

EpiData

Fichier d'aide

© Jens M. Lauritsen, Michael Bruus & Mark Myatt
Février 2001

Traduction française réalisée par les membres de l'association



coordonnée par



Table des matières

Table des matières	2
Introduction	4
Au sujet d'EpiData	5
Financement et remerciements	6
Enregistrement	8
But d'EpiData	8
Appui	8
Comment analyser des données après la saisie	9
Informations sur les versions	9
Fonctionnalités dans cette version d'EpiData	12
Compatibilité avec EpiInfo	14
Editeur.....	16
L'assistant champ	16
Éditeur de code	16
Prévisualisation du masque de saisie	17
Noms de variables	19
Premier mot comme nom de champ.....	19
Noms des champs: Automatique	20
Indentation automatique	21
Aligner les champs	21
Créer le fichier de données.....	22
Réviser un fichier de données	22
Contrôle de saisie	24
Ajoute/Révisé des Contrôles de saisie.....	26
Range / Legal : étendue, valeurs autorisées.	27
Jumps : Sauts.....	27
MUST ENTER : Saisie obligatoire.....	28
REPEAT : Répéter la valeur de l'enregistrement précédent	28
VALUE LABELS : Étiquettes de valeur.....	28
Éditez tous les contrôles pour la variable courante	30
Duplication des commandes de contrôle	31
Effacer tous les contrôles.....	31
Structure du fichier de contrôle (.CHK)	32
Exemple d'un fichier CHK.....	33
Liste des commandes de contrôle	36
AFTER ENTRY	36
AFTER FILE	36
AFTER RECORD	36
AUTOJUMP.....	37
AUTOSAVE	37
BEFORE ENTRY	37
BEFORE FILE	37
BEFORE RECORD	38
COMMENTS (*).....	38
COMMENT LEGAL	38
CONFIRM.....	39
CONFIRMFIELD	39
DEFINE	40
EXIT	40
HELP	41
HIDE, UNHIDE	41
IF	41
JUMPS	42
KEY	43
LABEL	43

LABELBLOCK	44
LEGAL	44
LET	44
MUSTENTER	45
NOENTER	45
RANGE	45
REPEAT	46
TYPE	46
TYPE COMMENT	46
TYPE STATUSBAR	47
UNHIDE	47
Opérateurs et fonctions	48
Opérateurs	48
Fonctions arithmétiques	49
Fonctions chaînes	50
Fonctions dates	51
Autres fonctions	52
Saisir des données	53
Navigation entre les champs	53
Navigation entre les enregistrements	53
Trouver des enregistrements	54
Documenter	55
Notes de saisie	55
Libellé du fichier de données	55
Liste des données	55
Table des codes	56
Double entrée et validation	57
Exportez les données	58
Sauvegarde des données	58
Exportation vers le texte-dossier	58
Export au format dBase III	58
Export Excel	59
Exportation vers Stata	59
Options	61
Barres d'utilis	63
Clefs de raccourci	64
Programmez les paramètres	66
Internationalisation	67
Format de champs EpiData	68
Numérique	68
Identifiant	69
Champs numériques	69
Champs texte	69
Champs texte majuscule	69
Champs booléens (oui/non ou vrai/faux)	69
Champs Date	70
Champs date du jour	71
Champs de Soundex	71
Marque de tabulation	72

Introduction

EpiData est un programme de saisie de données. Le but du programme est d'aider à la création de versions informatiques de questionnaires d'enquêtes, en facilitant la saisie. L'idée fondamentale est d'aider à la création de fichiers de données brutes de bonne qualité pour l'analyse ultérieure. Ceci est fait en définissant exactement quel genre de données est prévu dans chacun des champs (variables) du questionnaire, s'assurant par exemple que seulement les nombres, et non les lettres, peuvent être entrés dans les champs numériques. EpiData facilite également la définition de règles de validation pour chaque champ, par exemple un champ de numérique peut seulement n'être renseigné qu'avec certaines valeurs (1.2 ou 3 par exemple).

EpiData fonctionne avec trois types de fichiers de base:

- 1) fichiers .QES (abréviation pour fichiers 'questionnaire'), qui définissent la structure du questionnaire.
- 2) fichiers .REC (abréviation pour fichiers 'données'), qui contiennent les données plus une définition codée du questionnaire.
- 3) fichiers .CHK (abréviation pour fichiers de contrôles), qui contiennent les règles de validation définies pour les champs de saisie

Pour afficher l'introduction à EpiData et pour voir des exemples sur la façon de définir un questionnaire et la façon de saisir les données, reportez vous à EpiTour, à partir de l'option "Aide" du menu d'EpiData.

Au sujet d'EpiData

Design:

Jens M. Lauritsen, comté de Funen, Danemark.

Michael Bruus, Danemark

Mark Myatt, Brixton Health, Pays de Gales et Institut d'Ophtalmologie, Londres, R-U

Programmation

Michael Bruus, Danemark.

Financement:

Voir Remerciements

Suggestion de citations:

Lauritsen JM, Bruus M, Myatt M. EpiData, version 1.5. Un outil de base pour la saisie validée et la documentation des données. Comté de Funen Danemark et Brixton Health R-U, 2000.

N'hésitez pas à distribuer des copies du programme.

Reportez vous au site <http://www.EpiData.dk> pour toute information sur les mises à jour, les bogues connus etc., et les versions PDF du fichier d'aide, d'EpiTour, etc.

Le logiciel EpiData a été développé et testé pour assurer une saisie et une documentation fiable des données. Tous les efforts ont été faits pour produire un programme fiable, mais ne pouvons en aucune circonstance être tenus pour responsable d'éventuelles erreurs, perte de données ou de temps de travail ou d'autres pertes encourues par ou à cause du programme.

Financement et remerciements

Financement

Nous remercions les organismes suivants pour leur contribution au développement d'EpiData:

	Comté de Funen, Danemark	http://www.fyns-amt.dk
	Santé De Brixton, R-U	http://www.brixtonbooks.demon.co.uk
	Danish Data Archives/ERAS, Danemark	http://www.dda.dk
	Université du Danemark méridional, faculté de santé - Odense.	http://www.sdu.dk/indexE.html
	Valid International, R-U	http://www.validinternational.org/
	London School of Hygiene & Tropical Medicine, Londres, R-U	http://www.lshtm.ac.uk/
	Centre international pour la santé oculaire, R-U	http://www.ucl.ac.uk/iao/ircpb.htm
	Université d'Aarhus, Faculté des sciences de santé	http://www.au.dk

Contributions

La majeure partie de développement d'EpiData a été financée, mais si vous souhaitez soutenir le développement ultérieur de la documentation, de la programmation, d'une meilleure gestion de la vitesse, la maintenance du site web, n'hésitez pas à entrer en contact à l'adresse mail suivante : info@EpiData.dk

Remerciements

Les idées et les principes utilisés dans EpiData sont basés sur:
Dean AG, Dean JA, Coulombier D, Brendel Ka, Smith DC, Burton AH, Dicker RC, Sullivan K, Fagan RF, Arner TG. Epi Info, Version 6: Traitement de textes, base de données et programme de statistiques pour la santé publique sur micro-ordinateurs compatibles IBM. Centers for Disease Control and Prevention, Atlanta, Géorgie, États-Unis, 1995. Voir <http://www.cdc.gov/epo/epi/epiinfo.htm>.

Le code source d'Epi Info nous a été fourni par le CDC. Certaines parties d'EpiData sont basées sur ce code source.

Ni l'OMS ni le CDC ne peuvent accepter de responsabilités pour EpiData ou pour son support.

L'exportation vers Excel est basée sur une librairie Delphi gratuite, écrite par Eddy Sterckx.

L'analyse et l'interprétation des calculs, des conditions et des expressions IF sont basées sur une librairie Delhi gratuite écrite par Martin Lafferty, modifiée par Michael Bruus pour l'adapter à EpiData.

Enregistrement

Enregistrez vous comme utilisateur d'EpiData en remplissant le formulaire d'enregistrement disponible sur www.epidata.dk. Utilisez la fonction 'Enregistrement en ligne' depuis le menu d'aide d'EpiData.

Si vous ne disposez pas d'une connexion directe à Internet, envoyez un E-mail à info@epidata.dk en spécifiant "epidata registration" comme sujet:. Dans le corps du message, entrez votre adresse E-mail ainsi que votre pays de résidence.

Tous les utilisateurs enregistrés recevront une information sur les mises à jour et les changements principaux liés à des corrections de bogues. Certains utilisateurs volontaires pourront également être mis à contribution pour participer aux décisions sur les nouvelles fonctionnalités à inclure dans des prochaines versions.

Vos adresses de E-mail ne seront pas employés pour d'autres buts et ne seront pas cédées à qui que ce soit.

But d'EpiData

EpiData est un programme fonctionnant sous Windows 9x pour:

- La définition des structures de données
- La saisie simple
- L'utilisation de principes de validation
- L'édition de données préalablement entrées
- L'impression ou le listage des données pour la documentation des contrôles
- L'exportation des données pour l'analyse avec des programmes d'analyse statistique

Appui

Les principes de base de la création de formulaires de questionnaires, de la saisie de données, de la création de règles de contrôle etc...est similaire à ce qui est décrit dans le manuel Epi Info, disponible à l'adresse suivante : <http://www.cdc.gov/epo/epi/intro/manual/mantoc.htm>.

Un manuel pour EpiData est en préparation.

Une version PDF imprimable (avec le logiciel Acrobat Reader) du fichier d'aide peut être trouvé sur le site www.EpiData.dk.

Malheureusement nous n'avons pas les ressources pour fournir une aide directe.

Si vous souhaitez nous faire part d'erreurs ou de bogues à l'utilisation du programme ou si vous souhaitez nous faire part de suggestions visant à améliorer EpiData, envoyez un e-mail à Info@EpiData.dk. Une liste de bogues décrits est régulièrement tenue à jour sur <http://www.EpiData.dk>

Indiquez:

Description du problème. Est-il répliquable? Est-il apparu avec différents fichiers/structures de données ou seulement sur un fichier en particulier? L'erreur est-elle répliquable sur un PC différent? Sous quel système d'exploitation fonctionne la machine sur laquelle le problème apparaît ? De combien d'espace disque libre

dispose cette machine ? Quelle est la version d'EpiData utilisée? Indiquez une adresse e-mail de réponse au cas où nous puissions vous faire part de suggestions.

Comment analyser des données après la saisie

EpiData n'est destiné qu'à la saisie et la documentation des données, et ne peut être utilisé pour l'analyse.

Le format des fichiers de données utilisés est celui de la version 6 d'Epi Info. Les exceptions à cette règle sont décrites dans compatibilité avec Epi Info

Epi Info peut être utilisé directement pour l'analyse, sans aucune exportation. De plus, EpiData vous permet d'exporter les enregistrements dans un fichier texte, à champs séparés par des virgules, un fichier dBase III, un fichier Excel ou un fichier Stata. Il est aussi possible d'utiliser StatTransfer (voir <http://www.circlesys.com>), DBMS/Copy ou le module d'exportation d'Epi Info.

Des programmes d'analyse spécifiques (analyse de survie ou analyse de régression) fonctionnant avec des données au format Epi Info ont été développés. Pour plus d'informations, voir <http://www.cdc.gov/epo/epi/epiinfo.htm>

Informations sur les versions

Version 1.5 d'EpiData
diffusée le 22 février 2001

Diffusions précédentes :

La version 1.3 a été diffusée le 15 janvier 2001

La version 1,2 a été diffusée le 22 novembre 2000

La version 1.01 a été diffusée le 11 octobre 2000

La version 1.0 a été diffusée le 12 septembre 2000

Changements de la version 1,3 à la version 1,5

Nouvelles commandes check:

GOTO WRITEREC

TYPE COMMENT *nom_champ* (remplace le CODES/CODEFIELD d'Epi Info)

TYPE " mon propre texte " [*couleur*]

WeekNum(date)

DayOfWeek(date)

Autres additions

Option de choix de la langue

Imprimer la sélection (éditeur)

Fonction d'alignement des champs de saisie dans l'éditeur

De nombreux petits bogues ont été corrigés.

Changements de la version 1.2 à la version 1.3

Nouvelles commandes check:

Legal use *nom_champ*

Before file, after file, before record, after record

Exit

GOTO WRITE

DEFINE (e.g. DEFINE TempVariable ####)

Autosave

Confirm

Confirmfield

Nouvelles fonctions de commandes check:

Constantes de date (par exemple D1="14/09/2000") ajoutées

Point ajouté en tant que donnée manquante (exemple IF V1 =. THEN)

Option de couleur ajoutée à TYPE STATUSBAR et à TYPE COMMENT (exemple TYPE STATUSBAR "ID = " GREEN)

La valeur d'un champ peut être employée dans HELP (exemple HELP " V1 equals @v1")

Autres additions:

Recherche d'enregistrement lors de la saisie sur jusqu'à trois champs

Notes de saisie

Paramètres de programme / AUTOSAVE et / CONFIRM

Bogues corrigés:

Les guillemets doubles dans les conditions IF et les expressions LET ne fonctionnaient pas

Les fonctions copier / coller ne fonctionnaient pas dans les champs de saisie

Meilleure gestion des erreurs dans les fonctions d'export

Les champs booléens ne pouvaient être assignés avec LET

Plus plusieurs bogues mineurs

Changements de la version 1.01 à la version 1.2

- Utilisation d'index (KEY) ajoutée
- Validation de double saisie ajoutée
- Ajout de la Table des codes
- Ajout de Type Comment et Type Statusbar (commandes check)
- Application de Range, Legal et Jump à tous les types de champs
- Comment legal peut être employé dans les structures IF THEN (permet le codage hiérarchique)
- Ajout de la fonction de copie de la structure du fichier de données
- Amélioration de la vitesse de recherche d'enregistrements
- menu Outils ajouté
- bogues mineurs corrigés

Changements de la version 1.0 à la version 1.01

Bogues corrigés:

- Les commandes LET (sans le mot LET) ne fonctionnaient pas dans les fichiers checks avec des champs dont le nom était composé de 8 caractères.
 - les fichiers Epi Info créés avec StatTransfer ne pouvaient être lus par EpiData
 - l'exportation dans Excel transformait les champs booléens en leur affectant systématiquement la valeur VRAI
 - Les champs date européenne < dd/mm/yyyy > n'étaient pas considérés comme des dates lorsque des valeurs leur étaient assignées dans les fichiers check.
 -
- Ajout de mots-clés dans le fichier d'aide (inclusion de toutes les commandes de contrôle).
Ajout de l'information sur l'enregistrement dans le fichier d'aide

Fonctionnalités dans cette version d'EpiData

La version 1,5 d'EpiData inclut les fonctionnalités suivantes:

- Un éditeur permettant de créer ou de modifier les questionnaires (QES-files), en utilisant les fonctions chercher / remplace, copier dans / depuis le bloc-note, annuler la dernière commande.
- Une fonction d'alignement de champ facile à utiliser
- Une fonction de prévisualisation du fichier de données, permettant le test sans création du fichier de données
- Création des fichiers de données basés sur les fichiers QES
- Attribution automatique des noms des variables basée sur le texte avant la variable
- Validation de base à la saisie
- Règles de contrôle
- Création de nouvelles fiches, ou modification de fiches existant
- Exportation des fichiers de données au format séparé par des virgules ANSI, fichiers dBase III, fichiers Excel et fichiers Stata
- Compatibilité des données avec EpiInfo 6,04
- Une barre d'outils pour aider à structurer la création des fichiers de données
- Création de fichier QES à partir d'un fichier de données
- Sauvegarde des données
- Etiquettes de fichiers de données, étiquettes de variables et étiquettes de valeur
- Révision de la structure des données avec un fichier QES révisé
- Listage des données et fonctions de documentation de données améliorées
- Indexation des fichiers de données pour recherche rapide
- Double saisie et validation
- Possibilités d'utiliser des codes hiérarchiques.
- Possibilité d'utiliser différentes langues dans les menus etc.

Plan de développement:

Printemps 2001 – Sortie de la version 2, incluant:

- ❖ Saisie relationnelle.
- ❖ Fonctions de gestion des données pour fusionner et concaténer des fichiers de données
- ❖ Gestion améliorée des données manquantes, incluant la possibilité pour l'utilisateur de spécifier des codes pour les données manquantes et ignorées.
- ❖ Traitements du contrôle des données basés sur le contenu du fichier CHK (c.-à-d. des valeurs et des gammes légales), traitées comme un ensemble d'affirmations logiques semblables à la commande ASSERT pour la vérification de cohérence.
- ❖ Capacité d'employer des codes et des valeurs légales stockés dans les dossiers séparés (prolongation du COMMENT LEGAL). Ceci permettra l'utilisation de bibliothèques de codes susceptibles d'être employées dans de nombreux projets (par exemple une base de données de codes).
- ❖ Protection et restauration par accroissement / génération en utilisant les dossiers comprimés.
- ❖ La possibilité de naviguer (voir, sans possibilité de modifier) dans les fichiers de données à l'aide d'une interface du type feuille de calcul.
- ❖ L'importation des données d'autres systèmes pour permettre l'utilisation des de fonctionnalités comme le contrôle en série et les fonctions de documentation.
- ❖ EpiData restera un système pour la saisie de données, et il n'est PAS prévu d'y inclure de fonctions d'analyse dans aucune des versions planifiées. Une autre équipe a commencé le développement d'un module d'analyse. Il est probable que celui-ci inclura des commandes similaires à celles rencontrées dans les modules ANALYSIS et LOGISTIC d'Epi Info, ainsi que l'implémentation du langage 'S' pour l'analyse

statistique.

Compatibilité avec EpiInfo

EpiData, dans son idée directrice et ses principes, est basé sur le programme DOS EpiInfo créé pour l'OMS par les Centers for Disease Control and Prevention (CDC). Voir <http://www.cdc.gov/epo/epi/epiinfo.htm>.

La compatibilité des fichiers de données d'Epidata avec Epi Info (et vice versa) a été une préoccupation constante durant le développement.

Les quelques différences existant sont justifiées par le fait que certains types de champs existent dans EpiData, mais pas dans Epi Info, ou vice versa.

1. Différences dans les fichiers de données

Utilisation de fichiers de données EpiInfo avec EpiData

EpiData ne supporte pas les types de champs suivants::

- Champs 'Numéro de téléphone (Phonenum)
- Champs 'Numéro de poste' (extension de 'Numéro de téléphone)
- Les codes de couleur d'arrière plan, les champs de saisie, etc. sont ignorés par EpiData

Utilisation de fichiers de données EpiData avec EpiInfo

EpiInfo ne supporte pas les types de champs EpiData suivants::

- Date d'aujourd'hui au format européen (ddmmyyy)
- Champs Soundex
- Codes de tabulateur (@)
- Les codes de couleur d'arrière plan, les champs de saisie, etc. ne sont pas sauvegardés par EpiData

Une liste complète des champs supportés par EpiData est disponible dans [Description des types de champs](#).

Noms de champs

EpiData génère des noms de champs d'une longueur maximale de 8 caractères. Epi Info 6 supporte les noms de champs jusqu'à 10 caractères. Normalement, ce ne devrait pas poser de problème, EpiData pouvant lire les fichiers Epi Info, même si les noms de champs dépassent 8 caractères.

Si la fonction du module ENTER d'Epi Info "Réviser la structure du fichier de données en fonction d'un fichier .QES modifié" est utilisée avec un fichier de données créé avec EpiData des problèmes peuvent survenir. Le problème est qu' EpiData et Epi Info génèrent les noms de champs à partir du texte du fichier .QES, mais EpiData génère des noms de champs jusqu'à 8 caractères alors qu'Epi Info les génère jusqu'à 10 caractères. Si la fonction du module ENTER d'Epi Info "Réviser la structure du fichier de données en fonction d'un fichier .QES modifié" est utilisée, n'omettez pas d'insérer des accolades {} dans le fichier .QES pour indiquer le nom des champs. Cela permettra aux noms de champs de ne pas dépasser 8 caractères.

2. Différences dans le langage de Contrôle

- ❖ Dans les commandes IF complexes, utilisant AND ou OR, chaque partie doit être contenue dans des parenthèses.
- ❖ Les commandes suivantes ne sont pas supportées par Epi Info: Confirmfield, Labelblock, Autosave, Confirm, label.
- ❖ La commande HELP emploie une syntaxe légèrement différente dans EpiData que dans l'Epi Info
- ❖ Les codes de couleur et les coordonnées d'écran dans certaines commandes (TYPE, HELP, etc.) sont ignorées par EpiData
- ❖ Les constantes Date (LET date1="14/09/2000") doit dans EpiData avoir 10 caractères et au format européen
- ❖ EpiData emploie une syntaxe légèrement différente dans les calculs et l'expression de différentes fonctions (par exemple Copy et Pos pour des chaînes)

Editeur

Un éditeur de texte est inclus dans EpiData, dont le but essentiel est la création des questionnaires (fichiers QES). L'interface utilisateur de l'éditeur devrait être familière à la plupart des utilisateurs, car elle emploie des fonctions standard de Windows 9x.

Trois fonctions, cependant, n'existent pas dans d'autres programmes: L'assistant champ l'editeur de Code et la prévisualisation du fichier de données

L'assistant champ



L'assistant champ affiche tous les formats de champs disponibles dans EpiData. Quand la liste de sélection est ouverte, vous pouvez choisir un type de champ qui sera inséré à la position actuelle du curseur dans la fenêtre de l'éditeur. Un type de champ est choisi en sélectionnant la page contenant le type voulu et en plaçant les propriétés du champ suivi d'un clic sur 'Insérer' ou par la touche Entrée.

L'assistant peut être ouvert

- par la combinaison CTRL-Q
- En cliquant sur le bouton 'Assistant champ' de la barre d'outils de l'éditeur
- En sélectionnant 'Assistant champ' dans le menu Editer

L'appui sur CTRL-Q quand la liste de sélection est affichée bascule entre la fenêtre de l'éditeur et la fenêtre de l'assistant champ.

Pour fermer la fenêtre du sélecteur :

- Cliquer l'icône fermer (x)
- Appuyer sur Ctrl+F4 quand l'assistant est affiché

Éditeur de code

L'éditeur de code est une fonction d'aide à la rédaction des codes définissant les champs. Si l'éditeur de code est enclenché, certaines séquences de frappe seront interprétées comme le début d'un champ, et l'éditeur de code terminera le code ou demandera des informations relatives à la longueur du champ avant de mettre à jour le code dans le fichier QES.

Si par exemple vous entrez le caractère #, l'éditeur de code interprètera cela comme le début d'un champ numérique, dont il vous demandera la longueur. Dès que la longueur est entrée, le champ est inséré à l'endroit où se trouve le pointeur.

Les combinaisons de caractères suivantes sont gérées par l'éditeur de code:

#	Champ numérique. L'utilisateur est interrogé sur la longueur du champ. Entrez 5 pour obtenir un champ stockant un entier de 5 espaces. Entrez 5.2 ou 5,2 pour obtenir un champ stockant 5 chiffres avant la virgule et 2 chiffres après la virgule(#####.##)
_	Champ texte. . L'utilisateur est interrogé sur la longueur du champ
<A	Champ texte majuscule. . L'utilisateur est interrogé sur la longueur du champ La casse de la lettre A n'importe pas.
<d	Date au format européen. Insertion de <dd/mm/yyyy>.
<m	Date au format américain. Insertion de <mm/dd/yyyy>
<y	Champ booléen. <Y> sera inséré
<i	Insertion d'un numéro ID automatique ID. . L'utilisateur est interrogé sur la longueur du champ. Longueur par défaut (et plus petite longueur) : 5.
<s	Champ soundex. . L'utilisateur est interrogé sur la longueur du champ

Enclenchement et déclenchement de l'éditeur de code :

- par la séquence CTRL-W
- en cliquant sur le bouton 'Editeur de code' sur la barre d'outils de l'éditeur.
- en choisissant Editeur de code dans le menu Editer-

L'appui sur CTRL-Q pour ouvrir la liste de sélection de champ arrêtera l'éditeur de code et vice versa.

Prévisualisation du masque de saisie

Cette fonction affiche le masque de saisie tel qu'il se présentera, sans créer de fichier de données.

Les champs montrés dans la prévisualisation se comportent de la même manière que lors de la saisie de données, donnant une impression réaliste de la façon dont le questionnaire fonctionne. Notez que les fonctions de contrôle ne sont pas appliquées dans la prévisualisation, car aucun fichier de données n'est créé.

Il n'est pas nécessaire de fermer la fenêtre de prévisualisation pour lancer une nouvelle prévisualisation.

Quand un questionnaire est affiché dans une fenêtre de l'éditeur, la prévisualisation peut être lancée par

- la séquence CTRL-T
- en cliquant sur le bouton 'Voir le masque' sur la barre d'outils de l'éditeur
- en cliquant sur 'Voir le masque' dans le menu déroulant de l'éditeur
- en choisissant Créer fichier de données | Voir le masque.

Noms de variables

Les noms de variables sont créés automatiquement dans le masque de saisie à partir du texte du contenu du fichier .QES.

Les noms de variables peuvent être générés de 2 façons avec EpiData.

- 1) Le premier mot de la question (le texte à la gauche du champ) est utilisé comme nom de variable
- 2) Création automatique des noms de variables de la même façon qu'avec EpiInfo

Le choix de l'un ou de l'autre des systèmes dépend de la sélection se fait à partir de l'option Créer Fichier de données (menu Fichiers | Options).

Attention: EpiData crée des noms de variables d'une longueur maximale de 8 caractères. Epi Info 6 les crée avec une longueur maximale de 10 caractères. Cela ne devrait pas représenter un problème, car EpiData est capable de lire les fichiers Epi Info avec des noms de variables de 10 caractères. Il se peut néanmoins que des problèmes se posent à l'utilisation de l'option 'Réviser la structure du fichier de données' d'Epi Info. Voir [Compatibilité avec Epi Info](#)

Premier mot comme nom de champ

Si l'option "Nom des champs: premier mot" est sélectionnée dans 'créer fichier de données (menu fichier | options), le nom des variables sera créé en utilisant le premier mot de la zone de texte à la gauche du champ. Si ce mot fait plus de 8 caractères, seuls les 8 premiers caractères sont utilisés.

Exemples:

La ligne

v1 Entrez l'âge du patient ###

dans un fichier .QES attribuera le nom V1 à cette variable si l'option "Nom des champs: premier mot" est cochée.

La ligne

Entrez l'âge du patient ###

Attribuera le nom 'Entrez' à cette variable. Dans ce cas, la sélection de l'option 'Nom des champs: automatique' s'avérerait plus judicieuse.

Si un nom de champ est utilisé plusieurs fois, l'occurrence suivante est agrémentée d'un nombre. Exemple:

Les lignes:

v1 Entrez l'âge du patient ###

v1 Taille du patient ###

créeront 2 variables dont les noms ne peuvent être identiques (V1). La seconde s'appellera donc V2, en dépit du fait que le premier mot de cette ligne soit 'V1'.

La casse des noms de variables dépend de l'option choisie dans l'option 'style des noms de champs' (menu fichier | options | créer le fichier de données).

Noms des champs: Automatique

Si l'option 'Nom de champs: automatique' est sélectionnée, EpiData créera les noms de variables en se basant sur le texte à l' gauche du champ de saisie. Ce nom aura un maximum de 8 caractères, commençant par une lettre (A-Z). Les symboles internationaux sont ignorés. Le nom est généré à partir du début de la question du champ concerné.

Les règles suivantes sont utilisées:

1.	Texte entouré d'accolades. Si la question est "{mon} premier {champ}", la variable portera le nom MONCHAMP.
2.	Mots communs ignorés (mots comme 'what', 'the'...) [Anglais seulement]
3.	Les champs non précédés d'une question prennent le nom du champ précédent agrémenté d'un numéro. Si le champ précédent s'appelle CHAMP, le champ suivant (s'il n'est pas précédé d'un texte de question) portera le nom CHAMP1. S'il n'y a pas de champ précédent, le nom par défaut utilisé sera FIELD1.
4.	Si le premier caractère est un chiffre, la lettre N est insérée avant (à la question '3 souris grises' correspondra le nom 'N3SOURIS').
5.	La casse (min / MAJ) des noms de variables est gérée par l'option 'Style des noms de champs', (fichier / options / créer le fichier de données / style des noms de champs).

Exemples:

Question	Nom de variable	Règle
Quelle est votre {nation}alité	NATION	Règle 1
Vous {aime}z la {c}rème {g}lacée	AIMECG	Règle 1
What is your name	ISYOURNA	Règle 2
3. question:	N3QUESTI	Règle 4

Indentation automatique

La fonction d'auto indentation peut s'avérer utile lors de l'utilisation de l'éditeur de texte d'EpiData. Si cette option est enclenchée, chaque nouvelle ligne commence avec le même retrait à gauche que la ligne précédente. Cela s'avère particulièrement pratique lors de la création de fichiers de contrôle

Aligner les champs

La fonction d'alignement des champs de saisie peut être utilisée lors de la rédaction d'un fichier QES. Placer le curseur sur une ligne contenant un champ, et sélectionner l'option Editer | Aligner les champs.

Le résultat obtenu est dépendant du paramétrage de la fonction d'attribution de nom de variables (voir Options | Créer fichier de données). Si "Nom de champ: premier mot" est sélectionnée, les lignes suivantes

```
v1 A small text #####
v2 Other text <A > v3 ###.#
v3 Text ###
```

seront modifiées de la façon suivante

```
v1 A small text #####
v2 Other text <A > v3 ###.#
v3 Text ###
```

si le curseur était place sur la ligne de V1 avant que la fonction d'alignement des champs soit appelée.

Si l'option sélectionnée est "Noms de champs: automatique", le résultat sera:

```
v1 A small text #####
v2 Other text <A > v3 ###.#
v3 Text ###
```

Cette fonction peut être utilisée pour obtenir des questionnaires plus clairs et lisibles.

Créer le fichier de données

Le fichier de données est créé en :

- ❖ choisissant " nouveau fichier de données" dans le menu données dans l'écran principal, ou en
- ❖ cliquant le 2nd bouton sur la barre d'organisation du travail (2- Créer fichier de données), ou en
- ❖ choisissant Fichier de données | créer fichier de données

Il n'est pas nécessaire d'ouvrir un fichier QES dans l'éditeur pour créer un fichier de données. Si aucun fichier QES n'est ouvert dans l'éditeur, une boîte de choix de fichier sera affichée, permettant de choisir le fichier .QES à partir duquel le fichier de données sera créé.

Note : le système d'appellation des noms de variables dépend de la sélection en cours dans les options de Fichier de données. Voir noms de champ

Le fichier de données portera par défaut le même nom que le fichier QES, mais avec l'extension REC. Bien que n'étant pas une obligation, il s'avère fréquemment pratique d'affecter le même nom aux fichiers de questionnaires et aux fichiers de données.

Après l'attribution du nom du fichier de données, une courte description facultative (maximum 50 caractères) peut être entrée. C'est l'étiquette du fichier de données. Elle sera utilisée dans les fonctions de documentation du fichier , et elle est sauvegardée en tant qu'élément du fichier de données lors de l'exportation vers STATA

Avant qu'un fichier de données soit créé il peut être visionné préalablement si le fichier QES est ouvert dans l'éditeur en choisissant la prévisualisation du fichier de données à partir du menu de Fichier de données ou en appuyant sur Ctrl-T.

Notez qu'un fichier de données existant sera supprimé et les données seront perdues si un nouveau fichier de données est créé avec le même nom. Pour modifier un fichier de données, par exemple pour ajouter un champ ou pour changer le type d'un champ, employer l'option Outils | Modifier .REC avec .QES modifié

Réviser un fichier de données

Les fichiers de données peuvent être mis à jour sans perte de données. Les données déjà saisies seront copiées au nouveau format pour des champs conservant le même nom. Il est possible d'ajouter des champs, changer la définition des champs ou supprimer des champs. Employez la fonction Outils | Modifier .REC avec .QES modifié (tous les fichiers doivent avoir été préalablement fermés.)

Procédure à suivre pour réviser un fichier de données:

- ❖ Ouvrez le fichier QES dont dépend le fichier de données à modifier. Si ce fichier n'existe pas / plus, il est possible de le (re)créer à partir de l'option Outils | Convertir .REC en .QES.
- ❖ Sauvegardez le fichier QES modifié, et fermez le.
- ❖ Lancez Outils | Modifier .REC avec .QES modifié.
- ❖ Sélectionnez le fichier .QES modifié et le fichier de données que vous souhaitez modifier.

ATTENTION: Si vous supprimez des champs ou si vous en modifiez le nom (en modifiant le texte à la gauche de la zone de saisie), vous perdrez des données! Il est essentiel de vérifier immédiatement la structure du nouveau fichier de données obtenu. En cas d'erreur, le fichier de données original peut être restauré. Il est sauvegardé sous le nom *nomfichier old.rec*, situé dans le même répertoire de données que le fichier transformé.

Les noms de champs sous EpiData peuvent être générés de 2 façons différentes. Voir [Noms de champs](#) La méthode utilisée est choisie dans Fichier | Options. Le changement de système utilisé en cours de saisie peut entraîner la perte de données.

NOTE: Epi Info 6 crée des noms de variables d'une longueur maximale de 10 caractères. EpiData les crée avec une longueur maximale de 8 caractères. Il se peut que des problèmes se posent à l'utilisation de l'option 'Réviser la structure du fichier de données' d'EpiData. Il est alors préférable d'utiliser Epi Info.

La fonction Modifier .REC avec .QES modifié vérifie quel système d'appellation a été utilisé au moment de la création du fichier. Si ce système est différent de celui en cours, un avertissement suggérant d'utiliser le système original est affiché.

Le type de champ du fichier de données original peut être modifié en donnant au champ un nouveau type dans le fichier modifié.

Tous les champs peuvent être modifiés en alphanumériques ou alphanumériques majuscules.

Les champs numériques peuvent être changés en champs numériques comportant au moins autant de décimales. Il n'est pas recommandé de changer un champ décimal avec 2 décimales en entier; cela résulterait en une perte de données.

Type de champ original	Peut être change en:
Entier	Décimal, texte, texte majuscule
Décimal	Décimal comportant au moins le même nombre de décimales, texte, texte majuscule
IDNUM	Entier, decimal, texte, texte majuscule
Text	Texte majuscule
Upper-case text	Texte
Date (dd/mm/yyyy)	Date du jour (dd/mm/yyyy) , texte, texte majuscule
Date (mm/dd/yyyy)	Date du jour (mm/dd/yyyy) , texte, texte majuscule
Date du jour (dd/mm/yyyy)	Date (dd/mm/yyyy) , texte, texte majuscule
Date du jour (mm/mm/yyyy)	Date (mm/mm/yyyy) , texte, texte majuscule
Soundex	Texte, texte majuscule

Contrôle de saisie

Voici la manière la plus simple d'utiliser EpiData

- 1) Créer un fichier QES- pour décrire l'aspect du questionnaire,
- 2) Créer un fichier de données à partir du fichier QES, ce fichier aura une extension .REC
- 3) et saisir les données dans le fichier de données.

Ceci fonctionnera très bien.

Toutefois, au lieu de vérifier les données après les avoir toutes saisies, il est plus efficace de vérifier la validité des données en même temps qu'on les saisit. C'est ce que réalise le fichier de contrôle de saisie.

Un fichier de contrôle de saisie est un fichier qui décrit la manière de vérifier la validité des données d'une, plusieurs ou toutes les variables d'un questionnaire. Il peut aussi comporter des commandes de contrôle automatique de la manière de saisir les données, par exemple provoquer un saut d'une variable vers une autre en fonction des données saisies.

Il est nécessaire que le fichier de contrôle de saisie ait le même nom que celui des données (mais avec l'extension .CHK au lieu de .REC)

Voici quelques exemples d'opérations qui peuvent se dérouler pendant la saisie, si on prend soins de les programmer dans un fichier de contrôle de saisie :

- ❖ Limiter la saisie des nombres à un intervalle donné ou à un certain nombre de valeurs prédéfinies .
- ❖ rendre obligatoire la saisie d'une variable .
- ❖ copier le contenu d'une variable de l'enregistrement précédent dans le nouvel enregistrement à saisir .
- ❖ créer des sauts vers d'autres variables en fonction du contenu des données saisies.
- ❖ Calculez les valeurs de certaines variables en fonction de la valeur d'autres variables.
- ❖ Calculs complexes et opérations conditionnelles (opérations IF THEN) .
- ❖ afficher des messages d'aide à la personne qui saisit.

Une fois qu'on a créé un fichier de données le fichier de contrôle de saisie peut être créé de deux manières.

1) En utilisant l'option "Ajouter/Créer un contrôle de saisie" du menu CONTRÔLE de l'écran principal ou en cliquant sur le troisième bouton de la barre des tâches. Cette méthode peut être utilisée pour écrire des contrôles pour les variables. Cependant les blocs de commande qui concernent l'ensemble du fichier (par exemple les commandes Before File, Labelblock, etc.) ne peuvent être écrits qu'avec l'éditeur, décrit plus loin.

2) En tapant directement dans l'éditeur toutes les commandes de contrôle. N'oubliez pas de sauvegarder le fichier de contrôle de saisie avec le même nom que celui des données mais avec une extension .CHK au lieu de .REC.

Si un fichier de contrôle de saisie existe lorsqu'on choisit un fichier dans "Saisie de données", les commandes de contrôle sont chargées automatiquement .

Les commandes de base peuvent être programmées aisément dans l'option Ajoute/Réviser de "Contrôle de saisie". Elles comprennent le contrôle de l'étendue, la liste des valeurs autorisées, la saisie obligatoire ("Must Enter"), les sauts conditionnels entre variables, la répétition des valeurs de certaines variables de l'enregistrement précédent et l'utilisation des étiquettes de valeur.

Si vous vous contentez de ces commandes seulement passez à [Ajoute/Réviser des Contrôles de saisie.](#)

Si vous souhaitez utiliser d'autres commandes (BEFORE ENTRY, AFTER ENTRY, HIDE, UNHIDE, GOTO, HELP, IF-THEN-ENDIF) lisez structure des fichiers de contrôle de saisie et passez ensuite à exemple de fichier de contrôle de saisie.

Pour le détail des commandes de contrôle reportez vous à Liste des commande de contrôle de saisie.

Ajoute/Réviser des Contrôles de saisie.

Cette fonction ajoute ou modifie les contrôles - ou règles de validation - d'un fichier de données existant.

Lorsqu'on choisit un fichier de données, un masque d'écran est construit et une fenêtre des fonctions de contrôle s'affiche.

La touche F5 fait passer alternativement du masque d'écran à la fenêtre des fonctions de contrôle.

Raccourcis : Si le masque d'écran est actif, la combinaison de touches CTRL+flèche droite rend la fenêtre des fonctions de contrôle active. Si la fenêtre des fonctions de contrôle est active la combinaison de touches CTRL+flèche gauche fait basculer sur le masque d'écran.

Pour choisir une variable à laquelle ajouter des règles de validité, soit :

- la sélectionner dans le masque d'écran (par un clic de souris ou TAB et une série d'ENTRÉE pour atteindre la variable).
- utiliser la liste déroulante des noms de variables qui se trouve en haut de la fenêtre des fonctions de contrôle
- utiliser les raccourcis, combinaison de touches CTRL+ soit flèche vers le haut soit flèche vers le bas quand la fenêtre des fonctions de contrôle est active

Voir également la liste des touches de raccourci

La liste déroulante des noms de variables montre les noms des variables du fichier de données dans l'ordre d'affichage du masque d'écran.

Si la fenêtre des fonctions de contrôle est active, utilisez les flèches, la touche TAB ou ENTREE pour atteindre l'une des commandes suivantes,

Range/legal : étendue ou valeurs autorisées

Jumps Sauts

Must Enter Saisie obligatoire

RepeatValue Labels Répéter les données

Raccourcis : lorsque le masque d'écran est actif, les combinaisons de touches suivantes feront passer à

CTRL+ L pour changer l'étendue de la variable et/ou ses valeurs autorisées

CTRL+ J pour changer des sauts

CTRL+E pour activer ou annuler " saisie obligatoire " du champ courant

CTRL+R pour activer ou annuler " Répéter les données "

CTRL+A pour changer les étiquettes valeur du champ courant

Veuillez noter l'utilisation spéciale de CTRL+C, de CTRL+V et de CTRL+X quand le masque d'écran est actif.

Voir Copier les Contrôles

Enregistrer (ALT+S)

Cliquer sur le bouton Sauvegarder pour enregistrer tous les contrôles sans quitter la fonction Ajoute/Réviser des Contrôles de saisie

Éditez (Alt+D)

Cliquer sur Editer pour éditer tous les contrôles de la variable, voir éditer tous les contrôles de la variable

Sortir (ALT+X)

Cliquer sur Sortir pour quitter la fonction Ajoute/Révisé des Contrôles de saisie. Si vous avez fait des modifications il vous sera demandé si vous voulez les enregistrer.

Assurez vous d'appuyer sur ENTRÉE ou d'utiliser les flèches pour enregistrer les modifications.

Range / Legal : étendue, valeurs autorisées.

Si le curseur est sur une variable du masque d'écran, CTRL+L fera sauter le curseur à la ligne de définition de Range/Legal, (étendue, valeurs autorisées).

Une étendue est définie en tapant la valeur minimale et la valeur maximale, séparées par un signe moins (-). 2-5 signifie que seulement les nombres 2,3,4 ou 5 peuvent être saisis pour la variable courante. Si on veut seulement une valeur maximum, alors employer -INF (moins l'infini) comme valeur minimum. . Si on veut seulement une valeur minimum, alors employer +INF (plus l'infini) comme valeur maximum . Taper -INF-5 autorise la saisie de tous les nombres inférieurs ou égaux à 5. Taper 0-inf autorise la saisie de tous les nombres positifs.

Les valeurs autorisées sont définies en tapant toutes les valeurs acceptées séparées par un espace ou une virgule. Taper 4,6,8,10 signifie que seulement les nombres 4,6,8 or 10 peuvent être saisis pour cette variable.

Si on définit à la fois des valeurs autorisées et une étendue, il faut taper l'étendue avant les valeurs autorisées. Taper 2-6, 8 définit 2,3,4,5,6,8 comme valeurs autorisées.. Ecrire 8, 2-6 aura comme conséquence une erreur.

Si vous voulez employer la virgule au lieu du point comme séparateur décimal, entourez la définition de doubles apostrophes.

Jumps : Sauts

Les sauts indiquent sur quelle variable se rendra le curseur en fonction des données saisies dans la variable courante. Par exemple: si la variable courante est sexe (1=masculin, 2=féminin), les sauts peuvent définir que la valeur 1 fait passer le curseur à la variable V23 et que la valeur 2 fait passer le curseur à une autre variable, par exemple V40.

Raccourcis : La combinaison CTRL+J fait passer de la variable du masque d'écran à la ligne de définition des sauts.

Les sauts sont écrits en indiquant la valeur, puis le signe >(supérieur à) puis le nom de la variable à laquelle se rendre. 1>v23, 2>v40 définit que pour la valeur 1, la saisie continue à la variable V23 et que pour la valeur 2 la saisie continue à la variable V40.

Si des espaces ou des virgules sont employés dans une définition, entourez la définition entière par des doubles apostrophes, par exemple. " 2.5>v30 ", " 3,5>v35 ".

Au lieu d'indiquer un nom de variable, deux cibles spéciales peuvent être employées: END et WRITE. END signifie "sauter au dernier champ du masque de saisie", WRITE signifie "écrire l'enregistrement sur le disque après la saisie des données dans la variable courante ". Exemple ; " 1>v30", " 2>end", " 3>write " donne les règles suivantes: Si 1 est saisi alors la saisie continue à la variable appelée V30. Si 2 est saisi alors la saisie continue à la dernière variable du masque de saisie. Si 3 est saisi alors l'enregistrement en cours est écrit sur le disque.

Une commande générale de saut peut être écrite avec " AUTOJUMP V30 ". Ceci signifie que le curseur se rendra à la variable appelée V30 indépendamment des données saisies dans la variable contenant cette commande de saut. Si AUTOJUMP est employée ce doit être la seule commande de sauts pour cette variable.

Raccourcis : Quand la valeur pour un saut et " > " ont été écrits, cliquer avec la souris sur la variable à laquelle le saut devrait aller. Le nom de la variable cliquée sera inséré dans la ligne de commande après " > ". Utiliser la même méthode, pointer-cliquer, pour " autojump " (n'oubliez pas de taper un espace avant d'essayer de cliquer sur la variable).

MUST ENTER : Saisie obligatoire

Cette règle définit que des données doivent être saisies dans la variable courante.

La combinaison CTRL-E quand le curseur est sur une variable du masque d'écran (pendant Ajoute/Révisé des Contrôles de saisie) fait basculer la commande MUST ENTER de Yes à No.

REPEAT : Répéter la valeur de l'enregistrement précédent

Si cette règle est choisie, les données saisies dans l'enregistrement précédent seront répétées dans le suivant. Des données répétées peuvent être changées durant la saisie, mais cette fonction peut éviter beaucoup de frappe.

CTRL-R quand le curseur est sur une variable du masque d'écran fait basculer la commande REPEAT de Yes à No.

VALUE LABELS : Étiquettes de valeur

Les étiquettes de valeur sont un ensemble de valeurs combinées avec un texte qui explique leur signification.

Par exemple:

Une variable est créée pour décrire l'information sur le sexe du sujet. On peut décider qu'une valeur 1 signifie que le sujet est masculin et qu'une valeur 2 signifie féminin. Si une étiquette de valeur est définie une "table de correspondance" peut être montrée pendant la saisie des données si l'utilisateur appuie sur F9 (ou + du pavé numérique). L'étiquette de valeur dans cet exemple serait:

```
1 masculin
2 féminin
```

Pour définir une nouvelle étiquette

Cliquez le bouton à droite du cadre de la liste déroulante des étiquettes de valeur identifiée par " + ". ceci ouvre un petit éditeur la fenêtre avec ce texte:

```
LABEL [labelname]
END
```

Le texte " [labelname] " est en surbrillance et sera remplacé par le nom de l'étiquette tapé. Le nom dans l'exemple ci dessus pourrait être " sexe ".

Appuyez sur la touche ENTRÉE pour aller à la ligne suivante. Écrivez le nombre 1 suivi au moins de l'un espace et du texte de la valeur, ici " masculin ". Appuyez sur la touche ENTRÉE pour déplacer le curseur à la ligne et écrire la valeur et le texte pour la valeur suivante.

Le texte à entrer pour l'exemple ci-dessus est:

```
LABEL Sexe
```

```

1 Masculin
2 Féminin
END

```

Les espaces avant les valeurs 1 et 2 sont facultatifs, mais il facilitent la lecture.

Cliquez " accepter et fermer " ou appuyez sur ALT+A pour fermer la fenêtre d'édition. Le nom de la nouvelle étiquette s'affichera maintenant dans la liste déroulante d'étiquettes de valeur.

Éditer une étiquette existante

Assurez-vous que le nom de l'étiquette à éditer est affiché dans la liste déroulante d'étiquettes de valeur. Cliquez le bouton + . La fenêtre d'édition est maintenant affichée avec toutes les étiquettes définies. Éditez les étiquettes et cliquez " accepter et fermer " pour sortir ou appuyez sur ESC (ou cliquez Annuler) pour abandonner les modifications.

Assigner une étiquette existante à une variable

Cliquez sur la flèche vers le bas dans la liste déroulante des étiquettes de valeur et choisissez l'étiquette appropriée. Plusieurs champs peuvent partager la même étiquette de valeur, qui de cette façon doit être définie une fois seulement.

Effacer l'étiquette de valeur pour la variable.

Cliquez sur la flèche vers le bas dans la liste déroulante des étiquettes de valeur et choisissez " aucun ".

Employer les étiquettes prédéfinies

A l'installation d'EpiData une bibliothèque d'étiquettes de valeur a été enregistrée dans le répertoire du programme EpiData. La bibliothèque a pour nom de fichier EPIDATA.LBL.

La bibliothèque est destinée à devenir une aide quand les mêmes étiquettes de valeur sont employées souvent dans différents projets.

Quand la flèche vers le bas de la liste déroulante des étiquettes de valeur est cliquée, tous les noms des ensembles d'étiquette de valeur contenus dans le fichier de la bibliothèque sont montrés en police normale. Les noms "en gras" désignent des ensembles d'étiquettes de valeur qui font partie du fichier de contrôle de saisie en cours d'édition.

Si des étiquettes de valeur du fichier bibliothèque sont éditées, l'ensemble des étiquettes de valeur révisées est sauvegardé seulement dans le fichier de contrôle de saisie en cours d'édition. Les étiquettes de valeur du fichier bibliothèque ne peuvent pas être éditées dans EpiData, mais seulement en employant un éditeur.

Si un ensemble d'étiquettes de valeur du fichier de contrôle de saisie courant a le même nom qu'un ensemble d'étiquettes de valeur du fichier bibliothèque, alors les étiquettes du fichier courant sont utilisées, pas celles du fichier de bibliothèque.

Que se passe-t-il en réalité?

Quand un ensemble d'étiquettes de valeur est assigné à une variable, la définition d'étiquette est placée dans le LABELBLOCK (bloc des étiquettes) du fichier contrôle (.CHK) et la commande COMMENT LEGAL USE [<Nom d'étiquette>] est placée dans le bloc de la variable. Voir [la liste des commandes de contrôle](#) pour des détails sur ces commandes.

Éditez tous les contrôles pour la variable courante

La fenêtre Ajouter/réviser des contrôles de saisie possède un bouton " Editer ". En appuyant sur ce bouton (ou ALT+D), on ouvre une fenêtre d'édition dans laquelle on peut modifier directement le bloc de contrôle de la variable courante. Le fonctionnement est semblable à celui de l'éditeur qui permet de modifier l'ensemble du fichier de contrôle voir [structure du fichier de contrôle](#).

Si la variable courante n'a encore aucune commande de contrôle quand on clique le bouton d'édition, l'éditeur affiche seulement le nom de la variable (ce qui désigne le début du bloc de contrôle de la variable) et le mot END (qui désigne la fin du bloc de contrôle de la variable).

Si la variable courante a déjà une commande de contrôle elle apparaîtra dans la fenêtre de l'éditeur, et pourra être modifiée et de nouvelles commandes ajoutées.

Pressez ESC ou cliquez " annuler " pour abandonner les modifications.

Cliquez " accepter et fermer " ou pressez ALT+A pour enregistrer les changements.

Veuillez vous référer à [la structure des fichiers de contrôle](#) à la [liste des commandes de contrôle](#) si vous voulez écrire directement dans les commandes de contrôle d'une variable.

Notez que les blocs de contrôle autres que ceux des variables (par exemple BEFORE FILE....) ne peuvent pas être écrits en utilisant " Ajouter/réviser des contrôles de saisie ", mais seulement en ouvrant le fichier de contrôle entier avec un éditeur de texte.

Erreurs dans les commandes de contrôle

Quand " accepter et fermer " est choisi, les commandes tapées sont vérifiées. Si aucune erreur n'est trouvée, la fenêtre d'édition se ferme. Si des erreurs sont trouvées, la fenêtre d'édition est coupée en deux. Le haut de la fenêtre montre les commandes de contrôle et le bas les erreurs trouvées et les numéros de ligne où se trouvent les erreurs.

Double-cliquer sur la ligne montrant l'erreur pour faire sauter le curseur à la ligne correspondante du fichier de contrôle.

Cliquer " accepter et fermer " une fois que les erreurs ont été corrigées.

Noter que les expressions logiques et les calculs ne sont pas vérifiés, dans cette version d'EpiData, quand on clique " accepter et fermer ".

Duplication des commandes de contrôle

Les commandes de contrôles d'une variable peuvent facilement être copiées ou coupées vers un autre variable:

Cliquez sur une variable pour aller sur son bloc de contrôle. Pressez CTRL-C pour copier toutes les commandes de contrôle de la variable ou pressez CTRL-X pour les couper. Choisissez une autre variable et pressez CTRL-V pour insérer tous les contrôles dans la nouvelle variable.

La fonction de copier/coller/insérer copie les commandes étendue/valeurs autorisées, sauts, saisie obligatoire, répétition et étiquettes de valeur, mais pas les expressions étendues écrites avec la fenêtre " Editer tous les contrôles de la variable courante ".

Effacer tous les contrôles

Cette option efface tous les contrôles définis pour le fichier de données sélectionné. Les contrôles effacés ne peuvent pas être récupérés.

Structure du fichier de contrôle (.CHK)

Toutes les commandes dans un fichier CHK font partie d'un bloc de commandes. Cette version d'EpiData gère 2 types de blocs: les blocs de libellé (Labelblocks) et les blocs de champ (bloc de commandess).

Le bloc de libellé est décrit dans [la liste des commandes de contrôle](#)

Tout commande concernant un champ spécifique doit être placée dans un bloc de commandes. Un bloc de commandes commence par le nom du champ et se termine par la commande END (la casse utilisée pour rédiger les commandes est ignoré: "end" est interprété de la même manière que " END").

Quelques commandes sont elles-mêmes des blocs (par exemple LEGAL..END, JUMPS..END) tandis que d'autres commandes utilisent seulement une ligne (par exemple RANGE, GOTO). Tous les blocs sont terminés avec la commande END.

Exemple:

```
VAR1
  RANGE 1 5
  MUSTENTER
  JUMPS
    1 VAR4
    2 VAR5
    3 VAR10
  END
  BEFORE ENTRY
    VAR1 = F1 + 100
  END
  AFTER ENTRY
    IF (VAR1=2) AND (F2=1) THEN
      HELP " VAR1 ne peut pas avoir la 2 F2 a la valeur 1. Veuillez vérifier votre
entrée "
      GOTO VAR1
    ENDIF
  END
END
```

La première ligne marque le commencement d'un bloc de champ (c.-à-d. un ensemble des commandes) pour le champ appelé VAR1. La dernière ligne a la commande END qui marque la fin du bloc. Les lignes 2 et 3 contiennent des lignes de commande. La ligne 2 indique que seulement des nombres de 1 à 5 peuvent être entrés dans ce champ. La ligne 3 indique que des données doivent obligatoirement être saisies dans ce champ – Un message d'erreur sera affiché si le champ est laissé blanc.

La ligne 4 marque le commencement d'une commande de saut (JUMPS). Les commandes de saut sont des blocs en eux-mêmes et doivent être terminées avec un END (ligne 8). Dans cet exemple l'entrée de 1 fera sauter le curseur au champ appelé VAR4. L'entrée de 2 fera sauter le curseur au champ appelé VAR10.

Dans cet exemple les lignes dans le bloc de commandes sont indentées, ainsi que les lignes dans le bloc JUMPS. Cette indentation n'est pas nécessaire mais il facilite à l'utilisateur la lecture du fichier CHK, en rendant plus aisé le repérage des différents blocs.

2 commandes de contrôle importantes sont utilisées dans cet exemple: BEFORE ENTRY..END et AFTER ENTRY..END. Toutes les commandes dans BEFORE ENTRY sont exécutées à l'entrée du curseur dans le champ (avant que l'utilisateur n'ait saisi), Cette commande peut donc être utilisée pour entrer une valeur par défaut, que l'utilisateur pourra éventuellement conserver ou modifier. Toutes les commandes dans AFTER

ENTRY..END sont exécutés quand l'utilisateur déplace le curseur à un autre champ (= quitte le champ après la saisie). Les commandes d'un bloc de commandes sans BEFORE ENTRY ou AFTER ENTRY sont considérées comme étant AFTER ENTRY.

Il n'est pas nécessaire d'inclure de bloc à chaque champ dans le fichier de données. L'éditeur d'EpiData a une fonction d'autoindentation (dans le menu d'édition) qui facilite l'indentation des lignes.

Continuez avec [exemple d'un fichier CHK](#) ou voyez [la liste des commandes de contrôle](#)

Exemple d'un fichier CHK

Les 3 fichiers d'exemple sont stockés dans le répertoire où EpiData a été installé: SAMPLE.qes, SAMPLE.rec et SAMPLE.chk, qui sont censés rassembler les données d'une étude de croissance d'une cohorte d'enfants.

Si les valeurs rencontrées peuvent parfois sembler illogiques, il est important de garder à l'esprit qu'il ne s'agit pas de 'vraies' données, mais que l'objectif est de bâtir un exemple démontrant l'utilisation de certaines commandes de contrôle.

Dans la fenêtre principale d'EpiData, choisissez Fichier | ouvrir et choisissez le fichier SAMPLE.qes et cliquez sur Ouvrir. Retournez dans Fichier |Ouvrir, modifiez le type de fichier en 'Fichier de contrôle EpiData (*.CHK)', et choisissez SAMPLE.chk et cliquez sur Ouvrir. Les 2 fichiers sont maintenant affichés.Pour réorganiser l'affichage, choisissez Fenêtre | réorganiser horizontalement , qui l'affichage simultané des 2 fichiers. Noter que la barre de statut affiche le numéro de ligne courante.

Il est conseillé d'employer une police à chasse fixe (par exemple courier new) dans l'éditeur quand vous travaillez avec des fichiers CHK, qui rend la lecture et l'impression plus aisées.

La première ligne dans le fichier de contrôle commence par un signe *. Toutes lignes commençant par le signe * est considérée comme un commentaire et est ignorée quand le fichier est interprété.

Le premier bloc dans le fichier de contrôle (commençant à la ligne 3 et finissant à la ligne 22) est un LABELBLOCK. Dans ce bloc deux des ensembles d'étiquettes de valeur sont définis, une s'appelle " nationality ", l'autres " years ". Des étiquettes sont employées donnant à l'utilisateur la signification des codes. Par exemple le numéro 3 sera employé pour indiquer un citoyen américain.

Les deux ensembles d'étiquettes sont eux-mêmes des blocs. L'étiquette " nationality " commence à la ligne 4 et finit à la ligne 14.

Pour un explication complète sur les étiquettes voir [comment legal](#)

Le premier bloc de commandes commence à la ligne 24 et se termine à la ligne 26. Il concerne le champ appelé le " V1 " et comme vous pouvez le constater dans le fichier QES, ce champ est utilisé pour entrer la taille de la personne. Une seule commande se trouve dans ce bloc de commandes , c'est RANGE 130 230. Cette commande limite les entrées légales dans le champ V1 aux nombres entre (incluant) 130 et 230. Ceci s'assure que vous – par une faute de frappe – n'entrez pas une valeur de 3 centimètres de taille.

Le bloc de commandes suivant(lignes 28-30) concerne le champ B1 où la nationalité de la personne est entrée. Comme vous pouvez voir dans le fichier QES, le champ est un champ de nombre entier sur 1 caractère. L'utilisateur est aidé dans la saisie à l'aide des étiquettes. La commande de la ligne 29: " comment legal use nationality fait utiliser à ce champ l'ensemble d'étiquettes appelé 'nationality' situé au début du fichier. Au moment da la saisie de ce champ, l'utilisateur peut appuyer sur F9, provoquant l'ouverture d'une liste de choix. La commande 'comment legal' peut être utilisée de 3 façons. Voir [comment legal](#)

Le bloc de commandes du champ D1 commence à la ligne 32. Il finit à la ligne 46. La donnée concernée est la date de naissance de la personne. Le bloc de commandes contient une commande AFTER ENTRY entre les lignes 33 et 45. Les commandes contenues dans AFTER ENTRY sont exécutées après que l'utilisateur ait entré de la date de naissance; les commandes dans ce bloc ont deux objectifs: 1) vérifier si la date de naissance semble cohérente et 2) calculer l'âge de la personne et pour placer cette valeur dans le champ D2.

Le contrôle de la date est fait avec deux IF nichés dans des blocs. Le premier IF (ligne 34) est:

```
IF (Year(d1) < 1900) OR (d1 > Today) THEN
```

ce qui signifie:

SI l'année de la date entrée dans le champ D1 est inférieure à 1900

OU si la date D1 est supérieure à la date d'aujourd'hui (c.-à-d. date future)

ALORS une conséquence est provoquée.

Si l'une ou l'autre de ces deux conditions est vraie, alors les commandes entre THEN (ligne 34) et ELSE (ligne 37) sont exécutées. Dans ce cas, une boîte de message apparaîtra à l'écran demandant à l'utilisateur de vérifier la date de naissance sur le support papier. Ensuite le curseur reviendra au champ D1 au lieu de continuer sur le prochain champ.

Si les deux conditions dans IF (ligne 34) sont fausses, les commandes entre ELSE (ligne 37) et ENDIF (ligne 44) sont exécutées. Si aucune commande ELSE n'est présente (comme dans cet exemple), la prochaine commande exécutée est celle suivant la commande ENDIF.

Dans ce fichier CHK les commandes dans le bloc ELSE..ENDIF (ligne 37-44) contiennent un nouveau bloc IF-THEN-END. Cela montre que des blocs d'IF-ENDIF peuvent être nichés. Soyez très attentif à assortir chaque IF de ELSE et ENDIF.. Dans cet exemple ENDIF le plus indenté (ligne 43) dépend du IF le plus indenté (ligne 38), alors que ENDIF le plus extérieur (ligne 44) appartient au IF le plus extérieur (ligne 34).

Le bloc IF-THEN-ELSE-END suivant commence à la ligne 38:

```
IF (ROUND(int((today-d1)/365.25)) < 15) THEN
```

ce qui signifie:

SI la date d'aujourd'hui moins la date écrite dans le champ D1 est inférieure à 15 ans

ALORS une conséquence est provoquée.

Si c'est vrai alors les lignes 39 et 40 sont exécutées informant l'utilisateur que la date entrée indique un âge inférieur à 15 ans, la personne, qui ne peut donc être un ascendant. Si la condition à la ligne 38 est fausse, la commande dans le bloc ELSE (ligne 42) est exécutée. Cette ligne calcule l'âge de la personne basée sur la date entrée dans le champ D1 et assigne le résultat au champ D2.

La ligne 43 termine le bloc IF-ENDIF les plus indenté

La ligne 44 termine le bloc IF-ENDIF les plus extérieur

La ligne 45 termine le bloc AFTER ENTRY

La ligne 46 termine le bloc de commandes du champ D1

Cet exemple montre comment plusieurs conditions peuvent être vérifiées avec différentes actions en conséquence. Si les blocs IF-ENDIF ne sont pas limités à 2, un grand nombre de blocs IF-ENDIF peut être nichés. Attention: un grand nombre de blocs nichés IF-ENDIF peuvent rendre la maintenance du fichier difficile. Utilisez l'indentation dans le fichier .CHK pour en faciliter la lecture.

Le bloc de commandes pour le champ D2 commence à la ligne 48. Une seule commande est entrée: NOENTER qui interdit à l'utilisateur de saisir des données dans ce champ.

Le champ V1A demande si la personne a des enfants. Si la réponse est non, alors les champs V1B à V2 sont non pertinents et le processus de saisie de données continue avec le champ V6. Cette action est obtenue dans le bloc de commandes pour le champ V1A (lignes 52-76). Les lignes 53-55 (JUMPS) provoquent le déplacement du pointeur dans le champ V6 si la lettre N (pour non) est entrée. La ligne 56 (MUSTENTER) provoque l'obligation de saisie dans ce champ - il ne peut pas être laissé vide.

Un bloc AFTER ENTRY .. END est de la ligne 57 à la ligne 75. Il contient un bloc IF-THEN-ELSE-ENTIF. IF " NOT V1A " (c.-à-d. si V1A est 0 ou vide), des lignes 59-68 ne sont pas exécutés. Ces lignes réinitialisent le contenu des champs V1B, V1C, V1D, V1E et V2 et cachent ces champs (en rendant la saisie impossible, car ces informations sont non pertinentes si la personne n'a aucun enfant).

Si V1A est oui, les commandes des lignes 70-73 sont exécutés. Les commandes de ces lignes montrent (UNHIDE) ces quatre champs, permettant à l'utilisateur de saisir des données.

Le bloc de commandes pour le champ V1C (taille du 1er enfant - voir les lignes 90-104 dans le fichier CHK) donne un exemple de la façon d'employer BEFORE ENTRY. Les commandes contenues dans ce bloc sont exécutées au moment où le pointeur arrive sur ce champ, avant que l'utilisateur ait entré une information. Dans ce cas la commande BEFORE ENTRY est utilisée comme moyen d'entrer une valeur par défaut pour le champ V1C. Cette valeur par défaut est la moitié de la taille de la personne si cette donnée est entrée; autrement la valeur par défaut est de 50 centimètres. Les commandes du bloc AFTER ENTRY dans le champ V1C provoquent un déplacement (GOTO) conditionnel. S'il y a seulement un enfant (c.-à-d. V1B=1) alors la moyenne de la taille des enfants est égale à la taille du premier enfant (V2=V1C) et le champ suivant est V6 (GOTO V6). La même technique est employée pour les champs V1D et V1E (excepté le fait que ces 2 champs n'ont pas de bloc BEFORE ENTRY).

La dernière chose montrée par ce fichier d'exemple est la simplification entraînée par l'utilisation des blocs de libellés (labelblocks). Les champs V12 et V13 nécessitent l'entrée d'un code désignant une fourchette d'années. Plutôt que de d'entrer 2 fois les mêmes étiquettes, une seule ligne dans chacun des 2 champs fait référence à une seule liste. Si ces libellés doivent être changés, ils ne devront l'être qu'à un endroit.

Regardez le reste du fichier et essayez de saisir des données dans le fichier de données pour voir comment le déroulement de la saisie s'effectue.

Une liste de référence de toutes les commandes de contrôle peut être trouvée dans [la liste des commandes de contrôle](#)

Liste des commandes de contrôle

* : Voir COMMENTAIRES

AFTER ENTRY

Définit un bloc de commandes qui sont exécutées **après** l'entrée de la donnée dans le champ de saisie ou après un saut vers un autre champ de saisie. AFTER ENTRY est une commande de bloc et doit donc se terminer par END.

Des commandes spécifiées dans un bloc de champ de saisie, même si elle ne suivent pas l'instruction AFTER ENTRY, seront néanmoins interprétées comme des commandes AFTER ENTRY.

Exemple:

```
AFTER ENTRY
  <commande>
  <commande>
  . . .
END
```

AFTER FILE

Définit un bloc de commandes qui sont exécutées **après** la fermeture du fichier de données. Voir aussi BEFORE FILE.

Exemple:

```
AFTER FILE
  HELP "N'oubliez pas de faire une copie de sauvegarde de votre fichier!"
  TYPE=WARNING
END
```

AFTER RECORD

Définit un bloc de commandes qui sont exécutées juste avant de quitter un enregistrement nouveau ou modifié.

On utilise AFTER RECORD pour contrôler si les données ont été correctement saisies. Si une commande GOTO est exécutée dans un bloc AFTER RECORD, on ne quitte pas l'enregistrement en cours.

L'exemple ci-dessous permet de contrôler la saisie d'un numéro d'identification (ID-number). On a demandé à l'utilisateur d'entrer le numéro d'identification dans le premier champ ID1, puis de le saisir à nouveau dans le dernier champ ID2. Si les deux nombres ne sont pas identiques, ou si l'un des deux n'a pas été saisi, un message d'alerte est affiché, le curseur se place sur le champ ID1 et l'enregistrement n'est pas sauvé et ni quitté.

Exemple:

```
AFTER RECORD
  IF (ID1<>ID2) THEN
    HELP "ID1=@ID1 et ID2=@ID2\n\nVérifiez vos données, SVP" TYPE=WARNING
    GOTO ID1
    EXIT
  ENDIF
  IF (ID1 = .) OR (ID2 = .) THEN
    HELP "Le numéro d'identification doit obligatoirement être saisi dans les deux
champs, ID1 et ID2" TYPE=ERROR
    IF ID1 = . THEN
      GOTO ID1
    ELSE
      GOTO ID2
    
```

```

    ENDIF
  ENDIF
END

```

AUTOJUMP

Saut inconditionnel vers un autre champ. Le saut s'effectue lorsque l'utilisateur quitte le champ actuel. Voir aussi JUMPS. A la place du nom du champ visé, on peut utiliser les termes END ou WRITE.

AUTOJUMP END amène le curseur sur le dernier champ de l'enregistrement. AUTOJUMP WRITE active le message "Sauvez l'enregistrement sur le disque?". En cas de réponse OUI, l'enregistrement suivant est chargé ou un nouvel enregistrement est ouvert.

Exemple:

```

AUTOJUMP [nom du champ visé par le saut]
AUTOJUMP END
AUTOJUMP WRITE

```

AUTOSAVE

Lorsqu'un enregistrement modifié est quitté, le message "Sauvez l'enregistrement sur le disque" est proposé à l'utilisateur, qui a donc l'option de ne pas sauver cette modification. Cette option peut être supprimée en ajoutant la commande AUTOSAVE au fichier de contrôle. Cette commande peut également être utilisée comme paramètre de programme

Soyez prudent dans l'utilisation de AUTOSAVE. Un enregistrement existant peut être écrasé par l'enregistrement modifié, sans aucun message d'alerte.

Exemple:

```

BEFORE FILE
  AUTOSAVE
END

```

BEFORE ENTRY

Définit un bloc de commandes qui sont exécutées, lorsque le curseur se positionne sur le champ de saisie, mais **avant** la saisie de la donnée. BEFORE ENTRY est une commande de bloc et doit donc se terminer par END.

Des commandes situées dans un bloc de champ de saisie, ne suivant ni l'instruction AFTER ENTRY, ni l'instruction BEFORE ENTRY seront interprétées comme des commandes AFTER ENTRY.

Exemple:

```

BEFORE ENTRY
  <commande>
  <commande>
  ...
END

```

BEFORE FILE

Définit un bloc de commandes qui sont exécutées, lorsqu'un fichier de données est ouvert en saisie, mais **avant** qu'aucune donnée ne soit saisie. Voir aussi AFTER FILE.

On utilise volontiers BEFORE FILE pour définir des variables temporaires à utiliser dans le fichier.

Exemple:

```
BEFORE FILE
  HELP "Bienvenue dans mon fichier"
  DEFINE varAge ###
  DEFINE varRefDate <dd/mm/yyyy>
END
```

BEFORE RECORD

Définit un bloc de commandes qui sont exécutées, lorsqu'un enregistrement est ouvert, mais **avant** qu'aucune donnée ne soit saisie. Voir aussi AFTER RECORD.

Exemple:

```
BEFORE RECORD
  varAge = 33
END
```

CLEAR

Efface le contenu d'un champ spécifié. Le champ devient vide. Si aucun nom de champ n'est indiqué après CLEAR, le champ sur lequel s'applique la commande est effacé.

Exemples:

```
CLEAR
CLEAR champ5
```

COMMENTS (*)

Les lignes de commentaires doivent débuter par le signe *. L'ensemble de la ligne précédée de ce caractère est ignorée lors de l'exécution du fichier de contrôle. Attention: notez que l'utilisation des fonctions de contrôle Add/Revise peut changer l'emplacement de la ligne de commentaire lors de la sauvegarde d'un fichier de contrôle révisé.

COMMENT LEGAL

Agit de la même manière que LEGAL en indiquant quelles valeurs sont permises dans le champ de saisie. Mais COMMENT LEGAL offre en outre en plus la possibilité de visualiser une liste des valeurs admises et leur signification (en pressant F9 ou la touche numérique + au moment de la saisie)

COMMENT LEGAL peut être employé selon 3 syntaxes : 1) comme commande de bloc, 2) comme référence à un bloc COMMENT LEGAL d'un autre champ de saisie et 3) comme référence à un groupe d'étiquettes définie dans un LABELBLOCK. Si COMMENT LEGAL est utilisé comme une référence, le terme USE doit être ajouté avant le nom du champ contenant le bloc COMMENT LEGAL ou le nom du groupe d'étiquettes définie dans le LABELBLOCK.

COMMENT LEGAL peut aussi être utilisé avec des structures IF-THEN, par exemple dans les codages hiérarchisés. Cela peut être utile, si le groupe d'étiquettes (COMMENT LEGAL) utilisé dans un champ dépend des valeurs saisies d'un autre champ. Voir exemple ci-dessous, ou voir les fichiers échantillon Hierartest.rec et Hierartest.chk.

Exemples:

```
COMMENT LEGAL
  1 Homme
  2 Femme
END
```

COMMENT LEGAL USE [nom du champ]

COMMENT LEGAL USE [nom de l'étiquette]

Exemples de COMMENT LEGAL avec des structures IF-THEN

```
V1                {Sélection d'un pays}
  COMMENT LEGAL
    1  USA
    2  CANADA
  END
END
```

```
V2                {Sélection d'un état}
  IF V1=1 THEN
    COMMENT LEGAL
      1  Alabama
      2  New York
      3  Nevada
      4  Oklahoma
      5  .....
    END
  ENDIF
  IF V1=2 THEN
    COMMENT LEGAL
      1  Nova Scotia
      2  Quebec
      3  ....
    END
  ENDIF
END
```

CONFIRM

Par défaut, lorsqu'un champ de saisie est rempli, le curseur se déplace automatiquement vers le champ suivant. Cette fonction peut être supprimée en utilisant la commande CONFIRM dans le fichier de contrôle. Si CONFIRM est employé, le curseur ne se déplacera au champ suivant que si la touche Entrée est tapée.

Voir aussi CONFIRMFIELD.

Exemple:

```
BEFORE FILE
  CONFIRM
END
```

CONFIRMFIELD

Même action que CONFIRM, mais alors que CONFIRM agit sur tous les champs du fichier, CONFIRMFIELD n'agit que sur un seul champ. Cette commande ne peut être placée que dans un bloc de champ de saisie.

Exemple:

```
V1
```

```
CONFIRMFIELD
MUSTENTER
END
```

DEFINE

Permet de définir de nouvelles variables temporaires. Ces variables peuvent être utilisées pour recevoir des valeurs de calculs transitoires et pour reporter ces valeurs d'un enregistrement à l'autre.

La commande DEFINE peut être accompagnée du terme optionnel CUMULATIVE. Les variables cumulatives ne sont pas remises à zéro lorsqu'un nouvel enregistrement est saisi.

Si CUMULATIVE n'est pas précisé, la variable est remise valeur manquante entre chaque enregistrement.

Les noms des variables définies par DEFINE ne peuvent excéder 16 caractères. Notez que les variables temporaires ne sont pas sauvegardées.

Exemples:

```
DEFINE VarTemp #####          (definit un entier de 4 chiffres nommé VarTemp)
DEFINE varNom <A              > CUMULATIVE
DEFINE tempDate <dd/mm/yyyy>
```

EXIT

Stoppe l'exécution d'un bloc de commandes et quitte le bloc (par exemple un bloc AFTER ENTRY). EXIT est volontiers utilisé pour éviter une longue succession de IF-THEN-ELSE.

Dans l'exemple ci-dessous, plusieurs contrôles sont effectués sur un champ de saisie de type date (<dd/mm/yyyy>). Après chaque contrôle, EXIT fait quitter le bloc AFTER ENTRY avant qu'aucune autre instruction ne soit exécutée.

Exemple:

```
d1
  AFTER ENTRY
    IF D1 = . THEN
      HELP "Une date doit être saisie" TYPE=ERROR
      GOTO D1
      EXIT
    ENDIF
    IF D1 > TODAY THEN
      HELP "Une date postérieure à aujourd'hui n'est pas acceptée" TYPE=WARNING
      GOTO D1
      EXIT
    ENDIF
    IF D1 < TODAY-365 THEN
      HELP "Le @D1 date de plus d'un an. Recommencez la saisie, SVP" TYPE=ERROR
      GOTO D1
      EXIT
    ENDIF
    * Les commandes suivantes ne seront exécutées que si les
    * contrôles précédents vérifient que la date est valide
    D2 = D1+14
    V4 = Year(D1)
  END
END
```

GOTO

Fait déplacer le curseur sur un champ de saisie spécifié. Si aucun nom de champ n'est spécifié après GOTO, le curseur saute dans le champ contenant la commande.

Exemples:

```
GOTO
GOTO champ10
GOTO WRITEREC          (Termine l'enregistrement et demande "Sauvegarde de
l'enregistrement?"
```

HELP

Affiche un message personnalisé à l'écran. L'utilisateur doit cliquer sur le bouton OK avant que la saisie de données puisse continuer. Quatre types différents de messages existent: Information, Avertissement, Confirmation et Erreur. Si aucun type n'est indiqué, la boîte d'information est employée. L'indication du type peut se faire en utilisant le mot entier (ex: WARNING) ou sa première lettre (W pour WARNING)

Insérer "`\n`" pour provoquer un saut de ligne.

La valeur courante d'un champ utilisant DEFINE peut être montrés dans le message d'aide en employant `@nom_champ`. Si un caractère @ doit être montré (par exemple en tant qu'élément d'une adresse de E-mail), utilisez deux @.

Exemples:

```
HELP "ceci est une information"
HELP "ceci est une boîte\n d'avertissement sur 2 lignes" TYPE=warning
HELP "ceci est également une boîte d'avertissement"TYPE=w
HELP "confirmez svp " TYPE=confirmation
HELP "vous avez fait une erreur!" TYPE=error
HELP "La valeur du champ V1 est @v1"
HELP "l'adresse de E-mail d'EpiData est Info@@EpiData.dk"
```

HIDE, UNHIDE

Cache ou montre un champ. Quand un champ est caché sa couleur est modifiée, et l'utilisateur ne peut pas saisir des données dans le champ. Si aucun nom de champ n'est indiqué après le mot HIDE ou UNHIDE alors c'est le champ contenant la commande qui est soit caché soit montré.

Exemples:

```
HIDE
CACHEZ champ5
UNHIDE
UNHIDE champ5
```

IF

Structure de la commande IF:

```
IF <condition> THEN
  < commande à exécuter si la condition est vraie>
ENDIF
```

ou

```
IF <condition> THEN
  < commande à exécuter si la condition est vraie>
```

```
ELSE
< commande à exécuter si la condition est fausse>
ENDIF
```

Un bloc IF doit être terminé par la commande ENDIF. Les commandes à exécuter peuvent être sur plus d'une, et elles peuvent contenir d'autres commances IF nichées.

La condition doit avoir un résultat booléen (c.-à-d. vrai ou faux). Une liste des opérateurs et des fonctions qui peuvent être utilisés dans l'expression de condition, voir [Operators and fonctions](#)

L'expression de condition peut être composée de plusieurs parties séparées près AND ou OR chaque partie doit être exprimée entre parenthèses (différence avec Epi Info). Exemple: IF Champ2 > 5 AND Champ3 < 10 THEN... n'est pas autorisé. Entrez IF (Champ2 > 5) AND (Champ3 < 10) THEN...à la place.

Les erreurs dans les conditions IF- sont ignorées lors de la saisie. La détection d'erreurs peut être réalisée en utilisant l'option *messages d'erreurs*. Voir le chapitre [Options](#)

Exemples:

```
IF champ1 >10 THEN
  GOTO champ10
ENDIF
```

```
IF (Cos(champ1)*Sin(champ1)<0.3) AND (champ2<>0) THEN
  IF champ2 < champ3 THEN
    HELP "Erreur!"
    GOTO
  ENDIF
ELSE
  champ4=Tan(champ1)
  GOTO Champ23
ENDIF
```

```
SI champ10 =. THEN
  champ11 =.
  champ12 = 0
  date1 = "12/03/2001"
ENDIF
```

JUMPS

Saut conditionnel sur d'autres champs. JUMPS est une commande de bloc et doit être terminée par END. La liste entre les mots JUMPS et END indique 1) une valeur possible du champ après que l'utilisateur ait saisi des données et 2) un nom du champ sur lequel 'sauter' si l'utilisateur a entré la valeur indiquée. Voir également AUTOJUMP.

A la place d'un nom de champ sur lequel 'sauter', les mots END ou WRITE peuvent être employés. END fait sauter le curseur sur le dernier champ de l'enregistrement, WRITE fait aller directement à la question "Enregistrer sur le dique?". Après avoir répondu Oui à la question, l'enregistrement suivant ou un nouvel enregistrement est chargé.

Exemple:

```
JUMPS
  1 V5
  2 V10
```

```

3 END
4 WRITE
END

```

KEY

Syntaxe: KEY {UNIQUE} {keynumero}

Cette commande crée un index pour le champ dans lequel il est utilisé. L'index est un autre fichier qui accélère la recherche des enregistrements contenant une valeur particulière. Si le mot UNIQUE apparaît après CLEF, un index spécial est créé pour les identifiants uniques. JONES ne peut pas apparaître deux fois, et le NOM ne serait donc pas un choix judicieux pour un champ KEY UNIQUE. De tels champs sont ceux qui contiennent les numéros d'identification uniques qui ne peuvent donc être assignés qu'à un enregistrement chacun. KEY (sans UNIQUE) peut être assigné à n'importe quel champ et permet à des valeurs d'exister dans plus d'un enregistrement, de sorte qu'il puisse être employé avec des champs comme le NOM, le COMTÉ, ou l'ÂGE

KEY a trois utilités:

1. La recherche d'enregistrements lors de la saisie est accélérée de façon très significative dans le cas de fichiers contenant plusieurs centaines d'enregistrements.
2. L'affichage des données triées en utilisation la fonction Liste des données.
3. L'impossibilité de créer des doublons, par exemple d'un numéro d'identification (seulement KEY UNIQUE).

Le numéro de KEY est un numéro facultatif qui identifie l'ordre des clefs dans le dossier d'index. Si aucun numéro de KEY n'est donné, l'ordre des clefs est régi par l'ordre des champs contenant une commande KEY.

Si un champ KEY contient plus de 30 caractères, les 30 premiers caractères seront employés dans le KEY

Un maximum de 10 champs KEY peut être défini.

Les fichiers Index sont automatiquement créés quand un fichier est ouvert en saisie. Si le nombre d'enregistrements ou le nombre de champs KEY du fichier de données ne correspondent pas à l'index, celui-ci est recréé. Il est possible de reconstruire l'index manuellement en employant la fonction 'reconstruire les index' du menu outils.

Exemples:

```

IDNUMBER
  1 KEY UNIQUE 1
END

```

```

NOM
KEY
END

```

LABEL

Voir Labelblock

LABELBLOCK

Un Labelblock contient les définitions des ensembles d'étiquettes de certains codes. Le bloc doit finir avec END de commande et ne peut pas faire partie d'un champ. Chaque ensemble d'étiquettes de valeur doit commencer par la commande LABEL suivie du nom de l'étiquette et il doit finir avec la commande END. Les ensembles d'étiquettes de valeur définis dans le labelblock peuvent être employés avec la commande COMMENT LEGAL USE [nom_label].

L'exemple suivant montre comment définir deux ensembles d'étiquettes de valeur. L'un s'appelle "ouinon", l'autre "sexe".

Exemple:

```
LABELBLOCK
  LABEL ouinon
    1 oui
    2 non
  END
  LABEL sexe
    1 masculin
    2 féminin
    9 inconnu
  END
END
```

LEGAL

Indique des valeurs légales pour un champ, c.-à-d. les valeurs dont l'entrée est permise dans ce champ. LEGAL est une commande de bloc et doit être terminée par END.

Au lieu de définir les mêmes legals pour plusieurs champs vous pouvez définir des legals dans un domaine et puis se rapporter à la cette définition dans d'autres domaines ci-dessous. Ceci est fait en utilisant le code de commande LEGAL USE

Xxxx L'exposition d'exemple comment définissez seulement les chiffres pairs plus petit que 10 en tant qu'entrées légales dans le domaine. La même légal-définition est appliquée au champ V2 en employant LEGAL USE

Exemple:

```
LEGAL
  2
  4
  6
  8
END

V2
LEGAL USE V1
END
```

LET

Assigne une valeur ou le résultat d'un calcul à un champ ou à une variable de DEFINE. Le mot LET est facultatif. Pour une liste d'opérateurs et de fonctions qui peuvent être utilisés dans les calculs et d'autres expressions, voir [opérateurs et fonctions](#)

Les exemples suivants seront interprétés de la même manière

Exemples:

```
LET field5 = (field2/field3)+INT(field4)
```

```
field5 = (field2/field3)+INT(field4)
```

Autres exemples:

```
LET date1 = " 14/09/2000 "      (assigne une date à un champ date)
LET v1 = .                      (assigne le code de valeur manquante à V1)
LET b1 = ((15/2)>4)             (affecte la valeur "Y" à un champ booléen)
LET b2 = " Y "                 (affecte la valeur "Y" à un champ booléen)
LET b3 = faux                  (affecte la valeur "N" à un champ booléen)
LET v3 = integer(copy(s2,1,2))  (extrait les deux premiers caractères d'une
                                chaîne, et affecte le résultat à un champ de
                                numérique)

LET Text1 = " Q"+String(Number) (affecte la valeur Q14 au champ
                                alphanumérique Text1 si Le nombre est égal à
                                14)
```

MUSTENTER

Saisie requise (le champ ne peut être laissé vide).

Exemple:

```
MUSTENTER
```

NOENTER

Saisie manuelle impossible.

Exemple:

```
NOENTER
```

RANGE

Indique une gamme de valeurs légales pour le champ. Les mots - INFINITY et +INFINITY peuvent être employés si la limite inférieure ou la limite supérieure de la gamme est non pertinente.

Exemples:

```
RANGE -5 5 (permet la saisie de nombres entre -5 et 5)
RANGE - INFINITY 99 (permet tous les nombres inférieurs à 100)
100 INFINITY (permet tous les nombres supérieurs ou égaux à 100)
RANGE 1/3/2000 31/3/2000 (n'importe quelle date en mars 2000)
```

REPEAT

Le champ prend comme valeur par défaut la valeur qui lui a été attribuée dans l'enregistrement précédent.

Exemple:

```
REPEAT
```

TYPE

TYPE est employé pour afficher un texte à l'écran à côté d'un champ lors de la saisie. Une couleur facultative peut être indiquée.

Syntaxe: TYPE " *texte* " [*couleur*].

Les couleurs valides sont: Aqua,Black,Blue,Dkgray, Fuchsia, Gray, Green, Lime, Ltgray, Maroon, Navy, Olive, Purple, Red, Silver, Teal, White, Yellow.

Si aucune couleur n'est indiquée, la couleur bleue par défaut est employée.

Les valeurs courantes des champs de DEFINE peuvent être affichées dans le message d'aide en employant @ *nomdechamps* Si un caractère @- fait partie du texte, utilisez alors 2 @ (par exemple en tant qu'élément d'une adresse de E-mail)

Exemple:

```
V1
  AFTER ENTRY
  IF V1<5 THEN
    TYPE " le nombre @v1 est plus petit que 5" RED
  ELSE
    TYPE " @v1 est plus grands que 4"
  END
  END
  END
```

TYPE COMMENT

Peut être employé dans les champs ayant un ensemble de valeurs prédéfinies. Quand une valeur est écrite dans le champ le curseur est déplacé sur le champ suivant, alors que le texte lié à la valeur est écrit à la droite du champ ou dans un champ texte indiqué. Cela permet à la personne qui fait la saisie de s'assurer que la bonne valeur a été entrée. Cette fonction peut être employée en remplacement de la commande Epi Info CODES/codefield.

Note: la commande TYPE COMMENT ne peut pas être placée dans un bloc BEFORE ENTRY / AFTER ENTRY.

Un code facultatif de couleur peut être ajouté pour employer une couleur différente du bleu par défaut. Les couleurs valides sont: Aqua, Black, Blue, Dkgray, Fuchsia, Grey, Green, Lime, Ltgray, Dark red, Navy, Olive, Purple, Red, Silver, Teal, White, Yellow.

Syntaxe:

```
TYPE LE COMMENT [ couleur ] ou
TYPE COMMENT nomdechamps
```

Exemple:

Un champ contient ces commandes dans le fichier CHK:

```
V1
COMMENT LEGAL
 1 chien
 2 Chat
 3 Lion
 4 Rat
END
TYPE COMMENT yellow
END
```

Si la valeur 2 est entrée pour ce champ le texte le " chat " sera écrit à la droite du champ quand le curseur sort du champ.

Si TYPE COMMENT S1 de était employé à la place le mot " chat " serait entré dans le champ S1, à condition que S1 soit un champ alphanumérique.

TYPE COMMENT est recommandé à l'utilisation du codage hiérarchique. (voir commentaire légal).

TYPE STATUSBAR

Dans les questionnaires contenant de nombreux champs, cette fonction peut s'avérer utile en affichant par exemple le numéro d'une personne, quelle que soit la partie du questionnaire affichée.

L'emploi de TYPE STATUSBAR n'est possible que pour 1 champ..

Un libellé explicatif facultatif peut être ajouté à la commande pour indiquer quel champ a sa valeur affichée dans la barre de statut.

Un code facultatif de couleur peut être ajouté pour employer une couleur différente du bleu par défaut. Les couleurs valides sont: Aqua, Black, Blue, Dkgray, Fuchsia, Grey, Green, Lime, Ltgray, Dark red, Navy, Olive, Purple, Red, Silver, Teal, White, Yellow.

Exemple:

```
IDCODE
TYPE STATUSBAR " IDCODE = "
END
```

```
IDCODE
TYPE STATUSBAR " " LIME
END
```

UNHIDE

Voir HIDE

Opérateurs et fonctions

Cette liste montre les opérateurs et les fonctions disponibles dans les expressions booléennes dans les commandes IF et les calculs et expressions dans les commandes .

Opérateurs (arithmétiques, logiques et apparentés)

Fonctions:

Les fonctions arithmétiques

Fonctions pour chaîne

Fonctions pour dates

Opérateurs

Opérateurs arithmétiques

Opérateur	Opération	Type d'opérande	Type de résultat
^	exposant	nombre entier	Virgule flottante
		Virgule flottante	virgule flottante
+	addition	nombre entier	nombre entier
		virgule flottante	virgule flottante
		Chaîne	chaîne
-	soustraction	nombre entier	nombre entier
		virgule flottante	virgule flottante
*	multiplication	nombre entier	nombre entier
		virgule flottante	virgule flottante
/	division	nombre entier	virgule flottante
		virgule flottante	virgule flottante
division	division de nombre entier	nombre entier	nombre entier
mod	reste	nombre entier	nombre entier

Opérateurs logiques

Opérateur	Opération	Type d'opérande	Type de résultat
NOT	négation	booléen	Booléen
AND	et logique	booléen	Booléen
OR	ou logique	booléen	Booléen
xor	xor logique	booléen	Booléen

Opérateurs apparentés

Opérateur	Opération	Type d'opérande	Type de résultat
=	égale	compatible	Booléen

< >	Différent de	compatible	Booléen
<	Inférieur à	compatible	Booléen
>	Supérieur à	compatible	Booléen
< =	inférieur ou égal à	compatible	Booléen
> =	supérieur ou égal à	compatible	Booléen

Les opérateurs apparentés peuvent également être employés dans des champs alphanumériques.

Fonctions arithmétiques

fonction Abs(X): Virgule flottante

La fonction d'ABS renvoie la valeur absolue de l'argument. X est une expression numérique ou virgule flottante. ABS(4)=4, ABS(-4)=4

fonction ArcTan(X: Virgule flottante): Virgule flottante

ArcTan calcule l'arc tangent du nombre donné. D'autres fonctions trigonométriques sont utilisables, SIN, COS, ArcTan dans les expressions suivantes

$$\text{Tan}(x) = \text{Sin}(x) / \text{Cos}(x)$$

$$\text{ArcSin}(x) = \text{ArcTan}(x/\text{sqrt}(1-\text{sqr}(x)))$$

$$\text{ArcCos}(x) = \text{ArcTan}(\text{sqrt}(1-\text{sqr}(x)) / x)$$

fonction Cos(X: Virgule flottante): Virgule flottante

La fonction de Cos renvoie le cosinus de l'angle X.

fonction Exp(X: Virgule flottante): Virgule flottante

Exp renvoie la valeur de e augmentée à la puissance X, où e est la base des logarithmes normaux.

fonction Frac(X: Virgule flottante): Virgule flottante

La fonction de Frac renvoie la partie partielle de l'argument X. X est une expression de type virgule flottante. Le résultat est la partie partielle de X : Frac(X) = X - Int(X).

fonction Int(X: Virgule flottante): Virgule flottante

X est une expression de type virgule flottante. Le résultat est la valeur entière de X; c'est-à-dire, X arrondi. Veuillez noter que le résultat est du type virgule flottante, même s'il contient seulement la valeur entière de X. Round(X) permet d'obtenir un résultat qui peut être affecté à un champ de nombre entier.

fonction Entier(x): Nombre entier

Convertis X à une valeur de nombre entier. Si FIELD1 égale ' sb410 ' puis Entier(pos(FIELD1,3,2)) égale le nombre entier 41.

fonction Ln(X: Virgule flottante): Virgule flottante

La fonction de Ln renvoie le logarithme normal (Ln(e) = 1) de l'expression X. de type virgule flottante

fonction pi: Virgule flottante

Employez pi dans les calculs mathématiques qui exigent pi, le rapport de la circonférence d'un cercle à son diamètre. Pi est arrondi en tant que 3,1415926535897932385.

fonction Round(X: Virgule flottante): Nombre entier

La fonction de rond arrondit une valeur de type virgule flottante à une valeur de nombre-type. X est une expression de type virgule flottante. Rond renvoie une valeur qui est la valeur de X a arrondie au nombre entier le

plus proche. Si X est exactement à mi-chemin entre deux nombres entiers, le résultat est le nombre avec la plus grande valeur absolue.

fonction Sin(X: Virgule flottante): Virgule flottante

La fonction Sin renvoie le sinus de l'argument. X est une expression de type virgule flottante. Sin renvoie le sinus de l'angle X en radians.

fonction Sqr(X: Virgule flottante): Virgule flottante

La fonction Sqr renvoie le carré de l'argument. X est une expression à virgule flottante. Le résultat, du même type que X, est le carré de X, ou X*X.

fonction Sqrt(X: Virgule flottante): Virgule flottante

X est une expression à virgule flottante. Le résultat est la racine carrée du X.

fonction String(x): Chaîne

Convertit X en chaîne. Si FIELD1 est un champ de type nombre entier de valeur 41, 'sb'+String(FIELD1) renvoie la valeur 'sb41'

fonction Power(Base, exposant: Virgule flottante): Virgule flottante

La fonction de puissance élève la base à n'importe quelle puissance. Pour les exposants partiels ou les exposants supérieurs à MaxInt, base doit être supérieur à 0.

fonction Trunc(X: Virgule flottante):Entier

La fonction Trunc tronque une valeur de type virgule flottante à une valeur de type entier. X est une expression de type virgule flottante. Trunc renvoie une valeur entière qui est la valeur de X arrondie vers zéro.

Fonctions chaînes

fonction Upper(S: chaîne): chaîne

La fonction Upper renvoie une chaîne contenant le même texte que S, tous les caractères de 7-bit ASCII entre 'a' et 'z' étant convertis en majuscule.

fonction Lower(S: chaîne): chaîne

La fonction Lower renvoie une chaîne contenant le même texte que S, tous les caractères de 7-bit ASCII entre 'a' et 'z' étant convertis en minuscule.

fonction Copy(S: chaîne; Index, Count: Nombre entier): chaîne

La fonction de copie renvoie une sous-chaîne d'une chaîne. S est une expression de type chaîne. Index et Count sont des expressions de type numérique. Copy renvoie une chaîne contenant des caractères de compte commençant à la position de l'index dans la chaîne S. Si l'Index est supérieur à la longueur de S, une chaîne vide est renvoyée. Si le compte indique plus de caractères que disponibles, seuls les caractères depuis l'index jusqu'à la fin de S sont retournés.

fonction Pos(Substr: chaîne; S: chaîne): Nombre entier

Pos recherche une sous-chaîne, Substr, dans une chaîne, S. Substr et S sont des variables de type chaîne-. Pos recherche Substr dans S et renvoie un nombre entier qui est l'index du premier caractère de Substr dans S. Pos n'est pas casse-dépendant. Si Substr n'est pas trouvé, Pos retourne 0.

fonction Length(S: chaîne): Nombre entier

La fonction Length renvoie le nombre de caractères réellement utilisés dans la chaîne S.

fonction String(x): Chaîne

Convertit X en chaîne. Si FIELD1 est un nombre entier de valeur 41, 'sb'+String(FIELD1) renvoie le valeur 'sb41'

Fonctions dates

EpiData gère des dates en tant que champs de type virgule flottante, décomptant les jours depuis le 31/12 1899. La gestion des dates comme des nombres autorise l'emploi de dates dans les calculs. Pour calculer le nombre de jours entre deux dates, soustrayez simplement une date de l'autre.

fonction Date(d:entier, m:entier, y:entier): Date

La date a 3 nombres comme paramètres: le jour, le mois et l'année et retourne la date composée de ces trois paramètres. Cette fonction renvoie le résultat dans un champ date ou un entier, selon le type de champ auquel le résultat est assigné.

fonction Today: Date

Renvoie la date d'aujourd'hui. Cette fonction renvoie le résultat dans un champ date ou un entier, selon le type de champ auquel le résultat est assigné.

fonction Year(D: date): Nombre entier

Year renvoie l'année (en 4 chiffres) de la date D.

fonction Month(D: date): Nombre entier

Month renvoie le mois (c.-à-d. un nombre de 1 à 12) de la date D.

fonction Day(D: date): Nombre entier

Day renvoie le jour (c.-à-d. un nombre entre 1 et 31) de la date D.

fonction WeekNum(D: date):Entier

WeekNum renvoie le numéro de semaine de la date indiquée. Exemple: WeekNum("22/02/2001")=8.

fonction DayOfWeek(D: date):Entier

DayOfWeek renvoie un nombre représentant le jour de la semaine de la date indiquée. Exemple: DayOfWeek("22/02/2001")=4 (un jeudi). Veuillez noter que Lundi = 1, Dimanche = 7.

Au sujet des dates

EpiData considère les dates comme des nombres, décomptant le nombre de jours depuis le 31 décembre 1899, qui est le jour 1. Ainsi, 15 octobre 2000 a le 'nombre-date' 36814.

L'avantage des 'nombre-dates' est de faciliter les calculs avec des dates, par exemple en permettant d'ajouter 7 jours à une date ou de compter le nombre de jours entre date spécifique et aujourd'hui.

Quand des dates (c.-à-d. des 'nombre-dates') sont assignées à un champ date (en employant la commande LET la commande dans un fichier de contrôle), un format de date approprié sera utilisé. Quand des dates sont assignées à un champ de type nombre entier, le 'nombre-date' est affiché.

Les constantes-date peuvent être définies de deux manières: en entrant " 14/09/2000 " ou Date(14,9,2000), ces deux expressions renverront la date du 14 septembre 2000. Si " dd/mm/yyyy " est employé alors notez que la

chaîne est interprétée comme date si elle a 10 caractères de longueur et si la chaîne est composée au format européen.

Exemples:

Imaginez un fichier de données avec deux champs: D1 est un champ de type < dd/mm/yyyy > et INT1 est un champ de nombre entier à 5 chiffres (#####).

LET INT1=d1	Si l'utilisateur a entré 15/10/2000 dans D1, LET INT1=d1 entrera le nombre-date 36814 dans le champ INT1
LET D1=int1	Si l'utilisateur a entré le nombre-date 36814 dans le champ INT1, LET D1=INT1 entrera 15/10/2000 dans le champ D1
LET D1=d1+7	Ajoute une semaine à la date entrée dans D1 et formate le résultat comme date (dd/mm/yyyy).
LET INT1=d1+7	Ajoute une semaine à la date entrée dans D1 et formate le résultat comme nombre_date (ex. 36814).
LET INT1=Today - Date(1,10,2000)	Calcule le nombre de jours entre le 1er oct. 2000 et la date d'aujourd'hui et assigne le résultat à INT1.
LET INT1=(round(int((today-d1)/365.25))	Calcule l'âge d'une personne dont la date de naissance a été entrée dans D1.Round est nécessaire pour convertir le résultat du calcul à partir d'un réel en nombre en entier.
LET D1=Date(1, INT1, 2000)	Assigne la date 1-avr-2000 à D1 si l'utilisateur a entré 4 dans INT1.
LET D1="01/04/2000"+5	Assigne la date 6-avr-2000 à D1.

Autres fonctions

fonction IsBlank(nomdechamp): Booléen

IsBlank renvoie vrai si le champ, dont le nom est donné en paramètre, est blanc, c.-à-d. aucune donnée n'est saisie. IsBlank retourne Faux si le champ contient des données.

Au lieu d'employer IsBlank, un point peut être employée pour représenter une valeur absente. Ces deux exemples donne le même résultat:

```
IF IsBlank(v1) THEN...
IF v1 =. THEN...
```

Saisir des données

Choisissez cette fonction et sélectionnez le fichier de données d'extension. REC dans lequel vous voulez ajouter de nouvelles données ou éditer des données existantes. Si des règles de validation (contrôles) sont trouvées dans un fichier d'extension .CHK portant le même nom que le fichier .REC, elles seront appliquées pendant la saisie.

Quand un enregistrement est complet une demande est faite à l'utilisateur pour l'enregistrer.

Pour fermer le fichier utilisez le menu Fichier ou fermez la fenêtre de saisie.

[Navigation entre les champs](#)

[Navigation entre les enregistrements](#)

[Trouver des enregistrements](#)

Navigation entre les champs

Passer au champ suivant en :

- appuyant sur la touche Entrée
- appuyant sur la touche Tab
- appuyant sur la touche flèche vers le bas
- cliquant sur le champ souhaité avec la souris

Quand un champ est rempli le passage au champ suivant est automatique.

Choisissez le champ précédent en :

- appuyant sur les touches Shift et TAB
- appuyant sur la touche flèche vers le haut
- cliquant sur le champ souhaité avec la souris

Choisissez le premier champ par Ctrl+Début

Choisissez le dernier champ par Ctrl+Fin

Navigation entre les enregistrements




Des boutons de navigation sont présents en bas de la fenêtre de saisie. Toutes les fonctions de ces boutons peuvent être retrouvées dans le menu 'Aller à'.



Le panneau de navigation ci-dessus montre que l'enregistrement courant est le numéro 2 sur un total de 2. DEL signifie que le l'enregistrement courant est marqué comme supprimé.

Les boutons sont :

- ▶ Aller au premier enregistrement
- ◀ Enregistrement précédent (ou touches : Ctrl+Flèche rayée haute)
- ▶ Enregistrement suivant (ou touches : Ctrl+Flèche rayée basse)

-  Aller au dernier enregistrement
-  Écrire le nouvel enregistrement sur disque (ou touches : Ctrl+N)
-  Marquer l'enregistrement comme supprimé ou enlever la marque de suppression (ou touches : Shift+Suppr)

Veillez noter que des enregistrements peuvent seulement être marqués comme supprimés. Ils ne peuvent pas être physiquement supprimés dans cette version d'EpiData. Cette fonction, cependant, sera ajoutée à la prochaine version d'EpiData.

Trouver des enregistrements

Si le numéro de l'enregistrement est connu, on peut l'atteindre par le choix 'Atteindre enregistrement' du menu Aller à ou par la combinaison de touches Ctrl+G

Sinon, utiliser le choix 'Trouver enregistrement' ou Ctrl+F. Une boîte de dialogue s'ouvre dans laquelle on peut choisir les valeurs à chercher et les champs concernés. Le champ courant est mis par défaut. Les champs qui ne sont pas accessibles en saisie, comme les champs automatiques IDNUM, peuvent ici être choisis pour la recherche.

Une recherche peut concerner jusqu'à trois champs différents. La valeur peut être recherchée comme contenu exact du champ, comme présente au début du champ, ou présente n'importe où dans le champ. Les autres options sont : prise en compte de la casse (majuscules / minuscules), correspondance sur des mots entiers uniquement ou des parties de mots, recherche ou pas dans les enregistrements supprimés.

Appuyez F3 ou utilisez le choix 'Trouver le suivant' du menu Aller à pour répéter la recherche avec les mêmes paramètres.

Une recherche peut être arrêtée par un appui sur la touche Echap cliquant sur Annuler dans la boîte montrant le progrès de la recherche en cours.

Veillez noter que la recherche est beaucoup plus rapide si les champs (tous) sur lesquels porte la recherche sont indexés. Un champ est indexé quand son contrôle comporte la commande KEY. Se référer à la description de cette commande pour plus de détails.

Appuyez le bouton 'Réinitialiser' dans la boîte de recherche pour effacer tous les paramètres et revenir à une recherche dans le seul champ courant.

Documenter

Cette fonction permet d'accéder aux informations d'un fichier de saisie et des champs qu'il contient.

Les informations concernant le fichier de saisie sont les suivantes :

- le nom du fichier
- la taille du fichier
- la date des dernières modifications
- le nombre de champs d'entrée de données
- le nombre d'enregistrements
- l'existence de contrôles de la saisie

Pour chaque champ d'entrée de données, les informations suivantes sont précisées :

- Le nom du champ
- L'intitulé du champ (ou label)
- Le type de format du champ
- La longueur du champ
- La liste des contrôles actifs sur le champ

Les informations apparaissent dans un éditeur de texte, et peuvent être sauvegardées ou imprimées.

Notes de saisie

Lors de la création d'un questionnaire dans Epidata, il est souvent utile de créer des notes ou des commentaires. Ces commentaires servent par exemple à faciliter la compréhension d'un mot dans le questionnaire. La fonction *Notes de saisie* peut être utilisée pour rédiger de courts commentaires consultables aussi bien lors de la saisie des données que quand le fichier n'est pas ouvert.

Lors de la saisie des données, les commentaires peuvent être consultés en appuyant sur la touche F8. Si aucun commentaire n'existe pour le fichier en cours, un nouveau fichier sera créé. L'heure et la date courante (date système) sont insérées automatiquement dans les commentaires.

La fonction *Notes de saisie* peut également être consultée et modifiée dans le menu Document.

Les commentaires seront inclus si la fonction Copie de sauvegarde des données est sélectionnée.

Libellé du fichier de données

Quand un fichier est créé, son titre peut être enregistré dans un court texte (50 caractères). Le titre du fichier est sauvegardé dans le fichier REC.

Le titre du fichier est consultable par la fonction Documenter et il est exporté quand les données sont exportées au format STATA.

Le titre du fichier peut être modifié en utilisant le menu Outils, Editer le libellé du fichier de données.

Liste des données

La fonction Liste des données peut être utilisée pour éditer un listing de tout ou partie des données contenues dans le fichier sélectionné.

Sélectionner la fonction Liste des données dans le menu Documenter.

Une boîte de dialogue apparaît permettant de sélectionner le nom du fichier base de données dont on souhaite réaliser un listing.

Une nouvelle boîte de dialogue propose de paramétrer les options d'affichage du listing.

Les options sont :

Selectionner les champs	<p>Cocher ou décocher les champs de la base de données à sélectionner pour le listing.</p> <p>Cliquer sur Tout pour sélectionner tous les champs. Cliquer sur Aucun pour désélectionner tous les champs.</p>
Dimensions de la liste	<p>La longueur de ligne du listing peut être modifiée pour que la sortie soit celle désirée.</p> <p>Le nombre de colonnes dans le listing peut être modifié.</p> <p>Si la longueur de la ligne et le nombre de colonnes sont modifiés simultanément, un nouveau calcul de largeur des colonnes est effectué par le programme.</p> <p>Attention, les données dont le format sera plus long que la largeur des colonnes, seront tronquées (repérables par un "--").</p>
Utiliser l'index comme clé de tri	<p>Si cette option est activée, les données apparaissent ordonnées selon les champs index (KEY-field) du fichier CHECK.</p>
Afficher libellés à la place des données	<p>Cocher cette case, si les intitulés des variables doivent être visibles.</p>
Ignorer les enreg. effacés	<p>Cocher cette case, si les enregistrements marqués pour suppression ne doivent pas apparaître dans le listing.</p>

Le listing apparaît dans l'éditeur de texte d'Epidata. Si la liste est trop importante pour l'éditeur (plus de 64 kb), le listing est sauvegardé dans un fichier.

Table des codes

La fonction Table des codes se trouve dans le menu Documenter. Sélectionner le fichier à partir duquel doit être réalisé. Le résultat apparaît dans l'éditeur d'Epidata.

La table des codes contient des informations sur les données saisies dans la base, dont le nombre d'enregistrements et le nombre d'enregistrements marqués pour suppression. Pour chaque champs d'entrée de données, apparaissent l'intitulé du champs (libellé), les contrôles de saisie, et le nombre de valeurs non renseignées. Pour les champs numériques, l'étendue (valeur minimum – valeur maximum, à ne pas confondre avec les valeurs seuil des contrôles), la moyenne et l'écart type sont précisés. Pour les champs qualitatifs le tableau de fréquence est calculé.


Double entrée et validation

Afin de s'assurer de la qualité des données saisies, il est possible de recourir à la stratégie de la validation en double saisie (saisie de données identiques par 2 personnes différentes, dans 2 fichiers différents). Quand toutes les données ont été saisies les deux fichiers de données peuvent être comparés en utilisant la fonction " valider une double saisie" trouvé dans le menu outils dans l'écran principal (quand tous les dossiers sont fermés).

Il est possible de préparer la double entrée en utilisant la fonction " Copier la structure du fichier REC", dans le menu outils. La structure (et non les données) sera copiée d'un fichier de données à un nouveau fichier de données. La structure de Fichier de données de copie a une option permettant d'omettre les variables texte puisque celles-ci sont souvent entrées deux fois.

Choisi validez double saisie quand toutes les données ont été saisies deux fois dans deux fichier de données différents. Affectez un nom à chacun des deux dossiers. Ensuite une boîte de dialogue est affichée proposant les options de la validation.

Choisissez les zones de tri

Afin de comparer deux fichier de données, un ou plusieurs champs-clef doivent être choisis. Les champs-clef choisis sont employés pour croiser " le fichier 1 fichier au fichier 2. La liste des clés sélectionnables ne propose que les champs communs aux 2 fichiers. Les champs qui sont assortis de la commande KEY dans le fichier CHK sont repérés par un symbole . Ceci n'a qu'une valeur d'information, car il n'est pas nécessaire que les champs utilisés comme clé de validation soient des clés de tri (KEY) dans le fichier CHK.

Si aucune zone de tri n'est choisie alors les deux fichier de données sont comparés sur une base fiche à fiche, c.-à-d. que la fiche1 du fichier de données 1 est comparée à la fiche1 dans le fichier 2 . Les données des deux fichiers doivent donc avoir été saisies dans le même ordre dans ce cas.

Options

Ignorez les enregistrements supprimés - des fiches marquées comme supprimées ne sont pas traitées pendant le procédé de validation

Ignorez les champs texte - des champs du type texte et texte majuscule ne sont pas traités pendant le processus

Ignorez la casse des lettres dans des champs textes – dans ce cas, " Smith " est équivalent à " sMiTh "

Montrer les différences de types de champs- si cette option est cochée, le rapport de validation inclura une information sur les champs dans les deux fichier qui ont le même nom, mais dont le type est différent.

Cliquez OK pour démarrer la validation. Les 2 fichiers sont comparés, et un rapport de validation est affiché.

Exportez les données

L'exportation des données inclue [la sauvegarde des données](#) et [export au format texte](#), [au format dBase III](#), [au format Excel](#) and [au formatStata](#)

Sauvegarde des données

Choisissez un fichier de données sauvegardes, et choisissez un dossier de destination. Cliquez OK pour commencer la sauvegarde. Une copie du fichier de données choisi est créée, ainsi que des fichiers .QES et CHK portant le même nom, et des fichiers de documentation éventuels portant le même nom.

Exportation vers le texte-dossier

Exporte un fichier de données vers un fichier texte standard de windows, comportant un fiche par ligne. Les champs sont séparés par un caractère optionnel.

Choisissez le fichier de données à exporter et un nom du dossier de destination à créer. L'extension du nom de fichier de destination doit être TXT.

Les options sont:

- Séparateurde champs - choisit le caractère de séparation des champs.
- Utiliser ("") comme repère de texte: Si cette option est sélectionnée, tous les champs non-numériques seront encadrés de guillemets.
- Ne pas exporter les enregistrements effacés- si cette option est choisie, des fiches marquées comme supprimées ne sont pas exportées.

Export au format dBase III

Ce format est très répandu.

Mode d'exportation des champs

Champ EpiData

Nombre entier, Idnum

Nombres en virgule flottante

Texte, texte majuscule, soundex

Booléen

Dates (dmy et mdy), date d'aujourd'hui (dmy et mdy)

champ dBase III

N - nombre fixe avec 0 chiffre après le séparateur décimal

N - nombre fixe

C - Caractères

L - Logique (Y=true, N=false)

D - date au le format YYYYMMDD

Note : que le format dBase limite le nombre de champs à 128.

Les options sont:

- Ne pas exporter les enregistrements effacés- si cette option est choisie, des fiches marquées comme supprimées ne sont pas exportées.

Export Excel

Exporte un fichier de données vers un fichier excel. La version de excel utilisée est la 2.1 en raison de sa relative simplicité. Cette version de Excel- est compatible avec toutes autres versions de excel.

Des champs sont exportés As

Champ EpiData

Nombres en virgule flottante
 Texte, texte majuscule, soundex
 Booléen
 Dates (dmy et mdy), date d'aujourd'hui (dmy et mdy)

Format de cellule Excel

Nombre
 Étiquette
 Logique (1=true, 0=false)
 Nombre de date périodique (composé comme date)

Note les feuilles de calcul de Excel ont des limites en termes de nombre de colonnes et de nombre de lignes. Ces limites sont différentes dans chaque version d'Excel. Après l'export, assurez vous que toutes les données / fiches ont été exportées.

Les options sont:

- Ne pas exporter les enregistrements effacés- si cette option est choisie, des fiches marquées comme supprimées ne sont pas exportées.

Exportation vers Stata

Exporte un fichier de données vers Stata version 4, version 5 (format identique à la version 4) ou version 6.

Le format fichier de données pour Stata inclut l'étiquette, les étiquettes de fichier, de variables et de valeurs. Note : les versions 4/5 de Stata permettent seulement les étiquettes de valeur jusqu'à 8 caractères. Si les étiquettes sont trop longues elles seront tronquées.

Champ EpiData

Nombre entier
 - longueur < 3
 - longueur < 5
 - longueur < 10
 - longueur >= 10
 virgule flottante
 Booléen
 Dates (dmy et mdy), date d'aujourd'hui (dmy et mdy)

 Texte, texte majuscule
 Soundex

Type de variable Stata

Byte
 Nombre entier
 nombre entier Long
 Double réel
 Double réel
 Byte (0=faux, 1=vrai)
 Nombre de date périodique (composé comme date en utilisant %d)
 Chaîne
 Chaîne (longueur = 5)

Les options sont:

- Ne pas exporter les enregistrements effacés- si cette option est choisie, des fiches marquées comme supprimées ne sont pas exportées.
- Choix du format d'exportation à la version 4/5 de Stata ou à la version 6

Options

Quand toutes les fenêtres sont fermées ou quand une fenêtre d'éditeur est ouverte, les options d'EpiData peuvent être modifiées en choisissant fichier | options.

Les options sont réparties sur plusieurs onglets. Vous pouvez activer un onglet en cliquant dessus ou par Ctrl+TAB.

Note : que les changements des options ne sont saués sur le disque qu'à la fermeture d'EpiData.

Options de l'éditeur

Changement de police et de couleur de fond dans les fenêtres de l'éditeur. Veuillez noter que toutes les fenêtres d'éditeur ouverts sont mis à jour avec la nouvelles police et couleur quand des options sont validées en cliquant OK

Changement du nombre d'espaces insérés à l'aide de la touche Tab. EpiData ne peut pas utiliser les caractères de tabulation, ces caractères sont automatiquement remplacés par le nombre d'espaces- définis dans la page d'options de l'éditeur quand un fichier est ouvert ou quand du texte est inséré (collé) d'un autre programme.

Options du masque

Définit la façon dont un fichier de données est affiché

Vous pouvez changer la police et le la couleur d'arrière plan des masques de saisie. Vous pouvez également indiquer une couleur différente pour le fond des champs de saisie et choisir une autre couleur pour le champ actif.

D'autres options relatives à l'apparence des champs de saisie (Style 3D, à plat encadré ou à plat non encadré), hauteur des lignes (espacement), et le nombre de pixel insérés par le caractère tabulation (@) dans des fichiers de données.

Options créer le fichier de données

Définit comment des noms de champs sont définis quand 'créez fichier de données' est lancé. Il est possible de choisir les noms en majuscules, et le nom du champ peut être construit à partir du 1^{er} mot de la question ou automatiquement défini. Voir Noms de champ

Options de documentation

Définit l'apparence des fenêtres de l'éditeur à l'affichage de la documentation (documenter fichier de données, liste des données, table des codes).

Options avancées

ID-number: Définit le premier nombre utilisé dans les nouveaux dossiers qui contiennent un champ

Messages d'erreur: Si l'option est cochée, les messages dus à des erreurs dans l'utilisation des expressions IF et LET dans les fichiers CHK seront affichées lors de la saisie.. Cette fonction peut aider à trouver la cause du dysfonctionnement d'une condition lors de la saisie. Si l'option n'est pas cochée, les erreurs dans les expresions IF et LET les expressions sont ignorés pendant le la saisie.

Langue définit la langue employée dans les menues, les boutons, les messages d'erreur etc. dans EpiData. Veuillez vous référer à la section

Rétablir les options par défaut: Reconstitue toutes les options aux valeurs par défaut.

Barres d'utilis

EpiData affiche des barres d'outils en haut des fenêtre de programme, la barre d'outils d'organisation du travail- et labarre d'outils de l'éditeur. Toutes les fonctions affichées dans les barres d'outils peuvent être trouvées dans les menus (fichier, éditer...)) et une ou les deux barres d'outils peut être cachée.

Le masquage ou l'affichage d'une barre d'outils se fait en cliquant avec le bouton droit sur la barre concernée, et en cochant sur l'une ou l'autre des options proposées. Ceci peut également être fait en employant le menu Fenêtre | Barres d'utilis.

Dans le même menu se trouve la fonction 'masquer les barres d'outils durant la saisie'. Cette option est cochée par défaut. Les 2 barres d'outils ne seront pas visibles durant la saisie, libérant plus de surface écran.

Clefs de raccourci

Éditeur

CTRL+n	Nouvelle fenêtre de l'éditeur
CTRL+o	Ouvre le fichier QES existant
CTRL+s	Sauvegarde du fichier QES (sans affichage de la fenêtre de fermeture)
CTRL+p	Imprime le contenu de la fenêtre de éditeur
CTRL+a	Sélectionner tout le texte
CTRL+c	Copie le texte sélectionné dans le presse papier
CTRL+x	Coupe le texte sélectionné et le place dans le presse papier
CTRL+v	Insère le texte du presse papiers
CTRL+z	Annule la dernière modification
CTRL+g	Aller à la ligne...(Demande le numéro de ligne)
CTRL+f	Chercher texte
CTRL+r	Chercher et remplacer texte
CTRL+q	Montre/ active le sélecteur de champ
CTRL+t	Prévisualiser le masque de saisie

Ajouter / modifier contrôles

Lorsque le pointeur est sur un champ de saisie

CTRL+Home	Aller au premier champ
CTRL+End	Aller au dernier champ
F5	Active la fenêtre des fonctions de contrôle
CTRL+Flèche droite	Active la fenêtre des fonctions de contrôle
CTRL+i	Edite Range/Legal
CTRL+j	Editez JUMPS
CTRL+e	Bascule MUSTENTER entre oui et non
CTRL+r	Bascule REPEAT entre oui et non
CTRL+a	Edite les étiquettes de valeur
CTRL+d	Edite tous les contrôles pour le champ courant
CTRL+c	Copie tous les contrôles du champ courant au presse papier
CTRL+x	Coupe tous les contrôles du champ courant vers le presse papier
CTRL+v	Insérez les contrôles du presse papier
ALT+s	Sauvegarder le fichier
ALT+d	Édite tous les contrôles du champ courant
ALT+x	Quitter

Lorsque le pointeur est dans l'éditeur de fonctions de contrôle

F5	Retourne au masque de saisie
CTRL+Flèche gauche	Retourne au masque de saisie
Entrez	Va sur le contrôle suivant
Flèche haut	Va sur le contrôle précédent
Flèche bas	Va sur le contrôle suivant
CTRL+PgUp	Active le champ précédent
CTRL+PgDn	Active le champ suivant
ALT+s	Sauvegarder le fichier
ALT+d	Éditez tous les contrôles du champ courant
ALT+x	Quitter

Saisir des données

CTRL+n	Nouvelle fiche
SHIFT+Delete	Marquez la fiche comme supprimée / réactiver une fiche préalablement supprimée
CTRL+PgUp	Fiche précédente
CTRL+PgDn	Fiche suivante
CTRL+ALT+Home	Première fiche
CTRL+alt+end	Dernière fiche
CTRL+Home	Premier champ dans la fiche courante
CTRL+End	Dernier champ dans la fiche courante
CTRL+g	Aller à la niche numéro...
CTRL+f	Trouvez les enregistrements vérifiant la valeur du champ courant
F3	Continuer la recherche lancée avec CTRL-f
CTRL+ArrowKeyLeft	Recale le masque sur la marge gauche
F9 ou ++key numérique	Ouvre la liste de valeurs légales (si disponible)
F8	Ouvre <u>les notes de saisie</u>

Programmez les paramètres

EpiData peut être exécuté en utilisant un ou plusieurs des paramètres suivants. Les paramètres peuvent être indiqués dans un fichier .BAT ou en employant un Lien Windows.

<i>nom de fichier.</i> QES	Ouvre le dossier qes indiqué au démarrage du programme
<i>nom de fichier.</i> REC	Ouvre le fichier de données indiqué au démarrage du programme
/ NOTOOLBARS	Cache les deux barres d'outils
/ SAVE	Supprime le message " Sauvegarder sur le disque?" ; la sauvegarde est alors effectuée automatiquement. / AUTO a le même effet et est conservée pour assurer la compatibilité avec Epi Info 6,04. SAVE existe également comme commande de contrôle.
/ CONFIRM	Supprime la fonction qui déplace systématiquement le pointeur sur le champ suivant dès qu'un champ est rempli. CONFIRM existe également comme commande de contrôle.
/ LANG = <i>langue</i>	Choisit la langue . <i>langue</i> indique la première partie du nom de fichier d'un dossier de language.lang.txt. Pour forcer l'utilisation de l'anglais employez ce paramètre: / LANG=English Voir Internationalisation

Si plus d'un nom de fichier est employé, seulement le dernier dossier sera ouvert.

Exemple:

Un fichier de commande appelé CHILDPRJ.bat est créé contenant cette ligne de commande:

```
EPIDATA.exe CHILDPRJ.rec / NOTOOLBARS
```

Quand ce fichier est lancé, EpiData ouvrira le Childprj.Rec en saisie, en masquant les deux barres d'outils.

Internationalisation

EpiData est un logiciel de langue anglaise qui peut-être adapté dans d'autres langues afin d'obtenir des versions "langues locales" avec des menus, des boutons et des messages d'erreur dans la langue locale.

Le choix de la langue se fait au niveau du menu File de l'écran principal en choisissant l'option Advanced. Pour adapter le logiciel dans une autre langue que l'anglais, le fichier *langue* doit être présent dans le même répertoire que le fichier EpiData.exe.

Ces fichiers *langue* sont normés langue.lang.txt. Ainsi, pour une adaptation en version espagnole, il doit exister un fichier espagnol.lang.txt.

Les fichiers langue sont disponibles sur le site www.EpiData.dk. Si le fichier langue correspondant n'existe pas sur le site et que vous souhaitez participer à sa traduction, contactez Info@EpiDate.dk pour plus d'informations.

Lorsque la boîte de dialogue de l'option Advanced est affichée, la liste de tous les fichiers XXX.lang.txt disponibles dans le répertoire est à l'écran.

Pour ajouter une nouvelle langue, vous devez donc télécharger le fichier *langue* correspondant sur le site www.EpiData.dk. et l'enregistrer dans le même répertoire que le fichier EpiData.exe.

EpiData fait appel à trois fichiers d'aide dont le nom est, pour leurs versions standards c'est à dire en anglais :

EpiData.hlp : fichier d'aide

EpiTour.hlp : fichier de présentation

ReadMe.rtf : présentation de l'installation du logiciel

Ces fichiers peuvent eux aussi exister en version "langue locale". Dans le fichier langue.lang.txt est indiqué un code langue (Fr pour la France). Ce code est employé pour retrouver la version "langue locale" des trois fichiers d'aide. Le nom de ces fichiers est composé par adjonction du code langue au nom des fichiers standards.

Exemple :

EpiData_fr.hlp

EpiTour_fr.hlp

ReadMe-fr.hlp

Si la version "langue locale" de ces fichiers n'est pas disponible, le logiciel utilise les fichiers standards (en anglais).

Format de champs EpiData

Type de champ	Exemple
<u>Identifiant</u>	< IDNUM >
<i>Numérique</i>	### ###.##
<u>Texte</u>	_____
<u>Texte majuscule</u>	< A > < A >
<u>Booleen</u> (oui/non ou vrai/faux)	< Y >
<u>Date</u>	< dd/mm/yyyy > < mm/dd/yyyy >
<u>Date du jour</u>	< today-dmy > < today-mdy >
<u>Soundex</u>	< S > < S >
<u>Marque de tabulation</u>	@

Identifiant

< IDNUM >

IDNUM est un champ d'identification numérique automatique qui s'incrémente de 1 en 1 à chaque nouvel enregistrement. Cet identifiant numérique ne peut pas être modifié en cours de saisie.

La première valeur par défaut pour un nouveau fichier de données est égale à 1, mais elle peut-être modifiée par l'option Dataform du menu File.

Champs numériques

```
###
###.###
#####
##.####
```

Les champs numériques acceptent la saisie de chiffres, du signe "-" et de la marque de décimale. Dans les fichiers .QES et en cours de saisie de données les "." aussi bien que les "," sont acceptés comme marque de décimale.

Une seule marque de décimale est acceptée par le champ ce qui signifie que les séparateurs de 1000 ne peuvent pas être saisis.

Le nombre de caractères (marque décimale incluse) définit la longueur du champ. Sa longueur maximale est de 14 caractères.

Champs texte

```
_
_____
_____
```

Le nombre de caractères "_" définit la longueur du champ.

Les champs texte acceptent tous les caractères.

La longueur maximum de champ est de 80 caractères.

Champs texte majuscule

```
< A >
< A  >
```

Les champs texte majuscule acceptent tous les caractères, mais les entrées seront converties en majuscules. La longueur du champ est définie par le nombre de caractères entre les signes "<" et ">" y compris le "A". Dans les exemples donnés ci-dessus, les longueurs de champs sont respectivement 1 et 5.

Champs booléens (oui/non ou vrai/faux)

<Y>

Les champs booléens acceptent, à la saisie, les valeurs légales suivantes Y, N, 1, 0 et espace. Les valeurs " 1 " sont converties en " Y " et " 0 " en " N ".

La longueur des champs booléens est par définition 1. Ceci signifie qu'une formation de champs <Y > dans le .QES créera une erreur.

Champs Date

<dd/mm/yyyy>

<mm/dd/yyyy>

Les champs date existent sous 2 formats : un format européen avec le jour avant le mois et un format américain avec le mois avant le jour.

La longueur des champs de date est toujours de 10 caractères.

A la saisie, les caractères légaux sont les chiffres et le slash "/". Les dates peuvent être saisies sans employer de "/" si tous les chiffres sont saisis. Ainsi, et pour un format de date européenne, le 4 mai 1999 peut être saisi "04051999". Quand le curseur passera au champ suivant, le champ sera remis au format standard ("04/05/1999").

Il n'est pas nécessaire de saisir chacun des 8 chiffres. Si vous saisissez "040599" le champ sera automatiquement formaté en "04/05/1999". Les deux premiers chiffres de l'année sont extrapolés comme suit :

- Si vous saisissez un nombre entre 50 et 99, l'année est considérée comme appartenant au 19^{ème} siècle (19_ _).
- Si vous saisissez un nombre entre 00 et 49, l'année est considérée comme appartenant au 20^{ème} siècle (20_ _).

Si vous saisissez "0405", c'est l'année en cours qui est prise en compte.

Après la saisie, tous les champs date sont contrôlés pour vérifier que les dates saisies correspondent bien à un format de date autorisé.

EpiData n'accepte que les années sur 4 chiffres.

Champs date du jour

<today-dmy>

<today-mdy>

Des champs "date du jour" seront automatiquement remplis par la date du jour en cours dans le système de l'ordinateur au moment de la saisie. Ce type de champs ne peut pas être édité, ni saisi.

Si un enregistrement précédemment sauvé contenant un champ date du jour est modifié et que les modifications sont sauvées, le champ date du jour sera remis à la date courante. Ce format de champ peut être utilisé comme un marqueur de la date de dernière modification.

Alors qu'il n'existe pas dans Epi-Info un format date du jour en date européenne (jma) est disponible dans Epidata.

Champs de Soundex

<S>

<S >

Les champs Soundex acceptent toutes les entrées mais seules les lettres du dernier mot saisi seront utilisées pour créer le code soundex.

Le Soundex est un codage de texte qui peut être utilisé pour crypter le nom de famille des sujets d'une surveillance et ainsi anonymiser le recueil informatique. Un code Soundex est toujours composé d'une lettre majuscule, d'un signe moins et de trois chiffres.

1/ La première lettre du mot est toujours conservée. Le reste du mot est codé sur 3 chiffres en suivant le schéma de codage ci-dessous :

U N E I O U Y H W	Non codé
B F P V	Codé 1
C G J K Q S X Z	Codé 2
D T	Codé 3
L	Codé 4
M N	Codé 5
R	Codé 6

2/ Les consonnes situées après la première lettre du mot sont codées dans l'ordre où elles apparaissent dans le mot.

HOLMES = H-452

ADOMOMI = A-355

3/ Le code ne comprend que 3 chiffres correspondant aux 3 premières consonnes (hors 1^{ère} lettre). Les consonnes suivantes sont ignorées.

VONDERLEHR = V-536

4/ Des zéros sont employés pour compléter le codage des mots courts :

BALL = B-400

SHAW = S-000

5/ Les doubles consonnes sont traitées comme une seule :
BALL = B-400

6/ De même que les consonnes adjacentes du même groupe de consonnes :
JACKSON = J-250

7/ Si le mot commence par 2 (ou plusieurs) consonnes la deuxième (et énième) est ignorée :
SCANLON = S-545

8/ Les apostrophes et les traits d'union sont ignorés :
KING-SMITH = KINGSMITH = K-525

9/ Les consonnes du même mot à coder séparé par un W ou un H sont traitées comme une seule :
BOOTH-DAVIS = B-312

Marque de tabulation

Lorsqu'on crée un questionnaire, les champs du fichier .QES sont positionnés de façon automatique : nom de la variable suivit de son format.

Ce positionnement automatique peut occasionner un non alignement des champs de saisie.

La marque de tabulation permet d'aligner les champs de saisie. Cette marque de tabulation n'existe pas dans Epi-Info où elle serait traitée comme tout autre caractère.

La marque de tabulation n'affecte ni le format de la variable, ni son nom, ni le fichier de données (fichier .REC). Elle ne fait qu'aligner les champs de saisie dans le questionnaire.

Insérer une marque de tabulation "@" juste avant le format de la variable, permet d'aligner le champ au niveau du point de tabulation suivant.

```
v1@####  
v20@####
```

Ces lignes de questionnaire créeront deux champs à 4 chiffres. Le libellé de la question sera aligné sur la marge gauche de la page. Le bord gauche ces 2 champs de saisie sera aligné au point de tabulation gauche suivant. Les espaces de tabulation sont définis en Pixel. Par défaut, les espaces de tabulation sont de 40 Pixel (1 point de tabulation tous les 40 Pixel) mais cette valeur peut être modifiée en passant par le menu Fichier | options | Fichier de données.