



[Note Technique WinDesign]

Fonctions utilisables dans les critères ou les calculs

Date dernière révision	26/09/2018
Auteur	Cecima cecima@win-design.com
Versions	V0 : Initialisation du document
Document	PARA_FONCTIONS WINDESIGN_PNT_11700V00
Renvois	NC

Table des matières

1. GENERALITES	3
1.1 INTRODUCTION	3
1.2 ACCES A UNE CARACTERISTIQUE D'UN OBJET	3
1.3 ACCES A UNE CARACTERISTIQUE D'UN OBJET LIE	3
1.4 UTILISATION D'UNE FONCTION	3
2. OPERATEURS	4
2.1 INTRODUCTION	4
2.2 OPERATEURS ARITHMETIQUES	4
2.3 OPERATEURS LOGIQUES	4
2.4 OPERATEURS DE COMPARAISON	4
2.5 OPERATEURS DE CONCATENATION	4
3. FONCTIONS SUR LES CHAINES DE CARACTERES	5
3.1 FONCTION DE CONCATÉNATION	5
3.2 FONCTION MAKELOWER	5
3.3 FONCTION MAKEUPPER	5
3.4 FONCTION MID	5
3.5 FONCTION LEFT	5
3.6 FONCTION RIGHT	6
3.7 FONCTION COMPARE	6
3.8 FONCTION COMPARENOCASE	6
3.9 FONCTION BEGINWITH	6
3.10 FONCTION ENDWITH	6
3.11 FONCTION STRCONTAIN	7
4. FONCTIONS SUR LES DATES	8
4.1 FONCTION GETCURRENTDATE	8
4.2 FONCTION GETDAY	8
4.3 FONCTION GETMONTH	8
4.4 FONCTION GETYEAR	8
4.5 FONCTION GETDAYOFWEEK	8
5. FONCTIONS MATHEMATIQUES	9
5.1 FONCTION SCALE	9

1. Généralités

1.1 Introduction

Ce document permet de référencer les fonctions utilisables dans les critères et les caractéristiques étendues calculées.

1.2 Accès à une caractéristique d'un objet

Syntaxe : `§c:caractéristique§`
`§e:"caractéristique étendue"§`

Exemple :

```
§c:nom§  
§e:"Date version"§
```

1.3 Accès à une caractéristique d'un objet lié

Lorsque vous êtes sur un lien, il est possible d'accéder aux objets liés en spécifiant leur type/stéréotype ou en utilisant les mots clés object1 et object2 s'ils sont de même type.

Syntaxe : `§e:"type_objet.{stereotype.}caractéristique étendue"§`

Exemple :

```
§e:"Bloc applicatif.Date version"§  
§e:"Bloc applicatif.Application.Date version"§  
§e:"Object1.Date version"§
```

1.4 Utilisation d'une fonction

Les fonctions sont expliquées en détail dans les chapitres suivants.

Syntaxe : `§f: nom_fonction ([paramètre { , paramètre }])§`

Exemple :

```
§f:MakeLower( §c:nom§ )§
```

2. Opérateurs

2.1 Introduction

Les opérateurs sont utilisés dans les expressions et les conditions.
Le langage comprend :

- des opérateurs arithmétiques
- des opérateurs logiques
- des opérateurs de comparaison
- des opérateurs de concaténation

2.2 Opérateurs arithmétiques

+	addition
-	soustraction
*	multiplication
/	division entière

2.3 Opérateurs logiques

NOT	négation
AND	et logique
OR	ou logique

2.4 Opérateurs de comparaison

=	égal
==	égal
<>	différent
!=	différent
<	inférieur
>	supérieur
<=	inférieur ou égal
>=	supérieur ou égal

2.5 Opérateurs de concaténation

+	concatène deux chaînes de caractères
---	--------------------------------------

3. Fonctions sur les chaînes de caractères

3.1 Fonction de concaténation

Il suffit d'utiliser l'opérateur + entre deux chaînes de caractères.

Syntaxe : `expression + expression`

Exemple : `"Date" + " Version"` retourne `Date version`

3.2 Fonction MakeLower

La fonction **MakeLower** retourne le résultat de l'expression avec les caractères en minuscule.

Syntaxe : `MakeLower(expression)`

Exemple :
`$f:MakeLower($c:nom$)$`
`$f:MakeLower("Bonjour")$` retourne `bonjour`

3.3 Fonction MakeUpper

La fonction **MakeUpper** retourne le résultat de l'expression avec les caractères en majuscule.

Syntaxe : `MakeUpper(expression)`

Exemple :
`$f:MakeUpper($c:nom$)$`
`$f:MakeUpper("Bonjour")$` retourne `BONJOUR`

3.4 Fonction Mid

La fonction **Mid** extrait une chaîne de caractère de *nbCar* à partir de la position *début*.
Si *nbCar* est omis, extrait le nombre de caractères restant.

Syntaxe : `Mid(expression, début [, nbCar])`

Exemple :
`$f:Mid("abcdef", 2, 3)$` retourne `cde`
`$f:Mid("abcdef", 2)$` retourne `cdef`

3.5 Fonction Left

La fonction **Left** extrait une chaîne de caractère de *nbCar* en partant de la gauche.

Syntaxe : `Left(expression, nbCar)`

Exemple :

```
§f:Left("abcdef", 2)§ retourne ab
```

3.6 Fonction Right

La fonction **Right** extrait une chaîne de caractères de *nbCar* en partant de la droite.

Syntaxe : **Right(*expression*, nbCar)**

Exemple :

```
§f:Right("abcdef", 2)§ retourne ef
```

3.7 Fonction Compare

La fonction **Compare** compare deux chaînes de caractères sans ignorer la casse.

Retourne 0 si les chaînes sont identiques
 < 0 si *expression1* < *expression2*
 > 0 si *expression1* > *expression2*

Syntaxe : **Compare(*expression1*, *expression2*)**

Exemple :

```
§f:Compare("abc", "abc")§ retourne 0  
§f:Compare("ABC", "abc")§ retourne <0
```

3.8 Fonction CompareNoCase

La fonction **CompareNoCase** compare deux chaînes de caractères en ignorant la casse.

Retourne 0 si les chaînes sont identiques
 < 0 si *expression1* < *expression2*
 > 0 si *expression1* > *expression2*

Syntaxe : **CompareNoCase(*expression1*, *expression2*)**

Exemple :

```
§f:CompareNoCase("abc", "abc")§ retourne 0  
§f:CompareNoCase("ABC", "abc")§ retourne 0
```

3.9 Fonction BeginWith

La fonction **BeginWith** retourne TRUE si *expression* commence par *debut*.

Syntaxe : **BeginWith(*expression*, debut)**

Exemple :

```
§f:BeginWith("abcdef", "ab")§ retourne TRUE
```

3.10 Fonction EndWith

La fonction **EndWith** retourne TRUE si *expression* finit par *fin*.

Syntaxe : **EndWith**(*expression*, fin)

Exemple : `$f:EndWith("abcdef","ef")$` retourne **TRUE**

3.11 Fonction StrContain

La fonction **StrContain** retourne TRUE si contenu est dans *expression*.

Syntaxe : **StrContain**(*expression*, contenu)

Exemple : `$f:StrContain("abcdef","cd")$` retourne **TRUE**

4. Fonctions sur les dates

4.1 Fonction GetCurrentDate

Renvoie la date courante.

Syntaxe : **GetCurrentDate()**

Exemple :

`§f:GetCurrentDate()` § retourne la date actuelle

4.2 Fonction GetDay

La fonction **GetDay** retourne le jour d'une date.

Syntaxe : **GetDay(date)**

Exemple : Si la date courante est le 18/01/2015

`§f:GetDay(§f:GetCurrentDate())§` § retourne 18

4.3 Fonction GetMonth

La fonction **GetMonth** retourne le mois d'une date.

Syntaxe : **GetMonth(date)**

Exemple : Si la date courante est le 18/01/2015

`§f:GetMonth(§f:GetCurrentDate())§` § retourne 01

4.4 Fonction GetYear

La fonction **GetYear** retourne l'année d'une date.

Syntaxe : **GetYear(date)**

Exemple : Si la date courante est le 18/01/2015

`§f:GetYear(§f:GetCurrentDate())§` § retourne 2015

4.5 Fonction GetDayOfWeek

La fonction **GetDayOfWeek** retourne le jour de la semaine d'une date.

1 : dimanche, 2 : lundi, ...

Syntaxe : **GetDayOfWeek(date)**

Exemple : Si la date courante est le 18/01/2015

`§f:GetDayOfWeek(§f:GetCurrentDate())§` § retourne lundi

5. Fonctions mathématiques

5.1 Fonction Scale

Permet de changer d'échelle avec une règle de 3.

Cette fonction est utile si vous avez par exemple des nombres de 1 à 100 que vous voulez transformer en chiffre de 1 à 5 (souvent utilisé dans l'analyse des risques)

Syntaxe : **Scale**(*expression* , *nValMin* , *nValMax* , *nNouvValMin* , *nNouvValMax*)

expression : Valeur numérique à transformer

nValMin : valeur mini possible dans l'échelle actuelle (en général 0 ou 1)

nValMax : valeur maxi possible dans l'échelle actuelle

nNouvValMin : valeur mini possible dans la nouvelle échelle (en général 0 ou 1)

nNouvValMax : valeur maxi possible dans l'échelle actuelle

Exemple :

```
§f:Scale(75, 1, 100, 1, 5)§      retourne 4
```