessage("A² Graphic Menu - MenuRefresh - Boucle sur les frères");  
 'logmessage("A² Graphic Menu - MenuRefresh - Records : [ "+RecordCount+" ]");  
   
 if (RecordCount > 0) then  
 'logmessage("A² Graphic Menu - MenuRefresh - Loop sur les records récupérés");  
  
  
 'reset les variables  
 For j = 1 to 15  
 Me.Frere\_View[j] = "";  
 Next;  
  
 For j = 1 to RecordCount  
   
 Me.Frere\_View[j] = foundRows[j].ItemArray(2);  
 Me.Frere\_IsView[j] = foundRows[j].ItemArray(5);  
 Next;  
 endif;  
   
 ' Retrieve data from Array  
 foundRows[] = DataTable.Select("FatherViewID = "+Me.CurrentViewID,"CurrentViewID ASC");  
 RecordCount = foundRows[].GetUpperBound(0) + 1;  
   
 Me.FilsCount = RecordCount;  
   
 if (RecordCount > 0) then  
 'logmessage("A² Graphic Menu - MenuRefresh - Loop sur les records récupérés");  
  
 For k = 1 to 15  
 Me.Fils\_View[k] = "";  
 Next;  
  
 For k = 1 to RecordCount  
 Me.Fils\_View[k] = foundRows[k].ItemArray(2);  
 Me.Fils\_IsView[k] = foundRows[k].ItemArray(5);  
 Next;  
 endif;  
' endif;  
endif;  
  
Me.MenuRefresh = false;

MenuRefreshExpand

|  |  |
| --- | --- |
| Name | MenuRefreshExpand |
| Description |  |
| Trigger | WhileTrue of Me.MenuRefreshExpand and Me.XMLOK |

**Declarations :**

Dim MyDataSet as System.Data.DataSet;  
Dim DataTable as System.Data.DataTable;  
Dim foundRows[1] as System.Data.DataRow;  
Dim Sort as string;  
Dim i as Integer;  
Dim j as Integer;  
Dim RecordCount as integer;

**Script :**

'logmessage("A² Graphic Menu - MenuRefreshExpand");  
  
For j = 1 to 15  
 Me.PetitFils\_View[j] = "";  
Next;  
  
' Retrieve MyDataSet from AppDomain  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_"+ me.Tagname);  
  
' Add Data to MyDataSet  
DataTable = MyDataSet.Tables("NavMenu");  
  
' Retrieve data from Array  
foundRows[] = DataTable.Select("CurrentViewTxt like '"+Me.CurrentViewExpand+"'","");  
RecordCount = foundRows[].GetUpperBound(0) + 1;  
  
if (RecordCount > 0) then  
 'logmessage("A² Graphic Menu - MenuRefreshExpand - Vue courante : [ "+foundRows[1].ItemArray(2)+" ]");  
 Me.CurrentViewTxt = foundRows[1].ItemArray(1);  
 Me.CurrentView = foundRows[1].ItemArray(2);  
 Me.CurrentIsView = foundRows[1].ItemArray(5);  
 Me.CurrentViewID = foundRows[1].ItemArray(7);  
 Me.CurrentViewPereID = foundRows[1].ItemArray(3);   
   
 ' Retrieve data from Array  
 foundRows[] = DataTable.Select("FatherViewID = "+Me.CurrentViewID,"CurrentViewID ASC");  
 RecordCount = foundRows[].GetUpperBound(0) + 1;  
   
 Me.PetitFilsCount = RecordCount;  
   
 if (RecordCount > 0) then  
  
 For j = 1 to RecordCount  
 Me.PetitFils\_View[j] = foundRows[j].ItemArray(2);  
 Next;  
   
 endif;  
' endif;  
endif;  
  
Me.MenuRefreshExpand = false;

QuerySelectedView

|  |  |
| --- | --- |
| Name | QuerySelectedView |
| Description |  |
| Trigger | WhileTrue of Me.QuerySelectedView and Me.XMLOK |

**Declarations :**

Dim MyDataSet as System.Data.DataSet;  
Dim DataTable as System.Data.DataTable;  
Dim foundRows[1] as System.Data.DataRow;  
Dim Sort as string;  
Dim RecordCount as integer;

**Script :**

'logmessage("A² Graphic Menu - QuerySelectedView");  
  
' Retrieve MyDataSet from AppDomain  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_"+ me.Tagname);  
  
' Add Data to MyDataSet  
DataTable = MyDataSet.Tables("NavMenu");  
  
' Retrieve data from Array  
foundRows[] = DataTable.Select("CurrentViewTxt like '"+Me.SelectedViewTxt+"'","");  
RecordCount = foundRows[].GetUpperBound(0) + 1;  
  
if (RecordCount > 0) and (foundRows[1].ItemArray(5) == 1) then  
 Me.SelectedRecipe = foundRows[1].ItemArray(6);  
 Me.SelectedIsView = foundRows[1].ItemArray(5);  
 Me.SelectedViewFileTxt = foundRows[1].ItemArray(1);  
endif;  
  
Me.QuerySelectedView = false;

Startup

|  |  |
| --- | --- |
| Name | Startup |
| Description |  |
| Trigger | WhileTrue of (Me.Startup.ExecutionCnt == 0) OR (Me.Startup) |

**Declarations :**

Dim MyDataSet as System.Data.DataSet;  
Dim DataTable as System.Data.DataTable;  
Dim MyDataRow as System.Data.DataRow;  
Dim i as integer;

**Script :**

'Reset  
Me.NavigationArea = "";  
Me.CurrentView = "";  
Me.CurrentViewExpand = "";  
Me.CurrentViewID = 0;  
Me.CurrentViewTxt = "";  
Me.SelectedIsView = 0;  
Me.SelectedRecipe = "";  
Me.SelectedViewFileTxt = "";  
Me.SelectedViewTxt = "";  
  
' Create New DataSet  
MyDataSet = New System.Data.DataSet;  
  
'logmessage("A² Graphic Menu - Startup");  
  
' Add a Table to our new DataSet  
MyDataset.Tables.Add("NavMenu");  
  
DataTable = MyDataset.Tables("NavMenu");  
DataTable.Columns.Add("CurrentFileTxt", System.Type.GetType("System.String"));  
DataTable.Columns.Add("CurrentViewTxt", System.Type.GetType("System.String"));  
DataTable.Columns.Add("FatherViewID", System.Type.GetType("System.Int32"));  
DataTable.Columns.Add("AccessRight", System.Type.GetType("System.Int32"));  
DataTable.Columns.Add("IsView", System.Type.GetType("System.Int32"));  
DataTable.Columns.Add("Recipe", System.Type.GetType("System.String"));  
DataTable.Columns.Add("CurrentViewID", System.Type.GetType("System.Int32"));  
  
  
' Store DataSet in AppDom  
System.AppDomain.CurrentDomain.SetData("AppDomMyDataSet\_"+ me.Tagname,MyDataSet);  
  
'logmessage("A² Graphic Menu - Startup - TS en cours?");  
'  
  
' Read XML to retrieve information  
Me.XMLDSRead=True;  
Me.startup = false;

XMLDSRead

|  |  |
| --- | --- |
| Name | XMLDSRead |
| Description |  |
| Trigger | WhileTrue of Me.XMLDSRead |

**Declarations :**

Dim MyDataSet as System.Data.DataSet;  
Dim DataTable as System.Data.DataTable;  
Dim NavMenuFileName as String;  
Dim ErrorsFileName as String;

**Script :**

'logmessage("A² Graphic Menu - XMLDSRead - loading file: " + Me.xmlpath + Me.filename);  
  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_" + me.tagname);  
'logmessage("A² Graphic Menu - XMLDSRead - récupération dataset dans appdomain ok");  
   
NavMenuFileName = Me.XMLPath + Me.FileName;  
  
'logmessage("A² Graphic Menu - XMLDSRead - Lecture XML");  
  
If (System.IO.File.Exists( NavMenuFileName ) == True) then  
 'logmessage("A² Graphic Menu - XMLDSRead - file exists!");  
 ' Read XML File  
 MyDataset.ReadXml(NavMenuFileName, System.Data.XmlReadMode.IgnoreSchema);  
 'logmessage("A² Graphic Menu - XMLDSRead - Lecture XML OK !!!");  
 Me.XMLOK = true;  
else  
 Me.XMLOK = false;  
 'logmessage("A² Graphic Menu - XMLDSRead - Problème de lecture XML !!!");  
Endif;  
  
' Reset Trigger  
Me.XMLDSRead=False;

Reset

|  |  |
| --- | --- |
| Name | Reset |
| Description |  |
| Trigger | WhileTrue of Me.Reset |

**Declarations :**

Dim j as Integer;

**Script :**

'Reset array  
For j = 1 to 15  
 Me.Frere\_View[j] = "";  
 Me.Fils\_View[j] = "";  
 Me.PetitFils\_View[j] = "";  
Next;  
  
Me.NavigationArea = "";  
  
'CCO  
Me.CurrentView = "";  
Me.CurrentViewExpand = "";  
Me.CurrentViewID = 0;  
Me.CurrentViewTxt = "";  
Me.SelectedIsView = 0;  
Me.SelectedRecipe = "";  
Me.SelectedViewFileTxt = "";  
Me.SelectedViewTxt = "";  
  
Me.Reset = false;

#### Template ArchestrA $NAVI\_BEWHCLI\_old

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| CurrentIsView | boolean |  |  |  |  |  |
| CurrentView | string |  |  |  |  |  |
| CurrentViewExpand | string |  |  |  |  |  |
| CurrentViewID | integer |  |  |  |  |  |
| CurrentViewPere | string |  |  |  |  |  |
| CurrentViewPereID | integer |  |  |  |  |  |
| CurrentViewTxt | string |  |  |  |  |  |
| FileName | string |  |  |  |  |  |
| Fils\_IsView | boolean |  |  |  |  |  |
| Fils\_View | string |  |  |  |  |  |
| FilsCount | integer |  |  |  |  |  |
| Frere\_IsView | boolean |  |  |  |  |  |
| Frere\_View | string |  |  |  |  |  |
| FreresCount | integer |  |  |  |  |  |
| MenuRefresh | boolean |  |  |  |  |  |
| MenuRefreshExpand | boolean |  |  |  |  |  |
| NavigationArea | string |  |  |  |  |  |
| PetitFils\_View | string |  |  |  |  |  |
| PetitFilsCount | integer |  |  |  |  |  |
| QuerySelectedView | boolean |  |  |  |  |  |
| SelectedIsView | boolean |  |  |  |  |  |
| SelectedRecipe | string |  |  |  |  |  |
| SelectedViewFileTxt | string |  |  |  |  |  |
| SelectedViewTxt | string |  |  |  |  |  |
| Startup | boolean |  |  |  |  |  |
| TSEnCours | string |  |  |  |  |  |
| TSName | string |  |  |  |  |  |
| TSOK | boolean |  |  |  |  |  |
| XMLDSRead | boolean |  |  |  |  |  |
| XMLOK | boolean |  |  |  |  |  |
| XMLPath | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

InTouchStartup

|  |  |
| --- | --- |
| Name | InTouchStartup |
| Description |  |
| Trigger | DataChange of Me.TSEnCours |

**Declarations :**

Not Applicable

**Script :**

logmessage("A² Graphic Menu - Init");  
  
Me.NavigationArea = "";  
Me.CurrentView = "";  
Me.CurrentViewExpand = "";  
Me.CurrentViewID = 0;  
Me.CurrentViewTxt = "";  
  
Me.startup = true;

MenuRefresh

|  |  |
| --- | --- |
| Name | MenuRefresh |
| Description |  |
| Trigger | WhileTrue of Me.menurefresh and Me.tsok and Me.xmlok |

**Declarations :**

Dim MyDataSet as System.Data.DataSet;  
Dim DataTable as System.Data.DataTable;  
Dim foundRows[1] as System.Data.DataRow;  
Dim Sort as string;  
Dim i as Integer;  
Dim j as Integer;  
Dim k as Integer;  
Dim RecordCount as integer;

**Script :**

logmessage("A² Graphic Menu - MenuRefresh");  
  
' Retrieve MyDataSet from AppDomain  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_"+ me.Tagname);  
  
' Add Data to MyDataSet  
DataTable = MyDataSet.Tables("NavMenu");  
  
' Retrieve data from Array  
foundRows[] = DataTable.Select("CurrentViewTxt like '"+Me.NavigationArea+"'","");  
RecordCount = foundRows[].GetUpperBound(0) + 1;  
  
if (RecordCount > 0) then  
 logmessage("A² Graphic Menu - MenuRefresh - Vue courante : [ "+foundRows[1].ItemArray(2)+" ]");  
 Me.CurrentViewTxt = foundRows[1].ItemArray(1);  
 Me.CurrentView = foundRows[1].ItemArray(2);  
 Me.CurrentIsView = foundRows[1].ItemArray(5);  
 Me.CurrentViewID = foundRows[1].ItemArray(7);  
 Me.CurrentViewPereID = foundRows[1].ItemArray(3);  
' if(CurrentIsView == false) then  
  
 ' Retrieve data from Array  
 foundRows[] = DataTable.Select("CurrentViewID = "+Me.CurrentViewPereID,"");  
 RecordCount = foundRows[].GetUpperBound(0) + 1;  
   
 if (RecordCount > 0) then  
 Me.CurrentViewPere = foundRows[1].ItemArray(2);   
 endif;  
   
 logmessage("A² Graphic Menu - MenuRefresh - Lancement Query sur table NavMenu");  
 logmessage("A² Graphic Menu - MenuRefresh - QueryString : [ FatherViewID like "+Me.CurrentViewPereID+" ]");  
   
 ' Retrieve data from Array  
 foundRows[] = DataTable.Select("FatherViewID = "+Me.CurrentViewPereID,"CurrentViewID ASC");  
 RecordCount = foundRows[].GetUpperBound(0) + 1;  
   
 Me.FreresCount = RecordCount;  
   
 logmessage("A² Graphic Menu - MenuRefresh - Boucle sur les frères");  
 logmessage("A² Graphic Menu - MenuRefresh - Records : [ "+RecordCount+" ]");  
   
 if (RecordCount > 0) then  
 'logmessage("A² Graphic Menu - MenuRefresh - Loop sur les records récupérés");  
  
  
 'reset les variables  
 For j = 1 to 15  
 Me.Frere\_View[j] = "";  
 Next;  
  
 For j = 1 to RecordCount  
   
 Me.Frere\_View[j] = foundRows[j].ItemArray(2);  
 Me.Frere\_IsView[j] = foundRows[j].ItemArray(5);  
 Next;  
 endif;  
   
 ' Retrieve data from Array  
 foundRows[] = DataTable.Select("FatherViewID = "+Me.CurrentViewID,"CurrentViewID ASC");  
 RecordCount = foundRows[].GetUpperBound(0) + 1;  
   
 Me.FilsCount = RecordCount;  
   
 if (RecordCount > 0) then  
 'logmessage("A² Graphic Menu - MenuRefresh - Loop sur les records récupérés");  
  
 For k = 1 to 15  
 Me.Fils\_View[k] = "";  
 Next;  
  
 For k = 1 to RecordCount  
 Me.Fils\_View[k] = foundRows[k].ItemArray(2);  
 Me.Fils\_IsView[k] = foundRows[k].ItemArray(5);  
 Next;  
 endif;  
' endif;  
endif;  
  
Me.MenuRefresh = false;

MenuRefreshExpand

|  |  |
| --- | --- |
| Name | MenuRefreshExpand |
| Description |  |
| Trigger | WhileTrue of Me.MenuRefreshExpand and Me.TSOK and Me.XMLOK |

**Declarations :**

Dim MyDataSet as System.Data.DataSet;  
Dim DataTable as System.Data.DataTable;  
Dim foundRows[1] as System.Data.DataRow;  
Dim Sort as string;  
Dim i as Integer;  
Dim j as Integer;  
Dim RecordCount as integer;

**Script :**

'logmessage("A² Graphic Menu - MenuRefreshExpand");  
  
' Retrieve MyDataSet from AppDomain  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_"+ me.Tagname);  
  
' Add Data to MyDataSet  
DataTable = MyDataSet.Tables("NavMenu");  
  
' Retrieve data from Array  
foundRows[] = DataTable.Select("CurrentViewTxt like '"+Me.CurrentViewExpand+"'","");  
RecordCount = foundRows[].GetUpperBound(0) + 1;  
  
if (RecordCount > 0) then  
 'logmessage("A² Graphic Menu - MenuRefreshExpand - Vue courante : [ "+foundRows[1].ItemArray(2)+" ]");  
 Me.CurrentViewTxt = foundRows[1].ItemArray(1);  
 Me.CurrentView = foundRows[1].ItemArray(2);  
 Me.CurrentIsView = foundRows[1].ItemArray(5);  
 Me.CurrentViewID = foundRows[1].ItemArray(7);  
 Me.CurrentViewPereID = foundRows[1].ItemArray(3);   
   
 ' Retrieve data from Array  
 foundRows[] = DataTable.Select("FatherViewID = "+Me.CurrentViewID,"CurrentViewID ASC");  
 RecordCount = foundRows[].GetUpperBound(0) + 1;  
   
 Me.PetitFilsCount = RecordCount;  
   
 if (RecordCount > 0) then  
  
 For j = 1 to 15  
 Me.PetitFils\_View[j] = "";  
 Next;  
  
 For j = 1 to RecordCount  
 Me.PetitFils\_View[j] = foundRows[j].ItemArray(2);  
 Next;  
   
 endif;  
' endif;  
endif;  
  
Me.MenuRefreshExpand = false;

QuerySelectedView

|  |  |
| --- | --- |
| Name | QuerySelectedView |
| Description |  |
| Trigger | WhileTrue of Me.QuerySelectedView and Me.TSOK and Me.XMLOK |

**Declarations :**

Dim MyDataSet as System.Data.DataSet;  
Dim DataTable as System.Data.DataTable;  
Dim foundRows[1] as System.Data.DataRow;  
Dim Sort as string;  
Dim RecordCount as integer;

**Script :**

'logmessage("A² Graphic Menu - QuerySelectedView");  
  
' Retrieve MyDataSet from AppDomain  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_"+ me.Tagname);  
  
' Add Data to MyDataSet  
DataTable = MyDataSet.Tables("NavMenu");  
  
' Retrieve data from Array  
foundRows[] = DataTable.Select("CurrentViewTxt like '"+Me.SelectedViewTxt+"'","");  
RecordCount = foundRows[].GetUpperBound(0) + 1;  
  
if (RecordCount > 0) and (foundRows[1].ItemArray(5) == 1) then  
 Me.SelectedRecipe = foundRows[1].ItemArray(6);  
 Me.SelectedIsView = foundRows[1].ItemArray(5);  
 Me.SelectedViewFileTxt = foundRows[1].ItemArray(1);  
endif;  
  
Me.QuerySelectedView = false;

Startup

|  |  |
| --- | --- |
| Name | Startup |
| Description |  |
| Trigger | WhileTrue of Me.Startup |

**Declarations :**

Dim MyDataSet as System.Data.DataSet;  
Dim DataTable as System.Data.DataTable;  
Dim MyDataRow as System.Data.DataRow;  
Dim i as integer;

**Script :**

' Create New DataSet  
MyDataSet = New System.Data.DataSet;  
  
logmessage("A² Graphic Menu - Startup");  
  
' Add a Table to our new DataSet  
MyDataset.Tables.Add("NavMenu");  
  
DataTable = MyDataset.Tables("NavMenu");  
DataTable.Columns.Add("CurrentFileTxt", System.Type.GetType("System.String"));  
DataTable.Columns.Add("CurrentViewTxt", System.Type.GetType("System.String"));  
DataTable.Columns.Add("FatherViewID", System.Type.GetType("System.Int32"));  
DataTable.Columns.Add("AccessRight", System.Type.GetType("System.Int32"));  
DataTable.Columns.Add("IsView", System.Type.GetType("System.Int32"));  
DataTable.Columns.Add("Recipe", System.Type.GetType("System.String"));  
DataTable.Columns.Add("CurrentViewID", System.Type.GetType("System.Int32"));  
  
  
' Store DataSet in AppDom  
System.AppDomain.CurrentDomain.SetData("AppDomMyDataSet\_"+ me.Tagname,MyDataSet);  
  
'logmessage("A² Graphic Menu - Startup - TS en cours?");  
'  
  
if Me.TSName==stringright(Me.TSEnCours,3) then  
 Me.TSOK = true;  
else  
 for i = 1 to 1000  
 if Me.TSName==stringright(Me.TSEnCours,3) then  
 Me.TSOK = true;  
 exit for;  
 else  
 Me.TSOK = False;  
 endif;  
 next;  
endif;  
  
' Read XML to retrieve information  
Me.XMLDSRead=True;  
Me.startup = false;

XMLDSRead

|  |  |
| --- | --- |
| Name | XMLDSRead |
| Description |  |
| Trigger | WhileTrue of Me.XMLDSRead and Me.TSOK |

**Declarations :**

Dim MyDataSet as System.Data.DataSet;  
Dim DataTable as System.Data.DataTable;  
Dim NavMenuFileName as String;  
Dim ErrorsFileName as String;

**Script :**

logmessage("A² Graphic Menu - XMLDSRead - loading file: " + Me.xmlpath + Me.filename);  
  
MyDataSet = System.AppDomain.CurrentDomain.GetData("AppDomMyDataSet\_" + me.tagname);  
logmessage("A² Graphic Menu - XMLDSRead - récupération dataset dans appdomain ok");  
   
NavMenuFileName = Me.XMLPath + Me.FileName;  
  
'logmessage("A² Graphic Menu - XMLDSRead - Lecture XML");  
  
If (System.IO.File.Exists( NavMenuFileName ) == True) then  
 logmessage("A² Graphic Menu - XMLDSRead - file exists!");  
 ' Read XML File  
 MyDataset.ReadXml(NavMenuFileName, System.Data.XmlReadMode.IgnoreSchema);  
 logmessage("A² Graphic Menu - XMLDSRead - Lecture XML OK !!!");  
 Me.XMLOK = true;  
else  
 Me.XMLOK = false;  
 logmessage("A² Graphic Menu - XMLDSRead - Problème de lecture XML !!!");  
Endif;  
  
' Reset Trigger  
Me.XMLDSRead=False;

RetryRead

|  |  |
| --- | --- |
| Name | RetryRead |
| Description |  |
| Trigger | WhileFalse of Me.XMLOK |

**Declarations :**

Not Applicable

**Script :**

logmessage("Problème de lecture du fichier XML : " + Me.xmlpath + Me.filename);  
logmessage("Forcage de la relecture toutes les 30 secondes tant que le probleme persiste");  
  
Me.XMLDSRead =1;

#### Template ArchestrA $Pompe

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$Moteur1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $Resistance

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$Moteur1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $Roller

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$Moteur1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $Souflante

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$Moteur1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | DataChange of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $SQLData

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

Not Applicable

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $String

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$FieldReference

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $TaPing

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| iMilliSec | integer |  |  |  |  |  |
| iRead | integer |  | X |  |  |  |
| iWrite | integer | X |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

copie

|  |  |
| --- | --- |
| Name | copie |
| Description |  |
| Trigger | Periodic of Me.iRead |

**Declarations :**

dim dtRead as System.DateTime;  
dim dtWrite as System.DateTime;

**Script :**

dim ts as System.TimeSpan;  
dtRead = System.DateTime.Now;  
ts = dtRead.Subtract(dtWrite);  
Me.iMilliSec = ts.TotalMilliseconds;  
if Me.iRead == Me.iWrite then  
 if Me.iWrite <> 100 then  
 Me.iWrite = Me.iWrite + 1;  
 else  
 Me.iWrite = 0;  
 endif;  
 dtWrite=System.DateTime.Now;  
endif;

#### Template ArchestrA $Transporteur

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$Moteur1S

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| AffFleche | boolean |  |  |  |  |  |
| Degree | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $Transporteur2S

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$Moteur2S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $UserDefined

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

Not Applicable

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $UserDefined\_MES

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| \_Debug | boolean |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $Vanne1SBis

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

Not Applicable

##### Field Attributes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Attribute name | Data Type | IO Scaling | History | Limit Alarms | Rate Of Change Alarms | Target Deviation Alarms | Bad Value Alarm | Statistics |
| FCO | boolean |  |  |  |  |  |  |  |
| FCF | boolean |  |  |  |  |  |  |  |
| ALARM\_AU\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_AU\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_FERMETURE\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_FERMETURE\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_IS\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_IS\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_LOCAL\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_LOCAL\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_MTH\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_MTH\_A2 | boolean |  |  |  |  |  |  |  |
| ALARM\_OUVERTURE\_A1 | boolean |  |  |  |  |  |  |  |
| ALARM\_OUVERTURE\_A2 | boolean |  |  |  |  |  |  |  |
| CHECK\_DETECTEUR\_AU | boolean |  |  |  |  |  |  |  |
| FermeCmd | boolean |  |  |  |  |  |  |  |
| OuvertCmd | boolean |  |  |  |  |  |  |  |
| STATUT\_DEFAUT\_GENERAL | boolean |  |  |  |  |  |  |  |
| STATUT\_MODE\_LOCAL | boolean |  |  |  |  |  |  |  |
| STATUTS\_MODE\_AUTO | boolean |  |  |  |  |  |  |  |
| ANIM | integer |  |  |  |  |  |  |  |
| PV | float |  |  |  |  |  |  |  |
| TEMPS\_DEFAUT | integer |  |  |  |  |  |  |  |

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

AssignNom

|  |  |
| --- | --- |
| Name | AssignNom |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $Vanne2SensBis

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$UserDefined

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $Vibrant

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$Moteur1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $VIS

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$Moteur1S

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

AssignIo

|  |  |
| --- | --- |
| Name | AssignIo |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

## Control Modules

### Control Modules templates

#### Template ArchestrA $aINDEFF\_Refreshes

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$aINDEFF\_UserDefined\_V\_1\_0

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| DosingLines | integer |  |  |  |  |  |
| Loads | integer |  |  |  |  |  |
| Materials | integer |  |  |  |  |  |
| WorkOrders | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

ResetWorkOrders

|  |  |
| --- | --- |
| Name | ResetWorkOrders |
| Description |  |
| Trigger | WhileTrue of Me.WorkOrders > 10000 |

**Declarations :**

Not Applicable

**Script :**

Me.WorkOrders = 0;

#### Template ArchestrA $dwGenAnalogIn

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$AnalogDevice

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |
| PV\_FB | float |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenAnalogIn  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 IF MyArea.\_Config.Activate.FB Then  
 ' Affect Input source  
 Me.Object.Script.AffectInputSource = True;  
 ENDIF;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenAnalogIn  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.PV.Input.InputSource = "Me.PV\_FB";

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $dwGenAnalogInOut

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$AnalogDevice

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| fPV | float |  |  |  |  |  |
| Object.Script.AffectInputSource | boolean |  |  |  |  |  |
| PV\_FB | float |  |  |  |  |  |
| Ref\_Done | boolean |  |  |  |  |  |
| strArea | string |  |  |  |  |  |
| strName | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

osSecurity

|  |  |
| --- | --- |
| Name | osSecurity |
| Description |  |
| Trigger | WhileTrue of Me.osSecurity.ExecutionCnt <= 2 And Me.Ref\_Done == False |

**Declarations :**

Not Applicable

**Script :**

If Me.osSecurity.ExecutionCnt == 2 Then  
  
 Dim result as integer;  
  
 Me.strName = Me.Tagname;  
  
 result = StringInString(Me.strName, "\_", 1, 0) - 1;  
 Me.strName = StringLeft(Me.strName,result);  
  
 IF stringLeft(Me.strName,2) == "BR" OR Me.strName == "COM" OR Me.strName == "HGS" or Me.strName == "MA01" THEN  
 Me.strArea = "BROYAGE";  
 ELSEIF stringLeft(Me.strName,2) == "CF" OR stringLeft(Me.strName,3) == "FMT" OR stringLeft(Me.strName,3) == "CAC" OR stringLeft(Me.strName,2) == "FR" OR stringLeft(Me.strName,3) == "TSF" THEN  
 Me.strArea = "FOURS";  
 ENDIF;  
  
 Me.Ref\_Done = True;   
Endif;

01\_ObjectDeploy

|  |  |
| --- | --- |
| Name | 01\_ObjectDeploy |
| Description |  |
| Trigger | WhileTrue of Me.01\_ObjectDeploy.ExecutionCnt <= 2 |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenAnalogInOut  
' Script Name : ObjectDeploy  
'----------------------------------------------------------------------------------------   
'  
' Description : Script execute on deployement  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Declaration  
Dim i as integer;  
Dim bfind as boolean;  
  
' Second ExecutionCnt  
If Me.01\_ObjectDeploy.ExecutionCnt == 2 Then  
  
 If MyArea.\_Config.Activate.FB Then  
 ' Affect Input source  
 Me.Object.Script.AffectInputSource = True;  
 ENDIF;  
  
 ' Add object in Area object  
 bfind=False;  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == Me.Tagname Then  
 bfind = True;  
 Exit For;  
 Endif;  
 Next;  
 If Not bfind Then  
 For i=1 To MyArea.List.Object.dimension1   
 ' Already present  
 If MyArea.List.Object[i] == "" Then  
 MyArea.List.Object[i] = Me.Tagname;  
 Exit For;  
 Endif;  
 Next;   
 Endif;  
Endif;

02\_ObjectUnDeploy

|  |  |
| --- | --- |
| Name | 02\_ObjectUnDeploy |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

05\_AffectInputSource

|  |  |
| --- | --- |
| Name | 05\_AffectInputSource |
| Description |  |
| Trigger | WhileTrue of Me.Object.Script.AffectInputSource |

**Declarations :**

Not Applicable

**Script :**

'----------------------------------------------------------------------------------------  
' Object Name : $dwGenAnalogIn  
' Script Name : AffectInputSource  
'----------------------------------------------------------------------------------------   
'  
' Description : Script to affect input source  
'  
'----------------------------------------------------------------------------------------  
' Creator : LDE  
' Date : 01/08/2008  
'----------------------------------------------------------------------------------------  
  
' Reset Trigger  
Me.Object.Script.AffectInputSource=False;  
  
' Affect Inputsource  
Me.PV.Input.InputSource = "Me.PV\_FB";

#### Template ArchestrA $dwGenMoyenneGlissante

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$AnalogDevice

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| fPV | float |  |  |  |  |  |
| fTabValeurs | float |  |  |  |  |  |
| fValeur | float |  | X |  |  |  |
| iCompteur | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

pcMoyenne

|  |  |
| --- | --- |
| Name | pcMoyenne |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

DIM i as integer;  
DIM fTemp as float;  
  
' Mise de la valeur lue dans le PLC dans le tableau  
Me.fTabValeurs[Me.iCompteur] = Me.fValeur;  
  
' Mise à 0 de la variable temporaire  
fTemp = 0;  
  
' Moyenne des valeurs du tableau   
FOR i = 1 TO 30  
 fTemp = fTemp + Me.fTabValeurs[i];  
NEXT;  
  
Me.fPV = fTemp / 30;  
  
' Incrémentation du compteur du tableau fTabValeurs  
Me.iCompteur = Me.iCompteur + 1;  
  
' Si le compteur est plus grand ou égal à 30, on le remet à 0  
IF Me.iCompteur > 30 THEN  
 Me.iCompteur = 0;  
ENDIF;

#### Template ArchestrA $dwGenMoyennePLC

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$AnalogDevice

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarme | boolean |  | X |  | X |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGenRapportMensuel

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$AnalogDevice

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bNewDay | boolean |  |  |  |  |  |
| fPV | float |  |  |  |  | X |
| fTabMoyenneHoraireMois | float |  |  |  |  |  |
| fTabValeursMois | float |  |  |  |  |  |
| fValeur | float |  |  |  |  | X |
| iJourMois | integer |  |  |  |  |  |
| iLastJourMois | integer |  |  |  |  |  |
| iTabCompteurMois | integer |  |  |  |  |  |
| iTabDepassementDoubleSeuilMois | integer |  |  |  |  |  |
| iTabDepassementSeuilMois | integer |  |  |  |  |  |
| iTotalDepassementDoubleSeuil | integer |  |  |  |  | X |
| iTotalDepassementSeuil | integer |  |  |  |  | X |
| iTotalJourDoubleSeuil | integer |  |  |  |  | X |
| iTotalJourSeuil | integer |  |  |  |  | X |

##### Field Attributes

Not Applicable

##### Scripts

dcNewDay

|  |  |
| --- | --- |
| Name | dcNewDay |
| Description |  |
| Trigger | DataChange of Now().Day |

**Declarations :**

Not Applicable

**Script :**

' Affectation de la variable utilisée dans les tableaux  
Me.iLastJourMois = Me.iJourMois;  
Me.iJourMois = Now().Day;  
  
Me.bNewDay = true;

dcMonth

|  |  |
| --- | --- |
| Name | dcMonth |
| Description |  |
| Trigger | DataChange of Now().Month |

**Declarations :**

Not Applicable

**Script :**

DIM i as integer;  
'DIM fValeurMois as float;  
'DIM fMoyenneMois as float;  
'DIM iNbDepSeuilMois as integer;  
'DIM iNbDepDoubleSeuilMois as integer;  
  
'fValeurMois = 0;  
'fMoyenneMois = 0;  
'iNbDepSeuilMois = 0;  
'iNbDepDoubleSeuilMois = 0;  
  
'FOR i = 1 TO 31  
' fValeurMois = fValeurMois + Me.fTabMoyenneHoraireMois[i];'  
' fMoyenneMois = fMoyenneMois + Me.fTabValeursMois[i];  
' iNbDepSeuilMois = iNbDepSeuilMois + Me.iTabDepassementSeuilMois[i];  
' iNbDepDoubleSeuilMois = iNbDepDoubleSeuilMois + Me.iTabDepassementDoubleSeuilMois[i];  
'NEXT;  
  
'Me.fPV = fValeurMois / Me.iLastJourMois;  
'Me.fValeur = fMoyenneMois / Me.iLastJourMois;  
'Me.iTotalDepassementSeuil = iNbDepSeuilMois;  
'Me.iTotalDepassementDoubleSeuil = iNbDepDoubleSeuilMois;  
  
' RAZ  
FOR i = 1 TO 31  
 Me.fTabMoyenneHoraireMois[i] = 0;  
 Me.iTabDepassementDoubleSeuilMois[i] = 0;  
 Me.iTabDepassementSeuilMois[i] = 0;  
 Me.fTabValeursMois[i] = 0;  
NEXT;

otMinuitUne

|  |  |
| --- | --- |
| Name | otMinuitUne |
| Description |  |
| Trigger | OnTrue of Me.bNewDay AND Now().Hour == 0 AND Now().Minute == 1 |

**Declarations :**

Not Applicable

**Script :**

' Minuit 1  
  
Me.bNewDay = false;  
  
Me.fTabMoyenneHoraireMois[Me.iJourMois-1] = Me.dwGenRapportQuotidien.fMoyenneJour;  
Me.fTabValeursMois[Me.iJourMois-1] = Me.dwGenRapportQuotidien.fPVTotal;  
Me.iTabDepassementSeuilMois[Me.iJourMois-1] = Me.dwGenRapportQuotidien.iTotalDepassementSeuil;  
Me.iTabDepassementDoubleSeuilMois[Me.iJourMois-1] = Me.dwGenRapportQuotidien.iTotalDepassementDoubleSeuil;  
  
Me.iTotalJourSeuil = Me.dwGenRapportQuotidien.iTotalDepassementSeuil;  
Me.iTotalJourDoubleSeuil = Me.dwGenRapportQuotidien.iTotalDepassementDoubleSeuil;  
Me.fPV = Me.dwGenRapportQuotidien.fMoyenneJour;

#### Template ArchestrA $dwGenRapportMensuel.dwGenRapportQuotidien

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$AnalogDevice

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| fCalculMoyenneHoraire | float |  |  |  |  |  |
| fDoubleSeuil | float |  |  |  |  |  |
| fMoyenneJour | float |  |  |  |  | X |
| fPlcValeur | float |  | X |  |  |  |
| fPV | float |  |  |  |  | X |
| fPVTotal | float |  |  |  |  | X |
| fSeuil | float |  | X |  |  |  |
| fTabMoyenneHoraire | float |  |  |  |  |  |
| fTabTotalValeurs | float |  |  |  |  |  |
| iCompteur | integer |  |  |  |  |  |
| iDepassementDoubleSeuil | integer |  |  |  |  | X |
| iDepassementSeuil | integer |  |  |  |  | X |
| iHeure | integer |  |  |  |  |  |
| iTabCompteur | integer |  |  |  |  |  |
| iTabDepassementDoubleSeuil | integer |  |  |  |  |  |
| iTabDepassementSeuil | integer |  |  |  |  |  |
| iTotalDepassementDoubleSeuil | integer |  |  |  |  |  |
| iTotalDepassementSeuil | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcChangementHeure

|  |  |
| --- | --- |
| Name | dcChangementHeure |
| Description |  |
| Trigger | DataChange of Now().Hour |

**Declarations :**

Not Applicable

**Script :**

' Lorsqu'il y a un changement d'heure, il faut enregistrer la moyenne précédente et   
' lancer un nouveau calcul de moyenne d'une durée d'1h  
  
Me.iHeure = Now().Hour + 1;  
  
IF Me.iCompteur <> 0 THEN  
 ' Calcul de la moyenne et enregistrement  
 Me.fPV = Me.fCalculMoyenneHoraire / Me.iCompteur;  
ELSE  
 Me.fPV = 0;  
ENDIF;  
  
Me.fTabMoyenneHoraire[Me.iHeure] = Me.fPV;  
Me.iTabCompteur[Me.iHeure] = Me.iCompteur;  
Me.fTabTotalValeurs[Me.iHeure] = Me.fCalculMoyenneHoraire;  
  
' Remise à zéro du compteur  
Me.iCompteur = 0;  
  
' Remise à zéro de la moyenne  
Me.fCalculMoyenneHoraire = 0;  
  
' Dépassement des seuils  
IF Me.fPV > Me.fSeuil THEN  
 Me.iDepassementSeuil = 1;  
ELSE  
 Me.iDepassementSeuil = 0;  
ENDIF;  
  
IF Me.fPV > Me.fDoubleSeuil THEN  
 Me.iDepassementDoubleSeuil = 1;  
ELSE  
 Me.iDepassementDoubleSeuil = 0;  
ENDIF;  
  
Me.iTabDepassementSeuil[Me.iHeure] = Me.iDepassementSeuil;  
Me.iTabDepassementDoubleSeuil[Me.iHeure] = Me.iDepassementDoubleSeuil;

pcMoyenneHoraire

|  |  |
| --- | --- |
| Name | pcMoyenneHoraire |
| Description |  |
| Trigger | Periodic of |

**Declarations :**

Not Applicable

**Script :**

'Ajout de la condition du four en marche  
  
' Incrémentation du compteur  
Me.iCompteur = Me.iCompteur + 1;  
  
' Ajout dans la variable utilisée pour la moyenne de la valeur lue dans le PLC  
Me.fCalculMoyenneHoraire = Me.fCalculMoyenneHoraire + Me.fPLCValeur;

dcNewDay

|  |  |
| --- | --- |
| Name | dcNewDay |
| Description |  |
| Trigger | DataChange of Now().Day |

**Declarations :**

Not Applicable

**Script :**

DIM i as integer;  
DIM fTemp as float;  
DIM fTotalSeuil as float;  
DIM fTotalDoubleSeuil as float;  
DIM iTotalCompteur as integer;  
DIM fTotalValeurs as float;  
  
fTemp = 0;  
fTotalSeuil = 0;  
fTotalDoubleSeuil = 0;  
iTotalCompteur = 0;  
fTotalValeurs = 0;  
  
' Calcul de la moyenne journalière et du total des dépassements de seuils  
FOR i = 1 TO 24  
 iTotalCompteur = iTotalCompteur + Me.iTabCompteur[i];  
 fTemp = fTemp + Me.fTabMoyenneHoraire[i];  
 fTotalSeuil = fTotalSeuil + Me.iTabDepassementSeuil[i];  
 fTotalDoubleSeuil = fTotalDoubleSeuil + Me.iTabDepassementDoubleSeuil[i];  
 fTotalValeurs = fTotalValeurs + Me.fTabTotalValeurs[i];  
NEXT;  
  
Me.fMoyenneJour = fTemp/24;  
  
IF iTotalCompteur <> 0 THEN  
 Me.fPVTotal = fTotalValeurs/iTotalCompteur;  
ELSE  
 Me.fPVTotal = 0;  
ENDIF;  
  
Me.iTotalDepassementDoubleSeuil = fTotalDoubleSeuil;  
Me.iTotalDepassementSeuil = fTotalSeuil;  
  
FOR i = 1 TO 24  
 Me.iTabCompteur[i] = 0;  
 Me.fTabMoyenneHoraire[i] = 0;  
 Me.iTabDepassementSeuil[i] = 0;  
 Me.iTabDepassementDoubleSeuil[i] = 0;  
 Me.fTabTotalValeurs[i] = 0;  
NEXT;

dcSeuil

|  |  |
| --- | --- |
| Name | dcSeuil |
| Description |  |
| Trigger | DataChange of Me.fSeuil |

**Declarations :**

Not Applicable

**Script :**

Me.fDoubleSeuil = Me.fSeuil \* 2;

#### Template ArchestrA $MyAnalogDevice

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$AnalogDevice

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| ALARM\_GEN | boolean |  | X |  |  |  |
| ALARME\_B | boolean |  | X |  | X |  |
| ALARME\_BB | boolean |  | X |  | X |  |
| ALARME\_FORCAGE | boolean |  | X |  | X |  |
| ALARME\_H | boolean |  | X |  | X |  |
| ALARME\_HH | boolean |  | X |  | X |  |
| ALARME\_RUPTURE\_FIL | boolean |  | X |  | X |  |
| EA\_MAX | float | X |  |  |  |  |
| EA\_MIN | float | X |  |  |  |  |
| EU\_Max | float | X |  |  |  |  |
| EU\_Min | float | X |  |  |  |  |
| Nom | string |  |  |  |  |  |
| NomLogix | string |  |  |  |  |  |
| PV\_ANALOG | float |  | X |  |  |  |
| SEUIL\_B | float | X |  |  |  |  |
| SEUIL\_BB | float | X |  |  |  |  |
| SEUIL\_H | float | X |  |  |  |  |
| SEUIL\_HH | float | X |  |  |  |  |
| Seuil\_Hyste | float | X |  |  |  |  |
| VALEUR\_FORCAGE | float | X |  |  |  |  |
| VALIDATION\_B | boolean | X |  |  |  |  |
| VALIDATION\_BB | boolean | X |  |  |  |  |
| VALIDATION\_FORCAGE | boolean | X |  |  |  |  |
| VALIDATION\_H | boolean | X |  |  |  |  |
| VALIDATION\_HH | boolean | X |  |  |  |  |
| VALIDATION\_RUPTURE | boolean | X |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

AssignText

|  |  |
| --- | --- |
| Name | AssignText |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $MyAnalogInOut

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$AnalogDevice

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| NomLogix | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $AnalogDevice

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

Not Applicable

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

## Equipment Modules

### Equipment Modules templates

#### Template ArchestrA $DiscreteDevice

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

Not Applicable

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwGen\_Sonde

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$DiscreteDevice

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| NomLogix | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $Vanne1s

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$DiscreteDevice

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| ALARM\_AU\_A1 | boolean |  | X |  | X |  |
| ALARM\_AU\_A2 | boolean |  | X |  | X |  |
| ALARM\_FERMETURE\_A1 | boolean |  | X |  | X |  |
| ALARM\_FERMETURE\_A2 | boolean |  | X |  | X |  |
| ALARM\_IS\_A1 | boolean |  | X |  | X |  |
| ALARM\_IS\_A2 | boolean |  | X |  | X |  |
| ALARM\_LOCAL\_A1 | boolean |  | X |  | X |  |
| ALARM\_LOCAL\_A2 | boolean |  | X |  | X |  |
| ALARM\_MTH\_A1 | boolean |  | X |  | X |  |
| ALARM\_MTH\_A2 | boolean |  | X |  | X |  |
| ALARM\_OUVERTURE\_A1 | boolean |  | X |  | X |  |
| ALARM\_OUVERTURE\_A2 | boolean |  | X |  | X |  |
| ANIM | integer |  | X |  |  |  |
| CHECK\_DETECTEUR\_AU | boolean | X |  |  |  |  |
| FermeCmd | boolean |  |  | X |  |  |
| Nom | string |  |  |  |  |  |
| NomLogix | string |  |  |  |  |  |
| OuvertCmd | boolean |  |  | X |  |  |
| STATUT\_DEFAUT\_GENERAL | boolean |  | X |  |  |  |
| STATUT\_MODE\_LOCAL | boolean |  | X |  |  |  |
| STATUTS\_MODE\_AUTO | boolean |  | X |  |  |  |
| TEMPS\_DEFAUT | integer | X |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

AssignNom

|  |  |
| --- | --- |
| Name | AssignNom |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $Vanne2s

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$DiscreteDevice

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| ALARM\_AU\_A1 | boolean |  | X |  | X |  |
| ALARM\_AU\_A2 | boolean |  | X |  | X |  |
| ALARM\_FERMETURE\_A1 | boolean |  | X |  | X |  |
| ALARM\_FERMETURE\_A2 | boolean |  | X |  | X |  |
| ALARM\_IS\_A1 | boolean |  | X |  | X |  |
| ALARM\_IS\_A2 | boolean |  | X |  | X |  |
| ALARM\_LOCAL\_A1 | boolean |  | X |  | X |  |
| ALARM\_LOCAL\_A2 | boolean |  | X |  | X |  |
| ALARM\_MTH\_A1 | boolean |  | X |  | X |  |
| ALARM\_MTH\_A2 | boolean |  | X |  | X |  |
| ALARM\_OUVERTURE\_A1 | boolean |  | X |  | X |  |
| ALARM\_OUVERTURE\_A2 | boolean |  | X |  | X |  |
| ANIM | integer |  | X |  |  |  |
| CHECK\_DETECTEUR\_AU | boolean | X |  |  |  |  |
| FermeCmd | boolean |  |  | X |  |  |
| Nom | string |  |  |  |  |  |
| NomLogix | string |  |  |  |  |  |
| OuvertCmd | boolean |  |  | X |  |  |
| Status\_Fermeture | boolean |  | X |  |  |  |
| Status\_Ouverture | boolean |  | X |  |  |  |
| STATUT\_DEFAUT\_GENERAL | boolean |  | X |  |  |  |
| STATUT\_MODE\_AUTO | boolean |  | X |  |  |  |
| STATUT\_MODE\_LOCAL | boolean |  | X |  |  |  |
| sTexteDroite | string |  |  |  |  |  |
| sTexteGauche | string |  |  |  |  |  |
| TEMPS\_DEFAUT | integer | X |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

AssignNom

|  |  |
| --- | --- |
| Name | AssignNom |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $Vanne2s\_BIS

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$DiscreteDevice

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| ALARM\_AU\_A1 | boolean |  | X |  | X |  |
| ALARM\_AU\_A2 | boolean |  | X |  | X |  |
| ALARM\_FERMETURE\_A1 | boolean |  | X |  | X |  |
| ALARM\_FERMETURE\_A2 | boolean |  | X |  | X |  |
| ALARM\_IS\_A1 | boolean |  | X |  | X |  |
| ALARM\_IS\_A2 | boolean |  | X |  | X |  |
| ALARM\_LOCAL\_A1 | boolean |  | X |  | X |  |
| ALARM\_LOCAL\_A2 | boolean |  | X |  | X |  |
| ALARM\_MTH\_A1 | boolean |  | X |  | X |  |
| ALARM\_MTH\_A2 | boolean |  | X |  | X |  |
| ALARM\_OUVERTURE\_A1 | boolean |  | X |  | X |  |
| ALARM\_OUVERTURE\_A2 | boolean |  | X |  | X |  |
| ANIM | integer |  | X |  |  |  |
| CHECK\_DETECTEUR\_AU | boolean | X |  |  |  |  |
| cmd | boolean | X |  |  |  |  |
| Nom | string |  |  |  |  |  |
| NomLogix | string |  |  |  |  |  |
| Status\_Fermeture | boolean |  | X |  |  |  |
| Status\_Ouverture | boolean |  | X |  |  |  |
| STATUT\_DEFAUT\_GENERAL | boolean |  | X |  |  |  |
| STATUT\_MODE\_AUTO | boolean |  | X |  |  |  |
| STATUT\_MODE\_LOCAL | boolean |  | X |  |  |  |
| TEMPS\_DEFAUT | integer | X |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

AssignIO

|  |  |
| --- | --- |
| Name | AssignIO |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

AssignName

|  |  |
| --- | --- |
| Name | AssignName |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

## Unit Modules

### Unit Modules templates

#### Template ArchestrA $FieldReference

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

Not Applicable

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

## Base Function Modules

### Base Function Modules templates

#### Template ArchestrA $dwActionneur

##### Description

Not Applicable

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmeLocal | boolean |  | X |  | X |  |
| bApparaitDansRecette | boolean |  |  |  |  |  |
| bStatusDisponible | boolean |  | X |  |  |  |
| bStatusEnDefaut | boolean |  | X |  |  |  |
| bStatusModeAuto | boolean |  | X |  |  |  |
| bStatusModeLocal | boolean |  | X |  |  |  |
| iIndexRecette | integer | X |  |  |  |  |
| iPlcAlarme\_01 | integer |  | X |  |  |  |
| sAlarmeLocal | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

initTagDansListeurObjet

|  |  |
| --- | --- |
| Name | initTagDansListeurObjet |
| Description |  |
| Trigger | WhileTrue of (sTagDansListeObjets<>Me.tagname) and Me.bApparaitDansRecette |

**Declarations :**

dim sTagDansListeObjets as indirect;

**Script :**

dim index as integer;  
index=Me.iIndexRecette+1;  
'logmessage("maj tag dans "+index);  
sTagDansListeObjets=Me.tagname;  
if WriteStatus(sTagDansListeObjets) <> MxStatusOk then  
 sTagDansListeObjets="";  
 'logmessage(">>>>>>>>>>>>"+MyArea.sTagListeurObjets);  
 sTagDansListeObjets.BindTo(MyArea.sTagListeurObjets+".sTabTagElements["+index+"]");  
else  
 sTagDansListeObjets.BindTo(MyArea.sTagListeurObjets+".sTabTagElements["+index+"]");  
endif;

temp

|  |  |
| --- | --- |
| Name | temp |
| Description |  |
| Trigger | WhileTrue of Me.iIndexRecette <> tmp |

**Declarations :**

dim tmp as integer;

**Script :**

'logmessage("!!! index recette="+Me.iIndexRecette+"<>"+tmp+" forcage à "+tmp);  
Me.iIndexRecette = tmp;

#### Template ArchestrA $dwBoolean

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| Bool | boolean | X |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwDefaut

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarme | boolean |  | X |  | X |  |
| iTempsDefaut | integer | X |  |  |  |  |
| sAlarme | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwEchantillon

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| fPCFine | float |  | X |  |  |  |
| iHeure | integer |  | X |  |  |  |
| iMinute | integer |  | X |  |  |  |
| sTemps | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

CalcTemps

|  |  |
| --- | --- |
| Name | CalcTemps |
| Description |  |
| Trigger | DataChange of Me.iHeure + Me.iMinute |

**Declarations :**

Not Applicable

**Script :**

dim dt as System.DateTime;  
dt= new System.DateTime(2000,1,1,Me.iHeure,Me.iMinute,0);  
Me.sTemps=dt.ToShortTimeString();  
{dim ts as System.TimeSpan;  
ts = new System.TimeSpan(Me.iHeure,Me.iMinute,0);}

#### Template ArchestrA $dwEnginRecette

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| iIndex | integer |  | X |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwFloatIn

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| fValeur | float |  | X |  |  |  |
| iPrecision | integer |  |  |  |  |  |
| sUnite | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwFloatInOut

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| fValeur | float | X |  |  |  |  |
| iPrecision | integer |  |  |  |  |  |
| sUnite | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwIntegerMux

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| Bool00 | boolean | X |  |  |  |  |
| Bool01 | boolean | X |  |  |  |  |
| Bool02 | boolean | X |  |  |  |  |
| Bool03 | boolean | X |  |  |  |  |
| Bool04 | boolean | X |  |  |  |  |
| Bool05 | boolean | X |  |  |  |  |
| Bool06 | boolean | X |  |  |  |  |
| Bool07 | boolean | X |  |  |  |  |
| Bool08 | boolean | X |  |  |  |  |
| Bool09 | boolean | X |  |  |  |  |
| Bool10 | boolean | X |  |  |  |  |
| Bool11 | boolean | X |  |  |  |  |
| Bool12 | boolean | X |  |  |  |  |
| Bool13 | boolean | X |  |  |  |  |
| Bool14 | boolean | X |  |  |  |  |
| Bool15 | boolean | X |  |  |  |  |
| iPV | integer | X |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwMesureAnalogique

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmeB | boolean |  | X |  | X |  |
| bAlarmeBB | boolean |  | X |  | X |  |
| bAlarmeH | boolean |  | X |  | X |  |
| bAlarmeHH | boolean |  | X |  | X |  |
| bAlarmeRuptureFil | boolean |  | X |  | X |  |
| bInhibitionDefB | boolean | X |  |  |  |  |
| bInhibitionDefBB | boolean | X |  |  |  |  |
| bInhibitionDefH | boolean | X |  |  |  |  |
| bInhibitionDefHH | boolean | X |  |  |  |  |
| bInhibitionDefRuptureFil | boolean | X |  |  |  |  |
| bStatusEnDefaut | boolean |  | X |  |  |  |
| fFacteurLissage | float | X |  |  |  |  |
| iPrecision | integer |  |  |  |  |  |
| rHysterese | float | X |  |  |  |  |
| rMaxEA | float | X |  |  |  |  |
| rMaxEU | float | X |  |  |  |  |
| rMese | float |  | X |  |  | X |
| rMinEA | float | X |  |  |  |  |
| rMinEU | float | X |  |  |  |  |
| rSeuilB | float | X |  |  |  |  |
| rSeuilBB | float | X |  |  |  |  |
| rSeuilH | float | X |  |  |  |  |
| rSeuilHH | float | X |  |  |  |  |
| sAlarmeB | string |  |  |  |  |  |
| sAlarmeBB | string |  |  |  |  |  |
| sAlarmeH | string |  |  |  |  |  |
| sAlarmeHH | string |  |  |  |  |  |
| sAlarmeRuptureFil | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

dcPrecision

|  |  |
| --- | --- |
| Name | dcPrecision |
| Description |  |
| Trigger | DataChange of Me.iPrecision |

**Declarations :**

Not Applicable

**Script :**

Me.rMese.ValueDeadBand=System.Math.Pow( 10, - Me.iPrecision );

#### Template ArchestrA $dwPID

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmeEcartConsigne | boolean |  | X |  | X |  |
| bInhibitionAlarmeEcartConsigne | boolean | X |  |  |  |  |
| bModeInterne | boolean | X |  |  |  |  |
| bModeManu | integer | X |  |  |  |  |
| fACT | float | X |  |  |  | X |
| fACTMax | float | X |  |  |  |  |
| fACTMaxEU | float | X |  |  |  |  |
| fACTMin | float | X |  |  |  |  |
| fACTMinEU | float | X |  |  |  |  |
| fAlarmeEcartConsigne | float | X |  |  |  |  |
| fAlarmeEcartConsigneTemps | float | X |  |  |  |  |
| fCSG | float | X |  |  |  | X |
| fCSGMax | float | X |  |  |  |  |
| fCSGMin | float | X |  |  |  |  |
| fK | float | X |  |  |  |  |
| fMES | float | X |  |  |  | X |
| fMESMaxEU | float | X |  |  |  |  |
| fMESMinEU | float | X |  |  |  |  |
| fTD | float | X |  |  |  |  |
| fTI | float | X |  |  |  |  |
| sACTUnite | string |  |  |  |  |  |
| sAlarmeEcartConsigne | string |  |  |  |  |  |
| sMESUnite | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

#### Template ArchestrA $dwPretA

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bAlarmePretA | boolean |  | X |  | X |  |
| iPrioritePretA | integer |  |  |  |  |  |
| sAlarmePretA | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

PrioritePretA

|  |  |
| --- | --- |
| Name | PrioritePretA |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

#### Template ArchestrA $dwRecette

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bCmdAckDef | boolean | X |  |  |  |  |
| bCmdArretEnCharge | boolean | X |  |  |  |  |
| bCmdAuto | boolean | X |  |  |  |  |
| bCmdManu | boolean | X |  |  |  |  |
| bCmdStop | boolean | X |  |  |  |  |
| bStsArretEnCharge | boolean |  | X |  |  |  |
| bStsAuto | boolean |  | X |  |  |  |
| bStsManu | boolean |  | X |  |  |  |
| bStsStop | boolean |  | X |  |  |  |
| iNumero | integer |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

initTagDansListeurObjet

|  |  |
| --- | --- |
| Name | initTagDansListeurObjet |
| Description |  |
| Trigger | WhileTrue of sTagDansListeObjets<>Me.tagname |

**Declarations :**

dim sTagDansListeObjets as indirect;

**Script :**

sTagDansListeObjets=Me.tagname;  
if WriteStatus(sTagDansListeObjets) <> MxStatusOk then  
 sTagDansListeObjets="";  
' logmessage(">>>>>>>>>>>>"+MyArea.sTagListeurObjets);  
 sTagDansListeObjets.BindTo(MyArea.sTagListeurObjets+".sTabRecettes["+Me.iNumero+"]");  
endif;

#### Template ArchestrA $dwRecettes

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bCmdReset | string |  |  |  |  |  |
| bCopie | boolean |  |  |  |  |  |
| bGetRecette | boolean | X |  |  |  |  |
| bSetRecette | boolean | X |  |  |  |  |
| bStatusAutoRecette\_1 | boolean |  | X |  |  |  |
| bStatusAutoRecette\_10 | boolean |  | X |  |  |  |
| bStatusAutoRecette\_11 | boolean |  | X |  |  |  |
| bStatusAutoRecette\_12 | boolean |  | X |  |  |  |
| bStatusAutoRecette\_13 | boolean |  | X |  |  |  |
| bStatusAutoRecette\_14 | boolean |  | X |  |  |  |
| bStatusAutoRecette\_15 | boolean |  | X |  |  |  |
| bStatusAutoRecette\_16 | boolean |  | X |  |  |  |
| bStatusAutoRecette\_2 | boolean |  | X |  |  |  |
| bStatusAutoRecette\_3 | boolean |  | X |  |  |  |
| bStatusAutoRecette\_4 | boolean |  | X |  |  |  |
| bStatusAutoRecette\_5 | boolean |  | X |  |  |  |
| bStatusAutoRecette\_6 | boolean |  | X |  |  |  |
| bStatusAutoRecette\_7 | boolean |  | X |  |  |  |
| bStatusAutoRecette\_8 | boolean |  | X |  |  |  |
| bStatusAutoRecette\_9 | boolean |  | X |  |  |  |
| iIndexAlimentateurs\_1 | integer |  | X |  |  |  |
| iIndexAlimentateurs\_10 | integer |  | X |  |  |  |
| iIndexAlimentateurs\_11 | integer |  | X |  |  |  |
| iIndexAlimentateurs\_12 | integer |  | X |  |  |  |
| iIndexAlimentateurs\_13 | integer |  | X |  |  |  |
| iIndexAlimentateurs\_2 | integer |  | X |  |  |  |
| iIndexAlimentateurs\_3 | integer |  | X |  |  |  |
| iIndexAlimentateurs\_4 | integer |  | X |  |  |  |
| iIndexAlimentateurs\_5 | integer |  | X |  |  |  |
| iIndexAlimentateurs\_6 | integer |  | X |  |  |  |
| iIndexAlimentateurs\_7 | integer |  | X |  |  |  |
| iIndexAlimentateurs\_8 | integer |  | X |  |  |  |
| iIndexAlimentateurs\_9 | integer |  | X |  |  |  |
| iNumero | integer | X |  |  |  |  |
| iNumeroRecetteEnCours | integer |  |  |  |  |  |
| iStatusAncienAlim\_1 | integer | X |  |  |  |  |
| iStatusAncienAlim\_10 | integer | X |  |  |  |  |
| iStatusAncienAlim\_11 | integer | X |  |  |  |  |
| iStatusAncienAlim\_12 | integer | X |  |  |  |  |
| iStatusAncienAlim\_13 | integer | X |  |  |  |  |
| iStatusAncienAlim\_2 | integer | X |  |  |  |  |
| iStatusAncienAlim\_3 | integer | X |  |  |  |  |
| iStatusAncienAlim\_4 | integer | X |  |  |  |  |
| iStatusAncienAlim\_5 | integer | X |  |  |  |  |
| iStatusAncienAlim\_6 | integer | X |  |  |  |  |
| iStatusAncienAlim\_7 | integer | X |  |  |  |  |
| iStatusAncienAlim\_8 | integer | X |  |  |  |  |
| iStatusAncienAlim\_9 | integer | X |  |  |  |  |
| iStatusElement\_1 | integer | X |  |  |  |  |
| iStatusElement\_10 | integer | X |  |  |  |  |
| iStatusElement\_100 | integer | X |  |  |  |  |
| iStatusElement\_101 | integer | X |  |  |  |  |
| iStatusElement\_102 | integer | X |  |  |  |  |
| iStatusElement\_103 | integer | X |  |  |  |  |
| iStatusElement\_104 | integer | X |  |  |  |  |
| iStatusElement\_105 | integer | X |  |  |  |  |
| iStatusElement\_106 | integer | X |  |  |  |  |
| iStatusElement\_107 | integer | X |  |  |  |  |
| iStatusElement\_108 | integer | X |  |  |  |  |
| iStatusElement\_109 | integer | X |  |  |  |  |
| iStatusElement\_11 | integer | X |  |  |  |  |
| iStatusElement\_110 | integer | X |  |  |  |  |
| iStatusElement\_111 | integer | X |  |  |  |  |
| iStatusElement\_112 | integer | X |  |  |  |  |
| iStatusElement\_113 | integer | X |  |  |  |  |
| iStatusElement\_114 | integer | X |  |  |  |  |
| iStatusElement\_115 | integer | X |  |  |  |  |
| iStatusElement\_116 | integer | X |  |  |  |  |
| iStatusElement\_117 | integer | X |  |  |  |  |
| iStatusElement\_118 | integer | X |  |  |  |  |
| iStatusElement\_119 | integer | X |  |  |  |  |
| iStatusElement\_12 | integer | X |  |  |  |  |
| iStatusElement\_120 | integer | X |  |  |  |  |
| iStatusElement\_121 | integer | X |  |  |  |  |
| iStatusElement\_122 | integer | X |  |  |  |  |
| iStatusElement\_123 | integer | X |  |  |  |  |
| iStatusElement\_124 | integer | X |  |  |  |  |
| iStatusElement\_125 | integer | X |  |  |  |  |
| iStatusElement\_126 | integer | X |  |  |  |  |
| iStatusElement\_127 | integer | X |  |  |  |  |
| iStatusElement\_128 | integer | X |  |  |  |  |
| iStatusElement\_129 | integer | X |  |  |  |  |
| iStatusElement\_13 | integer | X |  |  |  |  |
| iStatusElement\_130 | integer | X |  |  |  |  |
| iStatusElement\_131 | integer | X |  |  |  |  |
| iStatusElement\_132 | integer | X |  |  |  |  |
| iStatusElement\_133 | integer | X |  |  |  |  |
| iStatusElement\_134 | integer | X |  |  |  |  |
| iStatusElement\_135 | integer | X |  |  |  |  |
| iStatusElement\_136 | integer | X |  |  |  |  |
| iStatusElement\_137 | integer | X |  |  |  |  |
| iStatusElement\_138 | integer | X |  |  |  |  |
| iStatusElement\_139 | integer | X |  |  |  |  |
| iStatusElement\_14 | integer | X |  |  |  |  |
| iStatusElement\_140 | integer | X |  |  |  |  |
| iStatusElement\_141 | integer | X |  |  |  |  |
| iStatusElement\_142 | integer | X |  |  |  |  |
| iStatusElement\_143 | integer | X |  |  |  |  |
| iStatusElement\_144 | integer | X |  |  |  |  |
| iStatusElement\_145 | integer | X |  |  |  |  |
| iStatusElement\_146 | integer | X |  |  |  |  |
| iStatusElement\_147 | integer | X |  |  |  |  |
| iStatusElement\_148 | integer | X |  |  |  |  |
| iStatusElement\_149 | integer | X |  |  |  |  |
| iStatusElement\_15 | integer | X |  |  |  |  |
| iStatusElement\_150 | integer | X |  |  |  |  |
| iStatusElement\_16 | integer | X |  |  |  |  |
| iStatusElement\_17 | integer | X |  |  |  |  |
| iStatusElement\_18 | integer | X |  |  |  |  |
| iStatusElement\_19 | integer | X |  |  |  |  |
| iStatusElement\_2 | integer | X |  |  |  |  |
| iStatusElement\_20 | integer | X |  |  |  |  |
| iStatusElement\_21 | integer | X |  |  |  |  |
| iStatusElement\_22 | integer | X |  |  |  |  |
| iStatusElement\_23 | integer | X |  |  |  |  |
| iStatusElement\_24 | integer | X |  |  |  |  |
| iStatusElement\_25 | integer | X |  |  |  |  |
| iStatusElement\_26 | integer | X |  |  |  |  |
| iStatusElement\_27 | integer | X |  |  |  |  |
| iStatusElement\_28 | integer | X |  |  |  |  |
| iStatusElement\_29 | integer | X |  |  |  |  |
| iStatusElement\_3 | integer | X |  |  |  |  |
| iStatusElement\_30 | integer | X |  |  |  |  |
| iStatusElement\_31 | integer | X |  |  |  |  |
| iStatusElement\_32 | integer | X |  |  |  |  |
| iStatusElement\_33 | integer | X |  |  |  |  |
| iStatusElement\_34 | integer | X |  |  |  |  |
| iStatusElement\_35 | integer | X |  |  |  |  |
| iStatusElement\_36 | integer | X |  |  |  |  |
| iStatusElement\_37 | integer | X |  |  |  |  |
| iStatusElement\_38 | integer | X |  |  |  |  |
| iStatusElement\_39 | integer | X |  |  |  |  |
| iStatusElement\_4 | integer | X |  |  |  |  |
| iStatusElement\_40 | integer | X |  |  |  |  |
| iStatusElement\_41 | integer | X |  |  |  |  |
| iStatusElement\_42 | integer | X |  |  |  |  |
| iStatusElement\_43 | integer | X |  |  |  |  |
| iStatusElement\_44 | integer | X |  |  |  |  |
| iStatusElement\_45 | integer | X |  |  |  |  |
| iStatusElement\_46 | integer | X |  |  |  |  |
| iStatusElement\_47 | integer | X |  |  |  |  |
| iStatusElement\_48 | integer | X |  |  |  |  |
| iStatusElement\_49 | integer | X |  |  |  |  |
| iStatusElement\_5 | integer | X |  |  |  |  |
| iStatusElement\_50 | integer | X |  |  |  |  |
| iStatusElement\_51 | integer | X |  |  |  |  |
| iStatusElement\_52 | integer | X |  |  |  |  |
| iStatusElement\_53 | integer | X |  |  |  |  |
| iStatusElement\_54 | integer | X |  |  |  |  |
| iStatusElement\_55 | integer | X |  |  |  |  |
| iStatusElement\_56 | integer | X |  |  |  |  |
| iStatusElement\_57 | integer | X |  |  |  |  |
| iStatusElement\_58 | integer | X |  |  |  |  |
| iStatusElement\_59 | integer | X |  |  |  |  |
| iStatusElement\_6 | integer | X |  |  |  |  |
| iStatusElement\_60 | integer | X |  |  |  |  |
| iStatusElement\_61 | integer | X |  |  |  |  |
| iStatusElement\_62 | integer | X |  |  |  |  |
| iStatusElement\_63 | integer | X |  |  |  |  |
| iStatusElement\_64 | integer | X |  |  |  |  |
| iStatusElement\_65 | integer | X |  |  |  |  |
| iStatusElement\_66 | integer | X |  |  |  |  |
| iStatusElement\_67 | integer | X |  |  |  |  |
| iStatusElement\_68 | integer | X |  |  |  |  |
| iStatusElement\_69 | integer | X |  |  |  |  |
| iStatusElement\_7 | integer | X |  |  |  |  |
| iStatusElement\_70 | integer | X |  |  |  |  |
| iStatusElement\_71 | integer | X |  |  |  |  |
| iStatusElement\_72 | integer | X |  |  |  |  |
| iStatusElement\_73 | integer | X |  |  |  |  |
| iStatusElement\_74 | integer | X |  |  |  |  |
| iStatusElement\_75 | integer | X |  |  |  |  |
| iStatusElement\_76 | integer | X |  |  |  |  |
| iStatusElement\_77 | integer | X |  |  |  |  |
| iStatusElement\_78 | integer | X |  |  |  |  |
| iStatusElement\_79 | integer | X |  |  |  |  |
| iStatusElement\_8 | integer | X |  |  |  |  |
| iStatusElement\_80 | integer | X |  |  |  |  |
| iStatusElement\_81 | integer | X |  |  |  |  |
| iStatusElement\_82 | integer | X |  |  |  |  |
| iStatusElement\_83 | integer | X |  |  |  |  |
| iStatusElement\_84 | integer | X |  |  |  |  |
| iStatusElement\_85 | integer | X |  |  |  |  |
| iStatusElement\_86 | integer | X |  |  |  |  |
| iStatusElement\_87 | integer | X |  |  |  |  |
| iStatusElement\_88 | integer | X |  |  |  |  |
| iStatusElement\_89 | integer | X |  |  |  |  |
| iStatusElement\_9 | integer | X |  |  |  |  |
| iStatusElement\_90 | integer | X |  |  |  |  |
| iStatusElement\_91 | integer | X |  |  |  |  |
| iStatusElement\_92 | integer | X |  |  |  |  |
| iStatusElement\_93 | integer | X |  |  |  |  |
| iStatusElement\_94 | integer | X |  |  |  |  |
| iStatusElement\_95 | integer | X |  |  |  |  |
| iStatusElement\_96 | integer | X |  |  |  |  |
| iStatusElement\_97 | integer | X |  |  |  |  |
| iStatusElement\_98 | integer | X |  |  |  |  |
| iStatusElement\_99 | integer | X |  |  |  |  |
| iStatusNouveauAlim\_1 | integer |  |  |  |  |  |
| iStatusNouveauAlim\_10 | integer |  |  |  |  |  |
| iStatusNouveauAlim\_11 | integer |  |  |  |  |  |
| iStatusNouveauAlim\_12 | integer |  |  |  |  |  |
| iStatusNouveauAlim\_13 | integer |  |  |  |  |  |
| iStatusNouveauAlim\_2 | integer |  |  |  |  |  |
| iStatusNouveauAlim\_3 | integer |  |  |  |  |  |
| iStatusNouveauAlim\_4 | integer |  |  |  |  |  |
| iStatusNouveauAlim\_5 | integer |  |  |  |  |  |
| iStatusNouveauAlim\_6 | integer |  |  |  |  |  |
| iStatusNouveauAlim\_7 | integer |  |  |  |  |  |
| iStatusNouveauAlim\_8 | integer |  |  |  |  |  |
| iStatusNouveauAlim\_9 | integer |  |  |  |  |  |
| sInputsources | string |  |  |  |  |  |
| sItemRecette\_1 | string |  | X |  |  |  |
| sItemRecette\_10 | string |  | X |  |  |  |
| sItemRecette\_11 | string |  | X |  |  |  |
| sItemRecette\_12 | string |  | X |  |  |  |
| sItemRecette\_13 | string |  | X |  |  |  |
| sItemRecette\_14 | string |  | X |  |  |  |
| sItemRecette\_15 | string |  | X |  |  |  |
| sItemRecette\_16 | string |  | X |  |  |  |
| sItemRecette\_2 | string |  | X |  |  |  |
| sItemRecette\_3 | string |  | X |  |  |  |
| sItemRecette\_4 | string |  | X |  |  |  |
| sItemRecette\_5 | string |  | X |  |  |  |
| sItemRecette\_6 | string |  | X |  |  |  |
| sItemRecette\_7 | string |  | X |  |  |  |
| sItemRecette\_8 | string |  | X |  |  |  |
| sItemRecette\_9 | string |  | X |  |  |  |
| sRecettesEnCours | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

initTemporaire

|  |  |
| --- | --- |
| Name | initTemporaire |
| Description |  |
| Trigger | WhileTrue of |

**Declarations :**

Not Applicable

**Script :**

Not Applicable

Reset

|  |  |
| --- | --- |
| Name | Reset |
| Description |  |
| Trigger | WhileTrue of Me.bCmdReset |

**Declarations :**

Not Applicable

**Script :**

Me.bCmdReset=false;  
dim x as indirect;  
dim j as integer;  
Me.iStatusElement\_1=0;  
for j=2 to 150  
 x.BindTo("Me.iStatusElement\_"+j);  
 x=-32768;  
next;

otCopieAlim

|  |  |
| --- | --- |
| Name | otCopieAlim |
| Description |  |
| Trigger | WhileTrue of Me.bCopie |

**Declarations :**

Not Applicable

**Script :**

'logmessage("debut copie");  
dim i as integer;  
dim nouveau as indirect;  
dim ancien as indirect;  
  
for i=1 to 13  
 nouveau.bindto("Me.iStatusNouveauAlim\_"+i);  
 ancien.bindto("Me.iStatusAncienAlim\_"+i);  
' logmessage("Me.iStatusAncienAlim\_"+i+"=Me.iStatusNouveauAlim\_"+i+"("+nouveau+")");  
 ancien=nouveau;  
next;  
Me.bCopie=false;  
'logmessage("fin copie");

dcIndexAlim

|  |  |
| --- | --- |
| Name | dcIndexAlim |
| Description |  |
| Trigger | DataChange of Me.iIndexAlimentateurs\_1^Me.iIndexAlimentateurs\_2^Me.iIndexAlimentateurs\_3^Me.iIndexAlimentateurs\_4^Me.iIndexAlimentateurs\_5^Me.iIndexAlimentateurs\_6^Me.iIndexAlimentateurs\_7^Me.iIndexAlimentateurs\_8^Me.iIndexAlimentateurs\_9^Me.iIndexAlimentateurs\_10^Me.iIndexAlimentateurs\_11 |

**Declarations :**

Not Applicable

**Script :**

dim i as integer;  
dim inputsource as indirect;  
dim indexAlim as indirect;  
dim index as integer;  
for i=1 to 13  
 inputsource.bindto("Me.iStatusAncienAlim\_"+i+".inputsource");  
 indexAlim.bindto("Me.iIndexAlimentateurs\_"+i);  
 index=indexAlim+1;  
 inputsource="Me.iStatusElement\_"+index;  
next;

CalcsRecettesEnCours

|  |  |
| --- | --- |
| Name | CalcsRecettesEnCours |
| Description |  |
| Trigger | DataChange of Me.bStatusAutoRecette\_1 + Me.bStatusAutoRecette\_2 + Me.bStatusAutoRecette\_3 + Me.bStatusAutoRecette\_4 + Me.bStatusAutoRecette\_5 + Me.bStatusAutoRecette\_6 + Me.bStatusAutoRecette\_7 + Me.bStatusAutoRecette\_8 + Me.bStatusAutoRecette\_9 + Me.bStatusAutoRecette\_10 + Me.bStatusAutoRecette\_11 + Me.bStatusAutoRecette\_12 + Me.bStatusAutoRecette\_13 + Me.bStatusAutoRecette\_14 + Me.bStatusAutoRecette\_15 + Me.bStatusAutoRecette\_16 |

**Declarations :**

Not Applicable

**Script :**

Me.sRecettesEnCours="";  
Me.iNumeroRecetteEnCours=0;  
if Me.bStatusAutoRecette\_1 == true then  
 Me.iNumeroRecetteEnCours=1;  
 if Me.sRecettesEnCours == "" then  
 Me.sRecettesEnCours = Me.sItemRecette\_1;  
 else  
 Me.sRecettesEnCours = Me.sRecettesEnCours + ", " + Me.sItemRecette\_1;  
 endif;  
endif;  
if Me.bStatusAutoRecette\_2 == true then  
 Me.iNumeroRecetteEnCours=2;  
 if Me.sRecettesEnCours == "" then  
 Me.sRecettesEnCours = Me.sItemRecette\_2;  
 else  
 Me.sRecettesEnCours = Me.sRecettesEnCours + ", " + Me.sItemRecette\_2;  
 endif;  
endif;  
if Me.bStatusAutoRecette\_3 == true then  
 Me.iNumeroRecetteEnCours=3;  
 if Me.sRecettesEnCours == "" then  
 Me.sRecettesEnCours = Me.sItemRecette\_3;  
 else  
 Me.sRecettesEnCours = Me.sRecettesEnCours + ", " + Me.sItemRecette\_3;  
 endif;  
endif;  
if Me.bStatusAutoRecette\_4 == true then  
 Me.iNumeroRecetteEnCours=4;  
 if Me.sRecettesEnCours == "" then  
 Me.sRecettesEnCours = Me.sItemRecette\_4;  
 else  
 Me.sRecettesEnCours = Me.sRecettesEnCours + ", " + Me.sItemRecette\_4;  
 endif;  
endif;  
if Me.bStatusAutoRecette\_5 == true then  
 Me.iNumeroRecetteEnCours=5;  
 if Me.sRecettesEnCours == "" then  
 Me.sRecettesEnCours = Me.sItemRecette\_5;  
 else  
 Me.sRecettesEnCours = Me.sRecettesEnCours + ", " + Me.sItemRecette\_5;  
 endif;  
endif;  
if Me.bStatusAutoRecette\_6 == true then  
 Me.iNumeroRecetteEnCours=6;  
 if Me.sRecettesEnCours == "" then  
 Me.sRecettesEnCours = Me.sItemRecette\_6;  
 else  
 Me.sRecettesEnCours = Me.sRecettesEnCours + ", " + Me.sItemRecette\_6;  
 endif;  
endif;  
if Me.bStatusAutoRecette\_7 == true then  
 Me.iNumeroRecetteEnCours=7;  
 if Me.sRecettesEnCours == "" then  
 Me.sRecettesEnCours = Me.sItemRecette\_7;  
 else  
 Me.sRecettesEnCours = Me.sRecettesEnCours + ", " + Me.sItemRecette\_7;  
 endif;  
endif;  
if Me.bStatusAutoRecette\_8 == true then  
 Me.iNumeroRecetteEnCours=8;  
 if Me.sRecettesEnCours == "" then  
 Me.sRecettesEnCours = Me.sItemRecette\_8;  
 else  
 Me.sRecettesEnCours = Me.sRecettesEnCours + ", " + Me.sItemRecette\_8;  
 endif;  
endif;  
if Me.bStatusAutoRecette\_9 == true then  
 Me.iNumeroRecetteEnCours=9;  
 if Me.sRecettesEnCours == "" then  
 Me.sRecettesEnCours = Me.sItemRecette\_9;  
 else  
 Me.sRecettesEnCours = Me.sRecettesEnCours + ", " + Me.sItemRecette\_9;  
 endif;  
endif;  
if Me.bStatusAutoRecette\_10 == true then  
 Me.iNumeroRecetteEnCours=10;  
 if Me.sRecettesEnCours == "" then  
 Me.sRecettesEnCours = Me.sItemRecette\_10;  
 else  
 Me.sRecettesEnCours = Me.sRecettesEnCours + ", " + Me.sItemRecette\_10;  
 endif;  
endif;  
if Me.bStatusAutoRecette\_11 == true then  
 Me.iNumeroRecetteEnCours=11;  
 if Me.sRecettesEnCours == "" then  
 Me.sRecettesEnCours = Me.sItemRecette\_11;  
 else  
 Me.sRecettesEnCours = Me.sRecettesEnCours + ", " + Me.sItemRecette\_11;  
 endif;  
endif;  
  
if Me.bStatusAutoRecette\_12 == true then  
 Me.iNumeroRecetteEnCours=12;  
 if Me.sRecettesEnCours == "" then  
 Me.sRecettesEnCours = Me.sItemRecette\_12;  
 else  
 Me.sRecettesEnCours = Me.sRecettesEnCours + ", " + Me.sItemRecette\_12;  
 endif;  
endif;  
  
if Me.bStatusAutoRecette\_13 == true then  
 Me.iNumeroRecetteEnCours=13;  
 if Me.sRecettesEnCours == "" then  
 Me.sRecettesEnCours = Me.sItemRecette\_13;  
 else  
 Me.sRecettesEnCours = Me.sRecettesEnCours + ", " + Me.sItemRecette\_13;  
 endif;  
endif;  
  
if Me.bStatusAutoRecette\_14 == true then  
 Me.iNumeroRecetteEnCours=14;  
 if Me.sRecettesEnCours == "" then  
 Me.sRecettesEnCours = Me.sItemRecette\_14;  
 else  
 Me.sRecettesEnCours = Me.sRecettesEnCours + ", " + Me.sItemRecette\_14;  
 endif;  
endif;  
  
if Me.bStatusAutoRecette\_15 == true then  
 Me.iNumeroRecetteEnCours=15;  
 if Me.sRecettesEnCours == "" then  
 Me.sRecettesEnCours = Me.sItemRecette\_15;  
 else  
 Me.sRecettesEnCours = Me.sRecettesEnCours + ", " + Me.sItemRecette\_15;  
 endif;  
endif;  
  
if Me.bStatusAutoRecette\_16 == true then  
 Me.iNumeroRecetteEnCours=16;  
 if Me.sRecettesEnCours == "" then  
 Me.sRecettesEnCours = Me.sItemRecette\_16;  
 else  
 Me.sRecettesEnCours = Me.sRecettesEnCours + ", " + Me.sItemRecette\_16;  
 endif;  
endif;

#### Template ArchestrA $dwSelecteur

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bCmdSelection | boolean | X |  |  |  |  |
| bStatusSelection | boolean |  | X |  |  |  |
| sTexteEnCours | string |  |  |  |  |  |
| sTexteFALSE | string |  |  |  |  |  |
| sTexteTRUE | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

calcsTexteEnCours

|  |  |
| --- | --- |
| Name | calcsTexteEnCours |
| Description |  |
| Trigger | DataChange of Me.bStatusSelection |

**Declarations :**

Not Applicable

**Script :**

if Me.bStatusSelection == TRUE then  
 Me.sTexteEnCours = Me.sTexteTRUE;  
else  
 Me.sTexteEnCours = Me.sTexteFALSE;  
endif;  
Me.sTexteEnCours.quality = Me.bStatusSelection.quality;

#### Template ArchestrA $dwSelecteur3

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bCmdSelection1 | boolean | X |  |  |  |  |
| bCmdSelection2 | boolean | X |  |  |  |  |
| bCmdSelection3 | boolean | X |  |  |  |  |
| bStatusSelection1 | boolean |  | X |  |  |  |
| bStatusSelection2 | boolean |  | X |  |  |  |
| bStatusSelection3 | boolean |  | X |  |  |  |
| sTexteSelection1 | string |  |  |  |  |  |
| sTexteSelection2 | string |  |  |  |  |  |
| sTexteSelection3 | string |  |  |  |  |  |
| sTexteSelectionEnCours | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

CalcsTexteSelectionEnCours

|  |  |
| --- | --- |
| Name | CalcsTexteSelectionEnCours |
| Description |  |
| Trigger | WhileTrue of Me.bStatusSelection1 + Me.bStatusSelection2 + Me.bStatusSelection3 |

**Declarations :**

Not Applicable

**Script :**

Me.sTexteSelectionEnCours="";  
if Me.bStatusSelection1 == true then  
 Me.sTexteSelectionEnCours = Me.sTexteSelectionEnCours + Me.sTexteSelection1;  
endif;  
if Me.bStatusSelection2 == true then  
 Me.sTexteSelectionEnCours = Me.sTexteSelectionEnCours + Me.sTexteSelection2;  
endif;  
if Me.bStatusSelection3 == true then  
 Me.sTexteSelectionEnCours = Me.sTexteSelectionEnCours + Me.sTexteSelection3;  
endif;

#### Template ArchestrA $dwSelecteur4

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bCmdSelection1 | boolean | X |  |  |  |  |
| bCmdSelection2 | boolean | X |  |  |  |  |
| bCmdSelection3 | boolean | X |  |  |  |  |
| bCmdSelection4 | boolean | X |  |  |  |  |
| bStatusSelection1 | boolean |  | X |  |  |  |
| bStatusSelection2 | boolean |  | X |  |  |  |
| bStatusSelection3 | boolean |  | X |  |  |  |
| bStatusSelection4 | boolean |  | X |  |  |  |
| sTexteSelection1 | string |  |  |  |  |  |
| sTexteSelection2 | string |  |  |  |  |  |
| sTexteSelection3 | string |  |  |  |  |  |
| sTexteSelection4 | string |  |  |  |  |  |
| sTexteSelectionEnCours | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

CalcsTexteSelectionEnCours

|  |  |
| --- | --- |
| Name | CalcsTexteSelectionEnCours |
| Description |  |
| Trigger | WhileTrue of Me.bStatusSelection1 + Me.bStatusSelection2 + Me.bStatusSelection3 + Me.bStatusSelection4 |

**Declarations :**

Not Applicable

**Script :**

Me.sTexteSelectionEnCours="";  
if Me.bStatusSelection1 == true then  
 Me.sTexteSelectionEnCours = Me.sTexteSelectionEnCours + Me.sTexteSelection1;  
endif;  
if Me.bStatusSelection2 == true then  
 Me.sTexteSelectionEnCours = Me.sTexteSelectionEnCours + Me.sTexteSelection2;  
endif;  
if Me.bStatusSelection3 == true then  
 Me.sTexteSelectionEnCours = Me.sTexteSelectionEnCours + Me.sTexteSelection3;  
endif;  
if Me.bStatusSelection4 == true then  
 Me.sTexteSelectionEnCours = Me.sTexteSelectionEnCours + Me.sTexteSelection4;  
endif;

#### Template ArchestrA $dwStatusBool

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| bStsFerme | boolean |  | X |  |  |  |
| bStsOuvert | boolean |  | X |  |  |  |
| sTexteFerme | string |  |  |  |  |  |
| sTexteIntermediaire | string |  |  |  |  |  |
| sTexteLesDeux | string |  |  |  |  |  |
| sTexteOuvert | string |  |  |  |  |  |
| sTexteStatusEnCours | string |  |  |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

CalcsTexteStatusEnCours

|  |  |
| --- | --- |
| Name | CalcsTexteStatusEnCours |
| Description |  |
| Trigger | DataChange of Me.bStsFerme + 2 \* Me.bStsOuvert |

**Declarations :**

Not Applicable

**Script :**

dim sts as integer;  
sts=Me.bStsFerme + 2 \* Me.bStsOuvert;  
if sts==0 then  
 Me.sTexteStatusEnCours=Me.sTexteIntermediaire;  
endif;  
if sts==1 then  
 Me.sTexteStatusEnCours=Me.sTexteFerme;  
endif;  
if sts==2 then  
 Me.sTexteStatusEnCours=Me.sTexteOuvert;  
endif;  
if sts==3 then  
 Me.sTexteStatusEnCours=Me.sTexteLesDeux;  
endif;

#### Template ArchestrA $dwTotaliseur

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

$dwTSFCObjet

##### UDA's

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UDA name | Data Type | IO | I | O | A | H |
| fMultiplicateur | float |  |  |  |  |  |
| fTotal | float |  | X |  |  |  |

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

## MCO Modules

### MCO Modules templates

#### Template ArchestrA $Sequencer

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

Not Applicable

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable

## Area

### Area templates

#### Template ArchestrA $Switch

##### Description

GalaxyToolkit.DataTransferObjects.GalaxyAttribute

##### Derived from

Not Applicable

##### UDA's

Not Applicable

##### Field Attributes

Not Applicable

##### Scripts

Not Applicable