

1. Des rapports sur mesure (objet ODSOUT)

1.1. Principe général

L'étape DATA `_NULL_` propose depuis longtemps la construction de rapports sur mesure, à l'aide des instructions `FILE PRINT` et `PUT`. Avec l'ODS, une instruction `FILE PRINT ODS=(STYLE=template)` était apparue, à utiliser conjointement avec l'instruction `PUT _ODS_ ;` pour produire avec une étape Data des sorties tabulaires.

Cependant, la flexibilité de l'étape Data ne permettait pas de résoudre tous les problèmes de formatage complexe de sorties, en particulier les cellules fusionnées. Il a donc été ajouté¹ aux fonctionnalités de l'étape Data un objet de type ODSOUT, qui représente un rapport, et dont le remplissage, à l'aide de méthodes définissant des contenants emboîtés, est d'une grande souplesse.

La syntaxe de base est la suivante.

```
ODS LISTING CLOSE ;
ODS RTF|PDF|HTML FILE = "..." ;

DATA _NULL_ ;
  SET|MERGE ... ;
  IF _N_=1 THEN DECLARE ODSOUT nomObjet ( ) ;

  ... /* remplissage du rapport */

RUN ;

ODS RTF|PDF|HTML CLOSE ;
```

L'instruction `DECLARE` permet de créer (instancier) l'objet ODSOUT. On lui donne un nom, qui respecte les conventions habituelles de SAS (32 caractères au maximum, composé de lettres non accentuées, de chiffres mais pas en 1^{er} et du signe `_`).



Il est indispensable de fermer au préalable la destination LISTING de l'ODS, la police SAS Monospace qu'elle utilise ne supportant pas le fonctionnement de l'objet ODSOUT. En revanche, cette étape DATA `_NULL_` fonctionne avec les principales destinations ODS telles que HTML, RTF et PDF.

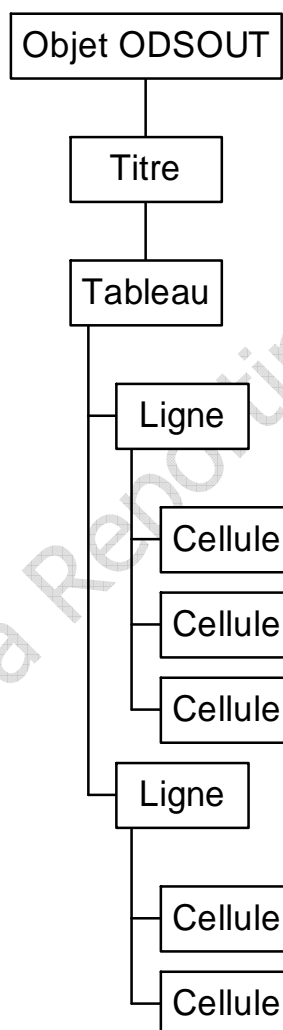
¹ de manière expérimentale en versions 9.1.3 et 9.2

1.2. Emboîtement des contenants

Les contenants que l'on décrit dans un objet ODSOUT sont tabulaires par nature. Il faut donc les penser comme tels, et repérer à l'avance où se situeront les fusions de cellules, verticalement et/ou horizontalement. Il est vivement conseillé de dessiner son tableau sur papier avant de commencer à programmer.

Leur organisation est hiérarchique, le contenant le plus élémentaire étant emboîté dans un contenant parent, et ainsi de suite (cf. Figure 1).

Figure 1 - arborescence des contenants dans l'objet ODSOUT



L'objet ODSOUT peut contenir un ou plusieurs tableaux. Ceux-ci peuvent courir sur une ou plusieurs pages (la méthode `PAGE()` permet de sauter une page à tout moment dans le document).

Chacun de ces tableaux est composé de lignes (*rows*) dont certaines peuvent être de type « en-tête » (*heading*).

Dans chaque ligne, on construit des cellules (*cells*) qui sont des contenants élémentaires.

On peut définir des titres qui s'insèrent avant le tableau (méthode TITLE) et des textes qui s'insèrent avant ou après un tableau selon son emplacement (méthode NOTE).

```
DATA _NULL_ ;
    SET|MERGE ... ;
    IF _N_=1 THEN DECLARE ODSOUT nomObjet ( ) ;

    nomObjet.TABLE_START() ; /* début du tableau */

    nomObjet.ROW_START() ; /* début d'une ligne */

    nomObjet.FORMAT_CELL(TEXT: "contenu de cellule") ;
    nomObjet.FORMAT_CELL(TEXT: "contenu de cellule") ;
    nomObjet.FORMAT_CELL(TEXT: "contenu de cellule") ;

    nomObjet.ROW_END() ; /* fin de la ligne */

    ...

    nomObjet.TABLE_END() ;

RUN ;
```

Les méthodes TABLE_START, TABLE_END, ROW_START et ROW_END servent à marquer le début et la fin des différents contenants (tableaux et lignes respectivement). La méthode FORMAT_CELL permet d'indiquer le contenu de la cellule.

Ces méthodes ont des arguments optionnels. Les principaux sont indiqués dans le Tableau 1.

Tableau 1 – Principales méthodes de gestion des contenants et leurs arguments

Méthode	Argument	Exemple de valeur	Description
TABLE_START	NAME	"RAPPORT"	Nom de l'objet ODS produit. Se retrouve dans la fenêtre RESULTS
TABLE_START	LABEL	"MON RAPPORT GENERE PAR L'ETAPE DATA"	Descriptif de l'objet ODS produit. Se retrouve dans la fenêtre RESULTS
TABLE_START	LABEL	"MON RAPPORT GENERE PAR L'ETAPE DATA"	Descriptif de l'objet ODS produit. Se retrouve dans la fenêtre RESULTS
TABLE_START	OVERRIDES	"RULES=NONE FRAME=VOID"	Altérations des attributs de style tels qu'ils sont définis pour l'ensemble de la destination ODS (style global)
ROW_START	TYPE	"HEADING"	Signale une ligne d'en-tête de tableau

			(répétée si le tableau court sur plusieurs pages)
--	--	--	---

Complément à Reporting avec SAS

Méthode	Argument	Exemple de valeur	Description
ROW_START	OVERRIDES	"BACKGROUND=YELLOW"	Altérations des attributs de style tels qu'ils sont définis pour l'ensemble de la destination ODS (style global)
ROW_START	KEEP_NEXT	2	Force la mise en forme à conserver cette ligne et les n suivantes sur la même page. Les lignes d'en-tête sont solidaires par défaut de la ligne suivante
FORMAT_CELL	TEXT	"Prix=" !! PUT(prixTTC,NUMX12.2)	Texte contenu dans la cellule. Est valide toute chaîne de caractères produite par des constantes, des fonctions caractère de SAS et des variables de type caractère
FORMAT_CELL	OVERRIDES	"FONT_WEIGHT=BOLD"	Altérations des attributs de style tels qu'ils sont définis pour l'ensemble de la destination ODS (style global)
FORMAT_CELL	SPLIT	"#"	Caractère de césure du contenu de la cellule (similaire à l'option SPLIT dans la procédure PRINT)
FORMAT_CELL	COLUMN_SPAN	2	Nombre de colonnes fusionnées que représente cette cellule
FORMAT_CELL	ROW_SPAN	3	Nombre de lignes fusionnées que représente cette cellule
FORMAT_CELL	UNDERLINE	0 ou 1	Trace une bordure en-dessous de la cellule (valeur 1) ou n'en trace pas (valeur 0)
FORMAT_CELL	OVERLINE	0 ou 1	Trace une bordure au-dessus de la cellule (valeur 1) ou n'en trace pas (valeur 0)

1.3. Fusion de cellules

La fusion de cellules se fait à l'aide des arguments COLUMN_SPAN et ROW_SPAN de la méthode FORMAT_CELL. La Figure 2 permet de mieux comprendre le fonctionnement de cette mise en forme.

Figure 2 – fusions de cellules et valeurs de COLUMN_SPAN et ROW_SPAN

❶	❷	
❸	❹	❺
❻		❼
❽		❾

Pour générer le tableau de la Figure 2, il faut écrire la syntaxe suivante :

```

nomObjet.TABLE_START() ; /* début du tableau */
  nomObjet.ROW_START() ; /* début d'une ligne */
    nomObjet.FORMAT_CELL(...) ; /* ❶ */
    nomObjet.FORMAT_CELL(..., COLUMN_SPAN : 2) ; /* ❷ */
  nomObjet.ROW_END() ; /* fin de la ligne */
nomObjet.ROW_START() ; /* début d'une ligne */
  nomObjet.FORMAT_CELL(...) ; /* ❸ */
  nomObjet.FORMAT_CELL(..., ROW_SPAN : 3) ; /* ❹ */
  nomObjet.FORMAT_CELL(...) ; /* ❺ */
nomObjet.ROW_END() ; /* fin de la ligne */
nomObjet.ROW_START() ; /* début d'une ligne */
  nomObjet.FORMAT_CELL(...) ; /* ❻ */
  nomObjet.FORMAT_CELL(...) ; /* ❼ */
nomObjet.ROW_END() ; /* fin de la ligne */
nomObjet.ROW_START() ; /* début d'une ligne */
  nomObjet.FORMAT_CELL(...) ; /* ❽ */
  nomObjet.FORMAT_CELL(...) ; /* ❾ */
nomObjet.ROW_END() ; /* fin de la ligne */
nomObjet.TABLE_END() ; /* fin du tableau */
    
```

La flexibilité de l'étape Data, qui fonctionne en boucle, permet de déclencher ces méthodes (création de lignes et de cellules) soit à chaque observation, soit en fonction de certaines conditions (avec des instructions IF). On peut ainsi construire aisément des rapports complexes.

Exemple 1 – Production d'un seul tableau avec l'objet ODSOUT

```

ODS LISTING CLOSE ;
ODS RTF FILE = "c:\odsout exemple1.doc" ;
DATA _NULL_ ;
  SET cours.produits END = fin ;
  IF _N_ = 1 THEN DO ;
    DECLARE ODSOUT rapport () ;
    
```

```
rapport.TABLE_START("listing") ;
  rapport.ROW_START(TYPE:"HEADING") ;
    rapport.FORMAT_CELL(TEXT:"Vin",
                        ROW_SPAN:2,
                        OVERRIDES:"VJUST=MIDDLE") ;
    rapport.FORMAT_CELL(TEXT:"Caractéristiques",
                        COLUMN_SPAN:3,
                        OVERRIDES:"JUST=CENTER") ;
  rapport.ROW_END() ;
  rapport.ROW_START(TYPE:"HEADING") ;
    rapport.FORMAT_CELL(TEXT:"Couleur") ;
    rapport.FORMAT_CELL(TEXT:"Millésime") ;
    rapport.FORMAT_CELL(TEXT:"Prix TTC") ;
  rapport.ROW_END() ;
END ;
rapport.ROW_START() ;
  rapport.FORMAT_CELL(TEXT:appellation,
                      SPLIT:"#") ;
  rapport.FORMAT_CELL(TEXT:couleur) ;
  rapport.FORMAT_CELL(TEXT:PUT(millesime,4.-L)) ;
  rapport.FORMAT_CELL(TEXT:PUT(prixTTC,NUMX7.2)) ;
rapport.ROW_END() ;
IF fin THEN rapport.TABLE_END() ;
RUN ;
ODS RTF CLOSE ;
```

Le programme de l'Exemple 1 produit le tableau suivant.

Vin	Caractéristiques		
	Couleur	Millésime	Prix TTC
Pessac-Léognan	blanc	1999	55,00
Alsace Grands Crus	blanc	2000	64,00
Alsace	blanc	2000	37,00
Alsace Grand Cru	blanc	2000	66,00
Alsace	blanc	2000	30,00
Alsace	blanc	2001	24,80
Alsace Grand Cru	blanc	1998	41,00

On peut également jouer sur la fusion de cellules sur plusieurs lignes, à condition de savoir à l'avance combien de lignes on devra fusionner. Comme dans l'Exemple 2, la procédure SQL peut aider à intégrer simplement cette information aux données à afficher.

Exemple 2 – Production de plusieurs tableaux et cellules fusionnées sur plusieurs lignes avec l'objet ODSOUT

```
ODS LISTING CLOSE ;
PROC SQL ;
/* préparation des données :
   ajout du nb de lignes / groupe */
CREATE TABLE work.produits AS
  SELECT *,
         COUNT(*) AS nb_lignes
  FROM cours.produits
  WHERE region = "Bourgogne"
  GROUP BY couleur,
         millesime
  ORDER BY couleur,
         millesime
  ;
QUIT ;
/* création du rapport */
ODS RTF FILE = "c:\odsout exemple2.doc" ;
DATA _NULL_ ;
  SET work.produits ;
  BY couleur millesime ;
  IF _N_ = 1 THEN DECLARE ODSOUT rapport ( ) ;
  IF FIRST.couleur THEN DO ;
    rapport.TABLE_START() ;
    rapport.ROW_START(TYPE:"HEADING") ;
    rapport.FORMAT_CELL(TEXT:"Vin "!!couleur,
                        COLUMN_SPAN:3,
                        OVERRIDES:"JUST=CENTER") ;

    rapport.ROW_END() ;
    rapport.ROW_START(TYPE:"HEADING") ;
    rapport.FORMAT_CELL(TEXT:"Appellation") ;
    rapport.FORMAT_CELL(TEXT:"Millésime") ;
    rapport.FORMAT_CELL(TEXT:"Prix TTC") ;
    rapport.ROW_END() ;
  END ;
  rapport.ROW_START() ;
  rapport.FORMAT_CELL(TEXT:appellation, SPLIT:"#") ;
  IF FIRST.millesime THEN DO ;
    rapport.FORMAT_CELL(TEXT:PUT(millesime,4.),
                        ROW_SPAN:nb_lignes,
                        OVERRIDES:"VJUST=TOP") ;

  END ;
  rapport.FORMAT_CELL(TEXT:PUT(prixTTC,NUMX7.2)) ;
  rapport.ROW_END() ;
  IF LAST.couleur THEN rapport.TABLE_END() ;
RUN ;
```

ODS RTF CLOSE ;

Vin blanc		
Appellation	Millésime	Prix TTC
Puligny-Montrachet 1 Er Cru	.	49,04
Puligny-Montrachet 1er Cru		29,90
Chablis Premier Cru		15,79
Chablis Premier Cru		17,82
Chassagne-Montrachet Premier Cru	1996	54,00
Meursault	1999	32,80
Puligny-Montrachet		30,20

Vin rouge		
Appellation	Millésime	Prix TTC
Volnay Premier Cru	.	29,42
Monthelie Premier Cru		14,11
Gevrey-Chambertin Premier Cru		55,02
Vosne-Romanée	1997	32,80
Chambolle-Musigny		32,80
Savigny-Les-Beaune Premier Cru		24,30
Pernand-Vergelesses Premier Cru		24,30
Pommard Premier Cru		45,00
La Romanée		305,00

1.4. Insertion d'éléments spéciaux

Des méthodes supplémentaires permettent d'insérer des éléments supplémentaires : titres au-dessus des tableaux, notes (textes hors tableaux), images ou liens hypertexte. Elles sont répertoriées dans le Tableau 2 avec leurs principaux arguments.

Tableau 2 – Principales méthodes d'insertion d'éléments spéciaux et leurs arguments

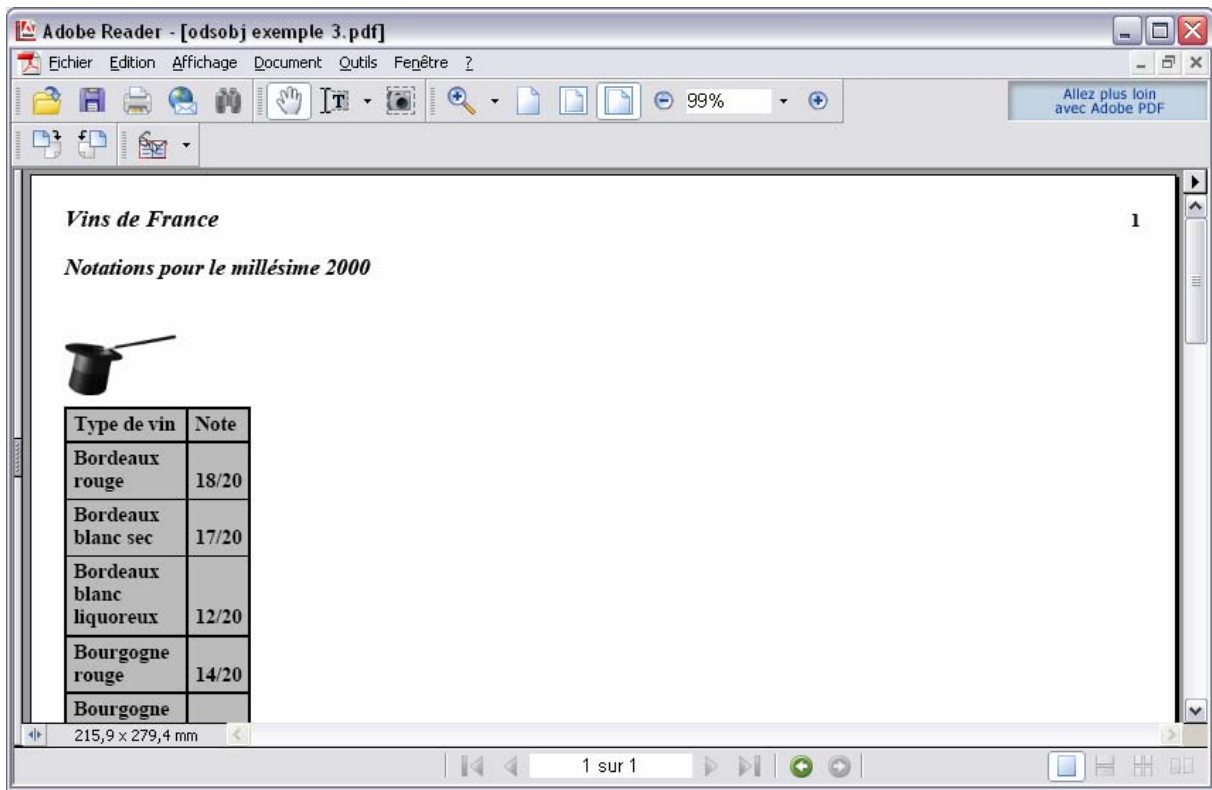
Méthode	Arguments	Description
TITLE	TEXT:"titre", JUST:LEFT CENTER RIGHT	Insertion de titre
PAGE	<i>aucun</i>	Insertion de saut de page
LINE	<i>aucun</i>	Insertion de ligne horizontale en travers de la page
NOTE	TEXT:"texte libre", JUST:LEFT CENTER RIGHT	Insertion de texte hors tableau
IMAGE	TEXT:"chemin et nom image"	Insertion d'image au-dessus du tableau

Exemple 3 – Insertion de titres et d'images dans l'objet ODSOUT

```

ODS LISTING CLOSE ;
ODS PDF FILE = "c:\odsout exemple 3.pdf" ;
TITLE1 "Vins de France" ;
DATA _NULL_ ;
    SET cours.millesimes (WHERE = (millesime=2000))
        END = fin ;
    IF _N_ = 1 THEN DO ;
        DECLARE ODSOUT rapport () ;
        rapport.TITLE(TEXT:"Notations pour le millésime 2000") ;
        rapport.IMAGE(TEXT:"c:\magie.gif") ;
        rapport.TABLE_START() ;
        rapport.ROW_START(TYPE:"HEADING") ;
        rapport.FORMAT_CELL(TEXT:"Type de vin", SPLIT:"#") ;
        rapport.FORMAT_CELL(TEXT:"Note") ;
        rapport.ROW_END() ;
    END ;
    rapport.ROW_START() ;
    rapport.FORMAT_CELL(TEXT:region, SPLIT:"#") ;
    rapport.FORMAT_CELL(TEXT:PUT(note,2.)!!"/20") ;
    rapport.ROW_END() ;
    IF fin THEN rapport.TABLE_END() ;
RUN ;
TITLE ;
    
```

```
ODS PDF CLOSE ;
```



Exemple 4 – Insertion de code HTML dans l'objet ODSOUT

```
ODS LISTING CLOSE ;
ODS HTML FILE = "c:\odsout exemple 4.htm" ;
DATA _NULL_ ;
  SET cours.millesimes (WHERE = (millesime=2000))
  END = fin ;
  IF _N_ = 1 THEN DO ;
    DECLARE ODSOUT rapport ( ) ;
    rapport.TITLE(TEXT:"Notations pour le millésime 2000") ;
    rapport.TABLE_START() ;
  END ;
  rapport.ROW_START() ;
  rapport.FORMAT_CELL(TEXT:region, SPLIT:"#") ;
  rapport.FORMAT_CELL(TEXT:
    REPEAT("<IMG SRC='etoile.jpg'>",
    note)) ;
  rapport.ROW_END() ;
  IF fin THEN rapport.TABLE_END() ;
RUN ;
ODS HTML CLOSE ;
```

La page Web produite par l'Exemple 4 contient un nombre d'étoiles proportionnel à la note du millésime. Le code HTML `` est inséré comme s'il s'agissait d'un texte normal.



Pour l'insertion d'images dans un document PDF ailleurs qu'au-dessus du tableau, on encadrera la méthode `IMAGE` telle qu'elle est décrite plus haute entre deux méthodes `CELL_START` et `CELL_END` sans arguments, qui servent à délimiter précisément le contenu d'une cellule de tableau.

Exemple 5 – Insertion d'images dans un fichier PDF

```
ODS LISTING CLOSE ;
TITLE "Millésimes de Bordeaux rouge" ;
ODS PDF FILE = "c:\millésimes.pdf" ;
DATA _NULL_ ;
  SET cours.millesimes (WHERE = (region =: "Bordeaux"))
  END = fin ;
  IF _N_ = 1 THEN DO ;
    DECLARE ODSOUT rapport ( ) ;
    rapport.TABLE_START ( ) ;
  END ;
  rapport.ROW_START ( ) ;
  rapport.FORMAT_CELL(TEXT:region) ;
  rapport.FORMAT_CELL(TEXT:PUT(millesime,4.) ) ;
```

```

DO i = 1 TO note ;
    rapport.CELL_START() ;
        rapport.IMAGE(TEXT:"c:\etoile.gif") ;
    rapport.CELL_END() ;
END ;
DO i = note+1 TO 20 ;
    rapport.FORMAT_CELL(TEXT:" ") ;
END ;
rapport.ROW_END() ;
IF fin THEN rapport.TABLE_END() ;
RUN ;
TITLE ;
ODS PDF CLOSE ;
    
```

Le résultat de l'Exemple 5 est analogue au résultat obtenu par l'Exemple 4, mais cette fois-ci en format PDF. On peut obtenir, avec le code de l'Exemple 5, en remplaçant ODS PDF par ODS HTML, le résultat dans une page Web sans autre aménagement du programme.

Millésimes de Bordeaux rouge

Bordeaux rouge	2001	★	★	★	★	★	★	★	★	★	★	★	★	★	★				
Bordeaux blanc sec	2001	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★			
Bordeaux blanc liquoreux	2001	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
Bordeaux rouge	2000	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
Bordeaux blanc sec	2000	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	
Bordeaux blanc liquoreux	2000	★	★	★	★	★	★	★	★	★	★	★							
Bordeaux rouge	1999	★	★	★	★	★	★	★	★	★	★	★	★	★					
Bordeaux blanc sec	1999	★	★	★	★	★	★	★	★	★	★	★	★						
Bordeaux blanc liquoreux	1999	★	★	★	★	★	★	★	★	★	★	★	★	★					
Bordeaux rouge	1998	★	★	★	★	★	★	★	★	★	★	★	★	★	★				
Bordeaux blanc sec	1998	★	★	★	★	★	★	★	★	★	★	★	★	★					
Bordeaux blanc liquoreux	1998	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★			
Bordeaux rouge	1997	★	★	★	★	★	★	★	★	★	★	★	★						
Bordeaux blanc sec	1997	★	★	★	★	★	★	★	★	★	★	★	★						
Bordeaux blanc liquoreux	1997	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★

1.5. Création de plusieurs fichiers

1.5.1. FICHIERS CONTENANT UN SEUL TABLEAU (HTML, PDF, RTF)

Si le fichier à créer doit contenir un seul tableau, on peut utiliser l'option `NEWFILE=TABLE`, valable pour les destinations RTF, PDF et HTML.

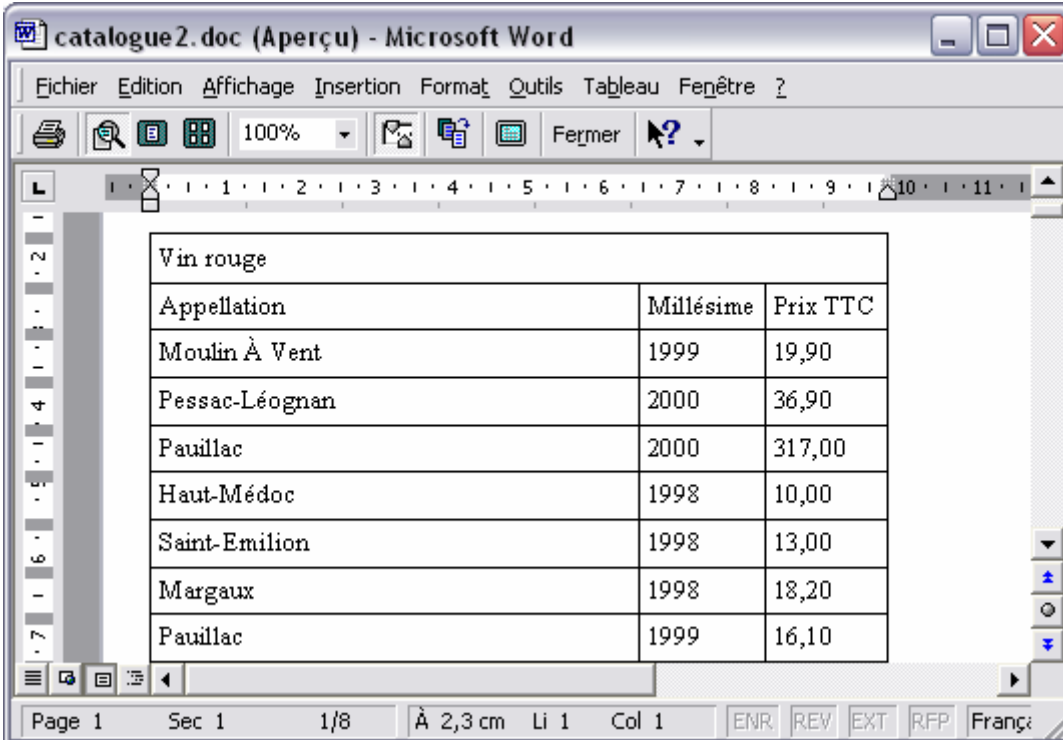
On doit ensuite s'arranger pour construire un tableau différent (avec les méthodes `TABLE_START` et `TABLE_END`) à chaque fois qu'on souhaite commencer un nouveau document.

Il n'est pas actuellement possible d'imbriquer plusieurs tableaux.

Exemple 6 – Création d'un fichier RTF par tableau

```
ODS LISTING CLOSE ;
PROC SORT DATA = cours.produits
      (WHERE = (millesime BETWEEN 1997 AND 2000))
      OUT = work.catalogue ;
  BY couleur ;
RUN ;
ODS RTF FILE = "c:\catalogue1.doc"
      NEWFILE = TABLE ;
DATA _NULL_ ;
  SET work.catalogue END = fin ;
  BY couleur ;
  IF _N_ = 1 THEN DECLARE ODSOUT rapport ( ) ;
  IF FIRST.couleur THEN DO ;
    rapport.TABLE_START( ) ;
    rapport.ROW_START( ) ;
    rapport.FORMAT_CELL(TEXT:"Vin "!!couleur,
                        COLUMN_SPAN:3) ;

    rapport.ROW_END( ) ;
    rapport.ROW_START( ) ;
    rapport.FORMAT_CELL(TEXT:"Appellation") ;
    rapport.FORMAT_CELL(TEXT:"Millésime") ;
    rapport.FORMAT_CELL(TEXT:"Prix TTC") ;
    rapport.ROW_END( ) ;
  END ;
  rapport.ROW_START( ) ;
  rapport.FORMAT_CELL(TEXT:appellation) ;
  rapport.FORMAT_CELL(TEXT:PUT(millesime,4.-L)) ;
  rapport.FORMAT_CELL(TEXT:PUT(prixTTC,NUMX12.2)) ;
  rapport.ROW_END( ) ;
  IF LAST.couleur THEN rapport.TABLE_END( ) ;
RUN ;
ODS RTF CLOSE ;
```



The screenshot shows a Microsoft Word window titled 'catalogue2.doc (Aperçu) - Microsoft Word'. The window displays a table with the following data:

Appellation	Millésime	Prix TTC
Moulin À Vent	1999	19,90
Pessac-Léognan	2000	36,90
Pauillac	2000	317,00
Haut-Médoc	1998	10,00
Saint-Emilion	1998	13,00
Margaux	1998	18,20
Pauillac	1999	16,10

1.5.2. FICHIERS CONTENANT PLUSIEURS TABLEAUX (PDF, HTML)

Pour les destinations ODS qui supportent l'option `NEWFILE=PAGE`, on peut changer de fichier créé en utilisant la méthode `PAGE`. Cela concerne les destinations PDF et HTML, la création d'un nouveau fichier à chaque saut de page n'étant pas possible en RTF.

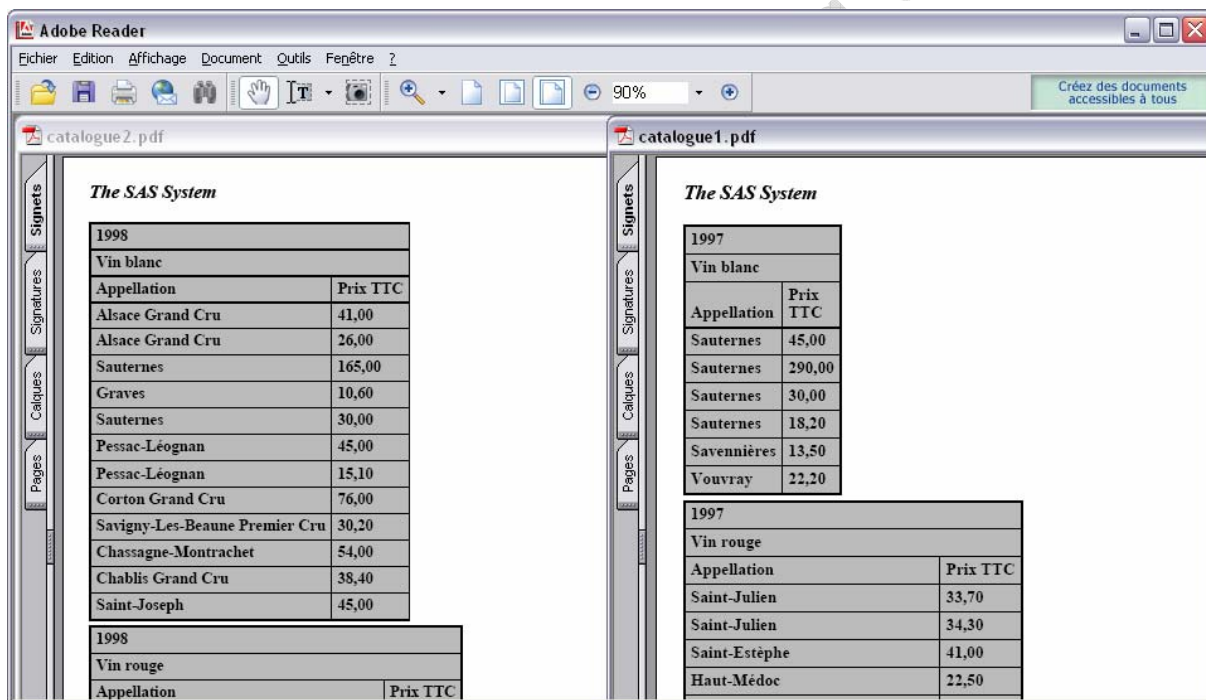
Exemple 7 – Création de plusieurs fichiers PDF contenant plusieurs tableaux

```
ODS LISTING CLOSE ;
ODS LISTING CLOSE ;
PROC SORT DATA = cours.produits
    (WHERE = (millesime BETWEEN 1997 AND 2000))
    OUT = work.catalogue ;
    BY millesime couleur ;
RUN ;
ODS PDF FILE = "c:\catalogue1.pdf"
    NEWFILE = PAGE ;
DATA _NULL_ ;
    SET work.catalogue END = fin ;
    BY millesime couleur ;
    IF _N_ = 1 THEN DECLARE ODSOUT rapport ( ) ;
    IF FIRST.couleur THEN DO ;
        rapport.TABLE_START ( ) ;
        rapport.ROW_START (TYPE:"Heading" ) ;
        rapport.FORMAT_CELL (TEXT:PUT(millesime,4.-L),
            COLUMN_SPAN:2) ;
        rapport.ROW_END ( ) ;
```

```
rapport.ROW_START(TYPE:"Heading") ;
  rapport.FORMAT_CELL(TEXT:"Vin "!!couleur,
                      COLUMN_SPAN:2) ;

rapport.ROW_END() ;
rapport.ROW_START(TYPE:"Heading") ;
  rapport.FORMAT_CELL(TEXT:"Appellation") ;
  rapport.FORMAT_CELL(TEXT:"Prix TTC") ;
rapport.ROW_END() ;
END ;

rapport.ROW_START(TYPE:"Data") ;
  rapport.FORMAT_CELL(TEXT:appellation) ;
  rapport.FORMAT_CELL(TEXT:PUT(prixTTC,NUMX12.2)) ;
rapport.ROW_END() ;
IF LAST.couleur THEN rapport.TABLE_END() ;
IF LAST.millesime AND NOT fin THEN rapport.PAGE() ;
RUN ;
ODS PDF CLOSE ;
```



1.6. Mise en régions (ODS PDF uniquement)

En conjonction avec la destination ODS PDF, il est possible de préciser que le rapport contenu dans l'objet ODSOUT est divisé en plusieurs régions. Il s'agit d'un découpage d'une page de résultats en zones autonomes, chacune contenant son propre tableau de résultats.

Le découpage est commencé par une méthode LAYOUT_GRIDDED comme dans la syntaxe suivante. Quand toutes les zones ont été alimentées, la méthode LAYOUT_END vient le signaler. Entre-temps, la méthode REGION a indiqué quelle zone on alimentait.

```
nomObjet.LAYOUT_GRIDDED(ROWS:nbLignes,  
                        COLUMNS:nbColonnes) ;  
nomObjet.REGION(positionEtForme) ;  
nomObjet.TABLE_START() ;  
...  
nomObjet.TABLE_END() ;  
...  
nomObjet.REGION(positionEtForme) ;  
...  
nomObjet.LAYOUT_END() ;
```

Afin d'indiquer quelle est la zone démarrée lors de l'invocation de la méthode REGION, on dispose de plusieurs arguments :

- COLUMN indique le n° de la colonne dans laquelle se trouve la zone ;
- ROW indique le n° de ligne dans laquelle se trouve la zone ;
- COLUMN_SPAN indique le nombre de colonnes fusionnées que représente la zone ;
- ROW_SPAN indique le nombre de lignes fusionnées que représente la zone.

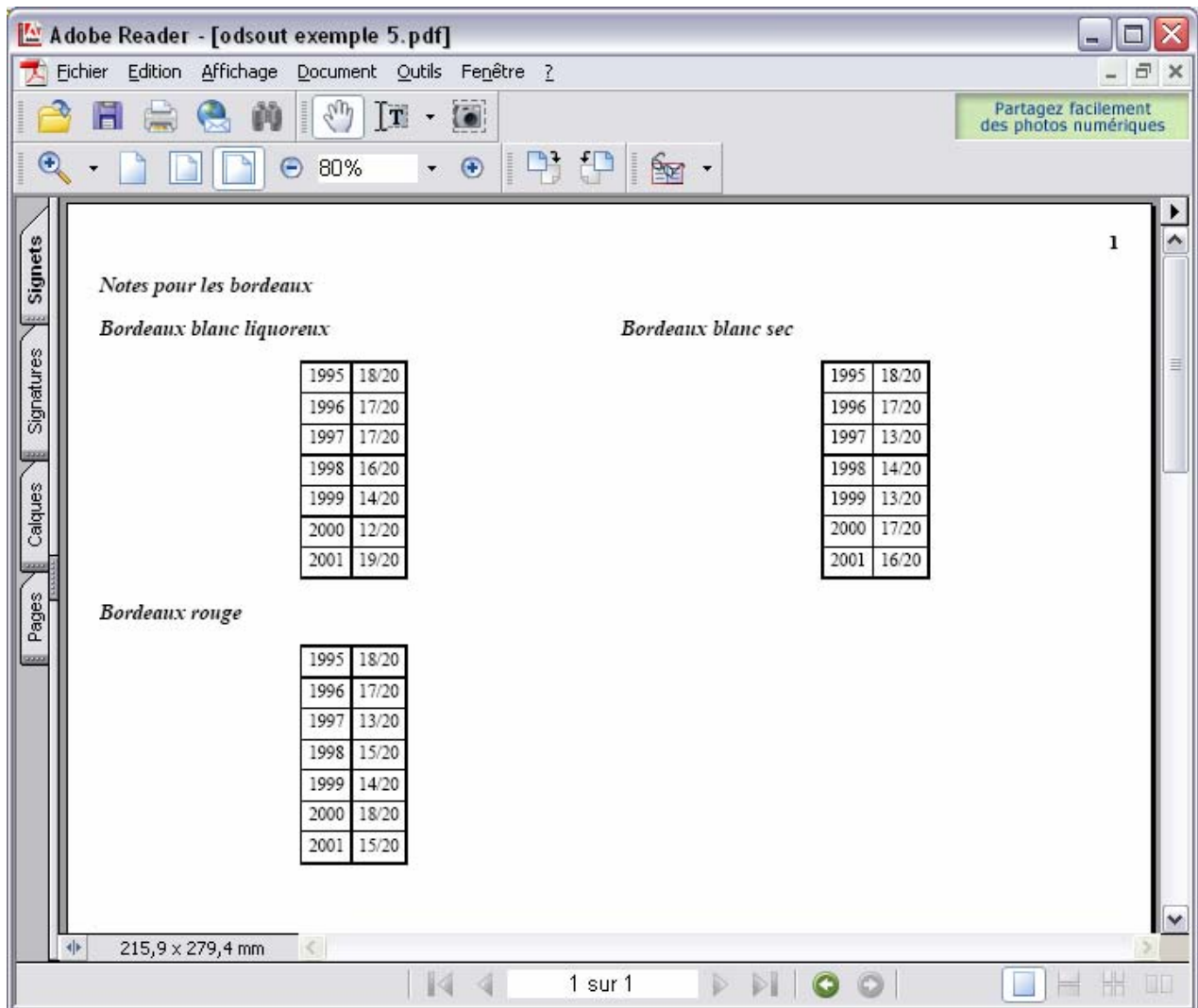
Plus que jamais, il est recommandé de dessiner sur papier le découpage que l'on souhaite réaliser avant de commencer à programmer.

Exemple 8 – Découpage en régions d'un objet ODSOUT

```
ODS LISTING CLOSE ;  
TITLE ;  
PROC SORT DATA = cours.millesimes  
            (WHERE = (region =: "Bordeaux"  
                    AND millesime >= 1995))  
    OUT = work.bordeaux ;  
    BY region millesime ;  
RUN ;
```

```
ODS PDF FILE = "c:\odsout exemple 5.pdf" ;
DATA _NULL_ ;
  SET work.bordeaux END = fin ;
  BY region millesime ;
  IF _N_ = 1 THEN DO ;
    DECLARE ODSOUT rapport ( ) ;
    rapport.TITLE(TEXT:"Notes pour les bordeaux) ;
    rapport.LAYOUT_GRIDDED(COLUMNS:2, ROWS:2) ;
  END ;
  IF FIRST.region THEN DO ;
    nbRegions + 1 ;
    IF nbRegions < 3 THEN
      rapport.REGION(COLUMN:nbRegions, ROW:1) ;
    ELSE
      rapport.REGION(ROW:2, COLUMN_SPAN:2) ;
    rapport.TITLE(TEXT:region) ;
    rapport.TABLE_START(OVERRIDES:"JUST=CENTER") ;
  END ;
  rapport.ROW_START() ;
  rapport.FORMAT_CELL(TEXT:PUT(millesime,4.-L)) ;
  rapport.FORMAT_CELL(TEXT:PUT(note,2.)!!"/20") ;
  rapport.ROW_END() ;
  IF LAST.region THEN rapport.TABLE_END() ;
  IF fin THEN rapport.LAYOUT_END() ;
RUN ;
ODS PDF CLOSE ;
```

Le résultat de l'Exemple 8 est découpé en 4 zones, dans un tableau de 2 lignes et 2 colonnes. Les deux colonnes de la dernière ligne sont ensuite fusionnées en une seule, ce qui donne au final une organisation en 3 zones.



Complément