

- Troisième écran : vous y précisez la bibliothèque dans laquelle vous souhaitez enregistrer la table créée par votre procédure d'importation, ainsi que le nom de cette table.
- Dernier écran : vous êtes invité à nommer, si vous le souhaitez, un fichier dans lequel sera enregistré le programme PROC IMPORT créé par le module d'importation. Via « Browse », vous pouvez naviguer sur votre disque dur afin de préciser l'endroit où ce fichier sera enregistré.

Exercice 2.28 : Importez la première feuille du classeur Excel HEURES.XLSX, que vous trouverez dans l'archive téléchargée sur www.sas-sr.com. Sauvegardez le programme PROC IMPORT. Comparez ce que vous obtenez dans la table créée avec les données présentées dans la feuille 1 du classeur Excel.

Si vous avez fait cet exercice, vous avez obtenu le programme 2.53.

Programme 2.53

```
PROC IMPORT OUT=WORK.TOT025
  DATAFILE="C:\intro_sas\chiers\heures.xlsx"
  DBMS=EXCEL REPLACE;
  RANGE="Feuil1$";
  GETNAMES=YES;
  MIXED=NO;
  SCANTEXT=YES;
  USEDATE=YES;
  SCANTIME=YES;
RUN;
```

À ce stade, il est très important de comprendre le principe de cette interface (qui fonctionne d'ailleurs comme toutes les interfaces de SAS) : **une interface SAS ne fait qu'écrire un programme SAS à votre place.**

Si vous demandez l'exécution du programme, vous obtiendrez exactement le même résultat qu'avec le module d'importation. Mais la production du programme finalement utilisé pour importer votre feuille Excel peut être intéressante, car vous constaterez de temps en temps que l'importation ne s'est pas bien déroulée. Vous devrez alors repartir du programme généré par l'interface d'importation et le modifier, pour ensuite en demander une nouvelle exécution. Nous n'explorerons pas ici en détail la syntaxe de PROC IMPORT (voir l'aide SAS à ce sujet¹). Cependant, à titre d'exemple, si vous avez réalisé l'exercice 2.28, vous avez dû constater que la variable DATE1 a été interprétée de façon imparfaite. Au lieu de

03/02/2011 12:26

pour la première observation dans Excel, vous obtenez dans SAS :

03FEB2011

La partie « heure » de la date est perdue.

Pour la retrouver à partir du programme 2.53, il faut remplacer la valeur de l'option USEDATE= par NO – ce problème est lié à une différence de codification des dates et heures entre Excel et SAS (voir la section suivante).

Autre exemple : dans la feuille 3 du classeur Excel heures.xlsx, les données que l'on souhaite importer (identiques à celles présentées dans la feuille 1) sont « noyées » dans la feuille et affichées dans les cellules C5:E15 (des données ont été ajoutées dans les cellules alentour). Pour utiliser uniquement ces données dans votre procédure d'importation, il suffit de modifier la valeur de l'option RANGE= en :

```
RANGE="Feuil1$C5:E15";
```

Attention cependant : même si vous disposez d'Excel 2007 (ou version ultérieure), et donc de la possibilité d'avoir plus de 256 colonnes (16 384 avec Excel 2007), vous ne pourrez pas importer plus de 255 colonnes vers une table SAS. Le nombre d'observations sera limité par le nombre de lignes que peut gérer votre version d'Excel (65 536 avec Excel 2005 et 1 048 576 lignes avec Excel 2007).

2.7.2 LIBNAME Excel

Vous savez déjà qu'au moyen d'une instruction LIBNAME, vous pouvez accéder à des tables SAS enregistrées sur votre disque dur. Cette instruction permet aussi de considérer un classeur Excel de la même manière que l'instruction LIBNAME « BASE » permet de définir une bibliothèque dans laquelle vous avez des tables. Votre classeur Excel peut ainsi obtenir le statut de bibliothèque : ses différentes feuilles constituent des tables que vous pourrez manipuler au même titre qu'une table SAS. Contrairement à l'interface d'importation, il n'y a pas création d'une table mais attribution au classeur Excel d'un statut de bibliothèque – les feuilles du classeur sont ensuite lues comme si elles étaient des tables.

Attention cependant : cette instruction LIBNAME diffère de l'instruction LIBNAME « BASE » dans le sens où, pour tirer profit de ses capacités, vous devez disposer du module SAS/ACCESS interface to PC Files.

En fonction de votre version de SAS, vous ne pourrez pas accéder à tous les fichiers Excel :

- Les utilisateurs de la version 9.1 ne peuvent accéder qu'aux classeurs Excel créés par Excel 2003 (ou versions antérieures).
- Les utilisateurs de la version 9.2 de SAS peuvent accéder à tous les types de classeurs Excel, y compris les versions XLSX générées par Excel 2007.

a. Principes

L'instruction LIBNAME « ACCESS » se présente, dans sa syntaxe, identiquement à la version « BASE » :

```
LIBNAME nom "chemin vers le classeur Excel" <options>;
```

Attention, avec LIBNAME, vous devez pointer non pas un dossier, comme nous le faisons auparavant, mais bien un classeur Excel :

```
LIBNAME toto "C:\intro_sas\chiers\client.xlsx";
```

Le classeur Excel que vous allez utiliser ici (présent dans l'archive à télécharger sur www.sas-sr.com) est reproduit à la figure 2.8.

A	B	C	D	E	F	G	H	I
1	dent							
2	1	64 rue de Valenciennes	58110 Valenciennes	N001	02/06/2010	21	6 892,54 €	
3	2	5 rue Pierre-Joseph Redouté	01650 Comblanchet-sur-Fismes	N002	09/06/2010	11	3 610,38 €	
4	3	2 rue Joseph Ceaur	07920 Vacheco	N003	04/06/2010	21	6 892,54 €	
5	4	95 place Polyvalente	11640 Marville	N004	04/06/2010	18	5 907,89 €	
6	5	39 rue du Pont de l'Arquet	33110 Bardenac-sur-Florentins	N005	04/06/2010	26	8 539,62 €	
7	6	20 place du Poudreau	04250 Mandargat	N006	06/06/2010	31	10 174,70 €	
8	7	37 rue Robert Schuman	14110 Guingou	N007	07/06/2010	9	2 953,85 €	
9	8	6 rue François de Crille	34460 Marillac	N008	08/06/2010	14	4 595,08 €	
10	9	12 rue Louis Braille	07090 Vidancé-en-Picardie	N009	10/06/2010	14	4 595,08 €	
11	10	44 rue Maladrene	28670 Bricouby	N010	10/06/2010	22	7 220,76 €	
12	11	23 place du Souffrenan	07920 Vacheco	N011	10/06/2010	19	6 236,11 €	
13	12	59 dossier Thomas More	54420 Prunoy	N012	12/06/2010	19	6 236,11 €	
14	13	feuille						

Figure 2.8 • Extrait du classeur Excel client.xlsx.

Divers formats Excel ont été imposés : un format code postal, un format date et un format monétaire. Comme dans SAS, certaines variables de ce classeur sont numériques, tandis que d'autres contiennent des chaînes de caractères.

Si vous avez exécuté l'instruction globale LIBNAME indiquée plus haut, vous pouvez maintenant, via l'explorateur SAS, accéder au classeur Excel devenu bibliothèque, si vous avez ce message dans votre fenêtre JOURNAL :

```
1 LIBNAME toto "C:\intro_SAS\chiers\client.xlsx";
NOTE: LIBREF TOTO attribué comme suit :
MOTEUR : EXCEL
Nom physique : C:\intro_SAS\chiers\client.xlsx
```

La figure 2.9 reprend ce que vous observez normalement dans l'onglet EXPLORATEUR.

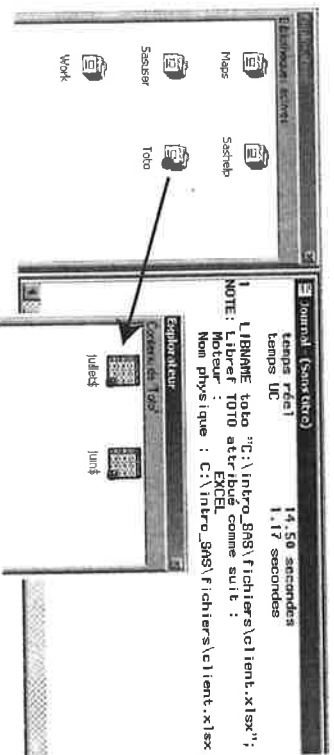


Figure 2.9 • Classeur client.xlsx et onglet EXPLORATEUR de SAS.

Vous constatez que les deux feuilles du classeur client.xlsx se présentent maintenant comme deux tables de la bibliothèque permanente TOTO. Les noms des feuilles Excel ont cependant été modifiés par l'ajout d'un signe \$. Cela pourrait être problématique puisque le signe \$ n'est pas autorisé dans les noms que vous pouvez donner aux tables. Ce signe \$ vous contraint en fait à référencer les tables contenues dans les classeurs Excel en suivant une certaine méthode, laquelle est illustrée par le programme 2.54.

Programme 2.54

```
PROC PRINT DATA=toto.'juin$';
RUN;
```

Le nom de votre table, avec son signe \$, doit être encadré par deux quotes (simples ou doubles) et être suivi d'un 'n' (majuscule ou minuscule). Ce 'n' autorise en fait la présence du signe \$ (et de tout autre caractère spécial) à l'intérieur d'un nom de table. Il est cependant possible que vous ayez un jour dans votre explorateur le résultat illustré à la figure 2.10.

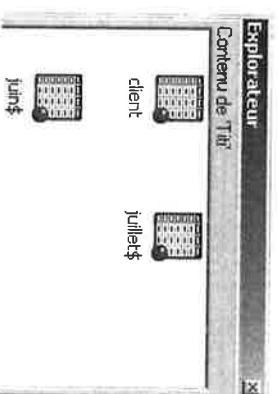


Figure 2.10 • Présence d'une plage nommée.

La bibliothèque TITI, qui pointe vers un classeur Excel, possède trois tables. Vous remarquez que le nom de deux des tables contient le signe \$, mais pas la table CLIENT. De fait, celle-ci correspond à une plage nommée (voir figure 2.11) dans une des feuilles Excel.

A	B	C	D	E	F	G	H	I
1	dent							
2	1	64 rue de Valenciennes	33410 Bardenac-sur-Florentins	N001	02/06/2010	21	6 892,54 €	
3	2	5 rue Pierre-Joseph Redouté	01650 Comblanchet-sur-Fismes	N002	09/06/2010	11	3 610,38 €	
4	3	2 rue Joseph Ceaur	07920 Vacheco	N003	04/06/2010	21	6 892,54 €	
5	4	95 place Polyvalente	11640 Marville	N004	04/06/2010	18	5 907,89 €	
6	5	39 rue du Pont de l'Arquet	33110 Bardenac-sur-Florentins	N005	04/06/2010	26	8 539,62 €	
7	6	20 place du Poudreau	04250 Mandargat	N006	06/06/2010	31	10 174,70 €	
8	7	37 rue Robert Schuman	14110 Guingou	N007	07/06/2010	9	2 953,85 €	
9	8	6 rue François de Crille	34460 Marillac	N008	08/06/2010	14	4 595,08 €	
10	9	12 rue Louis Braille	07090 Vidancé-en-Picardie	N009	10/06/2010	14	4 595,08 €	
11	10	44 rue Maladrene	28670 Bricouby	N010	10/06/2010	22	7 220,76 €	
12	11	23 place du Souffrenan	07920 Vacheco	N011	10/06/2010	19	6 236,11 €	
13	12	59 dossier Thomas More	54420 Prunoy	N012	12/06/2010	19	6 236,11 €	
14	13	feuille						

Figure 2.11 • Plage nommée dans Excel.

Les plages nommées sont particulièrement utiles lorsque les données que l'on souhaite utiliser en tant que tables SAS ne sont présentes que dans une partie de la feuille.

Il existe une limite importante à la capacité qu'a SAS de considérer une feuille Excel au même titre qu'il considère une table : il ne pourra pas lire plus de 255 variables dans une feuille Excel, même s'il s'agit d'une feuille réalisée sous Excel 2007 – qui peut contenir jusqu'à 16 384 colonnes¹.

¹ Cette caractéristique est propre à SAS 9.2. La version 9.3, annoncée pour fin 2011-début 2012 autorisera la lecture de plus

La transformation de feuilles Excel en tables accessibles par SAS n'est cependant pas sans poser de problème. Nous avons vu qu'un certain nombre de formats avaient été imposés dans Excel : comment ont-ils été compris par SAS ?

Nous commençons notre examen de la table JUIJIN\$ au moyen d'un PROC CONTENTS, afin d'accéder aux informations descriptives de la table (voir section 1.7 pour une première approche et section 5.1 pour plus d'informations sur PROC CONTENTS).

Programme 2.55

```
PROC CONTENTS DATA=toto.juin$ n;
```

Extrait 2.24 (extrait)

Procédure CONTENTS				Liste alphabétique des variables et des attributs	
Variable	Type	Long.	Format	Informat	Libellé
CP	Num.	8	\$23.	\$23.	CP
Ville	Texte	23	\$23.	\$23.	Ville
client	Texte	11	\$11.	\$11.	client
date	Num.	8	DATE9.	DATE9.	date
facture	Texte	4	\$4.	\$4.	facture
nombre_d_articles	Num.	8			nombre d'articles
num	Num.	8			num
rue	Texte	27	\$27.	\$27.	rue
total	Num.	8	DOLLAR21.2	DOLLAR21.2	total

La partie haute du résultat du PROC CONTENTS (non reproduite dans le résultat 2.24) vous indique déjà que vous ne disposez pas d'une table SAS comme les autres : nombre d'observations, longueur d'observation, date de création, dernière modification, etc., ne sont pas renseignés.

La lecture de ce résultat vous permet de voir (à peu près...) ce qui se produit lorsque vous demandez à SAS de lire un classeur Excel comme une table. Une synthèse de ce fonctionnement est illustrée à la figure 2.12.

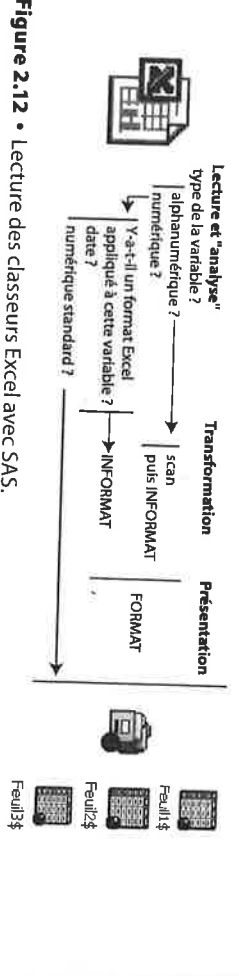


Figure 2.12 • Lecture des classeurs Excel avec SAS.

SAS cherche dans un premier temps à connaître le type de la variable. La première ligne est considérée comme contenant des informations sur le nom des variables. SAS analyse ensuite la nature des modalités observées sur la seconde ligne puis sur les 8 lignes suivantes et en fonction de ce qu'il observe déterminera si la colonne contient une variable numérique ou une variable caractère. Si la variable est caractère, SAS scanne

alors entièrement la colonne, regarde la longueur de la chaîne de caractères la plus longue et applique un INFORMAT de type \$w. adapté à cette longueur maximale. Ce fonctionnement garantit une fidélité totale entre vos variables caractères dans Excel et leur lecture dans SAS.

Si SAS décide que la variable est numérique, il regarde alors si un format est appliqué aux modalités et il s'en sert pour décider quel INFORMAT il devra utiliser pour lire les modalités de la variable. (En passant, SAS semble scanner entièrement la colonne, la aussi, et ne considère le format Excel que si celui-ci est appliqué à une majorité de cellules.)

La notion d'INFORMAT est cependant ici un peu différente de celle qui caractérise la création d'une table à partir d'un fichier TXT. Imaginons en effet que dans une cellule Excel, vous observiez la valeur suivante : 40212.42858

Vous appliquez un format de type date à cette valeur dans Excel et vous obtenez : 03/02/2010. Dans Excel en effet, une date correspond à un nombre de jours depuis le 1^{er} janvier 1900 ; la partie décimale indique à quel moment de la journée vous vous trouvez (0.5 correspondra à midi, 0.42858 correspond à 10:17:09).

Vous lisez maintenant votre table Excel dans SAS au moyen d'une instruction LIBNAME et vous obtenez dans SAS la valeur suivante : 03FEB2010

Retirez le FORMAT DATE9. et vous obtenez la valeur suivante : 18296.43 ; souvenez-vous que dans SAS, une date est un nombre de jours depuis le 1^{er} janvier 1960. Ce n'est donc pas la chaîne de caractères « 03/02/2010 » qui est lue par SAS au moyen de DATE9. Déjà parce que DATE9. ne sait pas lire cette chaîne, et ensuite parce que s'il l'avait lue, il n'y aurait pas de décimales dans cette date lue sans FORMAT dans SAS.

En réalité, SAS prend la valeur 40212.42858 et retire 21916 de sorte que, présentée au moyen d'un FORMAT DATE9., la date SAS apparaisse identique à la date Excel, malgré la différence de référence « temps 0 » entre SAS (1^{er} janvier 1960) et Excel (1^{er} janvier 1900). Cette transformation est dictée par la présence du format Excel appliqué aux données d'une colonne.

Certains noms de variables ont été modifiés, notamment, dans le cas présent, en remplaçant par le signe '_' les espaces et caractères spéciaux non acceptés dans les noms de variables SAS : 'nombre d'articles' est ainsi devenu 'nombre_d_articles'. Le libellé original est cependant conservé en tant que LABEL (voir section 5.2) de la variable.

Quant aux autres données numériques, SAS a analysé et transformé de diverses manières les formats imposés dans la feuille Excel.

Aucun FORMAT particulier ne semble avoir été appliqué à la variable CP (code postal). On devine donc que les codes postaux au format 06100 seront présentés, dans la feuille Excel vue par SAS, de la manière suivante : 6100. L'affichage ne sera donc pas correct et vous n'aurez plus qu'à créer un FORMAT spécifique pour présenter correctement ces codes postaux¹.

1. Ce FORMAT spécifique est très simplement créé par le programme suivant :

```
PROC FORMAT;
  PICTURE cp low-high="9999";
RUN;
```

En ce qui concerne la variable TOTAL, SAS applique un FORMAT de type DOLLAR.w,d, alors qu'EUROX.w,d aurait permis l'affichage du signe € au lieu du signe \$, ainsi qu'un séparateur décimal virgule. Il suffit de le savoir... De plus, le FORMAT demande l'affichage de 21 caractères, ce qui paraît un peu exagéré.

L'affichage des données au moyen d'un PROC PRINT donne des résultats conformes à ce que l'on pouvait attendre par la simple lecture du résultat du PROC CONTENTS (voir résultat 2.25).

Résultat 2.25 (extrait)

Obs	client	num	rue	CP	Ville	facture	date	nombre_d_articles	total
1	Agrès	54	rue de Haredeus	33410	Bedrieux-les-Pinurs	N001	02JUN2010	21	\$6,892.54
2	Applinaire	5	rue Pierre-Joseph Redoué	1630	Comillend-sur-Franche	N002	03JUN2010	11	\$3,610.38
3	Edwige	2	rue Joseph Court	7920	Vichevro	N003	04JUN2010	21	\$6,892.54
4	Ori	93	place Polyvalente	11640	Marville	N004	04JUN2010	18	\$5,907.89

b. Création et modification de tables

Les feuilles Excel transformées et vues comme des tables SAS peuvent servir à la création de nouvelles tables, mais ne peuvent pas être modifiées à partir de SAS. Les programmes qui suivent font le tour des différentes possibilités qui vous sont offertes.

Avec SAS, vous pouvez créer une table SAS standard à partir d'une feuille Excel¹ :

Programme 2.56

```
DATA test;
SET toto.'juin$'n (RENAME=(nombre_d_articles=nb_art));
pu=totol/nb_art;
FORMAT total EUROX10.2;
RUN;
```

Vous pouvez aussi créer une nouvelle feuille dans votre classeur Excel :

Programme 2.57

```
DATA toto.modif;
SEM toto.'juin$'n (RENAME=(nombre_d_articles=nb_art));
pu=totol/nb_art;
FORMAT total EUROX10.2;
RUN;
```

Attention : si vous exécutez ce programme et si vous explorez ensuite le contenu de la bibliothèque qui pointe vers le classeur Excel, vous constaterez la présence d'une table MODIF et d'une table MODIF\$. En réalité, vous ne disposez que d'une seule table : lorsque vous créez une nouvelle feuille dans votre classeur, SAS exporte les données dans une feuille qu'il va appeler MODIF et les données seront placées au sein de cette

feuille, dans une plage nommée elle aussi MODIF. Cela ne porte pas à conséquence car, une fois que les données se trouvent dans une feuille/plage nommée, vous ne pouvez plus les modifier (voir plus bas).

Vous pouvez aussi créer un nouveau classeur Excel *via* une instruction LIBNAME puis créer une feuille à l'intérieur de ce classeur :

Programme 2.58

```
LIBNAME new "C:\intro_sas\chiers\modif.xlsx";
DATA new.modif;
SET toto.'juin$'n (RENAME=(nombre_d_articles=nb_art));
pu=totol/nb_art;
FORMAT total EUROX10.2;
RUN;
```

Si vous exécutez le programme 2.58, vous observerez dans la fenêtre JOURNAL le message suivant :

NOTE: Les libellés, formats et longueurs des variables SAS ne sont pas copiés dans les tables SGBD.

Ce message indique que, dans les feuilles Excel que vous créez, les en-têtes de colonne portent le nom de vos variables SAS : les LABEL de variables ne peuvent pas devenir en-têtes de colonnes dans une feuille Excel. Les FORMAT propres à SAS ne peuvent pas non plus apparaître dans votre feuille Excel, ni les limitations que vous imposez au nombre de décimales. En revanche, la conversion de dates SAS (nombre de jours depuis le 1/1/1960) en dates Excel (nombre de jours depuis le 1/1/1900) ne pose ici aucune difficulté.

Il n'est pas possible de modifier votre feuille Excel de départ, qu'elle existe déjà dans votre classeur Excel ou que vous la créez. Si vous exécutez le programme 2.59 :

Programme 2.59

```
DATA toto.'juin$'n;
SET toto.'juin$'n (RENAME=(nombre_d_articles=nb_art));
pu=totol/nb_art;
FORMAT total EUROX10.2;
RUN;
```

Vous aurez dans la fenêtre JOURNAL le message suivant :

ERROR: La MS Excel table juin\$a a été ouverte en SORTIE. Cette table existe déjà, ou bien il y a un conflit de nom avec un objet existant.
Cette table ne sera pas remplacée. Ce moteur ne prend pas en charge l'option REPLACE.

c. Les options de LIBNAME

Il est utile d'étudier les options possibles de LIBNAME dans ce cadre, car elles vont nous permettre de mieux comprendre comment se fait la passerelle entre SAS et Excel. La syntaxe générale est :

```
LIBNAME toto "chemin vers votre fichier Excel" <options>;
```

Vous pouvez aussi appliquer ces différentes options à des bibliothèques déjà créées. La syntaxe devient :

```
LIBNAME toto <options>;
```


CLEAR - Avec Excel 2007, si vous avez ouvert une bibliothèque pointant vers un classeur Excel, il est possible que vous ne puissiez pas ouvrir ce même classeur avec Excel. Il peut alors être utile de fermer votre bibliothèque au moyen de l'option CLEAR.

LIST - Cette option permet de voir les caractéristiques d'une bibliothèque existante - celles-ci s'affichent dans la fenêtre JOURNAL.

Options de création des bibliothèques :

DBMAX_TEXT=n - Indique la longueur de la chaîne de caractères la plus longue. 'n' peut être compris entre 256 et 32767. Si une chaîne de caractères présente dans une cellule de votre feuille Excel est plus longue, la chaîne sera tronquée. Valeur par défaut : 1024.

GETNAMES=YES/NO - Par défaut, GETNAME=YES et SAS considère la première ligne de la table (ou de la plage nommée) comme contenant le nom des variables. Si vous indiquez NO, les variables seront nommées F1, F2, etc., et la première ligne sera considérée comme contenant des modalités de variables.

MIXED=YES/NO - Dans son fonctionnement par défaut (MIXED=NO), SAS attribue le type de la variable par rapport aux premières modalités rencontrées. Cela peut poser problème si les modalités caractères d'une colonne sont rares et qu'elles doivent absolument être conservées. SAS risque en effet de considérer que votre colonne contient une variable numérique et vos modalités caractères seront alors valeur manquante. Si vous entrez MIXED=YES, toutes les variables seront caractères : vous n'aurez aucune perte de modalités.

SCANTEXT=YES/NO - Par défaut, SCANTEXT=YES : lorsque vous avez une variable caractère, SAS attribue à la variable la longueur de la modalité la plus longue, en explorant l'intégralité des données contenues dans la colonne. Si vous utilisez SCANTEXT=NO, la longueur des variables caractères sera fixe et égale à 255.

SCANTIME=YES/NO - Si SCANTIME=YES, les colonnes sont toutes scannées à la recherche de modalités représentatives de dates et d'heures. SAS applique le format TIME. s'il rencontre dans une même colonne uniquement des valeurs représentant des heures. Lorsque SCANTIME=NO (valeur par défaut), les colonnes ne contenant que des modalités d'heures sont présentées au moyen de FORMAT DATE9. Cette option est rendue nécessaire par le fait que dates et heures ne font pas l'objet de codifications distinctes dans Excel, contrairement à SAS. Pour Excel, une heure correspond à la partie décimale d'un nombre de jours depuis le 1^{er} janvier 1900¹, alors que pour SAS, c'est un nombre de secondes depuis l'heure de minuit précédente.

USEDATE=YES/NO - Par défaut, USEDATE=YES et les dates/heures sont affichées au moyen de DATE9. ; si vous choisissez USEDATE=NO, les dates et heures seront affichées via FORMAT DATETIMEw. et les dates de votre feuille Excel seront donc réinterprétées en un nombre de secondes depuis le 1^{er} janvier 1960.

2.7.3 L'interface d'exportation

De même qu'il existe une interface d'importation vous permettant de créer des tables SAS à partir de fichiers extérieurs à SAS, SAS propose une interface d'exportation. Comme pour l'interface d'importation, vous accédez à cette interface via le menu **Fichier, puis Exporter données**.

Les différents écrans qui se succèdent lorsque vous activez cette interface vous demandent de préciser :

- La table SAS que vous voulez exporter.
- Le type de fichier que vous souhaitez : Excel (2007 et versions précédentes), Access (97 et 200x), CSV, TXT, DBASE, IMP, SPSS, STATA, PARADOX et LOTUS 1-2-3 ; pour cet exemple, nous créerons un classeur Excel.
- Le nom de ce fichier et l'emplacement de sauvegarde sur votre disque dur.
- S'il s'agit d'un classeur Excel, le nom de la feuille qui accueillera vos données.
- Le nom du fichier de programme SAS dans lequel sera sauvegardée la syntaxe du PROC EXPORT, équivalant à ce que produira l'interface d'exportation.

Exercice 2.29 : Exportez dans un classeur Excel les données de la table SAS que vous avez construite lors de l'exercice 2.28 ; sauvegardez le programme PROC EXPORT produit par l'interface d'exportation.

À l'issue de l'exercice 2.29, vous avez face à vous le programme 2.60.

Programme 2.60

```
PROC EXPORT DATA=WORK.POTO2S
  OUTFILE='C:\intro_sas\chiers\test.xls'
  DMS=EXCEL REPLACE;
  SHEET='mes données';
RUN;
```

Si vous constatez, dans le fichier exporté, des différences avec les données de votre table SAS, vous pourrez retrouver dans votre fichier des données fidèles à vos données initiales via les jeux d'options et les instructions propres à PROC EXPORT (voir l'aide SAS¹).

2.7.4 La création de fichiers TXT via FILE et PUT

Par ailleurs, vous pouvez créer, à partir d'une table SAS et sans utiliser le module d'exportation, un fichier TXT contenant vos données. Cette écriture passe par les instructions FILE et PUT : nous allons voir que cette paire d'instructions fonctionne, dans les grandes lignes, comme le couple INFILE/INPUT.

L'instruction PUT seule (sans instruction FILE) présente des attributs intéressants. Nous y avons déjà eu recours dans les programmes 2.40, 2.46 et 2.47.

Le « ? » qui précède le nom du FORMAT à utiliser est important. Si vous l'omettez, SAS ne placera pas de séparateur entre la variable X2 et votre date et il faut 10 caractères pour écrire cette dernière. Ici, le signe deux-points permet d'ajouter un séparateur de champs (l'espace, par défaut dans cet exemple) à la suite de la modalité.

Le séparateur décimal dans SAS étant le point, l'imposition d'un FORMAT, peut devenir obligatoire si la table contient des données décimales et si le logiciel auquel est destiné votre fichier TXT ne comprend pas le « ? » comme séparateur décimal (Excel par exemple). Cette modification de FORMAT peut aussi être nécessaire lorsque vos variables mesurent des montants dans une quelconque monnaie (\$1,000,000.00).

Ouvrir un fichier TXT dans Excel n'est certes pas la façon la plus simple d'exporter vos données SAS dans Excel, mais c'est possible : si vos modalités ne contiennent pas d'espace, vous pouvez ouvrir votre fichier sous Excel en indiquant que votre fichier est de type « largeur fixe ». Si des espaces interviennent dans vos modalités, introduisez un séparateur de champs particulier via l'option DLM= possible pour FILE – votre fichier est alors du type « délimité ». Sur l'écran suivant du module d'importation des données d'Excel, vous préciserez le séparateur de champs que vous avez adopté.

Exercice 2.30 : Parmi les fichiers d'accompagnement de cet ouvrage, vous trouverez une table SAS appelée EXPORTER. Exportez-la via FILE et PUT de sorte qu'Excel la comprenne parfaitement.

2.8 Le débogage des programmes de création de table

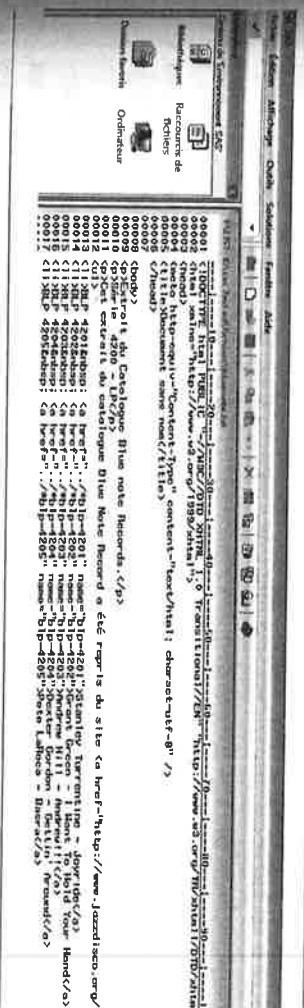
La procédure de création d'une table SAS à partir d'un fichier extérieur à SAS est une procédure délicate. En effet, il est absolument essentiel que votre table soit bien l'image exacte du fichier dont vous disposez. Une mauvaise spécification de votre ligne INPUT et un mauvais jeu d'options dans l'instruction INFILÉ vous conduiront immanquablement à créer des tables n'ayant qu'un rapport au mieux lointain avec votre fichier original. L'objet de cette dernière section est de vous donner quelques conseils qui vous permettront de vérifier que votre table est bien à l'image des données de votre fichier.

2.8.1 Regardez et comprenez votre fichier

Il est absolument impossible de créer une table SAS si l'on ne comprend pas la structure du fichier et l'organisation des données.

Pour examiner votre table, vous pouvez utiliser un traitement de texte extérieur ou les commandes SAS FSLIST et NOTEPAD. La commande FSLIST s'exécute dans la fenêtre COMMAND (en haut à gauche). SAS vous invite à ouvrir un fichier : demandez à pouvoir ouvrir tous types de fichiers et retrouvez votre fichier sur le disque dur. Dans la fenêtre COMMAND, vous pouvez alors demander COLS pour voir apparaître une réglette, ou NUMBER pour afficher des numéros de lignes. Vous obtenez ainsi l'écran du résultat 2.27.

Résultat 2.27



La réglette vous sera particulièrement utile pour connaître le nombre de caractères à réserver aux différentes variables caractères, si vous avez à le préciser. La commande NOTEPAD permet le même type de résultat, sauf que lorsque vous ordonnez NOTEPAD dans la fenêtre COMMAND, une fenêtre vide apparaît. Pour ouvrir un fichier, cliquez alors du bouton droit dans la fenêtre, demandez Ouvrir, puis retrouvez votre fichier. Pour faire apparaître la réglette, demandez NUMBER dans la fenêtre COMMAND, puis demandez, dans la fenêtre NOTEPAD, l'insertion de réglette à l'endroit où vous le souhaitez en tapant COLS, puis ENTER sur les numéros de lignes. Contrairement à FSLIST, NOTEPAD permet la modification des fichiers. L'enregistrement de votre fichier modifié avec NOTEPAD est automatique lorsque vous quittez la fenêtre. Si vous utilisez la fonction rechercher-remplacer, les remplacements ne seront effectifs que lorsque vous quitterez la fenêtre. De plus, seuls les rechercher-remplacer de caractères sont possibles. Contrairement à des traitements de texte plus évolués, SAS ne donne pas la possibilité de remplacer, par exemple, une chaîne de caractères par une tabulation.

Si la taille de votre fichier est importante, ne tentez pas immédiatement de créer votre table à partir du fichier complet : travaillez plutôt sur un échantillon d'environ 1 000 à 2 000 enregistrements. Ensuite, écrivez votre instruction INPUT et présentez la liste des variables que vous allez créer, puis réfléchissez aux INFORMAT que vous devez appliquer. Indiquez les options nécessaires à INFILÉ pour que votre fichier soit bien compris. Vous pourrez alors soumettre votre programme.

2.8.2 Lisez la fenêtre JOURNAL

Les utilisateurs débutants oublient très souvent de consulter la fenêtre JOURNAL. Pourtant, c'est dans cette fenêtre que vous lisez les informations qui vous aideront à comprendre si l'exécution de votre programme s'est bien déroulée et si votre table a bien été construite. Le JOURNAL doit être lu de haut en bas : remontez dans votre fenêtre JOURNAL jusqu'à l'apparition du programme que vous venez de soumettre. Vous n'apprenez rien du JOURNAL si vous commencez par la fin.

Imaginons un programme qui a effectivement créé une table SAS sans erreur de compilation (sans erreur de syntaxe, donc – l'erreur de syntaxe la plus courante étant l'absence de point-virgule).

Programme 2.61

```
DATA _NULL_ ;
X=12;
PUT "La valeur de X est égale à " X;
X=20;
PUT "maintenant, la valeur de X est égale à " X
/ "cette valeur est plus élevée que la précédente !"
/ @15 "c'est étonnant"
/ 18*"- " X EUROX12.2-L 5*"- "
/ 18*"- " X : EUROX12.2-L 5*"- "
/ X= ;
RUN;
```

L'instruction PUT demande l'écriture dans la fenêtre JOURNAL d'un texte ou de la valeur prise par une variable. Vous pouvez demander l'application d'un FORMAT (voir section 5.3), sauter une ligne, préciser l'emplacement où le texte (ou la variable) doit être écrit. Le programme 2.61 permet ainsi d'avoir dans votre fenêtre JOURNAL le résultat 2.26.

Résultat 2.26 (fenêtre JOURNAL)

```
La valeur de X est égale à 12
maintenant, la valeur de X est égale à 20
cette valeur est plus élevée que la précédente !
-----€20,00 -----
-----€20,00 -----
X=20
```

Puisque l'instruction INPUT vous est maintenant familière, vous constaterez immédiatement que les options que nous avons présentées pour INPUT peuvent aussi être utilisées avec PUT (pointeur @X ou / par exemple). EUROX12.2 est un FORMAT qui demande l'affichage de la modalité (ici 20) sur 12 caractères, avec deux chiffres après la virgule. Le '-L' qui suit le nom du FORMAT demande un alignement à gauche (*Left*) de la modalité dans le champ de 12 caractères (6 espaces apparaissent entre le dernier 0 et le '-' qui suit). Si le FORMAT est associé à '; un espace est ajouté après la modalité. Les deux-points demande à SAS d'utiliser au maximum 12 caractères pour présenter la modalité.

Vous notez aussi la présence du signe étoile, '*' option possible d'INPUT mais peu commentée dans l'aide et utilisable dans des cas très particuliers¹. Nous demandons ici la production de 18 '-' avant la présentation de X formaté via EUROX12.2, puis la production de 5 nouveaux signes '-'. Enfin, lorsque vous citez une variable suivie du signe égal, le nom de la variable et ce signe sont repris avant la production de la modalité.

Utilisée seule, l'instruction PUT vous sera particulièrement utile si vous souhaitez connaître, à un moment donné, la valeur prise par telle ou telle variable à l'intérieur d'un programme. Elle vous permettra ainsi de connaître le contenu du PDV à chaque étape du déroulement d'un programme (voir section 3.1).

Si vous introduisez une instruction FILE, le résultat de PUT n'est plus envoyé dans la fenêtre JOURNAL mais dans un fichier TXT extérieur à SAS. Vous pouvez donc aisément exporter une table dans un format relativement universel. Imaginons que vous disposiez d'une table de *n* individus pour lesquels 100 variables X1-X100 sont renseignées. Vous souhaitez créer un fichier qui contiendra uniquement les variables X1-X10.

Programme 2.62

```
DATA _NULL_ ;
SET test;
FILE 'C:\intro_SAS\chiers\extract.txt';
PUT x1-x10;
RUN;
```

Il n'est pas forcément utile d'écrire ici une nouvelle table SAS qui présenterait les caractéristiques du fichier que vous souhaitez créer. L'instruction DATA _NULL_ vous permet de tirer profit de l'ensemble des outils de programmation de l'étape DATA sans avoir besoin de créer une nouvelle table SAS.

Vous devez indiquer par l'instruction FILE l'endroit de votre disque dur où votre fichier doit être placé. L'instruction PUT liste ici les variables que vous souhaitez voir dans votre fichier.

Dans le fichier créé par le programme 2.62, les modalités sont séparées par des espaces. Cependant, rien ne vous empêche d'utiliser certaines des options vues avec les instructions INFILE et INPUT pour donner à votre fichier TXT la forme voulue et permettre une ouverture plus facile avec votre second logiciel.

Vous pouvez retirer les FORMATS qui sont associés à certaines de vos variables¹ dans la table originale grâce à l'instruction FORMAT _ALL_?. Mais cela ne conviendra pas toujours, notamment pour les dates : si SAS utilise comme référence le 1^{er} janvier 1960, ce n'est pas forcément le cas du logiciel auquel vous destinez votre fichier TXT. Il peut donc être utile de maintenir le FORMAT existant, voire d'imposer un FORMAT particulier pour être sûr que votre modalité sera correctement saisie par votre second logiciel. Par exemple, si votre date est présentée, dans SAS, au moyen d'un FORMAT MMDYY10. (affichage des dates « à l'américaine ») et que vous conserviez ce FORMAT dans le fichier d'exportation. Excel ne comprendra pas qu'il s'agit d'une date et considérera cette dernière comme une chaîne de caractères (et vous ne pourrez rien en faire). En revanche, si vous imposez un autre FORMAT (comme DIDMY10.), votre modalité sera parfaitement interprétée comme une date dans Excel.

Programme 2.63

```
DATA _NULL_ ;
SET test;
FILE 'C:\intro_SAS\chiers\extract.txt';
PUT x1 : DIDMY10. x2-x10;
RUN;
```

¹ Les FORMATS sont présentés en détail dans la section 5.3.

² ALL est une liste automatique de variables qui permet d'appeler toutes les variables présentes dans une table. Pour