

Résumé des possibilités des macros dans SAS

Partie 1

par
Louis-René Rheault

Date: 13 mai 2026

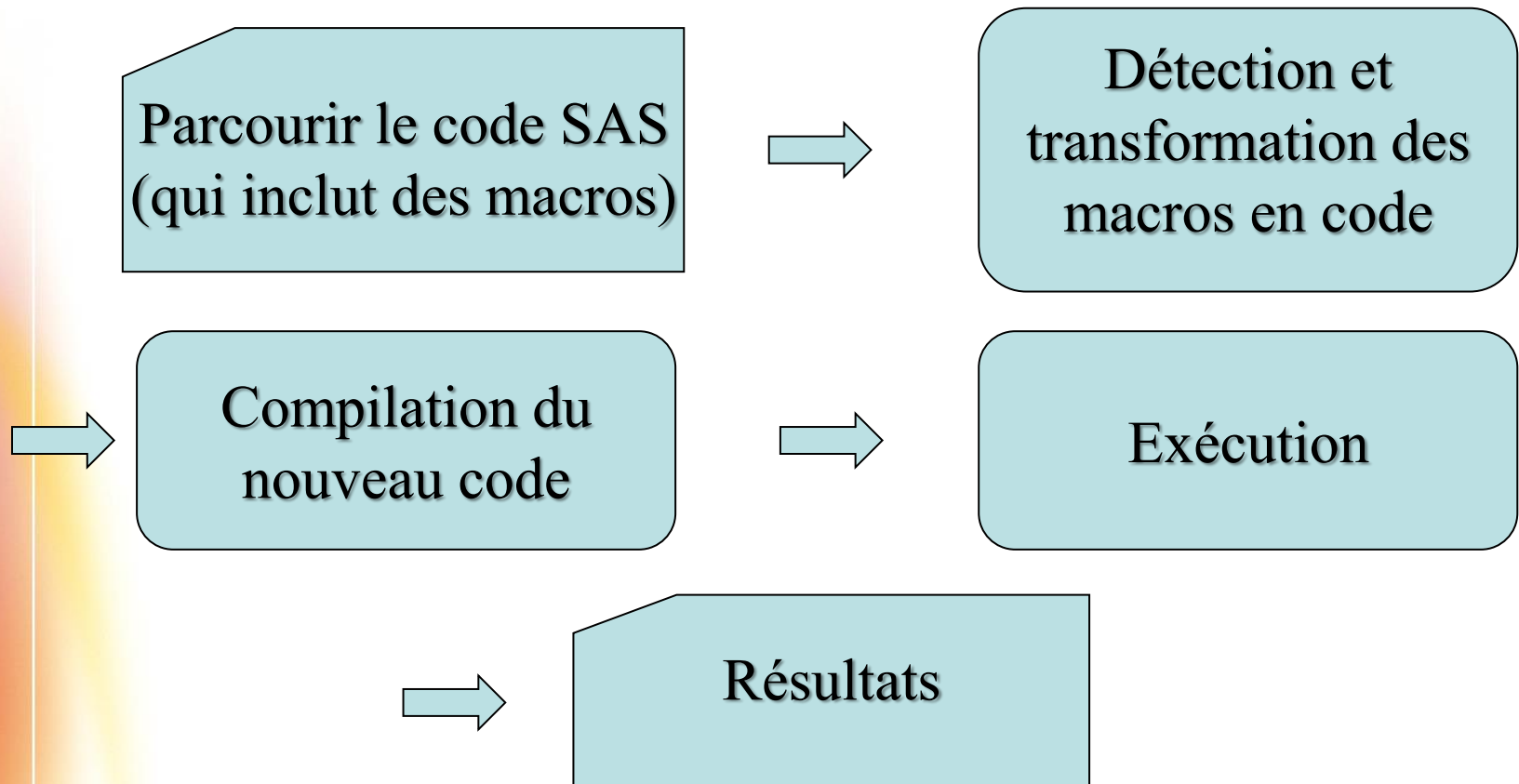
Plan de la présentation

1. Les macros, qu'est-ce que c'est?
2. Différents types de macro
3. Définir des macros-variables (partie 1)
4. Définir des routines (fonctions) macro (partie 2 cet après-midi)

Les macros, qu'est-ce que c'est?

- Le langage macro est un langage de programmation qui sert à remplacer et à générer automatiquement du code dans les programmes SAS. Il fait partie de SAS Base.
- Avant qu'un programme soit exécuté, le langage macro est compilé par SAS. Lorsque SAS repère des macros, elles sont interprétées (on dit aussi « résolues ») par le « *SAS Macro facility* » et transformées en code SAS. Par la suite, le programme « modifié » est compilé et exécuté par SAS.

Les macros, qu'est-ce que c'est?



Les différents types de macro

- Macro-variables
 - Prédéfinies dans SAS
 - Définies par l'utilisateur

- Routines (fonctions) macro
 - Fournies par SAS
 - Définies par l'utilisateur

Définir des macros-variables

- Macro-variables
 - Prédéfinies dans SAS

Dans SAS, plusieurs informations sont disponibles en tout temps, stockées dans une macro-variable:

La date du jour où la session SAS a commencé → &sysdate

L'heure où la session SAS a commencé → &systeme

La version de SAS utilisée → &sasver

Le nom d'utilisateur Windows → &_clientuserid

Le dossier du WORK → &sasworklocation

Si la dernière exécution a généré une erreur* → &saserr

Le nom de la dernière table créée → &syslast

Définir des macros-variables

➤ Macro-variables

- Prédéfinies dans SAS

Endroits où on peut trouver la liste des macro-variables dans SAS:

En écrivant la commande

« **%PUT _automatic_ ;** »*

Dans l'aide de SAS, dans la section

→ SAS 9.4 and SAS Viya 3.5

→ Macro Language Reference

→ Macro Language Dictionary

→ Automatic Macro Variables

* Ou « **%PUT _all_ ;** » pour avoir toutes les macro-variables)

SAS® Help Center

☰ SAS® 9.4 and SAS® Viya® 3.5 P

▼ Macro Language Reference

Syntax Conventions for the SAS Language

What's New

> Understanding and Using the Macro Facility

▼ Macro Language Dictionary

> AutoCall Macros

> **Automatic Macro Variables**

> DATA Step Call Routines for Macros

Définir des macros-variables

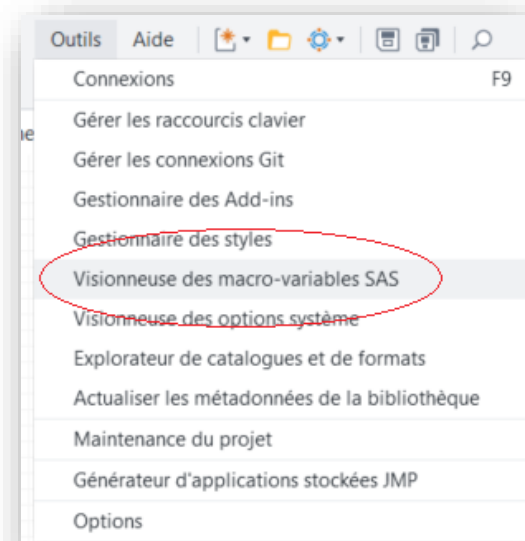
➤ Macro-variables

- Prédéfinies dans SAS

Endroits où on peut trouver la liste des macro-variables dans SAS:

Dans Enterprise Guide, dans le menu
Outils → Visionneuse de
macro-variables SAS

(*Tools → SAS macro variable viewer*)



Définir des macros-variables

➤ Macro-variables

- Prédéfinies dans SAS

Exemple d'utilisation:

```
3 DATA donnees_aujourd'hui;  
4     SET donnees;  
5  
6     IF date_production = "&SYSDATE"D THEN OUTPUT;  
7 RUN;  
8  
9 PROC SORT DATA=&SYSLAST; BY heure_production;
```

01MAY14

work.donnees_aujourd'hui

Définir des macros-variables

➤ Macro-variables

- Définies par l'utilisateur

Pourquoi?

Pour réutiliser une même valeur tout au long d'un programme, d'un projet ou même plus!

Ou pour qu'elle change de façon dynamique.

Note:

Vous aurez remarqué que

lorsqu'on appelle une macro-variable, elle est

toujours précédée par un E commercial (perluète) → &

Définir des macros-variables

- Macro-variables
 - Définies par l'utilisateur

4 principales méthodes pour assigner une valeur à une variable macro:

- La commande %LET
- La routine CALL SYMPUTX
- La procédure PROC SQL
- Dans Enterprise Guide, créer une invite (*prompt*)

Définir des macros-variables

- Macro-variables
 - Définies par l'utilisateur

- La commande %LET

En écrivant **%LET *nom_mamacrovariable* = LouisRene;**

vous pourrez appeler la variable **&*nom_mamacrovariable***

et SAS l'interprétera toujours comme étant **LouisRene**

*%LET traité à la compilation, avant l'exécution.
Jamais de & devant le nom au moment de l'assignation.*

Définir des macros-variables

- Macro-variables
 - Définies par l'utilisateur
- La routine CALL SYMPUTX

Dans une étape DATA, vous pouvez stocker la valeur d'une variable à l'aide de l'appel:

```
CALL SYMPUTX("nom_variable_macro",  
              variable_contenant_valeur);
```

Jamais de & devant le nom au moment de l'assignation.

CALL SYMPUTX est traité à chaque itération de l'étape DATA.

Si on veut créer une seule variable, la mettre entre apostrophes ou guillemets

Définir des macros-variables

- Macro-variables
 - Définies par l'utilisateur
- La procédure PROC SQL

Dans une procédure PROC SQL, il est possible de stocker le résultat d'une requête dans une variable macro:

PROC SQL;

```
SELECT MAX(date_production)  
INTO :derniere_production
```

```
FROM donnees;
```

QUIT;

Le résultat du SELECT ne doit contenir qu'une seule valeur

INTO traité au moment de l'exécution.

Jamais de & devant le nom au moment de l'assignation.

Dans un PROC SQL, c'est le : qui doit précéder le nom de la variable pour l'assignation.

Définir des macros-variables

- Macro-variables
 - Définies par l'utilisateur
- Dans Enterprise Guide (ici version 8.6)

Pour la définir:

Gestionnaire d'invites

+ | - | ✎

Cliquez sur Ajouter pour créer une nouvelle invite.

Ajouter une invite

Général Type d'invite et valeurs

Nom :
ma_macrovariable

Texte affiché :
Le nom complet de ma macro-variable

Description :
Une description détaillée

Options

Masquer à l'exécution Valeur non vide requise

Valeurs en lecture seule Utiliser la valeur de l'invite dans tout le projet

OK Annuler Aide

Définir des macros-variables

Gestionnaire d'invites

+ | |

Cliquez sur Ajouter pour créer une nouvelle invite.

Ajouter une invite

Général | Type d'invite et valeurs

Nom :
ma_macrovariable

Texte affiché :
Le nom complet de ma macro-variable

Description :
Une description détaillée

Options

Masquer à l'exécution Valeur non vide requise

Valeurs en lecture seule Utiliser la valeur de l'invite dans tout le projet

OK Annuler Aide

Pas de & devant le nom de la macro-variable dans la fenêtre de configuration

Définir des macros-variables

The image shows a software interface with two windows. The background window is titled 'Gestionnaire d'invites' and contains a plus icon, a trash icon, and an edit icon. Below these icons is the text: 'Cliquez sur Ajouter pour créer une nouvelle invite.' The foreground window is titled 'Ajouter une invite' and has a close button (X) in the top right corner. It contains the following fields and controls:

- Général** | **Type d'invite et valeurs**
- Type d'invite :
- Méthode d'alimentation de l'invite :
- Nombre de valeurs :
- Type de texte :
- Longueur min. :
- Longueur max. :
- Inclure les valeurs spéciales:
 - Toutes les valeurs possibles
 - Valeurs manquantes
- Valeur par défaut :
- Astuce :

At the bottom of the dialog are three buttons: **OK**, **Annuler**, and **Aide**.



Propriétés de Mon programme

Général
Résultats
Soumission du code
Invites 1
Récapitulatif

Invites

Invites du projet utilisées :

Nom SAS	Afficher le nom	Type de données
Cet élément n'utilise aucune des invites définies dans le projet.		

Sélectionner des invites

Sélectionnez les invites supplémentaires à utiliser :

3 ma_macrovariable

2 Ajouter... Supprimer Gestionnaire d'invites... OK Annuler

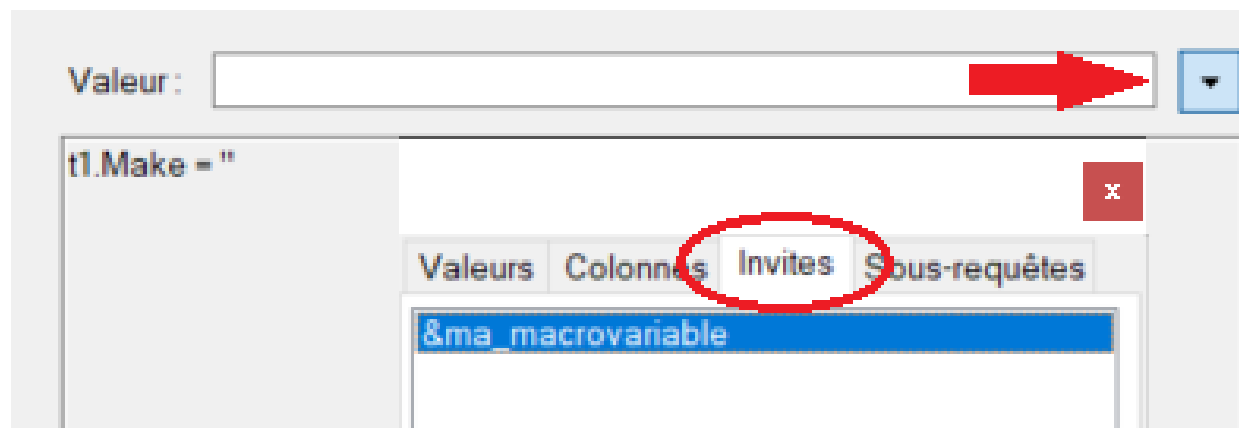
Définir des macros-variables

➤ Macro-variables

- Définies par l'utilisateur dans Enterprise Guide

Pour l'utiliser:

- Dans une requête (dans un filtre) →



Traité à la compilation, avant l'exécution.

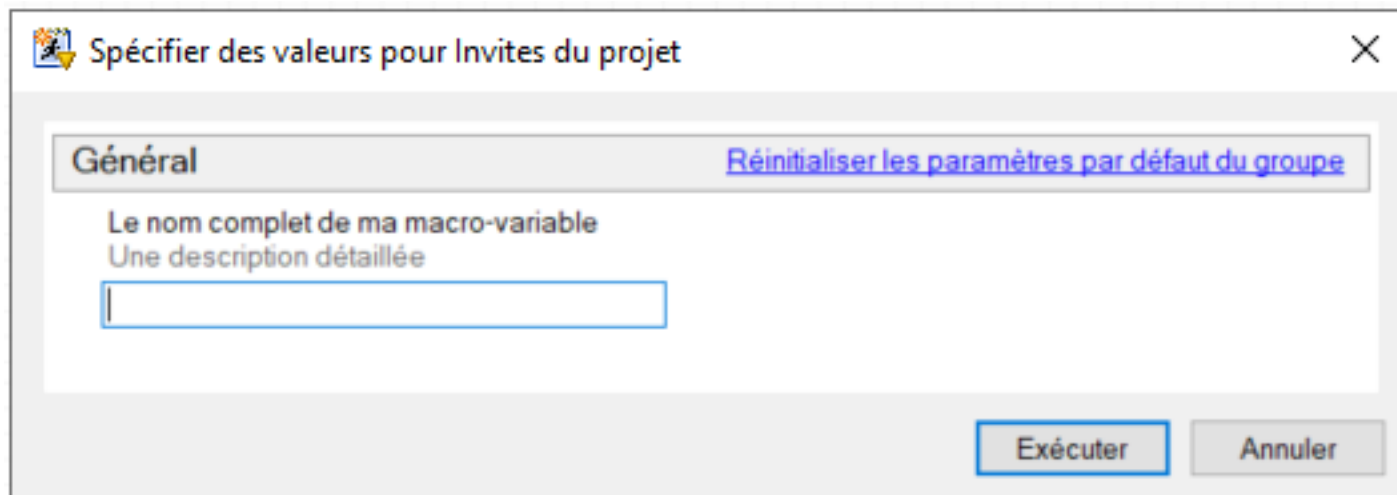
Définir des macros-variables

➤ Macro-variables

- Définies par l'utilisateur dans Enterprise Guide

Pour l'utiliser:

- À l'exécution, une boîte de dialogue s'ouvre:



The screenshot shows a dialog box with the title "Spécifier des valeurs pour Invites du projet". It has a close button (X) in the top right corner. The dialog is divided into a "Général" tab. In the "Général" tab, there is a text input field with the label "Le nom complet de ma macro-variable" and a sub-label "Une description détaillée". To the right of the input field is a link that says "Réinitialiser les paramètres par défaut du groupe". At the bottom right of the dialog, there are two buttons: "Exécuter" and "Annuler".

Définir des macros-variables

➤ Macro-variables

Une fois attribuée, la valeur d'une variable macro devient disponible dans le programme où elle est créée.

Attention :

- Le nom d'une variable doit commencer par une lettre ou un trait souligné
- Quand on veut utiliser la variable macro comme du texte, il faut utiliser les guillemets et non les apostrophes
- Certaines attributions / résolutions de macro-variables ont lieu avant l'exécution, ce qui explique parfois que la valeur de la macro-variable ne change pas en cours d'exécution

Définir des macros-variables

➤ Macro-variables

Une fois attribuée, la valeur d'une variable macro devient disponible dans la tâche (programme ou autre) où elle est créée.

Attention :

- le contenu d'une variable n'est qu'une suite de caractères. C'est au moment de son traitement qu'elle sera interprétée (comme un nombre, des caractères, une date, une fonction, ...)

Des questions?

**La suite (macro-fonctions)
cet après-midi**

Résumé des possibilités des macros dans SAS

Partie 2

par
Louis-René Rheault

Date: 13 mai 2026

Définir des routines macro

➤ Routines macro

Une routine macro sert à « encapsuler » une série d'énoncés (procédures, étapes DATA ou autres) qui pourront être appelées simplement en appelant le nom de la routine.

Par exemple:

%exporter_excel;

qui pourra servir à exporter la dernière table créée vers Excel.

Définir des routines macro

➤ Routines macro

Cela permet de réutiliser souvent du code très semblable sans avoir à le recopier chaque fois.

Cela permet aussi :

- d'intégrer des paramètres
- des traitements conditionnels
- des itérations / boucles

Définir des routines macro

➤ Routines macro

De la même manière que les macro-variables, les routines macro sont compilées puis interprétées par le SAS Macro Facility pour les remplacer par du code SAS. Le compilateur SAS compile le tout et l'exécution du programme peut être effectuée.

Définir des routines macro

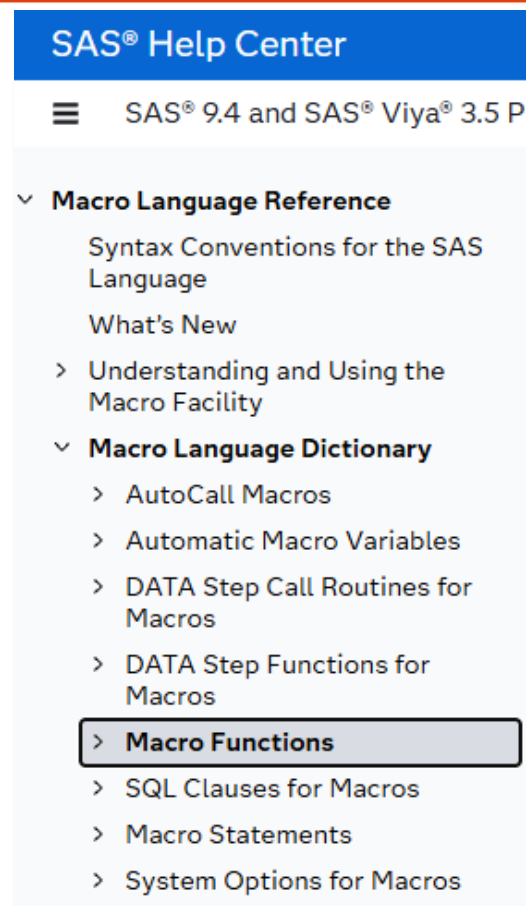
- Routines macro
 - Fournies par SAS

SAS permet d'utiliser une panoplie de routines macro déjà créées. Souvent, d'autres utilisateurs de SAS publient sur les forums ou sites Web leurs propres routines macro qui leur sont utiles.

Définir des routines macro

- Routines macro
 - Fournies par SAS

Il est possible de trouver une liste des routines macro fournies par SAS dans l'aide:



Définir des routines macro

- Routines macro
 - Définies par l'utilisateur

Un exemple de routine simple pourrait permettre d'exporter la dernière table créée vers Excel:

```
5  %MACRO exporter vers Excel;  
6      PROC EXPORT DATA= &SYSLAST  
7          OUTFILE= "c:\Productions quotidiennes\Fichier du &SYSDATE..xlsx"  
8          DBMS=Excel2007 REPLACE;  
9          SHEET = "Données";  
.0      RUN;  
.1  
.2  %MEND exporter_vers_Excel;  
.  
.3  
.4  %exporter_vers_Excel; → L'utilisation  
.5
```

La définition

Définir des routines macro

- Routines macro
 - Définies par l'utilisateur

Dans les routines macro, il est possible de passer des **paramètres** s'ils sont définis lors de la création:

```
13 %MACRO exporter_vers_Excel (nom_table = , nom_onglet = );
14     PROC EXPORT DATA= &nom_table
15         OUTFILE= "c:\Productions quotidiennes\Fichier du &SYSDATE..xlsx"
16         DBMS=Excel2007 REPLACE;
17         SHEET = "&nom_onglet";
18     RUN;
19 %MEND exporter_vers_Excel;
20
21 %exporter vers Excel(nom table = work.autres donnees, nom onglet = Données quotidiennes);
```

Définir des routines macro

- Routines macro
 - Définies par l'utilisateur

Dans les routines macro, il est possible de vérifier certaines conditions:

```
%MACRO exporter_vers_Excel (nom_table = , nom_onglet = );  
  /* Vérifier si la table est vide */  
  PROC SQL NOPRINT;  
    SELECT COUNT(*) INTO :nb_lignes FROM &nom_table;  
  QUIT;  
  
  %IF &nb_lignes GT 0 %THEN %DO;  
    PROC EXPORT DATA=&nom_table  
      OUTFILE = "c:\Productions quotidiennes\Fichier du &SYSDATE..xlsx"  
      DBMS=Excel2007 REPLACE;  
      SHEET = "&nom_onglet";  
    RUN;  
  %END;  
%MEND exporter_vers_Excel;  
  
%exporter_vers_excel(nom_table = work.mes_donnees_finales, nom_onglet = Donnees);
```

Définir des routines macro

```
%MACRO exporter_vers_Excel (nom_table = , nom_onglet = );  
  /* Vérifier si la table est vide */  
  PROC SQL NOPRINT;  
    SELECT COUNT(*) INTO :nb_lignes FROM &nom_table;  
  QUIT;  
  
  %IF &nb_lignes GT 0 %THEN %DO;  
    PROC EXPORT DATA=&nom_table  
      OUTFILE = "c:\Productions quotidiennes\Fichier du &SYSDATE..xlsx"  
      DBMS=Excel2007 REPLACE;  
      SHEET = "&nom_onglet";  
  
    RUN;  
  %END;  
%MEND exporter_vers_Excel;  
  
%exporter_vers_excel(nom_table = work.mes_donnees_finales, nom_onglet = Donnees);
```

Mêmes opérateurs
conditionnels mais
avec le %

Définir des routines macro

- Routines macro
 - Définies par l'utilisateur

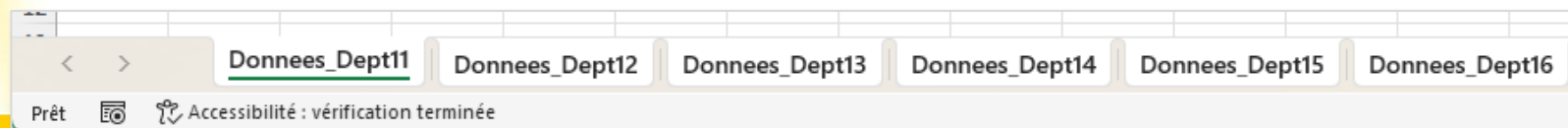
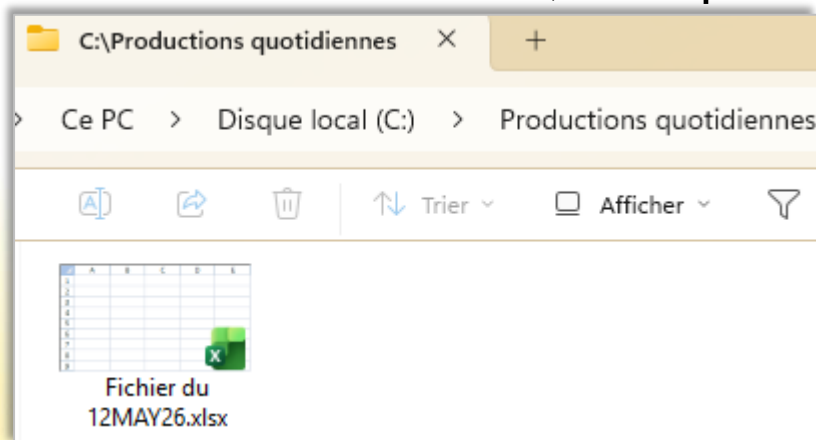
Dans les routines macro, il est possible d'exécuter des itérations:

```
%MACRO exporter_vers_Excel (nom_table = , nom_onglet = );  
  %DO i=1 %TO 15;  
    PROC EXPORT DATA=&nom_table (WHERE=(Departement EQ &i))  
      OUTFILE = "c:\Productions quotidiennes\Fichier du &SYSDATE..xlsx"  
      DBMS=Excel2007 REPLACE;  
      SHEET = "&nom_onglet Dept&i";  
  RUN;  
  %END;  
%MEND exporter_vers_Excel;  
  
%exporter_vers_excel(nom_table = work.mes_donnees_finales, nom_onglet = Donnees);
```

Définir des routines macro

- Routines macro
 - Définies par l'utilisateur

Dans les routines macro, il est possible d'exécuter des itérations:



Définir des routines macro

- Routines macro
 - Définies par l'utilisateur

Il ne faut pas oublier d'exécuter la macro après l'avoir définie...

NB:

Vous aurez remarqué que...
les commandes utilisées dans le langage macro commencent presque toutes par des %.

Ce sont les & et les % qui permettent à SAS de détecter l'utilisation des macros, d'appeler le SAS Macro Facility pour compiler et résoudre les macros en code pour ensuite redonner le contrôle au compilateur de code SAS.

Conclusion

- Le langage macro dans SAS est un outil extrêmement flexible et puissant
- Nous n'avons effleuré qu'une petite partie de la surface de l'iceberg!
- Beaucoup de documents, conseils, astuces et formations sont disponibles.

**Des
questions?**

Merci beaucoup!

Louis-René Rheault

Université Laval

Louis-Rene.Rheault@vrex.ulaval.ca