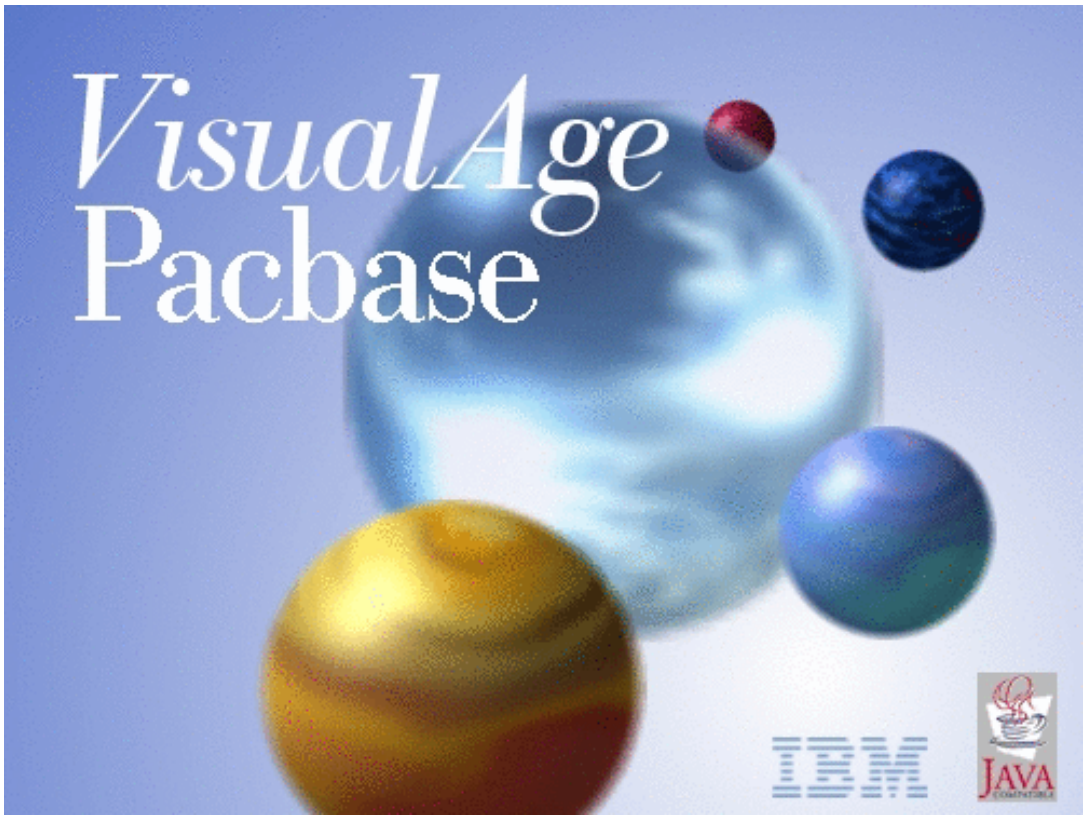
VisualAge Pacbase

# Guide d'utilisation de l'API COBOL

*Version 3.5*



VisualAge Pacbase

# Guide d'utilisation de l'API COBOL

*Version 3.5*

**Note**

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Notices» à la page v.

En application de votre contrat de licence, vous pouvez consulter ou télécharger la documentation de VisualAge Pacbase, régulièrement mise à jour, à partir de :

http://www.ibm.com/software/awdtools/vapacbase/productinfo_f.htm

La section Catalogue dans la page d'accueil de la Documentation vous permet d'identifier la dernière édition disponible du présent document.

**Deuxième édition (Mars 2004)**

La présente édition s'applique à :
- VisualAge Pacbase Version 3.5

Vous pouvez nous adresser tout commentaire sur ce document (en indiquant sa référence) via le site Web de notre Support Technique à l'adresse suivante : http://www.ibm.com/software/awdtools/vapacbase/support.htm ou en nous adressant un courrier à :

IBM Paris Laboratory
1, place Jean–Baptiste Clément
93881 Noisy-le-Grand, France.

IBM pourra disposer comme elle l'entendra des informations contenues dans vos commentaires, sans aucune obligation de sa part.

# Table des matières

# Notices

Ce document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM. Cela ne signifie pas qu'IBM ait l'intention de les annoncer dans tous les pays où la compagnie est présente. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante : IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk NY 10504–1785, U.S.A.

Les détenteurs de licences du présent produit souhaitant obtenir des informations sur celui-ci à des fins : (i) d'échange d'informations entre des programmes développés indépendamment et d'autres programmes (y compris celui-ci) et (ii) d'utilisation mutuelle des informations ainsi échangées doivent s'adresser à : IBM Paris Laboratory, SMC Department, 1 place J.B.Clément, 93881 Noisy-Le-Grand Cedex, France. De telles informations peuvent être mises à la disposition du Client et seront soumises aux termes et conditions appropriés, y compris dans certains cas au paiement d'une redevance.

IBM peut modifier ce document, le produit qu'il décrit ou les deux.

# Marques

IBM est une marque d'International Business Machines Corporation, Inc. AIX, AS/400, CICS, CICS/MVS, CICS/VSE, COBOL/2, DB2, IMS, MQSeries, OS/2, VisualAge Pacbase, RACF, RS/6000, SQL/DS et VisualAge sont des marques d'International Business Machines Corporation, Inc. dans certains pays.

Java et toutes les marques et logos incluant Java sont des marques de Sun Microsystems, Inc. dans certains pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation dans certains pays.

UNIX est une marque enregistrée aux Etats-Unis et/ou dans d'autres pays et utilisée avec l'autorisation exclusive de la société X/Open Company Limited.

D'autres sociétés peuvent être propriétaires des autres marques, noms de produits ou logos qui pourraient apparaître dans ce document.

# Chapitre 1. Introduction

L'objet de ce guide est de vous présenter toutes les informations nécessaires à l'utilisation de l'API COBOL à partir de Developer workbench.

L'API COBOL vous permet d'utiliser les Dossiers et les Composants Elémentaires non dans le cadre de Proxies mais de façon totalement ouverte, à l'intérieur de programmes applicatifs. Ces programmes bénéficient alors de la hiérarchie et de la réutilisation des informations du Dossier, du regroupement de ses traitements. Vous utilisez donc le Dossier en lui apportant des entrées externes à l'Application eBusiness.

**Exemple :** Pour gérer les prélèvements automatiques à date fixe de ses clients, une banque peut utiliser l'entité Programme qui va exploiter les informations du Dossier "Clients".

Il est conseillé d'écrire le programme applicatif dans l'entité Programme de VA Pac. Cependant, il est aussi possible de l'écrire dans l'entité Composant Elémentaire ou Ecran.

L'API COBOL vous permet d'indiquer des demandes de services (mise à jour, édition de listes...) sur le Dossier appelé dans votre programme applicatif. Vous bénéficiez pour cela de facilités d'écriture avec :

- des opérateurs en langage structuré qui vont formater le message en entrée du Dossier,
- la description COBOL des données de la zone de communication en entrée du Dossier.

Pour générer le programme applicatif contenant l'appel aux entités eBusiness, vous devez exécuter la procédure batch GPRC. Cette procédure est expliquée dans le 'Guide du Développeur' spécifique à votre plateforme, dans le chapitre 'Editions-générations'.

# Chapitre 2. Déclaration des zones de working

## Introduction

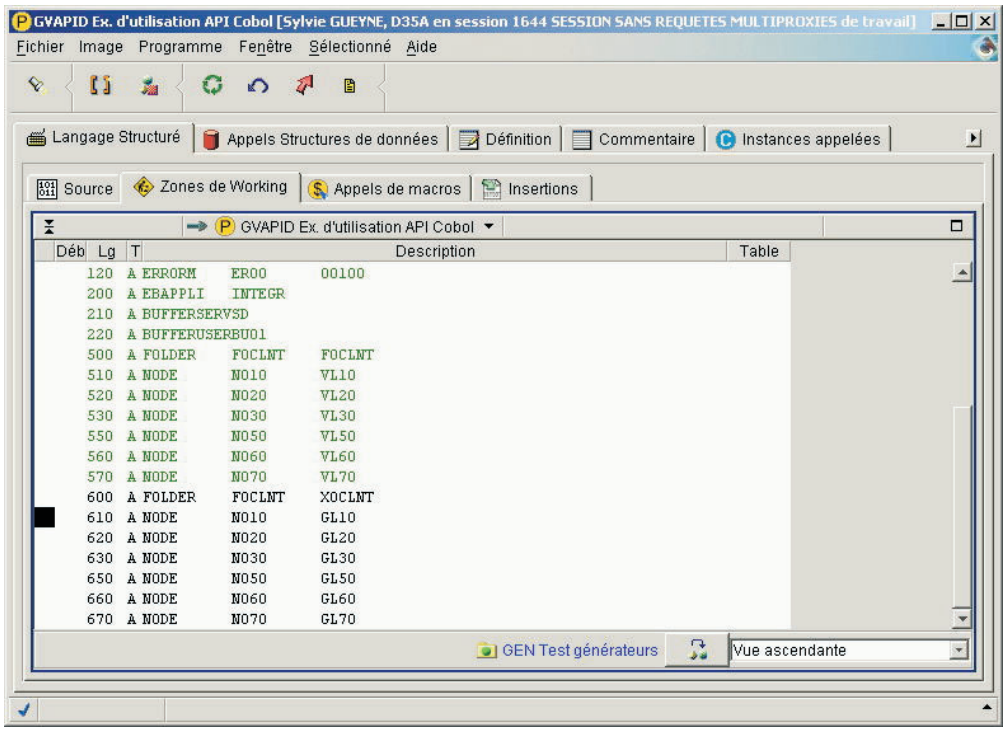
La première étape consiste à déclarer l'Application eBusiness et ses Dossiers dans la working de votre programme applicatif. Vous devez donc :

- déclarer l'opérateur qui va gérer les erreurs retournées par les services,
- appeler l'Application eBusiness voulue,
- déclarer les buffers utilisés par cette Application, si celle-ci en utilise,
- appeler les Dossiers et noeuds de Dossier utilisés.

Vous devez effectuer toutes ces déclarations dans l'onglet 'Langage Structuré' de votre programme applicatif, sous-onglet 'Zones de Working', sur des lignes de type 'A'.

La partie 'Description' de ces lignes est divisée en zones de 10 caractères. Vous devez indiquer dans la première zone le code de l'opérateur API COBOL, puis dans les autres les paramètres nécessaires à cet opérateur (le nombre de paramètres change suivant l'opérateur).

Attention, pour assurer le bon fonctionnement de l'API COBOL, vous devez respecter scrupuleusement le contenu et le cadrage des valeurs que vous saisissez car aucun contrôle n'est effectué dans la version actuelle.



## Gestion des erreurs

Vous devez tout d'abord déclarer l'opérateur qui va gérer les erreurs retournées par les services exécutés :

```
A ERRORM    PARAM1    PARAM2
```

Où

```
 PARAM1=XXXX  code du Segment généré
 PARAM2=nnnnn nombre de postes de cette structure
              (nnnnn=00001)
```

Toutes les structures nécessaires à la gestion des erreurs seront générées dans la working du programme applicatif où vous avez déclaré cet opérateur.

Le code défini par PARAM1 correspond au code Segment programme de la structure identifiant les code et message d'erreur. Il est obligatoire et ne doit pas être identique au code d'une structure qui existe déjà dans le programme.

Dans le COBOL généré, la structure définissant les erreurs retournées par les services est la suivante :

```
01                XXXX.
   05             XXXX-ERRNB     PICTURE 9(5) VALUE ZERO.
   05             XXXX-WERR.
     10           XXXX-ERR          OCCURS nnnnn.
      15          XXXX-ERRKEY    PICTURE X(25).
      15          XXXX-LABEL     PICTURE X(67).
   05             XXXX-ERRCOD.
     10           XXXX-STATUS    PICTURE X(09).
     10           XXXX-CODE      PICTURE X(27).
     10           XXXX-TYPE      PICTURE X(02).
```

ERRNB correspond au nombre d'erreurs rendues par le Gestionnaire de Dossier après l'exécution d'un service de l'Application eBusiness.

ERR contient ces erreurs, identifiants et messages.

ERRCOD contient les erreurs SQL.

CODE contient la valeur 'SQL ERRCODE'.

TYPE et STATUS contiennent les codes erreur.

## Déclaration d'une Application eBusiness

Vous déclarez une Application eBusiness dans votre programme applicatif de la façon suivante :

```
A EBAPPLI   PARAM1
```

Où

```
PARAM1=XXXXXX Code référentiel (code court) de l'Application
              eBusiness.
```

Un chaînage sera créé sur l'Application eBusiness.

Vous ne pouvez appeler qu'une seule Application eBusiness par programme applicatif.

L'ensemble des structures nécessaires à la déclaration de l'Application sera généré en working du programme applicatif où est définie cette instruction.

## Déclaration du buffer utilisateur

Si l'Application eBusiness utilise un buffer utilisateur, vous devez déclarer ce buffer de la façon suivante :

```
A BUFFERCSPARAM1
```

Où

```
 PARAM1 = XXXX  Code Segment généré
```

Ce code Segment est initialisé au code Segment référentiel qui définit le buffer utilisateur de l'Application eBusiness. Il peut être modifié pour des raisons de nommage dans le programme applicatif.

Dans le COBOL généré, la structure complète du buffer utilisateur de l'Application eBusiness se présente sous la forme :

```
01 XXXX.
  10 XXXX-CORUB1.
   15 XXXX-CORUB2 PIC X(nnn).
  10 etc.
```

## Déclaration du buffer serveur

Si l'Application eBusiness utilise un buffer serveur, vous devez déclarer ce buffer de la façon suivante :

```
A BUFFERSERVPARAM1
```

Où

```
PARAM1 = XX Code Structure généré
```

Le code Structure de Données est initialisé au code Structure de Données référentiel qui définit la Structure de données du buffer serveur de l'Application eBusiness. Il peut être modifié pour des raisons de nommage dans le programme.

Cette instruction permet de générer les structures du buffer serveur de l'Application eBusiness sous la forme d'un FILLER, ce buffer n'étant utilisé que dans les Composants Elémentaires.

```
01 XX00.
  05 FILLER  PICTURE X(nnnnn).
```

## Déclaration des Dossiers

La déclaration des Dossiers dans le programme applicatif comprend, pour chaque Dossier, la déclaration du Dossier ainsi que la déclaration des noeuds utilisés.

Vous déclarez un Dossier de la façon suivante :

```
A FOLDER    PARAM1    PARAM2
Où  PARAM1 = XXXXXX  Code référentiel (code court) d'un
                     Dossier de l'Application eBusiness.
    PARAM2 = XXXXXX  Code généré du Dossier.
```

Un chaînage sera créé sur le code référentiel du Dossier.

PARAM2 est initialisé au code référentiel. Il peut être modifié pour des raisons de nommage dans le programme applicatif.

Un même Dossier peut être déclaré plusieurs fois dans le programme à condition de l'utiliser sous des noms différents.

Cette instruction permet de générer toutes les zones de working utilisées dans les différentes demandes de service pour ce Dossier.

## Déclaration des noeuds de Dossier

Après avoir déclaré les Dossiers utilisés, vous devez spécifier quels en sont les noeuds utilisés par votre programme applicatif.

Vous déclarez un noeud de Dossier de la façon suivante :

```
A NODE      PARAM1    PARAM2
```

Où

```
 PARAM1= XXXX Identifiant du noeud dans le Dossier associé
 PARAM2= XXXX Code programme du noeud.
```

Le code programme du noeud est initialisé par l'identifiant du noeud dans le Dossier. Il peut être modifié pour des raisons de nommage dans le programme applicatif.

Vous pouvez faire référence plusieurs fois au même noeud à condition d'avoir des codes programme différents.

Le COBOL généré inclut alors :

- la structure des identifiants de la Vue Logique,
- la structure de la Vue Logique associée au noeud,
- la liste des méthodes d'extraction associées à la Vue Logique,
- la description d'une instance de Vue Logique et du vecteur de présence correspondant,
- la liste des services utilisateur définis pour ce noeud.

STRUCTURE DES IDENTIFIANTS DE LA VUE LOGIQUE

Attention, ces informations sont présentes pour consultation uniquement.

```
01       ID-xxxx-LEN    PIC 9(5) VALUE 00009.
```

Est indiquée ici la longueur des identifiants.

Premier niveau de sauvegarde des identifiants pour les sélections :

```
01        1-ID-xxxx.
     10     ID-xxxx-ident1 PICTURE nnnn
     10    ...
```

Deuxième niveau de sauvegarde des identifiants pour les sélections :

```
01        2-ID-xxxx.
     10     ID-xxxx-corub1 PICTURE nnnn
     10    ...
```

STRUCTURE DE LA VUE LOGIQUE ASSOCIEE AU NOEUD

Cette structure contient les codes action, le vecteur de présence, les identifiants, les paramètres d'extraction et les instances de la Vue Logique. Cette partie correspond à la structure que l'on trouve en entrée du Composant Elémentaire.

```
01             xxxx.
  02        A-xxxx.                              Code action
    05      A-xxxx-CATM.
       10   A-xxxx-CA    PICTURE X.
```

```
      10   A-xxx-CR     PICTURE X OCCURS nnnn.
  02       CH-xxxx.                      Vecteur de présence
   05      CH-xxxx-xxxx
               OCCURS nnnn.
      10   FILLER       PICTURE X(nnnnn).
  02       ID-xxxx.                             Identifiants
      10   1-xxxx-ident1 PICTURE xxxx
      10   ..
  02       EP-xxxx.                  Paramètres d'extraction
      10   2-xxxx-param1 PICTURE nnnn
  02       xxxx-xxxx       OCCURS nnnn.   Instances Vue Log.
      15   FILLER       PICTURE X(nnnnn).
```

En cas de services de mise à jour, l'alimentation de la première occurrence des codes action, du vecteur de présence et des champs de la Vue est à votre charge.

En cas de services de sélection, vous devez alimenter les identifiants et les paramètres d'extraction.

LISTE DES METHODES D'EXTRACTION ASSOCIEES A LA VUE LOGIQUE

```
01         EM-xxxx.
    05     EM-xxxx-EXTNAM   PIC X(10).
    05     EM-xxxx-CODES.
      10   FILLER  PIC X(10) VALUE "xxxxxxxxx ".
      10   ..
    05     EM-xxxx-CODESR REDEFINES EM-xxxx-CODES.
      10   EM-xxxx-VCODES  PIC X(10) OCCURS nnnnn.
```

En cas de sélection avec utilisation de méthodes d'extraction, la zone EXTNAM doit contenir le code de la méthode utilisée.

DESCRIPTION D'UNE INSTANCE DE VUE LOGIQUE ET DU VECTEUR DE PRESENCE CORRESPONDANT

```
01         RE-xxxx-xxxx.
        15  xxxx-corub1  PICTURE xxxxxx
        15  ...
01         RH-xxxx-xxx.
    05      CH-xxxx-NUCLIE PICTURE X.
    05      CH-xxxx-NOMCLI PICTURE X.
```

LISTE DE SERVICES UTILISATEUR

```
01         US-xxxx.
     05    US-xxxx-SRVUSR PIC X(25).
     05    US-xxxx-CODES.
       10  FILLER  PIC X(25) VALUE "PRINT".
       10  ..
     05    US-xxxx-CODESR REDEFINES US-xxxx-CODES.
       10  US-xxxx-VCODES   PIC X(25) OCCURS nnnn.
```

En cas d'utilisation de service utilisateur, la zone SRVUSR doit contenir le code de ce ce service.

# Chapitre 3. Demandes de services sur un Dossier

## Introduction

valeurs que vous saisissez.

L'API COBOL vous permet de spécifier, dans votre programme applicatif, des demandes de service sur le Dossier. Pour cela, vous indiquez des instructions qui vont gérer :

- les mises à jour,
- les lectures ou les collections,
- les services utilisateur,
- les services d'initialisation et de terminaison.

  Vous indiquez une instruction dans l'onglet 'Langage Structuré' de votre programme applicatif, sous-onglet 'Source', avec un opérateur EXS.

  La partie 'Opérande' de la ligne sur laquelle vous indiquez l'opérateur EXS est divisée en zones de 10 caractères. Vous devez indiquer dans la première zone l'opérande, puis dans les suivantes les paramètres nécessaires à cette opérande (le nombre de paramètres change suivant l'opérande).

  Attention, pour assurer le bon fonctionnement de l'API COBOL, vous devez respecter scrupuleusement le contenu et le cadrage des valeurs que vous saisissez, car aucun contrôle n'est effectué dans la version actuelle.

## Service de mise à jour

Vous indiquez un service de mise à jour de la façon suivante :

```
EXS UPDATE    PARAM1    PARAM2    PARAM3    PARAM4
```

Où

```
 PARAM1  = XXXXXX      Code programme du Dossier
 PARAM2  = XXXX        Code programme du noeud
 PARAM3  = 0 ou 1      Option de contrôle
 PARAM4  = 0 ou 1      Option de rafraîchissement
```

Pour un fonctionnement correct, l'Application eBusiness et le Dossier doivent avoir été définis en working.

Vous devez vous-même initialiser les structures nécessaires à la bonne exécution du service, c'est à dire les vecteurs de présence de la structure de la Vue Logique concernée, le code action et éventuellement le buffer utilisateur.

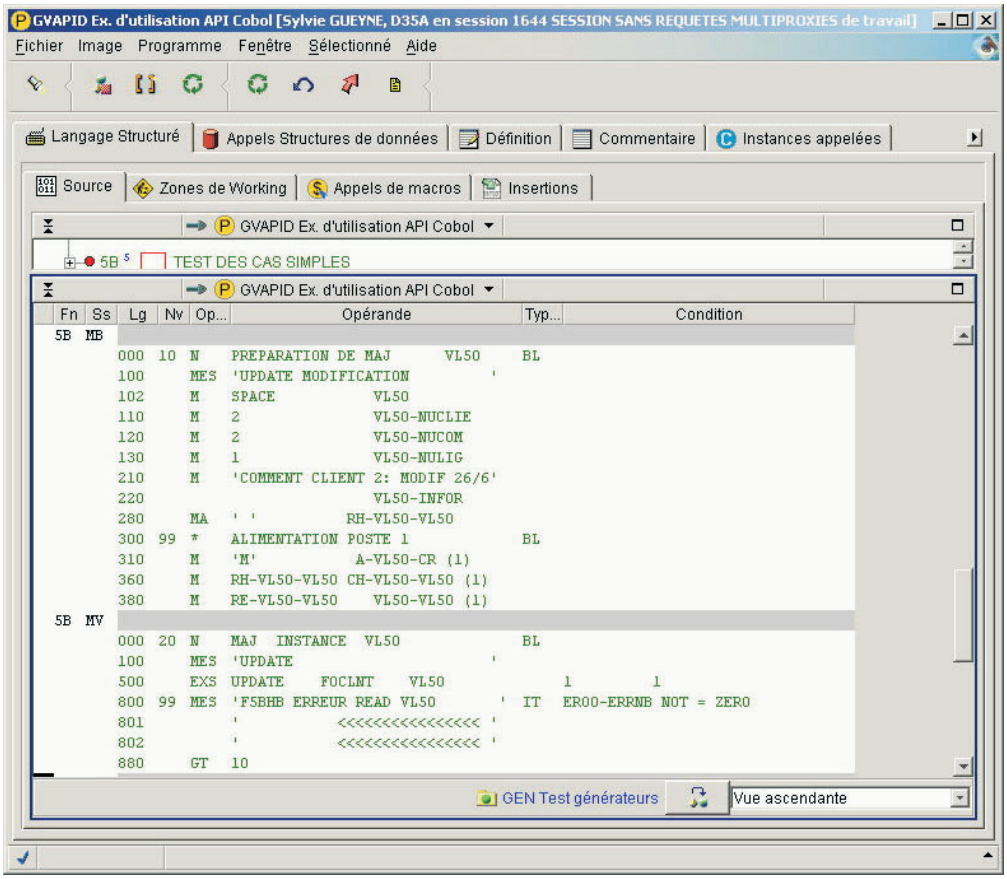
Vous devez tester le bon déroulement du service en vérifiant que le nombre d'erreurs (zone ERRNB dans le généré) n'est pas supérieur à 0 au retour du service.

Le contrôle de cohérence d'un Dossier en cas de mise à jour (contrôle de cardinalité par exemple) est à votre charge, ainsi que les instructions commit et rollback.

Si vous avez indiqué l'option de rafraîchissement, les vecteurs de présence et la structure de la Vue Logique associée au noeud sont initialisés au retour du service.

A chaque exécution d'une instruction de création sur un noeud dépendant, le Gestionnaire de Dossier contrôle que l'instance mère existe. Si elle n'existe pas, une erreur est renvoyée et l'instance proposée n'est pas créée. Vous devez donc respecter la hiérarchie des noeuds dans le Dossier.

En cas de suppression, ce service supprime systématiquement l'instance et toutes celles qui en dépendent si l'option 'Delete Cascade' a été sélectionnée dans le Composant Elémentaire.




VERROUILLAGE/DEVERROUILLAGE D'UN DOSSIER

```
EXS LOCK      PARAM1    PARAM2
EXS UNLOCK    PARAM1    PARAM2
```

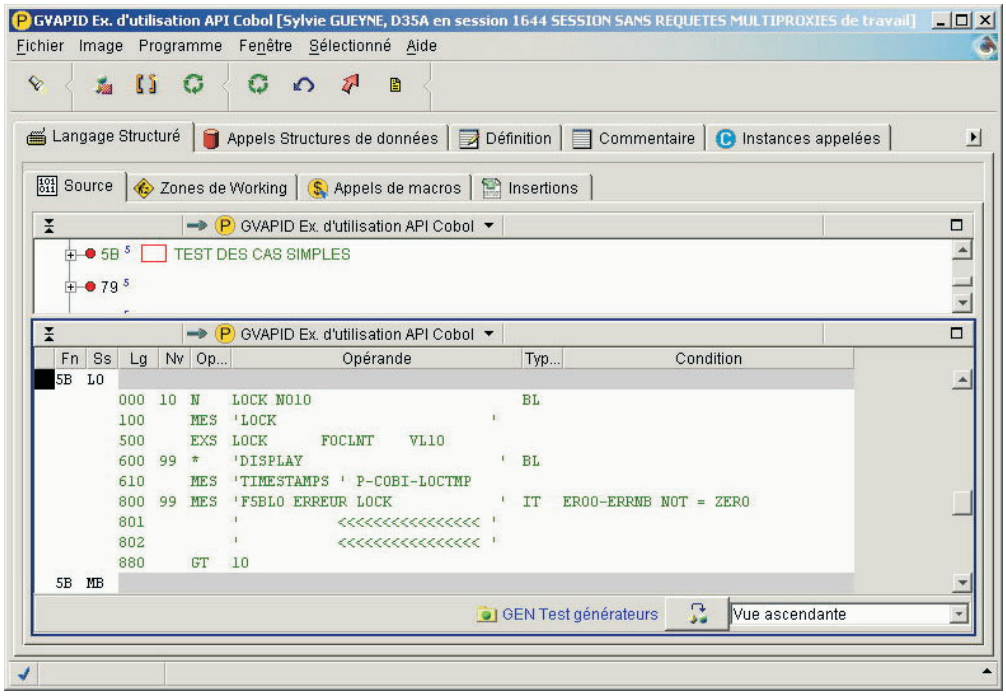
Où

```
 PARAM1  = XXXXXX Code programme du Dossier
 PARAM2  = XXXX   Code programme du noeud hiérarchique
```

Pour un fonctionnement correct, l'Application eBusiness et le Dossier doivent avoir été définis en working.

Vous devez initialiser les structures nécessaires à la bonne exécution du service, c'est-à-dire initialiser la structure des identifiants de la Vue Logique.

Vous devez tester le bon déroulement du service en vérifiant que le nombre d'erreurs (zone ERRNB dans le généré) n'est pas supérieur à 0 au retour du service.




## Service de lecture

Il existe deux types de services de lecture :

- les services dont le résultat correspond à au plus une instance. Un seul opérateur suffit pour spécifier ce type de service.
- les services dont le résultat peut être supérieur à une instance. Ce type de service nécessite deux opérateurs : le premier permet de définir les propriétés de la collection voulue tandis que le second permet de parcourir cette collection de manière séquentielle du début jusqu'à la fin.

Tous les services de lecture peuvent s'exécuter en dehors d'une phase de mise à jour.

### Lecture ramenant au plus une instance

Vous indiquez un service de lecture ne pouvant ramener qu'une instance de la façon suivante :

```
EXS READ      PARAM1    PARAM2
```

Où :

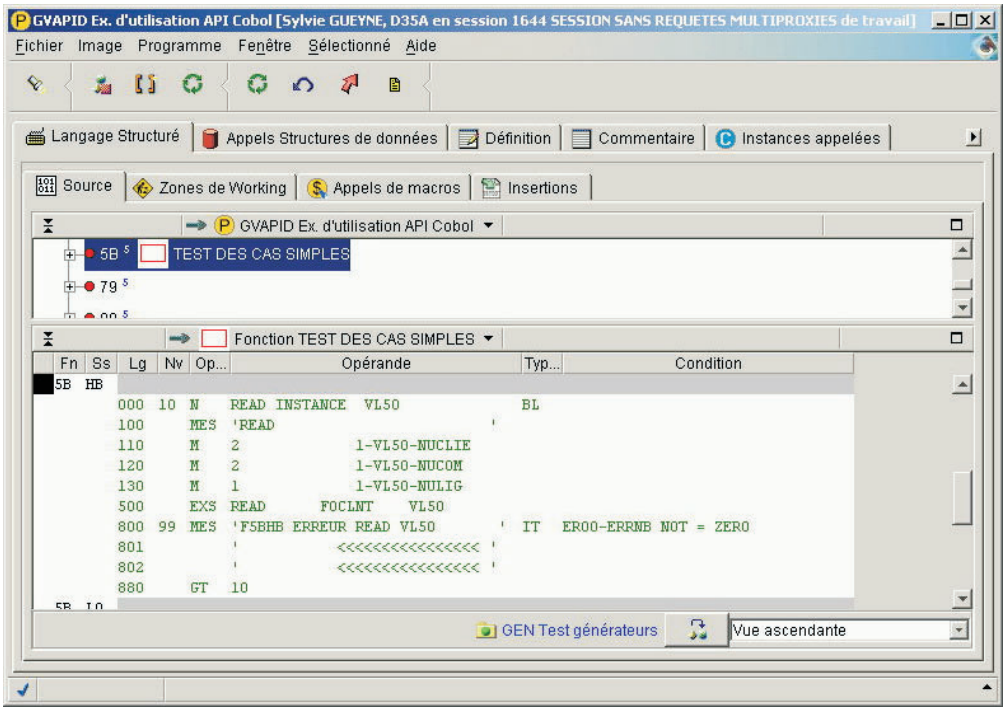
```
 PARAM1 = Code programme du Dossier
 PARAM2 = Code programme du noeud
```

Cet opérateur lit une instance de noeud d'un Dossier à partir de sa clé et la renvoie si elle existe dans la structure de la Vue Logique associée avec ses vecteurs de présence. En cas d'erreur ou si l'instance n'existe pas, un message d'erreur est positionné et la zone ERRNB est incrémentée. Quel que soit l'état de retour, le buffer utilisateur est initialisé.

Pour un fonctionnement correct, l'Application eBusiness et le Dossier doivent avoir été définis en working.

Vous devez initialiser les structures nécessaires à la bonne exécution du service, c'est à dire la structure des identifiants de la Vue Logique et éventuellement celle du buffer utilisateur.

Vous devez tester le bon déroulement du service en vérifiant que le nombre d'erreurs (zone ERRNB dans le généré) n'est pas supérieur à 0 au retour du service.




## Lecture ramenant au moins une instance

Pour ce service, vous devez indiquer deux opérateurs :

- le premier permet de définir la collection
- le second permet de la parcourir à travers une boucle ou une succession de performs.

La déclaration d'une collection doit intégrer les paramètres nécessaires au traitement des lectures d'un ou de plusieurs noeuds.

### Définition de la collection

Vous devez tout d'abord définir la collection de la façon suivante :

```
EXS OPENCOLLECPARAM1    PARAM2    PARAM3    PARAM4    PARAM5
```

Où

```
  PARAM1 = XXXXXX  Code programme de la collection
  PARAM2 = XXXXXX  Code programme du Dossier
  PARAM3 = XXXX    Code programme du noeud
  PARAM4 =         Option de sélection
           S  Single node (valeur par défaut)
           F  First children
           A  All children
  PARAM5 =         Option pour le résultat de la sélection
           H  Hierarchical
           N  By Node (valeur par défaut)
```

PARAM1 doit être unique dans le programme applicatif et sera utilisé dans le nommage des zones privées nécessaires à la gestion de la collection.

Il est possible d'ouvrir plusieurs collections sur le même Dossier et le même noeud.

Ce premier opérateur prépare la lecture d'une collection d'instances sur un noeud d'un Dossier à partir d'une clé de départ ou lit un noeud d'un Dossier et toutes les instances associées à ses noeuds fils de premier niveau ou à tous ses noeuds fils. La clé de départ à fournir correspond toujours à celle définie dans la structure associée au noeud défini par PARAM3. Chaque noeud concerné dans la collection peut prendre en compte une méthode d'extraction définie dans la zone EM-param3-EXTNAM de la structure associée au noeud. Les paramètres d'extraction sont à initialiser dans la structure correspondante.

Vous devez initialiser les structures nécessaire à la bonne exécution du service.

L'option 'hierarchical' permet de récupérer les instances de la collection dans l'ordre hiérarchique. L'option 'byNode' récupère les instances noeud par noeud.

Le service ne tient pas compte du contenu de 'resultOptionList' lorsque l'option 'singleNode' est positionnée.

Pour un fonctionnement correct, l'Application eBusiness et le Dossier doivent avoir été définis en working.

Vous devez tester le bon déroulement du service en vérifiant que le nombre d'erreurs (zone ERRNB dans le généré) n'est pas supérieur à 0 au retour du service.

## Parcours de la collection

Vous devez maintenant indiquer un second opérateur qui vous permettra de parcourir la collection afin de restituer les instances sélectionnées.

```
EXS FETCH     PARAM1
```

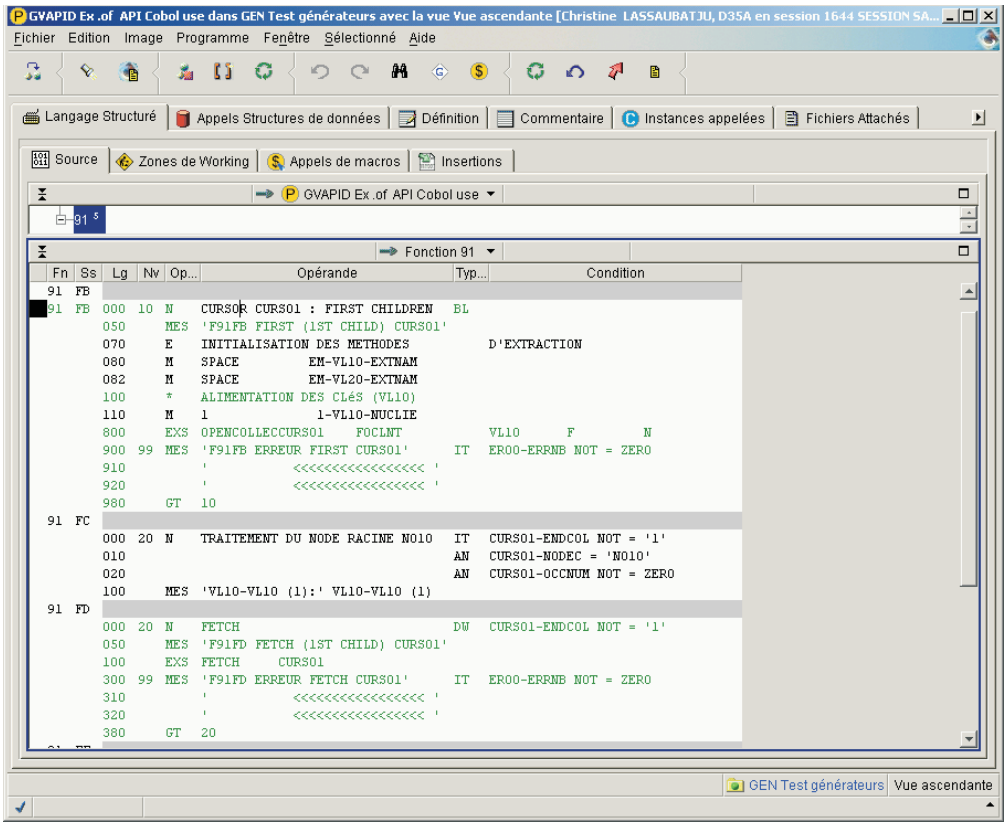
```
Où
  PARAM1  = XXXXXX Code d'utilisation de la collection
```

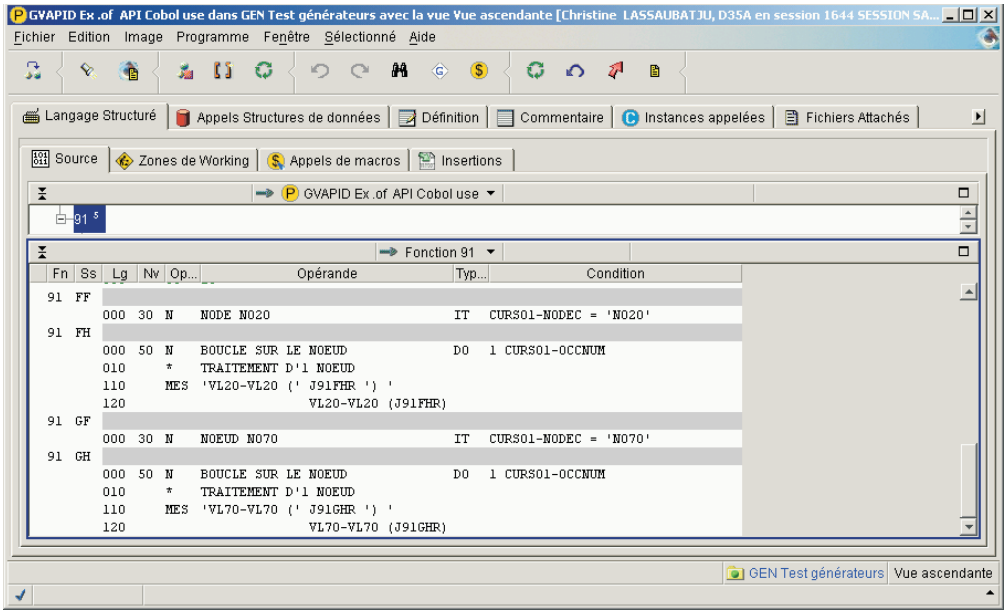
Cet opérateur ramène une instance de noeud de la collection identifiée par PARAM1. Cette instance est transférée dans la structure de la Vue Logique correspondant au noeud en cours de traitement.

A chaque instance rendue, le service initialise également les vecteurs de présence de cette instance.

La fin de liste sur un noeud est renvoyée dans la structure recevant les messages d'erreur en même temps que la dernière instance du noeud.

Vous devez tester le bon déroulement du service en vérifiant que le nombre d'erreurs (zone ERRNB dans le généré) n'est pas supérieur à 0 au retour du service.




## ZONES DE TRAVAIL GENEREES

Les zones de travail indiquées ci-dessous sont générées pour le parcours d'une collection. Elles sont à votre disposition en consultation uniquement ; vous ne devez en aucun cas les modifier. Ce sont des données techniques utilisées par le Dossier et ses Composants Elémentaires.

ENDCOL est l'indicateur de fin de collection. Il prend la valeur '1' quand la fin de la collection est atteinte, ou une autre valeur sinon.

ENDNOD est l'indicateur de fin de noeud. Il prend la valeur '1' à la fin du noeud ou une autre valeur sinon.

VIEWI représente la Vue Logique initiale.

VIEWC représente la Vue Logique courante.

NODEI représente le noeud initial.

NODEC représente le noeud courant.

OCCNUM représente le nombre d'instances retournées dans la table des Vues Logiques.

NBREC représente le nombre total d'enregistrements lus pour une collection.

## Service utilisateur

Vous indiquez une demande de service utilisateur de la façon suivante :

```
EXS EXUS      PARAM1    PARAM2
Où
 PARAM1 = Code programme du Dossier
 PARAM2 = Code programme du noeud
```

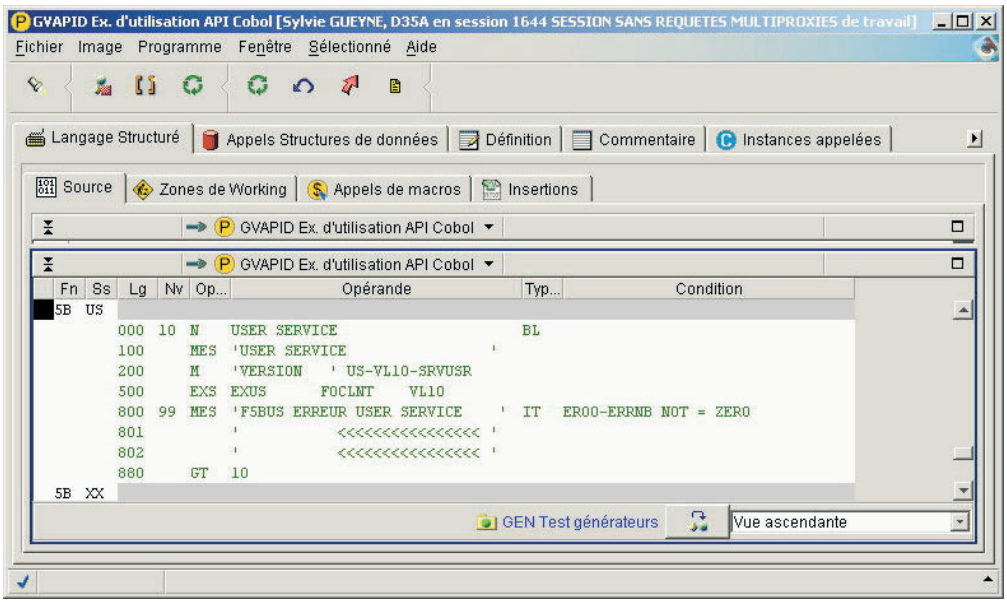
Pour un fonctionnement correct, l'Application eBusiness et le Dossier doivent avoir été définis en working.

Vous devez initialiser les structures nécessaires à la bonne exécution du service : les vecteurs de présence de la structure de la Vue Logique concernée, éventuellement le buffer utilisateur, et la zone US-xxxx-SRVUSR (xxxx étant le code du noeud correspondant au service).

Le contenu de la zone US-xxxx-SRVUSR doit correspondre à l'un des services disponibles sur le noeud concerné.

Le retour du service initialise les vecteurs de présence et la structure de la Vue Logique ou la structure des messages d'erreur.

Vous devez tester le bon déroulement du service en vérifiant que le nombre d'erreurs (zone ERRNB dans le généré) n'est pas supérieur à 0 au retour du service.




## Services d'initialisation et de terminaison

Grâce à ces services, vous définissez une unité de travail cohérente. Pour cela, vous devez appeler un Serveur d'Initialisation, qui ouvre l'unité de travail, et un Serveur de Terminaison qui ferme cette unité.

Vous indiquez les services d'initialisation et de terminaison de la façon suivante :

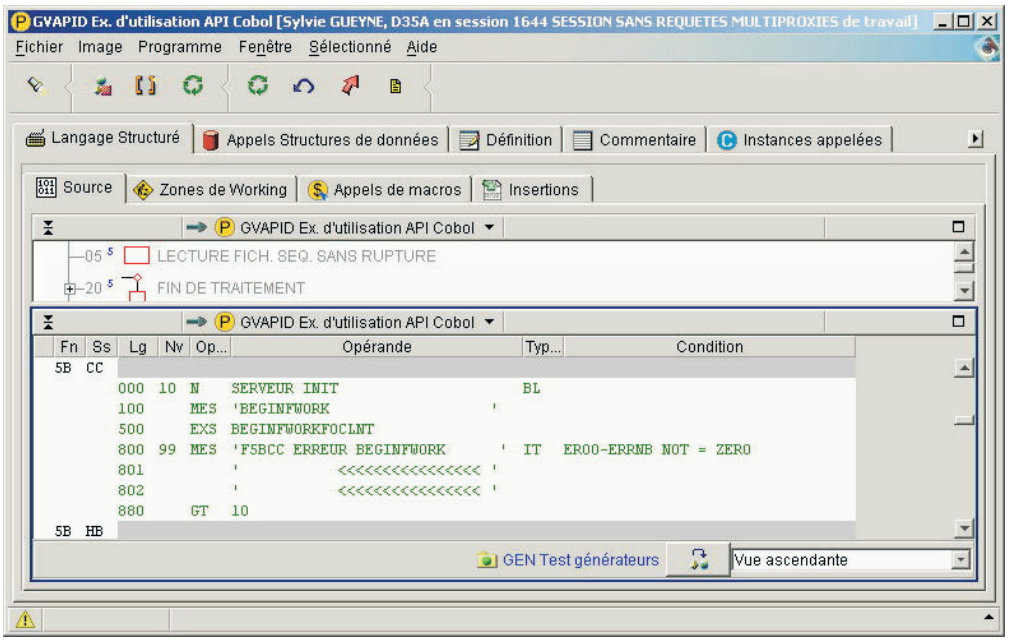
```
EXS BEGINFWORKPARAM1
EXS ENDFWORK  PARAM1
```

Où

```
   PARAM1  = XXXXXX  Code programme du Dossier
```

Le premier service assure l'appel du Serveur d'Initialisation associé à ce Dossier et le deuxième assure l'appel du Serveur de Terminaison.

Vous devez tester le bon déroulement du service en vérifiant que le nombre d'erreurs (zone ERRNB dans le généré) n'est pas supérieur à 0 au retour du service.


GVAPID Ex. d'utilisation API Cobol [Sylvie GUEYNE, D35A en session 1644 SESSION SANS REQUETES MULTIPROXIES de travail]
Fichier Image Programme Fenêtre Sélectionné Aide
Langage Structuré
Appels Structures de données
Définition
Commentaire
Instances appelées
Source
Zones de Working
Appels de macros
Insertions
GVAPID Ex. d'utilisation API Cobol
05 5 LECTURE FICH. SEQ. SANS RUPTURE
20 5 FIN DE TRAITEMENT
GVAPID Ex. d'utilisation API Cobol
Fn Ss Lg Nv Op... Opérande Typ... Condition
5B CC
000 10 N SERVEUR INIT BL
100 MES 'BEGINFWORK '
500 EXS BEGINFWORKFOCLNT
800 99 MES 'F5BCC ERREUR BEGINFWORK ' IT ERROO-ERRNB NOT = ZERO
801 ' <<<<<<<<<<<<<<<< '
802 ' <<<<<<<<<<<<<<<< '
880 GT 10
5B HB
GEN Test générateurs
Vue ascendante

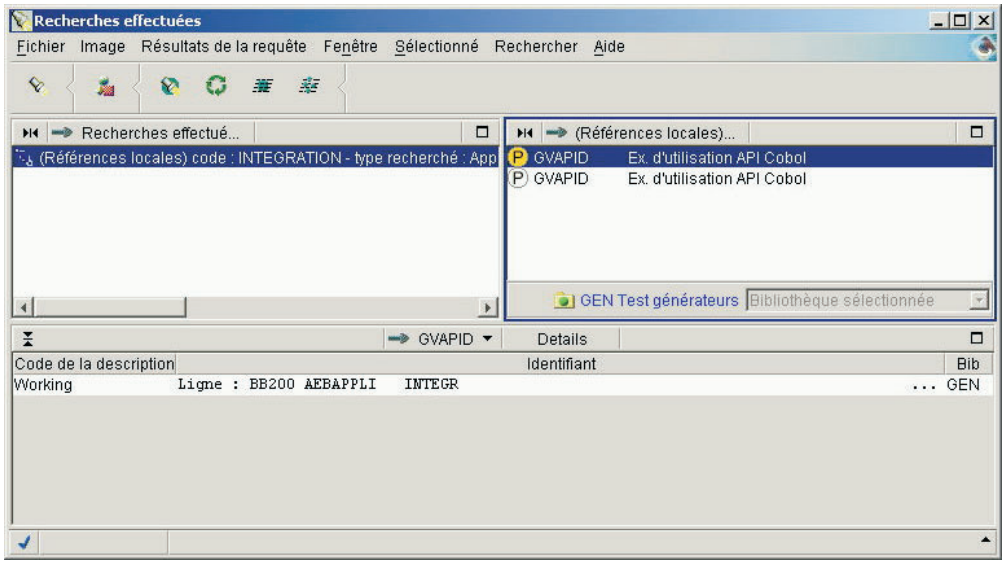
# Chapitre 4. Chaînages

Pour faciliter l'utilisation de l'API COBOL, des chaînages associent les entités eBusiness (Application eBusiness et Dossier) et le programme applicatif dans lequel elles sont appelées, sur des lignes de Working de type 'A'.

Des chaînages sont donc disponibles entre :

- Application eBusiness et Programme, Composant Elémentaire, Ecran.
- Dossier et Programme, Composant Elémentaire, Ecran.

La recherche des chaînages est disponible aussi bien pour la recherche locale que serveur.



Référence : DDAPI000352F - 6556

Imprimé en France