

Comment utiliser plusieurs requêtes pour un même état

par Jean BALLAT ([Espace perso de jeannot45](#))

Date de publication : 23/12/2007

Dernière mise à jour :

Cette article présente les possibilités qui s'offrent à nous
pour utiliser un même état avec plusieurs requêtes

I - Introduction

II - 1er cas : Filtrer la source au clic du bouton.

Présentation de l'exemple

II-B - De quoi avons nous besoin ?

II-B-1 - Les tables

II-B-2 - Le Formulaire

II-B-3 - Un état

II-C - Description des objets

II-C-1 - L'état

II-C-2 - Le formulaire

II-C-2-a - La liste déroulante

II-C-2-a-i - Créer une liste de valeurs

II-C-2-a-ii - A partir d'une table de la Base de données

II-C-2-a-iii - Avec une requête UNION

II-C-2-b - Le Bouton de commande

III - 2nd cas : Changer la source au clic du bouton.

III-A - Présentation de l'exemple

III-B - De quoi avons nous besoin ?

III-B-1 - Les tables

III-B-2 - Le formulaire

III-B-3 - L'état

III-C - Description des objets

III-C-1 - Le Formulaire

III-C-1-a - Création et Paramétrages par défaut

III-C-1-b - Initialisation du formulaire

III-C-1-c - Utilisation du formulaire

III-C-1-d - Le formulaire réduit

III-C-1-d-i - La fréquence quotidienne

III-C-1-d-ii - La fréquence hebdomadaire

III-C-1-d-iii - La fréquence mensuelle

III-C-1-e - L'état

III-C-1-e-1 - Description de l'état

III-C-1-e-2 - Ouverture de l'état

III-C-1-e-3 - Lancer l'impression

III-C-1-f - Conclusion sur le formulaire réduit

III-C-1-g - Le formulaire étendu

III-C-1-g-i - Activation du mode étendu

III-C-1-g-ii - Le Contrôle Calendrier

III-C-1-g-ii-1 - Implantation du contrôle

III-C-1-g-ii-2 - Paramétrage du contrôle

III-C-1-g-iii - Lancer l'impression

III-C-1-i - Conclusion sur le mode étendu

IV - 3ème Cas : Filtrer la source de l'état et changer la source des différents contrôles de la section Détail.

IV-A - Présentation de l'exemple

IV-B - De quoi avons nous besoin ?

IV-B-1 - Les tables

IV-B-2 - Le formulaire

IV-B-3 - Un état

IV-B-4 - Les requêtes

IV-C - Lancer l'impression

IV-C-1 - La section Entête d'Etat

IV-C-2 - La section Détail

IV-C-2-a - Affectation de la source aux Contrôles

IV-C-2-b - Formatage de la Section Détail

IV-D - Conclusion sur l'affectation de la source à l'état et aux contrôles
IV-E - Bibliographie
IV-F - Remerciements

I - Introduction

Par l'intermédiaire de l'étude de 3 cas, nous allons décrire comment alimenter un même état avec différentes sources.

Cet article s'adresse à toute personne désirent alléger les objets de sa base ou rencontrant des difficultés de paramétrages.

II - 1er cas : Filtrer la source au clic du bouton.

Présentation de l'exemple

Une entreprise de fabrication de fauteuils et canapés souhaite imprimer le planning de fabrication des différents ateliers.

II-B - De quoi avons nous besoin ?

II-B-1 - Les tables

La table **T Articles** qui contiendra toutes les informations sur l'article. Pour notre exemple nous n'aurons besoin que du CodeArticle (Clé Primaire) et de la désignation de l'article.

La table **T Ateliers** qui contiendra deux informations : le CodeAtelier (Clé Primaire) et le LibelAtelier.

La table **T Productions** qui contiendra les informations de la fabrication. A remarquer que la table ne contient aucune clé primaire. En effet le même n° OF sera repris pour le même article dans un atelier différent.

Les tables seront jointes par les relations ci-dessous :

II-B-2 - Le Formulaire

Il aura pour rôle de permettre à l'utilisateur de sélectionner un atelier et en cliquant sur le bouton " Imprimer le Planning " de lancer l'impression.

II-B-3 - Un état

Qui sera l'objet de notre étude.

II-C - Description des objets

II-C-1 - L'état

L'état sera bâti sur une requête dont la structure est montrée ci-dessous et la structure ci-contre :

II-C-2 - Le formulaire

Il s'agit en fait d'un formulaire indépendant, c'est-à-dire qu'aucune source ne lui est attachée.

On déposera sur celui-ci une liste déroulante indépendante et un bouton de commande.

II-C-2-a - La liste déroulante

Pour alimenter notre liste déroulante nous disposerons de 3 solutions :

Source de la liste déroulante (ComboBox)

- 1 Créer une liste de valeurs
- 2 A partir de la table T_Ateliers
- 3 A partir d'une requête UNION

II-C-2-a-i - Créer une liste de valeurs

(Cela suppose que nous ne disposons pas de table T_Ateliers)

soit en suivant les étapes de l'assistant.

soit en modifiant les valeurs des propriétés de la liste (voir ci-dessous).

On remarquera que dans cette méthode on interviendra dans 3 onglets :

Autres : pour nommer la liste ... cboAtelier (cbo pour ComboBox).

Format :

Le nombre de colonnes est 2 (une colonne représentant le Code et l'autre le LibelAtelier).

La largeur des colonnes : 0 cm ; 2.54 cm (0 pour masquer la première colonne).

Données :


Origine Source indique que nous ferons appel à une liste de valeurs.

Contenu : détaille les valeurs qui seront reprises dans la liste (le séparateur est le ; et le texte est à mettre en **double quotes**).

Colonne liée : indique la position de la colonne contenant l'info à récupérer.

Pour accéder à la fenêtre des "Propriétés" : Cliquez droit sur l'objet puis cliquez "Propriétés".

II-C-2-a-ii - A partir d'une table de la Base de données

 On procédera par l'assistant liste déroulante afin de générer l'objet. On n'oubliera pas de nommer notre liste **cboAtelier**.

Le nom de l'objet se définit dans la fenêtre des propriétés sur l'onglet "Autres".

II-C-2-a-iii - Avec une requête UNION

Par cette méthode on pourra intégrer la mention "TOUT" en entête de liste.

Après avoir créé la liste comme précédemment :

- Appelez les propriétés de la liste par clic droit sur l'objet.
- Cliquez dans l'onglet **Données**.
- Cliquez dans la ligne **Contenu** et modifiez la syntaxe comme précisé dans l'image de droite.

 Pour plus d'informations sur le langage **SQL**, [allez voir là](#).

II-C-2-b - Le Bouton de commande

Nous utiliserons le contrôle "Bouton" de la barre d'outils.

En fait l'impression se lancera sur le Clic de ce bouton. Il faut donc écrire une petite procédure événementielle.

Rappel : Affichez la fenêtre des **Propriétés** et dans l'onglet **Evènements**, cliquez dans l'évènement **sur clic**

```
Private Sub btnImprimerAtelier_Click()  
    'Déclaration d'une variable locale  
    Dim intCompteur As Integer  
  
    ' Test la valeur choisie dans la liste du formulaire  
    ' Si celle-ci est > 0 cela signifie qu'on a choisi un atelier précis  
    If cboAtelier > 0 Then  
        DoCmd.OpenReport "E_ImprimerPlanning", , , "CodeAtelier = " & cboAtelier  
    Else  
        ' Sinon on boucle pour imprimer l'ensemble des Ateliers en incrémentant un compteur de  
        boucle  
        For intCompteur = 1 To 7  
            DoCmd.OpenReport "E_ImprimerPlanning", , , "CodeAtelier = " & intCompteur  
        Next  
    End If  
End Sub
```

III - 2nd cas : Changer la source au clic du bouton.

III-A - Présentation de l'exemple

Une entreprise de fabrication de fauteuils et canapés souhaite imprimer les statistiques de production des différents ateliers (Par jour / par semaine / par mois).

III-B - De quoi avons nous besoin ?

III-B-1 - Les tables

La table **T Performances** qui contiendra toutes les informations sur les résultats de l'atelier (Quantité prévue, Quantité réalisée, date de production, mois de production, année de production).

La table **T Ateliers** qui contiendra deux informations : le CodeAtelier (Clé Primaire) et le LibeAtelier

Les tables seront reliées par les relations ci-dessous :

III-B-2 - Le formulaire

Il aura pour rôle de permettre à l'utilisateur de sélectionner une période. Cette période pourra, par défaut, être la période en cours ou par l'intermédiaire d'un bouton "Définir l'étendue ..." une période précisée par l'utilisateur grâce à un contrôle Calendrier. Enfin, en cliquant sur le bouton " Imprimer le Planning " l'utilisateur lancera l'impression.

III-B-3 - L'état

Qui sera l'objet de notre étude. Cet état dont la base est identique mais qui s'affichera sous trois aspects en fonction de la périodicité demandée.

III-C - Description des objets

III-C-1 - Le Formulaire

III-C-1-a - Création et Paramétrages par défaut

Il sera bâti sur un formulaire indépendant et sera la plaque tournante de notre process.

Etudions tout d'abord la structure de notre formulaire de choix de la périodicité.

On remarquera que le Pied de formulaire contient le Contrôle Calendrier ainsi que les champs correspondant aux choix des dates. Dans l'image de droite on retrouvera les différentes propriétés par défaut du formulaire.

*

Afin de pouvoir manipuler confortablement les infos contenues dans le formulaire, il est fortement conseillé de nommer **CHAQUE** objet utilisé.

Ainsi on trouvera :

1 - grpFrequence / 2 - lblQuotidien / 2 - lblHebdo / 2 - lblMensuel (*Chaque étiquette du grpFrequence*)

3 - btnEtendu / 4 - btnImprimer / 5 - objCalendrier / 6 - txtDebutPeriode / 7 - txtFinPeriode

III-C-1-b - Initialisation du formulaire

Imaginons ce que doit faire l'utilisateur lors de la demande d'impression des statistiques.

Celui-ci appelle le formulaire en cliquant sur un bouton du formulaire principal dédié à cet effet.

On doit donc gérer l'apparence du formulaire à l'ouverture de celui-ci. On fera donc déclencher une procédure événementielle à l'**Ouverture** du formulaire.

Pour créer cette procédure on affichera la fenêtre des propriétés et dans l'onglet "**Evènements**" on recherchera l'évènement : "**Sur Ouverture**"

Dans la fenêtre VBE nous écrivons la procédure suivante :

```
Private Sub Form_Open(Cancel As Integer)
    ' Initialisation du formulaire
    ' Rend le pied de formulaire invisible
    Me.Section(acFooter).Visible = False
    ' Initialise les textes dans les différentes options du groupe grpFrequence
    lblquotidien.Caption = "Tous les jours du mois en cours"
    lblHebdo.Caption = "Toutes les semaines du mois en cours"
    lblMensuel.Caption = "Tous les mois de l'année en cours"
    ' Initialise le texte du bouton
    btnEtendue.Caption = "Définir l'étendue ..."
    ' Initialisation de variables utilisées dans le process
    ' Ces variables ont été déclarées PUBLIC dans un module spécifique
    ' (c'est-à-dire reconnue dans l'ensemble du projet VBA)
    ' Remarque : le préfixe des variables permet de reconnaître à la lecture le type d'info
    stockée)
    boolEtendue = False
    strIntituleMois = ""
    intAnneeTraitee = Year(Date)
    ' Initialise le contrôle calendrier à la date du jour
    objCalendrier.Value = Date
    ' Exécute la commande "Ajuster à la taille du formulaire" du menu "Fenêtre"
    ' de la barre de menus "Formulaire" - Voir la remarque ci-dessous
    DoCmd.DoMenuItem acFormBar, 7, 6, 0, acMenuVer70
End Sub
```

 Cette syntaxe a été remplacée à partir de Access 97 par **RunCommand** avec l'utilisation d'une constante intrinsèque.

Voir la syntaxe ci-dessous :

```
DoCmd.RunCommand acCmdSizeToFitForm
```

III-C-1-c - Utilisation du formulaire

Lorsque l'utilisateur ouvre le formulaire deux situations peuvent se présenter :

- 1) Impression sur la date en cours (Formulaire en mode réduit)
- 2) Impression par rapport à une période précisée (Formulaire en mode étendu)

III-C-1-d - Le formulaire réduit

L'utilisateur peut donc choisir 3 périodes différentes.

De chacune d'entre-elles va découler l'extraction de données dépendantes de la fréquence choisie;

Ce même processus sera repris pour chaque situation sélectionnée;

III-C-1-d-i - La fréquence quotidienne

Formulaire	Requête associée
	Public Const cstPerformances As String = "SELECT R_StatProductionPrevue.DateProduction As DProduct, " & _ "R_StatProductionPrevue.Coupe AS CpP, " & _ "R_StatProductionRealisee.Coupe AS CpR, " & _ "[CpR]/[CpP] AS PerfCoupe, " & _ "R_StatProductionPrevue.Couture AS CtP, & _ "R_StatProductionRealisee.Couture AS CtR, " & _ "[CtR]/[CtP] AS PerfCt, " & _

```
"R_StatProductionPrevue.Bois AS BP, & _  
  
"R_StatProductionRealisee.Bois AS BR,  
" & _  
  
"[BR]/[BP] AS PerfB, " & _  
  
"R_StatProductionPrevue.Mousse AS MP,  
& _  
  
"R_StatProductionRealisee.Mousse AS  
MR, " & _  
  
"[MR]/[MP] AS PerfM, " & _  
  
"R_StatProductionPrevue.Assise AS AP,  
& _  
  
"R_StatProductionRealisee.Assise AS AR,  
" & _  
  
"[AP]/[AR] AS PerfA, " & _  
  
"R_StatProductionPrevue.Vernis AS VP,  
& _  
  
"R_StatProductionRealisee.Vernis AS VR,  
" & _  
  
"[VR]/[VP] AS PerfV, " & _  
  
"R_StatProductionPrevue.Enhoussage AS  
EP, & _  
  
"R_StatProductionRealisee.Enhoussage  
AS ER, " & _  
  
"[ER]/[EP] AS PerfE, " & _  
  
"R_StatProductionRealisee.MoisProduction,  
& _  
  
"R_StatProductionRealisee.AnneeProduction  
" & _  
  
"FROM R_StatProductionPrevue INNER  
JOIN R_StatProductionRealisee " & _  
  
"ON  
(R_StatProductionPrevue.AnneeProduction
```

	<pre> = R_StatProductionRealisee.AnneeProduction) " & _ "AND (R_StatProductionPrevue.MoisProduction = R_StatProductionRealisee.MoisProduction) " & _ "AND (R_StatProductionPrevue.DateProduction = R_StatProductionRealisee.DateProduction) " </pre>
--	--

 On remarquera deux points :

- La séquence SQL est stockée dans une constante Public afin d'alléger le code VBA lors de l'utilisation de la requête dans la procédure.
- Que l'on fait appel à deux requêtes préalables. Celles-ci sont des requêtes " Analyse croisée ". Voir ci-dessous les structures :

III-C-1-d-ii - La fréquence hebdomadaire

Formulaire	Requête associée
	<pre> Public Const cstPerformancesHebdo As String = "SELECT DatePart(""ww"",[R_StatProductionRealisee.DateProduction]) AS DProduct, " & _ "Sum(R_StatProductionPrevue.Coupe) AS CpP, " & _ "Sum(R_StatProductionRealisee.Coupe) AS CpR, " & _ "[CpR]/[CpP] AS PerfCoupe, " & _ "Sum(R_StatProductionPrevue.Couture) AS CtP, " & _ "Sum(R_StatProductionRealisee.Couture) AS CtR, " & _ "[CtR]/[CtP] AS PerfCt, " & _ "Sum(R_StatProductionPrevue.Bois) AS BP, " & _ "Sum(R_StatProductionRealisee.Bois) AS BR, " & _ </pre>

```
"[BR]/[BP] AS PerfB, " & _

    "Sum(R_StatProductionPrevue.Mousse)
AS MP, " & _

    "Sum(R_StatProductionRealisee.Mousse)
AS MR, " & _

    "[MR]/[MP] AS PerfM, " & _

    "Sum(R_StatProductionPrevue.Assise) AS
AP, " & _

    "Sum(R_StatProductionRealisee.Assise)
AS AR, " & _

    "[AP]/[AR] AS PerfA, " & _

    "Sum(R_StatProductionPrevue.Vernis) AS
VP, " & _

    "Sum(R_StatProductionRealisee.Vernis)
AS VR, " & _

    "[VR]/[VP] AS PerfV, " & _

"Sum(R_StatProductionPrevue.Enhoussage)
AS EP, " & _

"Sum(R_StatProductionRealisee.Enhoussage)
AS ER, " & _


    "[ER]/[EP] AS PerfE " & _

    "FROM R_StatProductionPrevue INNER
JOIN R_StatProductionRealisee " & _


                                "ON
(R_StatProductionPrevue.AnneeProduction
=
R_StatProductionRealisee.AnneeProduction)
" & _

                                "AND
(R_StatProductionPrevue.MoisProduction =
R_StatProductionRealisee.MoisProduction) "
& _
```

	"AND (R_StatProductionPrevue.DateProduction = R_StatProductionRealisee.DateProduction) "
--	--

 On remarquera l'utilisation de la fonction **DatePart("ww",DateProduction)** qui permettra de calculer le n° de la semaine de la date en cours.


(Les guillemets sont doublés parce qu'ils sont utilisés à l'intérieur d'une chaîne de caractères)

 Pour plus d'informations sur les fonctions de Date **c'est par ici**.

III-C-1-d-iii - La fréquence mensuelle

Formulaire	Requête associée
	Public Const cstPerformancesMensuelles As String = "SELECT T_Mois.LibelMois AS DProduct, " & _ "Sum(R_StatProductionPrevue.Coupe) AS CpP, " & _ "Sum(R_StatProductionRealisee.Coupe) AS CpR, " & _ "[CpR]/[CpP] AS PerfCoupe, " & _ "Sum(R_StatProductionPrevue.Couture) AS CtP, " & _ "Sum(R_StatProductionRealisee.Couture) AS CtR, " & _ "[CtR]/[CtP] AS PerfCt, " & _ "Sum(R_StatProductionPrevue.Bois) AS BP, " & _ "Sum(R_StatProductionRealisee.Bois) AS BR, " & _ "[BR]/[BP] AS PerfB, " & _ "Sum(R_StatProductionPrevue.Mousse) AS MP, " & _ "Sum(R_StatProductionRealisee.Mousse) AS MR, " & _

	<pre> "[MR]/[MP] AS PerfM, " & _ "Sum(R_StatProductionPrevue.Assise) AS AP, " & _ "Sum(R_StatProductionRealisee.Assise) AS AR, " & _ "[AP]/[AR] AS PerfA, " & _ "Sum(R_StatProductionPrevue.Vernis) AS VP, " & _ "Sum(R_StatProductionRealisee.Vernis) AS VR, " & _ "[VR]/[VP] AS PerfV, " & _ "Sum(R_StatProductionPrevue.Enhoussage) AS EP, " & _ "Sum(R_StatProductionRealisee.Enhoussage) AS ER, " & _ "[ER]/[EP] AS PerfE " & _ "FROM (R_StatProductionPrevue INNER JOIN R_StatProductionRealisee " & _ "ON (R_StatProductionPrevue.DateProduction = R_StatProductionRealisee.DateProduction) " & _ "AND (R_StatProductionPrevue.MoisProduction = R_StatProductionRealisee.MoisProduction) " & _ "AND (R_StatProductionPrevue.AnneeProduction = R_StatProductionRealisee.AnneeProduction) " & _ "INNER JOIN T Mois ON R_StatProductionRealisee.MoisProduction = T_Mois.CodeMois " </pre>
--	--

 On remarquera la présence de la table **T Mois** qui permettra de récupérer l'intitulé du mois par rapport au mois courant.

III-C-1-e - L'état

III-C-1-e-1 - Description de l'état

L'état sera bâti sur une requête dont la structure est montrée ci-dessous et la structure ci-contre :

On remarquera que le contrôle contenant "Semaine" sera modifié par VBA en fonction du choix de la fréquence (Nom de l'objet : txtFrequence).

La requête sous jacente est traitée dans le code VBA en fonction de la périodicité choisie.

On verra ci-dessous la séquence SQL pour chaque périodicité concernée.

III-C-1-e-2 - Ouverture de l'état

Nous avons d'une part récupéré la séquence SQL par rapport à la période sélectionnée, nous venons de générer la Clause WHERE.

Le reste du process s'exécute à l'ouverture de l'état.

```
Private Sub Report_Open(Cancel As Integer)
' La variable octFrequence déclarée publique a été initialisée lors du choix de la fréquence
dans le formulaire
Select Case octFrequence
Case Is = 1
'Initialisation du libelle de la zone de Texte txtFrequence
txtFrequence.Caption = "Date"
' C'est donc ici que sera initialisé la source de notre état
Me.RecordSource = cstPerformances & strEtendue
Case Is = 2
txtFrequence.Caption = "Semaine"
Me.RecordSource = cstPerformancesHebdo & strEtendue
Case Is = 3
txtFrequence.Caption = "Mois"
Me.RecordSource = cstPerformancesMensuelles & strEtendue
End Select
End Sub
```

III-C-1-e-3 - Lancer l'impression

Le choix de la fréquence étant fait, l'utilisateur lancera alors l'impression en cliquant sur le bouton concerné.

Bien sûr on aura écrit une procédure sur l'évènement Clic du bouton btnImprimer (n° 4)

```
Private Sub btnImprimer_Click()
' Variable booléenne qui permet de connaître l'état du formulaire
' test si le formulaire en mode Réduit
If boolEtendue = False Then
' Initialise la variable par rapport au mois de la date en cours
strIntituleMois = IntituleMois(Date)
```

```

' Création de la clause WHERE afin de la concaténer à la constante correspondante au choix
de la fréquence
' octFrequence est une variable publique qui est initialisée lors du clic sur une des
options dans le formulaire
Select Case octFrequence
Case Is = 1
' strEtendue : Variable publique récupérant la clause WHERE générée
strEtendue = "WHERE R_StatProductionRealisee.MoisProduction = " & Month(Date) _
& " AND R_StatProductionRealisee.AnneeProduction = " & Year(Date)
Case Is = 2
' Recherche le N° de semaine du premier jour du mois en cours et
'le N° de semaine du dernier jour du mois en cours
octNumSemaineDebut = DatePart("ww", DateSerial(Year(Date), Month(Date), 1))
octNumSemaineFin = DatePart("ww", DateSerial(Year(Date), Month(Date),
Day(DateSerial(Year(Date), Month(Date) + 1, 0))))
' Création de la clause WHERE
strEtendue = "WHERE DatePart(""ww"",[R_StatProductionRealisee.DateProduction]) BETWEEN
" _
& octNumSemaineDebut & " AND " & octNumSemaineFin & _
" GROUP BY DatePart(""ww"",[R_StatProductionRealisee.DateProduction])"
Case Is = 3
intNumAnnee = Year(Date)
strEtendue = " WHERE R_StatProductionRealisee.AnneeProduction = " & intNumAnnee & _
" GROUP BY T_Mois.LibelMois,
R_StatProductionRealisee.MoisProduction " & _
" ORDER BY R_StatProductionRealisee.MoisProduction"
End Select
' dans la partie ELSE on traite de la partie concernant le formulaire étendu
Else
#.
#.
End If
' Ferme le formulaire en cours
DoCmd.Close
' Ouvre l'état des Performances en mode aperçu avant impression
DoCmd.OpenReport "E_Performances", acViewPreview
End Sub

```

III-C-1-f - Conclusion sur le formulaire réduit

- 1 En cliquant sur une fréquence : Initialisation la variable octFrequence.
- 2 En cliquant sur le bouton "Imprimer".

Récupération de la constante SQL correspondante à la fréquence choisie.

Création de la clause WHERE correspondante à la fréquence choisie.

Ouverture de l'état "E_Performances".

- 3 Sur l'ouverture de l'état

Initialisation de l'étiquette txtFrequence.

Création de la clause WHERE correspondante à la fréquence choisie.


Affectation de la nouvelle source de l'état en concaténant La constante SQL et la Variable strEtendue.

III-C-1-g - Le formulaire étendu

III-C-1-g-i - Activation du mode étendu

En cliquant sur le bouton "**Définir l'étendu**", le formulaire affiche le pied de formulaire jusque là invisible et modifie les différents affichages de périodes ainsi que le texte affiché dans le bouton qui devient "**Masquer le calendrier**". Voici ci-dessous la description de la procédure qui se trouve implantée dans l'évènement "sur Clic" du bouton "btnEtendue" (n° 3)


```
Private Sub btnEtendue_Click()
' Test sur le texte affiché sur le bouton
If Screen.ActiveControl.Caption = "Définir l'étendue ..." Then
' Rend visible le pied de formulaire
Me.Section(acFooter).Visible = True
' Change la légende du bouton et des étiquettes des options
btnEtendue.Caption = "Masquer le Calendrier"
lblquotidien.Caption = "Tous les jours dans la période choisie"
lblHebdo.Caption = "Toutes les semaines dans la période choisie"
lblMensuel.Caption = "Tous les mois dans la période choisie"
' Initialise le calendrier à la date du jour
objCalendrier.Value = Date
intAnneeTraitee = Year(Date)
objCalendrier.Requery
' initialisation Pointeur de repérage du formulaire étendu
boolEtendue = True
Else
' Repositionne les valeurs du mode Réduit
Me.Section(acFooter).Visible = False
btnEtendue.Caption = "Définir l'étendue ..."
lblquotidien.Caption = "Tous les jours du mois en cours"
lblHebdo.Caption = "Toutes les semaines du mois en cours"
lblMensuel.Caption = "Tous les mois de l'année en cours"
' Réinitialise la variable booléenne sur FAUX
boolEtendue = False
End If
' Exécute la commande "Ajuste à la taille du formulaire" du Menu "Fenêtre"
'de la barre de menus "Formulaire" - Voir Remarque
DoCmd.DoMenuItem acFormBar, 7, 6, 0, acMenuVer70
End Sub
```

 Cette syntaxe a été remplacée à partir de Access 97 par RunCommand avec l'utilisation d'une constante intrinsèque (voir ci-dessous) :

```
DoCmd.RunCommand acCmdSizeToFitForm
```

Comme pour le mode réduit on retrouvera 3 situations.

Fréquence Quotidienne	Fréquence Hebdomadaire	Fréquence Mensuelle
-----------------------	------------------------	---------------------

 On remarquera que le texte des contrôles txtDebutPeriode et txtFinPeriode sont modifiés en fonction de la fréquence. Cette initialisation se fait sur le changement de fréquence (évènement Après MAJ de grpFrequence)

On trouvera ci-dessous la description de la procédure:

```
Private Sub grpFrequence_AfterUpdate()  
    ' Intitalisation de la variable octFrequence  
    octFrequence = grpFrequence  
    ' Intitalisation des zones de saisie des limites de période  
    txtDebutPeriode = Null  
    txtFinPeriode = Null  
    Select Case octFrequence  
        Case Is = 1  
            lblDebut.Caption = "1er Jour "  
            lblFin.Caption = "Dernier Jour"  
        Case Is = 2  
            lblDebut.Caption = "1ere Semaine "  
            lblFin.Caption = "Dernière Semaine"  
        Case Is = 3  
            lblDebut.Caption = "1er Mois "  
            lblFin.Caption = "Dernier Mois"  
    End Select  
End Sub
```

III-C-1-g-ii - Le Contrôle Calendrier

III-C-1-g-ii-1- - Implantation du contrôle

En mode création sur notre formulaire.

Cliquez sur l'outil d'appel aux contrôles ActiveX supplémentaires.

Dans la liste, choisir le contrôle calendrier.

Cliquez sur le formulaire. Un calendrier de dimensions par défaut s'affiche.

III-C-1-g-ii-2- - Paramétrage du contrôle

Pour paramétrer la procédure événementielle, il faut d'abord nommer notre objet dans l'onglet "Autres". (On lui donnera le nom de "objCalendrier")

Cliquez droit sur l'objet et choisir la commande "Créer une commande"

La fenêtre VBE s'affiche.

Dans la liste en haut à droite sélectionnez l'évènement "**sur clic**"

```
Private Sub objCalendrier_Click()  
    If IsNull(txtDebutPeriode) Then  
        Select Case octFrequence  
            Case Is = 1  
                txtDebutPeriode = CDate(objCalendrier)  
                ' Récupération de la date au format utilisé dans les requêtes  
                strDateDebut = Format(CDate(objCalendrier), "mm/dd/yy")  
            Case Is = 2  
                octNumSemaineDebut = DatePart("ww", CDate(objCalendrier.Value))  
                txtDebutPeriode = octNumSemaineDebut  
            Case Is = 3
```

```

        strMoisDebut = IntituleMois(CDate(objCalendrier))
        octMoisDebut = Month(CDate(objCalendrier))
        txtDebutPeriode = strMoisDebut
    End Select
Else
    Select Case octFrequence
    Case Is = 1
        If CDate(objCalendrier) < txtDebutPeriode Then
            MsgBox "Erreur de choix" & vbCrLf & "La fin de la période indiquée est antérieure
au début de la période", vbInformation, "Statistiques des performances"
            Exit Sub
        Else
            txtFinPeriode = CDate(objCalendrier)
            strDateFin = Format(CDate(objCalendrier), "mm/dd/yy")
        End If
    Case Is = 2
        octNumSemaineFin = DatePart("ww", CDate(objCalendrier.Value))
        ' Test de controle de saisie
        If octNumSemaineFin < octNumSemaineDebut Then
            ' Si le n° de semaine fin est inférieur au n° de semaine début, contrôle sur
l'année
            If Year(txtDebutPeriode) = Year(txtFinPeriode) Then
                MsgBox "Erreur de choix" & vbCrLf & "La fin de la période indiquée est
antérieure au
                début de la période", vbInformation, "Statistiques des performances"
            Exit Sub
            End if
        Else
            txtFinPeriode = octNumSemaineFin
        End If
    Case Is = 3
        strMoisFin = IntituleMois(CDate(objCalendrier))
        octMoisFin = Month(CDate(objCalendrier))
        If octMoisFin < octMoisDebut Then
            MsgBox "Erreur de choix" & vbCrLf & "La fin de la période indiquée est antérieure
au début de la période", vbInformation, "Statistiques des performances"
            Exit Sub
        Else
            txtFinPeriode = strMoisFin
        End If
    End Select
End If
intAnneeTraitee = Year(CDate(objCalendrier))
End Sub

```

III-C-1-g-iii - Lancer l'impression

Les constantes qui seront récupérées sont les mêmes que pour le mode réduit par contre les clauses WHERE dépendent de la fréquence choisie.

Rappelons que l'initialisation de celles-ci se fait sur le clic du bouton "btnImprimer"

```

Private Sub btnImprimer_Click()
    ' Première partie du IF : gère le formulaire mode réduit
    If boolEtendue = False Then
        #.
        #.
    Else
        ' Variable publique initialisée sur le choix d'une fréquence
        Select Case octFrequence
        ' Représente la Fréquence quotidienne
        Case Is = 1

```

```

' Création de la clause WHERE
strEtendue = "WHERE R_StatProductionRealisee.DateProduction BETWEEN #" _
& format(CDate(txtDebutPeriode),"mm/dd/yy") & "# AND #" _
& format(CDate(txtFinPeriode),"mm/dd/yy") & "#"
' Représente la Fréquence Hebdomadaire
Case Is = 2
' Création de la clause WHERE
strEtendue = "WHERE DatePart("ww",[R_StatProductionRealisee.DateProduction]) BETWEEN
" _
& octNumSemaineDebut & " AND " & octNumSemaineFin _
& " GROUP BY DatePart("ww",[R_StatProductionRealisee.DateProduction])"
' Représente la Fréquence Mensuelle
Case Is = 3
intNumAnnee = intAnneeTraitee
strEtendue = " WHERE R_StatProductionRealisee.AnneeProduction = " & intNumAnnee & _
" GROUP BY T_Mois.LibelMois,
R_StatProductionRealisee.MoisProduction " & _
" HAVING R_StatProductionRealisee.MoisProduction BETWEEN " &
octMoisDebut & _
" AND " & octMoisFin & " ORDER BY R_StatProductionRealisee.MoisProduction"
End Select
End If
DoCmd.Close
' Ouvre l'état des performances.
DoCmd.OpenReport "E_Performances", acViewPreview
End Sub

```



On retrouvera la même procédure à l'ouverture de l'état que pour le cas du formulaire en Mode réduit

III-C-1-i - Conclusion sur le mode étendu

- 1 En cliquant sur "Définir l'étendue ..."
 - Le Calendrier s'affiche
- 2 En cliquant sur une fréquence
 - Initialise la variable octFrequence
 - Modifie l'affichage des étiquettes en bas du formulaire
- 3 En cliquant sur le bouton "Imprimer"
 - Récupération de la constante SQL correspondante à la fréquence choisie
 - Création de la clause WHERE correspondante à la fréquence choisie
 - Ouverture de l'état "E_Performances"
- 4 Sur l'ouverture de l'état
 - Initialisation de l'étiquette txtFrequence
 - Affectation de la nouvelle source de l'état en concaténant La constante SQL et la Variable strEtendue

IV - 3ème Cas : Filtrer la source de l'état et changer la source des différents contrôles de la section Détail.

IV-A - Présentation de l'exemple

Une entreprise de fabrication de fauteuils et canapés souhaite imprimer le planning de fabrication des différents ateliers.

IV-B - De quoi avons nous besoin ?

IV-B-1 - Les tables

La table **T_Articles** qui contiendra toutes les informations sur l'article. Pour notre exemple nous n'aurons besoin que du CodeArticle (Clé Primaire) et de la désignation de l'article.

La table **T_Ateliers** qui contiendra deux informations : le CodeAtelier (Clé Primaire) et le LibelAtelier

La table **T_Productions** qui contiendra les informations de la fabrication.

A remarquer que cette table a une structure différente de notre premier exemple. En effet, on retrouve dans la structure de celle-ci 3 champs par atelier (voir ci-dessous).

De part cette structure, la génération de l'état de planification de la fabrication sera plus compliquée.

En effet, nous devons extraire les données dans des requêtes distinctes pour les affecter à notre état unique.

IV-B-2 - Le formulaire

De même structure que pour le premier cas étudié, il aura pour rôle de permettre à l'utilisateur de sélectionner un atelier et en cliquant sur le bouton "**Imprimer le Planning**" de lancer l'impression

IV-B-3 - Un état

On remarquera que dans cette structure tous les champs sont indépendants (donc par définition relié à aucune source).

Néanmoins pour faciliter l'utilisation de ces différents contrôles, on les nommera.

Pour nommer chaque zone de texte, il faut la sélectionner et appeler la fenêtre des "Propriétés.

Dans l'onglet "Autres" saisir un nom qui sera évocateur (exemple :txtTitreEtat et **non ZoneTexte1**)

1 - txtTitreEtat / 2 - txtDatePlanning / 3 - txtNumOrdre / 4 - txtOF / 5 - txtDesignation

IV-B-4 - Les requêtes

Comme pour le premier exemple, tout le process se déroule sur le clic du bouton "btnImprimer" du formulaire.

Les requêtes seront stockées dans une variable strSqlImpression qui sera utilisée lors de l'ouverture de l'état

```

' Génération de la requête pour l'atelier coupe
    strSqlImpression = "SELECT [T_Productions].[OF], T_Articles.DesignationArticle, " _
    & [T_Productions].DateCoupe, [T_Productions].CoupeTermine " _
    & " T_Articles INNER JOIN [T_Productions].ON T_Articles.CodeArticle =
[T_Productions].CodeArticle " & _
        "WHERE [T_Productions].DateCoupe=#" & varDateImpression & "# AND
[T_Productions].CoupeTermine=No"
' Génération de la requête de l'atelier Couture pour les articles dont l'atelier COUPE est
terminé
    strSqlImpression = "SELECT [T_Productions].OF, T_Articles.DesignationArticle, " _
    & [T_Productions].DateCouture, [T_Productions].CoupeTermine " & _
        "FROM T_Articles INNER JOIN T_Productions ON T_Articles.CodeArticle = ON
[T_Productions].OF " & _
        "WHERE T_Productions.CoupeTermine=Yes AND T_Productions.DateCouture=#" &
varDateImpression _
    & "# AND T_Productions.CoutureTermine=No"
' Génération de la requete de l'atelier Bois
    strSqlImpression = "SELECT [T_Productions].OF, T_Articles.DesignationArticle, " _
    & T_Productions.DateBois, T_Productions.BoisTermine " & _
        "FROM T_Articles INNER JOIN T_Productions ON T_Articles.CodeArticle =
T_Productions.CodeArticle & _
        "WHERE T_Productions.DateBois=#" & varDateImpression & "# AND
T_Productions.BoisTermine=No"
' Génération de la requête de l'atelier Mousse pour les articles dont l'atelier BOIS est
terminé
    strSqlImpression = "SELECT [T_Productions].OF, T_Articles.DesignationArticle, " _
    & T_Productions.DateMousse, T_Productions.MousseTermine " & _
        "FROM T_Articles INNER JOIN T_Productions ON T_Articles.CodeArticle =
T_Productions.CodeArticle) " & _
        "WHERE T_Productions.BoisTermine=Yes AND T_Productions.DateMousse=#" &
varDateImpression _
    & "# AND T_Productions.MousseTermine=No AND T_Productions.ProblemeMousse=No"
    
```

Ainsi de suite pour les 7 ateliers

Ces séquences SQL seront récupérées dans la procédure sur Clic du bouton btnImprimer décrite ci-dessous

```

Private Sub cmdImprimer_Click()
' Déclaration d'une variable compteur qui permettra de retrouver le nom de l'atelier en cours
d'impression
Dim intCompteurImpression As Integer
' cboAtelier est le nom de la liste déroulante du formulaire
' Le code 8 correspond au choix "Tout"
If cboAtelier = 8 Then
    For intCompteurImpression = 1 To 7
        ' La propriété Column permet de récupérer l'info contenu dans une colonne d'une liste
        déroulante,
        ' il faut indiquer la position de cette colonne -1 parce que l'indexation de colonnes commence
        à 0
        strNomAtelier = cboAtelier.Column(intCompteurImpression - 1)
        ' Récupération des Séquences SQL définies ci-dessus en fonction du compteur de boucle
        CreationRequeteImpression (intCompteurImpression)
    Next
Else
' Un atelier spécifique a été choisi
    
```

```
strNomAtelier = cboAtelier.Column(intCompteurImpression - 1)
' Récupération de la Séquence SQL définie ci-dessus par rapport à l'atelier choisi
CreationRequeteImpression (cboAtelier)
End If
' Ouverture de l'état
DoCmd.OpenReport "E_PlanningLancementFab", acPreview
End Sub
```

IV-C - Lancer l'impression

Il nous reste donc à découvrir les procédures qui vont s'exécuter à l'ouverture de l'Etat

IV-C-1 - La section Entête d'Etat

On créera une petite procédure sur le formatage de la section entête.

On passera donc par la fenêtre des "Propriétés" puis onglet "Evènements"

```
Private Sub EntêteEtat_Format(Cancel As Integer, FormatCount As Integer)
txtTitreEtat = "PLANNING ATELIER : " & strNomAtelier
txtDatePlanning = DateTraitee
End Sub
```

IV-C-2 - La section Détail

Pour la création de l'état il nous faudra écrire deux procédures

IV-C-2-a - Affectation de la source aux Contrôles


Cette affectation se fait sur l'évènement "A l'ouverture de l'état"

On procédera comme à l'accoutumé :

- Affichez l'état en mode création
- Cliquez droit dans la section "Détail"
- Cliquez "Propriétés"
- Activez l'onglet "évènement"
- Cliquez dans l'évènement "à l'ouverture"

```
Private Sub Report_Open(Cancel As Integer)
' Déclaration des variables
Dim rstImpression As DAO.Recordset
' Initialisation du jeu d'enregistrements en fonction de la séquence récupérée
Set rstImpression = CurrentDb.OpenRecordset(strSqlImpression)
' Traitement du jeu d'enregistrements
With rstImpression
' Contrôle du contenu
If .BOF = True And .EOF = True Then
MsgBox "Il n'y a rien à imprimer sur le planning de : " & strNomAtelier
' Annulation de l'évènement Ouverture de l'état
Cancel = True
End If
End With
End Sub
```

```
Exit Sub
End If
' Affectation de la source à l'état
Reports("E_PlanningFabrication").RecordSource = strSQLImpression
' Affectation des champs de la Source aux contrôles de la zone Détail - Voir remarque
Me.txtOF.ControlSource = .Fields(1).Name
Me.txtDesignationArticle.ControlSource = .Fields(2).Name
End With
End Sub
```

 On utilise l'objet *Fields(index)* pour affecter le nom du champ aux contrôles de la section Détail.

Les requêtes étant bâties toutes sur la même structure, cela nous permet d'utiliser une procédure générale

IV-C-2-b - Formatage de la Section Détail

Afin d'assurer une meilleure présentation de l'état nous souhaitons mettre une ligne sur deux avec un fond coloré. Cette procédure est attachée à l'évènement Sur Formatage de la section détail

```
Private Sub Détail_Format(Cancel As Integer, FormatCount As Integer)
' Déclaration d'une variable statique (la valeur est préservée durant le traitement)
Static NumLigne
NumLigne = NumLigne + 1
Me.txtNumOrdre = NumLigne
' affecte un fond grisé pour les lignes paires
If NumLigne Mod 2 = 0 Then
Me.Détail.BackColor = 14869218
Else
' Affecte un fond blanc pour les lignes impaires
Me.Détail.BackColor = 16777215
End If
End Sub
```

IV-D - Conclusion sur l'affectation de la source à l'état et aux contrôles

- Après avoir choisi un atelier, en cliquant sur le bouton btnImprimer :
 - Création de la séquence SQL en fonction de l'atelier.
- Impression de l'état
 - Affectation de la source de l'état.
 - Affectation de la source aux contrôles de la zone de Détail.
 - Formatage de la Section Entête d'état.
 - Formatage de la Section Détail.

IV-E - Bibliographie

"**Access en Action**" de Ken Getz, Paul Litwin et Andy Baron aux éditions O'REILLY

Excellent ouvrage dans lequel j'ai trouvé l'exemple de l'utilisation du formulaire étendu.

IV-F - Remerciements

Dolphy35 pour son incitation à la rédaction de cet article, son aide pour la prise en main de l'outil de Nono40.

Heureux-oli pour son soutien et ses explications lors de la première mise en place dans mon domaine.

Loufab pour ses précieux conseils de présentation. **Starec**, **RideKick** pour sa relecture attentive. **SQLPro** pour sa permission d'accéder à son site.

