



## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Optimisation d'un plan de vente

Bienfait, Bernadette

*Award date:*  
1988

*Awarding institution:*  
Universite de Namur

[Link to publication](#)

#### **General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### **Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre-Dame de la Paix  
Institut d'informatique

OPTIMISATION D'UN PLAN DE VENTE

Bernadette Bienfait

Promoteur: Monsieur J-P. LECLERCQ

Mémoire présenté en vue  
de l'obtention du grade  
de licenciée et maître  
en Informatique par  
Bernadette Bienfait.

année académique 87-88

Arrivée au terme de ce travail, je tiens à remercier Monsieur J-P. LECLERCQ pour l'intérêt avec lequel il a dirigé ce mémoire. Je remercie également le personnel du service informatique des 3 SUISSES, et plus particulièrement Monsieur HANOTIAU et Madame CASTELLAIN pour l'aide qu'ils m'ont apportée dans l'élaboration de ce travail.

## RESUME

Ce mémoire a été élaboré lors d'un stage effectué dans l'entreprise "3 Suisses" à Orcq en Belgique.

Il a pour but d'optimiser le plan de vente de l'entreprise par la méthode du Simplexe. Il montre comment arriver à une formulation correcte et linéaire du problème et les différentes étapes pour obtenir un programme qui résoudra ce problème.

## ABSTRACT

This memoir has been elaborated during a period of probation in the enterprise "3 Suisses" at Orcq in Belgium.

Its purpose is to optimise the enterprise's sale plan with the simplex methode. It shows how to be leaded to a correct and linear formulation of this problem, and also the different steps to arrive to a program which will resolve this problem.

## TABLE DES MATIERES

AVANT-PROPOS

INTRODUCTION

I.1. PRESENTATION DE L'ENTREPRISE 3 SUISSES

I.1.1. Historique général

I.1.2. Le groupe 3 Suisses International

I.1.3. Organigramme de la société 3 Suisses Belgique

I.2. SITUATION ACTUELLE

I.2.1. Le fichier clients 3 Suisses

I.2.2. Le programme de simulation du PVC

I.2.3. Exemple de plan de campagne

I.2.4. Application à réaliser

CHAPITRE 1. ADAPTATION DU PROBLEME A LA PROGRAMMATION  
LINEAIRE

1.1. MODELISATION DU PROBLEME

1.2. LINEARISATION DES DONNEES ET AUTRES PROBLEMES

1.3. TAILLE DU PROBLEME

1.4. CHOIX D'IMPLEMENTATION

CHAPITRE 2. LES OUTILS

2.1. OUTILS MATERIELS

2.2. OUTILS LOGICIELS

2.2.1 Ideal

CHAPITRE 3. RESULTATS ET MANUEL D'UTILISATION

3.1. LES RESULTATS

3.2. MANUEL D'UTILISATION

CONCLUSION

BIBLIOGRAPHIE

A1.2. LA DECOMPOSITION LU

A1.3. LA REGRESSION LINEAIRE

ANNEXE 2

A2.1. DESCRIPTION DE LA BASE DE DONNEES

A2.2. LES PANELS

A2.3. LES PROGRAMMES

## AVANT-PROPOS

Ce mémoire porte sur l'activité "Vente Par Correspondance" de l'entreprise 3 SUISSSES BELGIQUE. Cette activité recouvre l'envoi de documents tels que les catalogues, la prospection, la publicité...

Le problème rencontré par l'entreprise au niveau de cette activité est de déterminer combien et à qui envoyer ces documents de manière à maximiser la rentabilité de ces envois. Ceci implique l'existence d'un "Plan De Vente Par Correspondance" (PVC) pour résoudre ce problème.

Actuellement, le PVC est réalisé par le service Marketing à l'aide de calculs statistiques basés sur les historiques des résultats antérieurs. L'expérience professionnelle des personnes en place est aussi indispensable au bon fonctionnement du PVC.

Le but de ce mémoire est d'appliquer la Recherche Opérationnelle et en particulier la Méthode du Simplexe à ce problème.

Nous procéderons de la façon suivante :

Après une présentation de l'entreprise 3 SUISSSES, nous décrirons les outils existants actuellement dans l'entreprise pour résoudre le problème du PVC. Nous donnerons ensuite un exemple concret pour une meilleure compréhension. Cet exemple nous sera utile tout au long de notre travail.

Dans un premier chapitre, nous présenterons la modélisation du PVC que nous avons réalisée en vue de le rendre compatible avec la Méthode du Simplexe. Nous parlerons également des problèmes que nous avons rencontrés pour parvenir à cette modélisation, problèmes Mathématiques mais aussi problèmes liés à la difficulté de faire accepter une nouvelle méthode de résolution du PVC. Enfin, nous donnerons une estimation de la taille réelle du problème à résoudre et les difficultés qui peuvent apparaître à ce niveau.

Dans le chapitre 2, nous présenterons les outils que nous avons utilisés lors de l'implémentation du problème d'optimisation:

- outils logiciels
- outils matériels

Nous y développerons plus particulièrement le langage que nous avons utilisé pour programmer notre problème. il s'agit du langage IDEAL dans un environnement DATACOM.

Dans le chapitre 3, nous donnerons quelques résultats relatifs à notre programme. Nous suivrons menu après menu le chemin qui a mené à ces résultats. Ce chapitre pourra donc également servir de manuel d'utilisateur du programme.

Enfin dans une première annexe, nous trouverons quelques rappels théoriques relatifs aux différentes notions théoriques que nous avons utilisées lors de l'élaboration de ce travail:

- la méthode du simplexe
- la régression linéaire
- le calcul de l'inverse d'une matrice par la méthode de décomposition LU.

Et dans une deuxième annexe, nous trouverons

- la description de la Base De Données utilisée
- les programmes
- les écrans

## INTRODUCTION

Dans cette introduction, nous allons commencer par présenter l'entreprise 3 Suisses. Ensuite, nous décrirons les outils dont se sert le service Marketing pour résoudre le problème du PVC.

### I.1. PRESENTATION DE L'ENTREPRISE 3 SUISSES

#### I.1.1. HISTORIQUE GENERAL

En 1929, Xavier Toulemonde, l'un des héritiers d'une lignée d'industriels de la laine dans le nord de la France, gérant de la filature Destombe implantée en 1820 à Croix près de Roubaix, réussit à donner une nouvelle impulsion au marché du textile, alors en pleine crise, en ayant l'idée de vendre directement aux mères de familles la laine qu'il fabriquait. Il fit donc éditer un catalogue contenant ses échantillons de laine à tricoter et correspondit avec ses clientes via la poste à l'instar de la Redoute.

La société ainsi créée prit le nom d'un quartier voisin "Les 3 Suisses" en 1931.

Installée à proximité de la Belgique, la firme 3 Suisses s'y implanta dès mars 1933. Cinq ans plus tard, elle s'installe aux Pays-Bas.

Après la guerre, qui ralentit les activités de l'entreprise, celle-ci eut recours à de nouveaux procédés visant à un élargissement du secteur de vente: création de magasins, dont le premier fut ouvert à Anvers vers les années 1948-1950, publication d'un catalogue présentant désormais une gamme d'articles de textile autre que le fil à tricoter.

L'évolution et l'expansion de la firme ne cessa alors de continuer.

en 1963, L'informatique commence à être utilisée par la société.

En 1967, apparaît la prise de commandes par téléphone. L'année suivante s'ouvre le centième magasin belge à Zelzate. 1970 voit le lancement, dans toute la Belgique, du catalogue général français de 400 pages en édition bilingue. Trois ans plus tard, ce catalogue sera édité sur le marché hollandais.

En 1972, apparaît en Belgique la carte de crédit "3 Suisses Partner" gérée par Eurocard et autorisant des achats jusqu' à concurrence de 40 000 francs.

En 1973, chaque magasin reçoit l'autorisation d' effectuer et de réceptionner des commandes de vente par correspondance ("service élargi").

Cependant, l'incendie de l'usine mère de Roubaix contraignit "3 suisses Internationnal" à signer, en 1974, un accord de coopération avec "Otto Versand", deuxième allemand de la vente par correspondance (VPC). Le groupe financier détient au sein de la firme plus de 50 % des actions.

Néanmoins, le groupe n'a pas cessé de progresser. Actuellement, l'informatique est devenue une clef d'expansion et de souplesse d'une industrie qui, faut-il le dire, dépend constamment du changement des goûts et des modes de nos sociétés.

### I.1.2. LE GROUPE 3 SUISSES INTERNATIONAL

D'origine française, le groupe "3 Suisses International" est implanté dans cinq pays d'Europe (France, Belgique, Pays-Bas, Allemagne, Autriche) et s'organise comme suit:

#### Pour les achats

Une filature implantée à Dottignies produit le fil à tricoter et les chaussettes; pour les autres articles, une centrale d'achat est implantée à Roubaix.

#### Pour la vente

Dans un catalogue international, chaque centre national extrait des articles d'habillement, de lingerie, d'ameublement et d'équipement divers pour la maison afin de le proposer à sa clientèle. Le catalogue s'enrichit parfois d'articles spécialement destinés au marché national. Ces articles sont gérés différemment.

Une collection de fils à tricoter, commune à tous les pays, est mise en vente par correspondance en France et en Allemagne ou par un réseau de magasins en Belgique, aux Pays-Bas et en Autriche.

### I.1.3. ORGANIGRAMME DE LA SOCIETE 3 SUISSES BELGIQUE

#### DIRECTION D'EXPLOITATION

La direction d'exploitation est composée des services suivants:

#### service des expéditions

Il organise l'envoi des colis aux particuliers par la poste, par chemin de fer ou par camions. Il prévoit éventuellement d'autres solutions pour les expéditions de marchandises.

#### études informatiques et organe de décision

Ce service définit les diverses orientations à prendre à court et moyen terme.

#### service du personnel

Il s'occupe de la rémunération du personnel de la centrale et des magasins d'usine.

#### gestion stock central

Il s'agit du secteur s'occupant des achats, ventes et mises en rebut du stock national situé à Avelgem. Il dispose d'un ordinateur M.A.I. 710.

#### DIRECTION DE GESTION ET DE FINANCEMENT

Cette direction est composée des services suivants:

##### comptabilité

Ce secteur s'occupe des problèmes de comptabilité tant de la centrale que des magasins d'usine.

##### contrôle

Il s'agit du service vérifiant la gestion des magasins d'usine et la formation du personnel de magasin.

#### DIRECTION VENTE PAR CORRESPONDANCE

La VPC regroupe plus de 60 % du chiffre d'affaire des 3 Suisses Belgique.

Quelques 2 millions de colis sont vendus soit par la VPC pure, c'est à dire que les articles sont envoyés aux particuliers sur simple commande par la poste ou par téléphone (environ 2/3 des opérations), soit par la VPC magasin: dans ce cas, l'article est commandé et retiré par l'intermédiaire d'un magasin, ce qui représente le 1/3 restant des opérations de commande.

Quelques 30 000 articles à prédominance textile font l'objet de deux gros catalogues (printemps/été, automne/hiver) imprimés chacun à 700 000 exemplaires ainsi que de plus petits catalogues bimestriels intitulés "reflets 3 Suisses".

Tout achat est encodé sur un IBM 4381, puis commandé dans le stock soit à Roubaix qui groupe 78 % des commandes, soit à Avelgem à concurrence de 12 %, soit à d'autres endroits pour les 10 derniers %. Deux jours après, les articles commandés arrivent à Orcq où ils sont emballés et livrés avec la facture par la poste, le chemin de fer ou par le propre transport routier des 3 Suisses.

Le service VPC utilise un ordinateur IBM 4381 sous DOS/VSE et CMS, 16 Mbytes de mémoire réelle découpée en 12 partitions.

#### prêt à porter

Ce service gère 22 magasins qui se situent uniquement dans de grandes villes et ne vendent que des articles pour dames tels que sous-vêtements, robes, jupes, pantalons, pulls et autres accessoires féminins.

Créé en 1977, ce département, le plus récent des 3 Suisses, est parvenu en quelques années à atteindre un chiffre d'affaire représentant près de 10 % du chiffre d'affaire global, et ce grâce à une information poussée.

#### discount

Ce service, qui en 1987 regroupe 7 "discount", s'occupe de solder les fins de série ou les articles légèrement endommagés. Ces magasins, de grandes surfaces, se situent à Charleroi, Gent, Gilly, Wilrijk, Orcq, Blaton et Verviers.

Ces magasins sont gérés par à l'aide d'un ordinateur IBM 4381.

#### DIRECTION FIL ET OUVRAGE

Ce secteur s'occupe de la vente en magasin de fils à tricoter et d'ouvrages pour dames, secteur d'origine des 3 Suisses.

#### achat fil et ouvrage

Ce service, équipé d'un ordinateur M.A.I. 9520, centralise les commandes des magasins "fil et ouvrage", en assure la gestion selon la disponibilité des articles, les contraintes des transports ou autres aléas, passe les commandes aux fournisseurs et édite les factures et les documents de transport.

#### réseau fil et ouvrage

Ce service comprend le secrétariat de contact qui établit les liaisons entre les magasins de la chaîne "fil et ouvrage", un service d'animation et un secrétariat de développement des magasins.

## I.2. LA SITUATION ACTUELLE DANS L'ENTREPRISE

### 3 SUISSES

Dans cette section, nous allons présenter les outils actuellement utilisés par le service Marketing pour résoudre le PVC.

Nous donnerons également un exemple qui nous aidera à mieux comprendre la suite de notre travail.

Enfin nous exposerons brièvement l'application qu'il nous a été demandé d'effectuer.

Avant de décrire la résolution actuelle du PVC, définissons une notion qui interviendra tout au long de ce mémoire:

Les 3 SUISSES utilisent le mot ACTION pour parler de l'évènement qui correspond à l'envoi d'un document à la clientèle. Par exemple, on parlera de "l'action catalogue printemps-été" pour identifier les envois de ces catalogues à la clientèle.

Actuellement, le PVC est essentiellement basé sur un PROGRAMME DE SIMULATION écrit en Basic et les données qui lui sont nécessaires lui sont fournies par le FICHER-CLIENTS.

Il est utile pour la suite de ce mémoire de décrire ces deux outils.

### I.2.1 LE FICHER-CLIENTS 3 SUISSES

Ce fichier (DS 400) est composé de deux parties distinctes:

- le fichier prospection
- le fichier actif

#### Le fichier prospection

Il contient les références de personnes qui ne sont pas ou plus considérées comme des clientes actives des 3 SUISSES. C'est à elles que sont destinés les envois de documents du type "Prospection" dont le but est de les faire entrer dans le fichier actif.

Il contient actuellement environ 3 000 000 d'adresses. (1)

#### Le fichier actif

Il contient les références de clientes qui ont passé une commande sur les 5 derniers semestres.

Soit 1 000 000 d'adresses de clientes dont 700 000 ont commandé dans l'année et réparties de la façon suivante:

- 430 000 néerlandophones
- 480 000 francophones
- 22 000 luxembourgeoises. (1)

Ces clientes reçoivent les envois tels que les catalogues printemps-été, les catalogues automne-hiver, les catalogues soleil, les relances ...

(1) chiffres au 04.09.87

### Structure du fichier-clients

Le fichier-clients 3 SUISSES est défini de la façon suivante:  
Les clientes y sont classées par couches, chaque couche étant ordonnée par acquis croissant. L'acquis étant un nombre calculé pour chaque cliente de la façon suivante:

$$\begin{aligned} \text{ACQUIS DU SEMESTRE} &= \text{CA (SEMESTRE - 1)} * (1/100) + \\ &\quad \text{CA (SEMESTRE - 2)} * (1/(100 * \sqrt{2})) + \\ &\quad \text{CA (SEMESTRE - 3)} * (1/(100 * (\sqrt{2})^2)) + \\ &\quad \text{CA (SEMESTRE - 4)} * (1/(100 * (\sqrt{2})^3)) + \\ &\quad \dots \end{aligned}$$

où CA est le Chiffre d'Affaire de la cliente considérée.  
L'acquis nous fournit donc un classement des clientes d'une même couche. Un acquis élevé correspond à une bonne cliente de la couche considérée, de même un faible acquis correspond à une moins bonne cliente de la couche.

11 couches de clientes sont définies dans le fichier-clients :

CN1 (clientes nouvelles), CN2, CN3, CN4, CN5, AB2 (abonnées), AB3, AB4, AB5, ELM (éliminées), PRO (prospects).

L'ordre que nous avons indiqué ici ne représente qu'un ordre logique de classement des clientes, le classement physique des clientes dans le fichier DS 400 étant organisé autrement. Ce classement n'est d'aucun intérêt pour le problème qui nous préoccupe.

On peut remarquer que nous n'avons décrit ici que les informations élémentaires dont nous aurons besoin ultérieurement concernant les clientes. Il peut cependant être intéressant de donner une description plus générale de ce que contient le fichier clientes des 3 Suisses.

La figure 1 ci-dessous donne une idée des informations qui sont disponibles sur une cliente dans le fichier. Tout d'abord, une cliente est identifiée par un numéro, un nom, une adresse, un code langue et par les tops clientes. On peut définir ces tops de la façon suivante: A chaque cliente correspond un ensemble de 256 tops. Chaque top correspond à une action et est positionné à 1 si la cliente a reçu un envoi pour l'action correspondante.

Après les diverses informations sur les habitudes de commandes de la cliente, on trouve un récapitulatif des 7 derniers semestres de la cliente. Pour chaque semestre, on dispose d'un relevé en nombre, en valeur et par action des commandes du secteur général (le secteur général correspond à toutes les actions 3 Suisses), ainsi que des commandes VM (Il s'agit d'un petit catalogue Vitrine Magique parallèle aux actions 3 Suisses mais ne mentionnant pas le nom 3 Suisses). Suivent ensuite les informations concernant le déroulement de ces commandes, les avantages dont bénéficie la cliente, ...

NUMERO CLIENT NOM ET ADRESSE COMPLETE CODE LANGUE TOPS CLIENTS (PSA, RFP, ...)								
COUCHE PAR SECTEUR D'ACTIVITE ACQUIS PAR SECTEUR D'ACTIVITE								
HISTORIQUE PAR SECTEUR (1ère et dernière commandes)								
FREQUENTATION CANAUX (derniers magasins fréquentés) 50 DERNIERES COMMANDES (date, valeur, action, ...)								
DEROULEMENT DES 3 DERNIERES COMMANDES (date, canaux, lignes, factures, livrés directs, livrés après retard, annulés, ...)								
SM0	SM1	SM2	SM3	SM4	SM5	SM6	SM7	HIS
COMMANDES SG		- NOMBRE						
		- VALEUR						
		- ACTIONS						
ACQUIS								
COMMANDES VM								
COMMANDES PAR CANAUX				- VPM				
				- VPT				
				- CREDIT				
LIVRAISON		- DOMICILE						
		- MAGASIN						
RETOURS		- NOMBRE						
		- VALEUR						
RISTOURNE DE RELANCE								
BON D'ACHAT								
TOPS PUBLICITAIRES ET DE SUIVIS								
VALORISATION PUBLICITAIRE SG								
RAYONS FREQUENTES				- PRODUITS (25)				
				- CIBLAGE (40)				

Figure 1. Contenu du fichier.

La figure 2 nous montre une classification des clientes selon leurs attitudes de commandes.

## COUCHE A L'ECRAN

### LISTE DES INFORMATIONS "CLIENTE" DISPONIBLES :

1. TBC : TRÈS BONNES CLIENTES } ONT DROIT AU CATALOGUE  
BC+ : BONNES CLIENTES } EN 1ER JET  
  
CM : CLIENTES MOYENNES : FICHER ACTIF HORS 1ER JET CAT.  
AC : ANCIENNES CLIENTES  
PRO : ADRESSES DE PROSPECTION  
NC : NOUVELLES CLIENTES  
RET+ : CLIENTES ABUSIVES POUR LES RETOURS
2. CRP : CLIENTES INSCRITES CREDIT PARTNER
3. EXC : CLIENTES À RISQUE ÉLEVÉ DE RETOURS (=EXCESSIVES)
4. VPA5% : CLIENTES VPA1, 5% DE RISTOURNE  
VPA7% : CLIENTES VPA3, 7% DE RISTOURNE  
VPA9% : CLIENTES VPA3, 9% DE RISTOURNE
5. RFP : CLIENTES AYANT DEMANDÉ À ÊTRE RAYÉES DES LISTES D'ENVOI DE PUBLICITÉ.

figure 2. liste des informations clientes disponibles

Enfin, à partir des informations disponibles dans les enregistrements clientes, tous les croisements et tris peuvent être utilisés à des fins d'édition de statistiques. La figure 3 ci-dessous montre quelques statistiques existantes. Certaines nous ont été utiles pour obtenir les données que nous avons du traiter.

ADSTAR01 - ANALYSE FICHER/COUCHES

- \* CA ET COMMANDES
- \* SUPERS PAR CANAUX DE CDE OU LIV.
- \* CA ET COMMANDES/CANAUX
- \* RETOURS, B.A., RISTOURNES, PUBLICITÉ
- \* TOPS D'ENVOIS ET DE SUIVIS
- \* SUPERS PAR ACTIONS
- \* SUPERS PAR RAYONS
- \* CA ET COMMANDES/RAYONS

ADSTAR02 - VENTILATION DES COMMANDES ET C.A. PAR COUCHES ET TOPS (K3)

ADSTAR05 - INCIDENTS DE SERVICE (SUSPENS, ANNULÉS,....) VENTILÉS PAR ACTIONS ET NOMBRE DE LIGNES DE COMMANDE.

ADSTAR06 - TAUX DE SUPERS, C.A. ET COMMANDES DES AB2 ET AB3 PAR PROFILS (6 SEMESTRES)

ADSTAR21 - VENTILATION PAR COUCHES ET CANAUX DE CDE/LIV.

- \* CA COMMANDÉ
- \* CA LIVRÉ
- \* CA RETOURNÉ
- \* BA ET RISTOURNES
- \* CA NET
- \* PUBLICITÉ
- \* FRAIS D'EXÉCUTION ET DE RÉSEAU

Voyons maintenant comment est déterminée la couche à laquelle appartient une cliente.

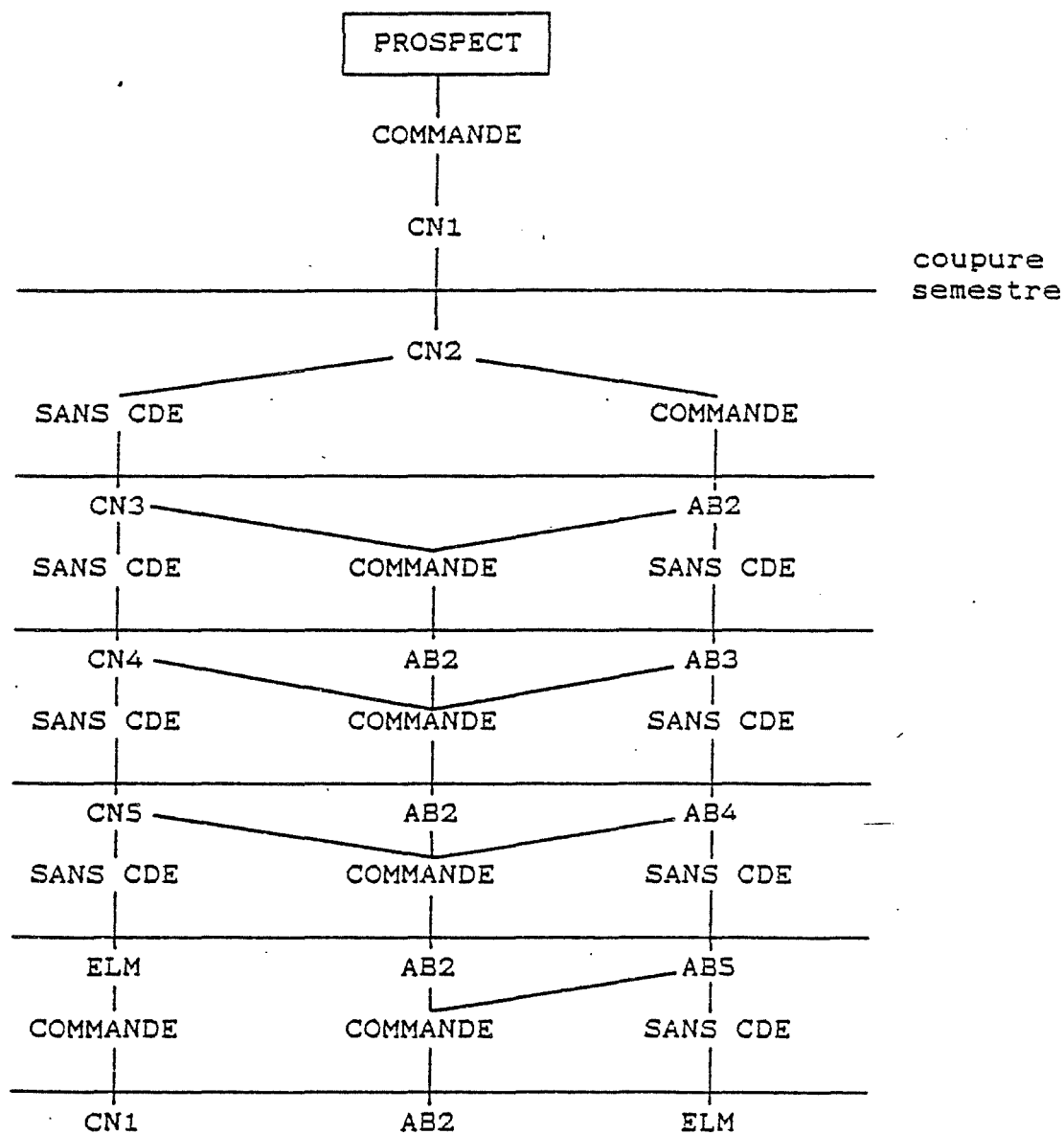


figure 4. Détermination de la couche

Supposons que nous disposions d'une adresse du fichier-clients. Pour déterminer à quelle couche appartient la cliente correspondant à cette adresse, il nous suffit de suivre l'algorithme donné par la figure 4 ci-dessus.

Expliquons le début de cet algorithme, la suite en découle automatiquement:

Toute cliente du fichier prospection qui commande devient CN1.

Au début du semestre suivant, elle devient automatiquement CN2.

Si elle commande au cours de ce semestre, elle devient AB2 sinon elle devient CN3. Une cliente CN3 qui commande au cours du semestre devient AB2, sinon elle devient CN4... La stratégie est la même pour les clientes des autres couches. Remarquons qu'une AB2 qui commande reste dans cette couche de clientes, sinon elle est déclassée en AB3. En fait, toute cliente du fichier actif qui commande devient une AB2 pour le semestre suivant. Cette couche contient donc les meilleures et les plus fidèles clientes de l'entreprise.

La figure 5 propose une représentation du fichier-clients actuel. Il s'agit d'une représentation logique du fichier clientes. L'organisation physique du fichier étant différente et ne nous intéresse pas ici. Cette figure peut nous aider à mieux visualiser les couches de clientes que nous aurons besoin ultérieurement.

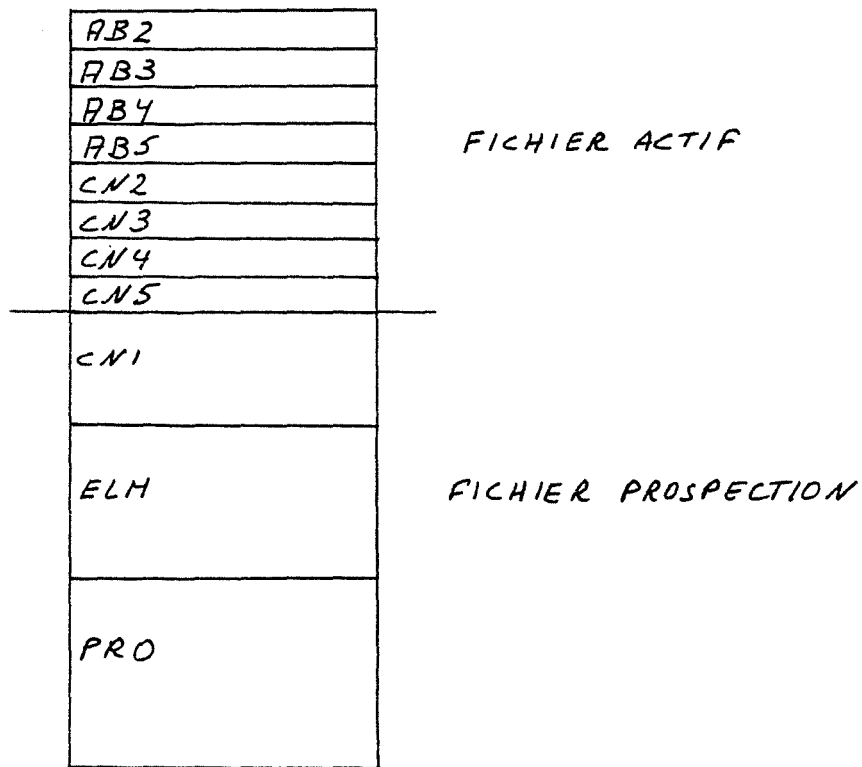


Figure 5. Le fichier-clients 3 SUISSES

### I.2.2 LE PROGRAMME DE SIMULATION DU PVC

La figure 6 ci-dessous propose le schéma général de ce programme qu' on peut expliquer de la façon suivante:

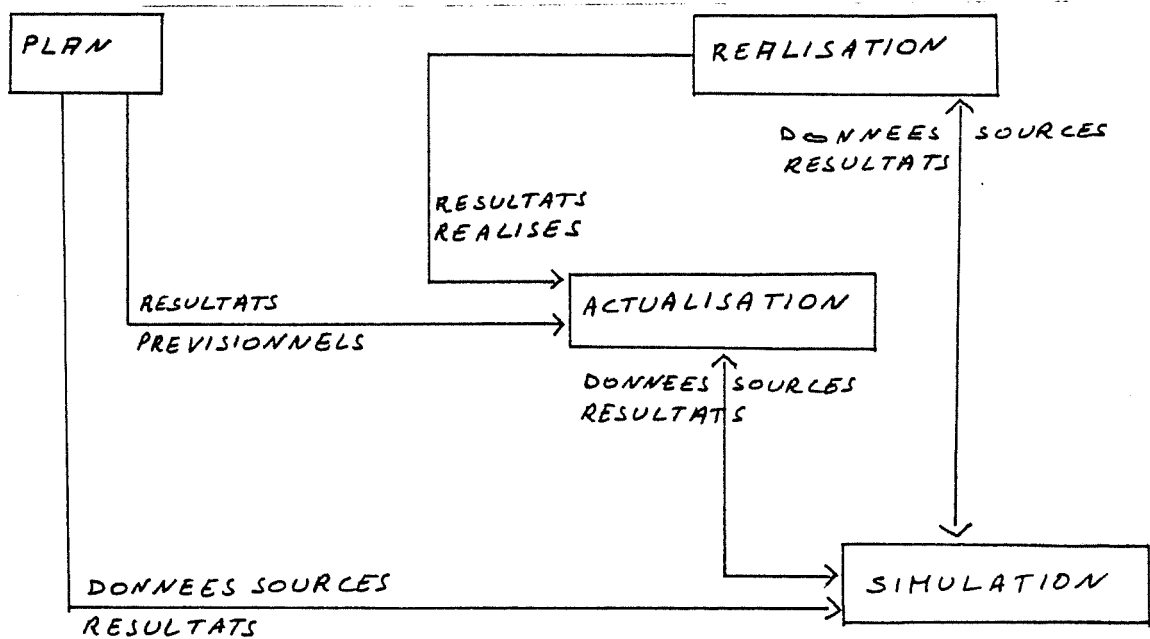


figure 6. programme de simulation du PVC

L'objectif de ce programme est d'obtenir le Chiffre d'Affaire Commandé Toute Taxe Comprise et Facturé Net Hors Taxe par action et au cumul, par semaine ou par mois et par exercice, aux différents types de niveaux:

- PLAN
- REALISATION
- ACTUALISATION
- SIMULATION

A l'aide des statistiques disponibles sur le fichier clientes et d'autres méthodes, mathématiques ou non, le service Marketing calcule les résultats prévisionnels du PLAN, il s'agit essentiellement du Chiffre d'Affaire prévu pour l'exercice courant, le nombre d'envois à effectuer par action et à qui envoyer ces actions. Ces résultats sont fournis à l'Actualisation et à la SIMULATION qui reçoit encore diverses données qui lui seront utiles.

La REALISATION contient les résultats effectivement réalisés au moment considéré, qu'elle fournit à la REALISATION et à la SIMULATION . On peut alors être confronté à une situation comme décrite à la figure 7.

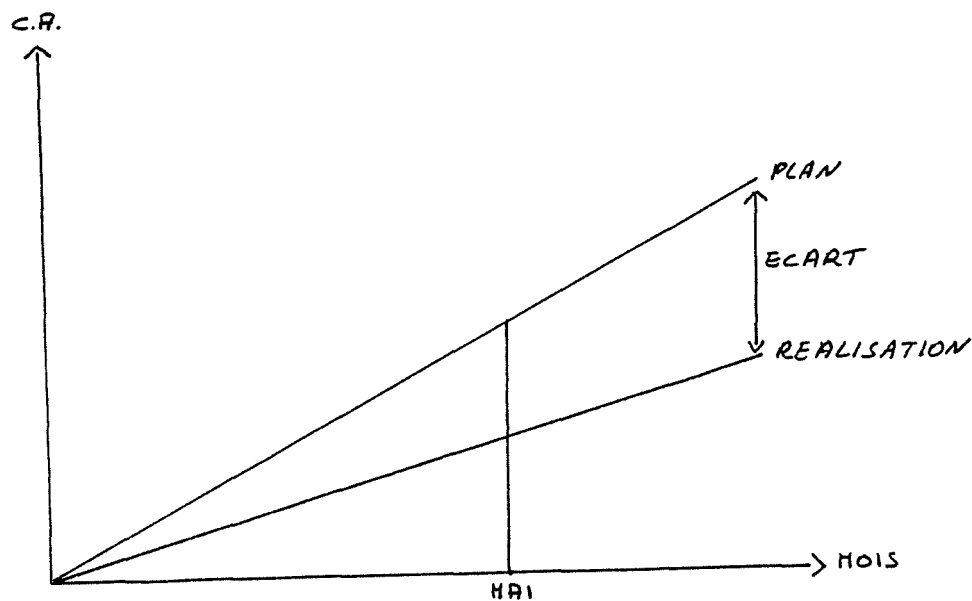


figure 7. Différence entre le PLAN et la REALISATION

Supposons qu'on ait constaté les résultats jusqu'en mai. Le programme de SIMULATION permet de faire une extrapolation de la situation pour le reste de l'exercice et ainsi de prévoir l'écart qui existera entre le PLAN et la REALISATION. Le PLAN peut alors être ACTUALISE en vue de réduire cet écart et d'arriver en fin d'exercice à l'objectif fixé.

La répartition des envois relatifs aux différentes actions de l'exercice est alors calculée en tenant compte des résultats fournis par le programme, et aussi sur base d'historiques des exercices antérieurs. La SIMULATION est faite quotidiennement et permet d'adapter la charge à la capacité productive.

Remarquons que l'expérience professionnelle des personnes responsables du PVC joue un grand rôle dans ces calculs.

### 1.2.3. EXEMPLE DE PLAN DE CAMPAGNE

Pour mieux comprendre le problème du PVC, il nous semble intéressant de décrire un exemple typique de Plan de Campagne auquel est quotidiennement confronté le service de Marketing. Il s'agit d'une partie du PVC correspondant à l'action Catalogue Printemps-Eté et les actions qui lui sont associées.

Pour cela il nous faut d'abord définir 2 nouvelles notions.

Une cliente est dite SUPER si elle a répondu à l'envoi qui lui a été adressé, qu'elle ait commandé ou non. Dans le cas contraire, elle est dite MUETTE. Remarquons ici qu'une cliente peut répondre à un envoi sans commander, il suffit pour cela qu'elle participe au jeu proposé ou qu'elle demande à recevoir le cadeau auquel elle a droit.

La figure 8 ci-dessous décrit la stratégie développée lors du Plan de Campagne de l'action Catalogue Printemps-Eté.

Le problème rencontré étant alors le nombre d'envois à effectuer pour chaque action considérée ici.

Nous reprendrons cet exemple dans la suite pour une meilleure compréhension de certains concepts et de plus, il nous servira lors de nos tests sur machine.

Expliquons le schéma de la figure 8: Le catalogue Printemps-Eté est envoyé à un nombre fixé de clientes. Peu après, ces clientes reçoivent une première Relance. A partir de cet envoi, la distinction est faite entre les clientes muettes et les clientes super. Les clientes muettes reçoivent une deuxième Relance avec un Bon d'Achat, quant aux clientes super, elles reçoivent un Message Choix. Ensuite, toutes les clientes reçoivent la Relance Mars avec 10 % de remise.

Les clientes super reçoivent un Rappel de la Relance Mars, puis, toutes les clientes reçoivent le Catalogue Soleil. Enfin, les clientes muettes reçoivent une Relance avec 20 % de remise et les clientes super reçoivent une action 20 % suivie d'une relance 20 %.

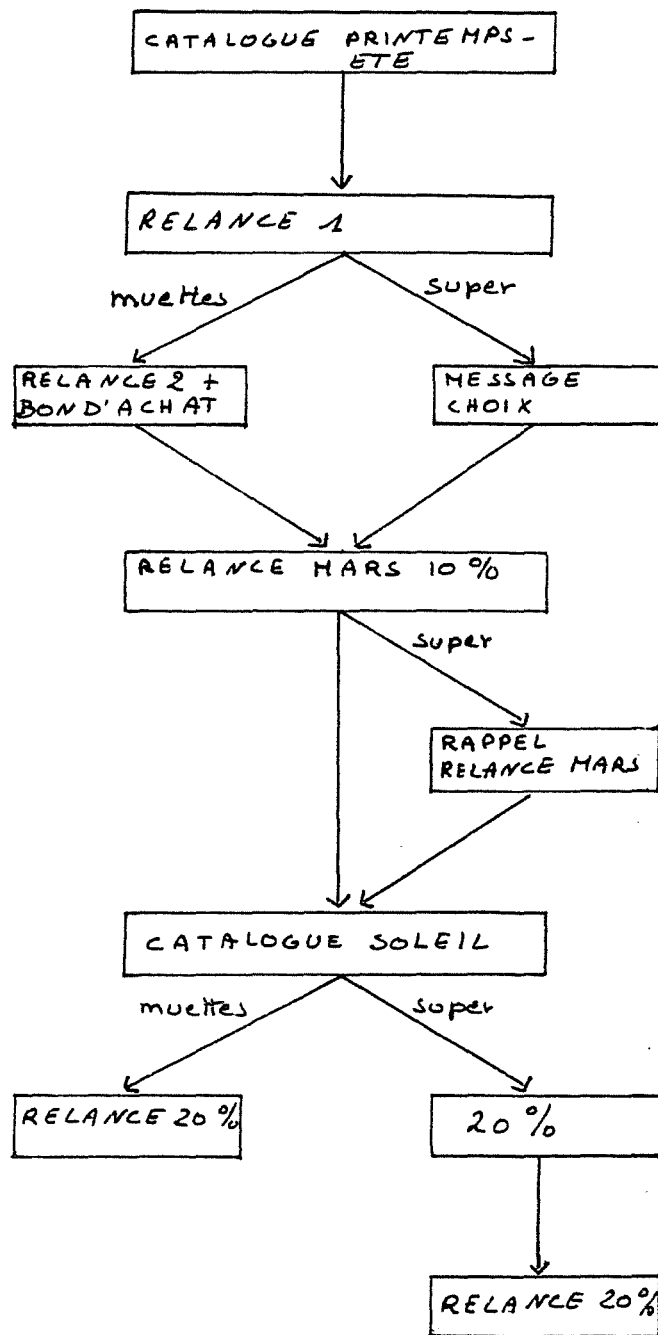


Figure 8. Plan de campagne Printemps-Eté

#### I.2.4. APPLICATION A REALISER

L'application que nous avons du réaliser pour l'entreprise 3 SUISSSES consiste en une optimisation mathématique du PVC. Ce qui revient à faire une optimisation de la marge de l'entreprise.

Il faut donc pour chaque couche du Fichier-Clients et pour chaque action du PVC, fournir le nombre d'envois à effectuer. Dans un premier temps, cette application tournera en parallèle avec le programme de simulation existant et l'on pourra alors en comparer les résultats et performances. Si les résultats du programme sont satisfaisants, on pourra dans un deuxième temps les entrer dans le programme de simulation, au niveau du PLAN. Le programme d'optimisation fournira en entrée au PLAN, le nombre d'envois à effectuer par action et par couche de clientes. Cette donnée étant cette fois-ci calculée à partir d'un programme linéaire et non plus statistiquement. Pour le reste, le programme de simulation restera tel qu'il est actuellement.

## CHAPITRE 1: ADAPTATION DU PVC A LA PROGRAMMATION LINEAIRE

Dans ce chapitre, nous allons présenter la modélisation du problème d'optimisation du PVC que nous avons réalisée.

Nous exposerons les différents problèmes que nous avons rencontrés lors de cette modélisation. Ce seront aussi bien des problèmes d'adaptation du PVC à notre modèle que des problèmes plus humains.

Nous exposerons les solutions que nous avons envisagées pour résoudre ces différents problèmes.

Enfin, nous terminerons ce chapitre en donnant quelques chiffres concernant la taille du problème.

Avant de commencer à développer ces différents points, il est utile de définir quelques notions dont nous aurons besoin plus loin.

RENDEMENT D'UNE ACTION: Nombre de commandes total pour l'action par rapport au nombre total de documents envoyés.

MOYENNE COMMANDEE D'UNE ACTION: Prix moyen des commandes pour l'action.

TYPES D'ENVOIS D'UNE ACTION: Les différents documents qui correspondent à une action, c'est à dire à une action peuvent correspondre différents types d'envois qui seront envoyés à des clientes ciblées: pour une même action, certaines clientes ont droit à un cadeau, d'autres à une remise, d'autres encore n'ont droit à aucun avantage, ...

COUT D'UN TYPE D'ACTION: Frais relatifs à ce type d'action: frais d'impression, frais d'envois...

Nous avons choisi de résoudre ce problème d'optimisation mathématique par la méthode du simplexe. Pour une présentation théorique de cette méthode, nous vous renvoyons à l'Annexe 1.

La résolution d'un tel problème de programmation linéaire commence par une phase de formulation d'un modèle, c'est-à-dire, la détermination de l'objectif que l'on désire atteindre, les variables de décision pour lesquelles on cherche une solution et les contraintes auxquelles les variables sont soumises.

Le problème est donc constitué de 3 éléments distincts:

- La fonction objectif
- Les variables
- Les contraintes

Il s'agira de maximiser (minimiser) cette fonction objectif tout en respectant les contraintes.

En programmation linéaire, on considère un objectif et des contraintes linéaires.

### 1.1 MODELISATION DU PROBLEME

Le problème qui nous a été soumis consiste donc à faire une optimisation du PVC en utilisant la Programmation Linéaire et plus particulièrement la méthode du Simplexe.

On peut traduire optimisation du PVC par optimisation de la marge de l'entreprise. La MARGE étant définie comme le CHIFFRE D'AFFAIRE - le COUT. Nous noterons CA pour Chiffre d'Affaire.

L'objectif à atteindre est donc de maximiser la marge, c-à-d

$$\text{MAX ( CA - COUT )}$$

Développons ces notions:

Nous supposons qu'il y a N actions (indice a), M couches de clientes (indice c), Ra types d'envois pour l'action a (indice t).

Le Coût total des actions programmées pour l'exercice courant est donné par :

$$\sum_c \sum_a \sum_t \text{nombre-envois (c,t,a) * coût (c,t,a)}$$

$c = 1, \dots, M$   
 $t = 2, \dots, Ra$   
 $a = 1, \dots, N;$

étant donné que nous connaissons le coût de chaque type d'envoi de chaque action. Remarquons que le coût d'un type d'action est le même pour toutes les couches de clientes. Néanmoins, nous avons introduit un indice de couche de clientes pour des raisons de cohérence avec la suite de la modélisation du problème.

Sachant que le Rendement (rdt) et la Moyenne Commandée (mcd) sont connus par action et par couche de clientes, le Chiffre d'Affaire de l'entreprise est donné par l'expression suivante:

$$\sum_c \sum_t \sum_a \text{rdt (c,t,a) * mcd (c,t,a) * nombre-envois (c,t,a)}$$

$c = 1, \dots, M$   
 $t = 1, \dots, Ra$   
 $a = 1, \dots, N$

Les variables de décision sont donc le nombre d'envois à effectuer par action, par type d'envoi et par couche de clientes.

La Fonction Objectif de notre problème est alors la suivante:

$$\text{MAX} \sum_c \sum_t \sum_a \text{rdt} (c,t,a) * \text{mcd} (c,t,a) * \\ \text{nombre-envois} (c,t,a) - \sum_c \sum_t \sum_a \text{coût} (c,t,a) *$$

nombre-envois (c,t,a)

Ce qu'on peut encore écrire de la façon suivante:

$$\text{MAX} \sum_c \sum_t \sum_a (\text{rdt} (c,t,a) * \text{mcd} (c,t,a) - \text{coût} (c,t,a)) * \\ \text{nombre-envois} (c,t,a)$$

$$c = 1, \dots, M$$

$$t = 1, \dots, R_a$$

$$a = 1, \dots, N$$

Nous avons déterminé trois types de contraintes sur les variables de décisions. Elles sont définies de la façon suivante:

(1) Pour chaque action considérée, le nombre d'envois effectué pour une couche de clientes ne peut évidemment pas dépasser le nombre de clientes de la couche.

On a donc,

Pour toute les couches,

Pour toutes les actions,

$$0 \leq \sum_t \text{nombre-envois} (c,t,a) \leq \text{nombre-cliente-couche}$$

$$t = 1, \dots, R_a$$

(2) Un des buts de ces actions est de faire augmenter le nombre de clientes du fichier actif.

Supposons que le taux de croissance du fichier soit fixé et posons le égal à X %. Posons  $Y = X + 1$ .

Alors,

$$\begin{aligned} & \text{Nombre-cli-actives} + \sum_a \sum_t \text{Pourcentage-reponse} (cn5,t,a) * \\ & \text{nombre-envois} (cn5,t,a) + \sum_a \sum_t \text{Pourcentage-reponse} (ab5,t,a) \\ & * \text{nombre-envois} (ab5,t,a) + \sum_a \sum_t \text{nouvelles-clientes} - \end{aligned}$$

$$\text{nombre-CN5} - \text{nombre-AB5} \geq \text{nombre-cli-actives} * Y$$

C'est à dire,

$$\begin{aligned} & \sum_a \sum_t \text{Pourcentage-reponse} (CN5,t,a) * \text{nombre-envois} \\ & (CN5,t,a) + \sum_a \sum_t \text{Pourcentage-reponse} (AB5,t,a) * \text{nombre-} \\ & \text{envois} (AB5,t,a) \geq \text{nombre-cli-actives} * (Y - 1) - \\ & \sum_a \sum_t \text{nouvelles-clientes} + \text{nombre-CN5} + \text{nombre-AB5} \end{aligned}$$

où les quantités suivantes sont connues :

- pourcentage-réponse (c,t,a)
- nombre-cli-actives
- nouvelles-clientes
- nombre-AB5
- nombre-CN5

En effet, les quantités nombre-cli-actives, nombre-AB5, nombre-CN5, sont calculables sur base des données du fichier clients. Quant aux deux autres quantités, pourcentage-réponse (c,t,a) et nouvelles-clientes, elles sont calculées à l'aide des historiques des années précédentes et de la connaissance que les personnes travaillant au service Marketing ont du problème.

(3) Enfin, il existe des contraintes entre les différentes actions programmées pour l'exercice courant.

En effet, si nous reprenons la figure 8 concernant le plan de campagne Printemps-Eté nous constatons les relations suivantes entre les actions concernées :

L'action principale étant le catalogue Printemps-Eté, il est évident que le nombre d'envois à effectuer pour les actions qui la suivent doit être inférieur ou égal au nombre d'envois effectué pour le catalogue Printemps-Eté.

De plus, certaines de ces actions ne sont pas destinées à toutes les clientes ayant reçu le catalogue Printemps-Eté, elles sont envoyées aux clientes SUPER ou MUETTES.

Remarquons qu'il serait long et inutile de faire la liste exhaustive des contraintes de ce type pour le PVC en entier. Pour plus de clarté, nous allons cependant développer ces contraintes pour le plan de campagne Printemps-Eté de la figure 8. Notons que le terme **Taux de Super** d'une couche de clientes utilisé ici correspond au pourcentage moyen de clientes super de la couche et est obtenu par une statistique sur le fichier clientes.

Pour toutes les couches, pour tous les types d'envois de l'action considérée,

nombre-envois (c,t,relance-1) = nombre-envois (c,t,catalogue-PE),

nombre-envois (c,t,relance-2-bon-achat) = (1 - taux de super (c)) \* nombre-envois (c,t,catalogue-PE),

nombre-envois (c,t,message-choix) = taux de super (c) \* nombre-envois (c,t,catalogue-PE),

nombre-envois (c,t,relance-mars) = nombre-envois (c,t,catalogue-PE),

nombre-envois (c,t,rappel-relance-mars) = taux de super (c) \* nombre-envois (c,t,catalogue-PE),

nombre-envois (c,t,catalogue-soleil) = nombre-envois (c,t,catalogue-PE),

nombre-envois (c,t,relance-20%) = (1 - taux de super (c)) \* nombre-envois (c,t,catalogue-PE),

nombre-envois (c,t,20%) = taux de super (c) \* nombre-envois (c,t,catalogue-PE),

Nous verrons plus loin comment nous avons résolu le problème de génération de ces contraintes.

### 1.2. Linéarisation des données et autres problèmes

La transformation du PVC sous forme de modèle mathématique linéaire résolvable par la méthode du simplexe ne semble donc pas poser de problèmes majeurs.

Seul le troisième type de contraintes ne peut être défini explicitement. Il est en effet impossible de leur donner une formulation mathématique générale indépendante du type de l'action et de son contexte.

Pour résoudre ce problème, nous avons implémenté un générateur de contraintes. Etant donné le type d'action et les clientes concernées par cette action, notre programme générera automatiquement la contrainte qui lui correspondra.

Un problème de linéarisation est apparu lors de l'étude de l'optimisation du PVC. En effet, le rendement et la moyenne commandée d'une action sont des valeurs connues pour chaque couche de clientes, et dans une couche on dispose des chiffres par tranches de 5 points d'acquis. Rappelons que l'acquis d'une cliente est une pondération de cette cliente à l'intérieur de la couche à laquelle elle appartient.

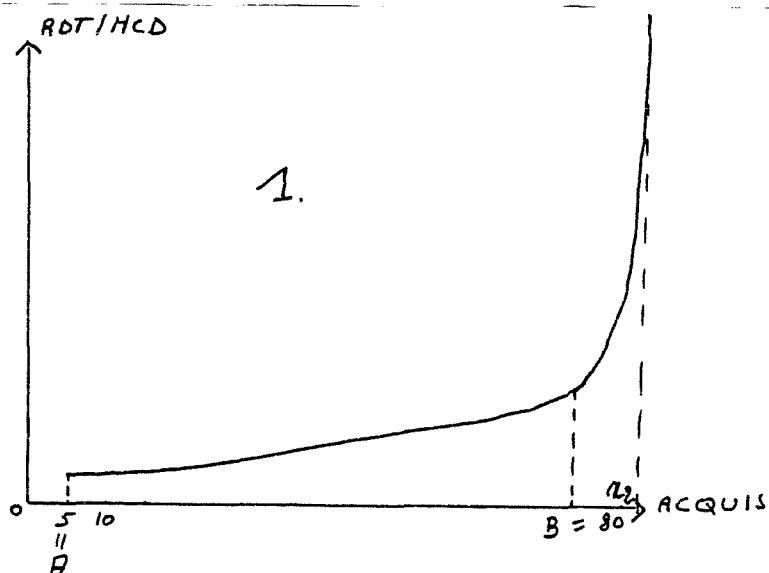
Nous disposons donc de courbes de rendement et de moyenne commandée pour chaque couche et pour chaque action.

Ces courbes sont disponibles pour les exercices précédents, les calculs pour l'année en cours étant basés sur ces historiques. De plus, ces courbes ne sont pas linéaires. Nous avons alors fait une étude des courbes de rendement et de moyenne commandée.

Nous avons plus spécialement étudié les chiffres des actions se rapportant au catalogue Printemps-Eté 87 (PE 87).

Nous avons obtenu les chiffres par action, par couche et par tranche de 5 points d'acquis. Nous avons constaté que l'allure générale des courbes était de 2 types, comme nous le montre la figure 9.

Lors de l'étude des courbes, nous avons ignoré les actions ou les couches de clientes pour lesquelles le nombre d'envois effectué était trop faible, les courbes obtenues étant peu fiables.



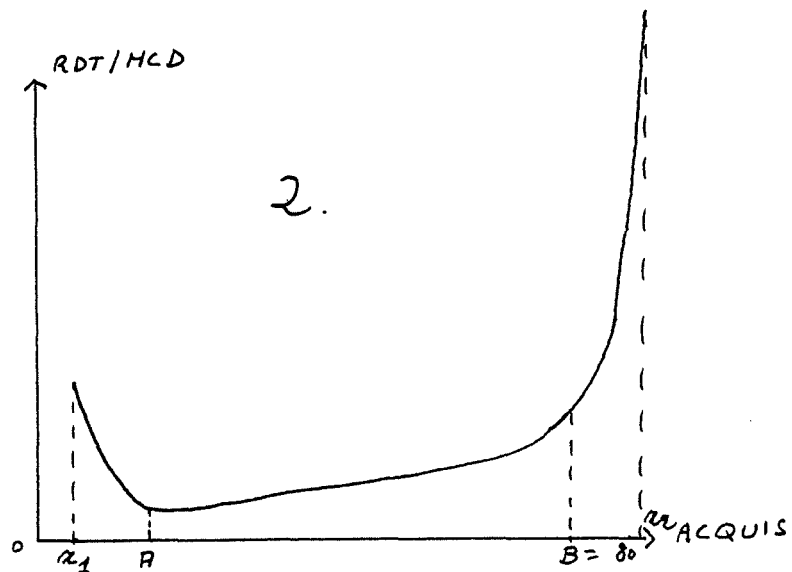


Figure 9. courbes de rendement et de moyenne commandée.

Nous constatons que ces courbes ne sont pas linéaires, il est donc impossible de prendre la moyenne de la courbe pour représenter le rendement ou la moyenne commandée d'une couche de clientes pour chaque action.

Mais si ces courbes ne sont pas linéaires, elles sont cependant assez simples.

Pour la majorité des courbes étudiées, nous avons constaté les particularités suivantes: A partir de 80 points d'acquis la croissance de la courbe augmente brusquement. De 0 à 80 points d'acquis, la croissance est plus lente. Sur certaines courbes, nous avons constaté un pic pour des faibles valeurs de l'acquis.

C'est pourquoi nous avons retenu ces 2 formes de courbes.

Initialement, nous avons pensé à faire une étude de reconnaissance des courbes, c'est à dire trouver à quel type de courbe nous étions confrontés: expo, log, polynôme, ..., et ensuite prendre la moyenne sur cette courbe. Mais nous n'avons pas implémenté ce procédé car l'allure assez simple des courbes ne demandait peut-être pas une analyse aussi complexe.

Nous avons alors choisi d'utiliser la régression linéaire morceau par morceau pour approcher ces courbes. En effet, leurs allures générales se prêtent bien à cette méthode d'approche et nous n'y perdons rien précision. Pour un rappel théorique de la régression linéaire, nous vous reportons à l'Annexe 1.

On remarque sur la figure 9 que ces courbes sont plus ou moins linéaires par morceaux. Dans le cas de la première courbe, nous aurions 2 régressions linéaires à effectuer et dans le cas de la deuxième courbe nous en aurions 3.

Nous avons adapté la régression linéaire à notre problème de la façon suivante:

Appelons A l'abscisse du point d'ordonnée minimale d'une courbe de rendement, par exemple. Et appelons B l'abscisse de valeur 80. Voir figure 9.

Le premier problème à résoudre est donc de déterminer s'il y aura 2 ou 3 régressions linéaires à effectuer. Pour cela, il faut déterminer A en prenant le minimum du rendement donné et en prenant pour A l'abscisse de ce point minimum.

Si cette abscisse correspond au premier acquis donné de la courbe, alors on est dans le cas 1 et il faudra faire 2 régressions linéaires, sinon on est dans le cas 2 et il faudra faire 3 régressions linéaires.

Ayant déterminé A, nous pouvons passer aux régressions linéaires proprement dites.

On calcule alors la droite de régression sur [A,B], soit  $F(x) = ax + b$  cette droite, par application des formules se trouvant en annexe 1. On calcule ensuite  $F(A) = aA + b$

$$F(B) = aB + b$$

S'il y a 3 régressions linéaires à effectuer, nous calculons alors la droite de régression sur  $[x_1, A]$ , avec  $x_1 < A$  et la contrainte que la droite de régression sur [A,B] et cette droite de régression-ci se coupent en  $F(A)$ . La figure 10 ci-dessous permet de visualiser pourquoi nous devons imposer une telle contrainte.

En effet, si nous n'avons pas cette contrainte, rien ne permettrait d'assurer que les 2 droites de régression se coupent et on pourrait avoir des droites comme dans le cas 1.

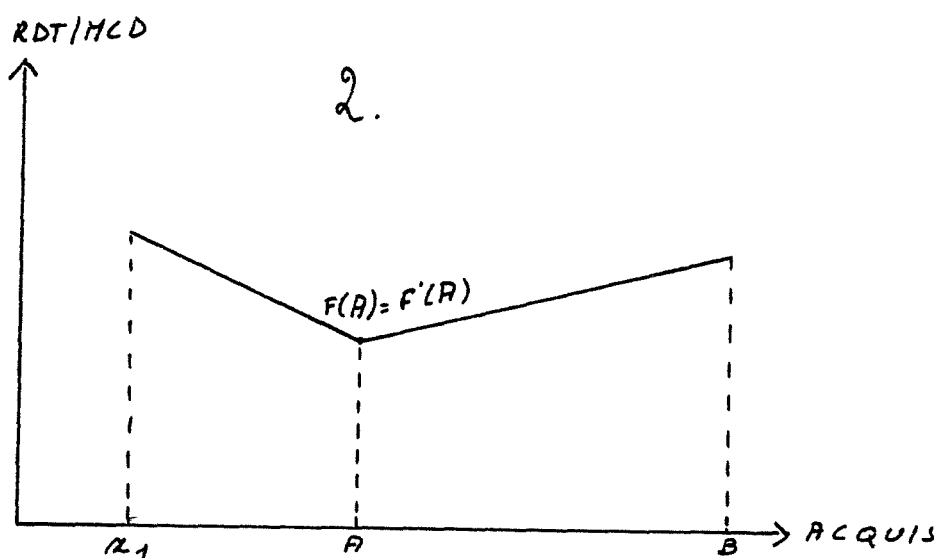
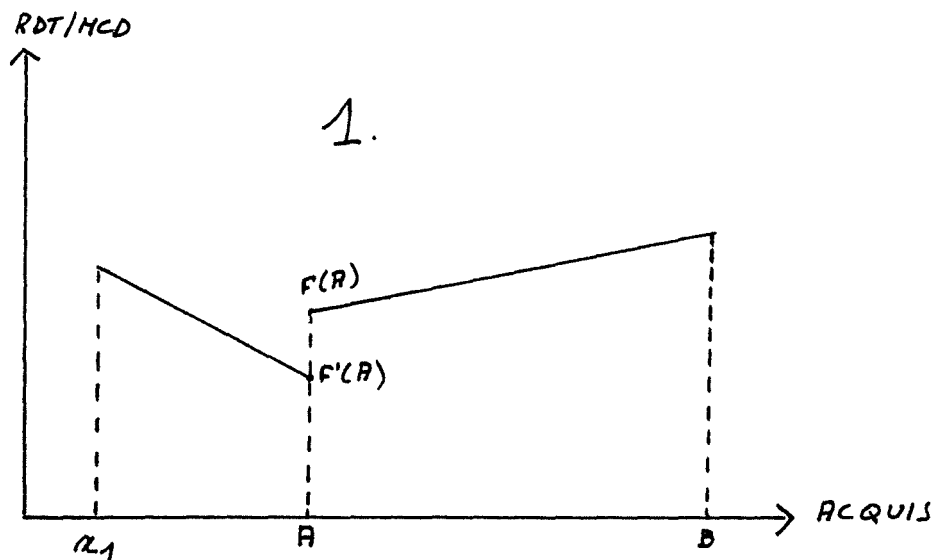


figure 10. les droites de régression

Mathématiquement, cela donne les formules suivantes:

Si  $F'(x) = a'x + b'$  est la droite de régression sur  $[x_1, A]$ ,

il faut que la condition suivante soit réalisée,

$$F(A) = F'(A) = a'A + b'$$

et donc,

$$b' = F(A) - a'A$$

$$a' = - \frac{\sum_{i=1}^n ((y_i - F(A)) * (A - x_i))}{\sum_{i=1}^n (A - x_i)^2}$$

De la même façon, et dans les 2 cas, nous faisons une régression linéaire sur [B,x2], en imposant que x2 <> B et que la droite de régression sur [A,B] et cette droite de régression-ci se coupent en F(B).

Les formules obtenues sont identiques au cas précédent en remplaçant A par B.

De plus, pour les besoins de notre programme, nous calculons les coordonnées de la valeur moyenne de chaque droite de régression.

Donc, par l'étude des chiffres de rendement et de moyenne commandée qui nous sont fournis, nous pouvons décider si nous devons découper la couche en 2 ou 3 tranches. Sur chacune de ces tranches, nous effectuons une régression linéaire. Ensuite, pour chaque tranche, nous calculons la moyenne de la régression.

Cette façon de faire suppose alors que nous ne travaillons plus par couche de clientes, mais par tranche de clientes.

Donc dans la modélisation du PVC, nous considérerons plutôt le nombre d'envois par tranche, par action et par type d'envoi, c'est à dire

$$\text{nombre-envois } (c,t,a) = \sum_{tr} \text{nombre-envois } (tr,t,a)$$

Il en est de même pour les autres quantités apparaissant dans la modélisation.

Cependant, lors de la sortie des résultats nous considérerons à nouveau

nombre-envois (c,a,t)

puisque c'est le nombre d'envois par couche qui nous intéresse et non le nombre d'envois par tranche d'une couche.

### 1.3. TAILLE DU PROBLEME

Le programme linéaire que nous avons implémenté sera capable de résoudre le problème de l'optimisation du PVC de l'entreprise 3 Suisses. Les dimensions du tableau du simplexe seront donc très importantes. Il est intéressant de se rendre compte des dimensions réelles du problème.

Au cours d'un exercice, il y en moyenne 30 actions lancées. En outre, nous savons qu'il y a 12 couches de clientes et comme nous avons décidé de travailler par tranches de clientes, il y aura au plus

$$12 * 3 = 36 \text{ tranches de clientes considérées}$$

et il y aura au moins

$$12 * 2 = 24 \text{ tranches de clientes considérées}$$

On notera encore que le nombre de types d'envois pour une action peut varier de 1 à 4.

On peut alors calculer les bornes inférieures et supérieures du nombre de variables de décision:

$$\text{Au minimum: } 30 * 24 * 1 = 720$$

$$\text{Au maximum: } 30 * 36 * 4 = 4320$$

En ce qui concerne les contraintes, nous avons les chiffres suivants:

contraintes de type (1)

$$\text{Au minimum: } 30 * 24 = 720$$

$$\text{Au maximum: } 30 * 36 = 1080$$

les contraintes de type (2) et (3) sont chacune unique.

Le tableau du simplexe aura donc dans le pire des cas 4000 variables et 1000 contraintes.

l'implémentation d'un problème de cette taille pose évidemment des problèmes de temps de calcul, de place mémoire et de précision des calculs.

### CHOIX D'IMPLEMENTATION

La solution que nous avons implémentée pour résoudre le problème de l'optimisation du PVC est donc la méthode du simplexe. Or, si nous considérons cette méthode, nous constatons qu'en pratique elle n'est pas optimale du point de vue occupation de place mémoire et du point de vue temps de calcul. C'est pourquoi, lorsqu'on veut implémenter la méthode du simplexe sur ordinateur, on implémente plutôt la forme révisée de la méthode du simplexe ou la forme produit de l'inverse. Dans notre cas, nous avons implémenté la forme révisée de la méthode du simplexe qui permet un gain de place mémoire et de temps de calcul. En effet, cette méthode ne demande que la mémorisation des données initiales du problème et de l'inverse de la base courante. De plus, il ne faut plus calculer l'entièreté du tableau du simplexe à chaque itération de la méthode.

Cependant, le problème de la perte de précision des calculs reste présent avec cette méthode. On peut en effet remarquer une détérioration des résultats au fur et à mesure des pivotages. A chaque itération, l'inverse de la base étant obtenue par pivotages successifs à partir de l'inverse de la base initiale.

Ces pivotages provoquent des erreurs d'arrondi à chaque itération, et ces erreurs se propagent dans les itérations suivantes.

Il faut alors prévoir une réinversion de la base à partir des données initiales pour éviter cette propagation d'erreurs.

C'est pourquoi nous avons prévu, dans notre programme, une réinversion de la base toutes les 20 itérations.

Nous avons utilisé la méthode de décomposition LU que nous avons adaptée à notre problème. Nous vous référons à l'annexe 1 pour un bref rappel de cette méthode.

De plus, comme nous l'avons déjà indiqué lors de l'exposé des problèmes, nous n'avons pas fait une simple implémentation de la méthode du simplexe, mais nous y avons inclu un générateur de programme linéaire qui générera l'expression du programme linéaire d'optimisation du PVC sur base des renseignements fournis par l'utilisateur, le programme créera donc la fonction objectif et le système des contraintes. Plus particulièrement ceci permet de résoudre le problème des contraintes du troisième type que nous avons définies dans la modélisation du PVC, c'est à dire les contraintes entre les actions.

L'utilisateur devra simplement préciser lors de l'entrée des données le type de l'action courante (catalogue principal, catalogue secondaire, relance, prospection) ainsi que le type de clientes qui sont concernées par cette action (muettes, super, toutes les clientes).

Avec ces données, le programme génère alors les contraintes qui existent entre cette action et les autres actions du PVC déjà introduites dans le programme par l'utilisateur.

Enfin, nous avons également prévu la possibilité pour l'utilisateur de résoudre tout autre problème d'optimisation que celui du PVC. Notre programme est donc capable de résoudre ces problèmes pour peu que l'utilisateur ait défini son problème sous forme de programme linéaire. La mise sous forme standard étant prise en charge par le programme.

Un dernier point à préciser concerne la Base de Données que nous avons utilisée. Il s'agit d'une base de données relationnelle permettant de stocker les données des différents problèmes de programmation linéaire que l'on pourrait vouloir résoudre à l'aide de la méthode du simplexe, ainsi que les résultats intermédiaires des itérations.

Cette base de données comprend 10 tables dont la description complète se trouve dans l'annexe 2.

Remarquons que la codification des noms est standardisée de façon à la construire de manière presque automatique à partir du libellé lui-même.

Par exemple,

PLS: Programme Linéaire Standard

VAR: Variable

LGN: Ligne

#: Numéro de

#N: Nombre de

#L: Libellé de

Tous les programmes écrits aux 3 SUISSES doivent respecter les abréviations standards. Ceci permet une plus grande cohérence et lisibilité des programmes.

## CHAPITRE 2.1 LES OUTILS

Dans ce chapitre, nous allons décrire les outils qui ont été mis à notre disposition pour mener à bien notre projet d'optimisation du PVC des 3 Suisses.

Il s'agit essentiellement d'outils matériels et logiciels. Plus particulièrement, nous allons présenter le langage que nous avons utilisé pour implémenter notre application: IDEAL de ADR (Applied Data Research), ses avantages, ses inconvénients.

### 2.1. OUTILS MATERIELS

Le service qui s'occupe de la vente par correspondance est doté d'un ordinateur IBM 4381 sous DOS/VSE. Une vingtaine de terminaux y sont connectés.

Il existe d'autres outils dans l'entreprise mais ne les ayant pas utilisés, nous les passons sous silence.

### 2.2. OUTILS LOGICIELS

Les langages disponibles sur l'IBM sont:

- Assembler
- RPG
- Ideal

Le logiciel que nous avons utilisé pour impémenter notre application est celui fourni par ADR. Les composants de ce logiciel sont les suivants

- Le DATACOM/DB (Database Management Component) qui est un SGBD relationnel. Dans un environnement DATACOM/DB, il suffit de connaître les données auxquelles on veut accéder, la structure interne et l'emplacement physique des données étant

connus du système.

- Le DATACOM/DATADICIONNARY.

Le dictionnaire est une base de données relationnelle gérée par Datacom et qui permet à la fois de documenter et de standardiser. Le dictionnaire sert en fait de répertoire de données descriptives.

- Le langage IDEAL est un langage de programmation structurée L4G qui contient une trentaine d'instructions.

#### 2.2.1. IDEAL

Notre programme d'optimisation ayant été implémenté en Ideal, il semble intéressant de décrire brièvement ce langage assez récent. Il est diffusé principalement sur les grands sites.

Il s'agit d'un système de développement d'applications qui fournit un environnement interactif pour le développement, la maintenance et l'exécution d'applications.

Les composants d' Ideal sont les suivants:

- panels
- reports
- dataviews
- langage structuré de haut niveau pour la conception et le développement des procédures d'application.

La structure d'Ideal est la suivante: A chacun de ses composants correspond un ensemble d'écrans que l'utilisateur doit ou peut remplir s'il en a besoin. décrivons les brièvement.

## panels

- **Identification:** description de l'écran.
- **Parameters:** Permet de spécifier des options générales pour la définition du panel.
- **Layout:** Ecran vide dans lequel l'utilisateur vient placer des champs aux positions désirées.
- **Field summary table:** Liste des champs entrés dans le layout. Entrée d'informations associées à chaque champ: (nom, type, occurrences, ...).
- **Extended field definition:** Définition des types de manière plus détaillée.
- **Facsimile:** Affichage d'un facsimile du panel tel qu'il apparaîtra lors de l'exécution d'un programme.

Un panel peut être imprimé, effacé, affiché ou dupliqué.

## Report

- **Identification:** Description du report.
- **Parameters:** Définition de plusieurs options (longueur de page, espace entre les lignes, ...).
- **Page heading définition:** Spécification plus détaillée des pages.
- **Report detail définition:** Spécification des champs qui apparaissent dans chaque "detail line" du corps du report, des tris à effectuer, ...

Un report peut être imprimé, effacé, affiché ou dupliqué.

Dataview: Les dataviews proviennent de la Base de données et permettent au programme utilisateur d'interagir avec cette base de données.

Program:

- **Identification:** Description du programme.
- **Ressources:** Spécification des ressources utilisées par le programme:
  - Dataview
  - panels
  - reports
  - sous-programmes

L'information qui identifie les ressources d'un programme est stockée dans ADR/DATADICTIONNARY et est utilisée par Ideal pour les reports et pour s'assurer que le programme utilise seulement les ressources pour lesquelles il a l'autorisation.

-**Working data definition:** Permet de nommer et de décrire les données locales d'un programme: niveau, nom de champ, type, nombre de caractères ou de champs, nombre d'occurrences, ...

-**Parameters definition:** Permet de nommer et de décrire les données qui sont passées entre le programme appelant et les sous-programmes.

Ideal dispose en outre d'un ensemble d'instructions et de primitives de base semblable à beaucoup de langages procéduraux: move, set, add, do, loop, if, select, for each, for any, for first, for new, for next ( ces cinq dernières instructions permettent de traiter un ensemble de records (ou un seul record) dans une dataview). transmit, refresh, reset permettent d'envoyer, de recevoir et de traiter des définitions de panels, ...

Nous ne détaillerons pas ici toutes les instructions disponibles en Ideal. Remarquons seulement que ce langage est très riche et très aisé d'apprentissage. En effet, que ce soit pour la définition des panels ou des programmes, l'éditeur affiche l'écran correspondant et l'utilisateur n'a plus qu'à remplir le squelette de définition qui lui est proposé. Par exemple, la définition des variables locales se fait en remplissant l'écran de Working Data correspondant au programme considéré.

De même, lors de l'écriture du programme, la connaissance parfaite de la syntaxe d'Ideal n'est pas indispensable, il suffit de taper le nom ou l'abréviation de l'instruction voulue et son squelette apparaît à l'écran. L'utilisateur n'a alors plus qu'à remplir les blancs.

Cependant, la syntaxe d'Ideal étant assez simple, ce mécanisme n'est pas indispensable, sauf peut être au début de l'apprentissage de Ideal.

On peut noter aussi l'avantage de pouvoir définir panels, reports et programmes dans un même formalisme et la facilité d'utilisation d'un panel ou d'un report par un programme où ils sont considérés comme des ressources au même titre que les sous-programmes.

Quant à l'accès aux Dataviews, elles sont prévues et gérées par les instructions de base du langage.

Un point négatif peut être, est le fait qu'Ideal travaille par écran. Tout passage à l'écran suivant ou à un autre écran du programme se fait par un accès au système et doit être demandé par l'utilisateur à l'aide d'une commande spécifique. Il en est de même pour avoir accès à la Working data ou aux ressources du programme.

Il serait parfois plus pratique de voir défiler le programme quand on en cherche une partie sans savoir exactement où elle se trouve.

Cependant, l'impression générale qui se dégage de Ideal est positive de part sa facilité d'utilisation et son efficacité. Pour une vue plus concrète de ces différentes notions, nous vous référons à l'annexe 2 où sont repris les panels et les programmes que nous avons réalisés dans le cadre de notre projet d'optimisation du PVC des 3 Suisses.

## CHAPITRE 3: RESULTATS ET MANUEL D'UTILISATION

Dans ce dernier chapitre, nous allons parler des résultats que nous avons obtenus pour notre programme d'optimisation du PVC des 3 Suisses. Nous ferons le point sur l'état d'avancement des tests relatifs à notre programme.

Nous terminerons en donnant le manuel d'utilisation de notre programme.

### 3.1. LES RESULTATS

Avant de passer aux résultats proprement dits de notre programme, il nous semble important de faire une remarque qui pourra peut-être expliquer l'état d'avancement des tests.

Un des problèmes que nous avons rencontré lors de notre stage ne concerne pas directement le travail qui nous a été confié. Il s'agit de la difficulté que nous avons rencontrée à faire accepter cette nouvelle technique d'étude de la gestion du PVC aux personnes en place utilisant une autre méthode de résolution. Cela peut s'expliquer par le fait qu'il est toujours difficile de s'habituer à une nouvelle technique de travail.

La conséquence de ceci a été la difficulté d'obtenir les données dont nous avons besoin pour étudier les courbes de rendement et de moyenne commandée.

De même, nous avons dû attendre assez longtemps pour avoir notre base de données.

Tout ceci pour dire que nous avons perdu un temps précieux qui aurait pu être utilisé à la fin de notre stage pour effectuer des tests supplémentaires.

Voici donc l'état actuel d'avancement des test et les résultats que nous avons obtenus:

Les tests que nous avons effectivement réalisés lors de notre stage portent uniquement sur la méthode du simplexe proprement dite. Les exemples que nous avons utilisés se trouvent tous dans [FICH]. Nous avons testé les types de programmes linéaires suivants:

- programme linéaire avec des contraintes d'égalité,
- programme linéaire avec des contraintes d'inégalité  $\leq$ ,
- programme linéaire avec des contraintes d'inégalité  $\leq$  et  $\geq$ ,
- programme linéaire dont le système des contraintes est redondant,
- programme linéaire dont les contraintes sont incompatibles,

Nous n'avons pas eu le temps de recouvrir tous les cas de figure qui peuvent se présenter pour un programme linéaire, nous nous sommes limités aux cas les plus généraux. Nous ne garantissons donc pas le parfait fonctionnement de notre programme face à tous les programmes linéaires. Cependant les résultats obtenus jusqu'ici sont encourageants.

Il faut préciser que les programmes linéaires que nous avons testés sont de petite taille, ils ont en moyenne 4 variables et 4 contraintes et ne demandent pas plus de 10 itérations.

Les résultats que nous a fournis le programme d'optimisation sont conformes aux résultats attendus en tenant évidemment compte des erreurs d'arrondis inévitables. Ces erreurs ne sont cependant pas très importantes et rappelons que l'algorithme de réinversion de la base est là pour remédier à cet inconvénient.

Nous avons testé cet algorithme de réinversion séparément du programme d'optimisation car les programmes d'optimisation que nous avons testés n'allaient jamais jusqu'à 20 itérations.

Donc, dans cet état d'avancement des tests, notre programme semble fonctionner d'une manière acceptable.

Ces tests ont clôturé notre stage aux 3 Suisses.

En ce qui concerne les tests sur le Plan de Vente, ils ont été pris en charge par le service informatique des 3 Suisses.

### 3.2. MANUEL D'UTILISATION

La présentation de l'application se fait par un menu général (PN0001) qui propose 2 possibilités:

- 1- optimisation d'un plan de vente
- 2- optimisation d'un autre problème

Derrière le mot optimisation ne se cache pas seulement la méthode du simplexe. En effet, dans le cas du PVC, il comprend également la génération du programme linéaire correspondant et le programme de régression linéaire.

Si le choix effectué est celui du PVC, alors le menu PN0222 s'affiche à l'écran et propose 4 possibilités:

- 1- création d'un PVC
- 2- exécution d'un PVC
- 3- suppression d'un PVC
- 4- modification d'un PVC

Remarquons que seuls les 3 premiers points ont été implémentés. Les menus pour la modification d'un PVC sont prévus, mais les programmes qui les gèrent ne sont pas écrits.

Si le choix effectué est la création d'un PVC, alors les menus suivants se succèdent pour permettre à l'utilisateur de rentrer les données nécessaires à l'optimisation du PVC:

1) PN0010: "données du fichier clients", l'utilisateur doit ici fournir manuellement les données nécessaires au programme concernant le fichier client. Il s'agit du nombre de clientes dans chaque couche, du taux de croissance du fichier qui a été prévu pour le PVC (ce taux de croissance est en général de 2%) et le nombre de nouvelles clientes estimé pour le PVC. Le nombre de clientes par couche est nécessaire pour définir les contraintes du premier type, - c'est à dire que le nombre d'envois par couche doit être inférieur au nombre de clientes dans la couche. Le taux de croissance du fichier est utilisé pour générer le deuxième type de contraintes. Cet écran n'apparaît qu'une seule fois, au début de la définition du PVC.

2) PN0011: "description générale du plan de vente",  
l'utilisateur peut ici, s'il le désire, donner une  
description littérale générale du PVC. Ensuite, il doit  
fournir le nombre d'actions que contient le PVC. Le programme  
lui fournit alors le numéro qui correspondra à ce nouveau  
problème. Cet écran n'apparaît, lui aussi, qu'une seule fois,  
au début de la définition du PVC.

3) PN0012: "saisie des données de la couche ---".

Cet écran apparaîtra autant de fois qu'il y a d'actions  
déclarées par l'utilisateur à l'écran précédent. L'utilisateur  
donne une description générale de l'action courante. Ensuite,  
il doit donner les informations suivantes concernant  
l'action: son numéro, son coût, le nombre de couches qui sont  
concernées par cette action, et enfin les clients concernés  
par l'action et le type de l'action. Ces deux dernières  
informations permettent de définir le troisième type de  
contraintes, c'est à dire les contraintes entre les actions.

4) PN012B: "saisie des données de la couche ---".

Pour l'action courante, cet écran apparaîtra autant de fois  
qu'il y a eu de couches déclarées par l'utilisateur dans  
l'écran précédent. L'utilisateur devra ici donner le nom de  
la couche, le taux de super de la couche et le nombre de  
points considérés pour les chiffres de rendement et de  
moyenne commandée, c'est à dire le nombre de points d'acquis  
que l'on va considérer pour traiter les chiffres de rendement  
et de moyenne commandée lors de la linéarisation des courbes  
formées par ces données.

5) PN0013: "saisie du rendement et de la moyenne commandée".  
Cet écran permet à l'utilisateur de donner les chiffres de rendement et de moyenne commandée pour l'action et la couche courante. Remarquons que le programme ne teste pas si l'utilisateur lui fournit des données acceptables. Nous avons en effet décidé de ne pas considérer les chiffres pour les couches où le nombre d'envois est trop faible et donc le résultat pas fiable pour nos calculs.

Ce dernier écran d'entrée des données va être suivi par la création proprement dite du programme linéaire du PVC, c'est à dire la fonction objectif et le système des contraintes. C'est donc à ce niveau qu'intervient le programme de régression linéaire. Sur base des données enregistrées par l'écran PN0013, il détermine le nombre de variables du problème et la valeur du rendement et de la moyenne commandée pour chacune de ces variables.

Si le choix effectué est l'exécution ou la suppression d'un PVC, l'utilisateur peut faire appel à l'écran PN0022 qui lui fournira la liste des PVC disponibles dans la base de données. L'écran PN0026 permet à l'utilisateur de donner le numéro du PVC à exécuter ou à supprimer.

Les résultats de l'exécution sont fournis à l'utilisateur par les écrans PN0220 et PN0221.

Si le choix dans le menu principal est l'optimisation d'un problème linéaire autre que le PVC, le principe est le même que pour le premier cas. Seule la partie concernant l'entrée des données est différente.

Les écrans définis pour cette partie sont les suivants:

FN0021, FN0022, FN0023, FN0024, FN0025 pour l'entrée des données par l'utilisateur, FN00218 et FN0219 pour la sortie des résultats fournis par le programme.

La représentation de ces écrans se trouve ci-dessous.

PANEL:

PN0001

VERSION: 001 STATUS: TEST

FACSIMILE:

```
!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!  
!          MENU PRINCIPAL          !  
!          -----          !  
!          1- OPTIMISATION DU PLAN DE VENTE          !  
!          2- OPTIMISATION D'UN AUTRE PROBLEME          !  
!          VOTRE CHOIX : _          !  
!          -----          !  
!          PF6 POUR FIN DE TRAVAIL          !  
!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!
```

09

PANEL:

PN0222

VERSION: 001 STATUS: TEST

FACSIMILE:

```
! .....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!  
!                                     menu pvc  
!                                     -----  
!                                     1. creation d'un pvc  
!                                     2. execution d'un pvc  
!                                     3. suppression d'un pvc  
!                                     votre choix : _  
! -----  
! $pf6 pour fin de travail  
! .....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!
```

19

PANEL: PN0010 VERSION: 001 STATUS: TEST

FACSIMILE:

```

.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!
                                     donnees fichier clientes
-----
! nombre cni : _____
! nombre cn1 : _____
! nombre cn2 : _____
! nombre cn3 : _____
! nombre cn4 : _____
! nombre cn5 : _____
! nombre ab2 : _____
! nombre ab3 : _____
! nombre ab4 : _____
! nombre ab5 : _____
! nombre elm : _____
! nombre pro : _____
! taux de croissance du fichier : _____
-----
! pf6 pour fin de travail
.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!

```

62

PANEL:

PN0011

VERSION: 001 STATUS: TEST

FACSIMILE:

```

!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!
!              description generale du plan de vente
!-----!
! libelle du probleme   :
!
! _____
! _____
! _____
! _____
! _____
!
! numero du probleme   :  ___
!
!
!
! nombre d' actions    :  ___
!
!-----!
! pf6 pour fin de travail
!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!

```

63

PANEL:

PN0012

VERSION: 001 STATUS: TFST

FACSIMILE:

```

.....1.....2.....3.....4.....5.....6.....7.....8!
      saisie des donnees action  _
-----
! libelle d'action   : _____
! numero d'action   : _____
! cout de l'action  : _____
! nombre de couches : _____
! clientes concernees : muettes (m)
                        supers (s)
                        toutes (t)
! type d'action : catalogue principal (cp)
                        catalogue secondaire (cs)
                        relance (rl)
                        prospection (pr)
-----
! pf6 pour fin de travail
.....1.....2.....3.....4.....5.....6.....7.....8!

```

h9

ANNEXES

## ANNEXE 1

Dans cette première annexe, nous allons faire un bref rappel théorique des notions indispensables à connaître pour comprendre l'ensemble du travail que nous avons effectué concernant l'optimisation du PVC.

Nous commencerons par rappeler la méthode du simplexe dans un cadre général. Nous en déduirons la forme révisée de la méthode du simplexe, ses particularités et ses raisons d'être.

Nous présenterons ensuite les principes généraux de la méthode de décomposition LU qui permet le calcul de l'inverse d'une matrice.

Et enfin, nous parlerons de la régression linéaire.

### A1.1. LA METHODE DU SIMPLEXE

Le problème général de la programmation linéaire simple s'écrit de la façon suivante:

$$\text{optimiser } z = \sum_{j=1}^n C_j X_j$$

sous des contraintes:

$$\sum_{j=1}^n A_{ij} X_j \geq B_i; \quad i = 1, \dots, p$$

$$\sum_{j=1}^n A_{ij} X_j \leq B_i; \quad i = p + 1, \dots, q$$

$$\sum_{j=1}^n A_{ij} X_j = B_i; \quad i = q + 1, \dots, m$$

$$X_j \geq 0; \quad j = 1, \dots, r$$

$$X_j \text{ quelconque}; \quad j = r + 1, \dots, n$$

où les  $A_{ij}$ ,  $B_i$  et  $C_j$  ( $i = 1, \dots, m$ ;  $j = 1, \dots, n$ ) sont des constantes données.

Dans un tel problème;

- Les  $X_j$  sont appelés les variables du problème;
- La fonction  $z$  est appelée la fonction objectif ou la fonction économique.

On peut définir la forme standard d'un problème linéaire:

$$\min z = \sum_{j=1}^n C_j X_j$$

sc.

$$\sum_{j=1}^n A_{ij} X_j = B_i \quad i = 1, \dots, m$$

$$j = 1$$

$$X_j \geq 0 \quad j = 1, \dots, n$$

Remarquons qu'il est très facile de passer de la forme générale à la forme standard, et ceci pour tout problème linéaire.

Nous distinguons ici algorithme et méthode du simplexe. L'algorithme du simplexe applicable aux problèmes linéaires standard et où l'on connaît une base réalisable de départ est destiné à l'obtention de la solution proprement dite: valeur de la fonction économique, variables en base et leur valeur. La méthode du simplexe se propose de lever l'hypothèse portant sur la connaissance d'une base réalisable de départ, permet de déterminer si les contraintes sont compatibles, si le système des contraintes contient des équations redondantes et si la fonction économique possède un optimum fini sous ces contraintes.

Enfin, cette méthode contient l'algorithme du simplexe comme technique de calcul.

Il existe diverses méthodes permettant de calculer une base réalisable de départ. En l'occurrence, nous nous sommes attachés à la méthode des 2 phases qui attaque ce problème par la résolution d'un problème auxiliaire qui fournit comme solution la base réalisable de départ pour le problème initial.

Pour une vue plus détaillée de cette méthode, nous vous référons à [Fich].

Lorsqu'il s'agit de programmer l'algorithme du simplexe, on constate qu'il est nécessaire, à chaque itération de la méthode de calculer un nouveau tableau du simplexe complet et de le substituer au tableau de l'itération précédente. Chaque tableau ayant la dimension  $(m + 1) \times (n + 1)$ .

On constate également que:

- Seule la ligne des coûts relatifs ( $C_j$ ), et même la partie de la ligne limitée aux coûts relatifs des variables indépendantes sert à choisir le nouveau vecteur qu'il faut faire entrer dans la base.
  - Seules la colonne des seconds membres des contraintes et la colonne correspondant au nouveau vecteur entrant dans la base servent à choisir le vecteur qu'il faut éliminer de la base.
  - L'inverse de la nouvelle base permet de calculer les nouveaux éléments du tableau du simplexe qui lui correspond.
- C'est pourquoi, il existe une forme révisée du simplexe, qui ne nécessite que la mémorisation de l'inverse de la base et les données du PLS original.

Dans [Fich], nous trouvons le détail de cette forme révisée, qui permet donc de gagner de la place mémoire et du temps de calcul.

#### A1.2. DECOMPOSITION LU

Le problème qui nous occupe est de calculer l'inverse d'une base  $B$  donnée.

Supposons  $B = (B_{ij})$  d'ordre  $n$ .

Calculer l'inverse  $B^{-1}$  de  $B$  revient à résoudre l'équation

$$B X = C \quad \text{où } C = I$$

Pour cela, on utilise l'algorithme de Gauss de résolution de systèmes d'équations linéaires.

Le détail de cet algorithme se trouve dans [KORG].

On peut alors montrer que toute matrice B peut se décomposer en un produit de 2 matrices:

$$B = L * U$$

où L est une matrice infratriangulaire,

U est une matrice supratrigulaire.

La résolution de tels systèmes triangulaires est très facile et de plus elle est assez précise et tient compte des matrices creuses.

### A1.3. LA REGRESSION LINEAIRE

La méthode de régression linéaire que nous avons considérée est la méthode des moindres carrés.

On peut l'expliquer brièvement de la façon suivante:

Soient n points  $(X_i, Y_i)$ ;  $i = 1, \dots, n$ .

La méthode des moindres carrés va nous fournir la droite qui se rapproche le plus de la courbe formée par les n points. La figure 7 ci-dessous permet de visualiser cette droite.

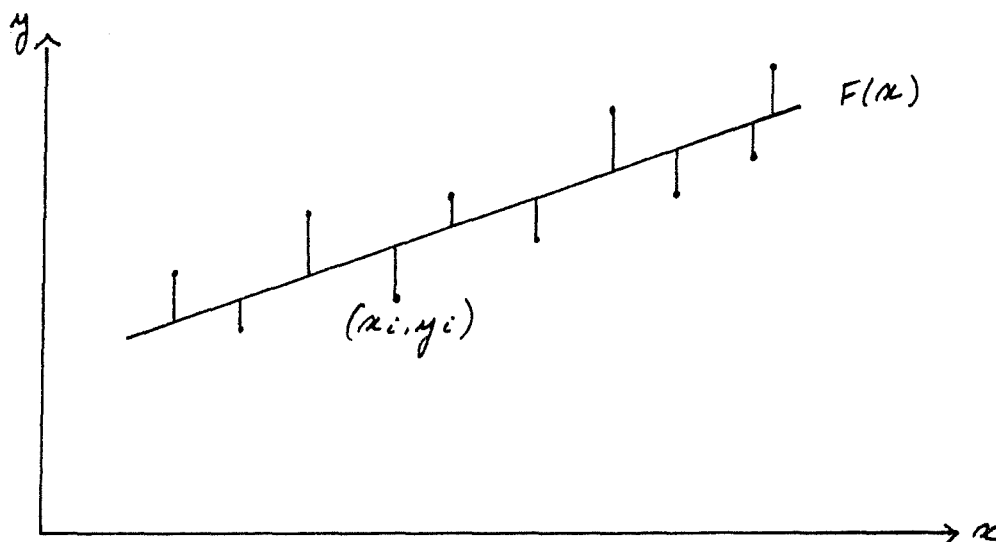


figure 7. La droite de régression.

La droite recherchée est donc telle que

$$\sum_{i=1}^n [Y_i - F(X_i)]^2 \text{ soit minimale.}$$

Appelons la droite régression  $F(X) = aX + b$ ,

on doit donc trouver  $a$  et  $b$  tels que

$$\sum_{i=1}^n [Y_i - (aX_i + b)]^2 \text{ soit minimale,}$$

$i = 1$

$a$  et  $b$  sont alors donnés par les formules suivantes:

$$a = \frac{n \left( \sum_{i=1}^n X_i * Y_i \right) - \left( \sum_{i=1}^n X_i \right) * \left( \sum_{i=1}^n Y_i \right)}{n \left( \sum_{i=1}^n X_i^2 \right) - \left( \sum_{i=1}^n X_i \right)^2}$$

$$b = \frac{\sum_{i=1}^n Y_i}{n} - a * \frac{\sum_{i=1}^n X_i}{n}$$

## ANNEXE 2

Dans cette deuxième annexe, nous trouverons les programmes et les panels que nous avons réalisés.

Nous y avons également mis la description de la base de données que nous avons créée.

LA BASE DE DONNEES

BAS	PLS PROGRAMME LINEAIRE STANDARD	T T001
REC	PLS-ACT ACTION	T T001
FLD	PLS-ACT.PLS# NUMERO PLS	T T001
FLD	PLS-ACT.PLS-ACT# NUMERO ACTION	T T001
FLD	PLS-ACT.PLS-ACT#L LIBELLE ACTION	T T001
REC	PLS-CRB COURBE	T T001
FLD	PLS-CRB.PLS# NUMERO DE PLS	T T001
FLD	PLS-CRB.PLS-ACT# NUMERO ACTION	T T001
FLD	PLS-CRB.PLS-CRB# NUMERO COURBE	T T001
FLD	PLS-CRB.PLS-CRB-TRA# NUMERO TRANCHE	T T001
FLD	PLS-CRB.PLS-CRB#L LIBELLE COURBE	T T001
FLD	PLS-CRB.PLS-CRB-MOE#V VALEUR MOYENNE TRANCHE	T T001
FLD	PLS-CRB.PLS-CRB-ACQ#V VALEUR ACQUIS	T T001
FLD	PLS-CRB.PLS-CRB-COU#A NOM DE LA COUCHE	T T001
REC	PLS-BAS INVERSE DE LA BASE	T T001
FLD	PLS-BAS.PLS# NUMERO PLI	T T001
FLD	PLS-BAS.PLS-CLN# NUMERO COLONNE	T T001
FLD	PLS-BAS.PLS-LGN# NUMERO LIGNE	T T001
FLD	PLS-BAS.PLS-BAS#V VALEUR ELEMENT DE BASE	T T001
FLD	PLS-BAS.PLS-BAS-IND#V VALEUR INDICE BASE CORRESPONDANT	T T001
FLD	PLS-BAS.PLS-BAS-FILLER FILLER	T T001
REC	PLS-CHB COUTS HORS BASE	T T001
FLD	PLS-CHB.PLS# NUMERO PLS	T T001
FLD	PLS-CHB.PLS-CLN# NUMERO COLONNE	T T001

FLD	PLS-CHB.PLS-CHB#V VALEUR COUT HORS BASE	T T001
FLD	PLS-CHB.PLS-CHB-IND#V VALEUR INDICE HORS BASE CORRESPOND.	T T001
FLD	PLS-CHB.PLS-CHB-FILLER FILLER	T T001
REC	PLS-PSS VECTEUR COLONNE AIS	T T001
FLD	PLS-PSS.PLS# NUMERO PLS	T T001
FLD	PLS-PSS.PLS-LGN# NUMERO LIGNE	T T001
FLD	PLS-PSS.PLS-PS#V VALEUR DES AIS	T T001
FLD	PLS-PSS.PLS-PS-ETR#V VALEUR INDICE ENTRE EN BASE	T T001
FLD	PLS-PSS.PLS-PS-FILLER FILLER	T T001
REC	PLS-XBN VECTEUR VARIABLES XB XN	T T001
FLD	PLS-XBN.PLS# NUMERO PLS	T T001
FLD	PLS-XBN.PLS-CLN# NUMERO COLONNE	T T001
FLD	PLS-XBN.PLS-XB#V VALEUR VARIABLE CORRESPONDANTE	T T001
FLD	PLS-XBN.PLS-XB-FILLER FILLER	T T001
REC	PLS-COE COEFFICIENTS	T T001
FLD	PLS-COE.PLS# NUMERO PLS	T T001
FLD	PLS-COE.PLS-CLN# NUMERO COLONNE	T T001
FLD	PLS-COE.PLS-LGN# NUMERO LIGNE	T T001
FLD	PLS-COE.PLS-COE#V VALEUR DU COEFFICIENT	T T001
FLD	PLS-COE.PLS-COE-FILLER FILLER	T T001
REC	PLS-CRL COUTS RELATIFS	T T001
FLD	PLS-CRL.PLS# NUMERO DU PLI	T T001
FLD	PLS-CRL.PLS-CLN# NUMERO COLONNE	T T001
FLD	PLS-CRL.PLS-CRL#V VALEUR COUT RELATIF	T T001

FLD	PLS-CRL.PLS-CRL-FILLER FILLER	T T001
REC	PLS-PLS PROGRAMME LINEAIRE STANDARD	T T001
FLD	PLS-PLS.PLS#	T T001
FLD	NUMERO DU PLS	
FLD	PLS-PLS.PLS-VAR#N	T T001
FLD	NOMBRE DE VARIABLES	
FLD	PLS-PLS.PLS-CTE#N	T T001
FLD	NOMBRE DE CONTRAINTES	
FLD	PLS-PLS.PLS#L	T T001
FLD	LIBELLE DU PLI	
FLD	PLS-PLS.PLS-ZO#V	T T001
FLD	VALEUR FONCTION ECONOMIQUE	
FLD	PLS-PLS.PLS-OPT#V	T T001
FLD	VALEUR OPTIMUM	
FLD	PLS-PLS.PLS-PLS-FILLER FILLER	T T001
REC	PLS-TIN TERMES INDEPENDANTS	T T001
FLD	PLS-TIN.PLS#	T T001
FLD	NUMERO DU PLS	
FLD	PLS-TIN.PLS-LGN#	T T001
FLD	NUMERO LIGNE	
FLD	PLS-TIN.PLS-TI#V	T T001
FLD	VALEUR TERME INDEPENDANT	
FLD	PLS-TIN.PLS-TI-SGN-CTE	T T001
FLD	SIGNE DE LA CONTRAINTE	
FLD	PLS-TIN.PLS-TIN-FILLER FILLER	T T001

LES PROGRAMMES

IDENTIFICATION: ME000

VERSION: 001 STATUS: TEST

## PROGRAM ME000

Created 11/03/87 By ETUD04  
Last Modified 12/07/87 at 14:00 By ETUD04  
Last Compiled 12/11/87 at 12:20

Run Status: PRIVATE  
Short Description: MENU PRINCIPAL PGM LIN  
Language: IDEAL Target Date \_\_\_ \_\_\_ \_\_\_ Actual Date \_\_\_ \_\_\_ \_\_\_

## Description:

MENU GENERAL DES PROGRAMMES RELATIFS AU PROBLEME DE  
PROGRAMMATION LINEAIRE.  
REGRESSION LINEAIRE POUR LE PVC  
ALGORITHME DU SIMPLEXE  
CREATION,SUPPRESSION,MODIFICATION,EXECUTION D'UN PLS

RESOURCES:

ME000

VERSION: 001 STATUS: TEST

DATAVIEW	VER	PANEL	VER	REPORT	VER	PROGRAM	VER	SYS
		PN0001	0001			ME001	0001	
						ME002	0001	

PARAMETER: ME000 VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	U	M	COMMENT/REDEF/DEP ON/COPY
-----	-------	------------	---	---	-------	-------	---	---	---------------------------

NO PARAMETERS FOR THIS PROGRAM

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND. NAME, F=FLAG,  
V=VARIABLE, D=DATE  
I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
U (PARAMETER UPDATING): U=UPDATABLE, I=INPUT ONLY  
M (PARAMETER MATCHING): BLANK OR D=DYNAMIC, I=IDENTICAL  
REDEF=REDEFINITION, DEP ON= DEPENDING ON

WORKING DATA: ME000 VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	VAL/COMMENT/REDEF/DEP ON/COPY
000100	1	W-PVC		F			

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND.NAME, F=FLAG,  
V=VARIABLE, D=DATE

I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY

CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS

REDEF= REDEFINITION, DEP ON= DEPENDING ON

PROCEDURE:

ME000

VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
000100 <<ME000>> PROCEDURE  
000200     DO INI-PN0001  
000300     LOOP  
000400         DO CHOIX  
000401     UNTIL $PF6  
000500     ENDL00P  
000600 ENDPROC  
000700 <<INI-PN0001>>PROCEDURE  
000800     REFRESH PN0001  
001000     TRANSMIT PN0001  
001100 ENDPROC  
001200 <<CHOIX>> PROCEDURE  
001300     SELECT PN0001.CHX  
001400         WHEN 1 SET W-PVC = TRUE  
001500             CALL ME001 USING INPUT W-PVC  
001501             RELEASE PROGRAM ME001  
001600         WHEN 2 SET W-PVC = FALSE  
001700             CALL ME002 USING INPUT W-PVC  
001701             RELEASE PROGRAM ME002  
001800     ENDSEL  
001900 ENDPROC
```

IDENTIFICATION: ME001

VERSION: 001 STATUS: TEST

PROGRAM ME001

Created 11/06/87 By ETUD04  
Last Modified 01/07/88 at 11:47 By ETUD04  
Last Compiled 01/07/88 at 16:32

Run Status: PRIVATE  
Short Description: menu regression  
Language: IDEAL Target Date \_\_\_ \_\_\_ \_\_\_ Actual Date \_\_\_ \_\_\_ \_\_\_

Description:

---

---

---

---

---



PARAMETER: ME001 VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	U	M	COMMENT/REDEF/DEP ON/COPY
000100	1	P-PVC		F				I D	

M

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND. NAME, F=FLAG,  
V=VARIABLE, D=DATE  
I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
U (PARAMETER UPDATING): U=UPDATABLE, I=INPUT ONLY  
M (PARAMETER MATCHING): BLANK OR D=DYNAMIC, I=IDENTICAL  
REDEF=REDEFINITION, DEP ON= DEPENDING ON

WORKING DATA: ME001                    VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	VAL/COMMENT/REDEF/DEP	ON/COPY
000300	1	W-PLS-MAX#	N		3			
000301	1	W-CHX-PLS#	X		3			
000400	1	W-OCC	N		3			
000500	1	W-CPT	N		3			
000501	1	W-CPT-2	N		2			
000502	1	W-CPT-3	N		2			
000503	1	W-CPT-4	N		2			
000504	1	W-ACQ#V						
000600	2	ACQ#V	N		3	20		
000601	1	W-RDT#V						
000700	2	RDT#V	N		7.8	20		
000701	1	W-MCD#V						
000800	2	MCD#V	N		7.8	20		
000801	1	W-RDT-MCD						
000900	2	RDT-MCD	N		7.8	20		
000901	1	W-ACQ						
001000	2	ACQ	N		3	4		
001001	1	W-REG-A#V						
001100	2	REG-A#V	N		7.8	3		
001101	1	W-REG-B#V						
001200	2	REG-B#V	N		7.8	3		
001300	1	W-TRA#N	N		1			
001301	1	W-REG-MOE						
001400	2	REG-MOE	N		7.8	3		
001500	1	W-CRB-MAX#	N		4			
001600	1	W-PLS#	X		3			
001601	1	W-PTS-TRA#N						
001700	2	PTS-TRA#N	N		3	3 000		
001800	1	W-RDT	F					
001900	1	W-RDT-TRA	N		7.8			
002000	1	W-COU#N	N		8	12		
002100	1	W-CRL-CLN#	N		4			
002200	1	W-TI-LGN#	N		4			
002300	1	W-COU-CLI#N	N		8			
002400	1	W-ACV-CLI#N	N		8			
002500	1	W-PVC	F					
002600	1	W-PLS-ZO#V	N		7.8			
002700	1	W-PLS-OPT#V	X		3			
002701	1	W-PLS-CTE#N	N		4			
002702	1	W-PLS-VAR#N	N		4			
002800	1	W-ITE#N	N		5			
002900	1	W-OK	F					
003000	1	W-I	N		2			
003100	1	W-IND	N		4	12		
003200	1	W-COU-TRA#N	N		1	12		
003300	1	W-CPT-4#V	N		2	12		
003400	1	W-COU	X		3	12		
003500	1	W-J	N		1			

0

WORKING DATA: ME001                   VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	VAL/COMMENT/REDEF/DEP ON/COPY
003600	1	W-PRT	N		3.2		
003700	1	W-K	N		4		
003800	1	W-PTS#N	N		3		

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND.NAME, F=FLAG,  
V=VARIABLE, D=DATE  
I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
REDEF= REDEFINITION, DEP ON= DEPENDING ON

PROCEDURE: ME001 VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
000001 <<PN0222>> PROCEDURE  
000002     LOOP  
000003         REFRESH PN0222  
000004         TRANSMIT PN0222  
000005     UNTIL $PF6  
000006         DO CHX-PLS  
000007     ENDLOOP  
000008 ENDPROC  
000009 <<CHX-PLS>> PROCEDURE  
000010     SELECT PN0222.CHX  
000011         WHEN 1 DO PN0010  
000012         WHEN 2 DO PN0026  
000013         WHEN 3 DO PN0026  
000014     ENDSEL  
000015 ENDPROC  
000017 <<PN0026>> PROCEDURE  
000018     REFRESH PN0026  
000019     IF PN0222.CHX = 2  
000020     THEN  
000021         SET PN0026.TYP = 'EXECUTION'  
000022     ELSE  
000023         IF PN0222.CHX = 3  
000024         THEN  
000025             SET PN0026.TYP = 'SUPPRESSION'  
000026         ELSE  
000027             SET PN0026.TYP = 'MODIFICATION'  
000028         ENDIF  
000029     ENDIF  
000030     LOOP  
000031         TRANSMIT PN0026  
000032         IF $PF2  
000033             DO PN0022  
000034         ENDIF  
000035         FOR EACH PLS-PLS-R  
000036             WHERE PLS# = PN0026.PLS#  
000037             LOOP VARYING W-CPT FROM 1 THRU 8  
000038                 SET PN0026.PLS#L (W-CPT) = $SUBSTR (PLS-PLS-R.PLS#L,  
000039                     START = 50 * (W-CPT - 1) + 1)  
000040             ENDLOOP  
000041             TRANSMIT PN0026  
000042         ENDFOR  
000043     UNTIL $PF6  
000044     ENDLOOP  
000045     SET W-CHX-PLS# = PN0026.PLS#  
000046     FOR EACH PLS-PLS-R  
000047     WHERE PLS# = W-CHX-PLS#  
000048         SET W-PLS-VAR#N = PLS-PLS-R.PLS-VAR#N  
000049         SET W-PLS-CTE#N = PLS-PLS-R.PLS-CTE#N
```

PROCEDURE:

ME001

VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
000050      SET W-PLS-ZO#V = PLS-PLS-R.PLS-ZO#V  
000051      SET W-PLS-OPT#V = PLS-PLS-R.PLS-OPT#V  
000052      SET W-ITE#N = 0  
000053      ENDFOR  
000054      SELECT PN0026.CON  
000055          WHEN 'N' DO PN0022  
000056          WHEN 'O'  
000057              IF PN0222.CHX = 2  
000058                  SET W-PVC = FALSE  
000059                  CALL UT002 USING INPUT W-CHX-PLS#,W-PLS-VAR#N,W-PLS-CTE#N,  
000060                      W-PVC  
000061                      UPDATE W-PLS-ZO#V,W-PLS-OPT#V,W-ITE#N  
000062                      : EXECUTION PLS  
000063                  RELEASE PROGRAM UT002  
000064              ELSE  
000065                  IF PN0222.CHX = 3  
000066                      CALL UT003 USING INPUT W-CHX-PLS#  
000067                      : SUPPRESSION PLS  
000068                      RELEASE PROGRAM UT003  
000069                  :ELSE  
000070                  : CALL UT004 USING INPUT W-CHX-PLS#,W-PLS-VAR#N,W-PLS-CTE#  
000071                  : UPDATE W-PLS-ZO#V,W-PLS-OPT#V  
000072                  : : MODIFICATION PLS  
000073                  : RELEASE PROGRAM UT004  
000074                  ENDIF  
000075              ENDIF  
000076          ENDSEL  
000077      ENDPROC  
000078      <<PN0022>> PROCEDURE  
000079          FOR EACH PLS-PLS-R  
000080              ORDERED BY ASCENDING PLS#  
000081                  LOOP  
000082                      REFRESH PN0022  
000083                      SET PN0022.PLS# = PLS-PLS-R.PLS#  
000084                      LOOP VARYING W-CPT FROM 1 THRU 8  
000085                          SET PN0022.PLS#L (W-CPT) = $SUBSTR (PLS-PLS-R.PLS#L,  
000086                              START = 50 * (W-CPT - 1) + 1)  
000087                      ENDLLOOP  
000088                      TRANSMIT PN0022  
000089                      UNTIL $PF6  
000090                  ENDLOOP  
000091          ENDFOR  
000092      ENDPROC  
000093      <<PN0010>> PROCEDURE  
000094          REFRESH PN0010  
000095          LOOP  
000096              TRANSMIT PN0010  
000097          UNTIL $PF6
```

PROCEDURE: ME001 VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
000098   ENDLOOP  
000099   SET W-COU#N (1) = PN0010.CN1#N  
000100   SET W-COU#N (2) = PN0010.CN1#N  
000101   SET W-COU#N (3) = PN0010.CN2#N  
000102   SET W-COU#N (4) = PN0010.CN3#N  
000103   SET W-COU#N (5) = PN0010.CN4#N  
000104   SET W-COU#N (6) = PN0010.CN5#N  
000105   SET W-COU#N (7) = PN0010.AB2#N  
000106   SET W-COU#N (8) = PN0010.AB3#N  
000107   SET W-COU#N (9) = PN0010.AB4#N  
000108   SET W-COU#N (10) = PN0010.AB5#N  
000109   SET W-COU#N (11) = PN0010.ELM#N  
000110   SET W-COU#N (12) = PN0010.PRO#N  
000111   DO PN0011  
000112   ENDPROC  
000113   <<PN0011>> PROCEDURE  
000114     REFRESH PN0011  
000115     FOR FIRST PLS-PLS-K  
000116     ORDERED BY DESCENDING PLS#  
000117       SET W-PLS-MAX# = $NUMBER (PLS-PLS-K.PLS#) + 1  
000118     WHEN NONE  
000119       SET W-PLS-MAX# = 1  
000120     SET W-PLS# = PLS-PLS-K.PLS#  
000121     ENDFOR  
000122     SET PN0011.PR# = W-PLS-MAX#  
000123     LOOP  
000124       TRANSMIT PN0011  
000125     UNTIL $PF6  
000126     ENDLOOP  
000127     DO CHARGEMENT-PLS  
000128     DO PN0012  
000129   ENDPROC  
000130   <<CHARGEMENT-PLS>> PROCEDURE  
000131     FOR NEW PLS-PLS  
000132       SET PLS-PLS.PLS# = $EDIT (PN0011.PR#, PIC = '999')  
000133       LOOP VARYING W-CPT FROM 8 DOWN THRU 1  
000134         SET PLS-PLS.PLS#L = $STRING(PN0011.GRP(W-CPT), PLS-PLS.PLS#L)  
000135       ENDLOOP  
000136       SET W-PLS# = PLS-PLS.PLS#  
000137     ENDFOR  
000138   ENDPROC  
000139   <<PN0012>>PROCEDURE  
000140     SET W-CPT-4 = 1  
000141     SET W-CRL-CLN# = 0  
000142     SET W-TI-LGN# = 2  
000143     LOOP VARYING W-CPT-2 FROM 1 THRU PN0011.ACT#N  
000144     REFRESH PN0012  
000145     SET PN0012.DON# = W-CPT-2
```

PROCEDURE:

ME001                   VERSION: 001 STATUS: TEST

```
-----  
SEQ    DATA  
-----  
000146        LOOP  
000147            TRANSMIT PNO012  
000148        UNTIL $PF6  
000149        ENDLOOP  
000150        DO CHGT-ACTION  
000151        LOOP VARYING W-CPT-3 FROM 1 THRU PNO012.COU#N  
000152            REFRESH PNO12B  
000153            SET PNO12B.ACT# = PNO012.ACT#  
000154            SET PNO12B.DON# = W-CPT-3  
000155        LOOP  
000156            TRANSMIT PNO12B  
000157            UNTIL $PF6  
000158        ENDLOOP  
000159        DO PNO013  
000160        DO SAUVE  
000161        DO APPEL-REG  
000162        DO CHGT-CRL-TI-COE  
000163        IF (PNO012.ACT-TYP = 'CS') OR (PNO012.ACT-TYP = 'RL')  
000164            IF PNO012.CLI-TYP = 'S'  
000165                SET W-PRT = PNO12B.SPR#T  
000166            ELSE  
000167                IF PNO012.CLI-TYP = 'M'  
000168                    SET W-PRT = 1 - PNO12B.SPR#T  
000169            ELSE  
000170                SET W-PRT = 1  
000171            ENDIF  
000172        ENDIF  
000173        DO CTE-ACT  
000174        ENDIF  
000175        DO CHGT-PLS  
000176        ENDLOOP  
000177        ENDLOOP  
000178        DO COE-ZERO  
000179        ENDPROC  
000180        <<CHGT-ACTION>>PROCEDURE  
000181            FOR NEW PLS-ACT  
000182                SET PLS-ACT.PLS# = W-PLS#  
000183                SET PLS-ACT.PLS-ACT# = $EDIT (PNO012.ACT#,PIC = '99999')  
000184                SET PLS-ACT.PLS-ACT#L = PNO012.ACT#L  
000185            ENDFOR  
000186        ENDPROC  
000187        <<PNO013>>PROCEDURE  
000188            LOOP  
000189                REFRESH PNO013  
000190                SET PNO013.ACT#L = PNO012.ACT#L  
000191                SET PNO013.COU = PNO12B.COU#A  
000192                TRANSMIT PNO013  
000193            UNTIL $PF6
```

PROCEDURE: ME001 VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
000194    ENDLOOP  
000195  ENDPROC  
000196 <<SAUVE>>PROCEDURE  
000197    SET W-OCC = $PANEL-GROUP-OCCURS(PN0013)  
000198    IF PN012B.PTS#N < W-OCC  
000199      THEN  
000200        LOOP  
000201          VARYING W-CPT FROM 1 THRU PN012B.PTS#N  
000202            SET W-ACQ#V.ACQ#V (W-CPT) = PN0013.ACQ#V (W-CPT)  
000203            SET W-RDT#V.RDT#V (W-CPT) = PN0013.RDT#V(W-CPT)  
000204            SET W-MCD#V.MCD#V (W-CPT) = PN0013.MCD#V (W-CPT)  
000205          ENDLOOP  
000206        ELSE  
000207          LOOP  
000208            VARYING W-CPT FROM 1 THRU W-OCC  
000209              SET W-ACQ#V.ACQ#V(W-CPT) = PN0013.ACQ#V(W-CPT)  
000210              SET W-RDT#V.RDT#V(W-CPT) = PN0013.RDT#V(W-CPT)  
000211              SET W-MCD#V.MCD#V(W-CPT) = PN0013.MCD#V(W-CPT)  
000212            ENDLOOP  
000213          DO PN0013  
000214            LOOP VARYING W-CPT FROM (W-OCC + 1) THRU PN012B.PTS#N  
000215              SET W-ACQ#V.ACQ#V (W-CPT) = PN0013.ACQ#V(W-CPT)  
000216              SET W-RDT#V.RDT#V (W-CPT) = PN0013.RDT#V(W-CPT)  
000217              SET W-MCD#V.MCD#V (W-CPT) = PN0013.MCD#V (W-CPT)  
000218            ENDLOOP  
000219          ENDIF  
000220        ENDPROC  
000221 <<APPEL-REG>>PROCEDURE  
000222    SET W-RDT = TRUE  
000223    SET W-PTS#N = PN012B.PTS#N  
000224    CALL UT001 USING UPDATE W-ACQ,W-REG-A#V,W-REG-B#V,W-TRA#N,W-REG-MOE,  
000225                          W-PTS-TRA#N  
000226                          INPUT W-RDT#V,W-ACQ#V,W-PTS#N,W-PLS#,W-RDT  
000227    RELEASE PROGRAM UT001  
000228    FOR FIRST PLS-CRB-K  
000229    ORDERED BY DESCENDING PLS-CRB#  
000230      SET W-CRB-MAX# = $NUMBER(PLS-CRB-K.PLS-CRB#) + 1  
000231    WHEN NONE  
000232      SET W-CRB-MAX# = 1  
000233    ENDFOR  
000234    FOR NEW PLS-CRB  
000235      LOOP VARYING W-CPT FROM 1 THRU W-TRA#N  
000236        SET PLS-CRB.PLS# = $EDIT (W-PLS-MAX#,PIC = '999')  
000237        SET PLS-CRB.PLS-ACT# = $EDIT (PN0012.ACT#,PIC = '99999')  
000238        SET PLS-CRB.PLS-CRB#L = 'RENDEMENT'  
000239        SET PLS-CRB.PLS-CRB# = $EDIT (W-CRB-MAX#,PIC = '9999')  
000240        SET PLS-CRB.PLS-CRB-TRA# = $EDIT (W-CPT,PIC = '9999')  
000241        SET PLS-CRB.PLS-CRB-MOE#V = W-REG-MOE.REG-MOE (W-CPT)
```

PROCEDURE:

ME001                   VERSION: 001 STATUS: TEST

```

-----
SEQ     DATA
-----
000242         SET PLS-CRB.PLS-CRB-ACQ#V = $EDIT (W-ACQ.ACQ (W-CPT),PIC = '999')
000243         SET PLS-CRB.PLS-CRB-COU#A = PNO12B.COU#A
000244         ENDLOOP
000245         ENDFOR
000246         SET W-RDT = FALSE
000247         FOR NEW PLS-CRB
000248             CALL UT001 USING UPDATE W-ACQ,W-REG-A#V,W-REG-B#V,W-TRA#N,
000249                             W-REG-MOE,W-PTS-TRA#N
000250                             INPUT W-MCD#V,W-ACQ#V,W-PTS#N,W-PLS#,
000251                             W-RDT
000252         RELEASE PROGRAM UT001
000253         LOOP VARYING W-CPT FROM 1 THRU W-TRA#N
000254             SET PLS-CRB.PLS# = $EDIT (W-PLS-MAX#,PIC = '999')
000255             SET PLS-CRB.PLS-ACT# = $EDIT (PNO012.ACT#,PIC = '99999')
000256             SET PLS-CRB.PLS-CRB#L = 'MOYENNE COMMANDEE'
000257             SET W-K = W-CRB-MAX# + 1
000258             SET PLS-CRB.PLS-CRB# = $EDIT (W-K,PIC = '9999')
000259             SET PLS-CRB.PLS-CRB-TRA# = $EDIT (W-CPT,PIC = '9999')
000260             SET PLS-CRB.PLS-CRB-MOE#V = REG-MOE (W-CPT)
000261             SET PLS-CRB.PLS-CRB-ACQ#V =
000262                             $EDIT (W-ACQ.ACQ (W-CPT),PIC = '999')
000263             SET PLS-CRB.PLS-CRB-COU#A = PNO12B.COU#A
000264             ENDLOOP
000265         ENDFOR
000266         CALL UT001 USING UPDATE W-ACQ,W-REG-A#V,W-REG-B#V,W-TRA#N,W-REG-MOE,
000267                             W-PTS-TRA#N
000268                             INPUT W-RDT-MCD,W-ACQ#V,W-PTS#N,W-PLS#,W-RDT
000269         RELEASE PROGRAM UT001
000270     ENDPROC
004501     <<CHGT-CRL-TI-COE>> PROCEDURE
004502     LOOP VARYING W-CPT FROM 1 THRU W-TRA#N
004503         FOR NEW PLS-CRL
004504             SET W-CRL-CLN# = W-CRL-CLN# + 1
004505             SET PLS-CRL.PLS# = $EDIT (W-PLS-MAX#,PIC = '999')
004506             SET PLS-CRL.PLS-CLN# = $EDIT (W-CRL-CLN#,PIC = '9999')
004507             FOR EACH PLS-CRB-R
004508                 WHERE (PLS# = W-PLS#) AND
004509                     (PLS-ACT# = $EDIT (PNO012.ACT#,PIC = '99999')) AND
004510                     (PLS-CRB# = $EDIT (W-CRB-MAX#,PIC = '9999')) AND
004511                     (PLS-CRB-TRA# = $EDIT (W-CPT,PIC = '9999'))
004512                 SET W-RDT-TRA = PLS-CRB-R.PLS-CRB-MOE#V
004513             ENDFOR
004514             FOR EACH PLS-CRB-R
004515                 WHERE (PLS# = $EDIT (W-PLS-MAX#,PIC = '999')) AND
004516                     (PLS-ACT# = $EDIT (PNO012.ACT#,PIC = '99999')) AND
004517                     (PLS-CRB# = $EDIT (W-CRB-MAX#,PIC = '9999')) AND
004518                     (PLS-CRB-TRA# = $EDIT (W-CPT,PIC = '9999'))
004519                 SET PLS-CRL.PLS-CRL#V = W-RDT-TRA * PLS-CRB.PLS-CRB-MOE#V

```

PROCEDURE:

ME001

VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
004520                                     - PN0012.ACT#V
004521          ENDFOR
004522          ENDFOR
004523          FOR NEW PLS-COE
004524              SET PLS-COE.PLS# = $EDIT (W-PLS-MAX#,PIC = '999')
004525              SET PLS-COE.PLS-CLN# = $EDIT (W-CRL-CLN#,PIC = '9999')
004526              SET PLS-COE.PLS-LGN# = $EDIT (W-TI-LGN#,PIC = '9999')
004527              SET PLS-COE.PLS-COE#V = 1
004528          ENDFOR
004529          IF (PN012B.COU#A = 'CN5') OR (PN012B.COU#A = 'AB5')
004530              FOR NEW PLS-COE
004531                  SET PLS-COE.PLS# = W-PLS#
004532                  SET PLS-COE.PLS-CLN# = $EDIT (W-CRL-CLN#,PIC = '9999')
004533                  SET PLS-COE.PLS-LGN# = $EDIT (1,PIC = '9999')
004534                  SET PLS-COE.PLS-COE#V = PN012B.PRT-RPS
004535              ENDFOR
004536          ENDFIF
004537          IF (PN0012.ACT-TYP = 'CP') OR (PN0012.ACT-TYP = 'PR')
004538              SET W-IND (W-CPT-4) = W-CRL-CLN#
004539              SET W-COU (W-CPT-3) = PN0013.COU
004540              SET W-COU-TRA#N (W-CPT-3) = W-TRA#N
004541              SET W-CPT-4#V (W-CPT-3) = W-CPT-4
004542              SET W-CPT-4 = W-CPT-4 + 1
004543          ENDFIF
004544          ENDLOOP
004545          FOR NEW PLS-TIN
004546              DO NOMBRE-CLI
004547                  SET PLS-TIN.PLS# = W-PLS#
004548                  SET PLS-TIN.PLS-LGN# = $EDIT (W-TI-LGN#,PIC = '9999')
004549                  SET PLS-TIN.PLS-TI#V = W-COU-CLI#N
004550                  SET PLS-TIN.PLS-TI-SGN-CTE = '<='
004551              ENDFOR
004552              SET W-TI-LGN# = W-TI-LGN# + 1
004553              IF (PN0012.DON# = 1) AND (PN012B.DON# = 1)
004554                  FOR NEW PLS-TIN
004555                      SET PLS-TIN.PLS# = W-PLS#
004556                      SET PLS-TIN.PLS-LGN# = $EDIT (1,PIC = '9999')
004557                      LOOP VARYING W-CPT-4 FROM 1 THRU 10
004558                          SET W-ACV-CLI#N = W-ACV-CLI#N + W-COU#N (W-CPT-4)
004559                      ENDFOR
004560                      SET PLS-TIN.PLS-TI#V = W-ACV-CLI#N * (PN0010.CRO#T - 1) -
004561                          PN0012.CLI-NOU#N + W-COU#N (6) + W-COU#N (10)
004562                      SET PLS-TIN.PLS-TI-SGN-CTE = '>='
004563                  ENDFOR
004564              ELSE
004565                  IF PN012B.DON# = 1
004566                      FOR EACH PLS-TIN
004567                          WHERE (PLS# = W-PLS#) AND

```

IDENTIFICATION: UT012

VERSION: 001 STATUS: TEST

PROGRAM UT012

Created 01/04/88 By ETUD04  
Last Modified 01/06/88 at 11:12 By ETUD04  
Last Compiled 01/06/88 at 11:25

Run Status: PRIVATE  
Short Description: met. primale phase 2.2  
Language: IDEAL Target Date \_\_\_ \_\_\_ Actual Date \_\_\_ \_\_\_

Description:

---

---

---

---

RESOURCES:

ME002                    VERSION: 001 STATUS: TEST

DATAVIEW	VER	PANEL	VER	REPORT	VER	PROGRAM	VER	SYS
PLS-COE	0001	PN0021	0001	_____	_____	UT001	0001	_____
PLS-CRL	0001	PN0022	0001	_____	_____	UT002	0001	_____
PLS-PLS	0001	PN0023	0001	_____	_____	UT003	0001	_____
PLS-PLS-K	0001	PN0024	0001	_____	_____	_____	_____	_____
PLS-PLS-R	0001	PN0025	0001	_____	_____	_____	_____	_____
PLS-TIN	0001	PN0026	0001	_____	_____	_____	_____	_____

PARAMETER:

ME002

VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	U	M	COMMENT/REDEF/DEP ON/COPY
000100	1	P-PVC		F				I D	

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND. NAME, F=FLAG,  
V=VARIABLE, D=DATE  
I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
U (PARAMETER UPDATING): U=UPDATABLE, I=INPUT ONLY  
M (PARAMETER MATCHING): BLANK OR D=DYNAMIC, I=IDENTICAL  
REDEF=REDEFINITION, DEP ON= DEPENDING ON

WORKING DATA: ME002                    VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	VAL/COMMENT/REDEF/DEP	ON/COPY
000100	1	W-PLS-MAX#	N			3		
000200	1	W-OCC	N			4		
000300	1	W-CPT	N			4		
000400	1	W-ITE	N			4		
000500	1	W-RET	N			4		
C 000600	1	W-MAX	N			4		
000700	1	W-MIN	N			4		
000800	1	W-I#V	N			4		
000900	1	W-CHX-PLS#	X			3		
001000	1	W-PLS-VAR#N	N			4		
001100	1	W-PLS-CTE#N	N			4		
001200	1	W-PLS-ZO#V	N		7.8			
001300	1	W-PLS-OPT#V	X			3		
001400	1	W-ITE#	N			5		
001500	1	W-PVC	F					
001600	1	W-PLS#	X			3		
001700	1	W-I	N			4		

LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND.NAME, F=FLAG,  
                   V=VARIABLE, D=DATE  
 I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
 CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
 REDEF= REDEFINITION, DEP ON= DEPENDING ON

PROCEDURE: ME002                   VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
000100  <<PN0021>> PROCEDURE
000101      LOOP
000200      REFRESH PN0021
000300      TRANSMIT PN0021
000400      UNTIL $PF6
000500      DO CHX-PLS
000600      ENDLOOP
000700  ENDPROC
C      000800  <<CHX-PLS>> PROCEDURE
000900      SELECT PN0021.CHX
001000      WHEN 1 DO PN0023
001100      WHEN 2 DO PN0026
001200      WHEN 3 DO PN0026
001300      WHEN 4 DO PN0026
001400      ENDSEL
001500  ENDPROC
001600  <<PN0023>> PROCEDURE
001700      REFRESH PN0023
001800      DO RECH-NO-PLS
001900      LOOP
001901          SET PN0023.PLS# = $EDIT (W-PLS-MAX#,PIC = '999')
002000          TRANSMIT PN0023
002100          UNTIL $PF6
002200          ENDLOOP
002300      FOR NEW PLS-PLS
002301          SET PLS-PLS.PLS# = W-PLS#
002302          LOOP VARYING W-CPT FROM 8 DOWN THRU 1
002303              SET PLS-PLS.PLS#L = $STRING (PN0023.GRP (W-CPT),PLS-PLS.PLS#L)
002304          ENDLOOP
002305          SET PLS-PLS.PLS-VAR#N = PN0023.VAR#N
002306          SET PLS-PLS.PLS-CTE#N = PN0023.CTE#N
002307          SET PLS-PLS.PLS-OPT#V = PN0023.OPT#V
002308          SET PLS-PLS.PLS-ZO#V = 0
002309          SET W-PLS-CTE#N = PLS-PLS.PLS-CTE#N
002310          SET W-PLS-VAR#N = PLS-PLS.PLS-VAR#N
002311          SET W-PLS-ZO#V = PLS-PLS.PLS-ZO#V
002312          SET W-PLS-OPT#V = PLS-PLS.PLS-OPT#V
002313      ENDFOR
002314      DO PN0024
002315  ENDPROC
002400  <<RECH-NO-PLS>> PROCEDURE
002500      FOR FIRST PLS-PLS-K
002600      ORDERED BY DESCENDING PLS#
002700          SET W-PLS-MAX# = $NUMBER (PLS-PLS-K.PLS#) + 1
P      002800      WHEN NONE
002900          SET W-PLS-MAX# = 1
003000      ENDFOR
003001      SET W-PLS# = $EDIT (W-PLS-MAX#,PIC = '999')

```

PROCEDURE:

ME002                   VERSION: 001 STATUS: TEST

```
-----  
SEQ    DATA  
-----  
003100  ENDPROC  
003200  <<PN0024>> PROCEDURE  
003300    REFRESH PN0024  
003400    SET PN0024.PLS# = W-PLS#  
003401    SET PN0024.VAR#N = PN0023.VAR#N  
003402    LOOP VARYING W-CPT FROM 1 THRU W-PLS-VAR#N  
003403      SET PN0024.CPT (W-CPT) = W-CPT  
003404    ENDLOOP  
003500    LOOP  
003600      TRANSMIT PN0024  
003700      UNTIL $PF6  
003800    ENDLOOP  
003900    SET W-OCC = $PANEL-GROUP-OCCURS (PN0024)  
004000    IF W-OCC >= PN0024.VAR#N  
004100      LOOP VARYING W-CPT FROM 1 THRU W-PLS-VAR#N  
004200        FOR NEW PLS-CRL  
004300          SET PLS-CRL.PLS-CLN# = $EDIT (W-CPT,PIC = '9999')  
004400          SET PLS-CRL.PLS# = W-PLS#  
004500          SET PLS-CRL.PLS-CRL#V = PN0024.CRL#V (W-CPT)  
004600        ENDFOR  
004700      ENDLOOP  
004800    ELSE  
004900      LOOP VARYING W-CPT FROM 1 THRU W-OCC  
005000        FOR NEW PLS-CRL  
005100          SET PLS-CRL.PLS-CLN# = $EDIT (W-CPT,PIC = '9999')  
005200          SET PLS-CRL.PLS# = W-PLS#  
005300          SET PLS-CRL.PLS-CRL#V = PN0024.CRL#V (W-CPT)  
005400        ENDFOR  
005500      ENDLOOP  
005501      SET W-ITE = 1  
005600      SET W-RET = PN0024.VAR#N - W-OCC  
005601      SET W-MIN = W-OCC * W-ITE  
005602      LOOP  
005700        LOOP  
005800          REFRESH PN0024  
005900          SET PN0024.PLS# = W-PLS#  
006000          SET PN0024.VAR#N = PN0023.VAR#N  
006001          LOOP VARYING W-CPT FROM 1 THRU W-RET  
006002            SET W-I = W-MIN + W-CPT  
006003            SET PN0024.CPT (W-CPT) = W-I  
006004          ENDLOOP  
006100          TRANSMIT PN0024  
006200          UNTIL $PF6  
006300          ENDLOOP  
006301          IF (W-RET - W-OCC) <= 0  
006302            SET W-MAX = W-RET  
006303          ELSE  
006304            SET W-MAX = W-OCC
```

R

PROCEDURE: ME002 VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
006305      ENDIF
006306      SET W-MIN = W-OCC * W-ITE
006307      LOOP VARYING W-CPT FROM 1 THRU W-MAX
006308          FOR NEW PLS-CRL
006309              SET W-I = W-MIN + W-CPT
006310                  SET PLS-CRL.PLS-CLN# = $EDIT (W-I,PIC = '9999')
006311                  SET PLS-CRL.PLS# = W-PLS#
006312                  SET PLS-CRL.PLS-CRL#V = PN0024.CRL#V (W-CPT)
006314          ENDFOR
006315      ENDLOOP
006316      SET W-ITE = W-ITE + 1
006400      SET W-RET = W-RET - W-MAX
006500      UNTIL W-RET <= 0
006600      ENDLOOP
006700      ENDIF
006701      DO PN0025
006800      ENDPROC
006900      <<PN0025>> PROCEDURE
007000      LOOP VARYING W-ITE FROM 1 THRU W-PLS-CTE#N
007100          REFRESH PN0025
007200          SET PN0025.PLS# = W-PLS#
007300          SET PN0025.CTE#N = PN0023.CTE#N
007400          SET PN0025.CTE# = W-ITE
007401      LOOP VARYING W-CPT FROM 1 THRU W-PLS-VAR#N
007402          SET PN0025.COE# (W-CPT) = W-CPT
007403      ENDLOOP
007500      LOOP
007600          TRANSMIT PN0025
007700      UNTIL $PF6
007800      ENDLOOP
007900      SET W-OCC = $PANEL-GROUP-OCCURS (PN0025)
008200      FOR NEW PLS-TIN
008300          SET PLS-TIN.PLS# = W-PLS#
008400          SET PLS-TIN.PLS-LGN# = $EDIT (W-ITE,PIC = '9999')
008500          SET PLS-TIN.PLS-TI#V = PN0025.TI#V
008600          SET PLS-TIN.PLS-TI-SGN-CTE = PN0025.SGN-CTE
008603      ENDFOR
008604      IF W-OCC >= PN0024.VAR#N
008700          LOOP VARYING W-CPT FROM 1 THRU W-PLS-VAR#N
008800              FOR NEW PLS-COE
008900                  SET PLS-COE.PLS# = W-PLS#
009000                  SET PLS-COE.PLS-CLN# = $EDIT (W-CPT,PIC = '9999')
009100                  SET PLS-COE.PLS-LGN# = $EDIT (W-ITE,PIC = '9999')
009200                  SET PLS-COE.PLS-COE#V = PN0025.COE#V (W-CPT)
009300              ENDFOR
009400          ENDLOOP
009500      ELSE
009600          LOOP VARYING W-CPT FROM 1 THRU W-OCC

```

PROCEDURE:

ME002

VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
009700      FOR NEW PLS-COE  
009800          SET PLS-COE.PLS# = W-PLS#  
009900          SET PLS-COE.PLS-CLN# = $EDIT (W-CPT,PIC = '9999')  
010000          SET PLS-COE.PLS-LGN# = $EDIT (W-ITE,PIC = '9999')  
010100          SET PLS-COE.PLS-COE#V = PN0025.COE#V (W-CPT)  
010200      ENDFOR  
010300      ENDLOOP  
010400      SET W-RET = W-PLS-CTE#N - W-OCC  
010401      SET W-I#V = 1  
010402      SET W-MIN = W-OCC * W-I#V  
010500      LOOP  
010600          LOOP  
010700              REFRESH PN0025  
010701              LOOP VARYING W-CPT FROM 1 THRU W-RET  
010702                  SET W-I = W-MIN + W-CPT  
010703                  SET PN0025.COE# (W-CPT) =W-I  
010704              ENDLOOP  
010800              SET PN0025.PLS# = W-PLS#  
010900              SET PN0025.CTE#N = W-PLS-CTE#N  
011000              SET PN0025.CTE# = W-ITE  
011100              TRANSMIT PN0025  
011200          UNTIL $PF6  
011300          ENDLOOP  
011500          IF (W-RET - W-OCC) <= 0  
011600              SET W-MAX = W-RET  
011700          ELSE  
011800              SET W-MAX = W-OCC  
011900          ENDIF  
012100          LOOP VARYING W-CPT FROM 1 THRU W-MAX  
012200              FOR NEW PLS-COE  
012300                  SET PLS-COE.PLS# = W-PLS#  
012301                  SET W-I = W-MIN + W-CPT  
012400                  SET PLS-COE.PLS-CLN# = $EDIT (W-I,PIC = '9999')  
012500                  SET PLS-COE.PLS-LGN# = $EDIT (W-ITE,PIC = '9999')  
012600                  SET PLS-COE.PLS-COE#V = PN0025.COE#V (W-CPT)  
012700              ENDFOR  
012800              ENDLOOP  
012900              SET W-I#V = W-I#V + 1  
013000              SET W-RET = W-RET - W-MAX  
013100          UNTIL W-RET <= 0  
013200          ENDLOOP  
013300      ENDIF  
013301      ENDLOOP  
013400      ENDPROC  
013500      <<PN0026>> PROCEDURE  
013600          REFRESH PN0026  
013700          IF PN0021.CHX = 2  
013800      THEN
```

PROCEDURE: ME002 VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
013900      SET PN0026.TYP = 'EXECUTION'  
014000      ELSE  
014100          IF PN0021.CHX = 3  
014200          THEN  
014300              SET PN0026.TYP = 'SUPPRESSION'  
014400          ELSE  
014500              SET PN0026.TYP = 'MODIFICATION'  
014600          ENDIF  
014700      ENDIF  
014800      LOOP  
014900          TRANSMIT PN0026  
015000          IF $PF2  
015100              DO PN0022  
015200          ENDIF  
015300          FOR EACH PLS-PLS-R  
015400              WHERE PLS# = PN0026.PLS#  
015401                  LOOP VARYING W-CPT FROM 1 THRU 8  
015500                      SET PN0026.PLS#L (W-CPT) = $SUBSTR (PLS-PLS-R.PLS#L,  
015501                          START = 50 * (W-CPT - 1) + 1)  
015502          ENDL  
015600          TRANSMIT PN0026  
015700          ENDFOR  
015800      UNTIL $PF6  
015900      ENDL  
015902      SET W-CHX-PLS# = PN0026.PLS#  
015904      FOR EACH PLS-PLS-R  
015905          WHERE PLS# = W-CHX-PLS#  
015908              SET W-PLS-VAR#N = PLS-PLS-R.PLS-VAR#N  
015910              SET W-PLS-CTE#N = PLS-PLS-R.PLS-CTE#N  
015912              SET W-PLS-ZO#V = PLS-PLS-R.PLS-ZO#V  
015913              SET W-PLS-OPT#V = PLS-PLS-R.PLS-OPT#V  
015914              SET W-ITE# = 0  
015915          ENDFOR  
016000      SELECT PN0026.CON  
016100          WHEN 'N' DO PN0022  
016200          WHEN 'O'  
016300              IF PN0021.CHX = 2  
016301                  SET W-PVC = FALSE  
016400                  CALL UT002 USING INPUT W-CHX-PLS#,W-PLS-VAR#N,W-PLS-CTE#N,  
016401                      W-PVC  
016402                      UPDATE W-PLS-ZO#V,W-PLS-OPT#V,W-ITE#  
016403                          : EXECUTION PLS  
016404                  RELEASE PROGRAM UT002  
016500              ELSE  
016600                  IF PN0021.CHX = 3  
016700                      CALL UT003 USING INPUT W-CHX-PLS#  
016702                          : SUPPRESSION PLS  
016703                  RELEASE PROGRAM UT003
```

PROCEDURE:

ME002

VERSION: 001 STATUS: TEST

```
-----  
SEQ DATA  
-----  
016800 :ELSE  
016900 : CALL UT004 USING INPUT W-CHX-PLS#,W-PLS-VAR#N,W-PLS-CTE#  
016901 : UPDATE W-PLS-ZO#V,W-PLS-OPT#V  
016902 : MODIFICATION PLS  
016903 : RELEASE PROGRAM UT004  
017000 : ENDIF  
017100 : ENDIF  
017200 : ENDSEL  
017300 : ENDPROC  
017400 <<PN0022>> PROCEDURE  
017500 FOR EACH PLS-PLS-R  
017600 ORDERED BY ASCENDING PLS#  
017700 LOOP  
017800 REFRESH PN0022  
017900 SET PN0022.PLS# = PLS-PLS-R.PLS#  
017901 LOOP VARYING W-CPT FROM 1 THRU 8  
018000 SET PN0022.PLS#L (W-CPT) = $SUBSTR (PLS-PLS-R.PLS#L,  
018001 : START = 50 * (W-CPT - 1) + 1)  
018002 : ENDLOOP  
018100 TRANSMIT PN0022  
018200 UNTIL $PF6  
018300 : ENDLOOP  
018400 : ENDFOR  
018500 : ENDPROC
```

IDENTIFICATION: UT001

VERSION: 001 STATUS: TEST

PROGRAM UT001

Created 11/02/87 By ETUD04  
Last Modified 01/07/88 at 15:13 By ETUD04  
Last Compiled 01/07/88 at 15:15

Run Status: PRIVATE  
Short Description: regression lineaire  
Language: IDEAL Target Date \_\_ \_\_ \_\_ Actual Date \_\_ \_\_ \_\_

Description:

---

---

---

---

RESOURCES:

UT001                    VERSION: 001 STATUS: TEST

DATAVIEW	VER	PANEL	VER	REPORT	VER	PROGRAM	VER	SYS
						UT002	0001	

PARAMETER:

UT001

VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	U	M	COMMENT/REDEF/DEP ON/COPY
000001	1	P-ACQ					U	D	
000100	2	ACQ	N		3	4			
000101	1	P-REG-A#V					U	D	
000200	2	REG-A#V	N		7.8	3			
000201	1	P-REG-B#V					U	D	
000300	2	REG-B#V	N		7.8	3			
000400	1	P-TRA#N	N		1		U	D	
000401	1	P-REG-MOE					U	D	
000500	2	REG-MOE	N		7.8	3			
000501	1	P-PTS-TRA#N			.		U	D	
000502	2	PTS-TRA#N	N		3	3			
000503	1	P-RDT-MCD					I	D	
000600	2	RDT-MCD	N		7.8	20			
000601	1	P-ACQ#V					I	D	
000700	2	ACQ#V	N		3	20			
000800	1	P-PTS#N	N		3		I	D	
000801	1	P-PLS#	X		3		I	D	
000900	1	P-RDT	F				I	D	

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND. NAME, F=FLAG,  
V=VARIABLE, D=DATE  
I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
U (PARAMETER UPDATING): U=UPDATABLE, I=INPUT ONLY  
M (PARAMETER MATCHING): BLANK OR D=DYNAMIC, I=IDENTICAL  
REDEF=REDEFINITION, DEP ON= DEPENDING ON

WORKING DATA: UT001            VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	VAL/COMMENT/REDEF/DEP	ON/COPY
000100	1	W-MIN	N		7.8			
000200	1	W-CPT	N		3			
000201	1	W-CPT-2	N		3			
000202	1	W-CPT-3	N		3			
000300	1	W-IND	N		3			
000400	1	W-PAS	N		3			
000600	1	W-I	N		3			
000700	1	W-PTS#N	N		3			
000800	1	W-REL-A#V	N		7.8			
000900	1	W-REL-B#V	N		8			
001000	1	W-MOE	N		7.8			
001100	1	W-SBS	N		7.8			
001200	1	W-SRD	N		7.8			
001300	1	W-SOA	N		7.8			
001400	1	W-SA2	N		7.8			
001500	1	W-COA	N		7.8			
001600	1	W-ORA	N		7.8			
001700	1	W-IND#V	N		7.8			
001800	1	W-IND-ARO	N		7.8			

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND.NAME, F=FLAG,  
V=VARIABLE, D=DATE  
I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
REDEF= REDEFINITION, DEP ON= DEPENDING ON

PROCEDURE: UT001 VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
000001  LIST 'P-RDT',P-RDT
000100  IF P-RDT
000101      DO CALCULA
000102  ELSE
000103      LOOP VARYING W-CPT FROM 1 THRU P-TRA#N
000104          SET W-PTS#N = P-PTS-TRA#N.PTS-TRA#N (W-CPT)
000105          DO REGLINEAIRE
000106          SET P-REG-A#V.REG-A#V (W-CPT) = W-COA
000107          SET P-REG-B#V.REG-B#V (W-CPT) = W-ORA
000108          DO MOYENNE
000109          SET P-REG-MOE.REG-MOE (W-CPT) = W-MOE
000110      ENDLOOP
000111  ENDIF
000200  <<CALCULA>>PROCEDURE
000201      SET W-IND = 1
000300      SET W-MIN = P-RDT-MCD.RDT-MCD (1)
000400      LOOP VARYING W-CPT FROM 2 THRU P-PTS#N
000500          IF P-RDT-MCD.RDT-MCD (W-CPT) < W-MIN
000600              SET W-MIN = P-RDT-MCD.RDT-MCD (W-CPT)
000601              SET W-IND = W-CPT
000700          ENDIF
000800      ENDLOOP
000900      SET W-MIN = P-ACQ#V.ACQ#V (W-IND)
000901  LIST W-MIN,'A'
000902      DO TRANCHES
001000  ENDPROC
001001  <<TRANCHES>>PROCEDURE
001002      SET W-PAS = P-ACQ#V.ACQ#V (2) - P-ACQ#V.ACQ#V (1)
001003  LIST W-PAS,'PAS',P-ACQ#V.ACQ#V (1),'ACQ-1',W-MIN,'W-MIN'
001004      IF W-MIN = P-ACQ#V.ACQ#V (1)
001005          SET P-TRA#N = 2
001006          SET P-PTS-TRA#N.PTS-TRA#N(1) = (80 - W-MIN) / W-PAS
001007          SET P-PTS-TRA#N.PTS-TRA#N (2) = 2
001008          SET P-ACQ.ACQ (1) = W-MIN
001009          SET P-ACQ.ACQ (2) = 80
001010          SET P-ACQ.ACQ (3) = 85
001011      ELSE
001012          SET P-TRA#N = 3
001013          SET P-PTS-TRA#N.PTS-TRA#N(1) = (W-MIN - P-ACQ#V.ACQ#V (1)) / W-PAS
001014          SET P-PTS-TRA#N.PTS-TRA#N(2) = (80 - W-MIN) / W-PAS
001015          SET P-PTS-TRA#N.PTS-TRA#N (3) = 2
001016          SET P-ACQ.ACQ (1) = P-ACQ#V.ACQ#V (1)
001017          SET P-ACQ.ACQ (2) = W-MIN
001018          SET P-ACQ.ACQ (3) = 80
001019          SET P-ACQ.ACQ (4) = 85
001020      ENDIF
001021  LIST P-TRA#N,'TRA#N'
001022      LOOP VARYING W-I FROM 1 THRU P-TRA#N

```

PROCEDURE: UT001 VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
001023      SET W-PTS#N = P-PTS-TRA#N.PTS-TRA#N (W-I)  
001024      DO REGLINEAIRE  
001025      SET P-REG-A#V.REG-A#V (W-I)= W-COA  
001026      SET P-REG-B#V.REG-B#V (W-I) = W-ORA  
001027      DO MOYENNE  
001028      SET P-REG-MOE.REG-MOE (W-I) = W-MOE  
001029      LIST P-REG-MOE.REG-MOE (W-I), 'MOE'  
001030      ENDLLOOP  
001031      ENDPROC  
001100      <<REGLINEAIRE>>PROCEDURE  
001101      LIST 'REGLINEAIRE-DBT'  
001200      LOOP VARYING W-CPT-2 FROM P-ACQ.ACQ (W-I) BY W-PAS  
001201          THRU P-ACQ.ACQ (W-I + 1)  
001202          SET W-CPT-3 = P-ACQ.ACQ (W-I) / W-PAS  
001300          SET W-SBS = W-SBS + W-CPT-2  
001400          SET W-SRD = W-SRD + P-RDT-MCD.RDT-MCD (W-CPT-3)  
001500          SET W-SOA = W-SOA + W-CPT-2 * P-RDT-MCD.RDT-MCD (W-CPT-3)  
001600          SET W-SA2 = W-SA2 + W-CPT-2 * W-CPT-2  
001700      ENDLLOOP  
001800      SET W-COA = (W-PTS#N * W-SOA - W-SBS * W-SRD) /  
001900          (W-PTS#N * W-SA2 - W-SBS * W-SBS)  
002000      SET W-ORA = W-SRD / W-PTS#N - W-COA * W-SBS / W-PTS#N  
002001      IF W-I = 1  
002100          SET W-REL-A#V = W-COA * W-MIN + W-ORA  
002200          SET W-REL-B#V = W-COA * 80 + W-ORA  
002201      ENDIF  
002202      LIST 'REGLINEAIRE-FIN'  
002300      ENDPROC  
002400      <<MOYENNE>> PROCEDURE  
002401      LIST 'MOYENNE-DBT'  
002402      LIST P-REG-A#V.REG-A#V (W-I), 'A', P-REG-B#V.REG-B#V (W-I), 'B'  
002403          SET W-IND#V = P-ACQ#V.ACQ#V (W-I + 1) - P-ACQ#V.ACQ#V (W-I) / 2  
002404          SET W-IND-ARO = $ROUND (W-IND#V, FACTOR = 1)  
002405          SET W-MOE = P-REG-A#V.REG-A#V (W-I) * W-IND-ARO + P-REG-B#V.REG-B#V  
002406              (W-I)  
002407      LIST 'MOYENNE-FIN'  
002500      ENDPROC
```

IDENTIFICATION: UT002

VERSION: 001 STATUS: TEST

PROGRAM UT002

Created 11/10/87 By ETUD04  
Last Modified 01/07/88 at 13:15 By ETUD04  
Last Compiled 01/07/88 at 13:24

Run Status: PRIVATE  
Short Description: execution pls  
Language: IDEAL Target Date \_\_\_ \_\_\_ \_\_\_ Actual Date \_\_\_ \_\_\_ \_\_\_

Description:

---

---

---

---

RESOURCES:

UT002                    VERSION: 001 STATUS: TEST

DATAVIEW	VER	PANEL	VER	REPORT	VER	PROGRAM	VER	SYS
PLS-BAS	0001					UT005	0001	
PLS-CHB	0001					UT006	0001	
PLS-COE	0001							
PLS-COE-R	0001							
PLS-CRL	0001							
PLS-PLS	0001							
PLS-PSS	0001							
PLS-TIN	0001							
PLS-TIN-R	0001							
PLS-XBN	0001							

PARAMETER:

UT002

VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	U	M	COMMENT/REDEF/DEP ON/COPY
000100	1	P-CHX-PLS#	X		3		I	D	
000200	1	P-PLS-VAR#N	N		4		I	D	
000300	1	P-PLS-CTE#N	N		4		I	D	
000301	1	P-PVC	F				I	D	
000400	1	P-PLS-ZO#V	N		7.8		U	D	
000500	1	P-PLS-OPT#V	X		3		U	D	
000600	1	P-ITE#	N		5		U	D	

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND. NAME, F=FLAG,  
V=VARIABLE, D=DATE

I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY

CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS

U (PARAMETER UPDATING): U=UPDATABLE, I=INPUT ONLY

M (PARAMETER MATCHING): BLANK OR D=DYNAMIC, I=IDENTICAL

REDEF=REDEFINITION, DEP ON= DEPENDING ON

WORKING DATA: UT002 VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	VAL/COMMENT/REDEF/DEP ON/COPY
000100	1	W-I	N		7.8		
000200	1	W-J	N		4		
000300	1	W-CPT	N		4		
000301	1	W-CPT-2	N		4		
000302	1	W-CPT-3	N		4		
000303	1	W-CPT-4	N		4		
000500	1	W-LGN	N		4		
000501	1	W-BAS-CLN	N		4		
000502	1	W-IND-ECR#V					
000600	2	IND-ECR#V	N		4	300	
000700	1	W-CHX-PLS#	X		3		
000800	1	W-PLS-VAR#N	N		4		
000900	1	W-PLS-CTE#N	N		4		
000901	1	W-VAR#N	N		4		
000902	1	W-ARF-VAR#N	N		4		
000903	1	W-ECR-BAS#N	N		3		
001000	1	W-PLS-ZO#V	N		7.8		
001100	1	W-PLS-OPT#V	X		3		
001200	1	W-ITE#N	N		5		
001400	1	W-PVC	F				
001500	1	W-PHASE	N		1		
001501	1	W-PI					
001600	2	PI	N		7.8	300	
001700	1	W-FIN	F				
001701	1	W-IND-ECR-BAS					
001800	2	IND-BAS	N		4	300	
001900	1	W-ECR-VAR#N	N		4		
002100	1	W-IND-PS-MIN	N		4		
002200	1	W-IND-CN-MIN	N		4		

LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND.NAME, F=FLAG,  
 V=VARIABLE, D=DATE  
 I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
 CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
 REDEF= REDEFINITION, DEP ON= DEPENDING ON

PROCEDURE: UT002 VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
000200 <<MIN-MAX>> PROCEDURE  
000203           : EXECUTION D'UN PLS  
000300     FOR EACH PLS-PLS  
000400     WHERE PLS# = P-CHX-PLS#  
000500       IF PLS-PLS.PLS-OPT#V = 'MAX'  
000501         SET PLS-PLS.PLS-OPT#V = 'MIN'  
000600         SET PLS-PLS.PLS-ZO#V = - PLS-PLS.PLS-ZO#V  
000700         FOR EACH PLS-CRL  
000800         WHERE PLS-CRL.PLS# = P-CHX-PLS#  
000900         SET PLS-CRL.PLS-CRL#V = - PLS-CRL.PLS-CRL#V  
001000         ENDFOR  
001001       ENDIF  
001100     ENDFOR  
001102     DO B-POSITIF  
001300 ENDPROC  
001400 <<B-POSITIF>> PROCEDURE  
001500     FOR EACH PLS-TIN  
001600     WHERE PLS# = P-CHX-PLS#  
001700       IF PLS-TIN.PLS-TI#V < 0  
001701         SET PLS-TIN.PLS-TI#V = - PLS-TIN.PLS-TI#V  
001800         FOR EACH PLS-COE  
001801         WHERE ( PLS-COE.PLS# = P-CHX-PLS# ) AND  
001900           ( PLS-COE.PLS-LGN# = PLS-TIN.PLS-LGN# )  
002100         SET PLS-COE.PLS-COE#V = - PLS-COE.PLS-COE#V  
002200         ENDFOR  
002300       ENDIF  
002400     ENDFOR  
002402     DO CTE-EGALITE  
002500 ENDPROC  
002600 <<CTE-EGALITE>> PROCEDURE  
002700     SET W-J = 1  
002701     LOOP VARYING W-CPT-4 FROM 1 THRU P-PLS-CTE#N  
002800     FOR EACH PLS-TIN-R WHERE ( PLS# = P-CHX-PLS# ) AND  
002801       ( PLS-LGN# = $EDIT ( W-CPT-4, PIC = '9999' ) )  
002900     IF PLS-TIN-R.PLS-TI-SGN-CTE NE '='  
002902       SET W-CPT = P-PLS-VAR#N + W-CPT-4  
003000       FOR NEW PLS-CRL  
003001         SET PLS-CRL.PLS# = P-CHX-PLS#  
003100         SET PLS-CRL.PLS-CLN# = $EDIT ( W-CPT, PIC = '9999' )  
003300         SET PLS-CRL.PLS-CRL#V = 0  
003500         SET W-IND-ECR#V.IND-ECR#V ( W-J ) = W-CPT  
003600         SET W-J = W-J + 1  
003601         SET W-CPT-3 = $NUMBER ( PLS-CRL.PLS-CLN# ) - P-PLS-VAR#N  
003603       ENDFOR  
003700     IF ( PLS-TIN-R.PLS-TI-SGN-CTE = '<' ) OR  
003701       ( PLS-TIN-R.PLS-TI-SGN-CTE = '<=' )  
003800       SET W-I = 1  
003900     ELSE
```



PROCEDURE: UT002 VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
007212          ELSE
007213          SET PLS-BAS.PLS-BAS#V = 0
007216          ENDIF
007300          ENDFOR
007400          ENDLOOP
007401          SET W-BAS-CLN = W-BAS-CLN + 1
007600          ENDIF
007700          ENDFOR
007701          ENDLOOP
007800          ENDLOOP
007900          DO CHARGEMENT-X
007901          SET W-CHX-PLS# = P-CHX-PLS#
007906          SET W-VAR#N = P-PLS-VAR#N + W-J - 1
007908          SET W-PLS-VAR#N = P-PLS-VAR#N
007909          SET W-PLS-CTE#N = P-PLS-CTE#N
007910          SET W-PLS-OPT#V = P-PLS-OPT#V
007911          SET W-PLS-ZO#V = P-PLS-ZO#V
007912          LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
007913          SET W-PI.PI (W-CPT)= 0
007914          ENDLOOP
007915          SET W-PHASE = 0
007916          SET W-ITE#N = 0
007918          IF P-PVC
007919          SET W-PVC = TRUE
007920          ELSE
007921          SET W-PVC = FALSE
007922          ENDIF
007923          SET W-ECR-VAR#N = W-J - 1
007924          SET W-FIN = FALSE
007925          SET W-BAS-CLN = W-BAS-CLN - 1
007926          SET W-ARF-VAR#N = 0
007927          SET W-IND-PS-MIN = 0
007928          SET W-IND-CN-MIN = 0
007929          IF W-BAS-CLN = P-PLS-CTE#N
007931          CALL UT005 USING INPUT W-CHX-PLS#,W-PLS-VAR#N,W-PLS-CTE#N,W-VAR#N,
007932          W-PHASE,W-ECR-VAR#N,W-IND-ECR#V,W-PVC,
007933          W-ARF-VAR#N
007934          UPDATE W-PLS-ZO#V,W-PLS-OPT#V,W-PI,
007935          W-ITE#N,W-FIN,
007936          : METHODE PRIMALE
007937          RELEASE PROGRAM UT005
007938          DO SUPPRESSION-DVW
007941          ELSE
007943          SET W-ECR-BAS#N = W-BAS-CLN
007944          CALL UT006 USING INPUT W-CHX-PLS#,W-PLS-VAR#N,W-PLS-CTE#N,W-VAR#N,
007945          W-ECR-BAS#N,W-IND-ECR#V,W-PVC
007946          UPDATE W-PLS-ZO#V,W-PLS-OPT#V,W-ITE#N
007947          : METHODE DES 2 F #SES

```

PROCEDURE: UT002                   VERSION: 001 STATUS: TEST

```
-----  
SEQ     DATA  
-----  
007948         RELEASE PROGRAM UT006  
007952             ENDIF  
008000     ENDPROC  
008100     <<CHARGEMENT-X>> PROCEDURE  
008200         LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
008300             FOR EACH PLS-TIN-R WHERE (PLS# = P-CHX-PLS#) AND  
008400                     (PLS-LGN# = $EDIT(W-CPT,PIC = '9999'))  
008500                 FOR NEW PLS-XBN  
008600                     SET PLS-XBN.PLS# = P-CHX-PLS#  
008700                     SET PLS-XBN.PLS-CLN# = $EDIT (W-CPT,PIC = '9999')  
008800                     SET PLS-XBN.PLS-XB#V = PLS-TIN-R.PLS-TI#V  
008900                 ENDFOR  
009000             ENDFOR  
009100         ENDLOOP  
009200     ENDPROC  
009300     <<SUPPRESSION-DVW>> PROCEDURE  
009400         FOR EACH PLS-XBN  
009500             WHERE PLS# = P-CHX-PLS#  
009600                 DELETE PLS-XBN  
009700         ENDFOR  
009800         FOR EACH PLS-BAS  
009900             WHERE PLS# = P-CHX-PLS#  
010000                 DELETE PLS-BAS  
010100         ENDFOR  
010200         FOR EACH PLS-CHB  
010300             WHERE PLS# = P-CHX-PLS#  
010400                 DELETE PLS-CHB  
010500         ENDFOR  
010600         FOR EACH PLS-PSS  
010700             WHERE PLS# = P-CHX-PLS#  
010800                 DELETE PLS-PSS  
010900         ENDFOR  
010901         LOOP VARYING W-CPT FROM 1 THRU W-ECR-VAR#N  
010902             FOR EACH PLS-CRL  
010903                 WHERE (PLS# = P-CHX-PLS#) AND  
010904                     (PLS-CLN# = $EDIT (W-IND-ECR#V.IND-ECR#V (W-CPT),  
010905                             PIC = '9999'))  
010906                     DELETE PLS-CRL  
010907                 ENDFOR  
010908             LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N  
010909                 FOR EACH PLS-COE  
010910                     WHERE (PLS# = P-CHX-PLS#)AND  
010911                         (PLS-CLN# = $EDIT (W-IND-ECR#V.IND-ECR#V (W-CPT),  
010912                             PIC = '9999')) AND  
010913                         (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))  
010914                     DELETE PLS-COE  
010915                 ENDFOR  
010916             ENDLOOP
```

PROCEDURE:

UT002

VERSION: 001 STATUS: TEST

-----  
SEQ DATA

-----  
010917 ENDLOOP  
011000 ENDPROC

IDENTIFICATION: UT003

VERSION: 001 STATUS: TEST

PROGRAM UT003

Created 11/10/87 By ETUD04  
Last Modified 11/30/87 at 10:51 By ETUD04  
Last Compiled 12/01/87 at 14:35

Run Status: PRIVATE  
Short Description: suppression pls  
Language: IDEAL Target Date \_\_\_ \_\_\_ \_\_\_ Actual Date \_\_\_ \_\_\_ \_\_\_

Description:

---

---

---

---

RESOURCES:

UT003                    VERSION: 001 STATUS: TEST

DATAVIEW	VER	PANEL	VER	REPORT	VER	PROGRAM	VER	SYS
PLS-ACT	0001							
PLS-BAS	0001							
PLS-CHB	0001							
PLS-COE	0001							
PLS-CRB	0001							
PLS-CRL	0001							
PLS-PLS	0001							
PLS-PSS	0001							
PLS-TIN	0001							
PLS-XBN	0001							

PARAMETER: UT003            VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	U	M	COMMENT/REDEF/DEP ON/COPY
000100	1	P-CHX-PLS#	X		3		I	D	

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND. NAME, F=FLAG,  
V=VARIABLE, D=DATE

I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY

CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS

U (PARAMETER UPDATING): U=UPDATABLE, I=INPUT ONLY

M (PARAMETER MATCHING): BLANK OR D=DYNAMIC, I=IDENTICAL

REDEF=REDEFINITION, DEP ON= DEPENDING ON

WORKING DATA: UT003 VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	VAL/COMMENT/REDEF/DEP ON/COPY
-----	-------	------------	---	---	-------	-------	-------------------------------

NO WORKING-DATA FOR THIS PROGRAM

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND.NAME, F=FLAG,  
V=VARIABLE, D=DATE  
I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
REDEF= REDEFINITION, DEP ON= DEPENDING ON

PROCEDURE:

UT003

VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
000100  FOR EACH PLS-PLS  
000200  WHERE PLS# = P-CHX-PLS#  
000300      DELETE PLS-PLS  
000400  ENDFOR  
000500  FOR EACH PLS-CRL  
000600  WHERE PLS# = P-CHX-PLS#  
000700      DELETE PLS-CRL  
000800  ENDFOR  
000900  FOR EACH PLS-TIN  
001000  WHERE PLS# = P-CHX-PLS#  
001100      DELETE PLS-TIN  
001200  ENDFOR  
001300  FOR EACH PLS-COE  
001400  WHERE PLS# = P-CHX-PLS#  
001500      DELETE PLS-COE  
001600  ENDFOR  
001700  FOR EACH PLS-BAS  
001800  WHERE PLS# = P-CHX-PLS#  
001900      DELETE PLS-BAS  
002000  ENDFOR  
002100  FOR EACH PLS-PSS  
002200  WHERE PLS# = P-CHX-PLS#  
002300      DELETE PLS-PSS  
002400  ENDFOR  
002500  FOR EACH PLS-CHB  
002600  WHERE PLS# = P-CHX-PLS#  
002700      DELETE PLS-CHB  
002800  ENDFOR  
002900  FOR EACH PLS-XBN  
003000  WHERE PLS# = P-CHX-PLS#  
003100      DELETE PLS-XBN  
003200  ENDFOR  
003300  FOR EACH PLS-ACT  
003400  WHERE PLS# = P-CHX-PLS#  
003500      DELETE PLS-ACT  
003600  ENDFOR  
003700  FOR EACH PLS-CRB  
003800  WHERE PLS# = P-CHX-PLS#  
003900      DELETE PLS-CRB  
004000  ENDFOR
```

S

IDENTIFICATION: UT005

VERSION: 001 STATUS: TEST

PROGRAM UT005

Created 11/12/87 By ETUD04  
Last Modified 01/06/88 at 10:45 By ETUD04  
Last Compiled 01/06/88 at 11:21

Run Status: PRIVATE  
Short Description: methode primale  
Language: IDEAL Target Date \_\_\_ \_\_\_ \_\_\_ Actual Date \_\_\_ \_\_\_ \_\_\_

Description:

---

---

---

---

RESOURCES:

UT005                    VERSION: 001 STATUS: TEST

DATAVIEW	VER	PANEL	VER	REPORT	VER	PROGRAM	VER	SYS
PLS-BAS	0001	_____	_____	_____	_____	UT008	0001	_____
PLS-BAS-R	0001	_____	_____	_____	_____	UT009	0001	_____
PLS-CHB	0001	_____	_____	_____	_____	UT010	0001	_____
PLS-CHB-R	0001	_____	_____	_____	_____			_____
PLS-COE-R	0001	_____	_____	_____	_____			_____
PLS-CRL-R	0001	_____	_____	_____	_____			_____
PLS-PLS	0001	_____	_____	_____	_____			_____
PLS-PSS	0001	_____	_____	_____	_____			_____
PLS-PSS-R	0001	_____	_____	_____	_____			_____
PLS-TIN-R	0001	_____	_____	_____	_____			_____
PLS-XBN	0001	_____	_____	_____	_____			_____
PLS-XBN-R	0001	_____	_____	_____	_____			_____

PARAMETER: UT005 VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	U	M	COMMENT/REDEF/DEP ON/COPY
000100	1	P-CHX-PLS#	X		3		I	D	
000200	1	P-PLS-VAR#N	N		4		I	D	
000300	1	P-PLS-CTE#N	N		4		I	D	
000301	1	P-VAR#N	N		4		I	D	
000302	1	P-PHASE	N		1		I	D	
000303	1	P-ECR-VAR#N	N		4		I	D	
000304	1	P-IND-ECR#V					I	D	
000305	2	IND-ECR#V	N		4	300			
000306	1	P-PVC	F				I	D	
000308	1	P-ARF-VAR#N	N		4		I	D	
000400	1	P-PLS-ZO#V	N		7.8		U	D	
000500	1	P-PLS-OPT#V	X		3		U	D	
000501	1	P-PI					U	D	
000700	2	PI	N		7.8	300			
000900	1	P-ITE#N	N		5		U	D	
001000	1	P-FIN	F				U	D	

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND. NAME, F=FLAG, V=VARIABLE, D=DATE  
I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
U (PARAMETER UPDATING): U=UPDATABLE, I=INPUT ONLY  
M (PARAMETER MATCHING): BLANK OR D=DYNAMIC, I=IDENTICAL  
REDEF=REDEFINITION, DEP ON= DEPENDING ON

WORKING DATA: UT005 VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	VAL/COMMENT/REDEF/DEP	ON/COPY
000001	1	W-Y	N		7.8			
000002	1	W-BAS-IND	N		4			
000100	1	W-CPT	N		4			
000200	1	W-CPT-2	N		4			
000300	1	W-K	N		4			
000400	1	W-I	N		4			
000401	1	W-PI						
000500	2	PI	N		7.8	300	0	
000600	1	W-HOR-BAS	F					
000601	1	W-IND-HOR-BAS						
000700	2	IND-HOR-BAS	N		4	3000		
000800	1	W-PN	N		7.8			
000900	1	W-CN#V	N		7.8			
000901	1	W-IND						
001000	2	IND	N		4	3000		
001100	1	W-CN-MIN	N		7.8			
001200	1	W-IND-CN-MIN	N		4			
001300	1	W-PS	N		7.8		0	
001400	1	W-X	N		7.8			
001500	1	W-PS-MIN	N		7.8			
001600	1	W-IND-PS-MIN	N		4			
001601	1	W-JR						
001700	2	JR	N		7.8	300	0	
001701	1	W-BAS						
001800	2	BAS	N		7.8	300	0	
001900	1	W-CN	F					
002000	1	W-FIN	F					
002001	1	W-IND-BAS						
002100	2	IND-BAS	N		4	300		
002200	1	W-ECR-VAR#N	N		4			
002300	1	W-ARF-VAR#N	N		4			
002400	1	W-CHX-PLS#	X		3			
002500	1	W-PLS-CTE#N	N		4			
002501	1	W-PLS-VAR#N	N		4			
002600	1	W-RET	N		5			
002700	1	W-J	N		4			

## LEGEND:

0

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND.NAME, F=FLAG,  
V=VARIABLE, D=DATE  
I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
REDEF= REDEFINITION, DEP ON= DEPENDING ON

PROCEDURE: UT005 VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
000001  <<UT005>> PROCEDURE
000003      IF P-PHASE = 0
000004          LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
000005              SET W-IND-BAS.IND-BAS (W-CPT) = P-IND-ECR#V.IND-ECR#V
000006                  (W-CPT)
000007          ENDLOOP
000008      ELSE
000009          LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
000010              FOR EACH PLS-BAS-R
000011                  WHERE (PLS# = P-CHX-PLS#) AND
000012                      (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND
000013                      (PLS-LGN# = $EDIT ( 1,PIC = '9999'))
000014                  SET W-IND-BAS.IND-BAS (W-CPT) =
000015                      $NUMBER (PLS-BAS-R.PLS-BAS-IND#V)
000016              ENDFOR
000017          ENDLOOP
000018      ENDIF
000020      IF P-FIN
000021          SET W-FIN = TRUE
000022      ELSE
000023          SET W-FIN = FALSE
000024      ENDIF
000127      SET W-CHX-PLS# = P-CHX-PLS#
000128      SET W-PLS-CTE#N = P-PLS-CTE#N
000129      LOOP UNTIL W-FIN
000130          SET P-ITE#N = P-ITE#N + 1
000136          SET W-RET = $REMAINDER (P-ITE#N, DIV = 20)
000137          IF W-RET = 0
000138              CALL UT008 USING INPUT W-CHX-PLS#,W-PLS-CTE#N
000139              RELEASE PROGRAM UT008
000140          ENDIF
000141          DO CALCUL-PI
000145      ENDLOOP
000146  ENDPROC
000147  <<CALCUL-PI>> PROCEDURE
000200      LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
000201          SET W-PI.PI (W-CPT) = 0
000300          LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N
000301              FOR EACH PLS-CRL-R
000400                  WHERE (PLS# = P-CHX-PLS#) AND
000401                      (PLS-CLN# = $EDIT (W-IND-BAS.IND-BAS (W-CPT-2),
000402                          PIC = '9999'))
000500                  FOR EACH PLS-BAS-R
000600                      WHERE (PLS# = P-CHX-PLS#) AND
000700                          (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND
000800                          (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))
000900                      SET W-PI.PI(W-CPT) = W-PI.PI(W-CPT) + PLS-CRL-R.PLS-CRL#V
000901                          * PLS-BAS-R.PLS-BAS#V

```

PROCEDURE: UT005 VERSION: 001 STATUS: TEST

```
-----  
SEQ  DATA  
-----  
001000      ENDFOR  
001100      ENDFOR  
001200      ENDLOOP  
001300      ENDLOOP  
001301      FOR EACH PLS-PLS  
001302      WHERE PLS# = P-CHX-PLS#  
001303      LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
001304      FOR EACH PLS-TIN-R  
001305      WHERE (PLS# = P-CHX-PLS#) AND  
001306      (PLS-LGN# = $EDIT (W-CPT,PIC = '9999'))  
001307      SET PLS-PLS.PLS-ZO#V = PLS-PLS.PLS-ZO#V + W-PI.PI (W-CPT)  
001308      * PLS-TIN-R.PLS-TI#V  
001310      ENDFOR  
001312      ENDLOOP  
001313      ENDFOR  
001314      SET W-CN = TRUE  
001317      DO IND-HORS-BASE  
001400      ENDPROC  
001500      <<IND-HORS-BASE>> PROCEDURE  
001502      IF P-ITE#N = 1  
001600      SET W-J = 1  
001700      LOOP VARYING W-CPT FROM 1 THRU P-VAR#N  
001800      SET W-HOR-BAS = TRUE  
001900      SET W-K = 1  
002000      LOOP WHILE (W-K <= P-PLS-CTE#N) AND (W-HOR-BAS)  
002100      FOR EACH PLS-BAS-R  
002200      WHERE (PLS# = P-CHX-PLS#) AND  
002201      (PLS-CLN# = $EDIT (W-K,PIC = '9999')) AND  
002202      (PLS-LGN# = $EDIT (1,PIC = '9999'))  
002300      IF W-CPT = $NUMBER (PLS-BAS-R.PLS-BAS-IND#V)  
002400      SET W-HOR-BAS = FALSE  
002500      ELSE  
002600      SET W-K = W-K + 1  
002700      ENDIF  
002800      ENDFOR  
002900      ENDLOOP  
003000      IF (W-K > P-PLS-CTE#N) AND W-HOR-BAS  
003100      SET W-IND-HOR-BAS.IND-HOR-BAS (W-J) = W-CPT  
003200      SET W-J = W-J + 1  
003300      ENDIF  
003400      ENDLOOP  
003401      ENDIF  
003402      DO CALCUL-CN  
003500      ENDPROC  
003600      <<CALCUL-CN>> PROCEDURE  
003700      LOOP VARYING W-CPT FROM 1 THRU (P-VAR#N - P-PLS-CTE#N)  
003701      SET W-PN = 0  
003702      FOR EACH PLS-CRL-R
```

PROCEDURE:

UT005

VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
003703      WHERE (PLS# = P-CHX-PLS#) AND  
003704      (PLS-CLN# = $EDIT (W-IND-HOR-BAS.IND-HOR-BAS (W-CPT),  
003705      PIC = '9999'))  
003706      SET W-CN#V = PLS-CRL-R.PLS-CRL#V  
003707      ENDFOR  
003800      LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N  
003900      FOR EACH PLS-COE-R  
004000      WHERE (PLS# = P-CHX-PLS#) AND  
004100      (PLS-CLN# = $EDIT (W-IND-HOR-BAS.IND-HOR-BAS (W-CPT),  
004101      PIC = '9999'))  
004200      AND (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))  
004300      SET W-PN = W-PN + W-PI.PI (W-CPT-2)* PLS-COE-R.PLS-COE#V  
004500      ENDFOR  
004600      ENDLOOP  
004700      SET W-CN#V = W-CN#V - W-PN  
004800      IF (P-ITE#N = 1) AND (P-PHASE NE 1) AND (P-PHASE NE 2)  
004900      FOR NEW PLS-CHB  
005000      SET PLS-CHB.PLS# = P-CHX-PLS#  
005100      SET PLS-CHB.PLS-CLN# = $EDIT (W-CPT,PIC = '9999')  
005200      SET PLS-CHB.PLS-CHB#V = W-CN#V  
005300      SET PLS-CHB.PLS-CHB-IND#V =  
005301      $EDIT (W-IND-HOR-BAS.IND-HOR-BAS (W-CPT),PIC = '9999')  
005400      ENDFOR  
005500      ELSE  
005600      FOR EACH PLS-CHB  
005700      WHERE (PLS# = P-CHX-PLS#) AND  
005701      (PLS-CLN# = $EDIT (W-CPT,PIC = '9999'))  
005800      SET PLS-CHB.PLS-CHB#V = W-CN#V  
005900      SET PLS-CHB.PLS-CHB-IND#V =  
005901      $EDIT (W-IND-HOR-BAS.IND-HOR-BAS (W-CPT),PIC = '9999')  
006000      ENDFOR  
006100      ENDIF  
006200      ENDLOOP  
006202      DO SIGNE-DE-CN  
006300      ENDPROC  
006400      <<SIGNE-DE-CN>> PROCEDURE  
006500      SET W-I = 1  
006501      SET W-IND-CN-MIN = 0  
006600      SET W-CN-MIN = 0  
006700      LOOP VARYING W-CPT FROM 1 THRU (P-VAR#N - P-PLS-CTE#N)  
006800      FOR EACH PLS-CHB-R  
006900      WHERE (PLS# = P-CHX-PLS#) AND  
006901      (PLS-CLN# = $EDIT (W-CPT,PIC = '9999'))  
007000      IF PLS-CHB-R.PLS-CHB#V < - 0.00000009  
007100      IF PLS-CHB-R.PLS-CHB#V < W-CN-MIN  
007200      SET W-CN-MIN = PLS-CHB-R.PLS-CHB#V  
007300      SET W-IND-CN-MIN = $NUMBER (PLS-CHB-R.PLS-CHB-IND#V)  
007400      ENDIF
```

PROCEDURE:

UT005

VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
007500          SET W-I = W-I + 1
007600          ENDIF
007700          ENDFOR
007800          ENDOLOOP
007900          IF W-I = 1
007901              SET W-FIN = TRUE
007902              SET W-CHX-PLS# = P-CHX-PLS#
007903              SET W-PLS-VAR#N = P-PLS-VAR#N
007904              SET W-PLS-CTE#N = P-PLS-CTE#N
007905              SET W-ECR-VAR#N = P-ECR-VAR#N
007906              SET W-ARF-VAR#N = P-ARF-VAR#N
007907              IF P-PVC
007908                  CALL UT009 USING INPUT W-CHX-PLS#,W-PLS-VAR#N,W-PLS-CTE#N,
007909                      W-ECR-VAR#N,W-ARF-VAR#N
007910              RELEASE PROGRAM UT009
007911              ELSE
007912                  CALL UT010 USING INPUT W-CHX-PLS#,W-PLS-VAR#N,W-PLS-CTE#N,
007913                      W-ECR-VAR#N,W-ARF-VAR#N
007914              RELEASE PROGRAM UT010
007915              ENDIF
008100          ELSE
008200              DO CALCUL-PS
008300          ENDIF
008400          ENDPROC
008500          <<CALCUL-PS>> PROCEDURE
008600              LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
008700                  SET W-PS = 0
008800                  LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N
008900                      FOR EACH PLS-BAS-R
009000                          WHERE (PLS# = P-CHX-PLS#) AND
009100                              (PLS-CLN# = $EDIT (W-CPT-2,PIC = '9999')) AND
009200                              (PLS-LGN# = $EDIT (W-CPT,PIC = '9999'))
009300                              FOR EACH PLS-COE-R
009400                                  WHERE (PLS# = P-CHX-PLS#) AND
009500                                      (PLS-CLN# = $EDIT (W-IND-CN-MIN,PIC = '9999')) AND
009600                                      (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))
009601                                  SET W-PS = W-PS + PLS-BAS-R.PLS-BAS#V * PLS-COE-R.PLS-COE#V
009602                              ENDFOR
009603                          ENDFOR
009604                      ENDOLOOP
009700                  IF (P-ITE#N = 1) AND (P-PHASE NE 2)
009800                      FOR NEW PLS-PSS
009900                          SET PLS-PSS.PLS# = P-CHX-PLS#
010000                          SET PLS-PSS.PLS-LGN# = $EDIT (W-CPT,PIC = '9999')
010100                          SET PLS-PSS.PLS-PS#V = W-PS
010200                          SET PLS-PSS.PLS-PS-ETR#V =
010201                              $EDIT (W-IND-CN-MIN,PIC = '9999')
010300                      ENDFOR

```

PROCEDURE: UT005 VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
010400      ELSE
010500          FOR EACH PLS-PSS
010600              WHERE (PLS# = P-CHX-PLS#) AND
010700                  (PLS-LGN# = $EDIT ( W-CPT,PIC = '9999'))
010800                  SET PLS-PSS.PLS-PS#V = W-PS
010900                  SET PLS-PSS.PLS-PS-ETR#V =
010901                      $EDIT (W-IND-CN-MIN,PIC = '9999')
011000          ENDFOR
011100      ENDIF
011200      ENDLOOP
011204      DO SIGNE-PS
011300      ENDPROC
011400      <<SIGNE-PS>> PROCEDURE
011500          SET W-I = 1
011501          SET W-K = 0
011600          LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
011700              FOR EACH PLS-PSS-R
011800                  WHERE (PLS# = P-CHX-PLS#) AND(PLS-LGN# = $EDIT(W-CPT,PIC = '9999'))
011900                  IF PLS-PSS-R.PLS-PS#V > 0.00000111
011901                      SET W-K = W-K + 1
012000                  FOR EACH PLS-XBN-R
012100                      WHERE (PLS# = P-CHX-PLS#) AND
012200                          (PLS-CLN# = $EDIT (W-CPT,PIC = '9999'))
012300                          SET W-X = PLS-XBN-R.PLS-XB#V / PLS-PSS-R.PLS-PS#V
012400                          IF (W-CPT = 1) OR (W-K = 1)
012500                              SET W-PS-MIN = W-X
012502                          FOR EACH PLS-BAS-R
012503                              WHERE (PLS# = P-CHX-PLS#) AND
012504                                  (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND
012505                                  (PLS-LGN# = $EDIT (1,PIC = '9999'))
012507                              SET W-IND-PS-MIN =
012508                                  $NUMBER (PLS-BAS-R.PLS-BAS-IND#V)
012509                              SET W-BAS-IND = W-CPT
012511                          ENDFOR
012600                      ELSE
012700                          IF W-X < W-PS-MIN
012800                              SET W-PS-MIN = W-X
012802                          FOR EACH PLS-BAS-R
012803                              WHERE (PLS# = P-CHX-PLS#) AND
012804                                  (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND
012805                                  (PLS-LGN# = $EDIT (1,PIC = '9999'))
012806                              SET W-IND-PS-MIN =
012807                                  $NUMBER (PLS-BAS-R.PLS-BAS-IND#V)
012808                              SET W-BAS-IND = W-CPT
012810                          ENDFOR
012900                      ENDIF
013000                  ENDIF
013100              ENDFOR

```

PROCEDURE:

UT005 VERSION: 001 STATUS: TEST

```
-----  
SEQ DATA  
-----  
013200 SET W-I = W-I + 1  
013900 ENDIF  
014000 ENDFOR  
014100 ENDLOOP  
014200 IF W-I = 1  
014300 LIST 'SOLUTIONS OPTIMALES NON BORNEES'  
014301 SET W-FIN = TRUE  
014400 ELSE  
014500 DO CALCUL-JR  
014600 ENDFOR  
014700 ENDPROC  
014800 <<CALCUL-JR>> PROCEDURE  
014900 FOR EACH PLS-PSS-R  
015000 WHERE (PLS# = P-CHX-PLS#) AND  
015001 (PLS-LGN# = $EDIT (W-BAS-IND,PIC = '9999'))  
015100 SET W-JR.JR (W-BAS-IND) = 1 / PLS-PSS-R.PLS-PS#V  
015200 ENDFOR  
015300 LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
015400 FOR EACH PLS-PSS-R  
015500 WHERE (PLS# = P-CHX-PLS#) AND  
015600 (PLS-LGN# = $EDIT (W-CPT, PIC = '9999'))  
015601 IF W-CPT NE W-BAS-IND  
015700 SET W-JR.JR (W-CPT) = - PLS-PSS-R.PLS-PS#V *  
015701 W-JR.JR (W-BAS-IND)  
015703 ENDFOR  
015800 ENDFOR  
015900 ENDLOOP  
015902 DO BASE  
016000 ENDPROC  
016100 <<BASE>> PROCEDURE  
016200 LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
016300 FOR EACH PLS-BAS  
016400 WHERE (PLS# = P-CHX-PLS#) AND  
016500 (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND  
016600 (PLS-LGN# = $EDIT (W-BAS-IND,PIC = '9999'))  
016700 SET W-BAS.BAS (W-CPT) = PLS-BAS.PLS-BAS#V  
016800 SET PLS-BAS.PLS-BAS#V = W-JR.JR (W-BAS-IND) *  
016801 PLS-BAS.PLS-BAS#V  
016900 ENDFOR  
017000 ENDLOOP  
017100 LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
017200 LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N  
017300 FOR EACH PLS-BAS  
017400 WHERE (PLS# = P-CHX-PLS#) AND  
017500 (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND  
017600 (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))  
017700 IF W-CPT-2 NE W-BAS-IND  
017800 SET PLS-BAS.PLS-BAS#V = PLS-BAS.PLS-BAS#V +
```

PROCEDURE: UT005 VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
017801                                W-JR.JR (W-CPT-2) * W-BAS.BAS (W-CPT)  
017900                ENDIF  
018000                ENDFOR  
018100                ENDLOOP  
018200                ENDLOOP  
018300                LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
018400                FOR EACH PLS-BAS  
018500                WHERE (PLS# = P-CHX-PLS#) AND  
018600                (PLS-CLN# = $EDIT (W-BAS-IND,PIC = '9999')) AND  
018700                (PLS-LGN# = $EDIT (W-CPT,PIC = '9999'))  
018800                SET PLS-BAS.PLS-BAS-IND#V =  
018801                                $EDIT (W-IND-CN-MIN,PIC = '9999')  
018900                ENDFOR  
019000                ENDLOOP  
019001                SET W-IND-BAS.IND-BAS (W-BAS-IND) = W-IND-CN-MIN  
019002                DO CALCUL-XB  
019100                ENDPROC  
019200                <<CALCUL-XB>> PROCEDURE  
019300                LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
019400                FOR EACH PLS-XBN  
019401                WHERE (PLS# = P-CHX-PLS#) AND  
019500                (PLS-CLN# = $EDIT (W-CPT,PIC = '9999'))  
019600                SET PLS-XBN.PLS-XB#V = 0  
019700                LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N  
019800                FOR EACH PLS-TIN-R  
019900                WHERE (PLS# = P-CHX-PLS#) AND  
020000                (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))  
020100                FOR EACH PLS-BAS-R  
020200                WHERE (PLS# = P-CHX-PLS#) AND  
020300                (PLS-CLN# = $EDIT (W-CPT-2,PIC = '9999')) AND  
020400                (PLS-LGN# = $EDIT (W-CPT,PIC = '9999'))  
020500                SET PLS-XBN.PLS-XB#V = PLS-XBN.PLS-XB#V  
020501                                + PLS-BAS-R.PLS-BAS#V * PLS-TIN-R.PLS-TI#V  
020600                ENDFOR  
020700                ENDFOR  
020800                ENDLOOP  
020900                ENDFOR  
021000                ENDLOOP  
021001                LOOP VARYING W-CPT FROM 1 THRU (P-VAR#N - P-PLS-CTE#N)  
021002                IF W-IND-HOR-BAS.IND-HOR-BAS (W-CPT) = W-IND-CN-MIN  
021003                SET W-IND-HOR-BAS.IND-HOR-BAS (W-CPT) = W-IND-PS-MIN  
021005                ENDIF  
021007                ENDLOOP  
021200                ENDPROC
```

IDENTIFICATION: UT006

VERSION: 001 STATUS: TEST

PROGRAM UT006

Created 11/12/87 By ETUD04  
Last Modified 01/07/88 at 13:19 By ETUD04  
Last Compiled 01/07/88 at 13:25

Run Status: PRIVATE  
Short Description: methode des 2 phases 1  
Language: IDEAL Target Date \_\_ \_\_ \_\_ Actual Date \_\_ \_\_ \_\_

Description:

---

---

---

---

RESOURCES: UT006 VERSION: 001 STATUS: TEST

DATAVIEW	VER	PANEL	VER	REPORT	VER	PROGRAM	VER	SYS
PLS-BAS	0001					UT011	0001	
PLS-BAS-R	0001					UT012	0001	
PLS-CHB	0001							
PLS-COE	0001							
PLS-COE-R	0001							
PLS-CRL	0001							
PLS-PLS	0001							
PLS-TIN-R	0001							
PLS-XBN	0001							

PARAMETER:

UT006

VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	U	M	COMMENT/REDEF/DEP ON/COPY
000100	1	P-CHX-PLS#	X		3		I	D	
000200	1	P-PLS-VAR#N	N		4		I	D	
000300	1	P-PLS-CTE#N	N		4		I	D	
000400	1	P-VAR#N	N		4		I	D	
000401	1	P-ECR-BAS#N	N		3		I	D	
000402	1	P-IND-ECR#V					I	D	
000403	2	IND-ECR#V	N		4	300			
000404	1	P-PVC	F				I	D	
000500	1	P-PLS-Z0#V	N		7.8		U	D	
000600	1	P-PLS-OPT#V	X		3		U	D	
000700	1	P-ITE#	N		5		U	D	

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND. NAME, F=FLAG,  
V=VARIABLE, D=DATE

I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY

CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS

U (PARAMETER UPDATING): U=UPDATABLE, I=INPUT ONLY

M (PARAMETER MATCHING): BLANK OR D=DYNAMIC, I=IDENTICAL

REDEF=REDEFINITION, DEP ON= DEPENDING ON

WORKING DATA: UT006 VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	VAL/COMMENT/REDEF/DEP ON/COPY
000100	1	W-ECR-VAR#N	N		4		
000200	1	W-ARF-VAR#N	N		4		
000300	1	W-CPT	N		4		
000400	1	W-CPT-2	N		4		
000500	1	W-CPT-3	N		4		
000501	1	W-CPT-4	N		4		
000600	1	W-I	N		4		
000700	1	W-J	N		4		
000701	1	W-IND-ECR-BAS					
000800	2	IND-ECR-BAS	N		4	300	
000801	1	W-IND-ARF					
000900	2	IND-ARF	N		4	300	
000901	1	W-IND-BAS					
001000	2	IND-BAS	N		4	300	
001001	1	W-IND-ECR					
001100	2	IND-ECR	N		4	300	
001200	1	W-CHX-PLS#	X		3		
001300	1	W-PLS-VAR#N	N		4		
001400	1	W-PLS-CTE#N	N		4		
001500	1	W-PLS-ZO#V	N		7.8		
001600	1	W-PLS-OPT#V	X		3		
001601	1	W-PI					
001700	2	PI	N		7.8	300	
001800	1	W-UT007	F				
001900	1	W-PVC	F				
002000	1	W-VAR#N	N		4		
002100	1	W-ITE#N	N		5		
002200	1	W-FIN	F			FALSE	
002300	1	W-PHASE	N		1		
002400	1	W-JR	F				

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND.NAME, F=FLAG,  
V=VARIABLE, D=DATE  
I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
REDEF= REDEFINITION, DEP ON= DEPENDING ON

PROCEDURE:

UT006

VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
001000  <<VAR-ARF>> PROCEDURE
002000  SET W-ECR-VAR#N = P-VAR#N - P-PLS-VAR#N
003000  SET W-ARF-VAR#N = P-PLS-CTE#N - P-ECR-BAS#N
005000  LOOP VARYING W-CPT FROM 1 THRU P-ECR-BAS#N
006000  FOR EACH PLS-BAS-R
007000  WHERE (PLS# = P-CHX-PLS#) AND
008000  (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND
009000  (PLS-LGN# = $EDIT (1,PIC = '9999'))
010000  SET W-IND-ECR-BAS.IND-ECR-BAS (W-CPT) =
011000  $NUMBER (PLS-BAS-R.PLS-BAS-IND#V)
012000  ENDFOR
013000  ENDLOOP
014000  SET W-CPT-3 = 1
015000  LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
016000  SET W-I = 0
017000  LOOP VARYING W-CPT-2 FROM 1 THRU P-ECR-BAS#N
018000  IF (P-PLS-VAR#N + W-CPT) NE W-IND-ECR-BAS.IND-ECR-BAS (W-CPT-2)
019000  SET W-I = W-I + 1
020000  ENDIF
021000  ENDLOOP
022000  IF W-I = P-ECR-BAS#N
023000  SET W-IND-ARF.IND-ARF (W-CPT-3) = W-CPT + P-VAR#N
024000  SET W-CPT-3 = W-CPT-3 + 1
025000  ENDIF
026000  ENDLOOP
027000  LOOP VARYING W-CPT FROM 1 THRU W-ARF-VAR#N
028000  FOR NEW PLS-CRL
029000  SET PLS-CRL.PLS# = P-CHX-PLS#
030000  SET PLS-CRL.PLS-CLN# = $EDIT (W-IND-ARF.IND-ARF (W-CPT),
031000  PIC = '9999')
032000  SET PLS-CRL.PLS-CRL#V = 0
033000  ENDFOR
034000  LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N
035000  FOR NEW PLS-COE
036000  SET PLS-COE.PLS# = P-CHX-PLS#
037000  SET PLS-COE.PLS-CLN# = $EDIT (W-IND-ARF.IND-ARF (W-CPT),
038000  PIC = '9999')
039000  SET PLS-COE.PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999')
040000  IF W-CPT-2 = W-IND-ARF.IND-ARF (W-CPT) - P-VAR#N
041000  SET PLS-COE.PLS-COE#V = 1
042000  ELSE
043000  SET PLS-COE.PLS-COE#V = 0
044000  ENDIF
045000  ENDFOR
046000  ENDLOOP
047000  ENDLOOP
048000  LOOP VARYING W-CPT FROM 1 THRU W-ARF-VAR#N
049000  LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N

```

W

PROCEDURE: UT006 VERSION: 001 STATUS: TEST

```
-----
SEQ  DATA
-----
050000      FOR NEW PLS-BAS
051000          SET PLS-BAS.PLS# = P-CHX-PLS#
052000          SET W-CPT-4 = P-ECR-BAS#N + W-CPT
053000          SET PLS-BAS.PLS-CLN# = $EDIT (W-CPT-4, PIC = '9999')
054000          SET PLS-BAS.PLS-LGN# = $EDIT (W-CPT-2, PIC = '9999')
055000          IF W-CPT-2 = (W-IND-ARF.IND-ARF (W-CPT) - P-VAR#N)
056000              SET PLS-BAS.PLS-BAS#V = 1
057000          ELSE
058000              SET PLS-BAS.PLS-BAS#V = 0
059000          ENDIF
060000          SET PLS-BAS.PLS-BAS-IND#V =
061000              $EDIT (W-IND-ARF.IND-ARF (W-CPT), PIC = '9999')
062000      ENDFOR
063000      ENDLOOP
064000  ENDLOOP
065000  DO FO-PLB
066000  ENDPROC
067000  <<FO-PLB>> PROCEDURE
068000      SET P-PLS-ZO#V = 0
069000      LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
070000          FOR EACH PLS-TIN-R
071000              WHERE (PLS# = P-CHX-PLS#) AND
072000                  (PLS-LGN# = $EDIT (W-CPT, PIC = '9999'))
073000                  SET P-PLS-ZO#V = P-PLS-ZO#V - PLS-TIN-R.PLS-TI#V
074000          ENDFOR
075000      ENDLOOP
076000  DO CALCUL-CHB-PLB
077000  ENDPROC
078000  <<CALCUL-CHB-PLB>> PROCEDURE
079000      LOOP VARYING W-CPT FROM 1 THRU P-ECR-BAS#N
080000          SET W-IND-BAS.IND-BAS (W-CPT) = W-IND-ECR-BAS.IND-ECR-BAS
081000                                          (W-CPT)
082000      ENDLOOP
083000      LOOP VARYING W-CPT FROM 1 THRU W-ARF-VAR#N
084000          SET W-IND-BAS.IND-BAS (P-ECR-BAS#N + W-CPT) =
085000                                          W-IND-ARF.IND-ARF (W-CPT)
086000      ENDLOOP
087000      SET W-J = 1
088000      LOOP VARYING W-CPT FROM 1 THRU (P-VAR#N - P-ECR-BAS#N)
089000          SET W-I = 0
090000          LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N
091000              IF W-CPT NE W-IND-BAS.IND-BAS (W-CPT-2)
092000                  SET W-I = W-I + 1
093000          ENDIF
094000      ENDLOOP
095000      IF W-I = P-PLS-CTE#N
096000          FOR NEW PLS-CHB
097000              SET PLS-CHB.PLS# = P-CHX-PLS#
```

PROCEDURE:

UT006

VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
098000      SET PLS-CHB.PLS-CLN# = $EDIT (W-J,PIC = '9999')
099000      SET PLS-CHB.PLS-CHB#V = 0
100000      SET PLS-CHB.PLS-CHB-IND#V = $EDIT (W-CPT,PIC = '9999')
101000      LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N
102000          FOR EACH PLS-COE-R
103000              WHERE (PLS# = P-CHX-PLS#) AND
104000                  (PLS-CLN# = $EDIT (W-CPT, PIC = '9999')) AND
105000                  (PLS-LGN# = $EDIT (W-CPT-2, PIC = '9999'))
106000              SET PLS-CHB.PLS-CHB#V = PLS-CHB.PLS-CHB#V
107000                              - PLS-COE-R.PLS-COE#V
109000          ENDFOR
110000      ENDLOOP
111000      ENDFOR
112000          SET W-J = W-J + 1
113000      ENDIF
114000      ENDLOOP
115000      DO APPEL
116000      ENDPROC
117000      <<APPEL>> PROCEDURE
118000          SET W-CHX-PLS# = P-CHX-PLS#
119000          SET W-PLS-VAR#N = P-PLS-VAR#N
120000          SET W-PLS-CTE#N = P-PLS-CTE#N
121000          LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
122000              SET W-PI.PI (W-CPT) = 0
123000          ENDLOOP
124000          SET W-PLS-ZO#V = P-PLS-ZO#V
125000          SET W-PLS-OPT#V = P-PLS-OPT#V
126000          SET W-UT007 = FALSE
127000          SET W-PHASE = 1
128000          IF P-PVC
129000              SET W-PVC = TRUE
130000          ELSE
131000              SET W-PVC = FALSE
132000          ENDIF
133000          SET W-VAR#N = P-VAR#N + W-ARF-VAR#N
134000          SET W-ITE#N = 0
135000          SET W-FIN = FALSE
136000          SET W-JR = FALSE
137000          LOOP VARYING W-CPT FROM 1 THRU W-ECR-VAR#N
138000              SET W-IND-ECR.IND-ECR (W-CPT) = P-IND-ECR#V.IND-ECR#V (W-CPT)
139000          ENDLOOP
141000          CALL UT011 USING INPUT W-CHX-PLS#,W-PLS-VAR#N,W-PLS-CTE#N,W-VAR#N,
142000                                W-PHASE,W-ECR-VAR#N,W-IND-ECR,W-PVC,W-JR
143000                                UPDATE W-PLS-ZO#V,W-PLS-OPT#V,W-PI,W-UT007,
144000                                W-ITE#N,W-FIN,W-IND-BAS
145000                                :PHASE 1
147004          RELEASE PROGRAM UT011
149000          SET W-FIN = FALSE

```

PROCEDURE:

UT006

VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
149001   SET W-VAR#N = P-PLS-VAR#N + W-ECR-VAR#N + W-ARF-VAR#N  
149002   SET W-ITE#N = 0  
150000   CALL UT012 USING INPUT W-CHX-PLS#,W-PLS-VAR#N,W-PLS-CTE#N,W-VAR#N,  
151000                                     W-ECR-VAR#N,W-IND-ARF,W-IND-ECR,W-PVC,  
151001                                     W-ARF-VAR#N  
152000                                     UPDATE W-IND-BAS,W-PLS-ZO#V,W-PLS-OPT#V,W-ITE#N,W-FIN  
153000                                     :PHASE 2  
155000   RELEASE PROGRAM UT012  
167000   ENDPROC
```

IDENTIFICATION: UT008

VERSION: 001 STATUS: TEST

PROGRAM UT008

Created 11/26/87 By ETUD04  
Last Modified 12/11/87 at 11:04 By ETUD04  
Last Compiled 12/11/87 at 12:20

Run Status: PRIVATE  
Short Description: reinversion de la base  
Language: IDEAL Target Date \_\_ \_\_ \_\_ Actual Date \_\_ \_\_ \_\_

Description:

---

---

---

---

RESOURCES:

UT008                    VERSION: 001 STATUS: TEST

DATAVIEW	VER	PANEL	VER	REPORT	VER	PROGRAM	VER	SYS
PLS-BAS	0001							
PLS-BAS-R	0001							
PLS-COE-R	0001							

PARAMETER:

UT008

VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	U	M	COMMENT/REDEF/DEP ON/COPY
000100	1	P-CHX-PLS#	X		3		I	D	
000200	1	P-PLS-CTE#N	N		4		I	D	

LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND. NAME, F=FLAG,  
V=VARIABLE, D=DATE

I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY

CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS

U (PARAMETER UPDATING): U=UPDATABLE, I=INPUT ONLY

M (PARAMETER MATCHING): BLANK OR D=DYNAMIC, I=IDENTICAL

REDEF=REDEFINITION, DEP ON= DEPENDING ON

WORKING DATA: UT008 VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	VAL/COMMENT/REDEF/DEP ON/COPY
000500	1	W-CPT	N		4		
000600	1	W-CPT-2	N		4		
000601	1	W-CPT-3	N		4		
000602	1	W-CPT-4	N		4		
000603	1	W-BAS-IND#V					
000700	2	BAS-IND#V	N		4	300	
000701	1	W-X					
000900	2	X	N		7.8	300	
000901	1	W-B					
001000	2	B	N		7.8	300	
001100	1	W-BAS					
001200	2	BAS#V	N		7.8	300	
001300	1	W-CP4					
001400	2	CP4#V	N		7.8	300	
001500	1	W-ELT					
001600	2	ELT#V	N		7.8	300	

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND.NAME, F=FLAG,  
V=VARIABLE, D=DATE

I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY

CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS

REDEF= REDEFINITION, DEP ON= DEPENDING ON

RESOURCES:

UT010                    VERSION: 001 STATUS: TEST

DATAVIEW	VER	PANEL	VER	REPORT	VER	PROGRAM	VER	SYS
PLS-BAS	0001	PN0218	0001					
PLS-BAS-R	0001	PN0219	0001					
PLS-CHB	0001							
PLS-COE	0001							
PLS-CRL	0001							
PLS-PLS-R	0001							
PLS-PSS	0001							
PLS-XBN	0001							
PLS-XBN-R	0001							

PARAMETER: UT010 VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	U	M	COMMENT/REDEF/DEP ON/COPY
000100	1	P-CHX-PLS#	X		3		I	D	
000200	1	P-PLS-VAR#N	N		4		I	D	
000300	1	P-PLS-CTE#N	N		4		I	D	
000400	1	P-ECR-VAR#N	N		4		I	D	
000500	1	P-ARF-VAR#N	N		4		I	D	

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND. NAME, F=FLAG,  
V=VARIABLE, D=DATE  
I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
U (PARAMETER UPDATING): U=UPDATABLE, I=INPUT ONLY  
M (PARAMETER MATCHING): BLANK OR D=DYNAMIC, I=IDENTICAL  
REDEF=REDEFINITION, DEP ON= DEPENDING ON

WORKING DATA: UT010 VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	VAL/COMMENT/REDEF/DEP ON/COPY
000100	1	W-OCC	N			4	
000300	1	W-ITE#N	N			4	
000500	1	W-MIN	N			4	
000600	1	W-CPT	N			4	
000700	1	W-CPT-2	N			4	
000701	1	W-CPT-3	N			4	

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND.NAME, F=FLAG,  
V=VARIABLE, D=DATE  
I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
REDEF= REDEFINITION, DEP ON= DEPENDING ON

PROCEDURE: UT010 VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
000100  <<PN0218>> PROCEDURE
000200      REFRESH PN0218
000300      SET PN0218.PLS# = P-CHX-PLS#
000400      SET PN0218.VAR#N = P-PLS-VAR#N
000500      SET PN0218.CTE#N = P-PLS-CTE#N
000600      SET PN0218.ECR-VAR#N = P-ECR-VAR#N
000700      SET PN0218.ARF-VAR#N = P-ARF-VAR#N
000800      FOR EACH PLS-PLS-R
000900          WHERE PLS# = P-CHX-PLS#
000901          LOOP VARYING W-CPT FROM 1 THRU 8
001000              SET PN0218.PLS#L (W-CPT) = $SUBSTR (PLS-PLS-R.PLS#L,
001001                  START = 50 * (W-CPT - 1) + 1)
001002          ENDLOOP
001100      ENDFOR
001200      LOOP
001300          TRANSMIT PN0218
001400          UNTIL $PF6
001500      ENDLOOP
001600      DO PN0219
001700  ENDPROC
005000  <<PN0219>> PROCEDURE
005100      SET W-OCC = $PANEL-GROUP-OCCURS (PN0219)
005200      IF P-PLS-CTE#N <= W-OCC
005300          SET W-ITE#N = 1
005301          SET W-MIN = P-PLS-CTE#N
005400      ELSE
005500          SET W-ITE#N = P-PLS-CTE#N / W-OCC
005600          IF $REMAINDER (P-PLS-CTE#N,DIV = W-OCC) NE 0
005700              SET W-ITE#N = W-ITE#N + 1
005800          ENDIF
005801          SET W-MIN = W-OCC
005900      ENDIF
006000      LOOP VARYING W-CPT FROM 1 THRU W-ITE#N
006100          REFRESH PN0219
006200          SET PN0219.PLS# = P-CHX-PLS#
006300          LOOP VARYING W-CPT-2 FROM 1 THRU W-MIN
006400              FOR EACH PLS-XBN-R
006500                  WHERE (PLS# = P-CHX-PLS#) AND
006600                      (PLS-CLN# = $EDIT (W-CPT-2,PIC = '9999'))
006700                  IF W-CPT-2 <= (W-OCC / 2)
006800                      SET PN0219.XB#V (W-CPT-2) = PLS-XBN-R.PLS-XB#V
006900                  FOR EACH PLS-BAS-R
007000                      WHERE (PLS# = P-CHX-PLS#) AND
007100                          (PLS-CLN# = $EDIT (W-CPT-2,PIC = '9999')) AND
007200                          (PLS-LGN# = $EDIT (1,PIC = '9999'))
007300                      SET PN0219.CPT (W-CPT-2) =
007301                          $NUMBER (PLS-BAS-R.PLS-BAS-IND#V)
007400              ENDFOR
-----

```

PROCEDURE:

UT010

VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
007500          ELSE  
007600          SET PNO219.XB2#V (W-CPT-2) = PLS-XBN-R.PLS-XB#V  
007700          FOR EACH PLS-BAS-R  
007800          WHERE (PLS# = P-CHX-PLS# ) AND  
007900          (PLS-CLN# = $EDIT (W-CPT-2,PIC = '9999')) AND  
008000          (PLS-LGN# = $EDIT (1,PIC = '9999'))  
008100          SET PNO219.CP2 (W-CPT-2) =  
008101          $NUMBER (PLS-BAS-R.PLS-BAS-IND#V)  
008200          ENDFOR  
008300          ENDIF  
008400          ENDFOR  
008500          ENDLOOP  
008501          TRANSMIT PNO219  
008502          UNTIL $PF6  
008600          ENDLOOP  
008601          DO SUPP-DVW  
008700          ENDPROC  
008800          <<SUPP-DVW>> PROCEDURE  
008900          FOR EACH PLS-BAS  
009000          WHERE PLS# = P-CHX-PLS#  
009100          DELETE PLS-BAS  
009200          ENDFOR  
009300          FOR EACH PLS-PSS  
009400          WHERE PLS# = P-CHX-PLS#  
009500          DELETE PLS-PSS  
009600          ENDFOR  
009700          FOR EACH PLS-CHB  
009800          WHERE PLS# = P-CHX-PLS#  
009900          DELETE PLS-CHB  
010000          ENDFOR  
010100          FOR EACH PLS-XBN  
010200          WHERE PLS# = P-CHX-PLS#  
010300          DELETE PLS-XBN  
010400          ENDFOR  
010600          LOOP VARYING W-CPT FROM 1 THRU (P-ECR-VAR#N + P-ARF-VAR#N)  
010601          SET W-CPT-2 = P-PLS-VAR#N + W-CPT  
010700          FOR EACH PLS-CRL  
010800          WHERE (PLS# = P-CHX-PLS#) AND  
010900          (PLS-CLN# = $EDIT (W-CPT-2,PIC = '9999'))  
011000          DELETE PLS-CRL  
011100          ENDFOR  
011200          FOR EACH PLS-COE  
011300          WHERE (PLS# = P-CHX-PLS#) AND  
011400          (PLS-CLN# = $EDIT (W-CPT-2,PIC = '9999'))  
011500          DELETE PLS-COE  
011600          ENDFOR  
011700          ENDLOOP  
011800          ENDPROC
```

IDENTIFICATION: UT011

VERSION: 001 STATUS: TEST

PROGRAM UT011

Created 12/16/87 By ETUD04  
Last Modified 01/06/88 at 11:11 By ETUD04  
Last Compiled 01/06/88 at 11:23

Run Status: PRIVATE  
Short Description: met. primale phase 2.1  
Language: IDEAL Target Date \_\_\_ \_\_\_ \_\_\_ Actual Date \_\_\_ \_\_\_ \_\_\_

Description:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

RESOURCES:

UT011                    VERSION: 001 STATUS: TEST

DATAVIEW	VER	PANEL	VER	REPORT	VER	PROGRAM	VER	SYS
PLS-BAS	0001					UT008	0001	
PLS-BAS-R	0001					UT009	0001	
PLS-CHB	0001					UT010	0001	
PLS-CHB-R	0001							
PLS-COE-R	0001							
PLS-CRL-R	0001							
PLS-PLS	0001							
PLS-PSS	0001							
PLS-PSS-R	0001							
PLS-TIN-R	0001							
PLS-XBN	0001							
PLS-XBN-R	0001							

PARAMETER:

UT011

VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	U	M	COMMENT/REDEF/DEP ON/COPY
000100	1	P-CHX-PLS#	X		3		I	D	
000200	1	P-PLS-VAR#N	N		4		I	D	
000300	1	P-PLS-CTE#N	N		4		I	D	
000301	1	P-VAR#N	N		4		I	D	
000302	1	P-PHASE	N		1		I	D	
000303	1	P-ECR-VAR#N	N		4		I	D	
000304	1	P-IND-ECR#V					I	D	
000305	2	IND-ECR#V	N		4	300			
000306	1	P-PVC	F				I	D	
000307	1	P-JR	F				I	D	
000400	1	P-PLS-ZO#V	N		7.8		U	D	
000500	1	P-PLS-OPT#V	X		3		U	D	
000501	1	P-PI					U	D	
000700	2	PI	N		7.8	300			
000800	1	P-UT007	F				U	D	
000900	1	P-ITE#N	N		5		U	D	
001000	1	P-FIN	F				U	D	
001100	1	P-IND-BAS					U	D	
001200	2	IND-BAS	N		4	300			

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND. NAME, F=FLAG,  
V=VARIABLE, D=DATE  
I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
U (PARAMETER UPDATING): U=UPDATABLE, I=INPUT ONLY  
M (PARAMETER MATCHING): BLANK OR D=DYNAMIC, I=IDENTICAL  
REDEF=REDEFINITION, DEP ON= DEPENDING ON

WORKING DATA: UT011 VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	VAL/COMMENT/REDEF/DEP	ON/COPY
000001	1	W-Y	N		7.8			
000002	1	W-BAS-IND	N		4			
000100	1	W-CPT	N		4			
000200	1	W-CPT-2	N		4			
000300	1	W-K	N		4			
000400	1	W-I	N		4			
000401	1	W-PI						
000500	2	PI	N		7.8	300	0	
000600	1	W-HOR-BAS	F					
000601	1	W-IND-HOR-BAS						
000700	2	IND-HOR-BAS	N		4	3000		
000800	1	W-PN	N		7.8			
000900	1	W-CN#V	N		7.8			
000901	1	W-IND						
001000	2	IND	N		4	3000		
001100	1	W-CN-MIN	N		7.8			
001200	1	W-IND-CN-MIN	N		4			
001300	1	W-PS	N		7.8		0	
001400	1	W-X	N		7.8			
001500	1	W-PS-MIN	N		7.8			
001600	1	W-IND-PS-MIN	N		4			
001601	1	W-JR						
001700	2	JR	N		7.8	300	0	
001701	1	W-BAS						
001800	2	BAS	N		7.8	300	0	
001900	1	W-CN	F					
002000	1	W-FIN	F					
002001	1	W-IND-BAS						
002100	2	IND-BAS	N		4	300		
002200	1	W-ECR-VAR#N	N		4			
002300	1	W-ARF-VAR#N	N		4			
002400	1	W-CHX-PLS#	X		3			
002500	1	W-PLS-CTE#N	N		4			
002501	1	W-PLS-VAR#N	N		4			
002600	1	W-RET	N		5			
002700	1	W-J	N		4			
003200	1	W-CRL						
003300	2	CRL	N		7.8	3000		

LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND.NAME, F=FLAG,  
 V=VARIABLE, D=DATE  
 I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
 CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
 REDEF= REDEFINITION, DEP ON= DEPENDING ON

PROCEDURE:

UT011

VERSION: 001 STATUS: TEST

```
-----
SEQ  DATA
-----
000001 <<UT011>> PROCEDURE
000103   IF P-PHASE = 0
000104     LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
000105       SET W-IND-BAS.IND-BAS (W-CPT) = P-IND-ECR#V.IND-ECR#V
000106         (W-CPT)
000107     ENDLOOP
000108   ELSE
000109     LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
000110       FOR EACH PLS-BAS-R
000111         WHERE (PLS# = P-CHX-PLS#) AND
000112           (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND
000113           (PLS-LGN# = $EDIT ( 1,PIC = '9999'))
000114         SET W-IND-BAS.IND-BAS (W-CPT) =
000115           $NUMBER (PLS-BAS-R.PLS-BAS-IND#V)
000116       ENDFOR
000117     ENDLOOP
000118   ENDIF
000122   IF NOT P-JR
000123     IF P-FIN
000124       SET W-FIN = TRUE
000125     ELSE
000126       SET W-FIN = FALSE
000127     ENDIF
000128   ENDIF
000129   SET W-CHX-PLS# = P-CHX-PLS#
000130   SET W-PLS-CTE#N = P-PLS-CTE#N
000131   SET W-ARF-VAR#N = P-VAR#N - P-PLS-VAR#N - P-ECR-VAR#N
000132   LOOP VARYING W-CPT FROM 1 THRU P-PLS-VAR#N
000133     SET W-CRL.CRL (W-CPT) = 0
000134   ENDLOOP
000135   LOOP VARYING W-CPT FROM (P-PLS-VAR#N + 1) THRU (P-PLS-VAR#N +
000136     P-ECR-VAR#N + W-ARF-VAR#N)
000137     SET W-I = 0
000138     LOOP VARYING W-CPT-2 FROM 1 THRU P-ECR-VAR#N
000139       IF W-CPT NE P-IND-ECR#V.IND-ECR#V (W-CPT-2)
000140         SET W-I = W-I + 1
000141       ENDIF
000142     ENDLOOP
000143   IF W-I = P-ECR-VAR#N
000144     SET W-CRL.CRL (W-CPT) = 1
000145   ELSE
000146     SET W-CRL.CRL (W-CPT) = 0
000147   ENDIF
000148   ENDLOOP
000149   IF P-UT007
000150     DO CALCUL-PI
000151   ENDIF
000152   IF P-JR
```

PROCEDURE:

UT011

VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
000162      DO CALCUL-JR  
000163      ENDIF  
000164      LOOP UNTIL W-FIN  
000165          SET P-ITE#N = P-ITE#N + 1  
000170          SET W-RET = $REMAINDER (P-ITE#N, DIV = 20)  
000171          IF W-RET = 0  
000172              CALL UT008 USING INPUT W-CHX-PLS#,W-PLS-CTE#N  
000173              RELEASE PROGRAM UT008  
000174          ENDIF  
000175          IF (P-ITE#N = 1) AND NOT P-UT007  
000176              SET W-CN = FALSE  
000177              DO IND-HORS-BASE  
000178              DO SIGNE-DE-CN  
000179          ELSE  
000180              DO CALCUL-PI  
000181          ENDIF  
000182      ENDLOOP  
000183  ENDPROC  
000184  <<CALCUL-PI>> PROCEDURE  
000200      LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
000201          SET W-PI.PI (W-CPT) = 0  
000300          LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N  
000301              SET W-I = W-IND-BAS.IND-BAS (W-CPT-2)  
000500              FOR EACH PLS-BAS-R  
000600                  WHERE (PLS# = P-CHX-PLS#) AND  
000700                      (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND  
000800                      (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))  
000900                  SET W-PI.PI(W-CPT) = W-PI.PI(W-CPT) +  
000901                      W-CRL.CRL (W-I) * PLS-BAS-R.PLS-BAS#V  
001000              ENDFOR  
001200          ENDLOOP  
001300      ENDLOOP  
001301      IF P-ITE#N NE 1  
001302          SET P-PLS-ZO#V = 0  
001304          LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
001305              FOR EACH PLS-TIN-R  
001306                  WHERE (PLS# = P-CHX-PLS#) AND  
001307                      (PLS-LGN# = $EDIT (W-CPT,PIC = '9999'))  
001308                  SET P-PLS-ZO#V = P-PLS-ZO#V - W-PI.PI (W-CPT)  
001309                      * PLS-TIN-R.PLS-TI#V  
001311              ENDFOR  
001312          ENDLOOP  
001314      ENDIF  
001315      SET W-CN = TRUE  
001316      IF P-PHASE NE 2  
001317          DO IND-HORS-BASE  
001318      ELSE  
001319          SET W-FIN = TRUE
```

PROCEDURE: UT011 VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
001320    ENDIF  
001400    ENDPROC  
001500    <<IND-HORS-BASE>> PROCEDURE  
001502    IF P-ITE#N = 1  
001600        SET W-J = 1  
001700        LOOP VARYING W-CPT FROM 1 THRU P-VAR#N  
001800            SET W-HOR-BAS = TRUE  
001900            SET W-K = 1  
002000            LOOP WHILE (W-K <= P-PLS-CTE#N) AND (W-HOR-BAS)  
002100                FOR EACH PLS-BAS-R  
002200                    WHERE (PLS# = P-CHX-PLS#) AND  
002201                        (PLS-CLN# = $EDIT (W-K,PIC = '9999')) AND  
002202                        (PLS-LGN# = $EDIT (1,PIC = '9999'))  
002300                        IF W-CPT = $NUMBER (PLS-BAS-R.PLS-BAS-IND#V)  
002400                            SET W-HOR-BAS = FALSE  
002500                        ELSE  
002600                            SET W-K = W-K + 1  
002700                        ENDIF  
002800                    ENDFOR  
002900                ENDOLOOP  
003000            IF (W-K > P-PLS-CTE#N) AND W-HOR-BAS  
003100                SET W-IND-HOR-BAS.IND-HOR-BAS (W-J) = W-CPT  
003200                SET W-J = W-J + 1  
003300            ENDIF  
003400        ENDOLOOP  
003401    ENDIF  
003402    IF W-CN DO CALCUL-CN ENDIF  
003500    ENDPROC  
003600    <<CALCUL-CN>> PROCEDURE  
003700    LOOP VARYING W-CPT FROM 1 THRU (P-VAR#N - P-PLS-CTE#N)  
003701        SET W-PN = 0  
003706        SET W-CN#V = W-CRL.CRL (W-CPT)  
003800        LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N  
003900            FOR EACH PLS-COE-R  
004000                WHERE (PLS# = P-CHX-PLS#) AND  
004100                    (PLS-CLN# = $EDIT (W-IND-HOR-BAS.IND-HOR-BAS (W-CPT),  
004101                        PIC = '9999'))  
004200                    AND (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))  
004300                    SET W-PN = W-PN + W-PI.PI (W-CPT-2)* PLS-COE-R.PLS-COE#V  
004500            ENDFOR  
004600        ENDOLOOP  
004700        SET W-CN#V = W-CN#V - W-PN  
004701        FOR EACH PLS-CHB  
004702            WHERE (PLS# = P-CHX-PLS#) AND  
004703                (PLS-CLN# = $EDIT (W-CPT,PIC = '9999'))  
005800                SET PLS-CHB.PLS-CHB#V = W-CN#V  
005900                SET PLS-CHB.PLS-CHB-IND#V =  
005901                    $EDIT (W-IND-HOR-BAS.IND-HOR-BAS (W-CPT),PIC = '9999')
```

PROCEDURE:

UT011                   VERSION: 001 STATUS: TEST

```
-----  
SEQ     DATA  
-----  
005903            ENDFOR  
006200            ENDLOOP  
006201            IF NOT P-UT007  
006202            DO SIGNE-DE-CN  
006203            ELSE  
006204            SET W-FIN = TRUE  
006205            ENDIF  
006300    ENDPROC  
006400    <<SIGNE-DE-CN>> PROCEDURE  
006500            SET W-I = 1  
006501            SET W-IND-CN-MIN = 0  
006600            SET W-CN-MIN = 0  
006700            LOOP VARYING W-CPT FROM 1 THRU (P-VAR#N - P-PLS-CTE#N)  
006701            FOR EACH PLS-CHB-R  
006702            WHERE (PLS# = P-CHX-PLS#) AND  
006703            (PLS-CLN# = $EDIT (W-CPT,PIC = '9999'))  
007000            IF PLS-CHB-R.PLS-CHB#V < - 0.00000009  
007100            IF PLS-CHB-R.PLS-CHB#V < W-CN-MIN  
007200            SET W-CN-MIN = PLS-CHB-R.PLS-CHB#V  
007300            SET W-IND-CN-MIN = $NUMBER (PLS-CHB-R.PLS-CHB-IND#V)  
007400            ENDIF  
007500            SET W-I = W-I + 1  
007600            ENDIF  
007601            ENDFOR  
007800            ENDLOOP  
007900            IF W-I = 1  
007901            SET W-FIN = TRUE  
007902            SET W-CHX-PLS# = P-CHX-PLS#  
007903            SET W-PLS-VAR#N = P-PLS-VAR#N  
007904            SET W-PLS-CTE#N = P-PLS-CTE#N  
007905            SET W-ECR-VAR#N = P-ECR-VAR#N  
007906            SET W-ARF-VAR#N = P-VAR#N - P-PLS-VAR#N - P-ECR-VAR#N  
007907            IF (P-PVC) AND P-PHASE = 2  
007908            CALL UT009 USING INPUT W-CHX-PLS#,W-PLS-VAR#N,W-PLS-CTE#N,  
007909            W-ECR-VAR#N,W-ARF-VAR#N  
007910            RELEASE PROGRAM UT009  
007911            ELSE  
007912            IF P-PHASE = 2  
007913            CALL UT010 USING INPUT W-CHX-PLS#,W-PLS-VAR#N,W-PLS-CTE#N,  
007914            W-ECR-VAR#N,W-ARF-VAR#N  
007915            RELEASE PROGRAM UT010  
007916            ENDIF  
007917            ENDIF  
008100            ELSE  
008200            DO CALCUL-PS  
008300            ENDIF  
008400    ENDPROC  
008500    <<CALCUL-PS>> PROCEDURE
```

PROCEDURE: UT011 VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
008600    LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
008700      SET W-PS = 0
008800      LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N
008900        FOR EACH PLS-BAS-R
009000          WHERE (PLS# = P-CHX-PLS#) AND
009100            (PLS-CLN# = $EDIT (W-CPT-2,PIC = '9999')) AND
009200            (PLS-LGN# = $EDIT (W-CPT,PIC = '9999'))
009300          FOR EACH PLS-COE-R
009400            WHERE (PLS# = P-CHX-PLS#) AND
009500              (PLS-CLN# = $EDIT (W-IND-CN-MIN,PIC = '9999')) AND
009600              (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))
009601          SET W-PS = W-PS + PLS-BAS-R.PLS-BAS#V * PLS-COE-R.PLS-COE#V
009602        ENDFOR
009603      ENDFOR
009604    ENDLOOP
009700    IF P-ITE#N = 1
009800      FOR NEW PLS-PSS
009900        SET PLS-PSS.PLS# = P-CHX-PLS#
010000        SET PLS-PSS.PLS-LGN# = $EDIT (W-CPT,PIC = '9999')
010100        SET PLS-PSS.PLS-PS#V = W-PS
010200        SET PLS-PSS.PLS-PS-ETR#V =
010201          $EDIT (W-IND-CN-MIN,PIC = '9999')
010300      ENDFOR
010400    ELSE
010500      FOR EACH PLS-PSS
010600        WHERE (PLS# = P-CHX-PLS#) AND
010700          (PLS-LGN# = $EDIT ( W-CPT,PIC = '9999'))
010800        SET PLS-PSS.PLS-PS#V = W-PS
010900        SET PLS-PSS.PLS-PS-ETR#V =
010901          $EDIT (W-IND-CN-MIN,PIC = '9999')
011000      ENDFOR
011100    ENDIF
011200  ENDLOOP
011201  DO SIGNE-PS
011300 ENDPROC
011400 <<SIGNE-PS>> PROCEDURE
011500   SET W-I = 1
011501   SET W-K = 0
011600   LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
011700     FOR EACH PLS-PSS-R
011800       WHERE (PLS# = P-CHX-PLS#) AND(PLS-LGN# = $EDIT(W-CPT,PIC = '9999'))
011900       IF PLS-PSS-R.PLS-PS#V > 0.00000111
011901         SET W-K = W-K + 1
012000       FOR EACH PLS-XBN-R
012100         WHERE (PLS# = P-CHX-PLS#) AND
012200           (PLS-CLN# = $EDIT (W-CPT,PIC = '9999'))
012300         SET W-X = PLS-XBN-R.PLS-XB#V / PLS-PSS-R.PLS-PS#V
012400         IF (W-CPT = 1) OR (W-K = 1)

```

PROCEDURE:

UT011

VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
012500      SET W-PS-MIN = W-X
012501      FOR EACH PLS-BAS-R
012502      WHERE (PLS# = P-CHX-PLS#) AND
012503      (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND
012504      (PLS-LGN# = $EDIT (1,PIC = '9999'))
012505      SET W-IND-PS-MIN =
012506      $NUMBER (PLS-BAS-R.PLS-BAS-IND#V)
012507      SET W-BAS-IND = W-CPT
012508      ENDFOR
012600      ELSE
012700      IF W-X < W-PS-MIN
012800      SET W-PS-MIN = W-X
012801      FOR EACH PLS-BAS-R
012802      WHERE (PLS# = P-CHX-PLS#) AND
012803      (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND
012804      (PLS-LGN# = $EDIT (1,PIC = '9999'))
012805      SET W-IND-PS-MIN =
012806      $NUMBER (PLS-BAS-R.PLS-BAS-IND#V)
012807      SET W-BAS-IND = W-CPT
012808      ENDFOR
012900      ENDIF
013000      ENDFOR
013100      SET W-I = W-I + 1
013200      ENDFOR
013900      ENDFOR
014000      ENDFOR
014100      ENDFOR
014200      IF W-I = 1
014300      LIST 'SOLUTIONS OPTIMALES NON BORNEES'
014301      SET W-FIN = TRUE
014400      ELSE
014500      DO CALCUL-JR
014600      ENDFOR
014700      ENDFOR
014800      <<CALCUL-JR>> PROCEDURE
014900      FOR EACH PLS-PSS-R
015000      WHERE (PLS# = P-CHX-PLS#) AND
015001      (PLS-LGN# = $EDIT (W-BAS-IND,PIC = '9999'))
015100      SET W-JR.JR (W-BAS-IND) = 1 / PLS-PSS-R.PLS-PS#V
015200      ENDFOR
015300      LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
015400      FOR EACH PLS-PSS-R
015500      WHERE (PLS# = P-CHX-PLS#) AND
015600      (PLS-LGN# = $EDIT (W-CPT, PIC = '9999'))
015601      IF W-CPT NE W-BAS-IND
015700      SET W-JR.JR (W-CPT) = - PLS-PSS-R.PLS-PS#V *
015701      W-JR.JR (W-BAS-IND)
015703      ENDFOR
-----

```

PROCEDURE:

UT011

VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
015800      ENDFOR  
015900      ENDLOOP  
015901      IF NOT P-JR  
015902          DO BASE  
015903      ELSE  
015904          SET W-FIN = TRUE  
015905      ENDIF  
016000      ENDPROC  
016100      <<BASE>> PROCEDURE  
016200      LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
016300          FOR EACH PLS-BAS  
016400              WHERE (PLS# = P-CHX-PLS#) AND  
016500                  (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND  
016600                  (PLS-LGN# = $EDIT (W-BAS-IND,PIC = '9999'))  
016700                  SET W-BAS.BAS (W-CPT) = PLS-BAS.PLS-BAS#V  
016800                  SET PLS-BAS.PLS-BAS#V = W-JR.JR (W-BAS-IND) *  
016801                                          PLS-BAS.PLS-BAS#V  
016900          ENDFOR  
017000      ENDLOOP  
017100      LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
017200          LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N  
017300              FOR EACH PLS-BAS  
017400                  WHERE (PLS# = P-CHX-PLS#) AND  
017500                      (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND  
017600                      (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))  
017700                  IF W-CPT-2 NE W-BAS-IND  
017800                      SET PLS-BAS.PLS-BAS#V = PLS-BAS.PLS-BAS#V +  
017801                          W-JR.JR (W-CPT-2) * W-BAS.BAS (W-CPT)  
017900                  ENDIF  
018000          ENDFOR  
018100      ENDLOOP  
018200      ENDLOOP  
018300      LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
018400          FOR EACH PLS-BAS  
018500              WHERE (PLS# = P-CHX-PLS#) AND  
018600                  (PLS-CLN# = $EDIT (W-BAS-IND,PIC = '9999')) AND  
018700                  (PLS-LGN# = $EDIT (W-CPT,PIC = '9999'))  
018800                  SET PLS-BAS.PLS-BAS-IND#V =  
018801                      $EDIT (W-IND-CN-MIN,PIC = '9999')  
018900          ENDFOR  
019000      ENDLOOP  
019001      LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
019002          IF W-IND-BAS.IND-BAS (W-CPT) = W-IND-PS-MIN  
019003              SET W-IND-BAS.IND-BAS (W-CPT) = W-IND-CN-MIN  
019005          ENDIF  
019006      ENDLOOP  
019007      DO CALCUL-XB  
019100      ENDPROC
```

PROCEDURE:

UT011                   VERSION: 001 STATUS: TEST

```
-----  
SEQ    DATA  
-----  
019200 <<CALCUL-XB>> PROCEDURE  
019300    LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
019400      FOR EACH PLS-XBN  
019401        WHERE (PLS# = P-CHX-PLS#) AND  
019500          (PLS-CLN# = $EDIT (W-CPT,PIC = '9999'))  
019600          SET PLS-XBN.PLS-XB#V = 0  
019700      LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N  
019800        FOR EACH PLS-TIN-R  
019900          WHERE (PLS# = P-CHX-PLS#) AND  
020000            (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))  
020100            FOR EACH PLS-BAS-R  
020200              WHERE (PLS# = P-CHX-PLS#) AND  
020300                (PLS-CLN# = $EDIT (W-CPT-2,PIC = '9999')) AND  
020400                (PLS-LGN# = $EDIT (W-CPT,PIC = '9999'))  
020500                SET PLS-XBN.PLS-XB#V = PLS-XBN.PLS-XB#V  
020501                    + PLS-BAS-R.PLS-BAS#V * PLS-TIN-R.PLS-TI#V  
020600            ENDFOR  
020700          ENDFOR  
020800        ENDLOOP  
020900      ENDFOR  
021000    ENDLOOP  
021001    LOOP VARYING W-CPT FROM 1 THRU (P-VAR#N - P-PLS-CTE#N)  
021002      IF W-IND-HOR-BAS.IND-HOR-BAS (W-CPT) = W-IND-CN-MIN  
021003        SET W-IND-HOR-BAS.IND-HOR-BAS (W-CPT) = W-IND-PS-MIN  
021005      ENDIF  
021007    ENDLOOP  
021200 ENDPROC
```

IDENTIFICATION: UT012

VERSION: 001 STATUS: TEST

PROGRAM UT012

Created 01/04/88 By ETUD04  
Last Modified 01/06/88 at 11:12 By ETUD04  
Last Compiled 01/06/88 at 11:25

Run Status: PRIVATE  
Short Description: met. primale phase 2.2  
Language: IDEAL Target Date \_\_\_ \_\_\_ \_\_\_ Actual Date \_\_\_ \_\_\_ \_\_\_

Description:

---

---

---

---

RESOURCES:

UT012                    VERSION: 001 STATUS: TEST

DATAVIEW	VER	PANEL	VER	REPORT	VER	PROGRAM	VER	SYS
PLS-BAS	0001					UT005	0001	
PLS-BAS-R	0001					UT008	0001	
PLS-CHB	0001					UT009	0001	
PLS-CHB-R	0001					UT010	0001	
PLS-COE	0001							
PLS-COE-R	0001							
PLS-CRL	0001							
PLS-CRL-R	0001							
PLS-PLS	0001							
PLS-PSS	0001							
PLS-PSS-R	0001							
PLS-TIN-R	0001							
PLS-XBN	0001							
PLS-XBN-R	0001							

PARAMETER:

UT012

VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	U	M	COMMENT/REDEF/DEP ON/COPY
000100	1	P-CHX-PLS#	X		3		I	D	
000200	1	P-PLS-VAR#N	N		4		I	D	
000300	1	P-PLS-CTE#N	N		4		I	D	
000301	1	P-VAR#N	N		4		I	D	
000303	1	P-ECR-VAR#N	N		4		I	D	
000304	1	P-IND-ARF					I	D	
000305	2	IND-ARF	N		4	300			
000306	1	P-IND-ECR#V					I	D	
000307	2	IND-ECR#V	N		4	300			
000308	1	P-PVC	F				I	D	
000309	1	P-ARF-VAR#N	N		4		I	D	
000310	1	P-IND-BAS					U	D	
000311	2	IND-BAS	N		4	300			
000400	1	P-PLS-ZO#V	N		7.8		U	D	
000401	1	P-PLS-OPT#V	X		3		U	D	
000900	1	P-ITE#N	N		5		U	D	
001000	1	P-FIN	F				U	D	

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND. NAME, F=FLAG, V=VARIABLE, D=DATE  
I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY  
CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS  
U (PARAMETER UPDATING): U=UPDATABLE, I=INPUT ONLY  
M (PARAMETER MATCHING): BLANK OR D=DYNAMIC, I=IDENTICAL  
REDEF=REDEFINITION, DEP ON= DEPENDING ON

WORKING DATA: UT012 VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	VAL/COMMENT/REDEF/DEP	ON/COPY
000001	1	W-PHA	F					
000002	1	W-Y	N		7.8			
000003	1	W-BAS-IND	N			4		
000100	1	W-CPT	N			4		
000200	1	W-CPT-2	N			4		
000201	1	W-CPT-3	N			4		
000202	1	W-CPT-4	N			4		
000300	1	W-K	N			4		
000400	1	W-I	N			4		
000401	1	W-PI						
000500	2	PI	N		7.8		300	0
000600	1	W-HOR-BAS	F					
000601	1	W-IND-HOR-BAS						
000700	2	IND-HOR-BAS	N			4	3000	
000800	1	W-PN	N		7.8			
000900	1	W-CN#V	N		7.8			
000901	1	W-IND						
001000	2	IND	N			4	3000	
001100	1	W-CN-MIN	N		7.8			
001200	1	W-IND-CN-MIN	N			4		
001300	1	W-PS	N		7.8			0
001400	1	W-X	N		7.8			
001500	1	W-PS-MIN	N		7.8			
001600	1	W-IND-PS-MIN	N			4		
001601	1	W-JR						
001700	2	JR	N		7.8		300	0
001701	1	W-BAS						
001800	2	BAS	N		7.8		300	0
001900	1	W-CN	F					
002000	1	W-FIN	F					
002001	1	W-IND-BAS						
002100	2	IND-BAS	N			4	300	
002200	1	W-ECR-VAR#N	N			4		
002300	1	W-ARF-VAR#N	N			4		
002400	1	W-CHX-PLS#	X			3		
002500	1	W-PLS-CTE#N	N			4		
002501	1	W-PLS-VAR#N	N			4		
002600	1	W-RET	N			5		
002700	1	W-J	N			4		
002800	1	W-ARF	F					
002900	1	W-IND-ARF-BAS						
003000	2	IND-ARF-BAS	N			4	300	
003100	1	W-ARF-BAS#N	N			4		
003200	1	W-PVC	F					
003300	1	W-IND-ECR						
003400	2	IND-ECR	N			4	300	
003500	1	W-PLS-ZO#V	N		7.8			
003600	1	W-PHASE	N			1		

WORKING DATA: UT012                    VERSION: 001 STATUS: TEST

SEQ	LEVEL	FIELD NAME	T	I	CH/DG	OCCUR	VAL/COMMENT/REDEF/DEP ON/COPY
003700	1	W-ITE#N	N		5		
003800	1	W-VAR#N	N		4		
003900	1	W-PLS-OPT#V	X		3		
004000	1	W-0	N		4		
004100	1	W-NEG	N		4		
004200	1	W-STR-NEG	F				
004300	1	W-TST	F				
004400	1	W-NUM					
004500	2	NUM	N		4	300	
004600	1	W-POS	N		4		
004700	1	W-PS-0	F				
004800	1	W-STU	F				
004900	1	W-PSS	F				
005000	1	W-INDICE	N		4		

## LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND.NAME, F=FLAG,  
V=VARIABLE, D=DATE

I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY

CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS

REDEF= REDEFINITION, DEP ON= DEPENDING ON

PROCEDURE:

UT012

VERSION: 001 STATUS: TEST

```
-----  
SEQ DATA  
-----  
000001 <<UT012>> PROCEDURE  
000009 LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
000010 FOR EACH PLS-BAS-R  
000011 WHERE (PLS# = P-CHX-PLS#) AND  
000012 (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND  
000013 (PLS-LGN# = $EDIT ( 1,PIC = '9999'))  
000014 SET W-IND-BAS.IND-BAS (W-CPT) =  
000015 $NUMBER (PLS-BAS-R.PLS-BAS-IND#V)  
000016 ENDFOR  
000017 ENDLLOOP  
000020 IF P-FIN  
000021 SET W-FIN = TRUE  
000022 ELSE  
000023 SET W-FIN = FALSE  
000024 ENDIF  
000036 SET W-CHX-PLS# = P-CHX-PLS#  
000037 SET W-PLS-CTE#N = P-PLS-CTE#N  
000038 LOOP UNTIL W-FIN  
000039 SET P-ITE#N = P-ITE#N + 1  
000044 SET W-RET = $REMAINDER (P-ITE#N, DIV = 20)  
000045 IF W-RET = 0  
000046 CALL UT008 USING INPUT W-CHX-PLS#,W-PLS-CTE#N  
000047 RELEASE PROGRAM UT008  
000048 ENDIF  
000049 SET W-ARF = TRUE  
000050 DO VAR-ARF-BAS  
000051 IF W-PHASE = 2  
000052 SET W-ARF = FALSE  
000053 DO CALCUL-PI  
000054 ENDIF  
000055 ENDLLOOP  
000057 ENDPROC  
000058 <<VAR-ARF-BAS>> PROCEDURE  
000060 SET W-FIN = FALSE  
000061 IF (P-PLS-Z0#V <= - 0.11111111) OR (P-PLS-Z0#V >= 0.11111111)  
000062 AND (P-ITE#N = 1)  
000064 LIST 'CONTRAINTES INCOMPATIBLES, PAS DE SOLUTIONS'  
000065 SET W-FIN = TRUE  
000066 ELSE  
000067 LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
000068 FOR EACH PLS-BAS-R  
000069 WHERE (PLS# = P-CHX-PLS#) AND  
000070 (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND  
000071 (PLS-LGN# = $EDIT (1,PIC = '9999'))  
000072 SET P-IND-BAS.IND-BAS (W-CPT) =  
000073 $NUMBER (PLS-BAS-R.PLS-BAS-IND#V)  
000075 ENDFOR  
000076 ENDLLOOP
```

PROCEDURE: UT012 VERSION: 001 STATUS: TEST

```
-----  
SEQ DATA  
-----  
000077 SET W-J = 0  
000078 LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
000079 SET W-I = 0  
000080 LOOP VARYING W-CPT-2 FROM 1 THRU P-ARF-VAR#N  
000081 IF P-IND-BAS.IND-BAS (W-CPT) NE  
000082 P-IND-ARF.IND-ARF (W-CPT-2)  
000083 SET W-I = W-I + 1  
000084 ENDIF  
000085 ENDLOOP  
000086 IF W-I NE P-ARF-VAR#N  
000087 SET W-J = W-J + 1  
000088 SET W-IND-ARF-BAS.IND-ARF-BAS (W-J) =  
000089 P-IND-BAS.IND-BAS (W-CPT)  
000091 ENDIF  
000092 ENDLOOP  
000093 SET W-ARF-BAS#N = W-J  
000095 IF (W-ARF-BAS#N = 0) :PAS DE VAR ARF EN BASE  
000097 SET W-PHASE = 1  
000098 DO PHASE-2-1  
000099 SET W-FIN = TRUE  
000100 ENDIF  
000101 IF (W-ARF-BAS#N NE 0)  
000102 SET W-PHASE = 2  
000103 DO PHASE-2-2 :VAR ARF EN BASE  
000104 ENDIF  
000105 ENDIF  
000107 ENDPROC  
000108 <<PHASE-2-1>> PROCEDURE  
000110 LOOP VARYING W-CPT FROM 1 THRU P-ARF-VAR#N  
000111 FOR EACH PLS-CRL  
000112 WHERE (PLS# = P-CHX-PLS#) AND  
000113 (PLS-CLN# = $EDIT (P-IND-ARF.IND-ARF (W-CPT),PIC = '9999'))  
000114 DELETE PLS-CRL  
000115 ENDFOR  
000116 LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N  
000117 SET W-CPT-4 = P-IND-ARF.IND-ARF (W-CPT)  
000118 FOR EACH PLS-COE  
000119 WHERE (PLS# = P-CHX-PLS#) AND  
000120 (PLS-CLN# = $EDIT (W-CPT-4,PIC = '9999')) AND  
000121 (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))  
000122 DELETE PLS-COE  
000123 ENDFOR  
000124 FOR EACH PLS-PSS  
000125 WHERE (PLS# = P-CHX-PLS#) AND  
000126 (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))  
000127 DELETE PLS-PSS  
000128 ENDFOR  
000129 ENDLOOP
```

PROCEDURE:

UT012

VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
000130   ENDLOOP  
000131   IF P-PVC  
000132       SET W-PVC = TRUE  
000133   ELSE  
000134       SET W-PVC = FALSE  
000135   ENDIF  
000136   LOOP VARYING W-CPT FROM 1 THRU P-ECR-VAR#N  
000137       SET W-IND-ECR.IND-ECR (W-CPT) = P-IND-ECR#V.IND-ECR#V (W-CPT)  
000138   ENDLOOP  
000139       SET W-ECR-VAR#N = P-ECR-VAR#N  
000140       SET W-CHX-PLS# = P-CHX-PLS#  
000141       SET W-PLS-CTE#N = P-PLS-CTE#N  
000142       SET W-PLS-ZO#V = 0  
000143   LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
000144       SET W-PI.PI (W-CPT) = 0  
000145   ENDLOOP  
000146       SET W-ARF-VAR#N = P-ARF-VAR#N  
000147       SET W-PHASE = 1  
000148       SET W-PLS-VAR#N = P-PLS-VAR#N  
000149       SET W-FIN = FALSE  
000150       SET W-ITE#N = 0  
000151       SET W-VAR#N = P-PLS-VAR#N + P-ECR-VAR#N  
000152       SET W-PLS-OPT#V = P-PLS-OPT#V  
000154       CALL UT005 USING INPUT W-CHX-PLS#,W-PLS-VAR#N,W-PLS-CTE#N,W-VAR#N,  
000155                               W-PHASE,W-ECR-VAR#N,W-IND-ECR,W-PVC,  
000156                               W-ARF-VAR#N  
000157                               UPDATE W-PLS-ZO#V,W-PLS-OPT#V,W-PI,W-ITE#N,W-FIN  
000159   RELEASE PROGRAM UT005  
000162 ENDPROC  
000163 <<PHASE-2-2>> PROCEDURE  
000166   DO CALCUL-PI  
000168   LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
000169       FOR EACH PLS-BAS-R  
000170           WHERE (PLS# = P-CHX-PLS#) AND  
000171                   (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND  
000172                   (PLS-LGN# = $EDIT (1,PIC = '9999'))  
000173           SET W-IND-BAS.IND-BAS (W-CPT) =  
000174                   $NUMBER (PLS-BAS-R.PLS-BAS-IND#V)  
000175       ENDFOR  
000176   ENDLOOP  
000231   FOR EACH PLS-CHB  
000232       WHERE (PLS# = P-CHX-PLS#)  
000233       DELETE PLS-CHB  
000234   ENDFOR  
000236   DO TEST-SORTIE  
000238 ENDPROC  
000239 <<TEST-SORTIE>> PROCEDURE  
000241       SET W-PSS = FALSE
```

PROCEDURE:

UT012

VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
000242   SET W-0 = 0  
000243   SET W-NEG = 0  
000244   SET W-STR-NEG = FALSE  
000245   SET W-TST = TRUE  
000246   SET W-FIN = FALSE  
000247   SET W-TST = FALSE  
000248   SET W-I = 1  
000249   LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
000250     FOR EACH PLS-BAS  
000251       WHERE (PLS# = P-CHX-PLS#) AND  
000252         (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND  
000253         (PLS-LGN# = $EDIT (1,PIC = '9999'))  
000254       LOOP VARYING W-CPT-2 FROM 1 THRU W-ARF-BAS#N  
000255         IF $NUMBER (PLS-BAS.PLS-BAS-IND#V) =  
000256           W-IND-ARF-BAS.IND-ARF-BAS (W-CPT-2)  
000257           SET W-NUM.NUM (W-I) = W-CPT  
000258           SET W-I = W-I + 1  
000261         ENDIF  
000262       ENDLOOP  
000263     ENDFOR  
000264   ENDLOOP  
000265   SET W-POS = 0  
000266   LOOP VARYING W-CPT FROM 1 THRU W-ARF-BAS#N  
000267     FOR EACH PLS-PSS-R  
000268       WHERE (PLS# = P-CHX-PLS#) AND  
000269         (PLS-LGN# = $EDIT (W-NUM.NUM (W-CPT), PIC = '9999'))  
000270       IF PLS-PSS-R.PLS-PS#V <= - 0.11111111  
000271         SET W-NEG = W-NEG + 1  
000272         IF (PLS-PSS-R.PLS-PS#V < 0) AND NOT W-STR-NEG  
000273           SET W-STR-NEG = TRUE  
000274           SET W-INDICE = W-IND-ARF-BAS.IND-ARF-BAS (W-CPT)  
000275           SET W-BAS-IND = W-NUM.NUM (W-CPT)  
000276         ENDIF  
000277         IF (PLS-PSS-R.PLS-PS#V >= - 0.11111111) AND  
000278           (PLS-PSS-R.PLS-PS#V <= 0.11111111)  
000279           SET W-0 = W-0 + 1  
000280         ENDIF  
000281       ELSE  
000282         SET W-POS = W-POS + 1  
000283       ENDIF  
000284     ENDFOR  
000285   ENDLOOP  
000286   IF W-0 = W-ARF-BAS#N  
000287     SET W-I = 1  
000288     SET W-STU = FALSE  
000289     LOOP WHILE (W-I <= W-0) AND NOT W-STU  
000290       SET W-J = 1  
000291       SET W-PS-0 = TRUE
```

PROCEDURE:

UT012

VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
000292      SET W-K = P-PLS-CTE#N - W-ARF-BAS#N
000293      LOOP WHILE (W-J <= W-K) AND (W-J NE W-IND-ARF-BAS.IND-ARF-BAS
000294              (W-I)) AND W-PS-O
000295              SET W-CPT-4 = W-NUM.NUM (W-I)
000296              FOR EACH PLS-BAS-R
000297                  WHERE (PLS# = P-CHX-PLS#) AND
000298                      (PLS-CLN# = $EDIT (W-J,PIC = '9999')) AND
000299                      (PLS-LGN# = $EDIT (W-CPT-4,PIC = '9999'))
000300              IF (PLS-BAS-R.PLS-BAS#V <= -0.11111111) OR
000301                  (PLS-BAS-R.PLS-BAS#V >= 0.11111111)
000302                  SET W-PS-O = FALSE
000303              ELSE
000304                  SET W-J = W-J + 1
000305              ENDIF
000306          ENDFOR
000307      ENDLOOP
000308      IF W-J = W-K + 1
000309          LIST 'PAS DE SOLUTION, LE SYSTEME DES CONTRAINTES'
000310          LIST 'EST REDONDANT'
000311          SET W-STU = TRUE
000312      ELSE
000313          SET W-I = W-I + 1
000314      ENDIF
000315      ENDLOOP
000316      ENDIF
000317      IF (W-NEG = W-ARF-BAS#N) AND W-STR-NEG
000318          SET W-IND-PS-MIN = W-INDICE
000319          DO CALCUL-JR
000320      ELSE
000321          IF W-POS = P-PLS-CTE#N
000322              SET W-PSS = TRUE
000323              DO SIGNE-PS
000324          ENDIF
000327      ENDIF
000329      ENDPROC
000413      <<CALCUL-PI>> PROCEDURE
000416          LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
000417              SET W-PI.PI (W-CPT) = 0
000418              LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N
000419                  FOR EACH PLS-CRL-R
000420                      WHERE (PLS# = P-CHX-PLS#) AND
000421                          (PLS-CLN# = $EDIT (W-IND-BAS.IND-BAS (W-CPT-2),
000422                              PIC = '9999'))
000500                      FOR EACH PLS-BAS-R
000600                          WHERE (PLS# = P-CHX-PLS#) AND
000700                              (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND
000800                              (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))
000900                      SET W-PI.PI(W-CPT) = W-PI.PI(W-CPT) + PLS-CRL-R.PLS-CRL#V

```

PROCEDURE: UT012 VERSION: 001 STATUS: TEST

```
-----  
SEQ      DATA  
-----  
000901                                * PLS-BAS-R.PLS-BAS#V  
001000          ENDFOR  
001100          ENDFOR  
001200          ENDLOOP  
001300          ENDLOOP  
001301          FOR EACH PLS-PLS  
001302          WHERE PLS# = P-CHX-PLS#  
001303          LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N  
001304          FOR EACH PLS-TIN-R  
001305          WHERE (PLS# = P-CHX-PLS#) AND  
001306          (PLS-LGN# = $EDIT (W-CPT,PIC = '9999'))  
001307          SET PLS-PLS.PLS-ZO#V = PLS-PLS.PLS-ZO#V + W-PI.PI (W-CPT)  
001308          * PLS-TIN-R.PLS-TI#V  
001310          ENDFOR  
001312          ENDLOOP  
001313          ENDFOR  
001314          SET W-CN = TRUE  
001317          DO IND-HORS-BASE  
001400          ENDPROC  
001500          <<IND-HORS-BASE>> PROCEDURE  
001502          IF P-ITE#N = 1  
001600              SET W-J = 1  
001700              LOOP VARYING W-CPT FROM 1 THRU P-VAR#N  
001800                  SET W-HOR-BAS = TRUE  
001900                  SET W-K = 1  
002000                  LOOP WHILE (W-K <= P-PLS-CTE#N) AND (W-HOR-BAS)  
002100                      FOR EACH PLS-BAS-R  
002200                          WHERE (PLS# = P-CHX-PLS#) AND  
002201                              (PLS-CLN# = $EDIT (W-K,PIC = '9999')) AND  
002202                              (PLS-LGN# = $EDIT (1,PIC = '9999'))  
002300                              IF W-CPT = $NUMBER (PLS-BAS-R.PLS-BAS-IND#V)  
002400                                  SET W-HOR-BAS = FALSE  
002500                              ELSE  
002600                                  SET W-K = W-K + 1  
002700                              ENDIF  
002800                          ENDFOR  
002900                      ENDLOOP  
003000                  IF (W-K > P-PLS-CTE#N) AND W-HOR-BAS  
003100                      SET W-IND-HOR-BAS.IND-HOR-BAS (W-J) = W-CPT  
003200                      SET W-J = W-J + 1  
003300                  ENDIF  
003400              ENDLOOP  
003401          ENDIF  
003403          DO CALCUL-CN  
003500          ENDPROC  
003600          <<CALCUL-CN>> PROCEDURE  
003700          LOOP VARYING W-CPT FROM 1 THRU (P-VAR#N - P-PLS-CTE#N)  
003701          SET W-PN = 0
```

PROCEDURE:

UT012 VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
003702   FOR EACH PLS-CRL-R
003703   WHERE (PLS# = P-CHX-PLS#) AND
003704   (PLS-CLN# = $EDIT (W-IND-HOR-BAS.IND-HOR-BAS (W-CPT),
003705   PIC = '9999'))
003706   SET W-CN#V = PLS-CRL-R.PLS-CRL#V
003707   ENDFOR
003800   LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N
003900   FOR EACH PLS-COE-R
004000   WHERE (PLS# = P-CHX-PLS#) AND
004100   (PLS-CLN# = $EDIT (W-IND-HOR-BAS.IND-HOR-BAS (W-CPT),
004101   PIC = '9999'))
004200   AND (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))
004300   SET W-PN = W-PN + W-PI.PI (W-CPT-2)* PLS-COE-R.PLS-COE#V
004500   ENDFOR
004600   ENDLOOP
004700   SET W-CN#V = W-CN#V - W-PN
004800   IF (P-ITE#N = 1) AND (W-PHASE = 1)
004900   FOR NEW PLS-CHB
005000   SET PLS-CHB.PLS# = P-CHX-PLS#
005100   SET PLS-CHB.PLS-CLN# = $EDIT (W-CPT,PIC = '9999')
005200   SET PLS-CHB.PLS-CHB#V = W-CN#V
005300   SET PLS-CHB.PLS-CHB-IND#V =
005301   $EDIT (W-IND-HOR-BAS.IND-HOR-BAS (W-CPT),PIC = '9999')
005400   ENDFOR
005500   ELSE
005501   IF W-PHASE = 2
005600   FOR EACH PLS-CHB
005700   WHERE (PLS# = P-CHX-PLS#) AND
005701   (PLS-CLN# = $EDIT (W-CPT,PIC = '9999'))
005800   SET PLS-CHB.PLS-CHB#V = W-CN#V
005900   SET PLS-CHB.PLS-CHB-IND#V =
005901   $EDIT (W-IND-HOR-BAS.IND-HOR-BAS (W-CPT),PIC = '9999')
006000   ENDFOR
006001   ENDIF
006100   ENDIF
006200   ENDLOOP
006202   DO SIGNE-DE-CN
006300   ENDPROC
006400   <<SIGNE-DE-CN>> PROCEDURE
006500   SET W-I = 1
006501   SET W-IND-CN-MIN = 0
006600   SET W-CN-MIN = 0
006700   LOOP VARYING W-CPT FROM 1 THRU (P-VAR#N - P-PLS-CTE#N)
006800   FOR EACH PLS-CHB-R
006900   WHERE (PLS# = P-CHX-PLS#) AND
006901   (PLS-CLN# = $EDIT (W-CPT,PIC = '9999'))
007000   IF PLS-CHB-R.PLS-CHB#V < - 0.11111111
007100   IF PLS-CHB-R.PLS-CHB#V < W-CN-MIN

```

PROCEDURE: UT012 VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
007200          SET W-CN-MIN = PLS-CHB-R.PLS-CHB#V
007300          SET W-IND-CN-MIN = $NUMBER (PLS-CHB-R.PLS-CHB-IND#V)
007400          ENDIF
007500          SET W-I = W-I + 1
007600          ENDIF
007700      ENDFOR
007800  ENDLOOP
007900  IF W-I = 1
007901      SET W-FIN = TRUE
007902      SET W-CHX-PLS# = P-CHX-PLS#
007903      SET W-PLS-VAR#N = P-PLS-VAR#N
007904      SET W-PLS-CTE#N = P-PLS-CTE#N
007905      SET W-ECR-VAR#N = P-ECR-VAR#N
007906      SET W-ARF-VAR#N = P-ARF-VAR#N
007907      IF (P-PVC) AND (W-PHASE = 2)
007908          CALL UT009 USING INPUT W-CHX-PLS#,W-PLS-VAR#N,W-PLS-CTE#N,
007909                          W-ECR-VAR#N,W-ARF-VAR#N
007910          RELEASE PROGRAM UT009
007911      ELSE
007912          IF W-PHASE = 2
007914              CALL UT010 USING INPUT W-CHX-PLS#,W-PLS-VAR#N,W-PLS-CTE#N,
007915                          W-ECR-VAR#N,W-ARF-VAR#N
007916              RELEASE PROGRAM UT010
007918          ENDIF
007919      ENDIF
008100  ELSE
008200      DO CALCUL-PS
008300  ENDIF
008400  ENDPROC
008500  <<CALCUL-PS>> PROCEDURE
008600      LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
008700          SET W-PS = 0
008800          LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N
008900              FOR EACH PLS-BAS-R
009000                  WHERE (PLS# = P-CHX-PLS#) AND
009100                      (PLS-CLN# = $EDIT (W-CPT-2,PIC = '9999')) AND
009200                      (PLS-LGN# = $EDIT (W-CPT,PIC = '9999'))
009300                  FOR EACH PLS-COE-R
009400                      WHERE (PLS# = P-CHX-PLS#) AND
009500                          (PLS-CLN# = $EDIT (W-IND-CN-MIN,PIC = '9999')) AND
009600                          (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))
009601                      SET W-PS = W-PS + PLS-BAS-R.PLS-BAS#V * PLS-COE-R.PLS-COE#V
009602                  ENDFOR
009603              ENDFOR
009604          ENDLOOP
009700          IF (P-ITE#N = 1) AND (W-PHASE = 1)
009800              FOR NEW PLS-PSS
009900                  SET PLS-PSS.PLS# = P-CHX-PLS#

```



PROCEDURE: UT012 VERSION: 001 STATUS: TEST

```

-----
SEQ      DATA
-----
012807          $NUMBER (PLS-BAS-R.PLS-BAS-IND#V)
012808          SET W-BAS-IND = W-CPT
012810          ENDFOR
012900          ENDIF
013000          ENDIF
013100          ENDFOR
013200          SET W-I = W-I + 1
013900          ENDIF
014000          ENDFOR
014100          ENDOLOOP
014200          IF W-I = 1
014300              LIST 'SOLUTIONS OPTIMALES NON BORNEES'
014301              SET W-FIN = TRUE
014400          ELSE
014401              IF W-PSS :NOT W-ARF
014500                  DO CALCUL-JR
014501              ENDIF
014600          ENDIF
014700          ENDPROC
014800          <<CALCUL-JR>> PROCEDURE
014900          FOR EACH PLS-PSS-R
015000              WHERE (PLS# = P-CHX-PLS#) AND
015001                  (PLS-LGN# = $EDIT (W-BAS-IND,PIC = '9999'))
015100              SET W-JR.JR (W-BAS-IND) = 1 / PLS-PSS-R.PLS-PS#V
015200          ENDFOR
015300          LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
015400              FOR EACH PLS-PSS-R
015500                  WHERE (PLS# = P-CHX-PLS#) AND
015600                      (PLS-LGN# = $EDIT (W-CPT, PIC = '9999'))
015601                  IF W-CPT NE W-BAS-IND
015700                      SET W-JR.JR (W-CPT) = - PLS-PSS-R.PLS-PS#V *
015701                          W-JR.JR (W-BAS-IND)
015703                  ENDIF
015800              ENDFOR
015900          ENDOLOOP
015902          DO BASE
016000          ENDPROC
016100          <<BASE>> PROCEDURE
016200          LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
016300              FOR EACH PLS-BAS
016400                  WHERE (PLS# = P-CHX-PLS#) AND
016500                      (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND
016600                      (PLS-LGN# = $EDIT (W-BAS-IND,PIC = '9999'))
016700              SET W-BAS.BAS (W-CPT) = PLS-BAS.PLS-BAS#V
016800              SET PLS-BAS.PLS-BAS#V = W-JR.JR (W-BAS-IND) *
016801                          PLS-BAS.PLS-BAS#V
016900          ENDFOR
017000          ENDOLOOP

```

PROCEDURE:

UT012

VERSION: 001 STATUS: TEST

```

-----
SEO    DATA
-----
017100    LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
017200        LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N
017300            FOR EACH PLS-BAS
017400                WHERE (PLS# = P-CHX-PLS#) AND
017500                    (PLS-CLN# = $EDIT (W-CPT,PIC = '9999')) AND
017600                    (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))
017700                IF W-CPT-2 NE W-BAS-IND
017800                    SET PLS-BAS.PLS-BAS#V = PLS-BAS.PLS-BAS#V +
017801                        W-JR.JR (W-CPT-2) * W-BAS.BAS (W-CPT)
017900                ENDIF
018000            ENDFOR
018100        ENDLLOOP
018200    ENDLLOOP
018300    LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
018400        FOR EACH PLS-BAS
018500            WHERE (PLS# = P-CHX-PLS#) AND
018600                (PLS-CLN# = $EDIT (W-BAS-IND,PIC = '9999')) AND
018700                (PLS-LGN# = $EDIT (W-CPT,PIC = '9999'))
018800            SET PLS-BAS.PLS-BAS-IND#V =
018801                $EDIT (W-IND-CN-MIN,PIC = '9999')
018900        ENDFOR
019000    ENDLLOOP
019001    SET W-IND-BAS.IND-BAS (W-BAS-IND) = W-IND-CN-MIN
019002    DO CALCUL-XB
019100 ENDPROC
019200 <<CALCUL-XB>> PROCEDURE
019300    LOOP VARYING W-CPT FROM 1 THRU P-PLS-CTE#N
019400        FOR EACH PLS-XBN
019401            WHERE (PLS# = P-CHX-PLS#) AND
019500                (PLS-CLN# = $EDIT (W-CPT,PIC = '9999'))
019600            SET PLS-XBN.PLS-XB#V = 0
019700        LOOP VARYING W-CPT-2 FROM 1 THRU P-PLS-CTE#N
019800            FOR EACH PLS-TIN-R
019900                WHERE (PLS# = P-CHX-PLS#) AND
020000                    (PLS-LGN# = $EDIT (W-CPT-2,PIC = '9999'))
020100                FOR EACH PLS-BAS-R
020200                    WHERE (PLS# = P-CHX-PLS#) AND
020300                        (PLS-CLN# = $EDIT (W-CPT-2,PIC = '9999')) AND
020400                        (PLS-LGN# = $EDIT (W-CPT,PIC = '9999'))
020500                    SET PLS-XBN.PLS-XB#V = PLS-XBN.PLS-XB#V
020501                        + PLS-BAS-R.PLS-BAS#V * PLS-TIN-R.PLS-TI#V
020600                ENDFOR
020700            ENDFOR
020800        ENDLLOOP
020900    ENDFOR
021000 ENDLLOOP
021001    LOOP VARYING W-CPT FROM 1 THRU (P-VAR#N - P-PLS-CTE#N)
021002        IF W-IND-HOR-BAS.IND-HOR-BAS (W-CPT) = W-IND-CN-MIN

```

PROCEDURE:

UT012

VERSION: 001 STATUS: TEST

-----  
SEQ DATA  
-----

021003 SET W-IND-HOR-BAS.IND-HOR-BAS (W-CPT) = W-IND-PS-MIN  
021005 ENDIF  
021007 ENDLOOP  
021200 ENDPROC

## LES ECRANS

PANEL: PN0001 VERSION: 001 STATUS: TEST

IDENTIFICATION:

Panel name PN0001

Created 11/03/87 By ETUD04
Last Modified 12/09/87 at 14:05 By ETUD04

Run Status PRIVATE
Short description MENU PRINCIPAL LIN

Description:

Three horizontal lines representing a description area.

PANEL PARAMETERS:

Start field symbol ^ New field symbol +
End field symbol ; Delete field symbol \*
Repeating group symbol a
Input fill character S (S=space, L=lowval, Z=zeros, U=\_, other=itself)
Output fill character U (S=space, L=lowval, U=\_, other=itself)
Non-display character S (S=space, other=as specified)
Error fill character \* (as specified)
Case translation U (U=upper, M=mixed)
Required N (Y=yes, N=no)
Error handling B (N=none, \*=fill w/errorfill, H=high intensity.)
(B=both: H if illegal value \* if rqd missing)
PF1=HELP, PF3=CLARIFY Y (Y=yes, N=no)
PF7=SCR -, PF8=SCR + N (Y=yes, N=no)
PF10=SCR TOP, PF11=SCR BOT
Edit-rule error proc C (C=clarify command, A=application)
Process appl on scroll Y (Y=yes, N=no)
Help panel name \_\_\_\_\_ Version \_\_\_\_
Prefix panel name \_\_\_\_\_ Version \_\_\_\_
Suffix panel name \_\_\_\_\_ Version \_\_\_\_



PANEL: PN0001 VERSION: 001 STATUS: TEST

## FIELD SUMMARY TABLE:

Seq	Lv	Field name	Attr	T	Len	In.dp	Occ	Comments
1	2		PSH	X	14			MENU PRINCIPAL
2	2		PSH	X	14			-----
3	2		PSL	X	32			1- OPTIMISATION DU PL
4	2		PSL	X	35			2- OPTIMISATION D'UN
5	2		PSH	X	13			VOTRE CHOIX :
6	2	CHX	UAH	N	1	1		
7	2		PSL	X	77			-----
8	2		PSL	X	23			PF6 POUR FIN DE TRAVA

## LEGEND:

Seq=sequence number

Attr=screen attributes: U=unprot H=highlight A=327X alphanumeric  
 P=prot I=invisible N=327X numeric  
 S=skip L=low-light E=ensure input  
 C=cursor

T=field type: X=alphanumeric, N=numeric

In.dp=Integer-places.decimal-places

Occ=number of occurrences

PANEL: PN0001 VERSION: 001 STATUS: TEST

## INPUT RULES:

Seq	Field name	E R H Q	Minimum value	Maximum value	J I C S F S	Mn D M DP S S	A C M C D F
1		B N			L S U		N
2		B N			L S U		N
3		B N			L S U		N
4		B N			L S U		N
5		B N			L S U		N
6	CHX	B N	1		2 R Z U	N N N N	N
7		B N			L S U		N
8		B N			L S U		N

## LEGEND:

Seq=sequence number

E/H=error handling: N=none, \*=fill with \*, H=high intensity.

B=both (H if illegal value, \* if required field missing)

R/Q=required field: Y=yes, N=no, C=conditional

J/S=justification: N=none, L=left, R=right, A=align by decimal point

I/F=input fill character:

S=spaces, L=low-values, Z=zeroes, U=underscore, Other=itself

C/S=case: U=upper case, M=mixed case

Mn/DP=minimum required decimals (for justification=A)

D/S=allow digit separator: Y=yes, N=no

M/S=allow minus sign: Y=yes, N=no

A/C=allow currency symbol: Y=yes, N=no

C/D=check digit: N=none, T=modulo 10, E=modulo 11

M/F=must fill: Y=yes, N=no

PANEL: PN0001 VERSION: 001 STATUS: TEST

OUTPUT RULES:

Seq	Field name	Edit pattern	O	E
			F	C H
1			U	N N
2			U	N N
3			U	N N
4			U	N N
5			U	N N
6	CHX	9	U	N N
7			U	N N
8			U	N N

LEGEND:

Seq=sequence number  
O/F=output fill character:  
S=spaces, L=low-values, Z=zeroes, U=underscore, Other=itself  
C=color:  
N=neutral, B=blue, R=red, P=pink, G=green  
T=turquoise, Y=yellow, W=white/black  
E/H=extended highlighting:  
N=none, B=blink, R=reverse video, U=underscore

PANEL: PN0010 VERSION: 001 STATUS: TEST

## IDENTIFICATION:

Panel name PN0010

Created 11/23/87 By ETUD04  
 Last Modified 12/09/87 at 14:06 By ETUD04

Run Status PRIVATE  
 Short description donnees fichier

Description:

\_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

## PANEL PARAMETERS:

Start field symbol ^	New field symbol +
End field symbol ;	Delete field symbol *
Repeating group symbol a	
Input fill character S	(S=space, L=lowval, Z=zeros, U=_, other=itself)
Output fill character U	(S=space, L=lowval, U=_, other=itself)
Non-display character S	(S=space, other=as specified)
Error fill character *	(as specified)
Case translation U	(U=upper, M=mixed)
Required N	(Y=yes, N=no)
Error handling B	(N=none, *=fill w/errorfill, H=high intensity.) (B=both: H if illegal value * if rpd missing)
PF1=HELP, PF3=CLARIFY Y	(Y=yes, N=no)
PF7=SCR -, PF8=SCR + N	(Y=yes, N=no)
PF10=SCR TOP, PF11=SCR BOT	
Edit-rule error proc C	(C=clarify command, A=application)
Process appl on scroll Y	(Y=yes, N=no)
Help panel name _____	Version _____
Prefix panel name _____	Version _____
Suffix panel name _____	Version _____

PANEL:

PN0010

VERSION: 001 STATUS: TEST

LAYOUT:

```

!.....1.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!
!                                     ^ donnees fichier clientes;
!                                     1
!                                     ^-----;
!                                     2
! ^nombre cni ;; ^_____ ;
! 3                                     4
! ^nombre cn1 ;; ^_____ ;
! 5                                     6
! ^nombre cn2 ;; ^_____ ;
! 7                                     8
! ^nombre cn3 ;; ^_____ ;
! 9                                     10
! ^nombre cn4 ;; ^_____ ;
! 11                                    12
! ^nombre cn5 ;; ^_____ ;
! 13                                    14
! ^nombre ab2 ;; ^_____ ;
! 15                                    16
! ^nombre ab3 ;; ^_____ ;
! 17                                    18
! ^nombre ab4 ;; ^_____ ;
! 19                                    20
! ^nombre ab5 ;; ^_____ ;
! 21                                    22
! ^nombre elm ;; ^_____ ;
! 23                                    24
! ^nombre pro ;; ^_____ ;
! 25                                    26
! ^taux de croissance du fichier ;; ^_____ :
! 27                                    28
!
!-----;
! 29
! ^pf6 pour fin de travail;
! 30
!.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!

```

PANEL: PN0010 VERSION: 001 STATUS: TEST

## FIELD SUMMARY TABLE:

Seq	Lv	Field name	Attr	T	Len	In.dp	Occ	Comments
1	2		PSH	X	24			donnees fichier clien
2	2		PSH	X	24			
3	2		PSL	X	12			nombre cni :
4	2	CNI#N	UAH	N	7	7		
5	2		PSL	X	12			nombre cn1 :
6	2	CN1#N	UAH	N	7	7		
7	2		PSL	X	12			nombre cn2 :
8	2	CN2#N	UAH	N	7	7		
9	2		PSL	X	12			nombre cn3 :
10	2	CN3#N	UAH	N	7	7		
11	2		PSL	X	12			nombre cn4 :
12	2	CN4#N	UAH	N	7	7		
13	2		PSL	X	12			nombre cn5 :
14	2	CN5#N	UAH	N	7	7		
15	2		PSL	X	12			nombre ab2 :
16	2	AB2#N	UAH	N	7	7		
17	2		PSL	X	12			nombre ab3 :
18	2	AB3#N	UAH	N	7	7		
19	2		PSL	X	12			nombre ab4 :
20	2	AB4#N	UAH	N	7	7		
21	2		PSL	X	12			nombre ab5 :
22	2	AB5#N	UAH	N	7	7		
23	2		PSL	X	12			nombre elm :
24	2	ELM#N	UAH	N	7	7		
25	2		PSL	X	12			nombre pro :
26	2	PRO#N	UAH	N	7	7		
27	2		PSL	X	31			taux de croissance du
28	2	CRO#T	UAH	N	6	3.2		
29	2		PSL	X	77			-----
30	2		PSL	X	23			pf6 pour fin de trava

## LEGEND:

Seq=sequence number

Attr=screen attributes: U=unprot H=highlight A=327X alphanumeric  
 P=prot I=invisible N=327X numeric  
 S=skip L=low-light E=ensure input  
 C=cursor

T=field type: X=alphanumeric, N=numeric

In.dp=Integer-places.decimal-places

Occ=number of occurrences

PANEL: PN0010 VERSION: 001 STATUS: TEST

INPUT RULES:

Seq	Field name	E R	Minimum	Maximum	J I C Mn D M A C M	J I C Mn D M A C M
		H Q	value	value	S F S DP S S C D F	S F S DP S S C D F
1		B N			L S U	N
2		B N			L S U	N
3		B N			L S U	N
4	CNI#N	B N	0	9999999	R Z U	N N N N N
5		B N			L S U	N
6	CN1#N	B N	0	9999999	R Z U	N N N N N
7		B N			L S U	N
8	CN2#N	B N	0	9999999	R Z U	N N N N N
9		B N			L S U	N
10	CN3#N	B N	0	9999999	R Z U	N N N N N
11		B N			L S U	N
12	CN4#N	B N	0	9999999	R Z U	N N N N N
13		B N			L S U	N
14	CN5#N	B N	0	9999999	R Z U	N N N N N
15		B N			L S U	N
16	AB2#N	B N	0	9999999	R Z U	N N N N N
17		B N			L S U	N
18	AB3#N	B N	0	9999999	R Z U	N N N N N
19		B N			L S U	N
20	AB4#N	B N	0	9999999	R Z U	N N N N N
21		B N			L S U	N
22	AB5#N	B N	0	9999999	R Z U	N N N N N
23		B N			L S U	N
24	ELM#N	B N	0	9999999	R Z U	N N N N N
25		B N			L S U	N
26	PRO#N	B N	0	9999999	R Z U	N N N N N
27		B N			L S U	N
28	CRO#T	B N			A Z U	0 Y Y Y N N
29		B N			L S U	N
30		B N			L S U	N

LEGEND:

Seq=sequence number  
 E/H=error handling: N=none, \*=fill with \*, H=high intensity.  
 B=both (H if illegal value, \* if required field missing)  
 R/Q=required field: Y=yes, N=no, C=conditional  
 J/S=justification: N=none, L=left, R=right, A=align by decimal point  
 I/F=input fill character:  
 S=spaces, L=low-values, Z=zeros, U=underscore, Other=itself  
 C/S=case: U=upper case, M=mixed case  
 Mn/DP=minimum required decimals (for justification=A)  
 D/S=allow digit separator: Y=yes, N=no  
 M/S=allow minus sign: Y=yes, N=no  
 A/C=allow currency symbol: Y=yes, N=no  
 C/D=check digit: N=none, T=modulo 10, E=modulo 11

PANEL: PN0011 VERSION: 001 STATUS: TEST

## IDENTIFICATION:

Panel name PN0011

Created 11/03/87 By ETUD04  
 Last Modified 12/09/87 at 14:19 By ETUD04

Run Status PRIVATE  
 Short description saisie des donnees du pv

Description:  
 description du probleme de plan de vente

---



---



---

## PANEL PARAMETERS:

Start field symbol ^	New field symbol +
End field symbol ;	Delete field symbol *
Repeating group symbol a	
Input fill character S	(S=space, L=lowval, Z=zeros, U=_, other=itself)
Output fill character U	(S=space, L=lowval, U=_, other=itself)
Non-display character S	(S=space, other=as specified)
Error fill character *	(as specified)
Case translation U	(U=upper, M=mixed)
Required N	(Y=yes, N=no)
Error handling B	(N=none, *=fill w/errorfill, H=high intensity, (B=both: H if illegal value * if rpd missing)
PF1=HELP, PF3=CLARIFY Y	(Y=yes, N=no)
PF7=SCR -, PF8=SCR + N	(Y=yes, N=no)
PF10=SCR TOP, PF11=SCR BOT	
Edit-rule error proc C	(C=clarify command, A=application)
Process appl on scroll Y	(Y=yes, N=no)
Help panel name _____	Version _____
Prefix panel name _____	Version _____
Suffix panel name _____	Version _____

PANEL:

PN0011

VERSION: 001 STATUS: TEST

LAYOUT:

```

!.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!
!          ^description generale du plan de vente;
!          1
!          ^-----;
!          2
! ^libelle du probleme   ;;
! 3
!          à _____;
!          5
! ^numero du probleme   ;; ^____;
! 6          7
!
! ^nombre d' actions    ;; ^____;
! 8          9
!
!-----;
!10
! ^pf6 pour fin de travail;
!11
!.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!

```

PANEL: PN0011 VERSION: 001 STATUS: TEST

## FIELD SUMMARY TABLE:

Seq	Lv	Field name	Attr	T	Len	In.dp	Occ	Comments
1	2		PSL	X	37			description generale
2	2		PSL	X	37			
3	2		PSL	X	23			libelle du probleme
4	2	GRP		G			8	
5	3	PLS#L	UAH	X	50			
6	2		PSL	X	23			numero du probleme
7	2	PRB#	PSH	N	3	3		
8	2		PSL	X	23			nombre d' actions
9	2	ACT#N	UAH	N	4	4		
10	2		PSL	X	77			
11	2		PSL	X	23			pf6 pour fin de trava

## LEGEND:

Seq=sequence number

Attr=screen attributes: U=unprot H=highlight A=327X alphanumeric  
 P=prot I=invisible N=327X numeric  
 S=skip L=low-light E=ensure input  
 C=cursor

T=field type: X=alphanumeric, N=numeric

In.dp=Integer-places.decimal-places

Occ=number of occurrences

PANEL: PN0011 VERSION: 001 STATUS: TEST

## INPUT RULES:

Seq	Field name	E R	Minimum	Maximum	J I C	Mn	D	M	A	C	M
		H Q	value	value	S F S	DP	S	S	C	D	F
1		B N			L S U						N
2		B N			L S U						N
3		B N			L S U						N
4	GRP										
5	PLS#L	B N			L S U						N
6		B N			L S U						N
7	PRB#	B N			R Z U		N	N	N	N	N
8		B N			L S U						N
9	ACT#N	B N			R Z U		N	N	N	N	N
10		B N			L S U						N
11		B N			L S U						N

## LEGEND:

Seq=sequence number  
 E/H=error handling: N=none, \*=fill with \*, H=high intensity.  
 B=both (H if illegal value, \* if required field missing)  
 R/Q=required field: Y=yes, N=no, C=conditional  
 J/S=justification: N=none, L=left, R=right, A=align by decimal point  
 I/F=input fill character:  
 S=spaces, L=low-values, Z=zeros, U=underscore, Other=itself  
 C/S=case: U=upper case, M=mixed case  
 Mn/DP=minimum required decimals (for justification=A)  
 D/S=allow digit separator: Y=yes, N=no  
 M/S=allow minus sign: Y=yes, N=no  
 A/C=allow currency symbol: Y=yes, N=no  
 C/D=check digit: N=none, T=modulo 10, E=modulo 11  
 M/F=must fill: Y=yes, N=no

PANEL: PN0012 VERSION: 001 STATUS: TEST

INPUT RULES:

Seq	Field name	E R	Minimum value	Maximum value	J I C	Mn DP	D M A	C M
		H Q			S F S		S S C D F	
1		B N			L S U			N
2	DON#	B N			R Z U		N N N N	N
3		B N			L S U			N
4		B N			L S U			N
5	ACT#L	B N			L S U			N
6		B N			L S U			N
7	ACT#	B N			R Z U		N N N N	N
8		B N			L S U			N
9	ACT#V	B N			R Z U		N N N N	N
10		B N			L S U			N
11	COU#N	B N	1		9 R Z U		N N N N	N
12		B N			L S U			N
13		B N			L S U			N
14		B N			L S U			N
15	CLI-TYP	B N			L S U			N
16		B N			L S U			N
17		B N			L S U			N
18		B N			L S U			N
19		B N			L S U			N
20	ACT-TYP	B N			L S U			N
21		B N			L S U			N
22		B N			L S U			N

LEGEND:

Seq=sequence number  
 E/H=error handling: N=none, \*=fill with \*, H=high intensity,  
 B=both (H if illegal value, \* if required field missing)  
 R/Q=required field: Y=yes, N=no, C=conditional  
 J/S=justification: N=none, L=left, R=right, A=align by decimal point  
 I/F=input fill character:  
 S=spaces, L=low-values, Z=zeros, U=underscore, Other=itself  
 C/S=case: U=upper case, M=mixed case  
 Mn/DP=minimum required decimals (for justification=A)  
 D/S=allow digit separator: Y=yes, N=no  
 M/S=allow minus sign: Y=yes, N=no  
 A/C=allow currency symbol: Y=yes, N=no  
 C/D=check digit: N=none, T=modulo 10, E=modulo 11  
 M/F=must fill: Y=yes, N=no

PANFI: PN0012 VERSION: 001 STATUS: TEST

## OUTPUT RULES:

Seq	Field name	Edit pattern	O	F	C	H	E
1			U	N	N		
2	DON#	Z9	U	N	N		
3			U	N	N		
4			U	N	N		
5	ACT#L		U	N	N		
6			U	N	N		
7	ACT#	ZZZZ9	U	N	N		
8			U	N	N		
9	ACT#V	ZZZZZ9	U	N	N		
10			U	N	N		
11	COU#N	9	U	N	N		
12			U	N	N		
13			U	N	N		
14			U	N	N		
15	CLI-TYP		U	N	N		
16			U	N	N		
17			U	N	N		
18			U	N	N		
19			U	N	N		
20	ACT-TYP		U	N	N		
21			U	N	N		
22			U	N	N		

## LEGEND:

Seq=sequence number

O/F=output fill character:

S=spaces, L=low-values, Z=zeros, U=underscore, Other=itself

C=color:

N=neutral, B=blue, R=red, P=pink, G=green

T=turquoise, Y=yellow, W=white/black

E/H=extended highlighting:

N=none, B=blink, R=reverse video, U=underscore

PANEL: PN012B VERSION: 001 STATUS: TEST

## IDENTIFICATION:

Panel name PN012B

Created 11/23/87 By ETUD04  
 Last Modified 05/18/88 at 13:24 By ETUD04

Run Status PRIVATE  
 Short description donnees couche

Description:  
 ecran de saisie des donnees generales concernant la couche

---



---



---

## PANEL PARAMETERS:

Start field symbol ^	New field symbol +
End field symbol ;	Delete field symbol *
Repeating group symbol à	
Input fill character S	(S=space, L=lowval, Z=zeros, U=_, other=itself)
Output fill character U	(S=space, L=lowval, U=_, other=itself)
Non-display character S	(S=space, other=as specified)
Error fill character *	(as specified)
Case translation U	(U=upper, M=mixed)
Required N	(Y=yes, N=no)
Error handling B	(N=none, *=fill w/errorfill, H=high intensity, B=both: H if illegal value & * if rcd missing)
PF1=HELP, PF3=CLARIFY Y	(Y=yes, N=no)
PF7=SCR -, PF8=SCR + N	(Y=yes, N=no)
PF10=SCR TOP, PF11=SCR BOT	
Edit-rule error proc C	(C=clarify command, A=application)
Process appl on scroll Y	(Y=yes, N=no)
Help panel name _____	Version _____
Prefix panel name _____	Version _____
Suffix panel name _____	Version _____

PANEL:

PN012B

VERSION: 001 STATUS: TEST

LAYOUT:

```

!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!
!^action;^_____!
!1          2          ^saisie des donnees de la couche;^_!
!          3          4          !
!          ^-----^!
!          5          !
!
!^couche ;;^_____!
!6          7          !
!^taux de supers ;;^_____!
!8          9          !
!^nombre de points consideres ;;^____!
!10         11        !
!
!-----!
!12
!^pf6 pour fin de travail;
!13
!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!

```

PANEL: PN012B VERSION: 001 STATUS: TEST

## FIELD SUMMARY TABLE:

Seq	Lv	Field name	Attr	T	Len	In.dp	Occ	Comments
1	2		PSH	X	6			action
2	2	ACT#	PAH	N	5	5		
3	2		PSH	X	31			saisie des donnees de
4	2	DON#	PAH	N	1	1		
5	2		PSL	X	34			-----
6	2		PSL	X	8			couche :
7	2	COU#A	UAH	X	3			
8	2		PSL	X	16			taux de supers :
9	2	SPR#T	UAH	N	6	3.2		
10	2		PSL	X	29			nombre de points cons
11	2	PTS#N	UAH	N	2	2		
12	2		PSL	X	77			-----
13	2		PSL	X	23			pf6 pour fin de trava

## LEGEND:

Seq=sequence number

Attr=screen attributes: U=unprot H=highlight A=327X alphanumeric  
 P=prot I=invisible N=327X numeric  
 S=skip L=low-light E=ensure input  
 C=cursor

T=field type: X=alphanumeric, N=numeric

In.dp=Integer-places.decimal-places

Occ=number of occurrences

PANEL: PN012B VERSION: 001 STATUS: TEST

## INPUT RULES:

Seq	Field name	E R	Minimum	Maximum	J I C	Mn	D	M	A	C	M
		H Q	value	value	S F S	DP	S	S	C	D	F
1		B N			L S U						N
2	ACT#	B N			R Z U		N	N	N	N	N
3		B N			L S U						N
4	DON#	B N			R Z U		N	N	N	N	N
5		B N			L S U						N
6		B N			L S U						N
7	COU#A	B N			R Z U						N
8		B N			L S U						N
9	SPR#T	B N			A Z U	0	Y	Y	Y	N	N
10		B N			L S U						N
11	PTS#N	B N			R Z U		N	N	N	N	N
12		B N			L S U						N
13		B N			L S U						N

## LEGEND:

Seq=sequence number

E/H=error handling: N=none, \*=fill with \*, H=high intensity.

B=both (H if illegal value, \* if required field missing)

R/Q=required field: Y=yes, N=no, C=conditional

J/S=justification: N=none, L=left, R=right, A=align by decimal point

I/F=input fill character:

S=spaces, L=low-values, Z=zeros, U=underscore, Other=itself

C/S=case: U=upper case, M=mixed case

Mn/DP=minimum required decimals (for justification=A)

D/S=allow digit separator: Y=yes, N=no

M/S=allow minus sign: Y=yes, N=no

A/C=allow currency symbol: Y=yes, N=no

C/D=check digit: N=none, T=modulo 10, E=modulo 11

M/F=must fill: Y=yes, N=no

PANEL: PN012B VERSION: 001 STATUS: TEST

## OUTPUT RULES:

Seq	Field name	Edit pattern	O	F	C	H	E
1			U	N	N		
2	ACT#	ZZZZ9	U	N	N		
3			U	N	N		
4	DON#	9	U	N	N		
5			U	N	N		
6			U	N	N		
7	COU#A		U	N	N		
8			U	N	N		
9	SPR#T	ZZ9.99	U	N	N		
10			U	N	N		
11	PTS#N	Z9	U	N	N		
12			U	N	N		
13			U	N	N		

## LEGEND:

Seq=sequence number

O/F=output fill character:

S=spaces, L=low-values, Z=zeros, U=underscore, Other=itself

C=color:

N=neutral, B=blue, R=red, P=pink, G=green

T=turquoise, Y=yellow, W=white/black

E/H=extended highlighting:

N=none, B=blink, R=reverse video, U=underscore

PANEL: PN0013 VERSION: 001 STATUS: TEST

## IDENTIFICATION:

Panel name PN0013

Created 11/04/87 By ETUD04  
 Last Modified 12/09/87 at 14:35 By ETUD04

Run Status PRIVATE  
 Short description rendement/moy-commandee

## Description:

saisie des donnees de rendement et de moyenne commandee pour une  
 action et une couche donnee

---



---

## PANEL PARAMETERS:

Start field symbol ^	New field symbol +
End field symbol ;	Delete field symbol *
Repeating group symbol a	
Input fill character S	(S=space, L=lowval, Z=zeros, U=_, other=itself)
Output fill character U	(S=space, L=lowval, U=_, other=itself)
Non-display character S	(S=space, other=as specified)
Error fill character *	(as specified)
Case translation U	(U=upper, M=mixed)
Required N	(Y=yes, N=no)
Error handling B	(N=none, *=fill w/errorfill, H=high intensity, B=both: H if illegal value * if rqd missing)
PF1=HELP, PF3=CLARIFY Y	(Y=yes, N=no)
PF7=SCR -, PF8=SCR + N	(Y=yes, N=no)
PF10=SCR TOP, PF11=SCR BOT	
Edit-rule error proc C	(C=clarify command, A=application)
Process appl on scroll Y	(Y=yes, N=no)
Help panel name _____	Version _____
Prefix panel name _____	Version _____
Suffix panel name _____	Version _____

PANEL:

PN0013

VERSION: 001 STATUS: TEST

LAYOUT:

```

!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!
!      ^saisie du rendement et de la moyenne commandee;
!      1
!      ^-----;
!      2
!
!      ^action ;; ^-----;
!      3      4
!      ^couche ;; ^____;
!      5      6
!^acquis;      ^rendement;      ^moyenne commandee;
!7      8      9
!-----;      ^-----;
!10      11      12
!à____;      à____;      à____;
!14      15      16
!-----;
!17
!^pf6 pour fin de travail;
!18
!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!

```

PANEL: PN0013 VERSION: 001 STATUS: TEST

## FIELD SUMMARY TABLE:

Seq	Lv	Field name	Attr	T	Len	In.dp	Occ	Comments
1	2		PSL	X	46			saisie du rendement e
2	2		PSL	X	46			-----
3	2		PSL	X	8			action :
4	2	ACT#L	PSH	X	50			
5	2		PSL	X	8			couche :
6	2	COU	PSH	X	3			
7	2		PSL	X	6			acquis
8	2		PSL	X	9			rendement
9	2		PSL	X	17			moyenne commandee
10	2		PSL	X	6			-----
11	2		PSL	X	9			-----
12	2		PSL	X	17			-----
13	2			G			*	
14	3	ACQ#V	UNH	N	3	3		
15	3	RDT#V	UNH	N	6	3.2		
16	3	MCD#V	UNH	N	6	6		
17	2		PSL	X	75			-----
18	2		PSL	X	23			pf6 pour fin de trava

## LEGEND:

Seq=sequence number

Attr=screen attributes: U=unprot H=highlight A=327X alphanumeric  
P=prot I=invisible N=327X numeric  
S=skip L=low-light E=ensure input  
C=cursor

T=field type: X=alphanumeric, N=numeric

In.dp=Integer-places.decimal-places

Occ=number of occurrences

PANEL: PN0013 VERSION: 001 STATUS: TEST

INPUT RULES:

Seq	Field name	E R	Minimum	Maximum	J I C	Mn D	M A C	M
		H Q	value	value	S F S	DP S	S S C	D F
1		B N			L S U			N
2		B N			L S U			N
3		B N			L S U			N
4	ACT#L	B N			L S U			N
5		B N			L S U			N
6	COU	B N			L S U			N
7		B N			L S U			N
8		B N			L S U			N
9		B N			L S U			N
10		B N			L S U			N
11		B N			L S U			N
12		B N			L S U			N
13								
14	ACQ#V	B N	0	100	R Z U		N N N N	N
15	RDT#V	B N			A Z U	0	Y Y Y N	N
16	MCD#V	B N			R Z U		N N N N	N
17		B N			L S U			N
18		B N			L S U			N

LEGEND:

Seq=sequence number  
 E/H=error handling: N=none, \*=fill with \*, H=high intensity.  
                                   B=both (H if illegal value, \* if required field missing)  
 R/Q=required field: Y=yes, N=no, C=conditional  
 J/S=justification: N=none, L=left, R=right, A=align by decimal point  
 I/F=input fill character:  
                   S=spaces, L=low-values, Z=zeroes, U=underscore, Other=itself  
 C/S=case: U=upper case, M=mixed case  
 Mn/DP=minimum required decimals (for justification=A)  
 D/S=allow digit separator: Y=yes, N=no  
 M/S=allow minus sign: Y=yes, N=no  
 A/C=allow currency symbol: Y=yes, N=no  
 C/D=check digit: N=none, T=modulo 10, E=modulo 11  
 M/F=must fill: Y=yes, N=no

PANEL: PN0013 VERSION: 001 STATUS: TEST

## OUTPUT RULES:

Seq	Field name	Edit pattern	O	E
			F	C
1			U	N
2			U	N
3			U	N
4	ACT#L		U	N
5			U	N
6	COU		U	N
7			U	N
8			U	N
9			U	N
10			U	N
11			U	N
12			U	N
13			U	N
14	ACQ#V	ZZ9	U	N
15	RDT#V	ZZ9.99	U	N
16	MCD#V	ZZZZZ9	U	N
17			U	N
18			U	N

## LEGEND:

Seq=sequence number  
 O/F=output fill character:  
   S=spaces, L=low-values, Z=zeroes, U=underscore, Other=itself  
 C=color:  
   N=neutral, B=blue, R=red, P=pink, G=green  
   T=turquoise, Y=yellow, W=white/black  
 E/H=extended highlighting:  
   N=none, B=blink, R=reverse video, U=underscore

PANEL: PN0021 VERSION: 001 STATUS: TEST

## IDENTIFICATION:

Panel name PN0021

Created 11/04/87 By ETUD04  
 Last Modified 12/09/87 at 14:37 By ETUD04

Run Status PRIVATE  
 Short description options simplexe

## Description:

menu des options possibles pour travailler avec l'algorithme du  
 simplexe: creation d'un pls  
 execution d'un pls  
 suppression d'un pls  
 modification d'un pls

## PANEL PARAMETERS:

Start field symbol ^	New field symbol +
End field symbol ;	Delete field symbol *
Repeating group symbol à	
Input fill character S	(S=space, L=lowval, Z=zeros, U=_, other=itself)
Output fill character U	(S=space, L=lowval, U=_, other=itself)
Non-display character S	(S=space, other=as specified)
Error fill character *	(as specified)
Case translation U	(U=upper, M=mixed)
Required N	(Y=yes, N=no)
Error handling B	(N=none, *=fill w/errorfill, H=high intensity, (B=both: H if illegal value * if rqd missing)
PF1=HELP, PF3=CLARIFY Y	(Y=yes, N=no)
PF7=SCR -, PF8=SCR + N	(Y=yes, N=no)
PF10=SCR TOP, PF11=SCR BOT	
Edit-rule error proc C	(C=clarify command, A=application)
Process appl on scroll Y	(Y=yes, N=no)
Help panel name _____	Version _____
Prefix panel name _____	Version _____
Suffix panel name _____	Version _____

PANEL:

PN0021

VERSION: 001 STATUS: TEST

LAYOUT:

```
!.....1.....2.....3.....4.....5.....6.....7.....8!  
!      ^menu du programme d'optimisation;  
!      1  
!      ^-----;  
!      2  
  
!      ^1- creation d'un pls;  
!      3  
!      ^2- execution d'un pls;  
!      4  
!      ^3- suppression d'un pls;  
!      5  
!      ^4- modification d'un pls;  
!      6  
  
!      ^choix:; ^_;  
!      7      8  
  
!-----;  
!9  
!^pf6 pour fin de travail;  
!10  
!.....1.....2.....3.....4.....5.....6.....7.....8!
```

PANEL: PN0021 VERSION: 001 STATUS: TEST

## FIELD SUMMARY TABLE:

Seq	Lv	Field name	Attr	T	Len	In.dp	Occ	Comments
1	2		PSH	X	32			menu du programme d'o
2	2		PSL	X	32			
3	2		PSL	X	20			1- creation d'un pls
4	2		PSL	X	21			2- execution d'un pls
5	2		PSL	X	23			3- suppression d'un p
6	2		PSL	X	24			4- modification d'un
7	2		PSL	X	6			choix:
8	2	CHX	UNH	N	1	1		
9	2		PSL	X	76			
10	2		PSL	X	23			pf6 pour fin de trava

## LEGEND:

Seq=sequence number  
 Attr=screen attributes: U=unprot H=highlight A=327X alphanumeric  
 P=prot I=invisible N=327X numeric  
 S=skip L=low-light E=ensure input C=cursor

T=field type: X=alphanumeric, N=numeric  
 In.dp=Integer-places.decimal-places  
 Occ=number of occurrences

PANEL: PN0021 VERSION: 001 STATUS: TEST

## INPUT RULES:

Seq	Field name	E R	Minimum	Maximum	J I C	Mn D M A C M				
		H Q	value	value	S F S	DP S S C D F				
1		B N			L S U					N
2		B N			L S U					N
3		B N			L S U					N
4		B N			L S U					N
5		B N			L S U					N
6		B N			L S U					N
7		B N			L S U					N
8	CHX	B N	1		4 R Z U		N N N N			N
9		B N			L S U					N
10		B N			L S U					N

## LEGEND:

Seq=sequence number  
 E/H=error handling: N=none, \*=fill with \*, H=high intensity.  
                                 B=both (H if illegal value, \* if required field missing)  
 R/Q=required field: Y=yes, N=no, C=conditional  
 J/S=justification: N=none, L=left, R=right, A=align by decimal point  
 I/F=input fill character:  
                         S=spaces, L=low-values, Z=zeros, U=underscore, Other=itself  
 C/S=case: U=upper case, M=mixed case  
 Mn/DP=minimum required decimals (for justification=A)  
 D/S=allow digit separator: Y=yes, N=no  
 M/S=allow minus sign: Y=yes, N=no  
 A/C=allow currency symbol: Y=yes, N=no  
 C/D=check digit: N=none, T=modulo 10, E=modulo 11  
 M/F=must fill: Y=yes, N=no

N

PANEL: PN0021 VERSION: 001 STATUS: TEST

## OUTPUT RULES:

Seq	Field name	Edit pattern	O	E
-----			F	C H
1			U	N N
2			U	N N
3			U	N N
4			U	N N
5			U	N N
6			U	N N
7			U	N N
8	CHX	9	U	N N
9			U	N N
10			U	N N

## LEGEND:

Seq=sequence number

O/F=output fill character:

S=spaces, L=low-values, Z=zeros, U=underscore, Other=itself

C=color:

N=neutral, B=blue, R=red, P=pink, G=green

T=turquoise, Y=yellow, W=white/black

E/H=extended highlighting:

N=none, B=blink, R=reverse video, U=underscore

PANEL: PN0022 VERSION: 001 STATUS: TEST

## IDENTIFICATION:

Panel name PN0022

Created 11/04/87 By ETUD04  
 Last Modified 01/06/88 at 14:57 By ETUD04

Run Status PRIVATE  
 Short description liste des pls

Description:  
 affichage de la liste des pls enregistres dans la bd

---



---



---

## PANEL PARAMETERS:

Start field symbol ^	New field symbol +
End field symbol ;	Delete field symbol *
Repeating group symbol à	
Input fill character S	(S=space, L=lowval, Z=zeros, U=_, other=itself)
Output fill character U	(S=space, L=lowval, U=_, other=itself)
Non-display character S	(S=space, other=as specified)
Error fill character *	(as specified)
Case translation U	(U=upper, M=mixed)
Required N	(Y=yes, N=no)
Error handling B	(N=none, *=fill w/errorfill, H=high intensity, (B=both: H if illegal value * if rqd missing)
PF1=HELP, PF3=CLARIFY Y	(Y=yes, N=no)
PF7=SCR -, PF8=SCR + N	(Y=yes, N=no)
PF10=SCR TOP, PF11=SCR BOT	
Edit-rule error proc C	(C=clarify command, A=application)
Process appl on scroll Y	(Y=yes, N=no)
Help panel name _____	Version _____
Prefix panel name _____	Version _____
Suffix panel name _____	Version _____

PANEL:

PN0022

VERSION: 001 STATUS: TEST

LAYOUT:

```

!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!
!
!                                     ^liste des pls;
!                                     1
!                                     ^-----;
!                                     2
!
! ^pls numero ; ^____;
! 3                4
!
!                                     à _____;
!                                     6
!
!
!
! -----
! 7
! ^pf6 pour fin de travail;
! 8
!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!

```

PANEL: PN0022 VERSION: 001 STATUS: TEST

## FIELD SUMMARY TABLE:

Seq	Lv	Field name	Attr	T	Len	In.dp	Occ	Comments
1	2		PSH	X	13			liste des pls
2	2		PSH	X	13			-----
3	2		PSH	X	11			pls numero
4	2	PLS#	UAH	X	3			
5	2			G			8	
6	3	PLS#L	PSL	X	50			
7	2		PSL	X	77			-----
8	2		PSL	X	23			pf6 pour fin de trava

## LEGEND:

Seq=sequence number

Attr=screen attributes: U=unprot H=highlight A=327X alphanumeric  
 P=prot I=invisible N=327X numeric  
 S=skip L=low-light E=ensure input  
 C=cursor

T=field type: X=alphanumeric, N=numeric

In.dp=Integer-places.decimal-places

Occ=number of occurrences

PANEL: PN0022 VERSION: 001 STATUS: TEST

## INPUT RULES:

Seq	Field name	E R	Minimum H Q value	Maximum value	J	I	C	Mn	D	M	A	C	M
					S	F	S	DP	S	S	C	D	F
1		B	N		L	S	U						N
2		B	N		L	S	U						N
3		B	N		L	S	U						N
4	PLS#	B	N		L	S	U						N
5													
6	PLS#L	B	N		L	S	U						N
7		B	N		L	S	U						N
8		B	N		L	S	U						N

## LEGEND:

Seq=sequence number

E/H=error handling: N=none, \*=fill with \*, H=high intensity,

B=both (H if illegal value, \* if required field missing)

R/Q=required field: Y=yes, N=no, C=conditional

J/S=justification: N=none, L=left, R=right, A=align by decimal point

I/F=input fill character:

S=spaces, L=low-values, Z=zeroes, U=underscore, Other=itself

C/S=case: U=upper case, M=mixed case

Mn/DP=minimum required decimals (for justification=A)

D/S=allow digit separator: Y=yes, N=no

M/S=allow minus sign: Y=yes, N=no

A/C=allow currency symbol: Y=yes, N=no

C/D=check digit: N=none, T=modulo 10, E=modulo 11

M/F=must fill: Y=yes, N=no

A

PANEL: PN0022 VERSION: 001 STATUS: TEST

## OUTPUT RULES:

Seq	Field name	Edit pattern	O	F	C	H
1			U	N	N	
2			U	N	N	
3			U	N	N	
4	PLS#		U	N	N	
5						
6	PLS#L		U	N	N	
7			U	N	N	
8			U	N	N	

## LEGEND:

Seq=sequence number

O/F=output fill character:

S=spaces, L=low-values, Z=zeros, U=underscore, Other=itself

C=color:

N=neutral, B=blue, R=red, P=pink, G=green

T=turquoise, Y=yellow, W=white/black

E/H=extended highlighting:

N=none, B=blink, R=reverse video, U=underscore

PANEL: PN0023 VERSION: 001 STATUS: TEST

## IDENTIFICATION:

Panel name PN0023

Created 11/04/87 By ETUD04  
 Last Modified 01/06/88 at 15:11 By ETUD04

Run Status PRIVATE  
 Short description description pls creation

Description:  
 ecran de saisie des donnees generales du pls dans le cas d'une  
 creation

---



---

## PANEL PARAMETERS:

Start field symbol ^	New field symbol +
End field symbol ;	Delete field symbol *
Repeating group symbol à	
Input fill character S	(S=space, L=lowval, Z=zeros, U=_, other=itself)
Output fill character U	(S=space, L=lowval, U=_, other=itself)
Non-display character S	(S=space, other=as specified)
Error fill character *	(as specified)
Case translation U	(U=upper, M=mixed)
Required N	(Y=yes, N=no)
Error handling B	(N=none, *=fill w/errorfill, H=high intensity, B=both: H if illegal value * if rqd missing)
PF1=HELP, PF3=CLARIFY Y	(Y=yes, N=no)
PF7=SCR -, PF8=SCR + N	(Y=yes, N=no)
PF10=SCR TOP, PF11=SCR BOT	
Edit-rule error proc C	(C=clarify command, A=application)
Process appl on scroll Y	(Y=yes, N=no)
Help panel name _____	Version _____
Prefix panel name _____	Version _____
Suffix panel name _____	Version _____

PANEL:

PN0023

VERSION: 001 STATUS: TEST

LAYOUT:

```

!.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!
!^creation pls;                ^description generale du pls;
!1                             2
!                             ^-----;
!                             3
!^libelle du pls              ;;
!4
!                             à _____;
!                             6
!^numero du pls               ;; ^ ____;
!7                             8
!^nombre de variables         ;; ^ ____;
!9                             10
!^nombre de contraintes       ;; ^ ____;
!11                            12
!^min/max                     ;; ^ ____;
!13                            14
!
!
!-----
!15
!^pf6 pour fin de travail;
!16
!.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!

```

PANEL: PN0023 VERSION: 001 STATUS: TEST

## FIELD SUMMARY TABLE:

Seq	Lv	Field name	Attr	T	Len	In.dp	Occ	Comments
1	2		PSH	X	12			creation pls
2	2		PSH	X	27			description generale
3	2		PSH	X	27			
4	2		PSL	X	23			libelle du pls
5	2	GRP		G			8	
6	3	PLS#L	UAH	X	50			
7	2		PSL	X	23			numero du pls
8	2	PLS#	PSL	X	3			
9	2		PSL	X	23			nombre de variables
10	2	VAR#N	UNH	N	4	4		
11	2		PSL	X	23			nombre de contraintes
12	2	CTE#N	UNH	N	4	4		
13	2		PSL	X	23			min/max
14	2	OPT#V	UAH	X	3			
15	2		PSL	X	77			
16	2		PSL	X	23			pf6 pour fin de trava

## LEGEND:

Seq=sequence number  
Attr=screen attributes: U=unprot H=highlight A=327X alphanumeric  
P=prot I=invisible N=327X numeric  
S=skip L=low-light E=ensure input  
C=cursor

T=field type: X=alphanumeric, N=numeric  
In.dp=Integer-places.decimal-places  
Occ=number of occurrences

PANEL: PN0023 VERSION: 001 STATUS: TEST

## INPUT RULES:

Seq	Field name	E R	Minimum	Maximum	J I C	Mn D	M A C	M
		H Q	value	value	S F S	DP S	S S C D	F
1		B N			L S U			N
2		B N			L S U			N
3		B N			L S U			N
4		B N			L S U			N
5	GRP							
6	PLS#L	B N			L S U			N
7		B N			L S U			N
8	PLS#	B N			L S U			N
9		B N			L S U			N
10	VAR#N	B N	1	9999	R Z U		N N N N	N
11		B N			L S U			N
12	CTE#N	B N	1	9999	R Z U		N N N N	N
13		B N			L S U			N
14	OPT#V	B N			L S U			N
15		B N			L S U			N
16		B N			L S U			N

## LEGEND:

Seq=sequence number

E/H=error handling: N=none, \*=fill with \*, H=high intensity,

B=both (H if illegal value, \* if required field missing)

R/Q=required field: Y=yes, N=no, C=conditional

J/S=justification: N=none, L=left, R=right, A=align by decimal point

I/F=input fill character:

S=spaces, L=low-values, Z=zeroes, U=underscore, Other=itself

C/S=case: U=upper case, M=mixed case

Mn/DP=minimum required decimals (for justification=A)

D/S=allow digit separator: Y=yes, N=no

M/S=allow minus sign: Y=yes, N=no

A/C=allow currency symbol: Y=yes, N=no

C/D=check digit: N=none, T=modulo 10, E=modulo 11

M/F=must fill: Y=yes, N=no

PANEL: PN0023 VERSION: 001 STATUS: TEST

## OUTPUT RULES:

Seq	Field name	Edit pattern	O	F	C	H	E
1			U	N	N		
2			U	N	N		
3			U	N	N		
4			U	N	N		
5	GRP						
6	PLS#L		U	N	N		
7			U	N	N		
8	PLS#		U	N	N		
9			U	N	N		
10	VAR#N	ZZZ9	U	N	N		
11			U	N	N		
12	CTE#N	ZZZ9	U	N	N		
13			U	N	N		
14	OPT#V		U	N	N		
15			U	N	N		
16			U	N	N		

## LEGEND:

Seq=sequence number

O/F=output fill character:

S=spaces, L=low-values, Z=zeros, U=underscore, Other=itself

C=color:

N=neutral, B=blue, R=red, P=pink, G=green

T=turquoise, Y=yellow, W=white/black

E/H=extended highlighting:

N=none, B=blink, R=reverse video, U=underscore

PANEL: PN0024 VERSION: 001 STATUS: TEST

## IDENTIFICATION:

Panel name PN0024

Created 11/04/87 By ETUD04  
 Last Modified 01/06/88 at 15:39 By ETUD04

Run Status PRIVATE  
 Short description couts creation

Description:  
 saisie des donnees des couts dans le cas d'une creation

---



---



---

## PANEL PARAMETERS:

Start field symbol ^	New field symbol +
End field symbol ;	Delete field symbol *
Repeating group symbol à	
Input fill character S	(S=space, L=lowval, Z=zeros, U=_, other=itself)
Output fill character U	(S=space, L=lowval, U=_, other=itself)
Non-display character S	(S=space, other=as specified)
Error fill character *	(as specified)
Case translation U	(U=upper, M=mixed)
Required N	(Y=yes, N=no)
Error handling B	(N=none, *=fill w/errorfill, H=high intensity.) (B=both: H if illegal value * if rqd missing)
PF1=HELP, PF3=CLARIFY Y	(Y=yes, N=no)
PF7=SCR -, PF8=SCR + N	(Y=yes, N=no)
PF10=SCR TOP, PF11=SCR BOT	
Edit-rule error proc C	(C=clarify command, A=application)
Process appl on scroll Y	(Y=yes, N=no)
Help panel name _____	Version _____
Prefix panel name _____	Version _____
Suffix panel name _____	Version _____

PANEL:

PN0024

VERSION: 001 STATUS: TEST

LAYOUT:

```

!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!
!^creation pls; ^____;
!1                2
!                ^saisie des couts;
!                3
!                ^-----;
!                4
!^nombre de variables ;; ^____;
!5                6
!
!àcout;à____;à=; à____;
!8    9    10  11
!
!
!-----;
!12
!^pf6 pour fin de travail;
!13
!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!

```

PANEL: PN0024 VERSION: 001 STATUS: TEST

## FIELD SUMMARY TABLE:

Seq	Lv	Field name	Attr	T	Len	In.dp	Occ	Comments
1	2		PSH	X	12			creation pls
2	2	PLS#	PAH	X	3			
3	2		PSH	X	16			saisie des couts
4	2		PSH	X	16			-----
5	2		PSL	X	21			nombre de variables :
6	2	VAR#N	PSL	N	4	4		
7	2	GRP		G				*
8	3		PSL	X	4			cout
9	3	CPT	PSL	N	4	4		
10	3		PSL	X	1			=
11	3	CRL#V	UNH	N	17	7.8		
12	2		PSL	X	77			-----
13	2		PSL	X	23			pf6 pour fin de trava

## LEGEND:

Seq=sequence number  
Attr=screen attributes: U=unprot H=highlight A=327X alphanumeric  
P=prot I=invisible N=327X numeric  
S=skip L=low-light E=ensure input  
C=cursor

T=field type: X=alphanumeric, N=numeric  
In.dp=Integer-places.decimal-places  
Occ=number of occurrences

PANEL: PN0024 VERSION: 001 STATUS: TEST

INPUT RULES:

Seq	Field name	E R	Minimum	Maximum	J I C	Mn D	M A C	M
		H Q	value	value	S F S	DP S S	C D F	
1		B N			L S U			N
2	PLS#	B N			L S U			N
3		B N			L S U			N
4		B N			L S U			N
5		B N			L S U			N
6	VAR#N	B N			R Z U	N N N N		N
7	GRP							
8		B N			L S U			N
9	CPT	B N			R Z U	N N N N		N
10		B N			L S U			N
11	CRL#V	B N			A Z U	O Y Y Y	N N	N
12		B N			L S U			N
13		B N			L S U			N

LEGEND:

Seq=sequence number  
 E/H=error handling: N=none, \*=fill with \*, H=high intensity,  
 B=both (H if illegal value, \* if required field missing)  
 R/Q=required field: Y=yes, N=no, C=conditional  
 J/S=justification: N=none, L=left, R=right, A=align by decimal point  
 I/F=input fill character:  
 S=spaces, L=low-values, Z=zeroes, U=underscore, Other=itself  
 C/S=case: U=upper case, M=mixed case  
 Mn/DP=minimum required decimals (for justification=A)  
 D/S=allow digit separator: Y=yes, N=no  
 M/S=allow minus sign: Y=yes, N=no  
 A/C=allow currency symbol: Y=yes, N=no  
 C/D=check digit: N=none, T=modulo 10, E=modulo 11  
 M/F=must fill: Y=yes, N=no

S

PANEL:

PN0024

VERSION: 001 STATUS: TEST

OUTPUT RULES:

Seq	Field name	Edit pattern	O	F	C	H	E
1			U	N	N		
2	PLS#		U	N	N		
3			U	N	N		
4			U	N	N		
5			U	N	N		
6	VAR#N	ZZZ9	U	N	N		
7	GRP						
8			U	N	N		
9	CPT	ZZZ9	U	N	N		
10			U	N	N		
11	CRL#V	-----9.99999999	U	N	N		
12			U	N	N		
13			U	N	N		

LEGEND:

Seq=sequence number

O/F=output fill character:

S=spaces, L=low-values, Z=zeroes, U=underscore, Other=itself

C=color:

N=neutral, B=blue, R=red, P=pink, G=green

T=turquoise, Y=yellow, W=white/black

E/H=extended highlighting:

N=none, B=blink, R=reverse video, U=underscore

PANEL: PN0025 VERSION: 001 STATUS: TEST

## IDENTIFICATION:

Panel name PN0025

Created 11/04/87 By ETUD04  
 Last Modified 01/06/88 at 15:41 By ETUD04

Run Status PRIVATE

Short description contraintes creation

## Description:

saisie des donnees des contraintes dans le cas d'une creation

---



---



---

## PANEL PARAMETERS:

Start field symbol ^	New field symbol +
End field symbol ;	Delete field symbol *
Repeating group symbol a	
Input fill character S	(S=space, L=lowval, Z=zeros, U=_, other=itself)
Output fill character U	(S=space, L=lowval, U=_, other=itself)
Non-display character S	(S=space, other=as specified)
Error fill character *	(as specified)
Case translation U	(U=upper, M=mixed)
Required N	(Y=yes, N=no)
Error handling B	(N=none, *=fill w/errorfill, H=high intensity, (B=both: H if illegal value * if rqd missing)
PF1=HELP, PF3=CLARIFY Y	(Y=yes, N=no)
PF7=SCR -, PF8=SCR + N	(Y=yes, N=no)
PF10=SCR TOP, PF11=SCR BOT	
Edit-rule error proc C	(C=clarify command, A=application)
Process appl on scroll Y	(Y=yes, N=no)
Help panel name _____	Version _____
Prefix panel name _____	Version _____
Suffix panel name _____	Version _____

PANEL: PN0025 VERSION: 001 STATUS: TEST

LAYOUT:

```

!.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!
!^creation pls;^____;
!1
!^nombre de contraintes ;;^____;
!3
!^saisie de la contrainte;^____;
!5
!^-----^
!7
!àcoefficient;à____;à=;à____;
!9
!^terme independant =;^____;
!13
!^signe ;;^____;
!15
!-----^
!17
!^pf6 pour fin de travail;
!18
!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!

```

PANEL: PN0025 VERSION: 001 STATUS: TEST

## FIELD SUMMARY TABLE:

Seq	Lv	Field name	Attr	T	Len	In.dp	Occ	Comments
1	2		PSH	X	12			creation pls
2	2	PLS#	PAH	X	3			
3	2		PSH	X	23			nombre de contraintes
4	2	CTE#N	PSH	N	4	4		
5	2		PAH	X	23			saisie de la contrain
6	2	CTE#	PAH	N	4	4		
7	2		PAH	X	29			-----
8	2	GRP		G			*	
9	3		PSL	X	11			coefficient
10	3	COE#	PSL	N	4	4		
11	3		PSL	X	1			=
12	3	COE#V	UNH	N	17	7.8		
13	2		PSL	X	20			terme independant =
14	2	TI#V	UNH	N	17	7.8		
15	2		PSL	X	20			signe :
16	2	SGN-CTE	UAH	X	2			
17	2		PSL	X	77			-----
18	2		PSL	X	23			pf6 pour fin de trava

## LEGEND:

Seq=sequence number

Attr=screen attributes: U=unprot H=highlight A=327X alphanumeric  
P=prot I=invisible N=327X numeric  
S=skip L=low-light E=ensure input  
C=cursor

T=field type: X=alphanumeric, N=numeric

In.dp=Integer-places.decimal-places

Occ=number of occurrences

PANEL: PN0025 VERSION: 001 STATUS: TEST

## INPUT RULES:

Seq	Field name	E R	Minimum	Maximum	J I C	Mn D M A C M					
		H Q	value	value	S F S	DP S S C D F					
1		B N			L S U						N
2	PLS#	B N			L S U						N
3		B N			L S U						N
4	CTE#N	B N			R Z U		N	N	N	N	N
5		B N			L S U						N
6	CTE#	B N			R Z U		N	N	N	N	N
7		B N			L S U						N
8	GRP										
9		B N			L S U						N
10	COE#	B N			R Z U		N	N	N	N	N
11		B N			L S U						N
12	COE#V	B N			A Z U		O	Y	Y	Y	N
13		B N			L S U						N
14	TI#V	B N			A Z U		O	Y	Y	Y	N
15		B N			L S U						N
16	SGN-CTE	B N			L S U						N
17		B N			L S U						N
18		B N			L S U						N

## LEGEND:

Seq=sequence number

E/H=error handling: N=none, \*=fill with \*, H=high intensity.

B=both (H if illegal value, \* if required field missing)

R/Q=required field: Y=yes, N=no, C=conditional

J/S=justification: N=none, L=left, R=right, A=align by decimal point

I/F=input fill character:

S=spaces, L=low-values, Z=zeros, U=underscore, Other=itself

C/S=case: U=upper case, M=mixed case

Mn/DP=minimum required decimals (for justification=A)

D/S=allow digit separator: Y=yes, N=no

M/S=allow minus sign: Y=yes, N=no

A/C=allow currency symbol: Y=yes, N=no

C/D=check digit: N=none, T=modulo 10, E=modulo 11

M/F=must fill: Y=yes, N=no

PANEL: PN0219 VERSION: 001 STATUS: TEST

## INPUT RULES:

Seq	Field name	E R	Minimum	Maximum	J I C	Mn	D	M	A	C	M
		H Q	value	value	S F S	DP	S	S	C	D	F
1		B N			L S U						N
2		B N			L S U						N
3		B N			L S U						N
4	PLS#	B N			L S U						N
5		B N			L S U						N
6	GRP										
7		B N			L S U						N
8	CPT	B N			R Z U		N	N	N	N	N
9		B N			L S U						N
10	XB#V	B N			R Z U		Y	Y	Y	N	N
11		B N			L S U						N
12	CP2	B N			R Z U		N	N	N	N	N
13		B N			L S U						N
14	XB2#V	B N			A Z U	0	Y	Y	Y	N	N
15		B N			L S U						N
16		B N			L S U						N
17		B N			L S U						N

## LEGEND:

Seq=sequence number

E/H=error handling: N=none, \*=fill with \*, H=high intensity,

B=both (H if illegal value, \* if required field missing)

R/Q=required field: Y=yes, N=no, C=conditional

J/S=justification: N=none, L=left, R=right, A=align by decimal point

I/F=input fill character:

S=spaces, L=low-values, Z=zeroes, U=underscore, Other=itself

C/S=case: U=upper case, M=mixed case

Mn/DP=minimum required decimals (for justification=A)

D/S=allow digit separator: Y=yes, N=no

M/S=allow minus sign: Y=yes, N=no

A/C=allow currency symbol: Y=yes, N=no

C/D=check digit: N=none, T=modulo 10, E=modulo 11

M/F=must fill: Y=yes, N=no

PANEL: PN0219 VERSION: 001 STATUS: TEST

## OUTPUT RULES:

Seq	Field name	Edit pattern	O	F	C	H
1			U	N	N	
2			U	N	N	
3			U	N	N	
4	PLS#		U	N	N	
5			U	N	N	
6	GRP		U	N	N	
7			U	N	N	
8	CPT	ZZZ9	U	N	N	
9			U	N	N	
10	XB#V	-----9.99999999	U	N	N	
11			U	N	N	
12	CP2	ZZZ9	U	N	N	
13			U	N	N	
14	XB2#V	-----9.99999999	U	N	N	
15			U	N	N	
16			U	N	N	
17			U	N	N	

## LEGEND:

Seq=sequence number  
O/F=output fill character:  
S=spaces, L=low-values, Z=zeroes, U=underscore, Other=itself  
C=color:  
N=neutral, B=blue, R=red, P=pink, G=green  
T=turquoise, Y=yellow, W=white/black  
E/H=extended highlighting:  
N=none, B=blink, R=reverse video, U=underscore

PANEL: PNO220 VERSION: 001 STATUS: TEST

## IDENTIFICATION:

Panel name PNO220

Created 11/05/87 By ETUD04  
 Last Modified 12/09/87 at 15:13 By ETUD04

Run Status PRIVATE  
 Short description res plan de vente 1

## Description:

ecran de rappel du plan de vente considere et de sortie de quelq  
 ues proprietes particulieres au probleme

---



---

## PANEL PARAMETERS:

Start	field symbol ^	New	field symbol +
End	field symbol ;	Delete	field symbol *
Repeating	group symbol à		
Input fill character	S	(S=space, L=lowval, Z=zeros, U=_, other=itself)	
Output fill character	U	(S=space, L=lowval, U=_, other=itself)	
Non-display character	S	(S=space, other=as specified)	
Error fill character	*	(as specified)	
Case translation	U	(U=upper, M=mixed)	
Required	N	(Y=yes, N=no)	
Error handling	B	(N=none, *=fill w/errorfill, H=high intensity, (B=both: H if illegal value * if rqd missing)	
PF1=HELP, PF3=CLARIFY	Y	(Y=yes, N=no)	
PF7=SCR -, PF8=SCR +	N	(Y=yes, N=no)	
PF10=SCR TOP, PF11=SCR BOT			
Edit-rule error proc	C	(C=clarify command, A=application)	
Process appl on scroll	Y	(Y=yes, N=no)	
Help panel name	_____	Version	___
Prefix panel name	_____	Version	___
Suffix panel name	_____	Version	___

PANEL: PN0220 VERSION: 001 STATUS: TEST

```

LAYOUT:
!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!
!                                     ^resultats plan de vente;
!                                     1
!                                     ^-----;
!                                     2
!^plan de vente numero ;^____;
!3                                     4
!^libelle du plan de vente ::
!5
!                                     à _____;
!                                     7
!^nombre de variables                ;;^____;
!8                                     9
!^nombre de contraintes                ;;^____;
!10                                    11
!^nombre de variables d'ecart          ;;^____;
!12                                    13
!^nombre de variables artificielles ;;^____;
!14                                    15
!
!-----;
!16
!^pf6 pour fin de travail;
!17
!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!

```

PANEL: PN0220 VERSION: 001 STATUS: TEST

## FIELD SUMMARY TABLE:

Seq	Lv	Field name	Attr	T	Len	In.dp	Occ	Comments
1	2		PSH	X	23			resultats plan de ven
2	2		PSH	X	23			-----
3	2		PSH	X	21			plan de vente numero
4	2	PBL#	PAH	X	3			
5	2		PSL	X	26			libelle du plan de ve
6	2	GRP		G			*	
7	3	PBL#L	PAL	X	50			
8	2		PSL	X	35			nombre de variables
9	2	VAR#N	PSL	N	4	4		
10	2		PSL	X	35			nombre de contraintes
11	2	CTE#N	PSL	N	4	4		
12	2		PSL	X	35			nombre de variables d
13	2	ECR-VAR#N	PSL	N	4	4		
14	2		PSL	X	35			nombre de variables a
15	2	ARF-VAR#N	PSL	N	4	4		
16	2		PSL	X	77			-----
17	2		PSL	X	23			pf6 pour fin de trava

## LEGEND:

Seq=sequence number

Attr=screen attributes: U=unprot H=highlight A=327X alphanumeric

P=prot I=invisible N=327X numeric

S=skip L=low-light E=ensure input

C=cursor

T=field type: X=alphanumeric, N=numeric

In.dp=Integer-places.decimal-places

Occ=number of occurrences

PANEL: PN0220 VERSION: 001 STATUS: TEST

## INPUT RULES:

Seq	Field name	E	R	Minimum	Maximum	J	I	C	Mn	D	M	A	C	M
		H	Q	value	value	S	F	S	DP	S	S	C	D	F
1		B	N			L	S	U						N
2		B	N			L	S	U						N
3		B	N			L	S	U						N
4	PBL#	B	N			L	S	U						N
5		B	N			L	S	U						N
6	GRP													
7	PBL#L	B	N			L	S	U						N
8		B	N			L	S	U						N
9	VAR#N	B	N			R	Z	U		N	N	N	N	N
10		B	N			L	S	U						N
11	CTE#N	B	N			R	Z	U		N	N	N	N	N
12		B	N			L	S	U						N
13	ECR-VAR#N	B	N			R	Z	U		N	N	N	N	N
14		B	N			L	S	U						N
15	ARF-VAR#N	B	N			R	Z	U		N	N	N	N	N
16		B	N			L	S	U						N
17		B	N			L	S	U						N

## LEGEND:

Seq=sequence number

E/H=error handling: N=none, \*=fill with \*, H=high intensity,  
B=both (H if illegal value, \* if required field missing)

R/Q=required field: Y=yes, N=no, C=conditional

J/S=justification: N=none, L=left, R=right, A=align by decimal point

I/F=input fill character:

S=spaces, L=low-values, Z=zeroes, U=underscore, Other=itself

C/S=case: U=upper case, M=mixed case

Mn/DP=minimum required decimals (for justification=A)

D/S=allow digit separator: Y=yes, N=no

M/S=allow minus sign: Y=yes, N=no

A/C=allow currency symbol: Y=yes, N=no

C/D=check digit: N=none, T=modulo 10, E=modulo 11

M/F=must fill: Y=yes, N=no

PANEL: PN0220 VERSION: 001 STATUS: TEST

## OUTPUT RULES:

Seq	Field name	Edit pattern	O	E
-----			F	C
-----			H	
1			U	N N
2			U	N N
3			U	N N
4	PBL#		U	N N
5			U	N N
6	GRP			
7	PBL#L		U	N N
8			U	N N
9	VAR#N	ZZZ9	U	N N
10			U	N N
11	CTE#N	ZZZ9	U	N N
12			U	N N
13	ECR-VAR#N	ZZZ9	U	N N
14			U	N N
15	ARF-VAR#N	ZZZ9	U	N N
16			U	N N
17			U	N N

## LEGEND:

Seq=sequence number

O/F=output fill character:

S=spaces, L=low-values, Z=zeros, U=underscore, Other=itself

C=color:

N=neutral, B=blue, R=red, P=pink, G=green

T=turquoise, Y=yellow, W=white/black

E/H=extended highlighting:

N=none, B=blink, R=reverse video, U=underscore

PANEL: PN0221 VERSION: 001 STATUS: TEST

## IDENTIFICATION:

Panel name PN0221

Created 11/05/87 By ETUD04  
 Last Modified 12/21/87 at 09:59 By ETUD04

Run Status PRIVATE  
 Short description res plan de vente 2

Description:  
 ecran de sortie des resultats de l'optimisation du plan de vente

## PANEL PARAMETERS:

Start field symbol ^	New field symbol +
End field symbol ;	Delete field symbol *
Repeating group symbol à	
Input fill character S	(S=space, L=lowval, Z=zeros, U=_, other=itself)
Output fill character U	(S=space, L=lowval, U=_, other=itself)
Non-display character S	(S=space, other=as specified)
Error fill character *	(as specified)
Case translation U	(U=upper, M=mixed)
Required N	(Y=yes, N=no)
Error handling B	(N=none, *=fill w/errorfill, H=high intensity, B=both: H if illegal value * if rqd missing)
PF1=HELP, PF3=CLARIFY Y	(Y=yes, N=no)
PF7=SCR -, PF8=SCR + N	(Y=yes, N=no)
PF10=SCR TOP, PF11=SCR BOT	
Edit-rule error proc C	(C=clarify command, A=application)
Process appl on scroll Y	(Y=yes, N=no)
Help panel name _____	Version _____
Prefix panel name _____	Version _____
Suffix panel name _____	Version _____

PANEL:

PN0221

VERSION: 001 STATUS: TEST

LAYOUT:

```

!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!
!          ^resultats de l'optimisation du plan de vente;
!          1-----;
!          2
!^plan de vente;^____;
!3          4
!^action ;^_____
!5          6

!^nombre d'envois pour la couche;^____;
!7          8
!          àtranche;à____;à::à_____
!          10      11      12 13

!-----;
!14
!^pf6 pour fin de travail;
!15
!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!

```

PANEL: PN0221 VERSION: 001 STATUS: TEST

## FIELD SUMMARY TABLE:

Seq	Lv	Field name	Attr	T	Len	In.dp	Occ	Comments
1	2		PSH	X	44			resultats de l'optimi
2	2		PSH	X	44			
3	2		PSH	X	13			plan de vente
4	2	PBL#	PAH	X	3			
5	2		PSH	X	7			action
6	2	ACT#L	PAL	X	50			
7	2		PSL	X	30			nombre d'envois pour
8	2	COU#A	PAL	X	3			
9	2			G			3	
10	3		PSL	X	7			tranche
11	3	TRA	PSL	N	3	3		
12	3		PSL	X	1			:
13	3	XB#V	PSH	N	17	7.8		
14	2		PSL	X	77			
15	2		PSL	X	23			pf6 pour fin de trava

## LEGEND:

Seq=sequence number

Attr=screen attributes: U=unprot H=highlight A=327X alphanumeric  
 P=prot I=invisible N=327X numeric  
 S=skip L=low-light E=ensure input  
 C=cursor

T=field type: X=alphanumeric, N=numeric

In.dp=Integer-places.decimal-places

Occ=number of occurrences

PANEL: PN0221 VERSION: 001 STATUS: TEST

## INPUT RULES:

Seq	Field name	E	R	Minimum	Maximum	J	I	C	Mn	D	M	A	C	M
		H	Q	value	value	S	F	S	DP	S	S	C	D	F
1		B	N			L	S	U						N
2		B	N			L	S	U						N
3		B	N			L	S	U						N
4	PBL#	B	N			L	S	U						N
5		B	N			L	S	U						N
6	ACT#L	B	N			L	S	U						N
7		B	N			L	S	U						N
8	COU#A	B	N			L	S	U						N
9														
10		B	N			L	S	U						N
11	TRA	B	N			R	Z	U		N	N	N	N	N
12		B	N			L	S	U						N
13	XB#V	B	N			A	Z	U		D	Y	Y	Y	N
14		B	N			L	S	U						N
15		B	N			L	S	U						N

## LEGEND:

Seq=sequence number

E/H=error handling: N=none, \*=fill with \*, H=high intensity,

B=both (H if illegal value, \* if required field missing)

R/Q=required field: Y=yes, N=no, C=conditional

J/S=justification: N=none, L=left, R=right, A=align by decimal point

I/F=input fill character:

S=spaces, L=low-values, Z=zeros, U=underscore, Other=itself

C/S=case: U=upper case, M=mixed case

Mn/DP=minimum required decimals (for justification=A)

D/S=allow digit separator: Y=yes, N=no

M/S=allow minus sign: Y=yes, N=no

A/C=allow currency symbol: Y=yes, N=no

C/D=check digit: N=none, T=modulo 10, E=modulo 11

M/F=must fill: Y=yes, N=no

PANEL: PN0221 VERSION: 001 STATUS: TEST

## OUTPUT RULES:

Seq	Field name	Edit pattern	O	E
-----			F	C H
1			U	N N
2			U	N N
3			U	N N
4	PBL#		U	N N
5			U	N N
6	ACT#L		U	N N
7			U	N N
8	COU#A		U	N N
9				
10			U	N N
11	TRA	ZZ9	U	N N
12			U	N N
13	XB#V	-----9.99999999	U	N N
14			U	N N
15			U	N N

## LEGEND:

Seq=sequence number

O/F=output fill character:

S=spaces, L=low-values, Z=zeroes, U=underscore, Other=itself

C=color:

N=neutral, B=blue, R=red, P=pink, G=green

T=turquoise, Y=yellow, W=white/black

E/H=extended highlighting:

N=none, B=blink, R=reverse video, U=underscore

PANEL: PNO222 VERSION: 001 STATUS: TEST

## IDENTIFICATION:

Panel name PNO222

Created 01/06/88 By ETUD04  
 Last Modified 01/06/88 at 14:13 By ETUD04

Run Status PRIVATE  
 Short description menu pvc

Description:

\_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

## PANEL PARAMETERS:

Start field symbol ^	New field symbol +
End field symbol ;	Delete field symbol *
Repeating group symbol à	
Input fill character S	(S=space, L=lowval, Z=zeros, U=_, other=itself)
Output fill character U	(S=space, L=lowval, U=_, other=itself)
Non-display character S	(S=space, other=as specified)
Error fill character *	(as specified)
Case translation U	(U=upper, M=mixed)
Required N	(Y=yes, N=no)
Error handling B	(N=none, *=fill w/errorfill, H=high intensity, (B=both: H if illegal value * if rqd missing)
PF1=HELP, PF3=CLARIFY Y	(Y=yes, N=no)
PF7=SCR -, PF8=SCR + N	(Y=yes, N=no)
PF10=SCR TOP, PF11=SCR BOT	
Edit-rule error proc C	(C=clarify command, A=application)
Process appl on scroll Y	(Y=yes, N=no)
Help panel name _____	Version _____
Prefix panel name _____	Version _____
Suffix panel name _____	Version _____

PANEL:

PN0222

VERSION: 001 STATUS: TEST

LAYOUT:

```

!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!
!          ^menu pvc;
!          1
!          ^-----;
!          2
!
!          ^1. creation d'un pvc;
!          3
!          ^2. execution d'un pvc;
!          4
!          ^3. suppression d'un pvc;
!          5
!
!          ^votre choix ::^_:
!          6          7
!
!-----;
!8
!^$pf6 pour fin de travail;
!9
!.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8!

```

PANEL: PNO222 VERSION: 001 STATUS: TEST

## FIELD SUMMARY TABLE:

Seq	Lv	Field name	Attr	T	Len	In.dp	Occ	Comments
1	2		PSL	X	8			menu pvc
2	2		PSL	X	8			-----
3	2		PSL	X	20			1. creation d'un pvc
4	2		PSL	X	21			2. execution d'un pvc
5	2		PSL	X	23			3. suppression d'un p
6	2		PSL	X	13			votre choix :
7	2	CHX	UAH	N	1	1		
8	2		PSL	X	76			-----
9	2		PSL	X	24			\$pf6 pour fin de trav

## LEGEND:

Seq=sequence number  
 Attr=screen attributes: U=unprot H=highlight A=327X alphanumeric  
 P=prot I=invisible N=327X numeric  
 S=skip L=low-light E=ensure input  
 C=cursor

T=field type: X=alphanumeric, N=numeric  
 In.dp=Integer-places.decimal-places  
 Occ=number of occurrences

PANEL: PNO222 VERSION: 001 STATUS: TEST

## INPUT RULES:

Seq	Field name	E	R	Minimum	Maximum	J	I	C	Mn	D	M	A	C	M
		H	Q	value	value	S	F	S	DP	S	S	C	D	F
1		B	N			L	S	U						N
2		B	N			L	S	U						N
3		B	N			L	S	U						N
4		B	N			L	S	U						N
5		B	N			L	S	U						N
6		B	N			L	S	U						N
7	CHX	B	N		1	3	R	Z	U		N	N	N	N
8		B	N			L	S	U						N
9		B	N			L	S	U						N

## LEGEND:

Seq=sequence number

E/H=error handling: N=none, \*=fill with \*, H=high intensity.

B=both (H if illegal value, \* if required field missing)

R/Q=required field: Y=yes, N=no, C=conditional

J/S=justification: N=none, L=left, R=right, A=align by decimal point

I/F=input fill character:

S=spaces, L=low-values, Z=zeros, U=underscore, Other=itself

C/S=case: U=upper case, M=mixed case

Mn/DP=minimum required decimals (for justification=A)

D/S=allow digit separator: Y=yes, N=no

M/S=allow minus sign: Y=yes, N=no

A/C=allow currency symbol: Y=yes, N=no

C/D=check digit: N=none, T=modulo 10, E=modulo 11

M/F=must fill: Y=yes, N=no

PANEL: PN0222 VERSION: 001 STATUS: TEST

## OUTPUT RULES:

Seq	Field name	Edit pattern	O	F	C	H	E
1			U	N	N		
2			U	N	N		
3			U	N	N		
4			U	N	N		
5			U	N	N		
6			U	N	N		
7	CHX	9	U	N	N		
8			U	N	N		
9			U	N	N		

## LEGEND:

Seq=sequence number

O/F=output fill character:

S=spaces, L=low-values, Z=zeros, U=underscore, Other=itself

C=color:

N=neutral, B=blue, R=red, P=pink, G=green

T=turquoise, Y=yellow, W=white/black

E/H=extended highlighting:

N=none, B=blink, R=reverse video, U=underscore