

Université IBM i 2018

16 et 17 mai

IBM Client Center Paris



S42 – Nouveautés RPG V7 et RDi 9.6

Philippe Bourgeois

IBM France

pbourgeois@fr.ibm.com



Plan de la présentation

- Nouveautés RPG récentes en 7.2 et 7.3
 - Nouvelles fonctions intégrées
 - Code-opération ON-EXIT
 - Code-opération DATA-INTO
 - Autres nouveautés

- Nouveautés RDi 9.5 et 9.6
 - Nouveautés éditeur LPEX
 - Nouveautés couverture de code
 - Intégration avec ACS

A background network diagram consisting of light blue lines connecting various circular nodes. The nodes vary in size and color, including shades of light blue, purple, and pink. The overall structure is a complex web of connections.

Nouveautés RPG récentes en 7.2 et 7.3

Rappel : RPG Full Free

```
1...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+....0....+..
**free
  ctl-opt dftactgrp(*no) option(*srcstmt:*nodebugio);

  dcl-f dcl_sin_d1 workstn sfile(chx_ct_sfl:rrn) sfile(chx_st_sfl:rrn);
  dcl-f ct_det_ctn keyed;
  dcl-f sinistres1 usage(*update:*output);

  dcl-s rrn packed(4:0);
  dcl-s sin_declare char(1) inz('N');
  dcl-s enquete_sin char(1);

  dcl-pr enq_h_sin1 extpgm('ENQ_H_SIN1');
    ncli like(cli_num);
    enq like(enquete_sin);
  end-pr;

  exfmt menu; // Affichage du menu principal
  dow not *in03 and not *in12;
    select;
    when (option = 1); // Déclaration d'un sinistre
      exfmt chx_crit; // Choix du critère de recherche
      dow not *in12;
        select;
        when (cont_num<>*zeros and cli_num=*zeros and deb_nom=*blanks); // Recherche par numéro de CONTRAT
          exsr rec_num_cont;
        when (cont_num=*zeros and cli_num<>*zeros and deb_nom=*blanks); // Recherche par numéro de CLIENT
          exsr rec_num_cli;
```

Outils de conversion automatique et de re-engineering :

- RPG Toolbox (ITHEIS)
- Transformer-RPG (ARCAD Software) (au catalogue IBM)

Fonctions %SCAN et %SCANR

- Nouveau paramètre longueur :

```
%scan('valeur' : zone : [debut [: longueur]]);
```

- Nouvelle fonction %SCANR

```
**free
dcl-s z1 char(50) inz('Le titre est *** Titre1 ***')
dcl-s p1 zoned(3:0);

p1 = %scan('***' : z1);
dsply p1;
// p1 = 14

p1 = %scanr('***' : z1);
dsply p1;
// p1 = 25

p1 = %scan('est' : z1 : 1 : 10);
dsply p1;
// p1 = 0

*inlr = *on;
```

```
DSPLY 14
DSPLY 25
DSPLY 0
```

DS imbriquées

- Il est désormais possible d'imbriquer **dcl-ds** dans **dcl-ds**

```
dcl-ds universite qualified;  
  nb_sessions int(5);  
  debut       date;  
  fin         date;  
  dcl-ds session dim(50);  
    id         char(3);  
    titre      varchar(100);  
    description varchar(1500);  
    speaker    varchar(50);  
  end-ds;  
end-ds;
```

DS imbriquées

- Préalablement il fallait faire

```
dcl-ds session_t template qualified;
```

```
  id          char(3);
```

```
  titre       varchar(100);
```

```
  description varchar(1500);
```

```
  speaker     varchar(50);
```

```
end-ds;
```

```
dcl-ds universite qualified;
```

```
  nb_sessions int(5);
```

```
  debut       date;
```

```
  fin         date;
```

```
  session     likeds(session_t) dim(50);
```

```
end-ds;
```

Fonctions %ELEM et %SIZE

- Acceptent des DS complexes, avec ou sans indice

```

**free
dcl-s i int(3);
dcl-s j int(3);
dcl-s total zoned(9:2);

dcl-ds transactions dim(30) qualified;
  numero zoned(5);
  dcl-ds acheteur;
    nom char(20);
    adresse char(50);
  end-ds;
  dcl-ds vendeur;
    nom char(20);
    adresse char(50);
  end-ds;
  dcl-ds produits dim(20);
    num zoned(7);
    nom char(50);
    prix zoned(7:2);
  end-ds;
end-ds;

for i = 1 to %elem(transactions);
  for j = 1 to %elem(transactions.produits);
    total = total + transactions(i).produits(j).prix;
    // ...
  endfor;
endfor;
// ...

```


Fonctions %MIN et %MAX

- %MIN : retourne la plus *petite* valeur d'une liste de valeurs
- %MAX : retourne la plus *grande* valeur d'une liste de valeurs

```

**free
dcl-s z1 zoned(5:2) inz(123,45);
dcl-s z2 packed(7:2) inz(100,68);
dcl-s z3 int(5) inz(300);

dsply %min(z1:z2:z3); // 100,68
dsply %max(z1:z2:z3); // 300

*inlr = *on;

```

- Exemple :

```

date_echeance = %min(date_livraison : date_facturation) + %days(60);

```

Fonction %PROC

- Retourne le nom de la procédure en cours d'exécution

```
BIFPROC2.RPGLE ☒  
Ligne 18      Colonne 2      Ret  
  . . . + . . . . 1 . . . + . . . . 2 . . . +  
000001  **free  
000002  ctl-opt dftactgrp(*no);  
000003  
000004  dcl-s msg char(50);  
000300  
001700  msg = %proc();  
001701  dsply msg;  
001702  
001703  proc1();  
001704  
003800  *inlr = *on;  
003801  
003802  dcl-proc proc1;  
003803    msg = %proc();  
003804    dsply msg;  
003806  end-proc;
```

```
DSPLY  BIFPROC2  
DSPLY  PROC1
```

Valeurs nulles – Mot-clé NULLIND

- Permet d'affecter la valeur nulle à une variable. Deux utilisations :
 - Sans paramètre : on utilise alors la fonction %NULLIND
 - Avec paramètre : on utilise alors ce paramètre (variable de type indicateur)

```

**free
ctl-opt alwnull(*usrctl);

dcl-s date1 date nullind;

dcl-s date2 date nullind(date2i);
dcl-s date2i ind;

dcl-s dates date dim(5) nullind(datesi);
dcl-s datesi ind dim(5);

```

```

%nullind(date1) = *on;
if %nullind(date1);
  dsply 'date1 contient *NULL';
endif;

date2i = *on;
if date2i;
  dsply 'date2 contient *NULL';
endif;

datesi(3) = *on;
if datesi(3);
  dsply 'dates(3) contient *NULL';
endif;

*inlr = *on;

```

```

DSPLY date1 contient *NULL
DSPLY date2 contient *NULL
DSPLY dates(3) contient *NULL

```

- Dans le cas d'un tableau ou d'une DS externe, le paramètre est un tableau ou une DS externe
- NULLIND peut-être indiqué sur les sous-zones d'une DS, sur un paramètre, et dans les prototypes et interfaces d'une procédure
- Les mots-clés EXTNAME et LIKEREK possèdent une nouvelle option *NULL

Nouveau code opération ON-EXIT

- Permet de forcer l'exécution d'une section de code, quelque soit la façon dont une procédure / un programme se termine

- Fonctionnement
 - Le code-opération ON-EXIT doit être placé en fin de procédure
 - Les instructions placées après ON-EXIT sont exécutées à chaque fois que la procédure se termine :
 - En cas de fin normale
 - En cas de fin anormale :
 - Erreur provoquant la sortie de la procédure
 - Arrêt du travail ou du sous-système

- Syntaxe : **ON-EXIT <indicateur>**
 - L'indicateur est optionnel
 - Il vaudra *OFF en cas de fin normale ou *ON en cas de fin anormale

Nouveau code opération ON-EXIT



■ Exemple

```
**free
ctl-opt main(pgm1) dftactgrp(*no);

dcl-proc pgm1;
  dcl-s ind1 ind;
  dcl-s z1 int(3) inz(100);
  dcl-s z2 int(3);

  exec sql set option commit = *none;
  // ...
  exec sql declare global temporary table fic1 (z1 char(10));
  // ...
  z1 = z1/z2;
  // ...

on-exit ind1;

exec sql drop table fic1;

if not ind1;
  dsply 'Fin normale du programme PGM1';
else;
  dsply 'Fin anormale du programme PGM1';
endif;

end-proc;
```

```
Tentative de division par zéro pour opération en virgule fixe.
Application error. MCH1211 unmonitored by ONEXIT at statement 0000000082,
instruction X'0000'.
DSPLY Fin anormale du programme PGM1
Application error. MCH1211 unmonitored by ONEXIT at statement 0000000082,
instruction X'0000'.
```

Nouveau code-opération DATA-INTO



- Permet de transférer le contenu d'un flux structuré de n'importe quel type (JSON, CSV, properties...) dans une DS
 - On avait déjà XML-INTO pour transférer un flux XML dans une DS
- Prérequis :
 - 7.2 ou 7.3 + PTFs : <http://ibm.biz/data-into-rpg-opcode-ptfs>

■ Syntaxes :

- `data-into` maDS
 - `%data('flux_structuré' : '<options>')`
 - `%parser('nom_du_parser' : '<options>');`

} Lecture de l'intégralité du flux
- `data-into` maDS
 - `%handler('nom_du_handler')`
 - `%data('flux_structuré' : '<options>')`
 - `%parser('nom_du_parser' : '<options>');`

} Lecture du flux par blocs

Nouveau code-opération DATA-INTO



- Vous devez écrire le programme **parser**
 - Même principe que les programmes handler de RPG Open Access

- Des exemples sont fournis dans le fichier source **SAMPLE** de la bibliothèque **QOAR** (copiez ces sources dans votre fichiers source **QRPGLESRC** de votre bibliothèque) :
 - Exemples pour des fichiers de type **JSON** :
 - Source du parser : **JSONPARSE**
 - Source d'un programme de test : **JSONTEST**
 - Exemples pour des fichiers de type **PROPERTIES** :
 - Sources des parsers : **PROPPARSE***
 - Sources des programmes de test : **PROPTTEST***

Nouveau code-opération DATA-INTO

- Les options disponibles dans `%data('flux_structuré' : '<options>')`

Option	Utilisation
<code>doc = <u>string</u>/file</code>	string : le flux structuré est une variable du programme file : le flux structuré est un fichier IFS
<code>case = <u>lower</u>/upper/any/convert</code>	lower : les données du flux sont en minuscules upper : les données du flux sont en majuscules any : le contenu du flux est mixte et sera converti en majuscules convert : le flux sera converti en majuscules et les caractères invalides en RPG seront transformés en _
<code>trim=<u>all</u>/none</code>	Enlève ou non les blancs des données du flux avant leur affectation auprès des variables
<code>allowmissing=<u>no</u>/yes</code>	Peut-il y avoir des sous-zones de DS n'ayant pas d'équivalent en tant que données du flux ?
<code>allowextra=<u>no</u>/yes</code>	Peut-il y avoir des données du flux n'ayant pas d'équivalent dans la DS ?

Nouveau code-opération DATA-INTO



- Les options disponibles dans `%data('flux_structuré' : '<options>')`

Option	Utilisation
path=donnée1/donnée2...	Indique le chemin de la donnée à extraire
ccsid= <u>best</u> /job/ucs2	CCSID utilisé pour parser le flux
datasubf=nom	Sous-zone de la DS pour placer le texte d'un élément
countprefix=préfixe	Sous-zone indiquant le nombre d'occurrences alimentées

Nouveau code-opération DATA-INTO



■ Exemple de parsing d'un fichier JSON (exemple IBM)

```
000001 **free
000002 dcl-ds petInfo qualified inz;
000003     numPets int(10);
000004     dcl-ds pets dim(3);
000005         name varchar(15);
000006         type varchar(10);
000007         age int(3);
000008     end-ds;
000009     veterinarian varchar(15);
000010 end-ds;
000011 dcl-s msg char(52);
000012 dcl-s i int(10);
000013 dcl-s jsonString varchar(1000);
000014
000015 jsonString =
000016     '{'
000017 +     '"petInfo":'
000018 +     '{'
000019 +         '"a?b" : "a dash b",'
000020 +         '"pets":'
000021 +         '['
000022 +             '{"name":"Spot", "type":"dog", "age":3, "faveToy":"ball"},'
000023 +             '{"name":"Puff", "type":"cat", "age":7, "faveToy":"string"}'
000024 +         '],'
000025 +         '"veterinarian":"Dr Smith"'
000026 +     '}'
000027 + '}';
```

```
000028 data-into petInfo
000029     %data(jsonString
000030         : 'case=any countprefix=num allowextra=yes')
000031     %parser('JSONPARSE');
000032 for i = 1 to petInfo.numPets;
000033     msg = petInfo.pets(i).name + ' ' + petInfo.pets(i).type + ' '
000034         + 'age=' + %char(petInfo.pets(i).age);
000035     dsply msg;
000036 endfor;
000037 msg = 'Vet is ' + petInfo.veterinarian;
000038 dsply msg;
000039 return;
```

```
DSPLY  Spot dog age=3
DSPLY  Puff cat age=7
DSPLY  Vet is Dr Smith
```

Nouveau code-opération DATA-INTO



- Pour en savoir plus : https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_73/rzasm/roaDataInto.htm



>|

The DATA-INTO operation imports the data from a structured document into an RPG data structure. The operation requires a parser that parses the data in the document, and uses [callback functions](#) to pass the information about named values in the document to the RPG runtime, which places the data into an RPG data structure.

- [Parameter passed to a DATA-INTO parser](#)
- [Callback procedures for DATA-INTO parsers](#)
- [Tracing a DATA-INTO parser](#)
- [Example of a DATA-INTO parser](#)

Nouveau code-opération DATA-INTO

- Pour en savoir plus :



IBM i > DEVELOPER > RPG

Web Exclusive

XML-INTO meets Open Access With RPG's New DATA-INTO

February 2018 | by Jon and Susan



This article is to introduce you to the latest addition to the RPG  2



IBM i > DEVELOPER > RPG

Web Exclusive

A Closer Look at RPG's DATA-INTO

March 2018 | by Jon Paris, Susan Gantner



In last month's article "XML_INTO Meets Open Access With  4

Date et heure – Amélioration des horodates



- Horodates : zones de type **timestamp** (type **Z** en spécification D)
 - Un seul format : yyyy-mm-dd-hh.mn.ss.uuuuuu
 - Longueur : 26
 - *LOVAL : 0001-01-01-00.00.00.000000
 - *HIVAL : 9999-12-31-24.00.00.000000
 - Les microsecondes sont optionnelles
- Depuis la 7.2 il est possible d'indiquer la précision des horodates, c'est-à-dire le nombre de fractions de secondes (de 0 à 12)
 - `dc1-s ts0 timestamp(0)` // 2018-05-17-15.17.03
 - `dc1-s ts3 timestamp(3)` // 2018-05-17.15.17.03.789
 - `dc1-s ts12 timestamp(12)` // 2018-05-17.15.17.03.789543218573

Date et heure – Amélioration de %SECONDS et %DIFF



- Support de valeurs avec des décimales lors de l'ajout de secondes avec la fonction %SECONDS
 - `t1 = t1 + %seconds(1,5);`
- Possibilité de récupérer la différence entre 2 horodates en secondes avec positions décimales
 - `diff1 = %diff(z_fin:z_debut:*SECONDS:6);`
`// diff1 = 1922,483019`

Autres nouveautés RPG



- Les sources en langage de liage (BND) peuvent être stockés dans l'IFS
- Plus besoin de spécifier **DFACTGRP (*NO)** si l'on spécifie **ACTGRP** ou **BNDDIR**
- Améliorations **PCML**
 - 1. Par CTL-OPT ou en spécification H : mot-clé **PGMINFO**
PGMINFO(*NO | *PCML | *DCLCASE < : *MODULE < : *V6 | *V7 >>)
 - *DCLCASE : les noms dans le PCML respectent la casse des noms RPG
 - *V7 : PCML V7 (support des zones de type VARCHAR)
 - 2. Sur la déclaration de procédure (DCL-PROC ou spécification P) : mot-clé **PGMINFO(*YES | *NO)**
 - Le PCML sera ou non généré pour cette procédure

A background network diagram consisting of light blue lines connecting various circular nodes. The nodes vary in size and color, including shades of purple, blue, and grey. The largest node is a purple circle on the left side. The text 'Nouveautés RDi' is centered in a bold, dark blue font.

Nouveautés RDi

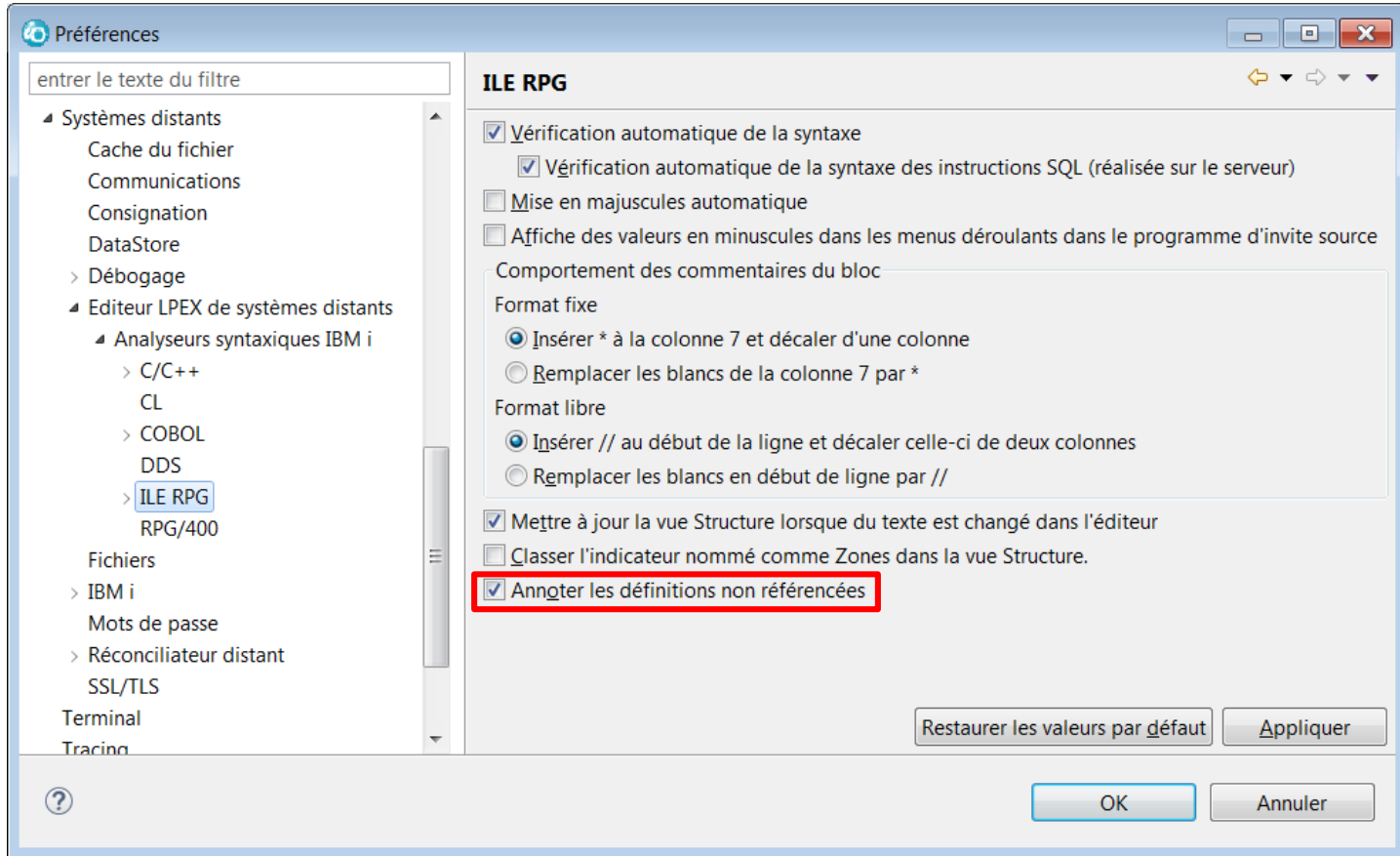
Rappel : RDi (Rational Developer for i)

- Environnement de développement de code RPG/COBOL/CL/DDS
- Successeur des outils traditionnels (SEU/PDM/SDA/RLU/ISDB)
- Le seul environnement de développement qui évolue
- Basé sur Eclipse – S'intègre avec d'autres outils IBM ou tiers
- S'installe sur un poste en Windows, Linux ou Mac OS X
- Améliore la productivité des développeurs
- Vous pouvez le tester gratuitement pendant 60 jours
- 2 jours de formation permettent de faire le tour du produit
- La version actuelle est la 9.6 (novembre 2017)

LPEX – Mise en évidence des zones non utilisées

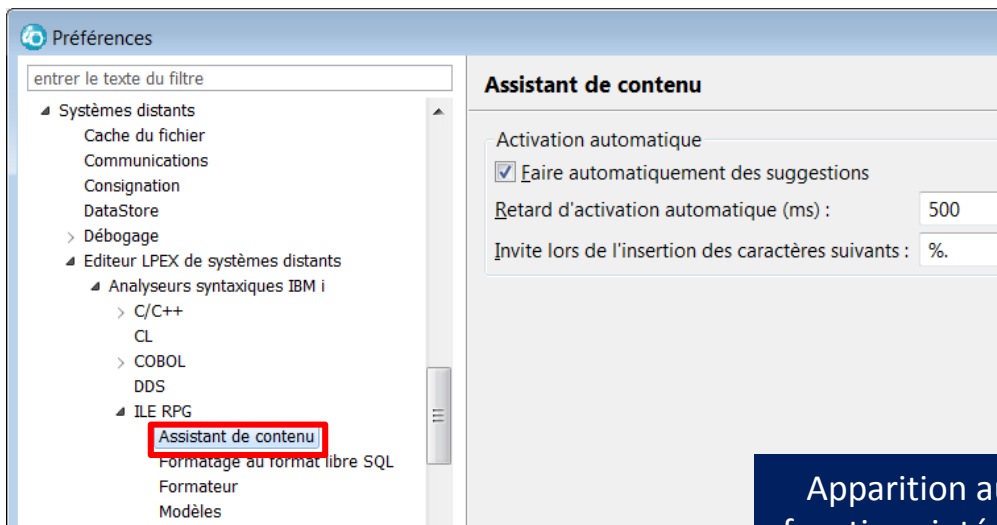
```
VIDEO.RPGLE ✕
Ligne 30      Colonne 3      Remplacement
...1...+...2...+...3...+...4...+...5...+...6...+...7
001300      D rrnfilm      s              4 0
001400      D rrnactf      s              like(rrnfilm)
001500      D rrnact       s              like(rrnfilm)
i 001600      D inconnu      s              1n
i 001700      D arg_rec      s              40  varying
001800      D nommax       s              20
i 001900      D nommi        s              like(nommax)
002000      D max          s              like(nommax) inz(*hival)
002100
002200      // Date du jour (date du travail)
002300      D              ds
002400      D date_job    d      inz(*job) datfmt(*iso)
i 002500      D an_job      4 0  overlay(date_job)
```

LPEX – Mise en évidence des zones non utilisées



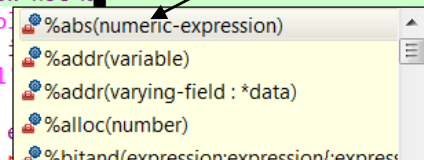
LPEX – Complétion de code automatique

- Complétion de code automatique (sans CTRL Espace)



Apparition automatique des fonctions intégrées sans avoir à faire CTRL+Espace

```
007800 // --> Si option "5=Afficher les
007900 readc films_sfl;
008000 dow not %
RNF604E Le symbole
008100 RNF5377E Fin de l
008200
008300
008400
```



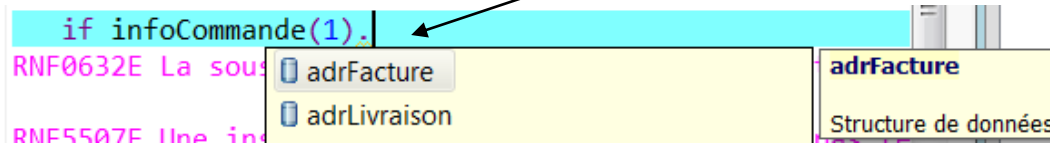
%ABS(numeric-expression)
%ABS (Valeur absolue d'une expression) renvoie la valeur absolue de l'expression numérique indiquée comme paramètre. Si cette valeur n'est pas négative, elle est renvoyée telle quelle ; si elle est négative, elle est renvoyée sans le signe moins (-).

LPEX – Complétion de code automatique

- Complétion de code automatique (sans CTRL Espace)

```
**free  
  dcl-ds adresse;  
    rue      char(40);  
    codePostal zoned(5);  
    ville    char(40);  
    pays     char(40);  
  end-ds;  
  
  dcl-ds infoCommande qualified dim(50);  
    adrFacture  likeds(adresse);  
    adrLivraison likeds(adresse);  
  end-ds;  
  
  // ...
```

Apparition automatique
des sous-zones sans avoir à
faire CTRL+Espace

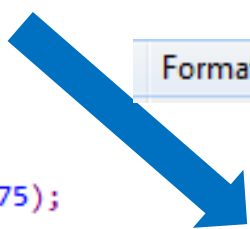


LPEX – Formatage de code

```
001110 dcl-s taux packed(4:2) inz(18,75);
001111     dcl-s pay like(taux:+2);
001115
001300 read payemp;
001400     dow not %eof;
001500     if nbrhrs <= 35;
001600         pay = nbrhrs * taux;
001700     else;
001800         pay = (35*taux) + (nbrhrs-35) * (taux*1,75);
001900     endif;
002000     exfmt fmt1;
002100     read payemp;
002200     enddo;
```

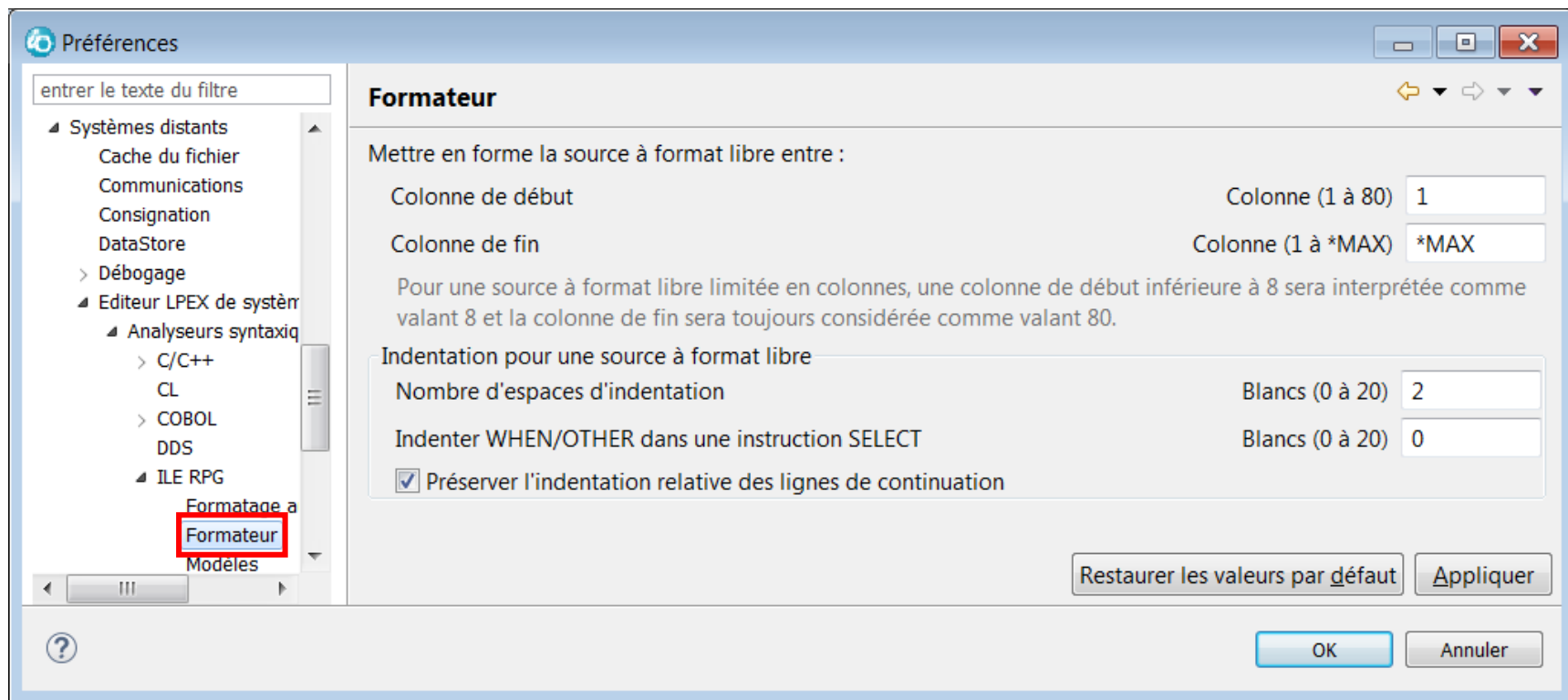
Formater

Ctrl+Maj+F

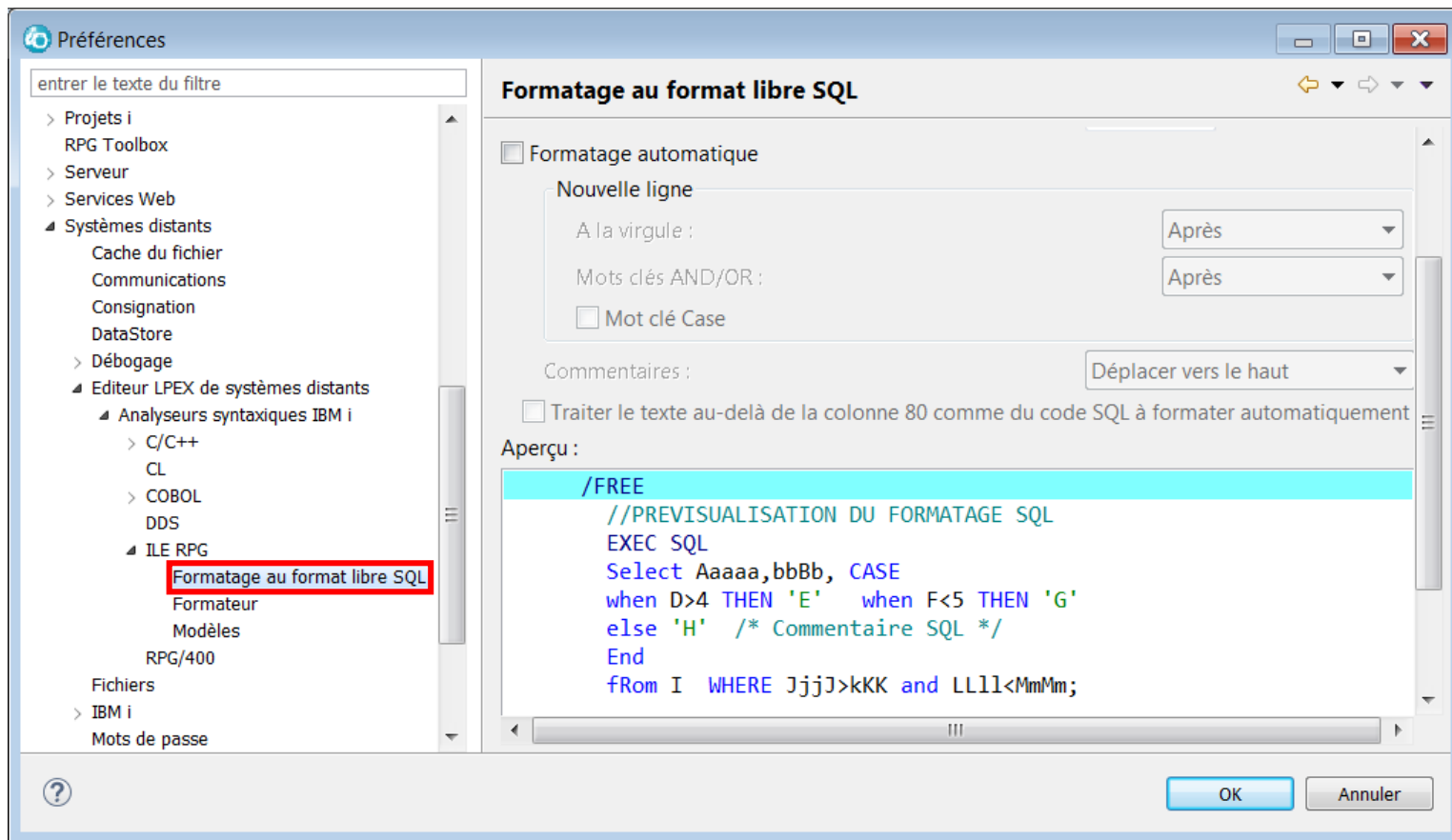


```
001110 dcl-s taux packed(4:2) inz(18,75);
001111 dcl-s pay like(taux:+2);
001115
001300 read payemp;
001400 dow not %eof;
001500     if nbrhrs <= 35;
001600         pay = nbrhrs * taux;
001700     else;
001800         pay = (35*taux) + (nbrhrs-35) * (taux*1,75);
001900     endif;
002000     exfmt fmt1;
002100     read payemp;
002200     enddo;
```

LPEX – Formatage de code



LPEX – Formatage de code



The screenshot shows the 'Préférences' (Preferences) dialog box in the LPEX editor. The left sidebar contains a tree view of settings categories, with 'Formatage au format libre SQL' highlighted in red. The main area is titled 'Formatage au format libre SQL' and contains the following options:

- Formatage automatique
- Nouvelle ligne**
 - A la virgule :
 - Mots clés AND/OR :
 - Mot clé Case
- Commentaires :
- Traiter le texte au-delà de la colonne 80 comme du code SQL à formater automatiquement

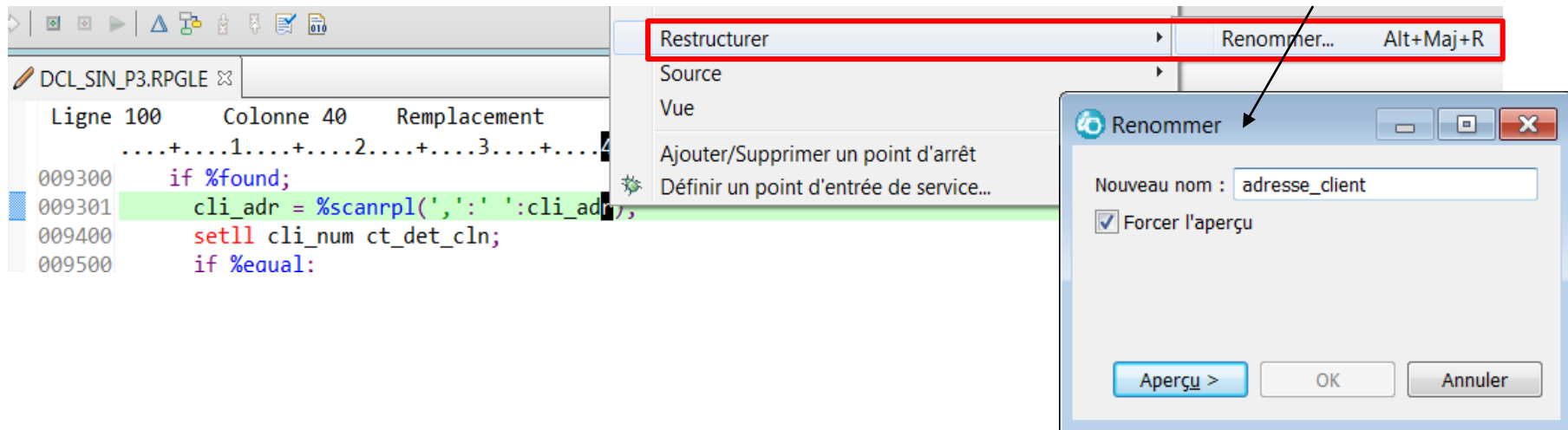
Aperçu :

```
/FREE
//PREVISUALISATION DU FORMATAGE SQL
EXEC SQL
Select Aaaaa,bbBb, CASE
when D>4 THEN 'E'   when F<5 THEN 'G'
else 'H' /* Commentaire SQL */
End
fFrom I  WHERE JjjJ>kKK and LLll<MmMm;
```

At the bottom of the dialog are 'OK' and 'Annuler' buttons.

LPEX – Restructuration (refactoring)

- RDi propose une fonctionnalité de refactoring de code, pour le moment uniquement pour le renommage de variables
 - Plus "intelligente" que la fonction de recherche / remplacement
 - Détection des noms définis en externe, des overflow en format fixe, des noms identiques dans les DS qualifiées et non qualifiées...
 - Aperçu des modifications et choix de modifier ou non, au cas par cas

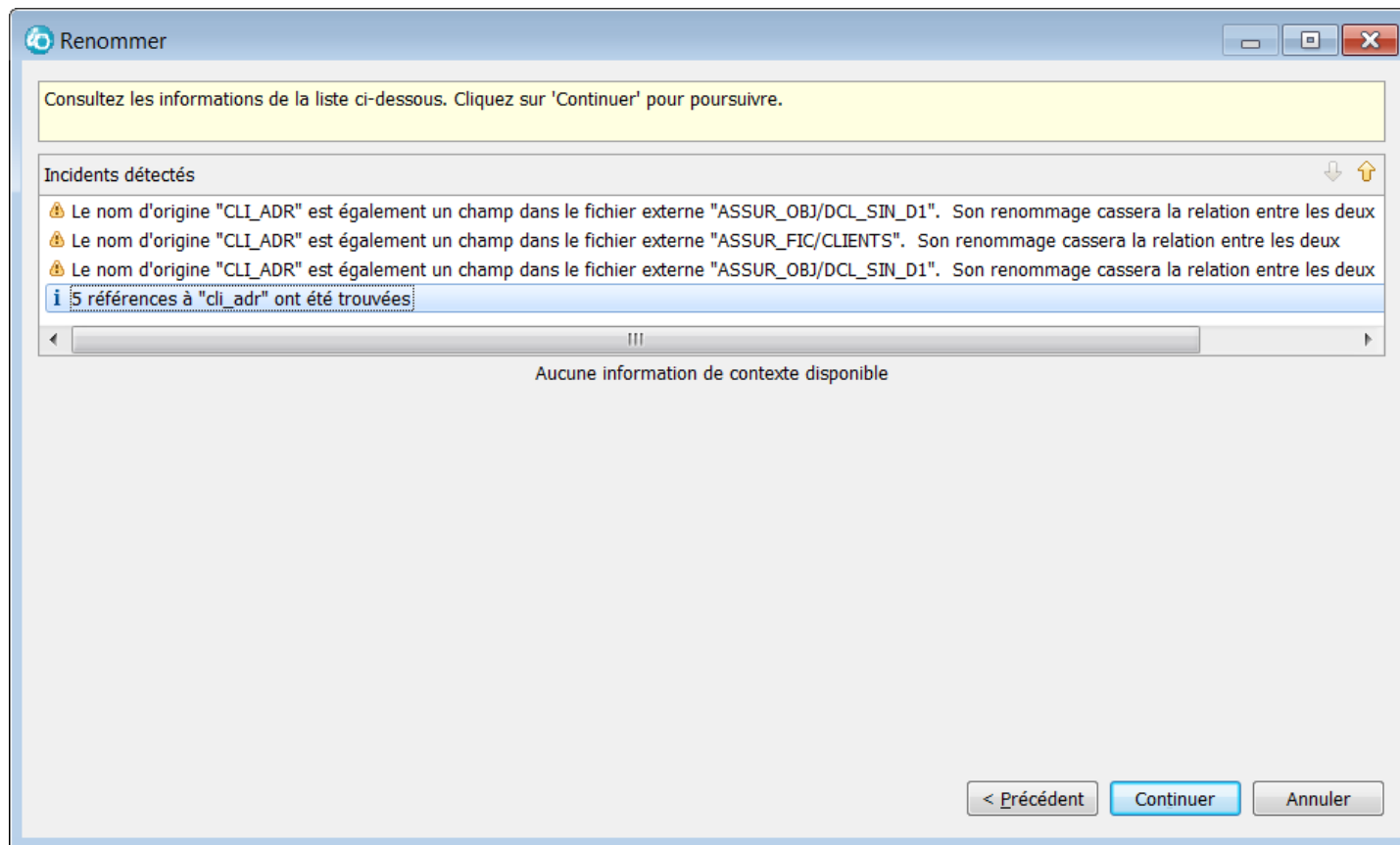


The screenshot shows the RDi IDE interface. The main editor displays a COBOL program with the following code:

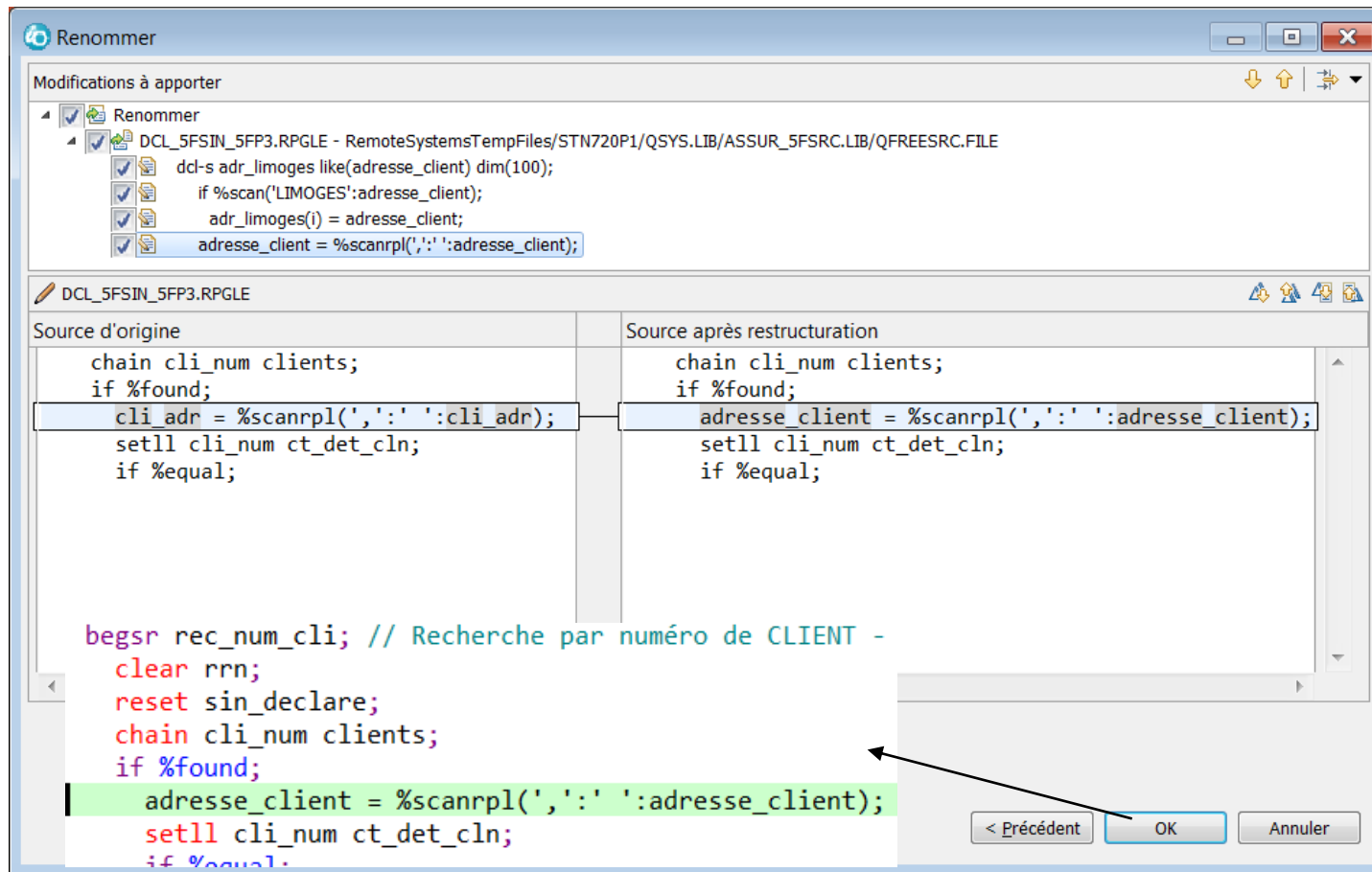
```
Ligne 100      Colonne 40      Remplacement
.....+...1.....+...2.....+...3.....+...4
009300      if %found;
009301      cli_adr = %scanrpl(',',' ':cli_adr);
009400      setll cli_num ct_det_cln;
009500      if %equal:
```

The menu path **Restructurer > Renommer...** is highlighted with a red box. A dialog box titled **Renommer** is open, showing the **Nouveau nom :** field with the value `adresse_client` and the **Forcer l'aperçu** checkbox checked. The **Aperçu >** button is highlighted.

LPEX – Restructuration (refactoring)



LPEX – Restructuration (refactoring)



Renommer

Modifications à apporter

- Renommer
 - DCL_5FSIN_5FP3.RPGLE - RemoteSystemsTempFiles/STN720P1/QSYS.LIB/ASSUR_5FSRC.LIB/QFREESRC.FILE
 - dcl-s adr_limoges like(adresse_client) dim(100);
 - if %scan('LIMOGES':adresse_client);
 - adr_limoges(i) = adresse_client;
 - adresse_client = %scanrpl(',':' ':adresse_client);

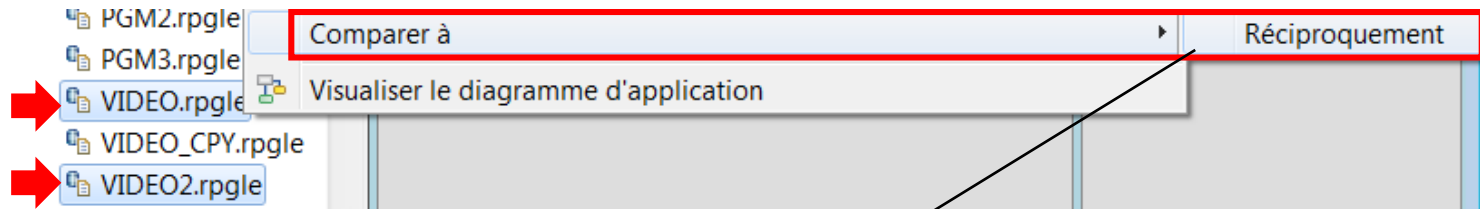
DCL_5FSIN_5FP3.RPGLE

Source d'origine	Source après restructuration
chain cli_num clients; if %found; cli_adr = %scanrpl(',':' ':cli_adr); setll cli_num ct_det_cln; if %equal;	chain cli_num clients; if %found; adresse_client = %scanrpl(',':' ':adresse_client); setll cli_num ct_det_cln; if %equal;

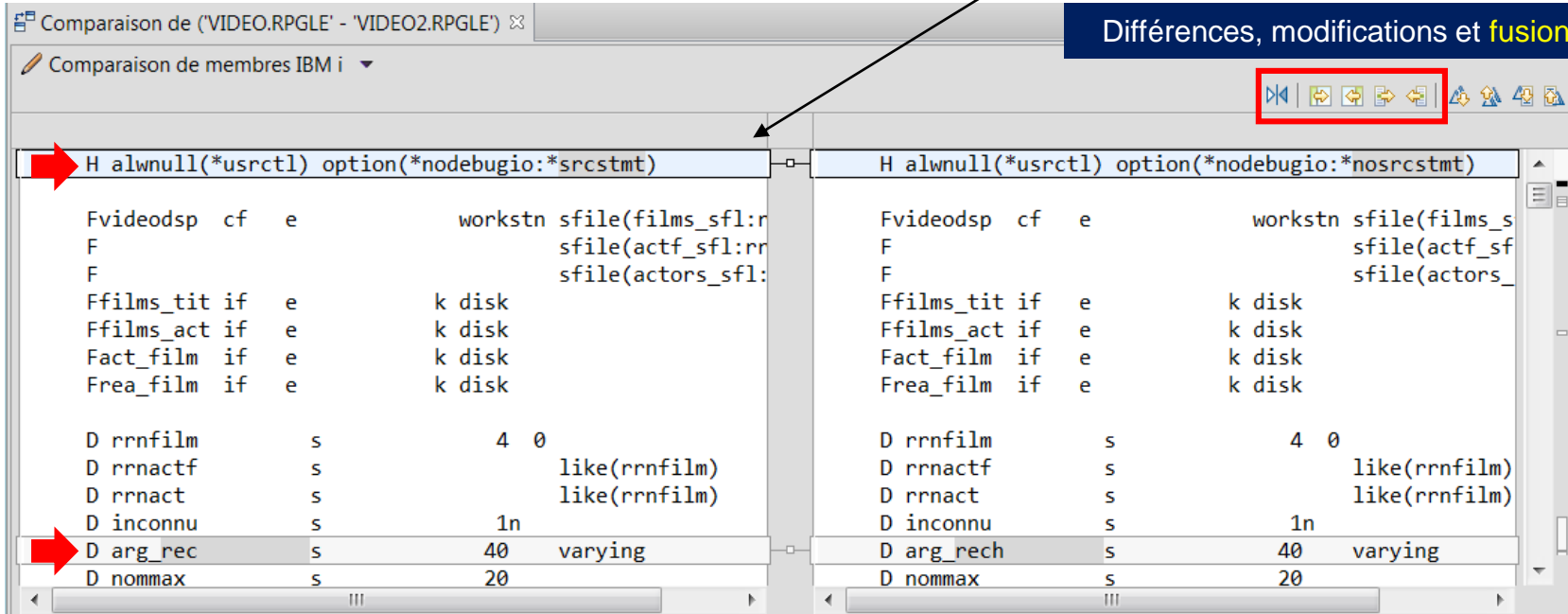
```
begsr rec_num_cli; // Recherche par numéro de CLIENT -  
clear rrr;  
reset sin_declare;  
chain cli_num clients;  
if %found;  
adresse_client = %scanrpl(',':' ':adresse_client);  
setll cli_num ct_det_cln;  
if %equal;
```

< Précédent OK Annuler

LPEX – Comparaison / fusion de sources

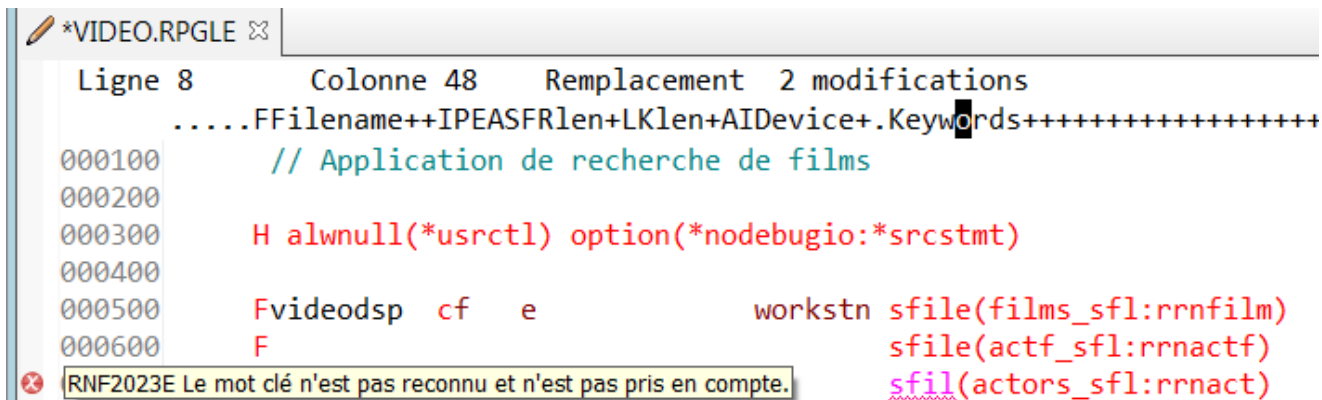
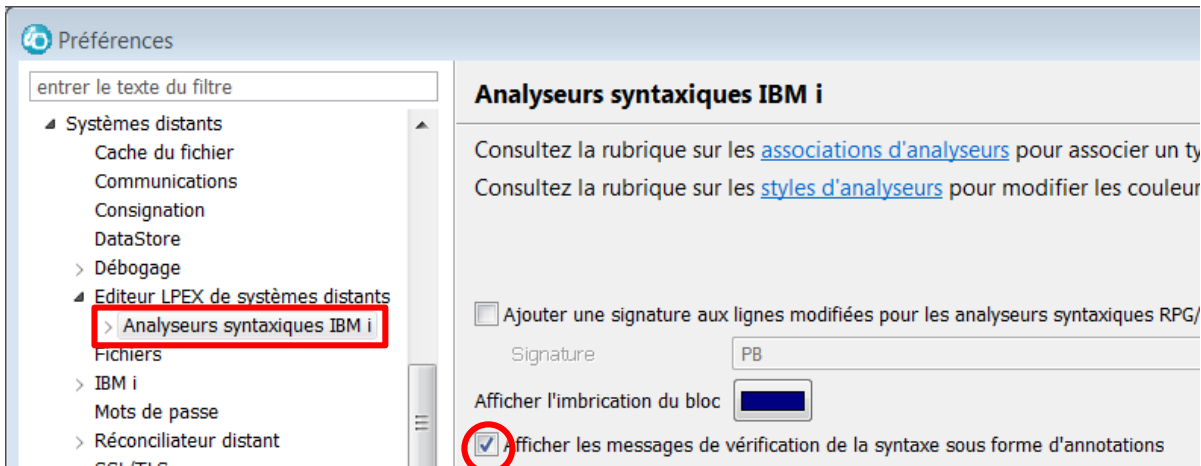


Différences, modifications et fusion



Line	VIDEO.RPGLE	VIDEO2.RPGLE
1	H alwnull(*usrctl) option(*nodebugio:*srcstmt)	H alwnull(*usrctl) option(*nodebugio:*nosrcstmt)
2	Fvideodsp cf e workstn sfile(films_sfl:r	Fvideodsp cf e workstn sfile(films_s
3	F sfile(actf_sfl:rr	F sfile(actf_sf
4	F sfile(actors_sfl:	F sfile(actors_
5	Ffilms_tit if e k disk	Ffilms_tit if e k disk
6	Ffilms_act if e k disk	Ffilms_act if e k disk
7	Fact_film if e k disk	Fact_film if e k disk
8	Frea_film if e k disk	Frea_film if e k disk
9	D rrnfilm s 4 0	D rrnfilm s 4 0
10	D rrnactf s like(rrnfilm)	D rrnactf s like(rrnfilm)
11	D rrnact s like(rrnfilm)	D rrnact s like(rrnfilm)
12	D inconnu s 1n	D inconnu s 1n
13	D arg_rec s 40 varying	D arg_rech s 40 varying
14	D nommax s 20	D nommax s 20

LPEX – Vérification de syntaxe et annotations



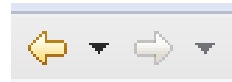
The screenshot shows a code editor window titled '*VIDEO.RPGLE'. The code is as follows:

```
Ligne 8      Colonne 48  Remplacement 2 modifications
.....FFilename++IPEASFRlen+LKlen+AIDevice+.Keywords+++++++
000100      // Application de recherche de films
000200
000300      H alwnull(*usrctl) option(*nodebugio:*srcstmt)
000400
000500      Fvideodsp  cf  e          workstn sfile(films_sfl:rrnfilm)
000600      F          sfile(actf_sfl:rrnactf)
RNF2023E Le mot clé n'est pas reconnu et n'est pas pris en compte.  sfil(actors_sfl:rrnact)
```

The error message 'RNF2023E Le mot clé n'est pas reconnu et n'est pas pris en compte.' is highlighted with a red box at the bottom of the editor. The word 'sfil' in the last line of code is underlined with a red wavy line, indicating the source of the error.

LPEX – Infobulles et hyperliens

- Affichage des déclarations en infobulle (survol souris)
- Lien hypertexte pour accéder aux déclarations
 - Déplacement par ALT → et ALT ← ou icônes



```
VIDEO.RPGLE
Ligne 44      Colonne 16  Rempla
.....+.....1.....+.....2.....+.....
004300
004400      clear chxopt;
004500      exfmt
004600      enddo;
```

Avec "CTRL + survol souris" la zone dans le source devient un lien hypertexte

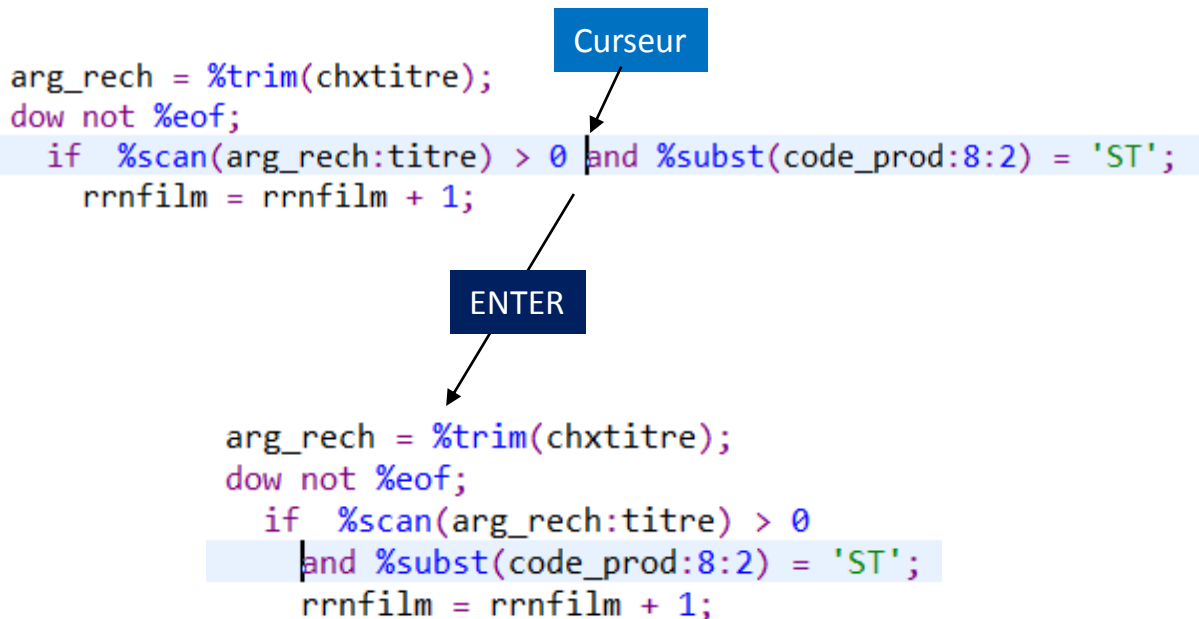
[chxopt](#) : Décimal condensé (1,0)
 Défini par:
 videodsp/fprincipal - Externe: VIDEO/VIDEODSP/FPRINCIPAL/CHXOPT

Appuyez sur 'F2' pour la mise en évidence

```
Fvideodsp  cf  e          workstn sfile(films_sfl:rrnfilm)
F                                                sfile(actf_sfl:rrnactf)
F                                                sfile(actors_sfl:rrnact)
```

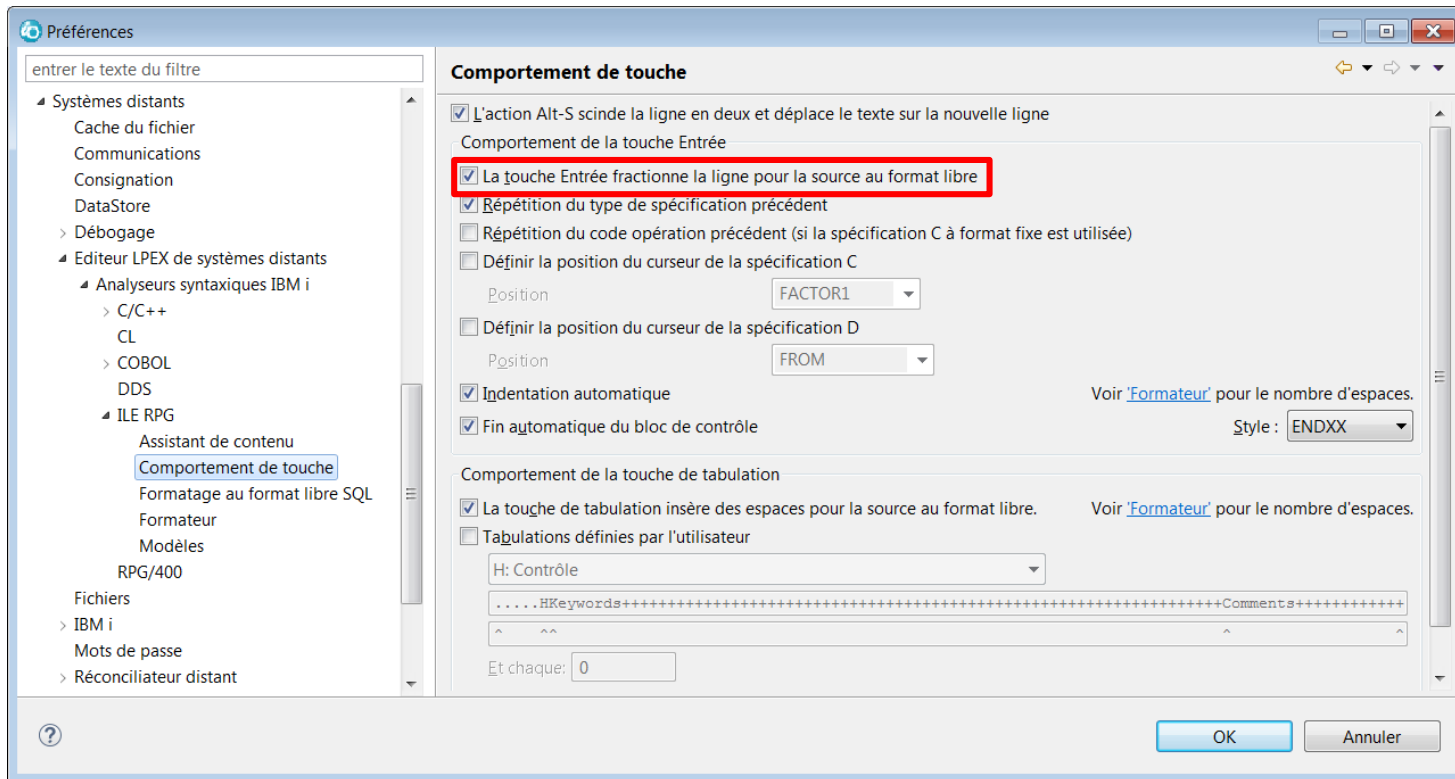
LPEX – RPG Format libre

- La touche ENTER divise la ligne en deux :



LPEX – RPG format libre

- La touche ENTER divise la ligne en deux



LPEX – RPG format libre

- La touche TAB insère des espaces

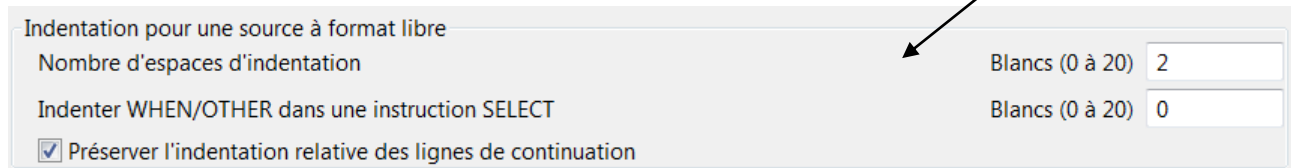
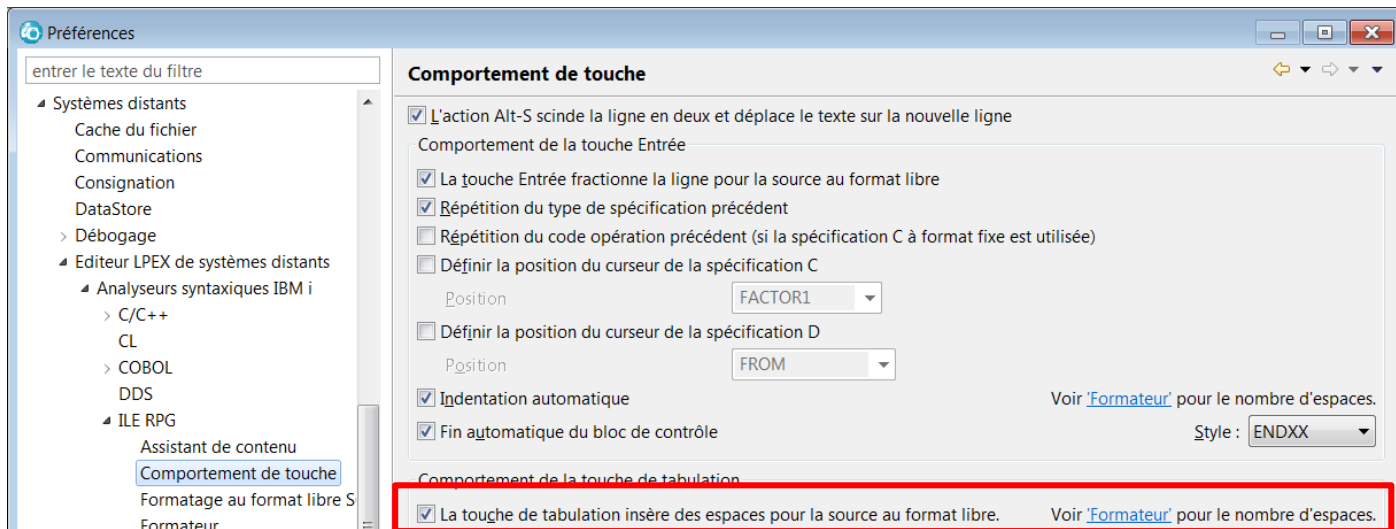
```
000169 // --> affichage
000170 write films_top;
000171 write films_bot;
000172 rang = 1;
000173 *in32 = *on;
000174 *in34 = *on;
000175 exfmt films_ctl;
```

Ici on a fait 2 fois TAB

```
000169 // --> affichage c
000170 write films_top;
000171 write films_bot;
000172 rang = 1;
000173 *in32 = *on;
000174 *in34 = *on;
000175 exfmt films_ctl;
```

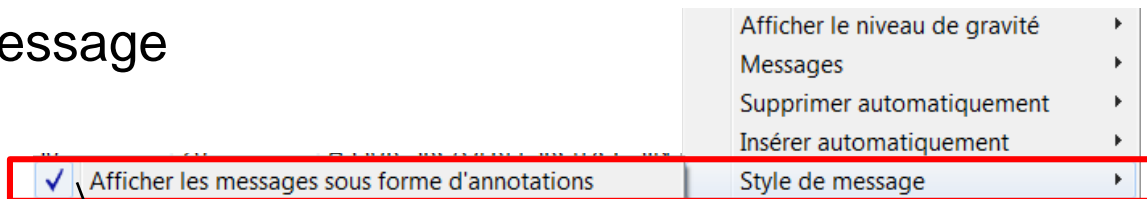
LPEX – RPG format libre

- La touche TAB insère des espaces

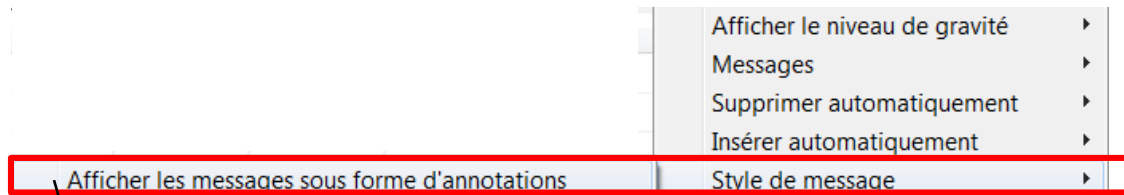


Résultat de la vérification / compilation

- Style de message



```
014500 reset sin_declare;  
014600 clear enquete_sin;  
014700 clear *in60;
```



```
014500 reset sin_declare;  
014600 clear enquete_sin;  
RNF7503: L'expression contient un opérande non défini.  
014700 clear *in60;
```

Couverture de code (code coverage)

- Fonctionnalité qui permet d'analyser et de visualiser la couverture des programmes (**quelles procédures et quelles lignes ont été exécutées**) afin de déterminer l'efficacité des **tests**, de détecter le code mort, etc.
- Valable pour tout programme ou programme de service IBM i qui peut être débogué, quel que soit son langage
- Peut être lancée par RDi ou par une commande CL
- Les résultats sont visualisables dans RDi ou sous forme HTML / PDF
- Les résultats peuvent être :
 - *Cumulés* pour voir la couverture d'une suite totale de tests
 - *Comparés* pour visualiser le % d'amélioration
 - *Fusionnés* pour ne visualiser que les différences

Couverture de code (code coverage)

- Mise en œuvre par l'ajout d'un point d'entrée de service

Puis choisir le mode "Couverture de code"

Bibliothèque	Programme	Type de progr...	Module	Procédure	ID utilisateur	Connexion	Activé
SIN_PB	DCL_SIN_P1	*PGM	*ALL	*ALL	BOURGEOIS	STN720P1	Oui

OU

Bibliothèque	Programme	Type de progr...	Module	Proc
SIN_PB	DCL_SIN_P1	*PGM	*ALL	*ALL

- Définir
- Modifier
- Ouvrir la configuration de lancement
- Supprimer
- Activer
- Désactiver
- Actualiser
- Mode débogage
- Mode couverture de code**
- Sélectionner tout

Couverture de code (code coverage)

Elément	Couverture	Couvert	Total
▾ *PGM ASSUR_SRC/DCL_SIN_P1	40 %	77	191
▾ DCL_SIN_P1	40 %	77	191
▾ DCL_SIN_P1.RPGLE	40 %	77	191
● DCL_SIN_P10	40 %	77	191

En vert : les lignes qui ont été exécutées
En rouge : les lignes qui n'ont pas été exécutées

Différences après 2 exécutions

Ligne 99 Colonne 24 Remplacement
.....1.....2...+.....3.....4.

```
chain cont_num ct_det_ctn;  
if %found;  
chain cli_num clients;  
write det_ct_fnn;  
exfmt det_cont;  
dow not *in12;  
if *in06;  
exsr decl_sin;  
if sin_declare = '0';  
leave;  
endif;  
endif;
```

```
chain cont_num ct_det_ctn;  
if %found;  
chain cli_num clients;  
write det_ct_fnn;  
exfmt det_cont;  
dow not *in12;  
if *in06;  
exsr decl_sin;  
if sin_declare = '0';  
leave;  
endif;  
endif;  
write det_ct_fnn;  
exfmt det_cont;  
enddo;  
else;  
*in90 = *on;  
endif;  
endsr;
```

Couverture de code en mode commande

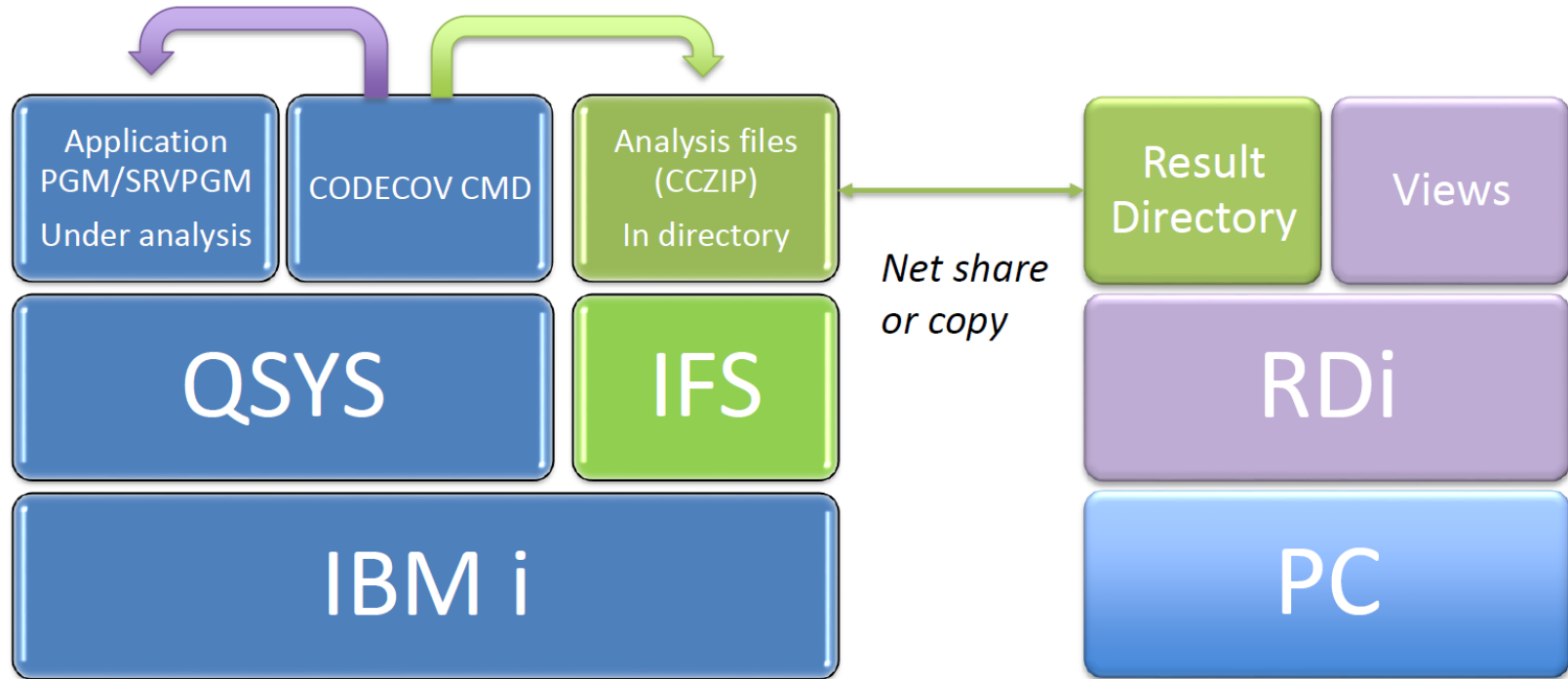


- Pour intégration du lancement de la couverture de code dans les scripts de test

- Nouvelle commande CL : CODECOV

- Prérequis :
 - PTFs sur le produit 5770-WDS Option 60 :
 - IBM i 7.2 : SI65228 et SI64561
 - IBM i 7.3 : SI65229 et SI64670
 - QDEVTOOLS dans la LIBL
 - Une bibliothèque qui porte le nom de votre profil utilisateur
 - RDi (ne sera nécessaire que pour visualiser le résultat)

Couverture de code en mode commande



Couverture de code en mode commande

```
Code coverage (CODECOV)

Indiquez vos choix, puis appuyez sur ENTREE.

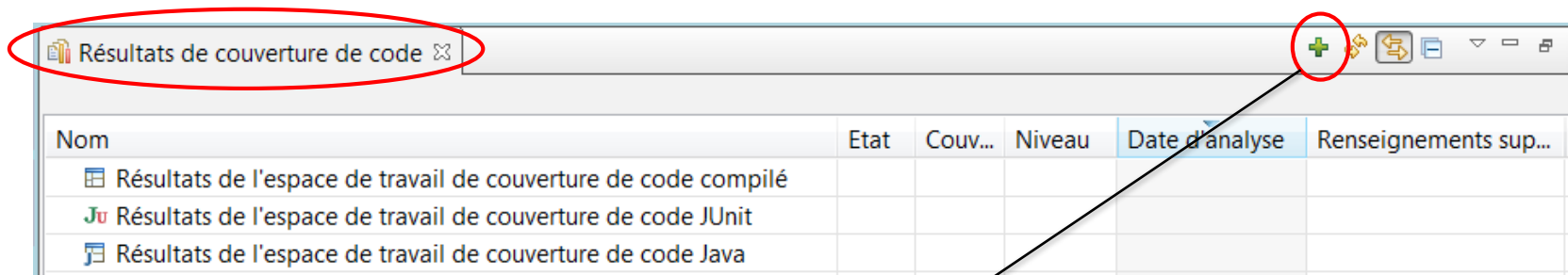
Output directory . . . . . '/pb/code_coverage'
-----
Output stream file . . . . .
-----
Code coverage test identifier .
-----
```

Le fichier généré se nommera "Nom_du_programme_Horodate.czip"

```
Message . . . . : Le fichier de résultats de couverture de code
                  (/pb/code_coverage/DCL_SIN_P1_1523997551.czip) a été créé.
```

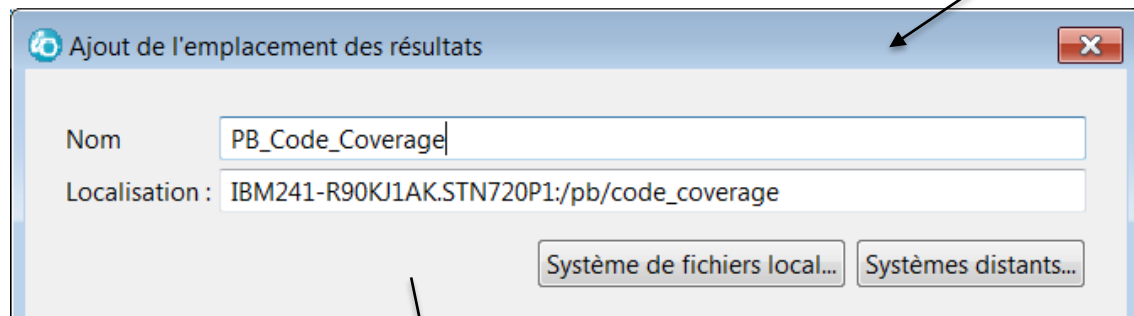
Couverture de code en mode commande

- Dans RDi :



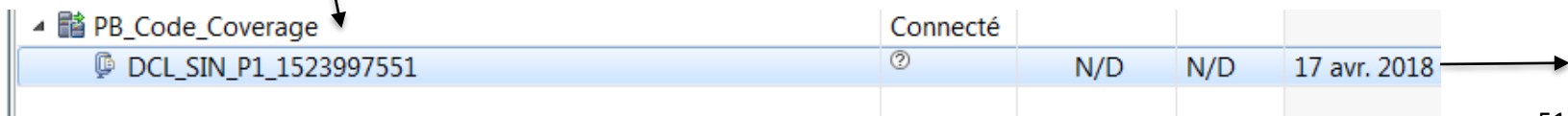
The screenshot shows the RDi interface with a tab titled "Résultats de couverture de code" circled in red. A green plus sign icon in the top right corner of the tab area is also circled in red, with an arrow pointing to a dialog box below. The table below the tab has the following structure:

Nom	Etat	Couv...	Niveau	Date d'analyse	Renseignements sup...
Résultats de l'espace de travail de couverture de code compilé					
Résultats de l'espace de travail de couverture de code JUnit					
Résultats de l'espace de travail de couverture de code Java					



The dialog box "Ajout de l'emplacement des résultats" contains the following fields and buttons:

- Nom: PB_Code_Coverage
- Localisation: IBM241-R90KJ1AK.STN720P1:/pb/code_coverage
- Buttons: "Système de fichiers local...", "Systèmes distants..."




The screenshot shows a table with a highlighted row. An arrow from the dialog box above points to the highlighted row. Another arrow points from the date "17 avr. 2018" to the right.

PB_Code_Coverage	Connecté			
DCL_SIN_P1_1523997551	?	N/D	N/D	17 avr. 2018

Couverture de code en mode commande

- Résultat :

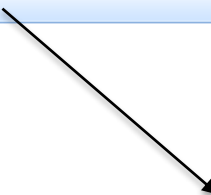
 **Résultats de la couverture de code niveau fichier**

Cette page affiche les résultats de la couverture de code au niveau fichier pour tous les résultats analysés.

Généré : 17 avril 2018 22:53:48 CEST

Résultats

Nom	Couverture	Lignes couve...	Lignes non c...	Total des lig...
QRPGLSRC/OL_BASE.file/DCL_SIN_P1.rpgle	75 %	140	46	186
• DCL_SIN_P1	75 %	140	46	186

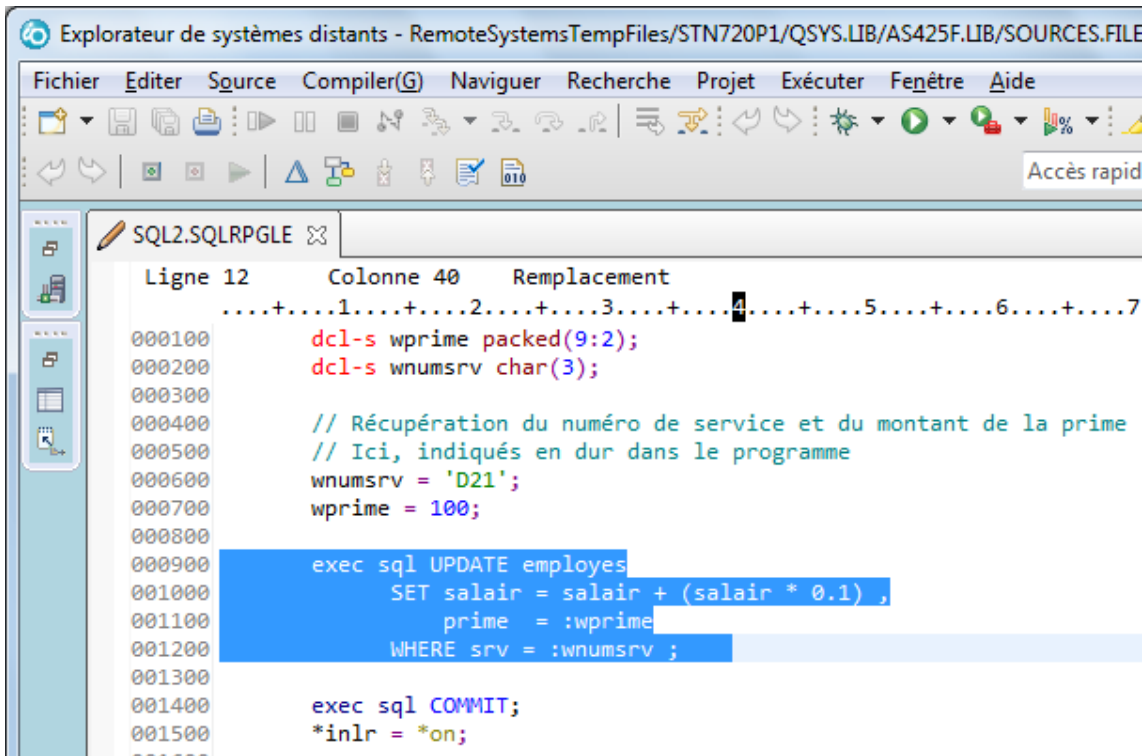


```
C      when      *in04
C      if        zone='SIN_TYP_L'
C      exsr     choix_typ_sin
C      eval     sin_typ_n = sin_typ_n_sav
C      endif
C      other
C      select
C      when     sin_typ_1 = *blanks
C      eval     *in70=*on
C      when     sin_desc = *blanks
C      eval     *in71=*on
C      other
```

Rappel : ACS (Access Client Solutions)

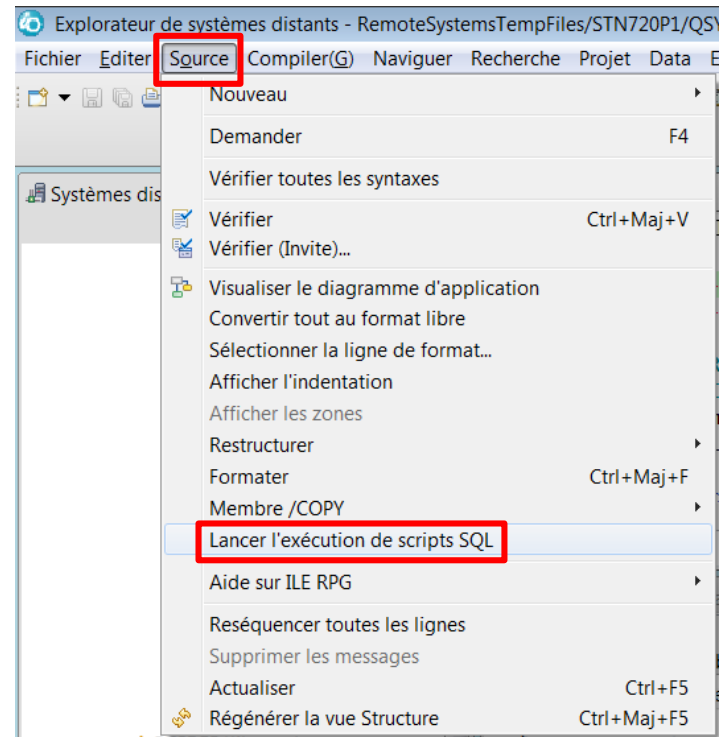
- Successeur de Client Access (IBM i Access for Windows)
- Utilise les licences de XW1 (IBM i Access Family)
- Téléchargeable depuis le site IBM
 - www.ibm.com/systems/power/software/i/access/solutions.html
- 100% Java
- Ne nécessite pas forcément d'installation
- Emulateur 5250, transfert de données, accès IFS, accès aux consoles LAN et HMC, terminal SSH, OUTQ et spoules (conversion en PDF)
- Gestion de la base de données : exécution de scripts SQL, gestion des schémas, centre de performances SQL
- La version actuelle est la 1.1.7.3

RDi – Intégration avec ACS

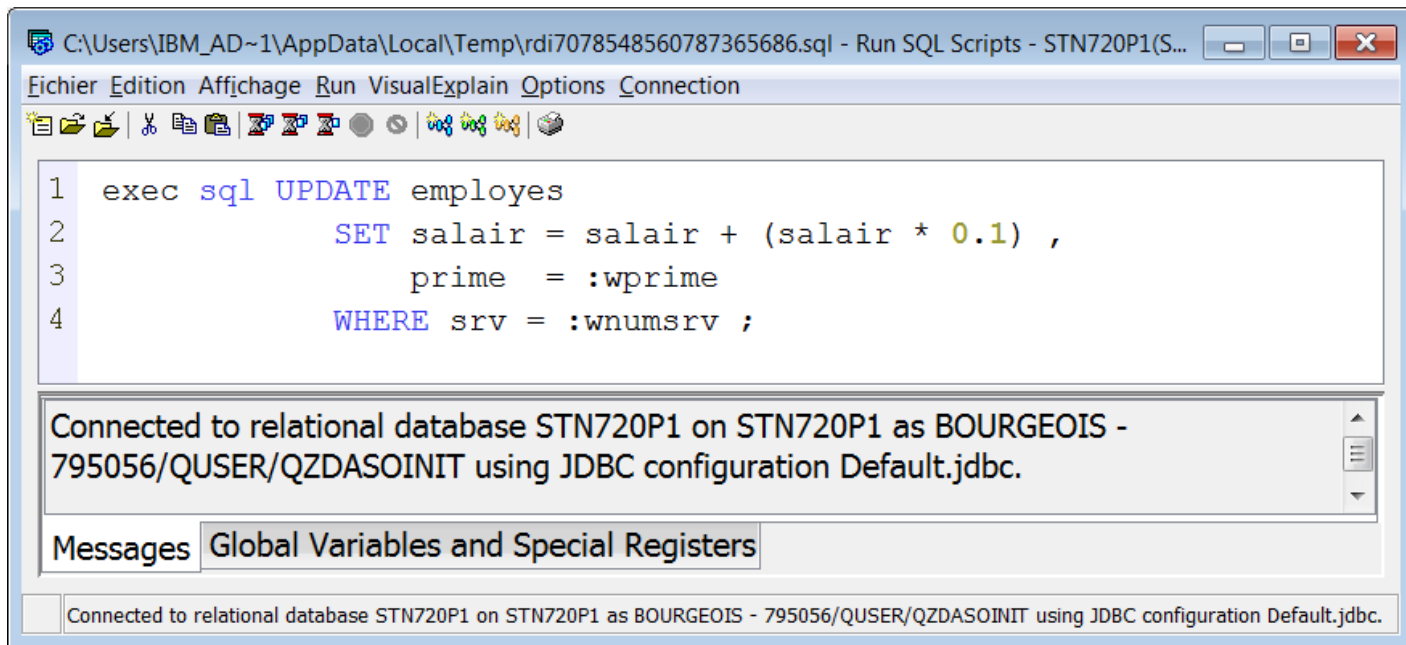


```
Explorateur de systèmes distants - RemoteSystemsTempFiles/STN720P1/QSYS.LIB/AS425F.LIB/SOURCES.FILE
Fichier Éditer Source Compiler(G) Naviguer Recherche Projet Exécuter Fenêtre Aide
SQL2.SQLRPGLE
Ligne 12      Colonne 40      Remplacement
.....1.....2.....3.....4.....5.....6.....7
000100      dcl-s wprime packed(9:2);
000200      dcl-s wnumsrv char(3);
000300
000400      // Récupération du numéro de service et du montant de la prime
000500      // Ici, indiqués en dur dans le programme
000600      wnumsrv = 'D21';
000700      wprime = 100;
000800
000900      exec sql UPDATE employes
001000          SET salaire = salaire + (salaire * 0.1) ,
001100          prime = :wprime
001200          WHERE srv = :wnumsrv ;
001300
001400      exec sql COMMIT;
001500      *inlr = *on;
```

Lancement de l'exécution de scripts SQL



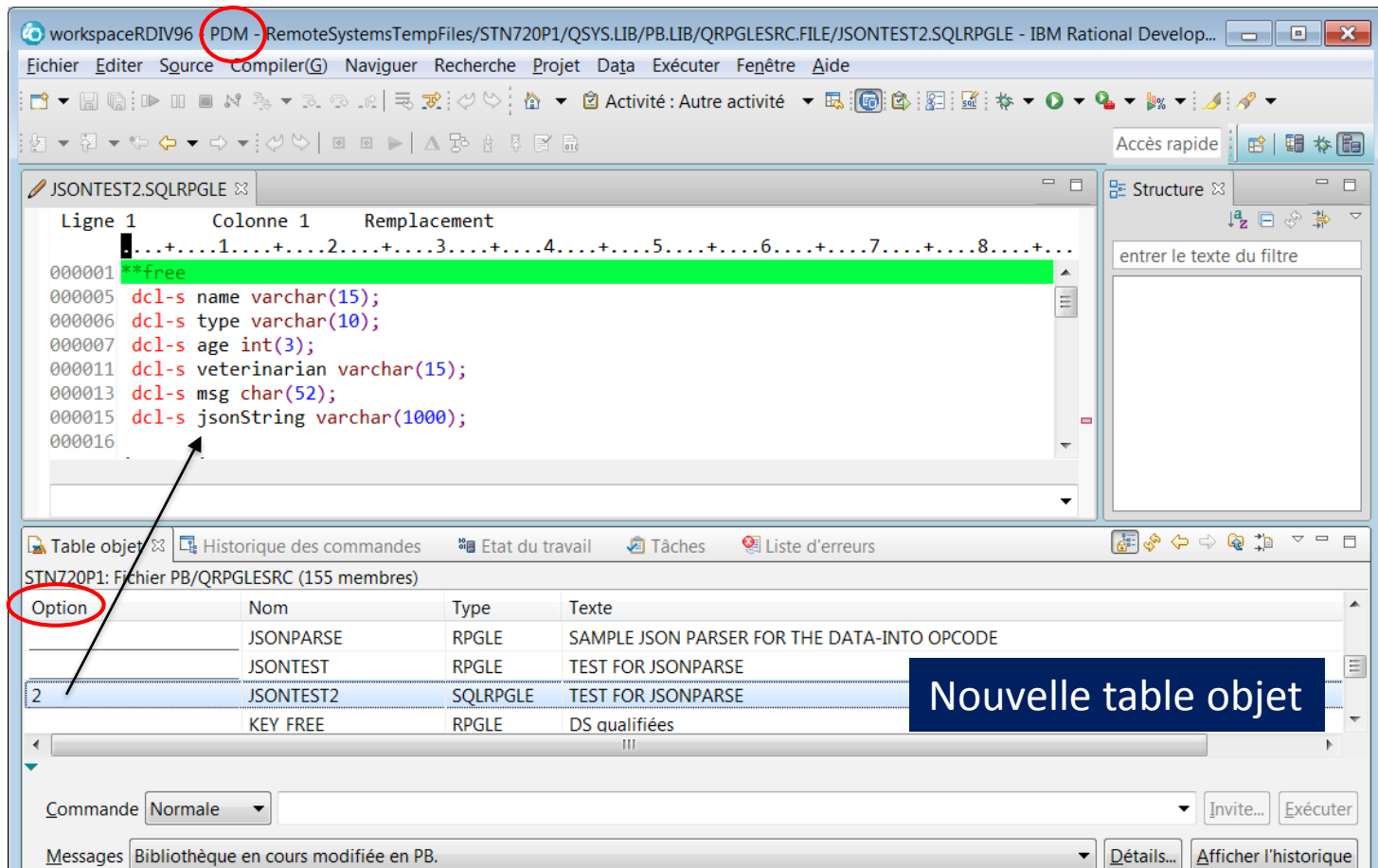
RDi – Intégration avec ACS

A screenshot of a SQL script editor window. The title bar shows the file path: C:\Users\IBM_AD~1\AppData\Local\Temp\rdi7078548560787365686.sql - Run SQL Scripts - STN720P1(S... The menu bar includes 'Fichier', 'Edition', 'Affichage', 'Run', 'Visual Explain', 'Options', and 'Connection'. The toolbar contains icons for file operations and execution. The main text area contains the following SQL code:

```
1  exec sql UPDATE employes
2      SET salaire = salaire + (salaire * 0.1) ,
3          prime = :wprime
4      WHERE srv = :wnumsrv ;
```

Below the code, a status bar indicates: 'Connected to relational database STN720P1 on STN720P1 as BOURGEOIS - 795056/QUSER/QZDASOINIT using JDBC configuration Default.jdbc.' At the bottom, there are tabs for 'Messages' and 'Global Variables and Special Registers', with the status bar text: 'Connected to relational database STN720P1 on STN720P1 as BOURGEOIS - 795056/QUSER/QZDASOINIT using JDBC configuration Default.jdbc.'

RDi 9.6.0.2 – Perspective PDM

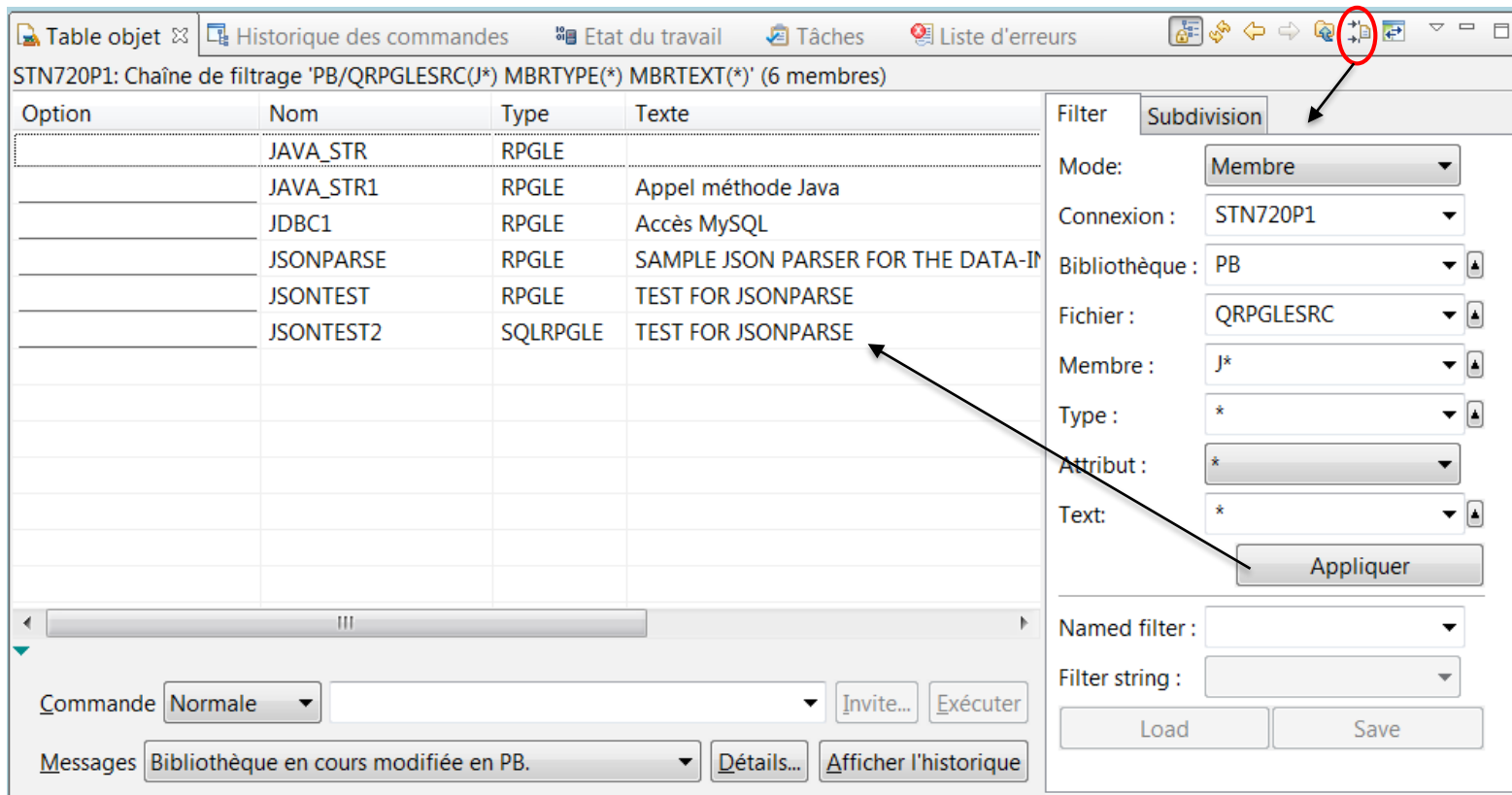


The screenshot displays the IBM Rational Developer for i (RDi) interface in the Perspective PDM. The top window shows the source code for the file `JSONTEST2.SQLRPGLE`. The code includes several `dcl-s` declarations for variables like `name`, `type`, `age`, `veterinarian`, `msg`, and `jsonString`. A red circle highlights the `PDM` tab in the title bar.

The bottom window shows the `Table objet` (Table Object) list for the project `STN720P1`. The `Option` column is circled in red. The table lists several objects, with `JSONTEST2` selected. A blue callout box with the text `Nouvelle table objet` (New table object) points to the `JSONTEST2` entry.

Option	Nom	Type	Texte
	JSONPARSE	RPGLE	SAMPLE JSON PARSER FOR THE DATA-INTO OPCODE
	JSONTEST	RPGLE	TEST FOR JSONPARSE
2	JSONTEST2	SQLRPGLE	TEST FOR JSONPARSE
	KEY FREE	RPGLE	DS qualifiées

RDi 9.6.0.2 – Filtrage rapide



STN720P1: Chaîne de filtrage 'PB/QRPGLESRC(J*) MBRTYPE(*) MBRTTEXT(*)' (6 membres)

Option	Nom	Type	Texte
	JAVA_STR	RPGLE	
	JAVA_STR1	RPGLE	Appel méthode Java
	JDBC1	RPGLE	Accès MySQL
	JSONPARSE	RPGLE	SAMPLE JSON PARSER FOR THE DATA-IN
	JSOATEST	RPGLE	TEST FOR JSONPARSE
	JSOATEST2	SQLRPGLE	TEST FOR JSONPARSE

Filter Subdivision

Mode: Membre

Connexion : STN720P1

Bibliothèque : PB

Fichier : QRPGLESRC

Membre : J*

Type : *

Attribut : *

Text: *

Appliquer

Named filter:

Filter string:

Load Save

Commande Normale

Invite... Exécuter

Messages Bibliothèque en cours modifiée en PB. Détails... Afficher l'historique

W E R C

The image features the letters 'W', 'E', 'R', and 'C' in a large, white, sans-serif font. Each letter is filled with a different photograph of a diverse group of business professionals. The 'W' shows a woman with long dark hair in a green top. The 'E' shows a man with a mustache in a green patterned shirt. The 'R' shows a woman with her hands clasped in a light blue top. The 'C' shows a man in a blue suit and yellow tie. To the right of the 'C' is a vertical strip showing a man with glasses in a blue suit. The letters have a slight drop shadow.