

Business Objects DE L'UTILISATION ETENDUE DES FONCTIONS DATE

par Bruno Roman-Ruiz (Bruno2r)

Date de publication : 14/01/2008

Dernière mise à jour : 16/01/2008

Cet article passe en revue les fonctions disponibles dans B.O. pour manipuler les dates, propose des exemples de fonctions imbriquées permettant de contourner les fonctions absentes expose une méthode permettant de calculer la durée écoulée entre deux dates à l'aide des fonctions disponibles dans B.O et présente une solution aux séries discontinues d'événements datés et à l'automatisation des requêtes portant sur une période N et N-1

I - LES FONCTIONS DATE SIMPLES OU IMBRIQUEES**I-A - CONVERSIONS**

I-A-1 - Convertir une Date en texte

I-A-2 - Convertir un texte en Date

I-A-3 - Convertir une Date Heure en Numéro de jour avec décimales

I-B - INFORMATIONS

I-B-1 - Informations de position

I-B-2 - Informations littérales

I-B-3 - Dates et Heures de référence

I-C - DATES RELATIVES (au 26/09/2007)**I-D - DUREES****II - CALCUL DE LA DUREE ECOULEE EN HEURES****II-A - DUREE ECOULEE EN HEURES : UNE FONCTION ABSENTE****II-B - COMPRENDRE LE BESOIN**

II-B-1 - Tester si les dates couvrent des jours entiers

II-B-2 - Calculer les durées exprimées en secondes

II-B-2-a - Pour des dates différentes

II-B-2-b - Pour des horaires à la même date

II-C - CREATION DES VARIABLES

II-C-1 - Calculs intermédiaires

II-C-2 - Durées exprimées en secondes

II-C-3 - Variable résultat

II-C-4 - Version texte du résultat en Heures Minutes Secondes

II-C-5 - Version décimale

II-D - TESTS ET VERIFICATIONS**III - COMMENT CREER UN CALENDRIER 2008****III-A - POUR QUOI FAIRE ?****III-B - DANS B.O. REPORTER EN SQL A LA CARTE**

III-B-1 - Oracle 8

III-B-2 - Oracle 9iR2 et supérieur

III-C - DANS DESIGNER A PARTIR DE LA V6**IV - AUTOMATISER DES REQUETES PARAMETREES ANNEE N / ANNEE N-1 (V6 minimum)****IV-A - OBJECTIF DE LA METHODE****IV-B - LA TECHNIQUE UTILISEE****IV-C - LES ETAPES DE LA MISE EN OEUVRE**

IV-C-1 - Création du SQL à la carte

V - REMERCIEMENTS

Par convention typographique les variables seront affichées entre crochets [Variable]

I - LES FONCTIONS DATE SIMPLES OU IMBRIQUEES

I-A - CONVERSIONS

I-A-1 - Convertir une Date en texte

Ce type de conversion peut être très utile, notamment pour identifier les périodes mensuelles de type AnnéeMois (200712) ou encore extraire des parties de DateHeure (Heures, Minutes, Secondes) pour lesquelles B.O. ne propose pas de fonctions.

Fonction : **FormatDeDate([Date], "FormatTexte")**

Exemples avec une variable [Date] = 25/12/2007 23:59:25

```
=FormatDeDate([Date], "aaaammjj hhmmss") >>> "20071225 235925"  
=FormatDeDate([Date], "Jjjj jj Mmmm aaaa") >>> "Mardi 25 Décembre 2007"  
=FormatDeDate([Date], "aaaamm") >>> "200712"
```

I-A-2 - Convertir un texte en Date

Cette fonction permet notamment de créer une variable Date utilisable comme référence dans un calcul.

Fonction : **EnDate("Texte", "FormatTexte")**

Exemples

```
=EnDate("20071225", "aaaammjj") >>> 25/12/2007  
=EnDate("20071225 235925", "aaaammjj hhmmss") >>> 25/12/2007 23:59:25
```

Au sujet du FormatTexte :

Le format en Français utilise les indicateurs suivants :

- aaaa pour l'année
- mm pour le mois
- j pour le jour
- hh pour les heures sur 12
- HH pour les heures sur 24
- mm pour les minutes
- ss pour les secondes

Le format en Anglais utilise les indicateurs suivants :

- y pour l'année
- m pour le mois

- d pour le jour

Pour connaître la langue de référence de votre application il suffit d'utiliser la formule suivante :

```
=ValeurApplication("BOLANGUAGE")
Résultat "FR" pour Français
```

I-A-3 - Convertir une Date Heure en Numéro de jour avec décimales

Cette conversion permet d'utiliser, pour un mois donné, les Date-Heures en décimales comme Axe X d'un graphe en nuage de points.

= NuméroDuJourDuMois([DATEHEURE]) +

```
(
(
(3600 * EnNombre(FormatDeDate([DATEHEURE] , "HH")))
+ (60 * EnNombre(Droite(FormatDeDate([DATEHEURE] , "HH:mm") ,2)))
+ EnNombre(Droite(FormatDeDate([DATEHEURE] , "HH:mm:ss") ,2))
)
```

/ 86400

)

26/09/2007 21:10:30 est convertie en 26.88229167

La même conversion peut se faire en utilisant pour la partie entière NuméroDuJourDansAnnée

I-B - INFORMATIONS

I-B-1 - Informations de position

Information relative à la date du 26/09/2007	Résultat obtenu	Fonction utilisée	Type en entrée	Type en sortie
Numéro du jour dans l'année	269	NuméroDuJourDeAnnée([Date])	Date	Numérique
Numéro du jour dans le mois	26	NuméroDuJourDuMois([Date])	Date	Numérique
Numéro du jour dans la semaine	3	NuméroDuJourDeLaSemaine([Date])	Date	Numérique
	39	Semaine([Date])	Date	Numérique

Numéro de la semaine dans l'année				
Numéro du mois dans l'année	9	NuméroDuMoisDeAnnée([Date])		Numérique
Numéro de trimestre	3	Trimestre([Date])	Date	Numérique
Numéro de l'année	2007	Année([Date])	Date	Numérique

 **Le Bug de la fonction Semaine(date) dans BO :**
Semaine(EnDate("20080101", "aaaammjj")) = 53 !

Cette fonction attribue la semaine 1 à la première semaine entière de l'année !!!

En 2008 c'est donc la semaine du lundi 7 janvier au dimanche 13 janvier qui est fixée comme Semaine 1, ce qui est, bien entendu, faux.

I-B-2 - Informations littérales

Information relative à la date du 26/09/2007	Résultat obtenu	Fonction utilisée	Type en entrée	Type en sortie
Semaine / Week-end	Semaine	Si(NuméroDuJourDeLaSemaine([Date]) > 5) Alors "Week_end" Sinon "Semaine"	Date	Texte
Nom du jour de semaine	mercredi	NomDuJour([Date])	Date	Texte
<i>variante</i>	Mercredi	FormatDeDate([Date], 'Jjj')	Date	Texte
Nom du mois	septembre	Mois([Date])	Date	Texte
<i>Variante</i>	Septembre	FormatDeDate([Date], 'Mmmm')	Date	Texte

I-B-3 - Dates et Heures de référence

Date-Heure recherchée	Résultat obtenu	Fonction utilisée	Type en entrée	Type en sortie
Date actuelle	30/12/2007	DateCourante()	Aucun	Date
Heure actuelle	01:28:46	HeureCourante()	Aucun	Date
Date du document	10/12/2007	DateDuDocument()	Aucun	Date
Heure du document	16:45:37	HeureDuDocument()	Aucun	Date
Date du dernier rafraîchissement	22/12/2007 10:24:34	DateDeLaDernièreExécution("Requête 1")	Aucun	Date
Date de dernière impression	22/12/2007 10:45:17	DateDeLaDernièreImpression()	Aucun	Date

I-C - DATES RELATIVES (au 26/09/2007)

Date recherchée	Résultat obtenu	Fonction utilisée	Type en entrée	Type en sortie
Date du dernier jour de la semaine	30/09/2007	DernierJourDeLaSemaine([Date])	Date	Date
Date du premier jour de la semaine suivante	01/10/2007	DateRelative(DernierJourDeLaSemaine([Date], + 1))	Date	Date
Date du premier jour de la semaine	24/09/2007	DateRelative([Date], - NuméroDuJourDeLaSemaine([Date]) + 1)	Date	Date
Date du dernier jour du mois	30/09/2007	DernierJourDuMois([Date])	Date	Date
Date du premier jour du mois	01/09/2007	DateRelative([Date], - NuméroDuJourDuMois([Date]) + 1)	Date	Date
Date du dernier jour du mois précédent	31/08/2007	DateRelative([Date], - NuméroDuJourDuMois([Date]))	Date	Date
Date du premier jour du mois précédent	01/08/2007	DateRelative(DateRelative([Date], - NuméroDuJourDuMois([Date])) , - NuméroDuJourDuMois(DateRelative([Date], - NuméroDuJourDuMois([Date])))+ 1)	Date	Date
Date du premier jour du mois suivant	01/10/2007	DateRelative(DernierJourDuMois([Date], + 1)	Date	Date
Date du dernier jour	31/10/2007	DernierJourDuMois(DateRelative(DernierJourDuMois([Date], + 1))	Date	Date

du mois suivant				
-----------------	--	--	--	--

I-D - DUREES

Information de durée	Résultat obtenu	Fonction utilisée	Type en entrée	Type en sortie
Nombre de jours séparant 2 variables date	268	JoursEntre([Date1] , [Date])	DateDébut DateFin	Numérique
Nombre de jours séparant une date saisie et une variable date	268	JoursEntre(EnDate("2010-01-01" , "aaaammjj") , [Date])	DateDébut DateFin	Numérique
Nombre de mois entre une variable date et aujourd'hui	2	MoisEntre([Date] , DateCourante())	DateDébut DateFin	Numérique

II - CALCUL DE LA DUREE ECOULEE EN HEURES

II-A - DUREE ECOULEE EN HEURES : UNE FONCTION ABSENTE

La difficulté réside dans la liste limitée de fonctions Date disponibles.

De plus la fonction `JoursEntre({Date 1} ,{Date 2})` ne renvoie que des jours entiers.

II-B - COMPRENDRE LE BESOIN

La méthode exposée ci dessous se propose de calculer la durée écoulée en secondes entre des `date_heures_minutes_secondes (datetime)` correspondant aux situations suivantes :

- le même jour du 13/12/2007 08:30 au 13/12/2007 12:00
- une date et son lendemain : du 12/12/2007 08:30 au 13/12/2007 12:00
- dates espacées de plusieurs jours : du 10/12/2007 08:30 au 13/12/2007 12:00

II-B-1 - Tester si les dates couvrent des jours entiers

Test d'amplitude

```
[JOURS_ENTIERS_ENTRE] = Nb de jours entiers séparant les deux dates
```

II-B-2 - Calculer les durées exprimées en secondes

II-B-2-a - Pour des dates différentes

Dates différentes

```
[PREMIER_JOUR] = 86400 secondes d'une journée de 24h - horaire exprimé en secondes
```

```
[JOURS_ENTRE] = Nb de jours entiers x 24h x 3600 secondes
```

```
[DERNIER_JOUR] = horaire exprimé en secondes
```

II-B-2-b - Pour des horaires à la même date

Même date

```
[DUREE_MEME_JOUR] = horaire de fin - horaire de début exprimé en secondes
```

II-C - CREATION DES VARIABLES

II-C-1 - Calculs intermédiaires

5 variables intermédiaires sont nécessaires

```

[JOURS_ENTIERS_ENTRE] = JoursEntre([DDEB] ,[DFIN]) -1

[PREMIER_JOUR] = 86400 - ((EnNombre(FormatDeDate([DDEB] ,"HH"))* 3600)
+ (EnNombre(Droite(FormatDeDate([DDEB] ,"hhmm") ,2))* 60)
+ EnNombre(Droite(FormatDeDate([DDEB] ,"hhmmss") ,2)) )

[JOURS_ENTRE]
= Si ([JOURS_ENTIERS_ENTRE] > 0) Alors ([JOURS_ENTIERS_ENTRE] * 24 * 3600 ) Sinon 0

[DERNIER_JOUR]
=(EnNombre(FormatDeDate([DFIN] ,"HH"))* 3600)
+ (EnNombre(Droite(FormatDeDate([DFIN] ,"hhmm") ,2))* 60)
+ EnNombre(Droite(FormatDeDate([DFIN] ,"hhmmss") ,2))

[DUREE_MEME_JOUR]
=
(
(EnNombre(FormatDeDate([DFIN] ,"hh"))* 3600)
+ (EnNombre(Droite(FormatDeDate([DFIN] ,"hhmm") ,2))* 60)
+ EnNombre(Droite(FormatDeDate([DFIN] ,"hhmmss") ,2))
)-(
(EnNombre(FormatDeDate([DDEB] ,"hh"))* 3600)
+ (EnNombre(Droite(FormatDeDate([DDEB] ,"hhmm") ,2))* 60)
+ EnNombre(Droite(FormatDeDate([DDEB] ,"hhmmss") ,2))
)
    
```

II-C-2 - Durées exprimées en secondes

- soit [DUREE_MEME_JOUR]
- soit [PREMIER_JOUR] + [JOURS_ENTRE] + [DERNIER_JOUR]

II-C-3 - Variable résultat

```

[DUREE_EN_SECONDES]
= Si ([JOURS_ENTIERS_ENTRE] = -1) Alors [DUREE_MEME_JOUR] Sinon [PREMIER_JOUR] + [JOURS_ENTRE] +
[DERNIER_JOUR]
    
```

II-C-4 - Version texte du résultat en Heures Minutes Secondes

Pour écrire une version texte du résultat en Heures Minutes Secondes :

Utiliser la fonction de **bastoonet**

Variable texte

```

=Tronque(Somme([DUREE_EN_SECONDES])/3600, 0) & " h "
& Tronque(Mod(Somme([DUREE_EN_SECONDES]), 3600)/60, 0) & " m "
& Mod(Somme([DUREE_EN_SECONDES]), 60) & " s"
    
```

II-C-5 - Version décimale

Pour exprimer la durée en heures dans sa version décimale :

Conversion en Heures et décimales

```
[DUREE_HEURES_DECIMALES]
=Tronque( Somme([DUREE_EN_SECONDES]) / 3600 , 0) + (Mod(Somme([DUREE_EN_SECONDES]) , 3600) / 3600)
```

II-D - TESTS ET VERIFICATIONS

Le SQL suivant nous fournira des exemples pour tester nos variables.

les DATETIME sont simplistes pour pouvoir vérifier mentalement.

```
select 'TEST1' as TEST, to_date('20071210 08:30','yyyymmdd hh24:mi') as DDEB, to_date('20071213
12:00','yyyymmdd hh24:mi') as DDFIN
from dual
UNION
select 'TEST2', to_date('20071212 08:30','yyyymmdd hh24:mi'), to_date('20071213 12:00','yyyymmdd
hh24:mi')
from dual
UNION
select 'TEST3', to_date('20071213 08:30','yyyymmdd hh24:mi') , to_date('20071213 12:00','yyyymmdd
hh24:mi')
from dual
```

VARIABLE	TEST1	TEST2	TEST3
DDEB	10/12/2007 08:30	12/12/2007 08:30	13/12/2007 08:30
DDFIN	13/12/2007 12:00	13/12/2007 12:00	13/12/2007 12:00
JOURS_ENTIERS_ENTRE (Variable test)	2	0	-1
PREMIER_JOUR	55 800	55 800	55 800
JOURS_ENTRE	172 800	0	0
DERNIER_JOUR	43 200	43 200	43 200
DUREE_MEME_JOUR	12 600	12 600	12 600
DUREE_EN_SECONDES	271 800	99 000	12 600

III - COMMENT CREER UN CALENDRIER 2008

III-A - POUR QUOI FAIRE ?

Un problème souvent rencontré dans les requêtes B.O. ramenant des événements datés est l'obtention de séries discontinues limitées aux seules dates correspondant aux événements.

Ainsi, le suivi des événements ayant eu lieu aux dates suivantes 03/01/2008, 06/01/2008, 07/01/2008, 08/01/2008 ne permet pas d'afficher les dates 04/01/2008 et 05/01/2008 en l'absence d'une table calendrier dans l'univers.

Néanmoins le recours au SQL à la carte procure une solution.

III-B - DANS B.O. REPORTER EN SQL A LA CARTE

III-B-1 - Oracle 8

```
SELECT to_date('20080101','yyyymmdd') + rownum - 1 as JOURID
FROM dual
WHERE
to_date('20080101','yyyymmdd') + (rownum - 1) <= to_date('20081231','yyyymmdd');
```

III-B-2 - Oracle 9iR2 et supérieur

```
SELECT to_date('20080101','yyyymmdd') + (rownum - 1)
FROM dual
connect BY to_date('20080101','yyyymmdd') + (rownum - 1) <= to_date('20081231','yyyymmdd');
```

III-C - DANS DESIGNER A PARTIR DE LA V6

La version 6 permet au Designer de créer des Tables Dérivées qui sont équivalentes à des vues Oracle, c'est à dire définies à l'aide de SQL s'appuyant sur les tables existantes.

Dans le cas qui nous concerne, la table dérivée sera construite de toutes pièces en pointant sur DUAL.

Procéder de la façon suivante :

- Créer une Table Dérivée,
- Créer les objets Dimension correspondants
- Etablir entre ce champ et le champ Date de la table de données une jointure externe.

```
SELECT
to_date('20080101','yyyymmdd') + (rownum - 1) as JOURID,
to_char(to_date('20080101','yyyymmdd') + (rownum - 1), 'yyyymm') as ANNEEMOIS,
to_char(to_date('20080101','yyyymmdd') + (rownum - 1), 'yyyy') as ANNEE
FROM dual
connect BY to_date('20080101','yyyymmdd') + (rownum - 1) <= to_date('20081231','yyyymmdd') as
CALENDAR;
```

Dans une Classe Calendrier, créer les objets Dimensions suivants :

- ANNEE
- ANNEEMOIS
- JOURID

En établissant des jointures externes entre le champ Date de chaque table de données et le champ JOURID de la table dérivée ces objets Dimensions permettront à l'utilisateur de rapatrier les données dans des suites ordonnées et continues donc exhaustives de dates provenant de la table dérivée CALENDAR et les données correspondantes lorsqu'elles existent.

```
CALENDAR.JOURID = TABLEDONNEES.LADATE (+)
```

IV - AUTOMATISER DES REQUETES PARAMETREES ANNEE N / ANNEE N-1 (V6 minimum)

IV-A - OBJECTIF DE LA METHODE

L'analyse des données de l'exercice en cours et la confrontation avec celles de l'année précédente est une pratique commune du reporting. Le Designer qui conçoit l'univers destiné aux utilisateurs a la possibilité de prévoir des filtres le permettant. Pourtant nombre d'univers en sont dépourvus.

La méthode exposée ici permet de pallier cette lacune en paramétrant la requête de façon à ce qu'elle se mette à jour automatiquement. En cela elle est préférable au recours à l'Invite qui, en imposant l'intervention de l'utilisateur avant exécution de la requête, rend impossible toute automatisation.

IV-B - LA TECHNIQUE UTILISEE

La version 6 de B.O. a apporté une fonctionnalité très utile dans la mise en place des requêtes : l'opérande "Sélectionner les résultats de la requête".

Il est donc possible de créer une requête en SQL à la carte, produisant deux valeurs Années N et N-1 calculées à partir de la date courante, puis, de se servir de ces valeurs de référence dans les conditions de la requête principale.

IV-C - LES ETAPES DE LA MISE EN OEUVRE

- 1) Créer un SQL à la carte produisant ANNEEREF et ANNEEANT
- 2) Utiliser ces références dans les conditions de la requête principale
- 3) Poser la condition Objet [CALENDRIER\ANNEE] Egal à résultats de la requête('Requête 2.ANNEEREF')
- 4) Poser la condition Objet [ANNEE] Egal à résultats de la requête('Requête 2.ANNEEANT')
- 5) Articuler ces deux conditions par OU

IV-C-1 - Création du SQL à la carte

Le SQL sera différent selon le type de donnée ramenée par l'objet Dimension Année.

Il devra, bien entendu être adapté selon la base de données, l'exemple ci-dessous supposant une base Oracle.

ANNEEREF ET ANNEEANT de type texte

```
SELECT
to_char( sysdate, 'yyyy') as ANNEEREF,
to_char(extract( YEAR from sysdate) - 1 , '0000') as ANNEEANT
FROM DUAL
```

ANNEEREF ET ANNEEANT de type numérique

ANNEEREF ET ANNEEANT de type numérique

```
SELECT
extract( YEAR from sysdate) as ANNEEREF,
extract( YEAR from sysdate) - 1 as ANNEEANT
FROM DUAL
```

V - REMERCIEMENTS

Merci à **LineLe** pour sa Qualité de lecture incomparable.

Merci à **bastoonet** pour son avis éclairé, à **Adrien Artero** et **LOPEZ**, pour leurs corrections.

Merci au **Forum Business Objects** qui nous permet de progresser ensemble.

Téléchargez la version pdf

