

**Ecole Nationale
Supérieure de
Bibliothécaires**

**Université
Claude Bernard
Lyon I**

**DESS Informatique
Documentaire**

Rapport de stage

CREATION D'UNE BASE DE DONNEES INTERNE A PARTIR DE REFERENCES
BIBLIOGRAPHIQUES TELEDECHARGEES POUR LA BIBLIOTHEQUE DE L'E.N.S.B.

Maria Elizabeth BROXADO

sous la direction de Mouhamed HASSOUN
Professeur à l'E.N.S.B.

et

Danielle ROGER
Conservateur de la bibliothèque de l'E.N.S.B.

1991

1991
Stages
57

**Ecole Nationale
Supérieure de
Bibliothécaires**

**Université
Claude Bernard
Lyon I**

**DESS Informatique
Documentaire**

Rapport de stage



**CREATION D'UNE BASE DE DONNEES INTERNE A PARTIR DE REFERENCES
BIBLIOGRAPHIQUES TELEDECHARGEES POUR LA BIBLIOTHEQUE DE L'E.N.S.B.**

Maria Elizabeth BROXADO

**sous la direction de Mouhamed HASSOUN
Professeur à l'E.N.S.B.**

et

**Danielle ROGER
Conservateur de la bibliothèque de l'E.N.S.B.**

1991

TABLE DE MATIERES

RESUME	1
INTRODUCTION.....	2

PREMIERE PARTIE

I. La Bibliothèque de l'E.N.S.B.	3
---------------------------------------	---

DEUXIEME PARTIE

I. Analyse	5
II. Le programme	9
1. Le langage de programmation Turbo Pascal	9
III. Choix du logiciel	11
1. Le logiciel Texto	11
2. Création d'un fichier Texto	11
3. Le document de paramètres	12
4. L'édition	15
5. La table d'appauvrissement.....	15
6. Les index	16
7. L'interrogation.....	17
7.1 La syntaxe	17
7.2 Les procédures.....	17
8. Le document de dialogue.....	19

CONCLUSION	20
BIBLIOGRAPHIE.....	21
ANNEXES	22

Création d'une base de données interne à partir de références
bibliographiques téléchargées pour la Bibliothèque de
l'E.N.S.B.

Maria Elizabeth BROKADO

Stage effectué du 2 Juin au 30 Septembre 1991 à l'Ecole
Nationale Supérieure de Bibliothécaires
17/21 Bv du 11 novembre 1918
69523 Villeurbanne - FRANCE
Tel.: 78 89 64 45

RESUME :

La Bibliothèque de l'E.N.S.B. a mis en place une base de données interne alimentée par des notices téléchargées des banques de données bibliographiques. Ce rapport fait la description des étapes suivies pour l'installation de cette base de données, à l'aide du logiciel documentaire TEXTO.

DESCRIPTEURS :

Base donnée interne ; Informatique ; PASCAL ; Logiciel ;
*TEXTO.

ABSTRACT :

The E.N.S.B. library has settled an in-house data base fed by down-loaded records from bibliographical databanks. This report describes the attended training course during the settlement of this data bank with TEXTO software.

KEYWORDS :

In-house data base ; Computer science ; PASCAL ; Software ;
*TEXTO.

[] Possibilité de diffusion

[] Confidentiel

INTRODUCTION

L'objectif du stage effectué pendant 4 mois (de juin à septembre 1991) à la Bibliothèque de l'Ecole Nationale Supérieure de Bibliothécaires était :

- l'analyse et la conception d'une base de données locale, qui contient des références provenant du téléchargement des banques de données bibliographiques;
- la réalisation, en langage TURBO PASCAL, d'un programme de conversion de notices ;
- le transfert des fichiers de données en fichier TEXTO, en format "ajout piloté";
- la mise en place de modules pour l'utilisation et la consultation de cette base de données.

PREMIERE PARTIE

I - LA BIBLIOTHEQUE DE L'E.N.S.B.

Créée en 1964, l'Ecole Nationale Supérieure de Bibliothécaires est un établissement public à caractère scientifique, culturel et professionnel. Sous la tutelle du Ministère de l'Education Nationale, de la Jeunesse et des Sports, elle est chargée de la formation de bibliothécaires et spécialistes de l'information des secteurs public et privé.

Sa bibliothèque, spécialisée en Sciences de l'Information et de la Communication, Bibliothéconomie et Documentation, offre en libre accès un fonds d'environ 27.000 ouvrages et 230 périodiques en cours, quelques documents sur CD-ROM. Elle compte au total 300 lecteurs inscrits provenant des élèves de l'Ecole (DSB), ses enseignants et son personnel, des élèves du CAFB, les élèves du DESS d'informatique documentaire, les étudiants du DEA en sciences de l'information et de la communication et quelques lecteurs autorisés.

Le prêt est réservé aux lecteurs inscrits et la bibliothèque se charge aussi de faire venir par prêt entre bibliothèques, le document qui n'a pas été localisé dans les bibliothèques de Lyon.

Etant donné que les élèves, à l'occasion de l'élaboration de leurs mémoires ou notes de synthèse, ont besoin d'interroger des banques de données, la bibliothèque prend en charge ce service et elle le propose aussi à tous ses lecteurs.

Quelques produits nous sont offerts par cette bibliothèque:

- revue de presse pour les enseignants de l'Ecole,
- dossiers documentaires,
- dossiers techniques,
- liste bimensuelle de divulgation des nouvelles acquisitions, et
- revue mensuelle des sommaires de périodiques.

En ce qui concerne sa gestion, elle n'utilise pas un système intégré. Les périodiques sont gérés par MICROLINK de Faxon, et le catalogage est réalisé en coopération avec la base OCLC. Le prêt est effectué par un logiciel maison qui a été mis en place depuis l'été 1990, réalisé par une élève stagiaire de l'Ecole.

Des équipements comme: un lecteur de microfiches, trois postes de CD-ROM, un lecteur de vidéodisques et une photocopieuse sont en libre service pour ses usagers.

Pour répondre à ses besoins et étendre sa croissance, la bibliothèque de l'E.N.S.B. est sur le point d'acquérir un système de gestion de bibliothèque informatisé.

DEUXIEME PARTIE

I - ANALYSE

La Bibliothèque mémorise certains des services de recherche dans les banques de données. Ces interrogations sont sauvegardées dans le répertoire KORTX, qui contient un nombre important de fichiers. Les sujets sont variés et certains d'entre eux sont d'intérêt pour l'Ecole. C'est pour cela qu'il a été décidé de mettre en place une base de données bibliographique locale, à l'usage strictement interne et qui pourra avoir une fonction pédagogique.

Elle sera alimentée par des fichiers provenant du téléchargement de notices des banques de données suivantes :

- 1- FRANCIS (CNRS - Centre de Documentation des Sciences Humaines);
- 2- PASCAL (CNRS - Centre de Documentation Scientifique et Technique);
- 3- LISA (The Library Association) et
- 4- INSPEC (Institute of Electrical Engineers).

Cette base multidisciplinaire sera installée sur un ordinateur compatible PC de la Bibliothèque et pourra être interrogée par les professeurs et les élèves de l'Ecole.

L'étape initiale de l'analyse a comporté une étude de la structure des notices de chacune des banques de données interrogées fréquemment à l'Ecole (voir Annexes 1/4). Pour mener à bien cette tâche, nous avons utilisé les informations diffusées dans les fiches techniques fournies par chacune des

banques de données. Au niveau de la présentation, cette analyse a mis en évidence des différences très importantes de format d'une banque de données à l'autre. A travers cette étude, notre objectif était d'établir une liste des champs que nous voulons conserver dans notre base de données. Une analyse plus détaillée a été reprise plus tard, avant l'écriture des algorithmes de traitement.

En fonction de l'application prévue pour cette base, nous avons retenu les champs suivants :

- 1- Le numéro d'accès du document;
- 2- Le nom de la banque de données;
- 3- L'auteur ou les auteurs;
- 4- Le titre du document;
- 5- La langue du document;
- 6- L'affiliation de l'auteur;
- 7- La source;
- 8- La cote;
- 9- Le résumé;
- 10- Les descripteurs.

La deuxième étape comprenait la création d'une table de transfert, permettant la transformation de ces champs en champs numériques.

Ce travail a été réalisé en utilisant une structure appauvrie du format MARC (Machine-Readable Cataloguing) adopté à la Bibliothèque du Congrès (LC-MARC). Les lignes directrices de ce format d'échange sont conformes aux règles de catalogage anglo-américaines (AACR2). Effectivement, ce format d'échange n'est pas adapté au type de document qui alimentera notre base de données. Mais, la Bibliothèque de l'E.N.S.B. veut conserver la possibilité ultérieurement dans la même base des notices provenant de la base OCLC (Online Computer Library Center).

Pour pouvoir définir les équivalences, nous avons examiné les 2 volumes du manuel "Format de catalogage des livres" (OCLC). Dans le cas où les champs pouvaient coexister, il ne s'agissait que de découvrir les champs LC-MARC correspondants. Pour les autres, nous les avons choisi parmi ceux qui étaient vides.

CHAMPS	EQUIVALENCES
Nom de la banque de données	035a
Numéro d'accès du document	035b
Langue du document	042
Cote	093
Auteur	100a
Affiliation de l'auteur	100u
Titre	200a
Source	440a
Résumé	520
Descripteurs en anglais	690e
Descripteurs en français	690f

Après cette étape, nous pouvons préciser quel type de fichier nous allons traiter.

Comme la structure d'une notice n'est pas la même dans les 4 banques de données, nous avons décidé de fixer les champs qui doivent obligatoirement figurer dans ces fichiers. Quelques champs sont essentiels pour une base de données, comme l'auteur, le titre, la source et les descripteurs. Un fichier ne sera traité que s'il fournit au moins ces informations. Cette décision nous a permis de connaître les formats d'affichage¹ propres au programme de conversion de notices :

- FRANCIS : max, fmax ou emax;
- PASCAL : max, fmax ou emax;
- INSPEC : 2, 5 ou 9;
- LISA : 2, 5 ou 9.

¹- Pour avoir des informations sur les formats d'affichage, consulter le manuel de l'utilisateur de chaque banque de données.

A la suite, nous avons effectué une recherche dans le répertoire KORTEX pour essayer de trouver un fichier de chaque banque de données au format d'affichage complet.

Ces fichiers sources ont été soumis à une analyse détaillée.

Pour pouvoir les lire nous les avons chargés dans l'éditeur de Turbo Pascal. Plusieurs problèmes se sont présentés au moment de la lecture :

- La structure des notices parfois n'était pas celle diffusée dans les fiches techniques des banques de données.

- Les fichiers PASCAL et FRANCIS affichaient quelques caractères bizarres, tout au long de l'interrogation au début de chaque ligne. Donc, nous avons décidé d'écrire un programme pour les éliminer.

- Le fichier INSPEC qui était trop volumineux ne pouvait pas être chargé dans l'éditeur de texte. Le système nous proposait sa troncature, mais la confirmation provoquait la perte des informations qui n'ont pas pu être chargées. Pour résoudre ce problème nous avons chargé ce fichier dans le traitement de texte WORD et nous l'avons divisé en deux.

- Par conséquence, ces fichiers ne sont pas normalisés. Pour pouvoir les traiter nous avons nettoyé toutes les interrogations, de façon qu'ils ne contiennent que des notices.

Ayant bien analysé ces fichiers, nous avons procédé à l'écriture des algorithmes de traitement. Cette étape nous a paru de grand intérêt, puisqu'elle nous a permis de connaître le cheminement à suivre en vue du traitement de ces notices.

II - LE PROGRAMME

1. Le langage de programmation Turbo Pascal :

Turbo Pascal est un langage de programmation très performant qui a été développé en Californie par Borland International. Le fait qu'il soit un langage compilé nous procure les avantages suivants :

- * fiabilité grâce à sa rigueur;
- * facilité et rapidité d'effectuer des modifications, de compiler le programme lui-même et de l'exécuter.

Très renommé pour sa puissance, Turbo Pascal est structuré de façon qu'on puisse découper les programmes en parties. Cette astuce facilite le traitement des données.

Les fichiers que nous avons traités sont de type texte. Nos fichiers résultats sont séquentiels, structurés de façon à respecter la convention TEXTO :

```

nom de champ
contenu
nom de champ
contenu
// (fin du premier document)
nom de champ
contenu
...

```

A la fin de chaque notice deux barres obliques (//) sont affichées. C'est le seul moyen en TEXTO pour reconnaître la fin d'une fiche (voir Annexe 5).

L'étude des fonctions et des procédures pour le traitement de chaînes de caractères au préalable à l'écriture du programme, s'est avérée fort utile. Nous avons donc consulté des ouvrages spécialisés, dont la liste se trouve à la fin du présent document.

Le programme de conversion de notices que nous avons écrit en utilisant la version 5.5 de Turbo Pascal s'appelle BASEDON (voir Annexe 4).

La normalisation d'un fichier s'est faite en 3 étapes :

1- Un nettoyage des caractères rajoutés par le système au cours du téléchargement utilisant le programme EFFACE (voir Annexe 6).

2- Un nettoyage manuel à l'intérieur de ces fichiers pour éliminer tout ce qui vient de l'interrogation.

3- Le traitement du fichier par le programme, qui comporte :

a) L'initialisation de certains éléments chargés par les procédures :

1- qui propose le choix du fichier à traiter et du fichier résultat;

2- qui dresse le nom des banques de données traitées par le programme;

3- qui liste le nom des champs à ignorer ou à conserver avec leur code.

b) Le traitement :

Cette étape correspond au traitement proprement dit de la notice. Il cherche le nom de la banque de données dans la première ou la deuxième ligne. S'il trouve, en fonction de chaque la banque de données il effectuera un traitement. S'il ne trouve pas il passera à la notice suivante, jusqu'à la fin du fichier.

Pour charger Turbo Pascal et le programme, il suffit de se placer sur C à partir du DOS, et taper TURBO BASEDON.

Pour l'exécuter on appuie à la fois les touches Ctrl et F9. Après l'exécution, nous avons les fichiers résultats contenant les notices traitées par le programme. Nous pouvons, donc passer à l'étape finale de notre stage, qui est l'installation de la base de données sur TEXTO.

III - CHOIX DU LOGICIEL

1. Le logiciel TEXTO :

Développé par CHEMDATA (Lyon, France), TEXTO est un logiciel de gestion documentaire d'application très répandue. Il a été choisi pour la création de cette base de données bibliographique, puisqu'il est déjà implanté sur les ordinateurs de la Bibliothèque de l'Ecole; ce qui évite l'achat d'un nouveau logiciel. En plus, son apprentissage est rapide.

Son système est paramétrable par l'utilisateur, ce qui rend possible la mise en oeuvre de traitements ajustés aux besoins des établissements.

TEXTO permet la création de fichiers liés entre eux, la saisie et la correction de données en mode "ligne à ligne". On peut également ajouter ou retirer un document, et modifier entièrement le contenu d'un ou plusieurs champs.

On avait besoin également, d'un système qui effectue le transfert de fichiers de données extérieures. Dans notre cas, il ne s'agit que d'un transfert physique, puisque TURBO PASCAL et TEXTO se trouvent installés sur le même ordinateur. TEXTO permet d'effectuer ces transferts en employant un format d'échange nommé "ajout piloté".

2. Création d'un fichier TEXTO :

Pour installer notre base de données bibliographique, nous avons utilisé la version 4.2.1. de TEXTO.

Nous avons créé un sous répertoire qui s'appelle BASEDON.

Dans TEXTO on travaille avec 3 types de fichiers fondamentaux :

1. Le catalogue, où on structure plusieurs types de documents, comme les fichiers qu'on veut créer (documents de paramètres). Il permet aussi de mémoriser les formats d'édition ou de tabulation, les suites d'instruction et les listes de mots vides;

2. Les fichiers documentaires, qui sont les fichiers proprement dits, qu'on peut créer autant qu'on veut, à la limite de l'espace disque; et

3. Les index ou "fichiers inversés" qui constituent en quelque sorte les index de fin de livres, qui sont générés à partir d'un ou de plusieurs champs. Pour chaque fichier documentaire il est possible d'avoir autant d'index qu'on désire. Ils permettent d'effectuer des tris et assurent une rapidité d'interrogation lors des recherches dans des fichiers de taille importante.

Tous les fichiers TEXTO sont caractérisés par un nom et contiennent une succession de documents de formalisme identique.

Un document se divise en champs, où chacun a un nom et un contenu. Le nombre de champs dans un document est limité à 99. Le choix du nom des champs est libre, mais il ne doit pas contenir plus de 8 caractères.

Le premier champs est formé toujours par la numérotation du document qui peut être soit définie par l'utilisateur, soit calculée par TEXTO.

Un contenu de champ est en format variable (sauf pour le premier qui contient le numéro), mais l'utilisateur peut imposer un format fixe. Ce contenu peut se diviser en articles, ce qui permet de rendre facile l'interrogation, ou bien former un tout. Le contenu est séparé du nom de son champ par un caractère spécial nommé séparateur champ/contenu.

Un document TEXTO ne peut pas comporter plus de 4000 caractères, mais certaines versions modifient ces limites, autorisant un plus grand nombre.

3. Le document de paramètres (DP):

Les champs que nous avons défini auparavant en utilisant la structure appauvrie de LC-MARC, ont servi à créer notre document de paramètres sur TEXTO.

Un DP comprend 3 champs :

1- Le champ "nom", qui contient le nom du document de paramètres, qui doit obligatoirement commencer par la lettre P et ne peut excéder 8 caractères;

2- Le champ "general", qui permet de définir le format et type de la numérotation, ainsi que les séparateurs d'articles et champ/contenu; et

3- Le champ "champs", où on liste le nom des champs qui figureront dans le document.

L'enregistrement d'un DP se fait par la commande ajout (a), suivie du nom du document dans le catalogue (/). Le nom doit obligatoirement commencer par la lettre P. Nous avons choisi le nom PBIBLIO pour notre document de paramètres.

```
nom      .pbiblio
general  .5 1 . {
champs   .REF 035a 035b 042 093 (?)100a,(a)ifusion 100u 200a,(a)titre 440a 520
          .690e,(a)ifusion 690f,(a)ifusion
```

Pour notre document de paramètres nous avons opté par :

- une numérotation automatique;
- le point comme séparateur champ/contenu et
- l'accolade comme séparateur d'articles.

Après avoir défini le DP, on doit choisir un nom pour le fichier, qui doit commencer par la lettre F. Notre fichier s'appelle F BIBLIO.

Maintenant il est possible de faire le transfert des fichiers traités par le programme, utilisant le format d'échange "ajout piloté". Pour cela, il suffit de lancer la commande "ap c:nom du fichier à transférer" et les notices sont déversées dans le fichier BIBLIO de TEXTO.

Voici l'exemple d'une notice traitée par le programme
et transférée utilisant ce format d'échange :

REF .00009
035a .PASCAL
035b .89-0363767
042 .ENG
093 .205A07A
100a .BURTON J; NEWPORT J; ROBINSON E
100u .Loughborough univ. technology, Pilkington/Loughborough Leics. LE11
.3TU/GBR
200a .OPACs and JANET: a simple technique for easy user access
440a .Program; ISSN 0033-0337; Coden PRGRDU; GBR; DA. 1989; VOL. 23; NO. 3;
.PP. 257-268; BIBL. 8 ref.
520 .Presentation illustree de copies d'ecran du logiciel realise par
.l'Universite Loughborough a partir de KERMIT et MENU, permettant
.d'accéder aux catalogues publics en ligne des autres bibliotheques
.d'enseignement superieur par l'intermediaire du reseau JANET (Joint
.Academic Network)
690f .Interface utilisateur; Logiciel; Acces information; Catalogue
.automatise; Traitement en ligne; Reseau bibliotheque; Reseau
.telecommunication; Bibliotheque universitaire; Royaume Uni; OPAC;
.JANET; KERMIT; MENU

4 - L'édition :

Nous avons réalisé une édition cataloguée. Nous avons opté pour une édition simple, qui comporte toutes les données d'une fiche classique.

On enregistre un document d'édition (DE) par la commande "a /nom de l'édition", ou bien "aw /nom de l'édition" pour travailler en plein écran. Ce nom doit nécessairement commencer par la lettre E.

```

nom      .edit
champs   .'BD      : ' 035a 'AUTEUR   : ' 100a 'TITRE    : ' 200a 'SOURCE   :
         .' 440a 'RESUME   : ' 520 'DESCRIPT : ' 690e 'DESCRIPT : ' 690f
titre    .
         .                      RESULTAT DE LA RECHERCHE
ligne    .65
marge    .12
interdoc.
-----
feuille  .45
page     .35
saut     .n
finition.n

```

Dans notre DE qui s'appelle EDIT, nous avons remplacé les champs numériques par leurs noms. Cela nous a paru nécessaire, car la plupart des usagers ne sont pas familiarisés avec les champs numériques. Pour l'activer on a qu'à taper */edit. Si on veut avoir une édition cataloguée d'une liste ou interrogation, il suffit d'utiliser la commande LE ou QE après son activation.

5 - La table d'appauvrissement :

Avant la génération des index, nous avons créé une table d'appauvrissement que nous avons appelé TXTAPP4.WHE (voir Annexe 8). Elle a été saisie par l'utilitaire TEXTABLE sous TEXUTIL, accessible en dehors de TEXTO. Pour le charger on se place sur C à partir du DOS et on tape TEXUTIL. La table est

automatiquement créée dans le répertoire WHEELS placée sous le répertoire TEXTO. Outil de grande valeur, elle nous a permis d'uniformiser la typographie et d'inverser le contenu des champs auteur et descripteurs qui n'étaient pas structurés.

6 - Les index :

Selon les instructions du Manuel d'Utilisation TEXTO, "sur micro-ordinateur l'interrogation par index devient vite nécessaire".

Pour notre base de données nous avons créé deux index d'interrogation :

- de titre (ITITRE) et
- de multi-champs (IFUSION) (voir Annexe 9)

Malheureusement nous n'avons pas eu le temps de faire un index titre qui permettrait l'interrogation par n'importe quel mot du titre, ni d'élaborer une liste des mots vides. L'index titre que nous avons créé, donne la possibilité d'interroger uniquement par le premier mot du titre suivi de la troncature.

L'index multi-champs est formé par les champs auteur, descripteurs français et descripteurs anglais. Les champs descripteurs français et descripteurs anglais englobent les auteurs cités. Il nous a donc semblé intéressant de regrouper tous ces champs dans un même index. Nous avons lié cet index à leurs champs respectifs et nous avons "implicité" le champ auteur. Ceci nous a permis :

- 1- D'accélérer la recherche puisque les trois champs peuvent être interrogés simultanément,
- 2- De les interroger directement sans avoir à indiquer le champ numérique.

Puisqu'il s'agit d'un index formé par plusieurs descripteurs composés, on doit utiliser des troncatures à l'interrogation.

Nous avons opté pour une mise à jour immédiate d'index. Pour créer un index on envoie la commande "index". Le système nous demande son nom, qui doit obligatoirement commencer par la lettre "i". Ensuite, il veut savoir si on veut créer un index d'interrogation. Nous avons tapé "y" suivi du numéro de notre table d'appauvrissement (4), pour qu'elle soit liée à notre index.

```

*index
index resultat(s)                               :ifusion
faire un index d'interrogation   v / n ? :y 4
champ(s)                         source(s)    :100a 690e 690f
longueur maxi des articles      ( < 47 ) :40

```

7 - L'interrogation :

7.1 - La syntaxe :

Le système accepte l'utilisation de masques (.), employés pour cacher un caractère dont la valeur est inconnue.

Les opérateurs logiques ("et", "ou", "sauf"), permettent de combiner plusieurs questions sur un même champ, ou bien de faire une interrogation sur plusieurs champs.

Les troncatures (*) gauches, droites et internes sont autorisées.

7.2 - Les procédures :

Les procédures d'interrogation sont de deux types : simple (q) ou composée (ques).

L'interrogation simple symbolisée par la commande "q" permet d'interroger un fichier et d'obtenir à l'instant même le contenu complet des références pertinentes. L'interrogation simple peut être faite à 4 niveaux :

q1 - donne le numéro de références qui répondent à la question;

q2 - donne le numéro de références et le numéro du document;

q3 - donne le numéro de références, le numéro du document et le contenu du champ concerné.

q ou q4 - donne le numéro de références et le contenu intégral du document.

La syntaxe utilisée pour une interrogation composée (ques) est la même que celle utilisée précédemment. Par contre,

sous l'interrogation composée on n'obtient que le nombre de réponses précédées par le signe \$. Chaque question crée un ensemble numéroté que TEXTO mémorise dans un fichier spécial. Ces ensembles peuvent être ensuite combinés entre eux.

Voici un exemple d'interrogation composée effectuée dans notre base de données :

*ques

***** Interrogation Composee *****

?logiciel*
LOGICIEL

\$1 1 reponse(s) pour : logiciel*

?interface*
INTERFACE UTILISATEUR

\$2 13 reponse(s) pour : interface*

?\$1 et \$2

\$3 1 reponse(s) pour : \$1 et \$2

?le

numero(s) :1

s v p , positionnez votre papier

RESULTAT DE LA RECHERCHE

BD : PASCAL
AUTEUR : BURTON J; NEWPORT J; ROBINSON E
TITRE : OPACs and JANET: a simple technique for easy user
.access
SOURCE : Program; ISSN 0033-0337; Coden PRGRDU; GBR; DA. 1989;
.VOL. 23; NO. 3; PP. 257-268; BIBL. 8 ref.
RESUME : Presentation illustree de copies d'ecran du logiciel
.realise par l'Universite Loughborough a partir de
.KERMIT et MENU, permettant d'accéder aux catalogues
.publics en ligne des autres bibliotheques
.d'enseignement superieur par l'intermediaire du
.reseau JANET (Joint Academic Network)
DESCRIPT : Interface utilisateur; Logiciel; Acces information;
.Catalogue automatise; Traitement en ligne; Reseau
.bibliotheque; Reseau telecommunication; Bibliotheque
.universitaire; Royaume Uni; OPAC; JANET; KERMIT; MENU

reference(s) :

8 - Le document de dialogue :

Pour enregistrer un document de dialogue on envoie la commande :

*a /d.....

Le nom d'un Dialogue doit toujours commencer par la lettre D.

Le nombre de Dialogues n'est pas limité. Mais selon la machine, son contenu ne doit pas dépasser la limite de 400 à 2000 caractères. Pour résoudre ce problème il est possible de les enchaîner (voir Annexe 10).

Nous n'avons pas pu finir d'enregistrer les Dialogues. Nous en avons enregistré 5 :

- /doption : demande si l'utilisateur veut faire une recherche ou bien quitter la base. Ce Dialogue enchaîne deux autres : /drech et /dfin.

- /drech : demande si l'utilisateur veut faire une recherche par auteur et descripteurs, une recherche par titre ou bien quitter la base. Ce Dialogue enchaîne trois autres : /dtitre, /dauteur et /dfin.

- /dauteur : explique comment procéder pour effectuer la recherche par auteur ou descripteurs. Ce module permet également d'éditer les résultats de la recherche, de poser une autre question, de changer de module de recherche ou bien de quitter la base.

- /dtitre : explique comment procéder pour effectuer la recherche par titre. Ce module permet de procéder aux mêmes fonctions que le module précédent.

- /dfin : représente une sortie commentée.

Ce mode Dialogue ne permet pas de faire des instructions conditionnelles du type :

si la réponse est...

alors...

sinon...

Dans ces documents nous devrions indiquer les ordres devant être mémorisés pour la mise en place de nos procédures d'interrogation. Ces Dialogues serviront comme moyen d'orienter l'utilisateur qui ne connaît pas le logiciel TEXTO.

CONCLUSION

La base de données fonctionne pour l'instant avec 75 références provenant des fichiers : Ndong2, Possoel, Soual et Opac.

Au terme de ce stage, nous pouvons faire un bilan de tout ce qui pourrait être fait pour compléter notre travail :

- Aucun traitement n'a été fait pour éliminer les doublons. Dans notre cas il ne s'agit pas seulement de repérer la similarité au niveau de l'auteur et du titre, mais aussi de la publication et même de la banque de données.

- Autres banques de données pourront être traitées par notre programme. Il suffit d'élargir le programme et d'opérer quelques modifications;

- Créer un index pour permettre l'interrogation par mots du titre et changer le Dialogue existant en y apportant les modifications nécessaires.

- Utiliser un langage de programmation (ex : Logotel) pour gérer le dialogue avec l'utilisateur.

D'autre part, l'expérience de concevoir et d'installer cette base de données a été enrichissante à tous les niveaux. Nous estimons que ce travail sera utile à ceux pour qui elle a été réalisée.

BIBLIOGRAPHIE

* BANQUES DE DONNEES

Dialog information retrieval service. Palo Alto, CA : Dialog Information Services. n.p.

LISA online user manual. 2nd ed. Oxford : Learned Information, 1987. 291p.

QUESTEL manuel d'utilisation. Paris : Questel. v.2.

* OCLC-MARC

Format de catalogage des livres. 3e éd. Dublin : Online Computer Library Center, 1986. 2v.

* TEXTO

Manuel d'utilisation TEXTO. Lyon : Chemdata, 1990. n.p.

* TURBO PASCAL

KOFFMAN, Elliot B. *PASCAL: problem solving and program design.* 3rd ed. New York : Addison-Wesley, 1989. 778p.

LE BEUX, Pierre. *Introduction à PASCAL avec TURBO PASCAL.* Paris: Sybex, 1990. 684p.

LE BEUX, Pierre, TAVERNIER, Henri. *PASCAL et TURBO PASCAL par la pratique.* Paris : Sybex, 1990. 593p.

Manuel de référence de TURBO PASCAL version 5.0. Velizy : Borland International, 1988. 712p.

VIRGA. *Aide-mémoire de Turbo Pascal,* avec la collab. de Jacques Morleghem. Alleur : Marabout, 1988. 221p.

ANNEXES

ANNEXE 1

EXEMPLE D'UNE NOTICE FRANCIS :

5/13 - (C) CNRS-FRANCIS
NO : 88-617-0696
DO : ECODOC
AU : MATHE (J. C.)
AF : Faculte de droit et des sciences economiques de Limoges et
CEREL (Universi
TI : Politique generale de l'entreprise: analyse et management
strategiques.
SO : Paris: Economica; Coll. Techniques de gestion; FRA; ISBN
2-7178-1331-4; 1987; 245 p., tabl., fig., graph., lexique,
index; bibl. Bibliogr. (28 ref.)
DT : LIVRE; E
LA : FRE
DP : 1987
CH : 617.VI
CC : 617.5
FG : Economie de l'entreprise
EG : Business finance
FD : gestion de l'entreprise; strategie
AB : Outils de gestion strategique de l'entreprise. Le systeme
d'information, la planification, le management. L'auteur
presente les bases de l'analyse concurrentielle et les
outils du diagnostic strategique interne
LO : SC. ECO./GRENOBLE

ANNEXE 2

EXEMPLE D'UNE NOTICE INSPEC :

6/2/1 (Item 1 from file: 4)
03814013 INSPEC Abstract Number: C91017936
Title: Review of cataloging at the Western Library Network
Author(s): Griffin, D.
Author Affiliation: WLN Bibliographic Center, Western Libr. Network,
Olympia, WA, USA
Journal: Resource Sharing and Information Networks vol.6, no.1 p.
17-37
Publication Date: 1990 Country of Publication: USA
CODEN: RSINEO ISSN: 0737-7797
Language: English Document Type: Journal Paper (JP)
Treatment: Applications (A); Practical (P)
Descriptors: bibliographic systems; cataloguing; computer networks;
library automation
Identifiers: cataloging; Western Library Network; WLN; MARC format
standards; database; Network level review; bibliographic standards;
customer service
Class Codes: C7240 (Information analysis and indexing); C7210L (Library
automation); C5620 (Computer networks and techniques)

ANNEXE 3

EXEMPLE D'UNE NOTICE LISA :

5/5/1

0206926 90-6492 Library and Information Science Abstracts (LISA)

TITLE: Automation and organisation of work in university libraries.

TITLE (Translated): Automazione e organizzazione del lavoro nelle biblioteche delle Universita.

AUTHOR(S): Salvi, Paolo

JOURNAL: Bollettino d'Informazioni

SOURCE: 29 (2-3) Apr-Sept 89, 301-309. 10 refs

LANGUAGES: Italian

ABSTRACT: Contribution to a thematic issue on university libraries in Italy and to the special section on their structure. Looks at the impact of automation on the organisation of work in Italian university libraries in terms of: the benefits of integrated systems; the reduction or redeployment of cataloguing staff with the advent of on-line cataloguing; the greater degree of precision and accuracy required when inputting records to standards which will be used by other libraries; the more open methods of work and team work required to sort out problems as they arise; considerations of physical space and ergonomics. Attributes the slow spread of automation in Italian university libraries to the fact that the head librarian is often a member of faculty who though lacking in information science training and spending only 12 hours per week on library matters is reluctant to delegate any power in order to facilitate change. B.T.

DESCRIPTORS: Data processing; Computers; University libraries; Italy

SECTION HEADINGS: COMPUTERS

SECTION HEADING CODES: OqGdD45

ANNEXE 4

EXEMPLE D'UNE NOTICE PASCAL :

1/13 - (C) CNRS
NO : PASCAL-INFODOC 90-0171216
FT : Realisation de l'interface utilisateur d'un catalogue
en ligne
ET : Designing the user-interface of an online public access
catalogue
GT : Die Gestaltung der Benutzeroberflaeche von Online-
Bibliothekskatalogen
AU : BUDNITZ H
AF : Bibliothek Deutschen Bundestages/Bonn 5300/DEU
DT : Periodique; LA
SO : ABI-Technik; ISSN 0720-6763; Coden ABITD7; DEU; DA.
1988; VOL. 8; NO. 4; ABS. ENG; BIBL. 72 ref.
LA : GER
CC : 205A07A; 001A01G
FD : Traitement en ligne; Interface utilisateur; Recherche
documentaire; Catalogue automatise; Allemagne
(republique federale); Etats Unis; Assistance
utilisateur; Conception; OPAC
LO : CNRS-20965

ANNEXE 5

EXEMPLE D'UNE NOTICE TRAITÉE PAR LE PROGRAMME :

035a

LISA

035b

0206926 90-6492

200a

Automazione e organizzazione del lavoro nelle
biblioteche delle Università.

100a

Salvi, Paolo

440a

Bollettino d'Informazioni

29 (2-3) Apr-Sept 89, 301-309. 10 refs

042

Ita

520

Contribution to a thematic issue on university libraries in Italy and to the special section on their structure. Looks at the impact of automation on the organisation of work in Italian university libraries in terms of: the benefits of integrated systems; the reduction or redeployment of cataloguing staff with the advent of on-line cataloguing; the greater degree of precision and accuracy required when inputting records to standards which will be used by other libraries; the more open methods of work and team work required to sort out problems as they arise; considerations of physical space and ergonomics. Attributes the slow spread of automation in Italian university libraries to the fact that the head librarian is often a member of faculty who though lacking in information science training and spending only 12 hours per week on library matters is reluctant to delegate any power in order to facilitate change.

B.T.

690e

Data processing; Computers; University libraries; Italy

//

ANNEXE 6

```
Program EFFACE;
uses dos;

Var
F, FRes : Text;
C : String [8];
C1 : String;
NomF, NomF2 : String [12];

Begin
Write ('Entrez le nom du fichier à traiter : ');
Readln (NomF);
Assign (F, NomF);
Reset (F);
Write ('Nom du fichier résultat : ');
Readln (NomF2);
Assign (FRes, NomF2);
Rewrite (FRes);
While not EOF (F) Do
Begin
Read (F, C);
Readln (F, C1);
Writeln (FRes, C1);
End;
Close (F);
Close (FRes);
End.
```

ANNEXE 7

```
Program BASEDON;
uses crt, dos;

Type
Tab = array [1..4] of string;
Codes = record
Champ : string;
Code : string [4];
End;
Numeros = record
Zone : string ;
Numero : string [4];
End;
TabFranPas = array [1..20] of codes;
TabLisalisa = array [1..27] of numeros;

Const
NombreBanque = 4;

Var
NomF, FRes : string [12];
VarF, VarF2 : text;
Banque_Donnee : Tab;
Trouve : boolean;
T : TabFranPas;
A : TabLisalisa;
NumBanque, Code : integer;
X1 : string;
I, j, n, z : integer;
```

```

Procédure AssignationFichiers;

Var
K : PathStr;

Begin (* début assignation fichier/entrée *)
  ClrScr;
  Repeat
    {$I-}
    Write ('Nom du fichier à traiter: ');
    Readln (NomF);
    Assign (VarF, NomF);
    Reset (VarF);
    {$I+}
    Code := IOResult;
    If Code <> 0 then
      Begin
        Writeln;
        Writeln ('Fichier non trouvé !');
        Writeln;
      End
    Until Code = 0;
End; (* fin assignation du fichier/entrée *)

```

```

Procédure RechercheFichier;

```

```

Var
Y : PathStr;

Begin (* début assignation fichier/sortie *)
  Writeln;
  Write ('Nom du fichier résultat: ');
  Repeat
    Readln (FRes);
    Y := FSearch (FRes, GetEnv ('PATH'));
    If Y = '' then
      Begin
        Assign (VarF2, FRes);
        Rewrite (VarF2);
      End
    Else
      Begin
        Writeln;
        Writeln ('Fichier ', FRes, ' trouvé comme: ', FExpand (Y));
        Writeln;
        Write ('Choisissez un autre nom: ');
      End
    Until Y = '';
End; (* fin assignation fichier/sortie *)

```

```

Procédure AffectBanqueDonnee;

```

```

Begin (* début affectation des banques de données *)

  Banque_Donnee[1] := 'LISA';
  Banque_Donnee[2] := 'INSPEC';
  Banque_Donnee[3] := 'PASCAL';
  Banque_Donnee[4] := 'FRANCIS';

End; (* fin affectation des banques de données *)

```

Procedure AffectChampCodeFranPas;

Begin (* début affectation champ/code FRANCIS et PASCAL *)

```
T[1].Champ := 'NO' ; T[1].Code := '035b';
T[2].Champ := 'LA' ; T[2].Code := '042';
T[3].Champ := 'CC' ; T[3].Code := '093';
T[4].Champ := 'AU' ; T[4].Code := '100a';
T[5].Champ := 'AF' ; T[5].Code := '100u';
T[6].Champ := 'ET' ; T[6].Code := '200a';
T[7].Champ := 'TI' ; T[7].Code := '200a';
T[8].Champ := 'SO' ; T[8].Code := '440a';
T[9].Champ := 'AB' ; T[9].Code := '520';
T[10].Champ := 'FA' ; T[10].Code := '520';
T[11].Champ := 'FD' ; T[11].Code := '690f';
T[12].Champ := 'DP' ;
T[13].Champ := 'LO' ;
T[14].Champ := 'DT' ;
T[15].Champ := 'CH' ;
T[16].Champ := 'FG' ;
T[17].Champ := 'EG' ;
T[18].Champ := 'DO' ;
T[19].Champ := 'FT' ;
T[20].Champ := 'GT' ;
```

End; (* fin affectation champ/code FRANCIS et PASCAL *)

Procedure AffectChampCodeLisInspec;

Begin (* début affectation champ/code LISA et INSPEC *)

```
A[1].Zone := 'Language' ; A[1].Numero := '042';
A[2].Zone := 'LANGUAGES' ; A[2].Numero := '042';
A[3].Zone := 'Author(s)' ; A[3].Numero := '100a';
A[4].Zone := 'AUTHOR(S)' ; A[4].Numero := '100a';
A[5].Zone := 'Author Affiliation' ; A[5].Numero := '100u';
A[6].Zone := 'Title' ; A[6].Numero := '200a';
A[7].Zone := 'TITLE (Translated)' ; A[7].Numero := '200a';
A[8].Zone := 'JOURNAL' ; A[8].Numero := '440a';
A[9].Zone := 'Journal' ; A[9].Numero := '440a';
A[10].Zone := 'Conference Title' ; A[10].Numero := '440a';
A[11].Zone := 'ABSTRACT' ; A[11].Numero := '520';
A[12].Zone := 'Abstract' ; A[12].Numero := '520';
A[13].Zone := 'DESCRIPTORS' ; A[13].Numero := '690e';
A[14].Zone := 'Descriptors' ; A[14].Numero := '690e';
A[15].Zone := 'TITLE:' ;
A[16].Zone := 'SOURCE' ;
A[17].Zone := 'Publication Date';
A[18].Zone := 'SECTION HEADINGS' ;
A[19].Zone := 'SECTION HEADING CODES' ;
A[20].Zone := 'NOTE' ;
A[21].Zone := 'CODEN' ;
A[22].Zone := ' Document Type' ;
A[23].Zone := 'Treatment' ;
A[24].Zone := 'Identifiers' ;
A[25].Zone := 'Class Codes' ;
A[26].Zone := 'Publisher' ;
A[27].Zone := 'Editor(s)' ;
```

End; (* fin affectation champ/code LISA et INSPEC *)

```

Procedure TraitementTexte;

Var
TailleX1 : integer;
X2 : string;

Begin (*début du traitement de texte des fichiers de notices*)

Case NumBanque of

1:Begin (* début du traitement LISA *)
  Writeln (VarF2, '035b');
  Writeln (VarF2, copy( X1 , 1, 17));
  Readln (VarF, X1);
  TailleX1 := length(X1);
  While not EOF(VarF) and (X1 <> '') and (TailleX1 > 3) Do
  Begin
    I:=1;
    While X1[I] = ' ' Do I := I + 1;
    X1 := copy(X1, I, length(X1));
    n:=1;
    While (n<=27) and (pos(A[n].Zone, X1) <> 1) Do n:= n+1;
    If n=16 then
      Begin
        I := 1;
        While X1[I] <> ':' Do I := I + 1;
        I := I + 1;
        While X1[I] = ' ' Do I := I + 1;
        Writeln (VarF2, copy(X1, I, length(X1)));
      End;
    If n<14 then
      Begin (* début du traitement d'un champ à copier *)
        Writeln (VarF2, A[n].Numero);
        I := 1;
        While X1[I] <> ':' Do I := I + 1;
        I := I + 1;
        While X1[I] = ' ' Do I := I + 1;
        If n=2 then
          Writeln (VarF2, copy(X1, I, 3))
        Else
          Writeln (VarF2, copy(X1, I, length(X1)));
        Repeat
          Readln (VarF, X1);
          TailleX1 := length(X1);
          X2 := copy(X1, 1, 5);
          If X2 = '      ' then
            Writeln (VarF2, copy(X1, 6, length(x1)));
          Until (X2<>'      ') or EOF(VarF) or (TailleX1 < 3);
        End (* fin du traitement d'un champ à copier *)
        Else Readln (VarF, X1);
      End; (* end of while *)
    Writeln (VarF2, '//');
  End; (* fin du traitement LISA *)

```

```

2:Begin (* début du traitement INSPEC *)
  Writeln (VarF2, '035b');
  Write (VarF2, copy(X1, 1, 10));
  Writeln (VarF2, copy(X1, 35, 10));
  Readln (VarF, X1);
  TailleX1 := length(X1);
  While not EOF(VarF) and (TailleX1 > 1) and (x1<>'') Do
  Begin
    I := 1;
    While X1[I] = ' ' Do I := I + 1;
    X1 := copy(X1, I, length(X1));
    n := 1;
    While (n<=27) and (pos(A[n].Zone, X1) <> 1) Do n :=n+1;
    If n=17 then
      Begin
        I := 1;
        While X1[I] <> ':' Do I := I + 1;
        I := I + 1;
        While X1[I] = ' ' Do I := I + 1;
        Writeln (VarF2, copy(X1, I, length(X1)));
      End;
    If n<15 then
      Begin (* début du traitement d'un champ à copier *)
        Writeln (VarF2, A[n].Numero);
        I := 1;
        While X1[I] <> ':' Do I := I + 1;
        I := I + 1;
        While X1[I] = ' ' Do I := I + 1;
        If n=1 then
          Writeln (VarF2, copy(X1, I, 3))
        Else
          Writeln (VarF2, copy(X1, I, length(X1)));
        Repeat
          Readln (VarF, X1);
          TailleX1 := length(X1);
          If X1[1] <> ' ' then
            Writeln (VarF2,copy(X1, 1, TailleX1));
          Until (X1[1]=' ') or EOF(VarF) or (TailleX1 < 3);
        End (* fin du traitement d'un champ à copier *)
      Else
        Readln (VarF, X1);
    End; (* end of while *)
    Writeln (VarF2, '//');
  End; (* fin du traitement INSPEC *)

```

```

3:Begin (* début du traitement PASCAL *)
  z := 1;
  While not (X1[z] in ['0'..'9']) do z := z + 1;
  Writeln (VarF2, copy(X1 ,z, length (X1)));
  Readln (VarF, X1);
  j := 1;
  While (j<20) and (pos(T[j].Champ, X1) <> 1) Do j:=j+1;
  While not EOF(VarF) and (X1 <> '') Do
  Begin

```

```

If j<12 then
Begin (* début du traitement d'un champ à copier *)
  Writeln (VarF2, T[j].Code);
  Writeln (VarF2, copy(X1, 8, length (X1)));
  Repeat
    Readln (VarF, X1);
    j := 1;
    While (j<=20)and (pos(T[j].Champ,X1)<>1) Do j:=j+1;
    If j=21 then
      Writeln (VarF2, copy(X1, 8, length (X1)));
      Until(j in [1..20])or EOF(VarF)or (X1='')or (X1=' ');
    End (* fin du traitement d'un champ à copier *)
  Else
    If j in [12..20] then
      Repeat
        Readln (VarF, X1);
        j := 1;
        While (j<=20) and (pos(T[j].Champ,X1)<>1) Do j:=j+1;
        Until (j in [1..20])or EOF(VarF)or (X1='')or (X1=' ');
      End; (* end of while *)
      Writeln (VarF2, '//');
    End; (* fin du traitement PASCAL *)

4:Begin (* début du traitement FRANCIS *)
  Readln (VarF, X1);
  j := 1;
  While (j<=20) and (pos(T[j].Champ, X1) <> 1) Do j:=j+1;
  While not EOF(VarF) and (X1 <>'') Do
  Begin
    If j<12 then
      Begin (* début du traitement d'un champ à copier *)
        Writeln (VarF2, T[j].Code);
        Writeln (VarF2, copy(X1, 8, length (X1)));
        Repeat
          Readln (VarF, X1);
          j := 1;
          While (j<=20)and (pos(T[j].Champ,X1)<>1) Do j:=j+1;
          If j=21 then
            Writeln (VarF2, copy(X1, 7, length(X1)));
            Until(j in [1..20])or EOF(VarF)or (X1='')or (X1=' ');
          End (* fin du traitement d'un champ à copier *)
        Else
          If j in [12..20] then
            Repeat
              Readln (VarF, X1);
              j := 1;
              While (j<=20) and (pos(T[j].Champ, X1)<>1) Do j:=j+1;
              Until (j in [1..20])or EOF(VarF)or (X1='')or (X1=' ');
            End; (* end of while *)
            Writeln (VarF2, '//');
          End; (* fin du traitement FRANCIS *)
        End; (* CASE *)
      End; (* fin du traitement de texte des fichiers de notices *)

```

```
Begin {programme principal}
  AssignationFichiers;
  RechercheFichier;
  Assign (VarF2, FRes);
  Rewrite (VarF2);
  AffectBanqueDonnee;
  AffectChampCodeFranPas;
  AffectChampCodeLisInspec;
  While not EOF (VarF) Do
  Begin
    RechercheBanqueDonnee;
    TraitementTexte;
  End;
  Close (VarF);
  Close (VarF2);
End. {programme principal}
```

ANNEXE 8

TABLE D'APPAUVRISSEMENT TXTAPP4.WHE :

16

141~0041.....

Transcodage en MAJU
Table employee pour retrait des mots vides
codes en caractere

a	A	#
â	A	#
ä	A	#
à	A	#
b	B	#
c	C	#
d	D	#
e	E	#
ë	E	#
è	E	#
é	E	#
è	E	#
f	F	#
g	G	#
h	H	#
i	I	#
ï	I	#
î	I	#
j	J	#
k	K	#
l	L	#
m	M	#
n	N	#
o	O	#
ö	O	#
ò	O	#
p	P	#
q	Q	#
r	R	#
s	S	#
t	T	#
u	U	#
ù	U	#
ü	U	#
û	U	#
v	V	#
w	W	#
x	X	#
y	Y	#
z	Z	#
;	{	#

ANNEXE 9**ECHANTILLON DE L'INDEX D'INTERROGATION MULTI-CHAMPS IFUSION :**

1 CAIN, J.
.00074.

1 CALIFORNIE
.00008.

1 CANADA
.00054.

1 CARD S
.00008.

1 CARPENTER, G.L.
.00067.

1 CATALOGUE
.00010.

12 CATALOGUE AUTOMATISE
.00006,00007,00008,00009,00011,00012,00013,00014,00015,00016
.00017,00018.

1 CATALOGUE COLLECTIF
.00015.

25 CATALOGUING
.00019,00022,00024,00027,00032,00035,00036,00038,00039,00040
.00041,00042,00044,00049,00050,00057,00058,00062,00063,00064
.00066,00067,00068,00069,00074.

5 CD-ROMS
.00043,00045,00055,00061,00072.

1 COMPLETE COMPUTER PROGRAMS
.00035.

2 COMPUTER AIDED INSTRUCTION
.00026,00028.

2 COMPUTER COMMUNICATIONS SOFTWARE
.00038,00070.

1 COMPUTER INSTALLATION
.00075.

1 COMPUTER INTERFACES
.00067.

3 COMPUTER NETWORKS
.00019,00030,00073.

1 COMPUTERS
.00051.

1 CONCEPTION
.00006.

1 CONTRAT
.00005.

1 CROUCHER C
.00018.

1 DARLING, K.
.00058.

1 DATA COMMUNICATION SYSTEMS
.00067.

1 DATA PROCESSING
 .00051.
1 DATA STRUCTURES
 .00035.
2 DATABASE MANAGEMENT SYSTEMS
 .00034,00050.
1 DAVIS, C.C.
 .00039.
1 DAVOUS (P.)
 .00003.
2 DECISION
 .00003,00004.
1 DECKER, L.
 .00070.
1 DIAGNOSTIC
 .00003.

ANNEXE 10

DOCUMENT DE DIALOGUE /DOPTION :

```
nom      .doption
dialogue.dial nnn
        ./f biblio
        ./pbiblio
        .*cls
        .ques
        .&
        .&
        .&
        .&
        .&
        .&
        .&
        .&
        .&Si vous voulez faire une recherche, tapez : /drech
        .&
        .&Si vous voulez quitter la base, faites <RC>
        .&
        .&Indiquez votre choix :
        .&
        .?
        ./dfin
```

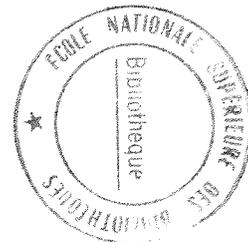
B I E N V E N U (E)

BD BIBLIOGRAPHIQUE

DOCUMENT DE DIALOGUE /DFIN :

```
nom      .dfin
dialogue.dial nnn
        .*cls
        .&
        .&
        .&
        .&
        .&
        .fin
```

AU REVOIR ET A BIENTOT !



BIBLIOTHEQUE DE L'ENSSIB



8016350